

**ggplot e
extensões**

**Frederico
Bertholini**

Visualização
de dados

ggplot

ggplot e extensões

Métodos Quantitativos Aplicados à Ciência Política

Frederico Bertholini

26.out.2020

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

1 Visualização de dados

2 ggplot

Rode seus pacotes!

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

```
lapply(c("tidyverse", "haven", "lubridate",  
        "janitor", "readxl",  
        "stringr", "magrittr", "srvyr",  
        "survey"), require, character.only=T)
```

**ggplot e
extensões**

**Frederico
Bertholini**

**Visualização
de dados**

ggplot

Visualização de dados

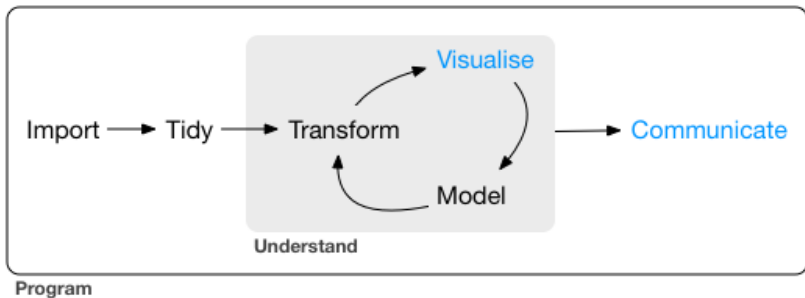
onde estamos?

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot



Uma exibição gráfica deve (1/2)

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

Mostrar os dados

Induzir o observador a pensar em sua substância, não em metodologia ou tecnologia de produção

Evitar distorcer o que os dados dizem

Apresentar muitos números em pequenos espaços

Tornar grandes conjuntos de dados coerentes

Uma exibição gráfica deve (2/2)

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

Encorajar o observador a comparar diferentes partes dos dados

Revelar diferentes níveis de detalhamento dos dados

Servir a um propósito claro e razoável: descrição, exploração, tabulação ou decoração

Estar integrada com as descrições estatísticas e verbais do conjunto de dados

4 princípios (Edward Tufte)

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

Miniaturas Múltiplas

Menor diferença efetiva

Causalidade (Respondendo a pergunta: “Comparado com o quê?”)

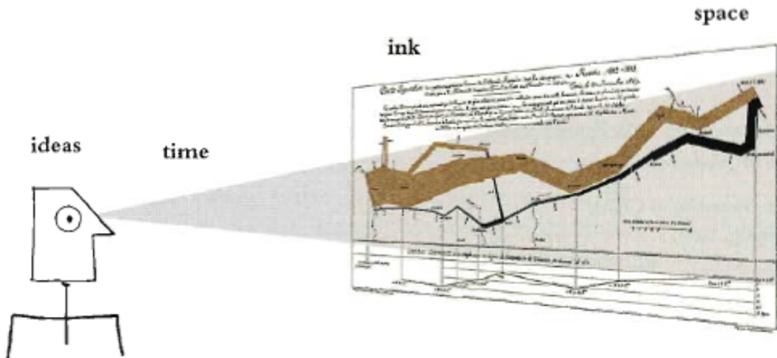
Contexto

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot



O que você quer mostrar?

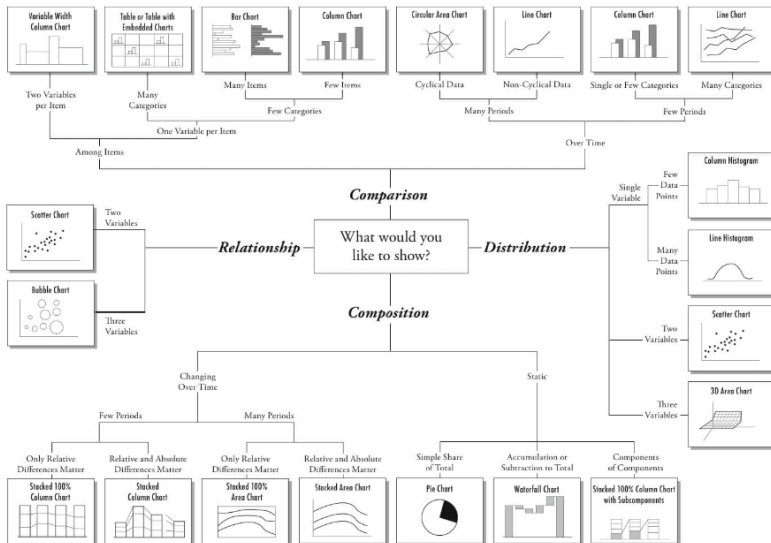
ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

