

Aula 6 - Gráficos com ggplot2

Marcelo Prudente e Rafael Giacomin

25 de janeiro de 2018

- 1 Tipos de gráficos
- 2 ggplot2
- 3 Gráfico por subgrupos
- 4 Camadas de ajuste
- 5 Temas

Tipos de gráficos

- **Histogramas:** representa a distribuição de dados numéricos contínuos.
 - ▶ densidade
- **Caixas:** representa a variação de categorias de dados numéricos.
- **Barras:** ótimo para summarizar dados categóricos.
- **Dispersão:** aponta a relação entre duas variáveis contínuas.

```
library(readr)
library(ggplot2)
dados_sociais <- read_csv2("C:/curso_r_enap/dados/dados_sociais.csv")
# criar variável para regiao
dados_sociais$regiao <- substr(dados_sociais$uf, 1, 1)
```

- os 4 gráficos com a base do sistema:

```
# Histograma  
hist(dados_sociais$esp_vida)  
  
# Diagrama de Caixas  
with(dados_sociais, boxplot(esp_vida ~ factor(uf)))  
  
# Barras  
p <- table(ds$uf)  
barplot(p)  
  
# Dispersão  
with(dados_sociais, plot(esp_vida, log(rdpc)))
```

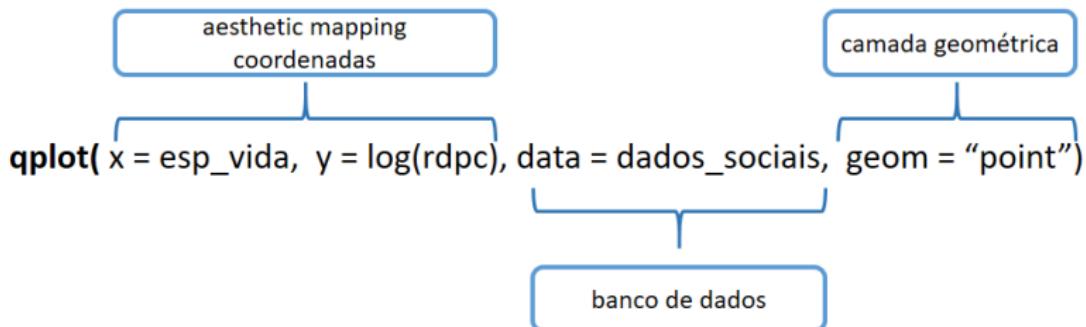
- Tente no seu pc

ggplot2

- É uma alternativa mais elegante aos gráficos com:
 - ▶ cores automáticas para os gráficos
 - ▶ temas pré-estabelecidos (como no excel)
 - ▶ facilidade de customização
 - ▶ **outros:** *veja aqui*
- Em resumo, é possível mexer em todos os aspectos do gráfico ## como funciona o ggplot2
- O ggplot2 aplica a gramática de gráficos de *Wilkinson*
- A ideia central é criar os gráficos em camadas com os seguintes componentes:
 - ▶ banco de dados
 - ▶ coordenadas (x , y) - estética
 - ▶ camadas geométricas

- Há duas abordagens:
 - ▶ **qplot()** - gráficos rápidos (*quick plot*)
 - ▶ **ggplot** - gráficos mais elaborados

- É possível plotar quase tudo com o **qplot**. Entretanto, você não alcançará todo o potencial do ggplot.
- **qplot()** é o mesmo que *quick plot* e segue a lógica de camadas. Veja!



- Veja que é possível criar histogramas e gráficos de caixas com facilidade

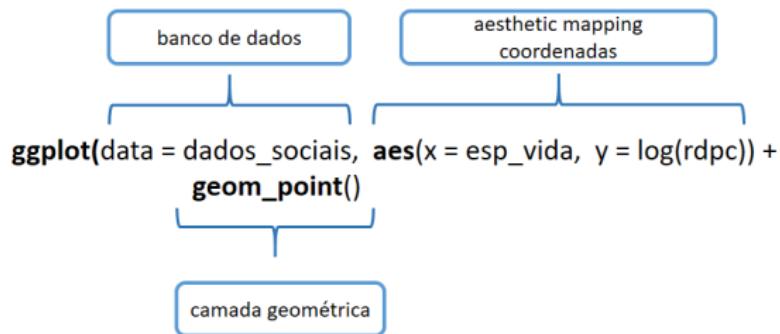
```
# Histograma  
qplot(esp_vida, data = dados_sociais)  
  
# Boxplot  
qplot(x = factor(uf), y = esp_vida,  
       data = dados_sociais, geom = "boxplot")
```

