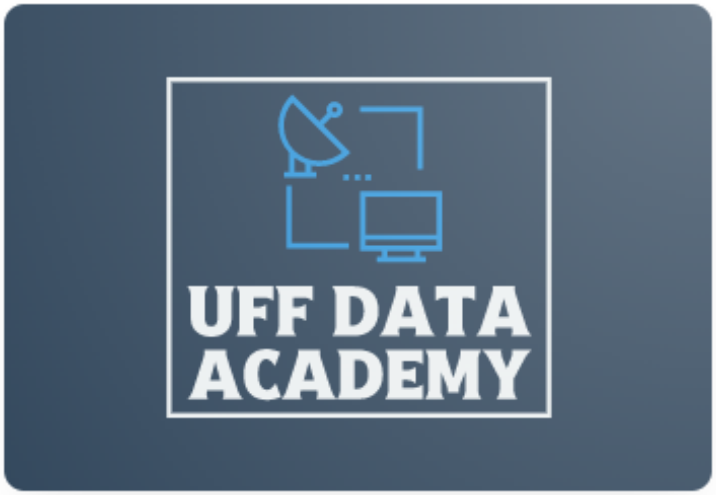
Universidade Federal Fluminense



Visualização, Representação e Análise de Dados

Professor Cledson de Sousa

Versão: 1.5.515 – 18 de agosto de 2025.

Conteúdo

PREFÁCIO

Bem-vindos ao curso de Visualização e Representação e Análise de Dados, uma disciplina criada para introduzir os conceitos fundamentais e práticas avançadas na análise e interpretação visual de dados, com um foco especial em aplicações na Engenharia de Telecomunicações. Este curso foi estruturado para fugir do puro aprendizado de fórmulas, para uma abordagem prática com exemplos e contrastes do que é correto e do que é enviesado nas análises dos parâmetros estatísticos necessários para garantir o rigor, fiabilidade e sanidade das representação dos resultados obtidos.

O objetivo é proporcionar aos alunos uma compreensão abrangente das técnicas de visualização de dados, desde a preparação e refinamento dos dados até a criação de representações visuais eficazes que facilitam a interpretação e a comunicação dos resultados. A visualização de dados é uma habilidade essencial em pesquisa científica e especificamente em engenharia, onde a capacidade de analisar grandes volumes de dados e extrair *insights* relevantes pode levar a soluções inovadoras e avanços tecnológicos.

Esta apostila, embora tenha um caráter acadêmico, não possui a pretensão de ser extremamente rigorosa, especialmente na forma. O autor empenhou-se em dar o devido crédito a todas as fontes utilizadas; no entanto, por vezes, estende-se o texto sem as devidas citações específicas dos autores originais acolhidos nas notas. O texto está coalhado de gráficos, figuras, exemplos, contra-exemplos e análises. O código de **todos** os gráficos, exercícios e listagens aqui apresentados estão disponíveis e comentados no github ou em minha página pessoal.

Ao longo do texto, há diversos sinais de parada 🛑, são as notas de margem, gráficos, figuras e outros sinais. Estes sinais estão lá para o leitor descontrair. Naturalmente, deve-se retornar ao texto tão logo possível. Essa estratégia funciona comigo!

Desejo que este curso seja leve, produtivo, enriquecedor!

Sobre o autor:

O autor obteve o título de Doutor em Computação pela Universidade Federal Fluminense (UFF) em 2019, bem como os títulos de graduação e de mestre em Engenharia de Telecomunicações pela mesma instituição, em 1997 e 2013, respectivamente. Com mais de 30 anos de experiência na indústria de telecomunicações, atua desde 2021 como Professor Adjunto no Departamento de Engenharia de Telecomunicações da Universidade Federal Fluminense. Seus atuais interesses de pesquisa incluem redes de sensores sem fio, SDN, rádios cognitivos e CSI.

Plataformas:

- `https://www.linkedin.com/in/cledsonsousa`
- `https://cledsonsousa.github.io`
- `http://lattes.cnpq.br/7195080748145566`
- cledsons@id.uff.br

1 Introdução

A evolução da tecnologia resultou em um aumento substancial na quantidade de dados, exigindo métodos eficientes de extração, medição e quantificação. A análise estatística desempenha um papel crucial ao representar dados e discernir padrões e tendências, fundamentais para entender e expressar complexidades dos resultados experimentais[1]

A habilidade de transformar dados complexos em representações visuais claras e compreensíveis é essencial na engenharia. Uma boa técnica de apresentação dos dados facilita a identificação de padrões, tendências e anomalias que podem não ser aparentes em uma análise puramente numérica. Essa habilidade é fundamental para a análise de dados experimentais e simulações, bem como para a comunicação eficaz dos resultados de pesquisa para diferentes públicos.

Na pesquisa científica, a habilidade de analisar e apresentar dados quantitativos de forma rigorosa é central. Uma análise detalhada permite a comunicação clara e eficaz dos resultados, facilitando a compreensão e a replicação dos estudos por outros pesquisadores. Além disso, a análise estatística é essencial para identificar contribuições relevantes, separando-as de resultados triviais.

A visualização de dados combina arte e ciência. Uma boa visualização deve ser esteticamente agradável e cientificamente precisa, transmitindo os dados de maneira clara e sem induzir a erros ou distorções. É importante que a visualização respeite a proporcionalidade: se um número é o dobro do outro, isso deve estar claramente representado.

Além de precisa, a apresentação de dados deve ser esteticamente agradável para reforçar a mensagem. Elementos distrativos, como cores fortes ou desbalanceadas, devem ser evitados, pois podem dificultar a interpretação correta dos dados ou desviar a atenção.

[1]Esta apostila é uma consolidação de minhas leituras. Alguns textos abordam melhor alguns tópicos que outros. E o Capítulo 2 é fortemente baseado nos trabalhos de [1] e [2]. Figuras em preto e branco são do segundo e as coloridas do primeiro. Pessoalmente ficam meus agradecimentos e os merecidos créditos pelo excelente texto dos autores. Os autores estão aí embaixo, nas Figuras 1.0.1, 1.0.2 e 3.0.1.

Figura 1.0.1: Elena A. Allen possui vasta experiência em análise de dados interdisciplinares. Atualmente, trabalha como Consultora na Rodin Scientific, LLC, foi Cientista Chefe na Rodin Scientific, Cientista Sênior na Medici *Technologies* e bolsista de pós-doutorado na Universidade de Bergen e na *The Mind Research Network*.

Figura 1.0.2: Erik Barry Erhardt, PhD, é Professor Associado de Estatística no Departamento de Matemática e Estatística da Universidade do Novo México, onde atuou como Diretor da Clínica de Consultoria em Estatística.Onde atualmente (2024) é Diretor do Núcleo de Bioestatística e Neuroinformática.

2 Boas Práticas e Rigor na Análise de Dados

A crescente quantificação de dados nas pesquisas científicas e no desenvolvimento tecnológico exige métodos eficazes de análise e interpretação. A análise estatística, por sua vez, torna-se indispensável para lidar com a grande quantidade de dados gerados. Ela permite identificar padrões, tendências e relações significativas que podem não ser percebidas por meio de uma simples observação. Além disso, a centralidade dos dados na pesquisa científica moderna é inegável: a habilidade de coletar e analisar dados de forma rigorosa é crucial para a comunicação e validação dos resultados obtidos.

Ao se discutir a visualização de dados, é essencial entender que ela combina arte e ciência. A visualização de dados (DataVis) não é apenas uma representação gráfica dos dados, mas uma ferramenta poderosa que pode influenciar a precisão e a clareza das informações transmitidas e a avaliação do leitor. Um dos maiores desafios na visualização é manter a precisão dos dados, evitando introduzir erros ou distorções que possam comprometer a análise. Além disso, é fundamental que a visualização seja proporcional: quando um elemento gráfico é duas vezes maior que outro, deve representar um valor duas vezes maior, mantendo a integridade dos dados.

Por outro lado, a estética não pode ser negligenciada. A visualização de dados deve ser agradável aos olhos, facilitando a compreensão da mensagem sem sobrecarregar o observador com elementos desnecessários. É vital evitar distrações que possam desviar a atenção do conteúdo principal, garantindo que os elementos visuais destacados contribuam para a interpretação correta dos dados.

Entretanto, é necessário ter cautela com as armadilhas da visualização de dados. Um gráfico deve considerar as capacidades perceptivas e cognitivas do leitor, aproveitando os pontos fortes das habilidades humanas de processamento de informações e minimizando os pontos fracos. Em Visualização de dados, as informações são frequentemente representadas por objetos geométricos, onde dados quantitativos ou categóricos são mapeados para atributos visuais como posição, tamanho e forma. Dessa forma, a visualização de dados não apenas complementa a análise estatística, mas também desempenha um papel crucial na comunicação eficaz dos resultados em pesquisas na área de engenharia.

2.1 CONTEXTUALIZAÇÃO METODOLÓGICA

A visualização de dados não é uma etapa isolada, mas parte integrante de um fluxo estruturado que vai da definição do objetivo à compilação resultados. Se bem planejado, esse *pipeline* garante que as decisões tomadas em cada fase estejam alinhadas com a pergunta central a ser respondida[2].

Nas subseções seguintes, descrevemos as etapas essenciais desse processo, ressaltando o papel da visualização tanto na exploração preliminar quanto na comunicação precisa e reprodutível das conclusões.

[2] Curiosamente, nem sempre o pipeline segue a ordem preconizada pelos manuais. Em projetos experimentais, ou quando usamos um *dataset* de terceiros, como faremos miutas vezes nesta disciplina, o "resultado" costuma aparecer antes de todo o resto e a partir dele, vamos montar nossos casos de uso. E, sim, no mundo real isso também acontece: há pesquisas em que os achados moldam a narrativa e a própria apresentação, e não o contrário. Os puristas vão torcer o nariz, mas a vida nem sempre é linear. 🙏

2.1.1 Definição do objetivo e aquisição dos dados

Em projetos de Engenharia, a definição do objetivo costuma cair em alguns cenários.

No primeiro, sabe-se exatamente o que se quer confirmar/melhorar ou medir. Por exemplo, ao implementar um esquema de fusão de dados provenientes de múltiplos sensores,como CSI de redes Wi-Fi combinado com imagens térmicas, o objetivo pode ser verificar se a integração das duas fontes aumenta a acurácia na classificação do fenômeno. Nesse caso, o planejamento da aquisição de dados envolve sincronizar as medições dos dois sistemas, garantir alinhamento temporal e espacial, e capturar variáveis de apoio (temperatura ambiente, intensidade de sinal, distância entre sensores) para controle da variabilidade do experimento. Nesse caso A coleta é pensada para gerar evidências quantitativas que nos permitam dizer algo sobre o ganho/perda de desempenho da fusão em relação ao uso isolado de cada sensor.

Em um segundo, parte-se de uma medida ou variável de interesse, mas sem uma hipótese claramente definida, e a expectativa é exploratória. O objetivo é investigar se determinadas grandezas apresentam correlação ou associação com outras e, assim, possam ajudar a explicar um comportamento ainda não compreendido. Não se sabe de antemão se tal padrão existe ou se será estatisticamente relevante, a análise pode confirmar uma intuição inicial, apontar ausência de relação significativa ou ainda revelar a influência de variáveis de confusão que distorcem a interpretação.

Há ainda casos híbridos, nos quais há simultaneamente uma hipótese inicial e a abertura para descobertas não previstas. Nessas situações, parte-se de uma expectativa, por exemplo, que duas métricas de desempenho estejam associadas, mas a análise também deve estar atenta a padrões secundários que possam surgir. Esse tipo de abordagem é comum quando se trabalha com conjuntos de dados, onde, além de confirmar a hipótese principal, podem emergir relações inesperadas entre variáveis, sugerindo novas linhas de investigação ou ajustes no desenho experimental.

E entre todos os outros possíveis, há ainda o caso em que pouco se sabe sobre o fenômeno em análise e a única opção é tatear nos resultados em busca de indícios. Nessas circunstâncias, o trabalho é essencialmente exploratório: examinam-se distribuições, testam-se múltiplas formas de visualização e cruzam-se variáveis de maneiras diversas até que surjam padrões dignos de investigação mais aprofundada. Veremos muitas ferramentas que ajudam o iniciante a examinar e correlacioar as grandezas.

Esse processo, embora mais demorado e sujeito a ajustes sucessivos, é muitas vezes o único caminho para abrir terreno em áreas pouco estudadas ou com dados inéditos. No entanto, podemos afirmar, independentemente do cenário, a aquisição de dados deve garantir:

- utilização de fontes confiáveis;

- registro completo de metadados relevantes (topologia, configurações, condições ambientais, período e contexto da coleta);
- definição de granularidade, tamanho de amostra e duração compatíveis com o fenômeno investigado;
- garantia de reprodutibilidade por meio de procedimentos padronizados e aceitos na comunidade.

Esses cuidados são fundamentais para que a etapa de visualização não seja apenas estética, mas também analiticamente válida e reprodutível [3, 4].

2.1.2 Preparação e avaliação da qualidade dos dados

Após a aquisição, os dados precisam ser preparados e avaliados quanto à sua qualidade antes de qualquer análise ou visualização. Essa etapa garante que os resultados não sejam distorcidos por inconsistências, lacunas ou erros sistemáticos introduzidos durante a coleta. A preparação inclui:

- **Limpeza e tratamento de valores anômalos:** remoção de registros corrompidos, imputação ou exclusão de valores ausentes e tratamento de *outliers* quando estes forem resultado de falhas de medição.
- **Padronização:** unificação de formatos, unidades e escalas para permitir comparação direta entre fontes distintas. *Exemplo:* converter/corrigir/ sincronizar tempos de atraso e unidades.
- **Integração de múltiplas fontes:** fusão de dados oriundos de plataformas ou sensores distintos, garantindo alinhamento temporal e espacial. *Exemplo:* sincronizar medições de CSI com imagens térmicas para análise de atividades humanas.
- **Verificação de integridade e completude:** checar se o conjunto cobre o intervalo de interesse e se não há perdas não documentadas.

Depois da coleta, para avaliar a qualidade, métricas objetivas devem ser empregadas, como taxa de dados ausentes, consistência interna entre variáveis relacionadas e estabilidade temporal das medições. Em projetos de engenharia de telecomunicações, esse cuidado é vital para evitar interpretações errôneas causadas por artefatos de coleta ou problemas operacionais, garantindo que as conclusões derivadas das visualizações sejam sustentadas por uma base de dados confiável. A isto chamamos de teste de sanidade[3]

2.1.3 Exploração e análise

Objetivo: compreender a estrutura dos dados antes de modelar ou concluir. A *Exploratory Data Analysis* (EDA) é a etapa em que o pesquisador deixa de ser um mero espectador do *dataset* e começa a interagir criticamente com ele.

[3] Em 1999, a NASA perdeu a sonda *Mars Climate Orbiter* por uma "pequena" falha: uma parte da equipe usava unidades imperiais (libras-força) e a outra usava o sistema métrico (newtons). O software integrou dados "corretos", mas com escalas diferentes — e ninguém conferiu. Resultado: a sonda queimou na atmosfera de Marte. Moral da história: não assuma nada. 🛰️ → 🔥

O foco aqui é observar, desconfiar e testar suposições iniciais: os dados são simétricos? Há *outliers*? Mudam no tempo, há indícios de agrupamento? O comportamento observado faz sentido? Como outros autores lidaram com esse problema?

Essas perguntas podem ser respondidas com ferramentas básicas e/ou poderosas: as medidas de posição e dispersão. Apesar de simples, elas fornecem as primeiras pistas sobre como os dados "respiram" — e às vezes já sinalizam onde vai emperrar.

Os primeiros passos da EDA envolvem quantificar aspectos fundamentais dos dados: onde se concentram, como se dispersam e quão simétricos (ou não) são. Esses sumários estatísticos permitem identificar padrões gerais, suspeitar de distorções e começar a desenhar o mapa do terreno antes de qualquer modelagem.

Procedimentos essenciais de EDA

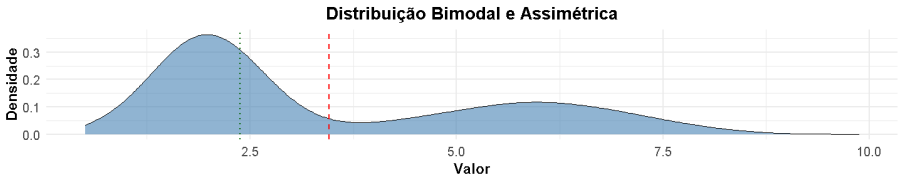
Sumários:

- **Média**: útil para dados simétricos; deve ser usada com cautela em distribuições assimétricas e/ou com outliers.
- **Mediana**: representa o centro robusto; preferível em distribuições assimétricas.
- **Variância e desvio padrão**: medem dispersão; essenciais para comparar homogeneidade entre grupos.
- **Assimetria e curtose**: caracterizam o formato da distribuição; indicam viés, concentração e cauda.
- **Percentis por grupo**: revelam diferenças internas entre subconjuntos de dados. Por exemplo, em duas redes com mesma mediana de latência, uma pode ter 90% dos pacotes abaixo de 100 ms, enquanto a outra tem metade dos pacotes acima de 200 ms — algo que só os percentis por grupo revelam.

Visualizações revelam o que os números sozinhos não mostram. A inspeção gráfica permite detectar distribuições incomuns, agrupamentos inesperados, comportamentos sazonais ou anomalias evidentes — mesmo antes de qualquer formalização estatística. A Figura 2.1.1 ajuda a entender tanto a necessidade de inspeção quanto o sumário.

Inspeção gráfica:

- Permite identificar padrões, desvios e inconsistências de forma intuitiva.



Métrica	Valor	Interpretação
Mínimo	0.50	Menor valor observado.
1º Quartil (Q1)	1.86	25% dos dados estão abaixo deste valor.
Mediana	2.38	Valor central (não sensível a outliers).
Média	3.45	Média aritmética (sensível a outliers).
3º Quartil (Q3)	5.42	75% dos dados estão abaixo deste valor.
Máximo	9.87	Maior valor observado.
Desvio padrão	2.06	Medida de dispersão dos dados em torno da média.
Assimetria	0.75	Indica a inclinação da distribuição (positivo = cauda à direita).
Curtose	2.13	Mede o 'peso' das caudas (valores altos indicam mais extremos).

Figura 2.1.1: Distribuição bimodal e assimétrica gerada como mistura de duas normais com médias diferentes. As linhas tracejada (vermelha) e pontilhada (verde) indicam, respectivamente, a média e a mediana. A figura ilustra como sumários estatísticos podem diferir significativamente em distribuições não simétricas, exigindo cautela na sua interpretação.

- Histogramas, densidades e boxplots são úteis para distribuições; dispersões e séries para relações temporais.
- Avaliar correlações, *lags* e sazonalidades ajuda a antecipar comportamentos ou ajustar

Muitos fenômenos em engenharia não são aleatórios: seguem padrões, ciclos ou dependem do passado. Identificar essas estruturas é crucial para escolher os modelos adequados e evitar interpretações ingênuas.

Antes da avalanche de IA, análises robustas já eram feitas com ferramentas estatísticas e bibliotecas programáveis. Elas continuam indispensáveis quando se exige controle fino, replicabilidade e domínio total sobre os dados e os métodos.

Ferramentas Tradicionais

- **R**: `tidyverse`, `ggplot2`, `data.table`.
- **Python**: `pandas`, `matplotlib`, `plotly` (modo offline), `statsmodels`.
- **Matlab/Octave**: análise de sinais e prototipagem numérica.

Com o avanço dos modelos de linguagem e da matemática simbólica computacional, surgiram ferramentas capazes de auxiliar desde tarefas simples até

auditorias formais de modelos. Algumas dispensam programação, outras aprofundam a análise simbólica — todas ampliam as possibilidades para quem precisa validar, testar e explorar hipóteses com precisão.

Plataformas de análise com uso de IA

- **Julius AI**, **ChatGPT (análise de dados)**: consulta em linguagem natural, geração de gráficos e modelos sem necessidade de código extenso.
- **Wolfram Alpha**, **Wolfram Language**: manipulação simbólica avançada, resolução de equações, derivadas, integrais e análise formal de modelos matemáticos.
- **SymPy (Python)**: biblioteca para álgebra simbólica, útil em ambientes programáveis para validar expressões analíticas, equações diferenciais, simplificações e verificação de consistência estrutural.
- **CoCalc + SageMath**: ambiente colaborativo com suporte a matemática simbólica, ideal para exploração formal de modelos estatísticos ou determinísticos.
- **MathGPT e similares**: ferramentas emergentes baseadas em LLMs treinadas com foco exclusivo em linguagem matemática, capazes de auditar etapas de modelagem com raciocínio simbólico.

Quando os dados excedem a capacidade de memória ou envolvem milhões de registros, métodos convencionais tornam-se ineficientes. A análise de bases extensas exige ferramentas otimizadas, formatos colunar-binários e técnicas específicas de indexação, particionamento e processamento em fluxo. Esta lista reúne alternativas robustas para lidar com esse cenário com desempenho e escalabilidade.

Bases extensas: métodos e formatos

- **SQL e time-series**: PostgreSQL, MySQL; **QuestDB**, InfluxDB para séries temporais.
- **Indexação**: *bitmap* e índices compostos para acelerar filtros por tempo/categoria.
- **Formatos binários**: `Parquet`, `Feather/Arrow` (colunar, compressão, leitura vetorizada).
- **Particionamento e compressão**: por data/enlace/subportadora; compactação *lossless*.

- **Streaming**: Kafka, Flink, Spark Streaming para ingestão e EDA quase em tempo real.

Antes de entrar na modelagem propriamente dita, é essencial reconhecer que a análise exploratória não serve apenas para ”olhar os dados”, ela estrutura o pensamento. Nesta etapa, hipóteses são descartadas com evidências, figuras ganham ou perdem relevância estratégica, e lacunas ocultas nos dados vêm à tona. Mesmo que essas entregas raramente apareçam nos artigos finais, elas moldam as decisões metodológicas e fortalecem os argumentos que sobreviverem até a submissão. Ignorá-las é caminhar às cegas com uma lanterna desligada.

Entregáveis da EDA - Oraganização do texto

- Hipóteses refinadas e descartes justificados.
- Conjunto curto de figuras “candidatas” e métricas para a próxima fase.
- Lista de lacunas de dados/metadados a sanear antes da modelagem.

A última etapa do pipeline não é apenas mostrar os resultados, mas garantir que eles possam ser verificados, reutilizados e compreendidos por outros. Isso envolve decisões cuidadosas sobre como apresentar visualmente as descobertas, documentar o processo de análise e manter um controle rigoroso de versões e *scripts*. A clareza na comunicação e o compromisso com a reprodutibilidade são marcas de trabalhos científicos robustos e úteis.

2.1.3.1 Comunicação e reprodutibilidade

- **Figuras finais**: devem ser legíveis, autocontidas e interpretáveis sem o auxílio do autor. Títulos, legendas, escalas e unidades devem ser explícitos.
- **Estático vs interativo**: gráficos estáticos são preferíveis para publicações formais (PDF, artigos), enquanto gráficos interativos (com `plotly`, `ggiraph`, `shiny`) favorecem exploração em ambientes digitais.
- **Acessibilidade**: figuras com boa distinção de cores, contraste adequado e uso mínimo de elementos ambíguos garantem compreensão ampla (incluindo daltônicos).
- **Documentação**: scripts devem ser comentados, com dependências explícitas, entradas bem definidas e estrutura modular para facilitar reuso.
- **Controle de versão**: uso de ferramentas como `git` garante rastreabilidade de alterações e facilita colaboração técnica.

- **Repositórios e persistência**: dados, códigos e figuras devem ser arquivados em locais de acesso aberto (como GitHub, Zenodo ou Figshare) sempre que possível, com versões bem etiquetadas.

3 Visualização de Dados, Dicas e Armadilhas

Ao criar gráficos, é fundamental considerar as capacidades perceptivas e cognitivas do leitor/avaliador. Gráficos eficazes são aqueles que aproveitam os pontos fortes das habilidades humanas de processamento de informações e evitam os pontos fracos. Na visualização de dados, as informações são representadas por objetos geométricos: dados quantitativos ou dividido em categorias devem ser mapeados para objetos visuais, normalmente através de atributos como posição, tamanho, forma ou cor.

Figura 3.0.1: Claus O. Wilke é um biólogo computacional e evolucionário e presidente do Departamento de Biologia Integrativa da Universidade do Texas em Austin, onde é o Dwight W. and Blanche Faye Reeder *Centennial Fellow* em Biologia Sistemática e Evolutiva, e atualmente (2024) ocupa a Cátedra de Professor Regente Biociências Moleculares na Universidade do Texas.

Teste você mesmo se algumas destas representações são mais fáceis de decodificar que outras. Cada coluna no gráfico apresentado na Figura 3.0.2 codifica os mesmos 7 valores usando um recurso gráfico diferente, como saturação, volume, área, ângulo, comprimento e posição. A precisão e a percepção de cada tipo de representação podem variar significativamente. Valores representados de cima para baixo são: 15, 30, 50, 10, 32, 29, 40. Avalie qual dessas representações transmite a informação de maneira mais clara e eficiente é essencial para evitar armadilhas na visualização de dados, garantindo que a interpretação dos dados seja precisa e intuitiva.

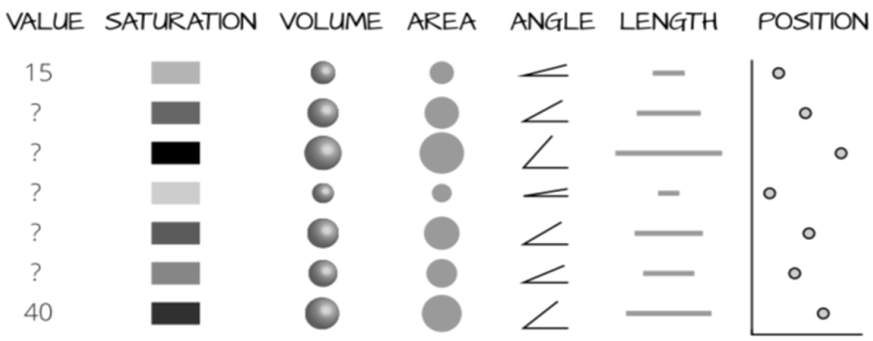


Figura 3.0.2: Valores da coluna *value* são mapeados para diferentes elementos visuais.

No próximo exemplo, Figura 3.0.3, no que diz respeito às limitações perceptivas, devemos ter cuidados especiais são ao se comparar visualmente curvas. A diferença entre as curvas A e B varia em função de x:

- (A) $(a - b)$ aumenta exponencialmente com x.

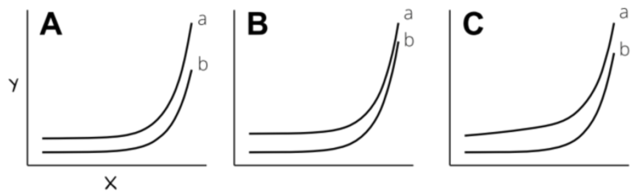


Figura 3.0.3: perceber a real diferença entre as curvas pode não ser trivial.

- (B) $(a - b)$ permanece constante em x.
- (C) $(a - b)$ aumenta linearmente com x.

Ao considerar estas variações, é importante garantir que as curvas sejam apresentadas de forma que as diferenças sejam perceptíveis e compreensíveis, evitando interpretações errôneas dos dados.

Devemos ter extremo cuidado com a descrição dos elementos gráficos[4], escolhemos cuidadosamente cada palavra e consideramos a sua integração ao texto ao redor. Da mesma forma, ao representarmos os dados, devemos seguir o mesmo processo e escolher cada elemento visual com base em sua função no gráfico. Boas escolhas de representação, assim como uma boa redação[5], tornam as ideias fáceis de entender, mas erros acontecem.

[4] A apresentação de uma elemento visual é tão importante para o trabalho que não podemos sequer deixar a interpretação por conta do leitor. Devemos guiar o leitor ao longo de **cada** figura, para que ele não se concentre em algo diferente do que gostaríamos.

[5] Outra dica importante: as legendas das figuras não precisam ter necessariamente apenas uma linha, é preciso ser breve e conciso, mas não deixe de se alongar se a figura precisar ser bem explicada.

3.1 EXTRAVAGANTE, CONFUSO, ERRADO

Podemos separar os erros de apresentacão em três casos principais:

- **Extravagante:** a figura 3.1.1 **b** tem problemas estéticos, mas por outro lado é clara e informativa.
- **Confuso:** a figura 3.1.1 **c** apresenta problemas relacionados à percepção: pode não ser clara ou até confusa.
- **Errado:** a figura 3.1.1 **d** está objetivamente incorreta.

Na visualização de dados, a escolha cuidadosa dos elementos visuais é tão importante quanto a seleção das palavras ao escrevermos um texto. Cada elemento gráfico deve ser escolhido com base em sua função específica dentro do gráfico, de forma a maximizar a clareza e a eficácia na transmissão da informação. Assim como uma boa redação torna as ideias mais fáceis de entender, uma representação visual bem planejada facilita a compreensão dos dados. Para isso, é essencial seguir princípios de design que orientem a escolha dos elementos visuais.

O objetivo de qualquer visualização de dados deve ser sempre claramente definido antes de sua implementação. Projetar uma visualização sem um objetivo claro é como viajar sem um destino definido: a eficácia da visualização

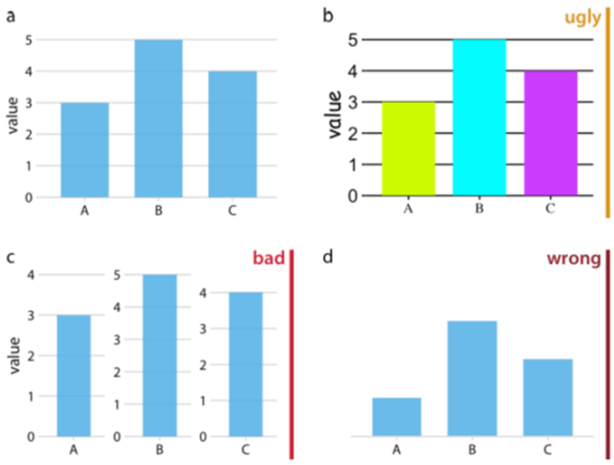


Figura 3.1.1: um exemplo de apresentação de resultados que podem ser classificadas como: b) extravagante, c) confuso e d) errado.

pode ser seriamente comprometida. Por exemplo, se o objetivo é responder a uma pergunta específica, como "Qual é a distribuição estatística dessa variável?", a escolha do gráfico deve ser direcionada para evidenciar essa informação. Além disso, é importante considerar questões secundárias, como a necessidade de transformação das variáveis ou a presença de observações extremas, para que a visualização seja tanto informativa quanto precisa.

Os princípios fundamentais da visualização de dados também incluem a primazia dos dados sobre os elementos decorativos. Um gráfico eficaz deve priorizar os dados e usar as anotações apenas como apoio, garantindo que a informação principal não seja obscurecida. Elementos gráficos como cores e saturação devem ser utilizados de forma a destacar o essencial, sem distrações desnecessárias. Se houver dúvidas sobre a importância relativa dos elementos visuais, a recomendação é minimizar a ênfase nos elementos menos relevantes para que os dados prevaleçam na percepção do leitor. Em suma, uma boa visualização de dados deve buscar o equilíbrio entre clareza, objetividade e a correta interpretação das informações apresentadas. Um gráfico tipicamente inclui duas partes:

- Os dados e as anotações que colocam os dados em contexto.
- Os dados devem sempre assumir o papel principal, com as anotações apenas como apoio.
- A compreensão pode ser diminuída quando detalhes menores são destacados. Ajuste o tamanho do objeto, a espessura da linha, a cor, a saturação, etc.

Se você tiver dúvidas quanto à proeminência visual dos elementos, tente olhar a imagem à distância: os dados devem ser a característica mais saliente e as anotações não devem desaparecer no fundo.

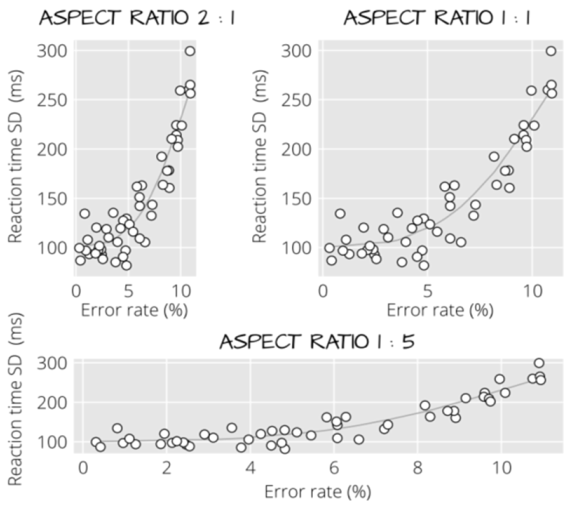


Figura 3.1.2: Qual a visualização explica melhor a correlação?

Na visualização de dados, a simplicidade é um princípio fundamental para garantir a clareza e a eficácia da comunicação. A competição visual entre os elementos de um gráfico pode dificultar a compreensão da mensagem central, por isso, é crucial utilizar o menor número possível de elementos. Um design eficiente prioriza a maximização da relação dado-tinta, onde a maior parte dos elementos gráficos representa dados relevantes, evitando elementos supérfluos que não contribuem para a interpretação da informação. Refinar os elementos restantes, removendo aqueles que não agregam valor, reforça a mensagem e melhora a legibilidade do gráfico.

Além disso, a consistência no uso de codificações, *layouts*, cores e outros elementos gráficos é essencial para alcançar uma continuidade visual que facilite a compreensão do leitor. Um *design* bem planejado reduz o esforço necessário para decifrar gráficos, evitando interpretações incorretas e garantindo que o leitor compreenda rapidamente a informação apresentada. No entanto, é importante equilibrar consistência com criatividade: enquanto designs inovadores têm seu lugar, os *displays* convencionais ou familiares geralmente exigem menos esforço cognitivo para serem compreendidos.

Por fim, a organização e a apresentação visual devem ser cuidadosamente planejadas para destacar as características mais importantes dos dados. Tipos comuns de gráficos podem ser organizados de acordo com a dimensionalidade e a ênfase desejada, como mostrado nos painéis de exemplos. Ao escolher o tipo de visualização, deve-se considerar a estrutura dos dados e a mensagem a ser

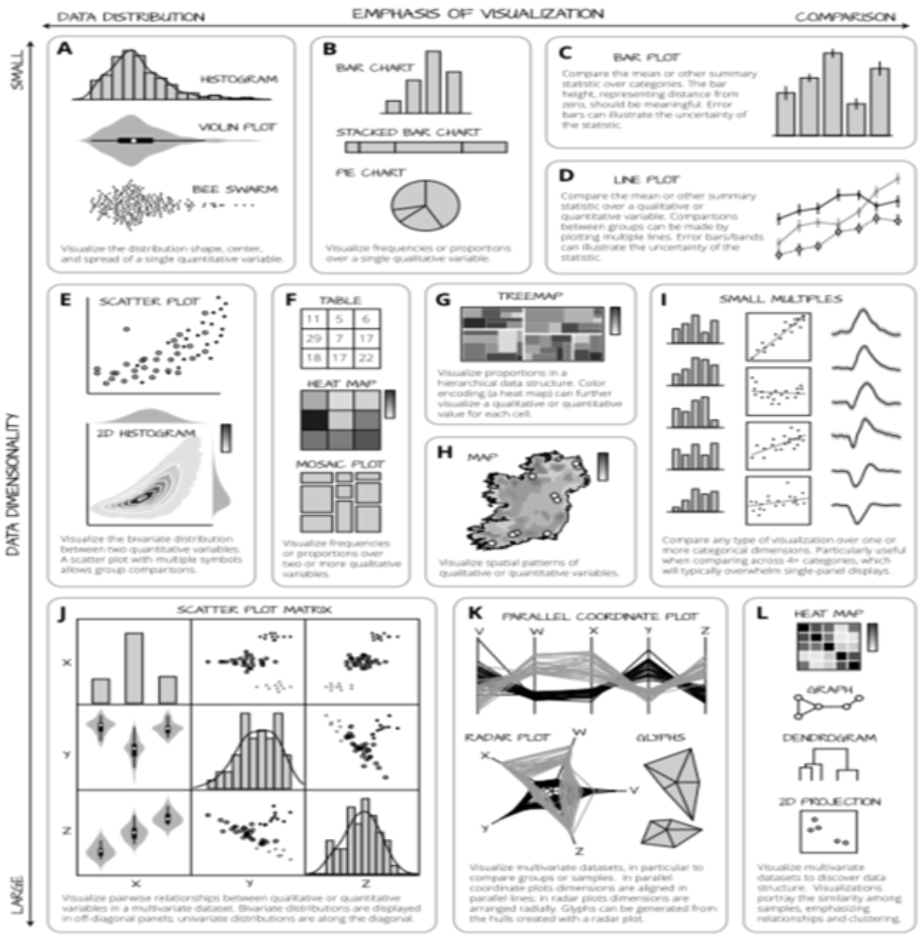


Figura 3.1.3: Tipos Comuns de Gráficos. Cada painel mostra possíveis visualizações baseadas no tipo de dados, dimensionalidade e ênfase desejada. Os painéis de A a L são organizados por dimensionalidade crescente dos dados (de cima para baixo) e ênfase crescente na comparação (da esquerda para a direita).

transmitida, sempre priorizando a clareza e a eficácia da comunicação.

Na visualização de dados, o equilíbrio entre consistência e criatividade é essencial para garantir a clareza e a eficácia na transmissão de informações. A escolha das proporções adequadas para os eixos X e Y é crucial para enfatizar as variações mais relevantes dos dados. Uma figura alongada ao longo do eixo Y e estreita no eixo X pode destacar variações em Y, enquanto uma figura baixa e larga pode fazer o oposto. Assim, ao representar os dados, deve-se escolher proporções que garantam que as diferenças mais importantes sejam perceptíveis e fáceis de interpretar.

Ao representar dados, é fundamental que a escolha do gráfico não apenas mantenha consistência visual, mas também revele aspectos relevantes da distribuição. Podemos usar a criatividade para municiar o leitor com mais informações sem nos alongar demais no texto.

A Figura 3.1.4 ilustra dois grupos (A e B) com médias semelhantes, como mostrado no gráfico de barras à esquerda. Embora as médias sejam praticamente iguais, isso não significa que as distribuições sejam idênticas.

O gráfico de violino (em azul, à direita, para o grupo B) revela que sua distribuição é **bimodal**, informação completamente oculta na visualização por médias. Já o grupo A (representado pelo violino em vermelho) apresenta uma distribuição mais próxima da normal, com menor variância.

Este exemplo evidencia que medidas-resumo, como a média, podem ocultar padrões importantes, e que a escolha inadequada de visualização pode levar a interpretações incorretas. Assim, a criatividade na seleção de gráficos deve sempre estar ancorada na consistência e na capacidade de expor a estrutura real dos dados.

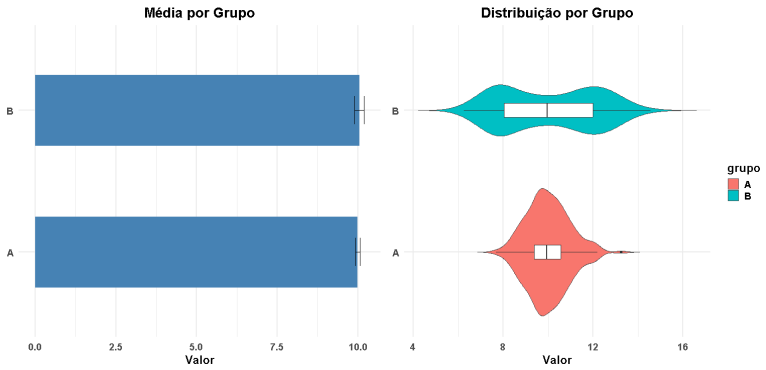


Figura 3.1.4: Comparação e complementaridade entre visualização por média e por distribuição (gráficos de violino) para dois grupos com médias semelhantes.

Quando trabalhamos com sistemas de coordenadas cartesianas, é comum que os eixos X e Y representem diferentes unidades de medida. Por exemplo, ao plotar a temperatura ao longo do tempo, os valores no eixo Y podem estar em

graus Celsius ou Fahrenheit, enquanto o eixo X pode representar o tempo em dias ou meses. A escolha das unidades deve ser cuidadosamente considerada, garantindo que a visualização permaneça clara e fiel à mensagem que se deseja transmitir.

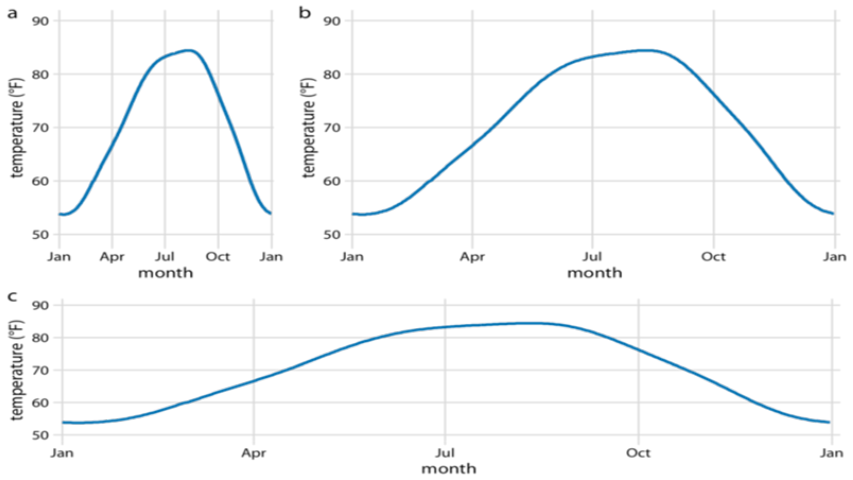


Figura 3.1.5: Temperatura diária normal para Houston, TX. A temperatura é mapeada para eixo y e dia do ano no eixo x. As partes (a), (b) e (c) mostram a mesma figura em diferentes proporções. Todas as três partes são visualizações válidas dos dados de temperatura. Fonte de dados: NOAA.

Além disso, quando os eixos X e Y são medidos nas mesmas unidades, o espaçamento da grade deve ser uniforme, e as distâncias nos eixos devem corresponder ao mesmo número de unidades de dados. Isso é particularmente importante para comparações diretas entre diferentes conjuntos de dados, como ao comparar a temperatura em diferentes cidades ao longo do tempo. Alterações nas unidades, como a conversão de Celsius para Fahrenheit, devem ser realizadas de forma a manter a integridade visual dos dados e a facilitar a compreensão do leitor. Em suma, a correta manipulação dos eixos e unidades é fundamental para a criação de visualizações que sejam, ao mesmo tempo, informativas e esteticamente equilibradas.

Ao criar visualizações, é crucial evitar erros comuns que podem comprometer a clareza e a precisão da informação apresentada. Visualizações extravagantes podem desviar a atenção do conteúdo principal, enquanto gráficos confusos ou incorretos podem levar a interpretações errôneas dos dados.

- Ao escrevermos, escolhemos cuidadosamente cada palavra e consideramos a sua integração ao texto ao redor.
- Ao representarmos os dados, devemos seguir o mesmo processo e escolher cada elemento visual com base em sua função no gráfico.

- Boas escolhas de representação, assim como uma boa redação, tornam as ideias fáceis de entender.
- Apresentamos cinco princípios de design que ajudarão a orientar suas escolhas.

Os princípios de design para visualização de dados incluem 1) simplicidade, 2) clareza, 3) precisão, 4) eficiência e 5) estética. Cada elemento no gráfico deve ter um propósito claro e contribuir para a compreensão da informação. É importante evitar o uso excessivo de cores e elementos decorativos que não agreguem valor à interpretação dos dados.

Ao seguir estas diretrizes, podemos criar visualizações que não só transmitem informações de forma eficaz, mas também são visualmente agradáveis e fáceis de entender. A visualização de dados é uma ferramenta poderosa na comunicação de resultados em engenharia de telecomunicações, facilitando a análise e a tomada de decisões com base em dados concretos.

Paleta de Cores

A escolha da paleta de cores é crucial na exibição de elementos gráficos em apresentações acadêmicas, pois influencia diretamente a clareza, a legibilidade e a percepção dos dados apresentados. Um dos aspectos mais importantes a considerar é a consistência do esquema de cores ao longo do trabalho: manter um esquema de cores consistente ajuda a criar uma identidade visual coesa, facilitando a interpretação e a comparação entre diferentes gráficos e seções do trabalho.

Pacotes de visualização de dados como `ggplot2` no  e no `matplotlib` do  ou oferecem seus próprios temas de cores, que podem ser ajustados conforme necessário. Esses pacotes incluem paletas de cores pré-definidas que foram cuidadosamente projetadas para maximizar a distinção entre categorias diferentes e garantir a acessibilidade para pessoas com daltonismo. Utilizar esses temas pode economizar tempo e garantir uma apresentação profissional, desde que as paletas escolhidas sejam aplicadas de forma consistente em todos os gráficos do trabalho. Aqui ainda vc pode usar as cores ao seu favor para evidenciar uma variável entre muitas.

Em algumas situações, especialmente em publicações impressas em preto e branco, a escolha de cores se torna ainda mais crítica. Nesses casos, é essencial utilizar diferentes padrões de hachura ou diferentes tons de cinza para distinguir entre elementos gráficos. Certifique-se de que a distinção entre as diferentes categorias ou séries de dados seja clara mesmo sem a ajuda das cores. Isso pode ser obtido através do uso de contrastes de luminosidade e texturas variadas. A Figura 3.1.6 exibe um contraste óbvio para um condado bem pequeno (chamado *Loving*) a oeste do Texas.

O uso de gradientes ao invés de cores sólidas pode ser uma ferramenta poderosa para representar variáveis contínuas e ajudar a visualizar transições suaves

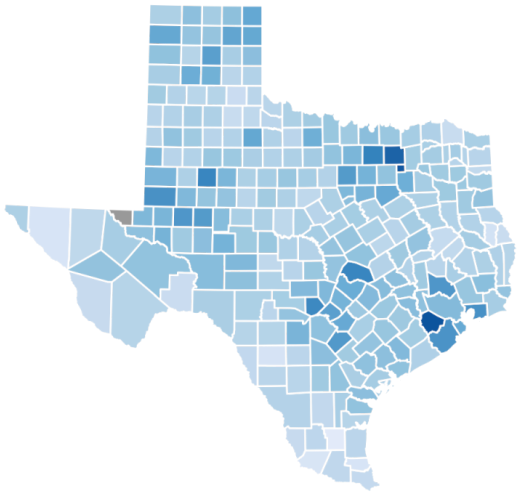


Figura 3.1.6: use as cores a seu favor, nesta imagem podemos destacar em cinza um condado específico a oeste do estado do Texas nos Estados Unidos.

entre valores. No entanto, deve-se tomar cuidado para que os gradientes não confundam o leitor. Gradientes devem ser aplicados de maneira que adicionem clareza, utilizando escalas de cores perceptualmente uniformes, onde mudanças na cor correspondem diretamente a mudanças nos valores dos dados. Na Figura 3.1.7 usamos quatro dimensões (uma está implícita) para apresentar os dados de altura x gordura corporal, neste caso está evidenciada a quarta dimensão na variável *track*. Então temos os eixos X e Y, polígonos diferentes nos rótulos (*point shapes* no ggplot) da legenda e temos a cor vermelha para a variável relevante.

Além desses pontos, a acessibilidade é um fator importante a ser considerado na escolha das cores. Paletas de cores acessíveis garantem que todas as audiências, incluindo aquelas com deficiências visuais como daltonismo, possam compreender as informações apresentadas. Ferramentas como simuladores de daltonismo podem ser usadas para verificar a acessibilidade das paletas de cores escolhidas. Dessa forma, uma escolha cuidadosa das cores não só melhora a estética da apresentação, mas também amplia seu alcance e impacto.

