R do Zero (3): Visualizando os dados e gerando documentos. ggplot2 e RMarkdown

João Pedro Oliveira, Pedro Cavalcante e Marina Merlo

13/10/2020

Markdown

Outra coisa que é possível fazer com o R é produzir documentos, para isso utilizamos a biblioteca `library(Rmarkdown)`. Aliás, a titulo de curiosidade: esse documento aqui foi escrito totalmente em `RMarkdown`.

O Markdown é uma simples linguagem de marcaçao, que serve para formataçao de textos, originalmente sendo utilizado para gerar código HTML válido a partir de textos simples. Esta linguagem aliada ao R (`Rmd`) oferece uma série de facilidades, uma delas é a possibilidade de incluir código (e o resultado das computaçoes que este código gera) de forma fluida dentro do seu documento, e exportá-lo em formatos legíveis pela maioria dos computadores de forma simples. Sabendo disso, é possível que utilizemos Rmarkdown para gerar relatórios automatizados em PDF, e documentos contendo informaçoes que sao frequentemente acessadas.

Com o `Rmd` é possível exportar arquivos nos formatos PDF, Docx e HTML. Uma particularidade do Rmarkdown é que quando se deseja exportar para PDF, o R converte o Markdown em um arquivo LaTeX, e depois compilar o LaTeX para PDF. Por conta da interaçao com o LaTeX, o Rmd também suporta equaçoes e expressoes matemáticas.

Para iniciarmos nosso, devemos criar um documento Rmd, o que é possível de ser feito dentro do RStudio indo no menu `File > New File` e selecionando a opçao `RMarkdown`. O arquivo gerado vai possuir algumas instruçoes básicas que devem ser lidas por quem nunca te contato com o Markdown.

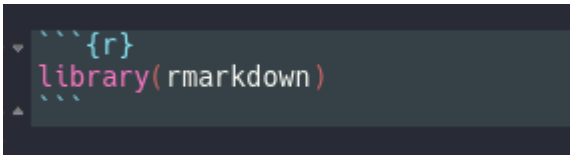
Primeiros Passos

Sintaxe básica

Para escrever em Markdown é simples, somente escreva sem se preocupar com cliques para definir formataçao e coisa do tipo, isso tudo poderá ser feito posterioremnte por meio de código. Mas, caso deseje inserir código, este deve ser colocado entre acentos graves (aquele lá da crase), isto é, o “`“. Um trecho colocado entre esses demarcadores é formatado como código...

`Este texto foi escrito entre dois acentos graves`

Contudo, esta nao é a única forma de inserirmos código, caso tenhamos certeza de que estamos inserindo código R, e que este possuirá mais de uma linha, o atralho CTRL+ALT+I (Command+ALT+I no Mac), irá criar automaticamente uma caixa em que é possível que digitemos código com mais de uma linha.



Mas texto deve ser dividido em blocos, certo? A forma mais simples de fazer isso é por meio de títulos e sub-títulos.

Títulos e subtítulos. Para que possamos fazer a divisao de blocos, normalmente recorremos a dividir os textos em seçoes, estas quais sao delimitadas por títulos e subtítulos. É possível fazer essas delimitaçoes no RMarkdown utilizando o operador `#` (sustenido), que controlará o tamanho da fonte de determinada linha de texto. O controle é feito por meio da quantidade de `#`, como pode ser visto nos exemplo abaixo:

Esse bloco contém somente um sustenido #.

Esse bloco contém dois sustenidos ##.

Esse bloco contém tres sustenidos ###.

Esse bloco contém quatro sustenidos ####. Quando formatados como título e salvas em formato PDF,

Listas e Links Em diversas situaçoes pode ser confortável ao leitor adicionar listas pontuadas ou numeradas para que possa transmitir uma ideia. Por serem tipos de listas diferentes, sintaxes diferentes sao pedidas:

Quando desejamos criar listas ordenadas/numeradas, a sintaxe é simples, se pode ser vista abaixo:

```
1. Primeiro Ponto
2. Segundo Ponto
3. Terceiro Ponto
```

O bloco de código acima gerará a seguinte lista:

1. Primeiro Ponto
2. Segundo Ponto
3. Terceiro Ponto

Como é possível ver, a sintaxe é extremamente simples, e isso facilita a escrita fluida de texto. Se nós queremos trabalhar com listas nao numeradas (unordered), podemos criar elas usando a seguinte sintaxe:

```
- Primeiro Ponto

- Segundo Ponto

- Terceiro Ponto
```

Se o interesse é inserir links no texto, a sintaxe é composta por `[Texto](Link)`. Por exemplo, um `[Google](www.google.com.br)`, irá abrir uma janela nova no navegador do leitor qual irá direcionar o usuário para a pagina inicial do Google. Vale notar que links podem estar em vários lugares, inclusive em listas, veja abaixo:

- Google
- R in Rio
- Twitter

Equaçoes em LaTeX

As equaçoes podem ser facilmente inseridas em um arquivo Rmarkdown por meio do uso de `$` como um marcador de inicio e fim de expressoes. Quem nunca teve contato com LaTeX pode estranhar a forma de escrita, contudo é somente uma questao de costume e de estudo, é possível encontrar material de LaTeX disponível na internet.

Para inserir LaTeX de em uma linha de texto, encapsule uma expressao válida dentro de cifrões, por exemplo `$f(x) = 1$`, que será transformada em $f(x) = 1$. Dando outro exemplo, temos `$\sum_{i=2}^{5}$`, que gera a visualizaçao $\sum_{i=2}^{5}$.

Para que a expressao tenha uma linha própria, é necessário que ela seja encapsulada por cifrões duplos, como `$$f(x) = 1$$`, uma boa prática é que essa expressao esteja destacada em uma linha própria, isso facilita potenciais alteraçoes futuras no texto.

$$f(x) = 1$$

$$\sum_{i=2}^{5}$$

Acima temos as duas expressoes em linhas próprias. A recomendaçao é o uso de linha própria quando a expressao é muito longa, ou deve receber destaque em seu trabalho.

ggplot2

Visualizar?

O trabalho com dados é popularmente visto como algo essencialmente numérico, e na maior parte do tempo realmente essa expectativa é atendida, porém uma parte da beleza da profissão está em transformar esse amontoado de tabelas em um gráfico capaz de condensar a informação para um público-alvo. Construir visualizações informativas é uma habilidade essencial para que cientistas de dados (e pesquisadores) consigam comunicar seus resultados de forma concisa.

Embora essencial muitos optam por ignorar a etapa da visualização de dados, o que acaba por gerar um gargalo de conhecimento que tem de ser corrigido já após estar colocado no mercado. Embora visualizar por si só seja uma concatenação de arte e ciência, existem simples técnicas que podem lhe ajudar, muitas das quais já estão inclusas no `ggplot2`. O `ggplot2` é *de-facto* a principal ferramenta de plotagem do `R`, apesar de existir a ferramenta `plot()` do `R-Base`, esta última utilizada nesta apostila.

Motivos e formas de representar dados graficamente.

Como dito antes, a visualização em R se faz principalmente com o uso de bibliotecas como o `ggplot2` e a `plotly`, sendo a última mais utilizada em contextos muito especificos que pedem interatividade ou integração com plataformas externas. O `ggplot2` foi escrito de forma a ser uma simples e poderosa ferramenta. Para tornar isso possível a potência da biblioteca deriva de algo que se asemelha a uma gramática, para entender melhor lembremos da escola, onde nos foi ensinado que uma gramática é "um sistema formal de regras para a construção de afirmações válidas" (Wilkinson, 2005, tradução própria). Ou seja, o `ggplot2` possui um sistema de regras que quando seguidas farão uma afirmação válida, esta qual será representada graficamente.

