

Introdução ao ggplot2

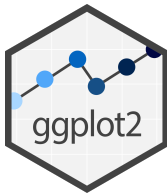
Isadora Real

2025-07-13

Introdução

O `ggplot2` é um pacote de código aberto para a visualização gráfica de dados para a linguagem de programação R. Foi criada por Hadley Wickham em 2005 (Wickham 2016), sendo uma implementação do livro *Grammar Graphics* de Leland Wilkison também lançado em 2005 (Wilkinson 2011).

Ele aborda que visualização gráfica dos dados pode ser dividida em componentes semânticos, como escalas e camadas.



Por que usar o ggplot2?

1. Alta customização gráfica.
2. Alta diversidade de modelos de gráficos.
3. Integração com outros pacotes do tidyverse, como por exemplo `dplyr` (Wickham et al. 2023), `forcats` (Wickham 2023) e o `plotly` (Sievert 2020).
4. Criação de gráficos a partir de camadas, podendo sobrepor diferentes gráficos.



Como instalar o ggplot2?

```
#instalando pacote ggplot2  
install.packages("ggplot2")
```

```
#instalando dplyr, forcats e patchwork  
install.packages("dplyr")  
install.packages("forcats")  
install.packages("patchwork")
```

Para usar o ggplot2 em seus scripts tem que carregá-lo

#Carregando o pacote ggplot2

```
library(ggplot2)
```

#Carregando dplyr, forcats e patchwork

```
library(dplyr)
```

```
library(forcats)
```

```
library(patchwork)
```

Banco de dados *iris*

Para essa oficina será utilizado bancos de dados **iris**.

iris - é referente tamanho de pételas e sepalas de 3 espécies do gênero *Iris* do trabalho de Fisher em 1936 (*Iris setosa*, *Iris versicolor* e *Iris virginica*)

`data(iris)`

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|--------------|-------------|--------------|-------------|---------|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |



Box-plot

```
iris%>%ggplot(aes(x=Species, y=Petal.Length))+  
  geom_boxplot()
```

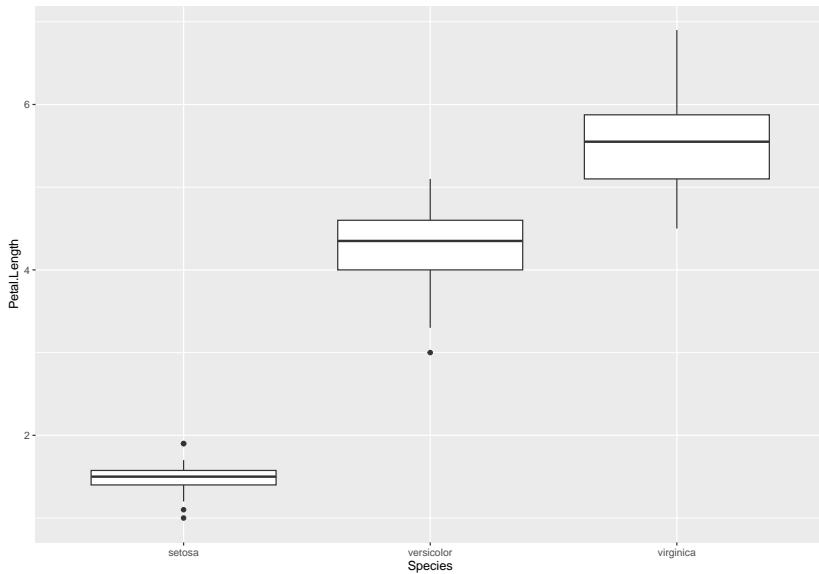
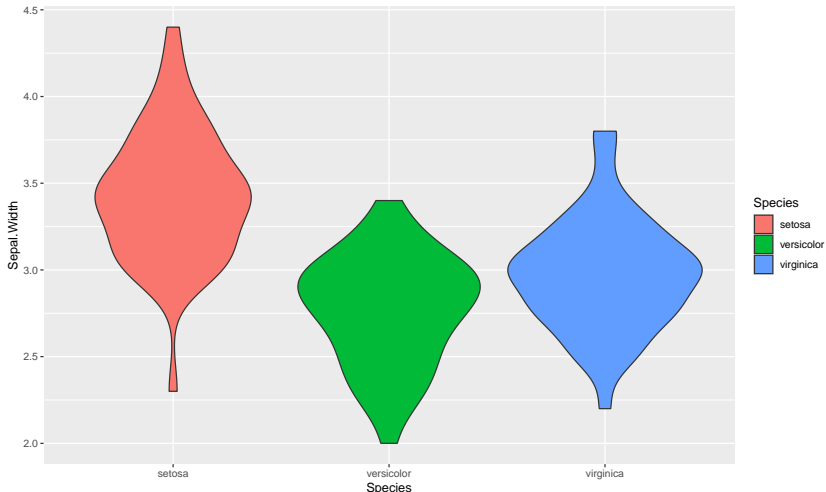


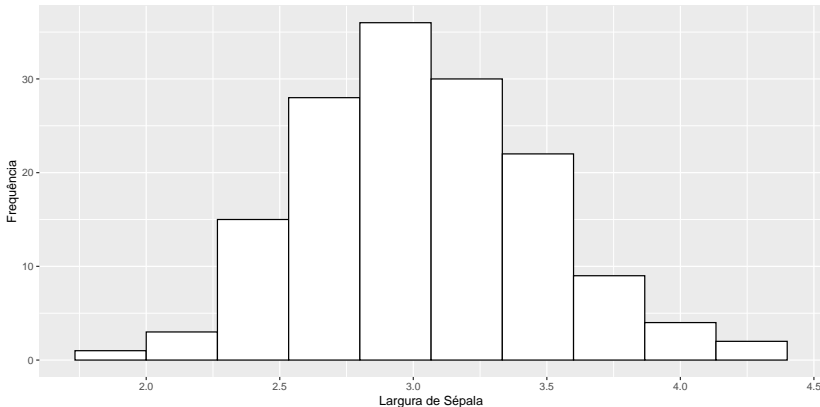
Gráfico violino

```
ggplot(iris, aes(x=Species,y=Sepal.Width, fill=Species))+  
  geom_violin()
```

