

Universidade Federal do Rio Grande do Norte
Instituto Metr pole Digital
IMD0601 - Bioestat stica

Visualiza  o dos dados em R

Prof. Dr. Tetsu Sakamoto
Instituto Metr pole Digital - UFRN
Sala A224, ramal 182
Email: tetsu@imd.ufrn.br



Baixe a aula (e os arquivos)

- Para aqueles que não clonaram o repositório:

```
> git clone https://github.com/tetsufmbio/IMD0601.git
```

- Para aqueles que já tem o repositório local:

```
> cd /path/to/IMD0601
```

```
> git pull
```

Visualização dos dados

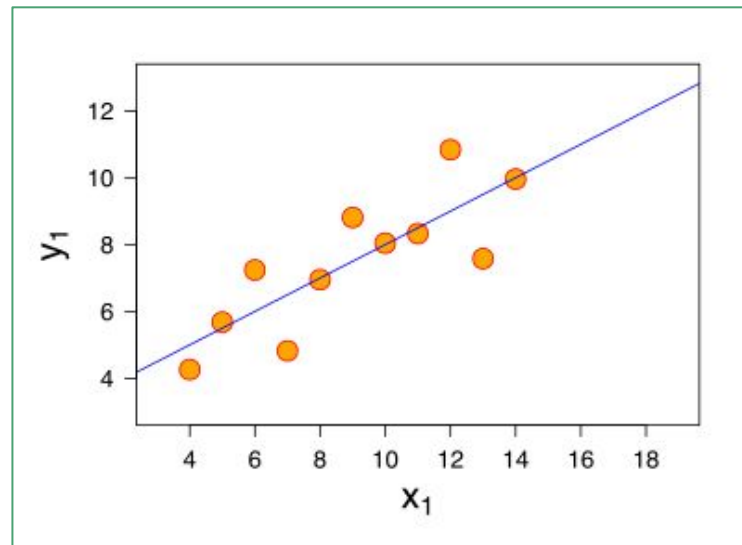
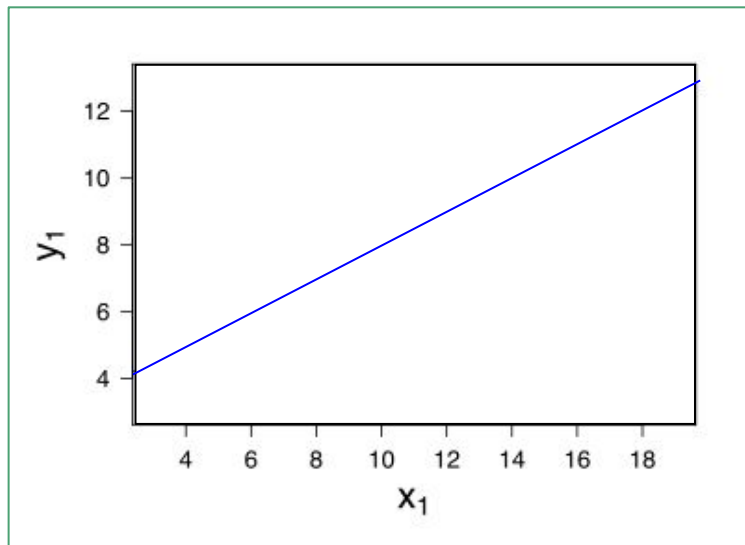
Cientistas de dados;

Combina a área de:

- **Estatística:** análise gráfica dos dados (Representação e interpretação dos dados);
- **Design:** Princípios de design (gráficos atrativos e que promove o melhor entendimento e comunicação);

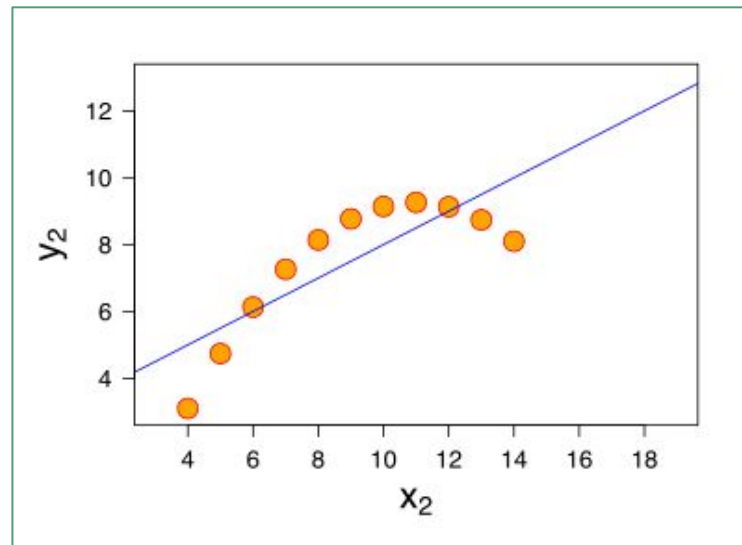
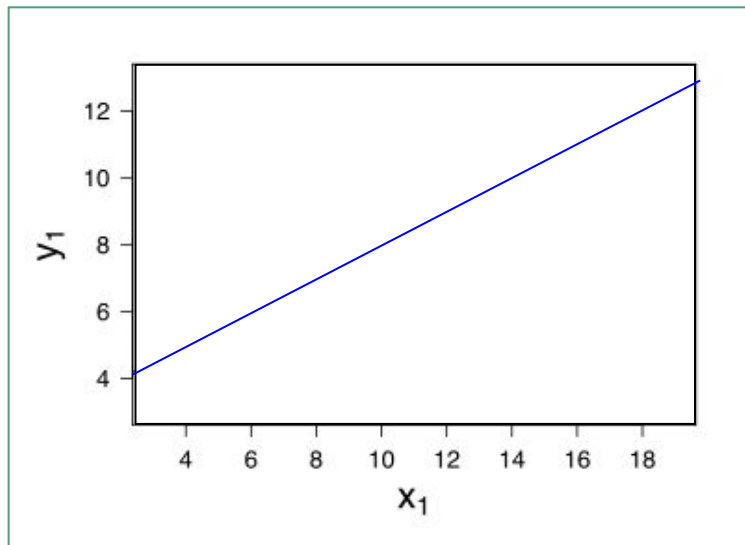
Visualização dos dados

Quarteto de Anscombe



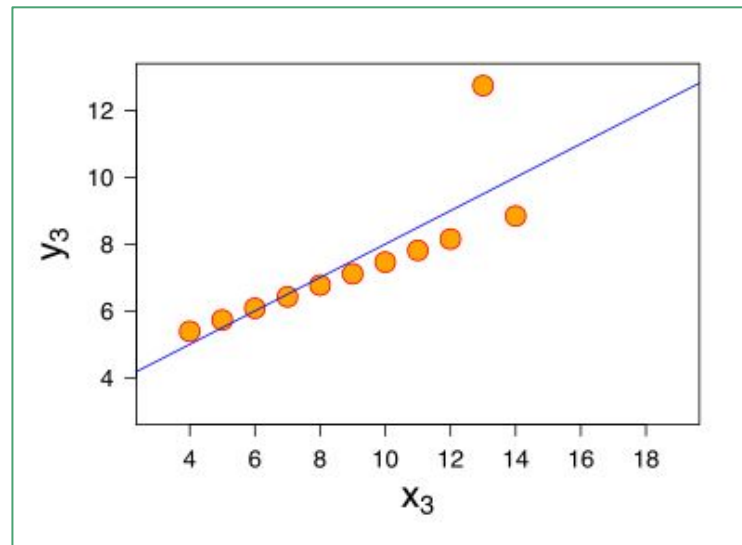
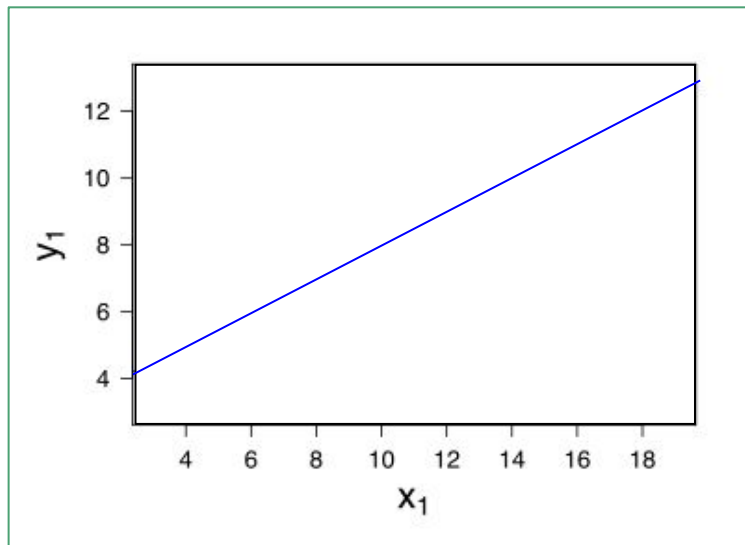
Visualização dos dados

