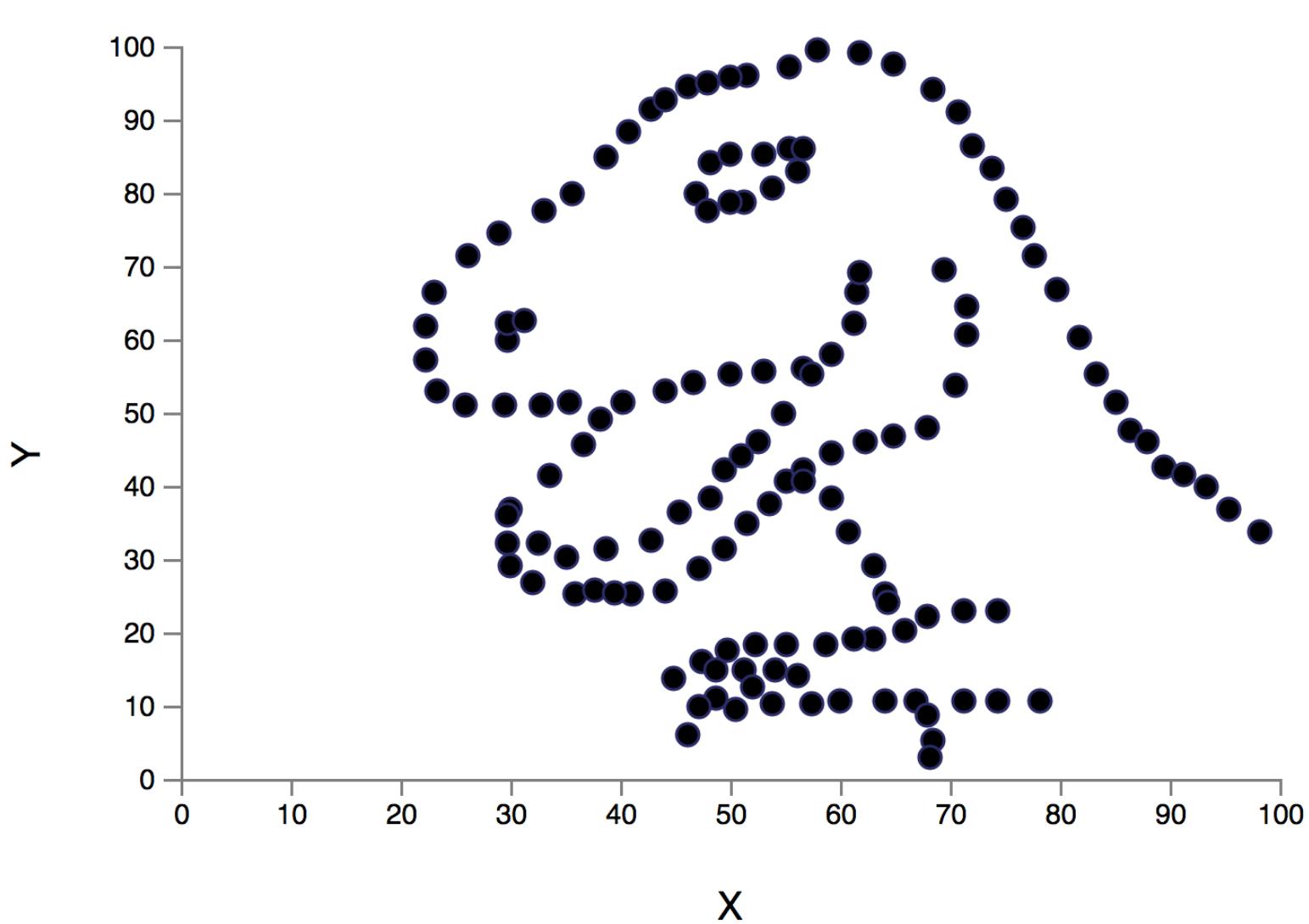


Introdução ao tidyverse

4 Visualização de dados

xaringan [presentation ninja]

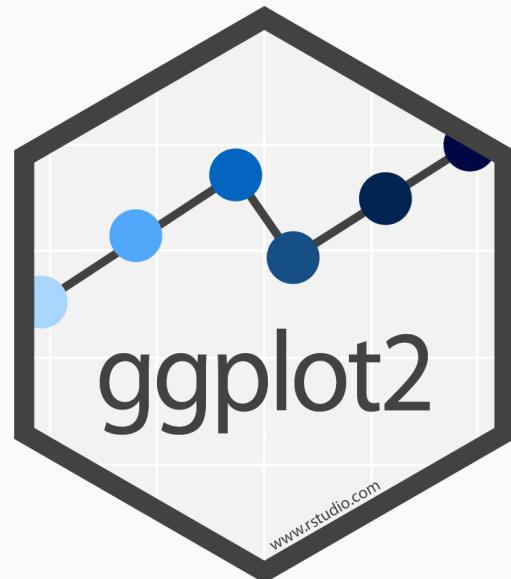
Maurício Vancine
25/04/2019



4 Visualização de dados

Conteúdo

- 4.1 Principais pacotes para gráficos
- 4.2 Principais livros e sites
- 4.3 Colunas como eixos do gráficos (*variáveis = colunas*)
- 4.4 Principais tipos de gráficos
- 4.5 Histograma (*histogram*)
- 4.6 Gráfico de setores (*pie chart* e *donut plot*)
- 4.7 Gráfico de barras (*bar plot*)
- 4.8 Gráfico de caixa (*box plot* e *violin plot*)
- 4.9 Gráfico de dispersão (*scatter plot*)
- 4.10 Gráfico pareado (*pairs plot*)



4 Visualização de dados

Script

```
script_aula_04.R
```

4 Visualização de dados

Importância dos gráficos

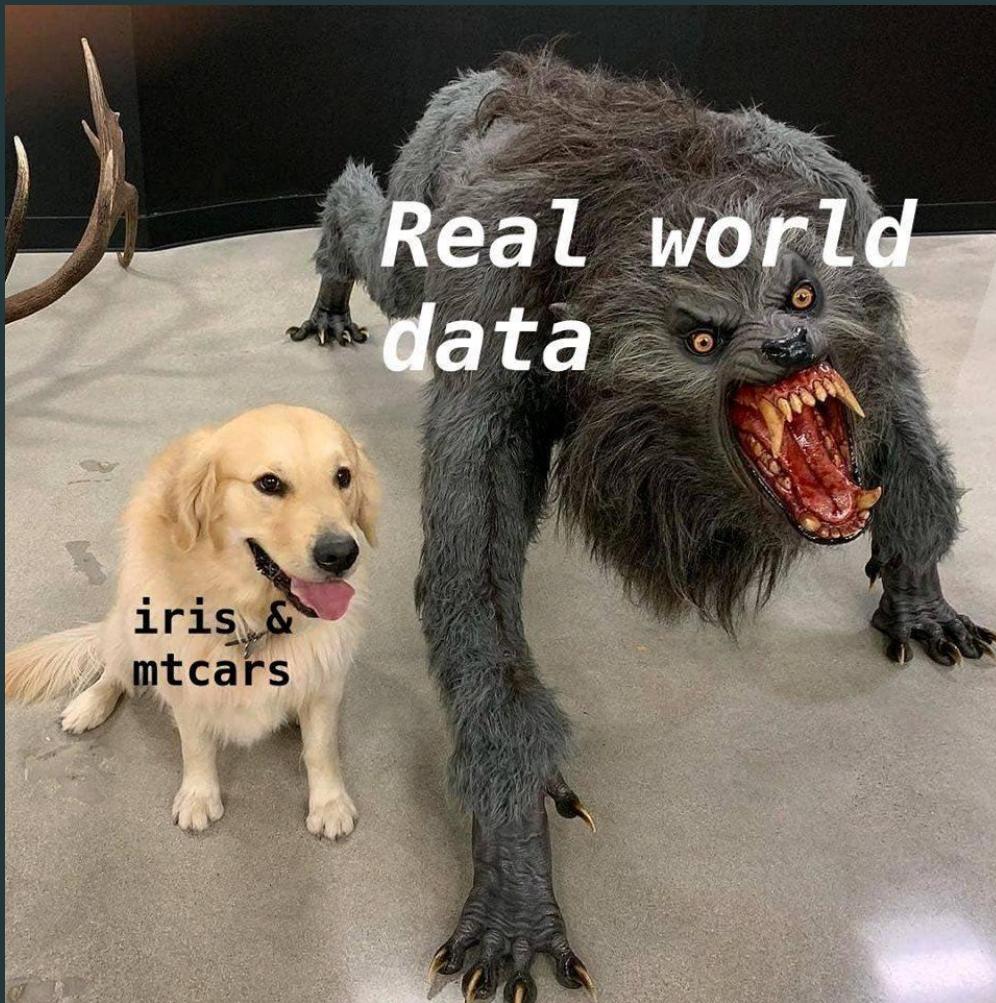
Melhor forma de **apresentar e discutir** seus dados

Faz uma **síntese** para melhor entendimento

Necessário em quase todas as **análises estatísticas**

Necessário em quase todas as **publicações**, trabalhos de consultoria,
TCC, dissertação, tese

E que dados vamos usar?

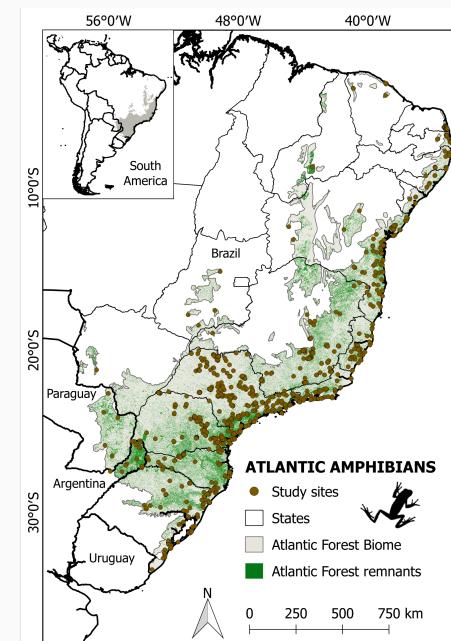
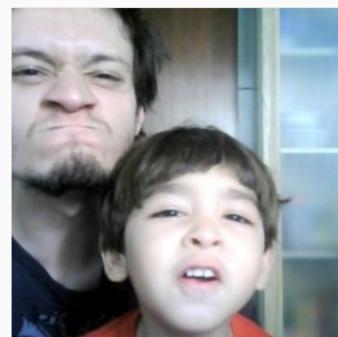
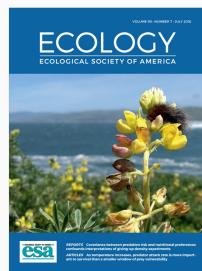


Vamos importar os dados do data paper de
anfíbios

Importar dados para gráficos

ATLANTIC AMPHIBIANS: a dataset of amphibian communities from the Atlantic Forests of South America

Eu mesmo et al. (2018)



Importar dados para gráficos

```
# package
library(tidyverse)

# directory
setwd("/home/mude/data/github/minicurso-tidyverse/03_dados")

# importar
da <- readr::read_csv("ATLANTIC_AMPHIBIANS_sites.csv")
da

# verificar
dplyr::glimpse(da)
```

4.1 Principais pacotes para gráficos

Atualmente, há **três principais pacotes** para gerar **gráficos gerais** no R:

1 graphics: simples, porém útil para visualizações rápidas de quase todos os formatos de arquivos

`plot()`

2 ggplot2: complexos, demandam mais tempo para realização, mas ficam muito bons

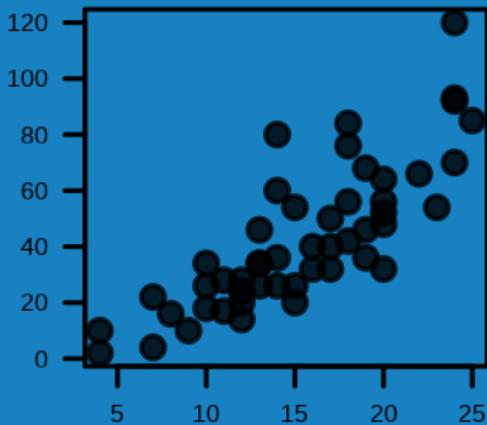
`ggplot()`

3 ggpubr: fornece algumas funções simplificadas para criar gráficos para publicação, baseados no

"`ggplot2`"

`gg*()`

graphics



4.1 Principais pacotes para gráficos

graphics

Default do R e mais simples

Pode ser utilizado para objetos de **diversas classes**

Possui funções como:

plot()

hist()

barplot()

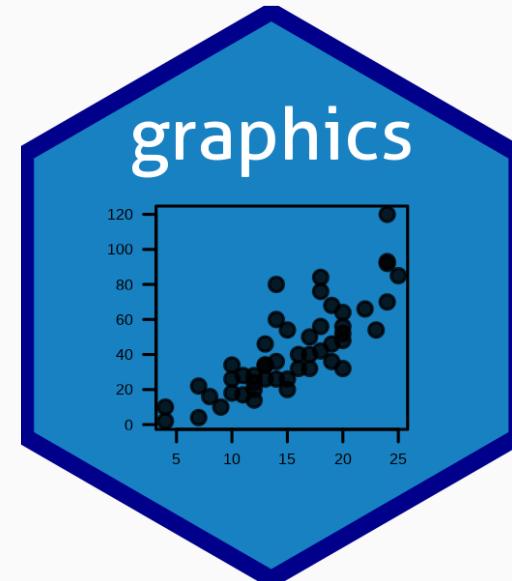
boxplot()

abline()

points()

lines()

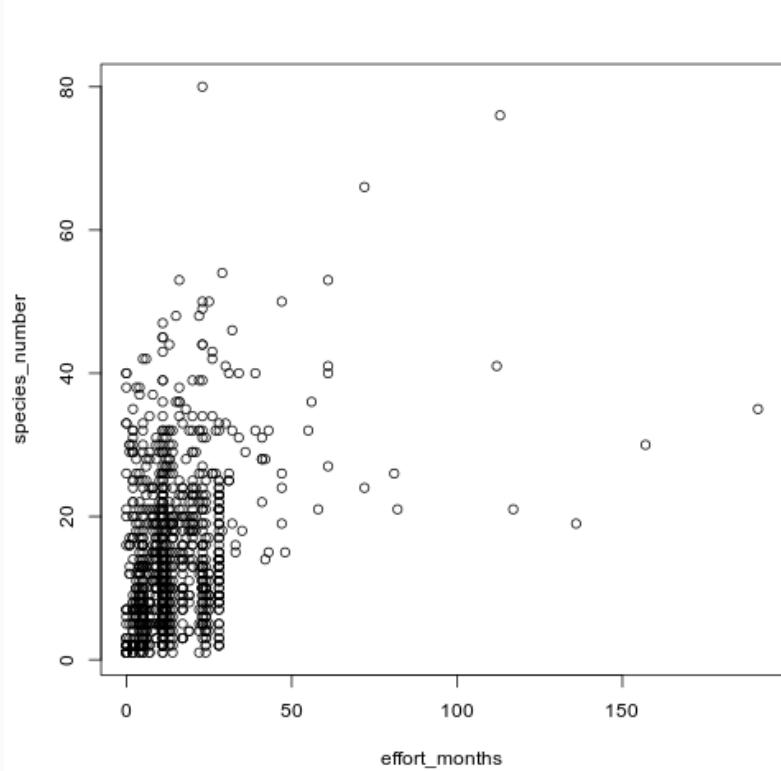
polygon()

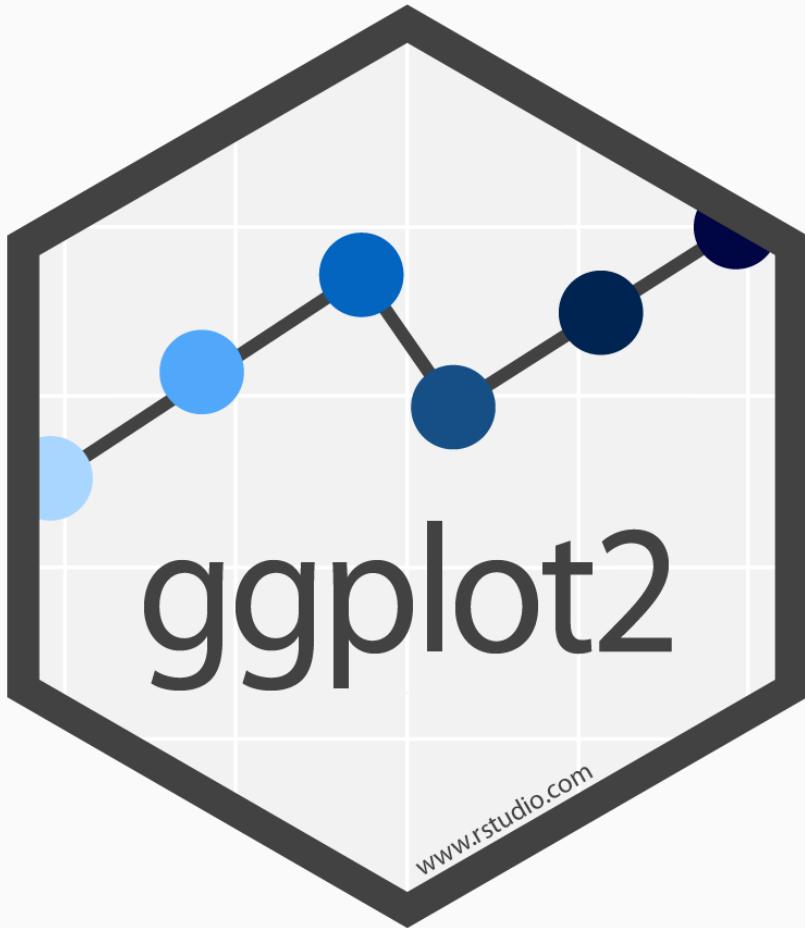


4.1 Principais pacotes para gráficos

graphics

```
# graphics  
plot(species_number ~ effort_months, data = da)
```





4.1 ggplot2

Data Visualization CheatSheet

Data Visualization with ggplot2 :: CHEAT SHEET

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and **geoms**—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<SETTINGS>),  
  stat = <STAT>, position = <POSITION>) +  
<>COORDINATE_FUNCTION> +  
<>FACET_FUNCTION> +  
<>SCALE_FUNCTION> +  
<>THEME_FUNCTION>
```

ggplot(data = mpg, aes(x = cyl, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

```
  aesthetic mappings | data | geom  
  plot(x = cyl, y = hwy, data = mpg, geom = "point")
```

Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot() Returns the last plot

```
ggsave("plot.png", width = 5, height = 5) Saves last plot as 5" x 5" file named "plot.png" in working directory. Matches file type to file extension.
```

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

```
a <- ggplot(economics, aes(date, unemploy))  
b <- ggplot(seals, aes(x = long, y = lat))  
  
a + geom_blank()  
  # (Useful for expanding limits)  
  
b + geom_curve(aes(yend = lat + 1,  
  xend = long + 1, curvature = -2)) x, yend, y, end,  
  alpha, angle, color, curvature, linetype, size  
  
a + geom_path(lineend = "butt", linejoin = "round",  
  x, y, alpha, color, group, linetype, size  
  
a + geom_polygon(aes(group = group)) x, y, alpha, color, fill, group, linetype, size  
  
b + geom_rect(aes(xmin = long, ymin = lat,  
  xmax = long + 1, ymax = lat + 1)) x, xmin, ymin, x, ymax, alpha, fill, group, linetype, size  
  
a + geom_ribbon(aes(ymin = unemploy - 900,  
  ymax = unemploy + 900)) x, y, alpha, color, fill, group, linetype, size
```

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size
b + geom_abline(aes(intercept=0, slope=1))
b + geom_hline(aes(intercept = 0))
b + geom_vline(aes(intercept = long))

b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:1155, radius = 1))

ONE VARIABLE continuous

```
c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)  
  
f + geom_area(stat = "bin") x, y, alpha, color, fill, group,  
  linetype, size  
f + geom_density(kernel = "gaussian") x, y, alpha, color, fill, group, linetype, size, weight  
  
f + geom_dotplot() x, y, alpha, color, fill  
  
f + geom_freqpoly() x, y, alpha, color, group,  
  linetype, size  
  
f + geom_histogram(binwidth = 5) x, y, alpha,  
  color, fill, linetype, size, weight  
  
f2 + geom_qq(aes(sample = hwy)) x, y, alpha,  
  color, fill, linetype, size, weight
```

discrete

```
d <- ggplot(mpg, aes(f))  
d + geom_bar()
```

```
d + geom_bar()
```

```
x, alpha, color, fill, linetype, size, weight
```

TWO VARIABLES

continuous x, continuous y
continuous x, discrete y

```
e + geom_label(aes(label = "A", nudge_x = 1,  
  nudge_y = 1, check_overlap = TRUE)) x, y, label,  
  alpha, angle, color, family, fontface, hjust,  
  lineheight, size, vjust  
  
e + geom_jitter(height = 2, width = 2)  
  x, y, alpha, color, fill, shape, size  
  
e + geom_point() x, y, alpha, color, fill, shape,  
  size, stroke  
  
e + geom_quantile() x, y, alpha, color, group,  
  linetype, size, weight
```

```
e + geom_rect(aes(xmin = long, ymin = lat + 1)) x, x, alpha, color,  
  color, fill, group, linetype, size, weight  
  
e + geom_smooth(method = "lm") x, y, alpha,  
  color, fill, group, linetype, size, weight  
  
e + geom_text(aes(label = "B", nudge_x = 1,  
  nudge_y = 1, check_overlap = TRUE)) x, y, label,  
  alpha, angle, color, family, fontface, hjust,  
  lineheight, size, vjust
```

discrete x, continuous y

```
f <- ggplot(mpg, aes(class, hwy))  
  
f + geom_boxplot() x, y, lower, middle, upper,  
  alpha, color, fill, group, linetype, size, weight  
  
f + geom_dotplot(binaxis = "y", stackdir =  
  "center") x, y, alpha, color, fill, group  
  
f + geom_hex() x, y, alpha, color, fill, group,  
  linetype, size, weight
```

discrete x, discrete y

```
g <- ggplot(diamonds, aes(cut, color))  
  
g + geom_count() x, y, alpha, color, fill, shape,  
  size, stroke
```

THREE VARIABLES

```
sealsSz <- with(seals, sqrt(delta_long^2 + delta_lat^2)) l <- ggplot(seals, aes(long, lat))  
l + geom_contour(aes(z = z)) x, y, z, alpha, colour, group, linetype,  
  size, weight  
  
l + geom_raster(aes(fill = z)) x, y, alpha, fill, hjust = 0.5, vjust = 0.5,  
  interpolate = FALSE  
l + geom_tile(aes(fill = z)) x, y, alpha, color, fill,  
  linetype, size, width
```



RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at <http://ggplot2.tidyverse.org> • ggplot2 2.1.0 • Updated: 2016-11

4.1 Principais pacotes para gráficos

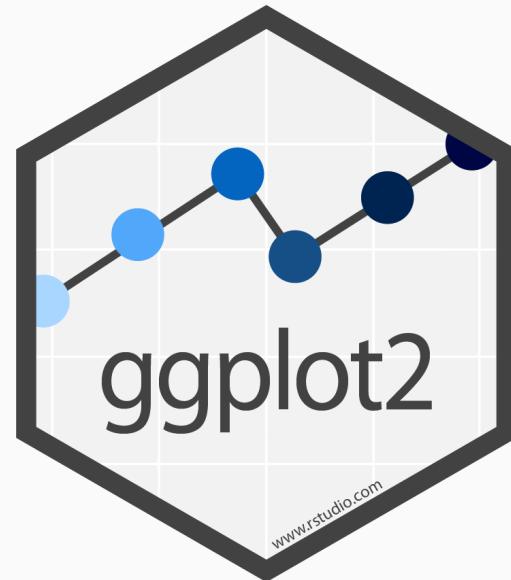
ggplot2

Integrado ao tidyverse, possui uma sintaxe própria

Necessita de funções específicas para objetos de **classes diferentes**

Estruturado dessa forma:

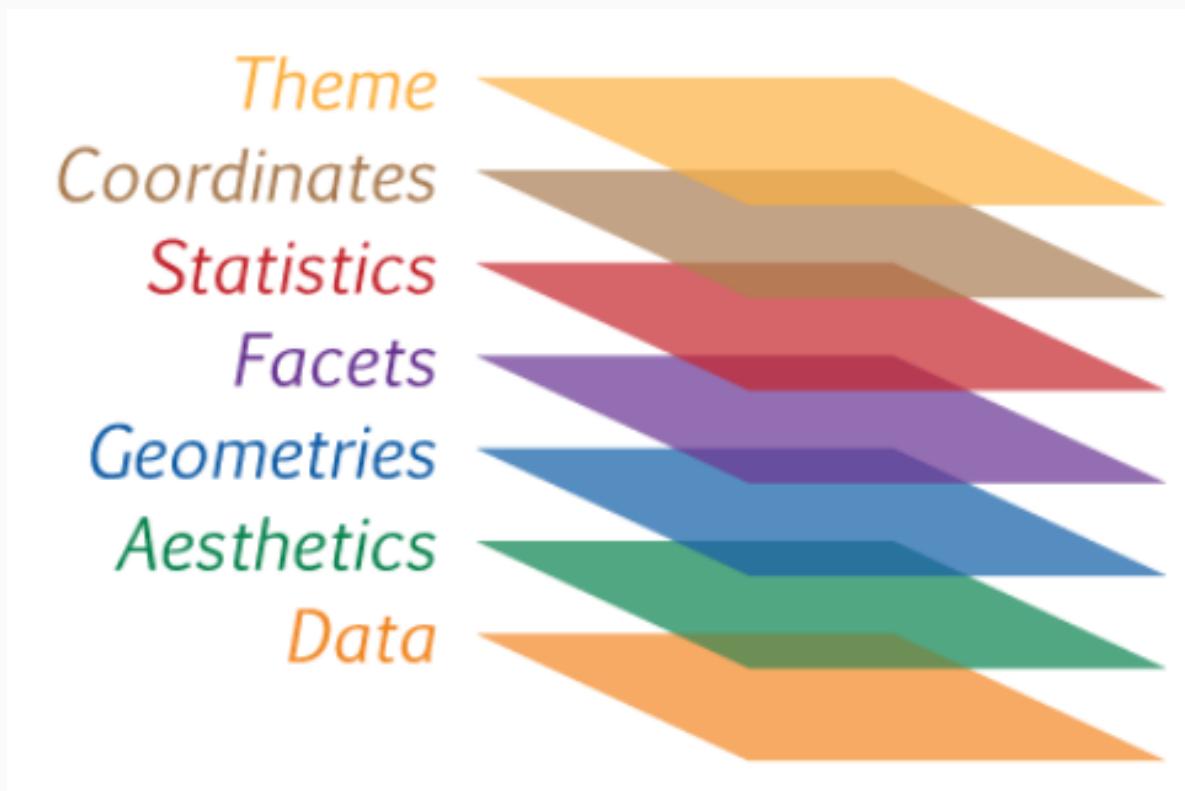
```
ggplot() +  
  aes() +  
  geom_() +  
  facet_() +  
  stats_() +  
  coord_() +  
  theme_()
```



www.rstudio.com

4.1 Principais pacotes para gráficos

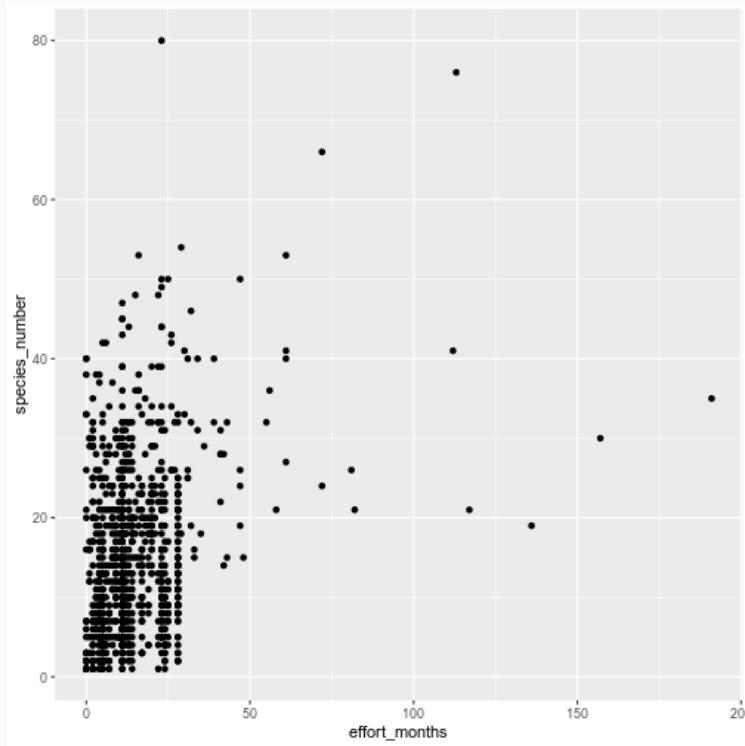
ggplot2



4.1 Principais pacotes para gráficos

ggplot2