Quando vamos nos comunicar com outras pessoas normalmente utilizamos do meio de comunicação que melhor se adequa à situação, e essa regra continua válida quando o assunto são dados. Apesar de existirem possibilidades como a criação de tabelas, a mais frequente é a exposição de informação por meio de gráficos.

A escolha por gráficos normalmente ocorre pela simplicidade com qual conseguem dispor situações complexas sem a necessidade do leitor possuir conhecimento profundo dos dados apresentados.

O exemplo clássico dado para motivar alguém a desenhar gráficos é quando se é necessário apresentar um relatório de descobertas feitas a um executivo. Sabemos que executivos não são conhecidos por dar muita atenção a detalhes de implementação de algoritmos, modelos e tabelas, isso ocorre por diversos motivos. Talvez um desses motivos seja desinteresse de parte dos engravatados, outra possibilidade é que estes podem não saber como ler a informação que está disposta de forma 'crua'. Para contornar esse problema é necessário que saibamos envelopar o que queremos exibir em uma máscara gráfica. Fazer com que estatísticas descritivas e resultados de modelos se tornem algo que possa ser lido por humanos, feito isso podemos comunicar nossas descobertas para o público que o trabalho deseja atingir.

Mas se queremos comunicar, por meio de um computador, o que números querem dizer, precisamos de gráficos. Adaptando para o mundo da visualização de dados podemos dizer que a biblioteca segue uma `Gramática dos Graficos`, isto é, um conjunto de regras gramaticais que fundamentam a construção matemática de gráficos, e como representá-los de forma visualmente agradável.

Sendo direto, é possível dizer que a gramática possibilita você "construir qualquer gráfico com alguns poucos componentes básicos: um conjunto de dados, um conjunto de geoms— marcações visuais que representam pontos de dados, e um sistema de coordenadas" (RStudio, 2016). A regra para construir qualquer tipo de gráfico sempre é a mesma.

Utilizando o ggplot2

Como construir gráficos

A construção de gráficos com o ggplot2 é como construir uma escultura de Lego, no sentido de que o gráfico a ser construido é composto de "blocos" que são empilhados e conforme essa empilhar ocorre, a nossa construção vai, progressivamente, tomando sua forma final.

Indo para a parte mais prática da coisa, precisamos primeiro entender que toda construção com ggplot2 se inicia com a declaração da função `ggplot()`, é ela que informa ao `R` que ali iremos começar a construir um gráfico. Dentro desta função iremos passar o argumento `data` que receberá o conjunto de dados que será utilizado no gráfico em questão. Essa função deve ser seguida do operador `+`, que indica que há uma continuação da sequência de instruções.

```
ggplot(data = dataset) +
```

Após dizer que iremos iniciar um gráfico, iremos falar ao `ggplot` qual será o formato do nosso gráfico, isto é, qual será o `geom` a ser utilizado. O geom receberá um `aes`, que define a estética do gráfico e que possuirá como principais argumentos os eixos `x` (horizontal) e `y` (vertical) do seu gráfico.

```
ggplot(data = dataset) +
  geom_point(aes(x = ?,y = ?))
```

> o `+` no ggplot2: dentro de um contexto do ggplot2, o operador `+` possui uma função diferente, ele não executa adições numéricas, ele executa adições de elementos ggplot. Melhor dizendo, o `+` acopla elementos gráficos à base do plot.

Nosso primeiro gráfico vem da Antártica Entendidas como montar as peças para fazer um gráfico simples, nada melhor do que fazer um teste, afinal, não temos nada a perder. Para isso utilizaremos um banco de dados chamado `palmerpenguins`, ele foi criado pela Profa. Dra. Kristen Gorman que coletou os dados para sua pesquisa em estudos marinhos. O banco de dados contém dados de 344 penguins, dispersos

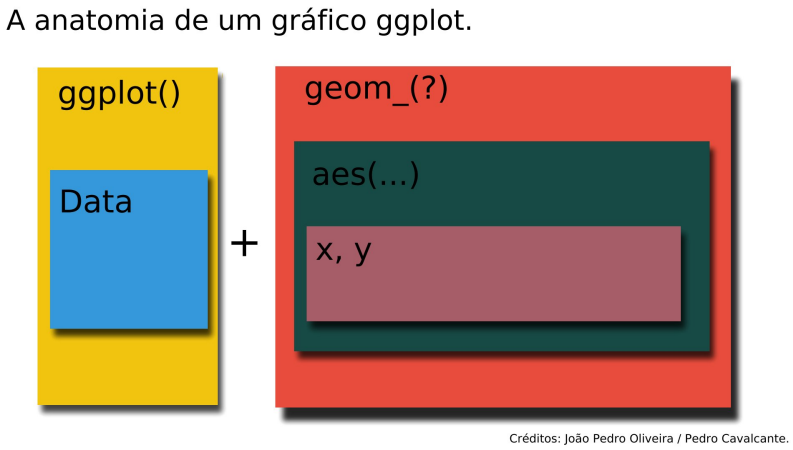


Figure 1: Anatomia de um ggplot

por 3 espécies diferentes e observados em 3 ilhas na Antártica. O nome `Palmer Penguins` vem do fato das ilhas fazerem parte do Arquipélago de Palmer - e por conta dos penguins, claro.

```
## # A tibble: 6 x 8
##   species island bill_length_mm bill_depth_mm flipper_length_~ body_mass_g sex
##   <fct>   <fct>           <dbl>         <dbl>            <int>       <int> <fct>
## 1 Adelie  Torge~           39.1          18.7              181        3750 male
## 2 Adelie  Torge~           39.5          17.4              186        3800 fema~
## 3 Adelie  Torge~           40.3          18                195        3250 fema~
## 4 Adelie  Torge~           NA            NA                 NA          NA <NA>
## 5 Adelie  Torge~           36.7          19.3              193        3450 fema~
## 6 Adelie  Torge~           39.3          20.6              190        3650 male
## # ... with 1 more variable: year <int>
```

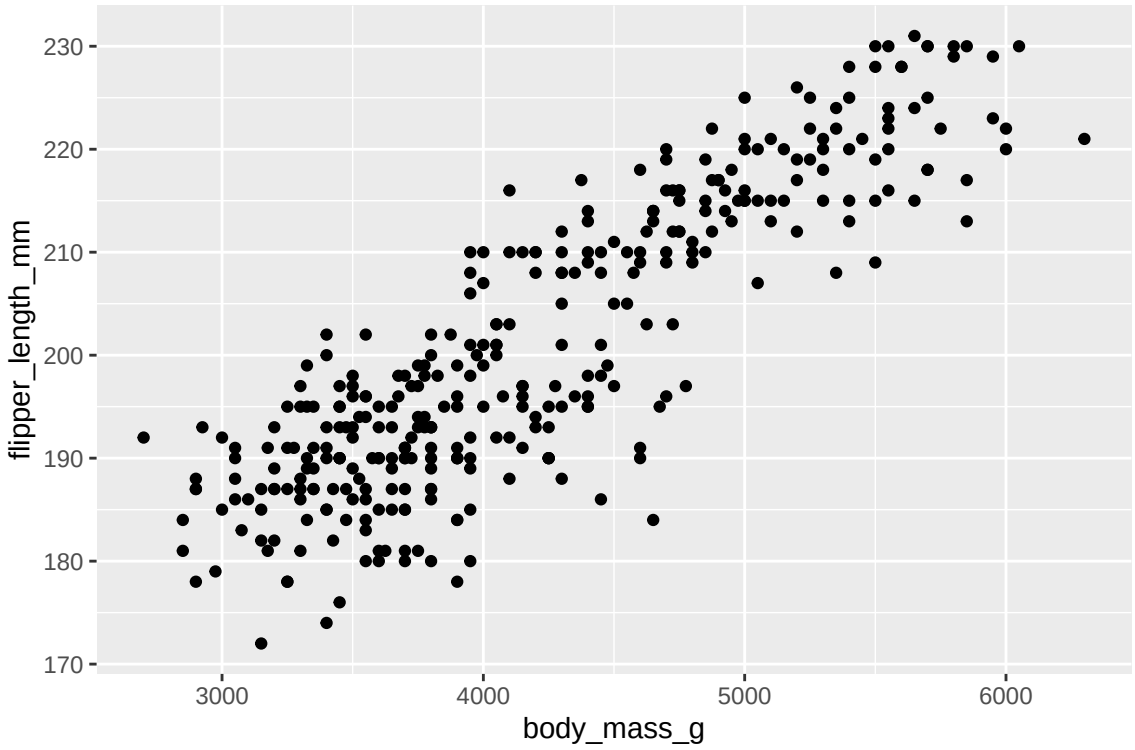
Acima temos um trecho de código que carrega o banco de dados na memória, e exibe as 5 primeiras observações, nosso `data.frame` possui 334 linhas (uma observação por penguim!) e 8 colunas, cada uma com um atributo. Abaixo temos um quadro com a explicação de cada um dos atributos:

atributo	descricao
species	Espécie do penguim
island	Ilha em qual foi feita a observação
bill_length_mm	Largura do bico em milimetros
bill_depth_mm	Altura do bico em milimetros
flipper_length_mm	Tamanho das nadadeiras em milimetros
body_mass_g	Peso do animal em gramas
sex	Sexo do animal
year	Ano do estudo

Observado o banco de dados, passamos a explorá-lo. Como todos nós temos um grande interesse acerca de penguins (afinal, como não gostar deles, são fofos), várias perguntas surgem na nossa cabeça. O primeiro grande questionamento que nos vem à cabeça é: "Existe alguma correlação entre o peso do animal e o tamanho de suas nadadeiras?", e essa pergunta é extremamente pertinente, imagina como deve ser difícil nadar com pequenas nadadeiras e grande peso.

Felizmente no nosso banco de dados temos as informações necessárias para respondermos essa pergunta, e elas se encontram nas colunas `body_mass_g` e `flipper_length_mm`.

```r
ggplot(data = penguins_data)+ # Colocamos nosso banco de dados no campo 'data'
  geom_point(aes(x= body_mass_g, y = flipper_length_mm)) # Então determinamos as variáveis x e y
```

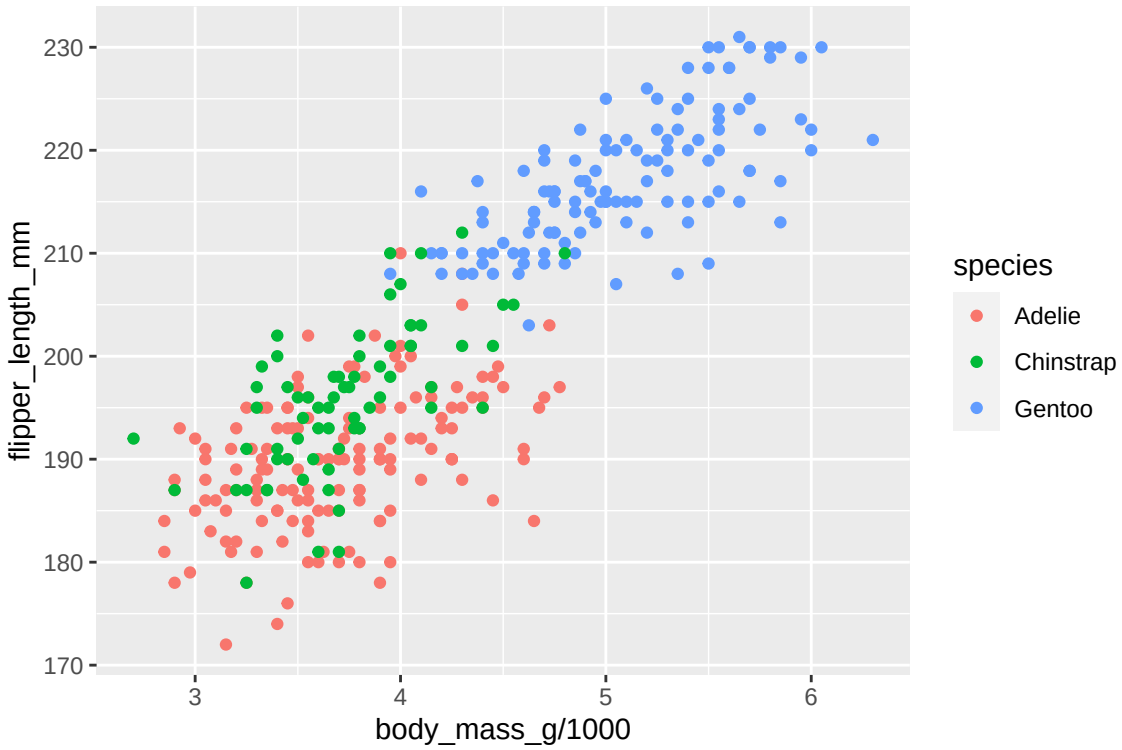


Eba! Parece que temos uma correlação positiva. Como podemos observar, o plot gerado acima possui pontos que representam a relação entre duas variáveis quantitativas. Essa situação é uma das quais o uso de gráficos de ponto (dot plots) é recomendada! Contudo, nosso gráfico está muito pouco claro e poderia mostrar mais. Um exemplo de melhoria é diferenciarmos as espécies de penguim, o que pode ser feito por meio do parâmetro `colour`/`color`, passado dentro do `aes()`.

A partir de agora, o nosso plot será representado somente pela letra `p`, leia `p` como `plot`.

```r
p = ggplot(data = penguins_data)+ # Colocamos nosso banco de dados no campo 'data'
  geom_point(aes(x= body_mass_g/1000,
                 y = flipper_length_mm,
                 colour=species)) # Então determinamos as variáveis x e y

p # exibe nosso gráfico na tela
```



Feito isso, novas informações são reveladas e agora conseguimos perceber coisas como a diferença de tamanho de nadadeira e peso entre as espécies 'Adelie', 'Chinstrap' e 'Gentoo'. Os Penguins Gentoo aparentam ser maiores e mais pesados, como os dados nos mostram. Abaixo podemos ver a média de peso de cada espécie de penguim e realmente os Gentoo's são mais pesados (na média).