Fontes em Rótulos, Títulos e Eixos:

Fontes muito pequenas dificultam a leitura e podem desviar a atenção do conteúdo principal, enquanto fontes muito grandes podem sobrecarregar a visualização e obscurecer detalhes importantes dos dados. É essencial encontrar um equilíbrio que permita a fácil leitura à distância, especialmente em ambientes de apresentação. Além disso, a consistência no tamanho da fonte ao longo

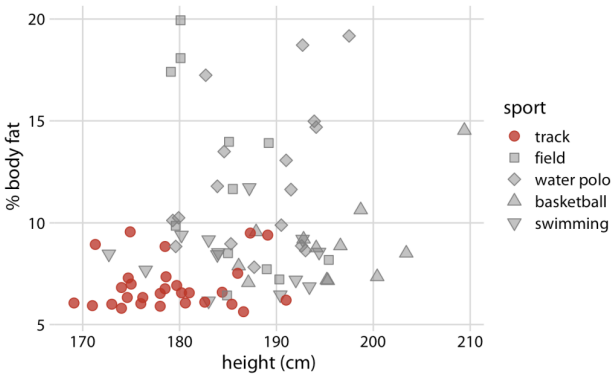


Figura 3.1.7: Representação multivariada de altura (x), gordura corporal (y), tipo de esporte (forma dos marcadores) e destaque cromático para a variável `track`. A legenda combina atributos visuais para codificar múltiplas dimensões em uma única visualização.

de todos os gráficos mantém uma aparência profissional e coesa. Ajustar os tamanhos das fontes para garantir que todos os elementos textuais sejam claramente visíveis e bem integrados ao *design* geral do gráfico. Uma boa prática é testar a visualização em diferentes dispositivos e distâncias[6] para assegurar que a informação esteja legível a toda a audiência, independentemente de sua posição na sala de apresentação.

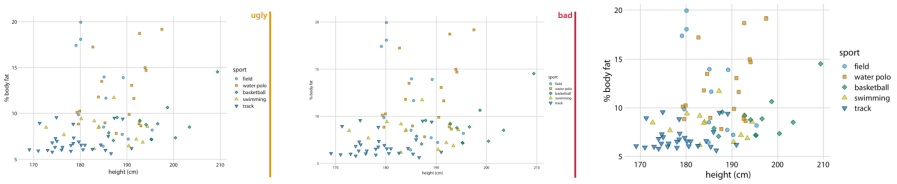


Figura 3.1.8: três diferentes visualizações (ruins), mostrando a importância do tamanho das fontes dos eixos e rótulos nos gráficos, da esquerda para direita veja o tamanho das fontes. Este é um exemplo e um contra-exemplo, Use o *zoom*!

Linhas Sobrepostas, Grids e Preenchimento:

Sempre que possível, apresente seus dados com formas sólidas e coloridas[7]. Evite sobrepor linhas de grade sobre a camada dos sólidos. Formas sólidas são mais facilmente percebidas como objetos coesos, menos propensas a criar ilusões de ótica [1] e transmitem quantidades de maneira mais imediata do que polígonos não preenchidos. Visualizações usando formas sólidas são mais claras e agradáveis de se ver do que versões equivalentes que usam hachuras.

[6] Quanto ao tamanho das fontes uma boa prática é:

1. Títulos dos Slides: fonte entre 32 e 44 pontos.
2. Texto Corporal: fonte entre 24 e 32 pontos.
3. Subtítulos: fonte entre 20 e 28 pontos é apropriado.
4. Legendas e Rótulos em Gráficos: fonte entre 18 e 24 pontos.
5. Texto em Tabelas: fonte entre 18 e 24 pontos.
6. Rodapés e Notas de Rodapé:fonte entre 14 e 18 pontos para rodapés e notas de rodapé.
7. Chegue cedo e teste a apresentação antes a partir de todos os pontos da sala.

[7] A presença de hachuras em ilustrações têm uma longa história na visualização. Durante parte do século XX, as visualizações científicas eram desenhadas à mão e eram impressas em preto e branco. Isso impedia o uso de áreas preenchidas ou coloridas, incluindo preenchimentos em tons de cinza. Áreas preenchidas eram às vezes simuladas por padrões de hachura. Ferramentas ainda imitam essas simulações, usando extensivamente linhas, padrões tracejados ou pontilhados, hachuras e símbolos.

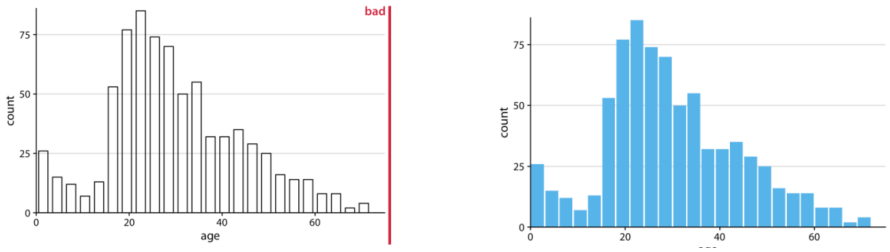


Figura 3.1.9: duas diferentes visualizações, demonstrando que sólidos não preenchidos e linhas de grade sobrepostas prejudicam a visualização.

O uso mais comum e inadequado de desenhos de linhas é em histogramas e gráficos de barras. O problema com barras desenhadas como contornos é que não é imediatamente claro qual lado de uma linha está dentro de uma barra e qual lado está fora. Isso resulta em um padrão visual confuso, especialmente quando há lacunas entre as barras, prejudicando a mensagem principal da figura. Preencher as barras com uma cor clara, ou cinza. ainda que a linha de *grid* cruze sobrepondo o elemento visual mais relevante.

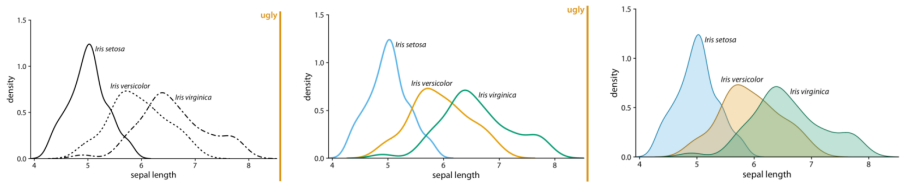


Figura 3.1.10: três diferentes visualizações, mostrando a importância do preenchimento, da transparência e do uso da paleta de cores para conferir uma visualização clara, rigorosa e agradável.

Na Figura 3.1.10, observa-se a importância do uso adequado da paleta de cores, preenchimento e transparência na visualização de dados. No primeiro gráfico, os contornos de linha podem criar confusão visual, dificultando a interpretação precisa de qual lado da linha pertence a qual categoria. No segundo gráfico, o uso de cores distintas para cada categoria melhora a diferenciação visual, facilitando a identificação imediata das áreas representadas. No terceiro gráfico, a adição de transparência e o preenchimento permite a sobreposição das áreas coloridas sem perder a visibilidade das informações subjacentes, destacando claramente onde os conjuntos de dados se sobrepõem ou divergem. Essas técnicas combinadas resultam em gráficos que são esteticamente agradáveis e eficazes na transmissão precisa e clara das informações complexas. Abaixo na Figura 3.1.11 mais um exemplo e um contra-exemplo deste tópico[8].

Pense bem antes de usar o 3D:

[8]Aqui é importante notar que as vezes temos que estar atentos a tradição da publicação. Não é incomum que alguns periódicos, bancas e avaliadores, prefiram as figuras e diagramas em tons de cinza ou preto e branco, seja por tradição, porque preferem ler impresso ou idiossincrasias particulares. Então fique atento ao seu público!

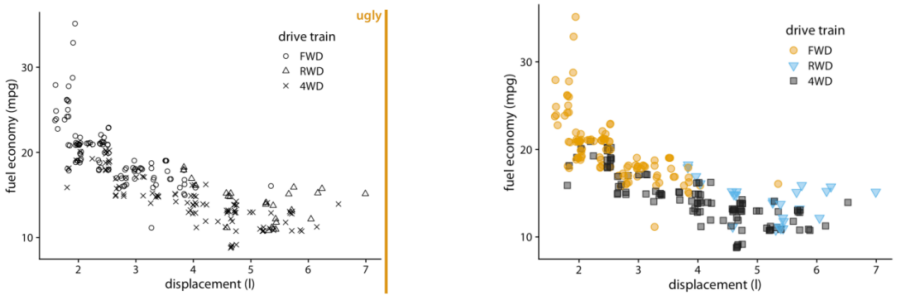


Figura 3.1.11: duas diferentes visualizações, mostrando a importância do preenchimento dos pontos, da transparência e do uso da paleta de cores para conferir uma visualização clara, rigorosa e agradável.

Gráficos 3D são populares em apresentações corporativas, mas frequentemente são usados de forma inadequada. A maioria dos gráficos 3D pode ser melhorada ao ser transformada em figuras 2D. Ferramentas de visualização frequentemente permitem transformar elementos gráficos em objetos tridimensionais, como gráficos de pizza em discos girados, gráficos de barras em colunas e gráficos de linhas em bandas. No entanto, esta terceira dimensão raramente transmite dados reais, sendo usada apenas para adornar o gráfico, o que é considerado desnecessário e prejudicial.

A projeção de objetos 3D em duas dimensões distorce os dados, tornando difícil para o sistema visual humano corrigir essa distorção. Por exemplo, um gráfico de pizza girado no espaço pode distorcer as proporções entre as fatias. Visualizações 3D exigem duas transformações de dados: uma mapeando os dados para o espaço 3D e outra mapeando do espaço 3D para o 2D da figura final. Esta última transformação é não invertível, dificultando a determinação precisa da posição dos dados no espaço 3D.

Raramente é necessário adicionar uma terceira dimensão como escala de posição (no espaço), pois variáveis podem ser mapeadas em escalas em cor, tamanho ou forma.

Na Figura 3.1.13, a terceira dimensão é utilizada como escala de cores e tamanhos, em vez de uma escala de posição, o que muitas vezes prejudica a interpretação dos dados. Neste gráfico de dispersão, a cor dos pontos representa o número de cilindros (4, 6 ou 8) e o tamanho dos pontos representa a eficiência de combustível (mpg - milhas por galão). Essa abordagem evita as distorções comuns associadas ao uso de gráficos 3D para representar dados em um espaço 2D, onde a projeção pode levar a interpretações incorretas. Usar cores e tamanhos para adicionar uma dimensão adicional de informação permite uma interpretação mais clara e direta dos dados, destacando padrões e relações sem confundir o observador com distorções visuais.

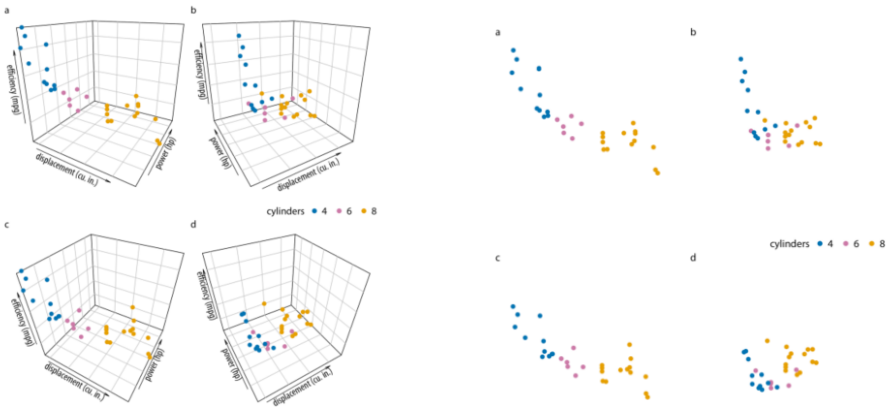


Figura 3.1.12: duas diferentes visualizações, mostrando um conjunto de pontos em 3D com e sem eixos e *grids*.

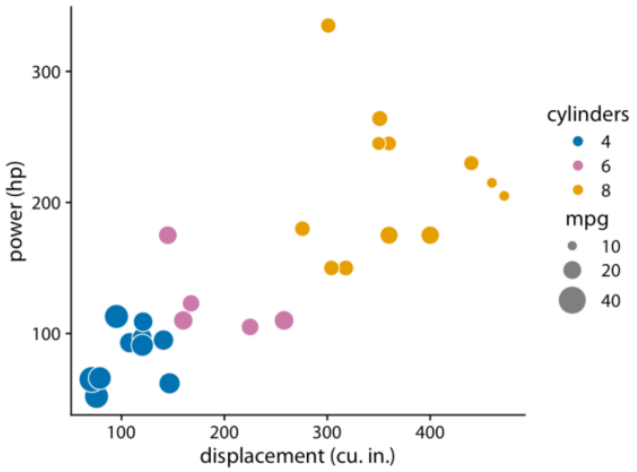


Figura 3.1.13: a figura mostra um gráfico no plano onde a 3a dimenão é representada como uma escala de cor e tamanho.

SEMPRE AJUDE O LEITOR

As duas Figuras 3.1.14a) e 3.1.14b) são diagramas que representam redes Clos cada uma representa a possibilidade ou não da conexão J ter sucesso.

Ao apresentar figuras, prefira sempre aquelas cuja mensagem visual seja evidente à primeira vista. Contudo, nem sempre isso será viável. Certos problemas, por sua própria natureza, exigem representações visuais mais densas, cuja interpretação correta requer um olhar mais atento por parte do leitor.

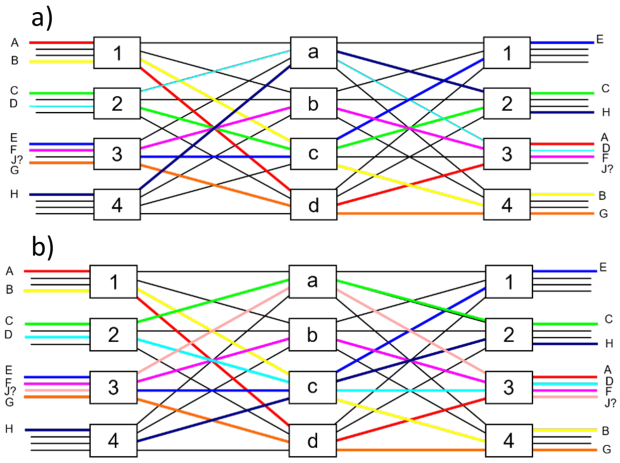


Figura 3.1.14: Veja o caso das Figuras a) e b) elas são bastantes similares, mas podem significar o funcionamento ou não um rearranjo não bloqueante. Embora esse seja um exemplo do que não deve ser feito, se for inevitável, o pesquisador deve conduzir o leitor a interpretação desejada.

Em ambos os casos, a legenda da figura e a descrição no corpo do texto devem atuar como guias interpretativos. Utilize a legenda para conduzir o leitor, destacando os aspectos relevantes da imagem e orientando a leitura de modo claro. Lembre-se de que a legenda não precisa se restringir a uma única linha: ela pode conter explicações mais detalhadas sempre que necessário. Da mesma forma, toda figura deve ser mencionada explicitamente no texto, preferencialmente no momento exato em que sua leitura se torna útil para a compreensão do argumento.

Um bom exemplo pode ser observado na comparação entre as Figuras 3.1.14a) e b). Uma olhada rápida pode levar o leitor a pensar que se trata da mesma imagem. No entanto, existe uma diferença fundamental: na Figura 3.1.14a), há um bloqueio na conexão para o nó J; já na Figura 3.1.14b), a conexão para J é bem-sucedida. Portanto, evidenciar essas distinções sutis — como o uso de cores diferentes para as linhas associadas a J (preta em a, laranja em b) — é essencial para garantir que a interpretação seja precisa.

Técnicas visuais que enfatizam variações críticas, em vez de reforçar semelhanças superficiais, são particularmente úteis nesse contexto. Elas ajudam a evitar mal-entendidos e asseguram que o leitor perceba as nuances importantes da análise[9].

4 Camadas e Separação Visual

Um gráfico tipicamente inclui duas partes: os dados e as anotações que colocam os dados em contexto. Os dados devem sempre assumir o papel principal, com

[9]Para garantir que o leitor compreenda corretamente as informações transmitidas por uma figura, é essencial guiá-lo na interpretação, sobretudo em representações visuais mais complexas. No exemplo das Figuras 3.1.14a) e b), embora inicialmente pareçam iguais, uma inspeção detalhada revela distinções cruciais, como a mudança de cor na conexão para o nó J. Apontar explicitamente essas diferenças é uma forma eficaz de conduzir a leitura.

as anotações apenas como apoio: a compreensão pode ser diminuída quando detalhes menores são destacados. Ajuste o tamanho do objeto, a espessura da linha e a cor para destacar os dados. Se houver dúvidas quanto à proeminência visual dos elementos, tente olhar a imagem à distância para garantir que os dados sejam a característica mais saliente e as anotações não desapareçam no fundo[10].

A importância do objetivo é crucial: projetar uma visualização sem um objetivo claro é como viajar por uma bela paisagem sem destino. O objetivo pode ser uma pergunta que a visualização deve responder, como "qual é a distribuição estatística da variável X?". Questões secundárias, como "a variável requer transformação?" e "há tendências nos extremos?", também devem ser consideradas.

A implementação visual dos dados deve escolher gráficos apropriados, mantendo o foco no objetivo: ter um objetivo ajuda a determinar quais informações são necessárias e a relevância de cada elemento visual. Cada elemento no gráfico deve ter um propósito claro e contribuir para a compreensão da informação, evitando o uso excessivo de cores e elementos decorativos que não agreguem valor à interpretação dos dados.

Ao seguir estas diretrizes, podemos criar visualizações que não só transmitem informações de forma eficaz, mas também são visualmente agradáveis e fáceis de entender. A visualização de dados é uma ferramenta poderosa na comunicação de resultados em engenharia de telecomunicações, facilitando a análise e a tomada de decisões com base em dados concretos.

[10] Ao escrevermos, escolhemos cuidadosamente cada palavra e consideramos a sua integração ao texto ao redor, ao representarmos os dados, devemos seguir o mesmo processo e escolher cada elemento visual com base em sua função no gráfico. Boas escolhas de representação, assim como uma boa redação, tornam as ideias fáceis de entender.

4.1 O CONCEITO DE CAMADAS

O conceito de camadas (*layering*) é uma estratégia essencial na construção de visualizações eficazes. Ao decompor um gráfico em componentes visuais distintos, organizados em camadas sobrepostas, é possível representar múltiplas dimensões de um conjunto de dados sem comprometer a legibilidade.

Essa ideia, levada ao nível de implementação, como ocorre no pacote `ggplot2` [5] do R, permite construir gráficos de forma incremental: camadas podem ser adicionadas, removidas ou modificadas individualmente sem a necessidade de reescrever toda a estrutura gráfica. Essa modularidade é uma das grandes características do `ggplot2`, permitindo flexibilidade e reuso com simplicidade, mantendo o rigor semântico[11]

[11] Hadley Wickham, ver Figura 4.1.1, não criou só um pacote. Criou um jeito de pensar gráficos. Com a invenção do + como adicionador de camada, o `ggplot2` virou Lego para dados.

Figura 4.1.1: Hadley Wickham. O cérebro por trás do `ggplot2`. Se hoje construímos gráficos por camadas como quem monta um sanduíche, a culpa (ou o mérito) é dele.

4.1.1 Separação de Elementos Visuais

Camadas permitem isolar diferentes aspectos da informação. Por exemplo, uma camada pode representar os valores centrais (como médias ou medianas), enquanto outra exibe a variabilidade (como barras de erro ou bandas de confiança). Essa separação evita sobrecarga visual e reforça o papel de cada elemento no raciocínio analítico — dados principais, incerteza, padrões esperados ou anotações.

Flexibilidade e Controle Semântico:
O uso de camadas confere controle refinado sobre a construção gráfica, permitindo que a complexidade visual seja ajustada conforme o objetivo analítico. Em ambientes como o `ggplot2`, cada função iniciada com `geom_` representa uma camada semântica independente, que pode ser adicionada, removida ou customizada isoladamente. Esse modelo modular favorece representações ricas, claras e adaptáveis ao propósito específico da visualização.

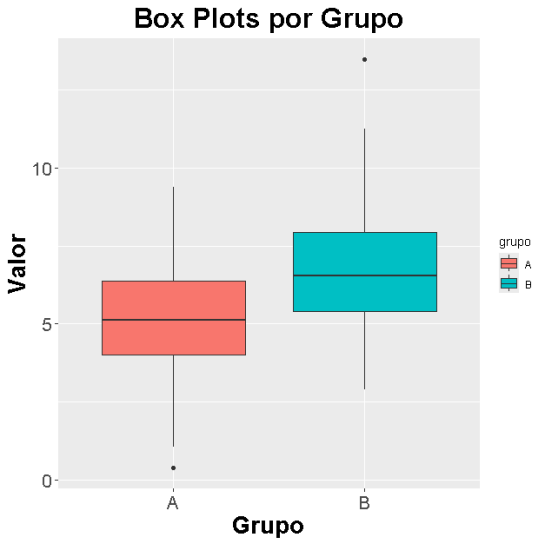


Figura 4.1.2: gráfico do tipo *boxplot*, composto por múltiplas camadas: a caixa representa os quartis e a mediana; os pontos sobrepostos indicam os *outliers*; e a legenda constitui mais uma camada visual.

Podemos acompanhar um exemplo prático no gráfico *boxplot*, ilustrado na Figura 4.1.2, são estruturados por camadas conceituais distintas. A camada principal evidencia a dispersão interquartil e a mediana dos dados; uma segunda camada destaca os *outliers*, que não são descartados, mas visualmente separados; e uma terceira camada corresponde à legenda. Essa separação visual é essencial em contextos de engenharia e análise estatística, permitindo distinguir padrões típicos de comportamentos atípicos ou ruídos experimentais.

O código R que gera esse gráfico modular pode ser visto na Listagem 4.1.

```r
library(ggplot2)

# Gerar dados simulados
set.seed(123)
data <- data.frame(
grupo = rep(c("A", "B"), each = 100),
valor = c(rnorm(100, mean = 5, sd = 2),
rnorm(100, mean = 7, sd = 2))
```

```
9   )
10
11  # Gráfico de boxplots com camadas
12  ggplot(data, aes(x = grupo, y = valor, fill = grupo)) +
13  geom_boxplot() +
14  labs(title = "Box Plots por Grupo",
15  x = "Grupo", y = "Valor") +
16  theme(
17  plot.title = element_text(size = 24, face = "bold", hjust =
      0.5),
18  axis.title = element_text(size = 20, face = "bold"),
19  axis.text = element_text(size = 16)
20  )
```

Listagem 4.1: gráfico de Boxplot com Camadas.

Camadas no ggplot2

O pacote `ggplot2`, amplamente adotado na visualização estatística em R, implementa o paradigma de camadas inspirado na gramática dos gráficos. Cada camada adicionada por meio do operador `+` corresponde a uma nova instrução visual, como caixas, pontos, linhas ou textos.
Componentes do gráfico (ver Listagem 4.1):

- `geom_boxplot()` cria a camada estrutural principal, com quartis e mediana;
- O argumento `fill = grupo` define a coloração das caixas por grupo;
- `labs()` adiciona o título e os rótulos dos eixos;
- `theme()` personaliza os elementos visuais como fontes e alinhamento.

Cada camada é independente e pode ser manipulada separadamente, o que garante flexibilidade analítica e clareza visual mesmo em situações mais complexas.

4.2 VIESES E LIMITAÇÕES COGNITIVAS

A visualização de dados depende, em última instância, da percepção humana. Isso implica reconhecer que nossos sentidos e interpretações estão sujeitos a vieses e limitações cognitivas. A seguir, discutimos alguns dos mais recorrentes, e perigosos, na análise gráfica.

Comparação por área ou volume: armadilha visual

O ser humano tem dificuldade em comparar superfícies com precisão. Quando dois círculos, quadrados ou figuras tridimensionais são usados para representar quantidades, a diferença percebida pode ser maior ou menor do que a real. Veja um exemplo deste contraste na Figura 4.2.1.

- **Exemplo clássico:** gráficos de pizza com setores visualmente parecidos, mas com valores muito distintos.
- **Boa prática:** prefira representações baseadas em comprimento (barras horizontais ou verticais).

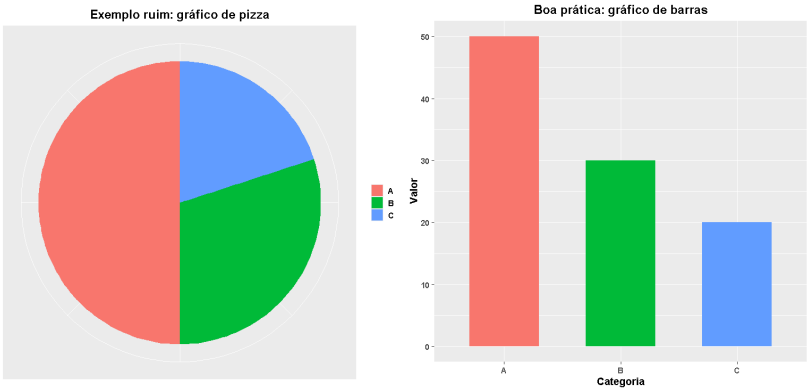


Figura 4.2.1: o gráfico de pizza da esquerda tende a distorcer a percepção de proporções, já gráfico de barras ordenado facilita a comparação visual precisa.

Escalas distorcidas e Viés Perceptivo

Truncar o eixo Y ou manipular escalas (como usar escalas logarítmicas ou deslocar o ponto de equilíbrio visual) pode acentuar ou atenuar a percepção de variação. A inclinação da curva e o posicionamento dos objetos influenciam fortemente a interpretação, ainda que os valores não tenham sido alterados.

- **Armadilha comum:** modificar o eixo Y para destacar oscilações irrelevantes ou, inversamente, suavizar variações relevantes.
- **Boa prática:** sempre explicitar e justificar a escolha de escalas não lineares, truncamentos ou deslocamentos, pois sua interpretação depende do contexto e do propósito da análise.

Por exemplo, a Figura 4.2.2 pode sugerir instabilidade ou equilíbrio, oscilação acentuada ou tendência modesta. O erro não está na escolha da escala, mas em omitir seus efeitos ou induzir conclusões visuais sem ancoragem analítica.

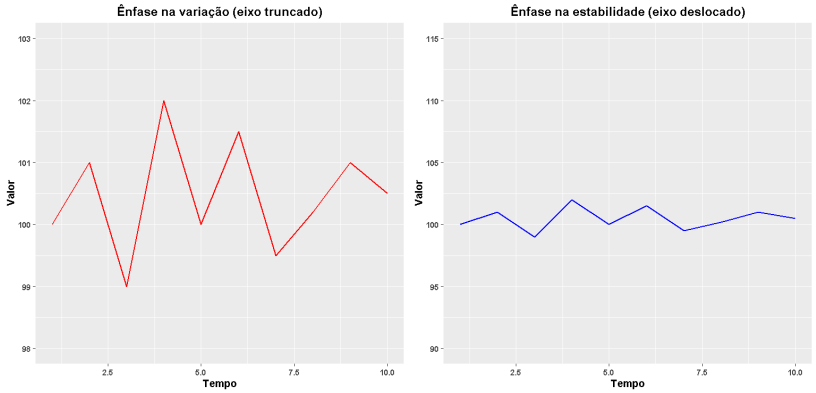


Figura 4.2.2: dois gráficos com os mesmos dados, porém com diferentes escalas no eixo Y. À esquerda, o eixo truncado exagera a percepção de variação; à direita, o deslocamento da escala suaviza visualmente as oscilações. Ambas as escolhas são válidas, desde que justificadas e explicitadas no texto. A manipulação da escala pode ser perigosa quando não há ancoragem analítica clara.

Pareidolia Estatística (ver padrão onde não há):

O cérebro humano é muito bom em detectar padrões, até onde eles não existem. Essa tendência é útil na vida cotidiana, mas perigosa em análise de dados. Correlações espúrias, flutuações aleatórias e agrupamentos ilusórios são facilmente interpretados como fenômenos reais[12].

- **Armadilha:** nuvens de dispersão que "sugerem" tendência sem evidência estatística robusta.
- **Boa prática:** combinar visualização com testes formais, evitando conclusões apressadas.

[12] A explicação da Figura 4.2.3 é simples, à medida que mais pais nomearam suas filhas Annabelle após a boneca assustadora nos filmes de terror, eles inadvertidamente desencadearam uma onda de energia de alguma forma atraiu OVNIs para a Carolina do Sul. Visite `https://www.tylervigen.com` 🧢+🪆.

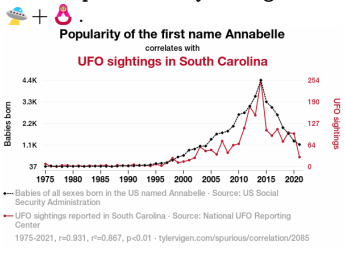


Figura 4.2.3: um exemplo de correlação espúrias. Fonte: `https://www.tylervigen.com`

Cores, Contraste e Agrupamento Visual

Cores mal escolhidas podem induzir a agrupamentos inexistentes ou mascarar diferenças reais. Além disso, contrastes inadequados prejudicam a legibilidade e tornam os gráficos inacessíveis a pessoas com daltonismo.

- **Armadilha:** uso de degradês com mesma luminância, que dificultam a distinção de categorias.
- **Boa prática:** empregar paletas perceptualmente uniformes e acessíveis (como Okabe–Ito).

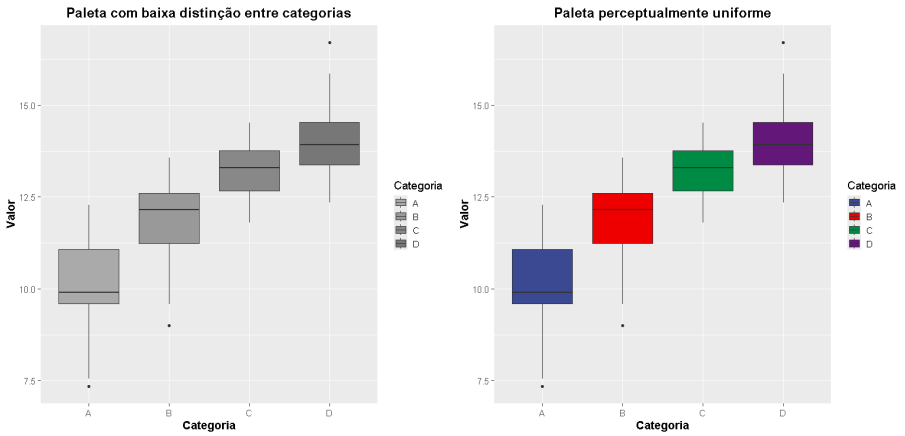


Figura 4.2.4: cores mal escolhidas podem induzir agrupamentos falsos ou mascarar padrões reais. A paleta da esquerda usa tons com luminância similar; a da direita utiliza uma paleta perceptualmente uniforme e acessível.

O viés do eixo X linear

Muitos fenômenos reais não se distribuem uniformemente no tempo ou espaço. Usar eixos X com marcações equidistantes em séries temporais pode induzir a uma percepção equivocada de ritmo ou frequência.

- **Armadilha:** comparar intervalos irregulares como se tivessem a mesma duração.
- **Boa prática:** respeitar a granularidade real dos dados ao construir o eixo X.

Vieses perceptivos não são exceções: são a regra. A responsabilidade do pesquisador é antecipá-los, mitigá-los e, quando inevitáveis, explicitá-los. Ignorar as limitações cognitivas é transferir a falha da análise para o leitor. Isso não é justo nem ético.

Interatividade – Usar ou Evitar

Recursos interativos ampliam a capacidade de exploração e detalhamento em visualizações de dados. Ferramentas como *tolltip*(hover), *zoom*, filtros dinâmicos, *sliders* e *drill-downs* permitem que o leitor acesse camadas adicionais de informação sob demanda, inspecione valores específicos e identifique outliers com precisão — o que seria impraticável em gráficos estáticos.

A interatividade é particularmente valiosa em análises exploratórias e dashboards, onde o objetivo é investigar padrões, relações ou anomalias nos dados. Ao permitir a navegação direta sobre pontos do gráfico com tooltips, torna-se insubstituível para a inspeção fina de elementos específicos.

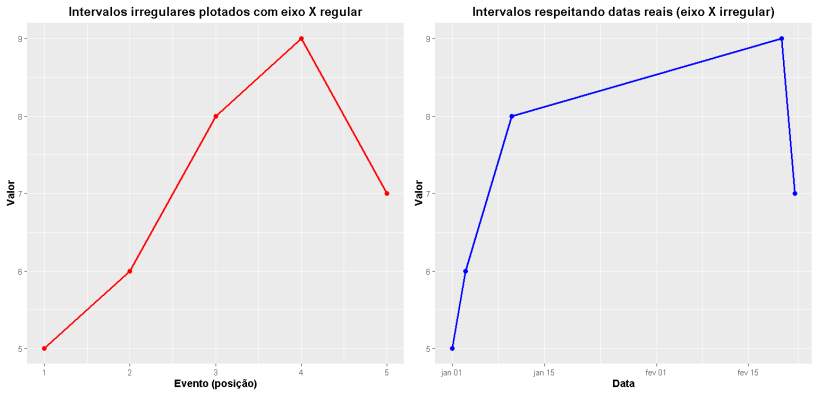


Figura 4.2.5: eixos com espaçamento uniforme pode sugerir periodicidade ou ritmo falso. O gráfico a direita respeita a granularidade temporal real dos dados.

Contudo, ela impõe uma carga cognitiva adicional e exige domínio da interface. Em materiais estáticos (como artigos, relatórios ou apresentações), qualquer dependência de interação compromete a compreensão e pode excluir parte do público.

- **Quando usar:** em *dashboards*, painéis exploratórios, sistemas de apoio à decisão e apresentações digitais onde a navegação é controlada pelo usuário.
- **Quando evitar:** em materiais estáticos (slides, relatórios em PDF, gráficos para impressão), nos quais a informação essencial deve ser imediatamente visível e autossuficiente.
- **Boa prática:** combine versões: gráficos interativos para exploração e estáticos bem formatados para comunicação. Nunca use a interatividade para esconder a complexidade ou compensar deficiências visuais.

Interatividade é uma lupa, não um atalho. Enriquecer a análise com ela é desejável, transferir ao leitor a responsabilidade por descobrir o essencial, não.

4.3 CHECKLIST DE REVISÃO CRÍTICA DE GRÁFICOS E FIGURAS

A visualização de dados exige o mesmo rigor metodológico que a redação científica. Um gráfico não é um ornamento — é uma proposição visual que precisa ser precisa, relevante e verificável. Em engenharia e ciência, gráficos mal projetados não são apenas feios — são perigosos.

A seguir, um checklist crítico para revisão de gráficos antes de publicar, apresentar ou compartilhar. Use-o como um roteiro de validação final, não como um lembrete genérico.

Checklist Final: Revisão Crítica de Gráficos e Figuras

- **Simplicidade:** Há elementos visuais redundantes, decorativos ou apenas bonitos? Remova-os. Se não comunica, atrapalha.
- **Clareza:** O gráfico pode ser entendido por alguém da área sem legenda adicional? O que você está tentando dizer está evidente à primeira leitura?
- **Precisão:** A escala está manipulando visualmente os dados? Cortou o eixo y em um gráfico de barras? O gráfico respeita proporções?
- **Eficiência:** O gráfico entrega muito ou pouco? Um gráfico com três barras não deveria ser uma tabela? Um gráfico 3D rotacionado transmite algo que um 2D não resolveria melhor?
- **Estética com função:** O layout facilita a leitura ou está apenas "bonito"? As cores têm contraste suficiente? Há consistência visual com os demais gráficos do material?
- **Consistência terminológica:** Os rótulos, títulos e eixos usam os mesmos termos do texto técnico? Está escrito "Mbps"no gráfico e "Mb/s"no texto?
- **Armadilhas cognitivas:** O gráfico sugere correlações espúrias, padrões ilusórios ou leva o leitor a generalizações não sustentadas? Use anotações para evitar falsas inferências.
- **Referência cruzada:** A figura foi mencionada no corpo do texto? O leitor foi conduzido à interpretação correta no ponto em que ela aparece?
- **Legenda informativa:** A legenda está clara, precisa e sem ambiguidade? Não tenha medo de usar mais de uma linha. Legenda curta demais raramente é uma virtude.
- **Inclusividade visual:** O gráfico é legível para leitores com daltonismo? As fontes têm contraste suficiente para visualização em projetores ruins ou luz ambiente?
- **Criatividade com propósito:** O gráfico explora formas visuais não convencionais que realmente ajudam a compreender o fenômeno? Criatividade é bem-vinda — desde que amplifique, não distraia.
- **Consistência visual global:** O estilo das figuras (cores, fontes, legendas, margens) deve estar consonante com os demais gráficos do trabalho? Padrões visuais coerentes reforçam credibilidade e leitura fluida.

Gráficos bem projetados são argumentos visuais. Eles antecipam dúvidas, evitam ruídos e reforçam interpretações corretas. Um gráfico ruim, ao contrário, cria falsas certezas com o verniz da credibilidade estatística[13].

[13]Lembre-se! Depois de ler diversos artigos com eixos truncados, legendas ambíguas e cores aleatórias, o avaliador já está no modo "*strong reject*". A partir dali, cada gráfico mal feito não só incomoda, mas irrita. O revisor vira um detector de erro enviesado com 100% de sensibilidade e 50% de falsos positivos 📉 🧐 ❌.

5 Descrição de dados

A descrição de dados muitas vezes recebe pouca atenção, sendo deixada de lado por aqueles mais ansiosos. A descrição dos dados, no entanto, desempenha várias funções importantes: ajuda-nos a escapar de algumas armadilhas e nos ajuda a avaliar a validade, fiabilidade e sanidade das medidas[14].

Para construir figuras úteis, precisamos conhecer a estrutura subjacente dos dados. Tentaremos ilustrar como as descrições dos dados podem ajudar rapidamente a avaliar a conexão entre nossas perguntas variáveis, *data sets* etc.

[14] O capítulo 5 suas Figuras e códigos foram fortemente baseados no trabalho publicado em [6]. O autor tomou a liberdade para fazer uma atualização dos códigos e figuras visando o emprego de *libraries* mais modernas e ferramentas mais flexíveis.

5.1 TIPOS DE VARIÁVEIS

Uma variável descreve uma característica particular de uma pessoa, lugar ou coisa. Um *data set* (conjunto de dados) é uma coleção de duas ou mais variáveis. Vamos a algumas definições:

Variável contínua: uma variável que registra contagens ou valores. Variáveis contínuas pode assumir qualquer valor entre o mínimo e o máximo.

Variável categórica: uma variável que indica diferenças de espécie. Variáveis categóricas registrar estados distintos de ser ou características das pessoas, lugares ou coisas.

Variável categórica ordenada: uma variável categórica cujas categorias podem ser ordenadas de acordo com uma dimensão subjacente (por exemplo, partes de um espectro conservador-liberal; escolaridade de completar o ensino fundamental até obter o doutorado; diferente classes de crimes, etc[15] .

[15] O exemplo presente na Figura 5.1.1 demonstra por que devemos resistir a aceitar os dados como eles chegam dados. Os indivíduos ou organizações que constroem os dados que usamos provavelmente têm diferentes agendas e diferentes hipóteses. Como resultado, as categorias são colocadas juntas em uma maneira que é fácil ou interessante para aquela análise deles, mas não para a nossa.

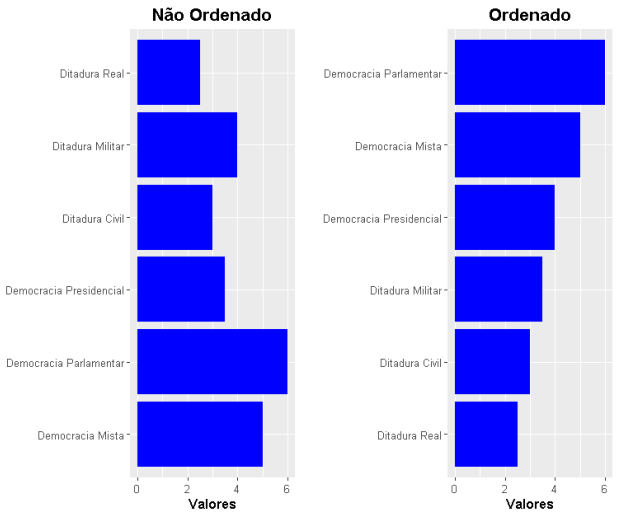


Figura 5.1.1: exemplo de clareza contrastando um gráfico com as variáveis ordenadas e desordenadas.

Conjunto de dados: uma coleção (uma série de colunas e linhas) de números ou categorias que especificam valores ou características para um grupo de pessoas, lugares ou coisas. Muitas vezes, cada coluna representará um característica e cada linha representará uma pessoa específica, lugar ou coisa.

Se houver uma ordem reconhecível, organizar as visualizações dos dados para refletir essa ordem pode ser útil. Variáveis cujas categorias podem ser ordenadas de acordo para alguma dimensão são chamadas de variáveis categóricas ordenadas, ver exemplo na Figura 5.1.1 .

5.2 FORMA E INTERVALO

Familiarizar-se com seus dados é importante por vários motivos. nos leva a melhores perguntas, nos ajuda a escolher sumários apropriados, e principalmente, nos ajuda a identificar erros. Nesta seção abordaremos duas características dos dados – forma e intervalo – Precisamos primeiro conhecer a forma dos dados para determinar o que resumos e modelos a serem usados. Em segundo lugar, precisamos conhecer a escala do problema. Conhecer a forma e o intervalo de uma variável são maneiras de economizar muito tempo e esforço.

Assim é importante refletor sobre algumas questões: nossa variável oscila entre seu mínimo e máximo? Os valores estão distribuídos uniformemente? Os casos se reúnem mais perto de o mínimo, mediana ou máximo? Para responder a essas perguntas, examinamos o forma ou distribuição dos dados. Nesse sentido, as palavras "forma" e "distribuição" são sinônimos. Eles descrevem uma característica dos dados que indica onde a maioria dos casos estão em relação ao intervalo de valores de uma variável. Compreender a forma de nossos dados são importantes porque alguns resumos simples dos dados podem ser influenciados por diferentes formas ou distribuições.

Primeiro, com a função `data.frame()` criarei um conjunto de dados chamado mydata que possui duas variáveis: um é distribuída "normalmente" usando a função `rnorm()`, e o outro tem distribuição "exponencial" (usando a função `rexp()`[16]. A curva normal é definida pela minha escolha de observações (1.000.000), a média (0), e o desvio padrão (1). Para sobrepor os dois gráficos usarei a função `melt()`.

[16]Para sobrepor gráfico de densidade a um histograma use a opção `..ncount..` no histograma e a opção `..scaled..` no gráfico de densidade.

Para a distribuição normal, a maioria das observações da variável estão próximas de x=0 , com muito poucas se aproximando de 4 e −4. A distribuição exponencial também tem a maioria das suas observações próximas de 0. e algumas observações próximas de 5. Em termos de forma, dizemos que a curva azul se aproxima de uma curva em forma de sino: é simétrica, pois metade da distribuição (valores acima de zero) reflete a outra metade (aqueles abaixo de zero).). A variável destacada em vermelho representa a distribuição exponencial. A curva vermelha não é simétrica, contendo observações acima de 7 e 8. simétrica, dizemos que a distribuição destacada em vermelho está desviada. o histograma da participação eleitoral está bastante próximo da curva ideal em forma de sino. Consideraríamos que a distribuição da participação eleitoral

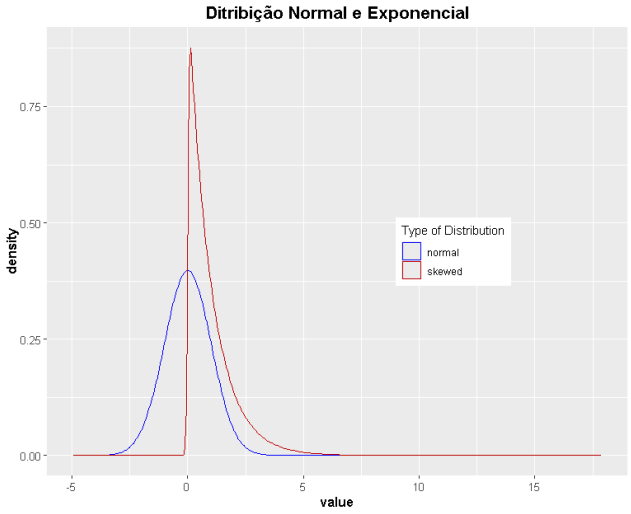


Figura 5.2.1: distribuições normal e exponencial.

seria normalmente distribuída com exceção de um valor bem inferior a 25%. Consideremos agora o formato da mortalidade infantil no mundo: o número de mortes em cada país por 1.000 nascidos vivos.

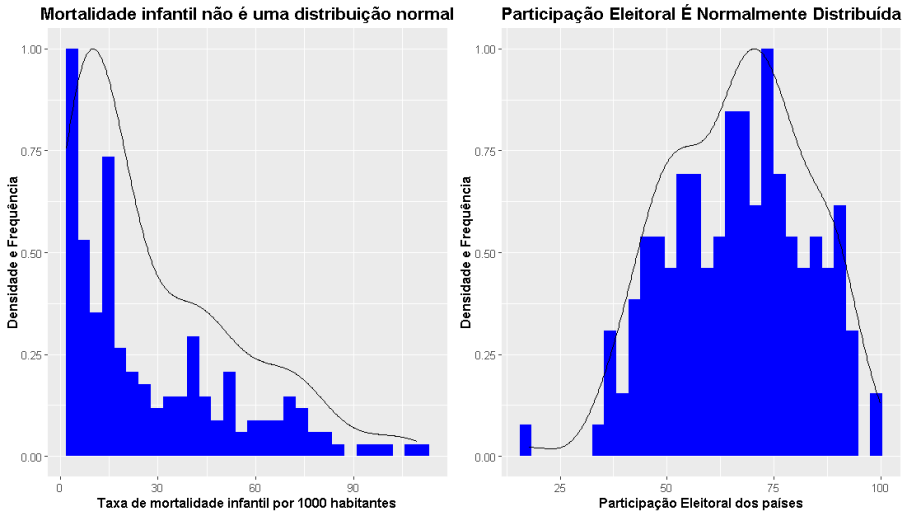


Figura 5.2.2: exemplos de distribuições: Normal (Participação eleitoral) e Não Normal (taxa de mortalidade por 1000 habitantes dos países).

5.3 AMPLITUDE DO DADO

A amplitude de uma variável é simplesmente a diferença entre seu valor mínimo e máximo. Entender a amplitude dos dados pode ajudar a identificar problemas que podem não valer o esforço de análise. Por exemplo, se queremos explicar o sucesso dos estudantes, podemos desenvolver um modelo considerando horas de estudo, número de cursos anteriores, horas de sono, etc. No entanto, se descobrirmos que a diferença entre a menor e a maior nota é menor que 3 pontos, isso indicaria que a análise não é relevante, pois a variação é muito pequena.

A moral da história: uma variável dependente com uma pequena amplitude pode sinalizar que não há muito a ser analisado. Por exemplo, ao analisar a porcentagem da população adulta de um estado que concluiu o ensino médio, é importante verificar a amplitude dos resultados antes de investir recursos na investigação.

5.4 REFINANDO O DADO

O *sumário estatístico* é uma das primeiras ferramentas no processo de refino e compreensão de dados. Ao condensar informações essenciais como média, mediana, desvio padrão, amplitude, assimetria e curtose, como na Figura 5.4.1, o sumário permite uma avaliação inicial da distribuição e da variabilidade dos dados sem a necessidade de examinar cada valor individualmente.

Essa visão panorâmica é fundamental para:

- Detectar tendências centrais e dispersões.
- Identificar indícios de assimetria ou caudas longas.
- Avaliar a presença de valores extremos (*outliers*).
- Comparar distribuições entre diferentes grupos ou experimentos.
- Apoiar a escolha de transformações (ex.: logarítmica) ou métodos estatísticos adequados (paramétricos ou não-paramétricos).

O sumário, quando apresentado em conjunto com a visualização gráfica, fornece um diagnóstico robusto que orienta decisões sobre filtragem, transformação e análise subsequente. Portanto, antes de aplicar técnicas complexas de modelagem ou inferência, é essencial começar com essa etapa descritiva para garantir clareza e relevância na interpretação.

Já para refinar e melhorar a visualização do dado, há múltiplas ferramentas possíveis, o autor em [6] preconiza e exemplifica com o `dplyr`. Este que vos escrever prefere o `sqldf` do R, que é extremamente útil para gerenciar e manipular dados e como bônus vc aprende cláusulas SQL, que são eficientes e rápidas em muitos casos de tratamento de dados. Muitas perguntas podem ser respondidas através destas *queries*.

