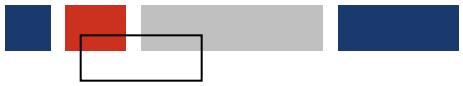
INSTITUTO NACIONAL DE ESTATÍSTICA
STATISTICS PORTUGAL

» R para a Ciência dos Dados

Pedro Sousa
João Lopes

DMSI / ME



Outubro de 2019

INSTITUTO NACIONAL DE ESTATÍSTICA
STATISTICS PORTUGAL

» R para a Ciência dos Dados

https://r4ds.had.co.nz/

Pedro Sousa
João Lopes

DMSI / ME

Outubro de 2019

Programa

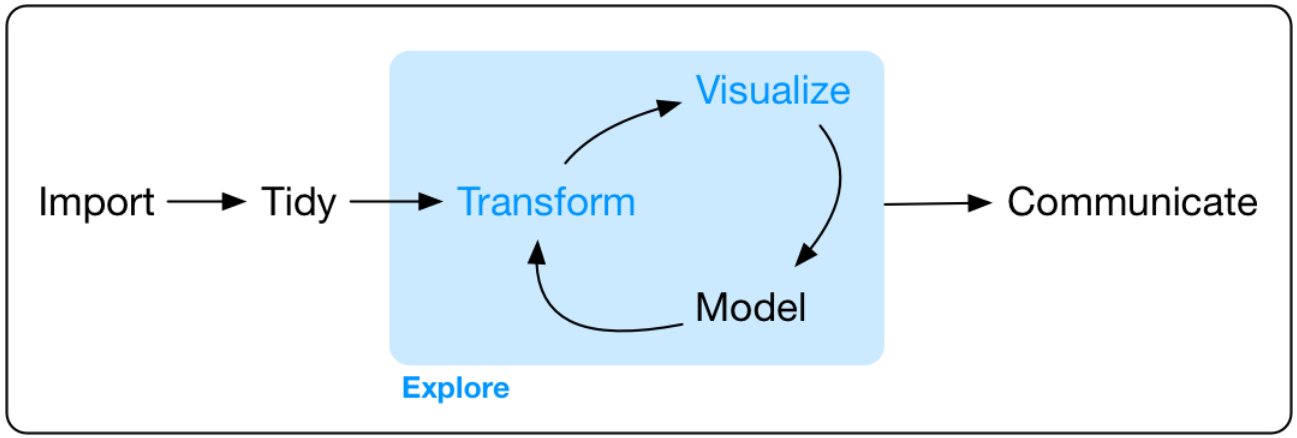
- Introdução
- Noções básicas do R
- Data *Wrangling*
- **Exploração dos dados**
- **Modelos e inferências**
- Comunicação

Programa

- **Exploração dos dados**
  - Visualização
  - Manipulação de dados
  - Exploração
- **Modelos e inferências**
  - Exemplo
  - Ajustamento
  - Diagnóstico

Exploração dos dados

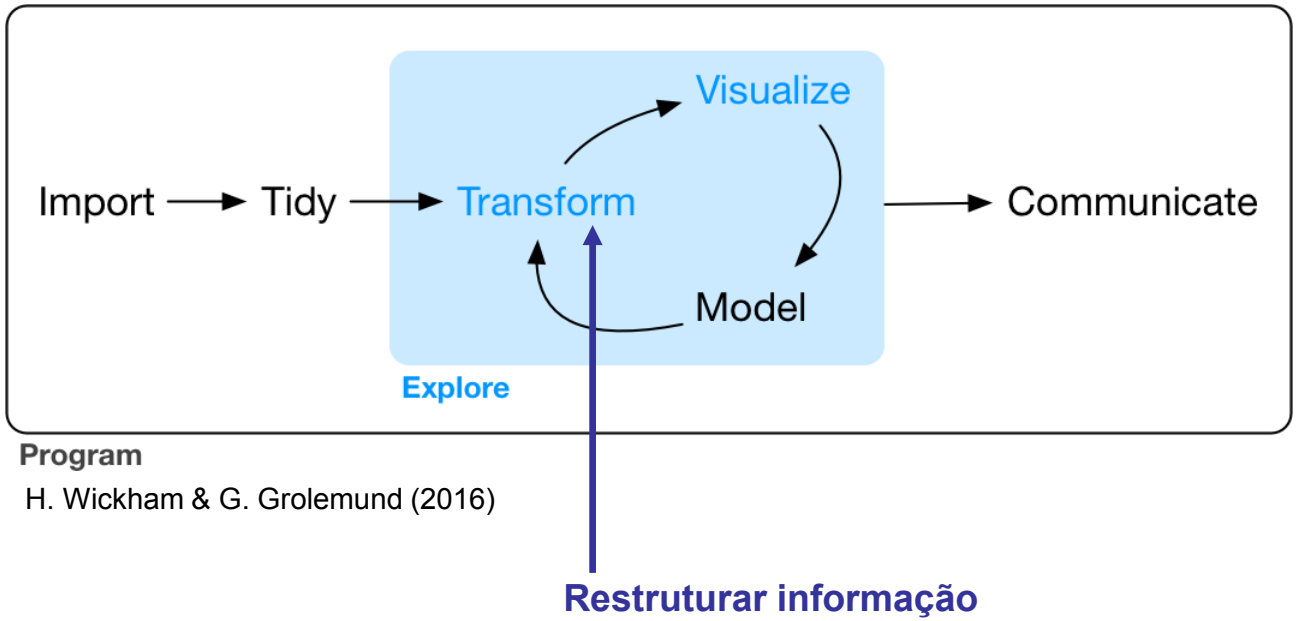
» Esquema geral



H. Wickham & G. Grolemund (2016)

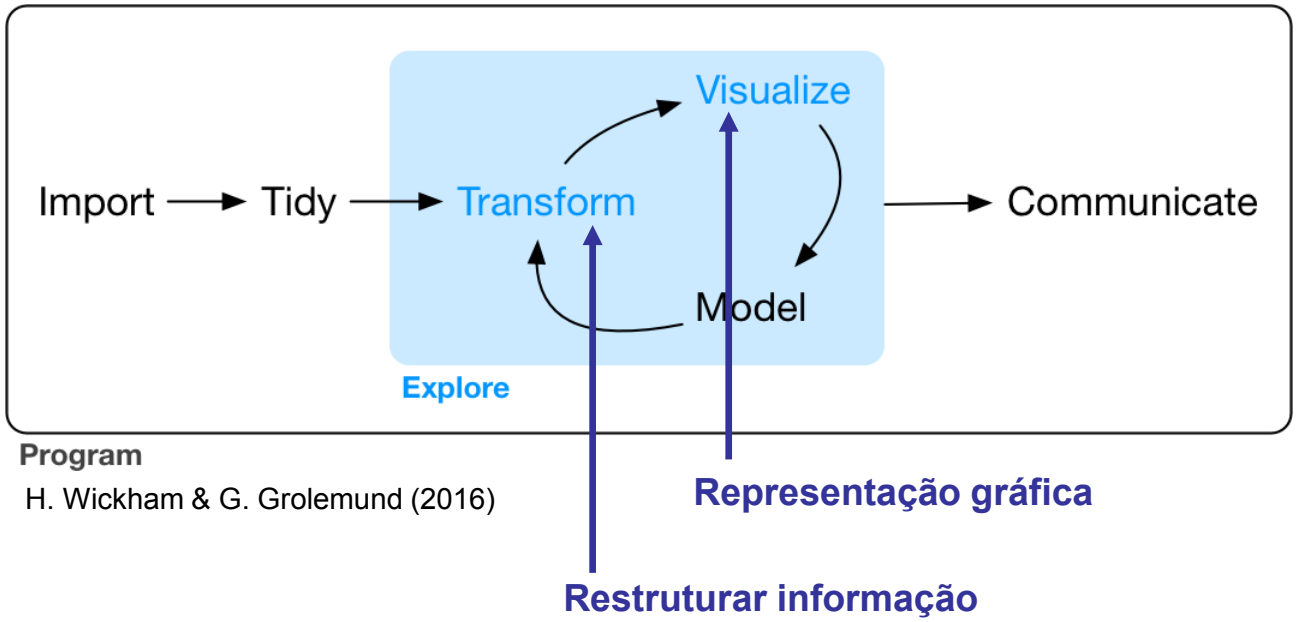
Exploração dos dados

» Esquema geral



Exploração dos dados

» Esquema geral



Exploração dos dados

» 1. Visualização (`ggplot2`):

Exploração dos dados

» 1. Visualização (`ggplot2`):

- *Scatterplot* (e *smoothplot*);
- Gráfico de barras;
- *Boxplot*;
- Histograma (e curva de densidade).

Exploração dos dados

» 1.1. Visualização: *scatterplot*

Exploração dos dados

» 1.1. Visualização: *scatterplot*

```
library("tidyverse")
?mpg
print(mpg)                                                    #table
```

Exploração dos dados

» 1.1. Visualização: *scatterplot*

```
# A tibble: 234 x 11
   manufacturer model      displ  year   cyl trans      drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
 1 audi         a4           1.8  1999     4 auto(l5)   f        18    29 p     compact
 2 audi         a4           1.8  1999     4 manual(m5) f        21    29 p     compact
 3 audi         a4           2    2008     4 manual(m6) f        20    31 p     compact
 4 audi         a4           2    2008     4 auto(av)   f        21    30 p     compact
 5 audi         a4           2.8  1999     6 auto(l5)   f        16    26 p     compact
 6 audi         a4           2.8  1999     6 manual(m5) f        18    26 p     compact
 7 audi         a4           3.1  2008     6 auto(av)   f        18    27 p     compact
 8 audi         a4 quattro   1.8  1999     4 manual(m5) 4        18    26 p     compact
 9 audi         a4 quattro   1.8  1999     4 auto(l5)   4        16    25 p     compact
10 audi         a4 quattro   2    2008     4 manual(m6) 4        20    28 p     compact
# ... with 224 more rows
```

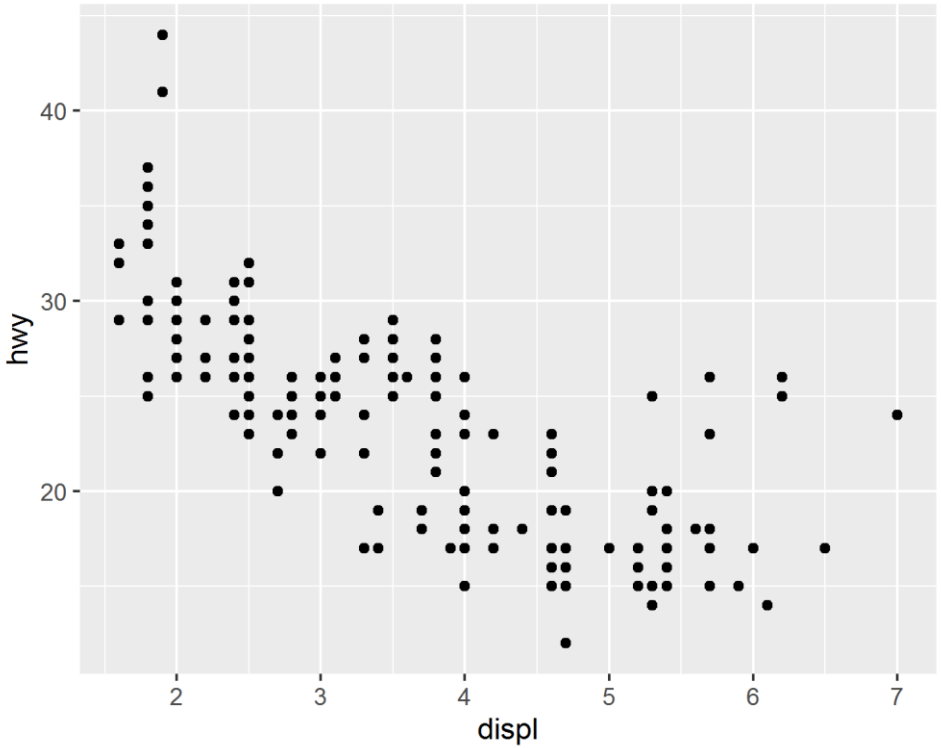
Exploração dos dados

» 1.1. Visualização: *scatterplot*

```
library("tidyverse")
?mpg
print(mpg)                                                    #table
ggplot(data=mpg) + geom_point(mapping=aes(x=displ,y=hwy))     #scatterplot
```

Exploração dos dados

» 1.1. Visualização: *scatterplot*



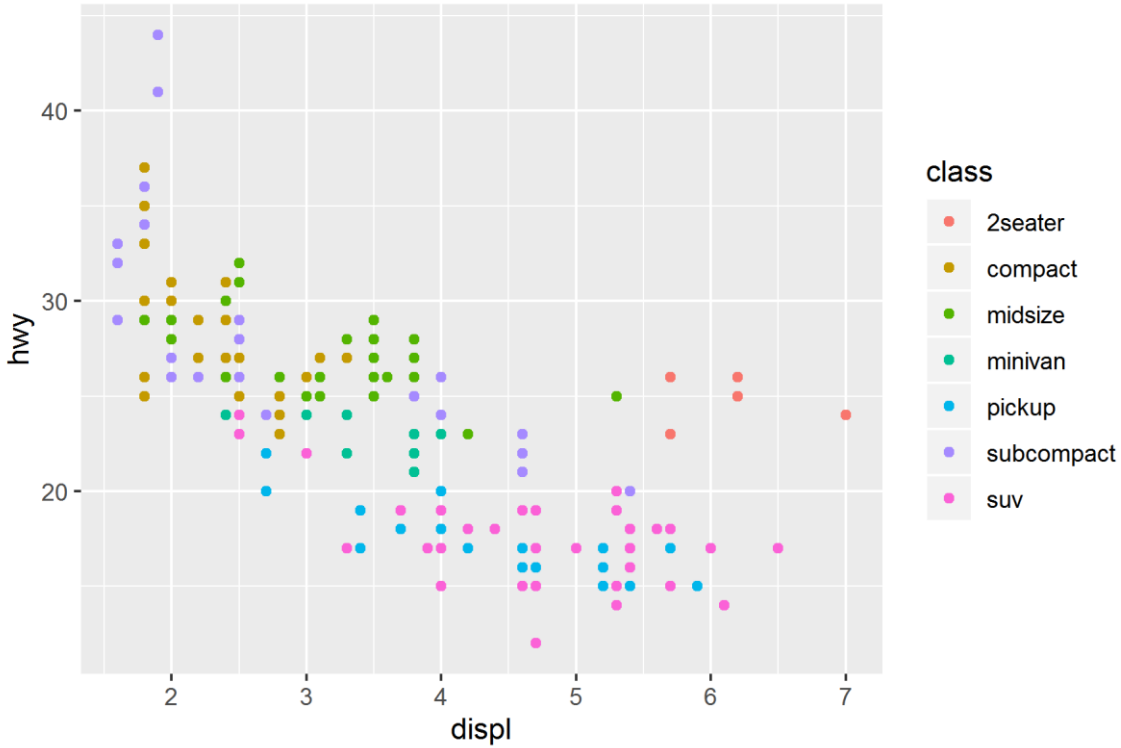
Exploração dos dados

» 1.1. Visualização: *scatterplot*

```
library("tidyverse")
?mpg
print(mpg)                                                    #table
ggplot(data=mpg) + geom_point(mapping=aes(x=displ,y=hwy))     #scatterplot
ggplot(data=mpg) + geom_point(mapping=aes(x=displ,y=hwy,color=class))   #w/ color
```

Exploração dos dados

» 1.1. Visualização: *scatterplot*



Exploração dos dados

» 1.2. Visualização: *smoothplot*

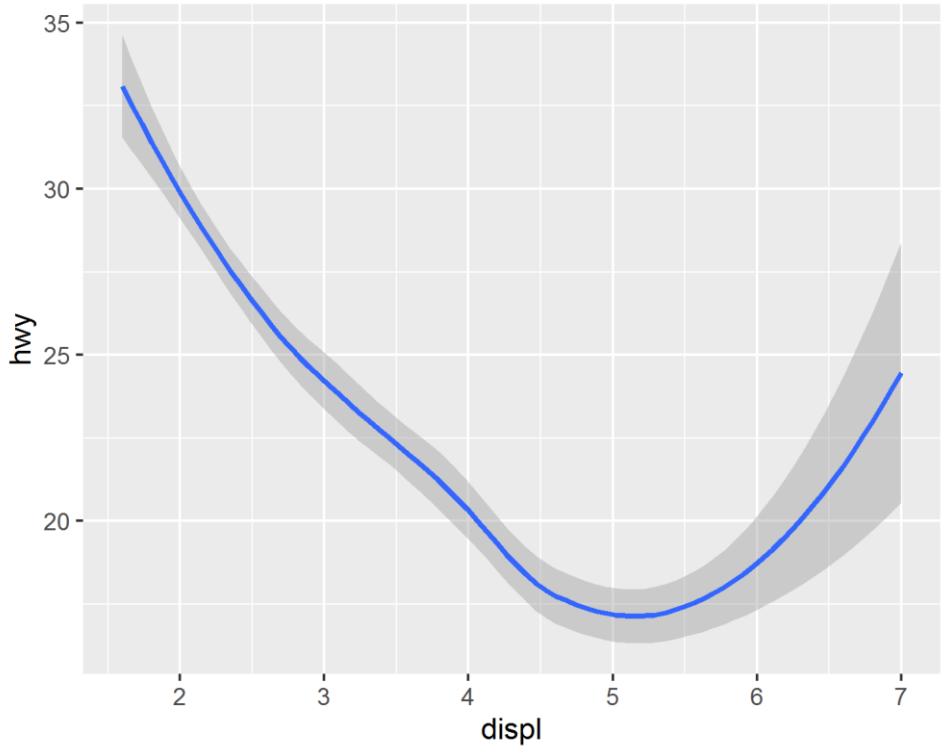
Exploração dos dados

» 1.2. Visualização: *smoothplot*

```
library("tidyverse")
?mpg
print(mpg)                                                          #table
ggplot(data=mpg) + geom_point(mapping=aes(x=displ,y=hwy))           #scatterplot
ggplot(data=mpg) + geom_point(mapping=aes(x=displ,y=hwy,color=class)) #w/ color
ggplot(data=mpg) + geom_smooth(mapping=aes(x=displ,y=hwy))          #smoothplot
```

Exploração dos dados

» 1.2. Visualização: *smoothplot*



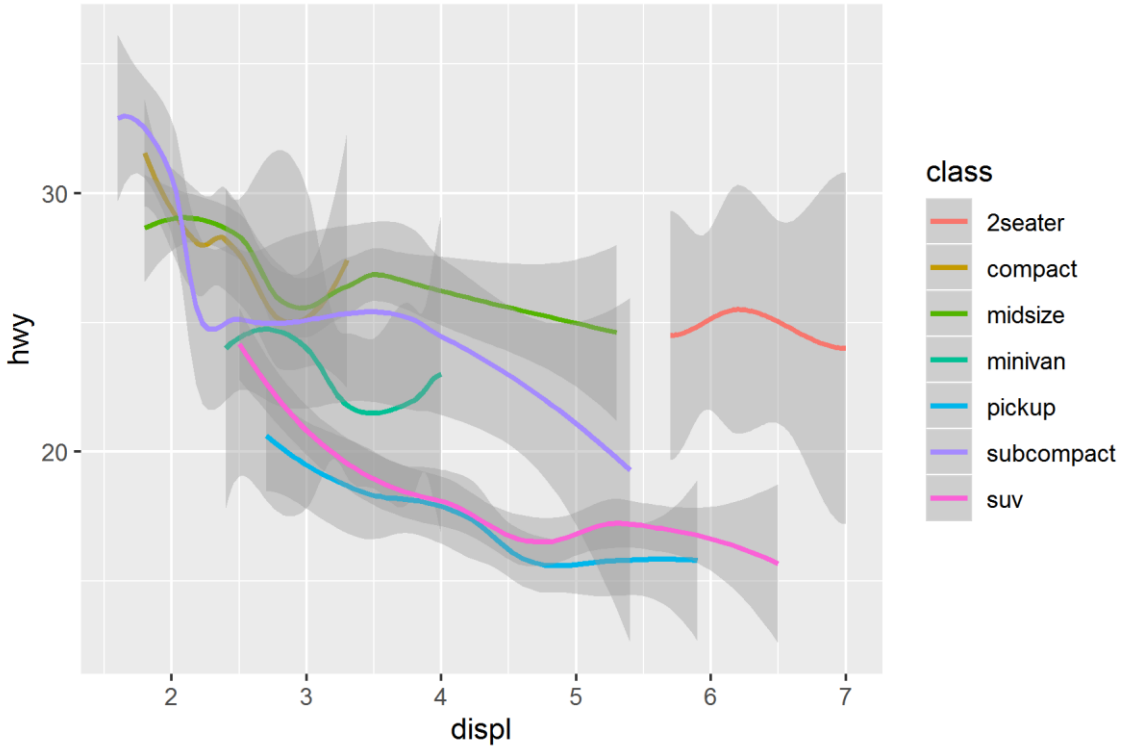
Exploração dos dados

» 1.2. Visualização: *smoothplot*

```
library("tidyverse")
?mpg
print(mpg)                                                          #table
ggplot(data=mpg) + geom_point(mapping=aes(x=displ,y=hwy))           #scatterplot
ggplot(data=mpg) + geom_point(mapping=aes(x=displ,y=hwy,color=class)) #w/ color
ggplot(data=mpg) + geom_smooth(mapping=aes(x=displ,y=hwy))          #smoothplot
ggplot(data=mpg) + geom_smooth(mapping=aes(x=displ,y=hwy,color=class)) #w/ color
```

Exploração dos dados

» 1.2. Visualização: *smoothplot*



Exploração dos dados

» 1.3. Visualização: gráfico de barras

Exploração dos dados

» 1.3. Visualização: gráfico de barras

```
library("tidyverse")
?diamonds
print(diamonds)                                                    #table
```

Exploração dos dados

» 1.3. Visualização: gráfico de barras

```
# A tibble: 53,940 x 10
   carat cut       color clarity depth table price     x     y     z
   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
 7 0.24  Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
 8 0.26  Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
 9 0.22  Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
10 0.23  Very Good H     VS1      59.4    61   338  4     4.05  2.39
# ... with 53,930 more rows
```

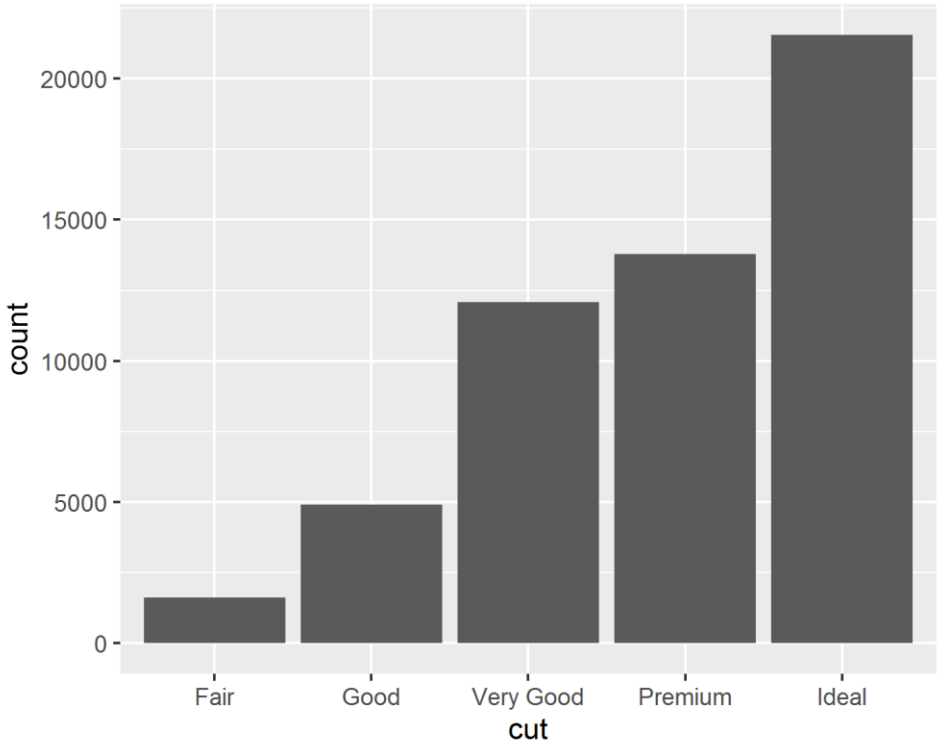
Exploração dos dados

» 1.3. Visualização: gráfico de barras

```
library("tidyverse")
?diamonds
print(diamonds)                                                    #table
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut))               #barplot
```

Exploração dos dados

» 1.3. Visualização: gráfico de barras



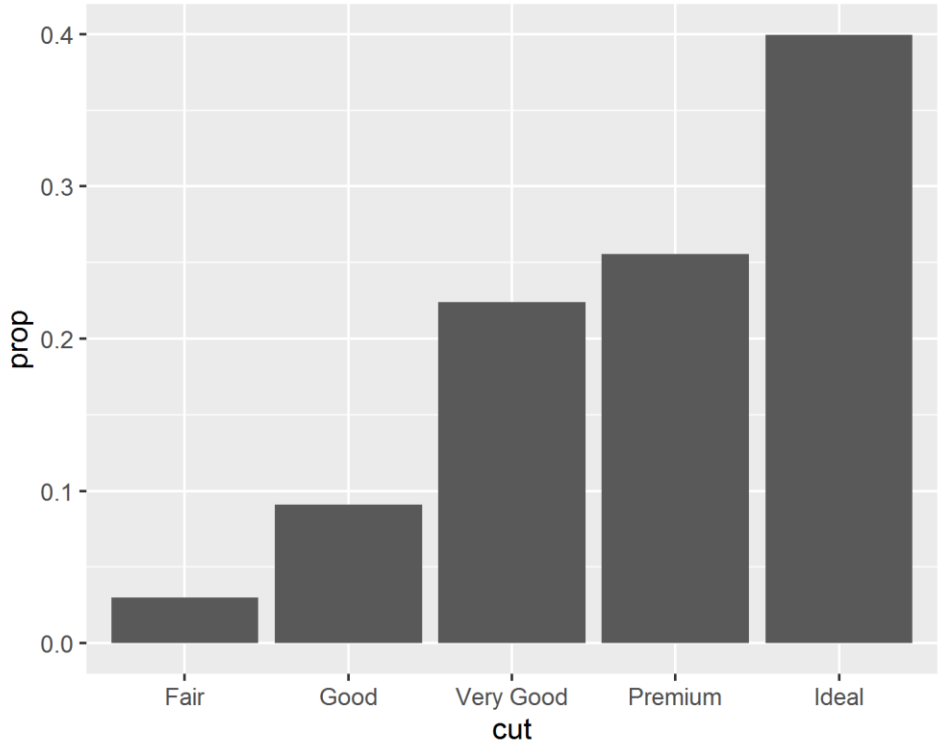
Exploração dos dados

» 1.3. Visualização: gráfico de barras

```
library("tidyverse")
?diamonds
print(diamonds)                                                    #table
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut))               #barplot
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut,y=..prop..,group=1)) #barplot %
```