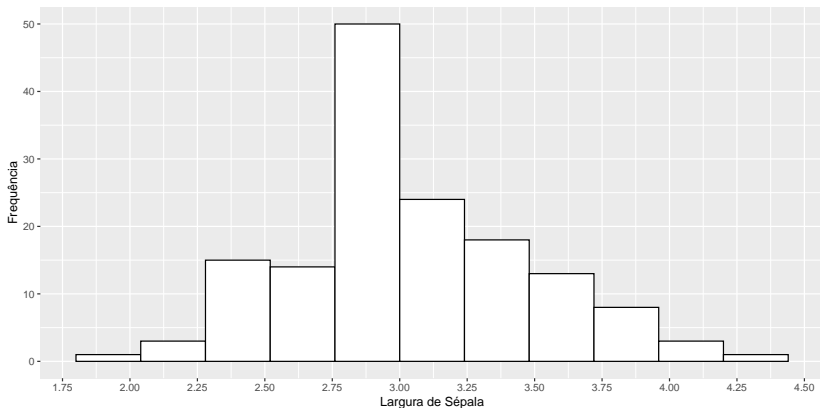


Histograma

```
ggplot(iris,aes(x=Sepal.Width))+  
  geom_histogram(bins=10, color="black",  
                 fill="white")+  
  labs(y="Frequência", x="Largura de Sépala")
```

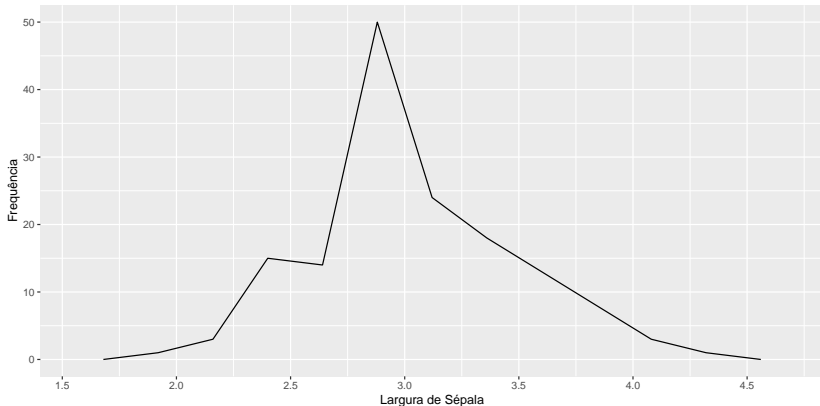


```
ggplot(iris,aes(x=Sepal.Width))+  
  geom_histogram(bins=11, color="black",  
                fill="white")+  
  labs(y="Frequência", x="Largura de Sépala")+  
  scale_x_continuous(n.breaks = 11)
```

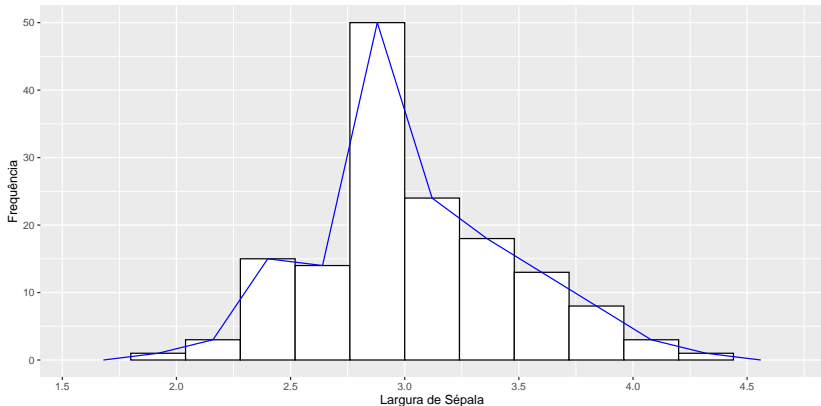


Polígono

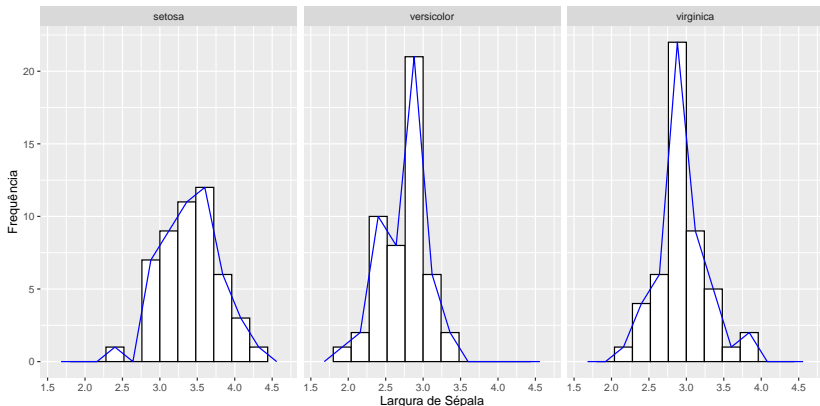
```
ggplot(iris,aes(x=Sepal.Width))+  
  geom_freqpoly(bins=11, color="black")+  
  labs(y="Frequência", x="Largura de Sépala")+  
  scale_x_continuous(n.breaks = 11)
```



```
ggplot(iris,aes(x=Sepal.Width))+  
  labs(y="Frequência", x="Largura de Sépala")+  
  scale_x_continuous(n.breaks = 11)+  
  geom_histogram(bins=11, color="black",  
                fill="white")+  
  geom_freqpoly(bins=11, color="blue")
```



```
ggplot(iris,aes(x=Sepal.Width))+  
  labs(y="Frequência", x="Largura de Sépala")+  
  scale_x_continuous(n.breaks = 11)+  
  geom_histogram(bins=11, color="black",  
                 fill="white")+  
  geom_freqpoly(bins=11, color="blue")+  
  facet_grid(~Species)
```



```
ggplot(iris,aes(x=Sepal.Width))+  
  labs(y="Frequência", x="Largura de Sépala")+  
  scale_x_continuous(n.breaks = 11)+  
  geom_histogram(bins=11, color="black",  
                fill="white")+  
  geom_freqpoly(bins=11, color="blue")+  
  facet_grid(Species~.)
```