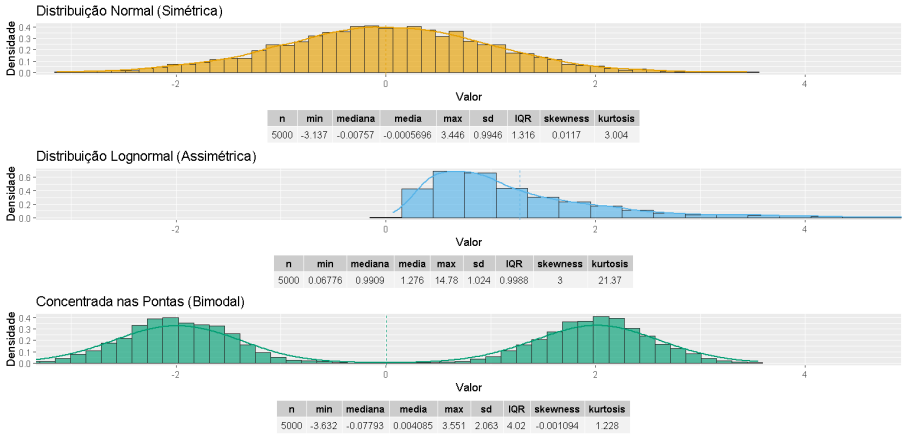


Figura 5.4.1: Comparação entre três distribuições distintas acompanhadas de seus respectivos sumários estatísticos. A tabela abaixo de cada histograma destaca as principais métricas descritivas, permitindo avaliar rapidamente diferenças de tendência central, dispersão, assimetria e curtose.

No código presente na Listagem 5.1 foi empregado o `sqldf` para filtrar apenas os casos em que a variável pid7 não tem dados ausentes (NA) e para eliminar os respondentes que responderam "not sure"e em seguida, calculei o valor médio dos sentimentos pela força policial para cada categoria político-partidária presente no campo do `pid7` e as relacionei. A listagem mostra o resultado [17].

```
>dt <- sqldf("
SELECT pid7,
ROUND(AVG(ftpolice), 2) AS ftpolice_mean
FROM nes
WHERE pid7 != 'NA' AND pid7 != 'Not sure'
GROUP BY pid7
")
> dt
pid7 ftpolice_mean
1       Independent       62.5821
2     Lean Democrat       61.4112
3   Lean Republican       73.7798
4   Strong Democrat       58.0441
5 Strong Republican       80.7261
6     Weak Democrat       62.6879
7   Weak Republican       76.3246
> xtable(dt)
% latex table generated in R 4.3.3 by xtable 1.8-4 package
% Sun Jun  9 17:26:55 2024
\begin{table}[ht]
  \centering
  \begin{tabular}{rlr}
    \hline
    & pid7 & ftpolice\_mean \\
    \hline
```

[17] o campo `ftpolice` representa um termômetro de sentimentos em relação à polícia, onde os respondentes avaliam seu sentimento de 0 a 100. Neste caso, o campo indica o nível de apoio ou sentimento positivo em relação à polícia, com 0 representando o menor apoio e 100 representando o maior apoio. Já o `Pid7` é um campo de identificação político-partidária com 7 categorias

```
26      1 & Independent & 62.58 \\
27      2 & Lean Democrat & 61.41 \\
28      3 & Lean Republican & 73.78 \\
29      4 & Strong Democrat & 58.04 \\
30      5 & Strong Republican & 80.73 \\
31      6 & Weak Democrat & 62.69 \\
32      7 & Weak Republican & 76.32 \\
33      \hline
34    \end{tabular}
35  \end{table}
36
```

Listagem 5.1: comando para instalar e carregar a biblioteca `easystat`.

Também é possível gerar tabelas em formato latex no R como visto na linha 17 da Listagem 5.1. a Tabela 1 foi criada a partir da saída do comando com o `xtable`.

Tabela 1: sentimentos pela polícia *vs.* ideologia política.

	pid7	ftpolice_mean
1	*Independent*	62.58
2	*Lean Democrat*	61.41
3	*Lean Republican*	73.78
4	*Strong Democrat*	58.04
5	*Strong Republican*	80.73
6	*Weak Democrat*	62.69
7	*Weak Republican*	76.32

Depois desta breve viagem na descrição de dados vamos reforçar alguns conceitos que ficaram implícitos nesta jornada. **Medição:** traduzir hipóteses em indicadores significativos é um desafio central na análise de dados em telecomunicações. Desde o início, é fundamental desenvolver medidas que sejam válidas e confiáveis para evitar conclusões equivocadas sobre o desempenho ou o comportamento de um sistema.

Validade: refere-se ao grau em que os dados representam de forma fiel o conceito que desejamos medir. Por exemplo, ao investigar a relação entre qualidade de serviço (QoS) e satisfação do usuário, é essencial que métricas como *latência*, *jitter* e *taxa de perda de pacotes* reflitam efetivamente a experiência real do usuário. Em outro caso, medir a eficiência espectral deve envolver indicadores adequados como bits/s/Hz e considerar o contexto da modulação e do esquema de acesso ao meio.

Confiabilidade: diz respeito à precisão e consistência das medidas. Uma medida confiável de potência de sinal recebida (RSSI) ou relação sinal-ruído (SNR) deve produzir resultados semelhantes quando o teste é repetido nas mesmas condições e com o mesmo equipamento. Sistemas de medição mal calibrados ou variações no ambiente de teste podem comprometer a confiabilidade, como no caso de medições de *throughput* em redes sem fio feitas em diferentes horários, sujeitos a congestionamento variável.

6 Medidas de Tendência Central e Dispersão

Duas perguntas devem ser respondidas no processo de descrição do dado. Qual é o caso típico do dado e Quão típico ele é? A primeira questão diz respeito ao centro da tendência e a segunda diz respeito à dispersão[18].

Nossas escolhas diárias são muitas vezes definidas pelo caso típico, por exemplo, o que vestir dependerá fortemente da temperatura típica ou média da estação. Diagnósticos são feitos com base em medidas de tendencia central. Os medicamentos são prescritos com base no batimento cardíaco do paciente. está abaixo ou acima da média.

O mesmo acontece com a dispersão. Por exemplo, ao calcular o tempo até o trabalho, podemos decidir não confiar na média, mas na dispersão daí saímos mais cedo.

A escolha da medida apropriada depende do tipo de dado por exemplo da presença ou não de valores extremos. A média é sensível a extremos, enquanto a mediana é mais estável nesses casos. Em dados categóricos, a moda identifica a categoria mais frequente, útil para entender preferências em situações como eleições. A escolha da medida correta causa grande impacto nas interpretações e decisões baseadas nos dados.

6.1 MODA

Para identificar a moda no conjunto de dados NES (`http://edge.sagepub.com/brownstats1e`), observa-se que a categoria com o maior número de respostas é "Strong Democrats", com quase 300 respostas, superando a próxima categoria, "Independents", por cerca de 100 respostas. Isso indica que "Strong Democrats"é a moda. Essa informação é relevante para estratégias políticas e pode influenciar políticos.

[18] Os Capítulos 6 e 9, suas figuras e exemplos foram baseados no trabalho publicado em [6] de David Brown (ver Figura 6.0.1. 👇)

Figura 6.0.1: Mr. Brown Tem mais de 20 anos de experiência como avaliador estatístico na MHRA e, antes do Brexit, foi membro do grupo de trabalho de bioestatística e do grupo de trabalho de aconselhamento científico da EMA. Ele fez parte do grupo que formulou a orientação recentemente publicada da MHRA sobre dados do mundo real.

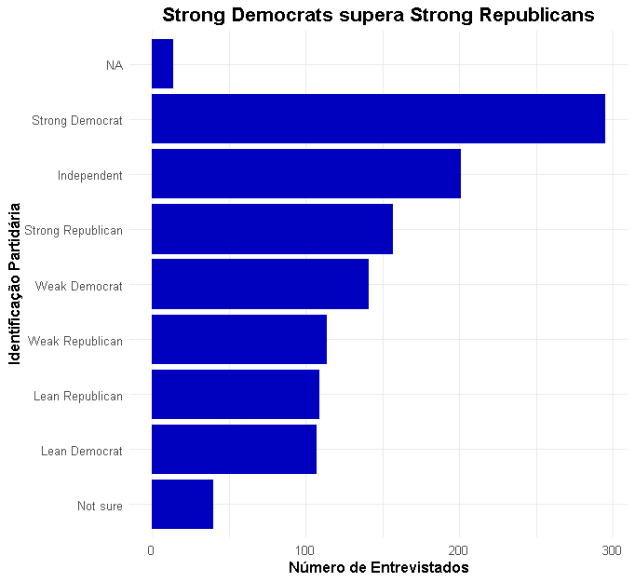


Figura 6.1.1: A moda é a categoria mais comum, neste caso: *strong democrat*.

6.2 MÉDIA

A média é um resumo numérico frequentemente usado devido à sua facilidade de cálculo e apelo intuitivo, sendo aplicada em diversas áreas, desde o desempenho de um jogador até como conclusão da competência de um curso com a média de alunos formados. Para calcular a média, soma-se todos os valores do conjunto de dados e divide-se pelo número de observações. Seu cálculo no é simples, mas não é trivial. Observe o comando `mean(world$homicide, na.rm = TRUE)`. A cláusula `na.rm = TRUE` tem o cuidado de remover os valores ausentes, para não poluir o cálculo.

A apresentação de gráficos também pode conter armadilhas, em um histograma e em um boxp-plot pode ser difícil perceber onde está a média. Já que essas ferramentas não apresentam explicitamente a média. De modo que pode ser interessante facilitar a vida do leitor, forçando sua presença identificando-a explicitamente.

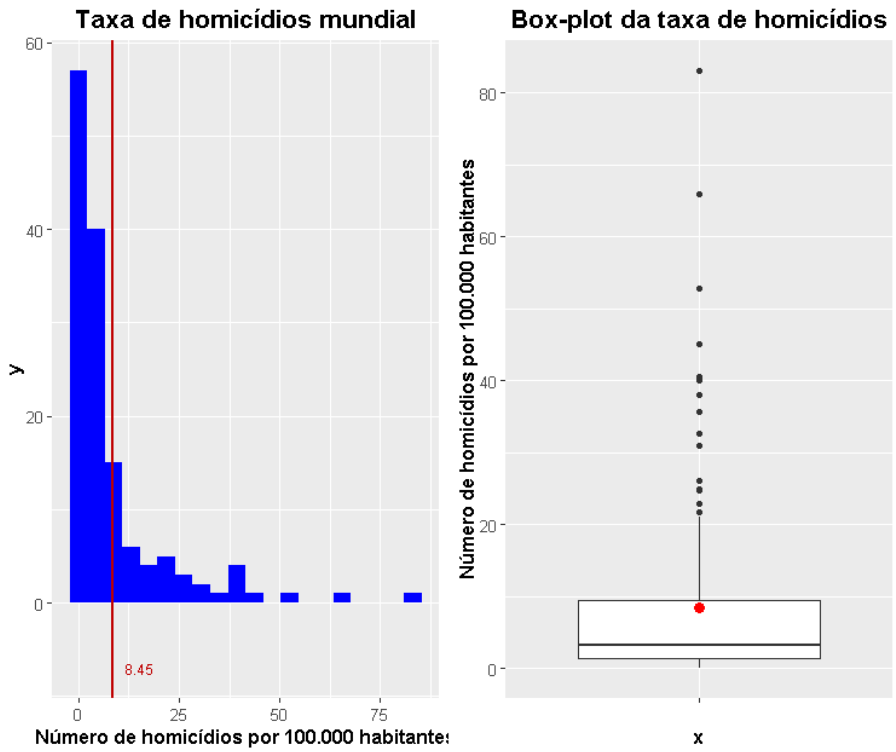


Figura 6.2.1: histograma e box-plot modificados para apresentação da média (em vermelho).

6.3 MEDIANA

A mediana sofre um preconceito indevido e e ela é muito subestimada, pois serve um importante função. A mediana deve ser usada quando existem valores extremos na variável. A mediana é simplesmente o caso intermediário. Por tanto ao invés de perguntarmos sempre qual é a média, devíamos perguntar pergunta "Qual é o caso típico? Quando existem valores extremos, a mediana pode dar a melhor representação do que é o típico.

O histograma na Figura 6.2.1 sugere que podemos querer usar a mediana em vez da média ao descrever a tendência central para as taxas de homicídio. A mediana representa o caso do meio ou o percentil 50. Ela é calculada organizando os dados em ordem do menor para o maior valor e, então, encontrando o valor do caso do meio. Se o conjunto de dados contém um número ímpar de casos, a mediana[19]. Comando: `median(world$homicide, na.rm = TRUE)` A mediana é o ponto de dados do meio, com um número igual de casos de cada lado. Se os dados contêm um número par de casos, a mediana é a média dos dois valores do meio na lista ordenada.

[19] A mediana pode ser calculada usando a função `median()` no R. Note que, embora o R calcule facilmente a média e a mediana com as funções simples *mean()* e `median()`, usar o comando `mode()` não calculará a moda. O comando `mode()` irá informar, ao invés disso, como uma variável foi armazenada no R (não sendo muito útil para entender a tendência central). A maneira mais fácil de descobrir qual categoria possui mais casos é gerar um gráfico de barras ou uma tabela de frequência.

Número ímpar de observações $= (1, 2, 3, 4, 5, 6, 7, 8, 9) = 5$

Número par de observações $= (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) = 5.5 = (5 + 6)/2$

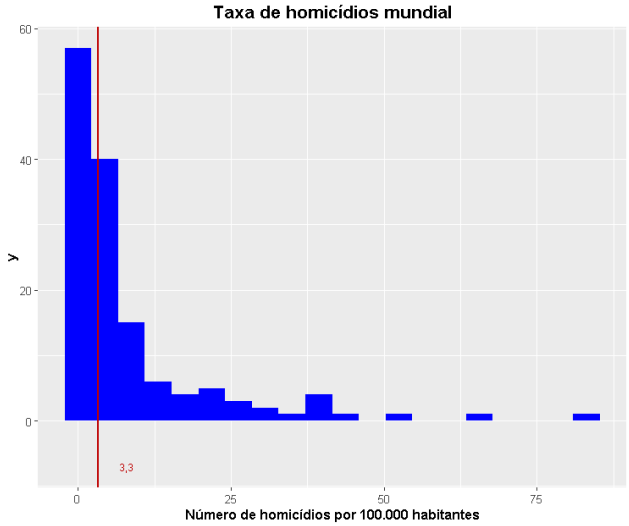


Figura 6.3.1: histograma com mediana representada.

A Figura 6.3.2 mostra dois conjuntos de dados similares, mas o Data2 [6] contém *outliers*, o que move a média. O gráfico também mostra que a mediana é pouco alterada.

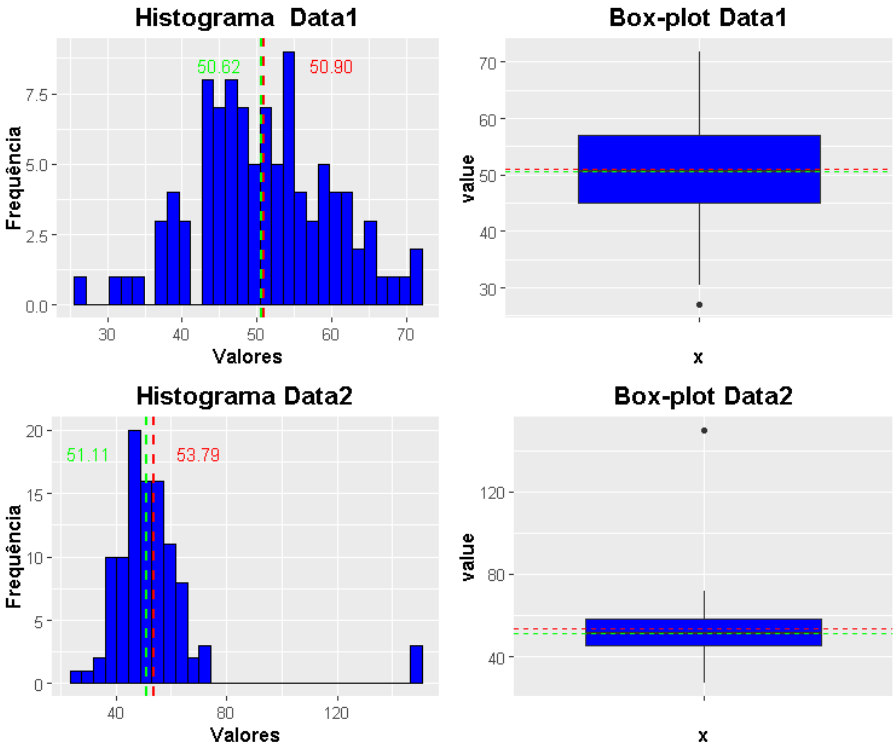


Figura 6.3.2: presença do *outlier* em Data2 deslocando a média (em vermelho) para a direita.

6.4 MEDIDAS DE DISPERSÃO – UMA NOVA ABORDAGEM

Depois de conhecer a tendência central de uma variável, surge a questão sobre a distância típica em relação à média. Não é suficiente saber se uma dada medida está acima ou abaixo da média. Gostaríamos de saber como essas medidas se comparam com o restante da população ou da amostra. Por exemplo, saber que a nota em um exame está acima da média é importante, mas é crucial saber se isso coloca você entre os melhores 10%, 20% ou 40%, pois isso pode significar a diferença entre a aprovação ou não. Conhecer a dispersão fornece essa perspectiva.

6.4.1 Amplitude

A amplitude de uma variável é a distância entre seu valor mínimo e máximo. Essa medida será útil para tomar decisões ou desenvolver políticas com base nos cenários do melhor, ou pior caso. Por exemplo, se a amplitude dos resultados possíveis de um novo medicamento inclui a morte, podemos decidir que seu uso é inaceitável.

6.4.2 Intervalo Inter Quartil

[20]Tanto o Desvio Padrão quanto o IIQ são medidas de dispersão, mas quando usar uma e quando usar outra? De cara podemos dizer que o IQQ, por usar a metade central dos dados, ele se torna quase imune a *outliers*. O Interquartil também não toma nenhum pressuposto do dado. O SD tipicamente serve para distribuições próximas da normal.

O intervalo interquartil (IIQ)[20], definido como a diferença entre o 75º e o 25º percentil, descreve a extensão da metade central dos dados. De forma mais precisa, esse intervalo é calculado identificando os valores correspondentes aos quartis primeiro e terceiro. Por exemplo, em uma amostra de 100 observações, o valor na 25ª posição após a ordenação crescente constitui o 25º percentil, enquanto o valor na 75ª posição representa o 75º percentil. O intervalo interquartil, portanto, é uma medida de dispersão entre esses dois pontos percentuais. Esses valores podem ser obtidos e reportados através da execução do comando `summary()`, mostrada na listagem 6.1 em um ambiente de programação estatística

```
1 >summary(states$infant)
2 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
3 4.520   5.938   6.785   6.981   7.878  11.470
```

Listagem 6.1: comando para exibir o sumário da variável `infant` presente no *dataframe* `states`.

6.4.3 Desvio Padrão

[21]Em contraste com a variância, o DP, que é a sua raiz quadrada, é expresso nas mesmas unidades dos dados originais. Isto confere ao desvio padrão uma superioridade prática significativa, pois facilita a interpretação e comparação dos resultados. Ao utilizar a mesma unidade da variável analisada, o desvio padrão permite uma compreensão mais direta e intuitiva da variabilidade dos dados. Por exemplo, se estamos considerando a altura de uma população em centímetros, o desvio padrão dessa altura também será expresso em centímetros, o que facilita entender o quanto de variação existe em relação à média da altura.

O desvio padrão[21] é uma medida de dispersão frequentemente utilizada, cuja popularidade decorre de dois atributos: sua simplicidade conceitual e sua unidade. Primeiramente, o desvio padrão de uma variável é expresso nas unidades originais da variável (metros, gramas, anos, porcentagens, etc.). Para ilustrar, consideraremos duas variáveis: as taxas de mortalidade infantil e a porcentagem de assentos legislativos ocupados por mulheres em um estado. No R o comando para calcular o desvio padrão (sd) é `sd(states$femleg, na.rm =TRUE)`.

Com o aumento do desvio padrão de 1 para 3, os dados tornam-se progressivamente mais dispersos. Isso significa que quanto maior o desvio padrão, maior a variabilidade ou dispersão dos dados em torno da média. Em termos práticos, um desvio padrão maior indica que as observações variam mais amplamente e estão menos concentradas em torno da média. Esse tipo de análise é fundamental para entender a consistência ou a volatilidade de um conjunto de dados, sendo crucial para aplicações em muitas áreas, incluindo finanças, ciências naturais, e engenharia.

7 Transformações de dados

A transformação de dados e a conversão de escalas são processos úteis na visualização e representação de dados, especialmente quando se lida com variáveis que abrangem amplos intervalos de valores ou que possuem distribuições não lineares. Esses métodos permitem que as informações sejam apresentadas de maneira mais clara e comparável, facilitando a interpretação dos padrões subjacentes e das relações entre as variáveis. Ao ajustar a escala dos dados, é possível

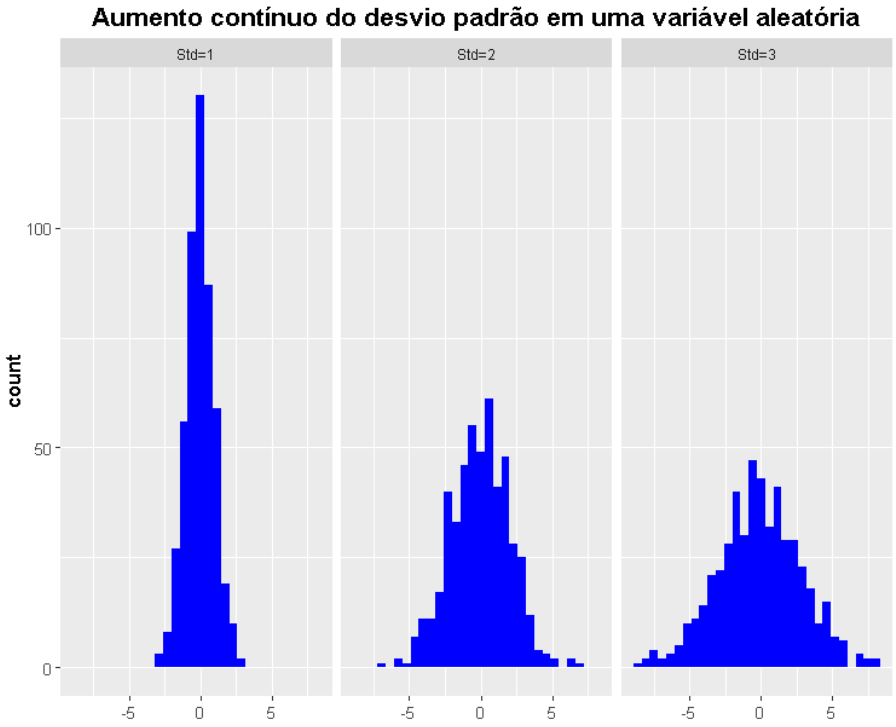


Figura 6.4.1: aumento do desvio padrão e consequentemente da dispersão da variável em torno da média.

destacar variações importantes que poderiam ser obscurecidas em uma representação linear simples, além de permitir uma análise mais equilibrada e justa, evitando que *outliers* ou grandes disparidades de escala distorçam a percepção dos resultados.

7.1 EIXOS LINEARES

Na representação gráfica de dados, a transformação linear dos eixos pode ser essencial para manter a integridade visual ao converter unidades de medida. Por exemplo, ao representar a temperatura, a transformação de graus Fahrenheit para Celsius mantém a forma das curvas intacta, sem alterar a relação entre as variáveis. Isso é importante para garantir que a análise visual permaneça consistente e as conclusões derivadas das figuras sejam precisas e comparáveis, independentemente da unidade de medida utilizada.

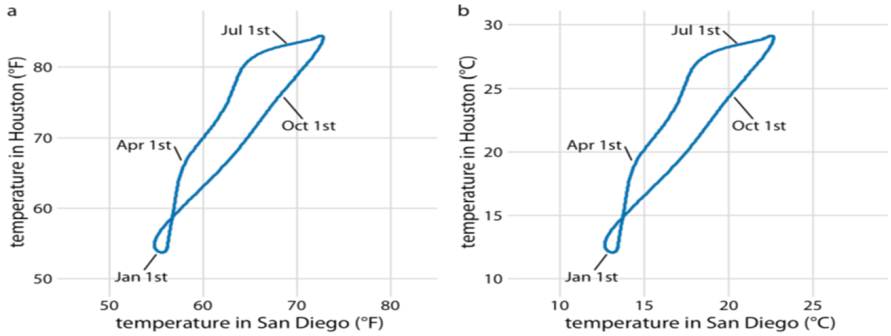


Figura 7.1.1: As temperaturas são mostradas em graus Fahrenheit. (b) As temperaturas são mostradas em graus Celsius e as figuras formadas pela curva não mudam

7.2 EIXOS NÃO LINEARES

Em alguns casos, no entanto, o uso de eixos não lineares pode ser mais apropriado, especialmente quando se deseja representar dados que variam exponencialmente ou em grandes intervalos. A escala logarítmica, por exemplo, é uma das escalas não lineares mais utilizadas e é particularmente útil para dados que cobrem uma ampla gama de valores. Nesse contexto, a transformação por logaritmo pode simplificar a visualização, reduzindo a carga cognitiva do leitor ao interpretar as diferenças relativas entre os dados.

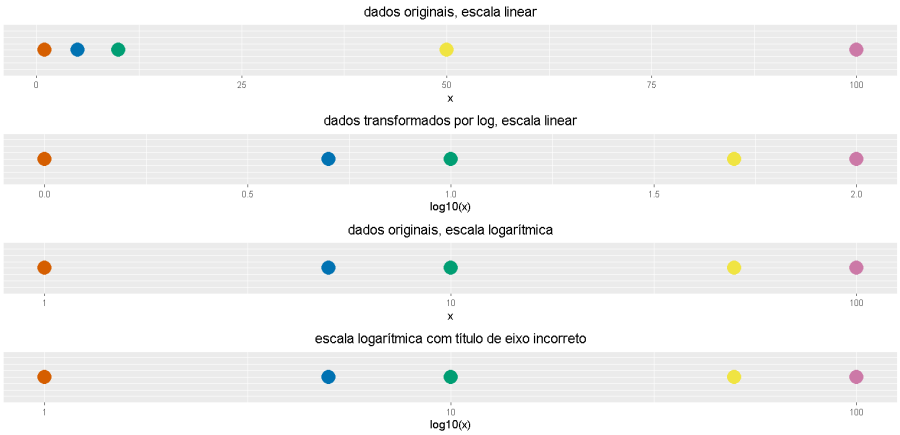


Figura 7.2.1: Comparação entre diferentes formas de representar dados e seus impactos na interpretação: dados originais em escala linear; dados transformados por logaritmo, mas representados em escala linear; dados originais em escala logarítmica; escala logarítmica com título rotulado como incorreto, $(log_{10}(x))$, que pode induzir erro de interpretação.

Apesar das vantagens, é crucial estar atento às possíveis confusões que podem surgir com o uso de escalas logarítmicas. A transformação logarítmica deve ser claramente indicada nos gráficos, especificando sempre a base utilizada (como logaritmo natural ou base 10), para evitar ambiguidade. A rotulagem inadequada ou ambígua pode levar a interpretações errôneas dos dados, comprometendo a análise. Portanto, ao optar por uma escala não linear, deve-se sempre assegurar que a escolha seja justificada pelo tipo de dados.

A utilização de eixos não lineares pode ser extremamente útil ao lidar com dados que apresentam grandes variações de escala. Um exemplo disso é a comparação da população de condados em relação ao valor mediano da população no estado do Texas. Quando representamos o número de habitantes de cada condado dividido pela mediana dos habitantes em todo o estado, obtemos uma razão que pode ser maior ou menor que 1. Essa razão nos permite avaliar a distribuição populacional de forma mais simétrica ao redor da mediana, especialmente quando utilizamos uma escala logarítmica.

7.2.1 Escala Logaritmica

Ao usar uma escala logarítmica, a visualização das razões mostra que a população dos condados está distribuída de forma aproximadamente simétrica em torno da mediana. Condados muito populosos terão uma razão significativamente maior que 1, enquanto condados menos populosos terão uma razão menor que 1, mas ainda compreensível dentro do contexto da mediana. Esse tipo de representação facilita a comparação entre condados de diferentes tamanhos populacionais sem distorcer as diferenças.

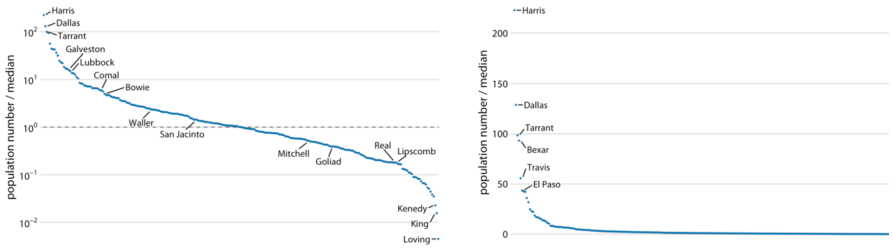


Figura 7.2.2: a escolha da escala adequada, como a escala logarítmica nesse caso, é fundamental para uma representação justa e clara dos dados.

Por outro lado, se esses mesmos dados fossem plotados em uma escala linear, as diferenças entre os condados com números de população próximos à mediana e aqueles com populações muito menores seriam drasticamente ampliadas ou distorcidas. Isso pode levar a interpretações errôneas ou a uma subestimação da importância relativa dos condados com menores populações. Portanto, a escolha da escala adequada, como a escala logarítmica nesse caso, é fundamental para uma representação justa e clara dos dados. A Figura 7.2.2 ilustra essa comparação, destacando a linha de razão igual a 1, que representa

o número populacional mediano dos condados, conforme o Censo Decenal dos EUA de 2010.

7.2.2 Escala Raiz Quadrada

Ao representar dados que apresentam grandes variações de magnitude, o uso de escalas não lineares, como a logarítmica, pode ser fundamental para uma visualização mais equilibrada e informativa. A escala logarítmica é particularmente útil quando os dados cobrem um amplo intervalo de valores, permitindo que multiplicações e divisões sejam representadas de maneira proporcional, com o valor 1 sendo o ponto médio natural, equivalente ao 0 em uma escala linear. No entanto, essa escala apresenta desafios, especialmente quando o conjunto de dados inclui o valor 0, que não pode ser representado em uma escala logarítmica. Nesse caso, recomenda-se o uso de transformações alternativas, como a escala de raiz quadrada, que permite a inclusão de 0 e comprime os números maiores em um intervalo menor.

A aplicação da escala de raiz quadrada oferece uma solução eficaz para representar dados que incluem o valor 0, mantendo ao mesmo tempo a capacidade de visualizar variações significativas em grandes intervalos de magnitude. A transformação de raiz quadrada comprime números grandes, tornando-os mais comparáveis e evitando a distorção que pode ocorrer em escalas lineares ao lidar com valores extremos. Essa abordagem é especialmente útil quando se busca uma distribuição mais uniforme dos dados, facilitando a interpretação visual e a comparação entre diferentes intervalos.

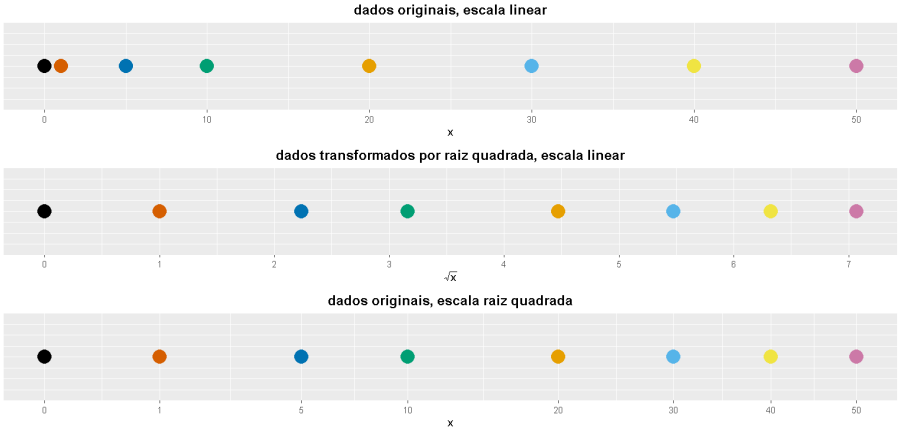


Figura 7.2.3: três formas de mostrar o mesmo conjunto de dados $x \in [0, 50]$: (1) valores originais em escala linear; (2) valores transformados por $\sqrt{x}$ em escala linear; (3) valores originais com o eixo x em *escala raiz* (mesmo posicionamento de (2), mas com rótulos em x). Intuição: a escala raiz "espalha" os valores pequenos e "aproxima" os grandes, de modo que passos iguais em x ocupam menos espaço conforme x aumenta.

Em resumo, a escolha entre escalas lineares, logarítmicas e de raiz quadrada deve ser orientada pela natureza dos dados e pelos objetivos da análise. Cada tipo de escala oferece vantagens e desafios específicos, e a seleção adequada pode melhorar significativamente a clareza e a precisão na comunicação dos resultados.

Embora a escala de raiz quadrada seja uma ferramenta valiosa para a compressão de dados em intervalos maiores, ela apresenta desafios específicos que devem ser cuidadosamente considerados ao aplicá-la em visualizações de dados. Um dos principais problemas é a falta de uma regra consistente para os passos unitários. Em uma escala linear, o incremento de um passo é constante, correspondendo à adição ou subtração de um valor fixo. Na escala logarítmica, cada passo representa uma multiplicação ou divisão por uma constante. No entanto, na escala de raiz quadrada, o significado de um passo unitário varia dependendo da posição no eixo, tornando-se uma operação não linear. Por exemplo, a diferença entre 1 e 4 na escala de raiz quadrada (um incremento de 1) é maior do que a diferença entre 9 e 16 (também um incremento de 1), o que não é intuitivo em comparação com as escalas lineares.

Além disso, a colocação de marcas de eixo igualmente espaçadas em uma escala de raiz quadrada requer uma posição baseada nas distâncias quadráticas, o que pode ser contra-intuitivo. Isso resulta em marcas de eixo em posições não uniformemente espaçadas, como 0, 4, 25 e 49, dificultando a interpretação visual (ver Figura 7.2.4). Outro desafio é a escolha de intervalos apropriados, que pode levar a um número insuficiente de marcas no início da escala ou a uma superlotação de marcas no final. Portanto, enquanto a escala de raiz quadrada pode ser útil para certos tipos de dados, sua aplicação exige um cuidado especial para garantir que a visualização seja precisa e intuitiva, evitando confusões na interpretação dos dados.

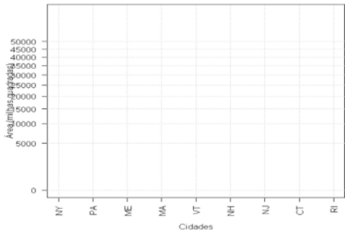
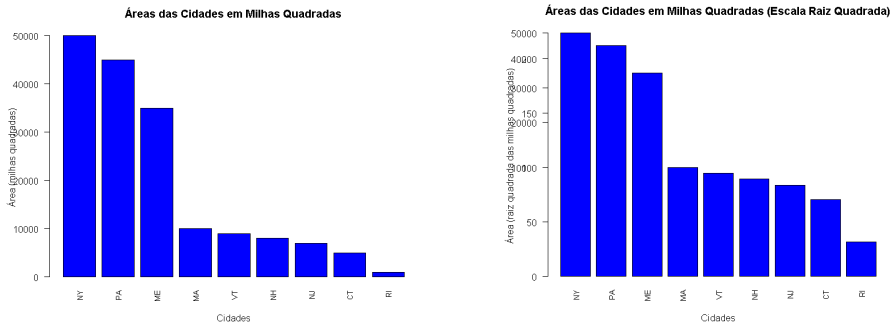


Figura 7.2.4: As marcas de eixo em posições não uniformemente espaçadas dificultam a interpretação visual.



(a) (b)

Figura 7.2.5: Exemplos de aplicações (a) sem e (b) com emprego da escala de raiz quadrada

A aplicação da escala de raiz quadrada em visualizações de dados pode ser particularmente útil para a análise de distribuições assimétricas ou para

destacar variações em intervalos de dados que, de outra forma, seriam difíceis de visualizar em uma escala linear. Por exemplo, ao comparar gráficos de áreas de cidades, a escala de raiz quadrada pode proporcionar uma percepção mais equilibrada das diferenças, especialmente quando se lida com valores que variam em ordens de magnitude. A compressão dos valores maiores em uma escala de raiz quadrada, como visto na Figura 7.2.6 permite que as diferenças entre áreas menores sejam mais visíveis, facilitando a interpretação.

Além disso, a linearização de funções de crescimento pode ser melhor representada utilizando a escala de raiz quadrada. Em uma função quadrática, como $y = x^2$, o uso da escala de $y = \sqrt{y}$ transforma o gráfico em uma relação linear, simplificando a visualização do crescimento e tornando a análise mais intuitiva. Esse tipo de transformação é particularmente útil em casos onde é necessário comparar taxas de crescimento ou mudanças em diferentes contextos.

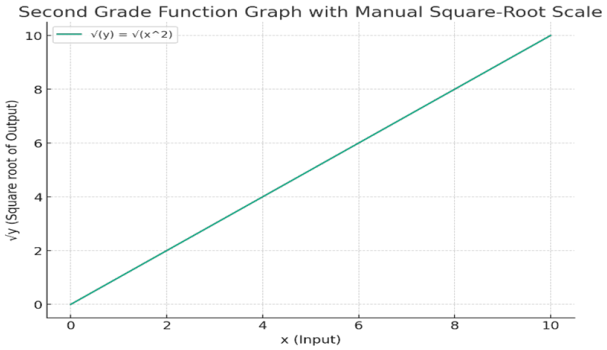


Figura 7.2.6: exemplo de linearização de uma parábola.

Outro uso importante da escala de raiz quadrada é na redução de assimetrias em distribuições de dados. Dados altamente assimétricos, com uma cauda longa à esquerda, podem ser transformados para reduzir essa assimetria, tornando padrões mais aparentes e facilitando a interpretação das estatísticas, como a média e a variância. Essa técnica pode ser essencial para melhorar a clareza visual e a interpretação dos dados, garantindo que as informações sejam comunicadas de forma mais clara para o leitor[22].

Em resumo, o uso de escalas não lineares, como a escala de raiz quadrada, deve ser considerado com cautela. Embora ofereçam soluções para a visualização de dados com grandes variações de magnitude, esses métodos demandam uma compreensão detalhada dos efeitos que produzem na interpretação visual. A aplicação correta dessas escalas pode melhorar a comunicação dos resultados, mas requer que o pesquisador esteja ciente das limitações e desafios inerentes a essas transformações.

[22] Nosso cérebro é preguiçoso, compara melhor diferenças relativas do que absolutas, então o mapeamento côncavo (uma transformação matemática onde o crescimento da saída diminui à medida que a entrada aumenta) $p(x) \propto \sqrt{x}$ expande o pequeno (onde tudo estava comprimido perto do 0) e comprime a cauda. Resultado: a primeira análise do leitor melhora sem trapacear nos números. É a lente correta para o córtex do leitor enxergar o que você quer. 👓 → 🧠

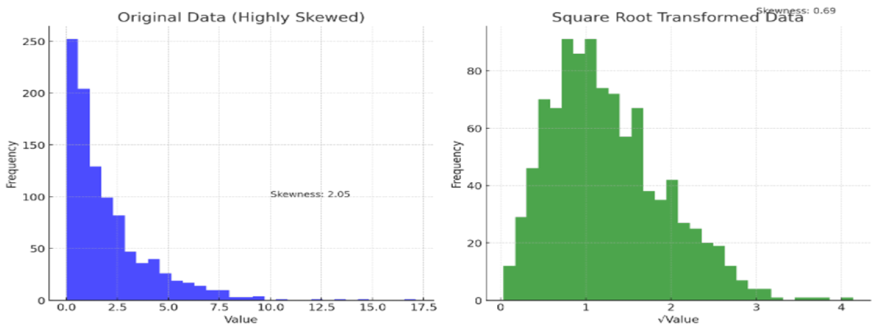


Figura 7.2.7: redução de assimetrias em distribuições de dados com a aplicação de transformações com escala de raiz quadrada.

7.3 TRANSFORMAÇÕES COMUNS

Transformações matemáticas ajudam a "equilibrar" a visualização. Em visualização de dados, representações lineares nem sempre são suficientes para revelar padrões relevantes, especialmente quando os dados apresentam alta variabilidade, assimetria ou crescimento acelerado. Nesses casos, transformações matemáticas nos eixos são estratégias fundamentais para aprimorar a legibilidade, estabilizar a variância e evidenciar relações ocultas.

Transformações bem escolhidas ajudam a adaptar a escala da visualização à estrutura estatística dos dados. Diferentemente de manipulações visuais arbitrárias, como truncar eixos ou alterar proporções gráficas sem justificativa, as transformações matemáticas operam sobre os próprios dados — e, quando corretamente aplicadas, preservam relações fundamentais como ordem, monotonicidade e proporção relativa.

- **Logaritmo:** Útil quando a variabilidade cresce proporcionalmente à magnitude dos dados (e.g., tráfego agregado em Gbps).
- **Raiz quadrada:** Reduz a amplitude de valores extremos mantendo a ordem dos dados.
- **Box–Cox:** Ajusta o grau de transformação de forma contínua, buscando simetrizar a distribuição.

As Figuras 7.3.1 e 7.3.2 ilustram o efeito de algumas transformações sobre a forma visual dos dados. Vale destacar que a escolha da transformação deve estar fundamentada em propriedades estatísticas do conjunto analisado e nos objetivos da análise — e não apenas em critérios estéticos ou de legibilidade gráfica.

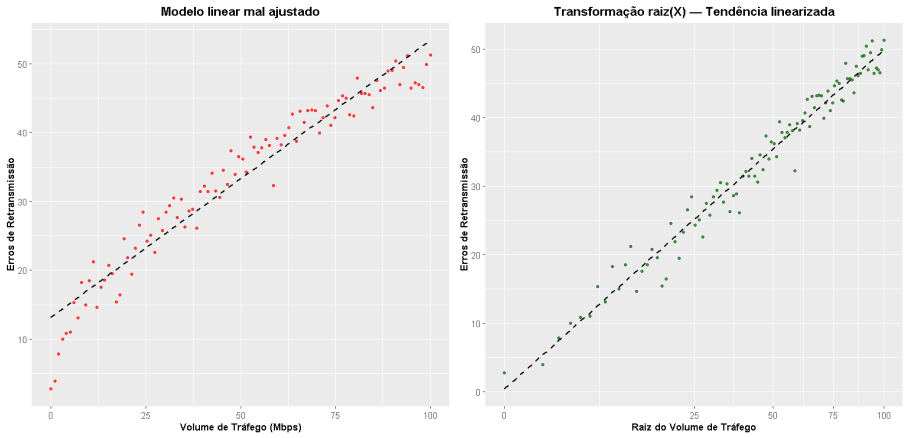


Figura 7.3.1: à esquerda, o ajuste de um modelo linear simples sobre o volume de tráfego mostra padrão sistemático, indicando inadequação do modelo. À direita, a aplicação de uma transformação na variável explicativa ($\sqrt{x}$) revela uma relação aproximadamente linear, evidenciando a importância de transformações apropriadas para linearizar relações não-lineares.

Armadilha de Escala

Armadilha comum: Manter escala linear em distribuições fortemente assimétricas. **Consequência:** Perda de detalhes nas regiões mais densas e distorção na percepção da magnitude. **Exemplo:** Ao plotar latências de pacotes onde 90% das medições estão abaixo de 20 ms, mas 10% chegam a 2000 ms, a escala linear torna os 90% quase indistinguíveis.

Boa Prática

Sempre testar múltiplas escalas antes de dar a figura como concluída. Escolher a transformação que melhor evidencia o fenômeno, mas informar no título ou legenda a transformação usada. Exemplo: "Latência (escala $\log_{10}$ em ms)".

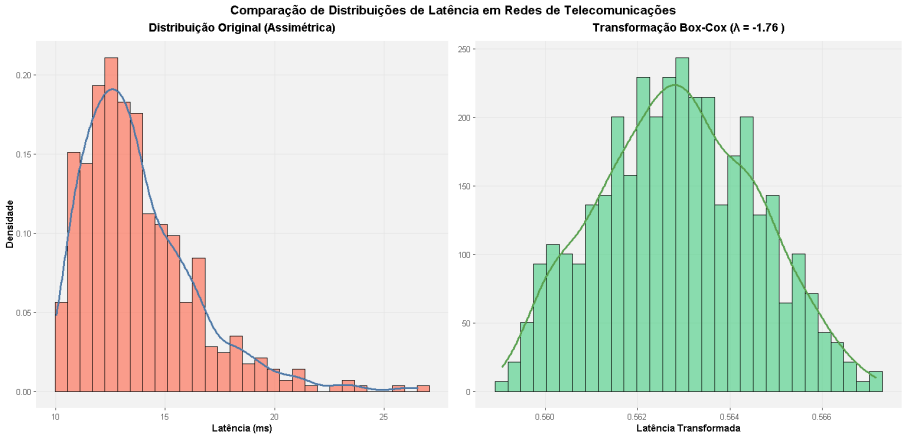


Figura 7.3.2: Distribuição da latência em redes de telecomunicações antes (à esquerda) e depois (à direita) da aplicação da transformação Box–Cox com $\lambda = 1.76$. A distribuição original apresenta forte assimetria à direita, dificultando análises baseadas em pressupostos de normalidade. A transformação suaviza a assimetria e aproxima a forma de uma distribuição simétrica, facilitando o uso de técnicas estatísticas paramétricas.

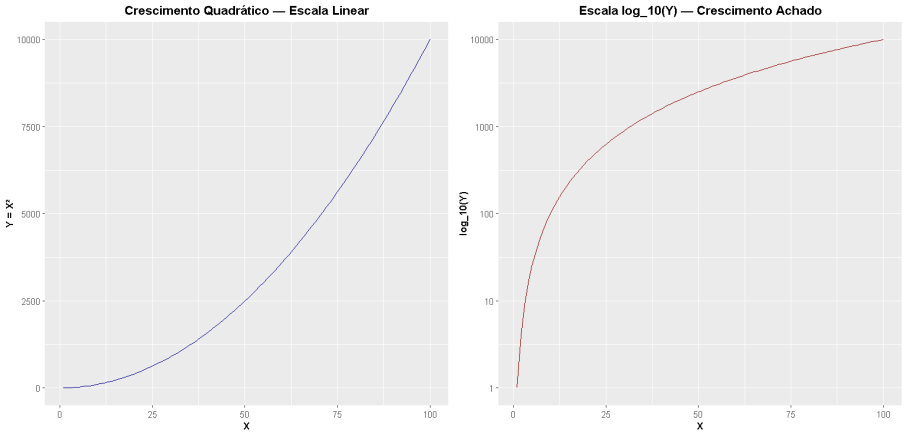


Figura 7.3.3: O gráfico à esquerda mostra uma função quadrática $y = x^2$ representada em escala linear, evidenciando seu crescimento acelerado. Já o gráfico à direita aplica uma transformação logarítmica ao eixo y, achatando visualmente o crescimento e dificultando a percepção da taxa de variação. O uso de escalas logarítmicas deve ser justificado, pois pode mascarar comportamentos importantes dos dados originais.

7.4 NORMALIZAÇÃO E PADRONIZAÇÃO

Nesta seção vamos diferenciar dois ajustes comuns de escala, Normalização e Padronização, ambos pertencentes à categoria de reescalonamento de variáveis (feature scaling), etapa fundamental no pré-processamento de dados para evitar que variáveis com amplitudes muito diferentes dominem as comparações.

A normalização (min–max) reescala os valores para um intervalo fixo, como $[0, 1]$. É útil quando precisamos que todas as variáveis fiquem na mesma faixa numérica, preservando a forma da distribuição original. É comum em visualizações e em algoritmos sensíveis à magnitude absoluta dos valores, como redes neurais ou k-vizinhos mais próximos.

A padronização (z-score) centraliza os dados em torno de 0 (subtraindo a média) e ajusta a dispersão para que o desvio padrão seja 1. Esse processo facilita comparações e é especialmente vantajoso para métodos que dependem de distâncias ou correlações.

7.4.1 Comparação Entre Grupos

A comparação entre grupos com tamanhos muito diferentes exige normalização ou padronização. Apresentar apenas valores absolutos favorece grupos maiores e pode ocultar padrões relevantes.

Armadilha: Comparar contagens brutas sem considerar proporções ou taxas.

Boa prática: Utilizar métricas relativas (por exemplo, taxas ou proporções) e escalas consistentes para comparação.

A etapa de escalonamento de características (*feature scaling*) é uma das mais importantes no pré-processamento de dados em *machine learning*. Algoritmos que calculam a distância entre as características tendem a ser enviesados em favor de valores numericamente maiores, caso os dados não sejam escalonados.