Quarteto de Anscombe



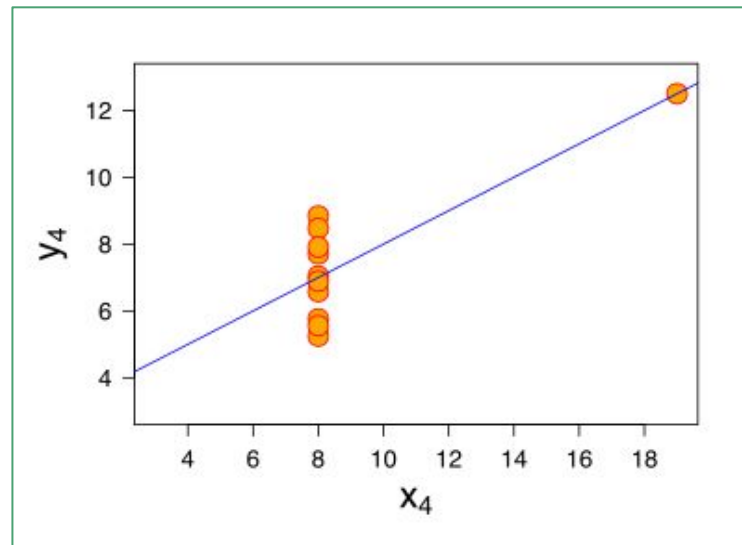
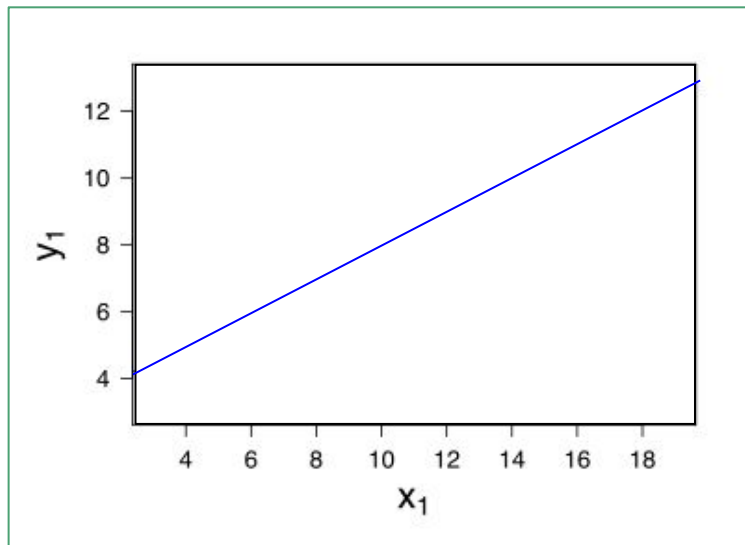
Visualização dos dados

Quarteto de Anscombe



Visualização dos dados

Quarteto de Anscombe



Visualização dos dados

Gráficos exploratórios X Gráficos explanatórios

- Exploratório:

- Gerado facilmente;
- Dados pesados;
- Para uma audiência específica (você e seus colegas);
- Análise gráfica dos dados;

R Base Graphics

- Explanatório:

- Laborioso;
- Específico para determinados dados;
- Para uma audiência ampla (publicação ou apresentação);
- Parte comunicativa do processo.

ggplot2
ggvis
lattice

R Base Graphics

```
library(swirl)
```

```
swirl()
```

```
R Programming
```

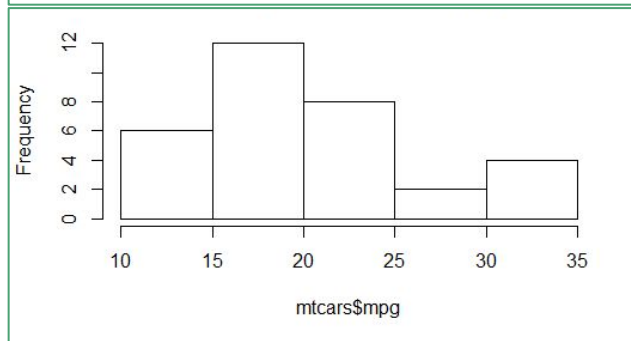
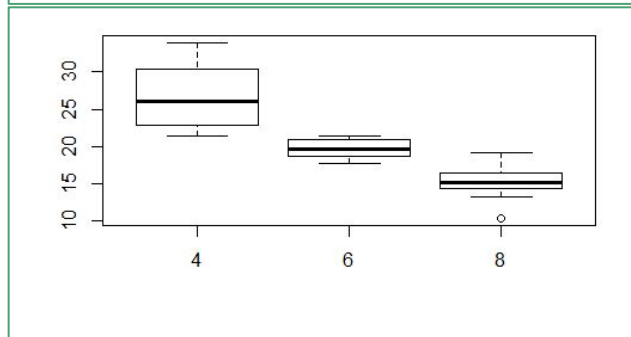
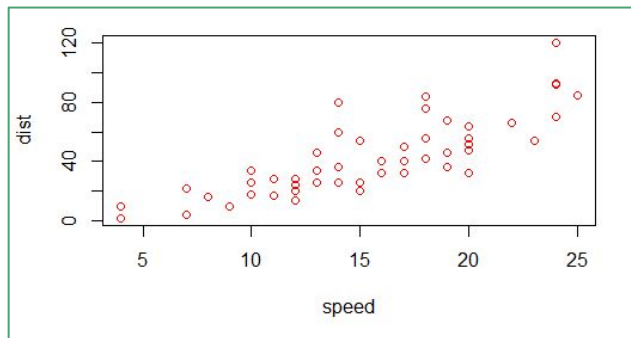
```
R Programming Base Graphics
```

R Base Graphics

```
plot(x = cars$speed, y = cars$dist,  
     xlab = "Speed", ylab = "Stopping  
     Distance", col = 2)
```

```
boxplot(formula = mpg~cyl, data =  
mtcars)
```

```
hist(mtcars$mpg)
```



ggplot2

Hadley Wickham

“The Grammar of Graphics”

Adiciona camadas nos gráficos
para melhor visualização dos
dados;

```
library(ggplot2)
```



ggplot2

Data

Dados (tabela) onde se encontram as variáveis a serem representados graficamente.

```
ggplot(data=mtcars)
```

```
ggplot(mtcars)
```



ggplot2

Aesthetics

Permite especificar as variáveis que queremos utilizar na representação gráfica.

```
ggplot(mtcars, aes(x=mpg,  
y=wt))
```



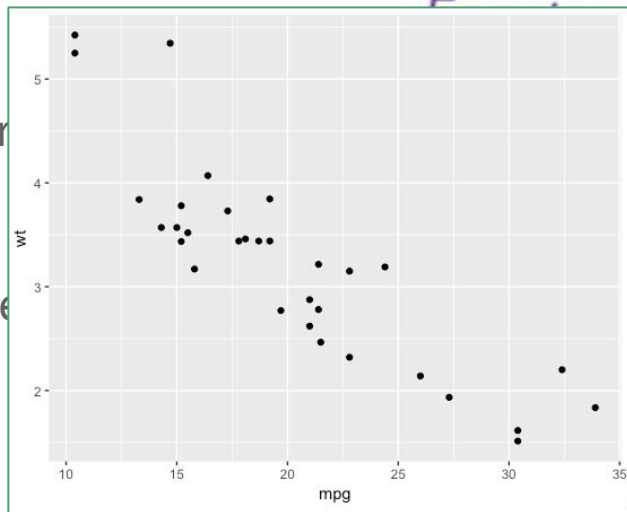
ggplot2

Geometries

Camada que indica a forma como os dados devem ser apresentados no gráfico.

```
ggplot(mtcars, aes(x=mpg, y=wt)) + geom_point()
```

```
g <- ggplot(mtcars, aes(x=mpg, y=wt)) + geom_point()
```



Theme
Coordinates
Statistics



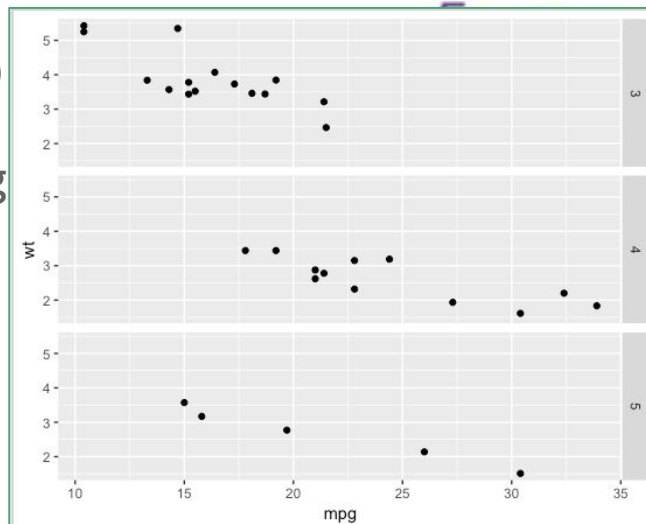
ggplot2

Facets

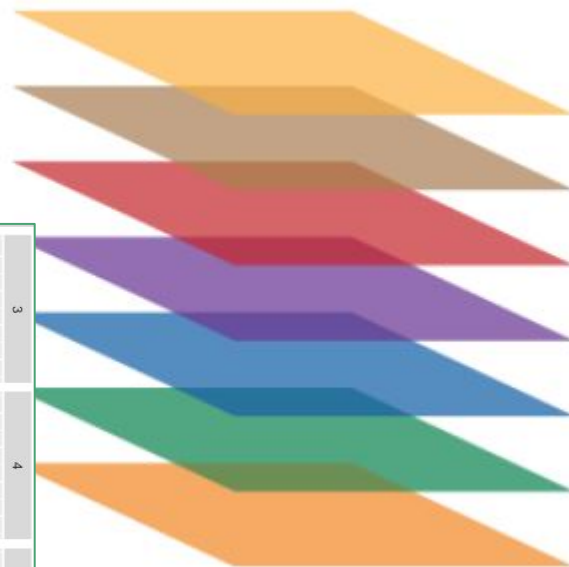
Permite colocar múltiplos gráficos em um canvas.

```
g + facet_grid(gear~)
```

```
g <- g + facet_grid(gear~)
```



