Insper EDUCAÇÃO EXECUTIVA ONLINE

Live Learning AO VIVO

Programação para Não Programadores

Aula 6

Prof. Magno Severino e Prof. Marina Muradian

29/04/2021

Objetivo de aprendizagem

- Identificar os componentes do pacote `ggplot2` para construção de gráficos.

Referências

- https://ggplot2-book.org/
- https://www.r-graph-gallery.com/index.html
- https://www.data-to-viz.com/
- https://rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

Por que usar o `ggplot2`?

Grammar of Graphics - o "gg" do `ggplot2`:

- analogia à gramática de uma língua: formação de frases a partir de alguns verbos, substantivos e adjetivos-chave
- conhecimento das *layers* do 'ggplot2' e sua gramática permite a criação de diversos gráficos
- diminui a necessidade de memorização

Layers: sintaxe intuitiva e relativamente simples de lembrar

Abrangência: o *default* do `ggplot2` satisfaz a grande maioria dos casos e é visualmente interessante.

O que é a *Grammar of Graphics*?

Um gráfico é um mapeamento de **dados** para **atributos estéticos** (*aesthetics*: cor, forma, tamanho) de **objetos geométricos** (pontos, linhas, barras)

Também pode conter **transformações estatísticas** (*stats*) dos dados e é desenhado num **sistema de coordenadas** específico.

Facetas podem ser utilizadas para gerar o mesmo gráfico para diferentes subconjuntos da base de dados

E o que NÃO é?

A *Grammar of Graphics* não sugere quais gráficos devem ser utilizados para cada tipo de dados e/ou objetivo da visualização (para sugestões, visite os sites listados nas Referências)

`ggplot2` - Componentes Básicos

- ***data***: conjunto de dados a ser visualizado no gráfico
- ***geometry***: tipo de gráfico (scatterplot, boxplot, barplot, histogram, qqplot, smooth density, etc.)
- ***aesthetics***: aspectos visuais (*visual cues*) de mapeamento de variáveis, como eixos x e y e cores.

Dados - atrasos x distância dos voos por destino

Vamos usar novamente a base de dados `flights`

e fazer uma tabela com o atraso médio e distância média dos vôos de cada destino:

```
library(tidyverse)

atrasos <- flights %>%
    filter(!is.na(distance), !is.na(arr_delay)) %>%
    group_by(dest) %>%
    summarise(distance = mean(distance),
              delay = mean(arr_delay))
head(atrasos)
```

```
## # A tibble: 6 x 3
##   dest  distance delay
##   <fct>    <dbl> <dbl>
## 1 ABQ      1826   4.38
## 2 ACK       199   4.85
## 3 ALB       143  14.4
## 4 ANC      3370  -2.5
## 5 ATL       757. 11.3
## 6 AUS      1514.  6.02
```

Vamos supor que nosso objetivo seja entender a relação entre distância do voo e atraso. Para isso, utilizaremos um **gráfico de dispersão**.

Criando um objeto ggplot

e gerar um objeto ggplot:

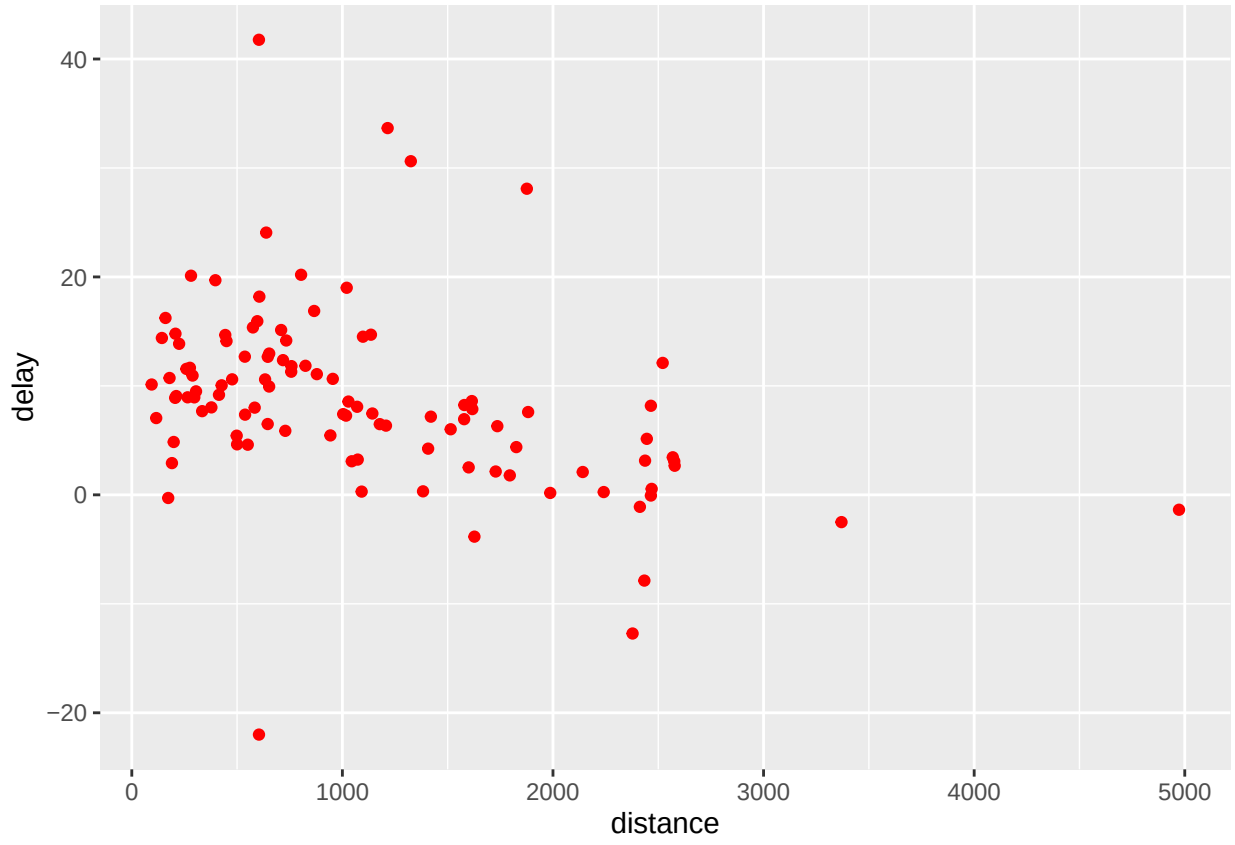
```
atrasos %>% ggplot()
```


Note que o objeto `ggplot` criado é só uma tela em branco. Isso ocorre, pois dos 3 componentes básicos, só informamos a base de dados `data`.

Gráfico de Dispersão

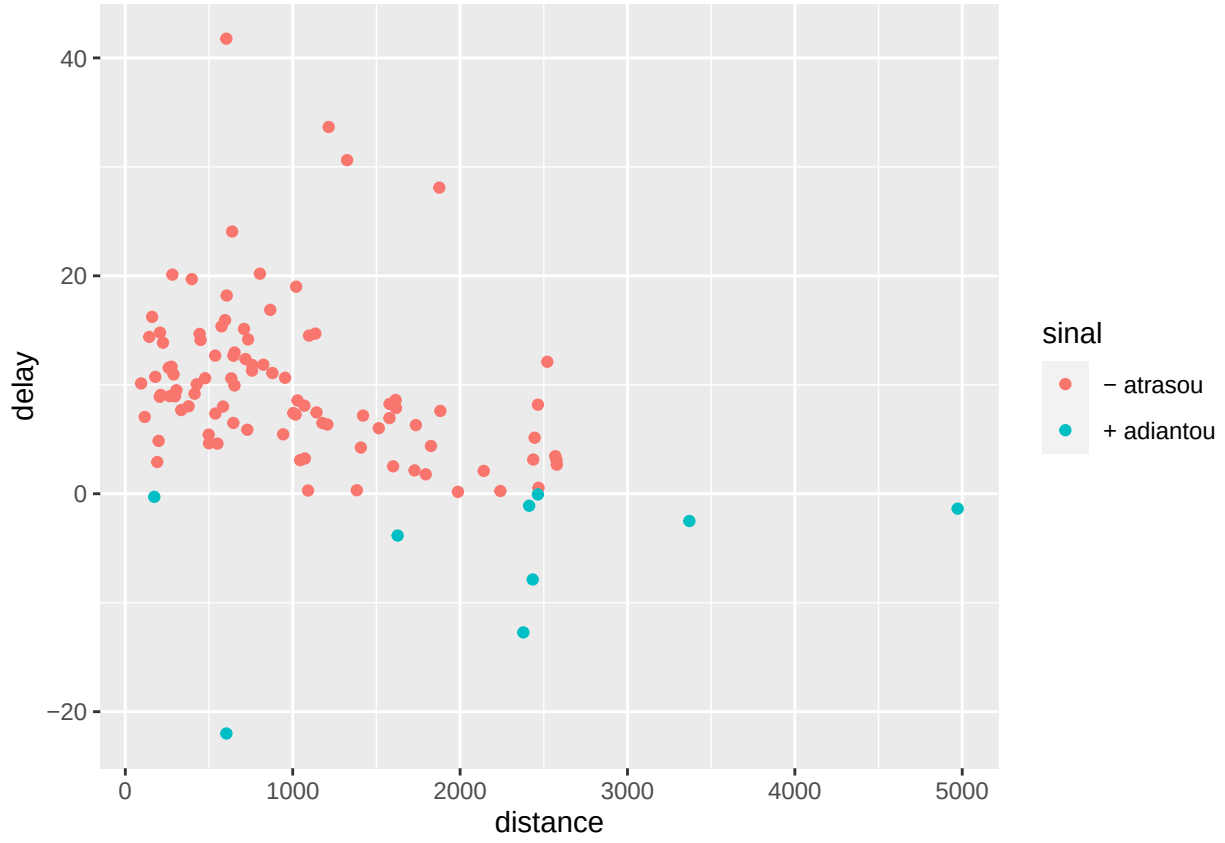
Como o nosso objetivo é criar um gráfico de dispersão, vamos utilizar a *geometry* `geom_point`, e informar o mapeamento das variáveis (x e y) dentro do componente de *aesthetics*:

```
atrasos %>%
  ggplot(aes(x = distance, y = delay))+
  geom_point(color = "red")
```