Na prática, frequentemente encontramos diferentes tipos de variáveis no mesmo conjunto de dados. Uma questão significativa é que o intervalo das variáveis pode diferir muito[23].

Vanos usar como exemplo um conjunto de dados (Tabela 2) que contém uma variável independente (Compra aprovada) e 3 variáveis dependentes (País, Idade e Salário). Podemos facilmente notar que as variáveis não estão na mesma escala porque a faixa de Idade vai de 27 a 50, enquanto a faixa de Salário vai de 48k a 83k. A faixa de Salário é muito maior que a faixa de Idade[24].

Ao calcular a distância euclidiana o resultado para o salário será muito maior que o da idade, o que significa que a distância euclidiana será dominada pelo salário se não aplicarmos algum tipo de transformação. Portanto, devemos usar o *Feature Scaling*. Para fazer isso, existem basicamente dois métodos chamados Padronização e Normalização.

Estas técnicas oferecem diversas vantagens que contribuem para o campo da análise de dados. Em primeiro lugar, permite a comparabilidade dos dados. Ao trabalhar com conjuntos de dados que possuem unidades ou escalas dife-

[23] Por exemplo ao fazer uma correlação entre idade e salário, . Usar a escala original pode colocar mais pesos nas variáveis com grande amplitude (enviesar a interpretação). Para lidar com este problema, precisamos aplicar alguma técnica de reescalonamento de dados na etapa de pré-processamento dos dados. Os termos normalização e padronização às vezes são usados de forma intercambiável, mas se referem a coisas diferentes.

[24] Note que deixamos propositalmente um valor em branco... Isso é muito comum e muitas vezes difícil de detectar, principalmente em dados massivos. O R naturalmente tem uma função para lidar com isso o `na.omit()`.

País	**Idade**	**Salario**	**Compra aprovada**
França	44	72000	Não
Espanha	27	48000	Sim
Alemanha	30	54000	Não
Esapanha	38	61000	Não
Alemanha	40	63730	
França	35	58000	Sim
Esapanha	30	52000	Não
França	48	79000	Sim
Alemanha	50	83000	Não
França	37	67000	Sim

Tabela 2: conjunto de dados para teste de distância entre os pontos, normalização e padronização.

rentes, a padronização fornece uma base comum para comparações significativas. Em segundo lugar, a padronização ajuda a identificar valores discrepantes (*outliers*). Ao padronizar os dados, os valores discrepantes tornam-se mais aparentes, auxiliando na investigação de observações incomuns. Por último, a padronização permite a normalização dos dados, garantindo que variáveis com diferentes intervalos ou distribuições possam ser analisadas e comparadas com maior precisão [7].

O resultado da padronização (ou padronização por *Z-score*) é que os dados serão redimensionados para garantir que a média e o desvio padrão sejam 0 e 1, respectivamente. E a fórmula para obtenção do *Z-score*[25], cuja fórmula é mostrada na Equação 1 :

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

[25] **O *Z-score***, é uma medida estatística que mostra a quantos desvios padrão um ponto de dados está em relação à média. É calculado subtraindo a média do valor do ponto em questão e dividindo o resultado pelo desvio padrão. O *Z-score* fornece informações valiosas sobre a posição relativa de um ponto de dados dentro de uma distribuiçãom

	Pais	Idade	Salario	Aprovação	Idade padr.	Sal. padr.
1	França	44	72000.00	Não	0.79	0.71
2	Espanha	27	48000.00	Sim	-1.40	-1.36
3	Alemanha	30	54000.00	Não	-1.02	-0.85
4	Espanha	38	61000.00	Não	0.01	-0.24
5	Alemanha	40	63730.00		0.27	-0.00
6	França	35	58000.00	Sim	-0.37	-0.50
7	Espanha	30	52000.00	Não	-1.02	-1.02
8	França	48	79000.00	Sim	1.30	1.32
9	Alemanha	50	83000.00	Não	1.56	1.66
10	França	37	67000.00	Sim	-0.12	0.28

Tabela 3: valores da Tabela 2 padronizados.

As Tabelas 3 e, 4 que apresentam respectivamente os dados padronizados e normalizados. Note que uma tabela tem valores negativos enquanto a outra

não, possui apenas valores positivos.

	Pais	Idade	Salario	Aprovação	Idade normaliz	Sal. normaliz.
. 1	França	44	72000.00	Não	0.74	0.69
2	Espanha	27	48000.00	Sim	0.00	0.00
3	Alemanha	30	54000.00	Não	0.13	0.17
4	Espanha	38	61000.00	Não	0.48	0.37
5	Alemanha	40	63730.00		0.57	0.45
6	França	35	58000.00	Sim	0.35	0.29
7	Espanha	30	52000.00	Não	0.13	0.11
8	França	48	79000.00	Sim	0.91	0.89
9	Alemanha	50	83000.00	Não	1.00	1.00
10	França	37	67000.00	Sim	0.43	0.54

Tabela 4: valores da Tabela 2 normalizados.

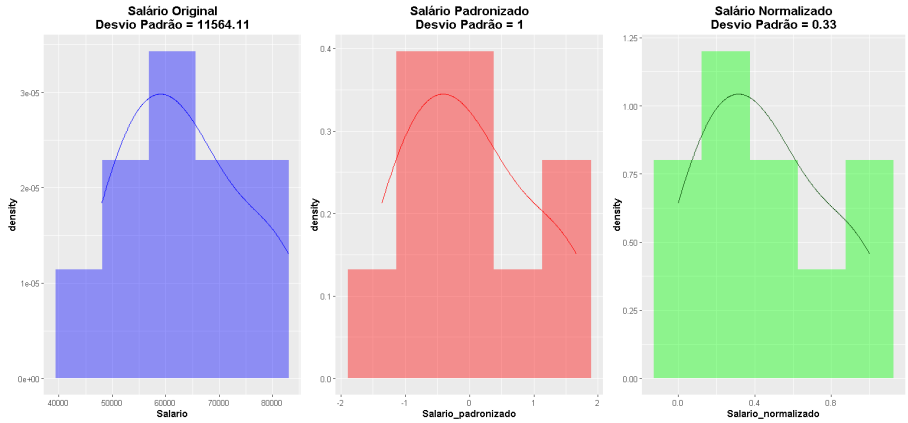


Figura 7.4.1: histograma das idades e salários com suas respectivas curvas de densidade sobrepostas.

A partir dos gráficos da Figura 7.4.1, observa-se que a Normalização Min–Max resultou em um desvio padrão menor do que o obtido pela Padronização. Isso significa que, após a normalização, os valores ficam mais concentrados em torno da média, enquanto na padronização há maior dispersão relativa.

Alguns modelos de aprendizado de máquina, como K-Vizinhos Mais Próximos, SVM e Redes Neurais, dependem fortemente do cálculo de distâncias. Nesses casos, o escalonamento das variáveis é fundamental, pois diferenças grandes de intervalo entre elas fazem com que as de maior amplitude tenham peso desproporcional no cálculo.

A Normalização Min–Max ajusta variáveis com escalas distintas para uma faixa comum, evitando que qualquer dimensão domine as estatísticas, e sem exigir suposições fortes sobre a distribuição dos dados — o que é conveniente para algoritmos como KNN e redes neurais. Entretanto, é muito sensível a

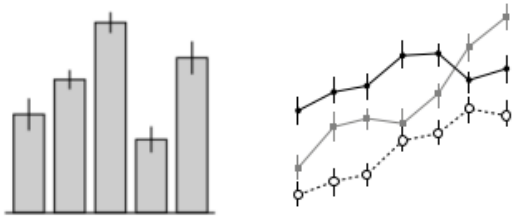
outliers, já que valores extremos determinam os limites de escala. Já a Padronização (z-score) lida melhor com *outliers* e pode favorecer a convergência de algoritmos baseados em otimização, como o gradiente descendente. Por isso, em muitos cenários práticos, a padronização tende a ser preferida à Normalização Min–Max.

8 Variação e Incertezas

É importante lembrar que dados experimentais tipicamente apresentam alguma incerteza e a visualização e representação destes dados deve "caracterizar a magnitude dessa incerteza em relação aos dados reais [8].

Alguns autores, infelizmente, apresentam resultados cujas figuras publicadas não atendem a esse padrão, especialmente quando a dimensionalidade dos dados aumenta [2]. Quando o objetivo de uma visualização é comparar uma quantidade medida ou derivada entre categorias ou condições, deve-se incluir dois elementos: (*i*) um elemento gráfico retratando a quantidade (correspondente às *geoms* do `ggplot`); e (*ii*) um elemento representando a incerteza associada a essa quantidade. Sem a representação da incerteza, uma comparação visual precisa não é possível, e os leitores podem chegar a conclusões incorretas ou mal informadas.

A variação e a incerteza podem ser retratadas com uma variedade de *geoms*, mas são mais comumente exibidas com barras de erro. Infelizmente, não há um padrão único para o que a barra de erro deve representar já que há uma pluralidade esmagadora de significados possíveis, como desvio padrão (DP) da amostra, erro padrão da média (EP ou SE) ou simplesmente erro padrão, Intervalo de Confiança (IC) paramétrico de $100(1-\alpha)\%$, intervalo de probabilidade bayesiano, um intervalo de previsão, etc. Cada quantidade tem sua própria interpretação estatística



(a) Gráficos de barra com barra de erros.

(b) Gráficos de linhas com barra de erros.

Figura 8.0.1: barras normalmente para comparar categorias e as linhas para comparar grupos e tendências [2].

Portanto, ao usar barras de erro, certifique-se de que (1) a quantidade codificada pela barra é consistente com o objetivo da visualização e (2) a quantidade é definida de forma inequívoca. Em relação ao primeiro ponto, oferecemos as seguintes diretrizes ao usar barras de erro para retratar a variação de uma estimativa de parâmetro ou a variação dos dados.

8.1 TIPOS DE ERRO: AMOSTRAL, RESIDUAL E INFERENCIAL

Além da variabilidade inerente aos dados, é fundamental compreender os diferentes tipos de erro envolvidos na análise estatística. Ignorar essas distinções pode levar a interpretações equivocadas, inferências frágeis ou visualizações enganosas. Cada tipo de erro está associado a uma etapa diferente da análise e serve a propósitos distintos:

Muitos desses conceitos são confundidos até por profissionais experientes. Por exemplo, é comum interpretar equivocadamente barras de erro como indicativas de variabilidade quando, na verdade, representam incerteza na estimativa. Ou então utilizar testes de hipótese sem considerar o risco de erro tipo II, produzindo estudos estatisticamente insignificantes, mas com potencia analítica insuficiente.

Erro Amostral:

O **erro amostral** é a diferença entre uma estatística observada na amostra,como a média, e o valor real (desconhecido) do parâmetro populacional, como . Trata-se de uma quantidade concreta, porém geralmente inacessível, já que o valor populacional raramente é conhecido.

$$\bar{x} - \mu$$

Esse erro decorre unicamente do fato de termos utilizado uma amostra finita, e não toda a população. Ele não é eliminável, mas tende a diminuir com o aumento do tamanho amostral[26].

Erro Residual:

Já o **erro residual** é definido, ponto a ponto, como a diferença entre o valor observado e o valor ajustado pelo modelo:

$$\text{resíduo}_i = y_i - \hat{y}_i$$

Ele expressa o desvio individual de cada observação em relação à curva, reta ou modelo ajustado. Portanto, enquanto o erro amostral diz respeito à estimativa de um parâmetro populacional a partir de uma amostra, o erro residual avalia o ajuste do modelo aos dados concretos.

Resíduos são insumos críticos em diagnósticos de regressão: padrões sistemáticos nos resíduos indicam problemas como não linearidade, heterocedasticidade ou dependência serial.

Erros Inferenciais do Tipo I e Tipo II:

[26] O Erro padrão é a medida de variabilidade esperada do erro amostral ao longo de repetições amostrais, dado por EP $= \frac{\sigma}{\sqrt{n}}$, onde σ é o desvio padrão populacional e n o tamanho da amostra. O erro padrão não é um erro em si, mas sim o desvio padrão da estatística $\bar{x}$. Ele serve como base para a construção de intervalos de confiança e testes de hipótese. O termo, infelizmente mal nomeado, foi popularizado por Ronald Fisher, que apesar de genial na formalização da estatística inferencial, não demonstrava o mesmo talento ao nomear conceitos. Sir Fisher também é responsável por outros termos de difícil apreensão como $p - value$ e hipótese nula.

Quando tomamos decisões inferenciais com base em testes de hipótese, estamos sujeitos a dois tipos clássicos de erro:

- **Erro Tipo I (α)**: rejeitar uma hipótese nula verdadeira. É o chamado "falso positivo".
- **Erro Tipo II (β)**: não rejeitar uma hipótese nula falsa. É o "falso negativo".

Esses erros não decorrem de falhas nos dados, mas sim da natureza probabilística da inferência estatística.

Tipo de Erro	Descrição
Erro Amostral	Diferença entre a estatística calculada na amostra (como $\bar{x}$) e o parâmetro populacional verdadeiro (como μ). Depende do tamanho da amostra e é conceitualmente distinto do erro padrão.
Erro Residual	Diferença entre o valor observado y_i e o valor estimado $\hat{y}_i$ por um modelo. Mede o desvio individual e local de cada ponto em relação à tendência ajustada.
Erro Tipo I (α)	Rejeição incorreta da hipótese nula verdadeira. Equivale a detectar um efeito que não existe (falso positivo).
Erro Tipo II (β)	Falha em rejeitar a hipótese nula quando ela é falsa. Representa a perda de um efeito real (falso negativo).

Tabela 5: resumo comparativo dos principais tipos de erro em análise estatística.

A Tabela 5 resume os principais tipos de erro envolvidos na análise estatística e na comunicação de resultados. Compreender essas distinções é essencial não apenas para interpretar corretamente os testes e intervalos construídos, mas também para elaborar visualizações que expressem, de forma honesta, os limites da evidência que os dados permitem sustentar.

8.2 INTERVALO DE CONFIANÇA E NÍVEL DE SIGNIFICÂNCIA ESTATÍSTICA

O nível de significância α é o limite máximo que aceitamos para a probabilidade de cometer um erro do Tipo I — isto é, rejeitar a hipótese nula H_0 quando ela é verdadeira. Por convenção, escolhemos valores pequenos (como $\alpha = 0.05$ ou 0.01) para tornar a rejeição de H_0 mais rigorosa. Note que α não mede a probabilidade de H_0 ser verdadeira, define a "tolerância ao risco" do teste.

Historicamente, Sir Ronald A. Fisher batizou esse limiar como "nível de significância" porque, para ele, um resultado que cruzasse α merecia atenção especial: era **estatisticamente suspeito**, no sentido de improvável sob H_0. Não significava "importante" no uso cotidiano, mas sim "digno de nota" para análise mais aprofundada. Esse batismo linguístico sobreviveu por tradição, mesmo sendo contra-intuitivo para iniciantes, pois a palavra "significância" no senso comum sugere importância prática, e não apenas raridade estatística[27].

Quando o objetivo é estimar parâmetros populacionais, como a média ou a variância, usamos o intervalo de confiança (IC) para expressar a incerteza dessa estimativa. O IC está diretamente ligado a α: um IC de 95% é construído de modo que, em média, 95% dos intervalos obtidos de amostras aleatórias contenham o valor verdadeiro do parâmetro.

IC, outro conceito muitas vezes mal entendido [10] e [11]
Um intervalo de confiança de $100(1-\alpha)\%$ para um parâmetro populacional é um intervalo calculado a partir dos dados amostrais que, em $100(1-\alpha)\%$ das amostras possíveis, conteria o verdadeiro valor do parâmetro.
Isto é: Suponha que você calcule a média de alturas de uma amostra de 100 pessoas e obtenha uma média de 170 cm com um desvio padrão de 10 cm. Um intervalo de confiança de 95% pode ser calculado, e você pode obter algo como (168 cm, 172 cm). Isso significa que, se você repetisse esse experimento várias vezes, 95% dos intervalos de confiança calculados conteriam a verdadeira média populacional (nesse caso 170 cm).

Ao compreender o significado de um intervalo de confiança e seu nível de significância α, torna-se natural relacionar esses conceitos à lógica da tomada de decisão em testes de hipótese. O desfecho de um teste não depende apenas dos dados observados, mas também da realidade subjacente se a hipótese nula H_0 é verdadeira ou falsa, e da decisão tomada pelo pesquisador. A Tabela 6 resume de forma sistemática as quatro combinações possíveis entre a situação real e a decisão adotada, destacando os conceitos de erro tipo I, erro tipo II, potência estatística e decisões corretas.

Figura 8.2.1: Sir Ronald Fisher (1925) criou o $p-value$ no livro *Statistical Methods for Research Workers* [9], Fisher apresentou o ***probability-value*** como uma medida de evidência contra H_0, interpretada como um grau de "surpresa estatística": quanto menor o $p-value$, mais improvável seria obter um resultado tão extremo se H_0 fosse verdadeira. O uso de valores fixos veio depois, com Neyman e Pearson, associando-o ao nível de significância α.

[27] Em resumo: Sir Fisher chamou de "significância" porque, para ele, cruzar o limiar α indicava que algo incomum estava ocorrendo sob o modelo nulo, merecendo investigação. (1) No inglês científico da época, *significant* queria dizer "com significado estatístico especial". (2) A expressão passou a ser repetida até se tornar padrão, mas gerou confusão e Deus sabe! Ainda gera, mas em estatística, "significativo estatisticamente" não implica importância estatística, mas um ponto de atenção, rejeitar ou não rejeitar H_0. ⚠

⚠ Armadilha — Mais confiança não é mais precisão

É um erro intuitivo frequente: achar que aumentar o nível de confiança deixa o resultado "mais preciso". Na verdade, ocorre o oposto — quanto maior a confiança, mais largo será o intervalo de confiança (e a barra de erro). Isso acontece porque, para garantir que a estimativa capture o valor verdadeiro em uma proporção maior de amostras, precisamos ampliar a margem de erro. Assim, um IC de 99% é "mais cauteloso" que um IC de 95%, mas também é menos informativo no sentido de precisão: suas barras de erro crescem, não encolhem. Confiança e precisão não andam sempre de mãos dadas — aumentar uma pode sacrificar a outra.

Tabela 6: Relação entre Situação Real, Decisão, Erro e a probabilidade p

Situação Real	**Decisão**	**Result. Estatístico**	**Nome Técnico**	p
H_0 é verdadeira	Rejeita H_0	Erro Tipo I	Falso Positivo	α
H_0 é verdadeira	Não rejeita H_0	Decisão correta	Verdadeiro Negativo	$1-\alpha$
H_1 é verdadeira	Rejeita H_0	Potência estatística	Verdadeiro Positivo	$1-\beta$
H_1 é verdadeira	Não rejeita H_0	Erro Tipo II	Falso Negativo	β

Um outro exemplo de interpretação geométrica dos intervalos de confiança são a sua sobreposição ou não entre duas amostras.

A sobreposição dos ICs [12] e [13]
Um intervalo de confiança de $100(1-\alpha)\%$ para um parâmetro populacional é um intervalo calculado a partir dos dados amostrais que, em $100(1-\alpha)\%$ das amostras possíveis, conteria o verdadeiro valor do parâmetro.
Isto é: Suponha que você calcule a média de alturas de uma amostra de 100 pessoas e obtenha uma média de 170 cm com um desvio padrão de 10 cm. Um intervalo de confiança de 95% pode ser calculado, e você pode obter algo como (168 cm, 172 cm). Isso significa que, se você repetisse esse experimento várias vezes, 95% dos intervalos de confiança calculados conteriam a verdadeira média populacional (nesse caso 170 cm).

⚠ Armadilha — interpretação visual de sobreposição de ICs

A sobreposição de intervalos de confiança pode sugerir que as estimativas de **diferentes grupos** não são estatisticamente diferentes, mas essa interpretação deve ser feita com cautela. Em geral, quando os ICs de 95% se sobrepõem, é comum assumir que a diferença entre os grupos não é significativa. Contudo, essa não é uma regra rígida, pois ainda pode haver diferença estatisticamente significativa, especialmente quando os ICs são estreitos ou o tamanho da amostra é grande.
De forma análoga, a não sobreposição visual entre ICs não garante significância: em alguns casos, a variabilidade pode ser alta e o teste formal ainda assim não rejeita H_0. A Figura 8.2.2 ilustra dois casos reais que evidenciam essas situações.
A conclusão correta sobre a diferença entre grupos não apenas em uma inspeção visual das barras individuais.

Além das tradicionais barras de erro vistas na Figura 8.2.2 , há outras formas eficazes de representar intervalos de confiança e variabilidade dos dados, como as bandas de confiança e os gráficos de violino e barras de erro com outras medidas de incerteza.

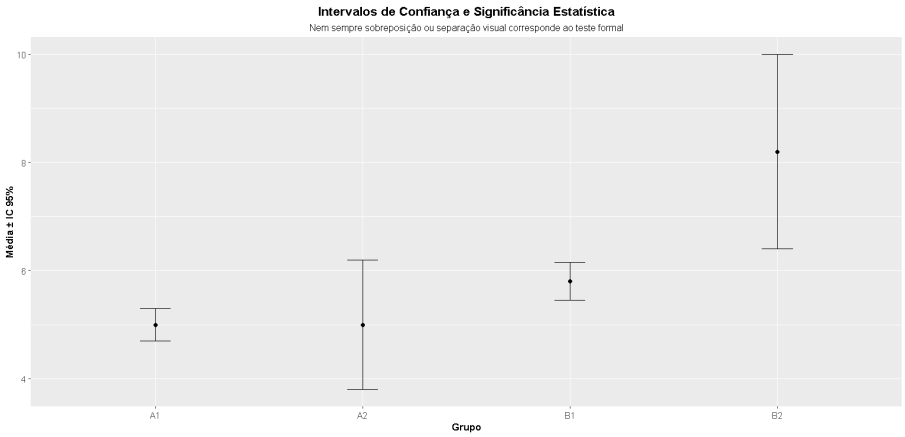


Figura 8.2.2: Todos os grupos (A1, A2, B1 e B2) foram plotados com o mesmo nível de confiança (95%), mas as larguras dos intervalos variam conforme o erro padrão da média ($EP = \sigma/\sqrt{n}$). A sobreposição visual entre os ICs não garante ausência de diferença estatística: no exemplo, A2 e B1 apresentam ICs que se sobrepõem parcialmente, mas um teste t para amostras independentes indica diferença estatisticamente significativa ($p \approx 0.0006$). Isso ocorre porque o teste considera simultaneamente as médias, a variabilidade e o tamanho amostral fatores que a simples inspeção visual das barras não captura plenamente. Portanto, a interpretação, correta exige teste formal, e não apenas observação do gráfico.

A Figura 8.2.2 apresenta três visualizações distintas que incorporam essa noção de incerteza. Já Figura 8.2.3 ilustra três formas distintas de representar incerteza estatística: à esquerda, uma dispersão com banda de confiança ajustada a um modelo; ao centro, barras de erro representando intervalos de confiança sobre médias de dois grupos; e à direita, pontos com barras de IC 95%, destacando a estimativa pontual e sua variação. Então toda atenção é pouca[28]

[28] À primeira vista, a linda banda azul do gráfico à esquerda da Figura 8.2.4 parece transmitir segurança estatística, precisão matemática e até bom gosto visual, mas cuidado: essa banda oscilante não é um intervalo de confiança — é apenas a variação empírica da simulação! Não se deixe seduzir por bandas coloridas, elas podem estar apenas "abraçando o ruído".

⚠ Armadilha — uso automático ou ornamental

O gráfico à esquerda na Figura 8.2.4, com dispersão e banda simulada de variabilidade construída via `geom_ribbon()` R, é útil quando se deseja representar a dispersão empírica dos dados em torno de uma tendência, especialmente para explorar aspectos como assimetrias, modas múltiplas ou caudas longas. No entanto, seu uso se torna irrelevante, ou mesmo enganoso, quando aplicado a conjuntos com poucos dados, distribuições triviais (como as simétricas e unimodais), ou quando a forma da densidade não é objeto de interesse analítico. Nesses casos, a banda atua apenas como um embelezamento herdado, muitas vezes ativado por padrão em bibliotecas de visualização ou plataformas automatizadas como *Large Language Models* (LLMs), sem aportar qualquer conteúdo estatístico relevante.

Já o gráfico à direita representa a banda de confiança da média estimada via `geom_smooth()` R, esta sim com interpretação estatística bem definida, desde que haja interesse inferencial na média ou tendência ajustada. Mesmo assim, tais bandas devem ser usadas com critério: em visualizações meramente descritivas, podem sugerir um nível de precisão indevido ou obscurecer a mensagem principal do gráfico.

Regra geral: toda camada visual deve responder a uma pergunta estatística concreta. Caso contrário, é ruído gráfico — ou pior, ornamento sem propósito.

As barras de erro são um recurso gráfico essencial para comunicar incerteza em resultados experimentais. Entretanto, a sua interpretação é ambígua quando não se declara explicitamente o que elas representam. Na literatura é possível encontrar três convenções distintas: o uso do desvio padrão (DP), do erro padrão da média (EP) ou do intervalo de confiança (IC).

Cada escolha transmite uma informação diferente. O **desvio padrão (DP)** descreve a dispersão natural dos dados em torno da média, enfatizando a variabilidade intrínseca da amostra. O DP é adequado quando o objetivo é mostrar quão heterogêneos são os valores individuais dentro de um grupo, como em medidas de desempenho de protocolos em múltiplas simulações independentes.

O **erro padrão da média (EP)** expressa a precisão da estimativa da média, sendo reduzido pelo aumento do número de observações. O EP é indicado quando se deseja destacar a confiabilidade da média como estimador, por

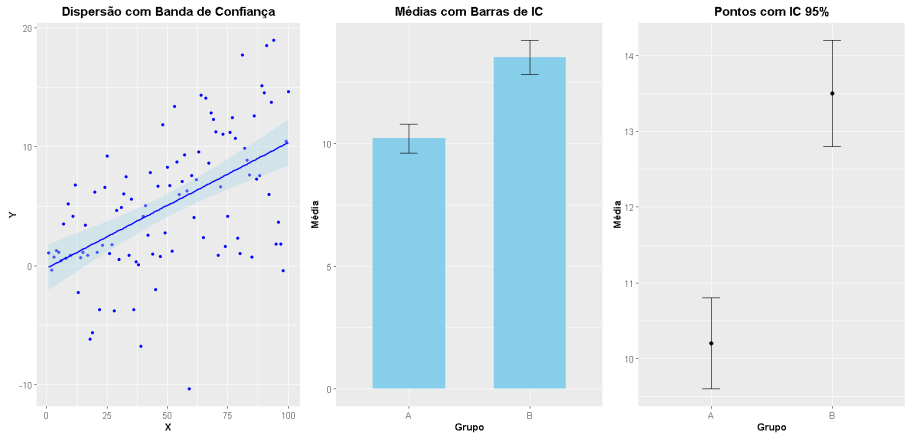


Figura 8.2.3: três representações visuais do intervalo de confiança (IC) em diferentes contextos. À esquerda, a banda de confiança expressa a incerteza na tendência estimada entre duas variáveis. No centro, barras com IC de 95% mostram a incerteza em torno das médias de dois grupos. À direita, pontos com barras de IC reforçam a comparação direta entre as estimativas, destacando a amplitude da incerteza entre grupos forma mais clara que os gráficos de barras.

exemplo ao comparar o tempo médio de resposta de dois algoritmos avaliados em várias execuções controladas.

Por fim, o **intervalo de confiança (IC)** combina o EP com um fator estatístico (como a distribuição t de Student), delimitando uma faixa onde se espera que a média populacional esteja contida com determinada confiança (geralmente 95%). O IC é a escolha apropriada quando se pretende extrapolar conclusões do experimento para além da amostra observada, como em estudos que buscam generalizar o desempenho de um sistema em condições operacionais típicas.

Assim, dois gráficos com as mesmas médias podem transmitir percepções muito distintas dependendo da métrica escolhida para as barras de erro. Por essa razão, recomenda-se fortemente que todo gráfico explicite de forma inequívoca se as barras representam DP, EP ou IC, evitando interpretações equivocadas pelo leitor. A Figura 8.2.5 ilustra esse contraste utilizando os três casos.

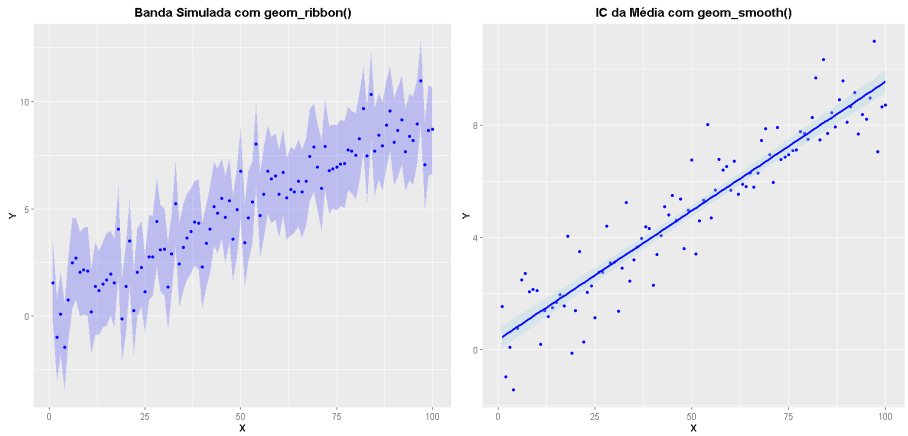


Figura 8.2.4: comparação entre banda empírica com `geom_ribbon()` (esquerda) e intervalo de confiança da média com `geom_smooth()` (direita), Ambas no R.

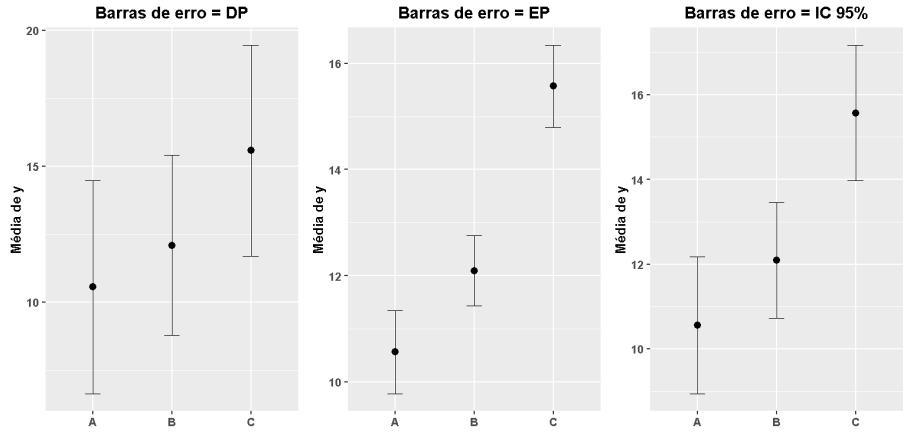


Figura 8.2.5: Exemplo de barras de erro aplicadas ao mesmo conjunto de dados. À esquerda, as barras representam o desvio padrão (DP), refletindo a dispersão dos valores individuais. No centro, utilizam-se o erro padrão da média (EP), mostrando a incerteza na estimativa da média. À direita, são exibidos intervalos de confiança de 95% (IC), que expressam a faixa de valores onde se espera encontrar a média populacional com esse nível de confiança. A comparação ressalta a importância de declarar explicitamente a métrica utilizada ao apresentar barras de erro.

8.2.1 Potência Estatística e o d de Cohen

A **potência estatística** de um teste, denotada por $1 - \beta$, representa a probabilidade de detectar um efeito real, isto é, de rejeitar corretamente a hipótese nula H_0 quando a hipótese alternativa H_1 é verdadeira. É o complemento do erro Tipo II (β), que ocorre quando falhamos em rejeitar H_0 apesar de existir um efeito real.

Interpretação Geométrica:

Antes de abordarmos a potência estatística em si, é importante compreender o cenário em que ela se manifesta: o teste de hipóteses. Ao confrontar uma hipótese nula (H_0) contra uma alternativa (H_1), estamos implicitamente lidando com duas distribuições de probabilidade distintas. A potência do teste está diretamente relacionada à capacidade de distinguir essas distribuições, ou seja, detectar um efeito real quando ele de fato existe.

A seguir, veremos como essa distinção se traduz graficamente na área sob a curva de H_1 que ultrapassa o limiar crítico definido por H_0. Essa área representa a probabilidade de rejeitarmos corretamente H_0, e é fortemente influenciada pela distância entre as duas curvas. Distância esta que pode ser quantificada pela métrica chamada d **de Cohen**.

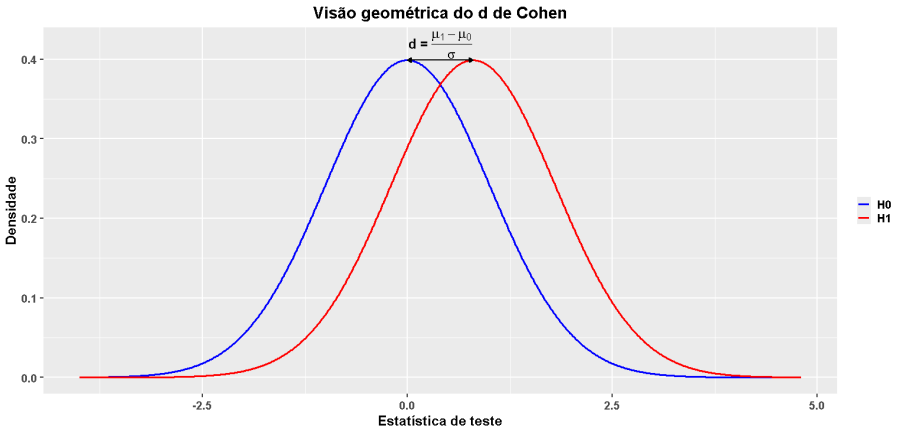


Figura 8.2.6: representação geométrica do d de Cohen como a distância padronizada entre as distribuições sob a hipótese nula (H_0) e a alternativa (H_1). Ambas as curvas são distribuições normais com o mesmo desvio padrão σ, mas com médias separadas por $d \cdot \sigma$. Essa distância quantifica o tamanho do efeito em unidades de desvio padrão, sendo visualmente interpretável como o afastamento entre os picos das curvas.

Na prática, a potência está ligada à capacidade de um experimento revelar evidência suficiente contra H_0, quando necessário. Ela depende de quatro fatores principais:

- o nível de significância α adotado,
- o tamanho da amostra n,
- a variabilidade dos dados (desvio padrão σ),
- e a *magnitude do efeito* a ser detectado.

Este último fator — a **magnitude do efeito**, é frequentemente negligenciado, mas possui grande importância prática. Mesmo efeitos estatisticamente significativos podem ser irrelevantes em termos reais. É nesse contexto que se introduz o d **de Cohen**, uma medida *padronizada* da diferença entre dois grupos, definida como:

$$d = \frac{E}{\sigma}$$

onde E representa a diferença observada (ou esperada) entre médias e σ é o desvio padrão da população (ou sua estimativa amostral). O valor de d quantifica essa diferença em unidades de desvio padrão, permitindo avaliar o tamanho do efeito independentemente da escala original da variável medida.

Por convenção em áreas como psicologia, medicina e ciências sociais, três faixas de interpretação foram propostas por Jacob Cohen:

$d \approx 0.2$ Pequeno efeito (diferença sutil)
$d \approx 0.5$ Efeito moderado (diferença perceptível)
$d \geq 0.8$ Grande efeito (diferença marcante)

Importante

Essas faixas são heurísticas, não limites rígidos. O significado prático de um d depende sempre do contexto.

Apesar da utilidade interpretativa, o d de Cohen é pouco utilizado em pesquisas de engenharia e computação. Isso se deve, em parte, ao foco dessas áreas em métricas absolutas (tempo, latência, throughput) e comparações percentuais, mais que em medidas padronizadas.

Contudo, há situações nas quais o uso do d pode ser esclarecedor, como por exemplo:

- Em estudos comparativos entre algoritmos, quando se deseja comunicar a magnitude da melhoria em relação à variabilidade observada;
- Em avaliações com participação humana, como experimentos de usabilidade ou análise de qualidade perceptual;

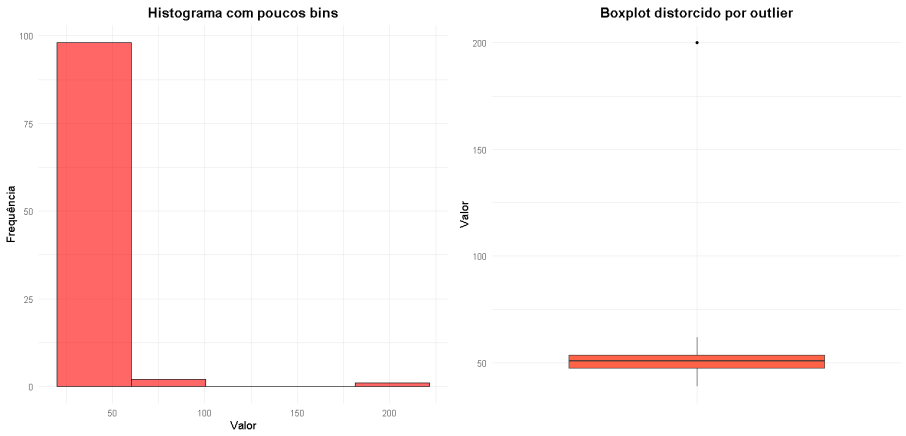


Figura 8.2.7: dois exemplos de visualizações comprometidas por *outliers*. À esquerda, um histograma com binagem inadequada cria artefatos visuais e oculta a presença do valor extremo. À direita, um boxplot que comprime todo o restante dos dados para acomodar o *outlier*, dificultando a leitura da distribuição principal. Deseja agora que eu monte a versão dos dois gráficos corrigidos, em azul, como segundo par?

- Em testes repetidos com amostras pequenas, quando se deseja justificar o tamanho de amostra com base em uma diferença mínima detectável relevante.

Assim, embora incomum em nosso domínio, o uso do d de Cohen pode enriquecer a argumentação estatística e ajudar a responder à pergunta essencial: *essa diferença, além de estatisticamente significativa, é grande o suficiente para importar?*

8.2.2 Detecção de Outliers e sua Representação

A identificação de valores atípicos (*outliers*) deve ocorrer logo nas primeiras etapas da visualização de dados. Outliers podem distorcer eixos, achatar distribuições, influenciar medidas de tendência central e, mais grave, afetar a percepção tanto do autor quanto do leitor. Uma armadilha comum está na construção de gráficos que, por omissão ou automatismo, ocultam ou exageram a influência desses pontos extremos.

Na Figura 8.2.7, temos dois exemplos de visualizações comprometidas. À esquerda, o histograma com poucos bins obscurece a presença do *outlier* e sugere uma distribuição artificialmente concentrada. À direita, o boxplot é totalmente comprimido pelo valor extremo, tornando ilegível a variação entre os dados principais.

Na Figura 8.2.8, são apresentadas versões corrigidas das mesmas visualizações. O histograma passa a utilizar mais bins e aplica um foco visual no

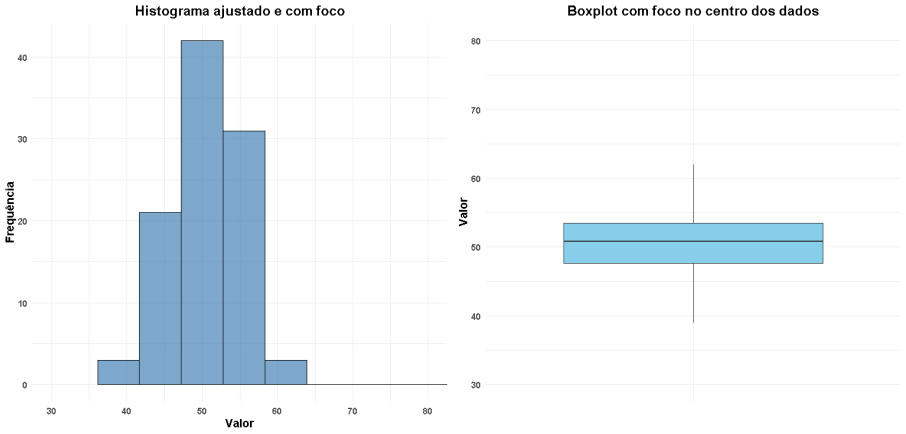


Figura 8.2.8: dois exemplos de visualizações corrigidas para lidar com outliers. À esquerda, o histograma com maior número de bins e foco no intervalo relevante evidencia melhor a distribuição central. À direita, o boxplot com escala ajustada permite observar a dispersão dos dados principais, sem eliminar o *outlier*, mas destacando-o de forma visualmente apropriada.

intervalo mais informativo, revelando a forma da distribuição real. O boxplot, por sua vez, utiliza uma escala truncada para priorizar a região densa dos dados, sem ocultar o outlier, que é destacado de forma clara e visualmente honesta. A boa visualização não deve esconder os *outliers*, mas também não deve deixá-los dominar indevidamente a narrativa visual.

⚠ Armadilha - bins mal definidos e outliers ignorados

Uma das armadilhas mais frequentes em visualizações é a combinação de **bins mal ajustados** e a **omissão silenciosa de valores atípicos**. A escolha inadequada de bins em histogramas pode ocultar padrões ou gerar falsas modas. Evite o número padrão automático de bins. Para estimar o número ideal, use:

- **Regra de Sturges:** $k = 1 + \log_2(n)$
- **Regra de Scott:** largura $= 3.5 \cdot \sigma / n^{1/3}$
- **Regra de Freedman–Diaconis:** largura $= 2 \cdot \text{IQR}/n^{1/3}$

Escolher o número de bins afeta a percepção de dispersão, simetria e existência de múltiplos modos.
Valores atípicos podem achatar visualizações e distorcer medidas como média e desvio padrão. a identificação visual inicial pode ser feita com:

- **Boxplot:** ponto fora de $[Q1 - 1.5 \cdot \text{IQR},\ Q3 + 1.5 \cdot \text{IQR}]$
- Teste**Z-score:** $|z| > 3$ pode indicar outlier sob normalidade

Para avaliação formal, use testes estatísticos como:

- **Teste de Grubbs (para 1 *outlier* extremo)**
- **Teste de Dixon (para pequenas amostras)**
- **Rosner (para múltiplos *outliers*)**, mais robusto

Nunca exclua outliers sem justificar. Se decidir ocultar graficamente (ex: no R, com `coord_cartesian()`), informe explicitamente que há pontos fora do intervalo visível. O silêncio visual pode induzir à falsa conclusão de ausência de anomalias.

8.2.3 Distribuições Assimétricas e Escalas Apropriadas

Distribuições assimétricas,também chamadas de enviesadas, ocorrem quando os dados não são distribuídos de forma simétrica em torno da média. Exemplos típicos incluem distribuições de renda, tempos de resposta e tráfegos de rede.

Impacto da Escala na Percepção

Quando apresentadas em escalas lineares, distribuições fortemente assimétricas podem achatar os valores mais concentrados ou comprimir excessivamente as caudas. Isso pode ocultar variações importantes, dificultando comparações entre categorias ou períodos.

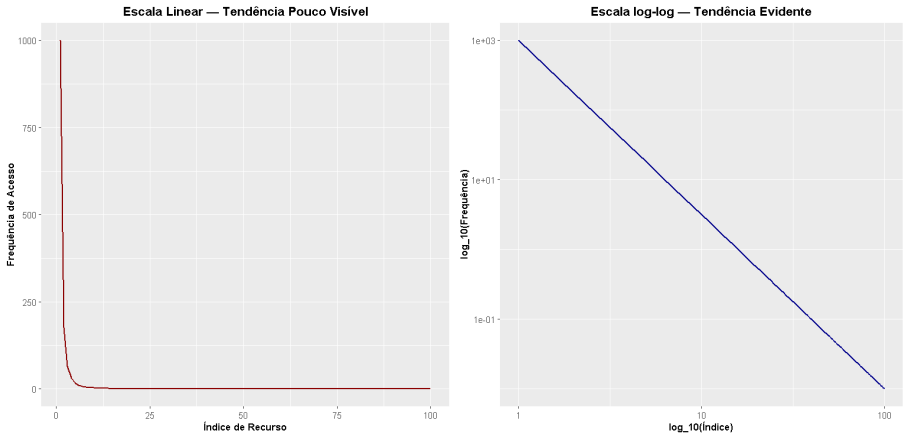


Figura 8.2.9: Comparação entre escalas linear e log–log na representação da frequência de acesso por índice de recurso. À esquerda, a escala linear comprime a maior parte dos dados, dificultando a identificação de padrões. À direita, a aplicação de escala logarítmica nos dois eixos revela uma tendência de decaimento aproximadamente linear em escala log–log.

9 Introdução a Análise de Dados

Experimentos na nossa área (Engenharia) tipicamente nos inundam com dados muitas vezes difíceis de entender e quiçá explicar. O objetivo aqui é escrutinar o método, a lógica, a arte da análise e representação de dados, empregando ferramentas essenciais para examinar dados. Concordamos com a filosofia do "aprender fazendo" para uma melhor compreensão dos dados [29].

[29] Esta Seção foi fortemente apoiada no trabalho publicado em [6].

9.1 CASOS REAIS

Ao perceber um aumento na quantidade de propagandas de fraldas, fórmulas infantis e macacões da Target, o destinatário fica se perguntando por que estão enviando tanta publicidade voltada para bebês. A Target explicou que os dados recentes de compras indicavam que havia uma mulher grávida na residência. Uma semana após ligar para a Target, ele descobre que sua filha está grávida.

Em um cenário como dos surtos de H1N1, Ebola, Zika ou COVID-19, o período comum de 2 semanas de análise é longo demais. Uma equipe do Google permitiu que a Internet descobrisse onde os surtos ocorriam. Para desenvolver seu modelo, eles rastrearam a disseminação do H1N1 e correlacionaram com termos de busca na Internet: febre alta, tosse e dores. Informando as autoridades quando e onde exatamente o novo surto de gripe estava ocorrendo.

Para ajudar a reduzir o crime em Chicago. Equipados com sensores que localizam disparos de armas pela cidade, junto com mapas de lojas de bebidas alcoólicas e acessos a rodovias, os pesquisadors identificam áreas onde o crime

provavelmente ocorrerá. Essa informação, combinada com dados sobre eventos esportivos televisionados, aumenta a precisão na localização de possíveis problemas[30].

[30] Esses e outros casos podem ser encontrados em [14].

9.2 COMPONENTES DA ANÁLISE DE DADOS

O foco está no processo de formação de hipóteses, teste de teorias e obtenção de inferências. Vamos abordar essas bases de investigação por meio da visualização de dados e trabalhando em problemas reais com dados reais.

São quatro os componentes da análise de dados nessa ordem:

1. Descrição de dados e formulação de hipóteses.
2. Construção e estimativa de modelos.
3. Diagnósticos.
4. Próxima pergunta.

Existem múltiplos conceitos e técnicas (ou seja, construção e estimativa de modelos, transformação de variáveis, diagnósticos etc.). O propósito deste capítulo é introduzir as linhas gerais estratégias de uma boa análise de dados com um exemplo.

Descrição de Dados e Formulação de Hipóteses:

Descrever dados significa, a princípio, identificar o caso típico (**tendência central**) e entender quão típico é esse caso típico (dispersão). No entanto com as novas ferramentas, deve-se ir muito além disso. Significa, entender, correlacionar e encontrar padões. E as hipóteses? Para nós uma hipótese se referirá a uma suposição específica sobre como duas coisas estão relacionadas (por exemplo, probabilidade de acesso ao meio e vazão total).

Construção e Estimativa de Modelos:

Modelos são versões simplificadas da realidade que nos ajudam a entender nosso mundo complexo. São argumentos para explicar um problema empírico. Por exemplo, se queremos explicar por que alguns países têm altas taxas de homicídio, construímos um modelo que pode incluir renda, idade da população, número de policiais e eficácia do sistema judiciário. Há uma infinidade de outras possíveis causas que poderíamos incluir, mas é útil manter as coisas simples, mas em engenharia não queremos recriar a realidade; queremos apenas aproximá-la.