```
# ggplot2
library(ggplot2)
ggplot(data = da) + aes(effort_months, species_number) + geom_point()
```





4.1 Principais pacotes para gráficos

ggpubr

Funções fáceis de usar para criar e personalizar plots para publicação baseadas no "ggplot2"

Necessita de funções específicas para gerar **gráficos específicos**

Funções:

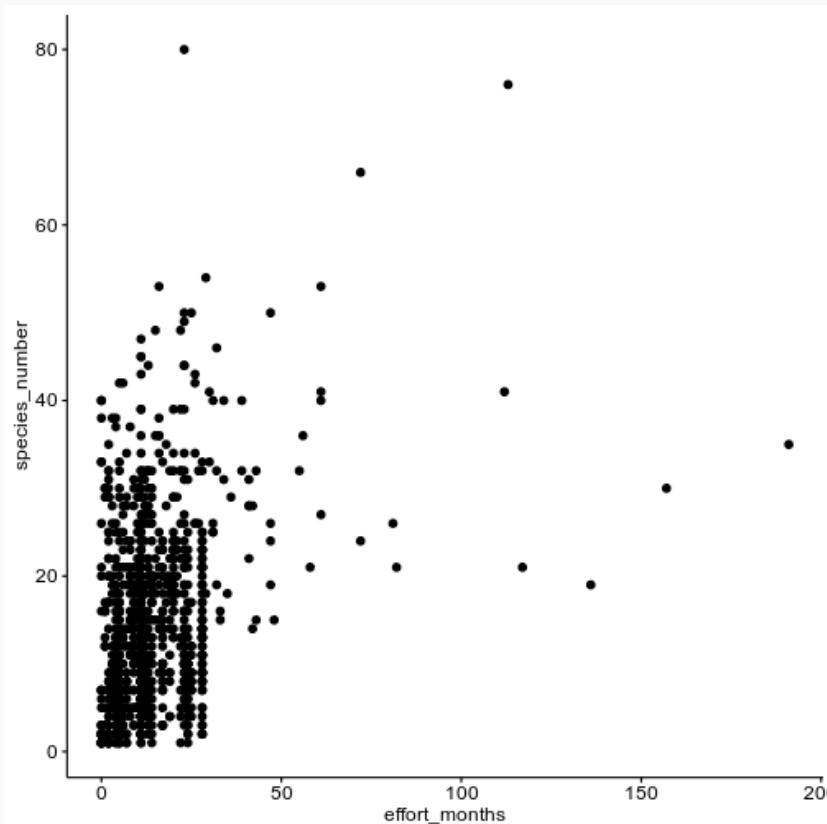
```
gghistogram()  
ggdensity()  
ggboxplot()  
ggviolin()  
ggbargraph()  
ggscatter()
```



4.1 Principais pacotes para gráficos

ggpubr

```
library(ggpubr)  
ggscatter(da, x = "effort_months", y = "species_number")
```



4.2 Principais livros e sites

Livros

O idealizador e mantenedor do pacote **ggplot2** foi de novo o **Hadley Wickham**

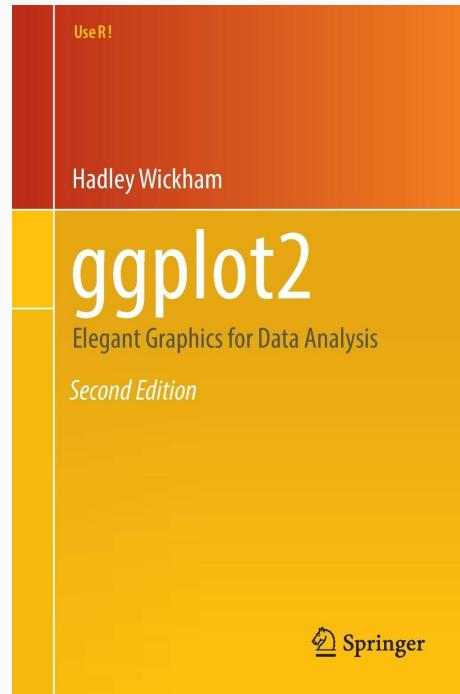


[*] <http://hadley.nz/>

4.2 Principais livros e sites

Livros

ggplot2 (2009, 2016)

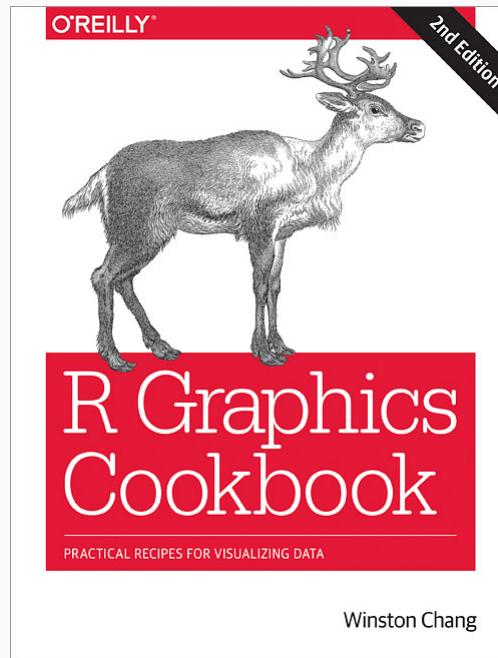


[*] <https://ggplot2.tidyverse.org/>

4.2 Principais livros e sites

Livros

R Graphics Cookbook (2018)

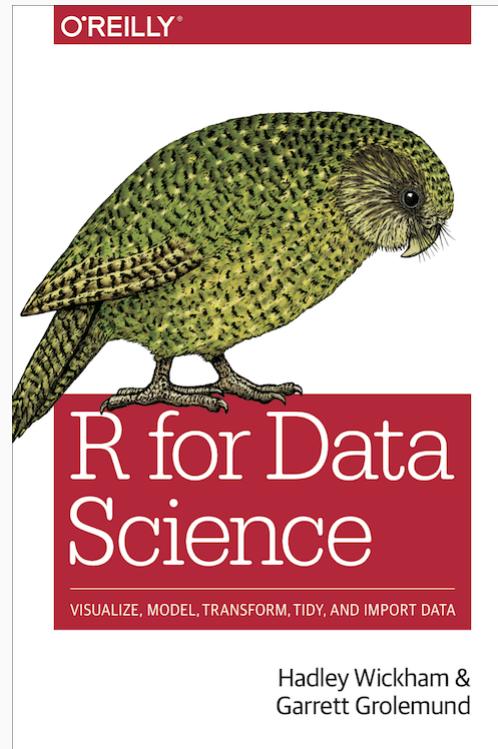


[*] <http://www.cookbook-r.com/Graphs/>

4.2 Principais livros e sites

Livros

R for Data Science (2017)

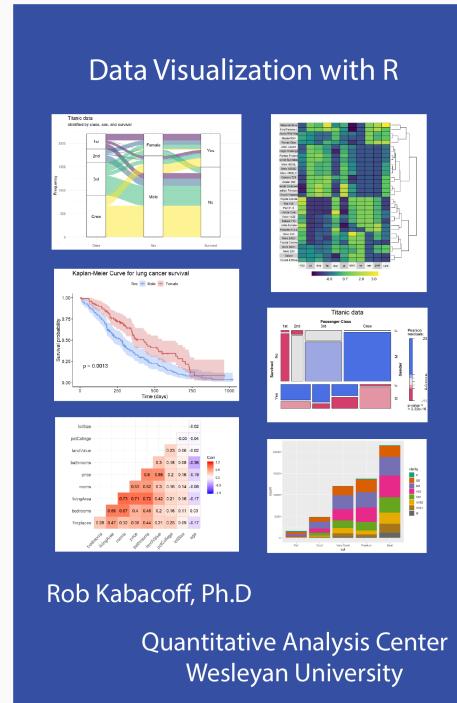


[*] <https://r4ds.had.co.nz/>

4.2 Principais livros e sites

Livros

Data Visualization with R (2018)

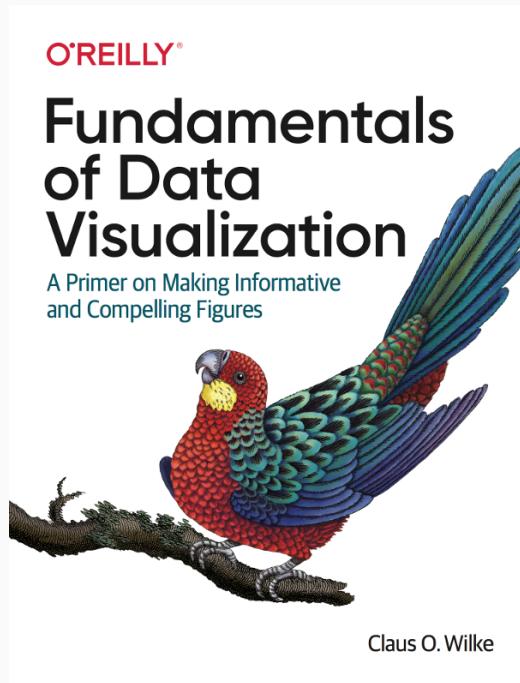


[*] <https://rkabacoff.github.io/datavis/>

4.2 Principais livros e sites

Livros

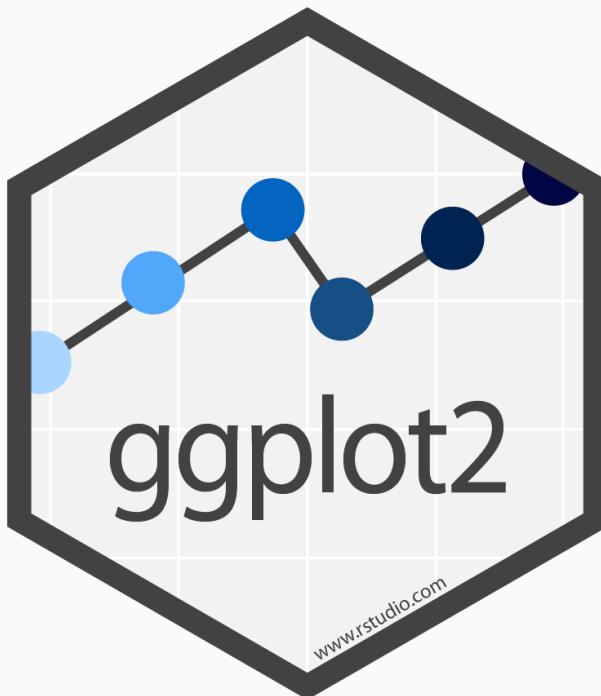
Fundamentals of Data Visualization (2019)



4.2 Principais livros e sites

Sites

ggplot2: Reference



[*] <https://ggplot2.tidyverse.org/reference/>

4.2 Principais livros e sites

Sites

R Graph Gallery



[*] <https://www.r-graph-gallery.com/>

4.2 Principais livros e sites

Sites

from Data to Viz



from **Data** to **Viz**

[*] <https://www.data-to-viz.com/>

4.2 Principais livros e sites

Sites

Statistical tools for high-throughput data analysis (STHDA)



[*] <http://www.sthda.com/english/>

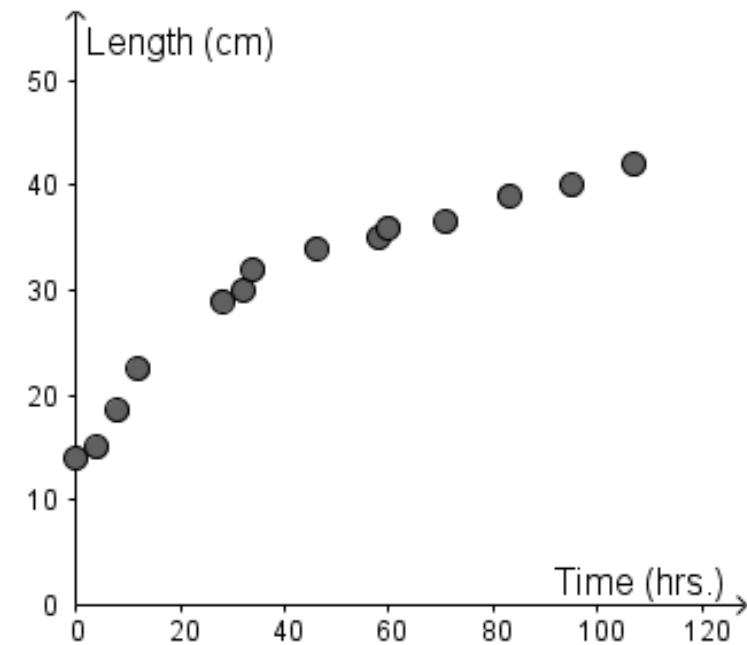
Dúvidas?

Os elementos de um gráfico são representações das colunas (eixos) e linhas (elementos) de nossas matrizes de dados

4.3 Colunas como eixos do gráficos

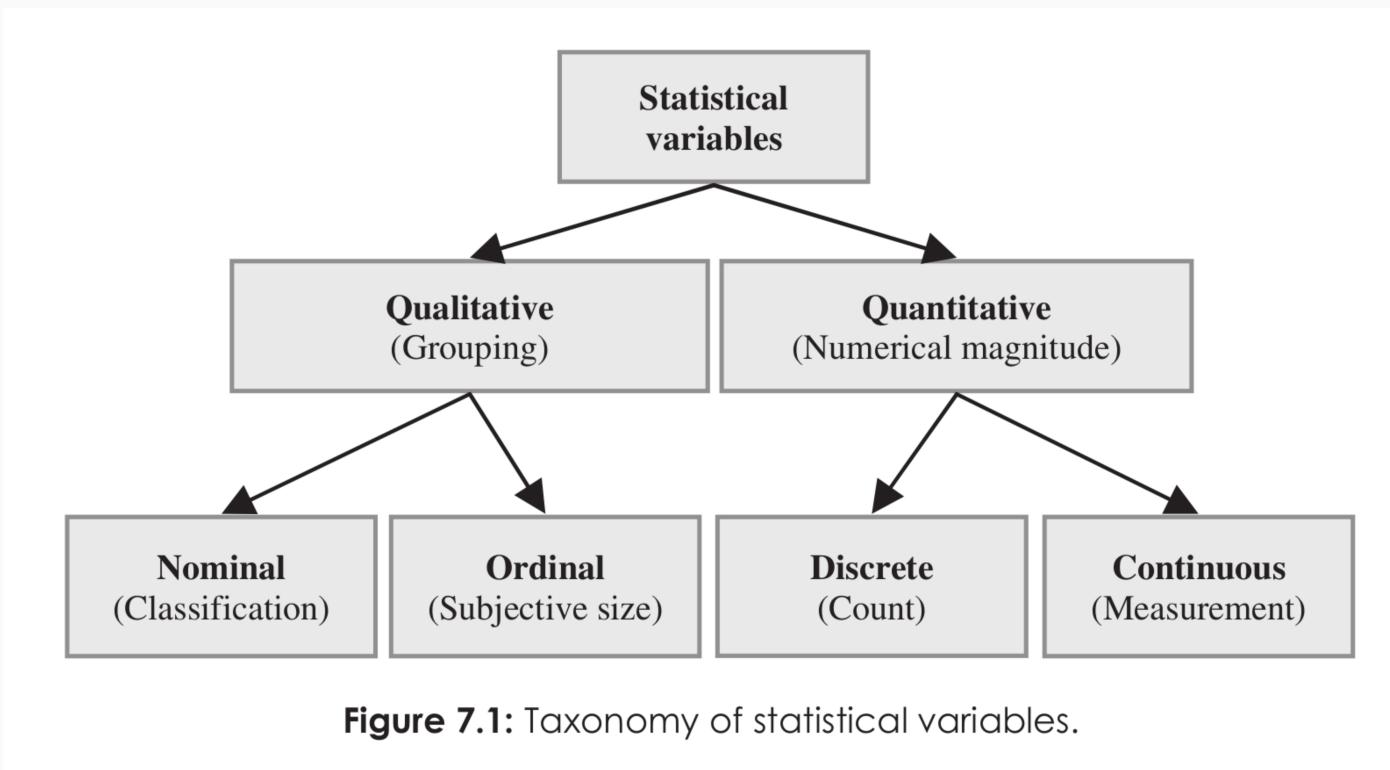
Colunas se tornam eixos e valores a representação

Amount of time from initial measurement (hrs.)	Length
0	14
4	15
8	18.5
12	22.5
28	29
32	30
34	32
46	34
58	35
60	36
71	36.5
83	39
95	40
107	42



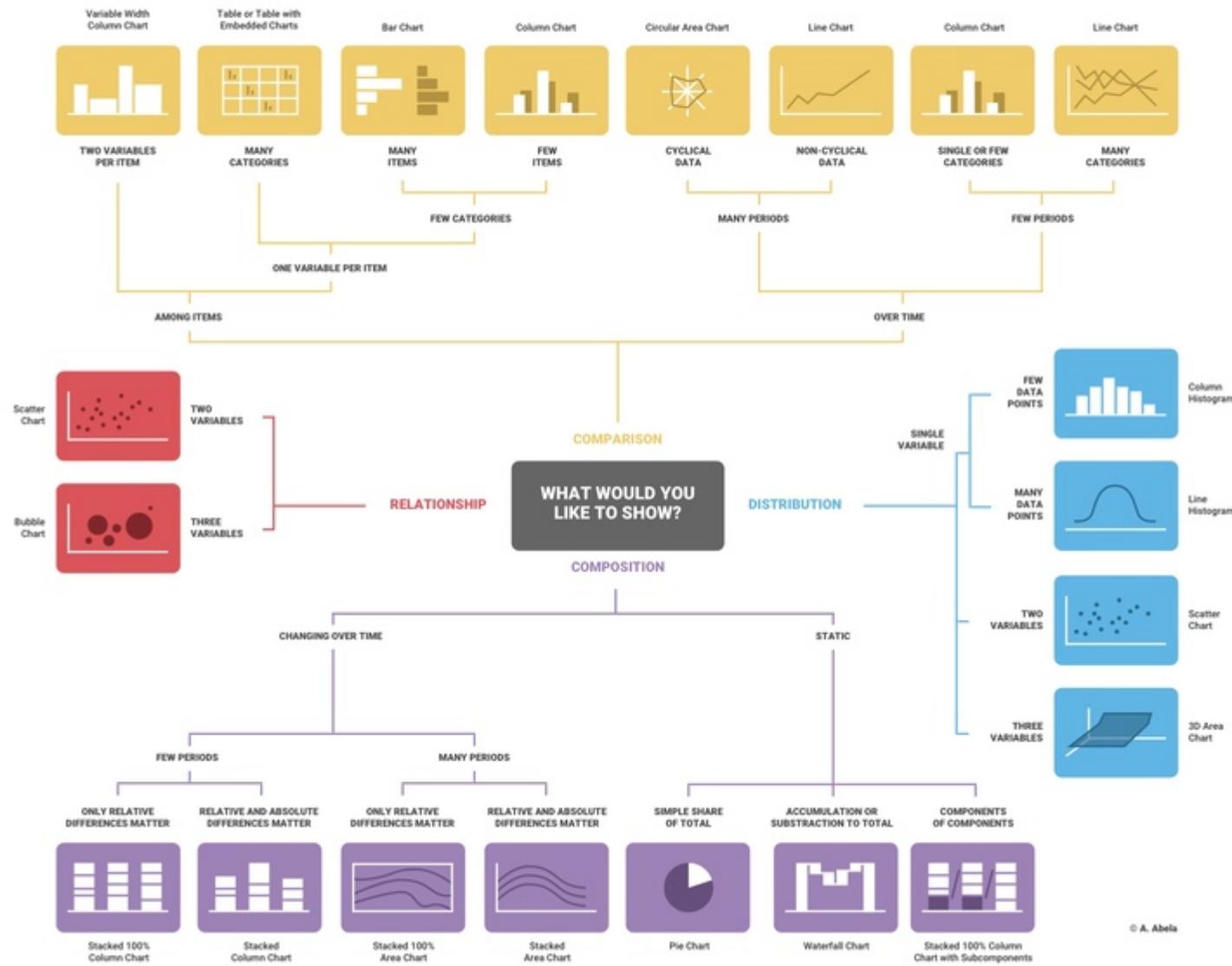
4.3 Colunas como eixos do gráficos

Tipos de variáveis



Ou seja tipo e quantidade das variáveis irá indicar o melhor tipo de gráfico para representar os dados

4.4 Principais tipos de gráficos



4.4 Principais tipos de gráficos

Sites

from Data to Viz



from **Data** to **Viz**

[*] <https://www.data-to-viz.com/>

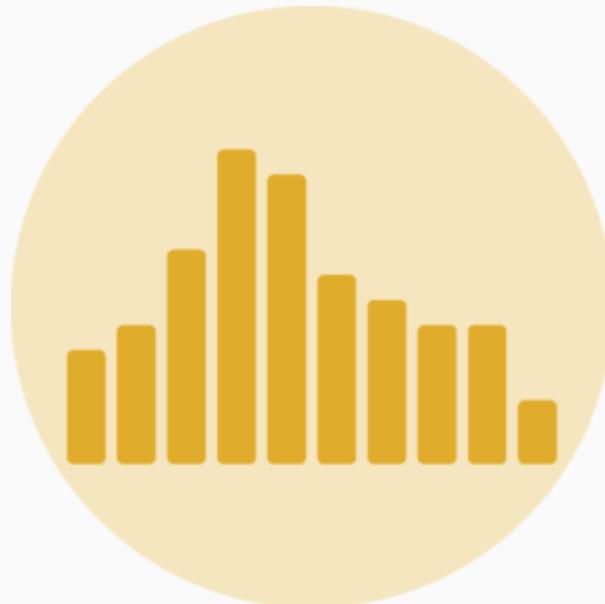
Dúvidas?

4.5 Histograma (histogram)

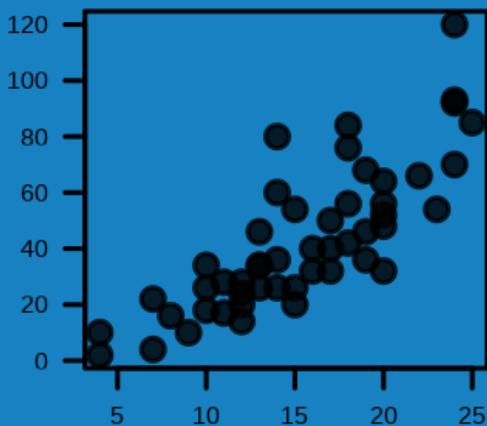
Representa os dados de: uma coluna

Tipo de dado: discreto ou contínuo

Distribuição de frequência e densidade de dados contínuos



graphics



4.5 Histogramma (histogram)

graphics

```
hist(da$species_number)
```

4.5 Histogramma (histogram)

graphics

```
hist(da$species_number,  
      col = "gray50",  
      border = "gray")
```

4.5 Histogramma (histogram)

graphics

```
hist(da$species_number,  
      col = "gray50",  
      border = "gray",  
      main = "Ti")
```

4.5 Histogramma (histogram)

graphics

```
hist(da$species_number,  
      col = "gray50",  
      border = "gray",  
      main = "Ti",  
      xlab = "Sp",  
      ylab = "Fr")
```

4.5 Histogramma (histogram)

graphics

```
hist(da$species_number,  
      col = "gray50",  
      border = "gray",  
      main = "Ti",  
      xlab = "Sp",  
      ylab = "Fr",  
      br = 50)
```

4.5 Histogramma (histogram)

graphics

