

Análise de Dados - UFPE/2019 - Lista 6

Ana Carolina Costa

20/05/2019

LINK PARA O GITHUB:

https://github.com/scostacarolina/lista_6_ana_carolina_costa.git

EXERCÍCIO 01

Descreva os conceitos abaixo:

a) Variável dependente Variáveis dependentes são apenas medidas ou registradas, em um experimento são as variáveis que esperamos ariáveis sejam “dependentes” da manipulação ou das condições experimentais. Ou seja, elas dependem “do que os sujeitos”, ou as variáveis independentes farão em resposta.

b) Variável independente Variável independente é a causa, o antecedente, e a origem de um fenômeno, um processo que constitui o objeto de estudo. São as variáveis que num experimento são manipuladas, ou seja, são “independentes” dos padrões de reação inicial.

c) Apresente qual a relação existente entre variáveis independentes e dependente:

Espera-se que outras variáveis sejam “dependentes” da manipulação ou das condições experimentais. Ou seja, elas dependem “do que os sujeitos farão” em resposta. Há uma relação de causalidade entre os dois tipos de variáveis.

EXERCÍCIO 02

Em análise de dados, qual o nome dado à equação mostrada na lista 06?

Trata-se de um modelo de regressão bivariado.

- Y^1 -> trata-se do valor observado da variável dependente
- α -> trata-se do parâmetro do intercepto estimado
- β -> trata-se do coeficiente de variação
- X_i -> trata-se do valor observado da variável independente
- ϵ -> trata-se do componente estocástico ou erro amostral

EXERCÍCIO 03

Com suas palavras, apresente uma definição para cada um dos componentes da equação apresentada no exercício 2.

- Y^1 -> é a Variável explicada (a variável dependente), representa o que o modelo tentará prever.
- α -> É uma constante, representa a interceptação da reta com o eixo vertical.
- β -> Representa a inclinação ou o coeficiente angular em relação à variável independente.
- X_i -> é a variável explicativa, é o que dá sentido a equação, sem ela não há efeitos sobre a variável que queremos explicar.
- ϵ -> Representa todos os resíduos da equação mais os possíveis erros de medição, representa todos aqueles componentes que não conseguimos explicar numa relação causal. Apresenta um comportamento aleatório.

EXERCÍCIO 04

Apresente o componente sistemático da equação apresentada no exercício 2. Descreva por quê é sistemático.

O componente sistemático é a parte da equação composta por Y^1 , alfa, beta e X_i , pode-se dizer que é a parte do experimento que o cientista tem controle sobre. Aqui as relações entre as variáveis conseguem ser definidas e explicadas. É onde de fato conseguimos observar a relação de causalidades entre as variáveis.

EXERCÍCIO 05

Apresente o componente estocástico da equação apresentada no exercício 2. Descreva por quê é estocástico.

O componente estocástico é a parte da equação composta por m_i , representa a aleatoriedade do comportamento humano que a ciência não consegue explicar, também pode representar o resíduo a realidade social e a mensuração da realidade social, de toda forma, representa a parte do modelo ou experimento que não está passível de controle.

EXERCÍCIO 06

Descreva a diferença entre Y^1 e \hat{Y}^1 . Qual a relação desses dois componentes com m_i ?

Como já dito Y^1 é a variável dependente explicada, enquanto \hat{Y}^1 é o valor predition de Y^1 . A presença do chapéu (^) significa que os valores dessas variáveis foram retirados de amostras populacionais, enquanto que sem o chapéu significa que o valor foi tirado de um dado populacional completo. A presença do chapéu é mais comum, pois dificilmente cientistas sociais conseguem trabalhar com o todo populacional. Quando há a presença de valores amostrais, comumente trabalhamos com um risco maior de resíduo amostral (m_i).

EXERCÍCIO 07

Com suas palavras, apresente o que é o modelo OLS e seu principal uso na análise de dados.

O modelo Ordinary Least Squares ou Mínimos Quadrados Ordinários é uma técnica de otimização matemática que procura encontrar o melhor ajuste para uma regressão tentando minimizar a soma dos quadrados das resíduos amostrais (ou populacionais) entre o valor estimado e os dados observados. O modelo OLS fornece estimativas mais precisas e não viciadas.

EXERCÍCIO 08

Com base no Google's R Style Guide, apresente exemplos de boas práticas para os seguintes tópicos:

a) nomes de arquivos:

Os nomes dos arquivos devem terminar em ".r" e devem ser práticos e concisos para você saber de cara a função desse arquivo.

Um bom exemplo: "banco_pnud_final_filtro_pe.r"

Um péssimo exemplo: "banco_finalagoravai.r"

b) Identificadores:

Tente não utilizar o underline "_" ou hífen "-" ao nomear suas variáveis. De preferência utilize apenas letras minúsculas. Tente ser conciso.

um bom exemplo: "matriculas.pe"

péssimo exemplo: "Matriculas_PE-PNUD"

c) identificação:

Ao recuar seu código, use dois espaços. Nunca use tabulações ou misture tabulações e espaços.

Um bom exemplo:

```

turmas_pe_sel <- turmas_pe %>% group_by(CO_MUNICIPIO) %>%
summarise(n_turmas = n(),
  turmas_disc_prof = sum(IN_DISC_PROFISSIONALIZANTE, na.rm = T),
  turmas_disc_inf = sum(IN_DISC_INFORMATICA_COMPUTACAO, na.rm = T),
  turmas_disc_mat = sum(IN_DISC_MATEMATICA, na.rm = T),
  turmas_disc_pt = sum(IN_DISC_LINGUA_PORTUGUESA, na.rm = T),
  turmas_disc_en = sum(IN_DISC_LINGUA_INGLES, na.rm = T))

```

d) espaçamento:

Coloque espaços ao redor de todos os operadores binários (=, +, -, <-, etc.). Exceção: Espaços ao redor de “=” são opcionais ao passar parâmetros em uma chamada de função. Não coloque um espaço antes de uma vírgula, mas sempre coloque um espaço após uma vírgula.

Um bom exemplo:

```
pnuad.pe.2010 <- pnuad %>% filter(ANO == 2010 & UF == 26)
```

Um péssimo exemplo:

```
pnuad_pe_2010<-pnuad %>% filter(ANO==2010 & UF==26)
```

e) Designação:

Use “<-” para designar uma função, variável ou rótulo, evite ao máximo usar “=”

Um bom exemplo:

```
albuns.ladygaga <- 5
```

Um péssimo exemplo:

```
Albuns_Lady_Gaga = 5
```

f) Comentando Diretrizes:

Ao comentar seu código utilize um “#” e um espaço para grandes comentários e utilize “##” após um código para fazer um comentário curto.

Um bom exemplo:

```

# aqui iremos inserir um grande comentário, porém as linhas não devem ser
# maiores que 80 caracteres. então iremos continuar fazendo a quebra utilizando
# apenas um "#" em cada linha até terminar o comentário

albuns.ladygaga <- 5 ## aqui comentamos rapidamente

```

um péssimo exemplo:

```

## a forma como vinha comentando o script até agora, utilizando dois "#" em cada
## linha sempre. Não há necessidade. Perdemos muito espaço fazendo assim.
## Faça o que eu digo mas não faça o que eu faço.

```

g) Definições de funções e chamadas:

As definições de função devem primeiro listar os argumentos sem valores padrão, seguidos daqueles com valores padrão.

h) documentação da função:

É interessante que haja comentários logo após o comando da função, explicando o que acabou de acontecer, os comentários devem ser concisos e explicativos.

EXERCÍCIO 09

Leia o Capítulo 7 do livro R para Ciência de Dados e entregue script no R que reproduza os exemplos apresentados no capítulo. Comente seu código indicando o que está para ser realizado em cada etapa do seu script.

Exemplos CAP 7.3.1

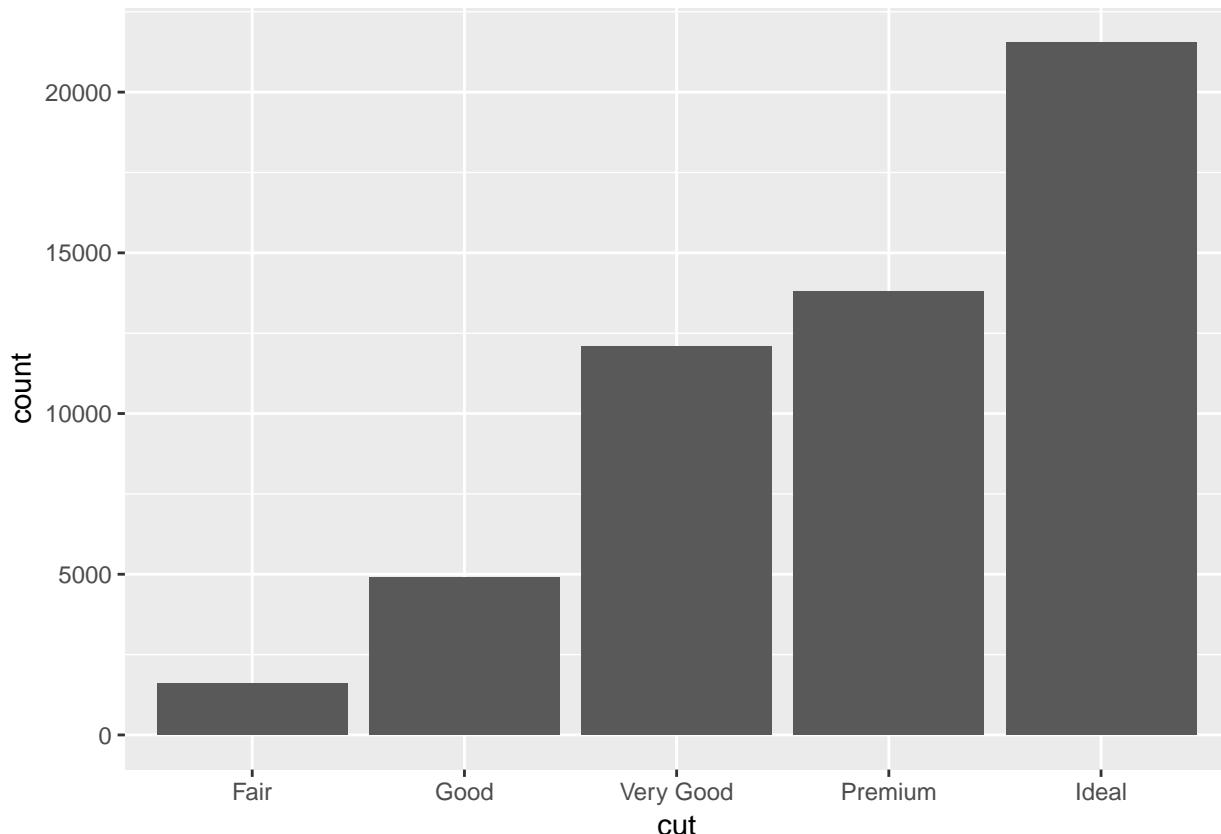
```
# CONHECER BASE DE DADOS
```

```
view(diamonds)
```