- Do mesmo modo, gráficos de barras e de dispersão

```
# Barplot  
p <- table(dados_sociais$uf)  
qplot(x = factor(uf), data = dados_sociais,  
      geom = "bar")  
  
# Dispersão  
qplot(x = esp_vida, y = log(rdpc), data = dados_sociais,  
      geom = "point")
```

ggplot()

- O gráfico rápido tem grande funcionalidade para observarmos relações de modo fácil.
- Porém, é possível extrair mais dos gráficos.
- A forma básica de expressar o ggplot é a seguinte:



ggplot(): entendendo os argumentos

Enap

- Ao mapear os dados e as coordenadas, o ggplot cria um quadro em branco.

```
g <- ggplot(dados_sociais, aes(esp_vida, log(rdpc)) )  
g
```

- O que você vê?

ggplot(): especificando a camada geométrica

Enap

- Isso ocorre pois é necessário especificar a camada geométrica.

```
g + geom_point()
```

- Em outras palavras, especificamos que os gráficos devem ser pontos. Também seria possível pedir linhas. Veja o que ocorre:

```
g + geom_line()
```

ggplot(): indo além dos argumentos

Enap

- Ainda, é possível especificar as cores dos pontos

```
g + geom_point(colour = "red")
```

- Ainda, é possível especificar ou a cor e o tamanho dos pontos de acordo com faixas populacionais

```
g + geom_point(colour = "red", aes(size = pop) )
```

- Ainda, é possível especificar as cores de acordo com o ano

```
g + geom_point(aes(colour = factor(ano)))
```

ggplot(): preciso saber de todos os verbos?

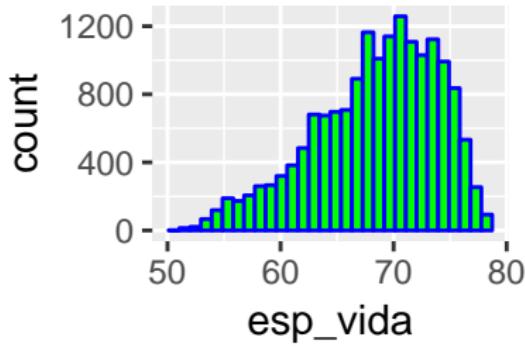
Enap

- De jeito nenhum! O site do *tidyverse* traz todos os exemplos necessários para fazer um bom gráfico.
 - ▶ <http://ggplot2.tidyverse.org/reference/>
 - ▶ <https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf>
- Porém, é importante saber o tipo de variável com que irá trabalhar.

Histogramas

Enap

```
p <- ggplot(dados_sociais, aes(x = esp_vida))  
p + geom_histogram(color = "blue", fill = "green")  
  
## `stat_bin()` using `bins = 30`. Pick better value with `bin
```