species	media_peso
Adelie	3700.662
Chinstrap	3733.088
Gentoo	5076.016

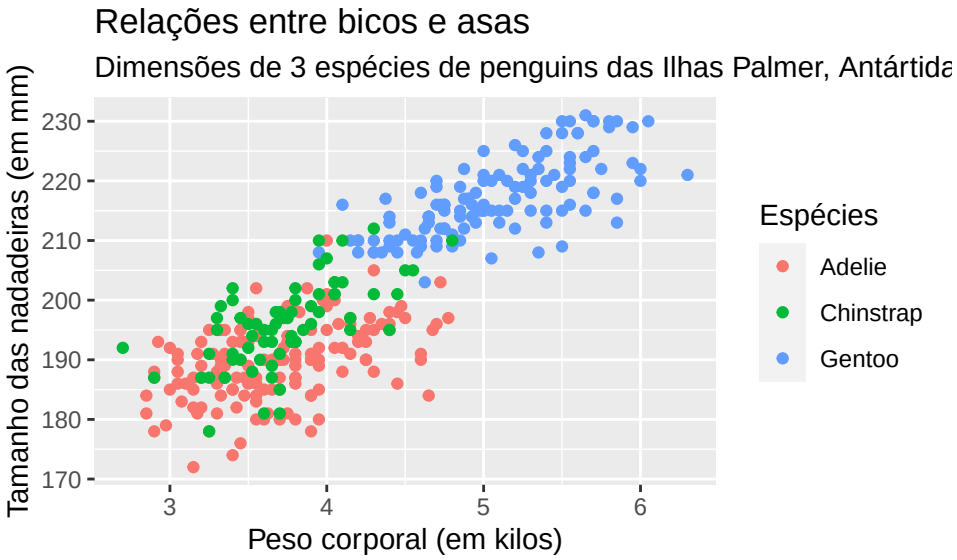
Um ajuste adicional que é necessário no nosso gráfico são as legendas, elas são extremamente importantes em um gráfico pois são responsáveis por informar ao leitor o que a visualização que está sendo apresentada a ele representa. No gráfico abaixo as legendas serão responsáveis por dizer ao leitor:

0. No que o leitor deve prestar atenção
1. O que aqueles pontos de dado representam
2. Em qual dos eixos está cada um dos elementos

Existem diversas formas de adicionar legendas aos nossos gráficos, a mais intuitiva delas é utilizando a função `ggplot2::labs()`. Como é possível observar em sua documentação, `labs` aceita uma infinidade de argumentos - quais não serão completamente explorados nesta obra -, as mais importantes são aquelas que representam texto considerado necessário para uma comunicação de sucesso, isto é, titulo, subtitulo e eixos. Utilizá-las é simples, atribui-se um texto ao parâmetro colocado dentro da função, façamos com nosso gráfico...

```r
p = p +
  labs(title= "Relações entre bicos e asas", # Título
       subtitle="Dimensões de 3 espécies de penguins das Ilhas Palmer, Antártida", #subtítulo
       x="Peso corporal (em kilos)", # eixo x
       y="Tamanho das nadadeiras (em mm)", #eixo y
       colour = "Espécies") # legenda

p
```

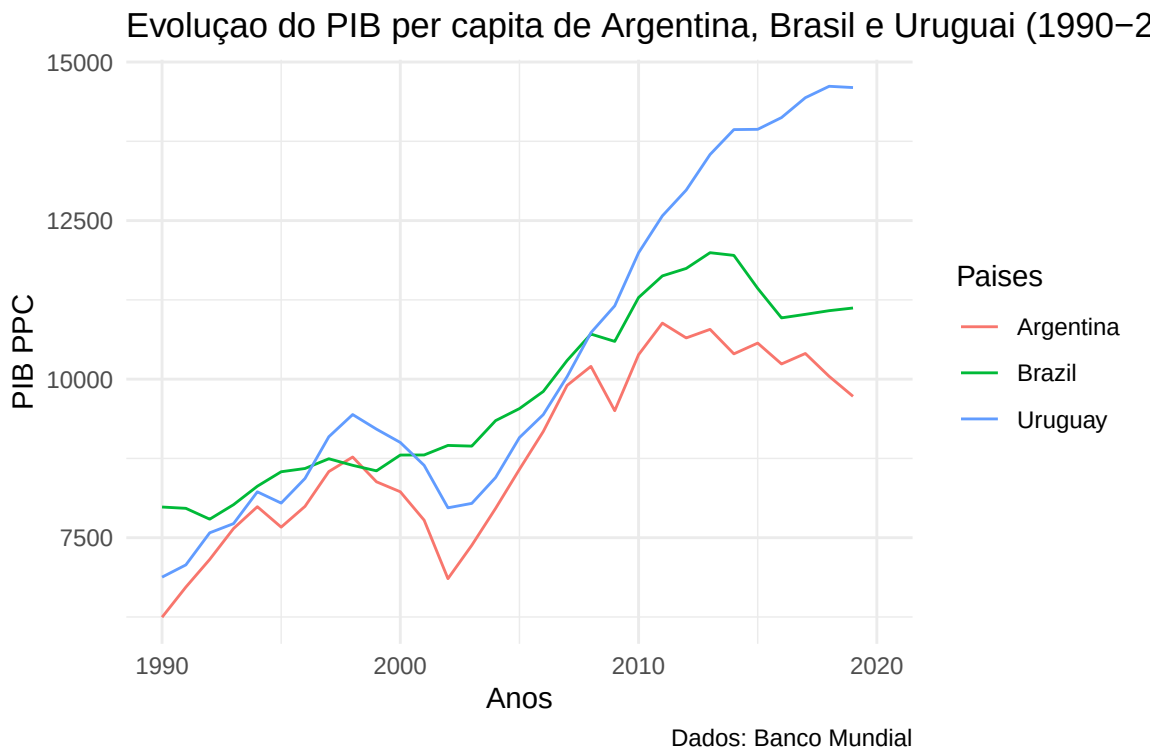


Agora nosso gráfico está muito melhor, e exibe perfeitamente o que nós queriamos. Na verdade, ele exibe até mais, contudo não esgotamos nosso banco de dados, e é possível termos insights ainda mais sofisticados com a mesma base. Agora que já entendemos como construir um gráfico, iremos mudar o `geom`, passando para um de linha.

Grafico de Linha Na seção anterior desenhamos um **dot-plot/scatterplot**, que tem como objetivo mostrar a relação entre duas variáveis, todavia nem sempre é essa a nossa intenção. Muitas vezes nós objetivamos demonstrar a existencia ou não de uma tendência com o passar do tempo, e não necessariamente uma relação. Portanto, se o objetivo é mostrar a variação de pontos na passagem do tempo, o gráfico de linha é uma alternativa excelente. Por favor, somente um eixo vertical.