Chart Suggestions—A Thought-Starter

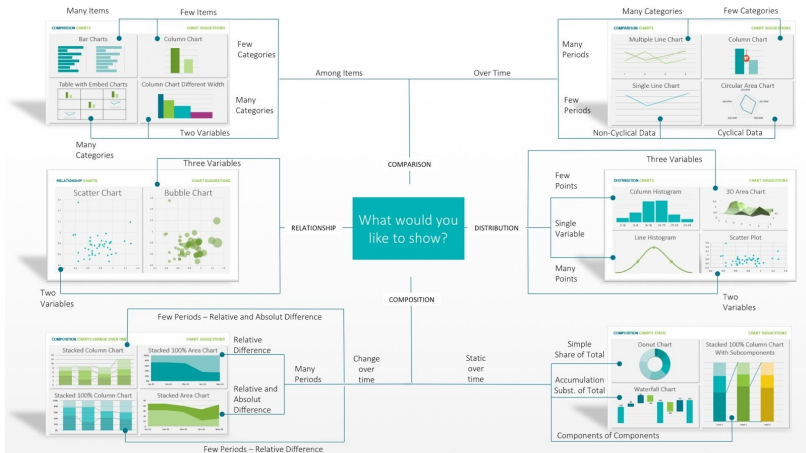


ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot



Animado

Princípios

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

- O que você quer mostrar?
- Elementos que podem **destacar** ou **confundir** o que você quer mostrar.
- vamos tentar alternar “teoria” com live code
- Ah, mas eu posso usar base R? Poder, pode. . .

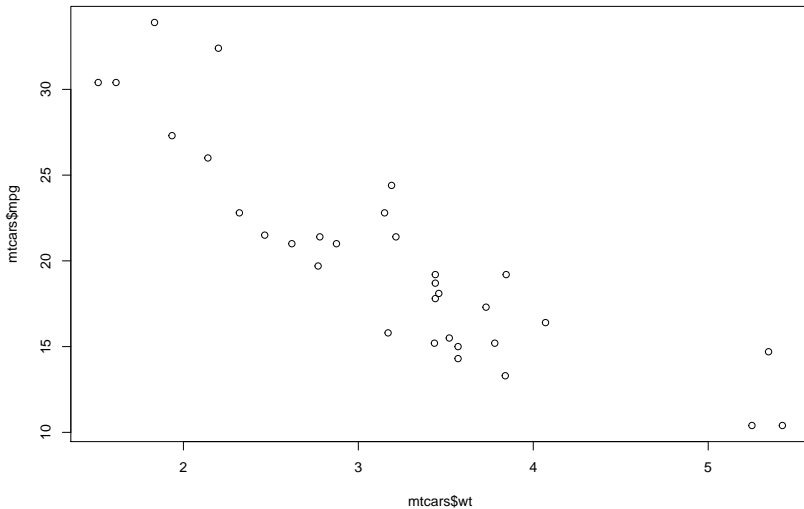
ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

```
plot(mtcars$wt, mtcars$mpg)
```



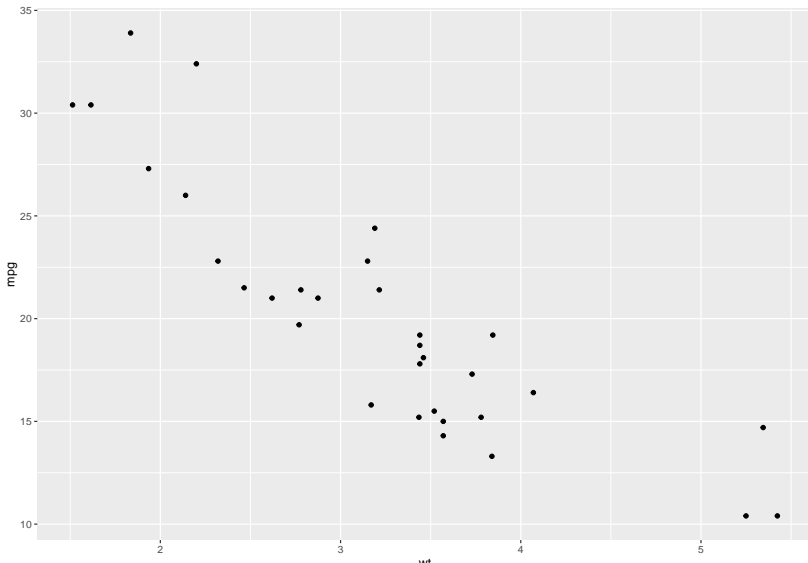
ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