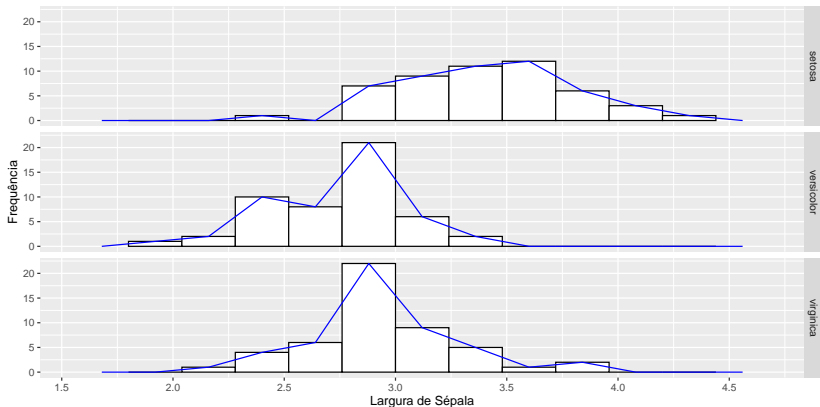


Gráfico de densidade

```
ggplot(iris,aes(x=Sepal.Width))+  
  geom_density(color="black", fill="white")+  
  labs(y="Frequência", x="Largura de Sépala")
```

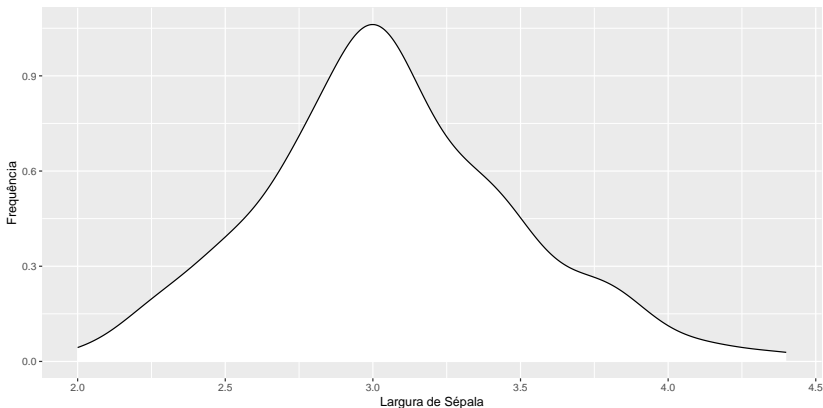
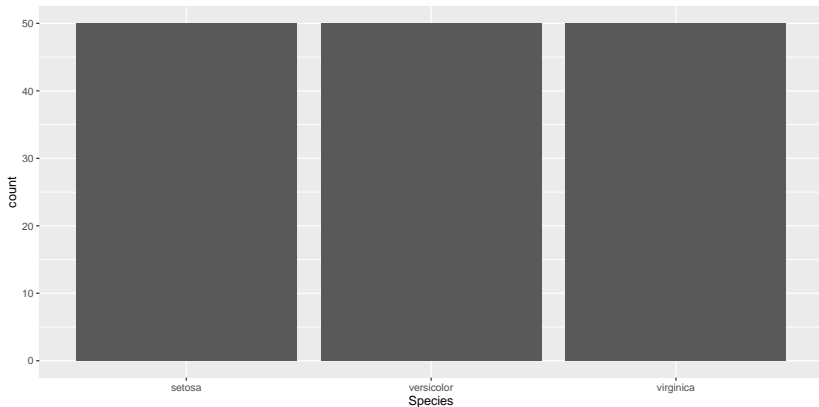


Gráfico de barras de frequência

```
iris%>%ggplot(aes(x=Species))+  
  geom_bar()
```



```
iris%>%group_by(Species)%>%  
  summarise(count=n())%>%  
  ggplot(aes(x=Species, fill=Species, y=count))+  
  geom_col(color="black")
```

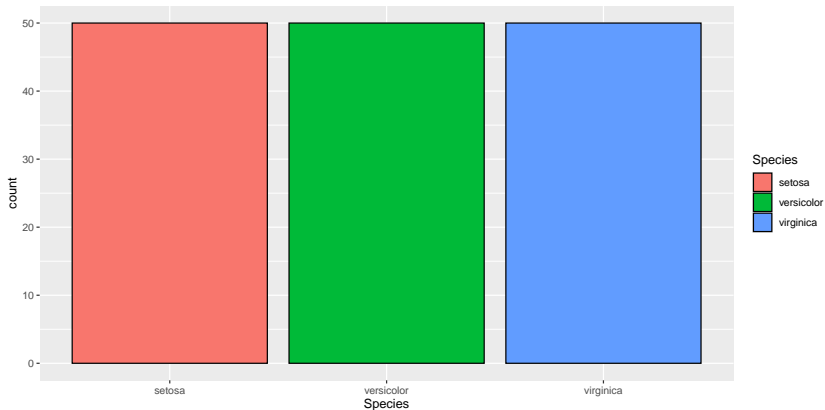
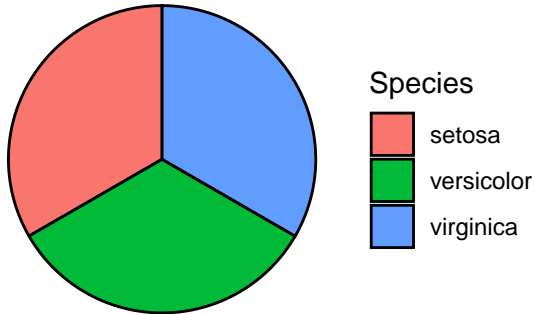


Gráfico de pizza

```
iris%>%group_by(Species)%>%  
  summarise(count=n()/150*100)%>%  
  ggplot(aes(x=" ", fill=Species, y=count))+  
  geom_col(color="black")+  
  coord_polar(theta="y")+  
  theme_void()
```



```
iris%>%group_by(Species)%>%  
  summarise(count=round(n()/150*100, 2))%>%  
  ggplot(aes(x=" ", fill=Species, y=count))+  
  geom_col(color="black")+  
  coord_polar(theta="y")+  
  geom_label(aes(label = count),  
            position = position_stack(vjust = 0.5),  
            show.legend = FALSE)+  
  theme_void()
```