VISUALIZANDO DISTRIBUIÇÕES CATEGÓRICAS

```
# SOLICITANDO GRAFICO QUE AJUDA A VISUALIZAR DISTRIBUICOES POR CORTES DE DIAMANTES
```

```
ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut))
```



```
diamonds %>%
```

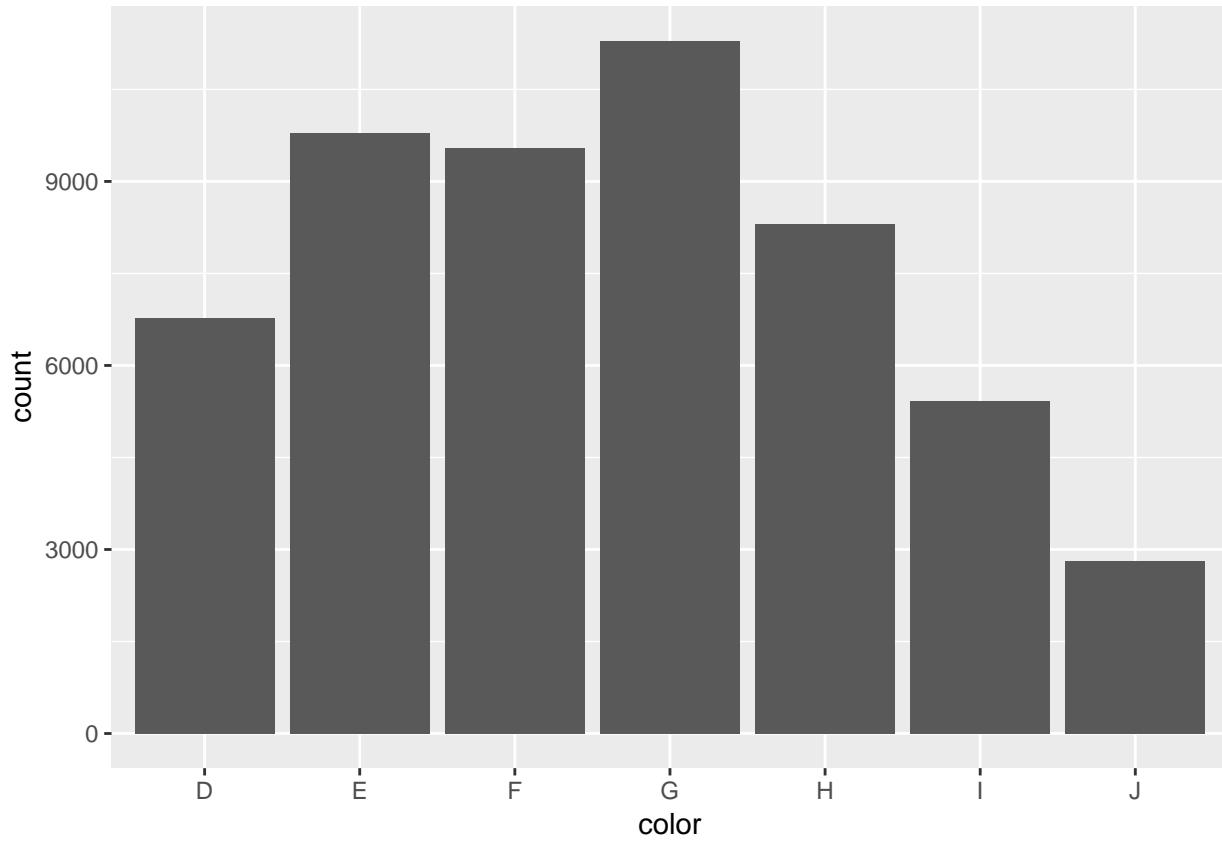
```
  count(cut) ## solicitando os dados manualmente com a funcao dplyr::count()
```

```

## # A tibble: 5 x 2
##   cut      n
##   <ord>    <int>
## 1 Fair     1610
## 2 Good     4906
## 3 Very Good 12082
## 4 Premium   13791
## 5 Ideal    21551

# DISTRIBUICAO POR COR DE DIAMANTES
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = color))

```



```

diamonds %>%
  count(color) ## ## solicitando os dados manualmente com a funcao dplyr::count()

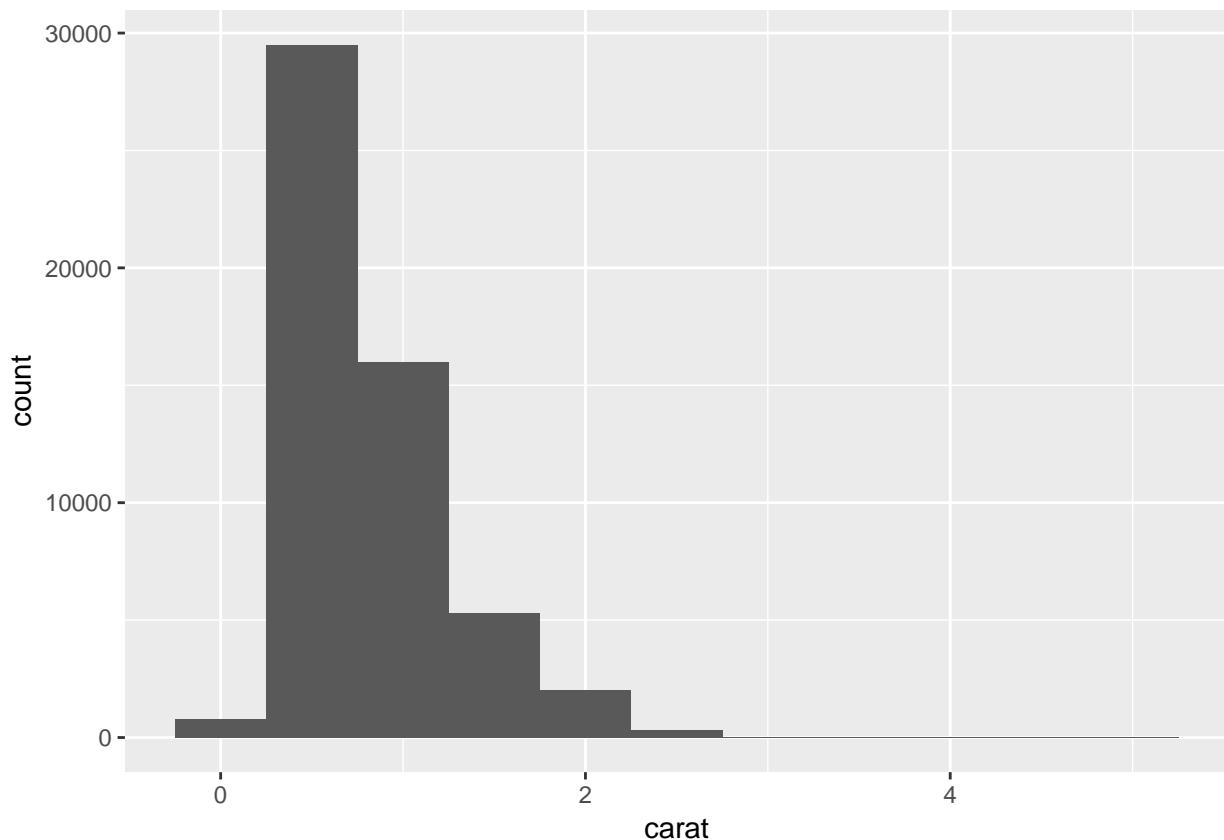
## # A tibble: 7 x 2
##   color      n
##   <ord>    <int>
## 1 D        6775
## 2 E        9797
## 3 F        9542
## 4 G       11292
## 5 H        8304
## 6 I        5422
## 7 J        2808

```

VISUALIZANDO DISTRIBUIÇÕES CONTÍNUAS

Distribuição de diamantes por quilates

```
ggplot(data = diamonds) +  
  geom_histogram(mapping = aes(x = carat), binwidth = 0.5)
```