Theme
Coordinates
Statistics



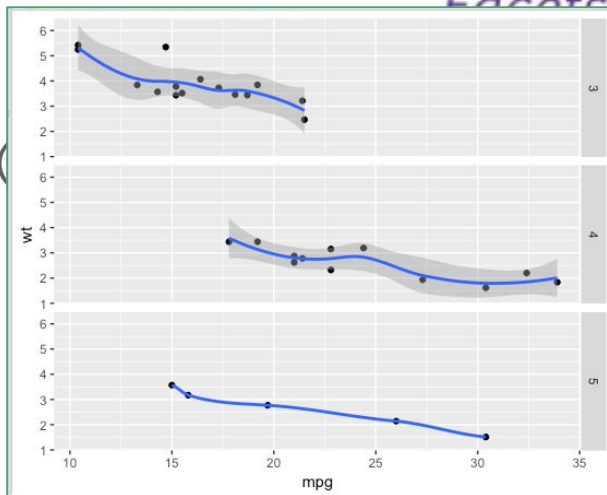
ggplot2

Statistics

Adiciona uma camada que representa uma análise estatística.

```
g + stat_smooth()
```

```
g <- g + stat_smooth()
```



Theme
Coordinates
Statistics
Facets



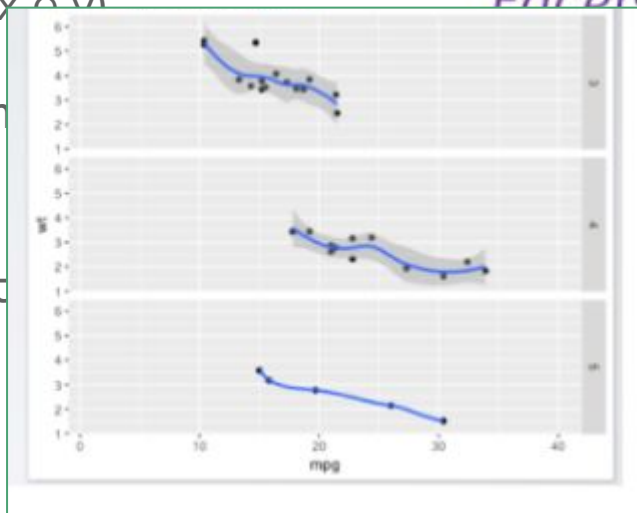
ggplot2

Coordinates

Camada que controla como as posições devem ser mapeadas no gráfico (limites do eixo x e y)

```
g + coord_cartesian(
  c(1, 40))
```

```
g <- g + coord_cartesian(
  = c(1, 40))
```



Theme
Coordinates
Statistics
Facets



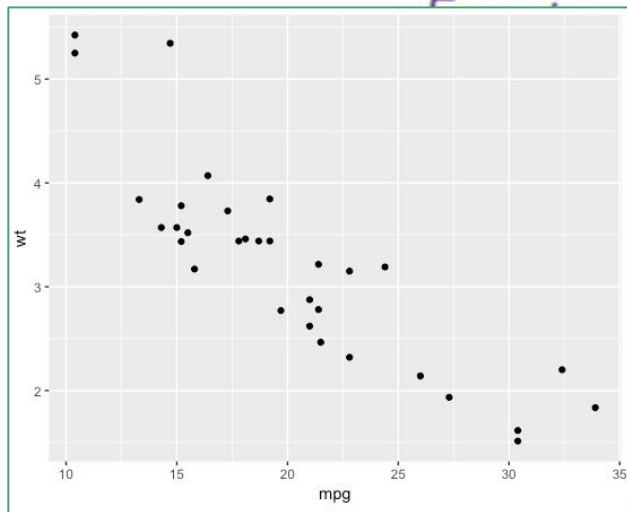
ggplot2

Themes

Camada que permite enriquecer a apresentação do gráfico (rótulos, fonte, cor, etc).

```
g + theme_dark()
```

```
g <- g + theme_dark()
```



Theme
Coordinates
Statistics



Iris dataset

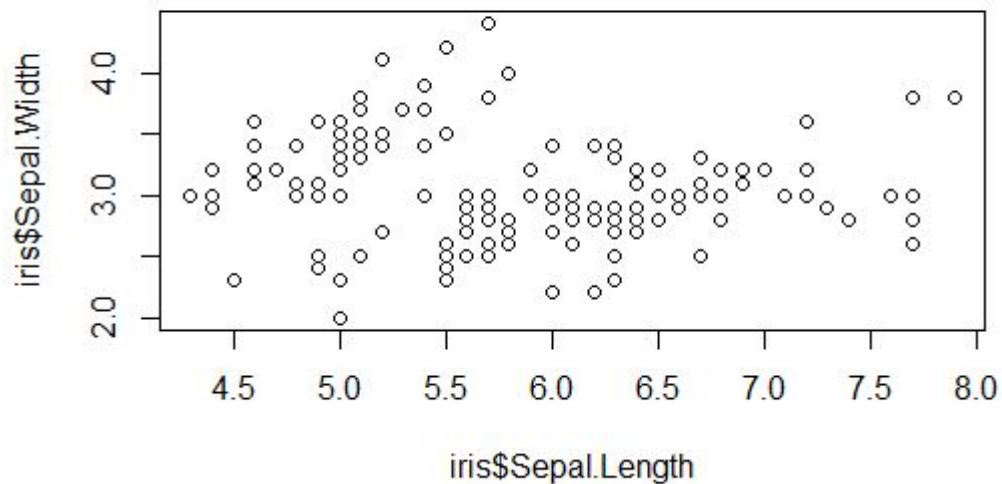
```
data(iris)
```

```
str(iris)
```

```
'data.frame': 150 obs. of 5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1
1 1 1 1 1 1 1 1 ...
```

Base plot

Sepal length X Sepal width



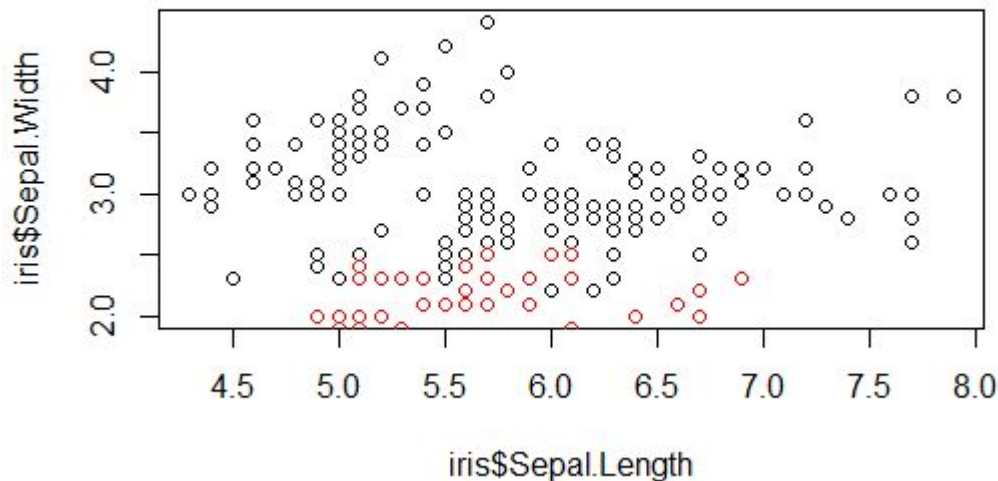
```
> plot(iris$Sepal.Length, iris$Sepal.Width)
```

Base plot

Como adicionar os dados de Petal length e Petal width?

```
> plot(iris$Sepal.Length, iris$Sepal.Width)
```

```
> points(iris$Petal.Length, iris$Petal.Width, col = "red")
```



