

Introdução ao Tratamento e Análise de Dados em R

Gráficos em R

Sérgio Rivero

PPGE-UFGA

19 de maio de 2019



Sumário

- 1 Objetivo da Aula
- 2 O que é o ggplot2?
- 3 Construindo Gráficos
- 4 Exercícios
- 5 Referências



Objetivo da Aula

Apresentar a biblioteca ggplot2,
seus principais casos de usos e exemplos



O que é o ggplot2?

O **ggplot2** é uma **Gramática de Gráficos**.

Uma gramática de gráficos é um arcabouço que segue uma abordagem em camadas para descrever e construir visualizações ou gráficos de uma maneira estruturada.

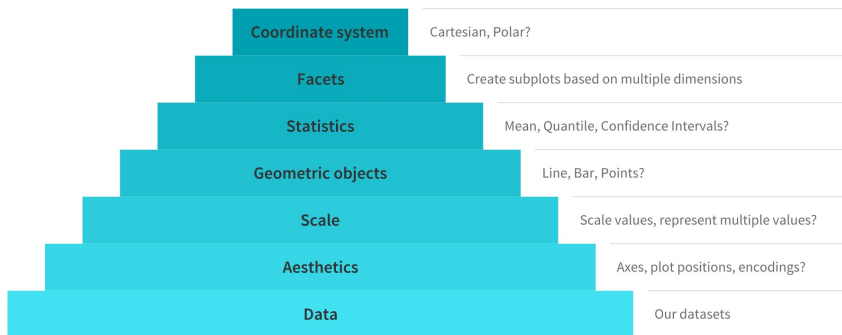
Uma visualização envolvendo dados multidimensionais geralmente tem múltiplos componentes ou aspectos, em termos de dados, estética, escala, objetos etc.

A gramática original da estrutura gráfica foi proposta por Leland Wilkinson [Wil99].



A estrutura do ggplot2

Major Components of the Grammar of Graphics



A estrutura de um gráfico no ggplot2 [Wil99]

FACECON
Faculdade de Ciências Econômicas



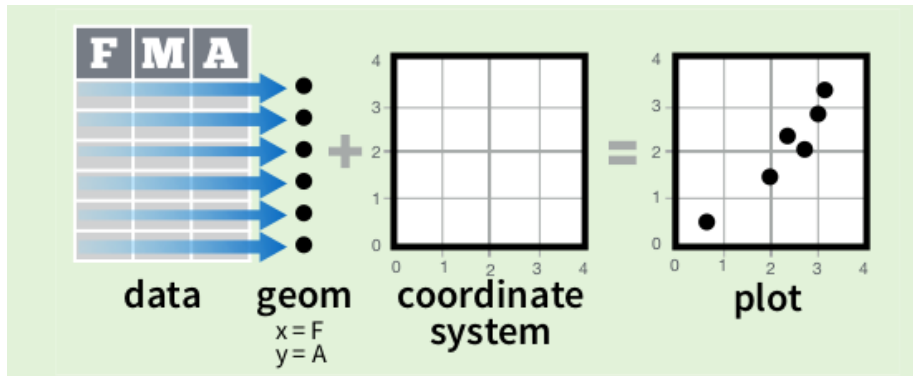
Explicando a estrutura

Explicando os componentes de um gráfico ggplot2

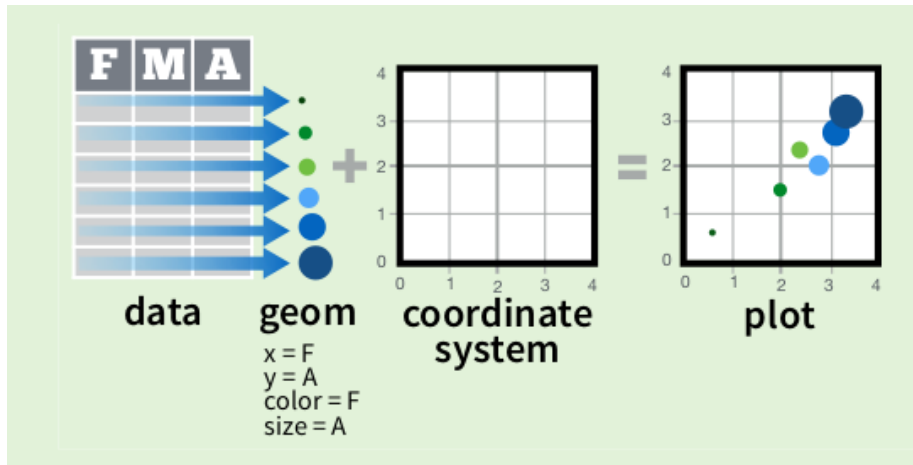
- **Dados:** identifique as dimensões que você deseja visualizar;
- **Estética:** Define-se os eixos , posições pontos de dados e eventualmente tamanho, forma, cor etc;
- **Escala:** uma escala específica para representar vários valores ou um intervalo;
- **Objetos geométricos:** "*geoms*". Devem ser pontos, barras, linhas, etc.;
- **Estatística:** medidas de tendência central, *spread*, intervalos de confiança, etc.;
- **Facetas:** espaço para sub-gráficos a partir de dimensões específicas
- **Sistema de coordenadas:** cartesiana, polar?