```
hist(da$species_number,  
      col = "gray50",  
      border = "gray",  
      main = "Ti",  
      xlab = "Sp",  
      ylab = "Fr",  
      br = 50,  
      cex.main = 2.5,  
      cex.lab = 2.2,  
      cex.axis = 2)
```

4.5 Histogramma (histogram)

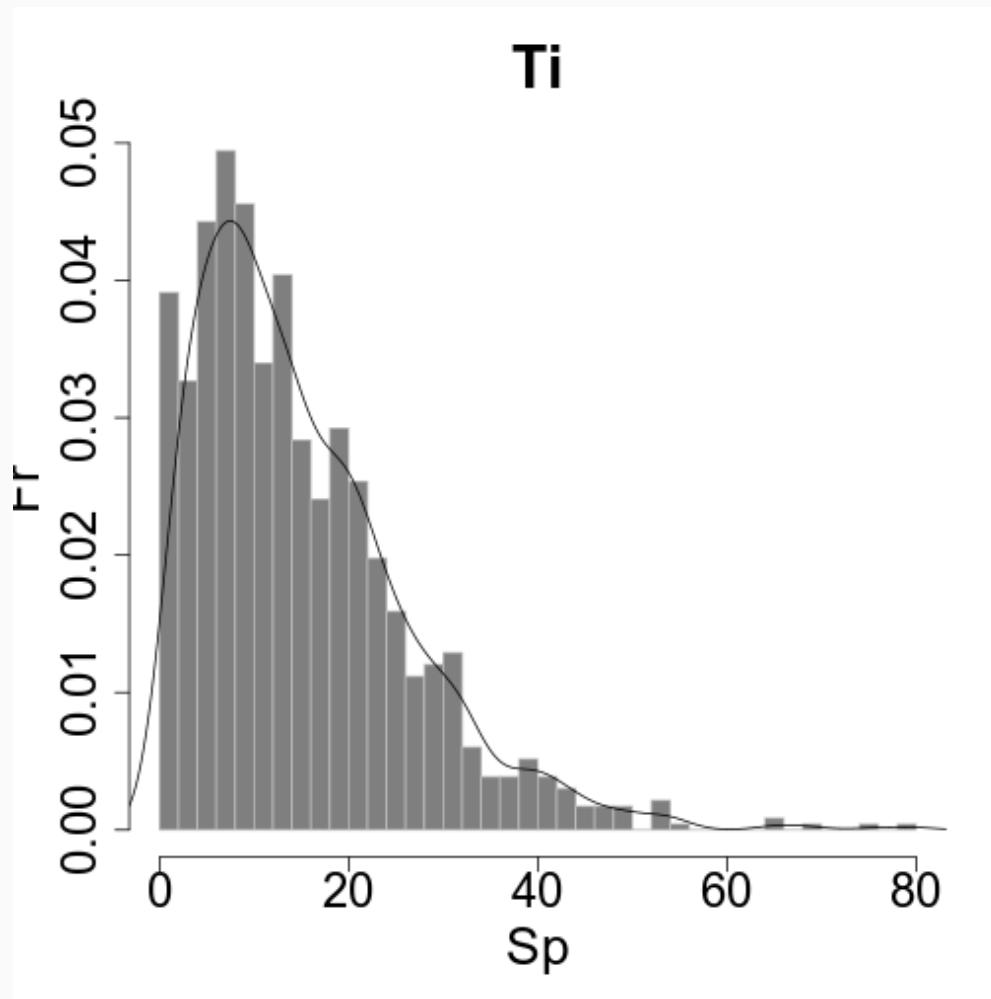
graphics

```
hist(da$species_number,
  col = "gray50",
  border = "gray",
  main = "Ti",
  xlab = "Sp",
  ylab = "Fr",
  br = 50,
  cex.main = 2.5,
  cex.lab = 2.2,
  cex.axis = 2,
  prob = TRUE)

lines(density(da$species_number))
```

4.5 Histogramma (histogram)

graphics



4.5 Histograma (histogram)

graphics

Exportar

```
# diretorio  
setwd("")
```

4.5 Histograma (histogram)

graphics

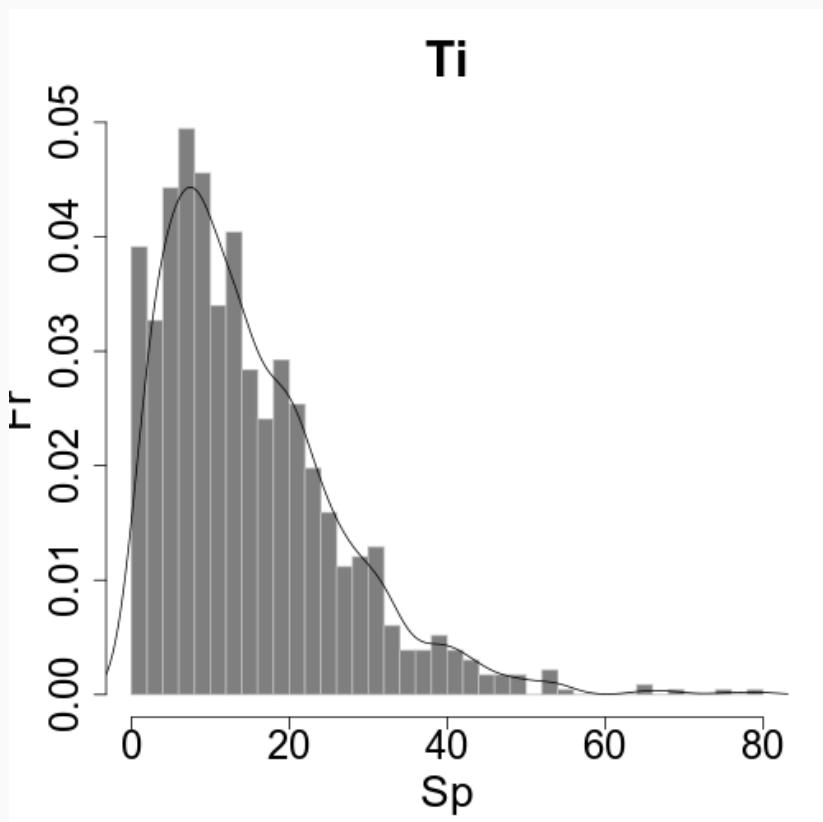
Exportar

```
png("meu_primeiro_histograma.png", wi = 15, he = 15, un = "cm", res = 300)
```

4.5 Histograma (histogram)

graphics

Exportar



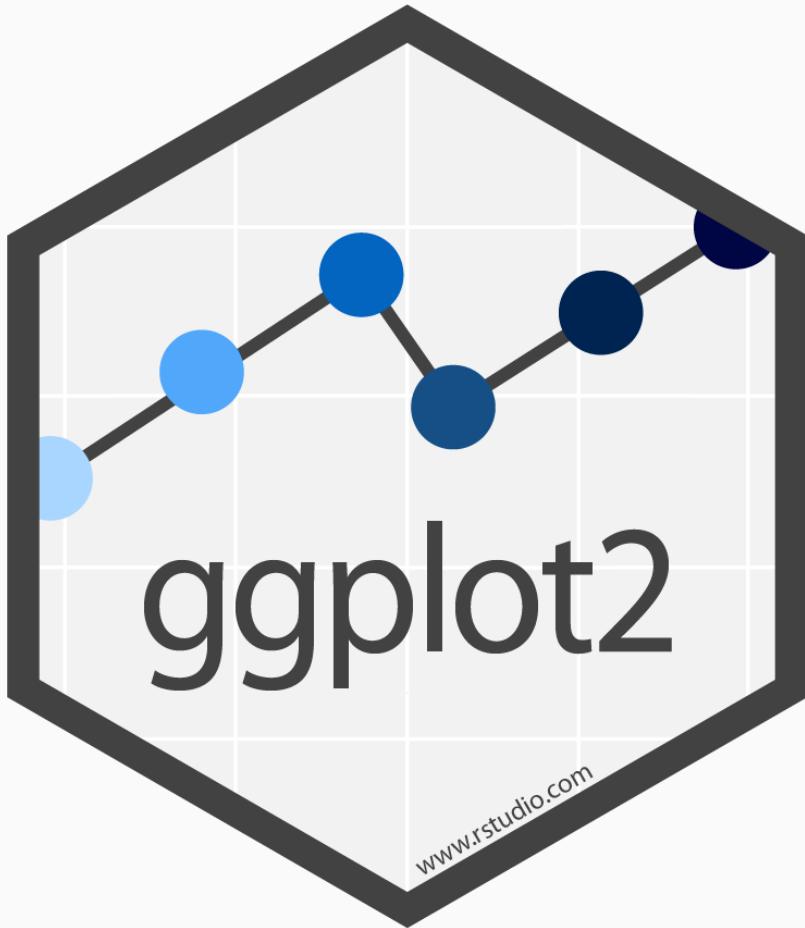
4.5 Histograma (histogram)

graphics

Exportar

```
dev.off()
```

```
## png  
##     3
```



4.5 Histograma (histogram)

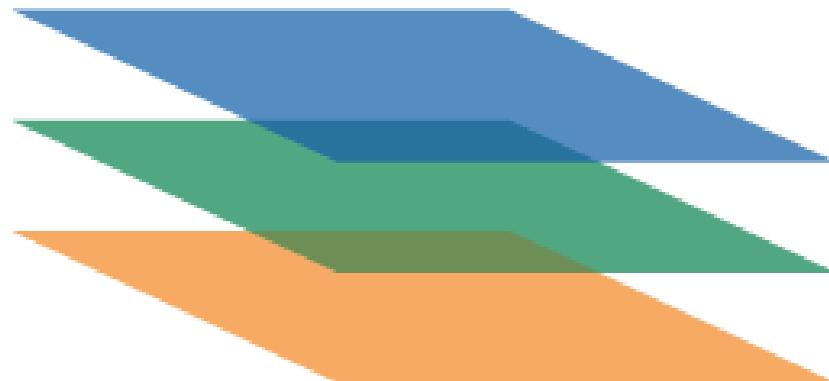
ggplot2

A gramática dos gráficos

Geometries

Aesthetics

Data



4.5 Histograma (histogram)

ggplot2

```
ggplot(data = da)
```

4.5 Histogramma (histogram)

ggplot2

```
ggplot(data = da) +  
  aes(x = species_number)
```

4.5 Histograma (histogram)

ggplot2

```
ggplot(data = da) +  
  aes(x = species_number) +  
  geom_histogram()
```

4.5 Histograma (histogram)

ggplot2

```
ggplot(data = da) +  
  aes(x = species_number) +  
  geom_density()
```

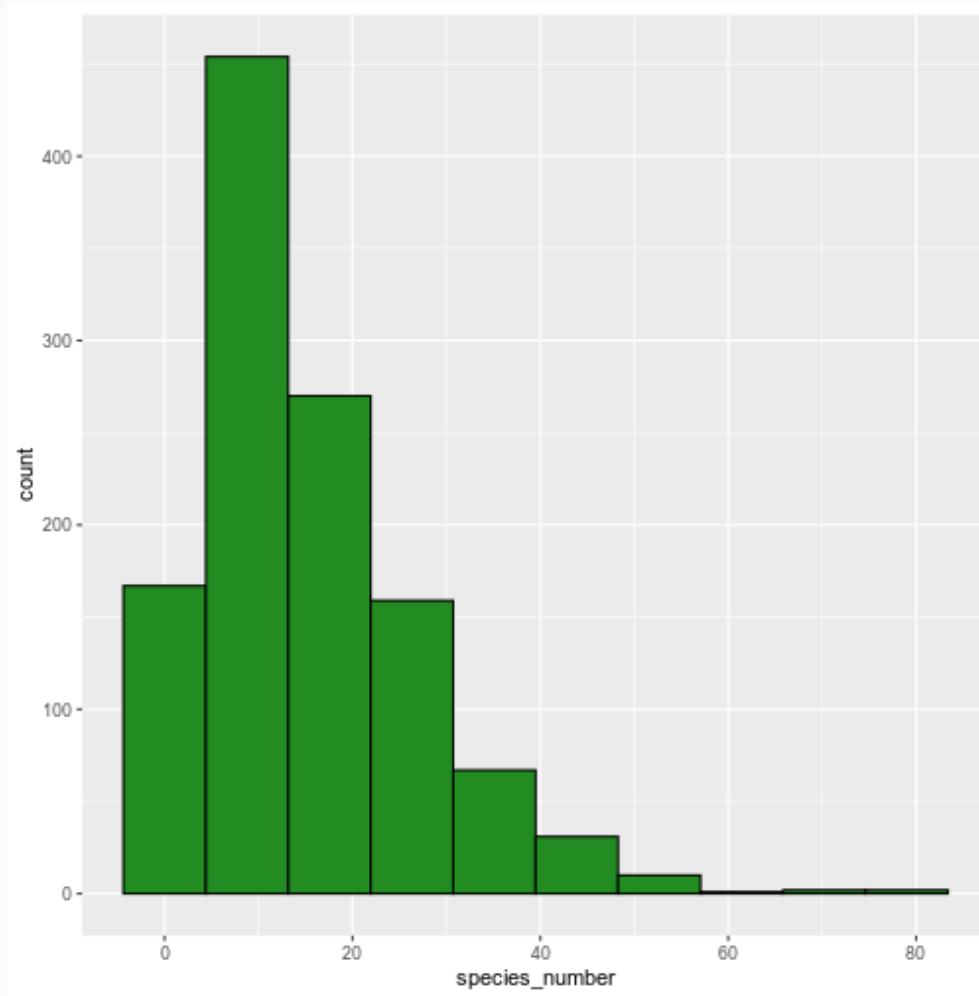
4.5 Histograma (histogram)

ggplot2

```
ggplot(data = da) +  
  aes(x = species_number) +  
  geom_histogram(color = "black", fill = "forest green", bins = 10)
```

4.5 Histogramma (histogram)

ggplot2



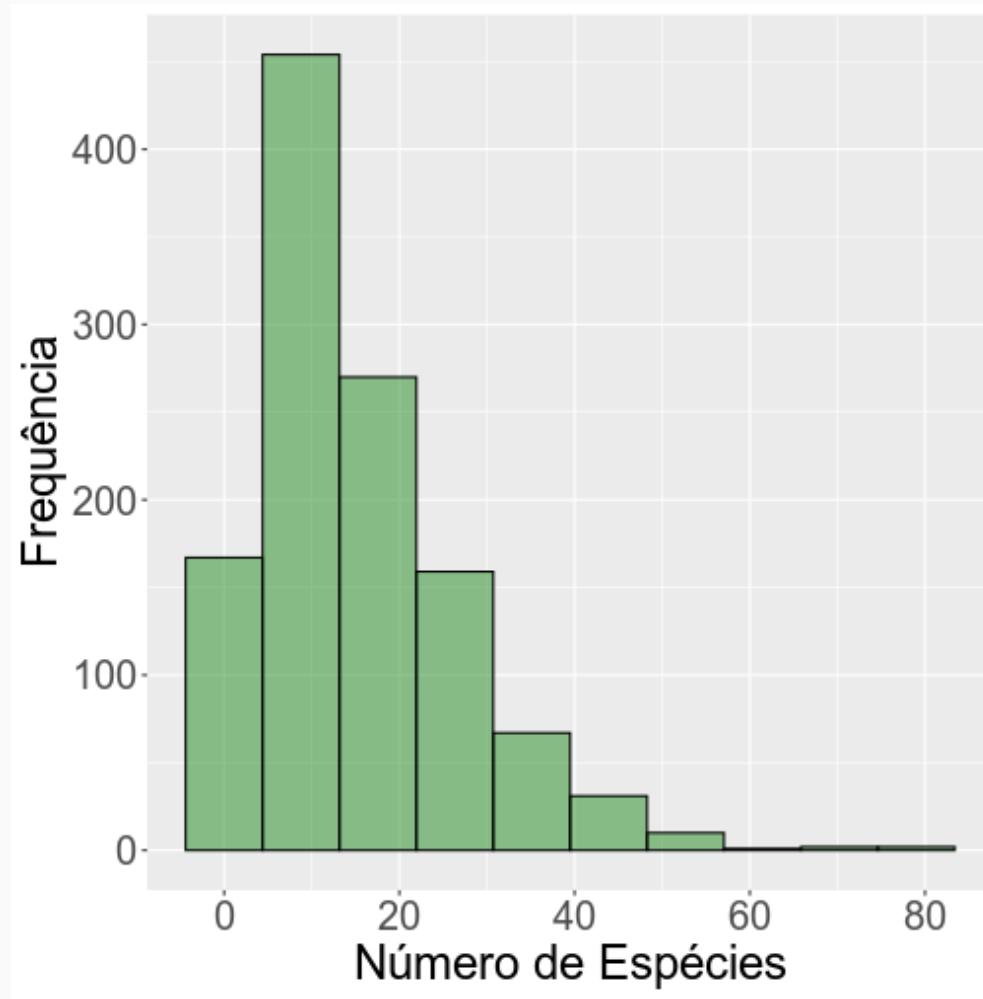
4.5 Histograma (histogram)

ggplot2

```
ggplot(data = da) +  
  aes(x = species_number) +  
  geom_histogram(color = "black", fill = "forest green", bins = 10, alpha =  
    labs(x = "Número de Espécies", y = "Frequência") +  
    theme(axis.title = element_text(size = 24),  
      axis.text.x = element_text(size = 20),  
      axis.text.y = element_text(size = 20))
```

4.5 Histograma (histogram)

ggplot2



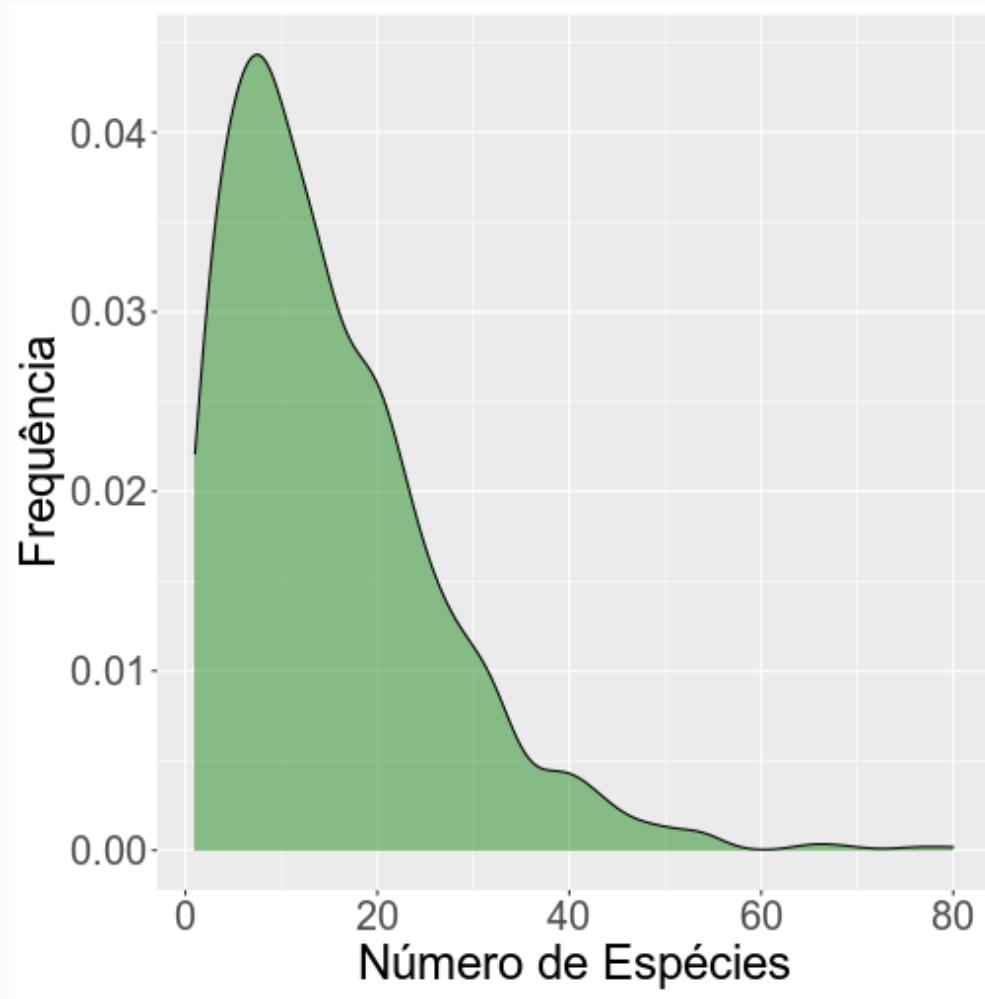
4.5 Histograma (histogram)

ggplot2

```
ggplot(data = da) +  
  aes(x = species_number) +  
  geom_density(color = "black", fill = "forest green", alpha = .5) +  
  labs(x = "Número de Espécies", y = "Frequência") +  
  theme(axis.title = element_text(size = 24),  
        axis.text.x = element_text(size = 20),  
        axis.text.y = element_text(size = 20))
```

4.5 Histograma (histogram)

ggplot2



4.5 Histogramma (histogram)

ggplot2



4.5 Histogramma (histogram)

ggplot2



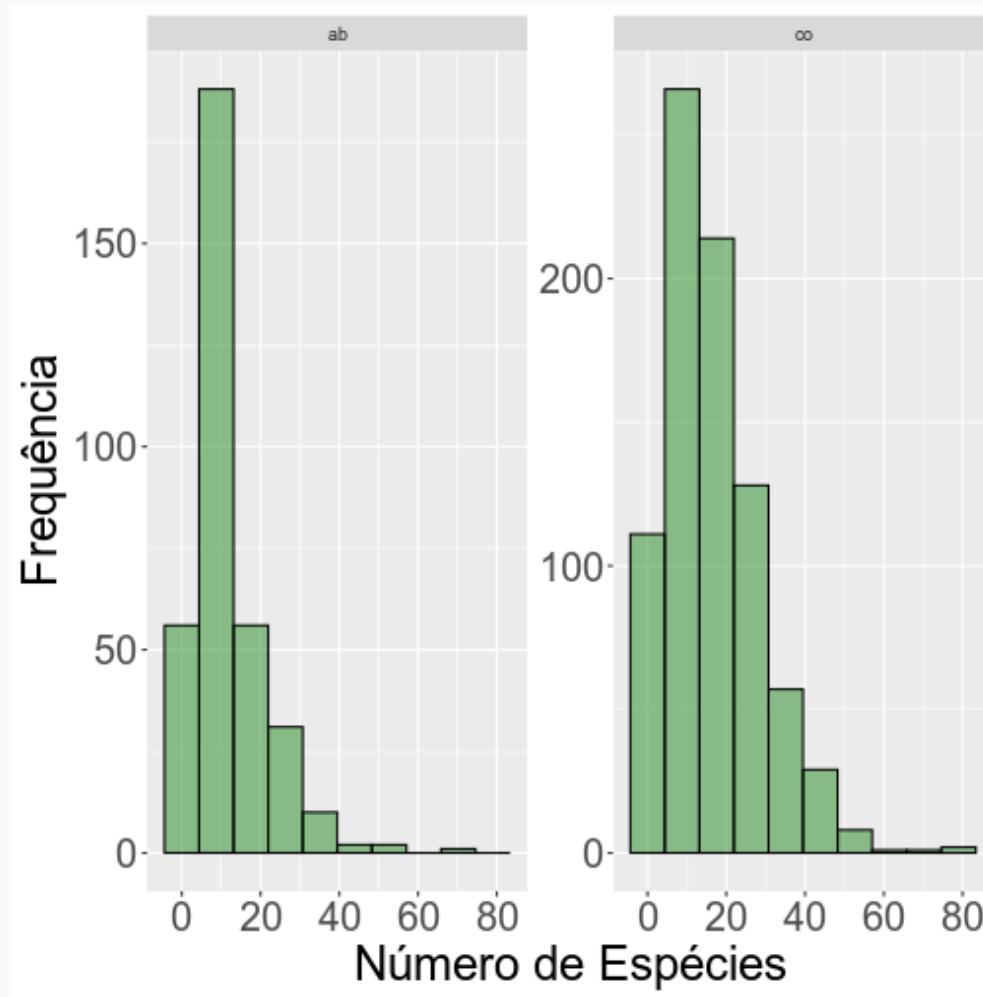
4.5 Histograma (histogram)

ggplot2

```
ggplot(data = da) +  
  aes(x = species_number) +  
  geom_histogram(color = "black", fill = "forest green", bins = 10,  
                 alpha = .5) +  
  facet_wrap(~ record, ncol = 2, scale = "free_y") +  
  labs(x = "Número de Espécies", y = "Frequência") +  
  theme(axis.title = element_text(size = 24),  
        axis.text.x = element_text(size = 20),  
        axis.text.y = element_text(size = 20))
```

4.5 Histograma (histogram)

ggplot2



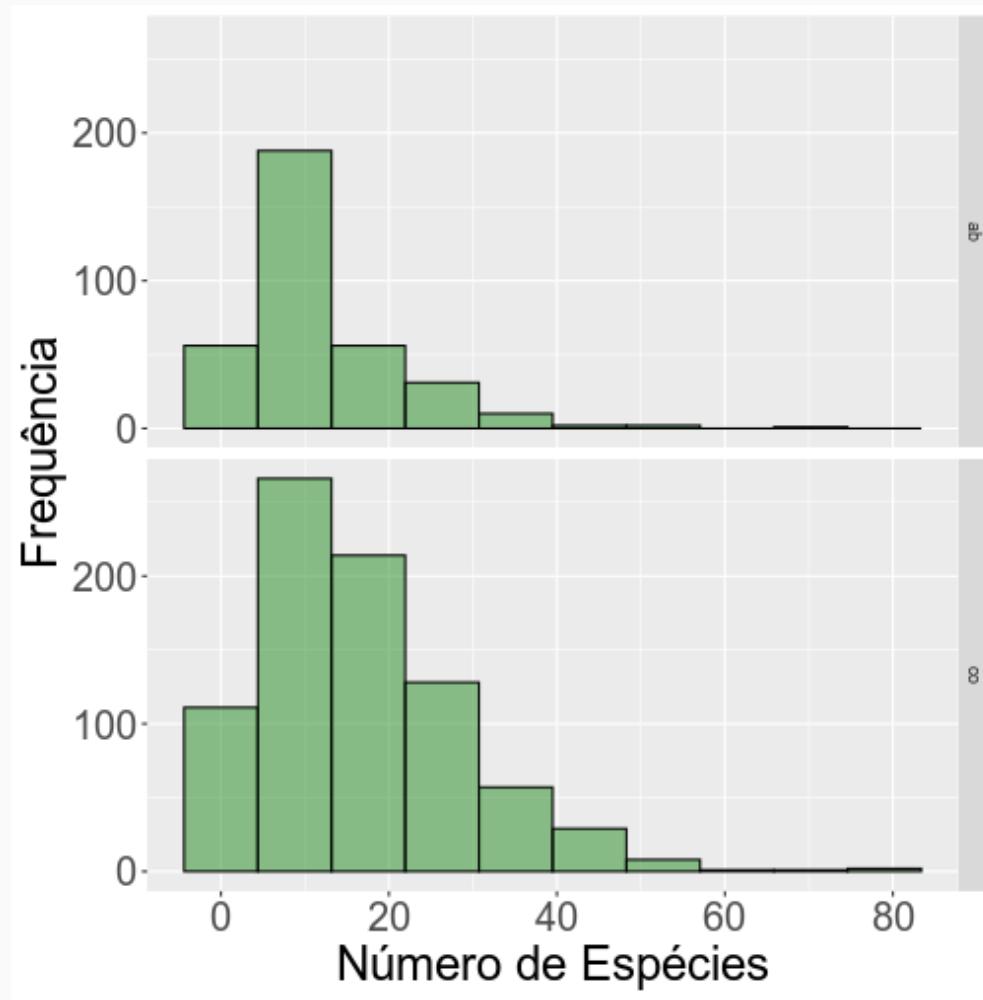
4.5 Histograma (histogram)

ggplot2