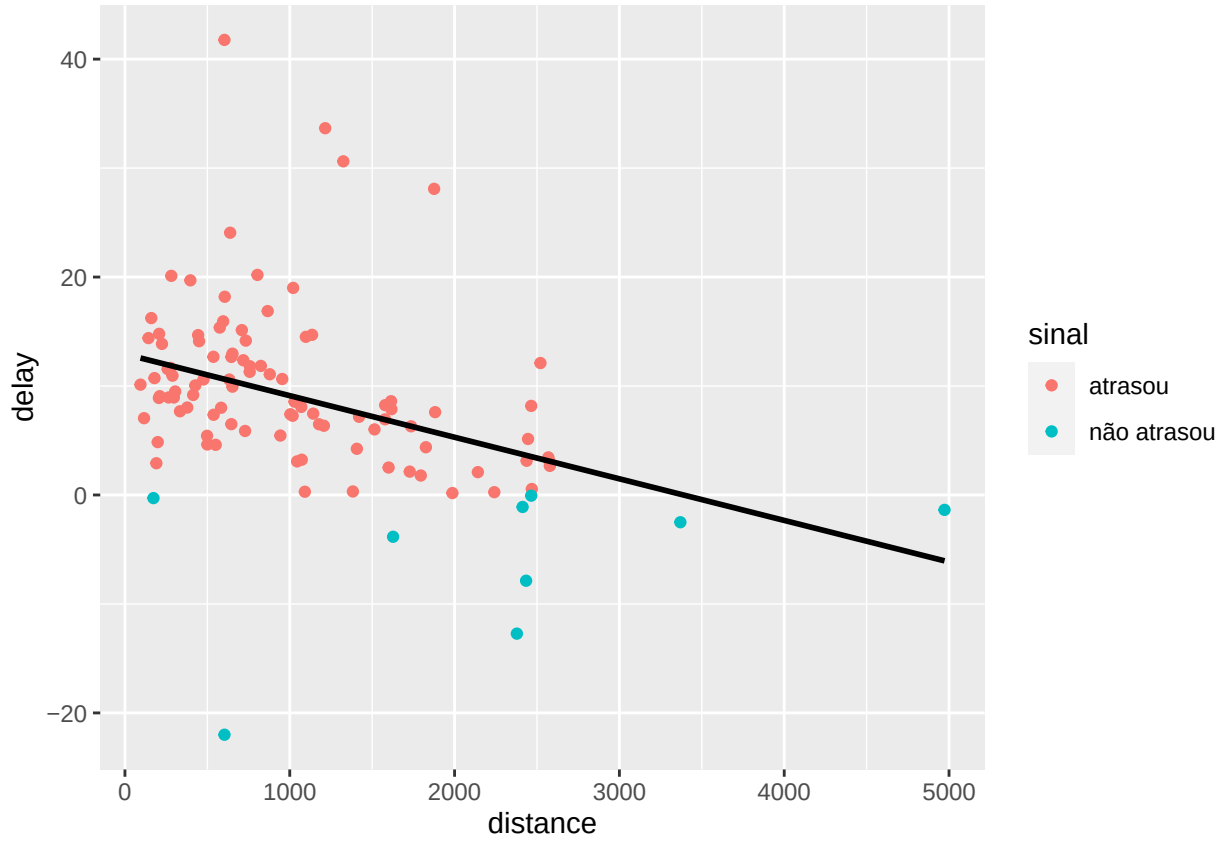
Podemos criar uma variável que indica se foi atraso ou adiantamento e colocar a cor dos pontos de acordo com ela:

```
atrasos %>%
mutate(sinal = ifelse(delay > 0, "- atrasou", "+ adiantou")) %>%
  ggplot(aes(x = distance, y = delay))+
  geom_point(aes(color = sinal))
```