Diagnósticos:
Depois de construirmos modelos e obtermos algumas estimativas, passamos para os diagnósticos. Diagnósticos são um conjunto de ferramentas que usamos para determinar se estamos usando o tipo certo de modelo. Para verificar se nosso modelo é apropriado, examinamos quão bem as previsões do nosso modelo correspondem à realidade. A diferença entre nossa previsão e a realidade é chamada de erro residual. Por exemplo, se nosso modelo faz um bom trabalho ao prever a vazão total (em bps) em todas as redes, exceto nas redes sem fio em áreas urbanas densas, os diagnósticos resultantes dirão isso. Ou seja, os resíduos para esses casos serão relativamente grandes. Talvez nossas estimativas de modelo estejam sendo excessivamente influenciadas por essas redes sem fio urbanas. Diagnósticos nos ajudam a determinar se nossas estimativas fornecem uma boa noção de como a vazão de rede realmente funciona, são o produto de alguns casos atípicos ou são o resultado de um modelo mal escolhido.

É importante lembrar que diagnósticos podem tanto detectar problemas quanto ajudar a descobrir relações interessantes, gerando explicações adicionais ou hipóteses.

Próximas Perguntas:
Se as estimativas que obtivemos estão corretas, então esperaríamos ver nossa variável acompanhar certo comportamento. Seguir cada conjunto de estimativas com essa declaração ajuda a descobrir explicações possíveis e hipóteses adicionais a serem testadas. Como é impossível provar qualquer coisa com total certeza, o exercício de gerar hipóteses adicionais para testar é extremamente importante.

9.3 DESCREVENDO E FORMULANDO HIPÓTESES

Os testes de hipóteses constituem uma pedra angular, o teste de hipótese trata de determinar a probabilidade de que uma determinada premissa sobre um conjunto de dados seja verdadeira. É um método usado para validar ou refutar suposições, muitas vezes levando a novos *insights* e entendimentos. Na sua essência, envolve a formulação de duas hipóteses concorrentes: a hipótese nula (H_0) e a hipótese alternativa (H_1).

A hipótese nula, H_0, representa uma crença básica. É uma afirmação de nenhum efeito ou nenhuma diferença, como "Não há diferença nas alturas médias entre duas espécies de plantas". Em contrapartida, a hipótese alternativa, H1, representa o que procuramos estabelecer. É uma afirmação de efeito ou diferença, como "Há uma diferença significativa nas alturas médias entre estas duas espécies".

Para decidir entre essas hipóteses, usamos um valor p, uma estatística crucial no teste de hipóteses. O valor p nos diz a probabilidade de observar nossos dados, ou algo mais extremo, se a hipótese nula fosse verdadeira. Um valor

p (*p-value*) baixo (geralmente abaixo de 0,05) sugere que os dados observados são improváveis sob a hipótese nula, levando-nos a considerar a hipótese alternativa.

No contexto do teste de correlação de Pearson, o valor p desempenha um papel fundamental na determinação da significância estatística da correlação observada entre duas variáveis. O teste de Pearson avalia a força e a direção da relação linear entre duas variáveis contínuas. Ao realizar este teste, calculamos o coeficiente de correlação de Pearson (r), que pode variar de -1 a 1. No entanto, para inferir se a correlação observada é estatisticamente significativa, analisamos o valor p associado.

Quando o valor p é baixo, indica que a probabilidade de obter um coeficiente de correlação tão extremo quanto o observado, se a hipótese nula de correlação zero fosse verdadeira, é pequena. Por exemplo, um valor p menor que 0,05 sugere que há menos de 5% (0.05) de chance de a correlação observada ser devida ao acaso, fornecendo evidências contra a hipótese nula e a favor de uma correlação verdadeira entre as variáveis. Portanto, a interpretação do valor p no teste de Pearson nos ajuda a decidir se podemos rejeitar a hipótese nula e aceitar a hipótese alternativa de que existe uma correlação significativa.

No entanto, o teste de hipóteses não está isento de riscos, nomeadamente erros do Tipo I e do Tipo II[31].

Um erro Tipo I, ou falso positivo, ocorre quando rejeitamos incorretamente uma hipótese nula verdadeira. Por exemplo, concluir que um novo medicamento é eficaz quando não o é, seria um erro do Tipo I. Este tipo de erro pode levar a uma falsa confiança em tratamentos ou intervenções ineficazes.

Por outro lado, um erro Tipo II, ou falso negativo, ocorre quando não conseguimos rejeitar uma hipótese nula falsa. Isto seria como não reconhecer a eficácia de um medicamento benéfico. Os erros do tipo II podem levar à perda de oportunidades de intervenções ou tratamentos benéficos.

O valor crítico é o ponto de corte que determina a fronteira entre a região onde rejeitamos a hipótese nula (H_0) e a região onde não a rejeitamos, com base no nível de significância (α) do teste. É calculado de modo que a probabilidade de cometer um erro do Tipo I (rejeitar H_0 quando H_0 é verdadeira) seja igual a α. Por exemplo, em um teste unilateral com nível de significância de 5% ($\alpha = 0.05$), o valor crítico na distribuição normal padrão seria aproximadamente 1.645. Se a estatística de teste exceder este valor crítico, rejeitamos H_0, caso contrário, não a rejeitamos.

Na prática, os erros do tipo II têm uma relação inversa com o poder estatístico. Alto poder estatístico terá baixo erro Tipo II.

O equilíbrio entre esses erros é crucial. O nível de significância, muitas vezes fixado em 0,05, ajuda a controlar a taxa de erros do Tipo I. No entanto, a redução dos erros do Tipo I pode aumentar a probabilidade de erros do Tipo II. Assim, a análise estatística não consiste apenas na aplicação de uma fórmula; requer uma consideração cuidadosa do contexto, dos dados e das implicações potenciais de ambos os tipos de erros.

[31] Uma outra maneira de ver é a seguinte: Ao vc sentir alguns sintomas, a Hipótese nula é o *status quo*: você está saudável. Então se o teste de dengue dá positivo, mas você não está doente de fato – isto é um falso positivo, ou seja o exame nos leva a rejeitar a H_0. O caso oposto é o falso negativo, na realidade vc tem dengue, mas o teste dá negativo, ou seja, um falso negativo. 🥵

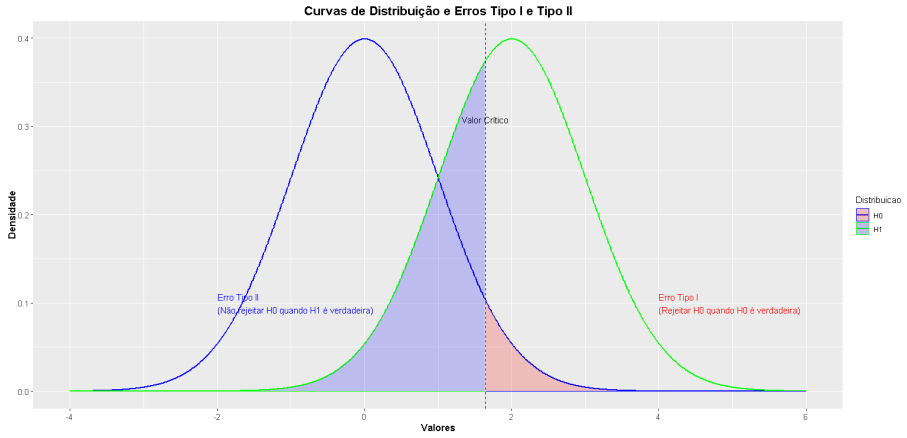


Figura 9.3.1: duas curvas de distribuição de probabilidades, uma para a hipótese nula (H_0) e outra para a hipótese alternativa (H_1). As áreas sombreadas em vermelho e azul ilustram os erros do Tipo I e do Tipo II, respectivamente.

A programação R, com seu conjunto abrangente de ferramentas estatísticas, simplifica a aplicação de testes de hipóteses. Ele não apenas realiza os cálculos necessários, mas também auxilia na visualização dos dados, o que pode fornecer *insights* adicionais. Através do R, podemos executar com eficiência vários testes de hipóteses, desde testes t simples até análises mais complexas, tornando-o uma ferramenta inestimável tanto para estatísticos quanto para pesquisadores de dados.

Em resumo, o teste de hipóteses é um método, que se observado com rigor e correção, suporta a decisão dentro de fatores repetíveis. Requer uma compreensão de conceitos estatísticos como hipóteses nulas e alternativas, valores p e os tipos de erros que podem ocorrer.

9.3.1 Teste de Hipótese – Um Exemplo Prático

Nesta seção, demonstraremos como conduzir um teste de hipóteses[32] no R com um conjunto de dados do mundo real. Exploraremos o conjunto de dados `PlantGrowth`[33], incluído no R, que contém dados sobre plantas em diferentes condições de crescimento. Nosso objetivo será determinar se há uma diferença estatisticamente significativa no crescimento das plantas entre dois grupos de tratamento.

[32] O teste apresentado na Seção 9.3.1 foi extraído de [15].

[33] O R possui pacotes, *libraries* e *datasets* para auxiliar a comunidade a aprender e ensinar. o `PlantGrowth` é um dentre tantos.

Formulação da Hipótese:

A hipótese nula (H_0) é a manutenção do *status quo*, ou seja, afirma que não há diferença no crescimento médio das plantas entre os dois grupos. A hipótese alternativa (H_1) postula que existe uma diferença significativa.

Conduzindo o Teste de Hipótese:
Precisamos descobrir o números de grupos do *dataset* e em seguida aplicar de um teste t para comparar os pesos médios das plantas entre dois dos grupos que se quer comparar. Este teste é apropriado para comparar as médias de dois grupos independentes.

Passo 1: instalar os pacotes apropriados ver Listagem 9.1.

```
>install.packages("easystats")
>library(easystats)
```

Listagem 9.1: comando para instalar e carregar a bibliotrca easystat.

Passo 2: exibição da estrutura e sumário dos dados, ver Listagem 9.2.

```
> head (PlantGrowth)
weight group
1   4.17  ctrl
2   5.58  ctrl
3   5.18  ctrl
4   6.11  ctrl
5   4.50  ctrl
6   4.61  ctrl
> summary (PlantGrowth)
weight       group
Min.   :3.590   ctrl:10
1st Qu.:4.550   trt1:10
Median :5.155   trt2:10
Mean   :5.073
3rd Qu.:5.530
Max.   :6.310
```

Listagem 9.2: comando para exibição da estrutura e sumário do *dataframe* no R.

Passo 4: formulando as hipóteses e conduzindo o teste da hipótese.

Nossa hipótese nula (H_0) afirma que não há diferença no crescimento médio das plantas entre os dois grupos (*status quo*). A hipótese alternativa (H_1) postula que existe uma diferença significativa. E aplicação do teste[34] t, para comparar os pesos médios das plantas entre dois dos grupos. Este teste é apropriado para comparar as médias de dois grupos independentes.

[34] O teste t é empregado para comparar as médias de dois grupos e determinar se elas são significativamente diferentes uma da outra. O teste t é baseado na distribuição t de Student e é especialmente útil quando as amostras são pequenas e a variância é desconhecida. E amostras independentes são aquelas em que as observações de uma amostra não têm qualquer relação com as observações da outra amostra.

```
> result <- t.test(weight ~ group,
> data = PlantGrowth,
> subset = group %in% c("ctrl", "trt1"))
```

Listagem 9.3: aplicação do test t para comparação do subgrupo (ctrl e trt1) no R.

Na Listagem 9.3, `weight` é a variável de interesse (dependente) e `group` é a variável que define os grupos (a variável independente). E a comparação deve se dar entre o grupo de controle e um segundo que queremos avaliar.

Passo 5: avaliação dos resultados, ver Listagem 9.4.

```
> report(result)
```

```
Effect sizes were labelled following Cohen (1988) recommendations.

The Welch Two Sample t-test testing the difference of weight by group
(mean in group ctrl = 5.03, mean in group trt1 = 4.66) suggests that
the effect is positive, statistically not significant, and
medium (difference = 0.37, 95% CI [-0.29, 1.03], t(16.52) = 1.19,
p = 0.250; Cohen d = 0.59, 95% CI [-0.41,1.56])
```

Listagem 9.4: apresentação dos resultados.

A função `result` gera um relatório teste t, incluindo a estimativa, o intervalo de confiança e o valor p.

Interpretando e visualizando os Resultados A saída da função de relatório nos dirá se a diferença nas médias é estatisticamente significativa. Um valor p menor que 0,05 geralmente indica que a diferença é significativa, e podemos rejeitar a hipótese nula em favor da alternativa. No entanto, se o valor p for maior que 0,05, não temos evidências suficientes para rejeitar a hipótese nula.

De modo que, olhando para nossos resultados, podemos avaliar que há uma certa diferença na medida que estamos verificando, mas de acordo com o valor de p alto, nos habilita a dizer que essa diferença pode ser simplesmente uma questão de acaso, não sendo estatisticamente significativa [35]. Acompanhar a tradução gráfica do resultado[36] na Figura 9.3.2. O código está na Listagem 9.5.

```
>library(ggplot2)
>ggplot(PlantGrowth, aes(x = group, y = weight)) +
geom_boxplot() +
theme_minimal() +
labs(title = "Crescimento das Plantas por Grupo de  Tratamento",
x = "Grupo",
y = "Peso")
```

Listagem 9.5: apresenta os resultados em forma gráfica.

É possível concluir que os grupos ctrl e trt1 realmente não têm grande diferença, seus intervalos superam um ao outro[37].

[35] **Significância Estatística:** se o valor p é menor que o nível de significância escolhido (por exemplo, 0,05), rejeitamos a hipótese nula em favor da hipótese alternativa, indicando que a diferença observada é estatisticamente significativa.

Não Significância Estatística: se o valor p é maior que o nível de significância, não rejeitamos a hipótese nula, indicando que não há evidências suficientes para afirmar que a diferença observada é estatisticamente significativa.

[36] **Explore Antes de Testar:** Familiarize-se com seu conjunto de dados antes de realizar testes de hipótese.
Verifique as Assunções: Cada teste estatístico tem suposições (como normalidade, independência ou variância igual).
Escolha o Teste Correto: diferentes testes são projetados para diferentes tipos de dados e objetivos. Por exemplo, use um teste t para comparar médias, testes qui-quadrado para dados categóricos e ANOVA para comparar mais de dois grupos.
Considere Opções Não Paramétricas: Se seus dados não atenderem às suposições dos testes paramétricos (são normais?).

[37] Que tal como exercício tentar verificar o par ctrl e trt2?

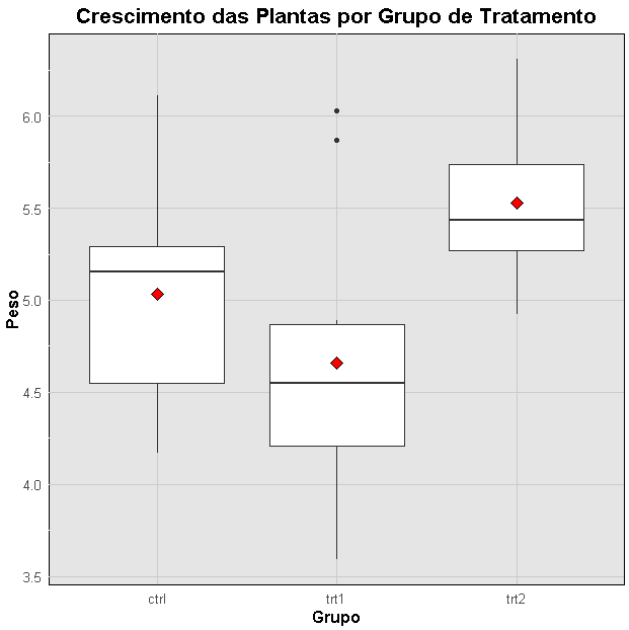


Figura 9.3.2: o *boxplot* exibe a média, a mediana, os quartis e possíveis *outliers*.

9.4 PODER ESTATÍSTICO E DETERMINAÇÃO DO TAMANHO DA AMOSTRA

Embora os pesquisadores compreendam a importância da análise estatística, um número significativo de investigadores admite que lhes falta conhecimento adequado sobre conceitos e princípios estatísticos [16]. De modo que intencionalmente ou não, os pesquisadores tendem a tirar conclusões que podem não ser suportadas pelos dados reais do estudo, muitas vezes devido ao uso inadequado de ferramentas estatísticas[38].

[38] Essa Seção foi fortemente baseada no trabalho publicado em [17].

Embora existam diversos erros estatísticos potenciais que podem ocorrer em qualquer tipo de pesquisa científica, observou-se que as fontes de erro mudaram devido ao uso de softwares dedicados que facilitam a estatística nos últimos anos. Uma síntese dos principais erros estatísticos frequentemente encontrados em estudos científicos é fornecida abaixo:

- Hipótese falha e/ou inadequada.
- Falta de uma condição/grupo de controle adequado,
- Viés de espectro.
- Supervalorização dos resultados da análise.
- Correlações espúrias.

- Tamanho de amostra inadequado.
- *p-hacking* (i.e., adição de novas covariáveis *post hoc* para tornar os valores de p significativos).
- Interpretação excessiva de resultados limitados ou insignificantes (subjetivismo).
- Confusão (intencionais ou não) entre correlações, relações e causações.
- Modelos de regressão defeituosos.
- Confusão entre *p-value* e significância estatística.
- Apresentação inadequada dos resultados e efeitos (tabelas, gráficos e figuras errôneos).

9.4.1 Como estes termos se relacionam

Esse conceito é tão importante que, se você não estiver claro entendimento ou tiver, no máximo, uma compreensão vaga, então você precisa aperfeiçoar sua compreensão antes de interpretar qualquer dado estatístico que faça inferências sobre parâmetros populacionais. Em vez de um argumento teórico longo e prolongado, vamos direto ao ponto e resumimos a questão com algo que você deve sempre ter em mente ao interpretar testes de significância:

Dado até mesmo o menor tamanho de efeito na amostra, com um tamanho de amostra suficientemente grande, a rejeição da hipótese nula é suficientemente garantida para qualquer teste estatístico que tivermos em mente.

O problema, no entanto, é que é totalmente possível obter o infame $p < 0,05$ mesmo em circunstâncias onde a medida do tamanho do efeito é extremamente pequeno. E como isso ocorre?

Pode ocorrer de algumas maneiras, mas de longe a maneira mais comum é quando o tamanho da amostra é grande. Por exemplo, uma diferença média na amostra de um grupo de controle versus um grupo experimental, independentemente de quão pequena essa diferença possa ser, pode se tornar estatisticamente significativa se o tamanho da amostra for grande o suficiente.

Portanto, suponha que nossa conclusão está baseada numa diferença entre a média de um evento entre dois grupos e encontremos uma diferença média na amostra de 0,00001 entre os escores de bem-estar daqueles tratados com terapia cognitiva *versus* terapia comportamental em algum experimento. Você teria que concordar que a diferença é insignificante, é trivial. É extremamente pequena a ponto de quase ser invisível. No entanto, com um tamanho de amostra grande o suficiente, pode-se obter $p < 0,05$ e rejeitar a hipótese nula de não diferença na população da qual os dados foram retirados! E, embora esse fato não seja alarmante em um nível teórico (ou seja, faz perfeito sentido para os teóricos).

Então, cuidado ao tirar conclusões substantivas de uma "diferença real" com base apenas nos valores de p. Encontrar $p < 0,05$ não é exatamente efeito científico, um senso de "sucesso" para o seu experimento. Agora demonstraremos esse efeito com um exemplo muito simples.

9.4.2 Como surge o $p < 0{,}05$?

Exemplos simples são mais simples... Suponha que você é um psicólogo que cuja hipótese é que a implementação de um novo programa para estudantes do ensino médio terá o efeito de melhorar as notas de um teste. Para simplificar, suponha que a média de **todos** os estudantes nesse teste seja igual a 100. Ou seja, $\mu = 100$, onde μ é a média populacional. Se o seu programa for eficaz, você esperaria mostrar uma média superior a 100 para aqueles que participam do seu programa. Ou seja, sua hipótese alternativa é que $\mu \neq 100$, ou mais especificamente, $\mu > 100$. Para ajudar a avaliar sua teoria, você seleciona aleatoriamente 100 estudantes do estado e os inscreve no programa de 3 meses. Então, ao final desse período, eles são submetidos a uma bateria de realização padronizada. Suponha que a média de realização da amostra seja 101. Ou seja, $\bar{y} = 101$. O que você acha do resultado?

Esqueça qualquer estatística por um momento. Você acha que o resultado é impressionante? Provavelmente não, porque a média da amostra é apenas uma unidade diferente da média populacional. Ou seja, $101 - 100 = 1$. Esse é um resultado impressionante? Provavelmente não, e se não realizarmos nenhuma inferência estatística, provavelmente concluiremos simplesmente que temos evidências insuficientes para pensar que o programa é eficaz. Afinal, se fosse eficaz, provavelmente esperaríamos uma diferença muito maior, digamos de 5 a 10 pontos, talvez mais. Mas a amostra resultou em 101, o que está muito próximo do que esperaríamos sob a hipótese nula (lembre-se de que a expectativa sob a nula era 100).

Você escolhe avaliar seu resultado para significância estatística de qualquer forma e, após calcular o desvio padrão na amostra como sendo igual a 10, você calcula o *t-statistic*[39] resultante na Equação 2:

$$t = \frac{\bar{y} - \mu_0}{\frac{s}{\sqrt{n}}} = \frac{101 - 100}{\frac{10}{\sqrt{100}}} = \frac{1}{1} = 1 \tag{2}$$

O *t-statistic* $= 1$ e, quando avaliada com 99 graus de liberdade[40] , o valor mostra seus resultados como sendo estatisticamente não significativo. Portanto, você não rejeita a hipótese nula não pode ser rejeitada. Ou seja, você tem evidências insuficientes para rejeitar $\mu = 100$. Nesse caso, o teste de significância mais ou menos concorda com sua intuição científica de que o programa não é eficaz. Ou seja, você não rejeitou a hipótese nula, obteve um $p - value$ relativamente alto (Ver Listagem 9.6), e tudo isso converge com a percepção de que não há nenhum efeito científico.

```
1  # Definir a estatística t e os graus de liberdade
```

[39] O **teste t** é um método estatístico utilizado para comparar as médias de dois grupos e determinar se elas são significativamente diferentes entre si. Ele é especialmente útil quando se tem um tamanho de amostra pequeno e a variância populacional é desconhecida. O valor resultante é então comparado com uma distribuição t de Student com $n - 1$ graus de liberdade para determinar a significância estatística.

[40] **Teste t e graus de liberdade**: para um teste t, os graus de liberdade (df) são calculados com base no tamanho da amostra. No caso de um teste t para uma média de uma amostra, os graus de liberdade são geralmente iguais ao tamanho da amostra menos um $(n - 1)$.

```
>t_statistic <- 1
>degrees_freedom <- 99

# Calcular o p-valor para um teste bicaudal
>p_value <- 2 * pt(-abs(t_statistic), df = degrees_freedom)

# Imprimir o p-valor
>print(p_value)
[1] 0.3197485
```

Listagem 9.6: cálculo do *p-value* a partir do teste *t-statistic*.

Em outras palavras, o programa não funciona, e tudo está bem, você pode seguir para outro experimento ou tentar algo que possa funcionar.

Agora, considere o mesmo estudo acima, exceto que, em vez de testar uma amostra de 100 estudantes, você decidiu testar uma amostra de 500. No entanto, a diferença nas médias permaneceu a mesma, assim como o desvio padrão. O novo cálculo para t pode ser acompanhado na Equação **??**:

$$t = \frac{\bar{y} - \mu_0}{\frac{s}{\sqrt{n}}} = \frac{101 - 100}{\frac{10}{\sqrt{500}}} = \frac{1}{0.45} = 2.22 \tag{3}$$

Observe que t saiu muito maior desta vez, mesmo que a diferença entre as médias no numerador tenha permanecido exatamente a mesma. Ou seja, note que a diferença entre as médias ainda é igual a 1 ponto. Este t particular de 2,22, se avaliado para significância estatística, atenderia ao critério de 0,05. Portanto, a hipótese nula seria rejeitada, e você poderia dizer que encontrou evidências de uma diferença nas médias! Assim, você estaria completamente honesto ao fazer uma declaração conclusiva como:

Rejeitamos a hipótese nula de que a média do programa μ é igual a 100 e concluímos que não é. O resultado é estatisticamente significativo em $p = 0.02 < 0.05$.

Embora a declaração acima seja tecnicamente correta, ela está sujeita a mal-entendidos se alguém não compreender primeiro a composição de um teste de significância e as influências sobre o que torna um *p-value* pequeno. Como vimos, um aumento no tamanho da amostra, tudo o mais constante (por exemplo, a diferença nas médias permanece a mesma), tem o efeito de virtualmente garantir que, em algum momento, a hipótese nula será rejeitada.

Mas, é claro, esse não foi o motivo do estudo. O estudo foi feito para verificar se o seu programa resultou em um aumento apreciável na média. O fato de ter obtido $p < 0,05$ não deve impressionar. O que deve impressionar ou não, é o tamanho do efeito, que, como veremos, não é suscetível a flutuações no tamanho da amostra como é o teste de significância.

9.4.3 Tamanho do Efeito e o *d* de Cohen

Lembre-se de que mencionamos anteriormente que os tamanhos de efeito, em essência, nos fornecem uma medida do "que aconteceu"em nosso estudo ou

experimento. Eles são uma medida do efeito científico. E, enquanto o teste de significância é sensível ao tamanho da amostra, os tamanhos de efeito são menos sensíveis. No mínimo, eles não são simplesmente uma função do tamanho da amostra. À medida que se coleta uma amostra maior, o tamanho do efeito pode ou não aumentar, enquanto para o teste de significância, o valor de z, t ou F (ou qualquer que seja a estatística inferencial que se está calculando para o estudo em questão) é praticamente garantido de aumentar em valor. No nosso exemplo acima, o de uma diferença de médias, uma medida comum de tamanho de efeito é calcular a diferença nas médias e dividir pelo desvio padrão. Ou seja, calculamos na Equação 4:

$$d = \frac{|\bar{y} - \mu_0|}{\sigma} = \frac{|101 - 100|}{10} = 0.1 \tag{4}$$

onde $\bar{y}$ é a média da amostra experimental, μ_0 é a média sob a hipótese nula, e σ é o desvio padrão da população, ou uma estimativa dele na forma de s se não tivermos o valor real. Tecnicamente, como $\bar{y}$ está servindo como a estimativa de μ, o numerador é mais precisamente representado como $\mu - \mu_0$. No entanto, notando $\bar{y}$ em vez de μ nos lembra de onde estamos obtendo essa média. Essa medida de tamanho de efeito é referida como d de Cohen[41] na literatura científica. É uma diferença padronizada e a padronização ocorre dividindo $\bar{y} - \mu_0$ por σ.

[41] O *d* **de Cohen** é uma medida de tamanho de efeito que expressa a diferença entre duas médias padronizada pelo desvio padrão. Ele é calculado como a diferença entre a média da amostra e a média da população sob a hipótese nula, dividida pelo desvio padrão da população. Os valores de *d* de Cohen são geralmente interpretados conforme as diretrizes de Cohen: 0.2 indica um efeito pequeno, 0.5 indica um efeito médio, e 0.8 indica um efeito grande.

Segundo todas os cálculos, incluindo as diretrizes originais de Cohen para o que constitui um tamanho de efeito pequeno (0,2), médio (0,5) ou grande (0,8), um valor de 0,1 é bastante pequeno. Note que o valor de d não é sujeito a flutuações como é o *p-value* com o aumento no tamanho da amostra. Ou seja, se aumentássemos o tamanho da amostra, σ no denominador pode aumentar ou diminuir. Isso é diferente do erro padrão da média apresentado anteriormente, para o qual um aumento no tamanho da amostra quase certamente faria o erro padrão diminuir, o que significaria que a estatística do teste resultante seria maior, levando a mais rejeições da hipótese nula. Os tamanhos de efeito não aumentam automaticamente simplesmente porque se está usando uma amostra maior. Já o *p-value* tipicamente diminui à medida que o tamanho da amostra aumenta.

9.4.3.1 O Veredito Final do Teste de Significância

Os testes de significância da hipótese nula têm seu lugar na ciência, mas devem ser usados com uma compreensão aguda do que os *p-values* podem e não podem fornecer ao pesquisador. A despeito do mau uso, ainda são úteis, pois fornecem uma medida de suporte inferencial para a descoberta científica, mas. Como discutimos, para fazer qualquer declaração de efeito, é necessário calcular o tamanho do efeito. Enquanto os p-valores podem indicar se o suporte inferencial é justificado, o que em si é importante, os tamanhos de efeito nos dão uma noção do grau em que a variável independente explica a variância

na variável dependente, ou uma noção da magnitude da associação encontrada entre as variáveis no experimento ou estudo.

9.4.4 Poder Estatístico e o Tamanho da Amostra

leia https://www.remesh.ai/resources/how-to-calculate-sample-size

A análise de poder estatístico nos permite determinar o tamanho da amostra necessário para detectar um efeito de um determinado tamanho com um grau de confiança especificado. E por outro lado, permite-nos determinar a probabilidade de detectar um efeito de um determinado tamanho com um nível de confiança dado, sob restrições de tamanho da amostra. Se a probabilidade for inaceitavelmente baixa, seria prudente alterar ou abandonar o experimento[42]. As seguintes quatro quantidades têm uma relação íntima:

[42]Esta Seção foi fortemente apoiada no trabalho publicado em [18].

- Tamanho da amostra
- Tamanho do efeito
- Nível de significância = $p = \alpha$, ou seja (Erro Tipo I), isto é, a probabilidade de encontrar um efeito que não está presente.
- Poder = $1 - p = \beta$, ou seja (Erro Tipo II), isto é, a probabilidade de encontrar um efeito que está presente

Dado qualquer três, podemos determinar o quarto.

A Figura 9.4.1, mostra os valores α, β, e os tipos de erro, o poder estatístico associado será analisado em seguida.

O pacote `pwr` implementa a análise de poder com o índice d de Cohen, suas principais funções são mostradas na Tabela 7 e em seguida uma breve explicação do emprego de cada.

A função `pwr.2p.test` realiza o cálculo do poder estatístico para comparar duas proporções com tamanhos de amostra iguais. Esta função determina o tamanho da amostra necessária para detectar uma diferença específica entre duas proporções populacionais.

A função `pwr.2p2n.test` é usada para calcular o poder estatístico ao comparar duas proporções com tamanhos de amostra desiguais. Isso permite uma análise mais flexível quando as amostras de comparação não têm o mesmo número de observações.

A função `pwr.anova.test` calcula o poder estatístico para uma ANOVA unifatorial balanceada. Este teste é empregado para avaliar se existem diferenças significativas entre as médias de três ou mais grupos.

A função `pwr.chisq.test` realiza o cálculo do poder estatístico para o teste qui-quadrado. Este teste é utilizado para examinar a independência entre variáveis categóricas ou a adequação de um modelo teórico às observações empíricas.

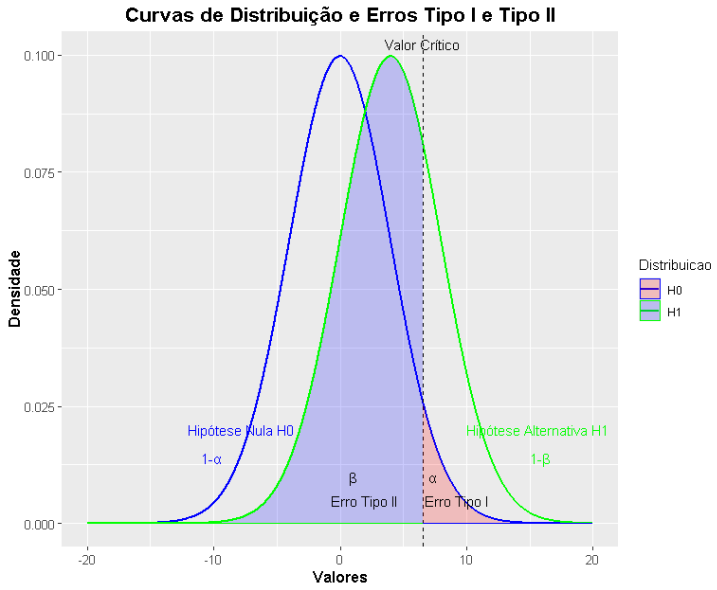


Figura 9.4.1: duas curvas de distribuição de probabilidades, uma para a hipótese nula (H_0) e outra para a hipótese alternativa (H_1). As áreas sombreadas vermelho e azul ilustram os erros do Tipo I e do Tipo II, α e β respectivamente.

Função	Aplicações
`pwr.2p.test`	duas proporções (n igual)
`pwr.2p2n.test`	duas proporções (n desigual)
`pwr.anova.test`	ANOVA unifatorial balanceada
`pwr.chisq.test`	teste qui-quadrado
`pwr.f2.test`	modelo linear geral
`pwr.p.test`	proporção (uma amostra)
`pwr.r.test`	correlação
`pwr.t.test`	testes t (uma amostra, duas amostras, pareado)
`pwr.t2n.test`	teste t (duas amostras com n desigual)

Tabela 7: algumas funções do Rassociadas ao `pwr` e suas aplicações.

A função `pwr.f2.test` é utilizada no cálculo do poder estatístico para modelos lineares gerais. Este teste avalia a magnitude dos efeitos das variáveis independentes sobre a variável dependente em um modelo de regressão.

A função `pwr.p.test` calcula o poder estatístico para proporções em uma amostra única. É útil para determinar se a proporção observada em uma amostra difere significativamente de uma proporção hipotética.

A função `pwr.r.test` realiza o cálculo do poder estatístico para correlação. Este teste avalia a força e a direção da relação linear entre duas variáveis contínuas.

A função `pwr.t.test` calcula o poder estatístico para testes t em uma amostra, duas amostras e amostras pareadas. Este teste é utilizado para comparar médias e determinar se há diferenças significativas entre os grupos ou condições.

A função `pwr.t2n.test` calcula o poder estatístico para testes t em duas amostras com tamanhos desiguais. Esta função é útil quando se comparam médias de dois grupos com diferentes números de observações.

9.4.4.1 Testes t e *d* de Cohen

Para o teste t, que empregamos na Seção 9.4.2 use a seguintes funções:

```
pwr.t.test(n=, d=, sig.level=, power=, type= c("sample_1",
"sample_2", "paired"))
```

onde n é o tamanho da amostra, d é o tamanho do efeito, e *type* indica um teste t de duas amostras, uma amostra ou pareado[43].

Se você tiver tamanhos de amostra desiguais, use:

```
pwr.t2n.test(n1=, n2=, d=, sig.level=, power=)
```

onde `n1` e `n2` são os tamanhos das amostras.

Para testes t, o tamanho do efeito é avaliado pelo d de Cohen Cohen sugere que valores de d de 0.2, 0.5 e 0.8 representam tamanhos de efeito pequenos, médios e grandes, respectivamente, acompanhe a Figura 9.7.

[43] Um **teste t pareado**, também conhecido como teste t de amostras pareadas, é um tipo de teste estatístico usado para comparar as médias de dois conjuntos de dados que estão emparelhados de alguma forma. Isso significa que cada observação em um conjunto de dados está diretamente relacionada ou emparelhada com uma observação no outro conjunto de dados. Exemplos comuns de dados emparelhados incluem medições antes e depois de um tratamento no mesmo grupo de indivíduos.

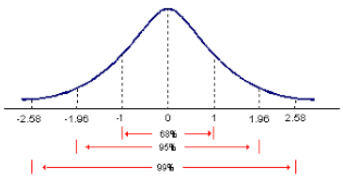


Figura 9.4.2: relação entre intervalo de confiança, valor crítico z e desvio padrão.

9.4.4.2 Determinação do Tamanho da amostra

Para garantir que um teste t tenha poder estatístico suficiente, é essencial determinar o tamanho da amostra necessário. O poder do teste é a probabilidade de rejeitar a hipótese nula quando ela é falsa, e depende de vários fatores, incluindo o tamanho do efeito, o nível de significância e o tamanho da amostra.

Fórmula Geral

A fórmula básica para calcular o tamanho da amostra n necessário para um teste t é derivada das fórmulas de poder de teste e envolve a resolução de n na equação de poder. Para um teste t de duas amostras, é dada pela Equação 5:

$$n = \left(\frac{Z_{\alpha/2} + Z_{\beta}}{d} \right)^2 \tag{5}$$

onde:

- n é o tamanho da amostra necessário para cada grupo,

- $Z_{\alpha/2}$ é o valor crítico da distribuição normal correspondente ao nível de significância α (por exemplo, 1,96 para 95% de confiança – Figura 9.4.2).
- Z_{β} é o valor crítico da distribuição normal correspondente ao poder do teste $1 - \beta$ (por exemplo, 0,84 para 80% de poder).
- d é o tamanho do efeito (d de Cohen), que é uma medida da magnitude da diferença entre as médias dos grupos.

Exemplo Prático Usando R Para calcular o tamanho da amostra necessário em R, podemos usar a função `pwr.t.test` da biblioteca `pwr`. Suponha que desejamos calcular o tamanho da amostra necessário para um teste t de duas amostras com um tamanho de efeito $d = 0.75$ (ver Figura 9.4.3), um nível de significância de 0.05 e um poder de 0.80. Ver Listagem 9.7.

```
1  # install.packages("pwr")
2  >library(pwr)
3
4  # Definir os parâmetros
5  >d <- 0.75          # Tamanho do efeito
6  >sig.level <- 0.05  # Nível de significância
7  >power <- 0.80      # Poder desejado
8
9  # Calcular o tamanho da amostra
10 >result <- pwr.t.test(d = d, sig.level = sig.level, power = power,
     type = "two.sample", alternative = "greater")
11
12 # Exibir o resultado
13 >print(result)
14
15 Two-sample t test power calculation
16
17 n = 22.69033
18 d = 0.75
19 sig.level = 0.05
20 power = 0.8
21 alternative = greater
22
23 NOTE: n is number in *each* group
24
```

Listagem 9.7: cálculo do tamanho da amostra, dados os parâmetros restantes.

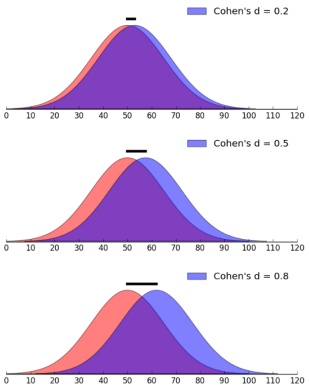


Figura 9.4.3: interpretação geométrica do d de Cohen. Fonte: `https://scientificallysound.org/2017/07/27/cohens-d-how-interpretation/`.

O resultado da função `pwr.t.test` incluirá o tamanho da amostra necessário para cada grupo no teste t de duas amostras, dado o tamanho do efeito, o nível de significância e o poder especificados.

Interpretação do Resultado: Este resultado indica que, com um tamanho de amostra maior que 22 em cada grupo, um tamanho de efeito de 0.75, um nível de significância de 0.05 e assumindo uma hipótese alternativa unicaudal (`greater` - na linha 21), o teste tem 80% de chance de detectar um efeito real, caso ele exista.

10 Construção e Estimativa de Modelos

Um objetivo comum para o desenvolvimento de um modelo é prever qual será o valor de saída de um sistema para conjunto de valores de entrada. A ideia é discutir como desenvolver o modelo, como avaliar até que ponto o modelo criado se ajusta aos dados e como interpretar os resultados[44].

[44] A Seção 10, seus exemplos e gráficos, embora refeitos pelo autor com uma biblioteca mais moderna, foram baseados no trabalho publicado em [19].

Suponha que medimos o desempenho de vários dispositivos computacionais. Podemos organizar as $n \times k$ medições, mostradas na Tabela 8 Como medimos o desempenho de n dispositivos diferentes, obteremos n linhas na tabela. Cada linha é chamada de "observação única".

Tabela 8: Um exemplo em que queremos prever o desempenho de novos sistemas $n+1, n+2$ e $n+3$ usando o medido anteriormente resultados dos outros n sistemas [19].

System	**Clock (MHz)**	**Cache (kB)**	**Transistors (M)**	**Output Performance**
1	1500	64	2	98
2	2000	128	2.5	134
⋮	⋮	⋮	⋮	⋮
i	...	...	...	...
⋮	⋮	⋮	⋮	⋮
n	1750	32	4.5	113
$n+1$	2500	256	2.8	?
$n+2$	1560	128	1.8	?
$n+3$	900	64	1.5	?

O objetivo da modelagem é usar essas k medidas independentes para determinar uma função f, que descreva as relações entre os parâmetros de entrada e a saída, por exemplo $desempenho = f(clock, cache, transistors)$. Um dito modelo de regressão pode assumir qualquer forma. Nos restringiremos a uma função que é uma combinação linear (regressão linear) dos parâmetros de entrada. Mas note que, embora a função seja linear, os parâmetros em si não precisam ser lineares.

10.1 A REGRESSÃO LINEAR SIMPLES - SLR

A regressão linear é um método estatístico utilizado para modelar a relação entre uma variável dependente (**resposta**) e uma ou mais variáveis independentes (**preditores**). A equação da regressão linear múltipla pode ser expressa da seguinte forma[45]:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \tag{6}$$

onde:

- Y é a variável resposta (dependente).

[45] Em notação matemática, letras maiúsculas denotam variáveis aleatórias ou conjuntos de dados:
– Y representa a variável resposta como um conjunto de observações.
– $X_1, X_2, \ldots, X_p$ representam as variáveis preditoras como conjuntos de observações.
Na prática, o intercepto (β_0) representa o ponto onde a linha de regressão cruza o eixo y no gráfico de dispersão.

- $X_1, X_2, \ldots, X_p$ são as variáveis preditoras (independentes).
- β_0 é o intercepto.
- $\beta_1, \beta_2, \ldots, \beta_p$ são os coeficientes de regressão associados às variáveis preditoras.
- ϵ é o erro, que captura a variação não explicada pelo modelo.

Os coeficientes de regressão (β_i) são estimados de forma a minimizar a soma dos quadrados dos resíduos (diferença entre os valores observados e os valores preditos pelo modelo).

$$\min_{\beta_0,\beta_1,\ldots,\beta_p} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{7}$$

onde $\hat{Y}_i$ é o valor predito pelo modelo para a i-ésima observação.

O primeiro passo no processo de modelagem com um único preditor (regressão linear simples - SLR) é determinar se parece haver uma relação entre o preditor e o valor de saída. Com base no conhecimento sobre projeto de dispositivos computacionais, sabemos que a frequência do *clock* influencia fortemente o desempenho do sistema. De forma que é esperada uma correlação entre o desempenho do processador [46] e sua frequência de *clock*.

[46] O *Integer Component of* SPEC CPU2000, cujo gráfico está apresentado na Figura 10.1.2 é uma parte do conjunto de *benchmarks* desenvolvido pela *Standard Performance Evaluation Corporation* (SPEC) para avaliar o desempenho de CPUs em tarefas que envolvem cálculos inteiros, é um conjunto de testes de desempenho de sistemas de computação, testa cálculos com inteiros e ponto flutuantes.

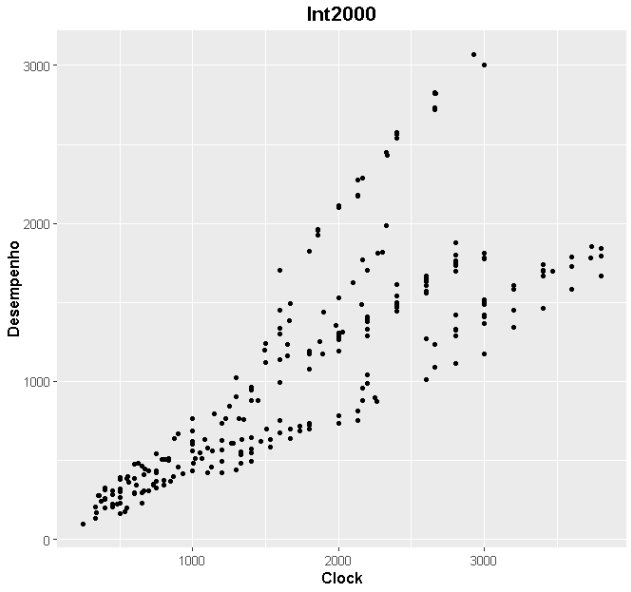


Figura 10.1.1: Um gráfico de dispersão do desempenho dos processadores que testamos usando o *benchmark* Int2000 versus a frequência do *clock*.

A varável independente neste caso é o *clock* e a variável dependente é o desempenho. Se sobrepusermos uma linha reta a este gráfico de dispersão, vemos que há uma aparente relação entre o preditor (a frequência do *clock*) e a saída (o desempenho). O gráfico ainda mostra que esta relação não é perfeitamente linear. À medida que a frequência do *clock* aumenta, vemos uma maior espalhamento em valores de desempenho. Nosso próximo passo é desenvolver uma regressão modelo que nos ajudará a quantificar o grau de linearidade na relação entre a saída e o preditor.

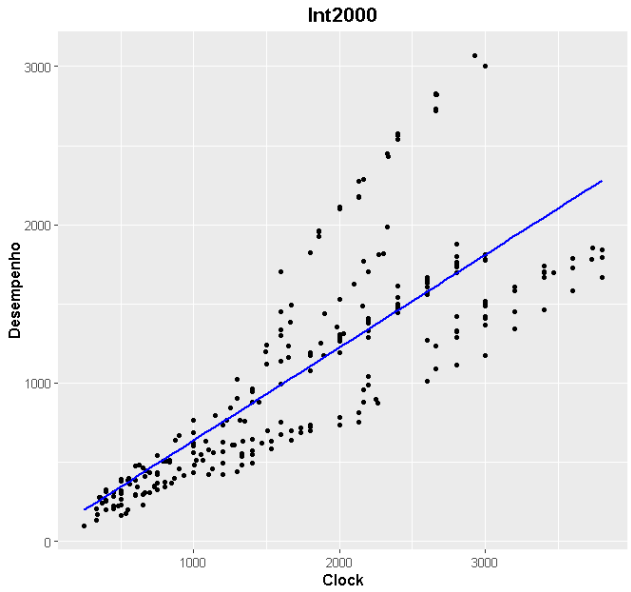


Figura 10.1.2: o modelo de regressão linear simples (em azul) sobreposto aos dados da Figura 10.1.1.

As informações que obtemos digitando o comando `int00.lm` alguns valores básicos do modelo, mas não nos diz nada sobre as qualidades do modelo.

```
> summary(int00.lm)

Call:
lm(formula = perf ~ clock, data = int00.dat)

Residuals:
Min      1Q  Median      3Q     Max
-634.61 -276.17  -30.83   75.38 1299.52

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 51.78709   53.31513   0.971    0.332
clock        0.58635    0.02697  21.741   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘’. 0.1 ‘ ’ 1

```

```
17      Residual standard error: 396.1 on 254 degrees of freedom
18      Multiple R-squared:  0.6505,    Adjusted R-squared:  0.6491
19      F-statistic: 472.7 on 1 and 254 DF,  p-value: < 2.2e-16
20
21
```

Listagem 10.1: apresentação dos resultados do comando `summary(int00.lm)`.

10.1.1 Análise dos Resultados da Regressão Linear

Primeiramente vamos dissecar os valores dos resíduos.

Resíduos:

- **Min:** -634.61
- **1Q:** -276.17
- **Mediana:** -30.83
- **3Q:** 75.38
- **Max:** 1299.52

Esses valores indicam a distribuição dos resíduos. A mediana (-30.83) próxima de de zero (em comparação ao valor máximo 1299.52) sugere que os resíduos estão aproximadamente balanceados em torno de zero, o que é um bom sinal. No entanto, a diferença entre os valores mínimo e máximo é bastante grande, indicando a presença de *outliers*.

Coeficientes:

- ***Residual standard error*:** 396.1 on 254 *degrees of freedom*
- ***Multiple R-squared*:** 0.6505,
- ***Adjusted R-squared*:** 0.6491
- ***F-statistic*:** 472.7 on 1 and 254 DF,
- ***p-value*** $<$ 2.2e-16

O valor de R-quadrado indica que aproximadamente 65.05% da variação no desempenho pode ser explicada pela variação no *clock*. Isso sugere um bom ajuste do modelo, mas ainda há 34.95% da variação que não é explicada pelo modelo linear.

No contexto de um modelo de regressão linear, o F-*value* é uma medida estatística que testa a significância global do modelo. Ele é usado para avaliar

se existe uma relação linear entre a variável dependente e as variáveis independentes no modelo.

Os graus de liberdade (DF, do inglês *Degrees of Freedom*) são importantes para o cálculo do F-*value*. No caso "1 and 254 DF", o primeiro número (1) representa os graus de liberdade do numerador, que correspondem ao número de variáveis independentes no modelo (neste caso, apenas uma variável independente: *clock*). O segundo número (254) representa os graus de liberdade do denominador, que são baseados no número de observações menos o número de parâmetros estimados (neste caso, 256 observações menos 2 parâmetros: o intercepto e a inclinação).