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



**ggplot e
extensões**

**Frederico
Bertholini**

Visualização
de dados

ggplot

ggplot

Recursos

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

- R Graphics Cookbook
- R Graph Gallery
- STHDA
- Fundamentals of Data Visualization
- r-statistics
- Extensões

Elementos do ggplot

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

- Dados
- Geometrias
- Estéticas
- Escalas (estética)
- Escalas (eixos)
- Tema
- Facet

Dados data =

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

- Dado empilhado?
- Cada coluna será uma entrada!

Geometrias geom_

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

- geom_tipo_de_geometria
- Recursos +
- cheat sheet
- manual ggplot

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 **aesthetics** like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

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

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

aesthetic mappings **data** **geom**
geom(x = cty, 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 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 = year))

a + geom_blank()
(Useful for expanding limits)

a + geom_curve(aes(yend = lat + 1,
  xend = long - 1, curvature = 1), x = wind, y = yend,
  alpha, angle, color, curvature, linetype, size)

a + geom_path(linetype = "butt", linetype = "round",
  linetype = "solid")
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)) - xmax, xmin, ymax,
  ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin = unemploy - 900,
  ymax = unemploy + 900)) - x, ymax, ymin,
  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 = lat))
b + geom_vline(aes(intercept = long))

b + geom_segment(aes(yend = lat + 1, xend = long - 1))
b + geom_spoke(aes(angle = 1:155, radius = 1))
```

ONE VARIABLE continuous

```
c <- ggplot(mpg, aes(hwy))
c2 <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size

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

c + geom_histogram(binwidth = 5)
x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy))
x, y, alpha, color, fill, linetype, size, weight
```

discrete

```
d <- ggplot(mpg, aes(flr))
d + geom_bar()
x, alpha, color, fill, linetype, size, weight
```

TWO VARIABLES

```
continuous x, continuous 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 = "bt", 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
```

discrete x, continuous y

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

f + geom_col()
x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot()
x, y, lower, middle, upper,
  ymax, ymin, alpha, color, fill, group, linetype,
  shape, size, weight

f + geom_dotplot(binwidth = "5", stackdir =
  "center")
x, y, alpha, color, fill, group

f + geom_violin(scale = "area")
x, y, alpha, color, fill, group, linetype, size, weight
```

discrete x, discrete y

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

g + geom_count()
x, y, alpha, color, fill, shape, size, stroke
```

THREE VARIABLES

```
seals$z <- with(seals, sort(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(z = z), hjust = 0.5, vjust = 0.5,
  interpolate = FALSE)
x, y, alpha, fill

l + geom_tile(aes(z = z))
x, y, alpha, color, fill, linetype, size, width
```

continuous bivariate distribution

```
h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500))
x, y, alpha, color, fill, linetype, size, weight

h + geom_density2d()
x, y, alpha, color, group, linetype, size

h + geom_hex()
x, y, alpha, colour, fill, size
```

continuous function

```
i <- ggplot(economics, aes(date, unemploy))

i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv")
x, y, alpha, color, group, linetype, size
```

visualizing error

```
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_crewbar(latten = 2)
x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar()
x, y, ymax, ymin, alpha, color, fill, group, linetype, size, weight

j + geom_linerange()
x, y, ymax, ymin, alpha, color, group, linetype, size

j + geom_pointrange()
x, y, ymax, ymin, alpha, color, fill, group, linetype, shape, size
```

maps