A estrutura básica de um gráfico



Incluindo componentes em um gráfico



Um comando ggplot

dados

```
ggplot(mpg, aes(hwy, cty)) +  
  geom_point(aes(color = cyl)) +  
  geom_smooth(method = "lm") +  
  coord_cartesian() +  
  scale_color_gradient() +  
  theme_bw()
```

**adiciona
camadas com +**

**camada = geom +
stat padrão +
mapeamentos
específicos da
camada**

**elementos
adicionais**



Mostrando e salvando gráficos com ggplot

mapeamentos estéticos

dados

geom

ggplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point")

Cria um gráfico completo com os dados, geom, e mapeamentos fornecidos. Possui vários valores padrões úteis.

last_plot()

Retorna o último gráfico.

ggsave("plot.png", width = 5, height = 5)

Salva o último gráfico em arquivo 5' x 5' nomeado de "plot.png" no diretório de trabalho. Define o tipo do arquivo pela extensão.



Gráficos de uma variável

Uma variável

Contínua

```
c <- ggplot(mpg, aes(hwy))
```



```
c + geom_area(stat = "bin")
```

x, y, alpha, color, fill, linetype, size

```
a + geom_area(aes(y = ..density..), stat = "bin")
```



```
c + geom_density(kernel = "gaussian")
```

x, y, alpha, color, fill, group, linetype, size, weight



```
c + geom_dotplot()
```

x, y, alpha, color, fill



```
c + geom_freqpoly()
```

x, y, alpha, color, group, linetype, size

```
a + geom_freqpoly(aes(y = ..density..))
```



```
c + geom_histogram(binwidth = 5)
```

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

```
a + geom_histogram(aes(y = ..density..))
```

Discreta

```
d <- ggplot(mpg, aes(fl))
```



```
d + geom_bar()
```

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



Duas variáveis contínuas

Contínua X, Contínua Y

```
e <- ggplot(mpg, aes(cty, hwy))
```



e + geom_label(aes(label = cty), 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_rug(sides = "bl")
x, y, alpha, color, linetype, size



e + geom_smooth(method = lm)
x, y, alpha, color, fill, group, linetype, size, weight



e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE)
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust



Duas variáveis: uma discreta, uma contínua

Discreta X, Contínua Y

```
f <- ggplot(mpg, aes(class, hwy))
```



f + geom_bar(stat = "identity")

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



f + geom_boxplot()

x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight



f + geom_dotplot(binaxis = "y", stackdir = "center")

x, y, alpha, color, fill, group



f + geom_violin(scale = "area")

x, y, alpha, color, fill, group, linetype, size, weight

Discreta X, Discreta Y

```
g <- ggplot(diamonds, aes(cut, color))
```



g + geom_count()

x, y, alpha, color, fill, shape, size, stroke



Mostrando o erro

Visualizando Erros

```
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))
```



j + geom_crossbar(fatten = 2)

x, y, ymax, ymin, alpha, color, fill, group,
linetype, size



j + geom_errorbar()

x, ymax, ymin, alpha, color, group, linetype,
size, width (also **geom_errorbarh()**)



j + geom_linerange()

x, ymin, ymax, alpha, color, group, linetype,
size



j + geom_pointrange()

x, y, ymin, ymax, alpha, color, fill, group,
linetype, shape, size

Mapas

```
data <- data.frame(murder = USArrests$Murder,
  state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))
```



**k + geom_map(aes(map_id = state), map = map) +
expand_limits(x = map\$long, y = map\$lat)
map_id, alpha, color, fill, linetype, size**



Facetas dividem um gráfico em sub-gráficos baseando-se em uma ou mais variáveis discretas.

```
t <- ggplot(mpg, aes(cty, hwy)) + geom_point()
```



t + facet_grid(. ~ fl)
divide em colunas baseado em fl



t + facet_grid(year ~ .)
divide em linhas baseado em year



t + facet_grid(year ~ fl)
divide em linhas e colunas



t + facet_wrap(~ fl)
ajusta as facetas em um formato retangular



"Decorations"

Ajustes de Posição

Ajustes de posição definem como os geoms se localizam, evitando que ocupem o mesmo espaço.

```
s <- ggplot(mpg, aes(fl, fill = drv))
```



s + geom_bar(position = "dodge")

Coloca os elementos lado a lado.



s + geom_bar(position = "fill")

Empilha os elementos um sobre o outro, normaliza a altura.



e + geom_point(position = "jitter")

Adiciona um ruído aleatório para as posições X e Y de cada elemento evitando a sobreposição.



e + geom_label(position = "nudge")

Afasta os rótulos dos pontos.



s + geom_bar(position = "stack")

Empilha os elementos um sobre o outro.

Cada ajuste de posição pode ser redefinido como um ajuste de posição manual dos argumentos **width** e **height**.

```
s + geom_bar(position = position_dodge(width = 1))
```

Rótulos

t + ggtitle("New Plot Title")

Adiciona um título principal sobre o gráfico

t + xlab("New X label")

Muda o rótulo do eixo X

t + ylab("New Y label")

Muda o rótulo do eixo Y

t + labs(title = "New title", x = "New x", y = "New y")

Todos acima

Use funções **scale** para atualizar o rótulo das legendas

Legendas

n + theme(legend.position = "bottom")

Coloca a legenda no "bottom", "top", "left", ou "right"

n + guides(fill = "none")

Define o tipo da legenda para cada estética: colorbar, legend, ou none (sem legenda)

n + scale_fill_discrete(name = "Title", labels = c("A", "B", "C", "D", "E"))

Define o título da legenda e os rótulos com uma função **scale**.



Exercícios

Exercícios



Referências I



Leland Wilkinson.

The Grammar of Graphics (Statistics and Computing).

Springer, 1999.