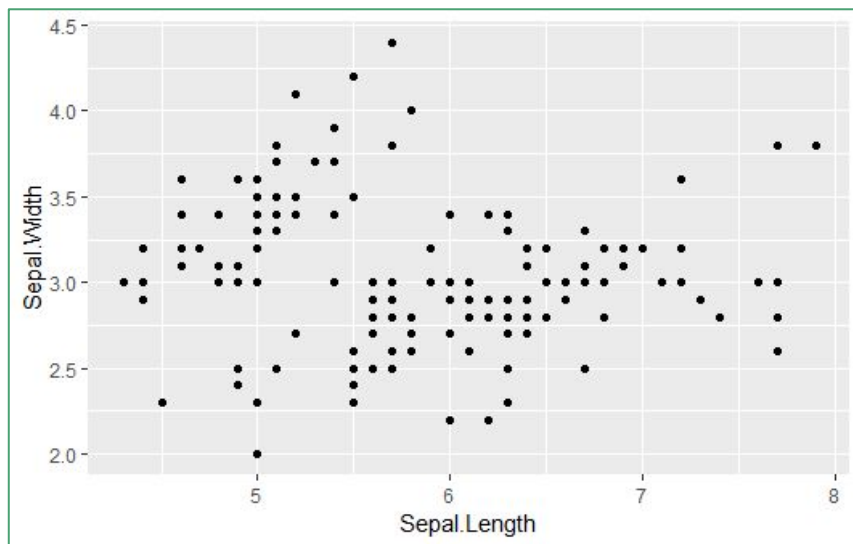
Limitações:

- O gráfico não é redesenhado;
- O gráfico gerado é uma imagem;
- A legenda é adicionada manualmente;
- Não há uma framework unificado para gerar diferentes tipos de gráficos.

ggplot2

Sepal length X Sepal width

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



ggplot2

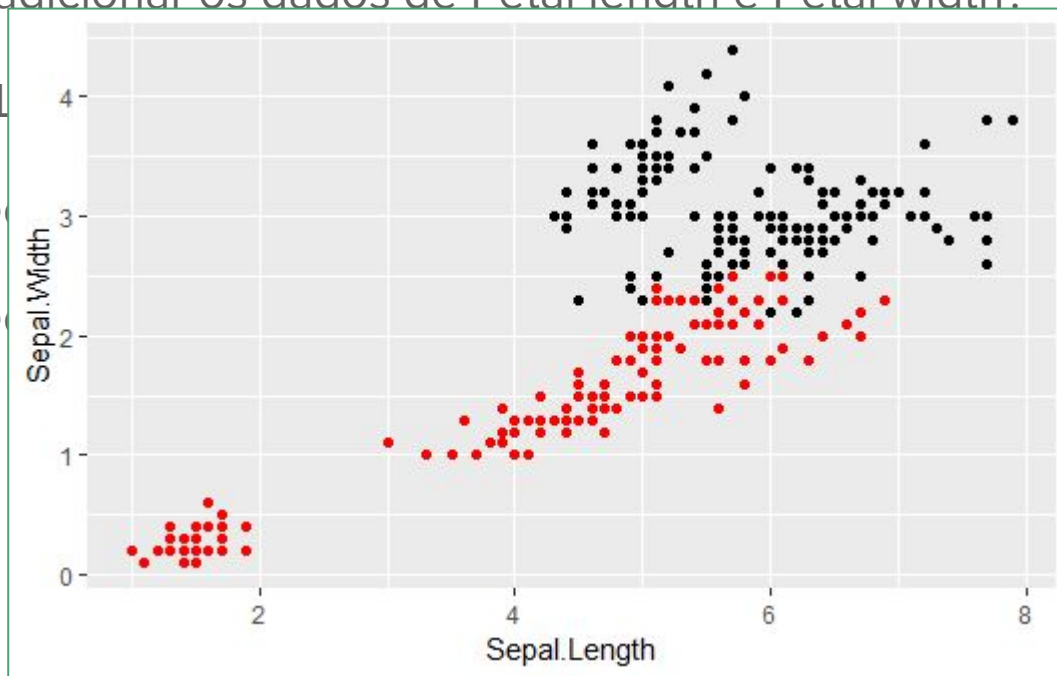
Como adicionar os dados de Petal length e Petal width?

```
> ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +  
  geom_point() +  
  geom_point(aes(x = Petal.Length, y = Petal.Width),  
col = "red")
```

ggplot2

Como adicionar os dados de Petal length e Petal width?

```
> ggplot(  
  ge  
  ge  
  col =  
  .Width)) +  
  .Width),
```



ggplot2

Exercício

Adicione uma coluna na tabela **iris** que corresponda a um identificador único de cada observação.

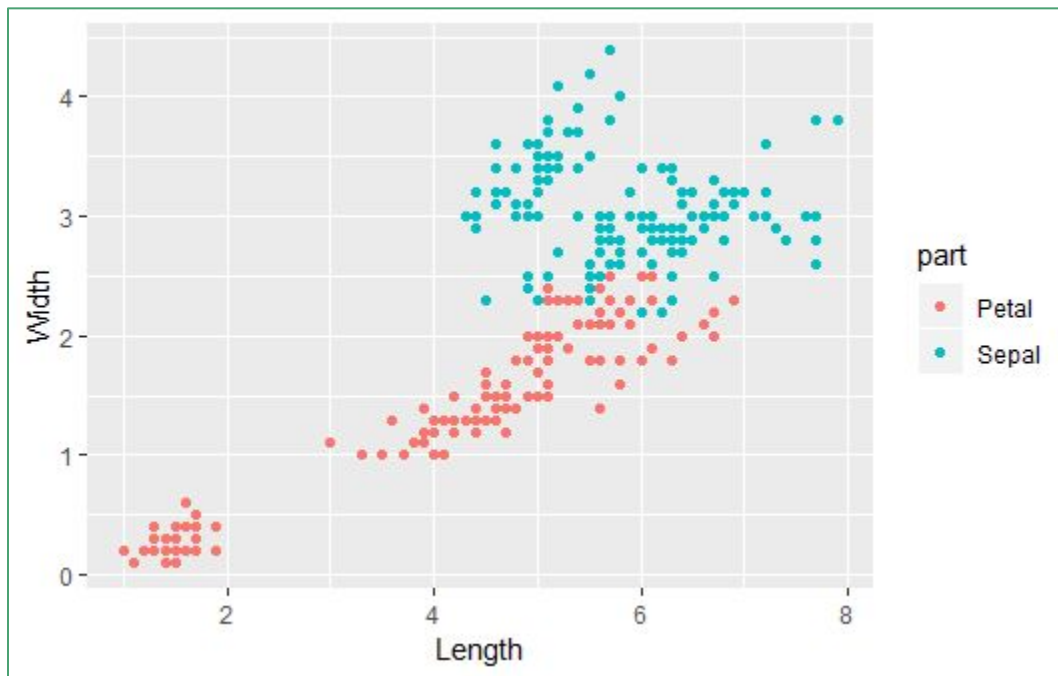
```
> iris$Flower <- 1:nrow(iris)
```

Crie uma tabela onde as variáveis Length e Width estejam em uma coluna.

	Species	Flower	part	Length	Width
1	setosa	1	Petal	1.4	0.2
2	setosa	1	Sepal	5.1	3.5
3	setosa	2	Petal	1.4	0.2
...					

ggplot2

```
ggplot(iris.wide, aes(Length, Width, col = part)) + geom_point()
```

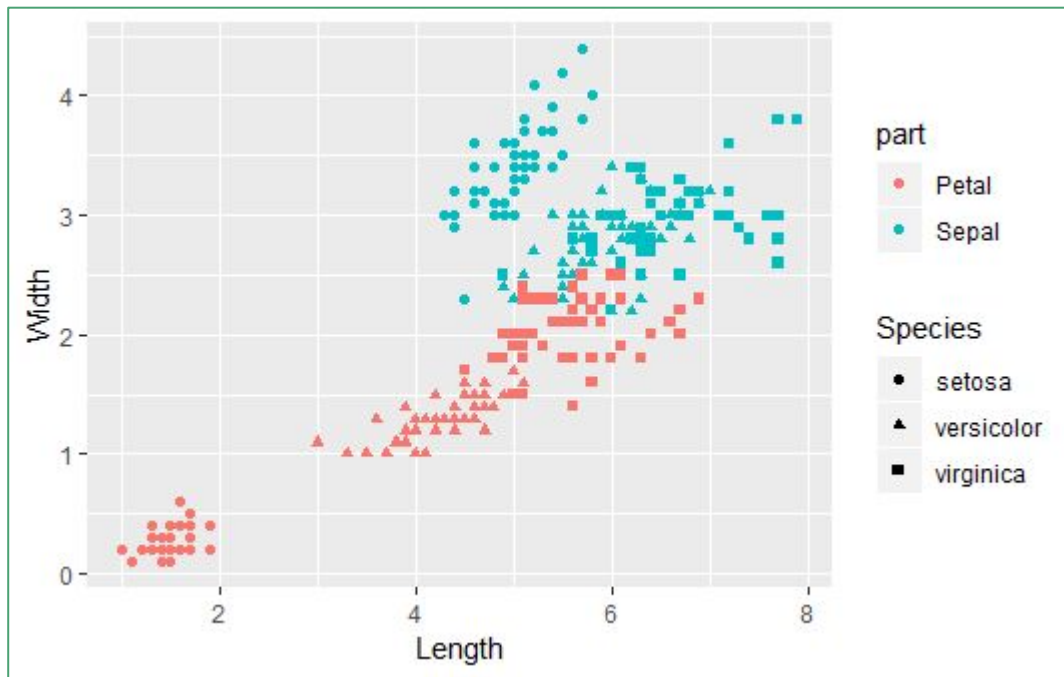


Parâmetros típicos da estética

- **x** → posição no eixo x;
- **y** → posição no eixo y;
- **col** → cor dos pontos, ou de outras formas;
- **fill** → cor a ser preenchido;
- **size** → diâmetro do ponto, largura da linha;
- **alpha** → transparência;
- **linetype** → padrão de tracejamento da linha;
- **labels** → texto no gráfico;
- **shape** → formas;