```
ggplot(data = da) +  
  aes(x = species_number) +  
  geom_histogram(color = "black", fill = "forest green", bins = 10,  
                 alpha = .5) +  
  facet_grid(record ~ .) +  
  labs(x = "Número de Espécies", y = "Frequência") +  
  theme(axis.title = element_text(size = 24),  
        axis.text.x = element_text(size = 20),  
        axis.text.y = element_text(size = 20))
```

4.5 Histograma (histogram)

ggplot2



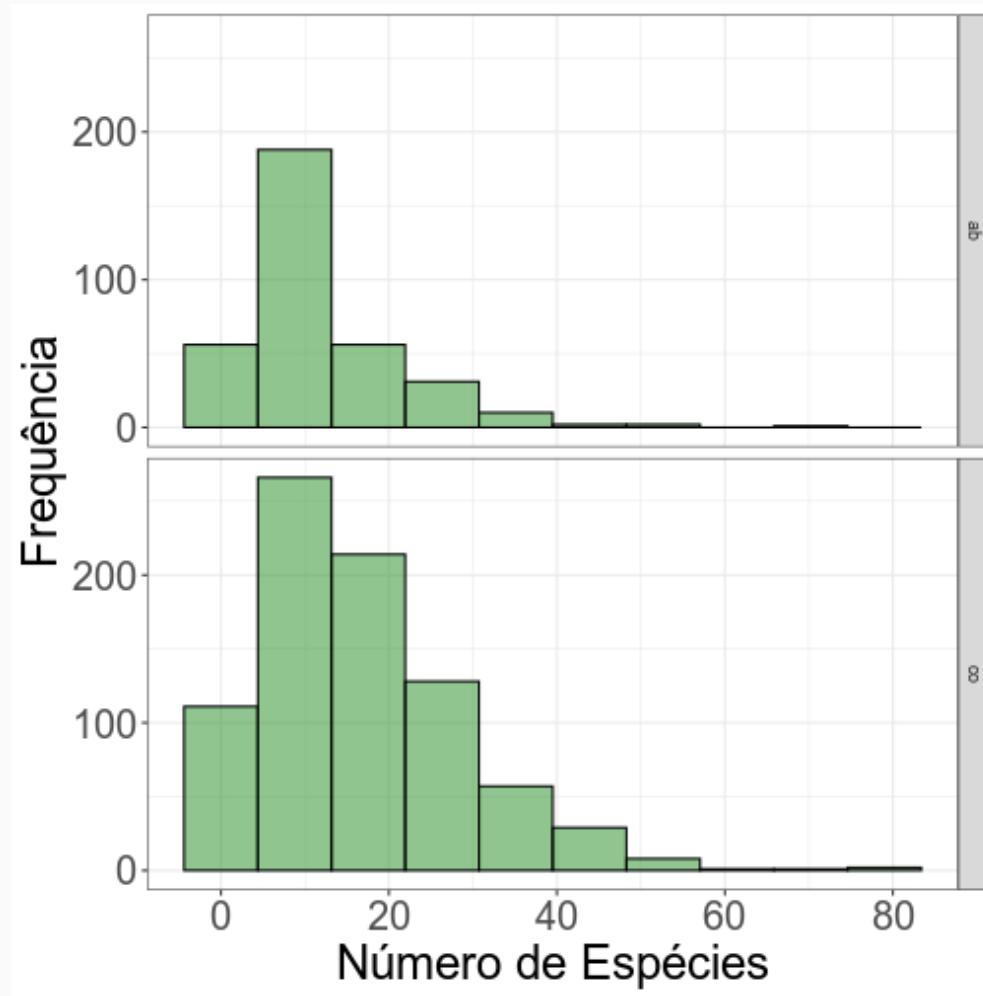
4.5 Histograma (histogram)

ggplot2

```
ggplot(data = da) +  
  aes(x = species_number) +  
  geom_histogram(color = "black", fill = "forest green", bins = 10,  
                 alpha = .5) +  
  facet_grid(record ~ .) +  
  labs(x = "Número de Espécies",  
        y = "Frequênci") +  
  theme_bw() +  
  theme(axis.title = element_text(size = 24),  
        axis.text.x = element_text(size = 20),  
        axis.text.y = element_text(size = 20))
```

4.5 Histograma (histogram)

ggplot2



4.5 Histograma (histogram)

ggplot2

```
ggplot(data = da) +  
  aes(x = species_number) +  
  geom_histogram(color = "black", fill = "forest green", bins = 10,  
                 alpha = .5) +  
  facet_grid(record ~ .) +  
  labs(x = "Número de Espécies",  
        y = "Frequência") +  
  theme_bw() +  
  theme(axis.title = element_text(size = 24),  
        axis.text.x = element_text(size = 20),  
        axis.text.y = element_text(size = 20))  
  
ggsave("histogram_ggplot2.png", wi = 20, he = 15, un = "cm", dpi = 300)
```



GGPUBR
Publication-
ready plots

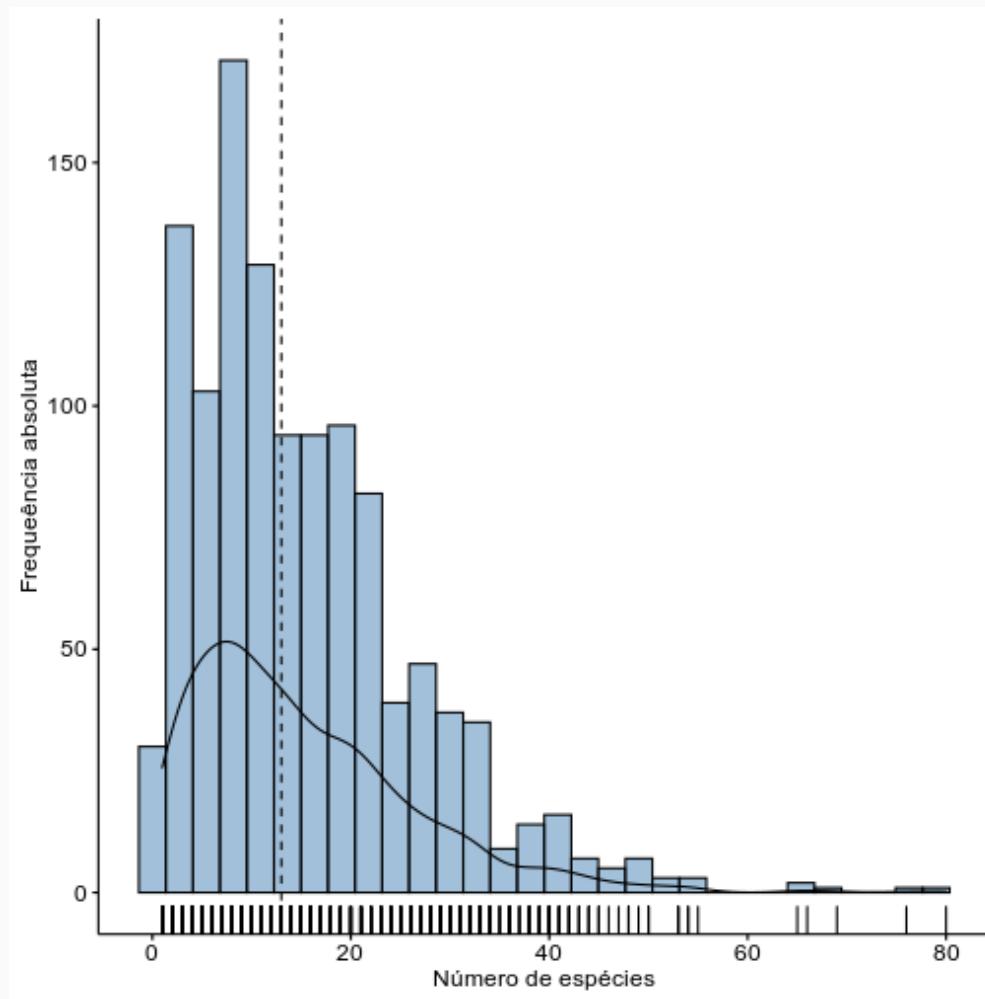
4.5 Histograma (histogram)

ggpubr

```
gghistogram(data = da,  
            x = "species_number",  
            add = "median",  
            fill = "steelblue",  
            rug = TRUE,  
            add_density = TRUE,  
            xlab = "Número de espécies",  
            ylab = "Frequência absoluta")
```

4.5 Histograma (histogram)

ggpubr



4.5 Histograma (histogram)

ggpubr

```
gghistogram(data = da,  
            x = "species_number",  
            add = "median",  
            fill = "steelblue",  
            rug = TRUE,  
            add_density = TRUE,  
            xlab = "Número de espécies",  
            ylab = "Frequência absoluta")  
  
ggsave("histogram_ggpubr.png", wi = 20, he = 15, un = "cm", dpi = 300)
```

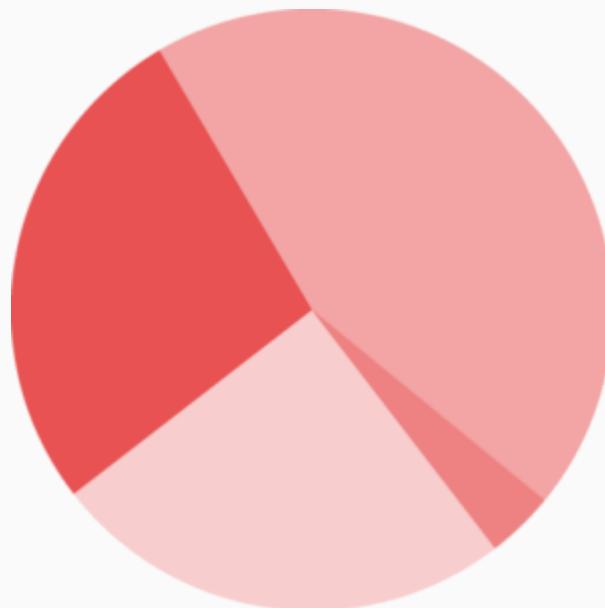
Dúvidas?

4.6 Gráfico de setores (pie chart)

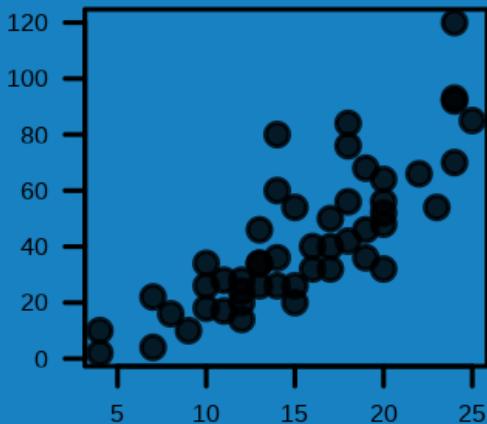
Representa os dados de: uma coluna

Tipo de dado: categórico

Proporção ou porcentagem de dados categóricos



graphics



4.6 Gráfico de setores (pie chart)

Clássico

```
par(mar = c(0, 1, 0, 1))
pie(c(280, 60, 20),
    c("Sky", "Sunny side of pyramid", "Shady side of pyramid"),
    col = c("#0292D8", "#F7EA39", "#C4B632"),
    init.angle = -50, border = NA)
```

4.6 Gráfico de setores (pie chart)

Tabela de frequênciā

```
# tabela de frequencia
ta <- table(da$record)
ta <- round(ta/sum(ta) * 100, 2)
ta
```

```
##  
##      ab      co  
## 29.75 70.25
```

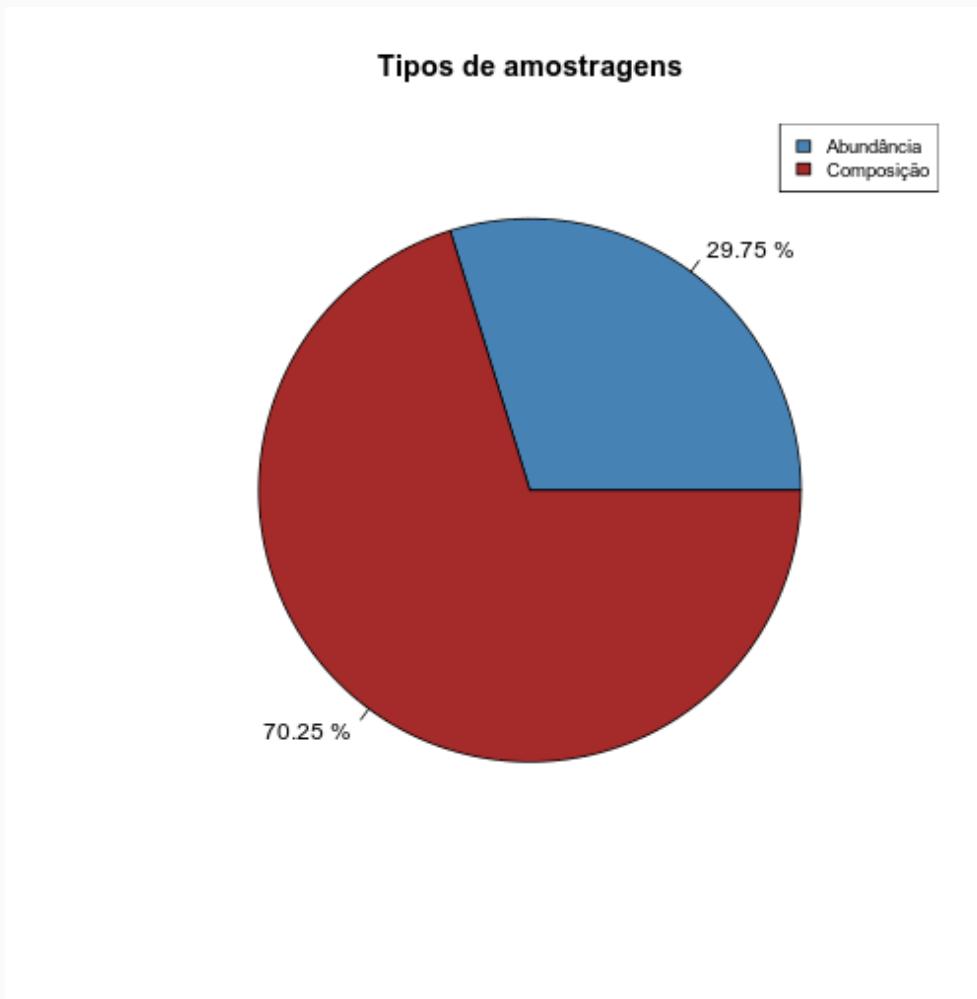
4.6 Gráfico de setores (pie chart)

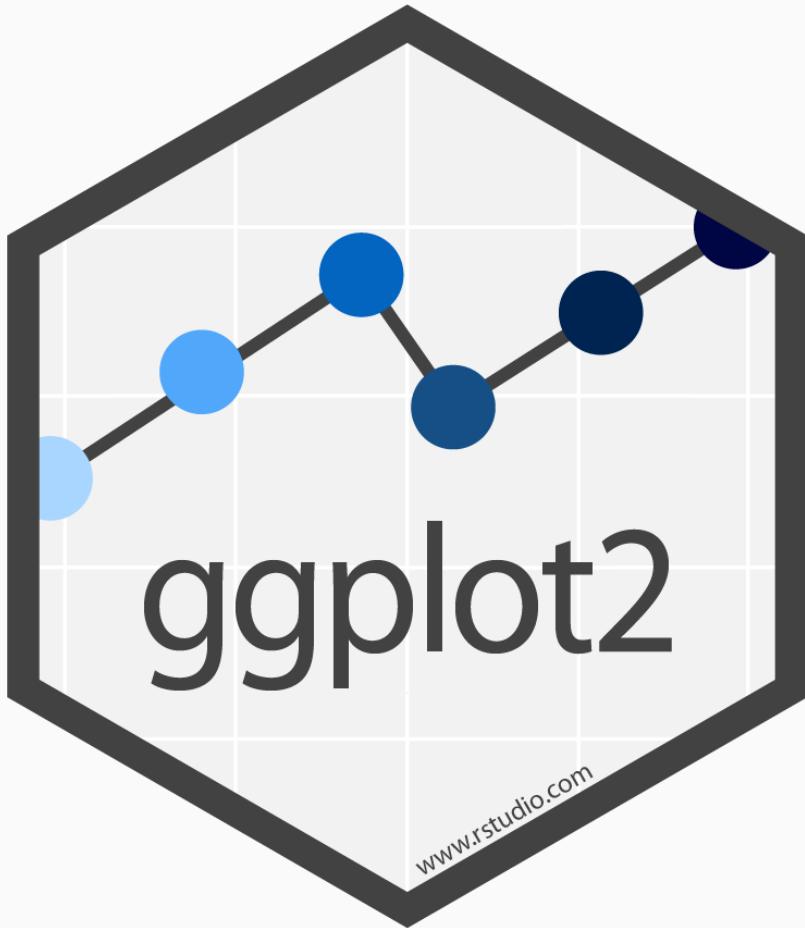
graphics

```
pie(ta,
    labels = paste(ta, "%"),
    main = "Tipos de amostragens",
    col = c("steelblue", "brown"))
legend("topright", c("Abundância", "Composição"), cex = 0.8,
       fill = c("steelblue", "brown"))
```

4.6 Gráfico de setores (pie chart)

graphics





4.6 Gráfico de setores (pie chart)

Tabela de frequênci

```
ta_por <- ta %>%
  as.data.frame %>%
  mutate(Amostragem = c("Abundância", "Composição"),
         porc = paste0(Freq, "%"))
ta_por
```

```
##   Var1   Freq Amostragem    porc
## 1 ab  29.75 Abundância 29.75%
## 2 co  70.25 Composição 70.25%
```

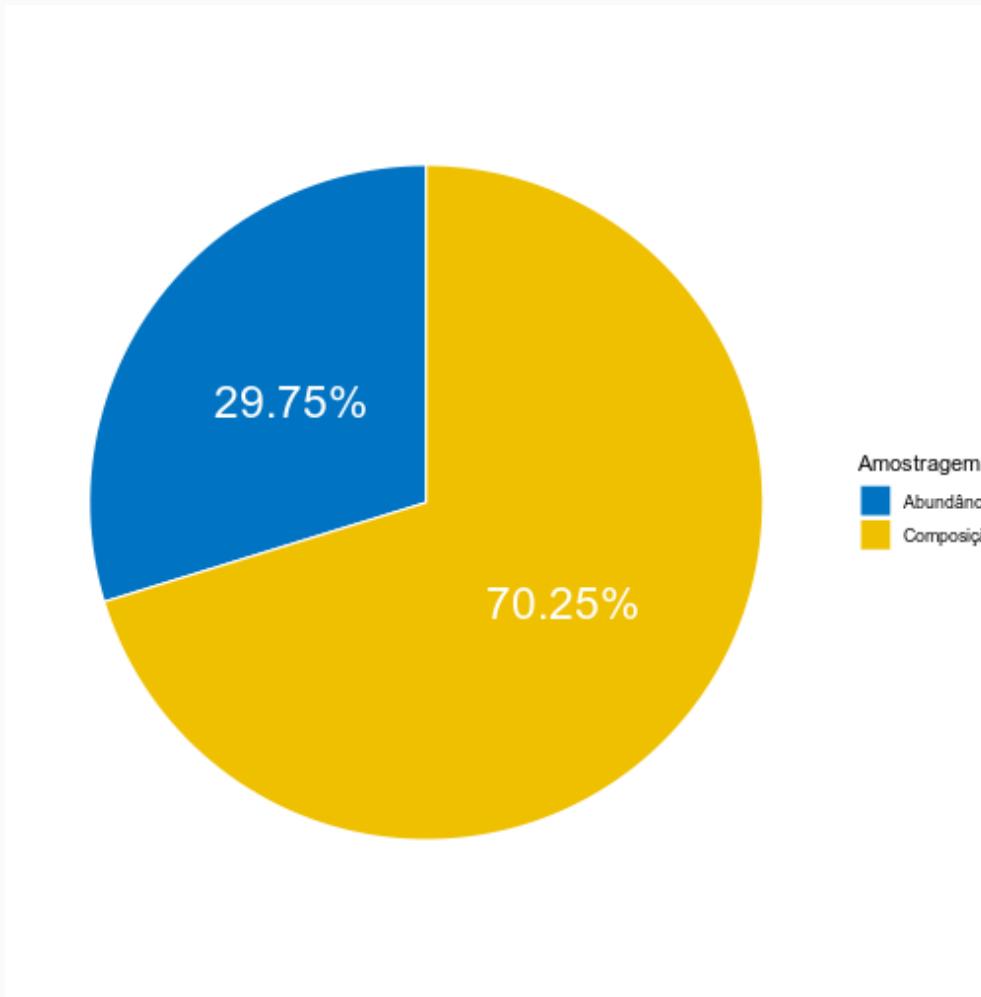
4.6 Gráfico de setores (pie chart)

ggplot2

```
# pie
ggplot(ta_por) +
  aes(x = "", y = Freq, fill = Amostragem) +
  geom_bar(width = 1, stat = "identity", color = "white") +
  coord_polar("y", start = 0) +
  geom_text(aes(label = porc), color = "white",
            position = position_stack(vjust = 0.5), size = 8) +
  scale_fill_manual(values = c(c("#0073C2FF", "#EFC000FF"))) +
  theme_void()
```

4.6 Gráfico de setores (pie chart)

ggplot2





GGPUBR
Publication-
ready plots

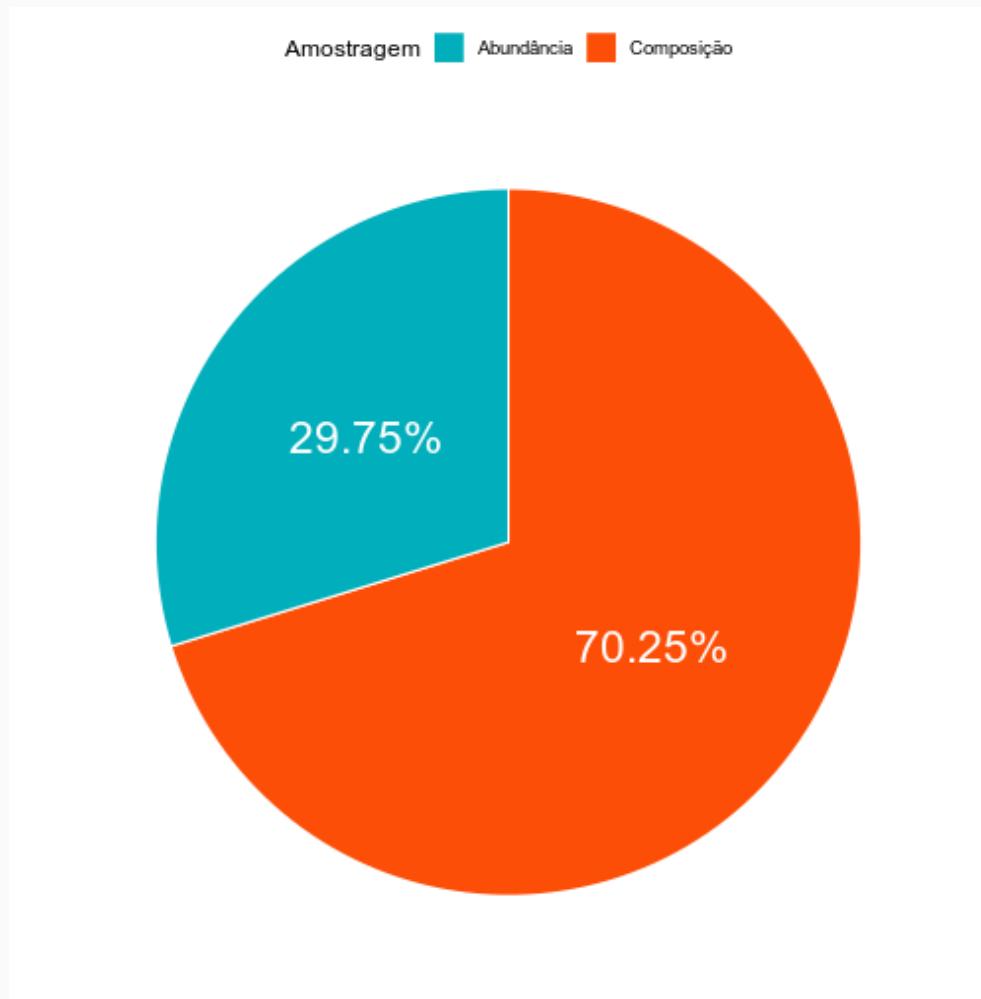
4.6 Gráfico de setores (pie chart)

ggpubr