Um *F-value* alto e um *p-value* muito baixo indicam que o modelo de regressão linear é estatisticamente significativo e que a variável independente (*clock*) está fortemente associada à variável dependente (desempenho).

Em resumo:

- O modelo de regressão linear sugere uma forte relação entre a frequência de *clock* e a performance do processador.
- O coeficiente do *clock* é altamente significativo, indicando que aumentos na frequência de clock estão fortemente associados a aumentos na performance.
- A mediana dos resíduos próxima de zero sugere um bom ajuste, embora a presença de outliers (valores mínimos e máximos distantes) possa indicar a necessidade de verificar a presença de pontos atípicos ou considerar modelos mais complexos.
- O R-quadrado de 65.05% indica que o modelo explica uma parte significativa da variação na performance, mas há espaço para melhorias.

10.1.2 Análise dos Resíduos

A função `summary()` fornece uma quantidade substancial de informações para nos ajudar a avaliar o ajuste de um modelo de regressão aos dados utilizados para desenvolvê-lo. Para aprofundar a análise da qualidade do modelo, precisamos examinar informações adicionais sobre os valores observados em comparação com os valores previstos pelo modelo. Em particular, a análise de resíduos examina esses valores residuais para entender melhor a qualidade do modelo.

Lembre-se de que o valor residual é a diferença entre o valor medido real e o valor que a linha de regressão ajustada prevê para aquele ponto de dados correspondente. Valores residuais maiores que zero significam que o modelo de regressão previu um valor muito pequeno em comparação com o valor medido real, e valores negativos indicam que o modelo previu um valor muito grande. Um modelo que se ajusta bem aos dados tenderia a superestimar e subestimar os valores com a mesma frequência. Assim, ao que parece a primeira vista, dado

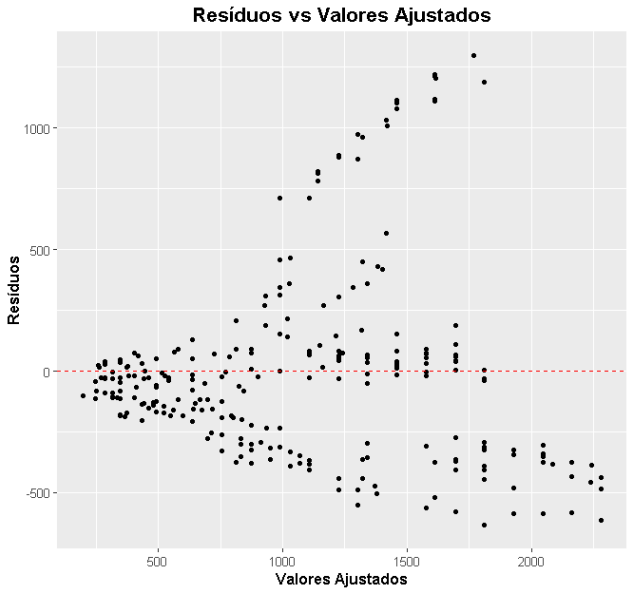


Figura 10.1.3: Os valores residuais versus os valores de saída do modelo SLR desenvolvido usando os dados do Int2000.

a conformação geométrica dos pontos em torno da reta ajustada, se plotarmos os valores residuais, esperaríamos vê-los distribuídos normalmente em torno de zero para um modelo bem ajustado.

Neste gráfico da Figura 10.1.3, vemos que os resíduos tendem a aumentar conforme nos movemos para a direita. Além disso, os resíduos não estão uniformemente dispersos acima e abaixo de zero. No geral, esse gráfico nos diz que usar o *clock* como único preditor no modelo de regressão não explica suficientemente ou totalmente os dados. Em geral, se você observar qualquer tipo de tendência ou padrão claro nos resíduos, provavelmente precisará gerar um modelo melhor. Isso não significa que nosso modelo de regressão linear simples seja inútil. Significa apenas que podemos construir um modelo que produza valores residuais mais ajustados e melhores previsões.

Outro teste dos resíduos utiliza o gráfico quantil-quantil[47], ou Q-Q plot. Anteriormente, dissemos que, se o modelo se ajustasse bem aos dados, esperaríamos que os resíduos fossem distribuídos normalmente (Gaussianamente) em torno de uma média de zero. O Q-Q plot fornece uma boa indicação visual de se os resíduos do modelo são distribuídos normalmente. As chamadas de função a seguir geram o Q-Q plot mostrado na Figura

[47] O gráfico Q-Q (Quantil-Quantil) é uma ferramenta diagnóstica usada para comparar a distribuição dos resíduos de um modelo de regressão com uma distribuição teórica, normalmente a distribuição normal

Se os resíduos fossem normalmente distribuídos, esperaríamos que os pontos plotados nesta figura seguissem uma linha reta. No entanto, com nosso modelo, vemos que as extremidades divergem consideravelmente dessa linha. Esse comportamento indica que os resíduos não são normalmente distribuídos. A forma

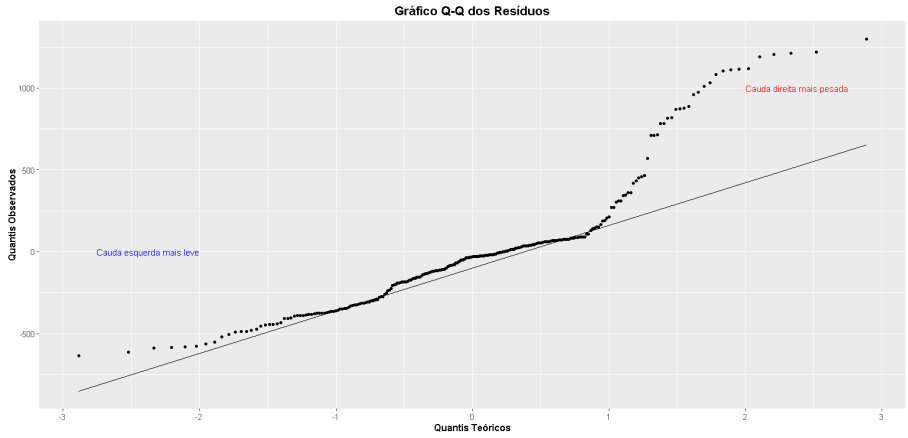


Figura 10.1.4: gráfico Q-Q para o modelo de regressão linear com os dados do Int2000.

como as caudas se desviam da linha de referência pode indicar como os resíduos observados se desviam do esperado caso fossem normalmente distribuídos.

Na verdade, este gráfico sugere que a cauda direita da distribuição é "mais pesada"do que o esperado de uma distribuição normal e que a cauda esquerda é "mais leve"do que o esperado. Esse padrão é indicativo de uma distribuição enviesada à direita. Este teste confirma ainda mais que usar apenas o *clock* como preditor no modelo é insuficiente para explicar os dados.

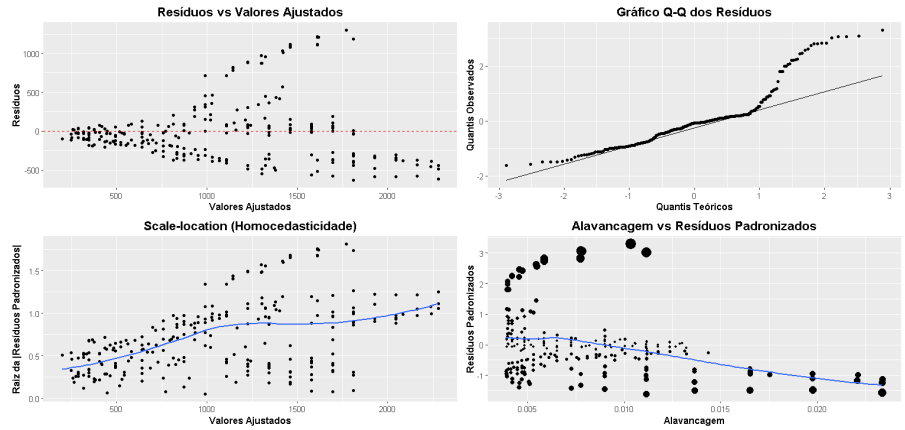


Figura 10.1.5: gráficos de análise mais profunda dos resíduos e sua possível normalidade.

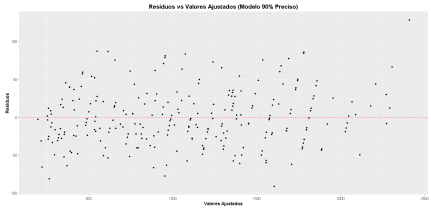
Dos 4 gráficos apresentados na Figura 10.1.5 os dois primeiros já foram explicados anteriormente. Os outros serão explicados a seguir.

O gráfico "*Scale-location*"é uma maneira alternativa de visualizar os resíduos em relação aos valores ajustados do modelo de regressão linear. Nesse gráfico, os resíduos são padronizados e depois transformados pela raiz quadrada. Isso essencialmente dobra os resíduos e pode ajudar a encontrar padrões nos resíduos.

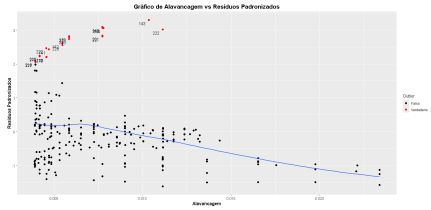
O gráfico de Resíduos vs Alavancagem pode ser usado para identificar possíveis *outliers*. Neste cenário, não há *outliers*.

Já o gráfico apresentado[48] na Figura 10.1.6 serve de comparação com a situação estudada até aqui, já que ele mostra um cenário onde o modelo se ajusta bem ao comportamento da variável dependente.

[48]Bom, esse curso é sobre visualização de dados, e os gráficos unidos desta maneira na Figura 10.1.6, dificulta sua visualização em detalhes. Mas eles foram unidos assim sobretudo por uma questão didática, mas estão em formato vetorial, de modo que para aumentar a visibilidade das legendas e pontos, basta aplicar o zoom. O que contorna a dificuldade inicial de visualizar detalhes.



(a) Um modelo de regressão linear básico ajustado para uma precisão próxima de 90%, para comparação com a Figura 10.1.3.



(b) Um gráfico de alavancagem para evidenciar os *outliers* em um cenário em que realmente há *outliers*, para comparação Figura 10.1.5.

Figura 10.1.6: as Figuras (a) e (b) devem ser empregadas como comparação em contraste com as Figuras 10.1.3 e 10.1.5.

10.1.3 Regressão Linear Múltipla

Um modelo de regressão linear múltipla é uma generalização da SLR discutido na Seção 10.1. Possui k variáveis com a forma:

$$\hat{y} = a_0 + a_1 x_1 + a_2 x_2 + ...a_k x_k, \tag{8}$$

onde os valores de x_i são as entradas do sistema, os coeficientes a_i são os parâmetros do modelo calculados a partir dos dados medidos, e $\hat{y}$ é um valor previsto pelo modelo. De modo geral todas as análise feitas na Seção 10.1, se aplicam neste modelo mais genérico. Da mesma forma para desenvolver este tipo de modelo de regressão linear múltipla (MLR), devemos selecionar cuidadosamente os preditores específicos para cada modelo.

Antes de iniciar o desenvolvimento do modelo, é útil ter uma noção visual do os relacionamentos dentro dos dados. Podemos fazer isso facilmente com o seguinte chamada de função[49] presente na Listagem 10.2:

```
1       >pairs(int00.dat, gap=0.5)
2
```

Listagem 10.2: comando para chamar a função que mostra um panorama geral das correlações entre as métricas do int2000

[49]Desta vez preferimos não usar o pacote `ggplot` pois para essa quantidade de gráficos ficar intelegível, é necessário uma visualização meno poluída possível. Use o zoom para maiores detalhes.

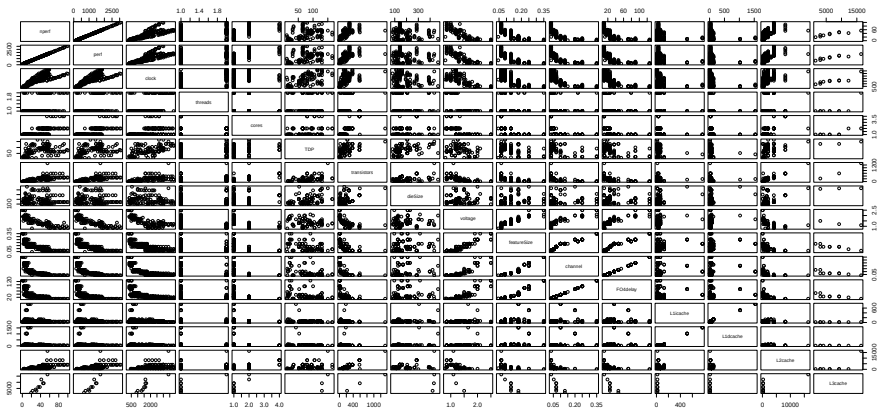


Figura 10.1.7: todas as comparações das métricas aos pares para o *data frame* Int2000.

A função apresentada na Listagem 10.2 produz o gráfico mostrado na Figura 10.1.7 .Este técnica fornece uma comparação pareada de todos os dados do *data frame* `int00.dat`.

Como exemplo para entender como ler este gráfico, localize o terceiro frame da primeierfra linha: é o gráfico de dispersão $clock \times nperf$. Já o terceiro bloco da segunda linha é o gráfico de dispersão $clock \times perf$.

Observe que existe uma aparente correlação linear perfeita entre perf e nperf. Esse relacionamento ocorre nõ à toa, nperf é um simples redimensionamento de perf[50].

[50]Como exercício para o bangalô: tente encontrar o $L2cache \times Transistors$ e explicar.

10.1.4 Identificando Bons Preditores

O segundo passo no desenvolvimento do modelo MLR, após entender o panorama geral das correlação par-a-par das métricas é identificar os possíveis preditores que poderíamos incluir no modelo. Para refinar o primeiro modelo. [51]. Vamos tentar manter as coisas simples. Assim, devemos usar o menor número de preditores necessários para fornecer boas previsões. Além disso, usar muitos preditores ou preditores redundantes incorporam o ruído aleatório nos dados no modelo. Preditores redundantes ou desnecessários também podem levar a instabilidades numéricas no cálculo dos coeficientes.

[51]Resista a ideia de incluir todas as variáveis disponíveis nos dados como preditores, por achar ser melhor muita informação que não ter informação suficiente.

De modo que devemos encontrar um equilíbrio entre incluir poucos e muitos preditores. Um modelo com poucos preditores pode produzir previsões tendenciosas. Por outro lado, adicionar mais preditores ao modelo sempre fará com que o Valor R^2 aumente[52]. Acompanhe na Figura 10.1.8 como se comporta o R^2 em modelos bem ajustados e fracamente ajustados.

[52]O coeficiente de determinação R^2 é uma métrica em análise de regressão, que fornece uma medida quantitativa da qualidade do ajuste do modelo aos dados observados. É a razão entre a variabilidade explicada pelo modelo e a variabilidade total (varia de 0 a 1). De modo que quanto mais próximo de 1, mais melhor este se ajusta aos dados reais.

O R^2 ajustado tenta compensar o comportamento do R^2 regular alterando seu valor de acordo com o número de parâmetros estimados (geralmente o nú-

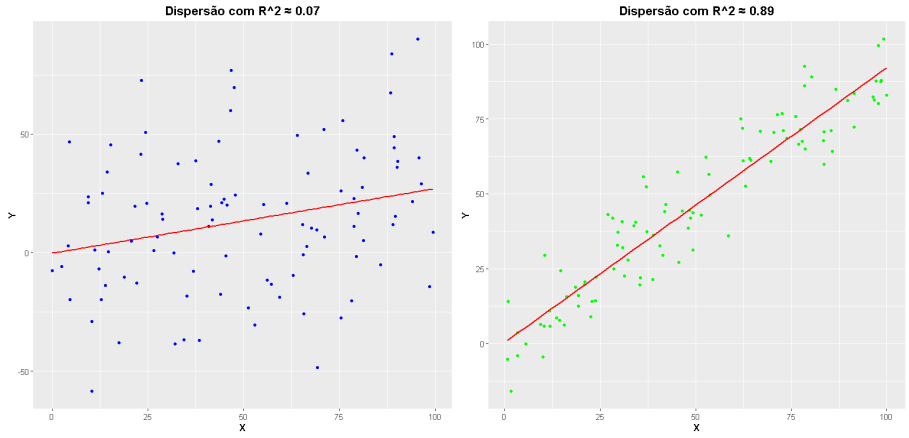


Figura 10.1.8: dois gráficos comparando um modelo que se ajusta bem, O coeficiente de determinação R^2 e outro com R^2 próximo de 0.

mero de preditores incluindo o interceptor). Esse ajuste nos ajuda a determinar se a adição de um preditor melhora o ajuste do modelo ou se é simplesmente uma melhor modelagem do ruído

Diferença entre R^2 Ajustado e R^2 Regular

O R^2 regular e o R^2 ajustado são métricas usadas para avaliar a qualidade de um modelo de regressão, mas têm diferenças importantes[53].

$$R^2 = 1 - \frac{SSR}{SST} \tag{9}$$

Limitação do R^2 Regular: sempre aumenta ou, no mínimo, permanece o mesmo quando novos preditores são adicionados ao modelo, independentemente de esses preditores realmente melhorarem o modelo. Isso pode levar à inclusão de preditores irrelevantes que apenas modelam o ruído.

$$R^2_{ajustado} = 1 - \frac{\frac{SSR}{n-p-1}}{\frac{SST}{n-1}} \tag{10}$$

onde n é o número de observações e p é o número de preditores[54] no modelo.

Benefício: o R^2 ajustado oferece uma medida mais realista do quão bem o modelo se ajusta aos dados, especialmente quando há muitos preditores. Ajuda a determinar se a adição de um novo preditor realmente melhora o modelo ou se apenas está capturando ruído.

Assim o R^2 ajustado é uma métrica mais confiável quando se lida com modelos de regressão com muitos preditores, pois ele ajusta para o número de

[53] O R^2 regular mede a proporção da variabilidade total da variável dependente que é explicada pelas variáveis independentes no modelo.
O R^2 ajustado tenta corrigir a tendência do R^2 regular de aumentar com a adição de preditores, ajustando seu valor de acordo com o número de preditores no modelo. Ele leva em consideração o número de preditores e o número de observações, penalizando a inclusão de preditores irrelevantes.

[54] Exemplo:

- **No modelo com Poucos Preditores**: R^2 regular e R^2 ajustado podem ser semelhantes, pois há poucos preditores e ambos medem quão bem o modelo explica a variabilidade.
- **No modelo com Muitos Preditores**: Adicionar características irrelevantes (como a cor dos cabos ou a potência das fontes de alimentação) pode aumentar o R^2 regular porque ele sempre aumenta com mais preditores. O R^2 ajustado, no entanto, pode diminuir ou aumentar muito pouco se esses novos preditores não melhorarem real-

preditores e evita a falsa impressão de melhora no modelo ao capturar ruído. Lembre-se de que o objetivo é usar o menor número possível de preditores, enquanto ainda produza um modelo que explica bem os dados.

Um bom começo é exibir os nomes de todas as colunas e tentar obter informações sobre seu conteúdo e sua influência no comportamento da variável independente. Ver Tabela 9.

Tabela 9: colunas do *Data Frame* original.

Nomes das Colunas	
nperf	perf
clock	threads
cores	TDP
transistors	dieSize
voltage	featureSize
channel	FO4delay
L1icache	L1dcache
L2cache	L3cache

Processo de Eliminação de Preditores:

Quase todos os possíveis preditores restantes parecem potencialmente úteis para o nosso modelo, então os mantemos disponíveis como preditores potenciais por enquanto[55].

Além de excluir algumas variáveis aparentemente inúteis (como TDP - ver nota 55). Também devemos considerar, por exemplo, que os termos individuais podem ser não lineares, como $a_i {x_i}^p$, onde p não precisa ser igual a um. Essa flexibilidade nos permite incluir potências adicionais das variáveis individuais. Devemos incluir esses termos não lineares, porém, apenas se tivermos alguma razão física forte para suspeitar que a saída possa ser uma função não linear de uma determinada entrada.

Alguns estudos [20] sugerem que as taxas de falhas de *cache* (*cache miss*) são aproximadamente proporcionais à raiz quadrada do tamanho do cache. Consequentemente, incluiremos termos para a raiz quadrada, então ($p = 1/2$) de cada tamanho de *cache* como possíveis preditores. Também devemos incluir termos de primeiro grau ($p = 1$) de cada tamanho de *cache* como possíveis preditores[56].

Notamos ainda que apenas algumas das entradas no *data frame* `int00.dat` incluem valores para o cache L3 (usando `summary(int00.dat)`), então decidimos excluir o tamanho do cache L3 como um preditor potencial. Explorar

[55] Deixo para vocês acompanhar os autores [19] nessa tarefa, vamos falar um pouco em sala desta análise, mas não vamos nos estender aqui.

[56] Ao usar ambos os preditores, linear e não linear, exploraremos tanto as relações lineares quanto não lineares entre o tamanho do cache e o desempenho do sistema. Isso permite que o modelo de regressão capture de forma mais completa as complexidades da relação entre os preditores e a variável dependente, resultando em um modelo mais robusto e preciso.

esse tipo de conhecimento específico do domínio ao selecionar preditores pode ajudar a produzir modelos melhores do que aplicar cegamente o processo de desenvolvimento de modelos.

Mais ainda, se mantivermos L3 como um preditor, apenas observações onde esse valor não está ausente serão incluídas no modelo, reduzindo o tamanho do conjunto de dados de 256 para 10 sistemas[57].

[57] Verifique os `NAs` no resultado do comando `summary(int00.dat)`

A lista final de preditores **potenciais** que disponibilizaremos para o processo de desenvolvimento do modelo é mostrada na Tabela 10.

Tabela 10: Colunas do *Data Frame* que escolhemos.

Nomes das Colunas	
clock	threads
cores	Transistors
dieSize	$\sqrt{L1icache}$
voltage	featureSize
channel	FO4delay
L1icache	L1dcache
L2cache	$\sqrt{L1dcache}$
$\sqrt{L2cache}$	

Eliminação Retroativa:

Usaremos um processo chamado eliminação retroativa (*backward elimination*) [21] para ajudar a decidir quais preditores manter e quais excluir. Na eliminação retroativa, depois dos passos já trilhados, usamos a função `summary()` para encontrar o *p-value* de cada preditor. Para preditores cuja contribuição ou inclinação é próxima de zero (com base na estimativa e no erro padrão), consideraremos a remoção do termo. Um valor p grande significa que a chance de observar o t-estatístico (razão entre a estimativa e o erro padrão), assumindo que a inclinação ou estimativa é zero, é bastante provável com base no acaso, indicando que o parâmetro não está contribuindo para o ajuste do modelo.

Se esse valor for maior que nosso limite predeterminado, removemos esse preditor do modelo e ajustamos outro modelo excluindo esse parâmetro. Um limite típico para manter preditores em um modelo é $p = 0,05$, o que significa que há pelo menos 95% de chance de que o preditor seja significativo. Um limite de $p = 0,10$ também não é incomum. Embora um limite específico possa ser usado para determinar se um valor p é "grande", pode haver razões para manter um termo no modelo se um valor p for apenas ligeiramente maior que os critérios. Repetimos esse processo até que os valores p de todos os preditores restantes no modelo estejam abaixo do nosso limite.

Observe que a eliminação retroativa não é a única abordagem para o desenvolvimento de modelos de regressão. Uma abordagem complementar é a seleção direta. Nesta abordagem, adicionamos sucessivamente preditores potenciais ao modelo de regressão, desde que seus valores de p no modelo computado permaneçam abaixo do

Se esse valor for maior do que nosso limite predeterminado, removemos esse preditor do modelo excluimos esse parâmetro. Um limite típico para manter preditores em um modelo é $p = 0,05$, significando que há pelo menos 95% de chance de que o preditor seja significativo. Um limite de $p = 0,10$ também não é incomum. Embora um limite específico possa ser usado para determinar se um valor p é "grande", pode haver razões para manter um termo no modelo se o valor p for apenas ligeiramente maior que o critério.

Note que a eliminação retroativa não é única abordagem possível. Uma abordagem complementar é a seleção adiante (*forward selection*). Nessa abordagem, adicionamos sucessivamente preditores potenciais ao modelo de regressão enquanto seus valores p permanecerem abaixo do limite. Esse processo segue até que todos os preditores tenham sido testados[58]

[58]Naturalmente há diferentes abordagens: como regressão passo a passo, seleção automatizada e outras. Pesquise!

Todas essas abordagens têm suas vantagens e desvantagens, seus apoiadores e detratores. Eu prefiro o processo de eliminação para trás porque geralmente é fácil determinar qual parâmetro devemos remover em cada etapa do processo. Determinar qual parâmetro tentar em cada etapa é mais difícil com a seleção para frente. A eliminação para trás tem uma vantagem adicional, na medida em que vários parâmetros juntos podem ter melhor poder preditivo do que qualquer subconjunto desses parâmetros. Como resultado, o processo de eliminação para trás é mais provável de incluir esses parâmetros como um grupo no modelo final do que o processo de seleção para frente.

Backward Elimination - Mão na Massa:

A chamada de função na Listagem 10.3 atribui o resultado do objeto de modelo linear à variável `int00.lm.full` onde `.full` é adicionado para indicar que o modelo inclui todos os preditores possíveis.

```
1       int00.lm.full <- lm(nperf ~ clock + threads + cores + transistors
2       + dieSize + voltage + featureSize + channel + FO4delay
3       + L1icache + sqrt(L1icache) + L1dcache + sqrt(L1dcache)
4       + L2cache + sqrt(L2cache), data=int00.dat)
5
```

Listagem 10.3: atribui o resultado do objeto de modelo linear à variável `int00.lm.full`.

O comando `summary(int00.lm.full)` nos fornece informações detalhadas sobre este novo modelo com mais preditores, apresentadas na Listagem 10.4 sobre o modelo linear que acabamos de criar:

```
1       > summary(int00.lm.full)
2
```

```
3       Call:
4       lm(formula = nperf ~ clock + threads + cores + transistors +
5       dieSize + voltage + featureSize + channel + FO4delay + L1icache +
6       sqrt(L1icache) + L1dcache + sqrt(L1dcache) + L2cache + sqrt(
    L2cache),
7       data = int00.dat)
8
9       Residuals:
10      Min      1Q  Median      3Q     Max
11      -10.804  -2.702   0.000   2.285   9.809
12
13      Coefficients:
14      Estimate Std. Error t value Pr(>|t|)
15      (Intercept)    -2.108e+01  7.852e+01  -0.268  0.78927
16      clock           2.605e-02  1.671e-03  15.594  < 2e-16 ***
17      threads        -2.346e+00  2.089e+00  -1.123  0.26596
18      cores           2.246e+00  1.782e+00   1.260  0.21235
19      transistors    -5.580e-03  1.388e-02  -0.402  0.68897
20      dieSize         1.021e-02  1.746e-02   0.585  0.56084
21      voltage        -2.623e+01  7.698e+00  -3.408  0.00117 **
22      featureSize     3.101e+01  1.122e+02   0.276  0.78324
23      channel         9.496e+01  5.945e+02   0.160  0.87361
24      FO4delay       -1.765e-02  1.600e+00  -0.011  0.99123
25      L1icache        1.102e+02  4.206e+01   2.619  0.01111 *
26      sqrt(L1icache) -7.390e+02  2.980e+02  -2.480  0.01593 *
27      L1dcache       -1.114e+02  4.019e+01  -2.771  0.00739 **
28      sqrt(L1dcache)  7.492e+02  2.739e+02   2.735  0.00815 **
29      L2cache        -9.684e-03  1.745e-03  -5.550 6.57e-07 ***
30      sqrt(L2cache)   1.221e+00  2.425e-01   5.034 4.54e-06 ***
31      ---
32      Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘’. 0.1 ‘ ’ 1
33
34      Residual standard error: 4.632 on 61 degrees of freedom
35      (179 observations deleted due to missingness)
36      Multiple R-squared:  0.9652,    Adjusted R-squared:  0.9566
37      F-statistic: 112.8 on 15 and 61 DF,  p-value: < 2.2e-16
38
```

Listagem 10.4: resultado do comando `summary(int00.lm.full)`.

Ao analisar os resíduos do modelo, observa-se que eles estão equilibrados em torno de uma mediana de valor zero, o que é desejável. No entanto, nota-se que 179 observações foram excluídas devido a valores ausentes (indica que, das linhas no *Data Frame* original int00.dat, 179 linhas foram removidas por que eram NA). Esses valores ausentes levaram o R a remover automaticamente essas linhas ao calcular o modelo linear. A desvantagem dessa remoção automática é que não sabemos de que variáveis os dados estão faltando, nem se há uma razão sistemática para a ausência desses valores. Isso também torna os valores de R^2-ajustado não comparáveis entre os modelos, pois pressupõe-se que as mesmas observações são usadas em ambos os modelos comparados.

O número total de observações no modelo é igual ao número de graus de liberdade[59] restantes (61 neste caso)[60].

Os valores de R^2 e R^2-ajustado são relativamente próximos de um, indicando que o modelo explica bem os valores de desempenho (*nperf*) ou esses

[59] Importante uma pausa aqui para rever a definição de Graus de liberdade. A mais simples que eu encontrei é a que diz que graus de liberdade podem ser entendidos pelo tamanho da amostra n menos o números de estimadores (variáveis ou preditores) mais o intercepto β, $gl = n - p - 1$

[60] Bom, e de onde surgiu o **61**? Dado $n - p$, onde n é o número total de observações e p. No nosso caso, o número total de observações (256)após a remoção das 179 observações ausentes, sobraram 77. O modelo tem 15 preditores mais o

altos valores de R^2-ajustado (ver nota 54) podem apenas indicar que o modelo está bom em modelar o ruído nas medições (*oufit*).

Para continuar desenvolvendo o modelo, aplicamos o procedimento de eliminação retroativa identificando do preditor com o maior valor p, que é *FO4delay* com um valor p de 0.99123, excedendo o limite de $p = 0.05$. Podemos usar a função `update()` para eliminar um preditor específico e recomputar o modelo em um único passo, indicando que o preditor FO4delay deve ser removido e o novo modelo será chamado de `int00.lm.2`.

Podemos usar a função `update()` para eliminar um determinado preditor e recalcular o modelo em uma única etapa. A notação " . . " significa que a função `update()` deve manter os lados esquerdo e direito do modelo iguais. Ao incluir " - *FO4delay* ", instamos a função a remover esse preditor do modelo. O *data frame* int00.lm.2 será nosso segundo modelo, passos a seguir na Listagem 10.5.

```
1   >  int00.lm.2 <- update(int00.lm.full, .~. - FO4delay, data =
2   + int00.dat)
3   > summary(int00.lm.2)
4
5   Call:
6   lm(formula = nperf ~ clock + threads + cores + transistors +
7   dieSize + voltage + featureSize + channel + L1icache + sqrt(
    L1icache) +
8   L1dcache + sqrt(L1dcache) + L2cache + sqrt(L2cache), data = int00
    .dat)
9
10  Residuals:
11  Min      1Q  Median      3Q     Max
12  -10.795  -2.714   0.000   2.283   9.809
13
14  Coefficients:
15  Estimate Std. Error t value Pr(>|t|)
16  (Intercept)    -2.088e+01  7.584e+01  -0.275 0.783983
17  clock           2.604e-02  1.563e-03  16.662  < 2e-16 ***
18  threads        -2.345e+00  2.070e+00  -1.133 0.261641
19  cores           2.248e+00  1.759e+00   1.278 0.206080
20  transistors    -5.556e-03  1.359e-02  -0.409 0.684020
21  dieSize         1.013e-02  1.571e-02   0.645 0.521488
22  voltage        -2.626e+01  7.302e+00  -3.596 0.000642 ***
23  featureSize     3.104e+01  1.113e+02   0.279 0.781232
24  channel         8.855e+01  1.218e+02   0.727 0.469815
25  L1icache        1.103e+02  4.041e+01   2.729 0.008257 **
26  sqrt(L1icache) -7.398e+02  2.866e+02  -2.581 0.012230 *
27  L1dcache       -1.115e+02  3.859e+01  -2.889 0.005311 **
28  sqrt(L1dcache)  7.500e+02  2.632e+02   2.849 0.005937 **
29  L2cache        -9.693e-03  1.494e-03  -6.488 1.64e-08 ***
30  sqrt(L2cache)   1.222e+00  1.975e-01   6.189 5.33e-08 ***
31  ---
32  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '. 0.1 ' ' 1
33
34  Residual standard error: 4.594 on 62 degrees of freedom
35  (179 observations deleted due to missingness)
36  Multiple R-squared:  0.9652,    Adjusted R-squared:  0.9573
37  F-statistic: 122.8 on 14 and 62 DF,  p-value: < 2.2e-16
```

38

Listagem 10.5: resultado do comando `summary(int00.lm.2)`, após a remoção do preditor *FO4delay*.

Bom, a ideia por trás deste processo é seguir removendo o próximo potencial preditor com o maior valor p, cque tenha excedido nosso limite predeterminado, neste caso $featureSize = 0.781232$. Se continuarmos aplicando a técnica de eliminação para o valores p maiores que 0.05, finalmente chegaremos ao seguinte modelo para o nperf estimado, mostrado na equação[61] 11.

[61] Lembrando que os preditores *L3cache* e *TDP*, foram removidos antes até da primeira eliminação, mesmo sem avaliação do *p-value*.

$$\begin{aligned}\widehat{nperf} &= -58.22 + 0.02482 \cdot clock + 2.397 \cdot cores - 23.58 \cdot voltage \\ &= +139.9 \cdot channel + 87.03 \cdot L1icache - 576.8 \cdot \sqrt{L1icache} \\ &= -89.03 \cdot L1dcache + 598 \cdot \sqrt{L1dcache} - 0.008621 \cdot L2cache \\ &= +1.085 \cdot \sqrt{L2cache}\end{aligned} \tag{11}$$

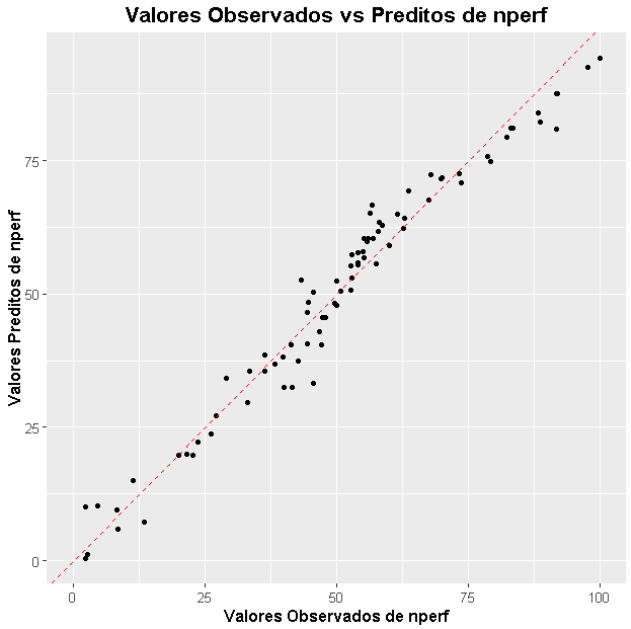


Figura 10.1.9: valores *nperf* medido *versus* *nperf* estimado com os novos preditores na escala 0 a 100 adimensional.

Analisando a Figura 10.1.9, vemos a linha vermelha tracejada, que representa a linha de identidade, onde os valores observados seriam iguais aos valores preditos ($y = x$). Os pontos bem próximos a essa linha indicam um bom ajuste do modelo.

O intercepto $\beta = -58.22$ não é diretamente observável no gráfico dos valores observados *vs* preditos. A importância do intercepto é mais relevante no contexto da fórmula do modelo, mas seu valor não se reflete diretamente na escala do gráfico, que é baseada nos valores observados e preditos.

Quanto aos valores extremos e *outliers*, há visualmente poucos, uma análise mais profunda dos resíduos pode nos subsidiar com maiores detalhes[62].

[62]Bom, parece que tudo deu certo, mas nem sempre é assim...Veremos adiante

Últimas análises sobre a MLR:

Se os caríssimos fizeram o trabalho de casa, devem ter observado que o número de graus de liberdade em cada modelo subsequente aumenta à medida que os preditores são excluídos (ver notas 60 e 59), como esperado. Em alguns casos, o número de graus de liberdade aumenta em mais de um quando apenas um único preditor é eliminado do modelo e aumenta em mais de um porque aumentam o número de observações descartadas (`NA`).

Observe também que, à medida que os preditores diminuem do modelo, os valores de R^2 ficam muito próximos de 0,965. No entanto, o valor de R^2ajustado tende a aumentar ligeiramente com cada preditor descartado. Este aumento, embora pequeno, indica que o modelo com menos preditores e mais graus de liberdade tende a explicar os dados um pouco melhor do que o modelo anterior, que tinha mais um preditor. É possível que essas mudanças sejam simplesmente devidas a flutuações aleatórias. No entanto, é bom ver um comportando esperado. teste F (*F-test*) compara o modelo atual a um modelo com um preditor a menos. Se o modelo atual for melhor que o modelo reduzido, o *valor p* será decrescente. No nosso caso com os resultados e dada a ordem de grandeza (10^{-16}), este teste não nos ajuda particularmente a discriminar claramente a melhora entre modelos.

Análise dos Resíduos do Modelo Final:

Além da análise visual presente na Figura 10.1.9, que claramente mostra uma correlação positiva forte, precisamos aplicar um pouco mais de rigor que a apenas uma inspeção visual. Os dados que obteremos aqui servirão para comparar com os dados presentes na análise da Figura 10.1.5.

A análise dos resíduos da última MLR[63] mostrada na Equação 11 produz o painel de gráficos mostrado na Figura 10.1.10. No gráfico Resíduos *vs* Valores Ajustados, vemos que os resíduos parecem estar uniformemente espalhados em torno de zero. Não há quaisquer padrões óbvios que nos levem a pensar que os resíduos não de mostram bem comportados. Consequentemente, este gráfico não nos dá razão para acreditar que produzimos um modelo pobre.

[63]Novamente, essa análise deve ser necessariamente comparada com análise similar feita sobre a Figura 10.1.5

No gráfico Q-Q da Figura 4.2, vemos que os resíduos seguem aproximadamente a linha indicada. Neste gráfico, podemos ver um pouco mais de padrão e algumas não linearidades óbvias, o que nos leva a ser um pouco mais cautelosos ao concluir que os resíduos são normalmente distribuídos. Mas um ou

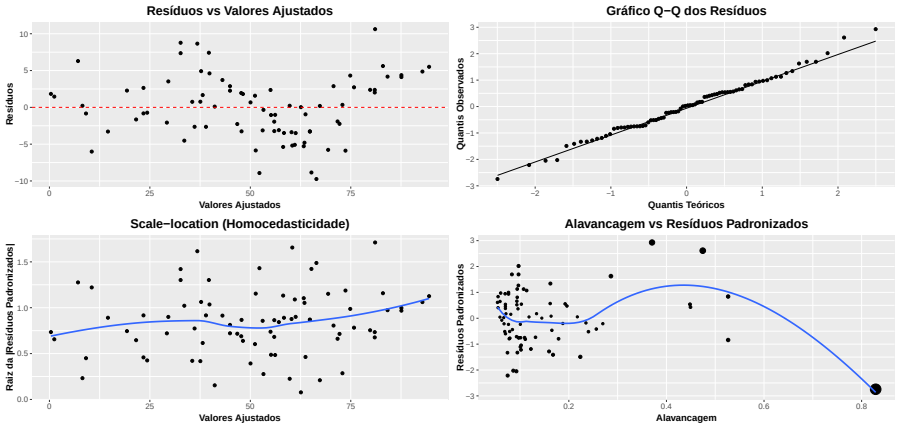


Figura 10.1.10: gráficos de análise mais profunda dos resíduos da MLR.

dois pontos que se desviem apenas ligeiramente dos valores esperados, nada excessivamente preocupante. Assim, não devemos rejeitar o modelo baseado neste teste, mas os resultados devem servir como um lembrete de que todos os modelos são imperfeitos.

10.1.5 Quando nem Tudo Vai Bem

Como já chamei atenção na nota 62, Às vezes, quando tentamos desenvolver um modelo utilizando o processo de eliminação retroativa, obtemos resultados que não parecem fazer sentido. Por exemplo, vamos tentar desenvolver um modelo de regressão linear múltipla para os dados `Int1992` usando este processo. Como antes, começamos por incluir todos os potenciais preditores da Tabela 10, veja os comandos presente na Listagem 10.6.

```
1       int92.lm.full <- lm(nperf ~ clock + threads + cores +
2       transistors + dieSize + voltage + featureSize + channel +
3       FO4delay + L1icache + sqrt(L1icache) + L1dcache +
4       sqrt(L1dcache) + L2cache + sqrt(L2cache), data=int92.dat)
5       >  summary(int92.lm.full)
6
7       Call:
8       lm(formula = nperf ~ clock + threads + cores + transistors +
9       dieSize + voltage + featureSize + channel + FO4delay + L1icache +
10      sqrt(L1icache) + L1dcache + sqrt(L1dcache) + L2cache + sqrt(
    L2cache),
11      data = int92.dat)
12
13      Residuals:
14      14       15       16       17       18       19
15      0.4096   1.3957 -2.3612   0.1498 -1.5513   1.9575
16
17      Coefficients: (14 not defined because of singularities)
18      Estimate Std. Error t value Pr(>|t|)
19      (Intercept)    -25.93278    6.56141  -3.952   0.0168 *
```

```
clock             0.35422    0.02184  16.215 8.46e-05 ***
threads                NA         NA      NA       NA
cores                  NA         NA      NA       NA
transistors            NA         NA      NA       NA
dieSize                NA         NA      NA       NA
voltage                NA         NA      NA       NA
featureSize            NA         NA      NA       NA
channel                NA         NA      NA       NA
FO4delay               NA         NA      NA       NA
L1icache               NA         NA      NA       NA
sqrt(L1icache)         NA         NA      NA       NA
L1dcache               NA         NA      NA       NA
sqrt(L1dcache)         NA         NA      NA       NA
L2cache                NA         NA      NA       NA
sqrt(L2cache)          NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘’. 0.1 ‘ ’ 1

Residual standard error: 1.868 on 4 degrees of freedom
(72 observations deleted due to missingness)
Multiple R-squared:  0.985,     Adjusted R-squared:  0.9813
F-statistic: 262.9 on 1 and 4 DF,  p-value: 8.463e-05
```

Listagem 10.6: criando um modelo de regressão dos dados presentes em int92.dat, e mostrando o seu sumário.

Observe a quantidade de itens ausentes (NA) no *data frame* para cada entrada.Vê-se uma linha contendo uma mesnsagem que alerta que quatorze coeficientes “não foram definidos devido a singularidades. ”[64] Assim o R não pode calcular um valor para esses coeficientes devido a algumas anomalias nos dados. (não foi possível inverter a matriz usada[65] na minimização de mínimos quadrados processo.)

O primeiro passo para resolver este problema é notar que 72 observações estão ausentes, deixando apenas quatro graus de liberdade. Aqui vamos seguir um caminho diferente dos autores em [19]. Vamos usar a função summary(int92.dat) para dar uma panorâmica no *data frame*.

```
> summary(int92.dat)
nperf              perf             clock           threads
Min.   :  0.00   Min.   : 36.70   Min.   : 48.00   Min.   :1
1st Qu.:  9.67   1st Qu.: 68.62   1st Qu.: 77.75   1st Qu.:1
Median : 20.02   Median :102.81   Median :105.50   Median :1
Mean   : 26.53   Mean   :124.29   Mean   :134.92   Mean   :1
3rd Qu.: 31.93   3rd Qu.:142.11   3rd Qu.:175.00   3rd Qu.:1
Max.   :100.00   Max.   :366.86   Max.   :350.00   Max.   :1

TDP              transistors        dieSize          voltage
Min.   : 3.00   Min.   : 0.580   Min.   : 76.0   Min.   :2.285
1st Qu.:13.25   1st Qu.: 1.680   1st Qu.:164.0   1st Qu.:3.300
Median :29.00   Median : 2.300   Median :196.0   Median :3.300
Mean   :24.95   Mean   : 3.248   Mean   :209.5   Mean   :3.492
3rd Qu.:33.00   3rd Qu.: 3.125   3rd Qu.:256.0   3rd Qu.:3.300
Max.   :56.00   Max.   :15.000   Max.   :335.0   Max.   :5.000
```

[64] O Problema da Singularidade: uma matriz é considerada singular se ela não possui uma inversa. No nosso caso, : o número de observações é menor que o número de preditores, então a matriz $X^T X$ não pode ser invertida. X é a matriz de preditores.

[65] A fórmula para encontrar os preditores pode ser expressa também assim $y = X\beta + \epsilon$, ao invés da forma polinomial vista na Equação 8. Portanto é daí que surge o erro.

```
      NA's   :36      NA's   :26      NA's   :23      NA's   :24
      featureSize        channel          FO4delay      L1icache
      Min.   :0.2900   Min.   :0.2500   Min.   : 90   Min.   :  1.00
      1st Qu.:0.5000   1st Qu.:0.5000   1st Qu.:180   1st Qu.:  8.00
      Median :0.6000   Median :0.6000   Median :216   Median :  8.00
      Mean   :0.5995   Mean   :0.5899   Mean   :214   Mean   : 13.87
      3rd Qu.:0.7500   3rd Qu.:0.7500   3rd Qu.:270   3rd Qu.: 16.00
      Max.   :1.0000   Max.   :1.0000   Max.   :360   Max.   :128.00
      NA's   :3        NA's   :3        NA's   :3     NA's   :15
      cores              L1dcache         L2cache         L3cache
      Min.   :1        Min.   :  0.00   Min.   : 96.0   Min.   : NA
      1st Qu.:1        1st Qu.:  8.00   1st Qu.: 96.0   1st Qu.: NA
      Median :1        Median :  8.00   Median : 96.0   Median : NA
      Mean   :1        Mean   : 18.95   Mean   :211.2   Mean   :NaN
      3rd Qu.:1        3rd Qu.: 16.00   3rd Qu.:256.0   3rd Qu.: NA
      Max.   :1        Max.   :256.00   Max.   :512.0   Max.   : NA
      NA's   :21       NA's   :68       NA's   :78
```

Listagem 10.7: sumário do *dataframe* `int92.dat`

Com muita cautela, podemos inferir que há 78 linhas no *data frame* (mas é bom conferir com `nrow()`. Temos!), de posse desta informação, vamos buscar anomalias que possam estar causando o problema das singularidades:

1. Nas colunas *cores* e *threads*, todos os dados tem valor 1.
2. A coluna *L3cache* está com seus dados ausentes.
3. A coluna *L2cache* possui 68 NAs, com apensa 10 valores, é impossível conseguir 14 preditores.

Então podemos descartar esses parâmetros para o nosso modelo, mas como int92.lm.full já havia partido do pressuposto que L3cache não fazia parte dos preditores desde o início. Nosso *data frame* reduzido (`int92.lm.reduced.2`) será o visto na Listagem 10.8:

```
      > int92.lm.full <- lm(nperf ~ clock + threads + cores +
      + transistors + dieSize + voltage + featureSize + channel +
      + FO4delay + L1icache + sqrt(L1icache) + L1dcache +
      + sqrt(L1dcache) + L2cache + sqrt(L2cache), data=int92.dat)

      > int92.lm.reduced.2 <- update(int92.lm.full, .~. - threads -
    cores - L2cache - sqrt(L2cache) - L3cache - sqrt(L3cache), data=
    int92.dat)
```

Listagem 10.8: criaçao do primeiro modelo, seguido do reduzido, a partir do *dataframe* `int92.dat`.

Acompanhe na Listagem 10.9, o sumário do último modelo[66].