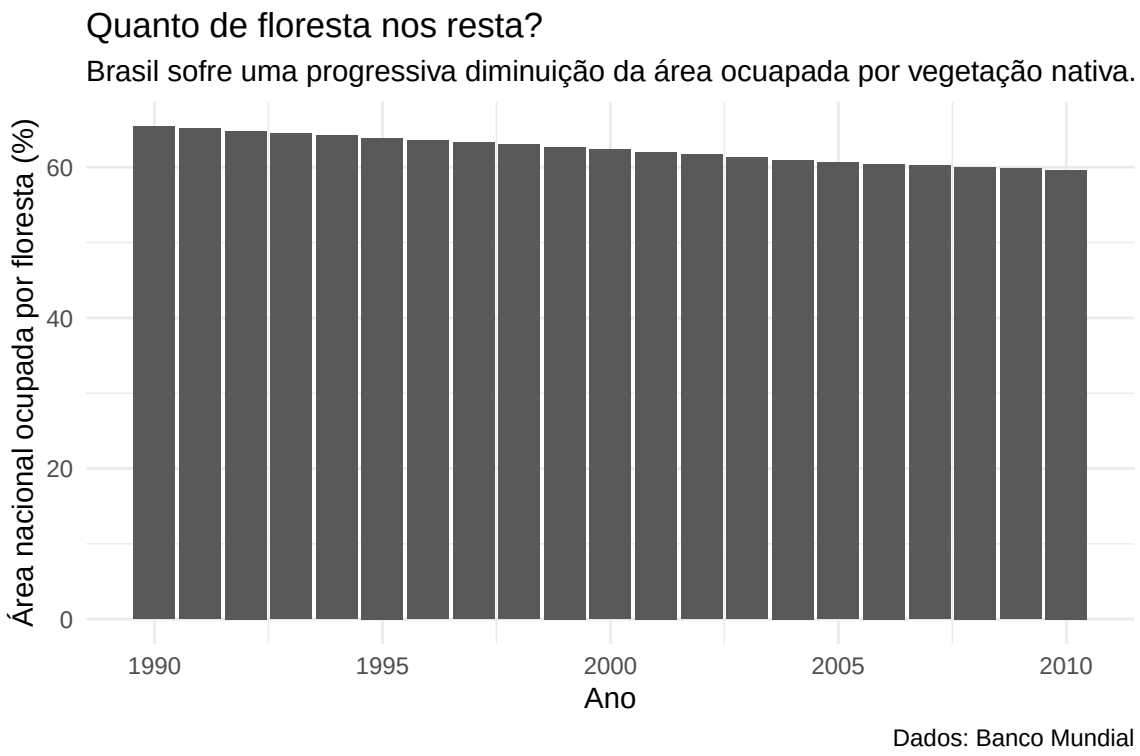
Usando a biblioteca `wbstats`, que opera como uma inferface de comunicaçao com os bancos de dados do Banco Mundial, por meio dele iremos coletas dados e depois usá-los para fazermos um simples gráfico para visualizarmos a evoluçao do PIB per capita de 3 países da América Latina: Argentina, Brasil, Uruguai.

```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```

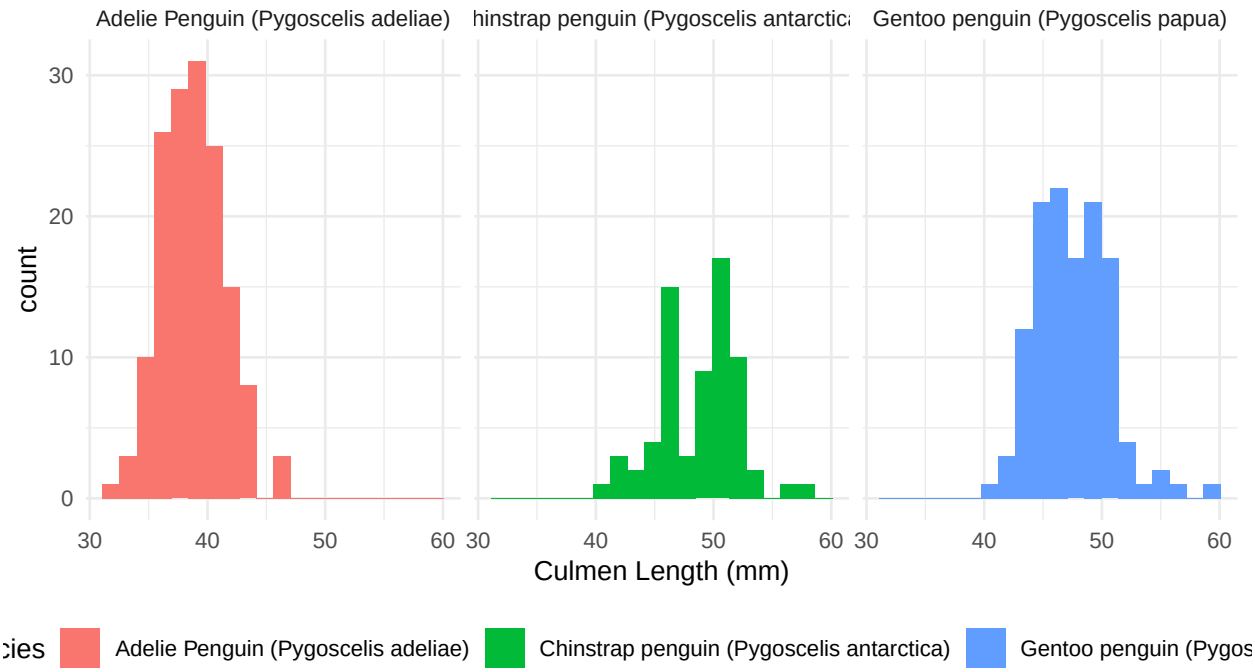


Graficos de Barra Se o objetivo é comunicar a grandeza de determinado conjunto de valores, o gráfico de barra pode ser a melhor escolha. Essa geometria é conhecida por sua versatilidade e diversidade de aplicaçoes, sendo possível ser utilizada em cenários diversos, como quando objetivamos exibir quantidades, ou grandes variaçoes no tempo. Todavia, devemos nos atentar ao fato de que barras *somente são boas alternativas para representar passagem temporal quando se sabe que as variaçoes de tempo sao grandes.*

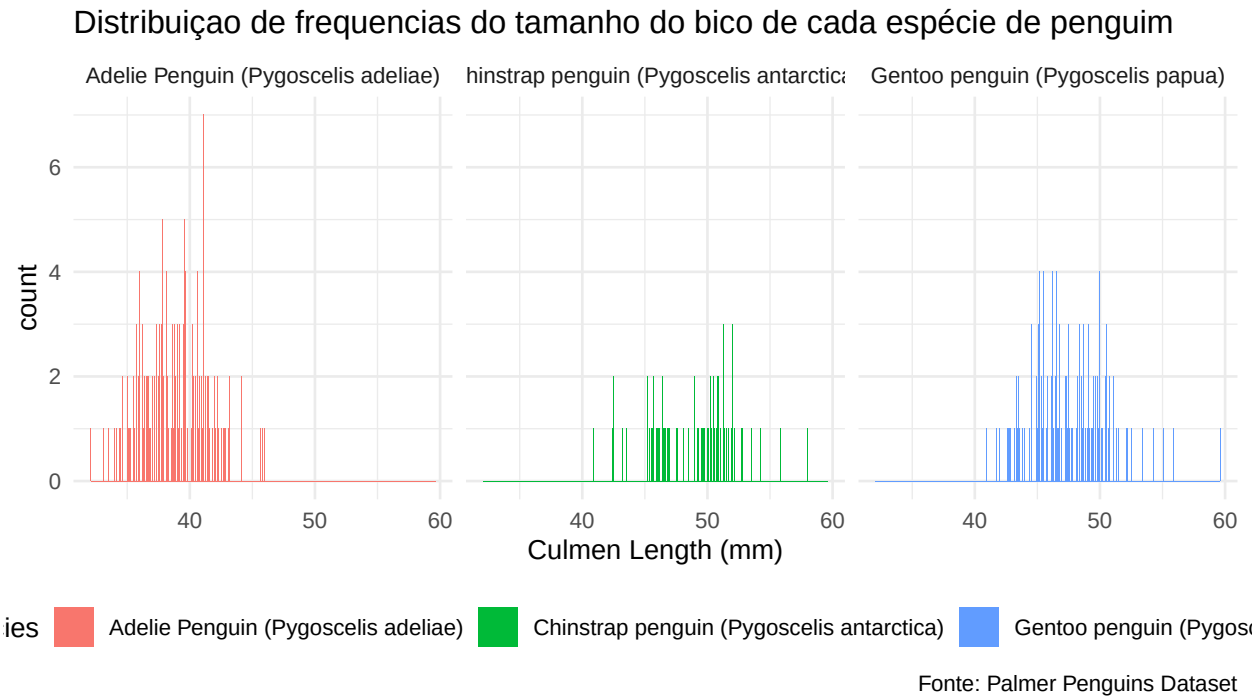
Abaixo dou um exemplo de situaçao na qual é tentada fazer uma série temporal com barras, contudo o periodo de tempo é curto:



Histogramas O desenho de histogramas é essencial na exploraçao de dados, pois ele é capaz de nos mostrar de forma simples a distribuição de um conjunto de dados, ou seja, quais os valores mais e menos frequentes. É comum que sejam confundidos com gráficos de barra, porém esse erro é crítico e não deve ser cometido, isso por que cada uma das geometrias irá exibir determinada informação e que há de ser interpretada de formas diferentes.



A leitura de um histograma parece ser mais complicada do que realmente é, neste tipo de geometria as barras mais elevadas correspondem aos valores que sao mais frequentes no banco de dados a ser visualizado. Nesta leitura, contudo, o leitor irá notar que nem todos os valores sao exibidos, esse fenomeno ocorre pois nesse tipo de gráfico os dados sao agrupados em classes (no ggplot2 estes se chamam: `bins`). O agrupamento ocorre pois caso contrário poderiamos ter cenários em que uma quantidade grande de valores está presente somente uma vez, o que impossibilita a visualizaçao, como podemos ver no exemplo abaixo...

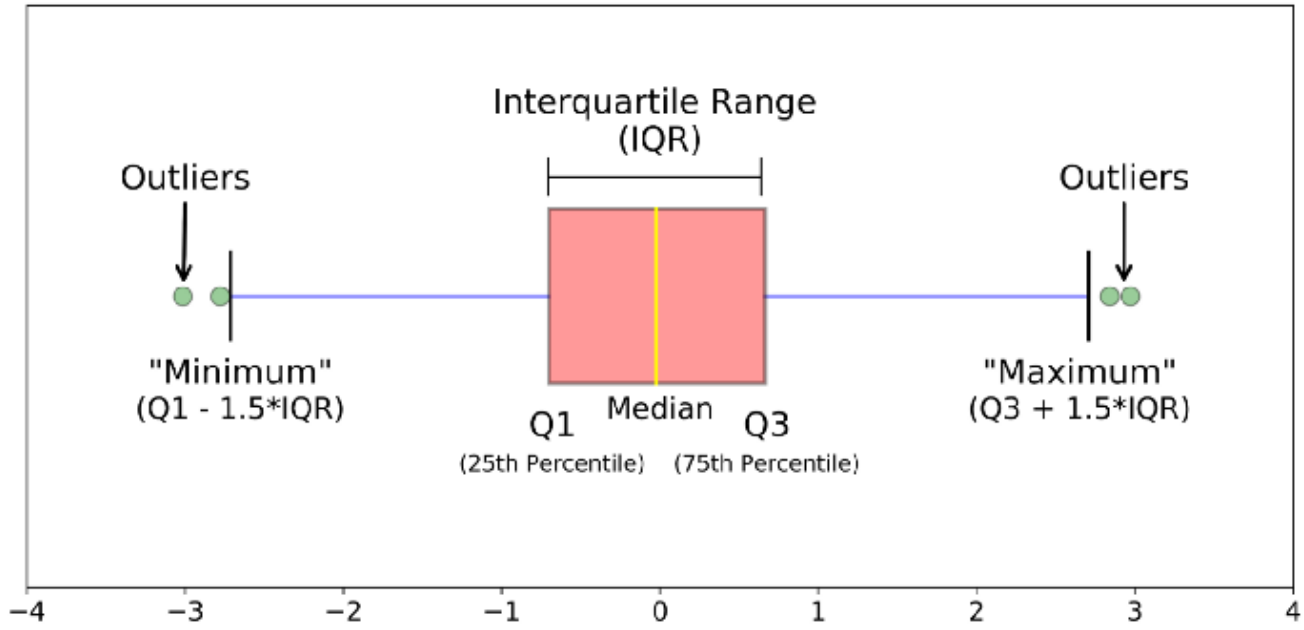


Fonte: Palmer Penguins Dataset

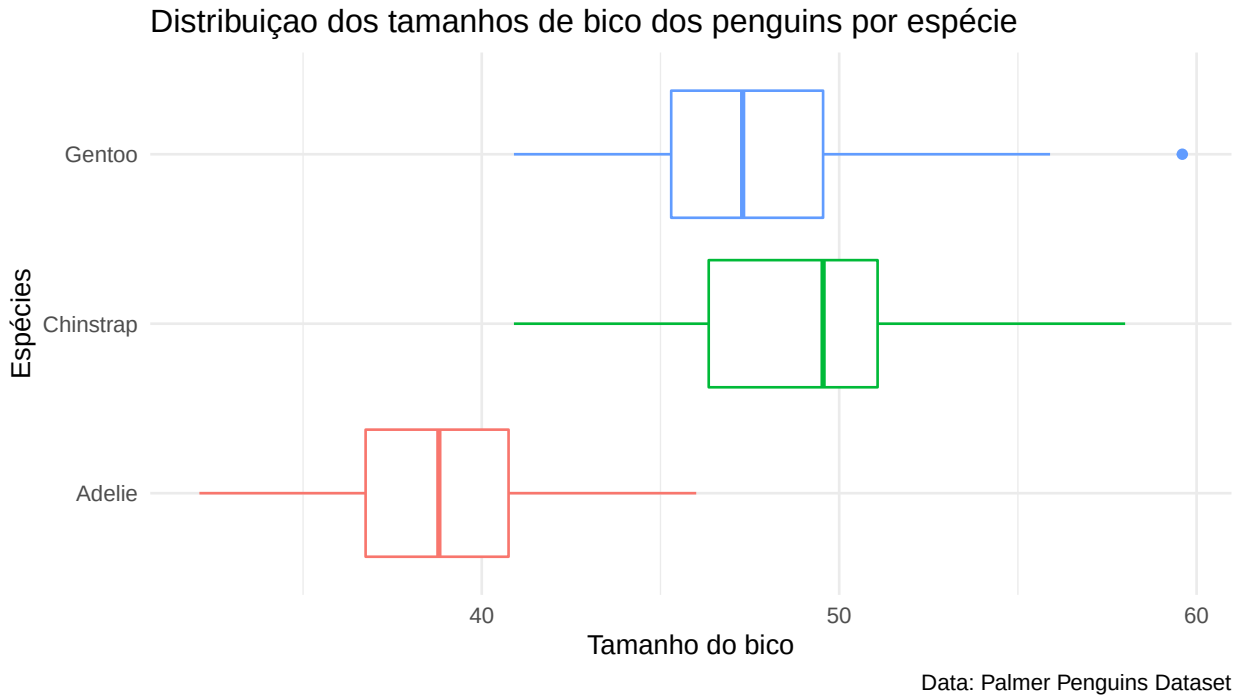
Mas como escolher o intervalo de classe ideal? A recomendaçao geral é que se mantenha um valor de 10 até 20, o que nao nos impede de ultrapassar caso achemos necessário. Como resultado dessa limitaçao há uma troca (ou trade-of) entre "perda de informaçao" (em intervalos de classe muito pequenos) e "extensao da distribuiçao" (no cenário com intervalos excessivos) feito pelo programador. Todavia, a regra 10-20 é um bom ponto de partida, e na maioria das vezes suficiente.