```
# pie
ggpie(ta_por,
       "Freq",
       label = "porc",
       lab.pos = "in",
       lab.font = c(8, "white"),
       fill = "Amostragem",
       color = "white",
       palette = c("#00AFBB", "#FC4E07"))
```

4.6 Gráfico de setores (pie chart)

ggpubr



Mas todos dizem para jogar o gráfico de pizza fora e pedir uma pizza...

... e eu concordo...

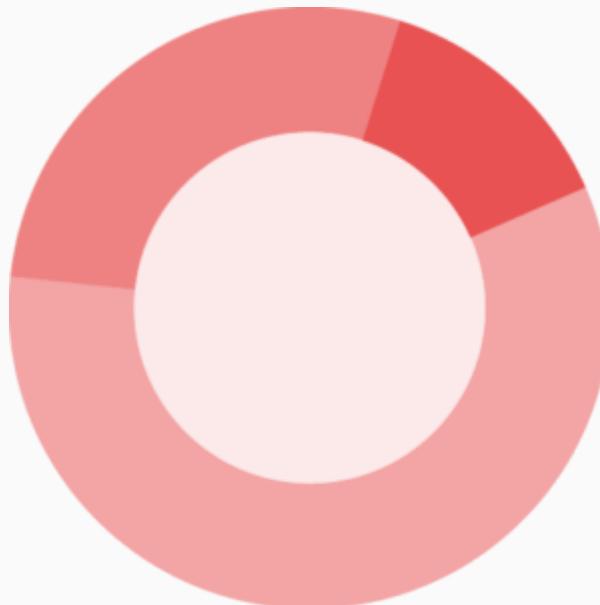
Então vamos usar algo mais saudável: vamos
retirar o recheio da pizza...

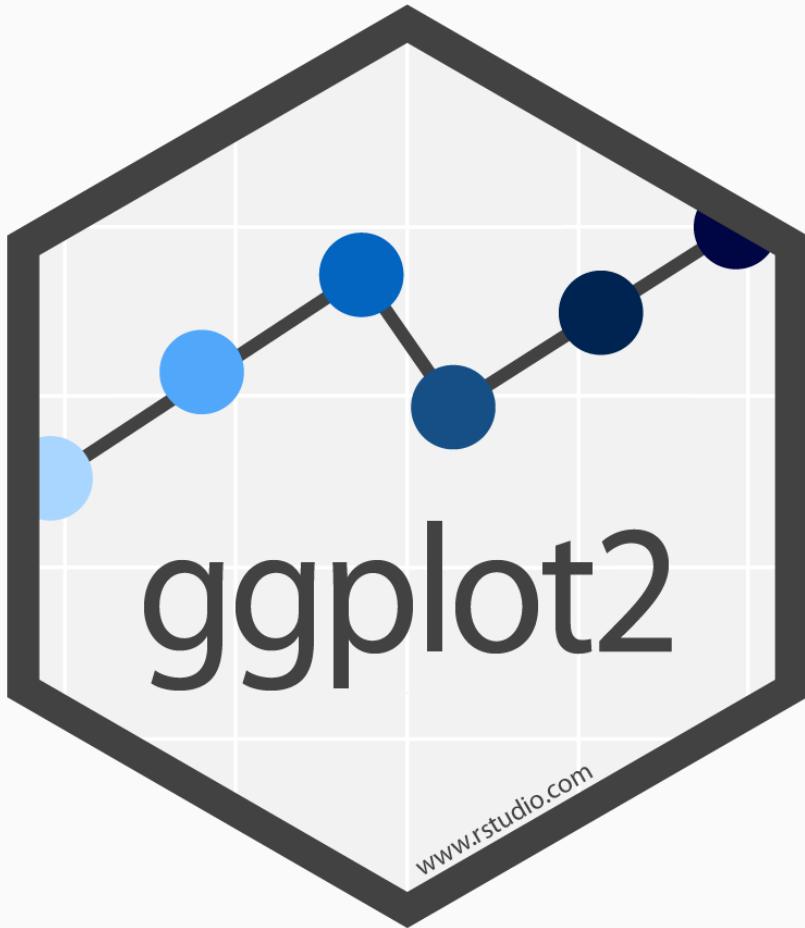
4.4 Gráfico de setores (donut chart)

Representa os dados de: uma coluna

Tipo de dado: categórico

Proporção ou porcentagem de dados categóricos





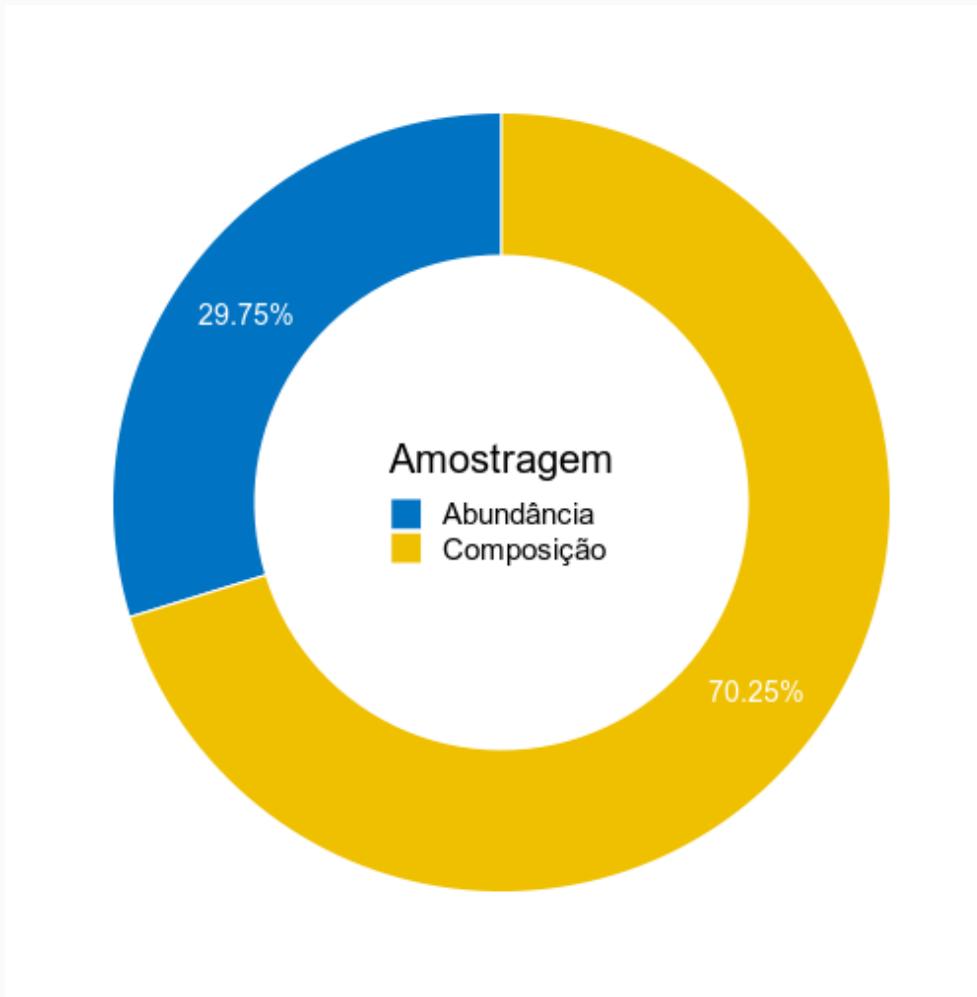
4.6 Gráfico de setores (donut chart)

ggplot2

```
# donut
ggplot(ta_por) +
  aes(x = 2, y = Freq, fill = Amostragem) +
  geom_bar(stat = "identity", color = "white") +
  xlim(0, 2.5) +
  coord_polar(theta = "y", start = 0) +
  geom_text(aes(label = porc), color = "white",
            position = position_stack(vjust = 0.5), size = 5) +
  scale_fill_manual(values = c(c("#0073C2FF", "#EFC000FF"))) +
  theme_void() +
  theme(legend.position = c(.5, .5),
        legend.title = element_text(size = 20),
        legend.text = element_text(size = 15))
```

4.6 Gráfico de setores (donut chart)

ggplot2





GGPUBR
Publication-
ready plots

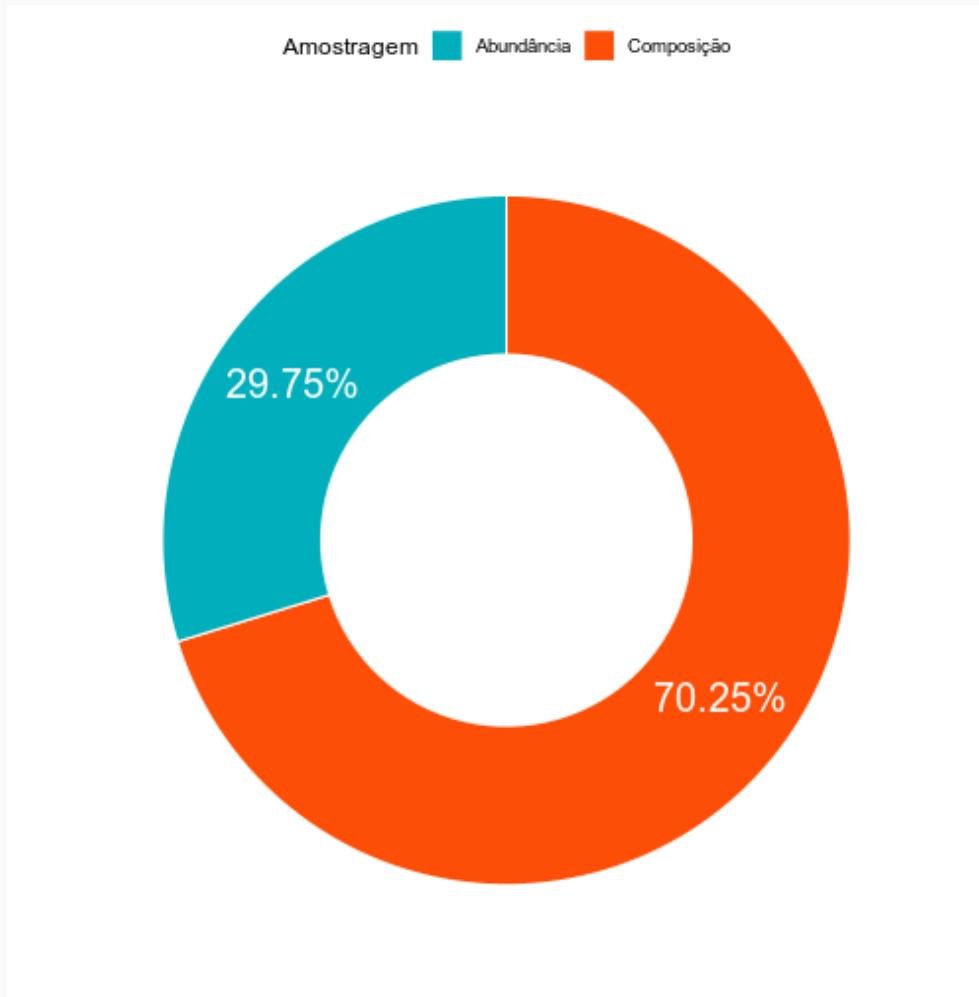
4.6 Gráfico de setores (donut chart)

ggpubr

```
# pie
ggdonutchart(ta_por,
  "Freq",
  label = "porc",
  lab.pos = "in",
  lab.font = c(7, "white"),
  fill = "Amostragem",
  color = "white",
  palette = c("#00AFBB", "#FC4E07"))
```

4.6 Gráfico de setores (donut chart)

ggpubr



Dúvidas?

4.7 Gráfico de barras (bar plot)

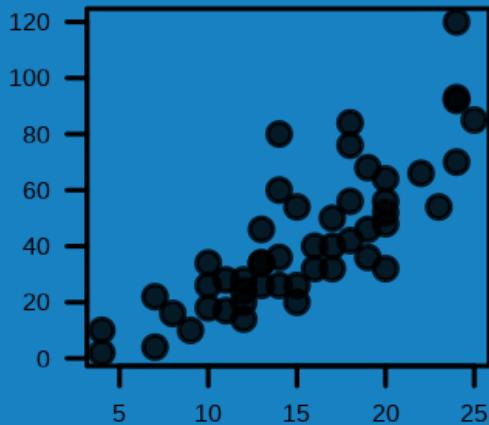
Representa os dados de: duas colunas

Modo das colunas: X = categórico e Y = categórico

Resume dados de contagens para uma coluna com diversos tipos de dados



graphics



4.7 Gráfico de barras (bar plot)

Tabela de frequência

```
ta <- table(da$record)
names(ta) <- c("Abundância", "Composição")
ta
```

```
## Abundância Composição
##           346          817
```

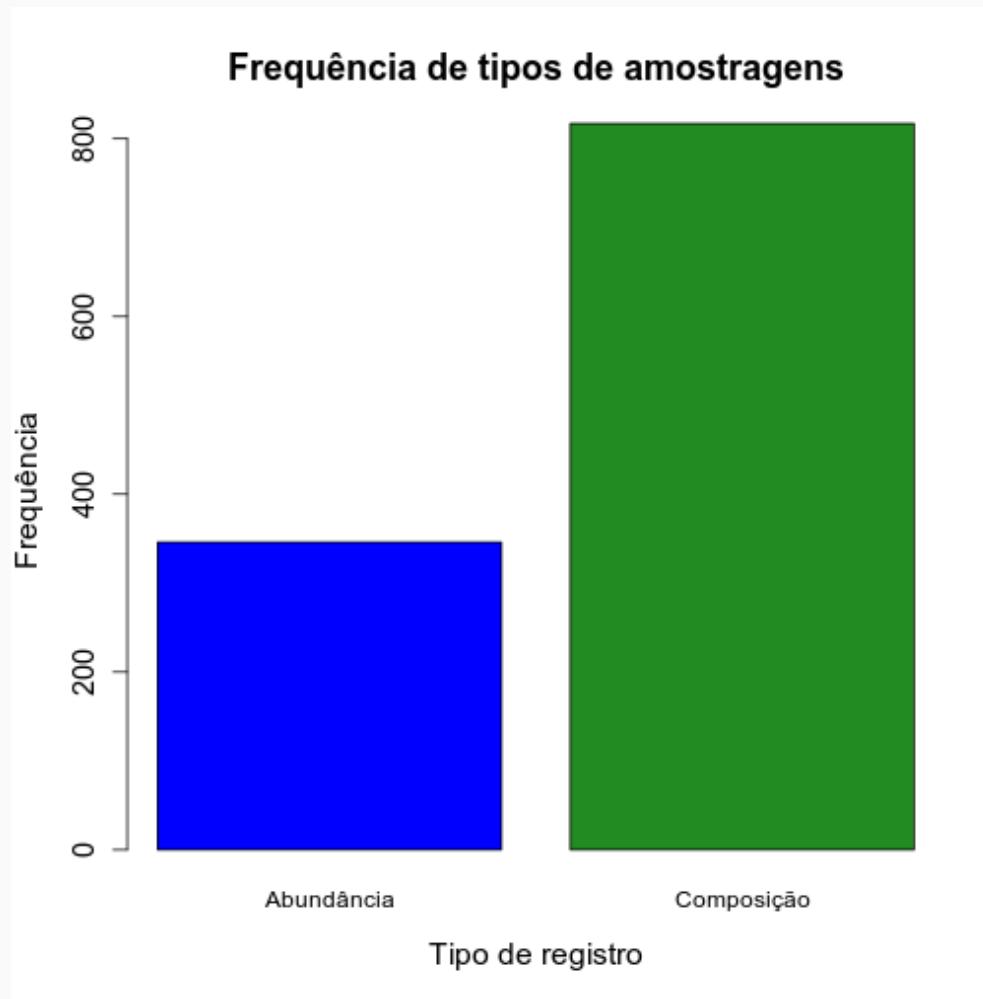
4.7 Gráfico de barras (bar plot)

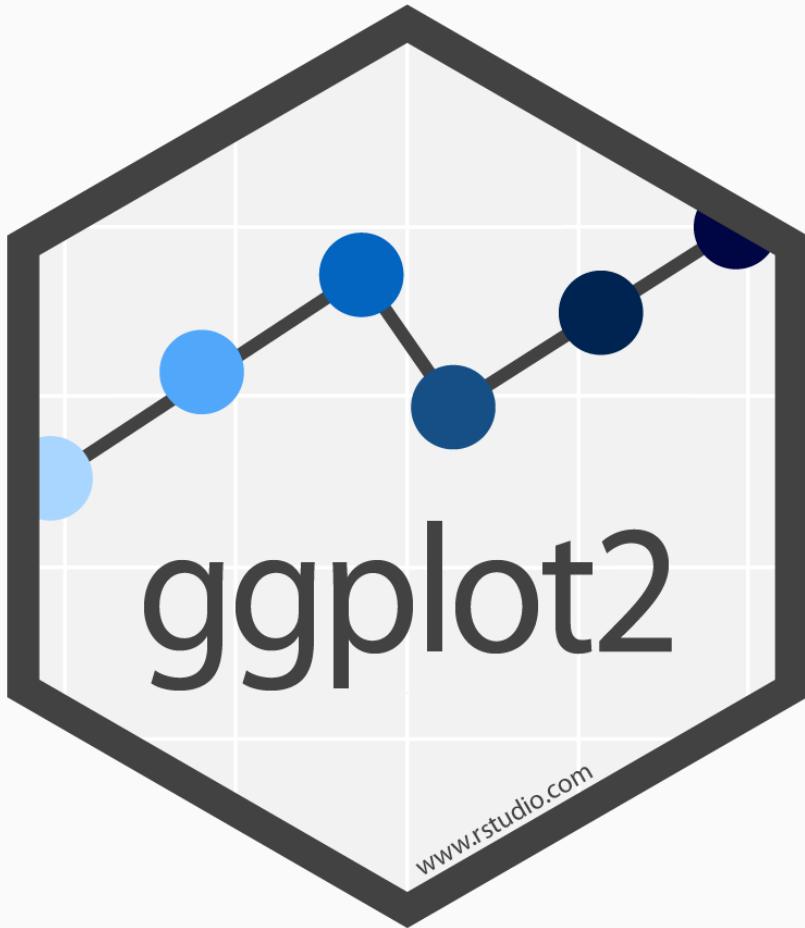
graphics

```
barplot(ta,
        col = c("blue", "forest green"),
        main = "Frequênciade tipos de amostragens",
        xlab = "Tipo de registro",
        ylab = "Frequênciia",
        cex.main = 1.5,
        cex.lab = 1.3,
        cex.axis = 1.2)
```

4.7 Gráfico de barras (bar plot)

graphics





4.7 Gráfico de barras (bar plot)

Tabela de frequênci

```
# tabela de frequencia
ta_por <- ta %>%
  as.data.frame
colnames(ta_por) <- c("record", "freq")
ta_por
```

```
##          record freq
## 1 Abundância   346
## 2 Composição  817
```

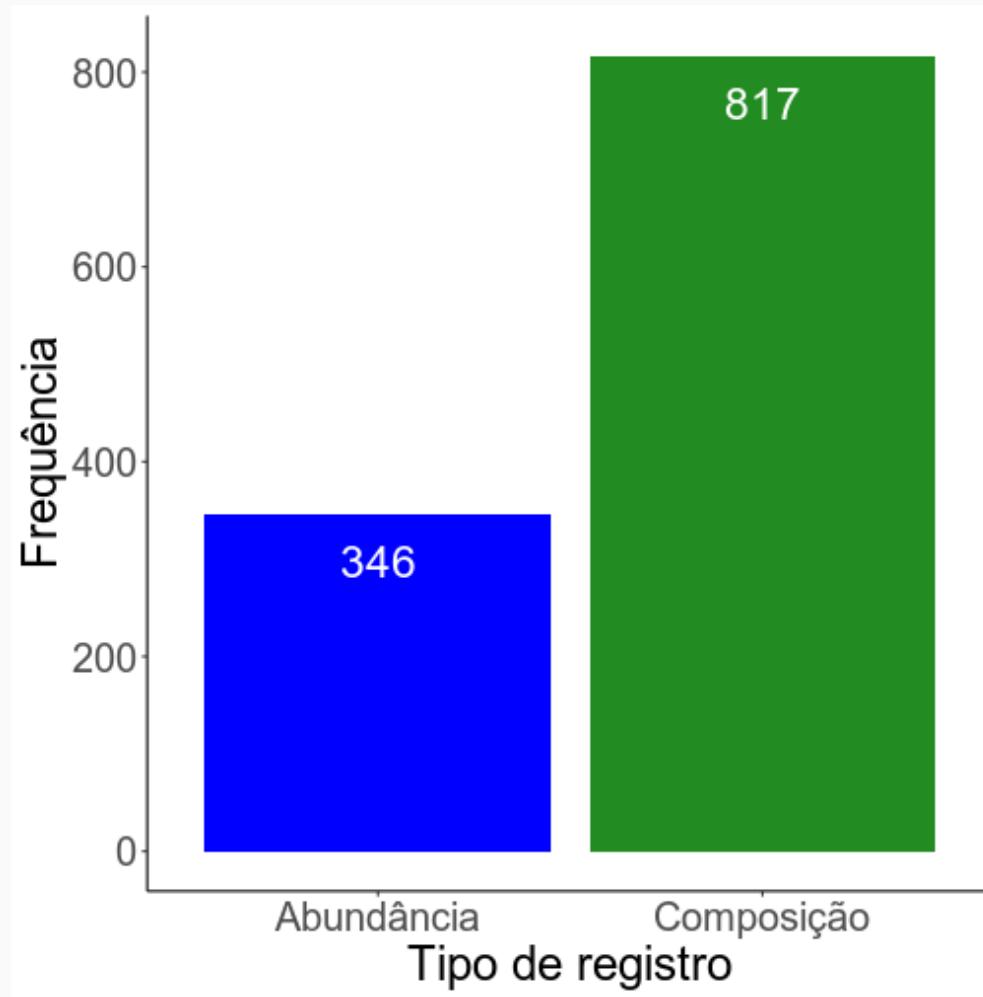
4.7 Gráfico de barras (bar plot)

ggplot2

```
ggplot(data = ta_por) +  
  aes(x = record, y = freq) +  
  geom_bar(fill = c("blue", "forest green"), stat = "identity") +  
  geom_text(aes(x = record, y = freq, label = freq), size = 8, color = "white")  
  labs(x = "Tipo de registro",  
       y = "Frequênciā") +  
  theme_classic() +  
  theme(axis.title = element_text(size = 24),  
        axis.text.x = element_text(size = 20),  
        axis.text.y = element_text(size = 20))
```

4.7 Gráfico de barras (bar plot)

ggplot2





GGPUBR
Publication-
ready plots

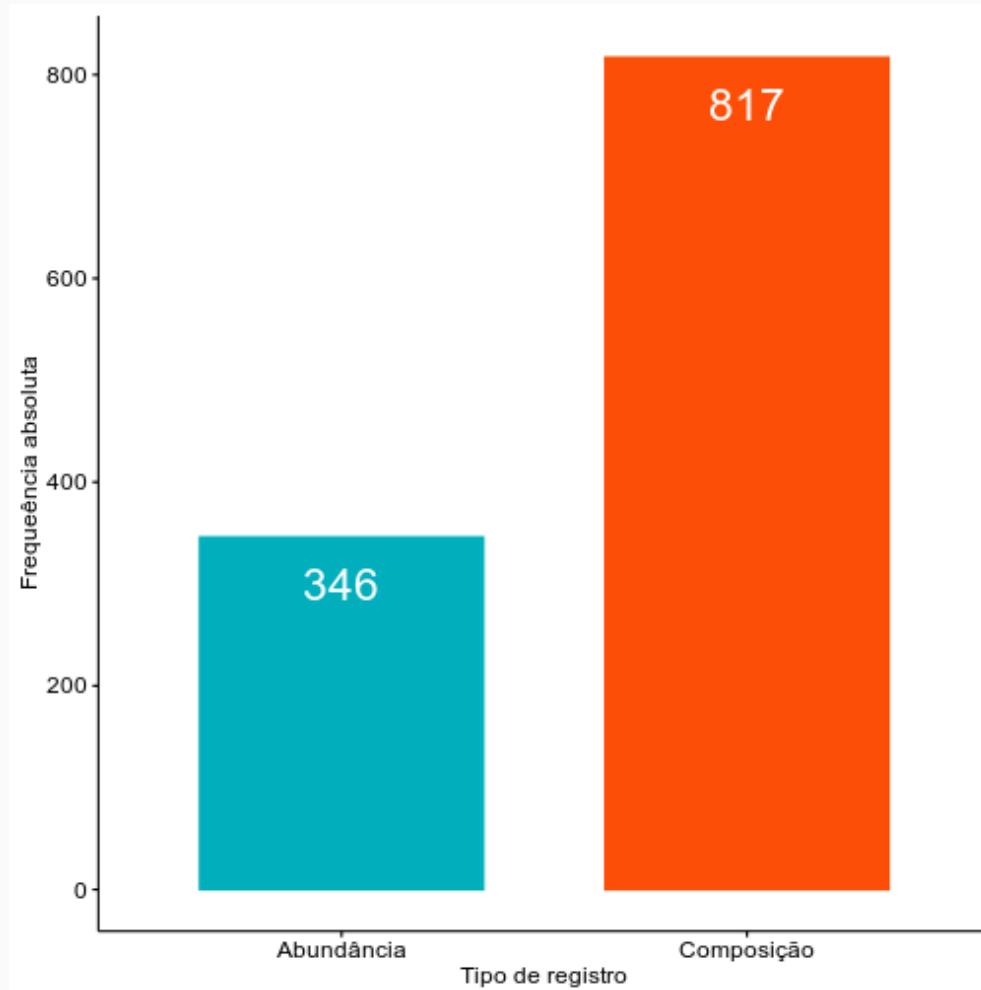
4.7 Gráfico de barras (bar plot)

ggpubr