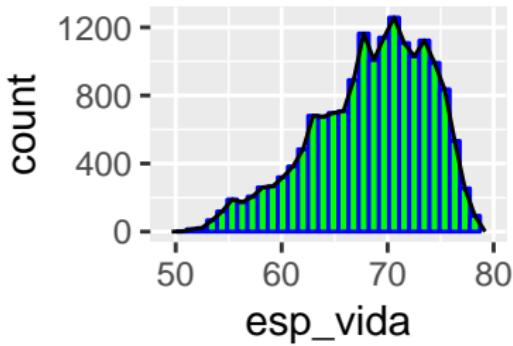


Histogramas 2

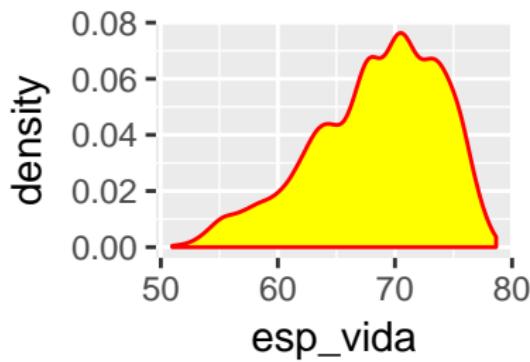
Enap

```
p <- ggplot(dados_sociais, aes(x = esp_vida))  
p + geom_histogram(color = "blue", fill = "green") +  
  geom_freqpoly()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`
```



```
p <- ggplot(dados_sociais, aes(x = esp_vida))  
p + geom_density(color = "red", fill = "yellow")
```



stat_summary()

- O ggplot oferece uma forma de summarizar algumas medidas de tendência central dos dados já no gráfico com o comando stat_summary().
- Se você quiser plotar um gráfico com a média da renda per capita por Estado é possível:

```
# usar dplyr
library(dplyr)
rdpc_uf <- dados_sociais %>%
  group_by(uf) %>%
  summarise(rdpc = mean(rdpc))

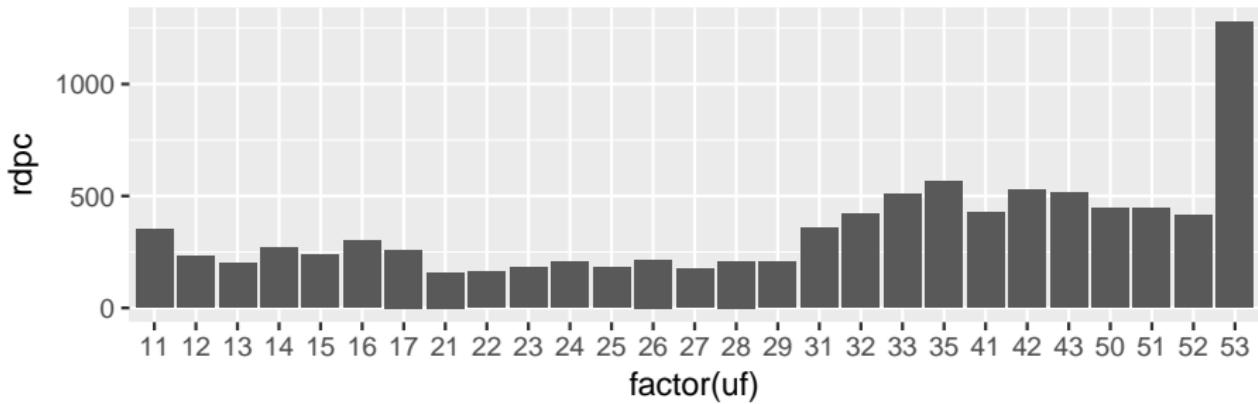
ggplot(rdpc_uf, aes(factor(uf), rdpc)) +
  stat_sum(geom = "bar")
```

- Tente em seu pc.

stat_summary()

- Bom, é preferível usar o **stat_summary()**.

```
dados_sociais %>%
  ggplot(aes(factor(uf), rdpc)) +
  stat_summary(fun.y = "mean", geom = "bar")
```



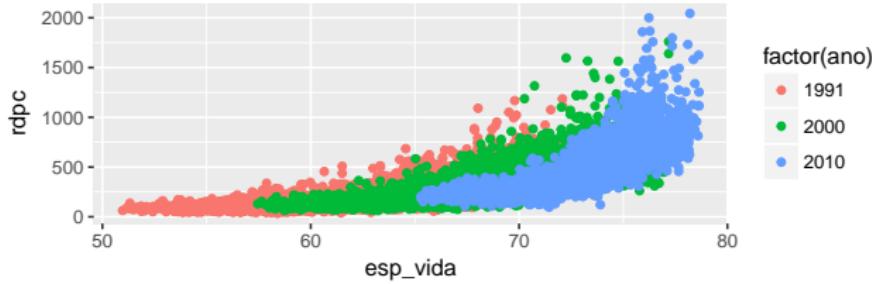
- Mais prático, não?