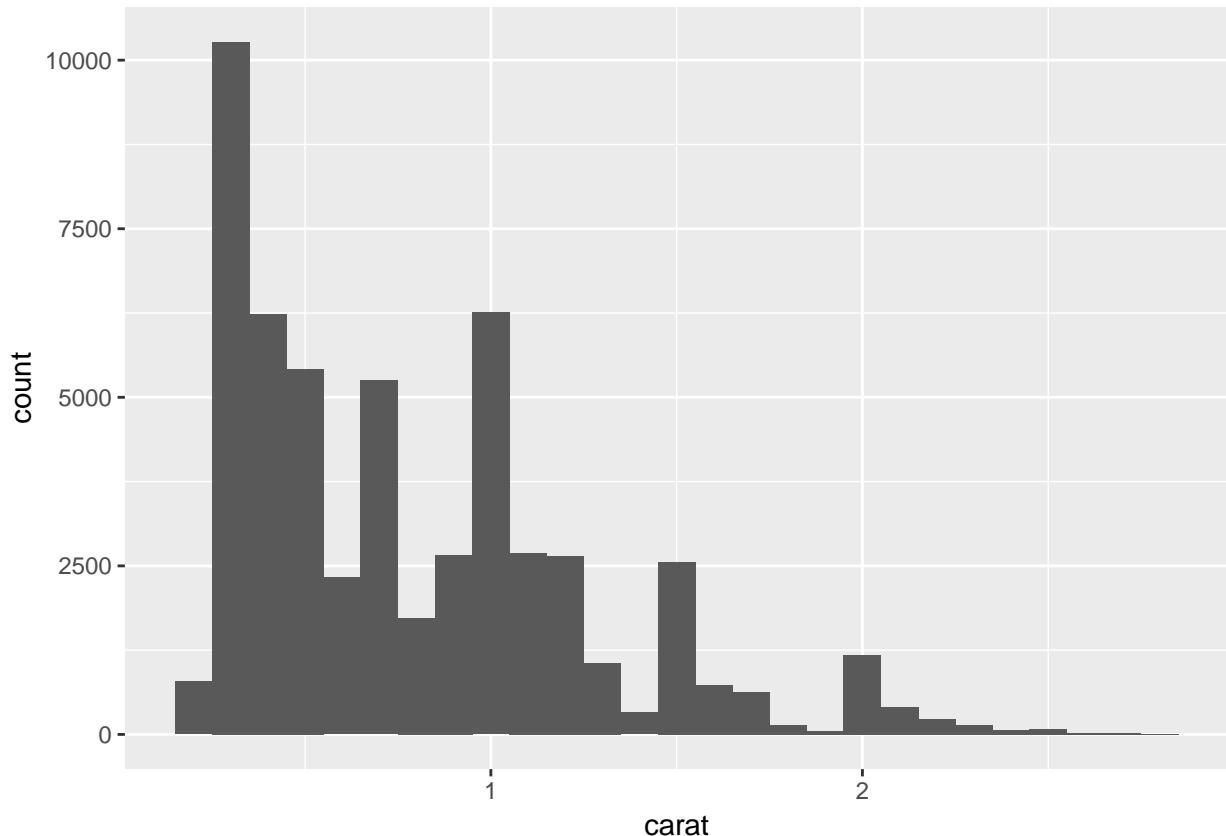
```
diamonds %>%  
  count(cut_width(carat, 0.5)) ## solicitando manualmente
```

```
## # A tibble: 11 x 2  
##   `cut_width(carat, 0.5)`     n  
##   <fct>                      <int>  
## 1 [-0.25,0.25]                785  
## 2 (0.25,0.75]                29498  
## 3 (0.75,1.25]                15977  
## 4 (1.25,1.75]                5313  
## 5 (1.75,2.25]                2002  
## 6 (2.25,2.75]                322  
## 7 (2.75,3.25]                32  
## 8 (3.25,3.75]                5  
## 9 (3.75,4.25]                4  
## 10 (4.25,4.75]               1  
## 11 (4.75,5.25]               1
```

```
# filtrando diamantes menores de 3 quilates

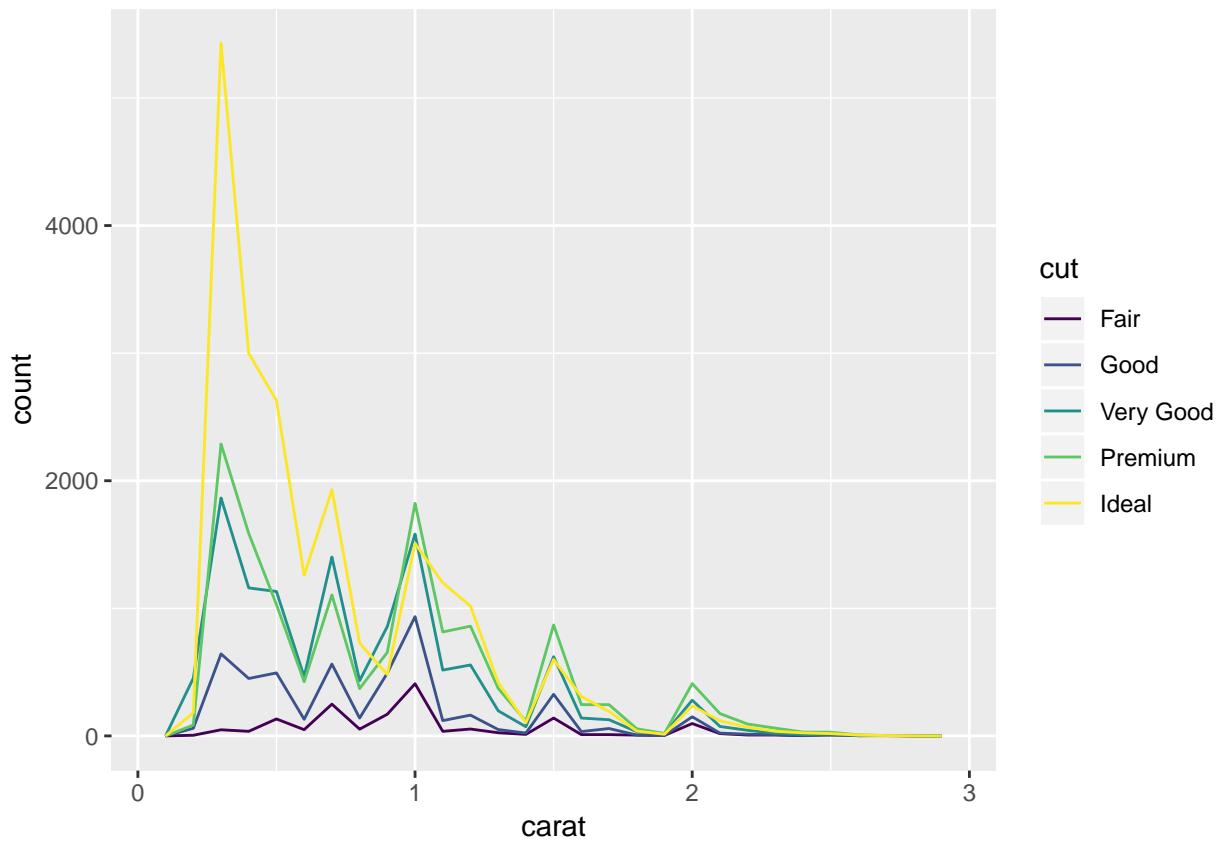
smaller <- diamonds %>%
  filter(carat < 3) ## criando nova base de dados

ggplot(data = smaller, mapping = aes(x = carat)) +
  geom_histogram(binwidth = 0.1) ## solicitando novo gráfico
```



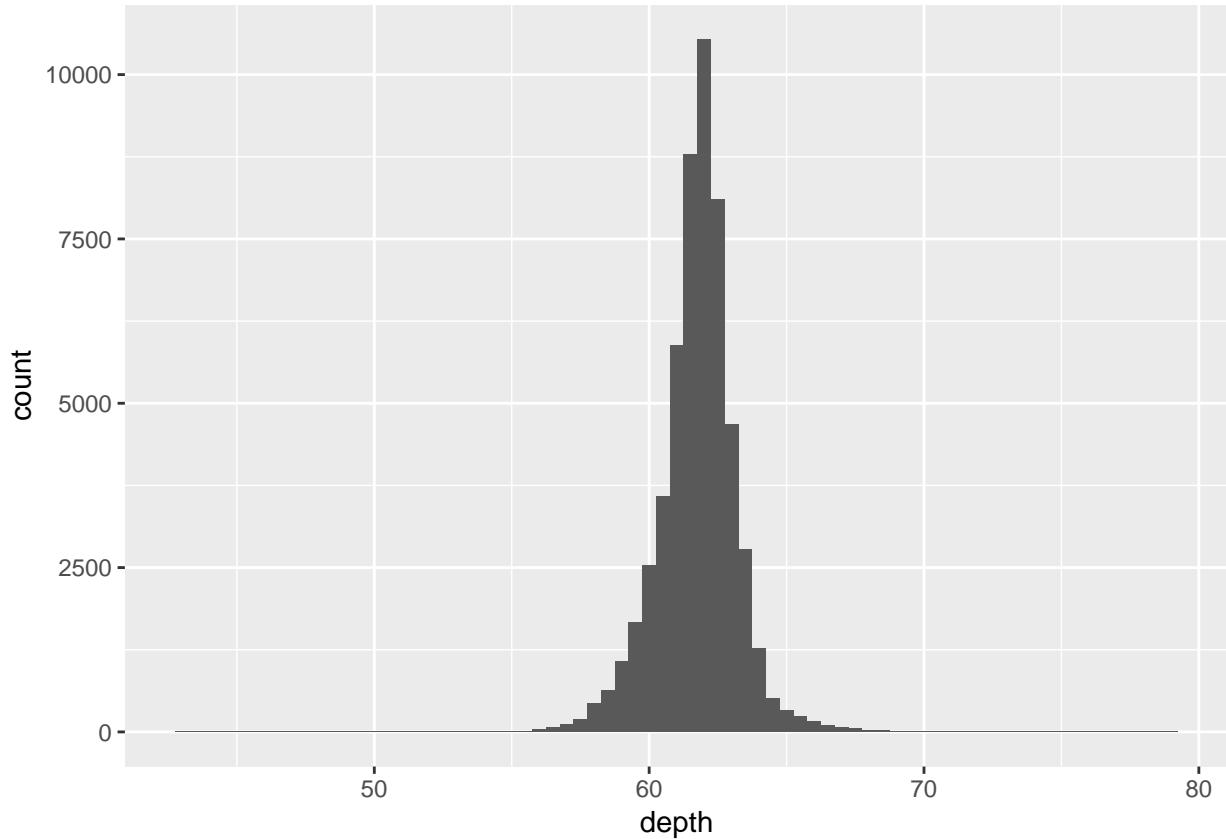
```
# Solicitando multiplos histogramas em um grafico
# Comparacao entre quilates e cortes de diamantes
# utilizamos nova funcao geom_freqpoly()

ggplot(data = smaller, mapping = aes(x = carat, colour = cut)) +
  geom_freqpoly(binwidth = 0.1)
```



Distribuição de diamantes por Profundidade

```
ggplot(data = diamonds) +  
  geom_histogram(mapping = aes(x = depth), binwidth = 0.5)
```



```
diamonds %>%
  count(cut_width(depth, 2)) ## solicitando manualmente
```

```
## # A tibble: 15 x 2
##   `cut_width(depth, 2)`     n
##   <fct>                  <int>
## 1 [43,45]                   3
## 2 (49,51]                   2
## 3 (51,53]                   4
## 4 (53,55]                  15
## 5 (55,57]                 187
## 6 (57,59]                1856
## 7 (59,61]               11498
## 8 (61,63]                33293
## 9 (63,65]                6247
## 10 (65,67]                 666
## 11 (67,69]                 127
## 12 (69,71]                  31
## 13 (71,73]                   7
## 14 (73,75]                   1
## 15 (77,79]                   3
```

```
# filtrando diamantes com profundidade menor de 60.
```