```
ggbarplot(ta_por,
           x = "record",
           y = "freq",
           fill = "record",
           color = "record",
           palette = c("#00AFBB", "#FC4E07"),
           label = TRUE,
           lab.pos = "in",
           lab.col = "white",
           lab.size = 8,
           xlab = "Tipo de registro",
           ylab = "Frequeênci a absoluta",
           legend = "none")
```

4.7 Gráfico de barras (bar plot)

ggpubr



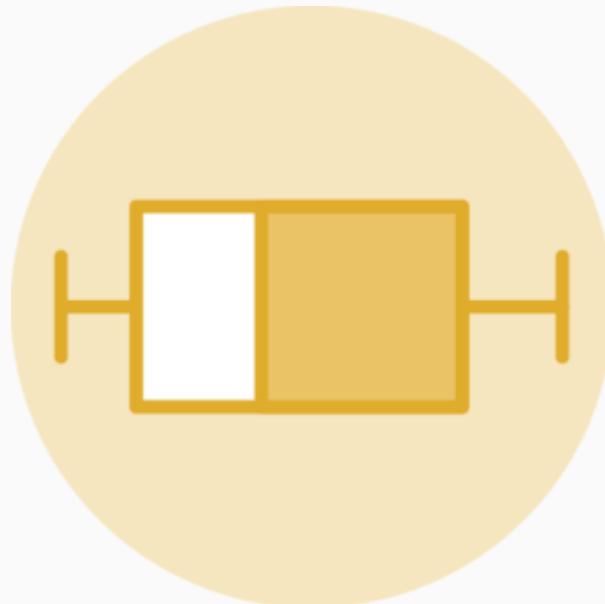
Dúvidas?

4.8 Gráfico de caixa (box plot)

Representa os dados de: duas colunas

Modo das colunas: X = categórico e Y = contínuo

Resume informações de medidas contínuas para dois ou mais fatores categóricos

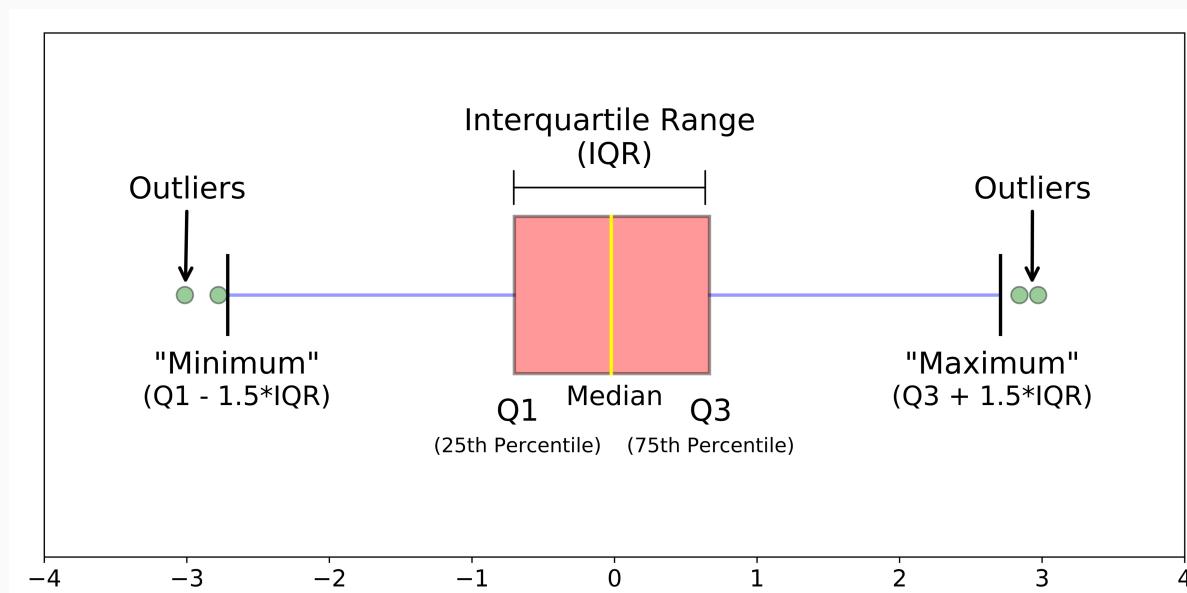


4.8 Gráfico de caixa (box plot)

Intervalo inter-quartil (*interquartile range* - IQR)

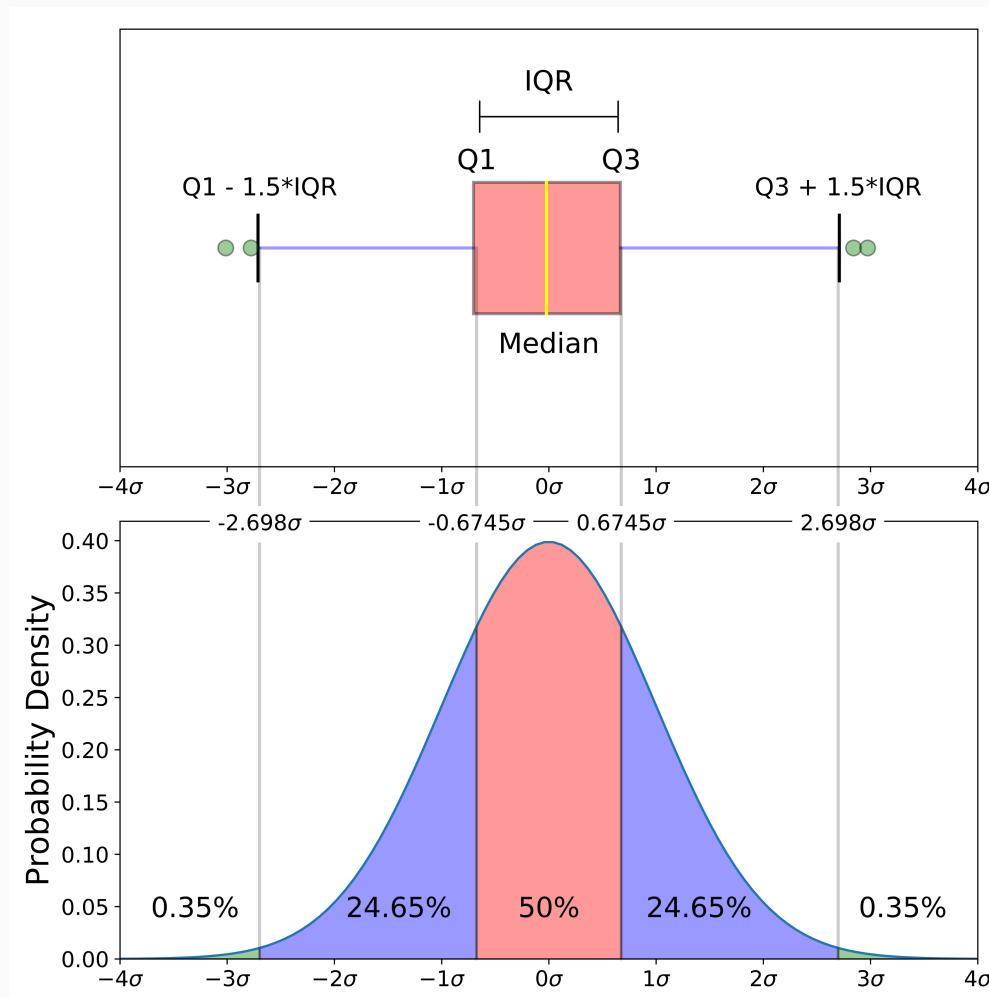
Límite inferior e limite superior ($1.5 \times \text{IQR}$)

Valores exteriores (*outliers*)



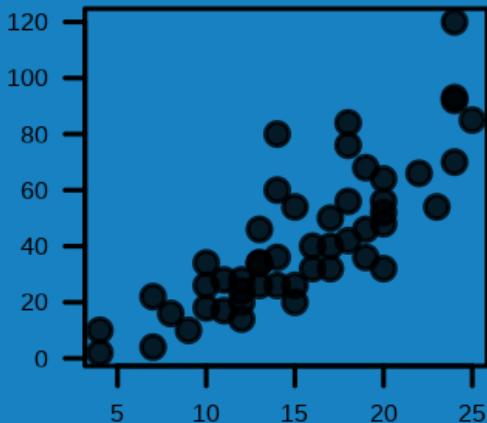
[*] <https://towardsdatascience.com/understanding-boxplots-5e2df7bcd51>

4.8 Gráfico de caixa (box plot)



[*] <https://towardsdatascience.com/understanding-boxplots-5e2df7bcd51>

graphics



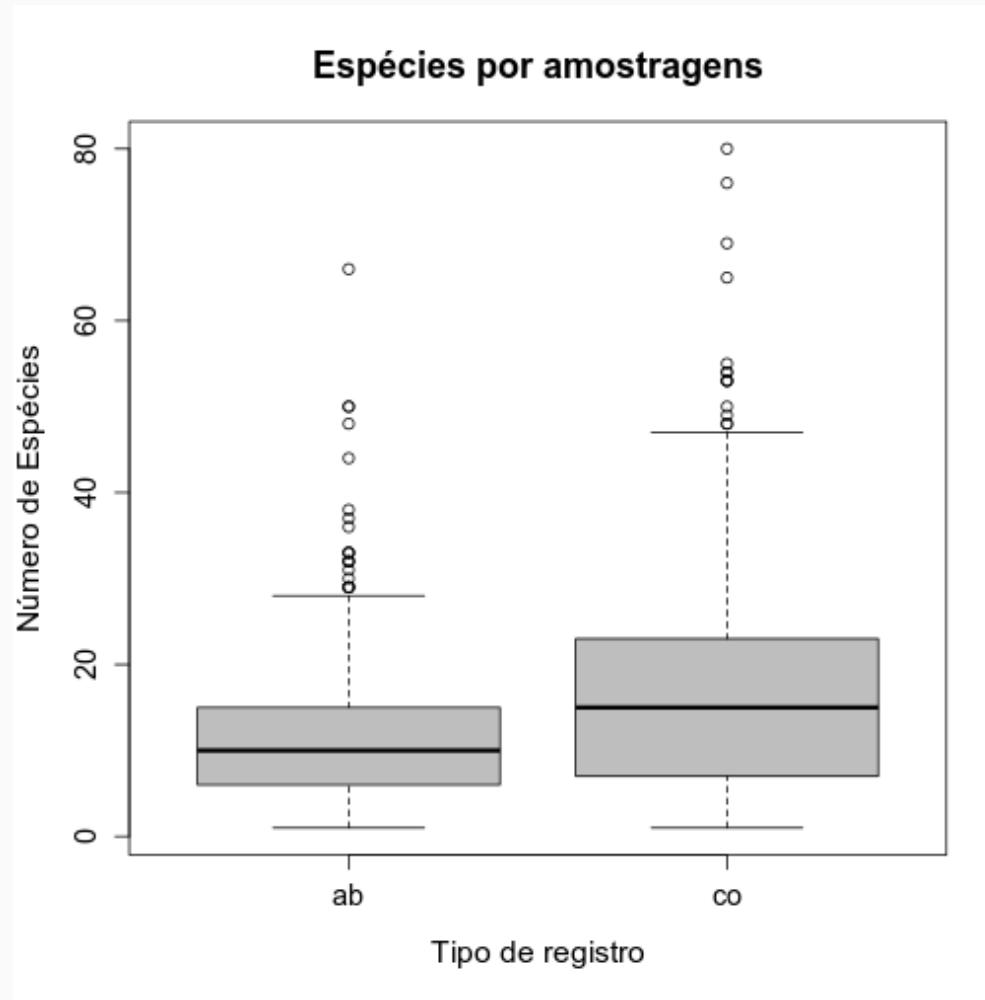
4.8 Gráfico de caixa (box plot)

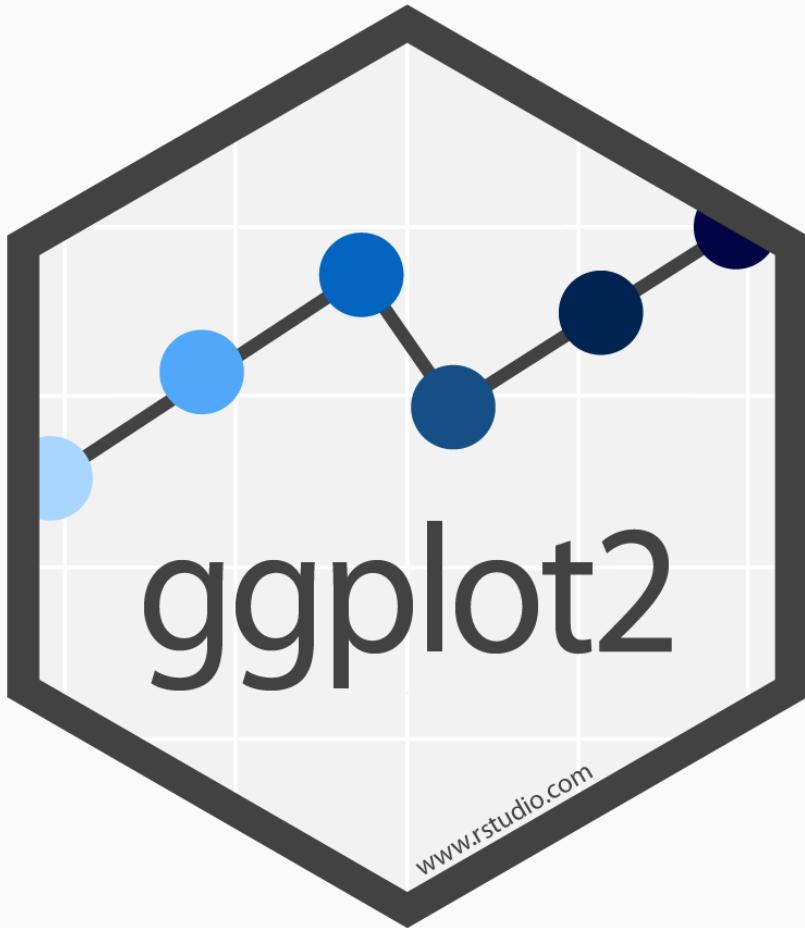
graphics

```
boxplot(species_number ~ as.factor(record),  
        data = da,  
        col = "gray",  
        border = "black",  
        main = "Espécies por amostragens",  
        xlab = "Tipo de registro",  
        ylab = "Número de Espécies",  
        cex.main = 1.5,  
        cex.lab = 1.3,  
        cex.axis = 1.2)
```

4.8 Gráfico de caixa (box plot)

graphics





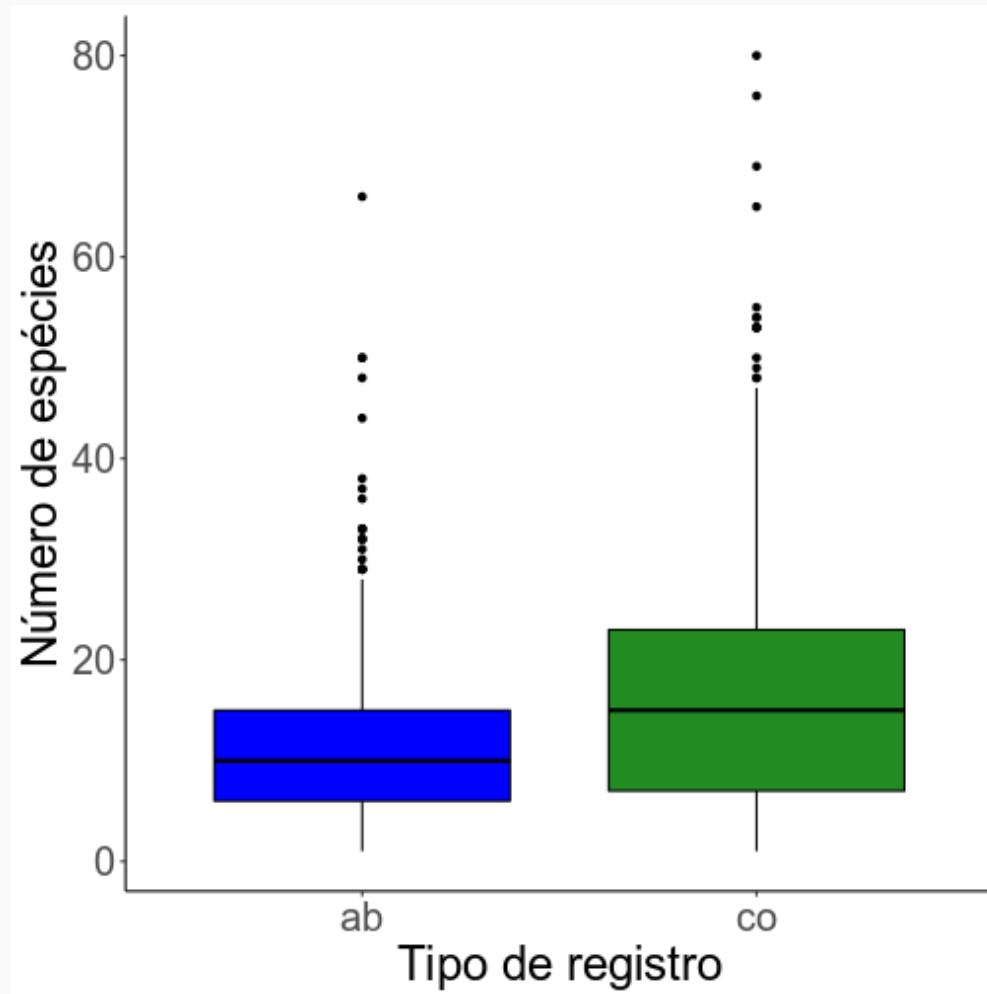
4.8 Gráfico de caixa (box plot)

ggplot2

```
ggplot(data = da) +
  aes(x = record, y = species_number) +
  geom_boxplot(fill = c("blue", "forest green"), color = "black") +
  labs(x = "Tipo de registro",
       y = "Número de espécies") +
  theme_classic() +
  theme(axis.title = element_text(size = 24),
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 20))
```

4.8 Gráfico de caixa (box plot)

ggplot2



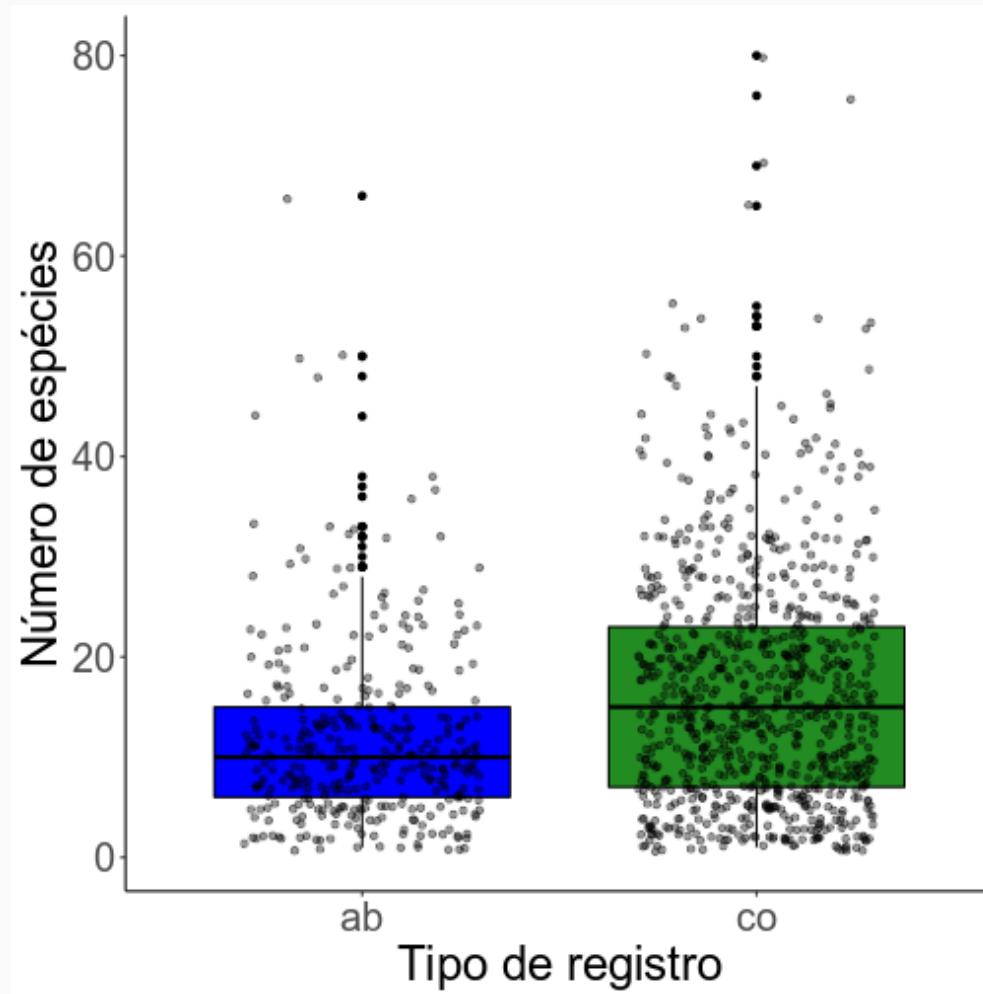
4.8 Gráfico de caixa (box plot)

ggplot2

```
ggplot(data = da) +
  aes(x = record, y = species_number) +
  geom_boxplot(fill = c("blue", "forest green"), color = "black") +
  geom_jitter(width = .3, alpha = .4) +
  labs(x = "Tipo de registro",
       y = "Número de espécies") +
  theme_classic() +
  theme(axis.title = element_text(size = 24),
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 20))
```

4.8 Gráfico de caixa (box plot)

ggplot2



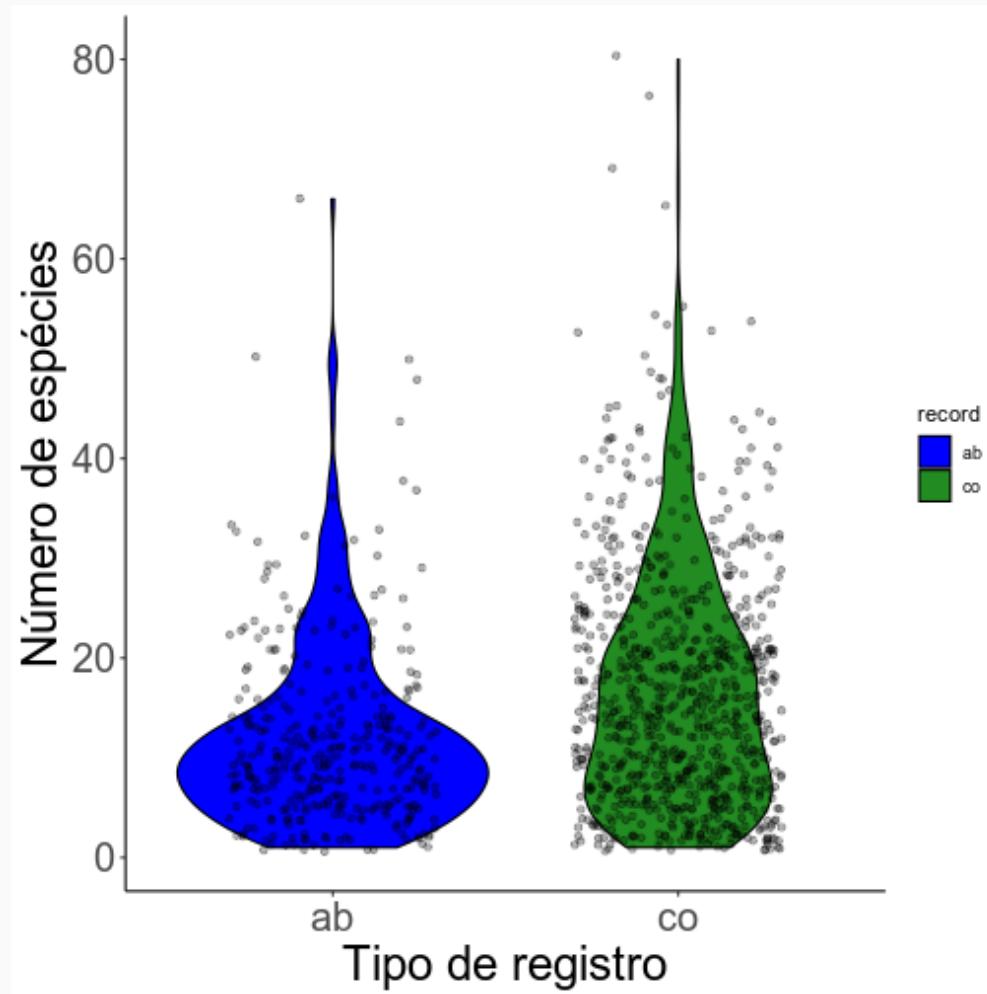
4.7 Gráfico de violino (violin plot)

ggplot2

```
ggplot(data = da) +
  aes(x = record, y = species_number) +
  geom_violin(aes(fill = record), color = "black") +
  scale_fill_manual(values = c("blue", "forest green")) +
  geom_jitter(width = .3, alpha = .3) +
  labs(x = "Tipo de registro",
       y = "Número de espécies") +
  theme_classic() +
  theme(axis.title = element_text(size = 24),
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 20))
```

4.7 Gráfico de violino (violin plot)

ggplot2





GGPUBR
Publication-
ready plots

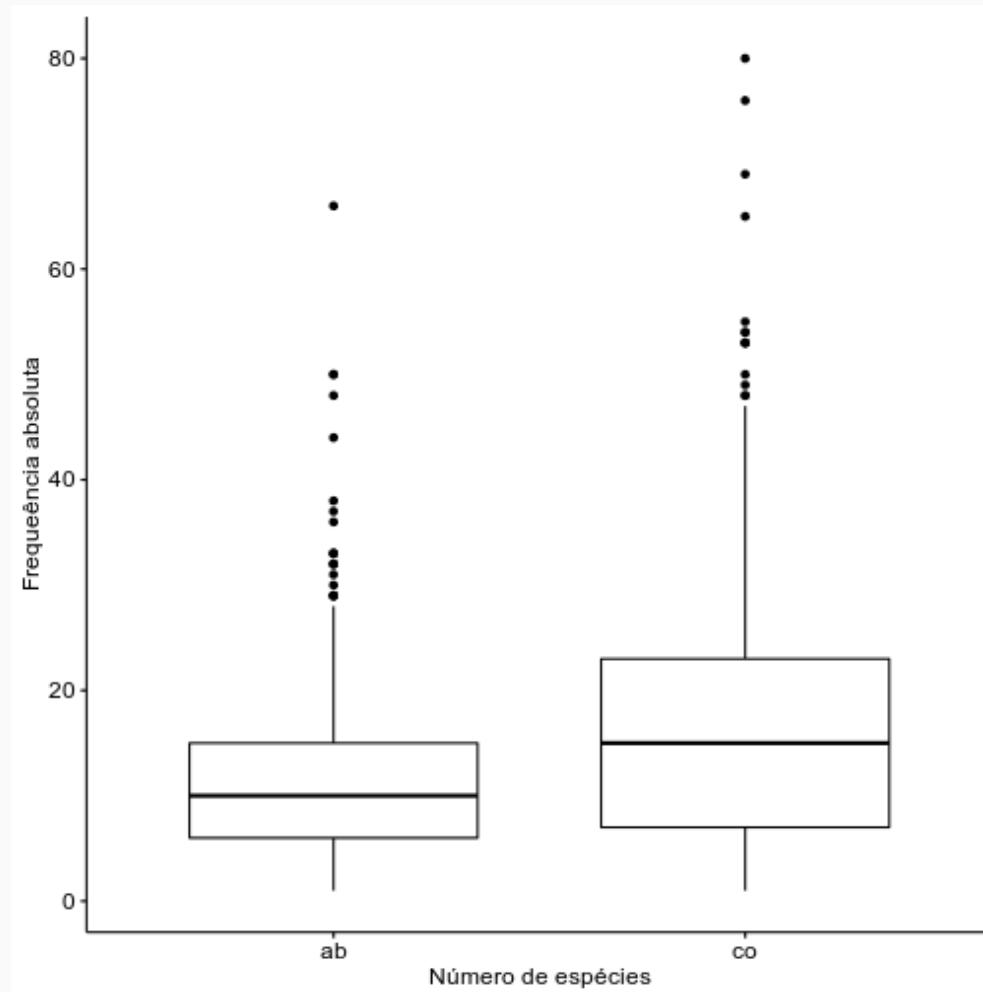
4.8 Gráfico de caixa (box plot)

ggpubr