Exploração dos dados

» 1.3. Visualização: gráfico de barras



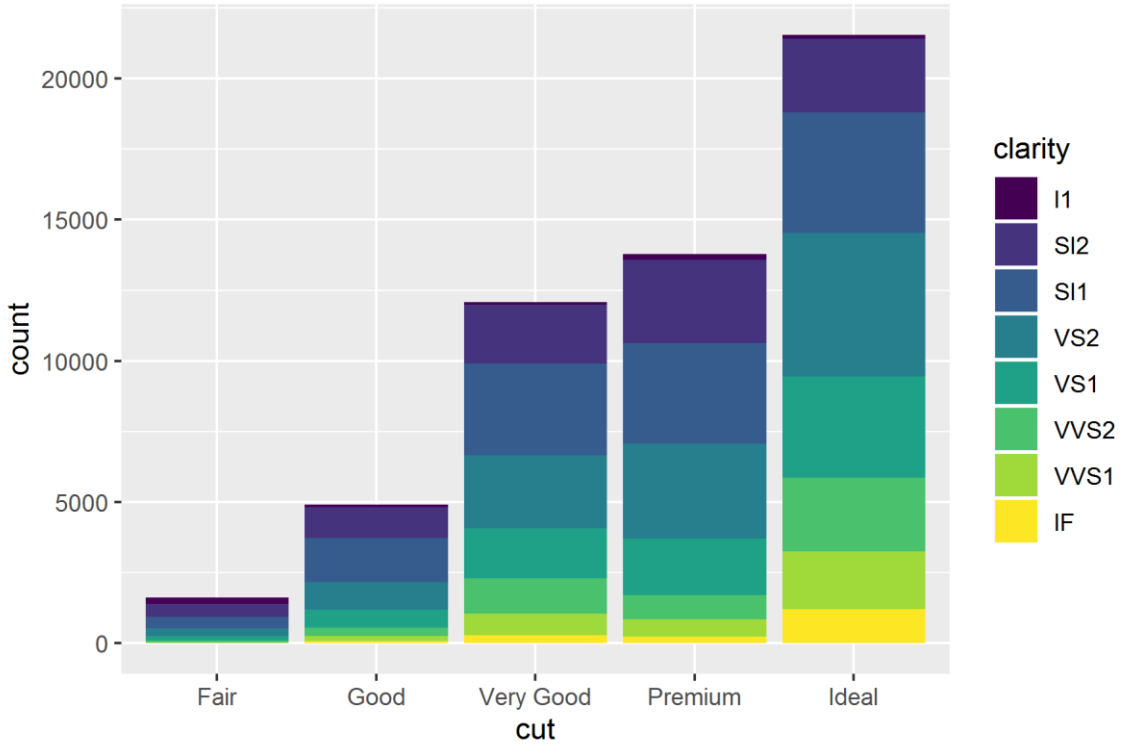
Exploração dos dados

» 1.3. Visualização: gráfico de barras

```
library("tidyverse")
?diamonds
print(diamonds)                                                    #table
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut))               #barplot
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut,y=..prop..,group=1)) #barplot %
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut,fill=clarity))  #stacked
```

Exploração dos dados

» 1.3. Visualização: gráfico de barras



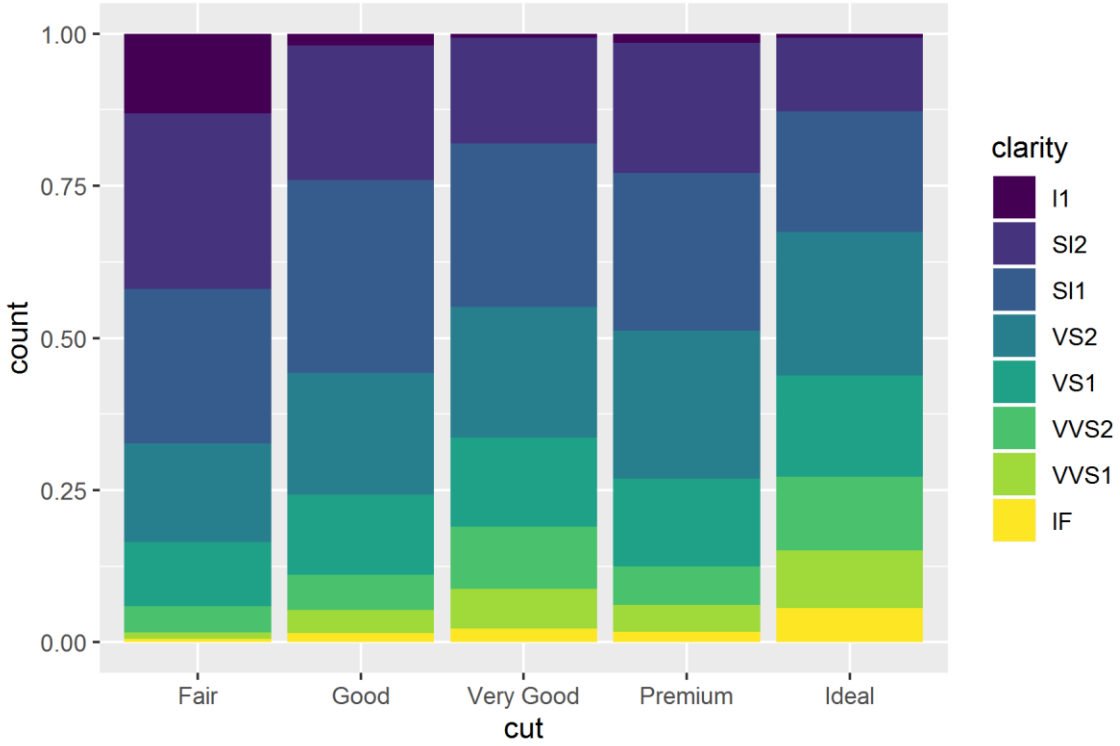
Exploração dos dados

» 1.3. Visualização: gráfico de barras

```
library("tidyverse")
?diamonds
print(diamonds)                                                    #table
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut))               #barplot
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut,y=..prop..,group=1)) #barplot %
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut,fill=clarity))  #stacked
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut,fill=clarity),
                                 position="fill")                  #stacked %
```

Exploração dos dados

» 1.3. Visualização: gráfico de barras



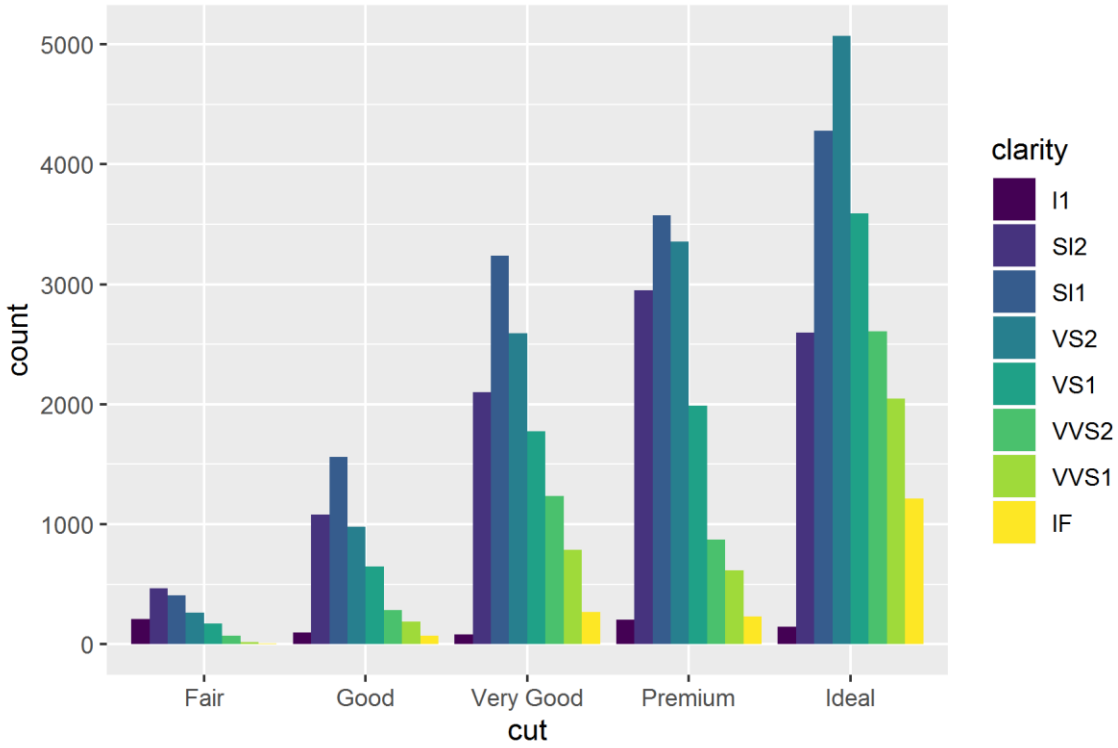
Exploração dos dados

» 1.3. Visualização: gráfico de barras

```
library("tidyverse")
?diamonds
print(diamonds)                                                    #table
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut))               #barplot
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut,y=..prop..,group=1)) #barplot %
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut,fill=clarity))  #stacked
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut,fill=clarity),
                                 position="fill")                  #stacked %
ggplot(data=diamonds) + geom_bar(mapping=aes(x=cut,fill=clarity),
                                 position="dodge")                 #clustered
```

Exploração dos dados

» 1.3. Visualização: gráfico de barras



Exploração dos dados

» 1.4. Visualização: *boxplot*

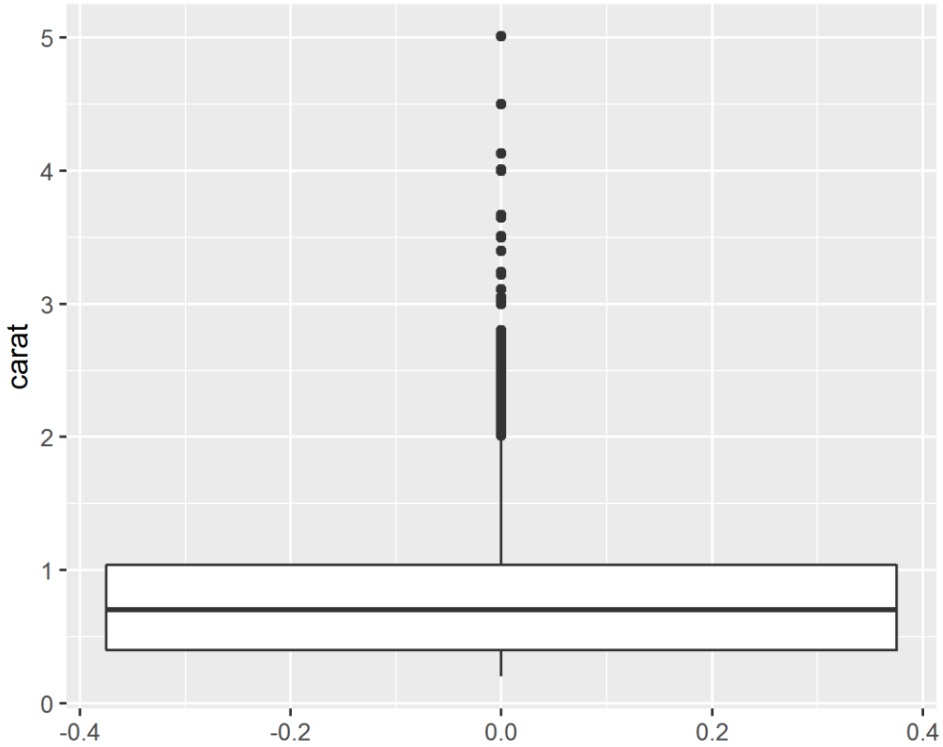
Exploração dos dados

» 1.4. Visualização: *boxplot*

```
library("tidyverse")
ggplot(data=diamonds) + geom_boxplot(mapping=aes(y=carat))        #boxplot
```

Exploração dos dados

» 1.4. Visualização: *boxplot*



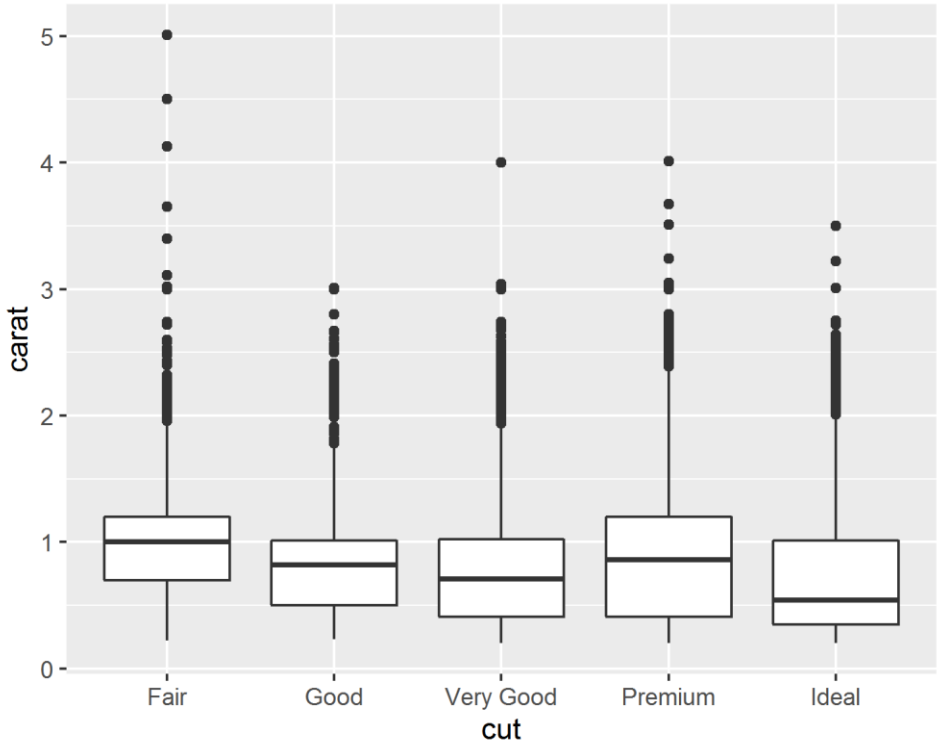
Exploração dos dados

» 1.4. Visualização: *boxplot*

```
library("tidyverse")
ggplot(data=diamonds) + geom_boxplot(mapping=aes(y=carat))         #boxplot
ggplot(data=diamonds) + geom_boxplot(mapping=aes(x=cut,y=carat)) #multiple boxplot
```

Exploração dos dados

» 1.4. Visualização: *boxplot*



Exploração dos dados

» 1.5. Visualização: histograma

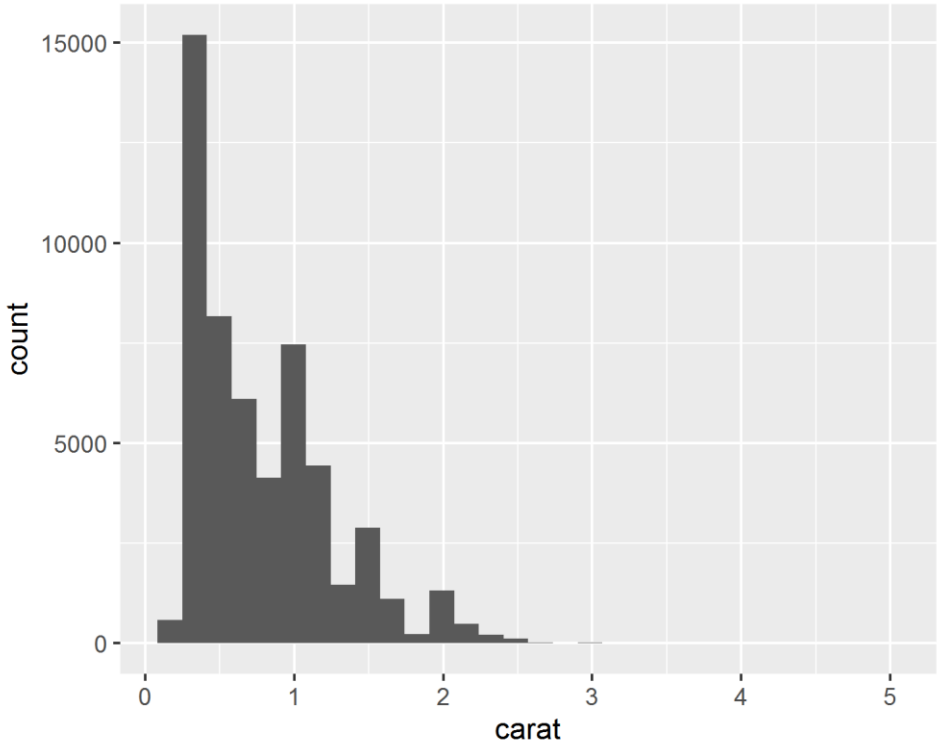
Exploração dos dados

» 1.5. Visualização: histograma

```
library("tidyverse")
ggplot(data=diamonds) + geom_histogram(mapping=aes(x=carat)) #histogram
```

Exploração dos dados

» 1.5. Visualização: histograma



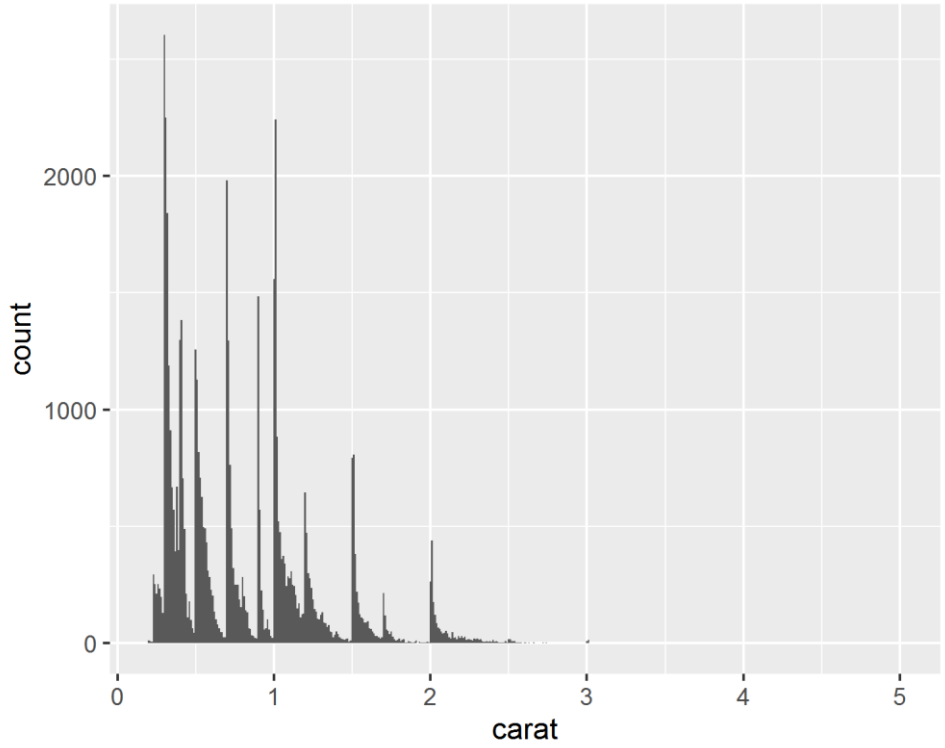
Exploração dos dados

» 1.5. Visualização: histograma

```
library("tidyverse")
ggplot(data=diamonds) + geom_histogram(mapping=aes(x=carat)) #histogram
ggplot(data=diamonds) + geom_histogram(mapping=aes(x=carat),
                                       binwidth=0.01)        #histogram
```

Exploração dos dados

» 1.5. Visualização: histograma



Exploração dos dados

» 1.6. Visualização: curvas de densidade

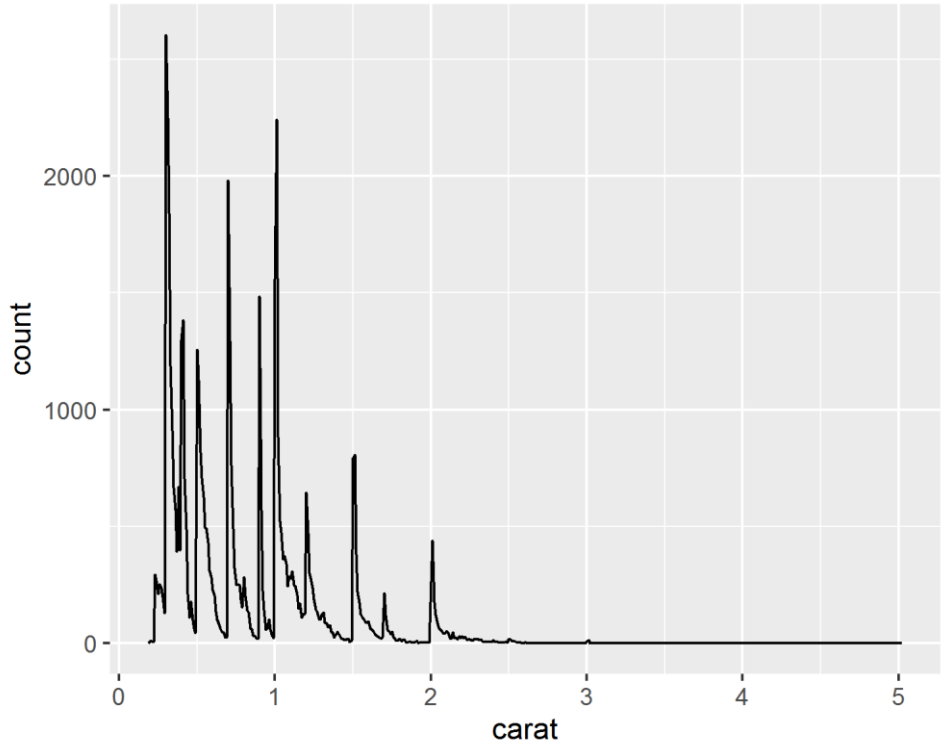
Exploração dos dados

» 1.6. Visualização: curvas de densidade

```
library("tidyverse")
ggplot(data=diamonds) + geom_histogram(mapping=aes(x=carat)) #histogram
ggplot(data=diamonds) + geom_histogram(mapping=aes(x=carat),
                                       binwidth=0.01)        #histogram
ggplot(data=diamonds) + geom_freqpoly(mapping=aes(x=carat),
                                      binwidth=0.01)         #density line
```

Exploração dos dados

» 1.6. Visualização: curvas de densidade



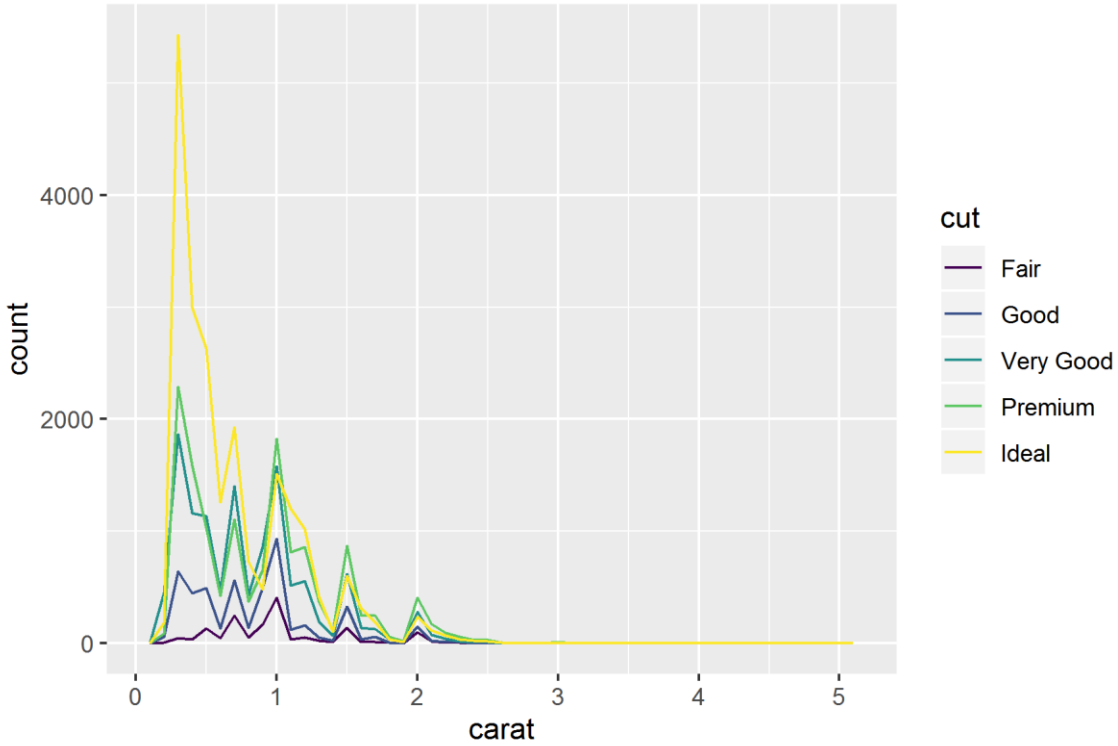
Exploração dos dados

» 1.6. Visualização: curvas de densidade

```
library("tidyverse")
ggplot(data=diamonds) + geom_histogram(mapping=aes(x=carat)) #histogram
ggplot(data=diamonds) + geom_histogram(mapping=aes(x=carat),
                                       binwidth=0.01)        #histogram
ggplot(data=diamonds) + geom_freqpoly(mapping=aes(x=carat),
                                      binwidth=0.01)         #density line
ggplot(data=diamonds) + geom_freqpoly(mapping=aes(x=carat,color=cut),
                                      binwidth=0.1)          #multiple density line
```

Exploração dos dados

» 1.6. Visualização: curvas de densidade



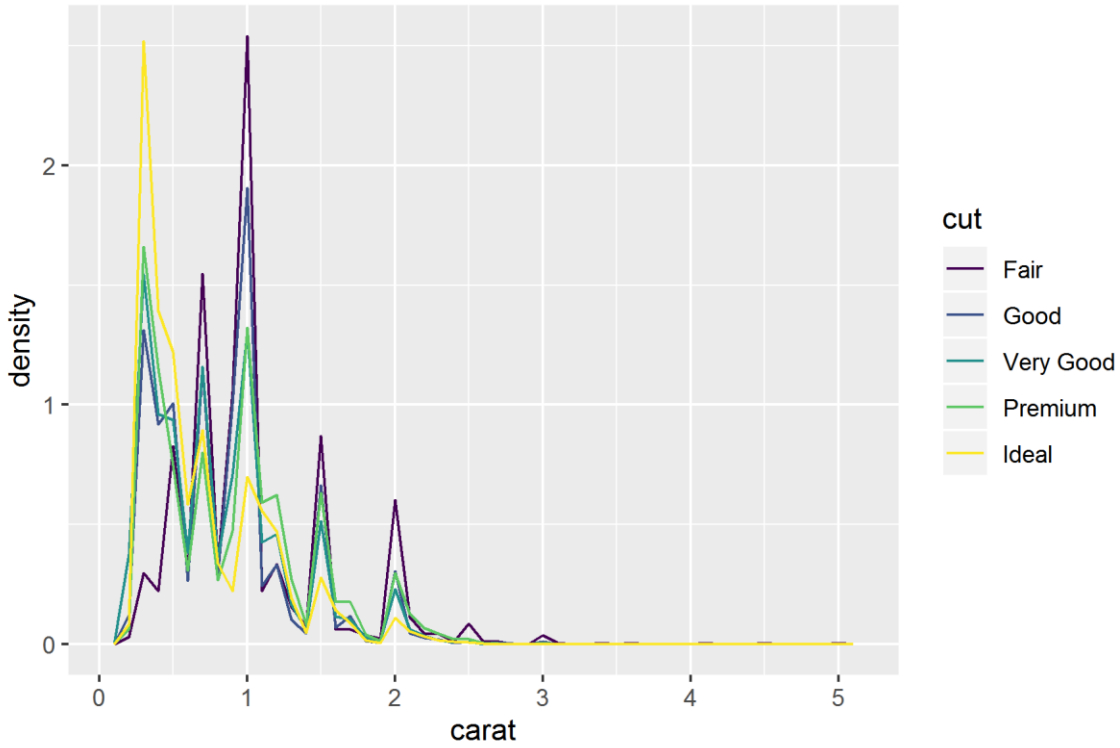
Exploração dos dados

» 1.6. Visualização: curvas de densidade

```
library("tidyverse")
ggplot(data=diamonds) + geom_histogram(mapping=aes(x=carat)) #histogram
ggplot(data=diamonds) + geom_histogram(mapping=aes(x=carat),
                                       binwidth=0.01)        #histogram
ggplot(data=diamonds) + geom_freqpoly(mapping=aes(x=carat),
                                      binwidth=0.01)         #density line
ggplot(data=diamonds) + geom_freqpoly(mapping=aes(x=carat,color=cut),
                                      binwidth=0.1)          #multiple density line
ggplot(data=diamonds) + geom_freqpoly(mapping=aes(x=carat,y=..density..,color=cut),
                                      binwidth=0.1)          #multiple density line %
```