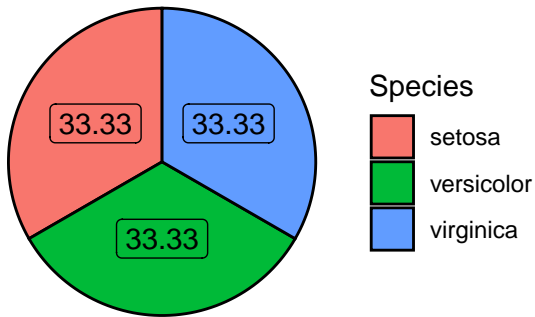
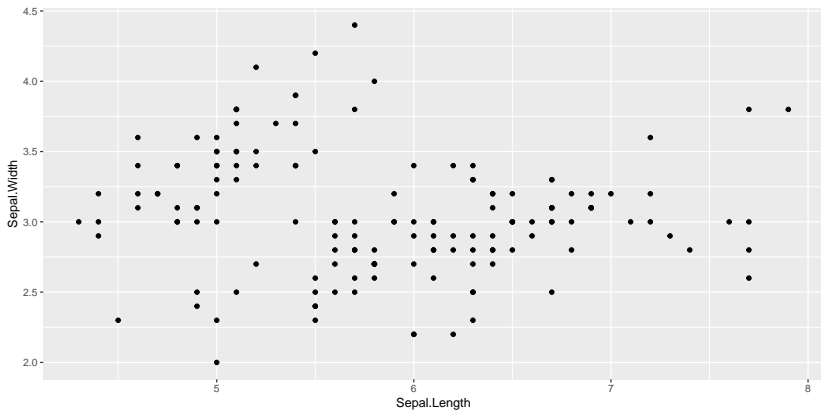
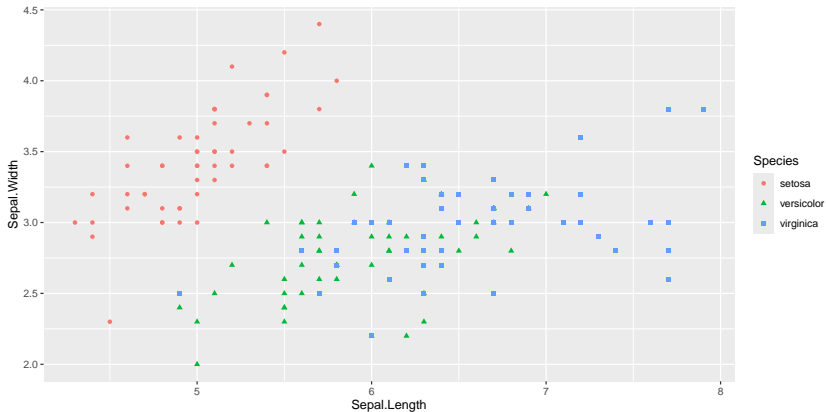


Gráfico de pontos

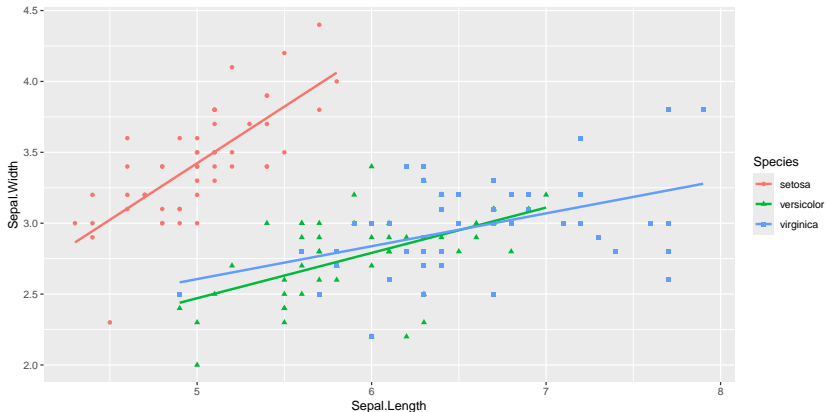
```
ggplot(iris,aes(x=Sepal.Length, y=Sepal.Width))+  
  geom_point()
```



```
ggplot(iris,aes(x=Sepal.Length, y=Sepal.Width,  
               color=Species, shape=Species))+  
  geom_point()
```



```
ggplot(iris,aes(x=Sepal.Length, y=Sepal.Width,  
               color=Species, shape=Species))+  
  geom_point()+  
  geom_smooth(se=FALSE, method="lm")
```



```
ggplot(iris,aes(x=Sepal.Length, y=Sepal.Width, color=Species,  
               shape=Species))+  
  geom_point()+  
  geom_smooth(se=FALSE, method="lm")+  
  coord_flip()
```

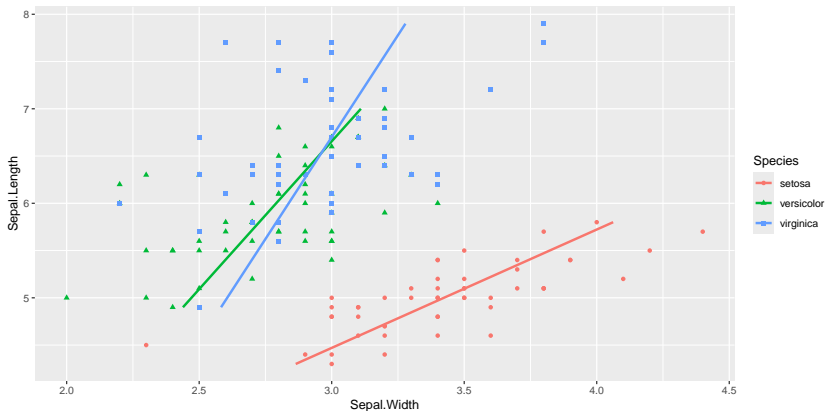
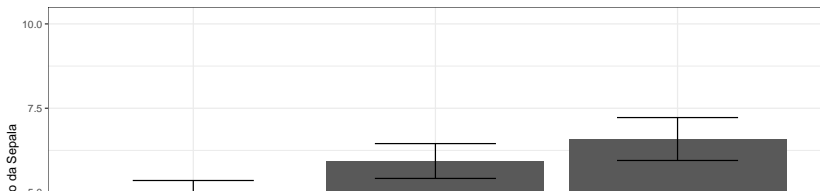
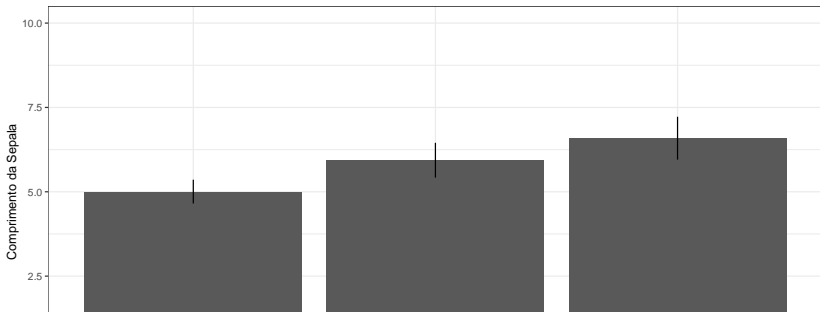