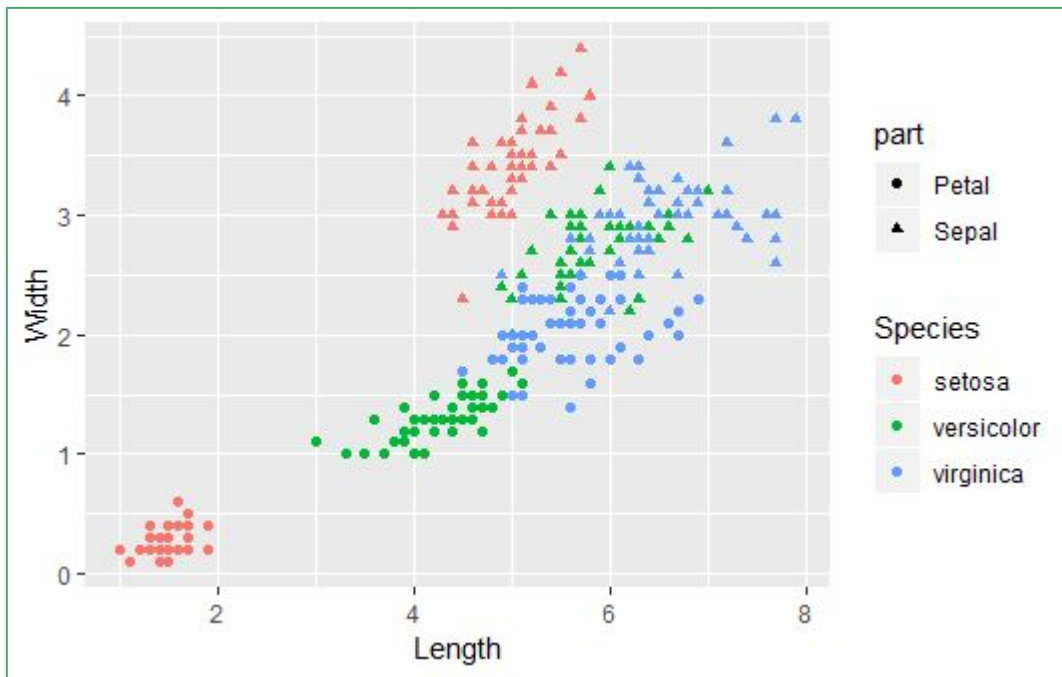
ggplot2

```
ggplot(iris.wide, aes(Length, Width, col = part, shape = Species))  
+ geom_point()
```



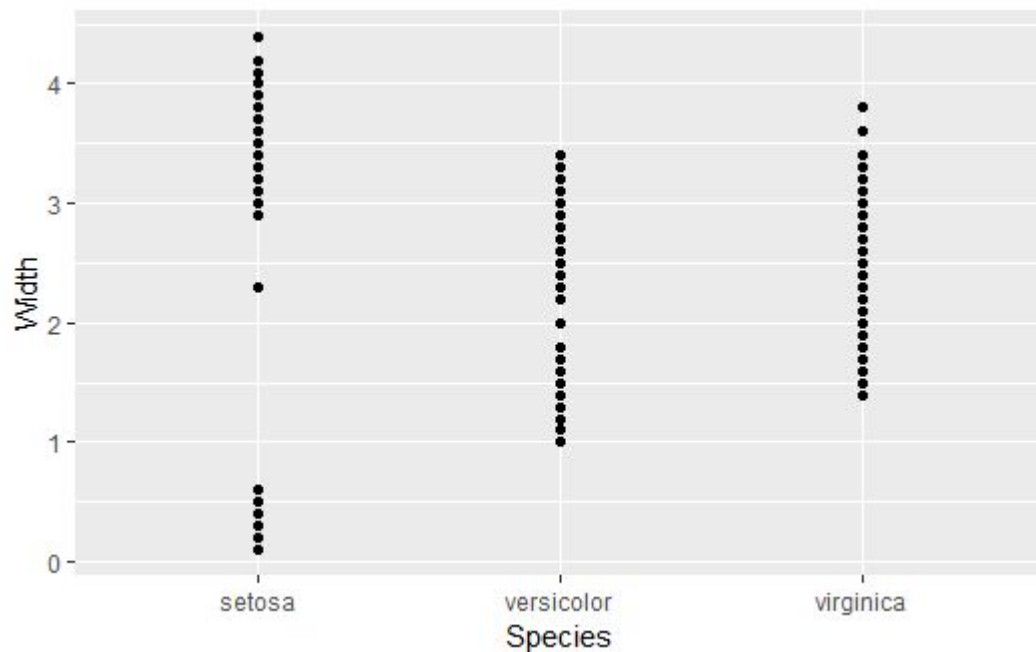
ggplot2

```
ggplot(iris.wide, aes(Length, Width, col = Species, shape = part))  
+ geom_point()
```



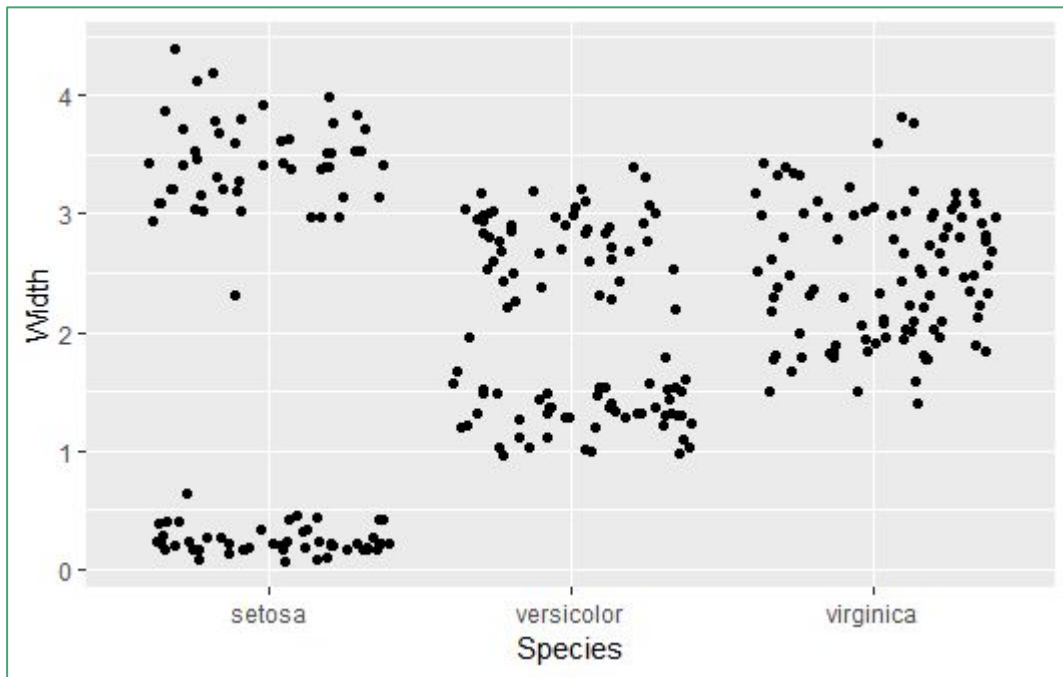
ggplot

```
ggplot(iris.wide, aes(Species, Width)) + geom_point()
```



ggplot

```
ggplot(iris.wide, aes(Species, Width)) + geom_point(position = "jitter")
```

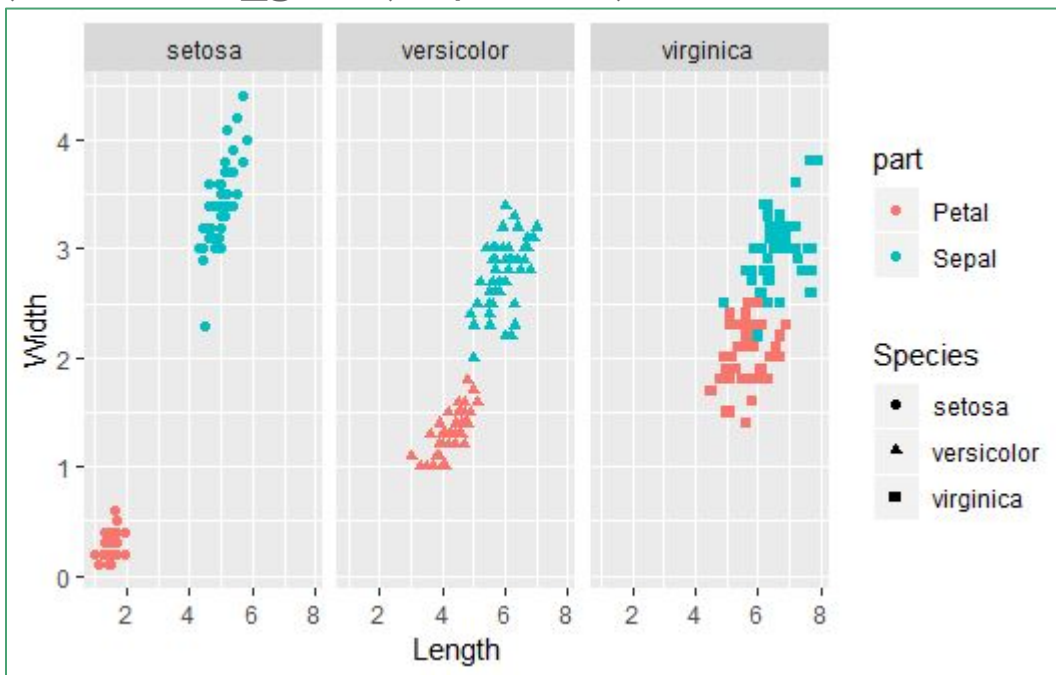


position:

- identity
- jitter
- dodge
- stack
- fill

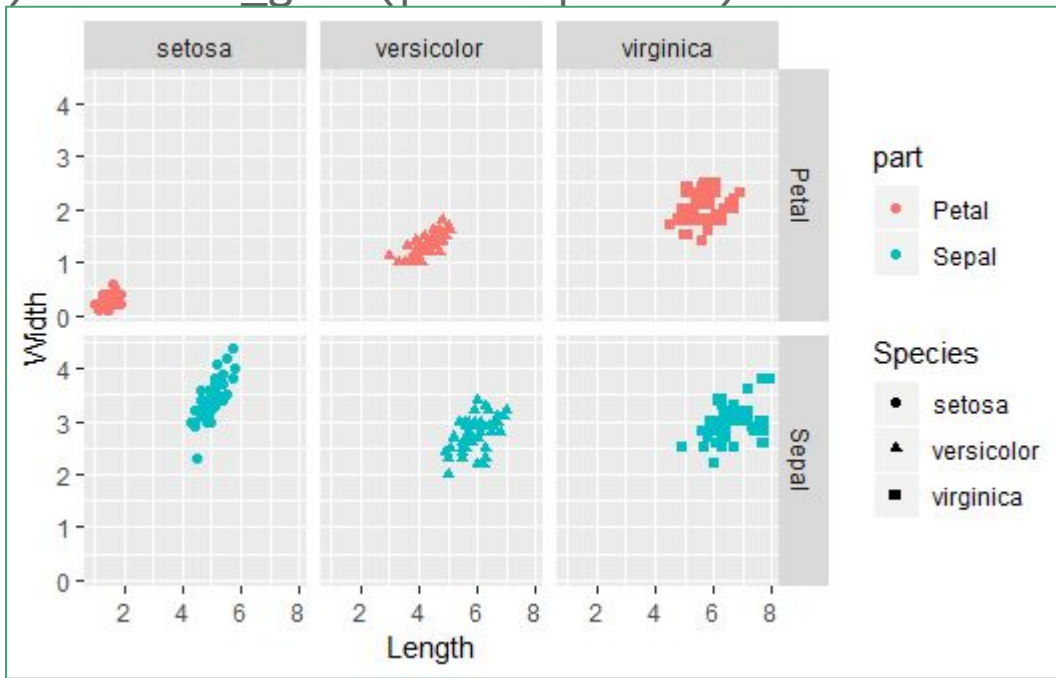
ggplot2 - facet

```
ggplot(iris.wide, aes(Length, Width, col = part, shape = Species))  
+ geom_point() + facet_grid(~Species)
```



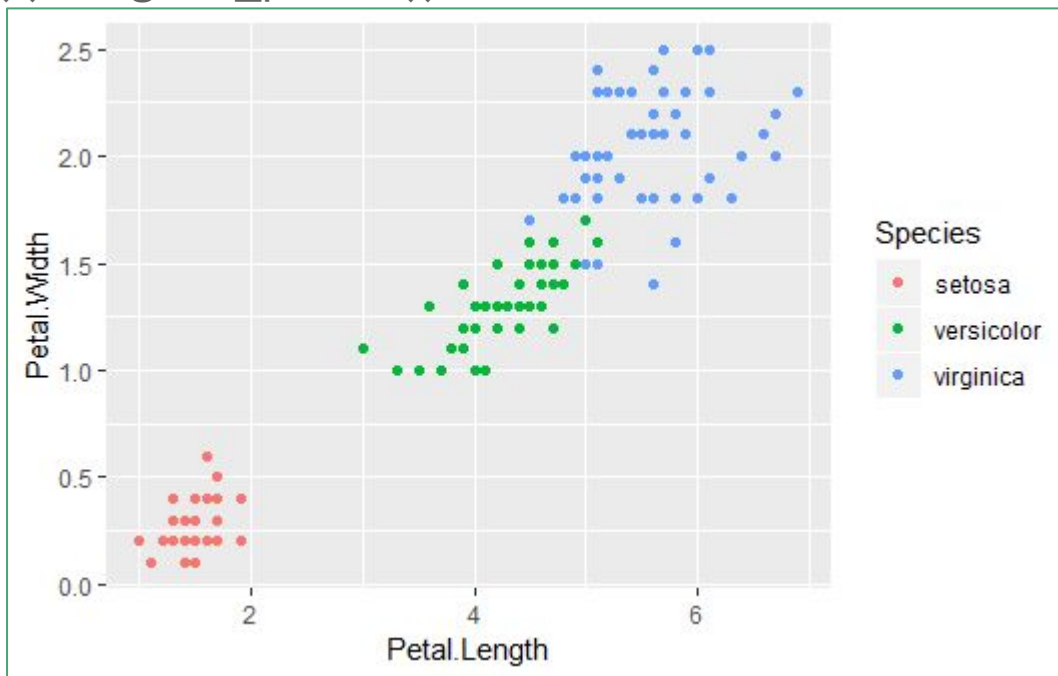
ggplot2

```
ggplot(iris.wide, aes(Length, Width, col = part, shape = Species))  
+ geom_point() + facet_grid(part~Species)
```



ggplot2 - adicionando camadas

```
> ggplot(iris, aes(x = Petal.Length, y = Petal.Width, col = Species)) + geom_point()
```



Como adicionar
a média do
comprimento e
da largura de
cada espécie?

ggplot2 - adicionando camadas

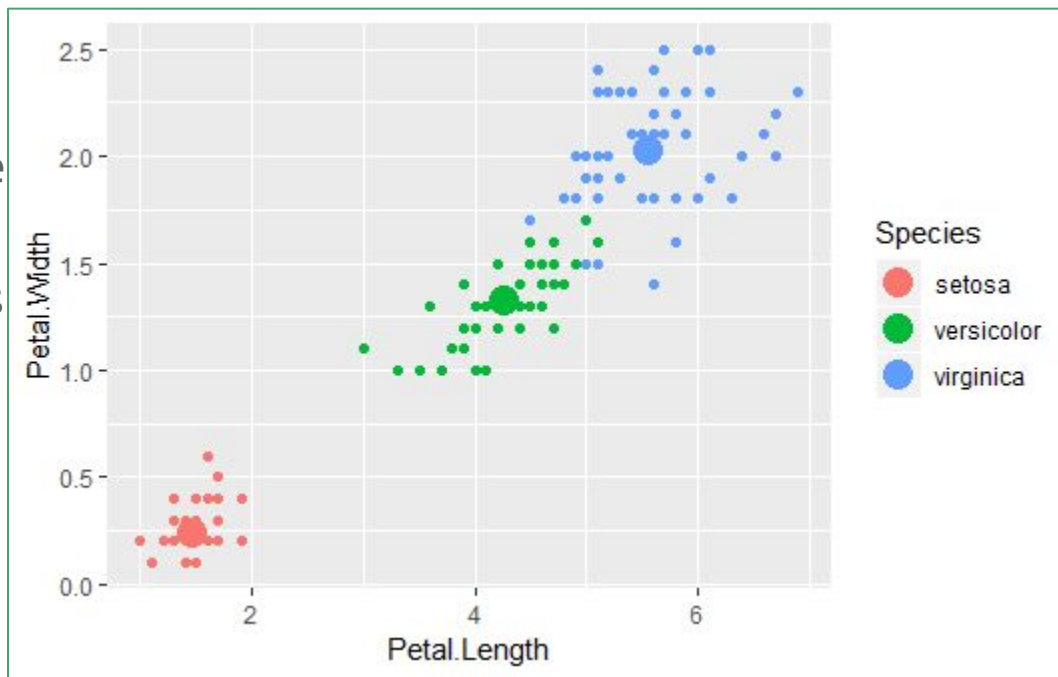
```
> iris.group <- group_by(iris, Species)
> iris.group <- summarise(iris.group, PLM = mean(Petal.Length),
PWM = mean(Petal.Width))
> str(iris.group)

> ggplot(iris, aes(x = Petal.Length, y = Petal.Width, col =
Species)) +
  geom_point() +
  geom_point(data = iris.group, aes(x = PLM, y = PWM), size = 5)
```

ggplot2 - adicionando camadas

```
> iris.group  
> iris.group  
PWM = mean(Pe
```

```
> ggplot(iris  
Species)) +  
  geom_point(  
  geom_point(
```