Gráfico por subgrupos

desagrupar os dados

- O campo **aes()** pode ser utilizado para desagrupar os dados. No caso abaixo, por cor.
 - Observe que o resultado seria distinto se a cor fosse especificada no **geom**

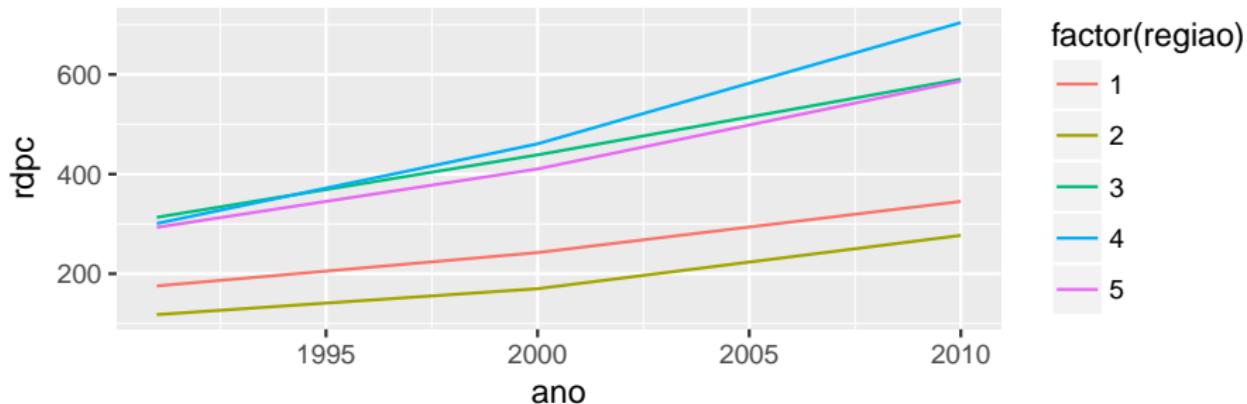
```
graf <- dados_sociais %>%
  ggplot(aes(esp_vida, rdpc, color = factor(ano)) ) +
  geom_point()
graf
```



Sumarizar por grupo

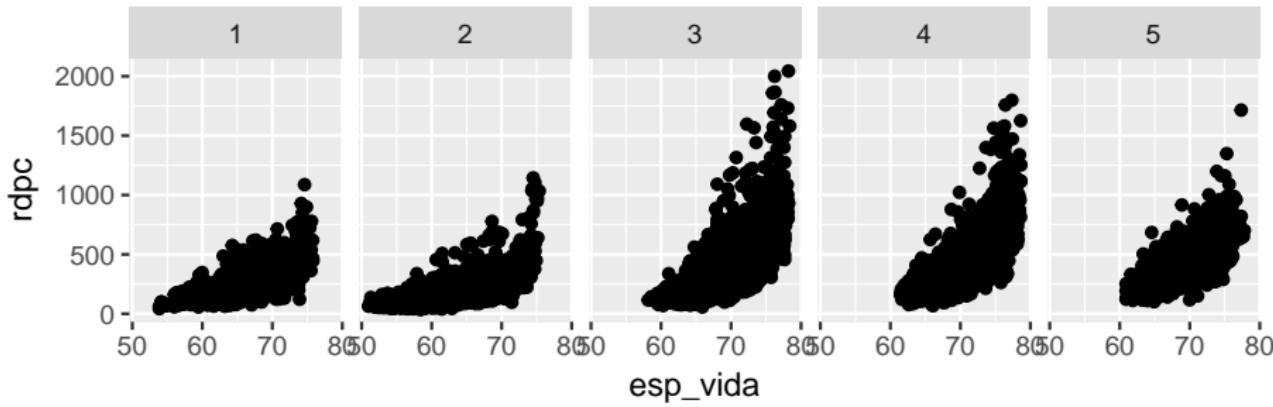
- É possível utilizar o `stat_summary()` para criar gráficos por grupos. Abaixo, o gráfico especifica a média por ano da variável renda per capita

```
# criar plot de base
dados_sociais %>%
  ggplot(aes(ano, rdpc, color = factor(regiao))) +
  stat_summary(fun.y = "mean", geom = "line")
```