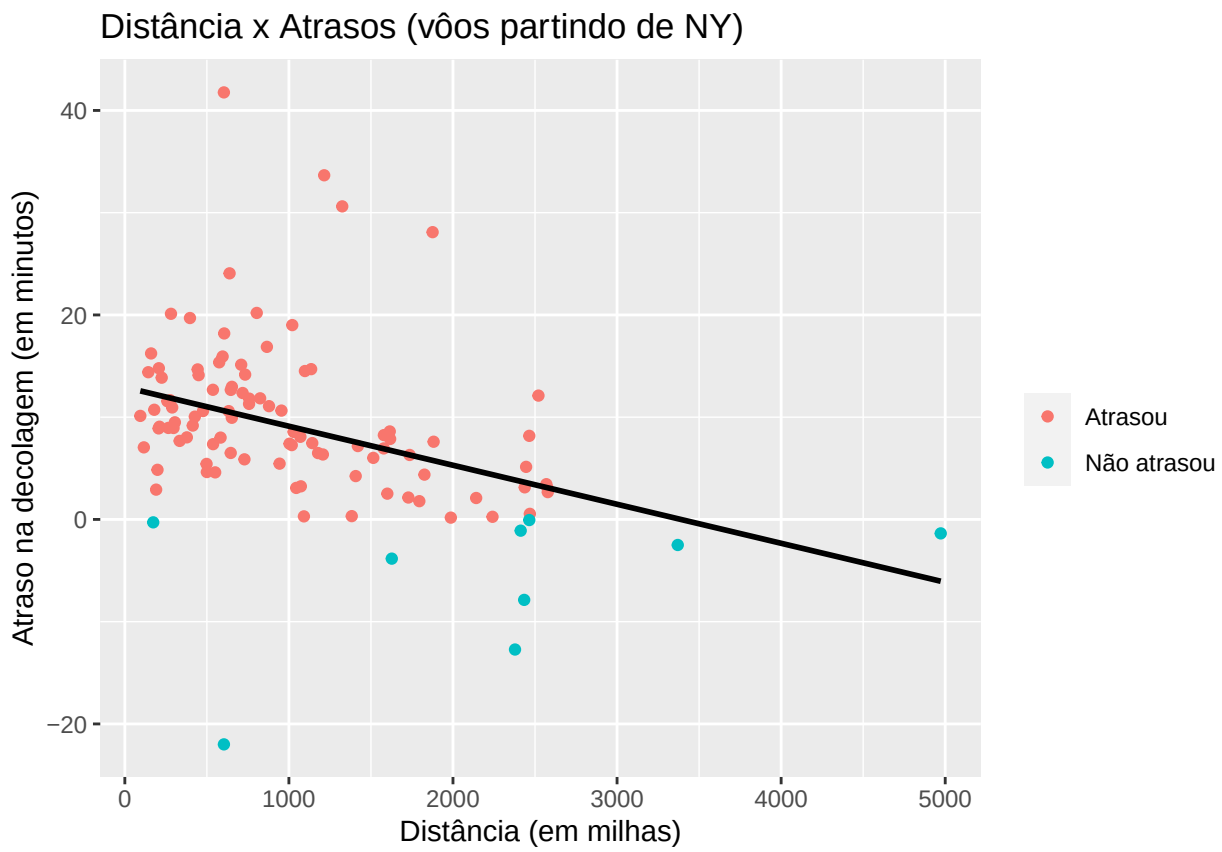
E adicionar uma *layer* com a linha de tendência (equação da reta):

```
atrasos %>%
mutate(sinal = ifelse(delay > 0, "atrasou", "não atrasou")) %>%
  ggplot(aes(x = distance, y = delay))+
  geom_point(aes(color = sinal))+
  geom_smooth(method = "lm",
              se = FALSE,
              color = "black")
```