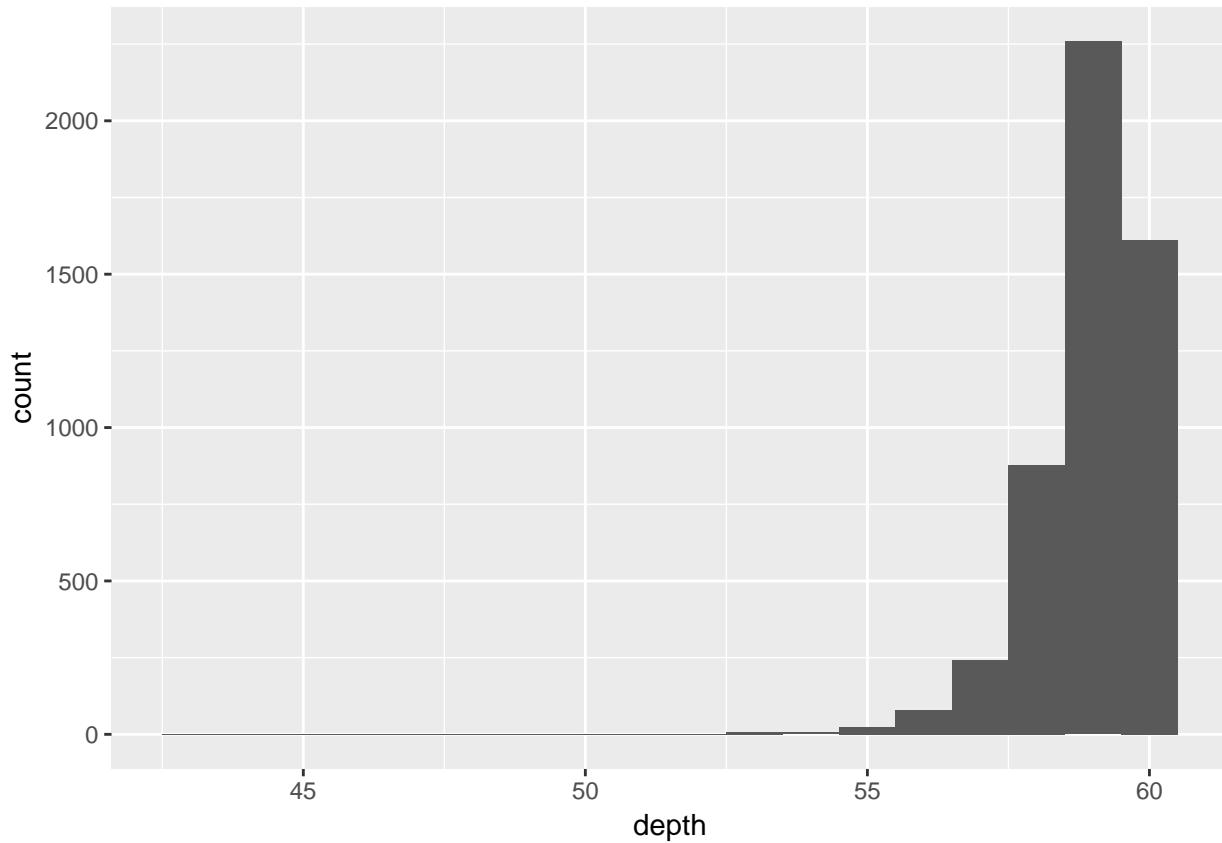
```
menor.profundidade <- diamonds %>%
```

```

filter(depth < 60) ## criando nova base de dados

ggplot(data = menor.profundidade, mapping = aes(x = depth)) +
  geom_histogram(binwidth = 1) ## solicitando novo gráfico

```

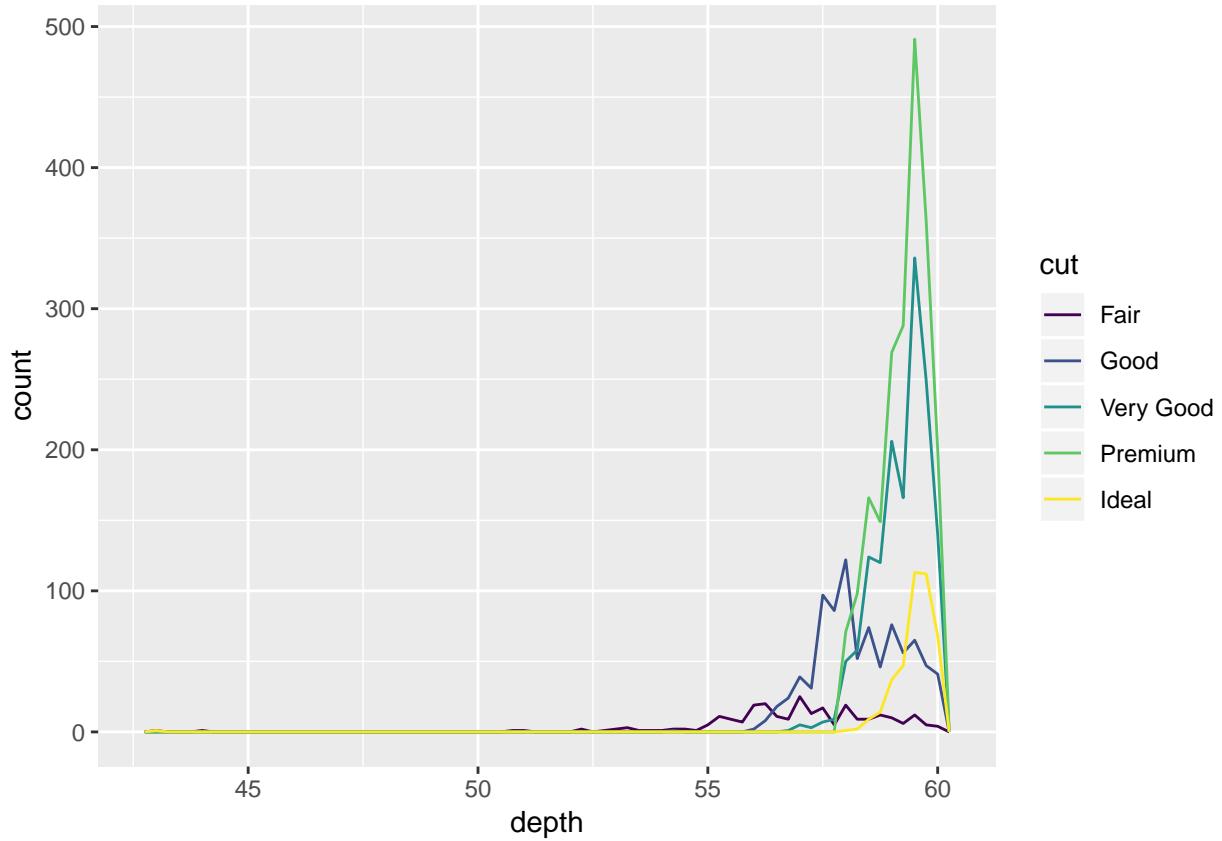


```

# Solicitando multiplos histogramas em um grafico
# Comparacao entre profundidade e cortes de diamantes
# utilizamos nova funcao geom_freqpoly()

ggplot(data = menor.profundidade, mapping = aes(x = depth, colour = cut)) +
  geom_freqpoly(binwidth = 0.25)

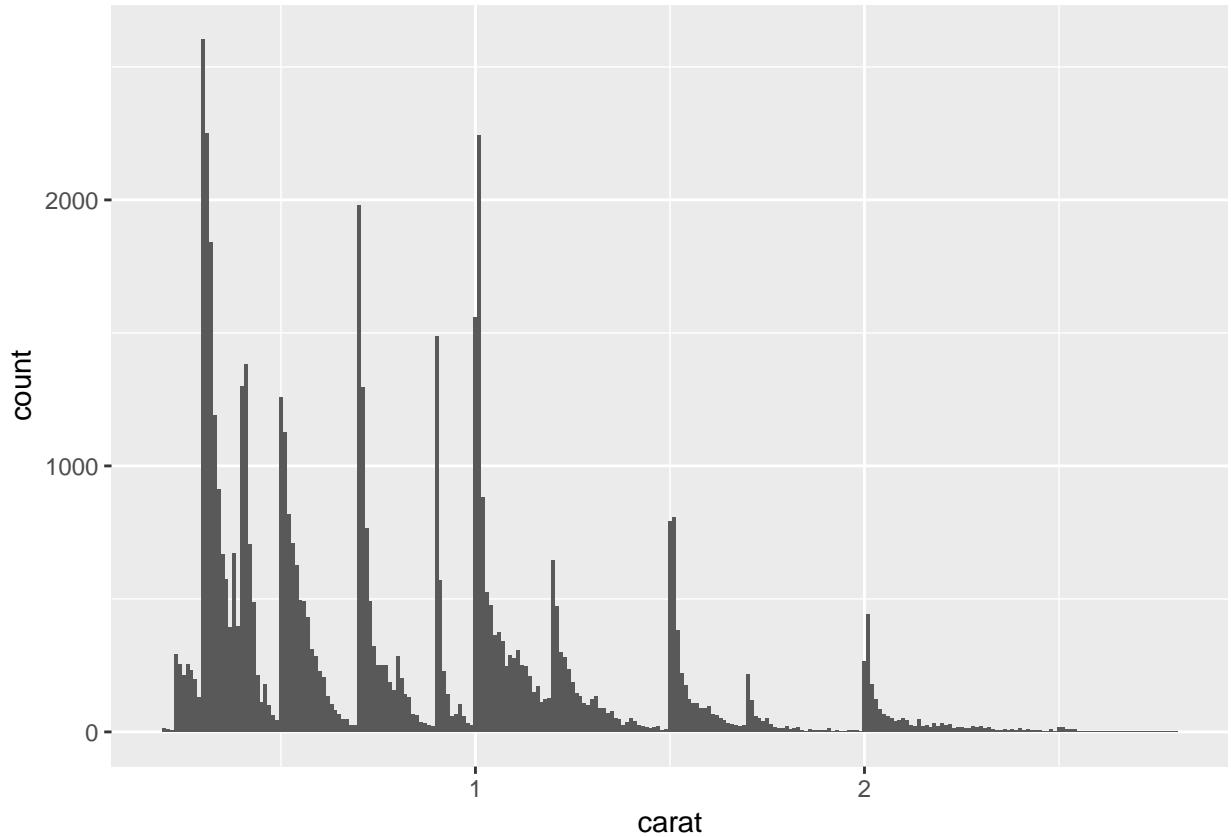
```