Entretanto, podemos nos ver tentados a ultrapassar o limite recomendado quando acreditamos conseguir extrair algum **insight** que está sendo suprimido pela limitaçao. Essa crença na possibilidade de extrair novas ideias do dataset muitas vezes está correta, e a experimentaçao de diferentes intervalos de classe é recomendada. Nesse momento, o bom senso se faz essencial.

Grafico de Caixa Gráficos de caixa (boxplot) tem sua principal utilidade quando queremos comparar a distribuiçao de valores entre grupos de um conjunto de dados.



Isso ocorre pois ele é capaz de resumir rapidamente estatísticas centrais para que possamos entender um banco de dados, veja a imagem acima. Na imagem, temos a explicaçao de como é feita a leitura de um boxplot. Ele apresenta os outliers, o valor mínimo, máximo, os percentis e a mediana das informaçoes. Para ele, utilizamos a geometria `geom_boxplot()`.

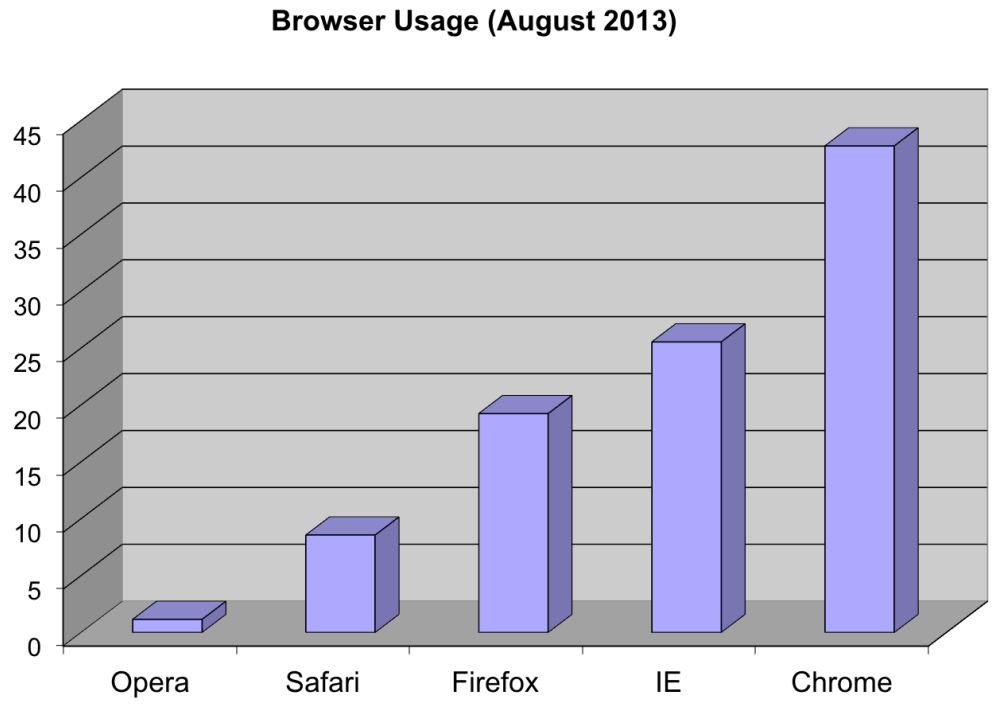


Boas práticas em visualização de dados.

Nessa seção eu escreverei, da forma mais concisa possível, dicas para que você possa se manter afastado de práticas consideradas ruins no mundo da visualização de dados. Noto porém que este não é um guia definitivo, nem uma escritura que deve sempre ser seguida pois é óbvio que existem exceções, ou momentos em que por força maior (normalmente seu chefe) você é impelido a executar um projeto de forma sub-ótima.

Não faça 3D

Muitas vezes voce estará utilizando uma ferramenta que irá lhe oferecer a tentadora opçao de fazer um gráfico 3D, evite-a. Embora fazer uma pizza virar um disco seja um truque de mágica muito interessante, quando estamos falando de gráficos a mágica não é tão boa assim. Na verdade, muitos argumentam que gráficos em 3-dimensoes são "inequivocadamente ruim, e devem ser removidos do vocabulário dos cientistas de dados" (Wilke, 2019). Essa repulsa por 3D existe pois este tipo de visualização é de dificil leitura e reduze a compreensão dos dados que estão sendo apresentados. Se a ideia dos gráficos é apresentar os dados de forma simples, não faz sentido apelar para tal artifício gráfico.



Na ilustraçao anterior temos um exemplo de qual tipo de visualização[1] evitar, a barra 3D temos um gráfico em que a profundidade aplicada pelo artificio visual prejudica a atribuiçao de uma porcentagem a um valor. Não faça.

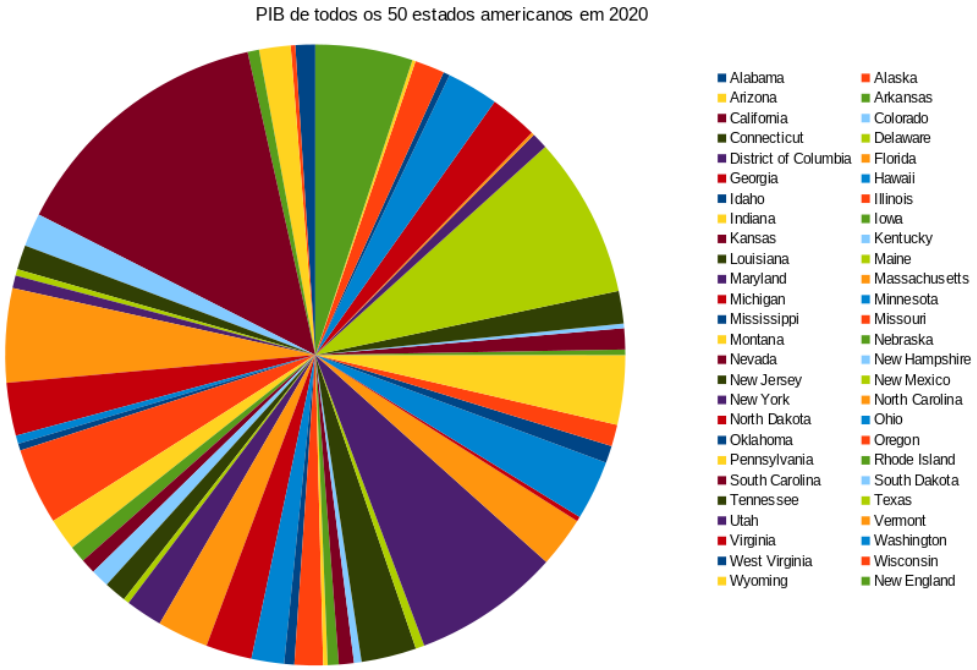
Evite Pizzas