Exploração dos dados

» 1.6. Visualização: curvas de densidade



Exploração dos dados

» 1.7. Visualização: curvas de densidade 2

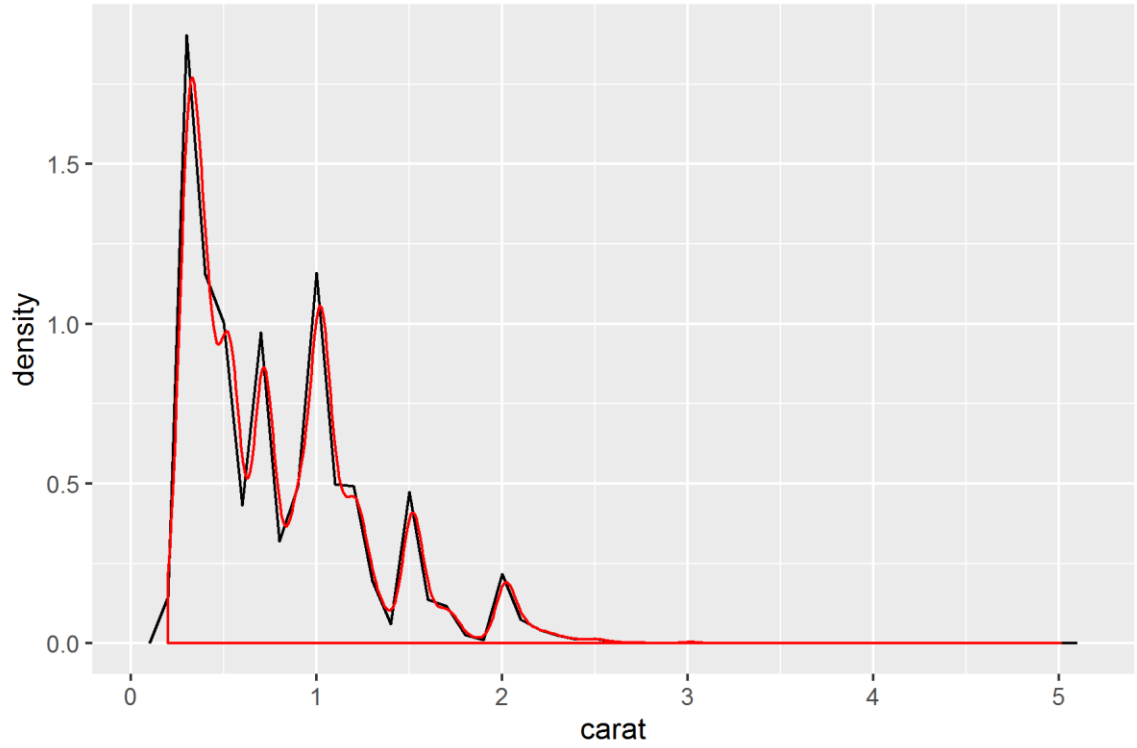
Exploração dos dados

» 1.7. Visualização: curvas de densidade 2

```
library("tidyverse")
ggplot(data=diamonds) + geom_histogram(mapping=aes(x=carat)) #histogram
ggplot(data=diamonds) + geom_histogram(mapping=aes(x=carat),
                                       binwidth=0.01)        #histogram
ggplot(data=diamonds) + geom_freqpoly(mapping=aes(x=carat),
                                      binwidth=0.01)         #density line
ggplot(data=diamonds) + geom_freqpoly(mapping=aes(x=carat,color=cut),
                                      binwidth=0.1)          #multiple density line
ggplot(data=diamonds) + geom_freqpoly(mapping=aes(x=carat,y=..density..,color=cut),
                                      binwidth=0.1)          #multiple density line %
ggplot(data=diamonds) + geom_freqpoly(mapping=aes(x=carat,y=..density..),
                                      binwidth=0.1) +
  geom_density(mapping=aes(x=carat),color="red")             #density line 2
```

Exploração dos dados

» 1.7. Visualização: curvas de densidade 2



Exploração dos dados

» 2. Manipulação dos dados (**`dplyr`**):

Exploração dos dados

» 2. Manipulação dos dados (**dplyr**):

- `filter()` Selecionar linhas (i.e. observações);
- `arrange()` Rearranjar linhas (i.e. observações);
- `select()` Selecionar colunas (i.e. variáveis);
- `mutate()` Criar novas colunas (i.e. variáveis);
- `summarize()` Calcular estatísticas descirtivas.

- `group_by()` Criar grupos de observações para manipulação.

Exploração dos dados

» 2.1. Manipulação dos dados: `filter()`

Exploração dos dados

» 2.1. Manipulação dos dados: `filter()`

```
library("tidyverse")
library("nycflights13")
?flights
print(flights)                                          #table
```

Exploração dos dados

» 2.1. Manipulação dos dados: `filter()`

```
# A tibble: 336,776 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>     <dbl>
 1  2013     1     1      517            515         2      830            819        11
 2  2013     1     1      533            529         4      850            830        20
 3  2013     1     1      542            540         2      923            850        33
 4  2013     1     1      544            545        -1     1004           1022       -18
 5  2013     1     1      554            600        -6      812            837       -25
 6  2013     1     1      554            558        -4      740            728        12
 7  2013     1     1      555            600        -5      913            854        19
 8  2013     1     1      557            600        -3      709            723       -14
 9  2013     1     1      557            600        -3      838            846        -8
10  2013     1     1      558            600        -2      753            745         8
# ... with 336,766 more rows, and 5 more variables: air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>, carrier <chr>, flight <int>, tailnum <chr>,
#   origin <chr>, dest <chr>
```

Exploração dos dados

» 2.1. Manipulação dos dados: `filter()`

```
library("tidyverse")
library("nycflights13")
?flights
print(flights)                                          #table
jan1 <- flights %>%
  filter(month==1,day==1)
print(jan1)                                             #filter 1
```

Exploração dos dados

» 2.1. Manipulação dos dados: `filter()`

```
# A tibble: 842 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>     <dbl>
 1  2013     1     1      517            515         2      830            819        11
 2  2013     1     1      533            529         4      850            830        20
 3  2013     1     1      542            540         2      923            850        33
 4  2013     1     1      544            545        -1     1004           1022       -18
 5  2013     1     1      554            600        -6      812            837       -25
 6  2013     1     1      554            558        -4      740            728        12
 7  2013     1     1      555            600        -5      913            854        19
 8  2013     1     1      557            600        -3      709            723       -14
 9  2013     1     1      557            600        -3      838            846        -8
10  2013     1     1      558            600        -2      753            745         8
# ... with 832 more rows, and 10 more variables: carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>
```

Exploração dos dados

» 2.1. Manipulação dos dados: `filter()`

```
library("tidyverse")
library("nycflights13")
?flights
print(flights)                                          #table
jan1 <- flights %>%
  filter(month==1,day==1)
print(jan1)                                             #filter 1
nov_dec <- flights %>%
  filter(month %in% c(11,12))
print(nov_dec)                                          #filter 2
```

Exploração dos dados

» 2.1. Manipulação dos dados: `filter()`

```
# A tibble: 55,403 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>     <dbl>
 1  2013    11     1        5           2359         6      352            345         7
 2  2013    11     1       35           2250       105      123           2356        87
 3  2013    11     1      455            500        -5      641            651       -10
 4  2013    11     1      539            545        -6      856            827        29
 5  2013    11     1      542            545        -3      831            855       -24
 6  2013    11     1      549            600       -11      912            923       -11
 7  2013    11     1      550            600       -10      705            659         6
 8  2013    11     1      554            600        -6      659            701        -2
 9  2013    11     1      554            600        -6      826            827        -1
10  2013    11     1      554            600        -6      749            751        -2
# ... with 55,393 more rows, and 10 more variables: carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>
```

Exploração dos dados

» 2.1. Manipulação dos dados: `filter()`

```
library("tidyverse")
library("nycflights13")
?flights
print(flights)                                          #table
jan1 <- flights %>%
  filter(month==1,day==1)
print(jan1)                                             #filter 1
nov_dec <- flights %>%
  filter(month %in% c(11,12))
print(nov_dec)                                          #filter 2
no_late <- flights %>%
  filter(arr_delay <= 120 & dep_delay <= 120)
print(no_late)                                          #filter 3
```

Exploração dos dados

» 2.1. Manipulação dos dados: `filter()`

```
# A tibble: 316,050 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>     <dbl>
 1  2013     1     1      517            515         2      830            819        11
 2  2013     1     1      533            529         4      850            830        20
 3  2013     1     1      542            540         2      923            850        33
 4  2013     1     1      544            545        -1     1004           1022       -18
 5  2013     1     1      554            600        -6      812            837       -25
 6  2013     1     1      554            558        -4      740            728        12
 7  2013     1     1      555            600        -5      913            854        19
 8  2013     1     1      557            600        -3      709            723       -14
 9  2013     1     1      557            600        -3      838            846        -8
10  2013     1     1      558            600        -2      753            745         8
# ... with 316,040 more rows, and 10 more variables: carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>
```

Exploração dos dados

» 2.2. Manipulação dos dados: `arrange()`

Exploração dos dados

» 2.2. Manipulação dos dados: `arrange()`

```
library("tidyverse")
library("nycflights13")
ord_date <- flights %>%
  arrange(year,month,day)
print(ord_date)                                    #arrange 1
```

Exploração dos dados

» 2.2. Manipulação dos dados: `arrange()`

```
# A tibble: 336,776 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>     <dbl>
 1  2013     1     1      517            515         2      830            819        11
 2  2013     1     1      533            529         4      850            830        20
 3  2013     1     1      542            540         2      923            850        33
 4  2013     1     1      544            545        -1     1004           1022       -18
 5  2013     1     1      554            600        -6      812            837       -25
 6  2013     1     1      554            558        -4      740            728        12
 7  2013     1     1      555            600        -5      913            854        19
 8  2013     1     1      557            600        -3      709            723       -14
 9  2013     1     1      557            600        -3      838            846        -8
10  2013     1     1      558            600        -2      753            745         8
# ... with 336,766 more rows, and 10 more variables: carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>
```

Exploração dos dados

» 2.2. Manipulação dos dados: `arrange()`

```
library("tidyverse")
library("nycflights13")
ord_date <- flights %>%
  arrange(year,month,day)
print(ord_date)                                    #arrange 1
ord_arr_delay <- flights %>%
  arrange(desc(arr_delay))
print(ord_arr_delay)                               #arrange 2
```

Exploração dos dados

» 2.2. Manipulação dos dados: `arrange()`

```
# A tibble: 336,776 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>     <dbl>
 1  2013     5     7     1715           1729       -14     1944           2110       -86
 2  2013     5    20      719            735       -16      951           1110       -79
 3  2013     5     2     1947           1949        -2     2209           2324       -75
 4  2013     5     6     1826           1830        -4     2045           2200       -75
 5  2013     5     4     1816           1820        -4     2017           2131       -74
 6  2013     5     2     1926           1929        -3     2157           2310       -73
 7  2013     5     6     1753           1755        -2     2004           2115       -71
 8  2013     5     7     2054           2055        -1     2317             28       -71
 9  2013     5    13      657            700        -3      908           1019       -71
10  2013     1     4     1026           1030        -4     1305           1415       -70
# ... with 336,766 more rows, and 10 more variables: carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
#   minute <dbl>, time_hour <dttm>
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
library("tidyverse")
library("nycflights13")
flights %>%
  select(year,month,day)                    #select 1
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
# A tibble: 336,776 x 3
    year month   day
   <int> <int> <int>
 1  2013     1     1
 2  2013     1     1
 3  2013     1     1
 4  2013     1     1
 5  2013     1     1
 6  2013     1     1
 7  2013     1     1
 8  2013     1     1
 9  2013     1     1
10  2013     1     1
# ... with 336,766 more rows
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
library("tidyverse")
library("nycflights13")
flights %>%
  select(year,month,day)                    #select 1
flights %>%
  select(year:day))                         #select 2
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
# A tibble: 336,776 x 3
    year month   day
   <int> <int> <int>
 1  2013     1     1
 2  2013     1     1
 3  2013     1     1
 4  2013     1     1
 5  2013     1     1
 6  2013     1     1
 7  2013     1     1
 8  2013     1     1
 9  2013     1     1
10  2013     1     1
# ... with 336,766 more rows
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
library("tidyverse")
library("nycflights13")
flights %>%
  select(year,month,day)                    #select 1
flights %>%
  select(year:day))                         #select 2
flights %>%
  select(-(year:day)))                      #select 3
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
# A tibble: 336,776 x 16
   dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay carrier flight
      <int>          <int>     <dbl>    <int>          <int>     <dbl> <chr>    <int>
 1      517            515         2      830            819        11 UA        1545
 2      533            529         4      850            830        20 UA        1714
 3      542            540         2      923            850        33 AA        1141
 4      544            545        -1     1004           1022       -18 B6         725
 5      554            600        -6      812            837       -25 DL         461
 6      554            558        -4      740            728        12 UA        1696
 7      555            600        -5      913            854        19 B6         507
 8      557            600        -3      709            723       -14 EV        5708
 9      557            600        -3      838            846        -8 B6          79
10      558            600        -2      753            745         8 AA         301
# ... with 336,766 more rows, and 8 more variables: tailnum <chr>, origin <chr>,
#   dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
#   time_hour <dttm>
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
library("tidyverse")
library("nycflights13")
flights %>%
  select(year,month,day)                    #select 1
flights %>%
  select(year:day))                         #select 2
flights %>%
  select(-(year:day)))                      #select 3
flights %>%
  select(contains("arr")))                  #select 4
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
# A tibble: 336,776 x 4
   arr_time sched_arr_time arr_delay carrier
      <int>          <int>     <dbl> <chr>
 1      830            819        11 UA
 2      850            830        20 UA
 3      923            850        33 AA
 4     1004           1022       -18 B6
 5      812            837       -25 DL
 6      740            728        12 UA
 7      913            854        19 B6
 8      709            723       -14 EV
 9      838            846        -8 B6
10      753            745         8 AA
# ... with 336,766 more rows
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
library("tidyverse")
library("nycflights13")
flights %>%
  select(year,month,day)                          #select 1
flights %>%
  select(year:day))                               #select 2
flights %>%
  select(-(year:day)))                            #select 3
flights %>%
  select(contains("arr")))                        #select 4
flights %>%
  select(time_hour,air_time,everything())) #select 5
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
# A tibble: 336,776 x 19
   time_hour           air_time  year month   day dep_time sched_dep_time dep_delay
   <dttm>                 <dbl> <int> <int> <int>    <int>          <int>     <dbl>
 1 2013-01-01 05:00:00      227  2013     1     1      517            515         2
 2 2013-01-01 05:00:00      227  2013     1     1      533            529         4
 3 2013-01-01 05:00:00      160  2013     1     1      542            540         2
 4 2013-01-01 05:00:00      183  2013     1     1      544            545        -1
 5 2013-01-01 06:00:00      116  2013     1     1      554            600        -6
 6 2013-01-01 05:00:00      150  2013     1     1      554            558        -4
 7 2013-01-01 06:00:00      158  2013     1     1      555            600        -5
 8 2013-01-01 06:00:00       53  2013     1     1      557            600        -3
 9 2013-01-01 06:00:00      140  2013     1     1      557            600        -3
10 2013-01-01 06:00:00      138  2013     1     1      558            600        -2
# ... with 336,766 more rows, and 11 more variables: arr_time <int>,
#   sched_arr_time <int>, arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
#   origin <chr>, dest <chr>, distance <dbl>, hour <dbl>, minute <dbl>
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
library("tidyverse")
library("nycflights13")
flights %>%
  select(year,month,day)                    #select 1
flights %>%
  select(year:day)                          #select 2
flights %>%
  select(-(year:day))                       #select 3
flights %>%
  select(contains("arr"))                   #select 4
flights %>%
  select(time_hour,air_time,everything())   #select 5
flights %>%
  rename(tail_num=tailnum)                  #rename 1
```

Exploração dos dados

» 2.3. Manipulação dos dados: `select()`

```
# A tibble: 336,776 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>     <dbl>
 1  2013     1     1      517            515         2      830            819        11
 2  2013     1     1      533            529         4      850            830        20
 3  2013     1     1      542            540         2      923            850        33
 4  2013     1     1      544            545        -1     1004           1022       -18
 5  2013     1     1      554            600        -6      812            837       -25
 6  2013     1     1      554            558        -4      740            728        12
 7  2013     1     1      555            600        -5      913            854        19
 8  2013     1     1      557            600        -3      709            723       -14
 9  2013     1     1      557            600        -3      838            846        -8
10  2013     1     1      558            600        -2      753            745         8
# ... with 336,766 more rows, and 10 more variables: carrier <chr>, flight <int>,
#   tail_num <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

Exploração dos dados

» 2.4. Manipulação dos dados: `mutate()`

Exploração dos dados

» 2.4. Manipulação dos dados: `mutate()`

```
library("nycflights13")
flights_sml <- flights %>%
  select(year:day,ends_with("delay"),distance,air_time)
print(flights_sml)                                          #table
```

Exploração dos dados

» 2.4. Manipulação dos dados: `mutate()`

```
# A tibble: 336,776 x 7
    year month   day dep_delay arr_delay distance air_time
   <int> <int> <int>     <dbl>     <dbl>    <dbl>    <dbl>
 1  2013     1     1         2        11     1400      227
 2  2013     1     1         4        20     1416      227
 3  2013     1     1         2        33     1089      160
 4  2013     1     1        -1       -18     1576      183
 5  2013     1     1        -6       -25      762      116
 6  2013     1     1        -4        12      719      150
 7  2013     1     1        -5        19     1065      158
 8  2013     1     1        -3       -14      229       53
 9  2013     1     1        -3        -8      944      140
10  2013     1     1        -2         8      733      138
# ... with 336,766 more rows
```

Exploração dos dados

» 2.4. Manipulação dos dados: `mutate()`

```
library("nycflights13")
flights_sml <- flights %>%
  select(year:day,ends_with("delay"),distance,air_time)
print(flights_sml)                                          #table
flights_sml %>%
  mutate(gain = arr_delay - dep_delay,
         speed = distance/air_time*60)                      #mutate 1
```

Exploração dos dados

» 2.4. Manipulação dos dados: `mutate()`

```
# A tibble: 336,776 x 9
    year month   day dep_delay arr_delay distance air_time  gain speed
   <int> <int> <int>     <dbl>     <dbl>    <dbl>    <dbl> <dbl> <dbl>
 1  2013     1     1         2        11     1400      227     9  370.
 2  2013     1     1         4        20     1416      227    16  374.
 3  2013     1     1         2        33     1089      160    31  408.
 4  2013     1     1        -1       -18     1576      183   -17  517.
 5  2013     1     1        -6       -25      762      116   -19  394.
 6  2013     1     1        -4        12      719      150    16  288.
 7  2013     1     1        -5        19     1065      158    24  404.
 8  2013     1     1        -3       -14      229       53   -11  259.
 9  2013     1     1        -3        -8      944      140    -5  405.
10  2013     1     1        -2         8      733      138    10  319.
# ... with 336,766 more rows
```

Exploração dos dados

» 2.4. Manipulação dos dados: `mutate()`

```
library("nycflights13")
flights_sml <- flights %>%
  select(year:day,ends_with("delay"),distance,air_time)
print(flights_sml)                                          #table
flights_sml %>%
  mutate(gain = arr_delay - dep_delay,
         speed = distance/air_time*60)                      #mutate 1
flights_sml %>%
  mutate(gain = arr_delay - dep_delay,
         hours = air_time/60,
         gain_per_hour = gain/hours)                        #mutate 2
```

Exploração dos dados

» 2.4. Manipulação dos dados: `mutate()`

```
# A tibble: 336,776 x 10
    year month   day dep_delay arr_delay distance air_time  gain hours gain_per_hour
   <int> <int> <int>     <dbl>     <dbl>    <dbl>    <dbl> <dbl> <dbl>         <dbl>
 1  2013     1     1         2        11     1400      227     9 3.78           2.38
 2  2013     1     1         4        20     1416      227    16 3.78           4.23
 3  2013     1     1         2        33     1089      160    31 2.67          11.6
 4  2013     1     1        -1       -18     1576      183   -17 3.05          -5.57
 5  2013     1     1        -6       -25      762      116   -19 1.93          -9.83
 6  2013     1     1        -4        12      719      150    16 2.5            6.4
 7  2013     1     1        -5        19     1065      158    24 2.63           9.11
 8  2013     1     1        -3       -14      229       53   -11 0.883        -12.5
 9  2013     1     1        -3        -8      944      140    -5 2.33          -2.14
10  2013     1     1        -2         8      733      138    10 2.3            4.35
# ... with 336,766 more rows
```

Exploração dos dados

» 2.4. Manipulação dos dados: `mutate()`

```
library("nycflights13")
flights_sml <- flights %>%
  select(year:day,ends_with("delay"),distance,air_time)
print(flights_sml)                                          #table
flights_sml %>%
  mutate(gain = arr_delay - dep_delay,
         speed = distance/air_time*60)                      #mutate 1
flights_sml %>%
  mutate(gain = arr_delay - dep_delay,
         hours = air_time/60,
         gain_per_hour = gain/hours)                        #mutate 2
flights_sml %>%
  transmute(gain = arr_delay - dep_delay,
            hours = air_time/60,
            gain_per_hour = gain/hours)                     #transmute 1
```

Exploração dos dados

» 2.4. Manipulação dos dados: `mutate()`

```
# A tibble: 336,776 x 3
    gain hours gain_per_hour
   <dbl> <dbl>         <dbl>
 1     9 3.78           2.38
 2    16 3.78           4.23
 3    31 2.67          11.6
 4   -17 3.05          -5.57
 5   -19 1.93          -9.83
 6    16 2.5            6.4
 7    24 2.63           9.11
 8   -11 0.883        -12.5
 9    -5 2.33          -2.14
10    10 2.3            4.35
# ... with 336,766 more rows
```

Exploração dos dados

» 2.4. Manipulação dos dados: `mutate()`

- Operadores aritméticos `+`, `-`, `*`, `/`, `^`;
- Moduladores aritméticos `%/%`, `%%`;
- Logs `log()`, `log2()`, `log10()`;
- Deslocador `lead()`, `lag()`;
- Acumuladores `cumsum()`, `cumprod()`, `cummin()`, `cummean()`;
- Comparadores lógicos `<`, `<=`, `>`, `>=`, `!=`;
- *Ranking* `min_rank()`, `percent_rank()`.

Exploração dos dados

» 2.5. Manipulação dos dados: `summarize()`

Exploração dos dados

» 2.5. Manipulação dos dados: `summarize()`

```
library("nycflights13")
flights %>%
  summarize(delay = mean(dep_delay,na.rm=TRUE))              #summarize 1
```

Exploração dos dados

» 2.5. Manipulação dos dados: `summarize()`

```
# A tibble: 1 x 1
  delay
  <dbl>
1  12.6
```

Exploração dos dados

» 2.5. Manipulação dos dados: `summarize()`

```
library("nycflights13")
flights %>%
  summarize(delay = mean(dep_delay,na.rm=TRUE))                    #summarize 1
delay <- flights %>%
  group_by(dest) %>%
  summarize(count = n(),
            dist = mean(distance,na.rm=TRUE),
            delay = mean(dep_delay,na.rm=TRUE)) %>%
  filter(count > 20,dest != "HNL")                                 #summarize 2
```

Exploração dos dados

» 2.5. Manipulação dos dados: `summarize()`

```
# A tibble: 96 x 4
   dest  count  dist delay
   <chr> <int> <dbl> <dbl>
 1 ABQ     254 1826  13.7
 2 ACK     265  199   6.46
 3 ALB     439  143  23.6
 4 ATL   17215  757. 12.5
 5 AUS    2439 1514. 13.0
 6 AVL     275  584.  8.19
 7 BDL     443  116  17.7
 8 BGR     375  378  19.5
 9 BHM     297  866. 29.7
10 BNA    6333  758. 16.0
# ... with 86 more rows
```

Exploração dos dados

» 2.5. Manipulação dos dados: `summarize()`

```
library("nycflights13")
flights %>%
  summarize(delay = mean(dep_delay,na.rm=TRUE))                    #summarize 1
flights %>%
  group_by(dest) %>%
  summarize(count = n(),
            dist = mean(distance,na.rm=TRUE),
            delay = mean(dep_delay,na.rm=TRUE)) %>%
  filter(count > 20,dest != "HNL")                                 #summarize 2
delay <- group_by(.data=flights,dest)
delay <- summarize(.data=delay,count = n(),
                                dist = mean(distance,na.rm=TRUE),
                                delay = mean(dep_delay,na.rm=TRUE))
delay <- filter(.data=delay,count > 20,dest != "HNL")              #summarize 3
```

Exploração dos dados

» 2.5. Manipulação dos dados: `summarize()`

```
# A tibble: 96 x 4
   dest  count  dist delay
   <chr> <int> <dbl> <dbl>
 1 ABQ     254 1826  13.7
 2 ACK     265  199   6.46
 3 ALB     439  143  23.6
 4 ATL   17215  757. 12.5
 5 AUS    2439 1514. 13.0
 6 AVL     275  584.  8.19
 7 BDL     443  116  17.7
 8 BGR     375  378  19.5
 9 BHM     297  866. 29.7
10 BNA    6333  758. 16.0
# ... with 86 more rows
```

Exploração dos dados

» 2.5. Manipulação dos dados: `summarize()`

- Medidas de tendência central **`mean()`**, **`median()`**;
- Medidas de dispersão **`sd()`**, **`IQR()`**, **`mad()`**;
- Medidas de *ranking* **`min()`**, **`quantile()`**, **`max()`**;
- Medidas de posição **`first()`**, **`nth()`**, **`last()`**;
- Medidas de contagem **`n()`** , **`sum(!is.na())`**, **`n_distinct()`**;

Exploração dos dados

» 2.6. Manipulação dos dados: `group_by()`

Exploração dos dados

» 2.6. Manipulação dos dados: `group_by()`

```
library("nycflights13")
flights %>%
  group_by(year) %>%
  summarize(nflights = n())                    #group 1
```

Exploração dos dados

» 2.6. Manipulação dos dados: `group_by()`

```
# A tibble: 1 x 2
   year nflights
  <int>    <int>
1  2013   336776
```

Exploração dos dados

» 2.6. Manipulação dos dados: `group_by()`

```
library("nycflights13")
flights %>%
  group_by(year) %>%
  summarize(nflights = n())                    #group 1
daily <- flights %>%
  group_by(year,month,day)                     #group 2
per_day <- daily %>%
  summarize(nflights = n())
```

Exploração dos dados

» 2.6. Manipulação dos dados: `group_by()`