Exemplos CAP 7.3.2 Distribuição de diamantes por quilates

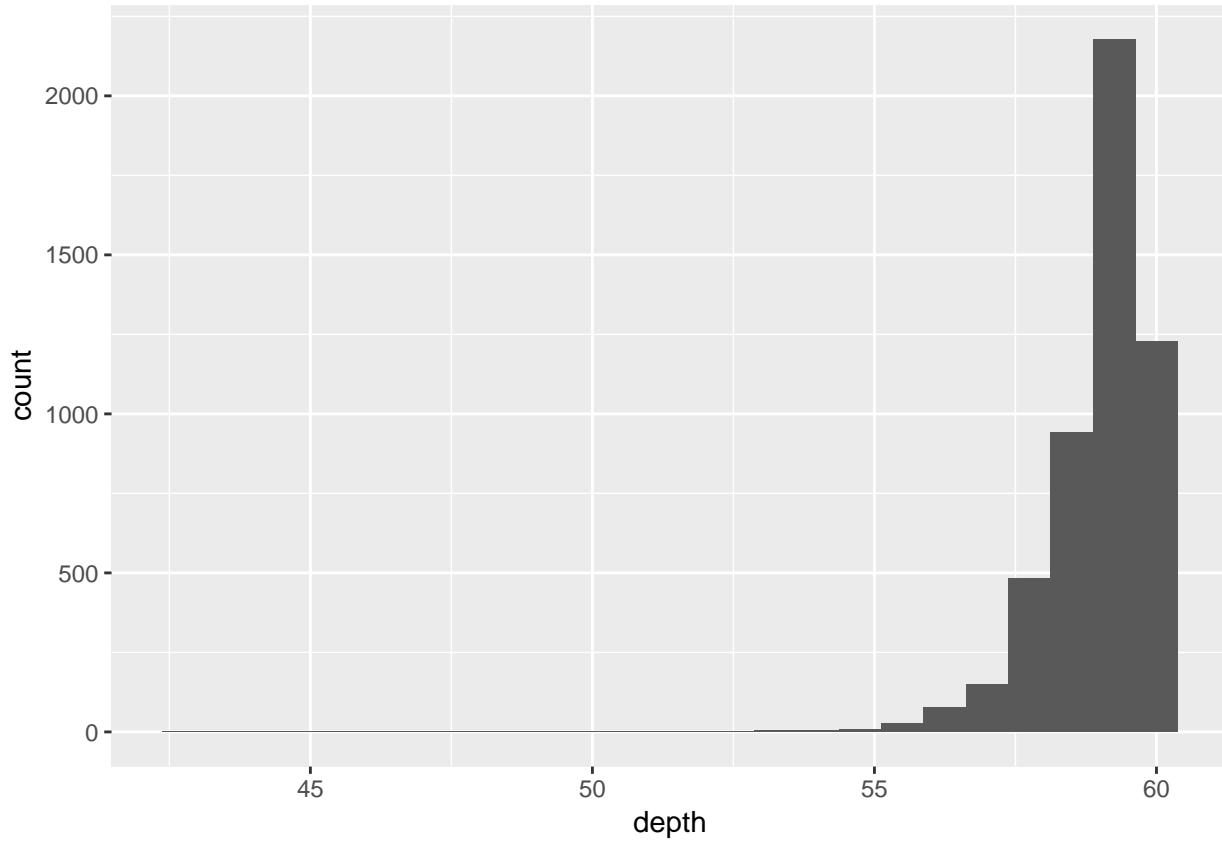
Observando pico na distribuição de diamantes por quilates

```
ggplot(data = smaller, mapping = aes(x = carat)) +  
  geom_histogram(binwidth = 0.01)
```



Distribuição de diamantes por Profundidade Observando pico na distribuição de diamantes por profundidade

```
ggplot(data = menor.profundidade, mapping = aes(x = depth)) +  
  geom_histogram(binwidth = 0.75)
```



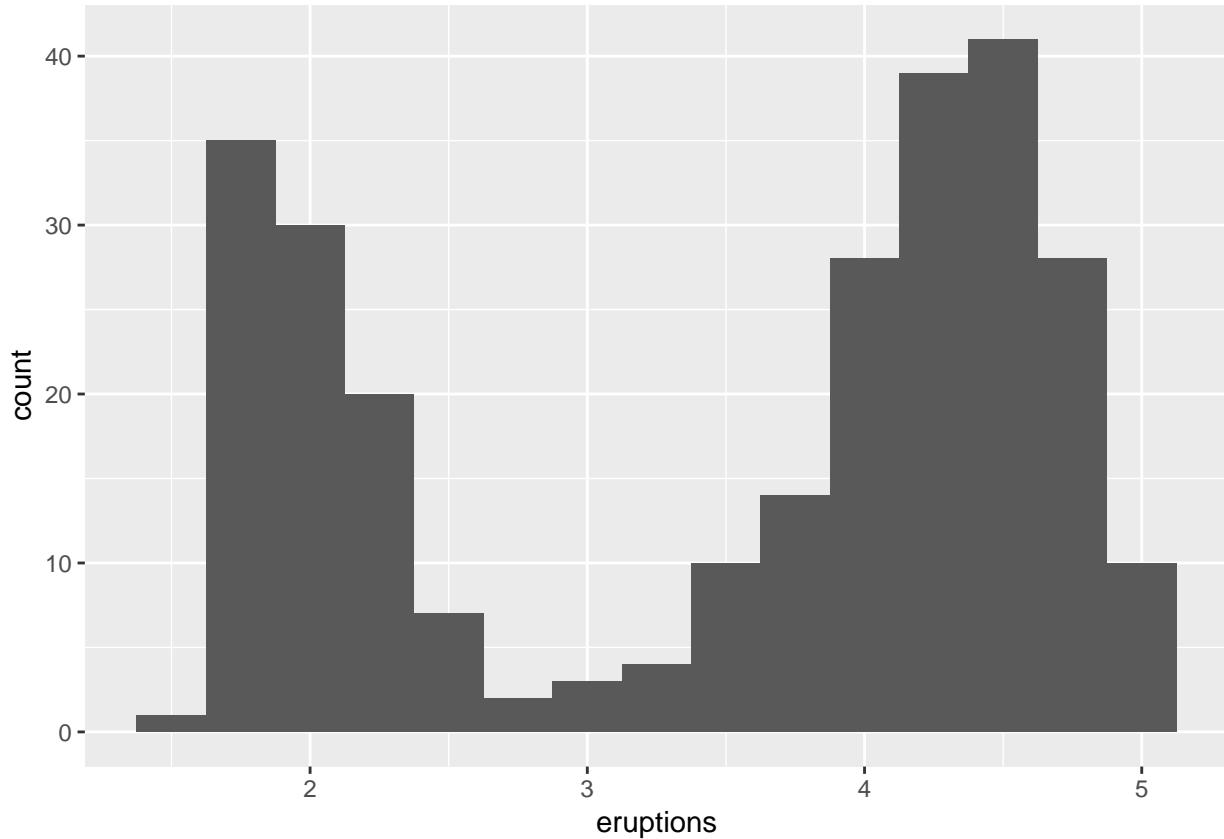
OBSERVANDO FORMAÇÃO DE CONGLOMERADOS EM NOVA BASE DE DADOS

```
#Conhecendo nova base de dados
view(faithful)
dim(faithful)

## [1] 272    2

# Solicitar gráfico para observar formacao de conglomerados

ggplot(data = faithful, mapping = aes(x = eruptions)) +
  geom_histogram(binwidth = 0.25)
```



Exemplos CAP 7.3.3

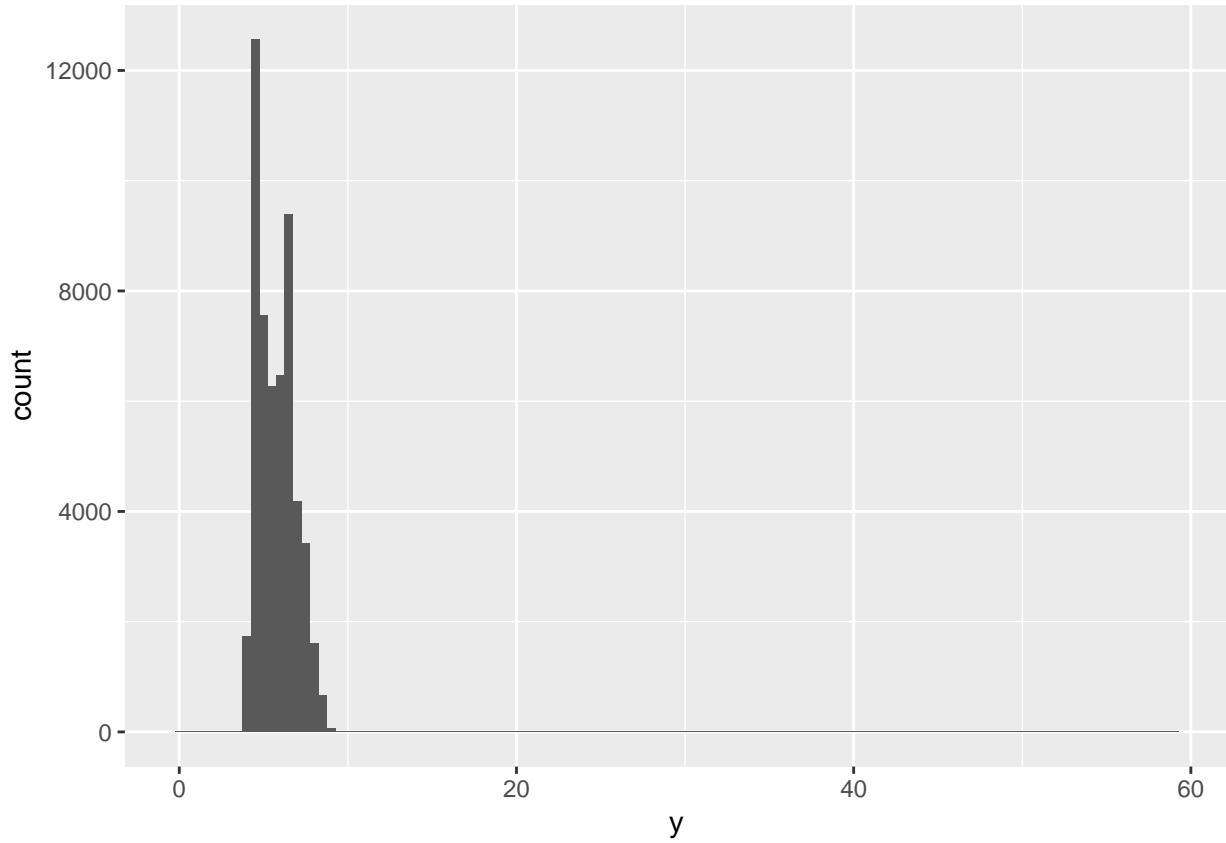
OBSERVANDO OUTLIERS

```
# retornando a base sobre diamantes  
dim(diamonds)
```

```
## [1] 53940    10
```