Gráfico de barras (média e desvio)

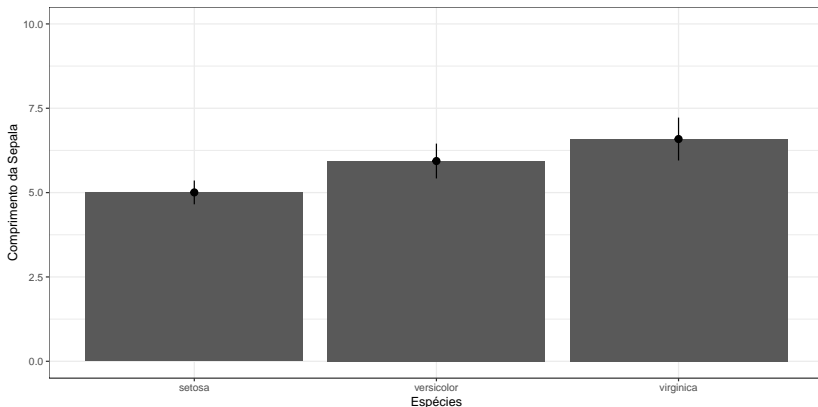
```
iris%>%group_by(Species)%>%
  summarise(mean=mean(Sepal.Length),
            sd=sd(Sepal.Length),
            se=sd(Sepal.Length)/sqrt(length(Sepal.Length)))
ggplot(aes(x=Species, y=mean))+
  geom_col()+
  geom_errorbar(aes(ymin=mean-sd,ymax=mean+sd), width=0.5)+
  labs(y="Comprimento da Sepala", x="Espécies")+
  theme_bw()+
  scale_y_continuous(limits=c(0,10))
```



```
iris%>%group_by(Species)%>%  
  summarise(mean=mean(Sepal.Length),  
            sd=sd(Sepal.Length),  
            se=sd(Sepal.Length)/sqrt(length(Sepal.Length)))+  
  ggplot(aes(x=Species, y=mean))+  
  geom_col()+  
  geom_linerange(aes(ymin=mean-sd,ymax=mean+sd))+  
  labs(y="Comprimento da Sepala", x="Espécies")+  
  theme_bw()+  
  scale_y_continuous(limits=c(0,10))
```



```
iris%>%group_by(Species)%>%  
  summarise(mean=mean(Sepal.Length), sd=sd(Sepal.Length), se=sd(Sepal.Length)/sqrt(n()))
```



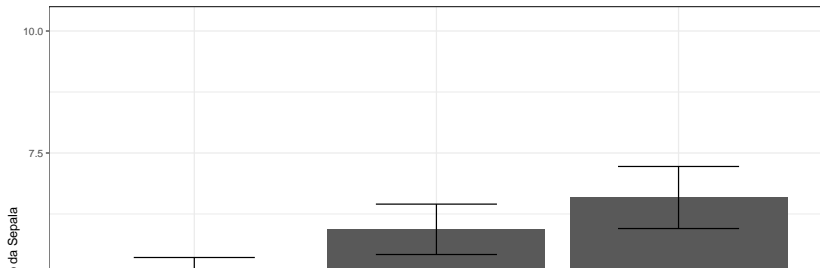
O exemplo abaixo é sem a coluna, apenas o geom_linerange.



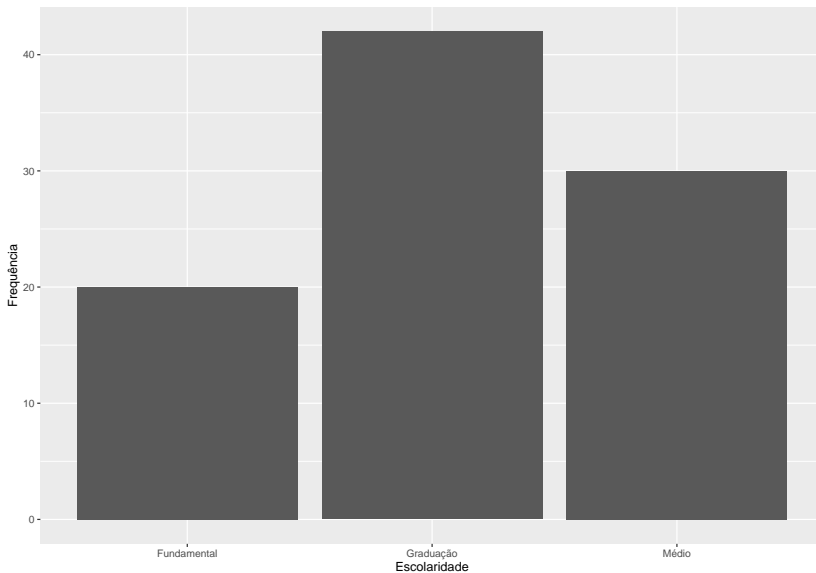
Alterando escalas, cores, fontes e temas