```
# A tibble: 365 x 4
# Groups:   year, month [12]
    year month   day nflights
   <int> <int> <int>    <int>
 1  2013     1     1      842
 2  2013     1     2      943
 3  2013     1     3      914
 4  2013     1     4      915
 5  2013     1     5      720
 6  2013     1     6      832
 7  2013     1     7      933
 8  2013     1     8      899
 9  2013     1     9      902
10  2013     1    10      932
# ... with 355 more rows
```

Exploração dos dados

» 2.6. Manipulação dos dados: `group_by()`

```
library("nycflights13")
flights %>%
  group_by(year) %>%
  summarize(nflights = n())                    #group 1
daily <- flights %>%
  group_by(year,month,day)                     #group 2
per_day <- daily %>%
  summarize(nflights = n())
per_month <- per_day %>%
  summarize(nflights = sum(nflights))
```

Exploração dos dados

» 2.6. Manipulação dos dados: `group_by()`

```
# A tibble: 12 x 3
# Groups:   year [1]
    year month nflights
   <int> <int>    <int>
 1  2013     1    27004
 2  2013     2    24951
 3  2013     3    28834
 4  2013     4    28330
 5  2013     5    28796
 6  2013     6    28243
 7  2013     7    29425
 8  2013     8    29327
 9  2013     9    27574
10  2013    10    28889
11  2013    11    27268
12  2013    12    28135
```

Exploração dos dados

» 2.6. Manipulação dos dados: `group_by()`

```
library("nycflights13")
flights %>%
  group_by(year) %>%
  summarize(nflights = n())                    #group 1
daily <- flights %>%
  group_by(year,month,day)                     #group 2
per_day <- daily %>%
  summarize(nflights = n())
per_month <- per_day %>%
  summarize(nflights = sum(nflights))
per_year <- per_month %>%
  summarize(nflights = sum(nflights))
```

Exploração dos dados

» 2.6. Manipulação dos dados: `group_by()`

```
# A tibble: 1 x 2
   year nflights
  <int>    <int>
1  2013   336776
```

Exploração dos dados

» 2.6. Manipulação dos dados: `group_by()`

```
library("nycflights13")
flights %>%
  group_by(year) %>%
  summarize(nflights = n())                    #group 1
daily <- flights %>%
  group_by(year,month,day)                     #group 2
per_day <- daily %>%
  summarize(nflights = n())
per_month <- per_day %>%
  summarize(nflights = sum(nflights))
per_year <- per_month %>%
  summarize(nflights = sum(nflights))
daily %>%
  ungroup() %>%
  summarize(nflights = n())                    #ungroup 1
```

Exploração dos dados

» 2.6. Manipulação dos dados: `group_by()`

```
# A tibble: 1 x 2
   year nflights
  <int>    <int>
1  2013   336776
```

Exploração dos dados

» 3. Exploração:

Exploração dos dados

» 3. Exploração:

- *Outliers* e *Missing values*;
- Distribuição de variáveis (Tendência, Variação, Normalidade);
- Distribuição conjunta de variáveis (Visualização, Correlação).

Exploração dos dados

» 3.1. Exploração: *outliers*

Exploração dos dados

» 3.1. Exploração: *outliers*

```
library("tidyverse")
?diamonds
print(diamonds)
```

Exploração dos dados

» 3.1. Exploração: *outliers*

```
# A tibble: 53,940 x 10
   carat cut       color clarity depth table price     x     y     z
   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
 7 0.24  Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
 8 0.26  Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
 9 0.22  Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
10 0.23  Very Good H     VS1      59.4    61   338  4     4.05  2.39
# ... with 53,930 more rows
```

Exploração dos dados

» 3.1. Exploração: *outliers*

```
library("tidyverse")
?diamonds
print(diamonds)
diamonds %>%
  summary()
```

Exploração dos dados

» 3.1. Exploração: *outliers*

```
     carat               cut        color        clarity          depth
 Min.   :0.2000   Fair      : 1610   D: 6775   SI1     :13065   Min.   :43.00
 1st Qu.:0.4000   Good      : 4906   E: 9797   VS2     :12258   1st Qu.:61.00
 Median :0.7000   Very Good:12082   F: 9542   SI2     : 9194   Median :61.80
 Mean   :0.7979   Premium   :13791   G:11292   VS1     : 8171   Mean   :61.75
 3rd Qu.:1.0400   Ideal     :21551   H: 8304   VVS2    : 5066   3rd Qu.:62.50
 Max.   :5.0100                     I: 5422   VVS1    : 3655   Max.   :79.00
                                    J: 2808   (Other): 2531
     table           price             x                y                z
 Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median :57.00   Median : 2401   Median : 5.700   Median : 5.710   Median : 3.530
 Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900   Max.   :31.800
```

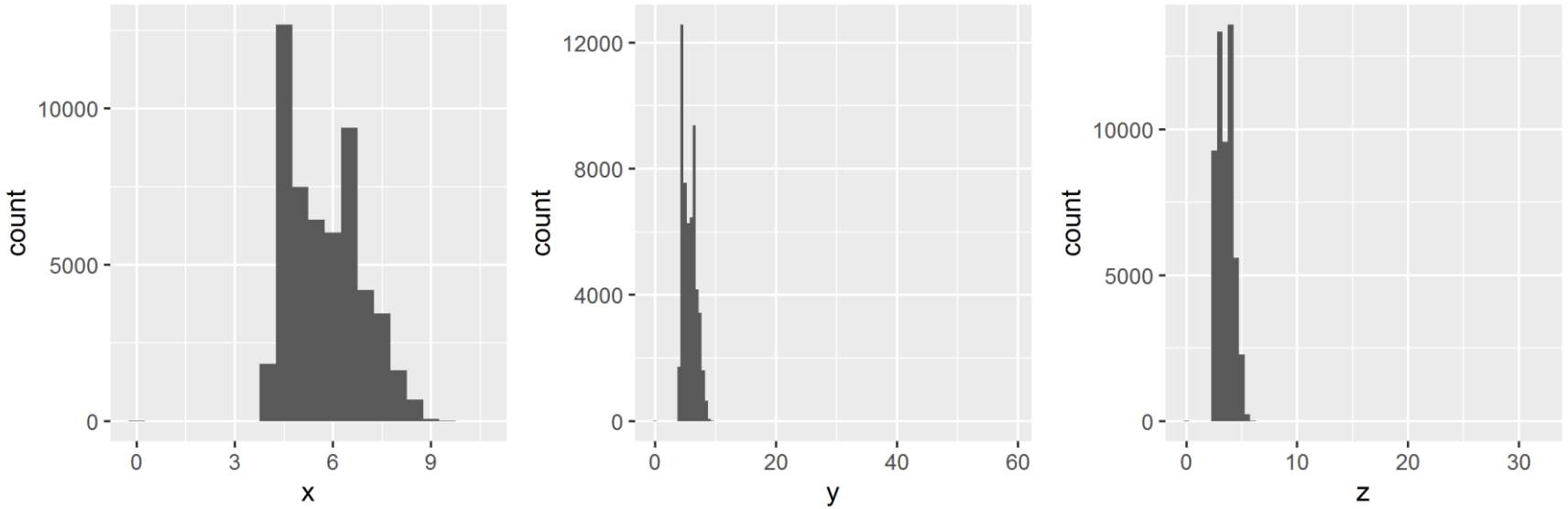
Exploração dos dados

» 3.1. Exploração: *outliers*

```
library("tidyverse")
?diamonds
print(diamonds)
diamonds %>%
  summary()
diamonds %>% ggplot() + geom_histogram(mapping=aes(x=x),binwidth=0.5)
diamonds %>% ggplot() + geom_histogram(mapping=aes(x=y),binwidth=0.5)
diamonds %>% ggplot() + geom_histogram(mapping=aes(x=z),binwidth=0.5)
```

Exploração dos dados

» 3.1. Exploração: *outliers*



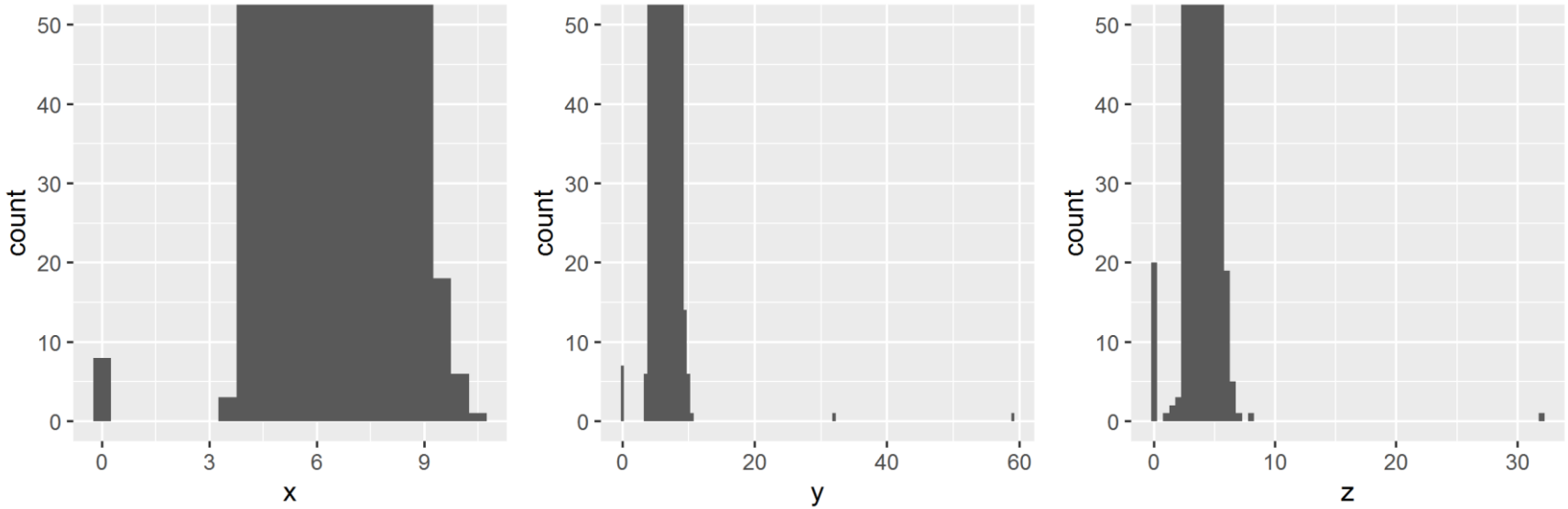
Exploração dos dados

» 3.1. Exploração: *outliers*

```
library("tidyverse")
?diamonds
print(diamonds)
diamonds %>%
  summary()
diamonds %>% ggplot() + geom_histogram(mapping=aes(x=x),binwidth=0.5)
diamonds %>% ggplot() + geom_histogram(mapping=aes(x=y),binwidth=0.5)
diamonds %>% ggplot() + geom_histogram(mapping=aes(x=z),binwidth=0.5)
diamonds %>% ggplot() + geom_histogram(mapping=aes(x=x),binwidth=0.5) +
                        coord_cartesian(ylim=c(0,50))
diamonds %>% ggplot() + geom_histogram(mapping=aes(x=y),binwidth=0.5) +
                        coord_cartesian(ylim=c(0,50))
diamonds %>% ggplot() + geom_histogram(mapping=aes(x=z),binwidth=0.5) +
                        coord_cartesian(ylim=c(0,50))
```

Exploração dos dados

» 3.1. Exploração: *outliers*



Exploração dos dados

» 3.1. Exploração: *outliers*

```
unusual <- diamonds %>%
  filter((x < 1) | (y < 1 | y > 20) | (z < 1 | z > 10)) %>%
  arrange(x)
print(unusual,n=23)
```

Exploração dos dados

» 3.1. Exploração: *outliers*

```
# A tibble: 23 x 10
   carat cut       color clarity depth table price     x     y     z
   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
 1  1.07 Ideal     F     SI2      61.6  56    4954  0     6.62  0
 2  1    Very Good H     VS2      63.3  53    5139  0     0     0
 3  1.14 Fair      G     VS1      57.5  67    6381  0     0     0
 4  1.56 Ideal     G     VS2      62.2  54   12800  0     0     0
 5  1.2  Premium   D     VVS1     62.1  59   15686  0     0     0
 6  2.25 Premium   H     SI2      62.8  59   18034  0     0     0
 7  0.71 Good      F     SI2      64.1  60    2130  0     0     0
 8  0.71 Good      F     SI2      64.1  60    2130  0     0     0
 9  0.51 Very Good E     VS1      61.8  54.7  1970  5.12  5.15 31.8
10  0.51 Ideal     E     VS1      61.8  55    2075  5.15 31.8   5.12
11  1.1  Premium   G     SI2      63    59    3696  6.5   6.47  0

[...]
```

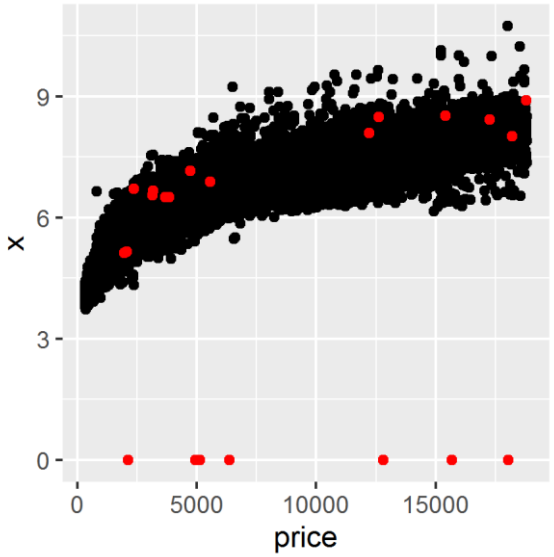
Exploração dos dados

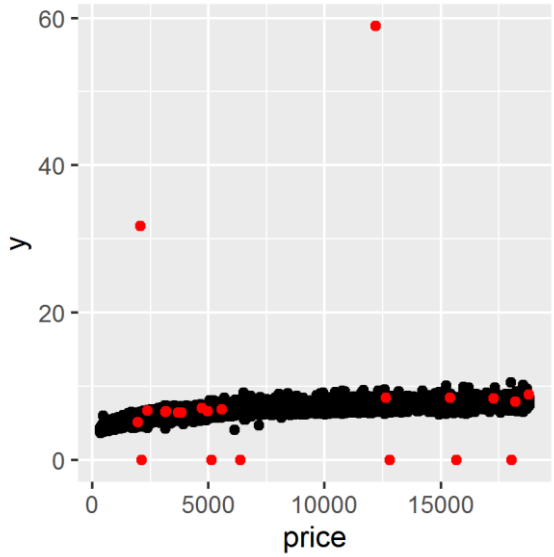
» 3.1. Exploração: *outliers*

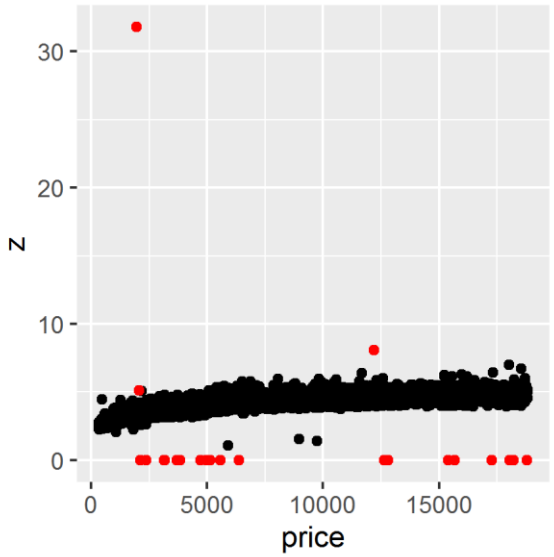
```
unusual <- diamonds %>%
  filter((x < 1) | (y < 1 | y > 20) | (z < 1 | z > 10)) %>%
  arrange(x)
print(unusual,n=23)
diamonds %>% ggplot() + geom_point(mapping=aes(x=price,y=x)) +
                        geom_point(data=unusual,mapping=aes(x=price,y=x),color="red")
diamonds %>% ggplot() + geom_point(mapping=aes(x=price,y=y)) +
                        geom_point(data=unusual,mapping=aes(x=price,y=y),color="red")
diamonds %>% ggplot() + geom_point(mapping=aes(x=price,y=z)) +
                        geom_point(data=unusual,mapping=aes(x=price,y=z),color="red")
```

Exploração dos dados

» 3.1. Exploração: *outliers*







Exploração dos dados

» 3.1. Exploração: *outliers*

```
unusual <- diamonds %>%
  filter((x < 1) | (y < 1 | y > 20) | (z < 1 | z > 10)) %>%
  arrange(x)
print(unusual,n=23)
diamonds %>% ggplot() + geom_point(mapping=aes(x=price,y=x)) +
                        geom_point(data=unusual,mapping=aes(x=price,y=x),color="red")
diamonds %>% ggplot() + geom_point(mapping=aes(x=price,y=y)) +
                        geom_point(data=unusual,mapping=aes(x=price,y=y),color="red")
diamonds %>% ggplot() + geom_point(mapping=aes(x=price,y=z)) +
                        geom_point(data=unusual,mapping=aes(x=price,y=z),color="red")
diamonds %>%
  filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10)) %>%
  summary()
```

Exploração dos dados

» 3.1. Exploração: *outliers*

```
     carat               cut        color        clarity          depth      
 Min.   :0.2000   Fair      : 1609   D: 6774   SI1     :13063   Min.   :43.00  
 1st Qu.:0.4000   Good      : 4902   E: 9795   VS2     :12254   1st Qu.:61.00  
 Median :0.7000   Very Good:12080   F: 9538   SI2     : 9184   Median :61.80  
 Mean   :0.7977   Premium   :13779   G:11284   VS1     : 8168   Mean   :61.75  
 3rd Qu.:1.0400   Ideal     :21547   H: 8297   VVS2    : 5066   3rd Qu.:62.50  
 Max.   :5.0100                     I: 5421   VVS1    : 3654   Max.   :79.00  
                                    J: 2808   (Other): 2528                  
     table           price             x                y                z       
 Min.   :43.00   Min.   :  326   Min.   : 3.730   Min.   : 3.680   Min.   :1.070  
 1st Qu.:56.00   1st Qu.:  949   1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.:2.910  
 Median :57.00   Median : 2401   Median : 5.700   Median : 5.710   Median :3.530  
 Mean   :57.46   Mean   : 3931   Mean   : 5.732   Mean   : 5.733   Mean   :3.539  
 3rd Qu.:59.00   3rd Qu.: 5323   3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.:4.040  
 Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :10.540   Max.   :6.980  
```

Exploração dos dados

» 3.2. Exploração: *missing values*

Exploração dos dados

» 3.2. Exploração: *missing values*

```
library("tidyverse")
diamonds_NA <- diamonds %>%
  mutate(x = ifelse(x < 1,NA,x)) %>%
  mutate(y = ifelse(y < 1 | y > 20,NA,y)) %>%
  mutate(z = ifelse(z < 1 | z > 10,NA,z))
print(diamonds_NA)
```

Exploração dos dados

» 3.2. Exploração: *missing values*

```
  # A tibble: 53,940 x 10
   carat cut       color clarity depth table price     x     y     z
   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
 7 0.24  Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
 8 0.26  Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
 9 0.22  Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
10 0.23  Very Good H     VS1      59.4    61   338  4     4.05  2.39
# ... with 53,930 more rows
```

Exploração dos dados

» 3.2. Exploração: *missing values*

```
diamonds_NA %>%
  group_by(cut) %>%
  summarize(mean_x = mean(x), mean_y = mean(y), mean_z = mean(z), n = n(),
            x_NA = sum(is.na(x)), y_NA = sum(is.na(y)), z_NA = sum(is.na(z)))
```

Exploração dos dados

» 3.2. Exploração: *missing values*

```
# A tibble: 5 x 8
  cut       mean_x mean_y mean_z     n  x_NA  y_NA  z_NA
  <ord>      <dbl>  <dbl>  <dbl> <int> <int> <int> <int>
1 Fair          NA     NA     NA  1610     1     1     1
2 Good          NA     NA     NA  4906     2     2     4
3 Very Good     NA     NA     NA 12082     1     1     2
4 Premium       NA     NA     NA 13791     2     3    11
5 Ideal         NA     NA     NA 21551     2     2     3
```

Exploração dos dados

» 3.2. Exploração: *missing values*

```
diamonds_NA %>%
  group_by(cut) %>%
  summarize(mean_x = mean(x), mean_y = mean(y), mean_z = mean(z), n = n(),
            x_NA = sum(is.na(x)), y_NA = sum(is.na(y)), z_NA = sum(is.na(z)))
diamonds_NA %>%
  group_by(cut) %>%
  mutate(xyz = x + y + z) %>%
  summarize(mean_x = mean(x,na.rm=TRUE), mean_y = mean(y,na.rm=TRUE),
            mean_z = mean(z,na.rm=TRUE), n=n(), x_NA = sum(is.na(x)),
            y_NA = sum(is.na(y)), z_NA = sum(is.na(z)), nNA = sum(is.na(xyz)))
```

Exploração dos dados

» 3.2. Exploração: *missing values*

```
# A tibble: 5 x 9
  cut       mean_x mean_y mean_z     n  x_NA  y_NA  z_NA   nNA
  <ord>      <dbl>  <dbl>  <dbl> <int> <int> <int> <int> <int>
1 Fair        6.25   6.19   3.99  1610     1     1     1     1
2 Good        5.84   5.85   3.64  4906     2     2     4     4
3 Very Good   5.74   5.77   3.56 12082     1     1     2     2
4 Premium     5.97   5.94   3.65 13791     2     3    11    12
5 Ideal       5.51   5.52   3.40 21551     2     2     3     4
```

Exploração dos dados

» 3.2. Exploração: *missing values*

```
diamonds_NA %>%
  group_by(cut) %>%
  summarize(mean_x = mean(x), mean_y = mean(y), mean_z = mean(z), n = n(),
            x_NA = sum(is.na(x)), y_NA = sum(is.na(y)), z_NA = sum(is.na(z)))
diamonds_NA %>%
  group_by(cut) %>%
  mutate(xyz = x + y + z) %>%
  summarize(mean_x = mean(x,na.rm=TRUE), mean_y = mean(y,na.rm=TRUE),
            mean_z = mean(z,na.rm=TRUE), n=n(), x_NA = sum(is.na(x)),
            y_NA = sum(is.na(y)), z_NA = sum(is.na(z)), nNA = sum(is.na(xyz)))
diamonds_NA %>%
  ggplot() + geom_point(mapping=aes(x=price,y=x))                #warning
diamonds_NA %>%
  ggplot() + geom_point(mapping=aes(x=price,y=x),na.rm=TRUE) #no warn
```

Exploração dos dados

» 3.3. Exploração: distribuição 1D

Exploração dos dados

» 3.3. Exploração: distribuição 1D

```
library("tidyverse")
library("moments")
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
```

Exploração dos dados

» 3.3. Exploração: distribuição 1D

```
# A tibble: 53,917 x 10
   carat cut       color clarity depth table price     x     y     z
   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
 7 0.24  Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
 8 0.26  Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
 9 0.22  Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
10 0.23  Very Good H     VS1      59.4    61   338  4     4.05  2.39
# ... with 53,907 more rows
```

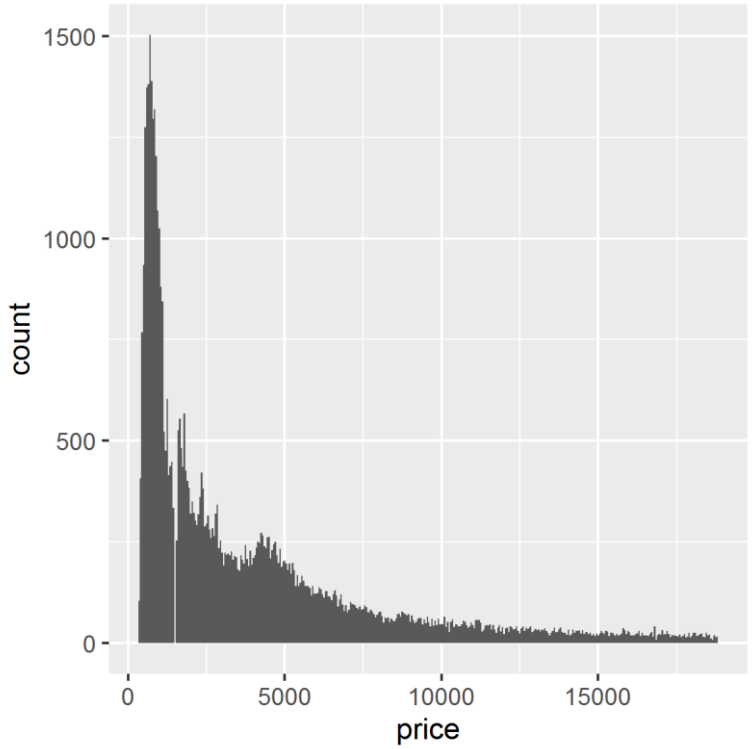
Exploração dos dados

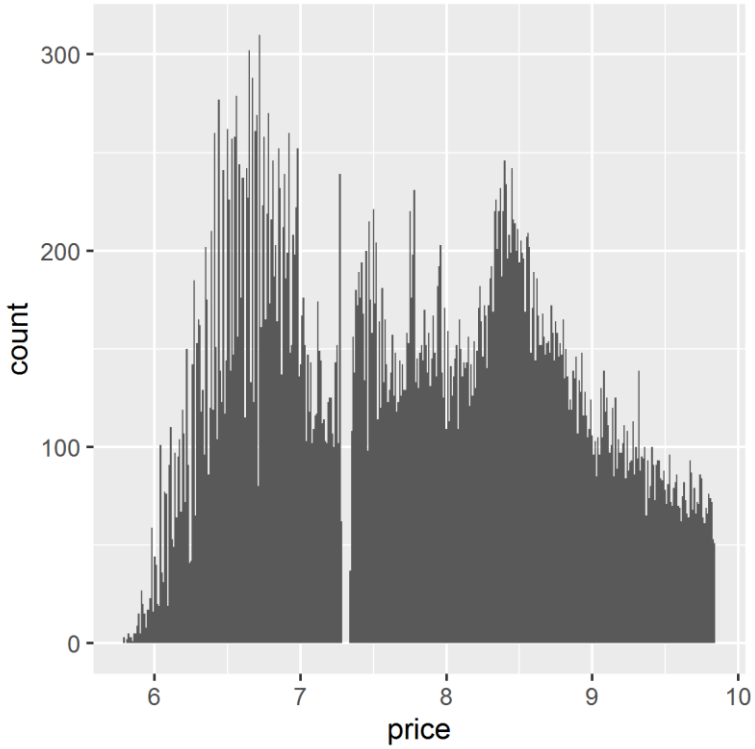
» 3.3. Exploração: distribuição 1D

```
library("tidyverse")
library("moments")
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
diamonds2 %>% ggplot() + geom_histogram(mapping=aes(x=price),binwidth=50)
ldiamonds2 <- diamonds2 %>%
  mutate(price = log(price))
ldiamonds2 %>% ggplot() + geom_histogram(mapping=aes(x=price),binwidth=0.01)
```

Exploração dos dados

» 3.3. Exploração: distribuição 1D





Exploração dos dados

» 3.3. Exploração: distribuição 1D

```
library("tidyverse")
library("moments")
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
diamonds2 %>% ggplot() + geom_histogram(mapping=aes(x=price),binwidth=50)
ldiamonds2 <- diamonds2 %>%
  mutate(price = log(price))
ldiamonds2 %>% ggplot() + geom_histogram(mapping=aes(x=price),binwidth=0.01)
param <- diamonds2 %>% summarize(n = n(), mean = mean(price), sd = sd(price),
                                 median = median(price), IQR = IQR(price))
lparam <- ldiamonds2 %>% summarize(n = n(), mean = mean(price), sd = sd(price),
                                   median = median(price), IQR = IQR(price))
print(round(cbind(t(param),t(lparam)),digits=2))
```

Exploração dos dados

» 3.3. Exploração: distribuição 1D