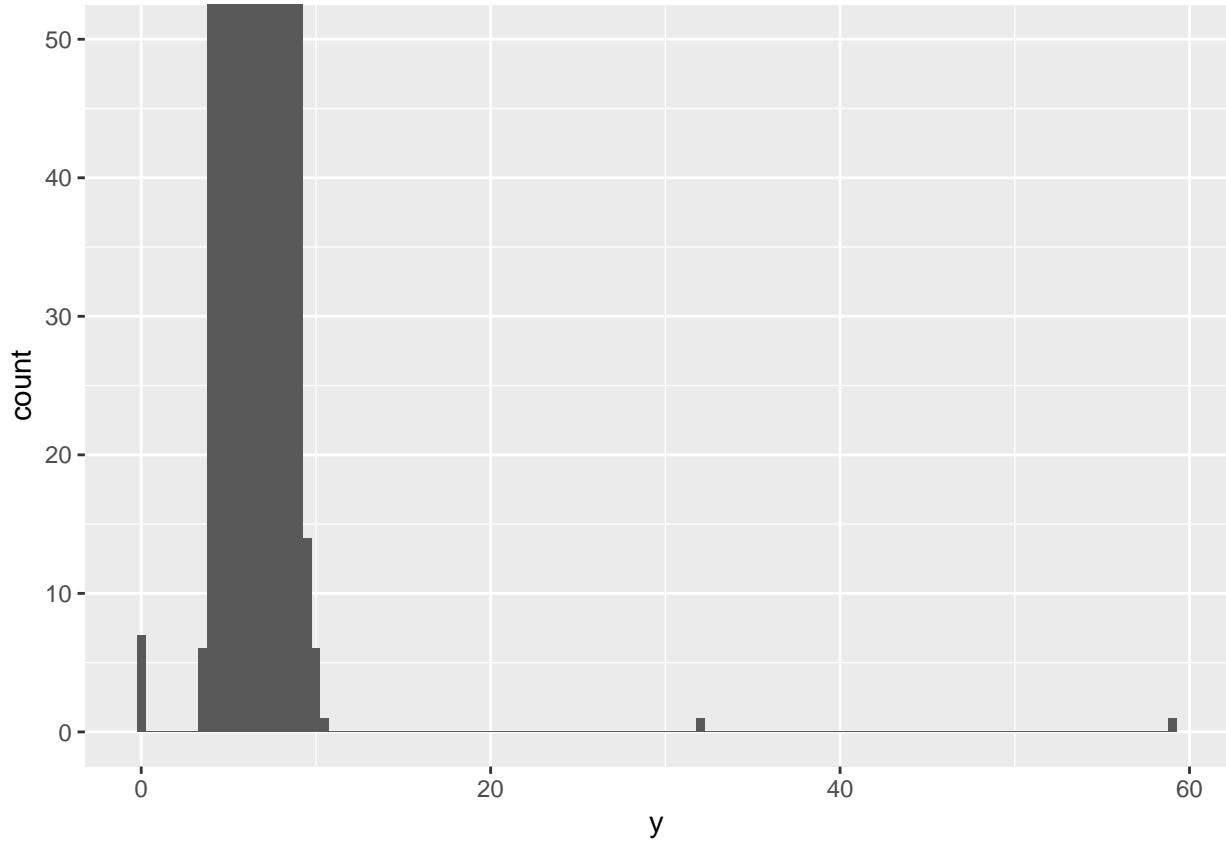
Solicitando gráfico de visão geral

```
ggplot(diamonds) +  
  geom_histogram(mapping = aes(x = y), binwidth = 0.5)
```



Utilizar função de zoom que facilita a observação de outliers

```
ggplot(diamonds) +  
  geom_histogram(mapping = aes(x = y), binwidth = 0.5) +  
  coord_cartesian(ylim = c(0, 50)) ## essa linha representa a nova função
```



```
# Usando o dplyr para visualizar e juntar os outliers
```

```
unusual <- diamonds %>%
  filter(y < 3 | y > 20) %>%
  select(price, x, y, z) %>%
  arrange(y)
```

Exemplos CAP 7.4

Quando encontramos outliers em nossos dados podemos agir para remove-los da nossa base. Existem duas maneiras:

- 1. remover as linhas completamente, da forma abaixo:

```
diamonds2.1 <- diamonds %>%
  filter(between(y, 3, 20))
```

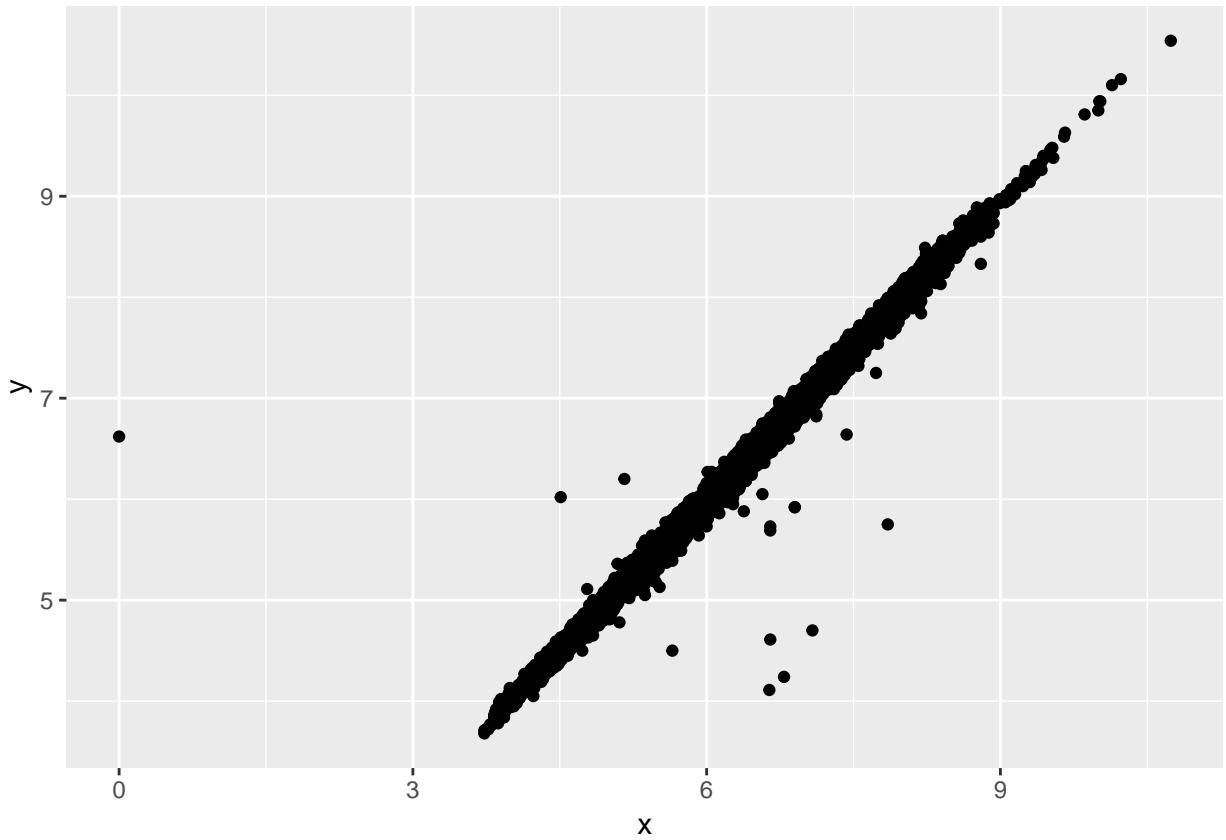
- 2. Subistituir os outlier por espacos sem informação:

```
diamonds2.2 <- diamonds %>%
  mutate(y = ifelse(y < 3 | y > 20, NA, y))
```

Se fizermos da primeira maneira, acabamos por perder muitas informações

Observe o grafico do exemplo 1

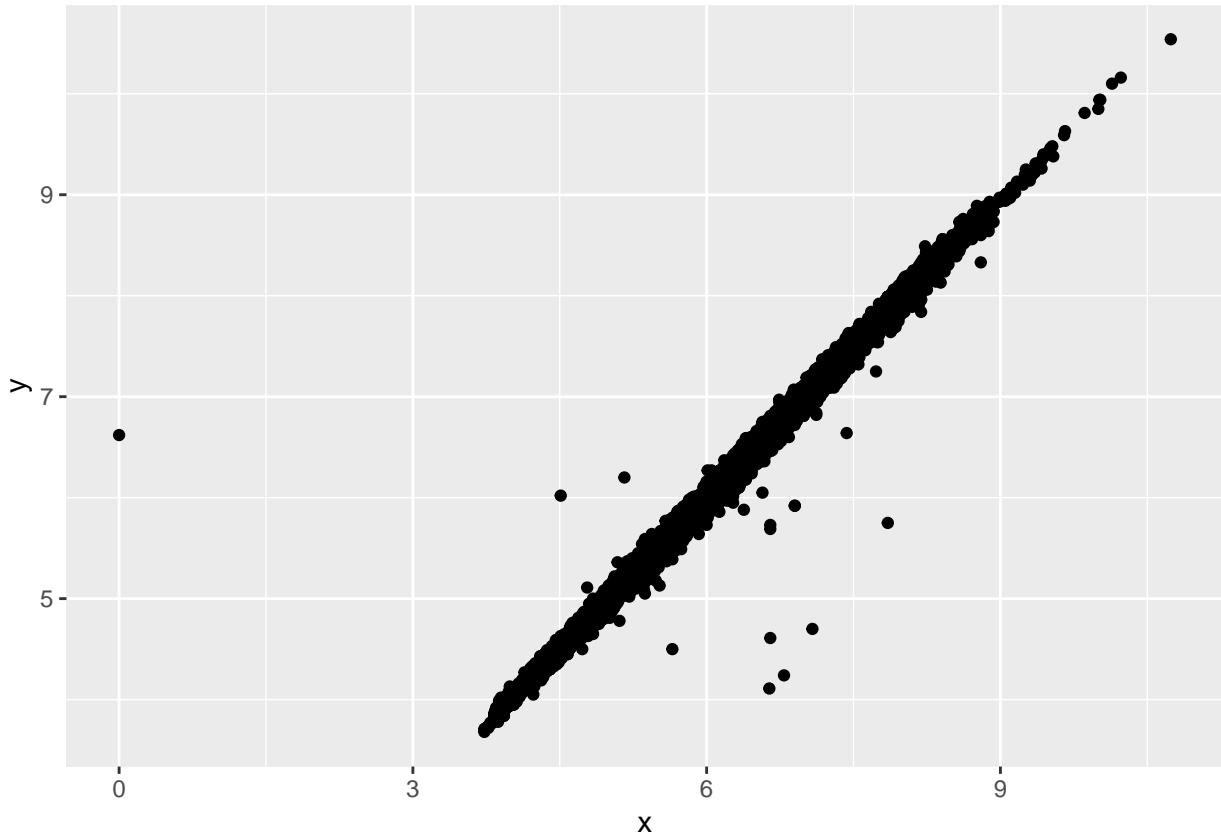
```
ggplot(data = diamonds2.1, mapping = aes(x = x, y = y)) +  
  geom_point()
```



```
# Perdemos 9 linhas completas de dados
```