Vamos editar o título do gráfico e dos eixos:

```
atrasos %>%
mutate(sinal = ifelse(delay > 0, "Atrasou", "Não atrasou")) %>%
  ggplot(aes(x = distance, y = delay))+
  geom_point(aes(color = sinal))+
  geom_smooth(method = "lm",
              se = FALSE,
              color = "black")+
  labs(title = "Distância x Atrasos (vôos partindo de NY)",
       x = "Distância (em milhas)",
       y = "Atraso na decolagem (em minutos)",
       color = "")
```



E finalmente, podemos alterar a aparência geral do gráfico:

```
atrasos %>%
mutate(sinal = ifelse(delay > 0, "Atrasou", "Não atrasou")) %>%
  ggplot(aes(x = distance, y = delay))+
  geom_point(aes(color = sinal))+
  geom_smooth(method = "lm",
              se = FALSE,
              color = "black")+
  labs(title = "Distância x Atrasos (vôos partindo de NY)",
       x = "Distância (em milhas)",
       y = "Atraso na decolagem (em minutos)",
       color = "")+
  theme_minimal()
```

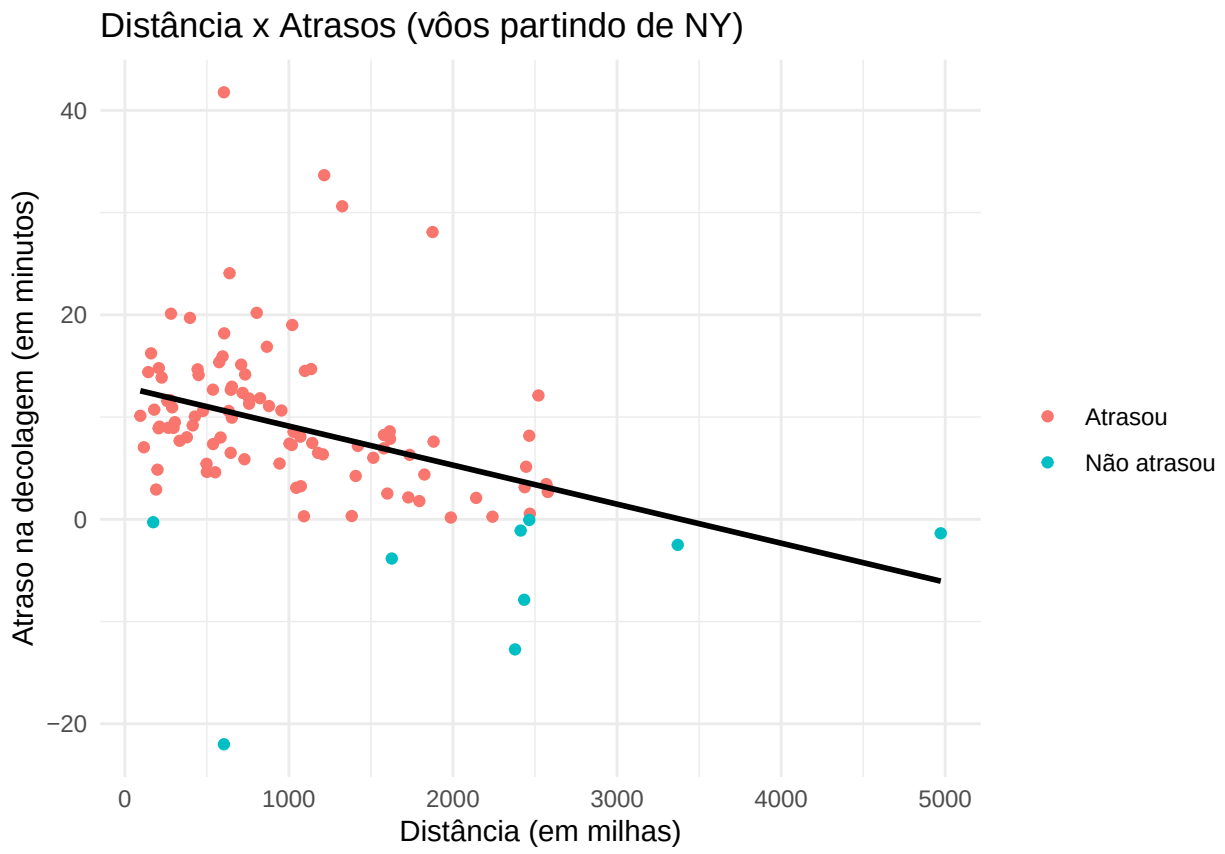


Gráfico de Colunas

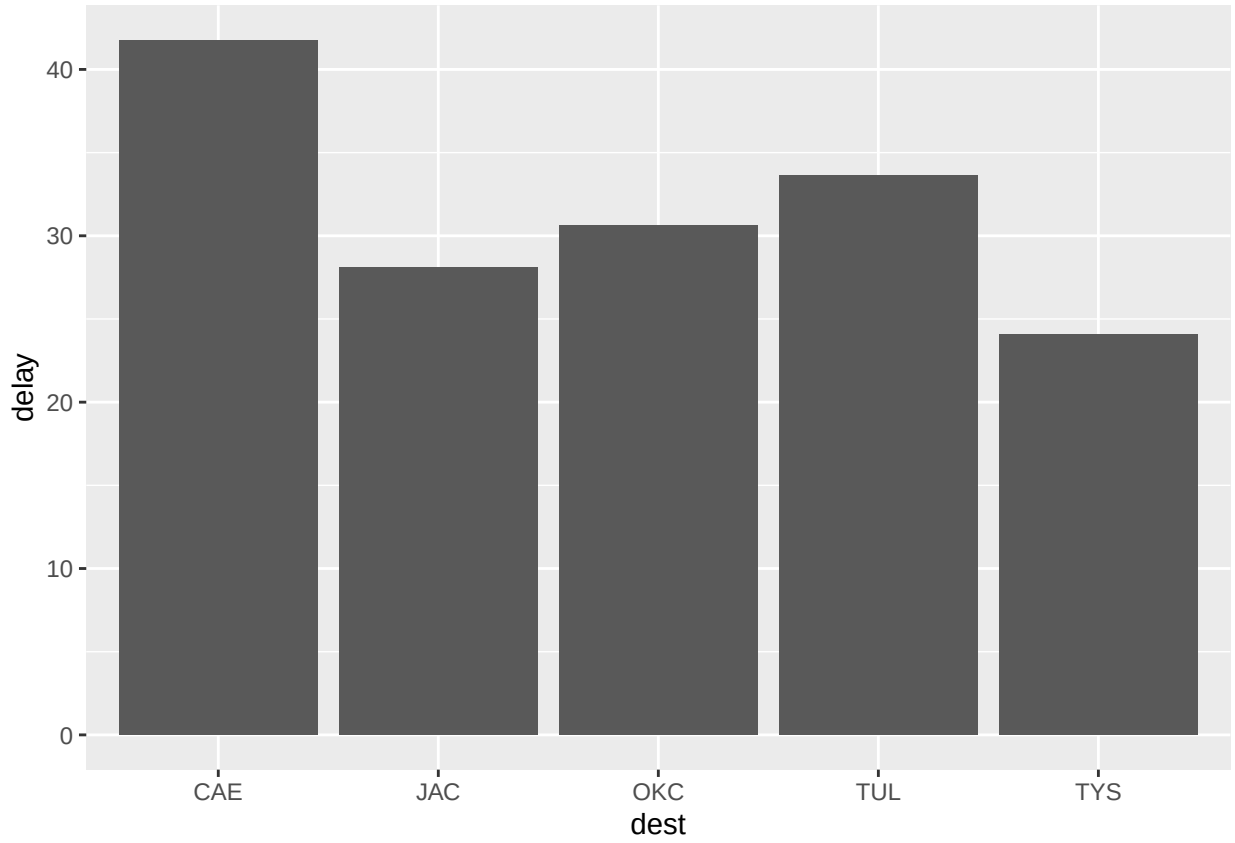
Vamos supor agora, que meu interesse seja em fazer um gráfico de colunas comparando o atraso médio dos 5 top destinos que mais atrasam:

```
top_5 <- atrasos %>% top_n(5, delay)
top_5
```

```
## # A tibble: 5 x 3
##   dest  distance delay
##   <fct>    <dbl> <dbl>
## 1 CAE       604.  41.8
## 2 JAC      1876.  28.1
## 3 OKC      1325   30.6
## 4 TUL      1215   33.7
## 5 TYS       638.  24.1
```