```
        original    log()
n       53917.00 53917.00
mean     3930.91     7.79
sd       3987.22     1.01
median   2401.00     7.78
IQR      4374.00     1.72
```

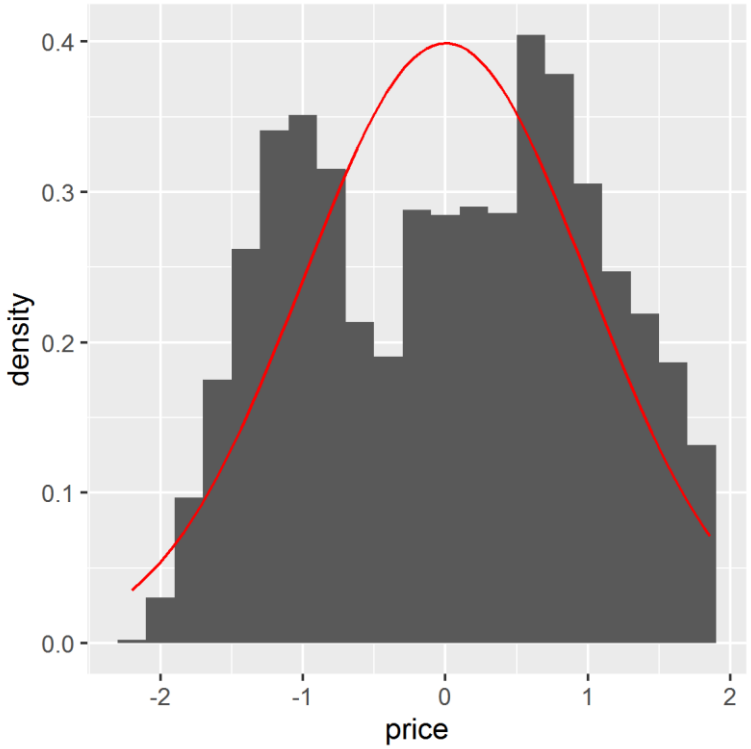
Exploração dos dados

» 3.3. Exploração: distribuição 1D

```
ldiamonds2_scale <- ldiamonds2 %>%
  mutate(price = (price - mean(price))/sd(price))
ldiamonds2_scale %>% ggplot() +
  geom_histogram(mapping=aes(x=price,y=..density..),binwidth=0.2) +
  stat_function(fun=dnorm,color="red",args=list(mean=0,sd=1))
```

Exploração dos dados

» 3.3. Exploração: distribuição 1D



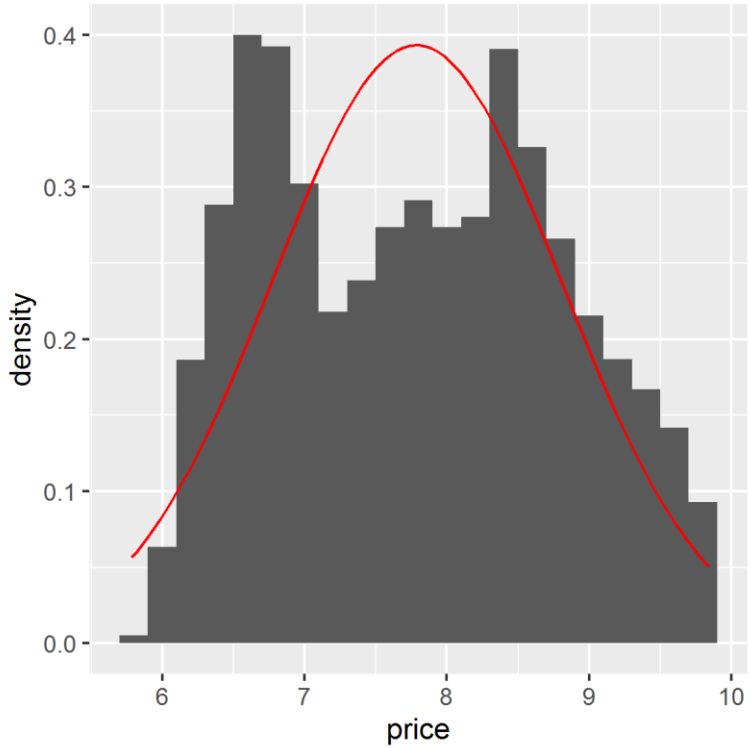
Exploração dos dados

» 3.3. Exploração: distribuição 1D

```
ldiamonds2_scale <- ldiamonds2 %>%
  mutate(price = (price - mean(price))/sd(price))
ldiamonds2_scale %>% ggplot() +
  geom_histogram(mapping=aes(x=price,y=..density..),binwidth=0.2) +
  stat_function(fun=dnorm,color="red",args=list(mean=0,sd=1))
price_mean <- mean(ldiamonds2$price)
price_sd <- sd(ldiamonds2$price)
ldiamonds2 %>% ggplot() +
  geom_histogram(mapping=aes(x=price,y=..density..),binwidth=0.2) +
  stat_function(fun=dnorm,color="red",args=list(mean=price_mean,sd=price_sd))
```

Exploração dos dados

» 3.3. Exploração: distribuição 1D



Exploração dos dados

» 3.3. Exploração: distribuição 1D

```
ldiamonds2_scale <- ldiamonds2 %>%
  mutate(price = (price - mean(price))/sd(price))
ldiamonds2_scale %>% ggplot() +
  geom_histogram(mapping=aes(x=price,y=..density..),binwidth=0.2) +
  stat_function(fun=dnorm,color="red",args=list(mean=0,sd=1))
price_mean <- mean(ldiamonds2$price)
price_sd <- sd(ldiamonds2$price)
ldiamonds2 %>% ggplot() +
  geom_histogram(mapping=aes(x=price,y=..density..),binwidth=0.2) +
  stat_function(fun=dnorm,color="red",args=list(mean=price_mean,sd=price_sd))
param <- ldiamonds2_scale %>% summarize(n = n(), mean = mean(price), sd = sd(price),
  skewness = skewness(price), kurtosis = kurtosis(price))
tab1 <- round(cbind(t(param),c(1/0,0,1,0,3)),digits=3)
colnames(tab1) <- c("original","Gaussian")
print(tab1)
```

Exploração dos dados

» 3.3. Exploração: distribuição 1D

```
           original Gaussian
n         53917.000      Inf
mean          0.000        0
sd            1.000        1
skewness     -0.055        0
kurtosis      1.866        3
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)

Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)

```
library("tidyverse")
?diamonds
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)

```
# A tibble: 53,917 x 10
   carat cut       color clarity depth table price     x     y     z
   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
 7 0.24  Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
 8 0.26  Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
 9 0.22  Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
10 0.23  Very Good H     VS1      59.4    61   338  4     4.05  2.39
# ... with 53,907 more rows
```

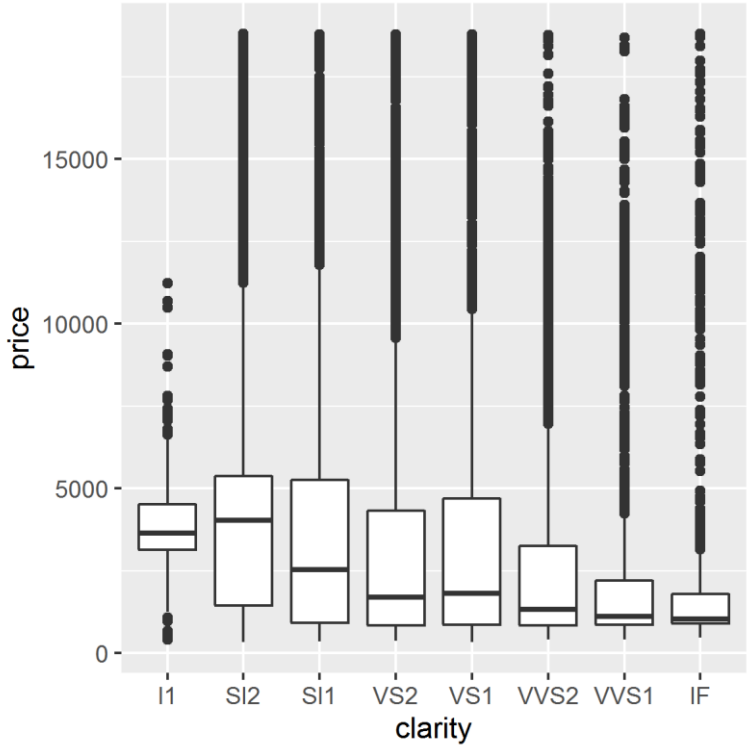
Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)

```
library("tidyverse")
?diamonds
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
diamonds2 %>%
  filter(cut=="Ideal") %>%
  ggplot() + geom_boxplot(mapping=aes(x=clarity,y=price))
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)



Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)

```
library("tidyverse")
?diamonds
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
diamonds2 %>%
  filter(cut=="Ideal") %>%
  ggplot() + geom_boxplot(mapping=aes(x=clarity,y=price))
diamonds2 %>%
  filter(cut=="Ideal") %>%
  group_by(clarity) %>%
  summarize(mean_price = mean(price))
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)

```
# A tibble: 8 x 2
  clarity mean.clarity
  <ord>          <dbl>
1 I1             4018.
2 SI2            4695.
3 SI1            3740.
4 VS2            3282.
5 VS1            3487.
6 VVS2           3250.
7 VVS1           2468.
8 IF             2273.
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)

```
diamonds2 %>%
  filter(cut=="Ideal") %>%
  group_by(clarity) %>%
  mutate(mean_price = mean(price))
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)

```
# A tibble: 21,547 x 11
# Groups:   clarity [8]
   carat cut   color clarity depth table price     x     y     z mean_price
   <dbl> <ord> <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>      <dbl>
 1  0.23 Ideal E     SI2      61.5    55   326  3.95  3.98  2.43      4756.
 2  0.23 Ideal J     VS1      62.8    56   340  3.93  3.9   2.46      3490.
 3  0.31 Ideal J     SI2      62.2    54   344  4.35  4.37  2.71      4756.
 4  0.3  Ideal I     SI2      62      54   348  4.31  4.34  2.68      4756.
 5  0.33 Ideal I     SI2      61.8    55   403  4.49  4.51  2.78      4756.
 6  0.33 Ideal I     SI2      61.2    56   403  4.49  4.5   2.75      4756.
 7  0.33 Ideal J     SI1      61.1    56   403  4.49  4.55  2.76      3752.
 8  0.23 Ideal G     VS1      61.9    54   404  3.93  3.95  2.44      3490.
 9  0.32 Ideal I     SI1      60.9    55   404  4.45  4.48  2.72      3752.
10  0.3  Ideal I     SI2      61      59   405  4.3   4.33  2.63      4756.
# ... with 21,537 more rows
```

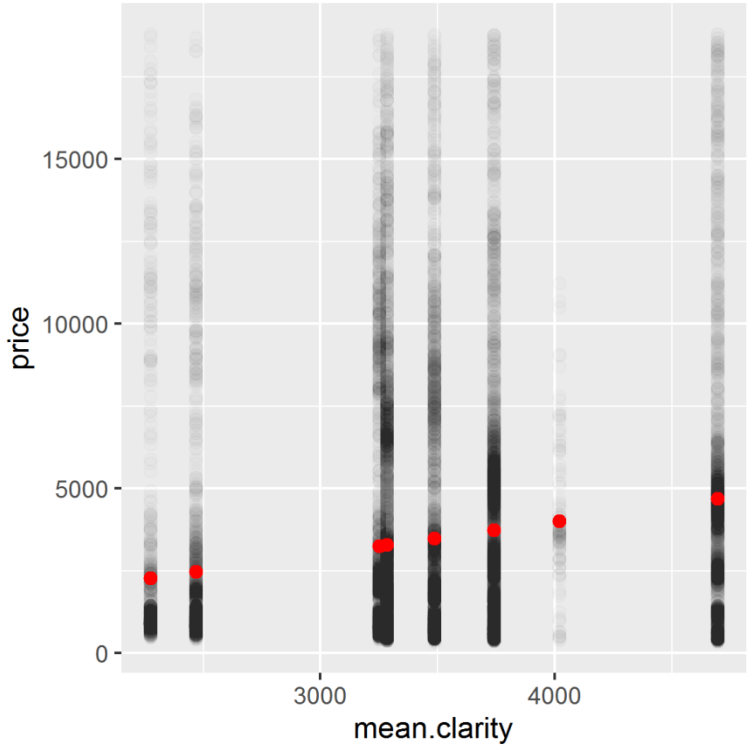
Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)

```
diamonds2 %>%
  filter(cut=="Ideal") %>%
  group_by(clarity) %>%
  mutate(mean_price = mean(price)) %>%
  ggplot() +
    geom_point(mapping=aes(x=mean_price,y=price),size=2,alpha=0.01) +
    geom_point(mapping=aes(x=mean_price,y=mean_price),size=2,color="red")
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)



Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)

```
diamonds2 %>%
  filter(cut=="Ideal") %>%
  group_by(clarity) %>%
  mutate(mean_price = mean(price)) %>%
  ggplot() +
    geom_point(mapping=aes(x=mean_price,y=price),size=2,alpha=0.01) +
    geom_point(mapping=aes(x=mean_price,y=mean_price),size=2,color="red")
diamonds2 %>%
  filter(cut=="Ideal") %>%
  group_by(clarity) %>%
  mutate(mean_price = mean(price)) %>%
  ungroup() %>%
  summarize(res = cor(mean_price,price))    #correlation observed vs. predicted values
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (cat vs. cont)

```
# A tibble: 1 x 1
    res
  <dbl>
1 0.164
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cat)

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cat)

```
library("tidyverse")
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cat)

```
# A tibble: 53,917 x 10
   carat cut       color clarity depth table price     x     y     z
   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
 7 0.24  Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
 8 0.26  Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
 9 0.22  Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
10 0.23  Very Good H     VS1      59.4    61   338  4     4.05  2.39
# ... with 53,907 more rows
```

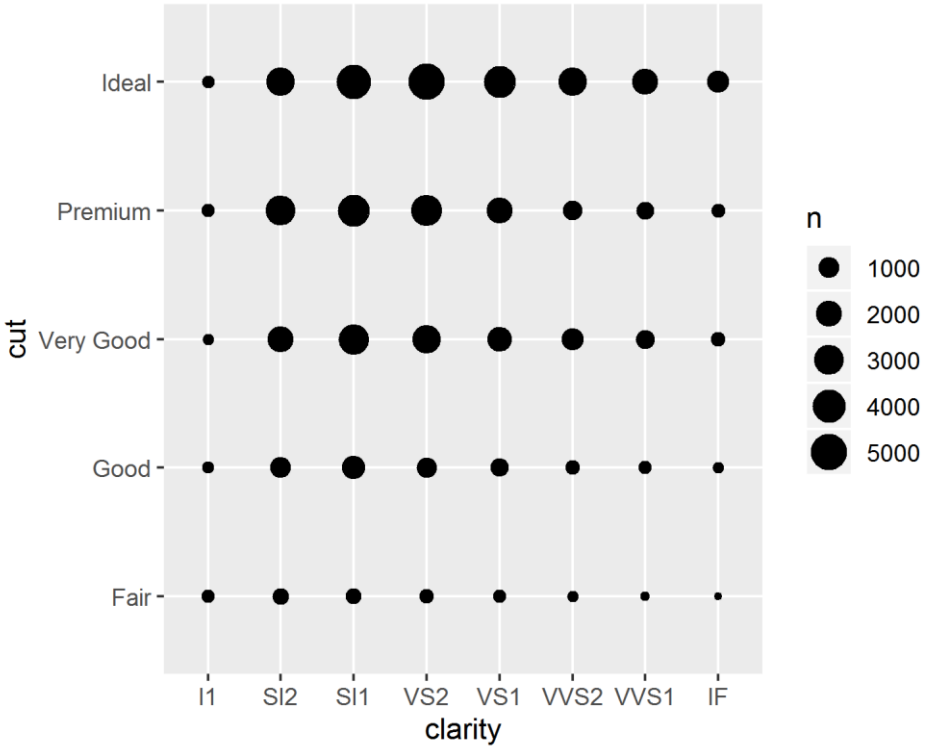
Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cat)

```
library("tidyverse")
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
diamonds2 %>% ggplot() +
  geom_count(mapping=aes(x=clarity,y=cut))
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cat)



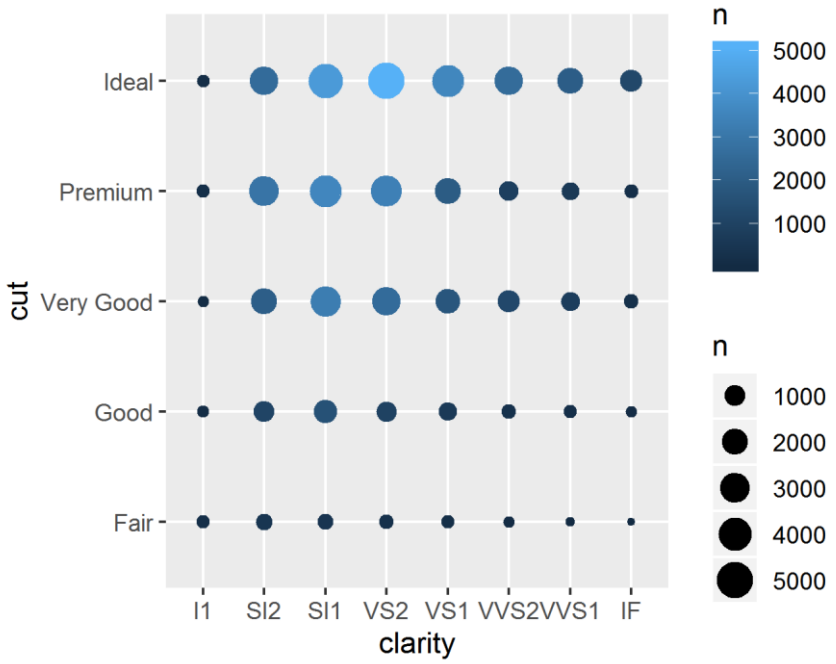
Exploração dos dados

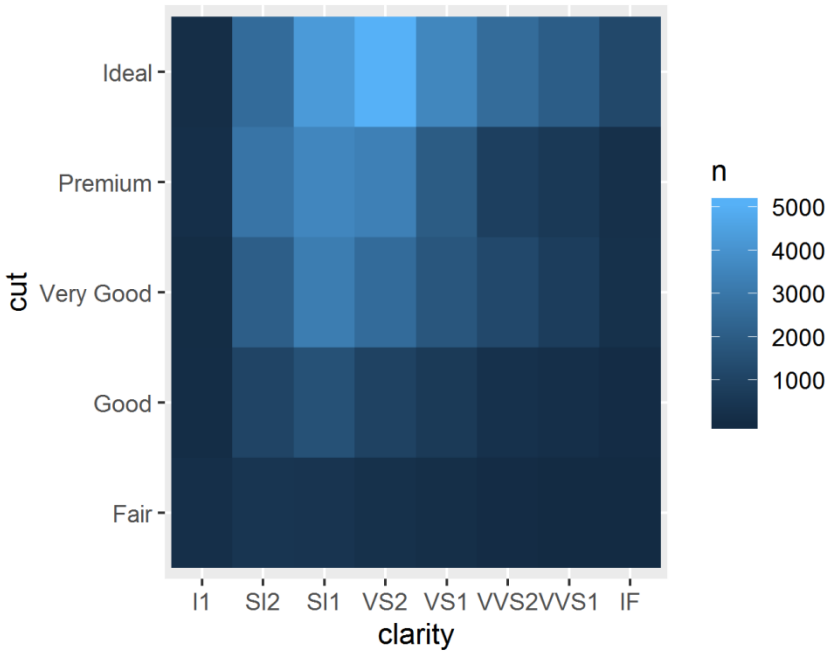
» 3.4. Exploração: distribuição 2D (2 cat)

```
library("tidyverse")
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
diamonds2 %>% ggplot() +
  geom_count(mapping=aes(x=clarity,y=cut))
diamonds2 %>%
  group_by(clarity,cut) %>%
  summarize(n = n()) %>%
  ggplot() + geom_count(mapping=aes(x=clarity,y=cut,color=n,size=n))
diamonds2 %>%
  group_by(clarity,cut) %>%
  summarize(n = n()) %>%
  ggplot() + geom_tile(mapping=aes(x=clarity,y=cut,fill=n))
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cat)





Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cat)

```
#Calculate Cramer's V from contingency table
calc_Cramers_V <- function(x){
    chi_stat <- chisq.test(x)$statistic #chi-sqrt
    n <- sum(x)                         #sample size
    min_dim <- min(dim(x)) - 1          #minimum number of dimensions - 1
    res <- sqrt(chi_stat/n/min_dim)     #Cramer's V
    names(res) <- "Cramers.V"
    return(res)
}
cont_table <- table(diamonds2$clarity,diamonds2$cut) #contingency table
print(cont_table)
round(calc_Cramers_V(cont_table),digits=3)
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cat)

```
> print(cont_table)
        cut
clarity Fair Good Very Good Premium Ideal
   I1    210   95        84     203   146
   SI2   466 1078      2100    2943  2597
   SI1   408 1560      3240    3573  4282
   VS2   261  978      2590    3356  5069
   VS1   169  648      1774    1989  3588
   VVS2   69  286      1235     870  2606
   VVS1   17  186       789     615  2047
   IF      9   71       268     230  1212
> calc_Cramers_V(cont_tab)
Cramers.V
0.143
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cont)

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cont)

```
library("tidyverse")
library("hexbin")
?diamonds
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cont)

```
# A tibble: 53,917 x 10
   carat cut       color clarity depth table price     x     y     z
   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
 7 0.24  Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
 8 0.26  Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
 9 0.22  Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
10 0.23  Very Good H     VS1      59.4    61   338  4     4.05  2.39
# ... with 53,907 more rows
```

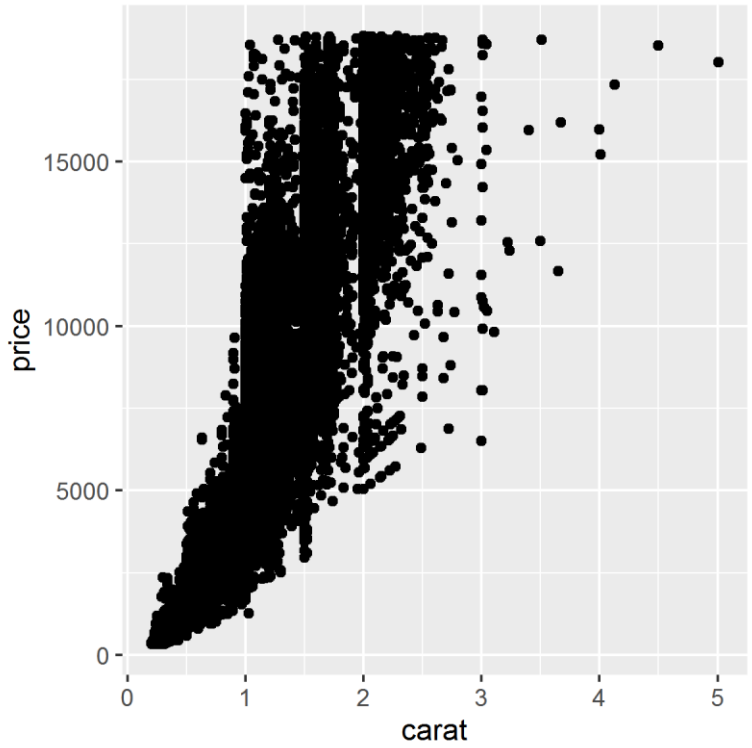
Exploração dos dados

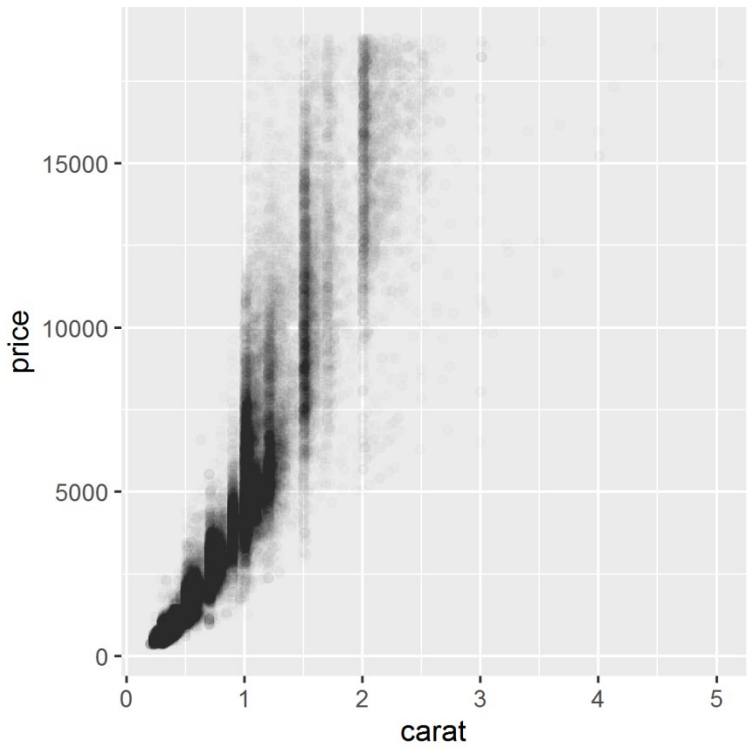
» 3.4. Exploração: distribuição 2D (2 cont)

```
library("tidyverse")
library("hexbin")
?diamonds
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
diamonds2 %>% ggplot() + geom_point(mapping=aes(x=carat,y=price))
diamonds2 %>% ggplot() + geom_point(mapping=aes(x=carat,y=price),alpha=0.01)
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cont)





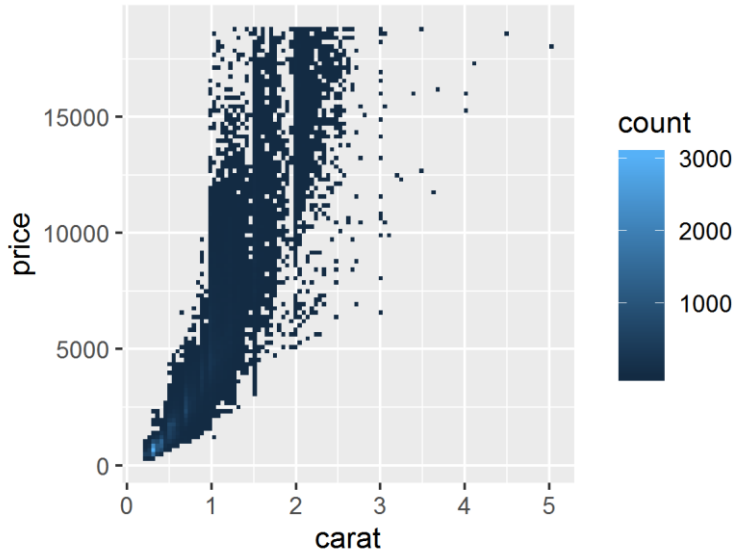
Exploração dos dados

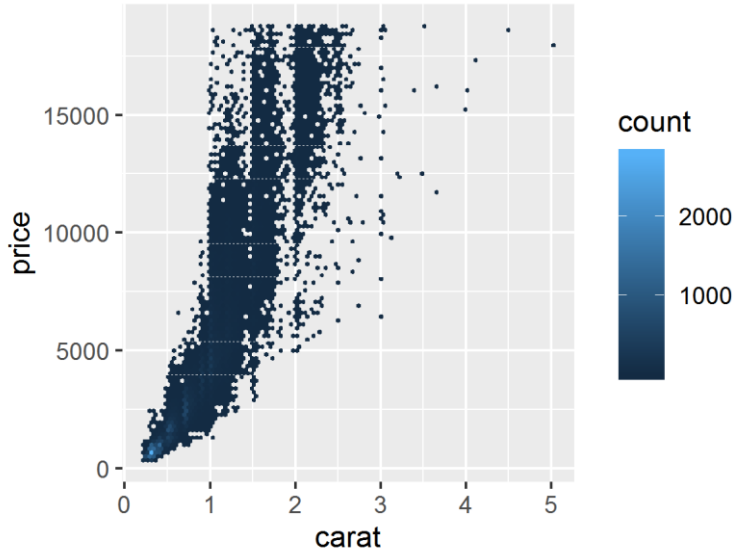
» 3.4. Exploração: distribuição 2D (2 cont)