- Ainda, é possível criar subgráficos por grupos

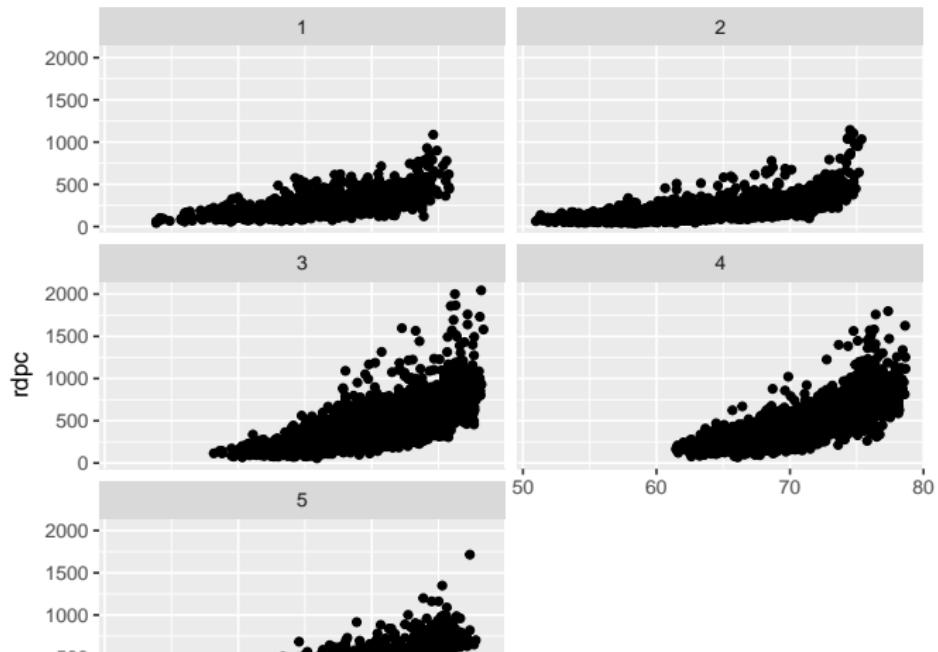
```
# criar plot de base
p <- dados_sociais %>%
  ggplot(aes(esp_vida, rdpc)) + geom_point()
# facet em um mesma grade de imagem
p + facet_grid(. ~ regiao)
```



facet()

- Também, é possível criar subgráficos por grupos com mais grades.

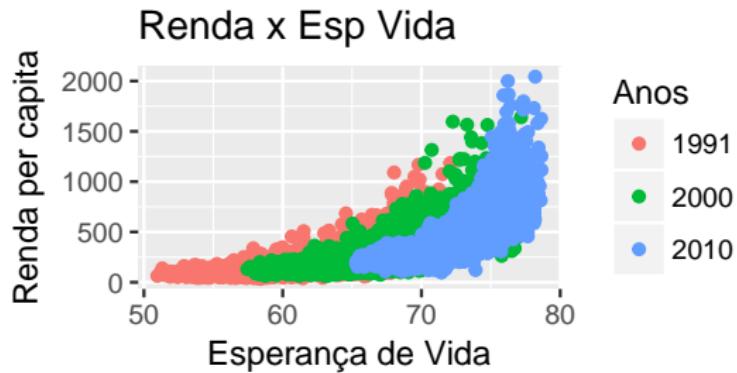
```
# facet wrap  
p + facet_wrap(~ regiao, ncol = 2)
```



Camadas de ajuste

- Veja como é possível modificar os títulos dos eixos e legendas

```
graf <- graf +  
  labs( title = "Renda x Esp Vida",  
        x = "Esperança de Vida", y = "Renda per capita",  
        color = "Anos")  
graf
```



Temas

- Por fim, o **ggplot()** permite modificar os temas dos gráficos.
 - Há alguns temas prontos, mas é possível editar o seu próprio:
help(theme)

```
graf + theme_minimal()
```