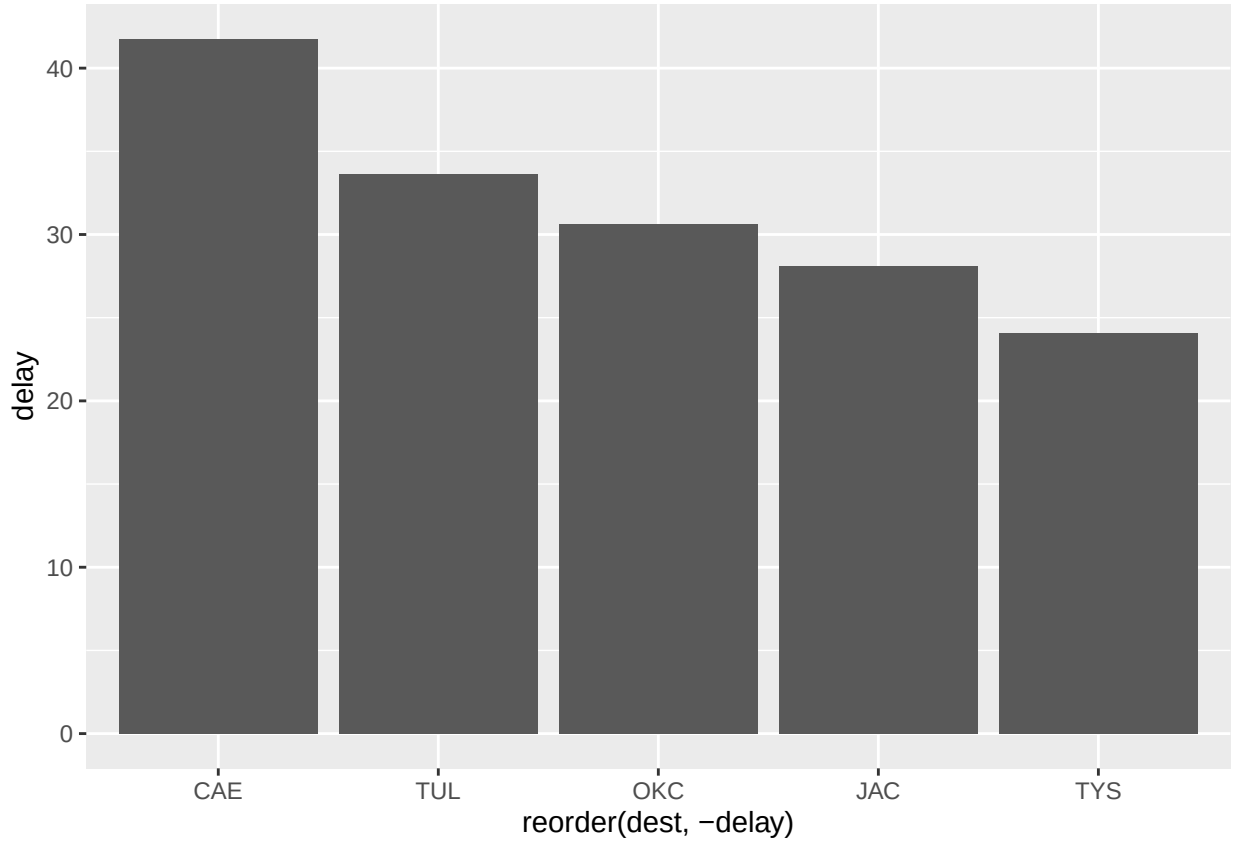
Construindo nosso gráfico de colunas `geom_col()`:

```
top_5 %>%
  ggplot(aes(x = dest, y = delay))+
  geom_col()
```