```
ggboxplot(data = da,  
          x = "record",  
          y = "species_number",  
          xlab = "Número de espécies",  
          ylab = "Frequência absoluta")
```

4.8 Gráfico de caixa (box plot)

ggpubr



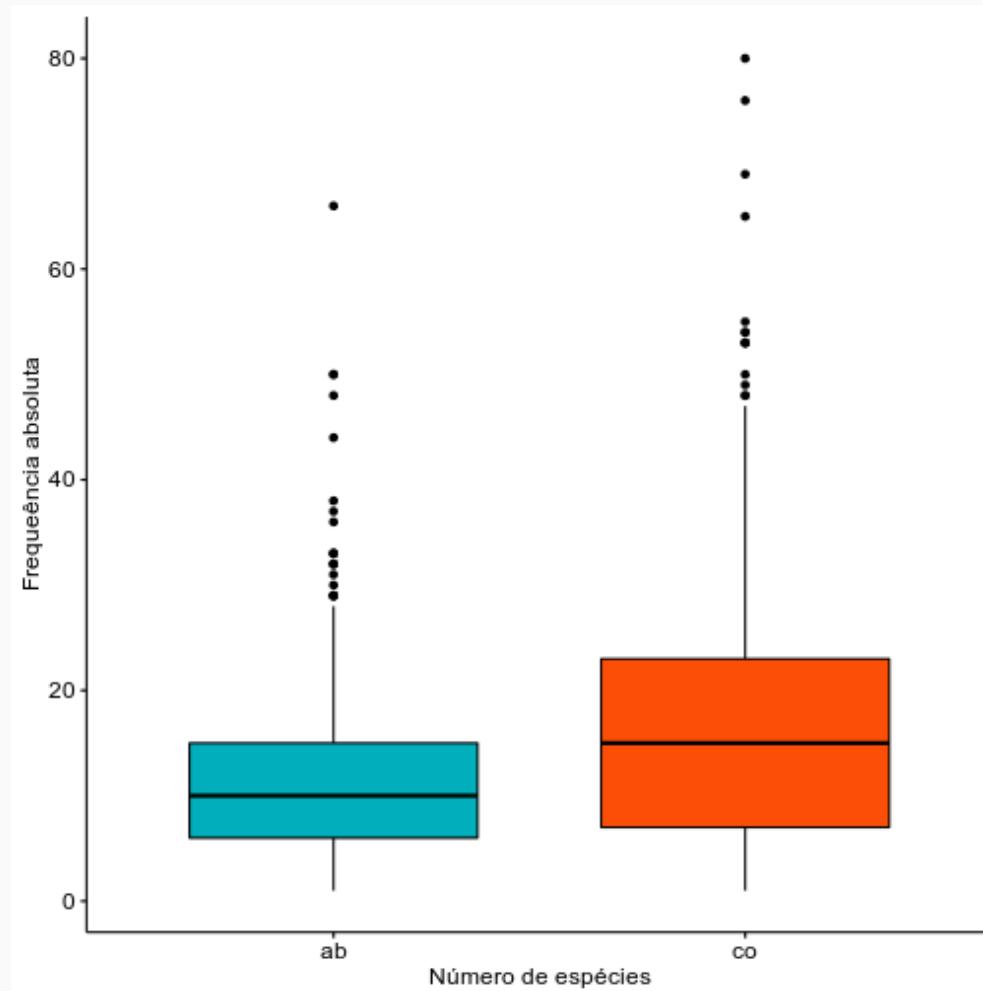
4.8 Gráfico de caixa (box plot)

ggpubr

```
ggboxplot(data = da,
           x = "record",
           y = "species_number",
           fill = "record",
           palette = c("#00AFBB", "#FC4E07"),
           xlab = "Número de espécies",
           ylab = "Frequência absoluta",
           legend = "none")
```

4.8 Gráfico de caixa (box plot)

ggpubr



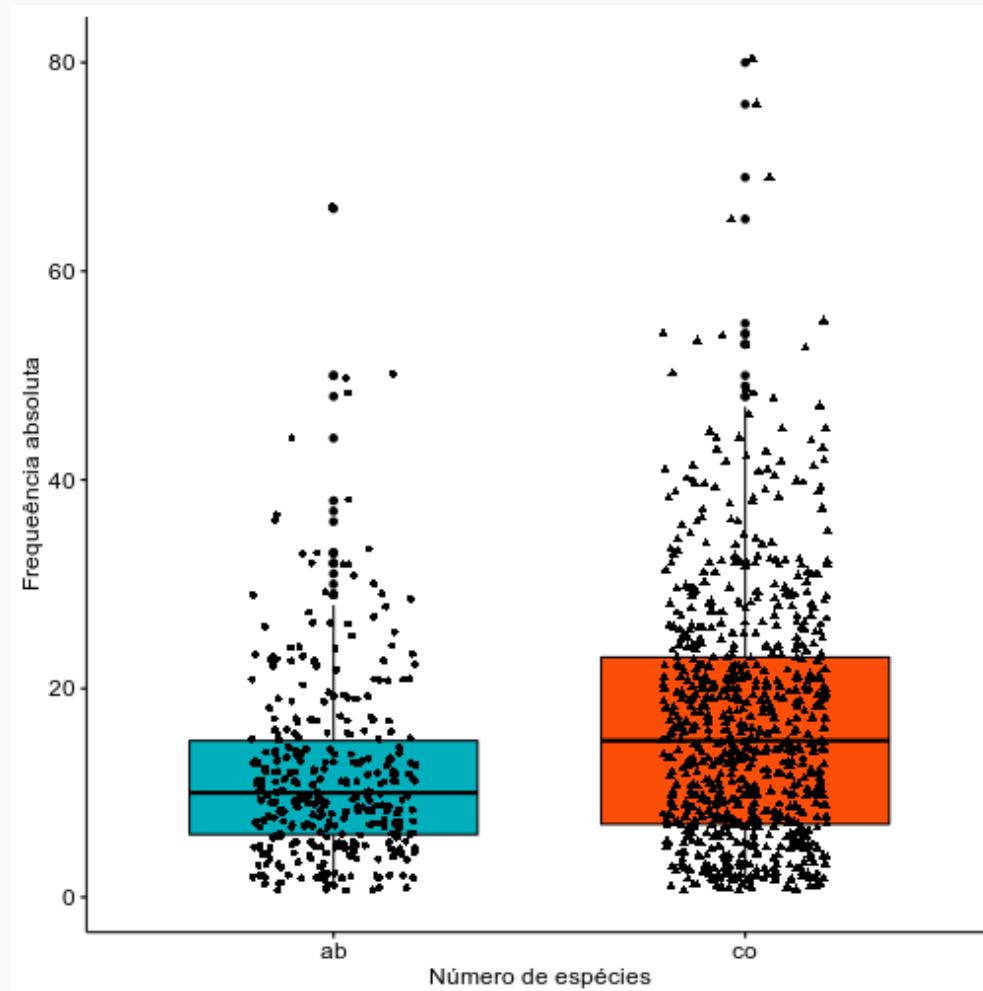
4.8 Gráfico de caixa (box plot)

ggpubr

```
ggboxplot(data = da,
           x = "record",
           y = "species_number",
           add = "jitter",
           shape = "record",
           fill = "record",
           color = "black",
           palette = c("#00AFBB", "#FC4E07"),
           xlab = "Número de espécies",
           ylab = "Frequeênciia absoluta",
           legend = "none")
```

4.8 Gráfico de caixa (box plot)

ggpubr



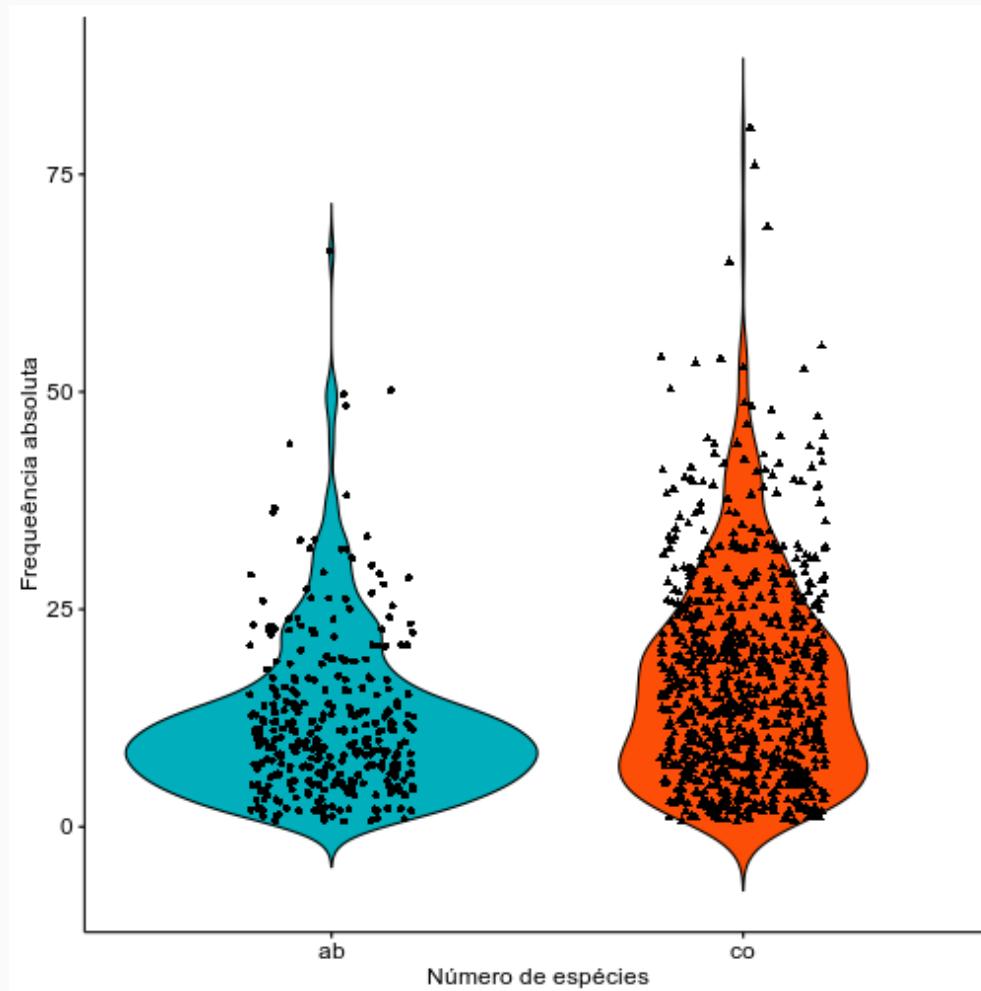
4.7 Gráfico de violino (violin plot)

ggpubr

```
ggviolin(data = da,
          x = "record",
          y = "species_number",
          add = "jitter",
          shape = "record",
          fill = "record",
          color = "black",
          palette = c("#00AFBB", "#FC4E07"),
          xlab = "Número de espécies",
          ylab = "Frequeênciia absoluta",
          legend = "none")
```

4.7 Gráfico de violino (violin plot)

ggpubr



Dúvidas?

4.9 Gráfico de dispersão

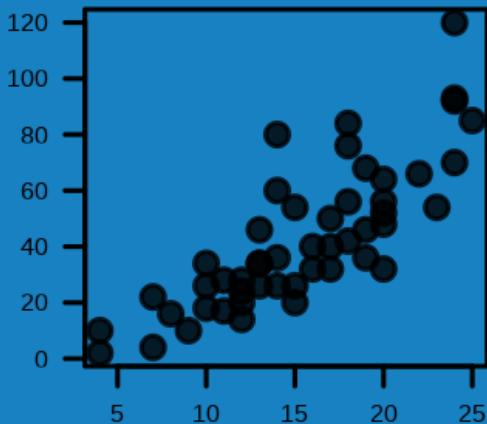
Representa os dados de: duas colunas

Modo das colunas: X = numérico e Y = numérico

Plota a relação entre duas variáveis contínuas



graphics



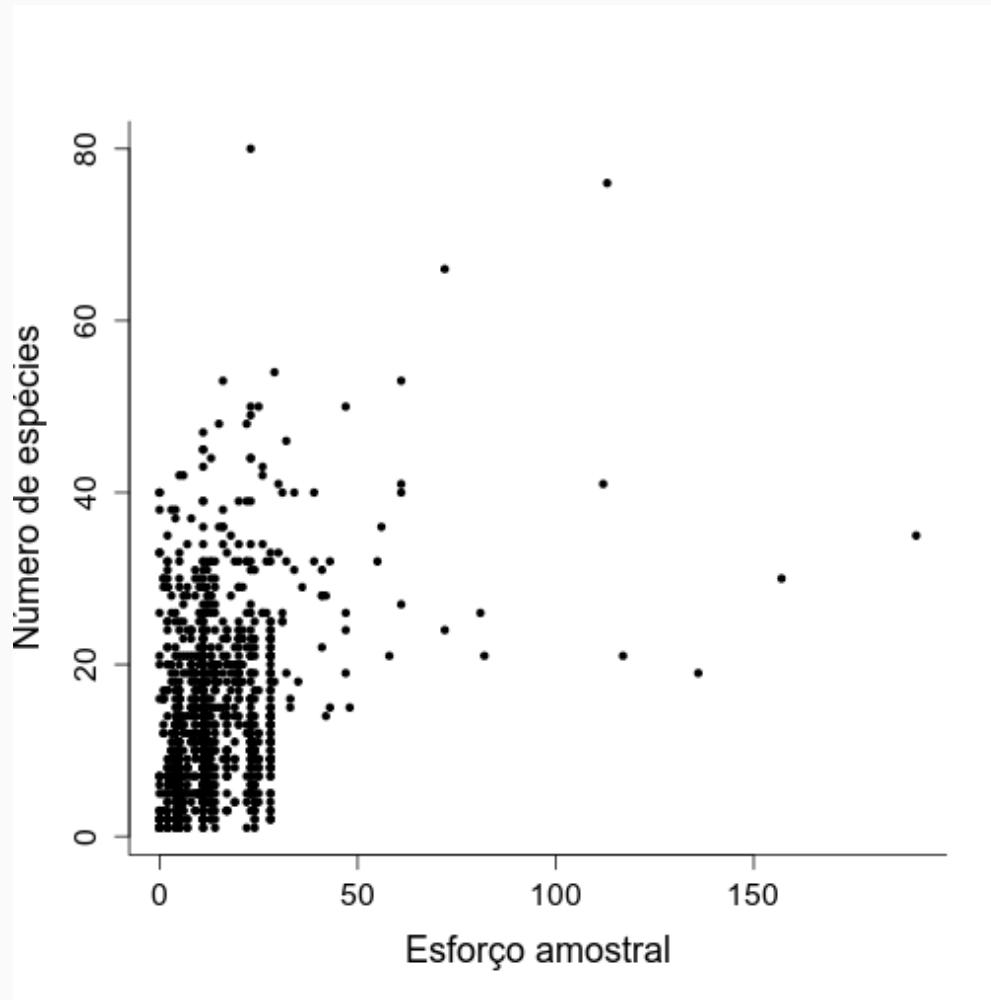
4.9 Gráfico de dispersão

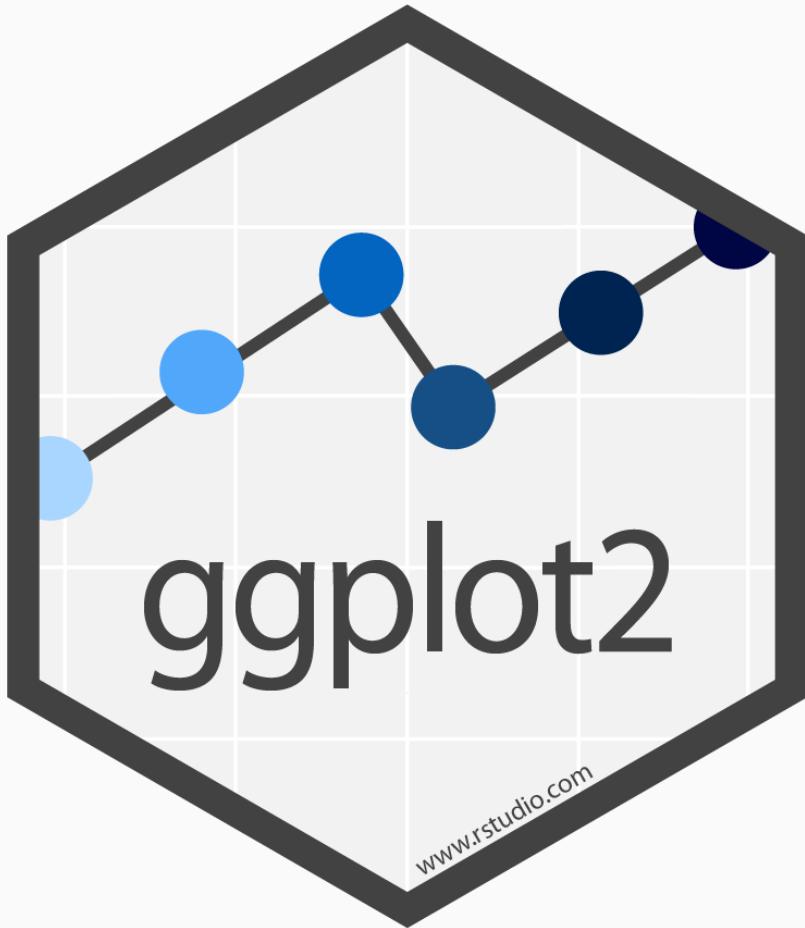
graphics

```
plot(species_number ~ effort_months,
      data = da,
      pch = 20,
      xlab = "Esforço amostral",
      ylab = "Número de espécies",
      cex.lab = 1.5,
      cex.axis = 1.3,
      bty = "l")
```

4.9 Gráfico de dispersão

graphics





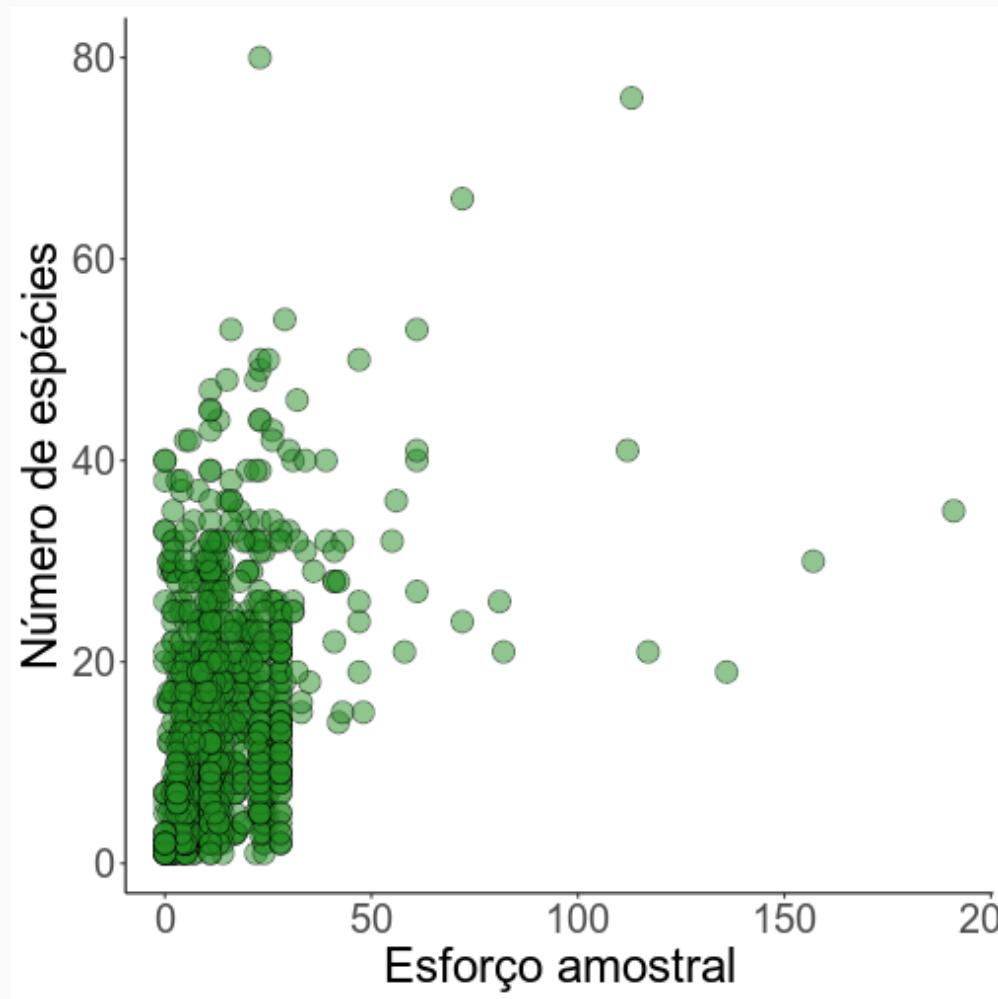
4.9 Gráfico de dispersão

ggplot2

```
ggplot(data = da) +  
  aes(x = effort_months, y = species_number) +  
  geom_point(colour = "black", fill = "forest green", size = 5,  
             alpha = .5, pch = 21) +  
  labs(x = "Esforço amostral", y = "Número de espécies") +  
  theme_classic() +  
  theme(axis.title = element_text(size = 24),  
        axis.text.x = element_text(size = 20),  
        axis.text.y = element_text(size = 20))
```

4.9 Gráfico de dispersão

ggplot2





GGPUBR
Publication-
ready plots

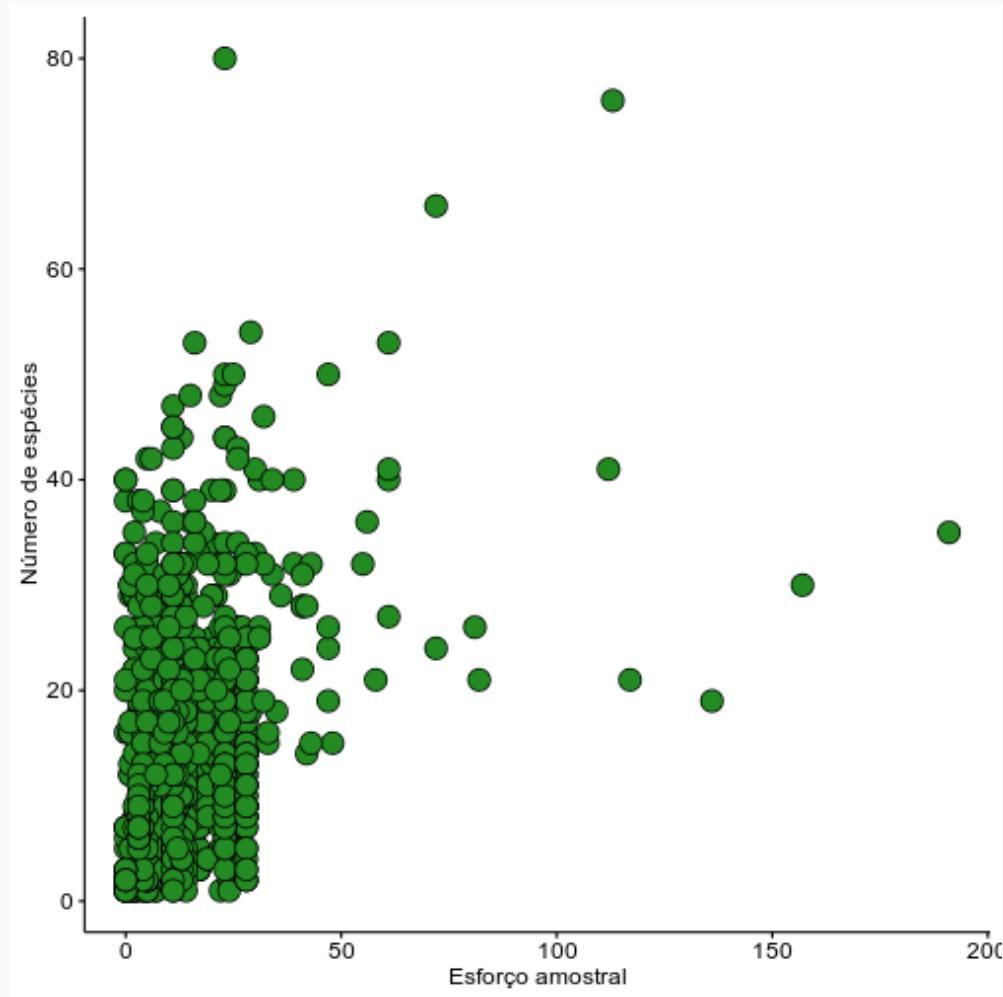
4.9 Gráfico de dispersão

ggpubr