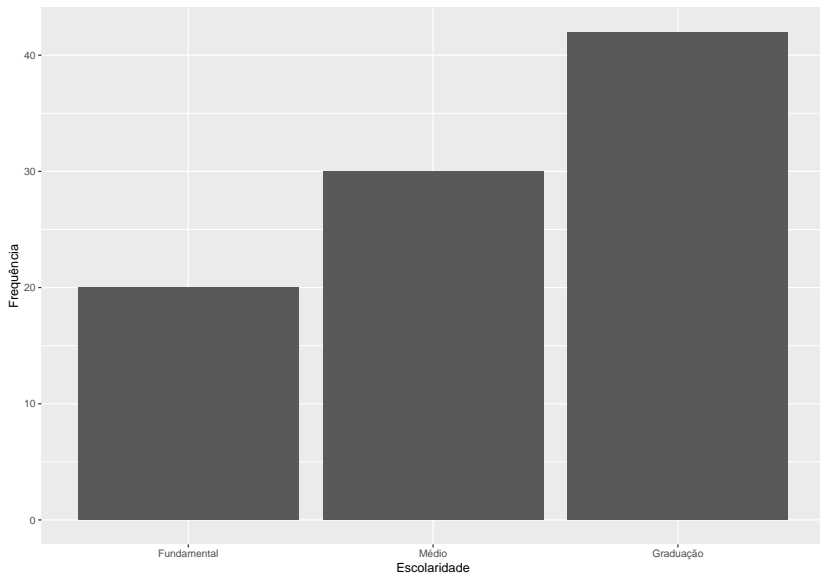
Ajustando escalas no ggplot

Retornamos com o gráfico de barra com a média e as barras de desvio. Para ajustar a escala de algum elemento gráfico no ggplot o comando inicia `scale_[parâmetro]_[característica do parâmetro ou tipo de função aplicada]()`. No caso abaixo foi utilizado o `scale_y_continuous()`, já que ajustaremos a escala do eixo y, sendo ela uma variável contínua. Dentro da função colocaremos o seguinte argumento `limits=c(0,10)`, onde zero é o limite inferior e o 10 o limite superior.



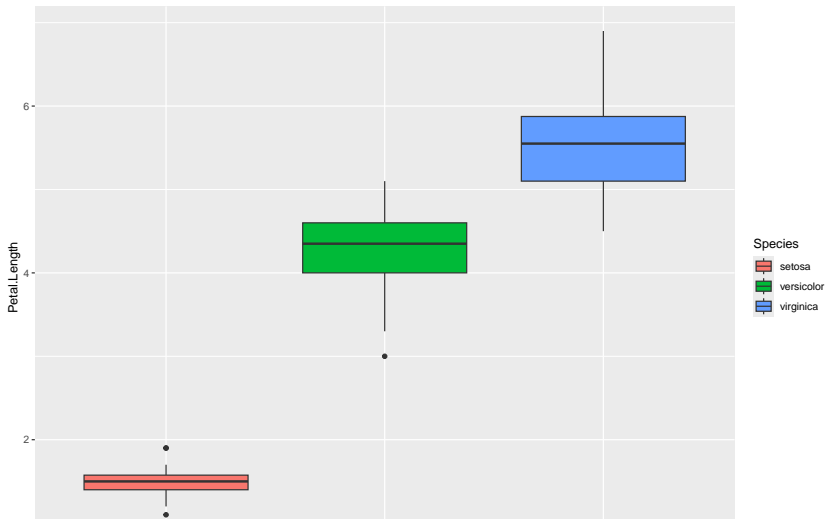
Ordenando variáveis ordinais no ggplot



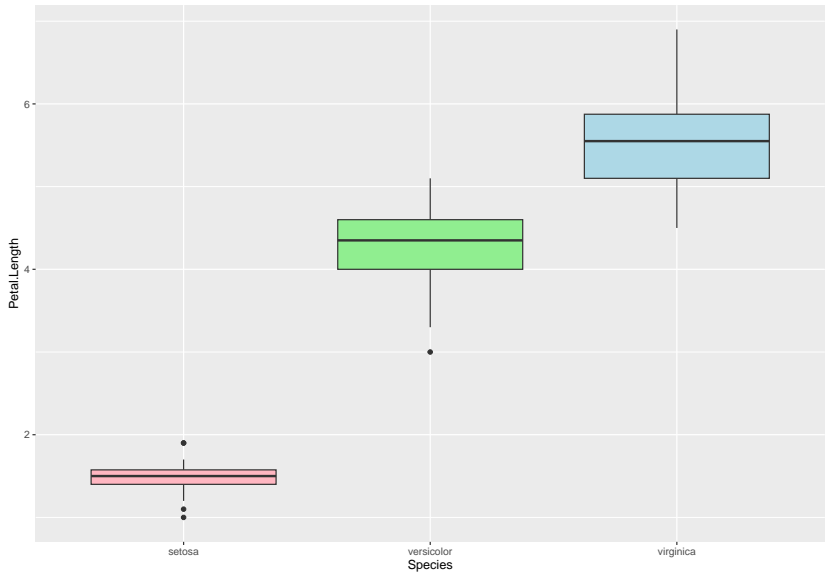


Mudando cores de preenchimento no ggplot

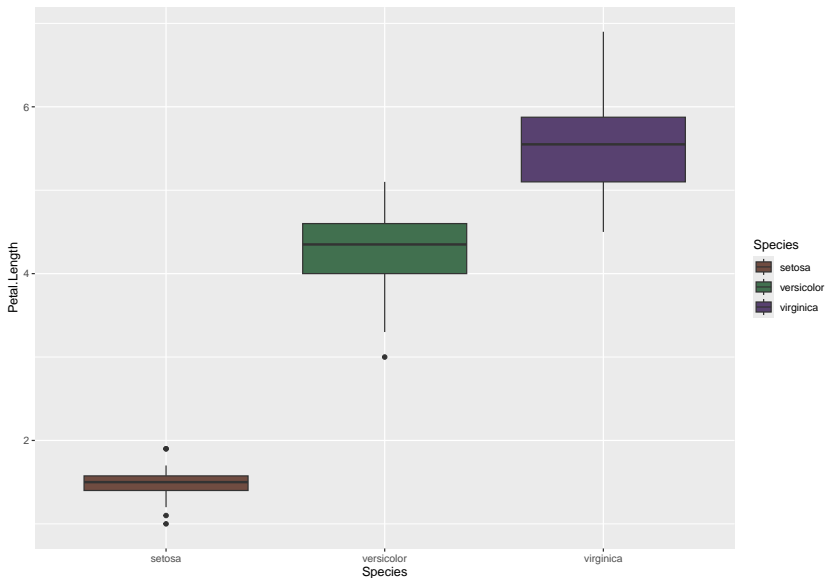
```
iris%>%ggplot(aes(x=Species, y=Petal.Length, fill=Species))
```




```
iris%>%ggplot(aes(x=Species, y=Petal.Length))+geom_boxplot
```