Gráficos de Pizza são amados no mundo corporativo, todavia não deveriam ser. Na verdade, se voce já teve alguma conversa com um cientista de dados ou designer, é possivel que em algum momento essa pessoa já expressou o quando abomina gráficos de pizza. Naturalmente esse sentimento não é infundado, o gráfico de pizza é extremamente problemático, e isso ocorre por conta de um aspecto do nosso cérebro. Acontece

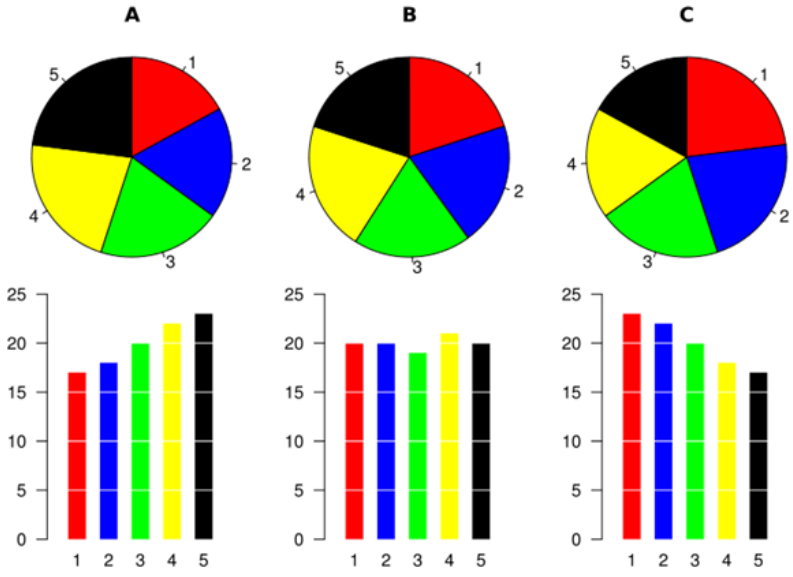
[1] Genomics Class

que a mente humana processa informação de forma linear, sendo assim essa característica facilita com que processemos dados dispostos em linhas e colunas, o que não abarca pizzas.

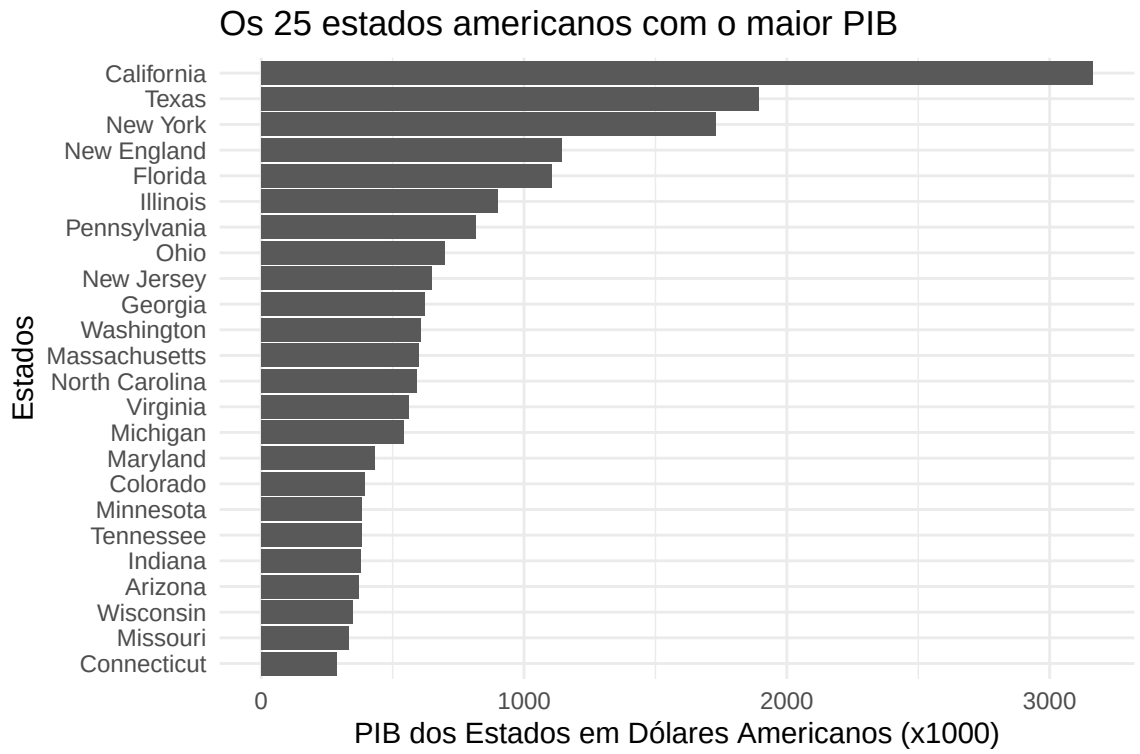
Podemos perceber as dificuldades que esse tipo de visualizaçao apresenta quando possuimos um gráfico com muitas fatias. Nesse tipo de situação a comparação entre os setores do gráfico se tornam extremamente dificeis de serem feitas a olho nu.



Na ilustraçao acima temos um caso de gráfico de pizza com 50 fatias, no caso observamos a dificuldade que nós possuimos em fazer comparaçoes entre os tamanhos e também identificar qual cor representa qual estado. Dessa forma conseguimos observar o quão problemático uma pizza pode se tornar mesmo com um numero pequeno de variáveis (50). Mas como resolver isso? O ideal normalmente é abandonarmos a tentação de comer uma deliciosa fatia de pizza e passar para geometrias menos apetitosas, como barras. As barras irão transmitir a mensagem ao leitor mais fácil e precisa a informação.



Nas imagens acima podemos observar como uma simples mudança de geometria facilita muito a compreensão de um mesmo conjunto de dados. Nesse exemplo acima vale prestar atençao na forma como o formato esférico de uma pizza dificulta a comparação entre valores muito próximos. Aplicando agora no mesmo banco de dados que gerou aquele gráfico horrível dos estados americanos, temos...



Em suma, mantenha suas pizzas somente no prato.

Eixos

Quando em gráficos não-circulares, como o de pizza, os eixos sao a principal guia ao leitor. Eles sao responsáveis por indicar qual é o valor que está sendo exibido no gráfico. Embora alguns possam pensar quando mais eixos melhor, nem sempre é o caso. Com o tempo percebeu-se que eixos devem seguir uma série de boas práticas:

- Na maioria dos casos gráficos devem possuir somente um eixo vertical.
- Na maioria dos casos, começar um gráfico com o eixo Y no zero é uma boa ideia. Todavia, essa regra pode ser ignorada caso de séries históricas em que comparaçoes com a base não são centrais para o problema.
- O espaçamento entre valores deve ser consistente.
- Quando necessário, use escala logaritmica.