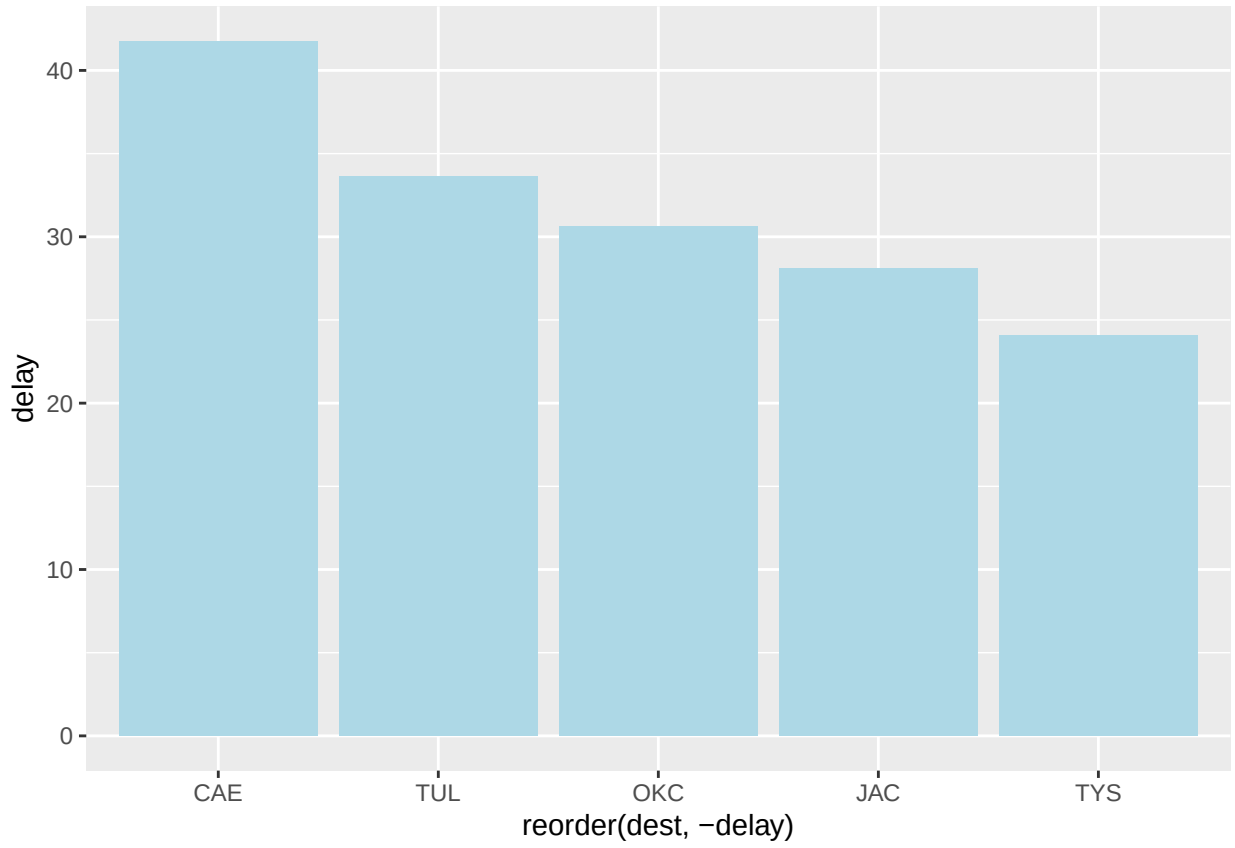
Perceba que as colunas estão organizadas por ordem alfabetica. Para colocarmos por ordem decrescente da variável delay, podemos usar o `reorder`:

```
top_5 %>%
  ggplot(aes(x = reorder(dest, -delay), y = delay))+
  geom_col()
```



Podemos pintar todas as colunas de uma cor só:

```
top_5 %>%
  ggplot(aes(x = reorder(dest, -delay), y = delay))+
  geom_col(fill = "lightblue")
```



Alterar os títulos e mudar a aparência:

```
top_5 %>%
  ggplot(aes(x = reorder(dest, -delay), y = delay))+
  geom_col(fill = "lightblue")+
  labs(title = "Destinos com maior atraso médio (top 5)",
       y = "Atraso (em minutos)",
       x = "")+
  theme_minimal()
```