L.Length),

col =

, size = 5)

ggplot2 - Camada Geométrica

37 geometrias

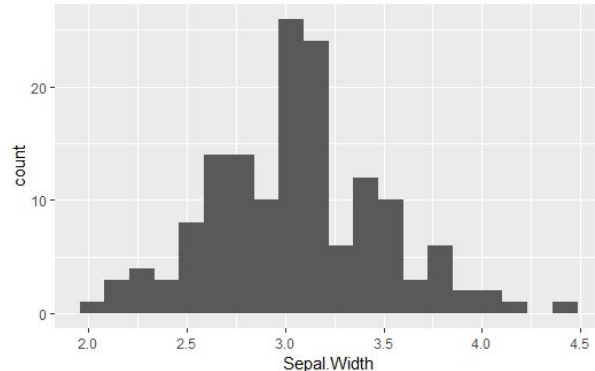
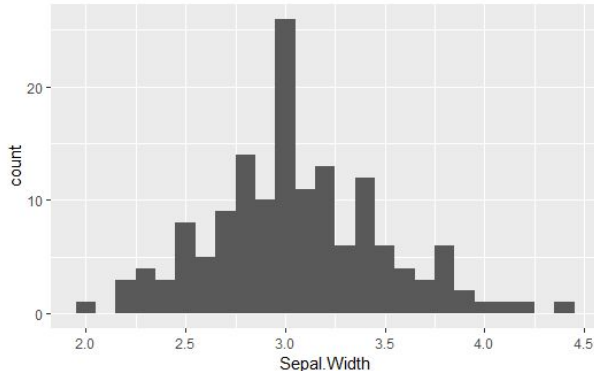
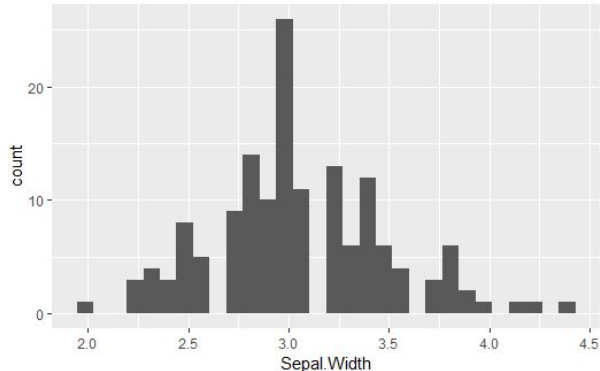
abline	contour	errorbarh	line	polygon	segment	vline
area	crossbar	freqpoly	linrange	quantile	smooth	
bar	density	hex	map	raster	step	
bin2d	density2d	histogram	path	rect	text	
blank	dotplot	hline	point	ribbon	tile	
boxplot	errorbar	jitter	poitrangle	rug	violin	

ggplot2 - Histogramas

```
ggplot(iris, aes(x = Sepal.Width)) + geom_histogram()
```

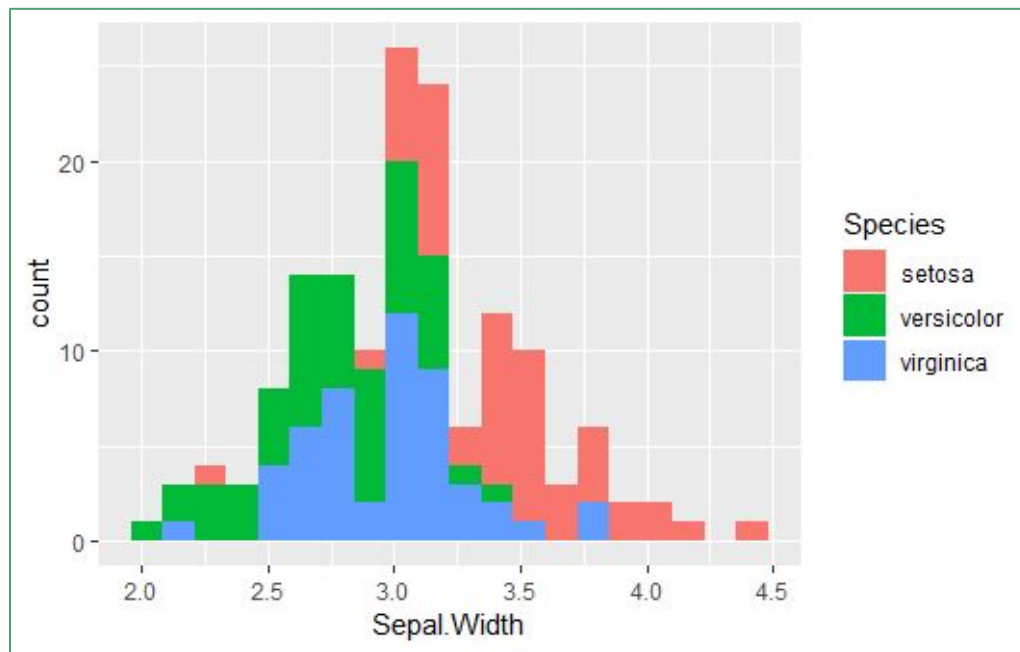
```
ggplot(iris, aes(x = Sepal.Width)) + geom_histogram(binwidth = 0.1)
```

```
ggplot(iris, aes(x = Sepal.Width)) + geom_histogram(bins = 20)
```



ggplot2 - Histogramas

```
ggplot(iris, aes(x = Sepal.Width, fill = Species)) +  
  geom_histogram(bins = 20)
```



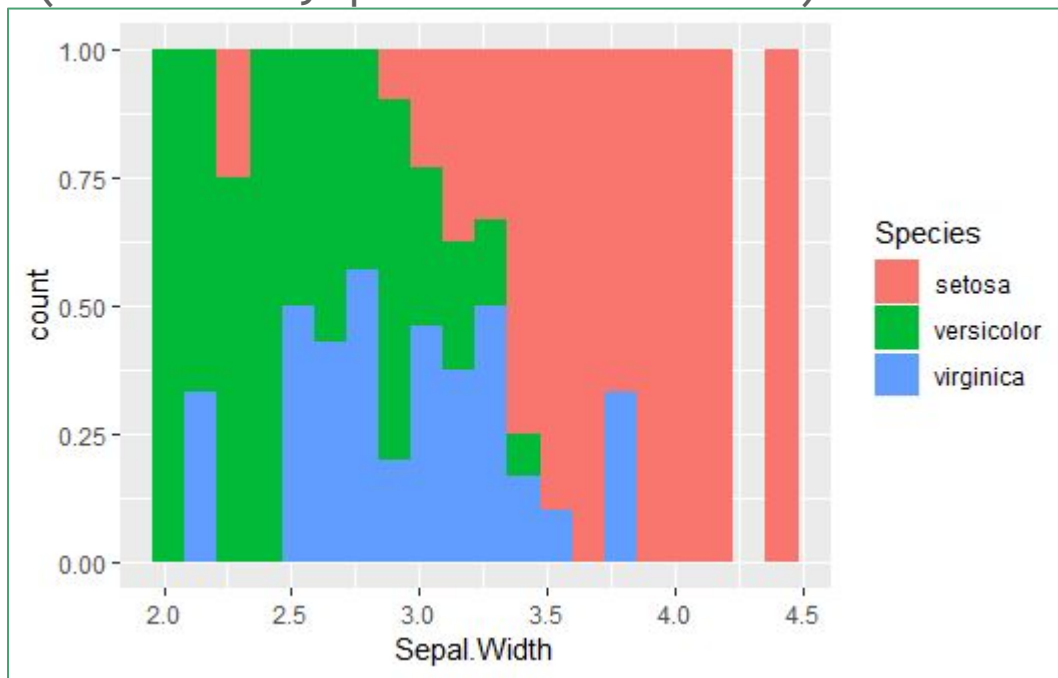
ggplot2 - Histogramas

```
ggplot(iris, aes(x = Sepal.Width, fill = Species)) +  
geom_histogram(bins = 20, position = "dodge")
```



ggplot2 - Histogramas

```
ggplot(iris, aes(x = Sepal.Width, fill = Species)) +  
geom_histogram(bins = 20, position = "fill")
```