Darei atenção especial aos dois primeiros citados, pois acredito que os dois últimos sao simples de serem resolvidos e também é fácil ao aluno perceber o motivo para que essas práticas sejam vistas como ruins.

Dois Eixos O problema com dois eixos é o mais comum dos que estao listados acima, ele ocorre quando o autor, querendo mostrar a relaçao entre duas variáveis, duplica o eixo Y (vertical) e coloca o valor de cada variável em um eixo diferente. Essa prática gera uma distorção visual, e por meio dela é possível enganar o leitor fazendo com que esse possa ser levado a acreditar que há uma relação entre as duas variáveis mesmo que esta não exista. O exemplo abaixo é capaz de ilustrar melhor o ponto.

Na imagem possuimos possuimos dois gráficos que apresentamos mesmos dados: A variação dos preços de imoveis na Espanha e na Alemanha, no periodo de 1990 até 2015. É possível perceber que os dois eixos no gráfico da esquerda geral uma distorção, é dada ao leitor a ideia de que há uma relação inversa entre os preços de imóveis nos países, e também de que há uma grande volatilidade nos preços dos imóveis. Olhando na imagem que está em sua direita, é possível observar que a realidade não é bem assim.

Fixando algumas ideias

1. O Banco Mundial possui uma grande quantidade de estatísticas, uma delas que pode ser interessante para a nossa realidade é a de código "EG.ELC.FOSL.ZS", este indicator mostra a porcentagem da produçao de eletricidade de um país por meio de fontes alimentadas por petróleo, gás e carvão. Gere uma visualizaçao para este indicador exibindo os dados de Brasil e México de 1972 até 2016. Atribua ao gráfico um titulo, nomeia corretamente seus eixos e sua legenda.

Dica: a funçao wbstats::wb_data() possui um argumento chamado `country` que permite filtrar previamente os países que voce deseja.

2. O que acontece quando voce coloca duas geometrias em um só gráfico? Volte ao nosso dataset Palmer Penguins e tente fazer isso! Declare dois geoms em uma mesma função ggplot. (dica: boxplot + dotplot sao uma bela combinação)

Material complementar

RStudio Cheatsheet em Português (2016): https://rstudio.com/wp-content/uploads/2016/03/ggplot2-cheatsheet-portuguese.pdf

Fundamentals of Data Visualization: https://clauswilke.com/dataviz/introduction.html

The Visual Display of Quantitative Information - Edward Tufte.

ggplot2: Elegant Graphics for Data Analysis (Wickham, 2009)

RMarkdown: The Definitive Guide: https://bookdown.org/yihui/rmarkdown/

Agradecimentos: Obrigado Laura Emerim, Jamil Civitarese e Leonardo Monastério pela revisao, dicas e gráficos horríveis.