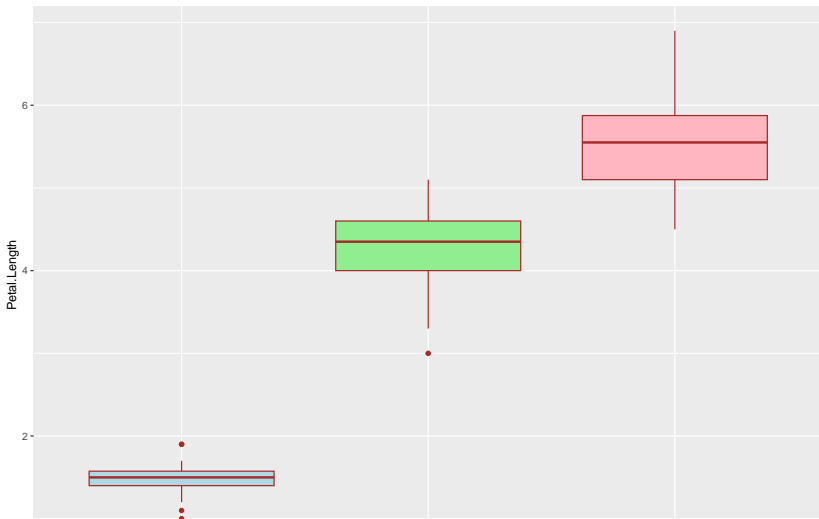


```
iris%>%ggplot(aes(x=Species, y=Petal.Length, fill=Species))
```



Mudando cores de contorno no ggplot

```
iris%>%ggplot(aes(x=Species, y=Petal.Length, fill=Species))
```



Alterando elementos textuais no ggplot

Os nomes dos eixos são alterados pela função `labs`, onde você indica qual elemento gráfico você quer renomear. Lembre-se: o nome que você quer renomear tem que estar entre aspas " ".

y para alterar o título do eixo y.

x para alterar o título do eixo x.

title para alterar o título ou acrescentar um título.

subtitle para alterar o subtítulo ou acrescentar um subtítulo.

fill para alterar o título da legenda referente ao fator colocado no fill.

color para alterar o título da legenda referente ao fator colocado no color.

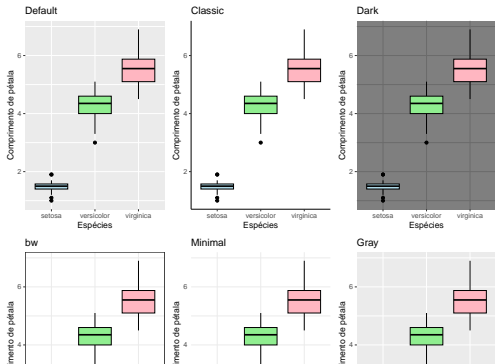
shape para alterar o título da legenda referente ao fator colocado no shape.

size para alterar o título da legenda referente ao fator colocado no size.

Temas

O ggplot2 por padrão dispõe de alguns temas para utilizar nos gráficos, adicionando `theme_`, aparece as sugestões disponíveis do pacote. Através do comando `theme()`, pode-se editar o tema ou criar temas novos.

Abaixo está os temas disponíveis do pacote ggplot2.

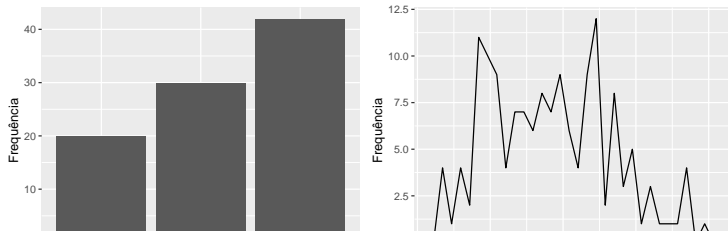


Unindo vários gráficos em uma imagem só

Unindo vários gráficos em uma imagem só

Em na seção do histograma vimos a função `facet_grid` ou `facet_wrap`, para ter gráficos separados de acordo a uma característica do *data frame*. Porém como colocamos ggplots lado a lado de diferentes *data frame* para uma única imagem? Há algumas opções, aqui irei mostrar utilizando o pacote `patchwork`,

1. Armazenar os ggplots em um objeto utilizando o comando `<-` ou `=`, como mostra o exemplo abaixo
2. Escrever os nomes do objetos colocar `+` entre eles



Extra

Mapas

```
#instalando o pacote raster e sf  
install.packages("raster")  
install.packages("sf")
```

```
#carregando o pacote raster e sf  
library(raster)  
library(sf)
```

```
# Importando dados
```

```
prec<-raster("pelprec.tiff")
```

```
pel<-read_sf("Pelotas/Pelotas.shp")
```

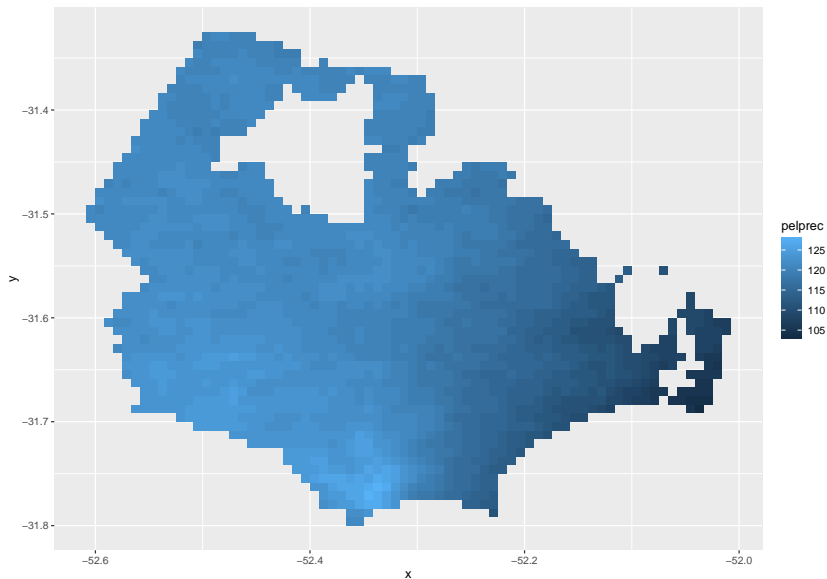
```
# Convertendo raster para data frame para o ggplot processar
```

```
prec_df<-as.data.frame(prec, xy = TRUE, na.rm = TRUE)
```

```
head(prec_df)
```

| | x | y | pelprec |
|----|-----------|-----------|---------|
| 14 | -52.49583 | -31.32917 | 120 |
| 15 | -52.48750 | -31.32917 | 121 |
| 16 | -52.47917 | -31.32917 | 121 |
| 17 | -52.47083 | -31.32917 | 120 |
| 18 | -52.46250 | -31.32917 | 120 |
| 19 | -52.45417 | -31.32917 | 120 |