```
library("tidyverse")
library("hexbin")
?diamonds
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
diamonds2 %>% ggplot() + geom_point(mapping=aes(x=carat,y=price))
diamonds2 %>% ggplot() + geom_point(mapping=aes(x=carat,y=price),alpha=0.01)
diamonds2 %>% ggplot() + geom_bin2d(mapping=aes(x=carat,y=price),bins=100)
diamonds2 %>% ggplot() + geom_hex(mapping=aes(x=carat,y=price),bins=100)
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cont)





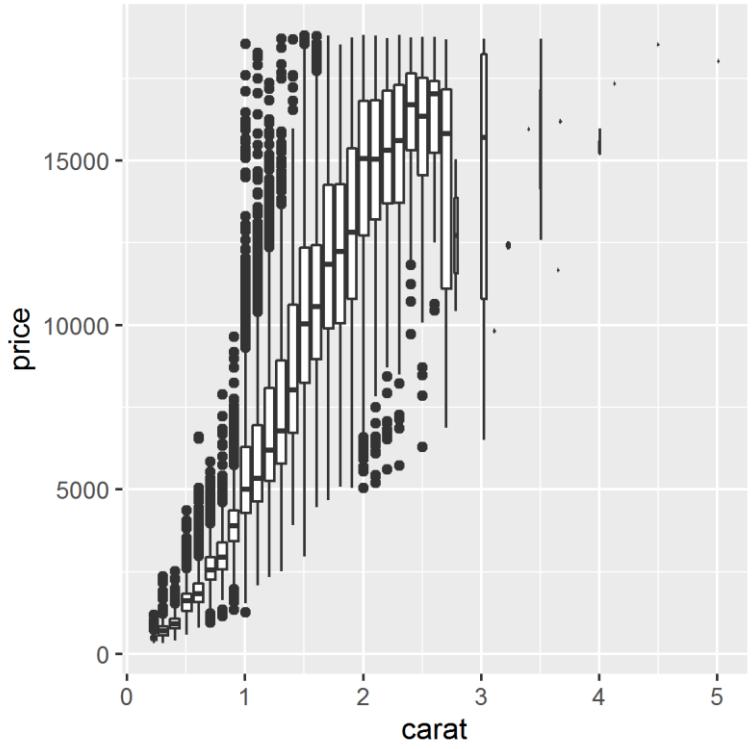
Exploração dos dados

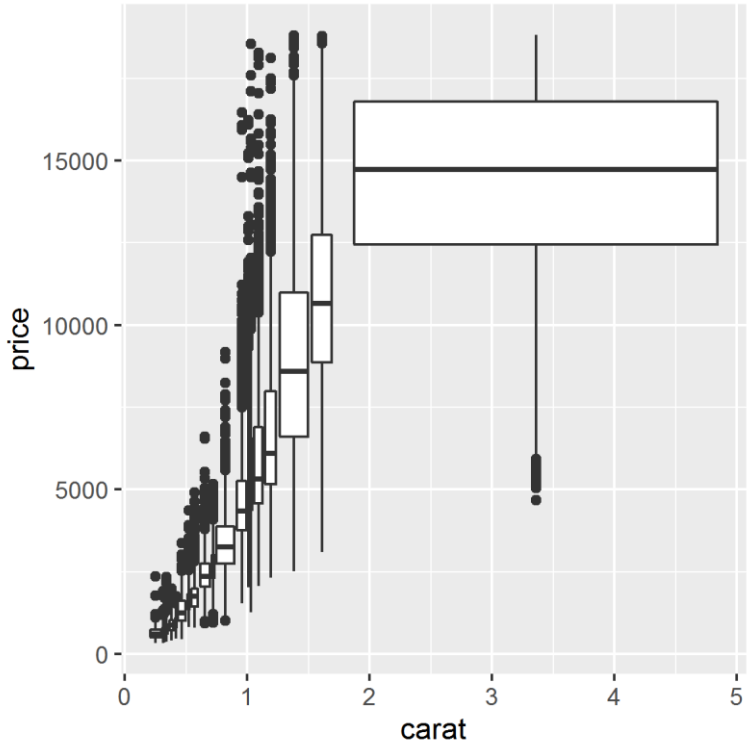
» 3.4. Exploração: distribuição 2D (2 cont)

```
library("tidyverse")
library("hexbin")
?diamonds
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
diamonds2 %>% ggplot() + geom_point(mapping=aes(x=carat,y=price))
diamonds2 %>% ggplot() + geom_point(mapping=aes(x=carat,y=price),alpha=0.01)
diamonds2 %>% ggplot() + geom_bin2d(mapping=aes(x=carat,y=price),bins=100)
diamonds2 %>% ggplot() + geom_hex(mapping=aes(x=carat,y=price),bins=100)
diamonds2 %>% ggplot() +
  geom_boxplot(mapping=aes(x=carat,y=price,group=cut_width(carat,0.1)))
diamonds2 %>% ggplot() +
  geom_boxplot(mapping=aes(x=carat,y=price,group=cut_number(carat,20)))
```

Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cont)





Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cont)

```
library("tidyverse")
library("hexbin")
?diamonds
diamonds2 <- diamonds %>%
 filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10))
print(diamonds2)
diamonds2 %>% ggplot() + geom_point(mapping=aes(x=carat,y=price))
diamonds2 %>% ggplot() + geom_point(mapping=aes(x=carat,y=price),alpha=0.01)
diamonds2 %>% ggplot() + geom_bin2d(mapping=aes(x=carat,y=price),bins=100)
diamonds2 %>% ggplot() + geom_hex(mapping=aes(x=carat,y=price),bins=100)
diamonds2 %>% ggplot() +
  geom_boxplot(mapping=aes(x=carat,y=price,group=cut_width(carat,0.1)))
diamonds2 %>% ggplot() +
  geom_boxplot(mapping=aes(x=carat,y=price,group=cut_number(carat,20)))
diamonds2 %>% summarize(res = cor(carat,price))
```

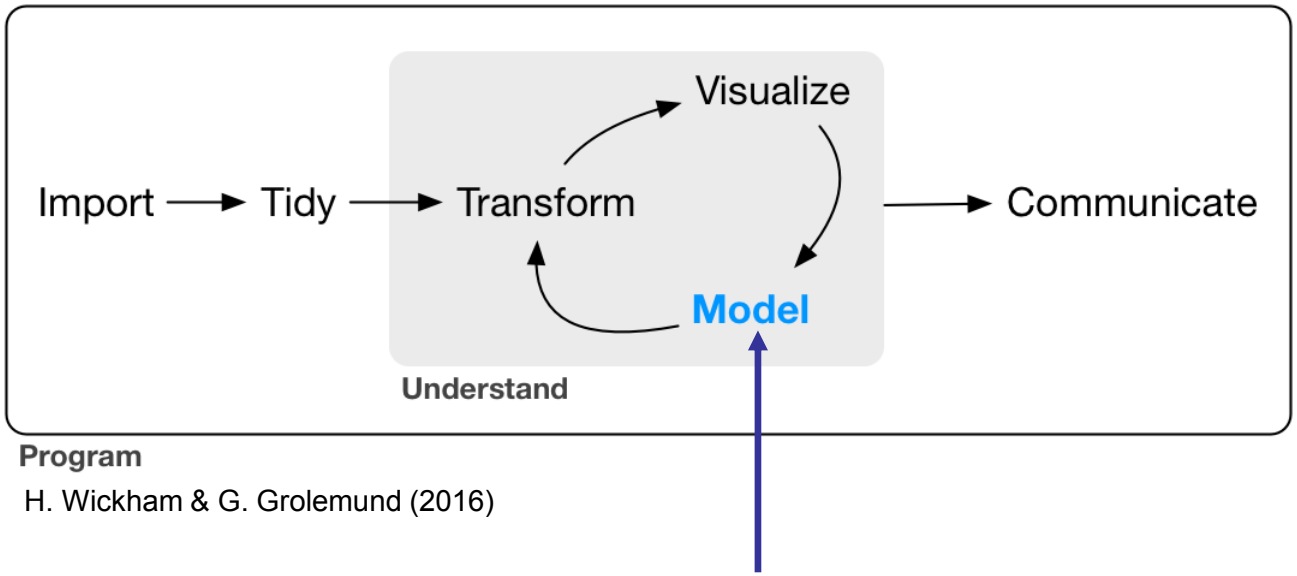
Exploração dos dados

» 3.4. Exploração: distribuição 2D (2 cont)

```
# A tibble: 1 x 1
    res
  <dbl>
1 0.922
```

Modelos e inferências

» Esquema geral



H. Wickham & G. Grolemund (2016)

Criação de modelos e inferências

Modelos e inferências

» 1. Modelação: exemplo

Modelos e inferências

» 1. Modelação: exemplo

"All models are wrong, but some are useful"

"Now it would be very remarkable if any system existing in the real world could be exactly represented by any simple model. However, cunningly chosen parsimonious models often do provide remarkably useful approximations."

George Box

Modelos e inferências

» 1. Modelação: exemplo

```
library("tidyverse")
library("hexbin")
library("modelr")
?diamonds
print(diamonds)
```

Modelos e inferências

» 1. Modelação: exemplo

```
# A tibble: 53,940 x 10
   carat cut       color clarity depth table price     x     y     z
   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
 7 0.24  Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
 8 0.26  Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
 9 0.22  Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
10 0.23  Very Good H     VS1      59.4    61   338  4     4.05  2.39
# ... with 53,930 more rows
```

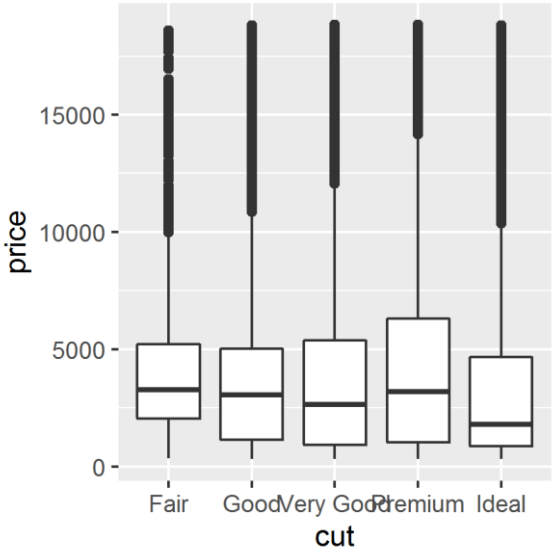
Modelos e inferências

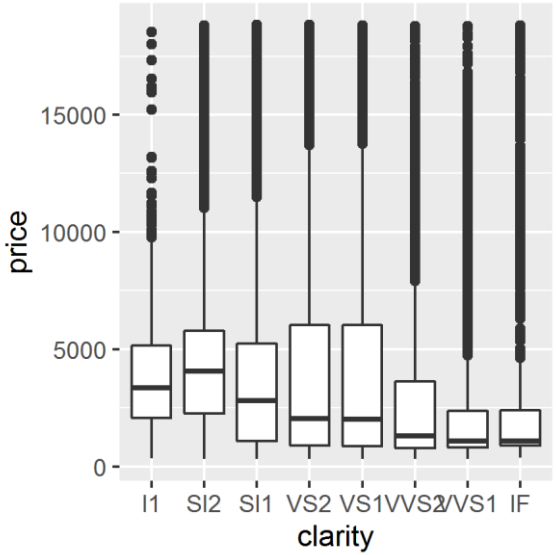
» 1. Modelação: exemplo

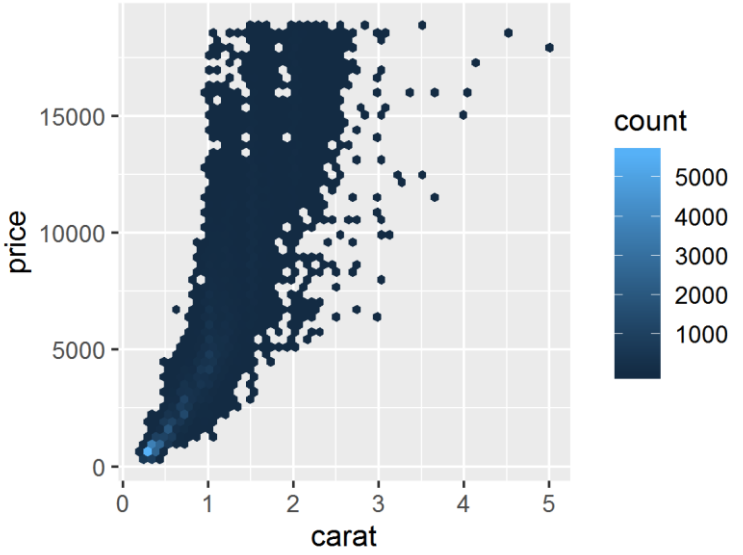
```
library("tidyverse")
library("hexbin")
library("modelr")
?diamonds
print(diamonds)
diamonds %>% ggplot() + geom_boxplot(aes(cut,price))          #price ~ cut
diamonds %>% ggplot() + geom_boxplot(aes(clarity,price))      #price ~ clarity
diamonds %>% ggplot() + geom_hex(aes(carat,price),bins=50)    #price ~ carat
```

Modelos e inferências

» 1. Modelação: exemplo







Modelos e inferências

» 1. Modelação: exemplo

```
library("tidyverse")
library("hexbin")
library("modelr")
?diamonds
print(diamonds)
diamonds %>% ggplot() + geom_boxplot(aes(cut,price))          #price ~ cut
diamonds %>% ggplot() + geom_boxplot(aes(clarity,price))      #price ~ clarity
diamonds %>% ggplot() + geom_hex(aes(carat,price),bins=50)    #price ~ carat
diamonds2 <- diamonds %>%
  filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10)) %>%
  filter(carat <= 2.5) %>%
  select(carat,cut,color,clarity,price) %>%
  mutate(lprice = log(price), lcarat = log(carat))
print(diamonds2)
```

Modelos e inferências

» 1. Modelação: exemplo

```
# A tibble: 53,792 x 7
   carat cut       color clarity price lprice lcarat
   <dbl> <ord>     <ord> <ord>   <int>  <dbl>  <dbl>
 1 0.23  Ideal     E     SI2       326   5.79  -1.47
 2 0.21  Premium   E     SI1       326   5.79  -1.56
 3 0.23  Good      E     VS1       327   5.79  -1.47
 4 0.290 Premium   I     VS2       334   5.81  -1.24
 5 0.31  Good      J     SI2       335   5.81  -1.17
 6 0.24  Very Good J     VVS2      336   5.82  -1.43
 7 0.24  Very Good I     VVS1      336   5.82  -1.43
 8 0.26  Very Good H     SI1       337   5.82  -1.35
 9 0.22  Fair      E     VS2       337   5.82  -1.51
10 0.23  Very Good H     VS1       338   5.82  -1.47
# ... with 53,782 more rows
```

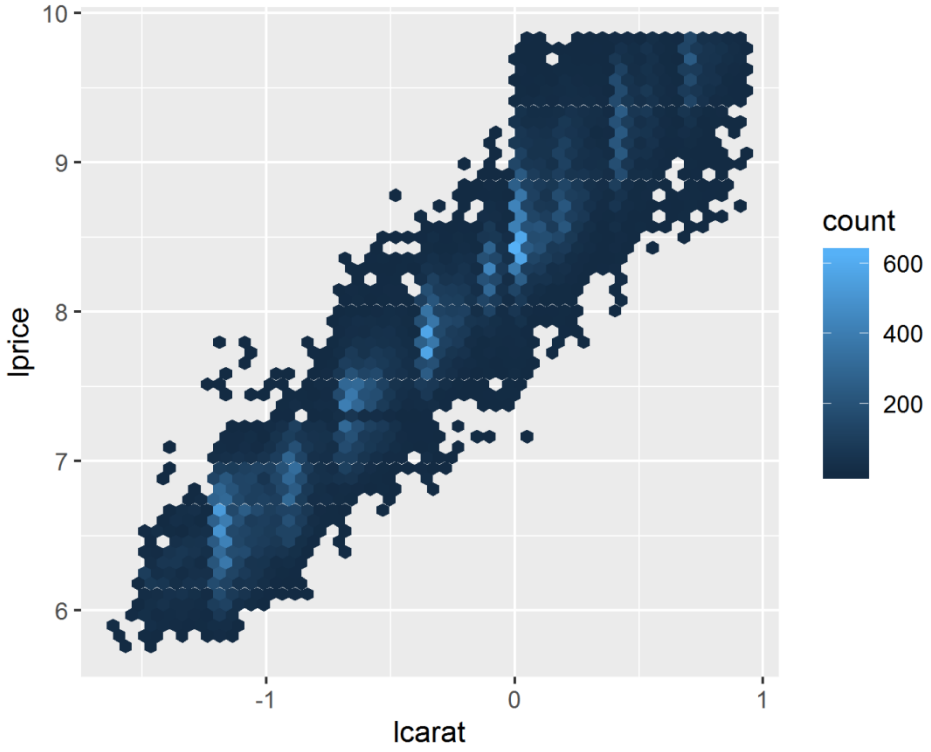
Modelos e inferências

» 1. Modelação: exemplo

```
library("tidyverse")
library("hexbin")
library("modelr")
?diamonds
print(diamonds)
diamonds %>% ggplot() + geom_boxplot(aes(cut,price))          #price ~ cut
diamonds %>% ggplot() + geom_boxplot(aes(clarity,price))      #price ~ clarity
diamonds %>% ggplot() + geom_hex(aes(carat,price),bins=50)    #price ~ carat
diamonds2 <- diamonds %>%
  filter((x > 0) & (y > 0 & y < 20) & (z > 0 & z < 10)) %>%
  filter(carat <= 2.5) %>%
  select(carat,cut,color,clarity,price) %>%
  mutate(lprice = log(price), lcarat = log(carat))
print(diamonds2)
diamonds2 %>% ggplot() + geom_hex(aes(lcarat,lprice),bins=50) #lprice ~ lcarat
```

Modelos e inferências

» 1. Modelação: exemplo



Modelos e inferências

» 1. Modelação: exemplo

```
diamonds_mod <- lm(lprice ~ lcarat,data=diamonds2)                    #fit model
summary(diamonds_mod)
```

Modelos e inferências

» 1. Modelação: exemplo

```
lm(formula = lprice ~ lcarat, data = diamonds2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.36147 -0.17011 -0.00586  0.16587  1.34117

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 8.452218   0.001365  6193.2   <2e-16 ***
lcarat      1.681463   0.001936   868.4   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2611 on 53790 degrees of freedom
Multiple R-squared:  0.9334,     Adjusted R-squared:  0.9334
F-statistic: 7.541e+05 on 1 and 53790 DF,  p-value: < 2.2e-16
```

Modelos e inferências

» 1. Modelação: exemplo

```
diamonds_mod <- lm(lprice ~ lcarat,data=diamonds2)                    #fit model
summary(diamonds_mod)
diamonds2 <- diamonds2 %>%
  add_predictions(diamonds_mod,"lpred") %>%
  add_residuals(diamonds_mod,"lresid") %>%
  mutate(pred = exp(lpred))
print(diamonds2)
```

Modelos e inferências

» 1. Modelação: exemplo

```
# A tibble: 53,792 x 10
   carat cut       color clarity price lprice lcarat lpred  lresid  pred
   <dbl> <ord>     <ord> <ord>   <int>  <dbl>  <dbl> <dbl>   <dbl> <dbl>
 1 0.23  Ideal     E     SI2       326   5.79  -1.47  5.98 -0.194   396.
 2 0.21  Premium   E     SI1       326   5.79  -1.56  5.83 -0.0411  340.
 3 0.23  Good      E     VS1       327   5.79  -1.47  5.98 -0.191   396.
 4 0.290 Premium   I     VS2       334   5.81  -1.24  6.37 -0.560   585.
 5 0.31  Good      J     SI2       335   5.81  -1.17  6.48 -0.669   654.
 6 0.24  Very Good J     VVS2      336   5.82  -1.43  6.05 -0.235   425.
 7 0.24  Very Good I     VVS1      336   5.82  -1.43  6.05 -0.235   425.
 8 0.26  Very Good H     SI1       337   5.82  -1.35  6.19 -0.367   486.
 9 0.22  Fair      E     VS2       337   5.82  -1.51  5.91 -0.0862  367.
10 0.23  Very Good H     VS1       338   5.82  -1.47  5.98 -0.158   396.
# ... with 53,782 more rows
```

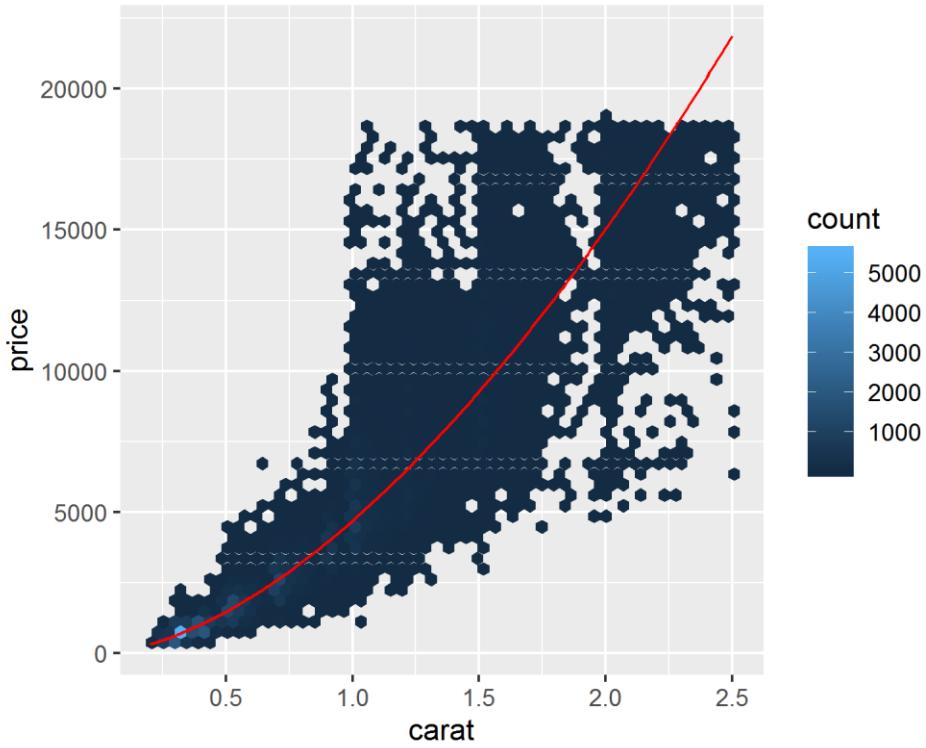
Modelos e inferências

» 1. Modelação: exemplo

```
diamonds_mod <- lm(lprice ~ lcarat,data=diamonds2)                    #fit model
summary(diamonds_mod)
diamonds2 <- diamonds2 %>%
  add_predictions(diamonds_mod,"lpred") %>%
  add_residuals(diamonds_mod,"lresid") %>%
  mutate(pred = exp(lpred))
print(diamonds2)
diamonds2 %>% ggplot() +
  geom_hex(aes(carat,price),bins=50) +
  geom_line(aes(carat,pred),color="red")                              #price ~ carat
```

Modelos e inferências

» 1. Modelação: exemplo



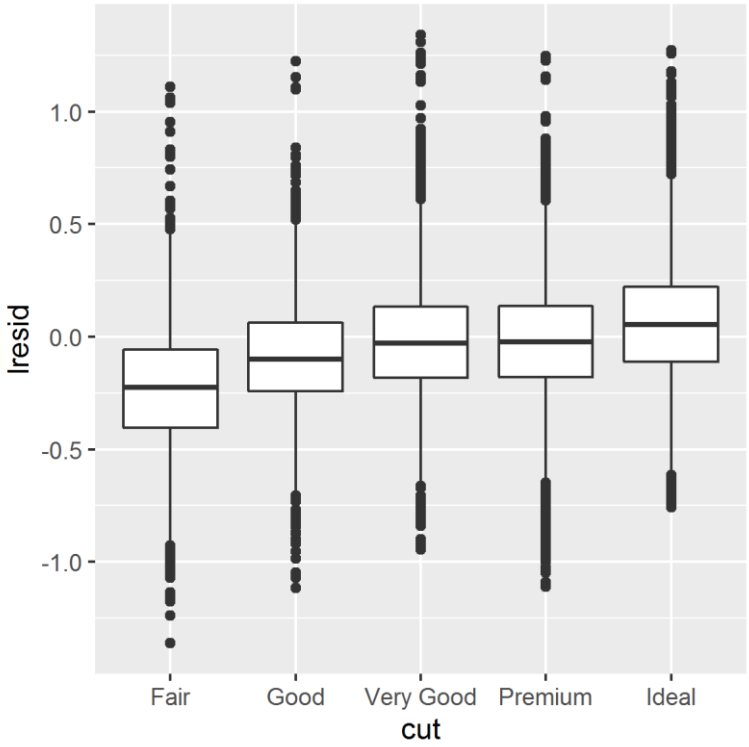
Modelos e inferências

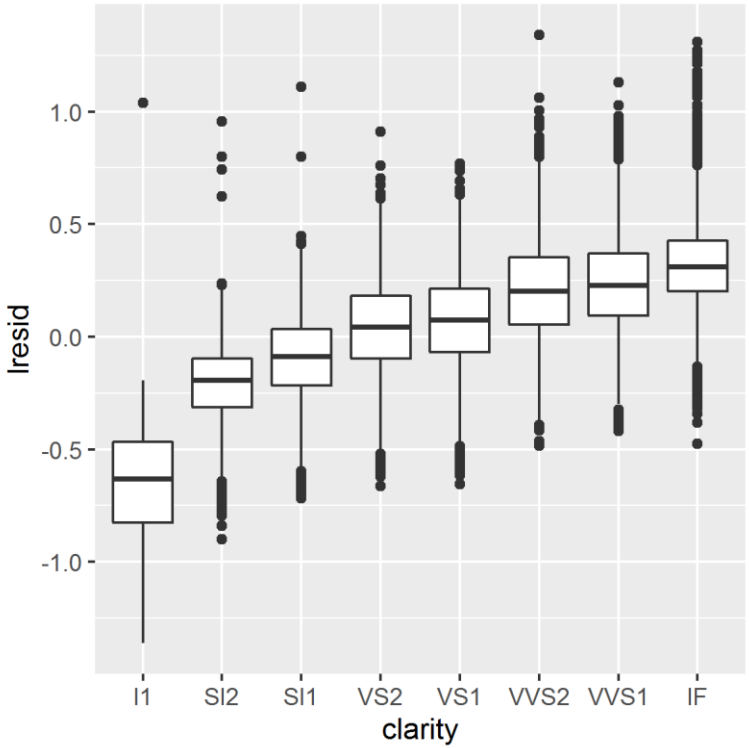
» 1. Modelação: exemplo

```
diamonds_mod <- lm(lprice ~ lcarat,data=diamonds2)                #fit model
summary(diamonds_mod)
diamonds2 <- diamonds2 %>%
  add_predictions(diamonds_mod,"lpred") %>%
  add_residuals(diamonds_mod,"lresid") %>%
  mutate(pred = exp(lpred))
print(diamonds2)
diamonds2 %>% ggplot() +
  geom_hex(aes(carat,price),bins=50) +
  geom_line(aes(carat,pred),color="red")                          #price ~ carat
diamonds2 %>% ggplot() + geom_boxplot(aes(cut,lresid))            #lresid ~ cut
diamonds2 %>% ggplot() + geom_boxplot(aes(clarity,lresid))        #lresid ~ clarity
```