Observe o grafico do exemplo 2

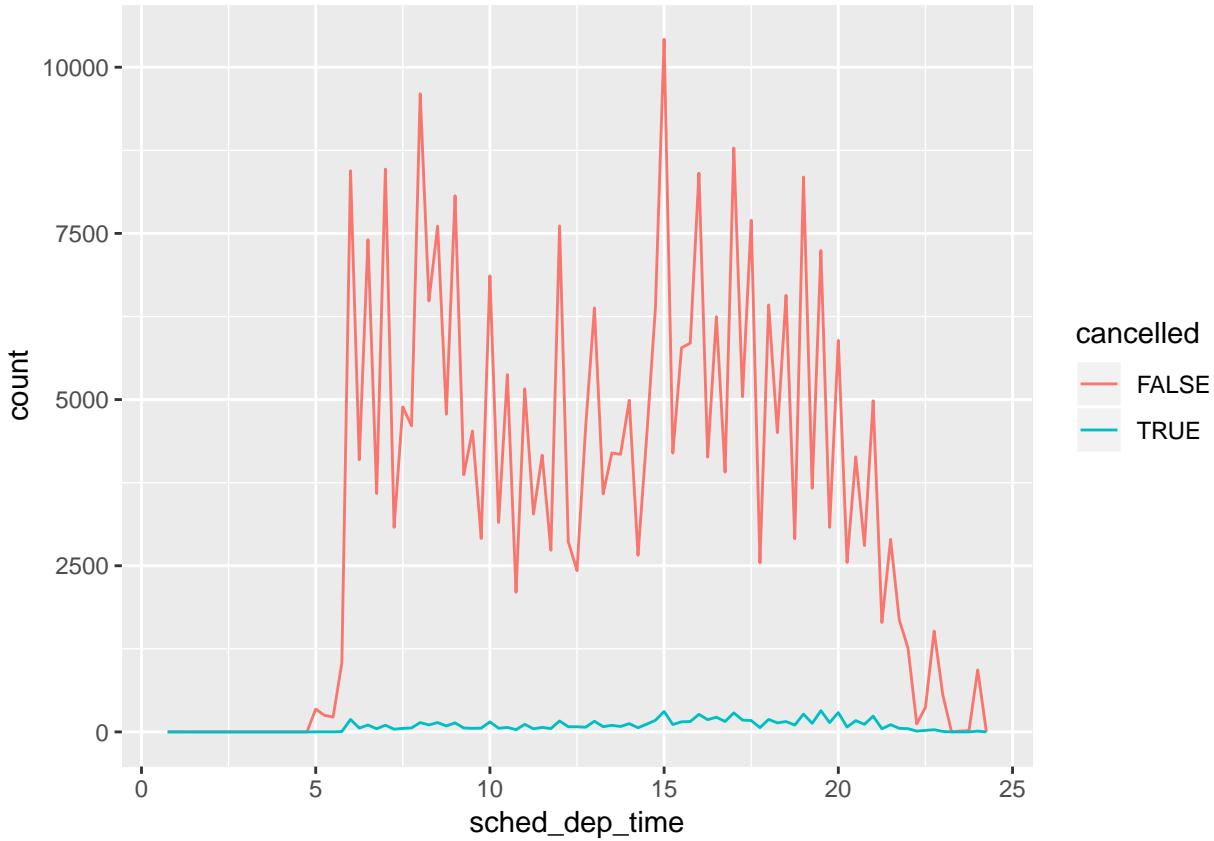
```
ggplot(data = diamonds2.2, mapping = aes(x = x, y = y)) +  
  geom_point(na.rm = TRUE)
```



Quando voce quiser que os valores em branco aparecam em seu gráfico, você pode usar a função nycflights13::

```
# Aqui o exemplo e com uma base de dados sobre voos

nycflights13::flights %>%
  mutate(
    cancelled = is.na(dep_time),
    sched_hour = sched_dep_time %/% 100,
    sched_min = sched_dep_time %% 100,
    sched_dep_time = sched_hour + sched_min / 60
  ) %>%
  ggplot(mapping = aes(sched_dep_time)) +
  geom_freqpoly(mapping = aes(colour = cancelled), binwidth = 1/4)
```



Observe que podemos comparar a quantidade de vôos feitos em relação a quantidade de vôos cancelados.

Exemplos CAP 7.5.1 Covariância

Entre Variável categórica e contínua é mais difícil visualizar a diferença na distribuição pois as contagens gerais são muito diferentes:

gráfico sobre corte e preço

```
ggplot(data = diamonds, mapping = aes(x = price)) +
  geom_freqpoly(mapping = aes(colour = cut), binwidth = 500)
```

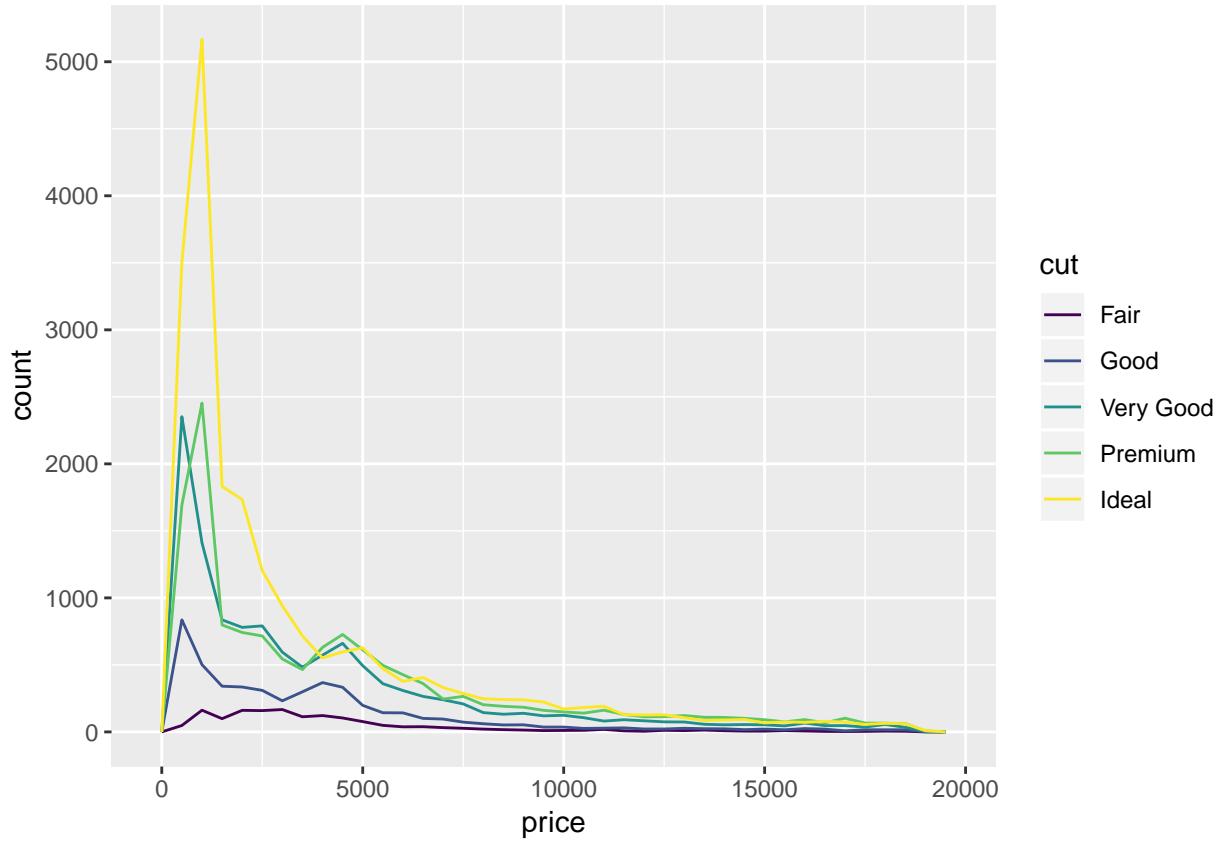
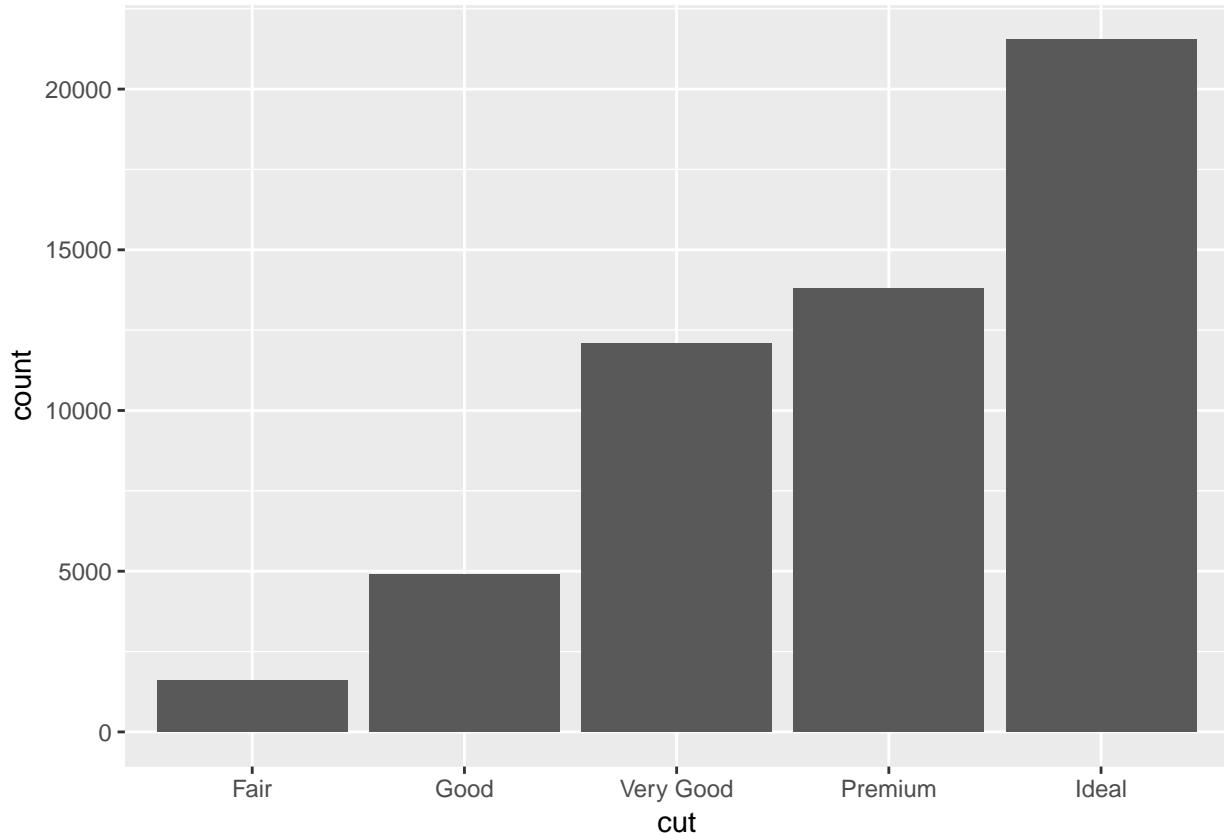


gráfico sobre distribuição do corte na população

```
ggplot(diamonds) +  
  geom_bar(mapping = aes(x = cut))
```