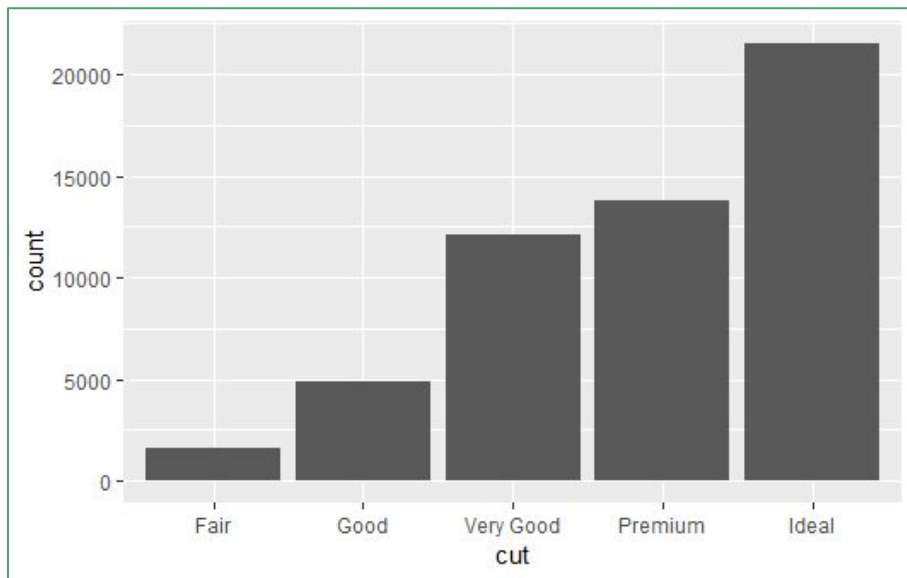


ggplot2 - gráfico de barra

```
data(diamonds)
```

```
str(diamonds)
```

```
ggplot(diamonds, aes(cut)) + geom_bar()
```

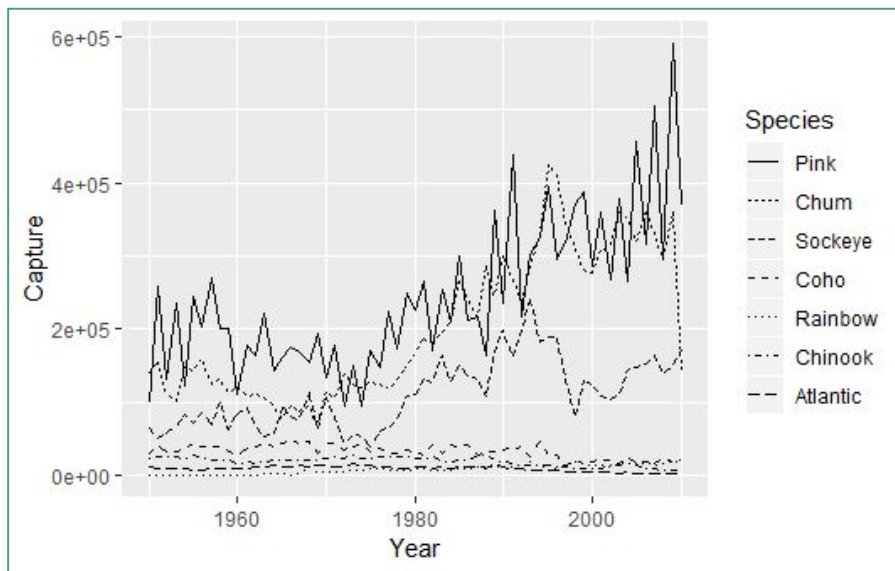


ggplot2 - gráfico de linha

```
load(fish.RData)
```

```
str(fish.tidy)
```

```
ggplot(fish.tidy, aes(x=Year, y=Capture, linetype = Species)) +  
geom_line()
```

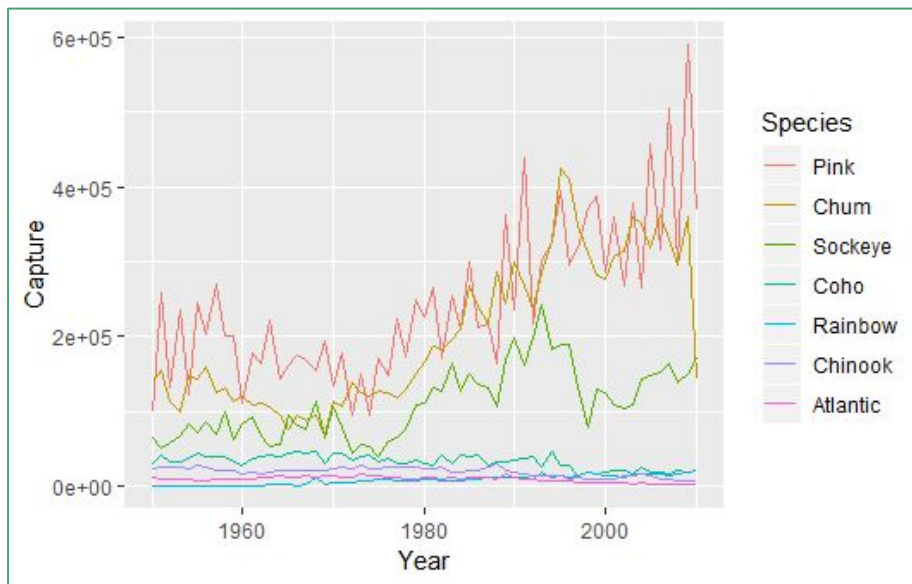


ggplot2 - gráfico de linha

```
load(fish.RData)
```

```
str(fish.tidy)
```

```
ggplot(fish.tidy, aes(x=Year, y=Capture, col = Species)) +  
geom_line()
```

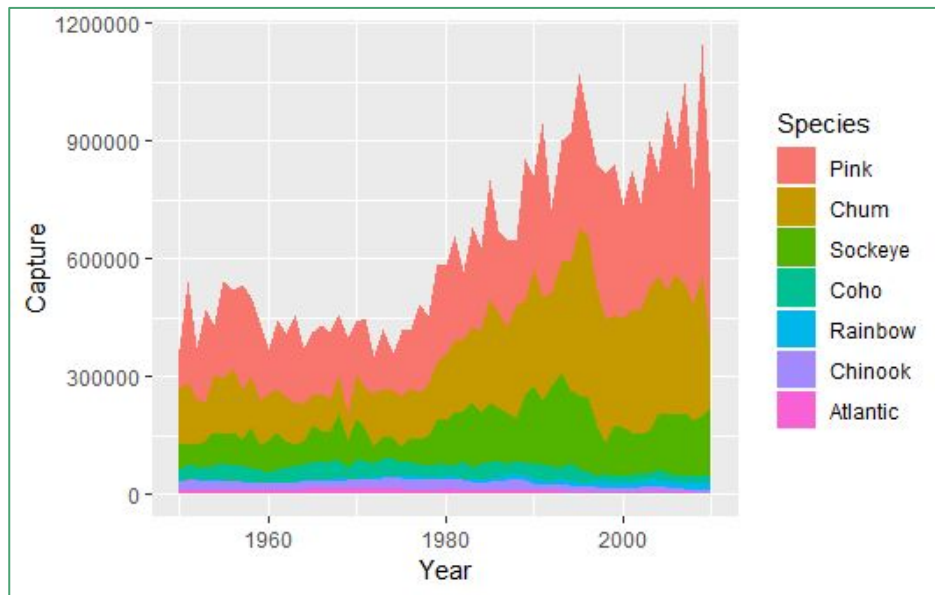


ggplot2 - gráfico de área

```
load(fish.RData)
```

```
str(fish.tidy)
```

```
ggplot(fish.tidy, aes(x=Year, y=Capture, fill = Species)) +  
geom_area()
```

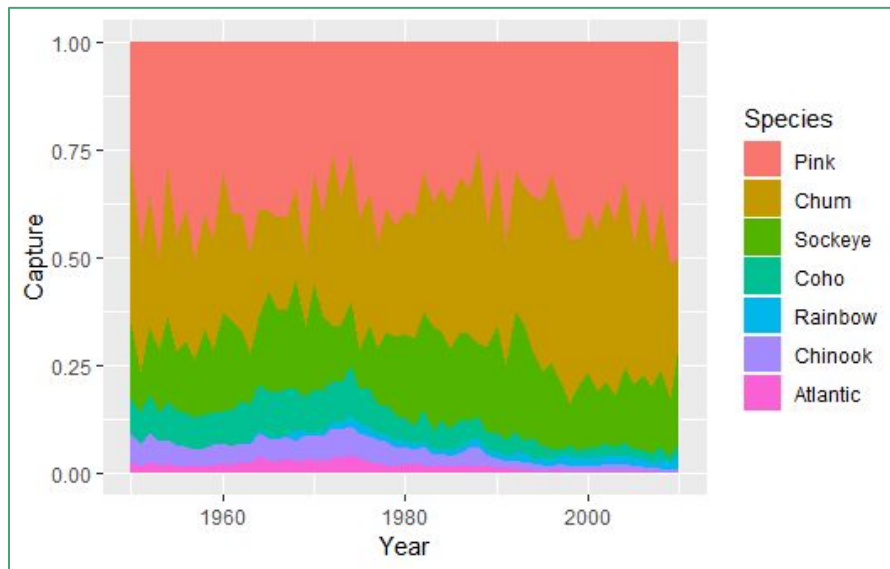


ggplot2 - gráfico de área

```
load(fish.RData)
```

```
str(fish.tidy)
```

```
ggplot(fish.tidy, aes(x=Year, y=Capture, fill = Species)) +  
geom_area(position = "fill")
```



Referência

<https://skillgaze.com/2017/10/31/understanding-different-visualization-layers-of-ggplot/>

Esta aula foi baseada no curso “**Data Visualization with ggplot2 (Part 1)**” de Rick Scavetta (<https://www.datacamp.com/courses/data-visualization-with-ggplot2-1>)