Modelos e inferências

» 1. Modelação: exemplo





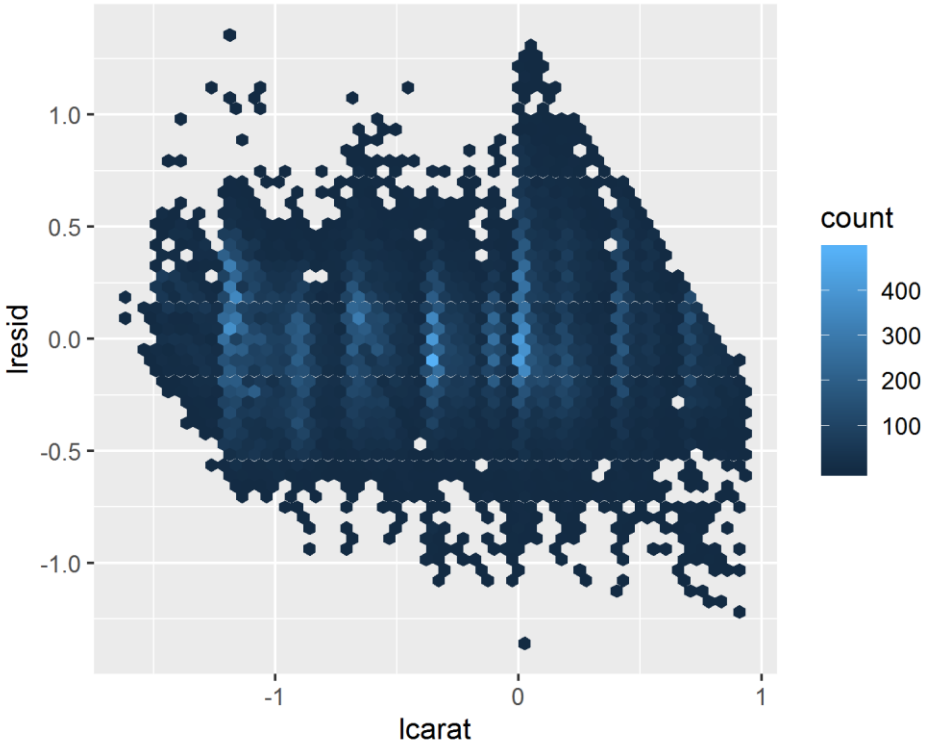
Modelos e inferências

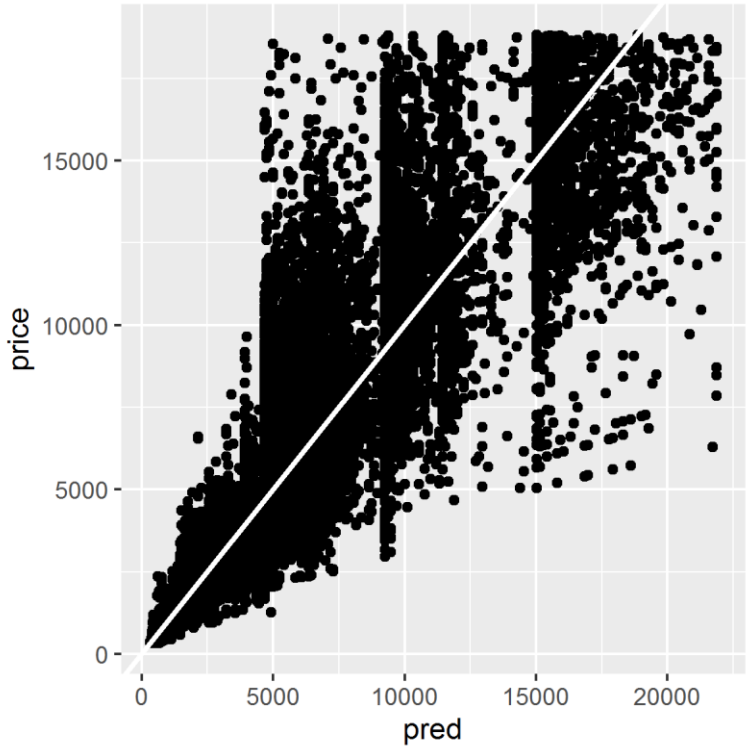
» 1. Modelação: exemplo

```
diamonds_mod <- lm(lprice ~ lcarat,data=diamonds2)                #fit model
summary(diamonds_mod)
diamonds2 <- diamonds2 %>%
  add_predictions(diamonds_mod,"lpred") %>%
  add_residuals(diamonds_mod,"lresid") %>%
  mutate(pred = exp(lpred))
print(diamonds2)
diamonds2 %>% ggplot() +
  geom_hex(aes(carat,price),bins=50) +
  geom_line(aes(carat,pred),color="red")                          #price ~ carat
diamonds2 %>% ggplot() + geom_boxplot(aes(cut,lresid))            #lresid ~ cut
diamonds2 %>% ggplot() + geom_boxplot(aes(clarity,lresid))        #lresid ~ clarity
diamonds2 %>% ggplot() + geom_hex(aes(lcarat,lresid),bins=50)     #lresid ~ lcarat
diamonds2 %>% ggplot() + geom_point(aes(pred,price)) +
  geom_abline(aes(intercept=0,slope=1),size=1,color="white")      #price ~ pred
```

Modelos e inferências

» 1. Modelação: exemplo





Modelos e inferências

» 1. Modelação: exemplo

```
diamonds_mod2  <- lm(lprice ~ lcarat + clarity + cut,data=diamonds2) #fit model
summary(diamonds_mod2)
```

Modelos e inferências

» 1. Modelação: exemplo

```
lm(formula = lprice ~ lcarat + clarity + cut, data = diamonds2)

Residuals:
     Min       1Q   Median       3Q      Max
-0.76285 -0.11807  0.01112  0.12249  1.94161

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept)  8.4703037  0.0015922 5319.725  < 2e-16 ***
lcarat       1.8152405  0.0015005 1209.730  < 2e-16 ***
```

[...]

```
Residual standard error: 0.185 on 53779 degrees of freedom
Multiple R-squared:  0.9666,    Adjusted R-squared:  0.9666
F-statistic: 1.297e+05 on 12 and 53779 DF,  p-value: < 2.2e-16
```

Modelos e inferências

» 1. Modelação: exemplo

```
diamonds_mod2  <- lm(lprice ~ lcarat + clarity + cut,data=diamonds2) #fit model
summary(diamonds_mod2)
diamonds2 <- diamonds2 %>%
  add_predictions(diamonds_mod2,"lpred2") %>%
  add_residuals(diamonds_mod2,"lresid2") %>%
  mutate(pred2 = exp(lpred2))
print(diamonds2)
```

Modelos e inferências

» 1. Modelação: exemplo

```
# A tibble: 53,792 x 13
   carat cut       color clarity price lprice lcarat lpred  lresid  pred lpred2 lresid2 pred2
   <dbl> <ord>     <ord> <ord>   <int>  <dbl>  <dbl> <dbl>   <dbl> <dbl>  <dbl>   <dbl> <dbl>
 1 0.23  Ideal     E     SI2       326   5.79  -1.47  5.98 -0.194   396.   6.04 -0.257   421.
 2 0.21  Premium   E     SI1       326   5.79  -1.56  5.83 -0.0411  340.   5.81 -0.0190  332.
 3 0.23  Good      E     VS1       327   5.79  -1.47  5.98 -0.191   396.   5.89 -0.0976  361.
 4 0.290 Premium   I     VS2       334   5.81  -1.24  6.37 -0.560   585.   6.11 -0.295   448.
 5 0.31  Good      J     SI2       335   5.81  -1.17  6.48 -0.669   654.   6.01 -0.198   408.
 6 0.24  Very G~   J     VVS2      336   5.82  -1.43  6.05 -0.235   425.   5.64  0.175   282.
 7 0.24  Very G~   I     VVS1      336   5.82  -1.43  6.05 -0.235   425.   5.77  0.0425  322.
 8 0.26  Very G~   H     SI1       337   5.82  -1.35  6.19 -0.367   486.   6.02 -0.195   410.
 9 0.22  Fair      E     VS2       337   5.82  -1.51  5.91 -0.0862  367.   5.65  0.169   285.
10 0.23  Very G~   H     VS1       338   5.82  -1.47  5.98 -0.158   396.   5.80  0.0227  330.
# ... with 53,782 more rows
```

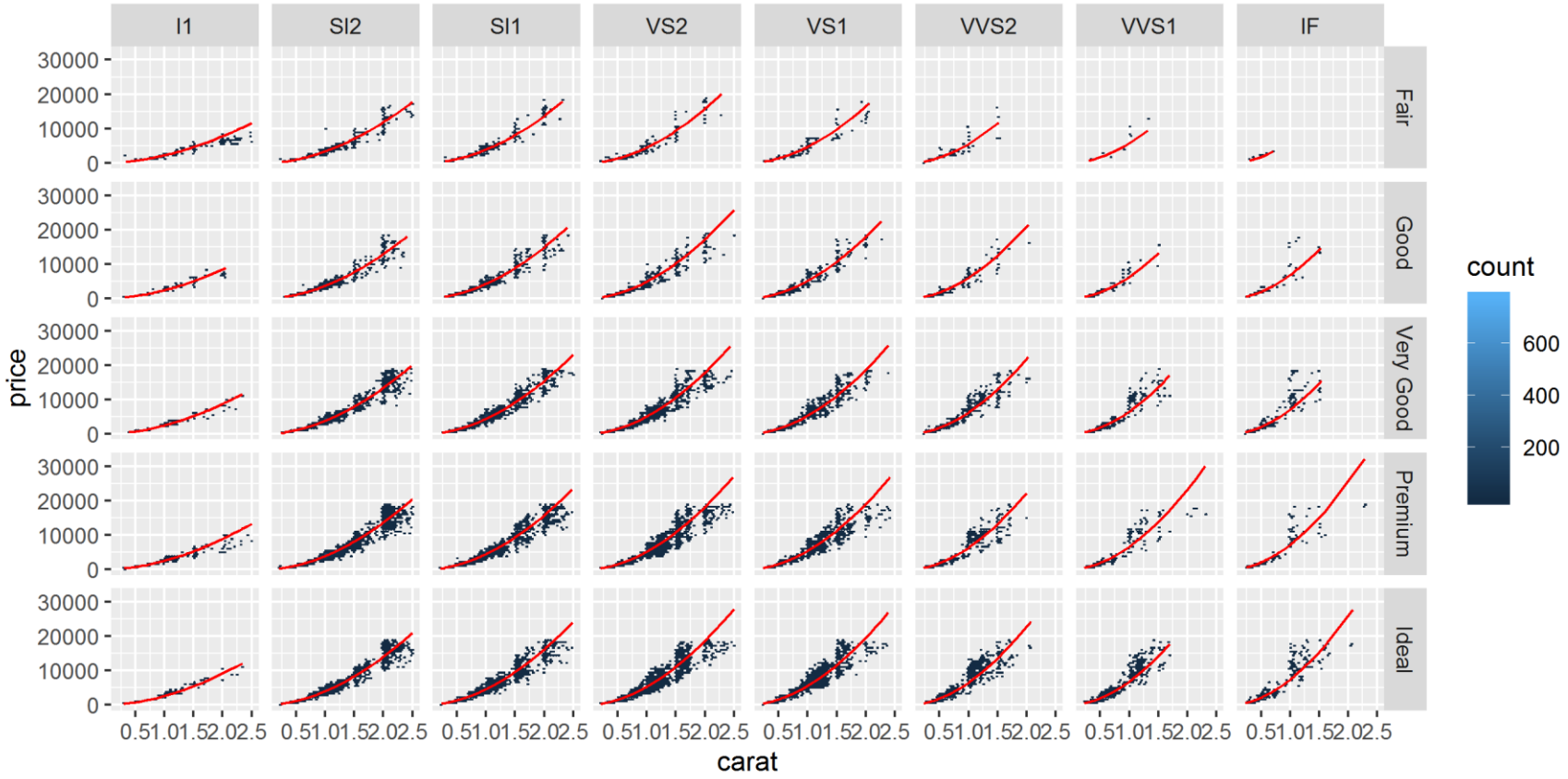
Modelos e inferências

» 1. Modelação: exemplo

```
diamonds_mod2  <- lm(lprice ~ lcarat + clarity + cut,data=diamonds2) #fit model
summary(diamonds_mod2)
diamonds2 <- diamonds2 %>%
  add_predictions(diamonds_mod2,"lpred2") %>%
  add_residuals(diamonds_mod2,"lresid2") %>%
  mutate(pred2 = exp(lpred2))
print(diamonds2)
diamonds2 %>%
  ggplot() +
    geom_hex(aes(carat,price),bins=50) +
    geom_line(aes(carat,pred2),color="red") +
    facet_grid(cut ~ clarity)                                    #price ~ carat
```

Modelos e inferências

» 1. Modelação: exemplo



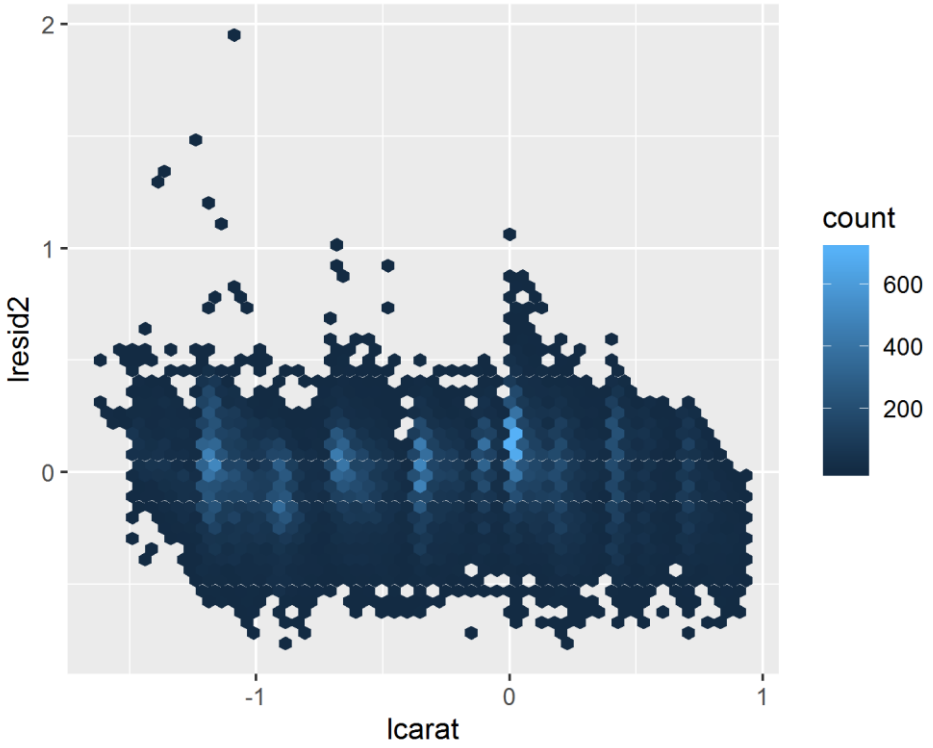
Modelos e inferências

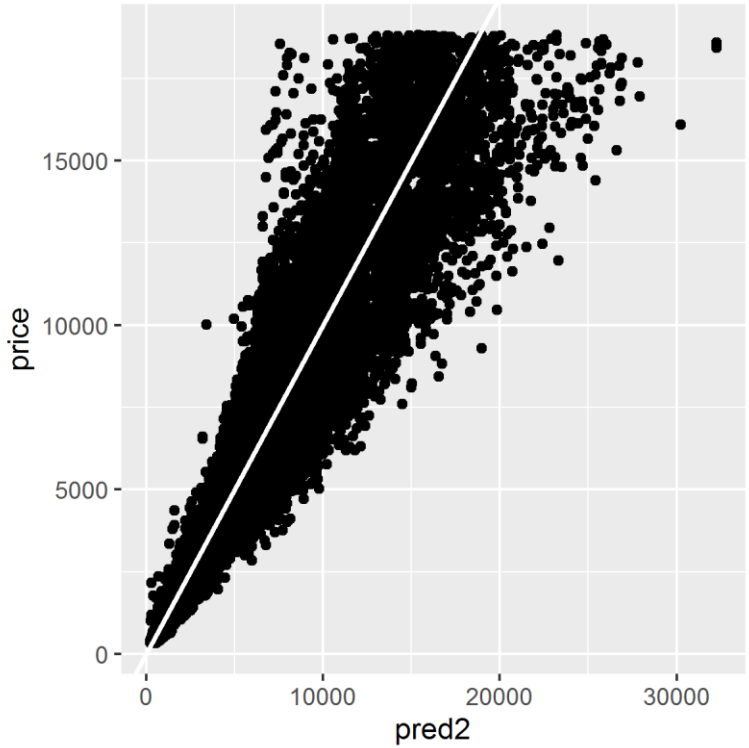
» 1. Modelação: exemplo

```
diamonds_mod2  <- lm(lprice ~ lcarat + clarity + cut,data=diamonds2) #fit model
summary(diamonds_mod2)
diamonds2 <- diamonds2 %>%
  add_predictions(diamonds_mod2,"lpred2") %>%
  add_residuals(diamonds_mod2,"lresid2") %>%
  mutate(pred2 = exp(lpred2))
print(diamonds2)
diamonds2 %>%
  ggplot() +
    geom_hex(aes(carat,price),bins=50) +
    geom_line(aes(carat,pred2),color="red") +
    facet_grid(cut ~ clarity)                                    #price ~ carat
diamonds2 %>% ggplot() + geom_hex(aes(lcarat,lresid2),bins=50)   #lresid ~ lcarat
diamonds2 %>% ggplot() + geom_point(aes(pred2,price)) +
  geom_abline(aes(intercept=0,slope=1),size=1,color="white")     #price ~ pred2
```

Modelos e inferências

» 1. Modelação: exemplo





Modelos e inferências

» 2. Ajustamento (`modelr`)

Modelos e inferências

» 2. Ajustamento (`modelr`)

```
library("tidyverse")
library("modelr")
set.seed(12345)
real_a1 <- 4.22
real_a2 <- 2.05
x <- round(runif(n=30,min=0,max=10),digits=2)
y <- round(real_a1*x + real_a2 + rnorm(n=30,mean=0,sd=1),digits=2)
sim1 <- tibble(x,y)
print(sim1)
```

Modelos e inferências

» 2. Ajustamento (`modelr`)

```
# A tibble: 30 x 2
       x     y
   <dbl> <dbl>
 1  7.21 33.3
 2  8.76 38.1
 3  7.61 33.8
 4  8.86 40.6
 5  4.56 21.6
 6  1.66  9.83
 7  3.25 17.2
 8  5.09 22.9
 9  7.28 31.2
10  9.9  42.2
# ... with 20 more rows
```

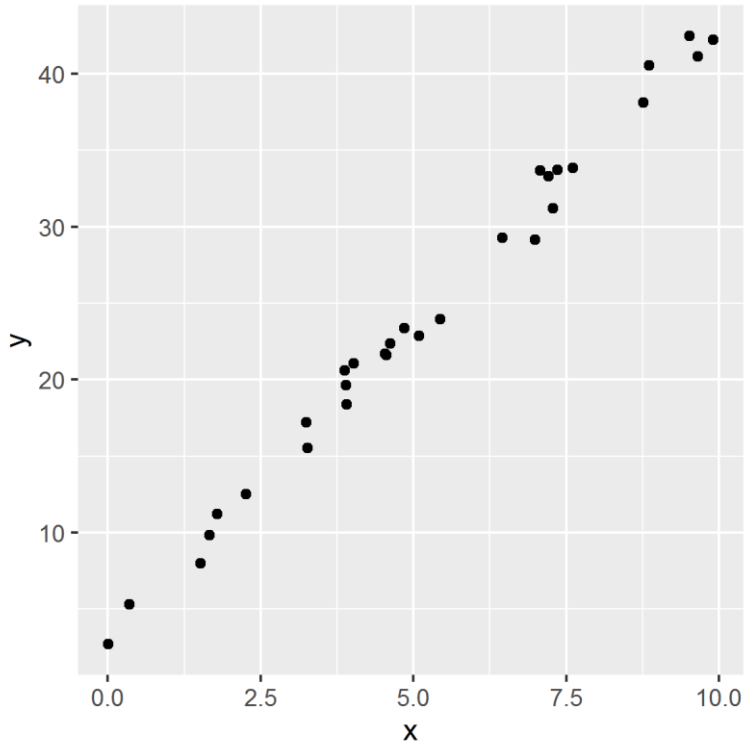
Modelos e inferências

» 2. Ajustamento (`modelr`)

```
library("tidyverse")
library("modelr")
set.seed(12345)
real_a1 <- 4.22
real_a2 <- 2.05
x <- round(runif(n=30,min=0,max=10),digits=2)
y <- round(real_a1*x + real_a2 + rnorm(n=30,mean=0,sd=1),digits=2)
sim1 <- tibble(x,y)
print(sim1)
sim1 %>% ggplot() + geom_point(aes(x,y))
```

Modelos e inferências

» 2. Ajustamento (`modelr`)



Modelos e inferências

» 2.1. Ajustamento: *random search*

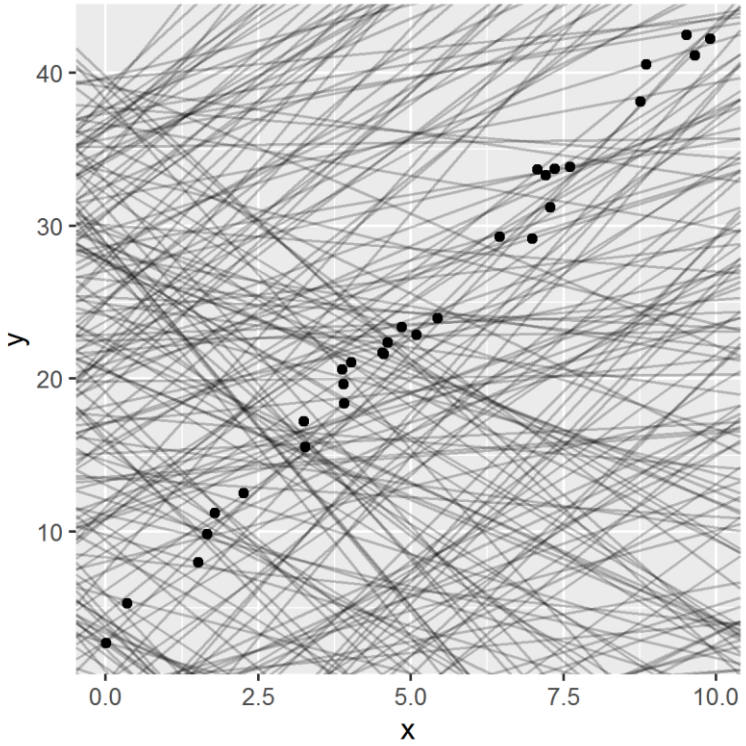
Modelos e inferências

» 2.1. Ajustamento: *random search*

```
models <- tibble(
  a1 = runif(250,-20,40),
  a2 = runif(250,-5,5)
)
sim1 %>% ggplot() +
  geom_abline(data=models,aes(intercept=a1,slope=a2),alpha=0.25) +
  geom_point(aes(x,y))
```

Modelos e inferências

» 2.1. Ajustamento: *random search*



Modelos e inferências

» 2.1. Ajustamento: *random search*

```
#Linear regression model
model1 <- function(a,dat){
  a[1] + dat$x*a[2]
}
#Calculate Root-mean-squared-deviation
measure_distance <- function(params,dat){
  diff <- dat$y - model1(params,dat)
  sqrt(mean(diff^2))
}
#Helper function to calculate RMSD for synthetic data "sim1"
sim1_dist <- function(a1,a2){
  measure_distance(c(a1,a2),dat=sim1)
}
models <- models %>% mutate(RMSD = map2_dbl(a1,a2,sim1_dist))
print(models)
```

Modelos e inferências

» 2.1. Ajustamento: *random search*

```
# A tibble: 250 x 3
       a1     a2  RMSD
    <dbl>  <dbl> <dbl>
 1  35.7   0.792 17.9
 2  28.5   1.94  15.5
 3 -15.3   4.11  18.2
 4  16.1   3.38   9.57
 5  22.9   0.144 10.7
 6  10.8   1.27  10.2
 7  23.2   2.95  14.6
 8  25.0   4.44  23.8
 9 -14.3  -4.56  66.4
10   3.87  2.53   8.34
# ... with 240 more rows
```

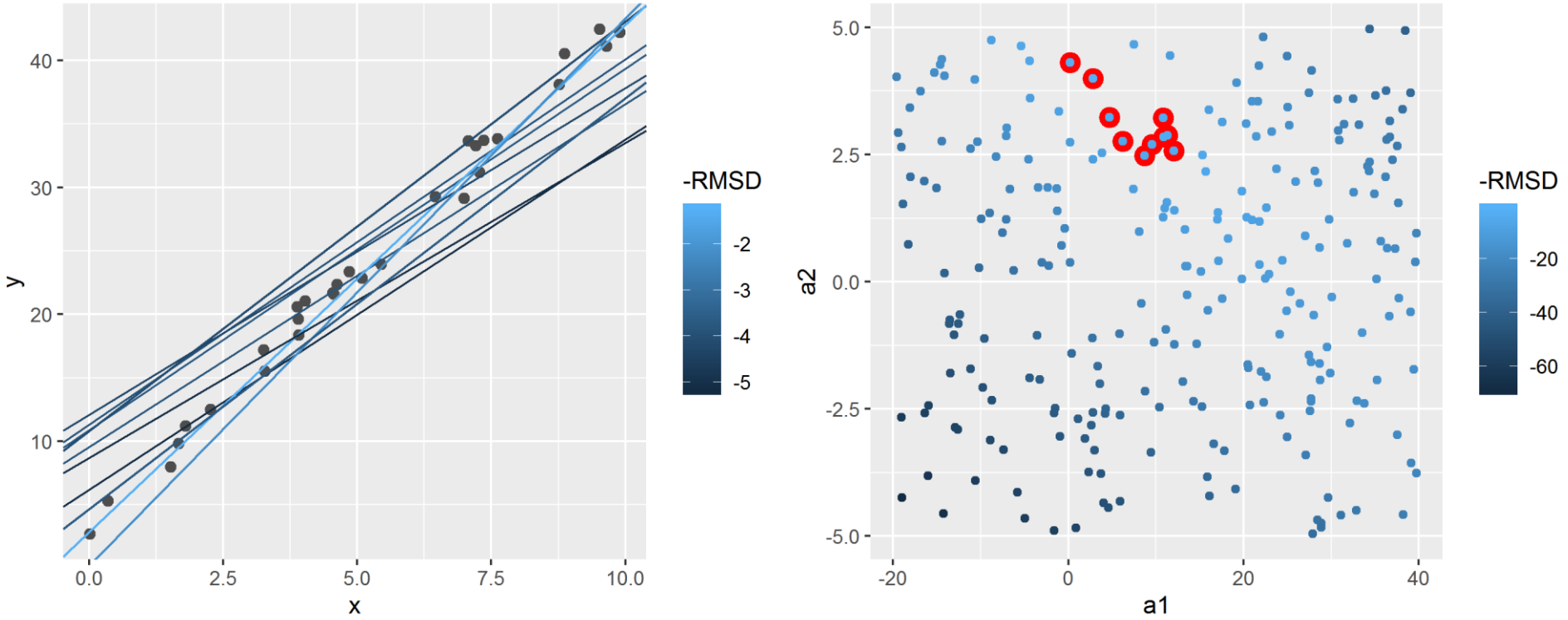
Modelos e inferências

» 2.1. Ajustamento: *random search*

```
best_models <- models %>%
  filter(rank(RMSD) <= 10)
sim1 %>% ggplot() +
  geom_point(aes(x,y),size=2,color="grey30") +
  geom_abline(data=best_models,aes(intercept=a1,slope=a2,color=-RMSD))
models %>% ggplot() +
  geom_point(data=best_models,aes(a1,a2),size=4,color="red") +
  geom_point(aes(a1,a2,color=-RMSD))
```