[66] Para o bangalô fica a análise gráfica.

```
      > summary(int92.lm.reduced.2)

      Call:
```

```
4       lm(formula = nperf ~ clock + transistors + dieSize + voltage +
5       featureSize + channel + FO4delay + L1icache + sqrt(L1icache) +
6       L1dcache + sqrt(L1dcache), data = int92.dat)
7
8       Residuals:
9       Min      1Q  Median      3Q     Max
10      -7.3233 -1.1756  0.2151  1.0157  8.0634
11
12      Coefficients:
13      Estimate Std. Error t value Pr(>|t|)
14      (Intercept)     -58.51730   17.70879  -3.304  0.00278 **
15      clock             0.23444    0.01792  13.084 6.03e-13 ***
16      transistors      -0.32032    1.13593  -0.282  0.78018
17      dieSize           0.25550    0.04800   5.323 1.44e-05 ***
18      voltage           1.66368    1.61147   1.032  0.31139
19      featureSize     377.84287   69.85249   5.409 1.15e-05 ***
20      channel        -493.84797   88.12198  -5.604 6.88e-06 ***
21      FO4delay          0.14082    0.08581   1.641  0.11283
22      L1icache          4.21569    1.74565   2.415  0.02307 *
23      sqrt(L1icache)  -12.33773    7.76656  -1.589  0.12425
24      L1dcache         -5.53450    2.10354  -2.631  0.01412 *
25      sqrt(L1dcache)   23.89764    7.98986   2.991  0.00602 **
26      ---
27      Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘’. 0.1 ‘ ’ 1
28
29      Residual standard error: 3.68 on 26 degrees of freedom
30      (40 observations deleted due to missingness)
31      Multiple R-squared:  0.985,     Adjusted R-squared:  0.9786
32      F-statistic:   155 on 11 and 26 DF,  p-value: < 2.2e-16
33
```

Listagem 10.9: resultado da função `summary()` aplicada ao modelo `int92.lm.reduced.2`.

10.2 FAZENDO ESTIMATIVAS

Predição é normalmente o objetivo principal para a construção de modelos de regressão. O desenvolvedor do modelo deseja usar o modelo para estimar ou prever a resposta do sistema se ele fosse operado com valores de entrada que nunca estiveram realmente disponíveis em nenhum dos sistemas medidos. Por exemplo, podemos querer usar o modelo que desenvolvemos usando o conjunto de dados Int2000 para prever o desempenho de um novo processador com uma frequência de *clock*, um tamanho de *cache* ou alguma outra combinação de parâmetros que não existe no conjunto de dados. Ao inserir esta nova combinação de valores de parâmetros no modelo, podemos calcular o desempenho esperado do novo processador.

10.2.1 Segregação de Dados Para Testes e Treinamento.

Previamente dados disponíveis no *data frame* `int00.dat` para selecionar os preditores apropriados para incluir no modelo de regressão final. Como calculamos o modelo para se ajustar a este conjunto de dados específico,se empregarmos

este mesmo conjunto de dados para testar as capacidades preditivas do mesmo processador, obteriamos um resultado enviesado (ver Figura 10.1.9). Em vez disso, devemos usar um conjunto de dados para treinar o modelo e outro conjunto de dados para testá-lo.

A dificuldade deste processo de teste de trem é que precisamos de conjuntos de dados semelhantes. Uma maneira padrão de encontrar esses dois conjuntos de dados diferentes é para dividir os dados disponíveis em duas partes. Pegamos uma porção aleatória de tudo os dados disponíveis e chame-os de nosso conjunto de treinamento. Usamos então esta parte do os dados na função lm() para calcular os valores específicos do modelo coeficientes. Usamos a parte restante dos dados como nosso conjunto de testes para veja quão bem o modelo prevê os resultados, em comparação com os dados deste teste. Para fins de demonstração, uma semente de número aleatório será definida para que os resultados serão reproduzíveis sempre. Na prática, você provavelmente irá deseja resultados aleatórios verdadeiros e não deseja definir uma semente. Uma semente pode ser qualquer valor inteiro.

A dificuldade deste processo de teste é encontrar conjuntos de dados semelhantes. Uma maneira prática de encontrar de lidar com este problema é dividir os dados disponíveis em duas partes. Pegamos uma porção aleatóriad os dados disponíveis e use-os como conjunto de treinamento. Usamos então esta parte do os dados na função `lm()` para calcular os valores específicos dos coeficientes. A parte restante dos dados será nosso conjunto de testes para ver quão bem o modelo prevê os resultados, em comparação com os dados deste teste. Uma semente será definida para que a escolha de cada conjunto seja aleatória [67], acompanhe na Listagem 10.10.

```
> set.seed(1234)
> rows <- nrow(int00.dat)
> f <- 0.5
> upper_bound <- floor(f * rows)
> permuted_int00.dat <- int00.dat[sample(rows),]
> train.dat <- permuted_int00.dat[1:upper_bound,]
> test.dat <- permuted_int00.dat[(upper_bound+1):rows,]

```

Listagem 10.10: sequência de operações para seguir divide o conjunto de dados int00.dat em conjuntos de treinamento e teste.

A função `floor()` arredonda o valor do argumento para o número inteiro mais próximo. A linha: `upper_bound <- floor(f * rows)` atribui o número de índice da linha do meio à variável `upper_bound`. A função `sample()` [68] retorna uma permutação dos inteiros entre 1 e n quando lhe damos o valor inteiro n como argumento de entrada. Neste código, a expressão `sample(rows)` retorna um vetor que é uma permutação dos inteiros entre 1 e `rows`, onde `rows` é o número total de linhas no *data frame* `int00.dat`.

[67] Em muitos processos de modelagem e análise de dados, é comum que um conjunto de dados seja dividido aleatoriamente em subconjuntos, como conjuntos de treinamento e de teste. Essa divisão aleatória pode resultar em diferentes subconjuntos a cada execução do código, levando a resultados que podem variar ligeiramente. Para contornar essa variação e permitir que os experimentos sejam reproduzíveis, utilizamos uma semente de número aleatório.

[68] A função `sample(rows)` gera uma permutação aleatória dos índices das linhas do conjunto de dados `int00.dat` usando a semente definida anteriormente, essa permutação será a mesma em todas as execuções do programa, garantindo a reprodutibilidade. A permutação é então utilizada para reordenar as linhas do conjunto de dados, resultando nos respectivos conjuntos.

10.2.2 Treinos e Testes

Na Listagem 10.11 mostramos o cálculo dos coeficientes do modelo de treinamento (`train.dat`) do modelo de regressão. Na mesma Listagem, linha 4, A função `predict()` toma este novo modelo como um de seus argumentos.

```
1  >int00_new.lm <- lm(nperf ~ clock + cores + voltage + channel +
2  L1icache + sqrt(L1icache) + L1dcache + sqrt(L1dcache) +
3  L2cache + sqrt(L2cache), data = train.dat)
4  >predicted.dat <- predict(int00_new.lm, newdata=test.dat)
5  >delta <- predicted.dat - test.dat$nperf
6
```

Listagem 10.11: cálculo dos coeficientes do modelo de treinamento.

Definimos a diferença entre o desempenho previsto e medido para cada processador $\Delta_1 = predito_i - medido_i$. A instrução da Linha 5 calcula o vetor desses valores i e atribui o vetor à variável delta .

A média dessas diferenças para n processadores diferentes é mostrada na equação 12:

$$\overline{\Delta} = \frac{1}{n}\sum_{i=1}^{n}\Delta_i \tag{12}$$

Um intervalo de confiança calculado para esta média nos dará alguma indicação de quão bem o modelo treinado no conjunto de dados `train.dat` previu o desempenho dos processadores no conjunto de dados test.dat. A função `t.test()` calcula um intervalo de confiança para o nível de confiança desejado para i na Listagem 10.12 [69]:

[69] Lembre-se sempre que *p-value* **"grande"**, não rejeitamos a hipótese nula.

```
1  > t.test(delta, conf.level = 0.95)
2
3  One Sample t-test
4
5  data:  delta
6  t = -1.0552, df = 41, p-value = 0.2975
7  alternative hypothesis: true mean is not equal to 0
8  95 percent confidence interval:
9  -3.0825338  0.9668025
10 sample estimates:
11 mean of x
12 -1.057866
13
```

Listagem 10.12: cálculo do intervalo de confiança de 95% para $\overline{\Delta}$.

Se a previsão fosse perfeita, então $\Delta_i = 0$. $Se\Delta_i > 0$, então o modelo previu que o desempenho seria maior do que realmente foi. um $\Delta_i < 0$, por outro lado, significa que o modelo previu que o desempenho foi inferior ao que realmente foi. Consequentemente, se as previsões fossem razoavelmente boas, esperaríamos ver um intervalo de confiança apertado em torno de zero. Neste caso, obtemos um intervalo de confiança de 95 por cento de [-3,08, 0,97]. Dado que nperf é escalonado entre 0 e 100, este é um intervalo de confiança razoavelmente restrito que inclui zero. Assim, concluímos que o modelo é razoavelmente bom em prever valores no conjunto de dados `test.dat` quando treinado no conjunto de dados `train.dat`.

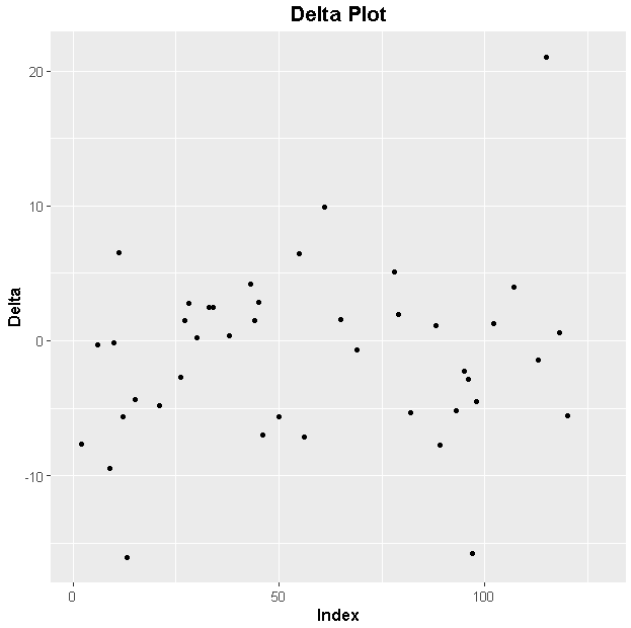


Figura 10.2.1: o gráfico de dispersão produziu uma faixa estreita de valores uniformemente espalhados em torno de zero.

A Figura 10.2.1 nos ajuda a confirmar as conclusões acima, o gráfico de dispersão produziu uma faixa estreita de valores uniformemente espalhados em torno de zero. Nesta figura, vemos essa distribuição, embora existam alguns valores discrepantes que estão mais de dez pontos acima ou abaixo de zero [70].

É importante perceber que se não tivéssemos definido uma semente de número aleatório, a função `sample()` retornará uma permutação aleatória diferente cada vez que a executarmos. Essas diferentes permutações particionarão diferentes processadores (ou seja, linhas no quadro de dados) nos conjuntos de treinamento e teste. Assim, se executarmos este experimento novamente com exatamente as mesmas entradas sem definir `set.seed(1234)` primeiro, provavelmente obteremos um intervalo de confiança diferente e um

[70] Lembrando que Δ no nosso contexto são os erros de predição (as diferenças entre os valores previstos e os valores reais). Então se um modelo está fazendo previsões boas, esses erros devem estar pouco espalhados e próximos de zero. E Index são as posições do valores no *data frame*.

gráfico de dispersão Δ_i. Por exemplo, quando repetimos o mesmo teste cinco vezes com entradas idênticas, obtemos os seguintes intervalos de confiança: $[-1,94,1,46]$, $[-1,95,2,68]$, $[-2,66,3,81]$, $[-6,13,0,75]$, $[-4,21,5,29]$. Variar a fração dos dados que atribuímos aos conjuntos de treinamento e teste alterando `f` também altera os resultados.

10.2.3 Previsões entre *Data Sets* Diferentes

Temos vários resultados de *benchmark* adicionais no arquivo de dados que podemos usar para esses testes. Como exemplo, utilizaremos o modelo que desenvolvemos a partir dos dados do Int2000 para prever o desempenho do benchmark Fp2000. Primeiro treinamos o modelo desenvolvido usando os dados do Int2000, `int00.lm`, usando todos os dados do Int2000 disponíveis no quadro de dados int00.dat. Em seguida, prevemos os resultados do Fp2000 usando este modelo e os dados do `fp00.dat`. Novamente, atribuímos as diferenças entre os resultados previstos e reais ao vetor `delta`. Os comandos correspondentes estão na Listagem 10.13:

```
> t.test(delta, conf.level = 0.95)
> int00.lm <- lm(nperf ~ clock + cores + voltage + channel +
L1icache + sqrt(L1icache) + L1dcache + sqrt(L1dcache) +
L2cache + sqrt(L2cache), data = int00.dat)
> predicted.dat <- predict(int00.lm, newdata=fp00.dat)
> delta <- predicted.dat - fp00.dat$nperf
> t.test(delta, conf.level = 0.95)
One Sample t-test
data: delta
t = 1.5231, df = 80, p-value = 0.1317
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
-0.4532477 3.4099288
sample estimates:
mean of x
1.478341

```

Listagem 10.13: treinando modelos com *data sets* diferentes.

O intervalo de confiança resultante para os valores `delta` contém zero é é relativamente pequeno. Este resultado sugere que o modelo desenvolvido utilizando os dados do Int2000 é razoavelmente bom na previsão dos resultados do programa de referência Fp2000 [71]. O gráfico de dispersão na Figura 10.2.2 mostra os valores delta resultantes para cada um dos processadores que usamos na previsão. Os resultados tendem a ser distribuídos aleatoriamente em torno de zero, como seria de esperar de um bom modelo de regressão. Observe, entretanto, que alguns dos valores diferem um pouco de zero. O desvio positivo máximo é quase 20 e a magnitude do maior valor negativo é superior a 43. O intervalo de confiança sugere resultados relativamente bons, mas este gráfico de dispersão mostra que nem todos os valores são bem previstos.

[71] Para o bangalô, fazer o gráfico de dispersão usando os valores de `nperf` int2000 × `nperf` Fp2000.

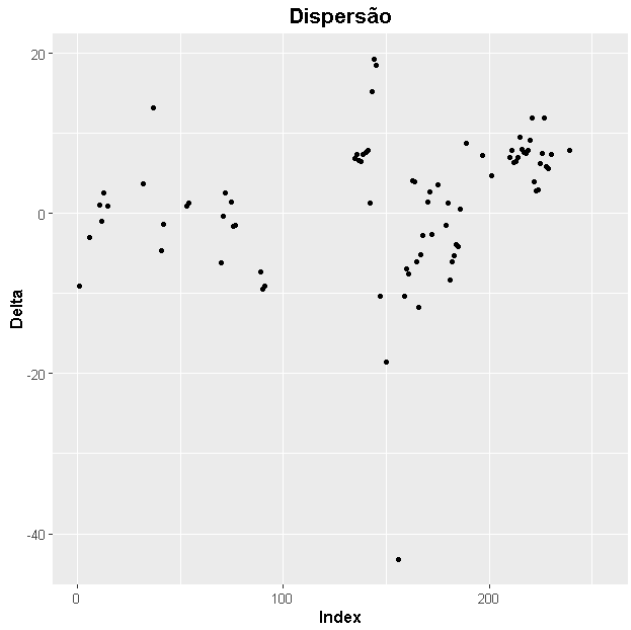


Figura 10.2.2: o gráfico de dispersão empregando os dados de int2000 para prever o fp2000.

11 Detecção de Agrupamentos e de Valores Discrepantes

Grandes quantidades de dados são coletadas diariamente a partir de imagens de satélite, equipamentos biomédicos, de segurança, marketing, buscas na web, geoespaciais ou outros dispositivos automáticos. A extração de conhecimento desses grandes volumes de dados excede em muito as capacidades humanas[72].

[72]Essa Seção foi fortemente apoiada no trabalho de [22].

A clusterização é um dos métodos importantes de mineração de dados para a descoberta de *insights* em dados multidimensionais. O objetivo da clusterização é identificar padrões ou grupos de objetos similares dentro de um conjunto de dados de interesse.

Na literatura, é referida como ”reconhecimento de padrões”ou ”aprendizado de máquina não supervisionado- ”não supervisionado”porque não somos guiados por ideias a priori sobre quais variáveis ou amostras pertencem a quais clusters. ”Aprendizado”porque o algoritmo ”aprende”a clusterizar.

A análise de cluster é popular em muitos campos, incluindo:

- Na pesquisa sobre câncer, para classificar pacientes em subgrupos de acordo com seu perfil de expressão gênica. Isso pode ser útil para identificar o perfil molecular de pacientes com prognóstico bom ou ruim, bem como para entender a doença.

- No *marketing*, para segmentação de mercado, identificando subgrupos de clientes com perfis similares que possam ser receptivos a uma forma particular de publicidade.
- No planejamento urbano, para identificar grupos de edificações de acordo com seu tipo, valor e localização.

A classificação de observações em grupos requer alguns métodos para calcular a distância ou a (dis)similaridade entre cada par de observações. O resultado de este cálculo é conhecido como matriz de dissimilaridade ou distância.

11.1 MÉTODOS DE MEDIÇÃO DE DISTÂNCIAS

A escolha das medidas de distância é um passo crítico no agrupamento. Ele define como a similaridade de dois elementos (x, y) é calculada e influenciará a forma dos aglomerados. Os métodos clássicos para medidas de distância são as distâncias euclidiana e de Manhattan, que são definidos como segue:

Distância Euclidiana: A distância euclidiana entre dois pontos $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ e $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ em um espaço n-dimensional é dada por:

$$d_{\text{Euclidiana}}(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

Note a conformação de p e q na Figura 11.1.1.

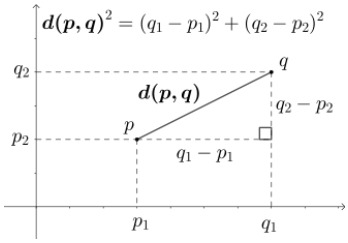


Figura 11.1.1: distância euclidiana entre p e q.

Distância Manhattan:

A distância de Manhattan (também conhecida como distância do táxi) entre dois pontos $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ e $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ em um espaço n-dimensional é dada por[73]:

$d_{\text{Manhattan}}(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} |p_i - q_i|$.

Há ainda outros como Eisen cosine correlation distance, kendall, spearman and pearson correlation distance[74].

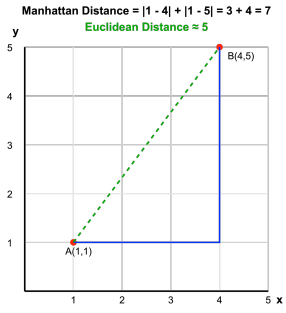


Figura 11.1.2: distância Manhattan entre A e B.

11.2 MEDIDAS DE DISTÂNCIA

As medidas de distância são um componente essencial de muitas análises de resultados. Aqui, revisaremos as propriedades das medidas de distância mais comumente utilizadas..

As medidas de distância podem ser calculadas entre parcelas (também conhecidas como unidades amostrais; as linhas em sua matriz de dados) ou espécies (também conhecidas como variáveis; as colunas em sua matriz de dados).

[73] Da Figura 11.1.1 podemos extrair também a Distância Manhattan como $|p_1 - q_1| + |p_2 - q_2|$. Expressa mais claramente na Figura 11.1.2.

[74] Pearson correlation analysis is the most commonly used method. It is also known as a parametric correlation which depends on the distribution of the data. - Kendall and Spearman correlations are non-parametric and they are used to perform rank-based correlation analysis

No entanto, a maioria das análises é baseada nas distâncias entre parcelas, e é isso que assumirei ao longo destas notas.

11.2.1 Propriedades Desejáveis de Medidas de Distância

A primeira propriedade é o **zero se idêntico**. Se as unidades amostrais A e B tiverem os mesmos valores para todas as variáveis, a distância entre elas deve ser zero. Essa propriedade é verdadeira para todas as medidas de distância que consideraremos.

A segunda propriedade é a **positividade**. Se as unidades amostrais A e B não tiverem os mesmos valores para todas as variáveis (ou seja, não são idênticas), a distância entre elas deve ser positiva. Como a distância é zero quando elas são idênticas (propriedade 1), pois o que significaria uma distância negativa?

A terceira propriedade desejável é a **simetria**. Uma medida de distância é simétrica se a distância de A para B for igual à distância de B para A. Isso é verdadeiro para todas as medidas de distância que consideraremos[75].

[75] Apenas como discussão: há situações em que a simetria não se aplica, como em aplicações de mapeamento. Por exemplo, em rotas de condução com ruas de mão única, a distância percorrida para ir de um ponto a outro pode não ser a mesma na volta. Outra situação é a estimativa de tempo de viagem que incorpora a elevação do terreno, onde subir pode levar mais tempo do que descer.

A quarta propriedade refere-se à **métrica ou semimétrica**. Em contextos onde se calcula a distância entre múltiplas unidades amostrais, uma medida métrica segue o teorema da desigualdade triangular da geometria euclidiana (ver Figura 11.2.1). Isso significa que a distância direta entre duas unidades amostrais (por exemplo, de A a C) é sempre menor ou igual à soma das distâncias entre outras unidades intermediárias (por exemplo, de A a B e de B a C). Medidas que seguem essa regra são chamadas de "medidas de distância". Já as medidas que não seguem essa desigualdade são conhecidas como semimétricas ou "medidas de dissimilaridade", onde a geometria euclidiana pode não ser válida.

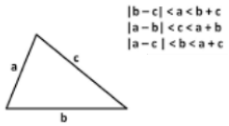


Figura 11.2.1: desigualdade triangular.

Por fim, a quinta propriedade considera se existe um **máximo constante**. Algumas medidas não possuem um limite superior, como a distância física entre unidades amostrais, que pode sempre ser maior em outro par de unidades. Outras medidas têm um máximo constante, o que significa que a distância não pode exceder certo valor. Isso ocorre frequentemente quando a distância é expressa como uma proporção de um total, limitando os valores a serem menores ou iguais a 1. A presença de um máximo constante permite que a dissimilaridade seja convertida em similaridade. Por exemplo, se a dissimilaridade entre duas unidades for 0,25 em uma medida cujo máximo é 1, isso é equivalente a dizer que a similaridade entre elas é 0,75 (ou seja, 1 – 0,25) e *vice-versa*.

11.2.2 A Matriz Distância

A **matriz de distâncias** é formada quando uma medida de distância é aplicada a múltiplas amostras, resultando no cálculo da distância para cada combinação par de amostras. Essas distâncias são então organizadas em uma matriz de distâncias (ou matriz de dissimilaridade). Por exemplo, uma matriz de distâncias entre três amostras (A, B, C) pode ser representada da seguinte forma:

	A	B	C
A	0.0	0.8	0.4
B	0.8	0.0	0.5
C	0.4	0.5	0.0

Essa matriz possui algumas características importantes:

- **Quadrada**: Como discutido na álgebra matricial, muitas operações possíveis com matrizes são aplicáveis apenas a matrizes quadradas. - **Simétrica** (propriedade desejável nº 3): Por exemplo, a distância de A para B é a mesma que de B para A, o que faz com que a parte superior da matriz seja um espelho da parte inferior[76]. - **Diagonais com valor zero** (propriedade desejável nº 1): A distância entre uma parcela e ela mesma é zero, portanto, todos os valores ao longo da diagonal são zero. - **Primeira linha e última coluna não informativas**: Contêm informações que também são relatadas em outras partes da matriz.

[76] Cuidado! Nem sempre isso nos ajuda. Vamos ver um caso adiante.

Devido a essas características, uma matriz de distâncias é frequentemente apresentada de forma mais concisa como uma matriz triangular inferior:

	A	B
B	0.8	
C	0.4	0.5

Embora essa forma não se pareça com uma matriz tradicional e não seja quadrada nem simétrica, ainda é descrita como uma matriz de distâncias. Mesmo que pareça ter apenas duas linhas e duas colunas, ela continua sendo uma matriz de distâncias 3 x 3.

Uma matriz de distâncias resume as distâncias entre cada par de unidades amostrais. O número de distâncias pareadas únicas escala de acordo com o número de unidades amostrais. Para n unidades amostrais, há $n \times n = n^2$ combinações pareadas. No entanto, como a matriz de distâncias é simétrica com zeros na diagonal, o número de combinações pareadas únicas é $\frac{n(n-1)}{2}$.

11.2.3 Equivalente a uma Matriz Distância de um Grafo de Redes

Vamos imaginar uma situação em que temos um grafo que sumariza os nós e arestas (enlaces) de uma rede como na Figura 11.2.2

Vamos criar o equivalente a uma matriz distância, mas aqui não é a matriz euclidiana, mas as distâncias entre os saltos. Acompanhe a Listagem 11.1.

```
1       # Carregar os pacotes necessários
2       >library(tidyverse)
3       >library(igraph)
4
5       # Criar o conjunto de arestas como um dataframe
6       > edges_set <- tibble::tribble(
```

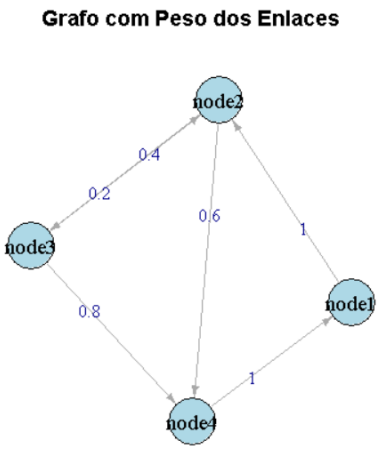


Figura 11.2.2: Um grafo que representa uma rede com quatro nós e cinco arestas. Atenção a aresta ente o nó 2 e o nó 3.

```
      ~from, ~to, ~weight,
      "node1", "node2", 1,
      "node2", "node3", 0.4,
      "node2", "node4", 0.6,
      "node3", "node4", 0.8,
      "node3", "node2", 0.2,
      "node4", "node1", 1
      ) %>% as.data.frame()
      > graph <- graph_from_data_frame(edges_set, directed = TRUE)
      > distance_matrix <- distances(graph, mode = "out", weights =
    edges_set$weight)
      > print(distance_matrix)


      node1 node2 node3 node4
      node1   0.0   1.0   1.4   1.6
      node2   1.6   0.0   0.4   0.6
      node3   1.8   0.2   0.0   0.8
      node4   1.0   2.0   2.4   0.0

```

Listagem 11.1: criando o **equivalente** a uma Matriz Distância para um grafo de redes.

11.3 DISTÂNCIA MAHALANOBIS E A DETECÇÃO DE VALORES DISCREPANTES

A distância de Mahalanobis é uma métrica usada para encontrar a distância entre um ponto e uma distribuição e é mais comumente usada em dados multivariados. Ele calcula a distância entre um ponto e a distribuição considerando

quantos desvios padrão os dois pontos estão, tornando-o útil para detectar valores discrepantes[77].

[77] Esta Seção foi extraída de [23]

A distância Euclidiana é comumente utilizada para calcular a distância entre dois pontos em um espaço de duas ou mais dimensões. No entanto, ao contrário da Euclidiana, a distância de Mahalanobis utiliza uma matriz de covariância. Por causa disso, a distância de Mahalanobis é empregada quando duas ou mais variáveis estão altamente correlacionadas, mesmo que suas escalas não sejam as mesmas. Quando duas ou mais variáveis não estão na mesma escala, os resultados da distância Euclidiana podem ser enganosos. Portanto, os *Z-scores* das variáveis devem ser calculados antes de encontrar a distância entre esses pontos. Além disso, a distância Euclidiana não não é tão eficaz se as variáveis estiverem altamente correlacionadas, já que foi criada para encontrar diferenças.

Dado dois pontos $P_1 = (6, 8)$ e $P_2 = (9, 11)$, vamos calcular a sua Distância Mahalanobis[78].

[78] Algumas manobras matemáticas (transposição) foram feitas para chegar em um valor escalar positivo que representasse uma distância. Além disso, a matriz covariância, quando aplicada diretamente (sem inversão), amplifica as distâncias nas direções de maior variância e reduz nas de menor variância. Ao inverter a matriz de covariância, as variáveis são normalizadas, reduzindo a influência das variáveis mais dispersas e aumentando a influência das menos dispersas. O que em última análise significa que aumenta a influência das variáveis com menor variância. Portanto, um pequeno desvio em uma direção onde os dados são menos dispersos (menor variância) será amplificado, tornando mais fácil a identificação de *outliers* nessa direção. Veja o exemplo a seguir na Seção 11.3.1.

$$
\begin{aligned}
D^2(P_1, P_2) &= |[6,8] - [9,11]|^T \begin{pmatrix} 4.8 & 6.9 \\ 6.9 & 10.7 \end{pmatrix}^{-1} |[6,8] - [9,11]| \\
&= \begin{pmatrix} -3 \\ -3 \end{pmatrix}^T \begin{pmatrix} 2.85 & -1.84 \\ -1.84 & 1.28 \end{pmatrix} \begin{pmatrix} -3 \\ -3 \end{pmatrix} \\
D^2(P_1, P_2) &= 4.08
\end{aligned}
$$

11.3.1 Exemplo de Detecção de Valor Discrepante

A distância de Mahalanobis é bastante eficaz na identificação de *outliers* em dados multivariados. Se houver relações lineares entre variáveis, a distância de Mahalanobis pode detectar quais observações rompem essa linearidade. Diferente de outros métodos, para encontrar os *outliers*, precisamos calcular a distância de cada ponto até o centro, que pode ser representado como o valor médio de cada variável nos dados multivariados.

Neste exemplo, podemos utilizar dados pré-definidos no R, chamados `airquality`. Usando os valores de "Temp"e "Ozone" como nossas variáveis, os passos a seguir são:

1. Encontrar o ponto central (média) de "Ozone"e "Temp".
2. Calcular a matriz de covariância de "Ozone"e "Temp".
3. Calcular a distância de Mahalanobis de cada ponto até o centro.
4. Determinar o valor de corte a partir da distribuição qui-quadrado.
5. Selecionar as distâncias que são menores que o valor de corte. Esses são os valores que não são outliers.

Antes de calcular as distâncias, vamos plotar nossos dados e desenhar uma elipse considerando o ponto central e a matriz de covariância. Podemos encontrar as coordenadas da elipse usando a função "`ellipse`"que está no pacote "`car`".

```r
# Carregar pacotes necessários
library(car)
library(ggplot2)

# Encontrando distâncias
distances <- mahalanobis(x = air , center = air.center , cov = air.cov)

# Valor de corte para distâncias da distribuição qui-quadrado
# com p = 0.95 df = 2, que é ncol(air)
cutoff <- qchisq(p = 0.95 , df = ncol(air))

## Exibir observações cujas distâncias são maiores que o valor de corte
air[distances > cutoff ,]
## Retorna: observações 30, 62, 99, 117

# Criar o data frame com os valores fornecidos
air         = airquality[c("Ozone" , "Temp")]
air         = na.omit(air)

# Calcular o ponto central (média de Ozone e Temp)
air_center <- colMeans(air)

# Calcular a matriz de covariância
cov_matrix <- cov(air)

# Gerar a elipse com base no centro e na matriz de covariância
ellipse_coords <- ellipse(center = air_center, shape = cov_matrix, radius = sqrt(qchisq(0.95, df = 2)), draw = FALSE)

# Converter as coordenadas da elipse para um data frame
ellipse_df <- as.data.frame(ellipse_coords)

# Criar o gráfico de dispersão com a elipse envolvente
figure <- ggplot(air, aes(x = Ozone, y = Temp)) +
  geom_point(size = 2) +
  geom_polygon(data = ellipse_df, aes(x = x, y = y), fill = "orange", color = "orange", alpha = 0.5) +
  geom_point(aes(x = air_center[1], y = air_center[2]), size = 5, color = "blue") +
  geom_text(aes(label = row.names(air)), hjust = 1, vjust = -1.5, size = 2.5) +
  ylab("Temperatura") + xlab("Ozônio") +
  ggtitle("Elipse Envolvendo Dados sem Outliers") +
  theme(plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
  axis.title = element_text(size = 12, face = "bold"),
  axis.text = element_text(size = 10))

# Exibir o gráfico
print(figure)
```

```
46
47
```

Listagem 11.2: detecção de outlier com Malahanobis.

A função `ellipse` toma três argumentos importantes: centro, forma e raio. O centro representa os valores médios das variáveis, a forma representa a matriz de covariância, e o raio deve ser a raiz quadrada do valor qui-quadrado[79] com dois graus de liberdade e 0,95 de probabilidade. Usamos o valor de 0,95 porque qualquer ponto fora desse intervalo será considerado um outlier, e os dois graus de liberdade se aplicam porque temos duas variáveis, "Ozone"e "Temp".

[79]Para dados multivariados, as distâncias de Mahalanobis seguem uma distribuição qui-quadrado com k graus de liberdade (onde k é o número de variáveis, como "Ozone"e "Temp"). Usamos essa distribuição para definir um valor de corte com base em um nível de confiança (como 95%). Se a distância de Mahalanobis de um ponto exceder esse valor de corte, o ponto é considerado um *outlier*.

Depois de encontrar as coordenadas da elipse, podemos criar nosso gráfico de dispersão usando o pacote `ggplot2`:

O ponto azul no gráfico mostra o ponto central. Os pontos pretos são as observações das variáveis Ozone e Temp. Como podemos ver, os pontos 30, 62, 117 e 99 estão fora da elipse laranja. Isso indica que esses pontos podem ser outliers. Considerando que essa elipse foi desenhada com base na covariância, centro e raio, podemos dizer que esses pontos também seriam identificados como *outliers* pela distância de Mahalanobis. Na distância de Mahalanobis, não desenhamos uma elipse, mas calculamos a distância entre cada ponto e o centro. Depois de encontrar as distâncias, usamos o valor qui-quadrado como valor de corte para identificar os outliers. Isso é equivalente ao raio da elipse no exemplo anterior.

A função `mahalanobis` disponível no R, no pacote `stats`, retorna as distâncias entre cada ponto e o ponto central dado. Esta função também requer três argumentos: "x", "center"e "cov". Como esperado, "x"são os dados multivariados (matriz ou dataframe), "center"é o vetor dos pontos centrais de cada variável, e "cov"é a matriz de covariância dos dados. Ao obter o valor de corte do qui-quadrado, não devemos calcular a raiz quadrada, pois a distância de Mahalanobis já retorna as distâncias ao quadrado, conforme visto na fórmula da distância de Mahalanobis.

Finalmente, identificamos os *outliers* em nossos dados multivariados. Os *outliers* são as observações (linhas) 30, 62, 99 e 117, que são os mesmos pontos fora da elipse no gráfico de dispersão vistos na Figura 11.3.1.

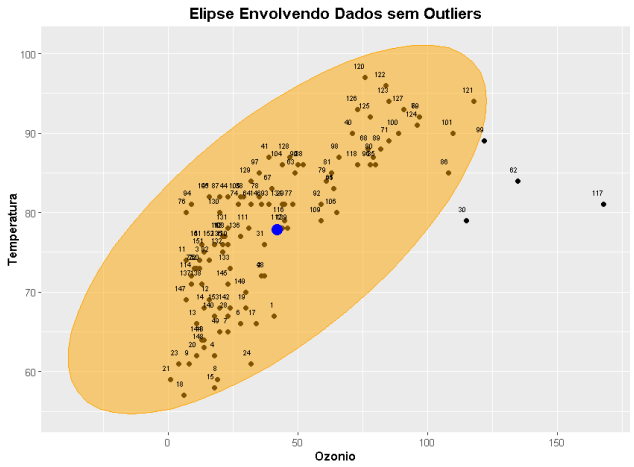


Figura 11.3.1: uso da Distância de Mahalanobis para detecção de valores discrepantes.

11.4 DETECÇÃO DE AGRUPAMENTO

Agrupamento (*clustering*) é uma técnica de tratamento de dados, que devido à sua característica não supervisionada – para agrupar não precisamos de rótulos pré-definidos dos grupos nos dados[80].

[80]Esta Seção foi fortemente apoiada nos trabalhos publicados em [22] e [24]

Agrupamento, é a tarefa de agrupar um conjunto de objetos de tal forma que objetos no mesmo grupo (chamado de *cluster*) sejam de alguma forma mais similares entre si do que com aqueles em outros grupos.

É uma técnica de aprendizado não supervisionado, portanto, o agrupamento é utilizado quando não há informações prévias disponíveis sobre os dados. Isso torna o agrupamento uma técnica muito poderosa para obter insights sobre os dados para subsidiar decisões.

Técnicas de agrupamento podem ser empregados para subsidiar hipóteses, detectar anomalias e identificar características relevantes, identificar graus de similaridade entre objetos (por exemplo, organismos) ou como um método para organizar e resumir os dados através de protótipos de *clusters* (compressão).

Os métodos de agrupamento (*clustering*) particional são utilizados para classificar observações, dentro de um conjunto de dados, em vários grupos com base em sua similaridade. Os algoritmos exigem que o pesquisador especifique o número de *clusters* a ser gerado.

Este capítulo descreve os métodos de agrupamento particional mais utilizados, incluindo:

- Agrupamento *K-means* [25], no qual cada *cluster* é representado pelo centro ou pela média dos pontos de dados pertencentes ao *cluster*. O método *K-means* é sensível a pontos de dados anômalos e *outliers*.

- Agrupamento K-medoids ou PAM [26], no qual cada *cluster* é representado por um dos objetos no *cluster*. O PAM é menos sensível a *outliers* em comparação com o K-means.
- Algoritmo CLARA (*Clustering Large Applications*), que é uma extensão do PAM adaptada para grandes conjuntos de dados.

11.4.1 *K-Means*

A ideia básica por trás do agrupamento *k-means* consiste em definir *clusters* de forma que a variação *intra-cluster* total (conhecida como variação total dentro dos clusters) seja minimizada.

Existem vários algoritmos k-means disponíveis. O algoritmo padrão é o algoritmo publicado em [27], que define a variação total dentro dos *clusters* como a soma das distâncias euclidianas ao quadrado entre os itens e o centróide correspondente:

$$W(C_k) = \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

- x_i representa um ponto de dado pertencente ao cluster C_k;
- μ_k é o valor médio dos pontos atribuídos ao cluster C_k.

Cada observação (x_i) é atribuída a um dado *cluster* de forma que a soma dos quadrados (SS, do inglês *Sum of Squares*) da distância da observação ao seu centro de cluster atribuído μ_k seja mínima.

Definimos a variação total dentro dos *clusters* (*total within-cluster variation*) da seguinte forma:

$$\text{tot.withinss} = \sum_{k=1}^{k} W(C_k) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

A soma dos quadrados dentro dos clusters (*total within-cluster sum of squares*) mede a compacidade (ou qualidade) do agrupamento, e queremos que ela seja o menor possível.

11.4.2 Inicialização dos Centróides

Como o agrupamento *k-means* busca convergir para um conjunto ótimo de centros de *clusters* (centróides) e de membros do *cluster* com base na distância desses centróides via iterações sucessivas, é intuitivo que quanto mais bem posicionados estiverem esses centróides iniciais, menos iterações serão necessárias para a convergência do algoritmo *k-means*. Isso sugere que alguma consideração estratégica para inicialização desses centróides iniciais pode ser útil. Quais métodos de inicialização de centróides existem?

- **Pontos aleatórios**: Neste método, descrito no caso "tradicional", k pontos de dados aleatórios são selecionados do conjunto de dados e usados como os centróides iniciais. Este método é obviamente altamente volátil e pode resultar em centróides mal posicionados ao longo de todo o espaço de dados.
- ***k-means*++**: este método começa atribuindo o primeiro centróide à localização de um ponto de dado selecionado aleatoriamente, e então escolhendo os centróides subsequentes dos pontos de dados restantes com base em uma probabilidade proporcional ao quadrado da distância do ponto mais próximo ao centróide existente. O efeito é uma tentativa de empurrar os centróides o mais longe possível uns dos outros, cobrindo o máximo possível do espaço de dados ocupado desde a inicialização.
- ***Sharding naive***: Este método menos conhecido de inicialização de centróides foi o tema de algumas das minhas próprias pesquisas de pós-graduação. Ele depende principalmente do cálculo de um valor de somatório composto que reflete todos os valores de atributos de uma instância. Uma vez que esse valor composto é calculado, ele é usado para ordenar as instâncias do conjunto de dados. O conjunto de dados é então dividido horizontalmente em k pedaços, ou *shards*. Finalmente, os atributos originais de cada shard são somados independentemente, suas médias são calculadas, e a coleção resultante de linhas de valores médios dos shards se torna o conjunto de centróides a ser usado na inicialização. A expectativa é que, como um método determinístico, ele deve ser mais rápido do que os métodos estocásticos e aproximar a distribuição dos centróides iniciais no espaço de dados via o valor de somatório composto. Mais detalhes podem ser encontrados em trabalhos relacionados.

11.4.3 *Partition Around Medoids* ou K-Medoids

Partition Around Medoids (PAM ou K-Medoids) é uma alternativa ao K-*Means* que utiliza pontos de dados reais, conhecidos como medoids, como centros dos *clusters*. Um medoid é definido como o objeto em um *cluster* com a soma mínima das dissimilaridades em relação a todos os outros objetos dentro do mesmo cluster. Ao contrário do K-*Means*, o K-Medoids é mais robusto a ruídos e *outliers*, o que os torna adequados para certos conjuntos de dados[81].

[81] Veja na Figura 11.4.1 um exemplo de caso de uso em redes de sensores [28].

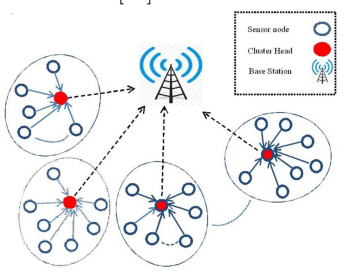


Figura 11.4.1: extensão da vida útil da rede de sensores sem fio por meio de técnicas K-Medoids em ambiente incerto.

K-*Means vs.* PAM – Casos de Uso O uso do algoritmo K-Means em mineração de texto [29] e do algoritmo PAM [30] em agrupamento de grafos estão relacionados às características dos dados e aos pontos fortes de cada algoritmo.

K-Means em Mineração de Texto

- **Natureza dos Dados**: Em *text mining*, os dados são tipicamente representados por vetores de alta dimensionalidade (por exemplo, vetores de

frequências de palavras). Esses vetores geralmente residem em espaços euclidianos, onde a média (centróide) é uma boa representação central dos dados dentro de um cluster.

- **Eficiência Computacional**: *Text mining* frequentemente lida com grandes volumes de dados (milhares ou milhões de documentos). O K-Means é um algoritmo eficiente em termos computacionais e pode ser escalado para esses grandes conjuntos de dados.
- **Homogeneidade dos Dados**: Os dados textuais tendem a ser relativamente homogêneos, com poucos outliers extremos. Isso faz com que o K-Means, que não é robusto a outliers, funcione bem na maioria dos casos.

PAM em Agrupamento de Grafos O K-Medoids é um algoritmo de agrupamento particularmente eficaz para dados de grafos devido à sua capacidade de lidar com métricas de distância arbitrárias e sua robustez a ruídos e outliers. A seguir, é apresentada uma visão geral de sua aplicação em agrupamento de grafos.

Definição de *Clusters*: no contexto do agrupamento de grafos, o K-Medoids identifica k nós no grafo como medoids. Cada nó no grafo é então atribuído ao cluster do medoid mais próximo com base na distância do caminho mais curto. Isso permite um agrupamento intuitivo dos nós com base em sua conectividade dentro do grafo.

Métricas de Distância: o K-Medoids pode utilizar diferentes métricas de distância, tornando-o versátil para dados de grafos. A métrica mais comum é o comprimento do caminho mais curto entre nós, particularmente relevante na análise de redes. Essa flexibilidade permite que o K-Medoids capture efetivamente a estrutura do grafo sem precisar transformá-lo em um espaço vetorial, como é necessário no K-Means.

11.4.4 Avaliação de Desempenho

A qualidade do agrupamento pode ser avaliada usando métricas como *silhouette scores*, que ajudam a determinar o número ideal de clusters (k). Essa avaliação é crucial para garantir que os resultados do agrupamento sejam significativos e relevantes para a análise.

11.4.5 Escolha do Hiperparâmetro Correto (k)

Determinar o número ideal de clusters (k) é uma etapa crítica tanto no K-Means quanto no K-Medoids. O método do cotovelo (*Elbow Method*) e o método da silhueta (*Silhouette Method*) são abordagens comuns para avaliar a qualidade de um valor dado de k. Esses métodos fornecem insights sobre a estrutura dos dados e auxiliam na seleção do hiperparâmetro apropriado.

11.4.6 Algoritmos de Detecção de Agrupamentos

Nosso primeiro exemplo foi criado artificialmente de modo a ser mais didático. Na Listagem 11.3 os passos estão comentados e facilitarão o acompanhamento e em seguida na Figura 11.4.2 o resultado gráfico.

Depois do carregamento dos pacotes, note que o pacote `cluster` fornece a implementação do algoritmo PAM, que é uma variação do K-means, que utiliza medoids em vez de centróides.

Em seguida carrega os dados criados artificialmente, aplica o Algoritmo PAM (K-medoids), solicitando a identificação de 3 *clusters*[82]. E a função `define_region` é uma função auxiliar para definir a posição dos gráficos no layout.

[82]Note que aqui definimos manualmente quantos clusters queremos identificar, nesse caso 03. Se não soubéssemos a priori quantos clusters existem, precisaríamos determinar o número ideal de clusters antes de aplicar o algoritmo. Há várias abordagens para fazer isso, algumas das mais comuns incluem: Método do cotovelo, da silhueta, etc.

```
1   # Carregar pacotes necessários
2   library(data.table) # data handling
3   library(ggplot2) # visualisations
4   library(gridExtra) # visualisations
5   library(grid) # visualisations
6   library(cluster) # PAM - K-medoids
7
8   set.seed(54321)
9   # Aqui tres distribuição gaussianas e 3 outliers foram
    adicionados para evidenciar o agrupamento
10  data_example <- data.table(x = c(rnorm(10, 3.5, 0.1), rnorm(10,
    2, 0.1),
11  rnorm(10, 4.5, 0.1), c(5, 1.9, 3.95)),
12  y = c(rnorm(10, 3.5, 0.1), rnorm(10, 2, 0.1),
13  rnorm(10, 4.5, 0.1), c(1.65, 2.9, 4.2)))
14
15  gg1 <- ggplot(data_example, aes(x, y)) +
16  geom_point(alpha = 0.75, size = 8) +
17  theme(plot.title = element_text(size = 16, face = "bold", hjust =
     0.5),
18  axis.title = element_text(size = 12, face = "bold"),
19  axis.text = element_text(size = 10))
20
21  kmed_res <- pam(data_example, 3)$clustering
22
23  data_example[, class := as.factor(kmed_res)]
24
25  gg2 <- ggplot(data_example, aes(x, y, color = class, shape =
    class)) +
26  geom_point(alpha = 0.75, size = 8) +
27  theme(plot.title = element_text(size = 16, face = "bold", hjust =
     0.5),
28  axis.title = element_text(size = 12, face = "bold"),
29  axis.text = element_text(size = 10))
30
31
32  define_region <- function(row, col){
33    viewport(layout.pos.row = row, layout.pos.col = col)
34  }
35
36  grid.newpage()
37  # Create layout : nrow = 2, ncol = 2
```

```
38      pushViewport(viewport(layout = grid.layout(1, 2)))
39      # Arrange the plots
40      print(gg1, vp = define_region(1, 1))
41      print(gg2, vp = define_region(1, 2))
42
43
```

Listagem 11.3: conjunto de dados com claro agrupamento.

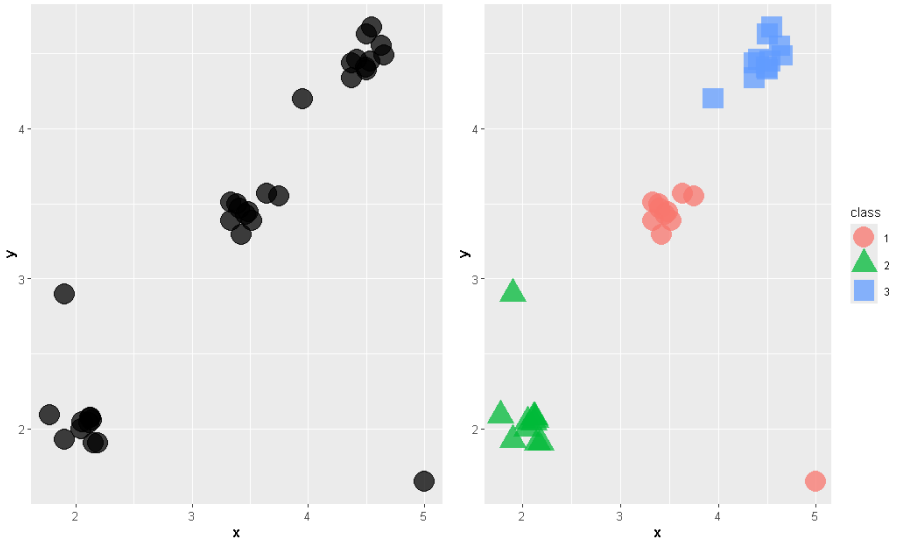


Figura 11.4.2: dados com três agrupamentos claramente evidenciados.