```
k <- data.frame(murder = USArrests$Murder,
  state = tolower(names(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)
```

geometrias

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

<https://ggplot2.tidyverse.org/reference/index.html#section-layers>

Estéticas `aes()`

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

- `x` (`xmax` e `xmin`)
- `y` (`ymax` e `ymin`)
- `color`
- `fill`
- `shape`
- `group`
- `size`

Escalas (estética) scale_

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

- `scale_color_xx`
- `scale_fill_xx`
- `scale_shape_xx`

Escalas (eixos) `scale_x`

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

- Contínua `scale_x_continuous`
- Discreta `scale_x_discrete`
- Tempo `scale_yearmon`
- Série de tempo `zoo` `lubridate` -> `scale_yearmon`

Tema

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

- Customização total da visualização
- Eixos
- Texto `element_text`
- linhas de grade

Facet

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

- facet_grid
- facet_wrap

Adicionais

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

Gráficos com interatividade:

- ggiraph
- plotly (ggplotly)

Combinação de gráficos

- patchwork
- cowplot

Exercício

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

- Carregue os dados de exemplo do pacote `survey data(api)`, use o `data.frame apisrs`
- Crie o objeto `tbl_svy` com o nome `amostra_expandida` expandindo a amostra aleatória simples usando apenas a variável (coluna) “`pw`”, contendo o peso amostral. Dica: execute `as_survey(weight=pw)`.
- Usando a variável `stype` crie uma nova variável indicando se a escola é de nível fundamental (categorias **E** e **M** de `stype`) ou de nível médio (categoria *H* de `stype`). Dica: use `mutatee case_when`.
- Faça um gráfico de barras comparando a variação média das notas de 1999 (`api99`) e 2000 (`api00`) por nível e utilize as estimativas intervalares. Dica: olhe o código da aula 07, utilize `geom_errorbar` para a estimativa intervalar.

Resolução

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

```
data(api)

amostra_expandida <- apisrs %>%
  as_survey(weight = pw) %>%
  mutate(nivel=case_when(
    stype=="E"~"Fundamental",
    stype=="M"~"Fundamental",
    stype=="H"~"Médio"
  ))
```

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

```
out <- amostra_expandida %>%  
  group_by(nivel) %>%  
  summarise(api_diff =  
    survey_mean(api00 - api99, vartype = "ci"))
```

ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

```
grafico <- out %>%  
  ggplot(aes(x = nivel, y = api_diff,  
             fill = nivel,color=nivel,  
             ymax = api_diff_upp,  
             ymin = api_diff_low)) +  
  geom_bar(stat = "identity",alpha=0.6) +  
  geom_errorbar(width = 0,size=3)
```

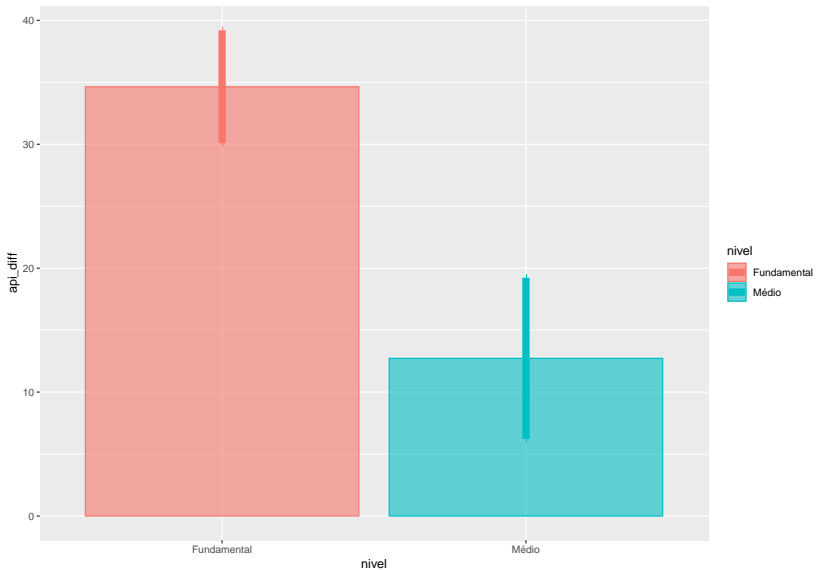
ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

grafico



ggplot e
extensões

Frederico
Bertholini

Visualização
de dados

ggplot

```
grafico + labs(y="Variação das notas",x="",color="Nível",fill="Nível")
```