```
ggscatter(data = da,
          x = "effort_months",
          y = "species_number",
          color = "black",
          fill = "forestgreen",
          shape = 21,
          size = 5,
          xlab = "Esforço amostral",
          ylab = "Número de espécies")
```

4.9 Gráfico de dispersão

ggpubr

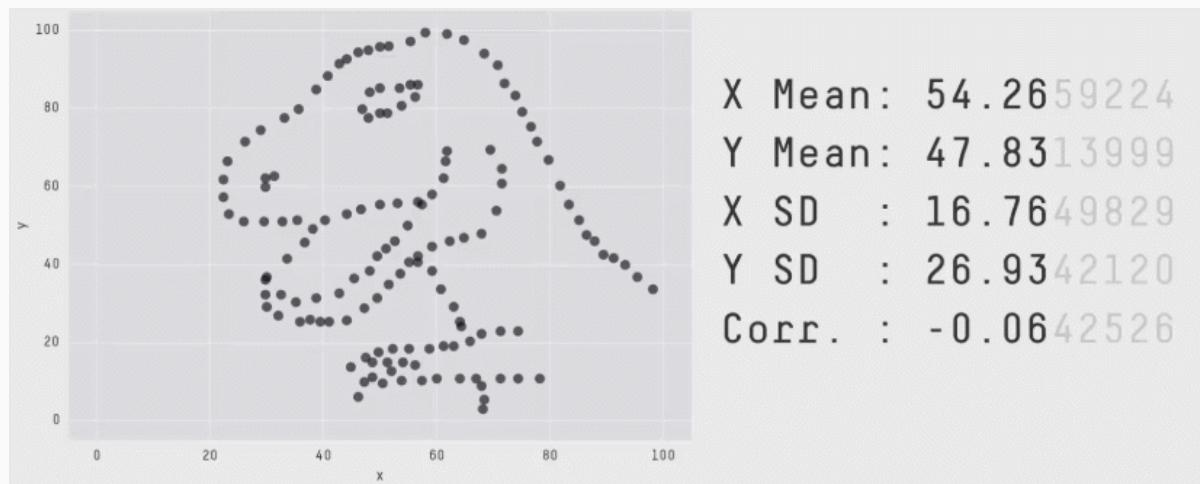


Bônus!

4.9 Gráfico de dispersão

Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing

Justin Matejka, George Fitzmaurice



[*] <https://www.autodeskresearch.com/publications/samestats>

4.9 Gráfico de dispersão

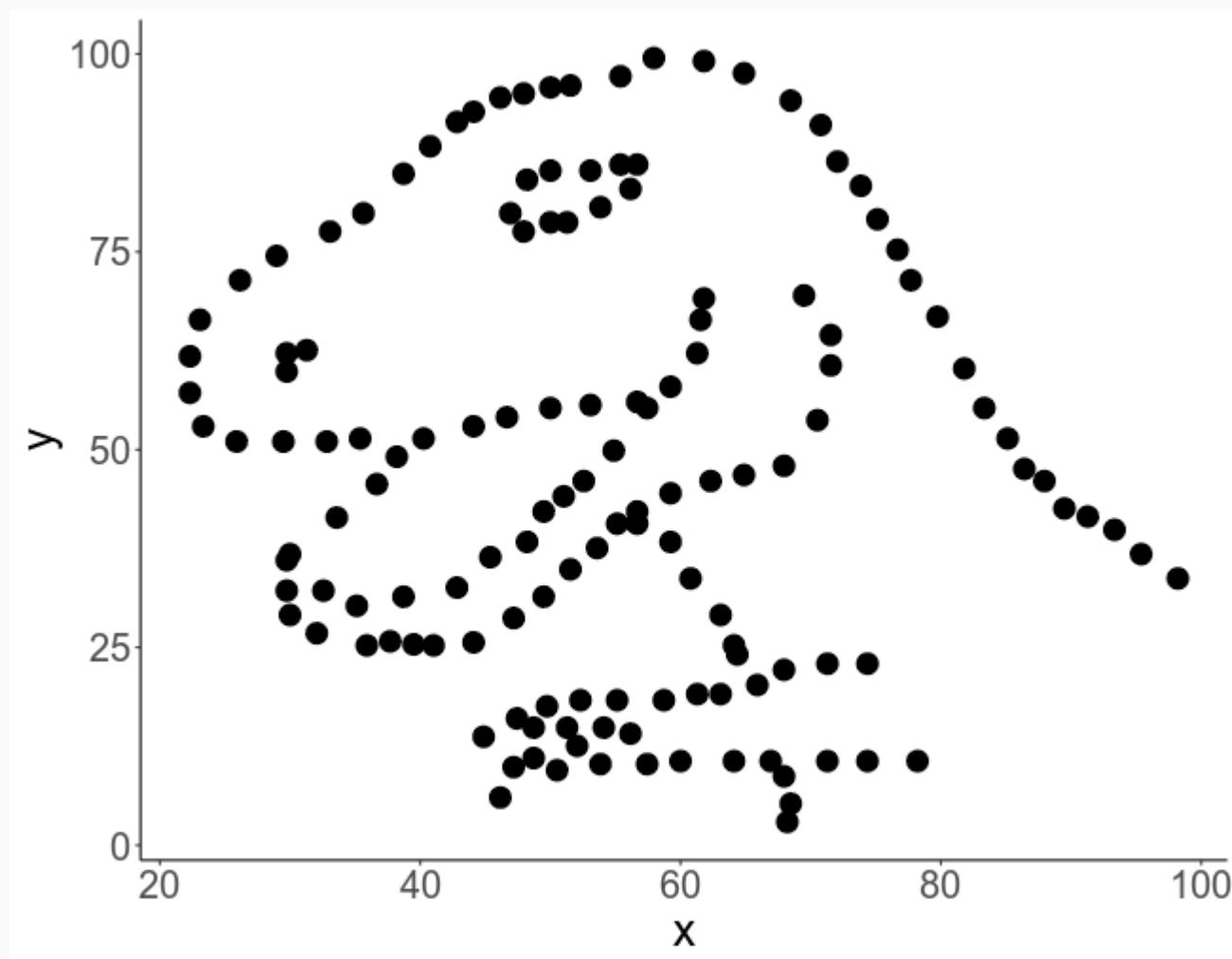
dino

```
# package
library(datasauRus)

# data + plot
datasaurus_dozen %>%
  dplyr::filter(dataset == "dino") %>%
  ggplot() +
  aes(x = x, y = y) +
  geom_point(colour = "black", fill = "black",
             size = 5, alpha = .75, pch = 21) +
  theme_classic() +
  theme(axis.title = element_text(size = 24),
        axis.text.x = element_text(size = 20),
        axis.text.y = element_text(size = 20))
```

4.9 Gráfico de dispersão

dino



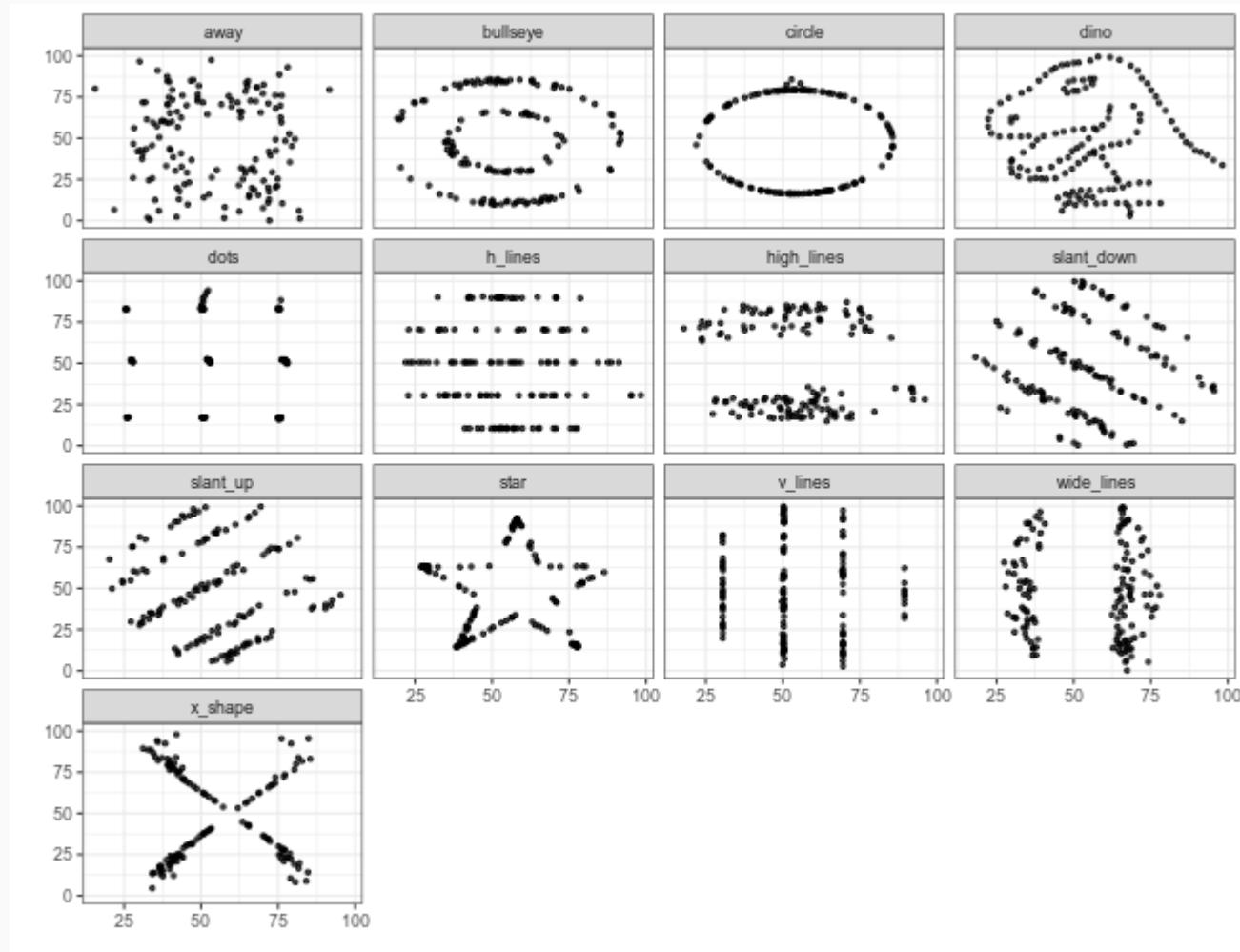
4.9 Gráfico de dispersão

All plots

```
# all plot
datasaurus_dozen %>%
  ggplot() +
  aes(x = x, y = y) +
  geom_point(colour = "black", fill = "black",
             size = 1, alpha = .75, pch = 21) +
  facet_wrap(~dataset) +
  theme_bw()
```

4.9 Gráfico de dispersão

All plots

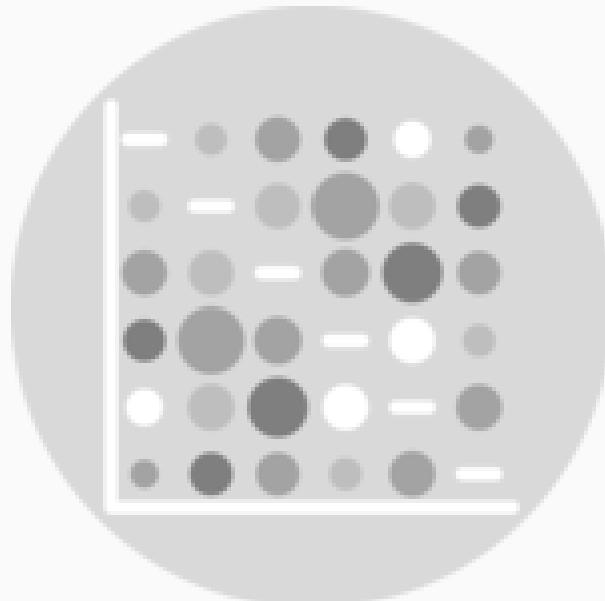


4.10 Gráfico pareado (pairs plot)

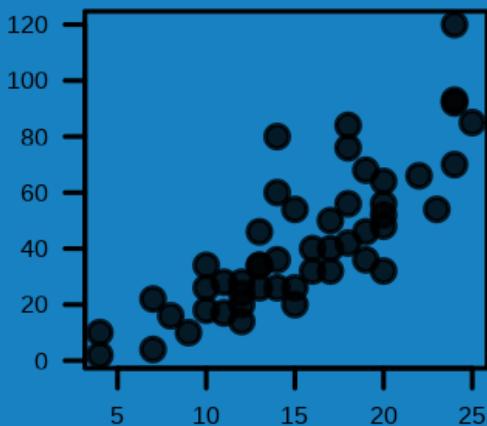
Representa os dados de: muitas colunas

Modo das colunas: X = numérico e Y = numérico

Plota a relação entre duas variáveis contínuas, mas para várias colunas



graphics



4.10 Gráfico pareado (pairs plot)

Selecionar as colunas

```
# seleccionar as colunas
da_sel <- da %>%
  select(temperature, precipitation, altitude)
da_sel
```

```
## # A tibble: 1,163 x 3
##   temperature precipitation altitude
##       <dbl>          <dbl>     <dbl>
## 1      25.0          853      543
## 2      26.5         1318      15
## 3      26.4         1248      29
## 4      26.6         1376      25
## 5      21.4         1689      750
## 6      20.4         1249      745
## 7      21.6         1520      863
## 8      21.5         1474      878
## 9      21.8         1450      826
## 10     25.4         1332      93
```

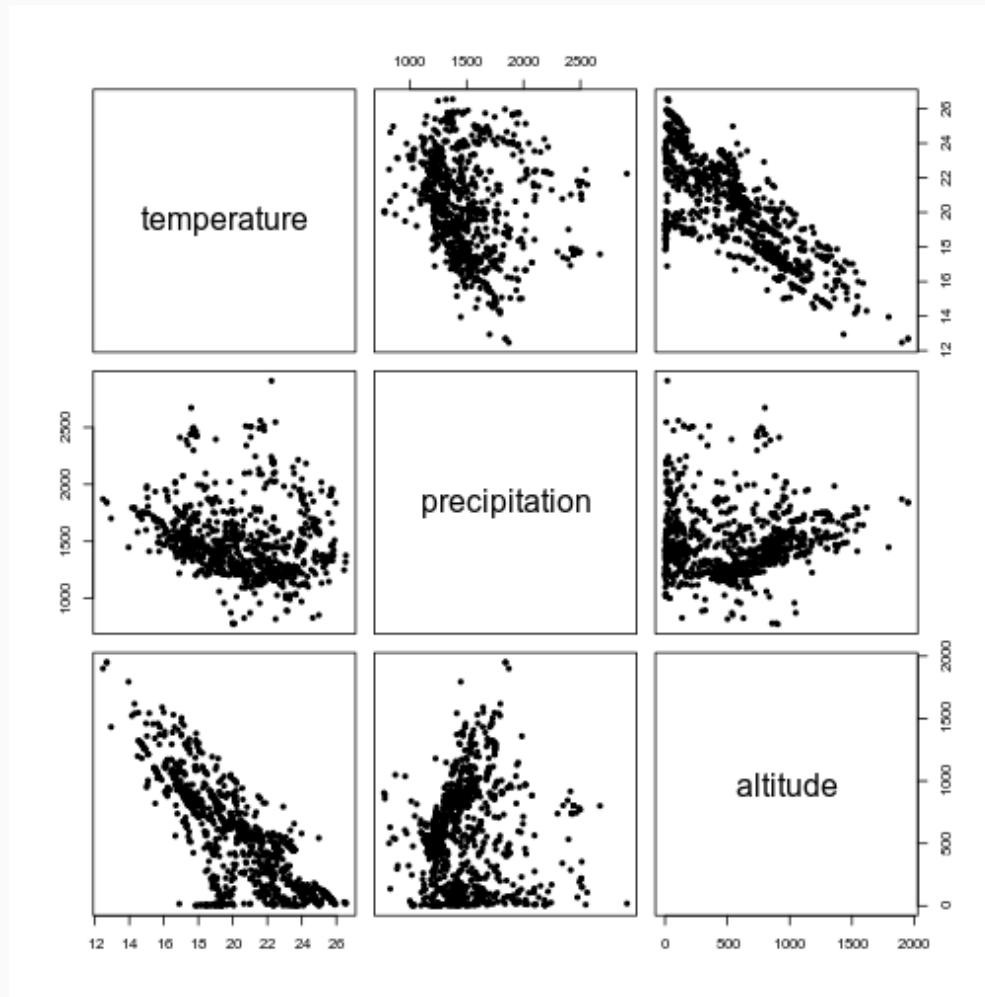
4.10 Gráfico pareado (pairs plot)

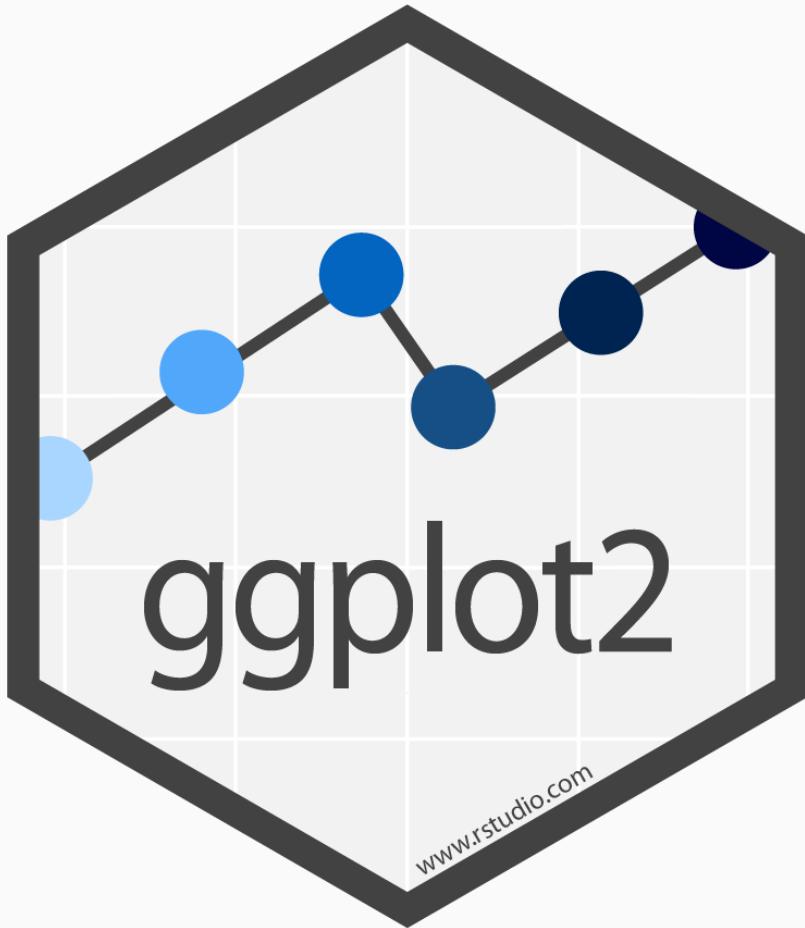
graphics

```
# pairs plot
pairs(da_sel,
      pch = 20,
      upper.panel = NULL)
```

4.10 Gráfico pareado (pairs plot)

graphics





4.10 Gráfico pareado (pairs plot)

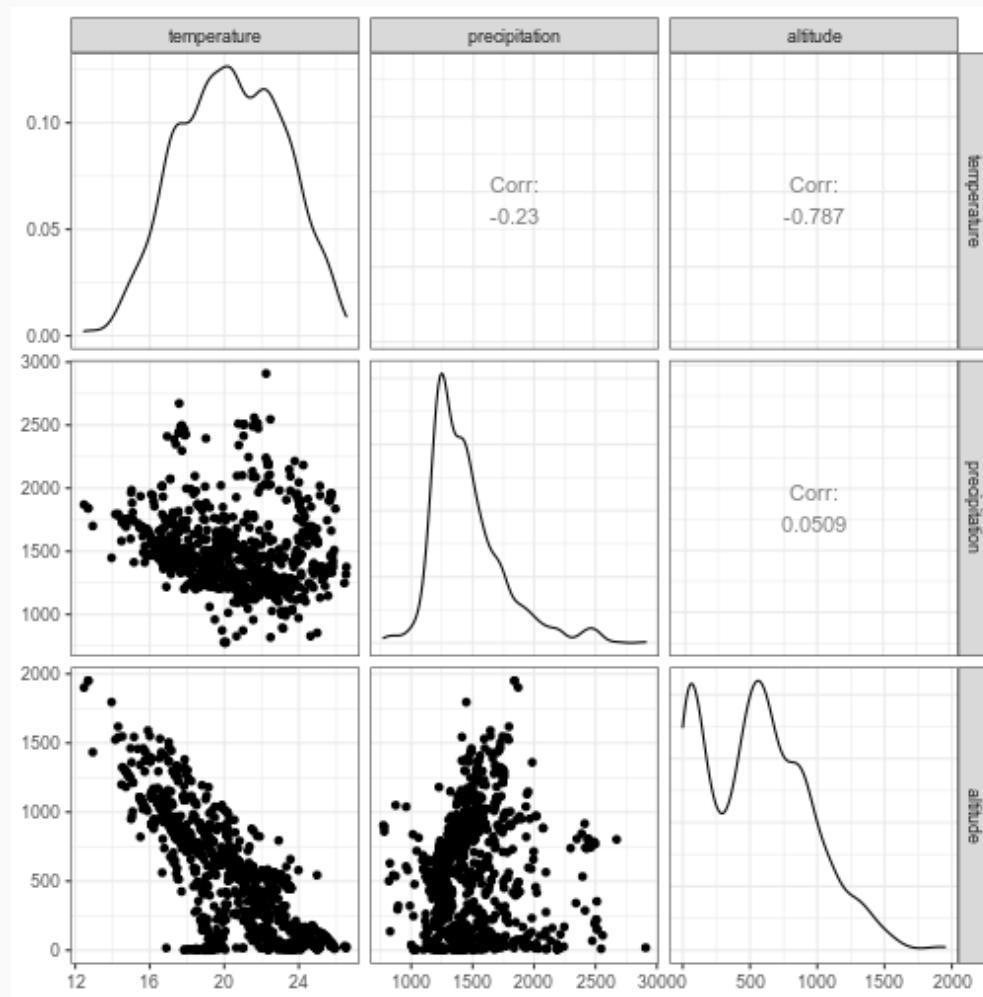
GGally

```
# package
install.packages("GGally")
library(GGally)

# ggpairs
ggpairs(data = da_sel) +
  theme_bw()
```

4.10 Gráfico pareado (pairs plot)

GGally





GGPUBR
Publication-
ready plots

4.10 Gráfico pareado (pairs plot)

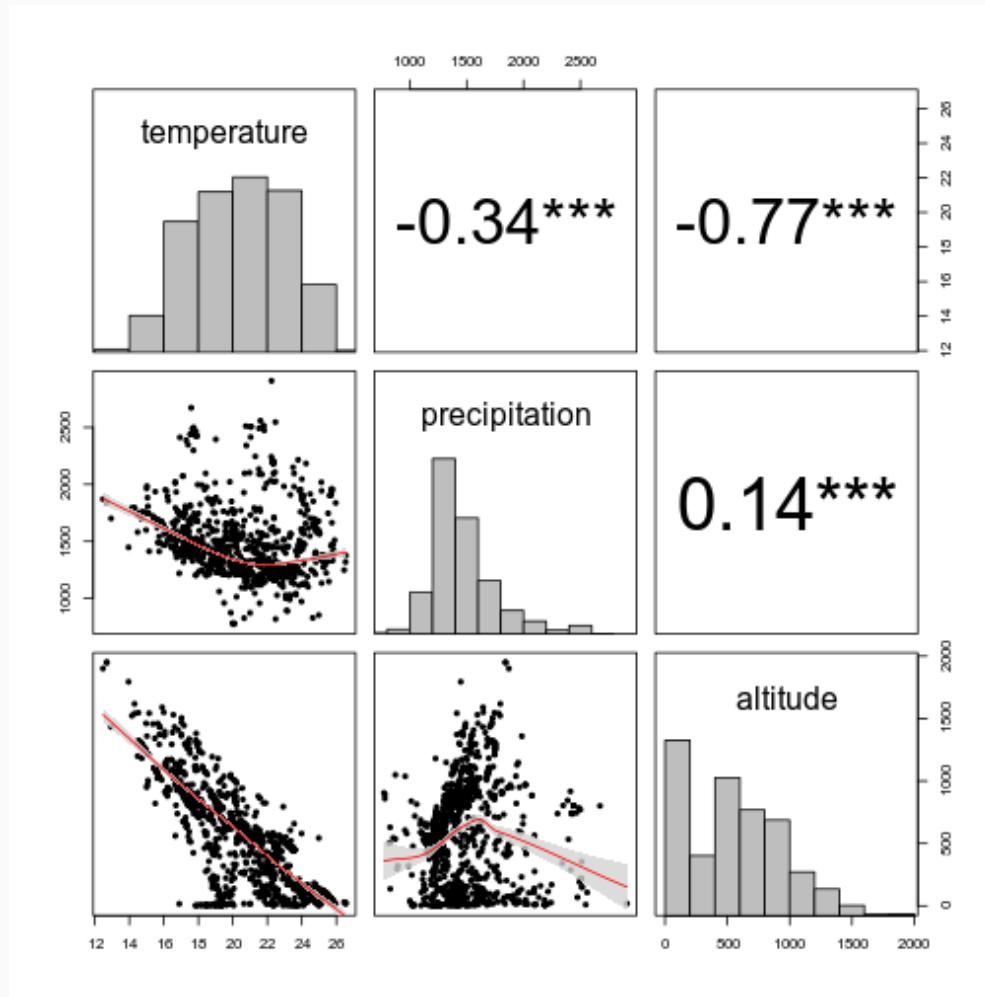
psych

```
# package
install.packages("psych")
library(psych)

# pairs
pairs.panels(da_sel,
              method = "spearman",
              pch = 20,
              ellipses = FALSE,
              density = FALSE,
              stars = TRUE,
              hist.col = "gray",
              digits = 2,
              rug = FALSE,
              breaks = 10,
              ci = TRUE)
```

4.10 Gráfico pareado (pairs plot)

psych



Dúvidas?

Maurício Vancine

Contatos:

 mauricio.vancine@gmail.com

 [mauriciovancine](https://twitter.com/mauriciovancine)

 mauriciovancine.netlify.com

Slides criados via pacote [xaringan](#) e tema [Metropolis](#)