Modelos e inferências

» 2.1. Ajustamento: *random search*



Modelos e inferências

» 2.2. Ajustamento: *grid search*

Modelos e inferências

» 2.2. Ajustamento: *grid search*

```
models_grid <- expand.grid(
  a1 = seq(0,15,length=25),
  a2 = seq(2,5,length=25)) %>%
    mutate(RMSD=map2_dbl(a1,a2,sim1_dist))
head(models_grid,n=15)
```

Modelos e inferências

» 2.2. Ajustamento: *grid search*

```
      a1 a2      RMSD
1  0.000  2 14.947724
2  0.625  2 14.370600
3  1.250  2 13.797648
4  1.875  2 13.229409
5  2.500  2 12.666519
6  3.125  2 12.109723
7  3.750  2 11.559901
8  4.375  2 11.018099
9  5.000  2 10.485559
10 5.625  2  9.963766
11 6.250  2  9.454500
12 6.875  2  8.959897
13 7.500  2  8.482522
14 8.125  2  8.025451
15 8.750  2  7.592352
```

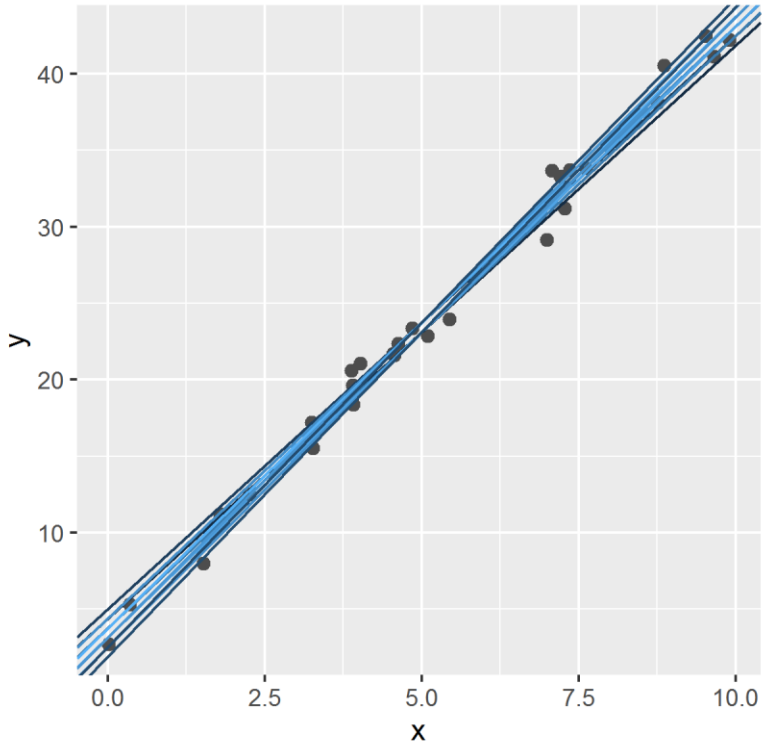
Modelos e inferências

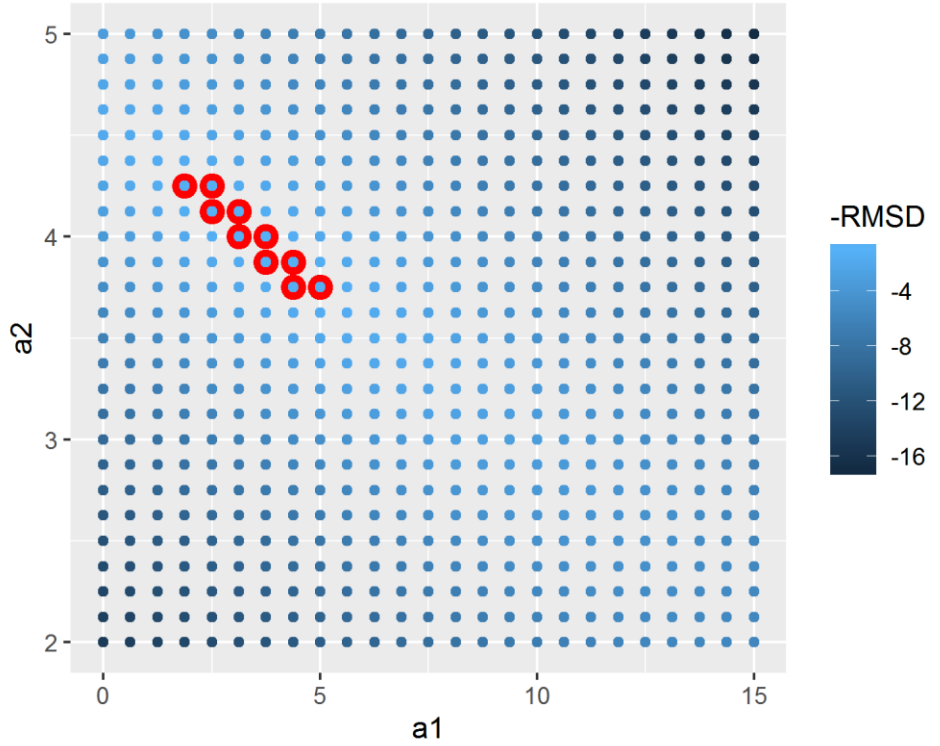
» 2.2. Ajustamento: *grid search*

```
models_grid <- expand.grid(
  a1 = seq(0,15,length=25),
  a2 = seq(2,5,length=25)) %>%
    mutate(RMSD=map2_dbl(a1,a2,sim1_dist))
head(models_grid,n=15)
best_grid <- models_grid %>%
  filter(rank(RMSD) <= 10)
sim1 %>% ggplot() +
  geom_point(aes(x,y),size=2,color="grey30") +
  geom_abline(data=best_grid,aes(intercept=a1,slope=a2,color=-RMSD))
models_grid %>% ggplot() +
  geom_point(data=best_grid,aes(a1,a2),size=4,color="red") +
  geom_point(aes(a1,a2,color=-RMSD))
```

Modelos e inferências

» 2.2. Ajustamento: *grid search*





Modelos e inferências

» 2.3. Ajustamento: *optimization*

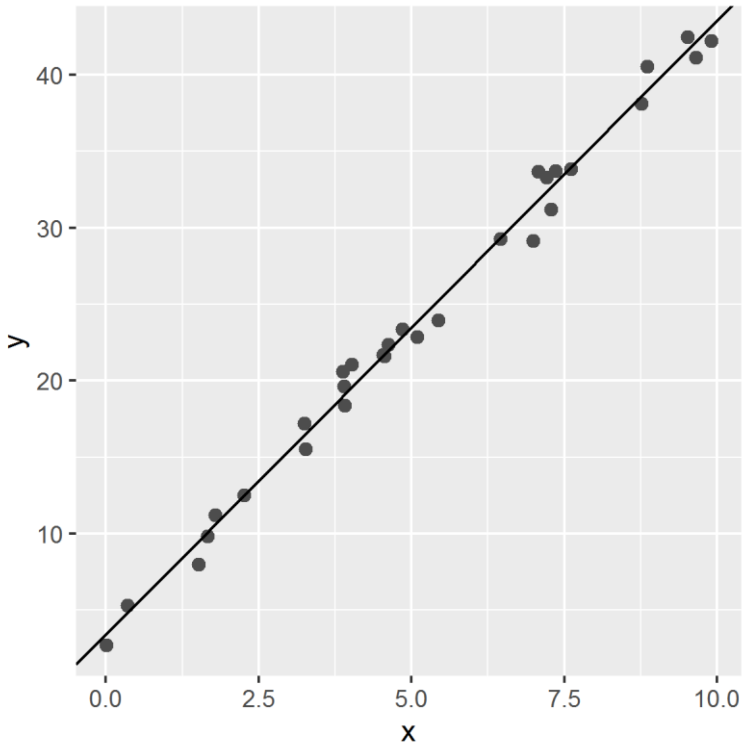
Modelos e inferências

» 2.3. Ajustamento: *optimization*

```
best_optim <- optim(c(0,0),measure_distance,dat=sim1)
best_a1 <- best_optim$par[1]
best_a2 <- best_optim$par[2]
sim1 %>% ggplot() +
  geom_point(aes(x,y),size=2,color="grey30") +
  geom_abline(aes(intercept=best_a1,slope=best_a2))
```

Modelos e inferências

» 2.3. Ajustamento: *optimization*



Modelos e inferências

» 2.4. Ajustamento: *least-squares*

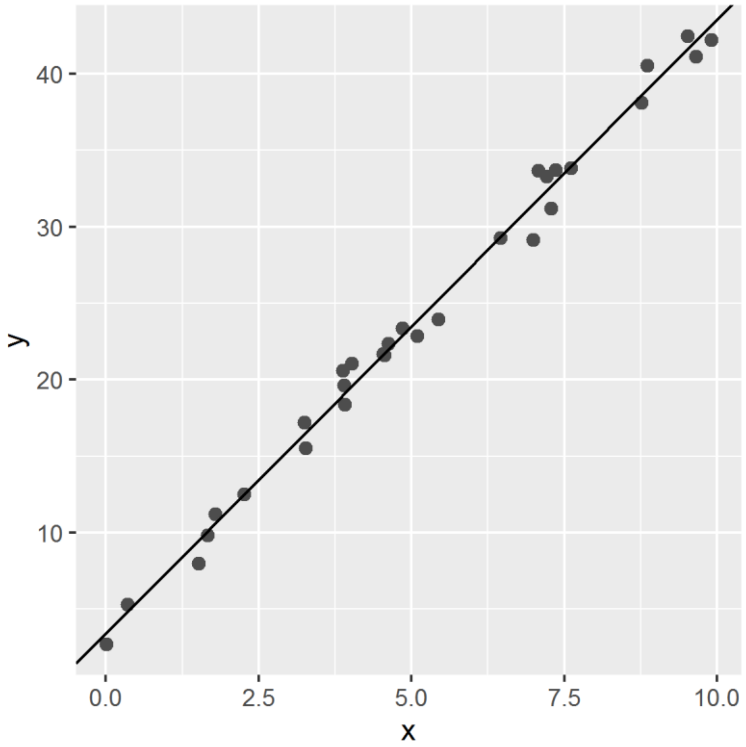
Modelos e inferências

» 2.4. Ajustamento: *least-squares*

```
best_lm <- lm(y ~ x,data=sim1)
best_a1 <- best_lm$coef[1]
best_a2 <- best_lm$coef[2]
sim1 %>% ggplot() +
  geom_point(aes(x,y),size=2,color="grey30") +
  geom_abline(aes(intercept=best_a1,slope=best_a2))
```

Modelos e inferências

» 2.4. Ajustamento: *least-squares*



Modelos e inferências

» 2.5. Ajustamento: comparação

Modelos e inferências

» 2.5. Ajustamento: comparação

```
res1 <- models %>%
  filter(rank(RMSD) == 1)
res2 <- models_grid %>%
  filter(rank(RMSD) == 1)
res3 <- tibble(
  a1 = best_optim$par[1],
  a2 = best_optim$par[2],
  RMSD = best_optim$value)
res4 <- tibble(
  a1 = best_lm$coef[1],
  a2 = best_lm$coef[2],
  RMSD = sqrt(mean(best_lm$residuals^2)))
tab1 = round(data.frame(rbind(res1,res2,res3,res4)),digits=2)
rownames(tab1) = c("random","grid","NR","LS")
print(tab1)
```

Modelos e inferências

» 2.5. Ajustamento: comparação

```
          a1   a2 RMSD
random 2.84 4.00 1.23
grid   3.75 4.00 1.06
NR     3.40 4.02 1.03
LS     3.40 4.02 1.03
```

Modelos e inferências

» 3. Diagnóstico (`modelr`)

Modelos e inferências

» 3. Diagnóstico (`modelr`)

1. Linearidade entre variável de resposta e variáveis explicativas;
2. Não há multicolinearidade entre variáveis explicativas;
3. Resíduos têm média igual a zero;
4. Resíduos têm variância constante (homocedasticidade);
5. Resíduos não estão autocorrelacionados;
6. Resíduos são independentes das variáveis explicativas;
7. Resíduos têm distribuição Normal.

Greene, 2012, Econometric Analysis

Modelos e inferências

» 3. Diagnóstico (`modelr`)

```
library("tidyverse")
library("modelr")
set.seed(12345)
real_a1 <- 4.22
real_a2 <- 2.05
x <- round(runif(n=30,min=0,max=10),digits=2)
y <- real_a1*x + real_a2 + rnorm(n=30,mean=0,sd=1)
sim1 <- tibble(x,y)
sim1_mod <- lm(y ~ x,data=sim1)
sim1 <- sim1 %>%
  add_predictions(sim1_mod) %>%
  add_residuals(sim1_mod)
print(sim1)
```

Modelos e inferências

» 3. Diagnóstico (`modelr`)

```
# A tibble: 30 x 4
       x     y  pred  resid
   <dbl> <dbl> <dbl>  <dbl>
 1  7.21 33.3   32.3  0.945
 2  8.76 38.1   38.6 -0.441
 3  7.61 33.8   34.0 -0.122
 4  8.86 40.6   39.0  1.59
 5  4.56 21.6   21.7 -0.115
 6  1.66  9.83  10.1 -0.227
 7  3.25 17.2   16.4  0.774
 8  5.09 22.9   23.8 -0.950
 9  7.28 31.2   32.6 -1.41
10  9.9  42.2   43.1 -0.919
# ... with 20 more rows
```

Modelos e inferências

» 3. Diagnóstico (`modelr`)

```
sim1 %>% ggplot() + geom_point(aes(x,y),size=2,color="grey30") +
  geom_line(aes(x,pred)) +
  geom_segment(aes(x=x,xend=x,y=y,yend=pred),alpha=0.2)          #scatterplot y ~ x
```

Modelos e inferências

» 3. Diagnóstico (`modelr`)

```
sim1 %>% ggplot() + geom_point(aes(x,y),size=2,color="grey30") +
  geom_line(aes(x,pred)) +
  geom_segment(aes(x=x,xend=x,y=y,yend=pred),alpha=0.2)          #scatterplot y ~ x
sim1 %>% ggplot() + geom_abline(aes(intercept=0,slope=1),size=2,color="white") +
  geom_point(aes(pred,y),size=2,color="grey30")                  #scatterplot y ~ pred
```

Modelos e inferências

» 3. Diagnóstico (`modelr`)

```
sim1 %>% ggplot() + geom_point(aes(x,y),size=2,color="grey30") +
  geom_line(aes(x,pred)) +
  geom_segment(aes(x=x,xend=x,y=y,yend=pred),alpha=0.2)          #scatterplot y ~ x
sim1 %>% ggplot() + geom_abline(aes(intercept=0,slope=1),size=2,color="white") +
  geom_point(aes(pred,y),size=2,color="grey30")                  #scatterplot y ~ pred
sim1 %>% ggplot() + geom_density(aes(resid)) +
  stat_function(fun=dnorm,color="red",args=list(mean=0,sd=1))    #density(resid)
```

Modelos e inferências

» 3. Diagnóstico (`modelr`)

```
sim1 %>% ggplot() + geom_point(aes(x,y),size=2,color="grey30") +
  geom_line(aes(x,pred)) +
  geom_segment(aes(x=x,xend=x,y=y,yend=pred),alpha=0.2)          #scatterplot y ~ x
sim1 %>% ggplot() + geom_abline(aes(intercept=0,slope=1),size=2,color="white") +
  geom_point(aes(pred,y),size=2,color="grey30")                  #scatterplot y ~ pred
sim1 %>% ggplot() + geom_density(aes(resid)) +
  stat_function(fun=dnorm,color="red",args=list(mean=0,sd=1))    #density(resid)
sim1 %>% ggplot() + geom_hline(aes(yintercept=0),size=2,color="white") +
  geom_point(aes(x,resid))                                       #scatterplot res ~ x
```

Modelos e inferências

» 3. Diagnóstico (`modelr`)

```
sim1 %>% ggplot() + geom_point(aes(x,y),size=2,color="grey30") +
  geom_line(aes(x,pred)) +
  geom_segment(aes(x=x,xend=x,y=y,yend=pred),alpha=0.2)          #scatterplot y ~ x
sim1 %>% ggplot() + geom_abline(aes(intercept=0,slope=1),size=2,color="white") +
  geom_point(aes(pred,y),size=2,color="grey30")                  #scatterplot y ~ pred
sim1 %>% ggplot() + geom_density(aes(resid)) +
  stat_function(fun=dnorm,color="red",args=list(mean=0,sd=1))    #density(resid)
sim1 %>% ggplot() + geom_hline(aes(yintercept=0),size=2,color="white") +
  geom_point(aes(x,resid))                                       #scatterplot res ~ x
sim1 %>% ggplot() + geom_qq_line(aes(sample=resid),size=2,color="white") +
  geom_qq(aes(sample=resid)) +
  labs(x="Theoretical Quantiles",y="Sample Quantiles")           #qqplot(resid)
```

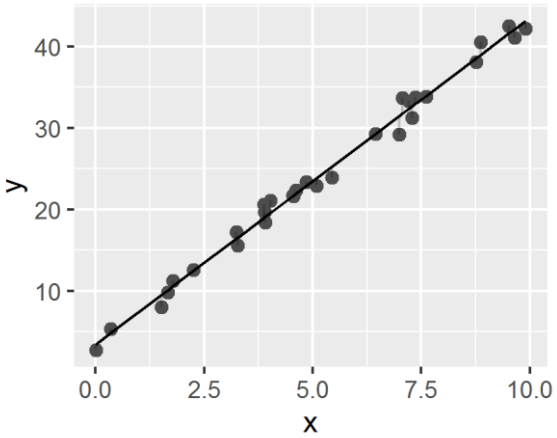
Modelos e inferências

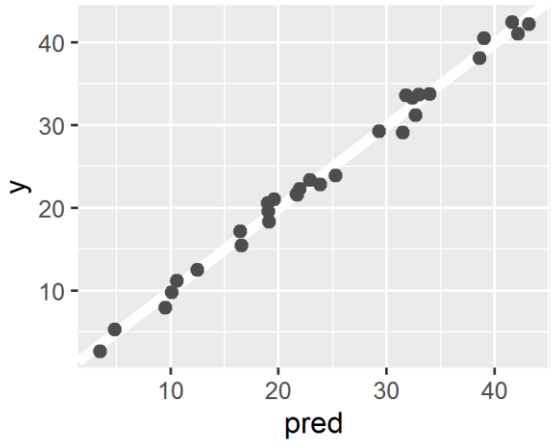
» 3. Diagnóstico (`modelr`)

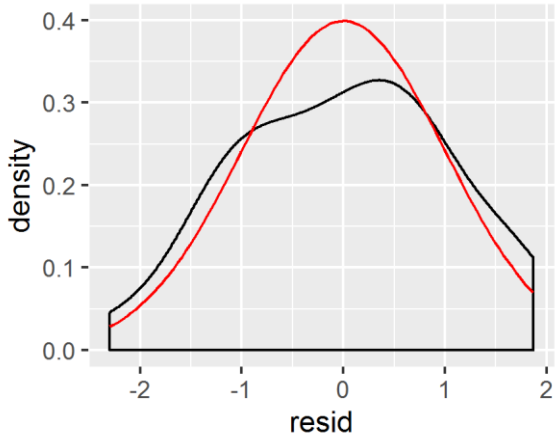
```
sim1 %>% ggplot() + geom_point(aes(x,y),size=2,color="grey30") +
  geom_line(aes(x,pred)) +
  geom_segment(aes(x=x,xend=x,y=y,yend=pred),alpha=0.2)          #scatterplot y ~ x
sim1 %>% ggplot() + geom_abline(aes(intercept=0,slope=1),size=2,color="white") +
  geom_point(aes(pred,y),size=2,color="grey30")                  #scatterplot y ~ pred
sim1 %>% ggplot() + geom_density(aes(resid)) +
  stat_function(fun=dnorm,color="red",args=list(mean=0,sd=1))    #density(resid)
sim1 %>% ggplot() + geom_hline(aes(yintercept=0),size=2,color="white") +
  geom_point(aes(x,resid))                                       #scatterplot res ~ x
sim1 %>% ggplot() + geom_qq_line(aes(sample=resid),size=2,color="white") +
  geom_qq(aes(sample=resid)) +
  labs(x="Theoretical Quantiles",y="Sample Quantiles")           #qqplot(resid)
sim1 %>% ggplot() + geom_point(aes(resid,c(resid[-1],NA))) +
  labs(x=expression(resid[i-1]),y=expression(resid[i]))          #lagplot(resid)
```

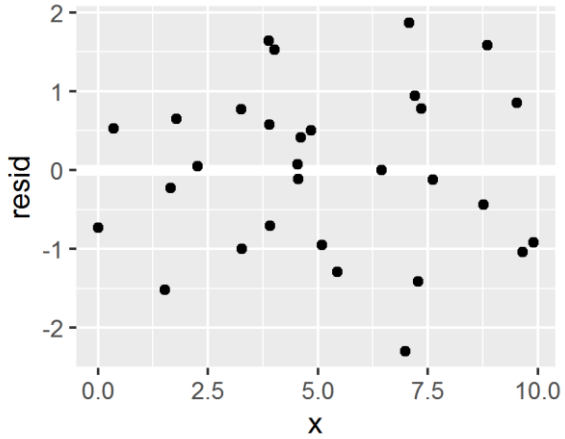
Modelos e inferências

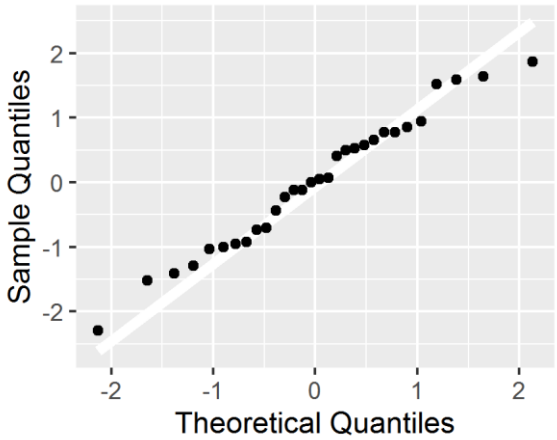
» 3. Diagnóstico (`modelr`)

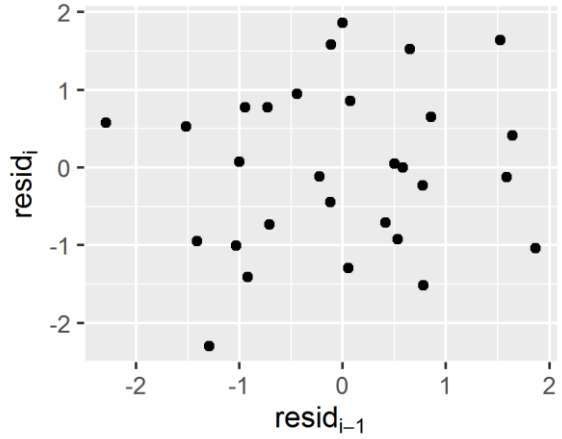












Gestão de projectos

» Boas práticas

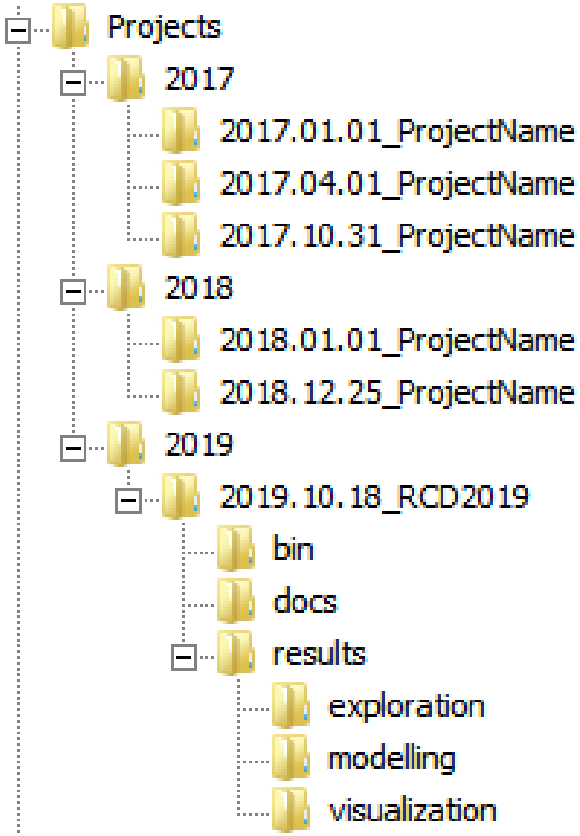
Gestão de projectos

» Boas práticas

- Estrutura de pastas;
- Ficheiro README (e nomeação de ficheiros).

Gestão de projectos

» 1. Estrutura de pastas



Gestão de projectos

» 2. Ficheiro README

```
#autor:      Joao Sollari Lopes
#local:      INE, Lisboa
#criado:     18.10.2019
#modificado: 18.10.2019
+bin
  |exploration.r                          #exploracao de dados
  |install_packages.r                     #instalar pacotes necessarios
  |manipulation.r                         #manipulacao de dados
  |modelling.r                            #modelação de dados
  |visualization.r                        #visualizacao de dados
+docs
  |RCD2019_programa.pdf                   #programa
  |RCD2019_slides.pdf                     #slides [versao final]
  |RCD2019_slides_short.pdf               #slides principais [versao final]
  |RCD2019_slides_20191014.pptx           #slides [v2019-10-14]
  |RCD2019_slides_short_20191014.pptx     #slides principais [v2019-10-14]
+results
  +exploration                           #resultados de "exploration.r"
  +modelling                             #resultados de "modelling.r"
  +visualization                         #resultados de "visualization.r"
README.txt                               #Este ficheiro
```

Comunidade R

» help!

Comunidade R

» help!

- https://www.r-project.org/
- https://www.r-project.org/foundation/

- https://www.r-project.org/mail.html
- https://stackoverflow.com/
- https://www.r-bloggers.com/
- https://community.rstudio.com/
- comunidade R no INE

Bibliografia

» Azzalini A & Scarpa B (2012) Data Analysis and Data Mining - An Introduction. Oxford University Press, New York.

» Due KL & Swamy MNS (2016) Search and Optimization by Metaheuristics - Techniques and Algorithms Inspired by Nature. Springer, Switzerland.

» Gama J, Carvalho APL, Faceli K, Lorena AC e Oliveira M (2012) Extração de Conhecimento de Dados. *Data Mining*. Edições Sílabo. Lisboa.

» Larose DT & Larose CD (2014) Discovering Knowledge in Data – An Introduction to Data Mining. John Wiley & Sons, New Jersey.

» Torgo L, (2017) Data Mining with R – Learning with Case Studies. Taylor & Francis Group, New York.

» Wickham H & Grolemund G (2017) R for Data Science. O'Reilly Media Inc., Sevastopol.