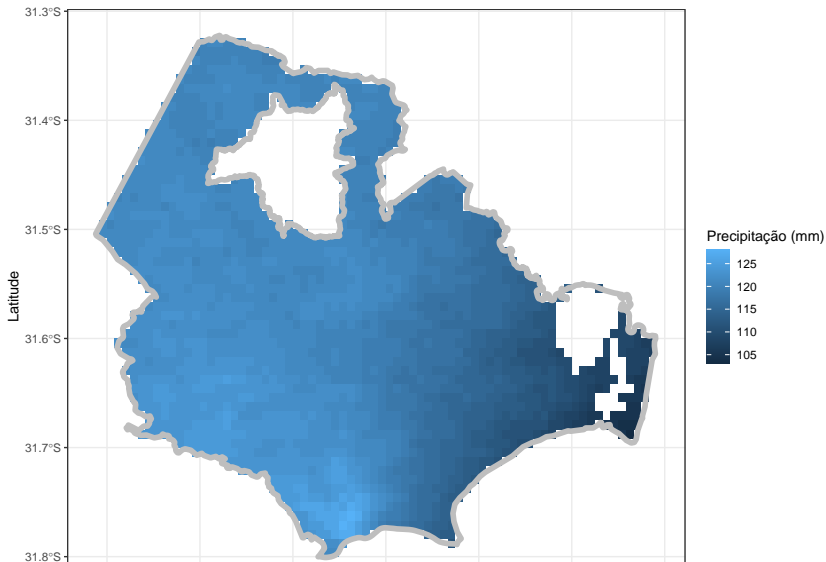
```
ggplot(prec_df,aes(x=x,y=y,fill=pelprec))+geom_raster()
```



```
# Cores padrão
```

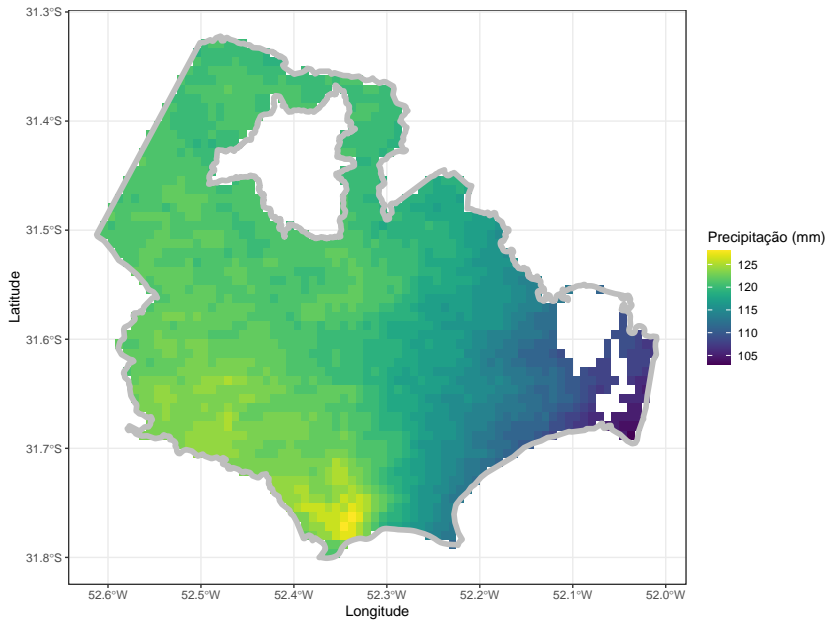
```
ggplot()+geom_raster(data=prec_df,aes(x=x,y=y,fill=pelprec))
```

Mapa da média anual da precipitação
em Pelotas-RS entre 1970–2000



```
#instalando pacote viridis  
install.packages("viridis")  
#carregando pacote viridis  
library(viridis)
```

Mapa da média anual da precipitação
em Pelotas-RS entre 1970–2000



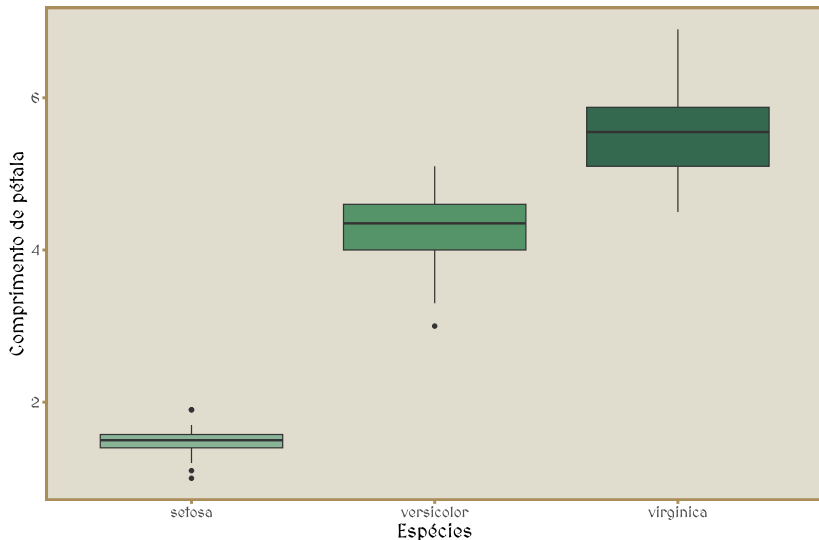


Temas divertidos

```
install.packages("remotes")  
remotes::install_github("MatthewBJane/ThemePark")  
library(ThemePark)
```

```
iris%>%ggplot(aes(x=Species, y=Petal.Length, fill=Species))
```

Tema Senhor dos Anéis



```
iris%>%ggplot(aes(x=Species, y=Petal.Length, fill=Species))
```


Referências

- Sievert, Carson. 2020. *Interactive Web-Based Data Visualization with r, Plotly, and Shiny*. Chapman; Hall/CRC.
<https://plotly-r.com>.
- Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.
<https://ggplot2.tidyverse.org>.
- . 2023. *Forcats: Tools for Working with Categorical Variables (Factors)*. <https://forcats.tidyverse.org/>.
- Wickham, Hadley, Romain François, Lionel Henry, Kirill Müller, and Davis Vaughan. 2023. *Dplyr: A Grammar of Data Manipulation*. <https://dplyr.tidyverse.org>.
- Wilkinson, Leland. 2011. “The Grammar of Graphics.” In *Handbook of Computational Statistics: Concepts and Methods*, 375–414. Springer.