11.4.6.1 K-Means

Detecção de *cluster* com K-means no R(pela função kmeans), precisa a apenas do número de *clusters*, veja na Listagem 11.5 e na Figura

```
1
2       km_res <- kmeans(data_example, 3)$cluster
3
4       data_example[, class := as.factor(km_res)]
5
6       centroids <- data_example[, .(x = mean(x), y = mean(y)), by =
    class]
7
8       ggplot(data_example, aes(x, y, color = class, shape = class)) +
9       geom_point(alpha = 0.75, size = 8) +
10      geom_point(data = centroids, aes(x, y), color = "black", shape =
    "+", size = 18) +
11      theme_bw() +
12      theme(plot.title = element_text(size = 16, face = "bold", hjust =
     0.5),
13      axis.title = element_text(size = 12, face = "bold"),
14      axis.text = element_text(size = 10))
```

```
15
16
```

Listagem 11.4: agrupamento de dados.

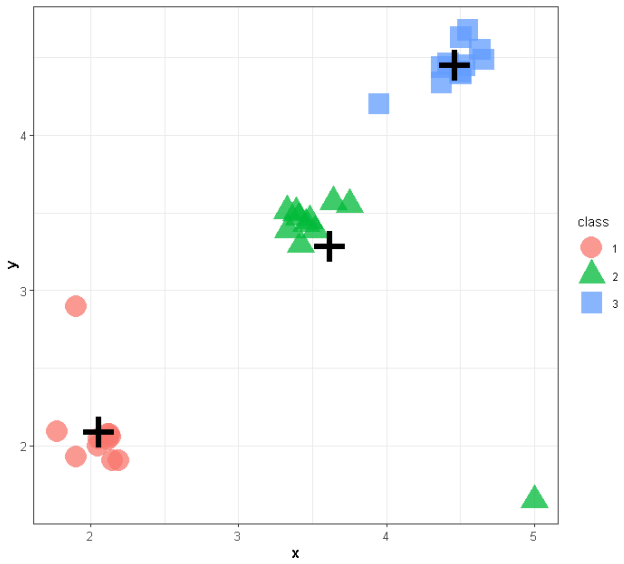


Figura 11.4.3: K-Means se mostrou falho ao alocar os centroides quando há *outliers*.

11.4.6.2 K-Medoids

O problema K-medoids pode ser resolvido pelo algoritmo Partition Around Medoids (PAM) (função pam no pacote cluster).

```
1
2       km_res <- kmeans(data_example, 3)$cluster
3
4       data_example[, class := as.factor(km_res)]
5
6       centroids <- data_example[, .(x = mean(x), y = mean(y)), by =
    class]
7
8       ggplot(data_example, aes(x, y, color = class, shape = class)) +
9       geom_point(alpha = 0.75, size = 8) +
10      geom_point(data = centroids, aes(x, y), color = "black", shape =
    "+", size = 18) +
11      theme(plot.title = element_text(size = 16, face = "bold", hjust =
     0.5),
12      axis.title = element_text(size = 12, face = "bold"),
13      axis.text = element_text(size = 10))
14
15
```

Listagem 11.5: agrupamento de dados.

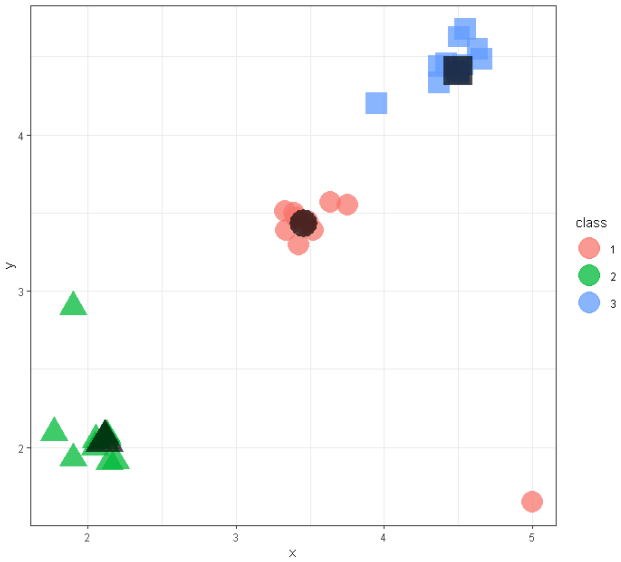


Figura 11.4.4: o K-Medoids apresenta maior robustez frente a presença de *outliers*.

11.4.7 Determinando o número ótimo de Agrupamentos

A determinação do número ótimo de clusters em um conjunto de dados é uma questão fundamental em algoritmos de particionamento, como o *k-means* , que requer que o usuário especifique o número de clusters k a ser gerado. Esses métodos incluem abordagens diretas e métodos de testes estatísticos:

1. **Métodos diretos**: consistem em otimizar um critério, como a soma total dos quadrados dentro dos *clusters* (*within-cluster sum of squares*, WSS) ou a silhueta média. Os métodos correspondentes são chamados de métodos do cotovelo (*elbow*) e da silhueta (*silhouette*), respectivamente.
2. **Métodos de testes estatísticos**: consistem em comparar evidências contra uma hipótese nula. Um exemplo é a estatística do gap (*gap statistic*).

Além dos métodos do cotovelo e GMM, apresentaremos códigos em Rpara calcular todos esses índices, de modo a entender a abordagem para então decidir qual a direção tomar para escolher o k.

11.4.7.1 Método do cotovelo

Lembre-se de que a ideia básica por trás dos métodos de particionamento, como o K-Means, é definir *clusters* de forma que a variação total dentro dos *clusters* (ou a soma total dos quadrados dentro dos clusters, WSS) seja minimizada.

O WSS total mede a compacidade do agrupamento, e queremos que ele seja o menor possível.

O método do cotovelo analisa o WSS total como uma função do número de clusters: deve-se escolher um número de *clusters* de forma que adicionar outro *cluster* não melhore significativamente o WSS total.

O número ótimo de clusters pode ser definido da seguinte forma:

1. Calcular o algoritmo de clustering (por exemplo, *k-means*) para diferentes valores de k. Por exemplo, variando k de 1 a 10 clusters.
2. Para cada k, calcular a soma total dos quadrados dentro dos clusters (WSS).
3. Traçar a curva do WSS em função do número de clusters k.
4. A localização de uma inflexão (joelho) na curva é geralmente considerada um indicador do número apropriado de *clusters*[83].

[83] Rode diversas vezes este código que gera o gráfico e veja o que acontece. Consegue explicar por que?

É importante notar que o método do cotovelo às vezes é ambíguo, e nesse caso estamos utilizando o Índice Davies Bouldin[84],a Figura 11.6, mostra isso com clareza. Além do mais no *dataframe* em que estamos trabalhando (o `data_example`), tem claramente apenas 3 *clusters*.

[84] Além do índice Davies-Bouldin, existem vários outros índices e métricas que podem ser usados para avaliar a qualidade dos *clusters* em um conjunto de dados, como Índice de Silhueta e *Gap statistics*, entre outros.

```
    library(clusterCrit)
    # Calcula clusters usando k-means para k variando de 2 a 6
    km_res_k <- lapply(2:6, function(i) kmeans(data_example[, .(x, y)], i)$cluster)

    # Calcula o índice Davies-Bouldin para cada solução de clustering
    db_km <- lapply(km_res_k, function(j) intCriteria(data.matrix(data_example[, .(x, y)]),
    j,
    "Davies_bouldin")$davies_bouldin)

    # Converte os resultados em um data.table e plota o índice Davies-Bouldin para cada k
    ggplot(data.table(K = 2:6, Dav_Boul = unlist(db_km)), aes(K, Dav_Boul)) +
    geom_line() +
    geom_point() +
    labs(y = "Elbow-Davies_bouldin", x = "Número de Clusters K") +
    theme(plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    axis.title = element_text(size = 12, face = "bold"),
    axis.text = element_text(size = 10))


```

Listagem 11.6: Metodo *elbow* de deterninação do número de cluster. O x do ponto de inflexão ou cotovelo, mostra o k.

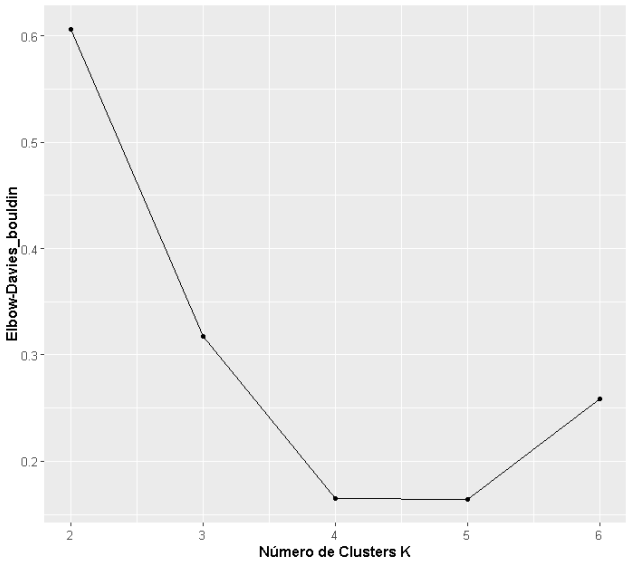


Figura 11.4.5: na Figura qual o x referente ao ponto do cotovelo? 4 ou 5?

11.4.7.2 Modelos de Mistura Gaussiana – GMM

O agrupamento com modelos de mistura gaussiana (*Gaussian Mixture Model*, GMM) é uma técnica utilizada em aprendizado de máquina não supervisionado que agrupa pontos de dados com base em suas distribuições de probabilidade[85]. No contexto da linguagem de programação R, a versatilidade do GMM reside na sua capacidade de modelar *clusters* de diferentes formas e tamanhos, tornando-o aplicável a uma variedade de cenários. A abordagem assume que os dados consistem em uma mistura de distribuições, cada uma representando um *cluster* distinto. Ao estimar os parâmetros desses componentes, o GMM identifica e separa os pontos de dados pertencentes a diferentes *clusters*.

[85] Esta seção foi baseada no trabalho publicado em [31]

Conceito Matemático No GMM, os dados são representados como uma mistura de várias distribuições gaussianas. Cada distribuição gaussiana é caracterizada pelo seu vetor de médias (centro) e matriz de covariância (dispersão e forma). Para determinar a função de densidade de probabilidades de um ponto de dado pertencer a um cluster específico, é utilizada a distribuição gaussiana correspondente. Assim, o GMM permite uma análise detalhada e flexível dos dados, possibilitando a identificação dos *clusters* em diferentes contextos.

O Algoritmo O algoritmo de agrupamento com modelos de mistura gaussiana (*Gaussian Mixture Model*, GMM) segue um processo iterativo composto por várias etapas. Inicialmente, é realizada a **inicialização**: os parâmetros, como médias ou matrizes de covariância, são definidos, escolhendo-se o número de *clusters* k. Em seguida, ocorre a **etapa de Expectativa (E)**: cada ponto de

dado é atribuído a um *cluster* com base nos parâmetros atuais, considerando a probabilidade de pertencer a cada *cluster*. Na **etapa de Maximização (M)**, os parâmetros (médias e covariâncias) são atualizados com base nas atribuições de *clusters* atuais. Esses passos de expectativa e maximização são repetidos iterativamente até que se atinja a convergência, ou seja, até que as mudanças nos parâmetros sejam mínimas.

Entendendo a Arquitetura do GMM Ao considerar o conjunto de dados como uma "paisagem"preenchida por pontos, em vez de estarem dispersos aleatoriamente, esses pontos são organizados em *clusters*, cada um com seu próprio "centro de gravidade"e "forma". O centro do *cluster* é representado por um vetor que inclui valores para cada uma das características, enquanto as matrizes de covariância descrevem a variação dos pontos de dados em relação à média, determinando assim a forma do *cluster*.

No modelo de mistura gaussiana, assume-se que cada ponto de dado pertence a um desses *clusters* e é atribuído a uma probabilidade de pertencimento. Diferentemente de algoritmos de agrupamento como o *k-means*, essa abordagem probabilística oferece vantagens: os GMMs permitem que pontos de dados pertençam a múltiplos *clusters*, reconhecendo que as fronteiras entre os *clusters* nem sempre são bem definidas. Além disso, os GMMs são capazes de determinar automaticamente o número de *clusters* por inferência, ao contrário do *k-means*, que requer a especificação prévia do número de *clusters*. Graças ao uso da matriz de covariância, os GMMs são suficientemente flexíveis para agrupar *clusters* com diferentes formas e orientações.

O R oferece vários pacotes para o agrupamento com GMMs, como o `clusterCrit` e o `ClusterR`. Este último disponibiliza um conjunto de funções para análise de GMMs, incluindo:

- `GMM()`: função principal para ajustar um GMM ao conjunto de dados.
- `predict()`: função utilizada para prever a atribuição de *cluster* para novos pontos de dados.
- A seleção do número ótimo de *clusters* pode ser realizada usando as funções `BIC()` e `AIC()`.

O pacote `mixtools` também é útil, com a função `normalmixEM()` para ajustar GMMs com opções específicas de modelo. A função `predict()` é usada para prever a probabilidade de pertencimento a um cluster, enquanto as funções `AIC` e `BIC` facilitam a comparação entre diferentes modelos de GMM.

Passos Envolvidos no Agrupamento com GMM Usando R Os passos básicos para realizar o agrupamento com GMM no R incluem:

1. Carregar e processar os dados para garantir que estejam devidamente codificados para o agrupamento.

2. Utilizar métodos como análise do cotovelo (*elbow analysis*) ou análise de silhueta (*silhouette analysis*) para selecionar o número de *clusters*.

3. Aplicar o GMM aos dados usando o pacote *mclust* no R.

4. Atribuir cada ponto de dado ao *cluster* ao qual tem maior probabilidade de pertencer.

5. Avaliar os resultados do agrupamento utilizando métricas como a pontuação de silhueta (*silhouette score*) ou o índice de Calinski-Harabasz.

Exemplo 1 – GMM:

Vamos aplicar a biblioteca mclust sobre os dados presentes em `data_example`. Pode-se otimizar várias formas de misturas (*clusters*) pelo parâmetro modelNames (verifique a função `mclustModelNames` com o comando `?mclustModelNames` para mais detalhes). Ver Listagem 11.7 e a Figura 11.4.6.

```
library(mclust)
library(ggplot2)
library(data.table)

# Executa o Mclust com 3 clusters
res <- Mclust(data_example[, .(x, y)], G = 3, modelNames = "VVV",
verbose = FALSE)

# Adiciona as classificações ao data_example
data_example[, cluster := as.factor(res$classification)]

# Plota a classificação usando ggplot2 com elipses
ggplot(data_example, aes(x = x, y = y, color = cluster, shape =
cluster)) +
  geom_point(size = 3) +
  stat_ellipse(level = 0.95) +  # Desenha as elipses com um nível
de confiança de 95%
  labs(title = "Classificação com Mclust ",
  x = "Variável X",
  y = "Variável Y",
  color = "Cluster", shape = "Cluster") +
  theme(plot.title = element_text(size = 16, face = "bold", hjust =
0.5),
  axis.title = element_text(size = 12, face = "bold"),
  axis.text = element_text(size = 10))

```

Listagem 11.7: detecção de *cluster* com o GMM através da library `mclust`.

Na Figura 11.4.7, vamos aplicar uma pequena correção para ajuste a esses dados[86].

Embora esse ajuste seja pós descoberta dos *clusters* para evidenciar os *outliers* com outra cor, em um cenário em que haja centenas ou milhares de pontos, será difícil perceber apenas visualmente quais são ou não os pontos que se pode descartar. A Listagem 11.8 mostra como imprimir os *outliers* neste cenário.

[86]Em elipses baseadas em níveis de confiança: cria-se uma elipse que inclui 95% da densidade de probabilidade do *cluster*. Os pontos fora dessa elipse estão na"cauda" da distribuição gaussiana do *cluster*, mas ainda fazem parte da mesma distribuição.Isso é esperado em uma distribuição normal, onde sempre há uma pequena probabilidade de pontos extremos (*outliers*) que ainda pertencem ao *cluster*, mas a Figura 11.4.7 tenta ajustar esse erro. Quais são as maneiras de fazer este ajuste. Para pensar no Bangalô.

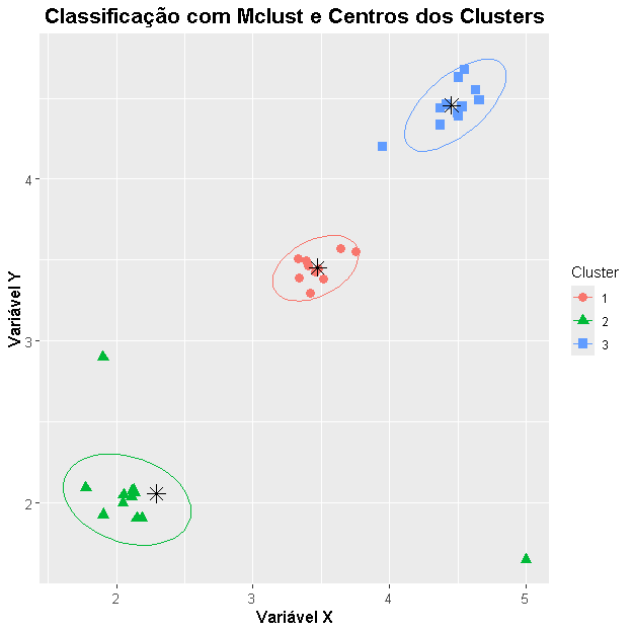


Figura 11.4.6: dado o gráfico, é possível destacar claramente os *outliers*, no entanto ainda estão classificados como parte do *cluster*.

```
>outliers <- data_example[outlier == TRUE]
>print(cbind(outliers$x, outliers$y))
[,1]    [,2]
[1,] 3.751605 3.55252
[2,] 5.000000 1.65000
[3,] 1.900000 2.90000
[4,] 3.950000 4.20000

```

Listagem 11.8: comandos para imprimir os outliers.

Exemplo 2 – BIC:
Ainda é possível empregar a função `mclust` para definir o número possível de *cluster* a partir de um *data set*. Essa função pode agrupar os dados baseada em modelos, onde os dados são assumidos como provenientes de uma mistura de várias distribuições Gaussianas. Em vez de dividir arbitrariamente os dados em grupos, o Mclust ajusta diferentes modelos estatísticos aos seus dados e usa um critério chamado *Bayesian Information Criterion* (BIC) para encontrar o número mais adequado de *clusters* e o melhor modelo para descrever seus dados[87].

- `data_example[, .(x, y)]`) Esse comando seleciona as colunas `x` e `y` do conjunto de dados `data_example` para a clusterização, representando va-

[87] Mais exemplos de modelos e algoritmos de agrupamento, assim como gráficos e códigos no R podem ser encontrados a granel em [22].

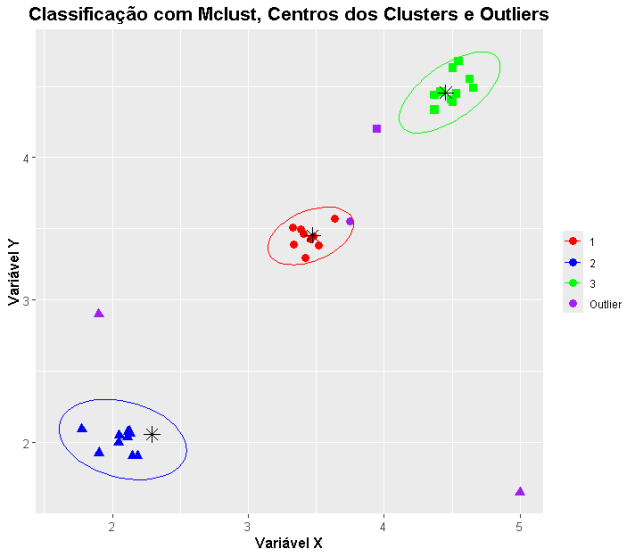


Figura 11.4.7: gráfico ajustado com os agrupamentos definidos como 1, 2 e 3 e com *outliers* rotulados de púrpura.

riáveis de interesse como medições de sinais ou características de sistemas.

- `G = 2:6`: Indica à função `Mclust` para considerar modelos com 2 a 6 clusters, testando várias opções para determinar o número ideal de grupos para os dados.
- `modelNames = c("VVV", "EEE", "VII", "EII")`: Especifica diferentes modelos estatísticos de distribuição dos clusters:
  - `VVV`: com tamanhos, formas e orientações diferentes.
  - `EEE`: semelhantes em tamanho, forma e orientação.
  - `VII`: com tamanhos diferentes, mas mesma forma e orientação.
  - `EII`: esféricos e de tamanho igual.

```
>outliers <- data_example[outlier == TRUE]
>res <- Mclust(data_example[, .(x, y)], G = 2:6, modelNames = c("VVV", "EEE", "VII", "EII"), verbose = FALSE)
>res
'Mclust' model object: (EII,6)

Available components:
[1] "call"    "data"    "modelName" "n"    "d"       "G"
[7]  "BIC"    "loglik"  "df"        "bic"  "icl"     "hypvol"
[13] "params" "z"       "classification"  "uncertainty"
>plot(res, what = "BIC")
>plot(res, what = "classification")
```

12

Listagem 11.9: uso de *Bayesian Information Criterion* , para definir o número de agrupamentos (levemente modificada por uma questão de formatação.).

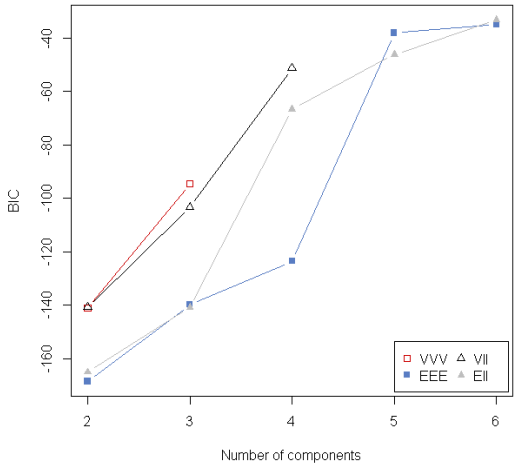


Figura 11.4.8: resultado com número de *clusters* previstos pelo modelo BIC.

O comando na Listagem 11.9 retorna o melhor modelo de agrupamento de acordo com o BIC. A partir da interpretação da resposta presente na linha 4 da Listagem, podemos concluir que o modelo escolhido foi o `BIC EII` com 6 *clusters*. Vamos acompanhar o gráfico na Figura 11.4.8, resultado co comando apresentado na linha 10. Em seguida a Figura 11.4.9 com *cluster* e *outliers* referenciados, desta vez, sem ajuste manual.

11.4.7.3 DBSCAN e HDBSCAN

DBSCAN (*Density-Based Spatial Clustering and Application with Noise*). Diferente dos GMM, que são melhor aplicados em cenários comportados onde se tem uma ideia de que seus dados podem ser modelados como uma combinação de gaussianas e quando você conhece o número aproximado de *clusters*. sendo mais adequado para dados onde os clusters são aproximadamente elípticos e sobrepostos. O DBSCAN é ideal para dados que contêm *clusters* em formatos arbitrários e que em cenários com ruído significativo[88].

[88]Esta Seção é baseada no trabalho publicado em [22].

DBSCAN é um algoritmo de clusterização baseado em densidade, introduzido em [32], que pode ser usado para identificar *clusters* de qualquer forma em um conjunto de dados que contenha ruído e outliers. A ideia básica por trás da abordagem de clusterização baseada em densidade é derivada de um método intuitivo de clusterização humana. Por exemplo, ao olhar para a Fi-

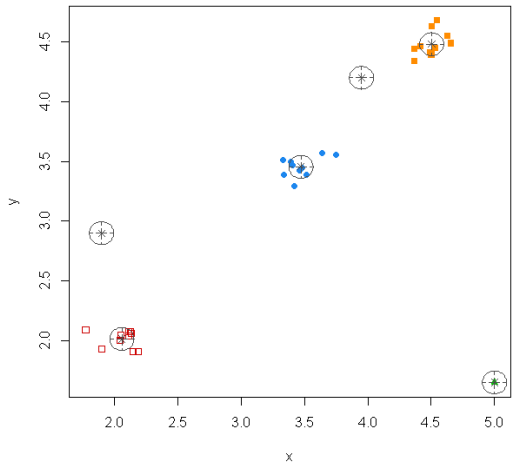


Figura 11.4.9: gráfico ajustado com os agrupamentos e *outliers* definidos.

gura 11.4.10, é fácil identificar os agrupamentos *clusters* juntamente com vários pontos de ruído, devido às diferenças na densidade dos pontos.

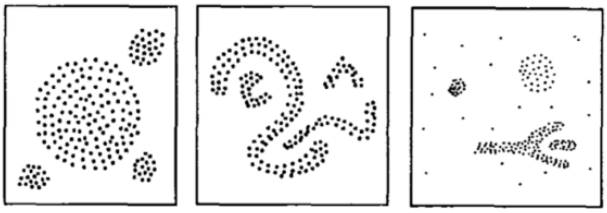


Figura 11.4.10: imagens em que para o humano é fácil detectar os agrupamentos. Fonte: [32].

Clusters são regiões densas no espaço de dados, separadas por regiões de menor densidade de pontos. O algoritmo DBSCAN é baseado nesta noção intuitiva de "*clusters*"e "ruído". A ideia central é que, para cada ponto de um cluster, a vizinhança de um determinado raio deve conter pelo menos um número mínimo de pontos.

Por que DBSCAN?

Os métodos de partição (K-means, clusterização PAM) e a clusterização hierárquica são adequados para encontrar *clusters* de forma esférica ou *clusters* convexos. Em outras palavras, eles funcionam bem apenas para *clusters* com-

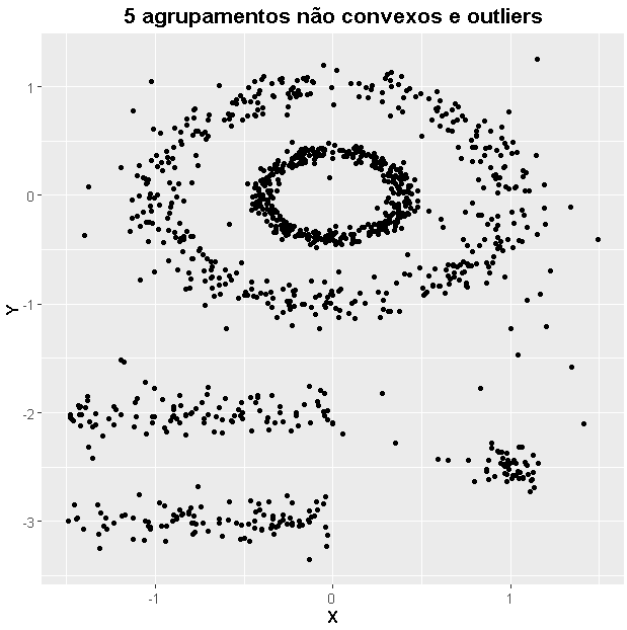


Figura 11.4.11: agrupamento não convexos com múltplos outliers.

pactos e bem separados. Além disso, eles também são severamente afetados pela presença de ruído e outliers nos dados.

Infelizmente, os dados da vida real podem conter: *i) clusters* de forma arbitrária, como os mostrados na figura abaixo (*clusters* ovais, lineares e em forma de "S"); *ii*) muitos *outliers* e ruído.

A Figura 11.4.11 mostra o gráfico de um conjunto de dados contendo *clusters* não convexos e outliers/ruídos. O conjunto de dados simulado `multishapes` do pacote `factoextra` é usado.

Dado esse tipo de dado, o algoritmo K-means terá dificuldades para identificar esses *clusters* com formas arbitrárias. Para ilustrar essa situação, usaremos o  para identificar os cluster com o K-means. Ver Listagem 11.10 e Figura 11.4.12.

```
>data("multishapes")
>df <- multishapes[, 1:2]
>set.seed(123)
>km.res <- kmeans(df, 5, nstart = 25)
>fviz_cluster(km.res, df, geom = "point",
+ ellipse= FALSE, show.clust.cent = FALSE,
+ palette = "jco", ggtheme = theme_classic())

```

Listagem 11.10: uso do K-*means* para identificar agrupamentos não convencionais.

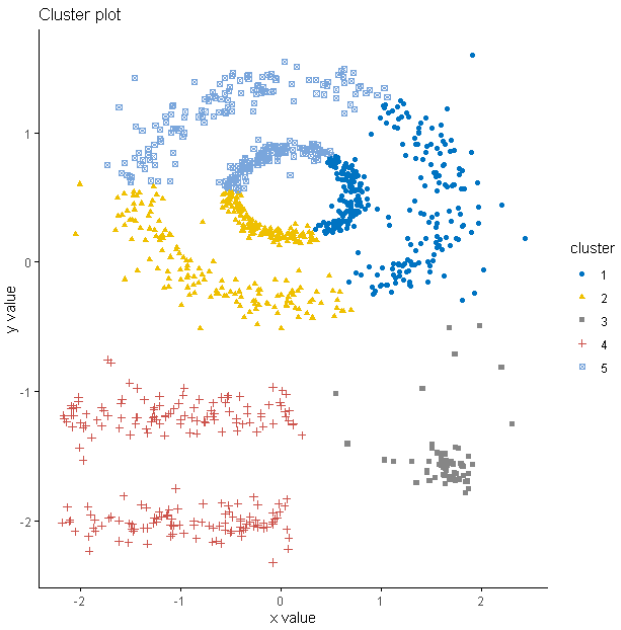


Figura 11.4.12: emprego do K-*means* para identificar agrupamentos não convencionais.

É possível acompanhar na Listagem 11.10 que o K-*means* sendo um método supervisionado precisa ser informado de quantos agrupamentos queremos identificar. No caso, são 5 agrupamentos e foi especificado na linha 5[89].

[89] Fácil ver que o K-*means* não teve um bom desempenho. 👎

O Algoritmo

O objetivo é identificar regiões densas, que podem ser medidas pelo número de objetos (pontos) próximos a um determinado ponto.

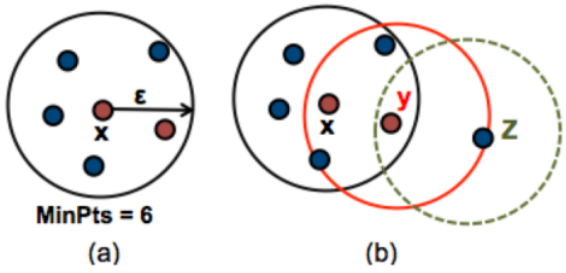


Figura 11.4.13: ϵ, x, y, z e `MinPts`=6.

Dois parâmetros importantes são necessários para o DBSCAN: ϵ e o número mínimo de pontos `MinPts`. O parâmetro ϵ define o raio da vizinhança em torno de um ponto x. É chamado de $\epsilon_{neighbour}$ de x. O parâmetro `MinPts` é o número mínimo de vizinhos dentro do raio ϵ.

Qualquer ponto `x` no conjunto de dados, com um número de vizinhos maior ou igual a `MinPts`, é marcado como um ponto central. Dizemos que `x` é um ponto de borda se o número de seus vizinhos for menor que `MinPts`, mas ele pertence à $neighbour_\epsilon$ de algum ponto central `z`. Finalmente, se um ponto não é nem central, nem de borda, então ele é chamado de ponto de ruído ou *outlier*.

A Figura 11.4.13b. mostra os diferentes tipos de pontos (centrais, de borda e *outliers*) usando `MinPts` $= 6$. Aqui x é um ponto central porque $neighbour_\epsilon = 6$, y é um ponto de borda porque $neighbour_\epsilon(y) <$ `MinPts` , mas pertence à $neighbour_\epsilon$ do ponto central x. Finalmente, z é um ponto de ruído.

Começamos definindo 3 termos necessários para entender o algoritmo DBSCAN:

- **Acessível por densidade direta**: Um ponto "A" é acessível por densidade direta a partir de outro ponto "B" se: *i)* "A" está na $neighbour_\epsilon$ de "B" e *ii)* "B" é um ponto central.
- **Acessível por densidade**: Um ponto "A" é acessível por densidade a partir de "B" se houver um conjunto de pontos centrais ligando "B" a "A".
- **Conectado por densidade**: Dois pontos "A" e "B" são conectados por densidade se houver um ponto central "C", tal que tanto "A" quanto "B" sejam acessíveis por densidade a partir de "C".

Um *cluster* baseado em densidade é definido como um grupo de pontos conectados por densidade. O algoritmo de clusterização baseado em densidade (DBSCAN) funciona da seguinte maneira:

1. Para cada ponto x_i, calcule a distância entre x_i e os outros pontos. Encontre todos os pontos vizinhos dentro da distância ϵ do ponto inicial (x_i). Cada ponto, com um número de vizinhos maior ou igual a `MinPts`, é marcado como ponto central ou visitado.
2. Para cada ponto central, se ainda não estiver atribuído a um *cluster*, crie um novo *cluster*. Encontre recursivamente todos os seus pontos conectados por densidade e atribua-os ao mesmo cluster que o ponto central.
3. Itere pelos pontos restantes não visitados no conjunto de dados. Aqueles pontos que não pertencem a nenhum cluster são tratados como outliers ou ruído.

Estimativa de Parâmetros

- `MinPts`: Quanto maior o conjunto de dados, maior deve ser o valor escolhido para `minPts`. `MinPts` deve ser escolhido como pelo menos 3.
- ϵ: O valor para ϵ pode ser escolhido usando um gráfico de k-distância, plotando a distância até o k-ésimo vizinho mais próximo, onde $k =$ `MinPts`. Bons valores de `eps` são onde esse gráfico mostra uma inflexão acentuada.

Exemplo 3 – DBSCAN
Aqui, usaremos o pacote `fpc` do R para calcular o DBSCAN. Também é possível usar o pacote `dbscan`, que fornece uma reimplementação mais rápida do algoritmo DBSCAN em comparação com o pacote `fpc`.

Usaremos novamente o pacote `factoextra` para visualizar os *clusters*.

```
install.packages("fpc")
install.packages("dbscan")
install.packages("factoextra")
# Carregar os dados
data("multishapes", package = "factoextra")
df <- multishapes[, 1:2]

# Computar DBSCAN usando o pacote fpc
library("fpc")
set.seed(123)
db <- fpc::dbscan(df, eps = 0.15, MinPts = 5)

# Identificar os outliers (pontos que não estão em nenhum cluster)
outliers <- df[db$cluster == 0, ]



# Plotar os resultados do DBSCAN
>library("factoextra")
>fviz_cluster(db, data = df, stand = FALSE,
+ellipse = FALSE, show.clust.cent = FALSE,
+geom = "point", palette = "jco", ggtheme = theme_classic())
print(db)
dbscan Pts=1100 MinPts=5 eps=0.15
        0   1   2   3  4  5
border 31  24   1   5  7  1
seed    0 386 404  99 92 50
total  31 410 405 104 99 51

# Imprimir os valores x e y dos outliers
>print("Outliers encontrados:")
>print(outliers)
x           y
71    0.0008941857  0.83613574
167   0.6183618859 -0.29393984
915  -0.7576114777 -2.68320624
925  -0.0347896077 -3.23179376
1005 -0.1322602804 -3.35346216
1006  1.3421913749 -1.58451781
1007  1.1502225716  1.25387415
.
.
```

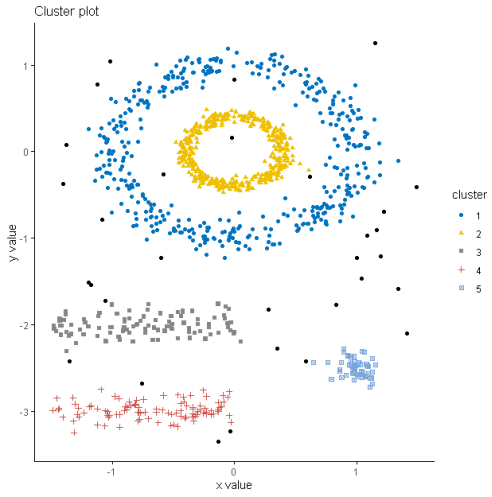


Figura 11.4.14: pode-se observar que o DBSCAN tem melhor desempenho para esses conjuntos de dados e pode identificar o conjunto correto de *clusters* em comparação com algoritmos K-*means*.

```
43 .
```

Listagem 11.11: uso do DBSCAN para identificar agrupamentos não convencionais.

Ao acompanhar os comandos da Listagem 11.11, surgem algumas perguntas, como setar valor de `MinPts`. Isso pode ser feito de forma empírica depois de plotar o gráfico através de uma inspeção visual que seja coerente com seu cenário. Já o ϵ, precisa de um método mais rigoroso. Que será visto adiante nesta Seção.

Determine optimal ϵ

O algoritmo DBSCAN requer que os usuários especifiquem os valores ótimos de `eps` e o parâmetro `MinPts`. No código do Exemplo 3, usamos `eps` = 0.15 e `MinPts` = 5.

Uma limitação do DBSCAN é que ele é sensível à escolha de `eps`, especialmente se os *clusters* tiverem densidades diferentes. Se `eps` for muito pequeno, clusters mais esparsos serão definidos como ruído. Se `eps` for muito grande, *clusters* mais densos podem ser mesclados. Se houver *clusters* com densidades locais diferentes, então um único valor de ϵ pode não ser suficiente.

O método proposto aqui consiste em calcular as distâncias dos k-vizinhos mais próximos em uma matriz de pontos.

A ideia é calcular a média das distâncias de cada ponto aos seus k vizinhos mais próximos. O valor de k será especificado pelo usuário e corresponde ao

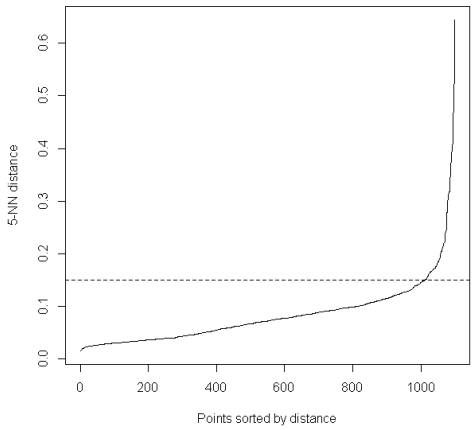


Figura 11.4.15: note que no eixo y o ponto de inflexão se aproxima muito de $\epsilon = 0.15$.

`MinPts`. Em seguida, essas k-distâncias são plotadas em ordem crescente. O objetivo é determinar o "joelho", que corresponde ao valor ótimo do parâmetroϵ.

Um joelho/cotovelo corresponde a um limiar onde ocorre uma mudança abrupta ao longo da curva de k-distâncias.

Exemplo 4 – Hyerarchical DBSCAN

A força do DBSCAN reside em sua capacidade de descobrir *clusters* com base na densidade dos pontos de dados, o que o torna particularmente útil para conjuntos de dados onde os *clusters* não são necessariamente esféricos ou linearmente separáveis. No entanto, o DBSCAN também tem suas limitações, especialmente ao lidar com clusters de densidades variáveis ou quando as configurações ótimas de parâmetros são difíceis de determinar.

```
>library("dbscan")
>data("moons")
>plot(moons, pch=20)
cl <- hdbscan(moons, minPts = 5)
>cl
HDBSCAN clustering for 100 objects.
Parameters: minPts = 5
The clustering contains 3 cluster(s) and 0 noise points.

1  2  3
25 25 50

Available fields: cluster, minPts, coredist, cluster_scores,
membership_prob, outlier_scores, hc
```

```
15 >plot(moons, col=cl$cluster+1, pch=20)
```

Listagem 11.12: uso do HDBSCAN para identificar agrupamentos não convencionais.

No R, a função `hdbscan` não usa um parâmetro ϵ explícito como DBSCAN. Em vez disso, o HDBSCAN determina automaticamente o limite de densidade apropriado para o agrupamento com base nos dados[90]. Ver gráfico gerado na Listagem 11.12 na Figura 11.4.16.

[90] Os números dos *clusters* são tipicamente 1, 2, 3, etc. Adicionar 1 `plot(moons, col=cl$cluster+1, pch=20)`, como visto na linha 15 da Listagem 11.12 a esses números desloca os índices de cor para garantir que o cluster 1 não use o índice de cor 1, que geralmente é preto no R. Isso é feito para tornar os *clusters* mais visualmente distintos, já que o preto pode não se destacar bem em um gráfico. Exemplo extraído de `https://cran.r-project.org/web/packages/dbscan/vignettes/hdbscan.html`.

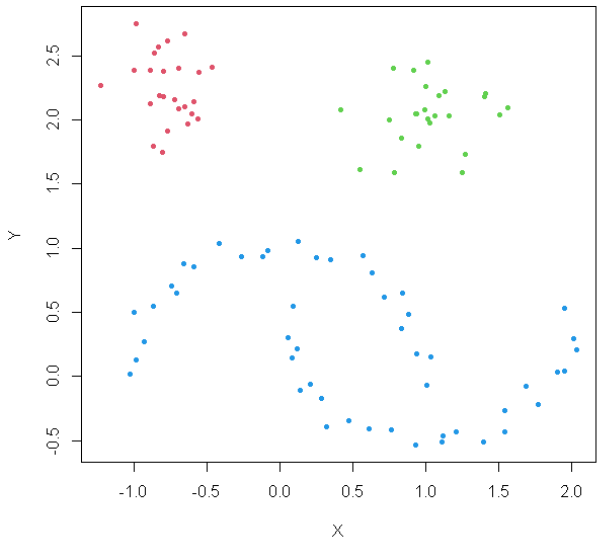


Figura 11.4.16: emprego do do HDBSCAN para determinar o número ótimo de *cluster* com o valor de ϵ também determinado automaticamente.

Bibliografia

[1] Claus O Wilke. *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*. O'Reilly Media, 2019.

[2] Elena A Allen, Erik B Erhardt, and Vince D Calhoun. Data visualization in the neurosciences: overcoming the curse of dimensionality. *Neuron*, 74(4):603–608, 2012.

[3] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.

[4] Victoria Stodden, Friedrich Leisch, and Roger D Peng. *Implementing reproducible research*, volume 546. Crc Press Boca Raton, FL, 2014.

[5] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York, 2nd edition, 2016.

[6] David S Brown. *Statistics and data visualization using R: the art and practice of data analysis*. SAGE Publications, 2021.

[7] Decoding Data Science. Standardization in statistics: Understanding and applying. Online at `https://decodingdatascience.com/standardization-statistics-understanding-and-applying/`, 2021. Accessed: 2024-08-11.

[8] Howard Wainer. Depicting error. *The American Statistician*, 50(2):101–111, 1996.

[9] Ronald A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh, 1st edition, 1925.

[10] S Belia, F Fidler, J Williams, and G Cumming. Researchers misunderstand confidence intervals and standard error bars. *Psychological Methods*, 10(4):389–396, 2005.

[11] R Hoekstra, RD Morey, JN Rouder, and EJ Wagenmakers. Robust misinterpretation of confidence intervals. *Psychonomic Bulletin & Review*, 21(5):1157–1164, 2014.

[12] G Cumming, F Fidler, and DL Vaux. Error bars in experimental biology. *The Journal of Cell Biology*, 177(1):7–11, 2007.

[13] M Krzywinski. Points of view: Elements of visual style. *Nature Methods*, 10(5):371–371, 2013.

[14] Viktor Mayer-Schönberger and Kenneth Cukier. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.

[15] Number Around Us. Hypothesis testing in r: Elevating your data analysis skills. Online at `https://medium.com/number-around-us`, 2024. Accessed: 2024-05-31.

[16] Nicholas J Horton and Suzanne S Switzer. Statistical methods in the journal. *New England Journal of Medicine*, 353(18):1977–1979, 2005.

[17] Daniel J Denis. *Univariate, bivariate, and multivariate statistics using R: quantitative tools for data analysis and data science*. John Wiley & Sons, 2020.

[18] Robert I. Kabacoff. Power analysis. Online at `https://www.statmethods.net/stats/power.html`, 2023. Accessed: 2024-08-08.

[19] David J Lilja and Greta M Linse. *Linear regression using R: An introduction to data modeling*. University of Minnesota Libraries Publishing, 2022.

[20] Jason F Cantin and Mark D Hill. Cache performance for selected spec cpu2000 benchmarks. *ACM SIGARCH Computer Architecture News*, 29(4):13–18, 2001.

[21] Peter Dalgaard. Logistic regression. *Introductory Statistics with R*, pages 227–248, 2008.

[22] Alboukadel Kassambara. *Practical guide to cluster analysis in R: Unsupervised machine learning*, volume 1. Sthda, 2017.

[23] Sergen Cansiz. What is mahalanobis distance? Online at `https://builtin.com/data-science/mahalanobis-distance`, 2023. Accessed: 2024-08-12.

[24] Petra Lauš. Overview of clustering methods in r. Online at `https://petolau.github.io/Overview-clustering-methods-in-R/`, 2023. Accessed: 2024-08-12.

[25] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, CA, USA, 1967. University of California Press.

[26] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, New York, NY, USA, 1990.

[27] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[28] Supriyan Sen, Laxminarayan Sahoo, Kalishankar Tiwary, Vladimir Simic, and Tapan Senapati. Wireless sensor network lifetime extension via k-medoids and mcdm techniques in uncertain environment. *Applied Sciences*, 13(5):3196, 2023.

[29] Siti Ramadhani, Dini Azzahra, and Z Tomi. Comparison of k-means and k-medoids algorithms in text mining based on davies bouldin index testing for classification of student's thesis. *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, 13(1):24–33, 2022.

[30] Seo Woo Hong, Pierre Miasnikof, Roy Kwon, and Yuri Lawryshyn. Market graph clustering via qubo and digital annealing. *Journal of Risk and Financial Management*, 14(1):34, 2021.

[31] deekshashukla. What is gaussian mixture model clustering using r? Online at `https://www.geeksforgeeks.org/what-is-gaussian-mixture-model-clustering-using-r/`, 2023. Accessed: 2024-08-14.

[32] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press, 1996.

A Descrição dos CSVs

Neste apêndice vamos sumarizar nossas tabelas: O arquivo nes.csv contém dados de uma pesquisa com variáveis relacionadas a características demográficas e opiniões políticas. Aqui está uma breve explicação das suas colunas:

Campo e Descrição	**Campo e Descrição**
amer_ident: Identificação como americano.	birthright_b: Opinião sobre direito de nascimento.
birthyr: Ano de nascimento.	bo_muslim: Se Barack Obama é muçulmano.
compromise: Opinião sobre comprometer-se para realizar coisas.	disc_b: Percepção de discriminação contra negros.
disc_fed: Percepção de discriminação pelo governo federal.	disc_g: Percepção de discriminação contra gays.
disc_h: Percepção de discriminação contra hispânicos.	disc_m: Percepção de discriminação contra muçulmanos.
disc_police: Percepção de discriminação pela polícia.	disc_w: Percepção de discriminação contra brancos.

dpolice.new: Percepção de discriminação pela polícia (nova variável).	econnow: Opinião sobre a economia atual.
educ: Nível de educação.	employ: Situação de emprego.
faminc: Renda familiar.	finwell: Bem-estar financeiro.
follow: Frequência com que acompanha notícias.	freetrade: Opinião sobre comércio livre.
ftblack: Sentimentos em relação a negros.	ftfem: Sentimentos em relação a feministas.
ftgay: Sentimentos em relação a gays.	fthisp: Sentimentos em relação a hispânicos.
fthrc: Sentimentos em relação a Hillary Clinton.	ftmuslim: Sentimentos em relação a muçulmanos.
ftobama: Sentimentos em relação a Barack Obama.	ftpolice: Sentimentos em relação à polícia.
ftsanders: Sentimentos em relação a Bernie Sanders.	ftsci: Sentimentos em relação a cientistas.
fttrump: Sentimentos em relação a Donald Trump.	ftwhite: Sentimentos em relação a brancos.
gender: Gênero.	healthspend: Opinião sobre gastos com saúde.
ideo5: Identificação ideológica (5 categorias).	immig_numb: Opinião sobre o número de imigrantes.
lcself: Identificação ideológica (esquerda-direita).	marstat: Estado civil.
march: Participação em marchas nos últimos 4 anos.	meet: Probabilidade de participar de reuniões.
pew_churatd: Frequência de frequência à igreja.	pid3: Identificação partidária (3 categorias).
pid7: Identificação partidária (7 categorias).	race: Raça.
stop_ever: Se já foi parado pela polícia.	stopblack: Frequência de paradas policiais de negros.
terror_worry: Preocupação com terrorismo.	turnout12: Participação na eleição de 2012.
vote12: Candidato votado em 2012.	warmcause: Causa do aquecimento global.
weight: Peso da amostra.	