Para facilitar a comparação, precisamos trocar o que é mostrado no eixo y. Em vez de exibir a contagem, exibiremos densidade, que é a contagem padronizada para que a área sob cada polígono de frequência seja uma.

```
ggplot(data = diamonds, mapping = aes(x = price, y = ..density..)) +  
  geom_freqpoly(mapping = aes(colour = cut), binwidth = 500)
```

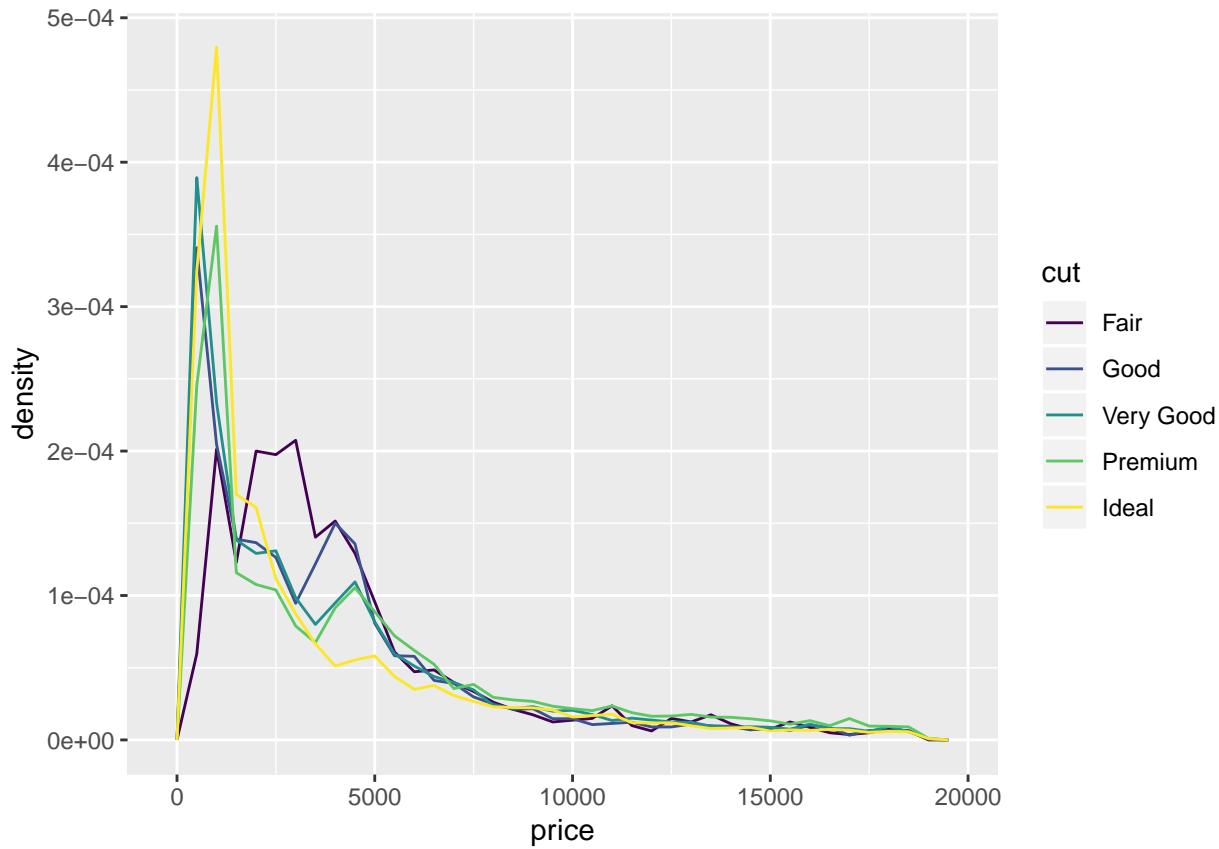
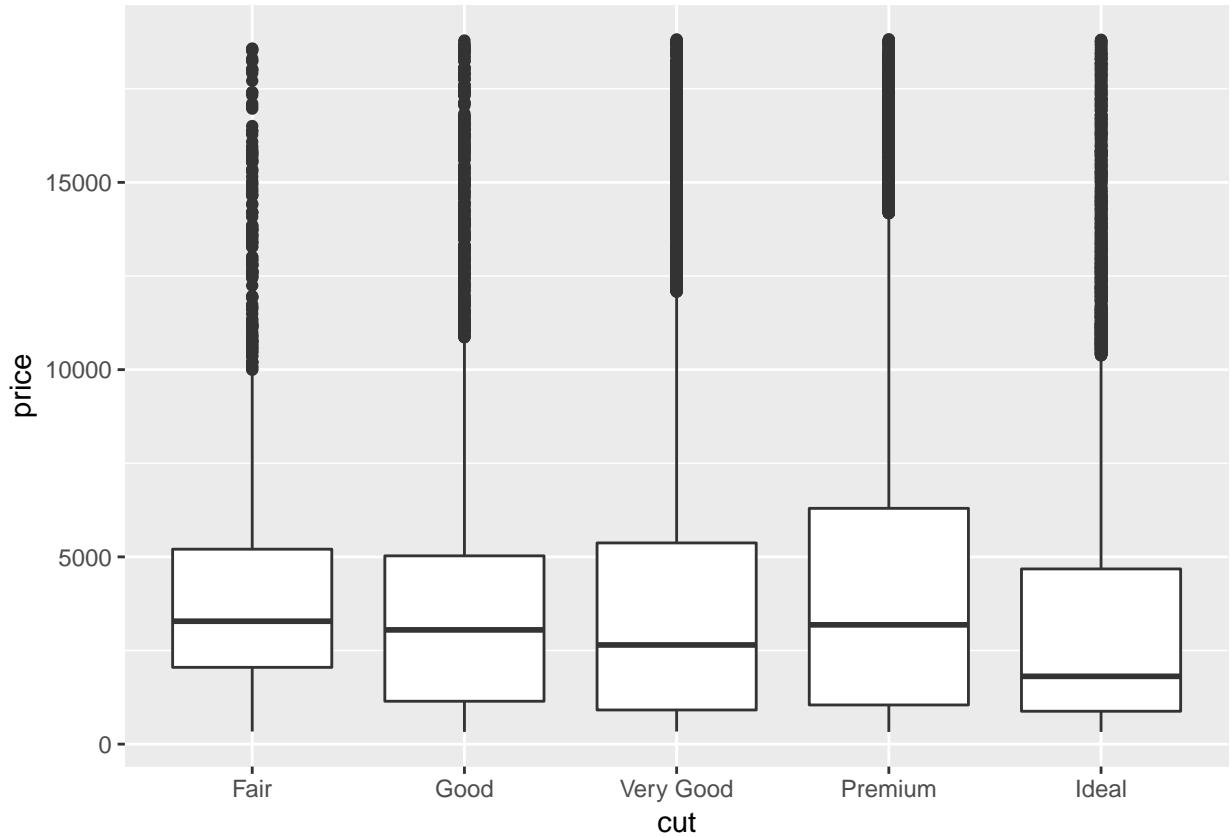


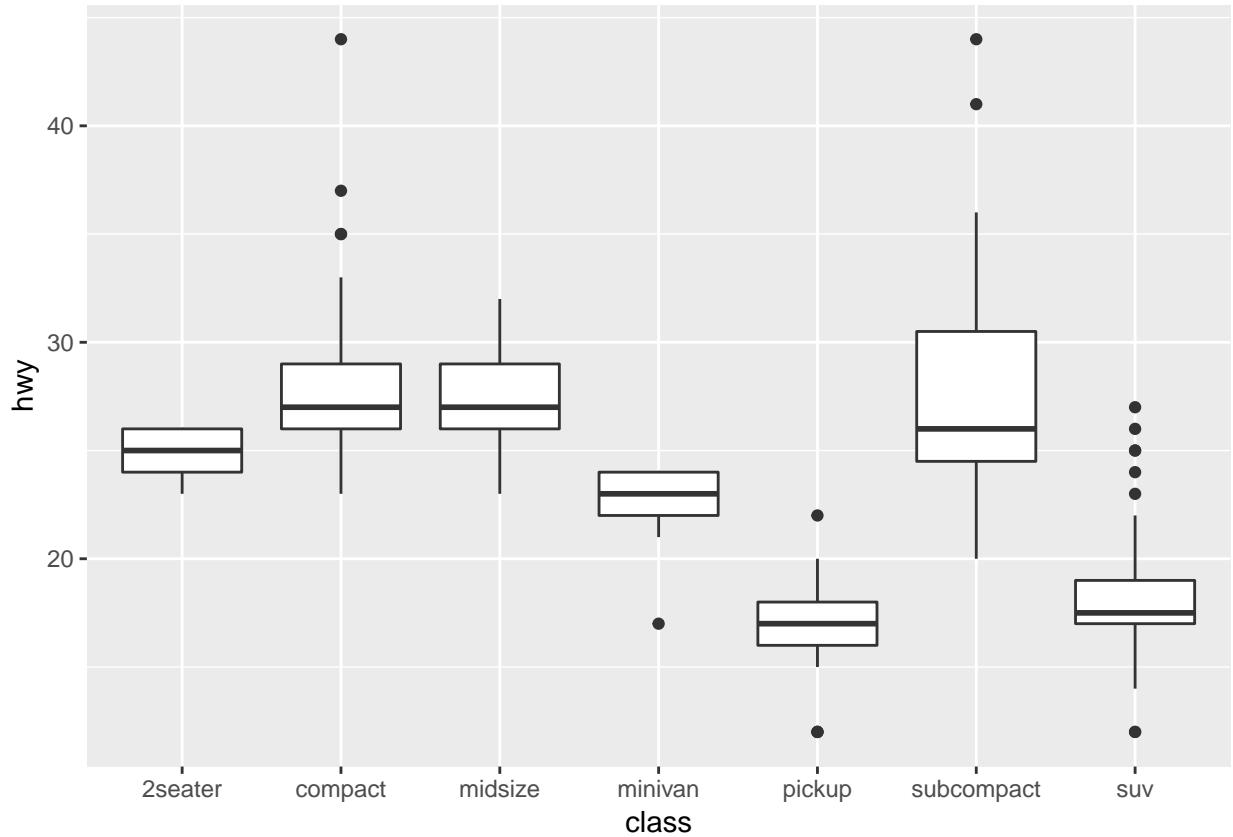
gráfico de distribuição de preço por corte usando geom_boxplot ():

```
ggplot(data = diamonds, mapping = aes(x = cut, y = price)) +  
  geom_boxplot()
```



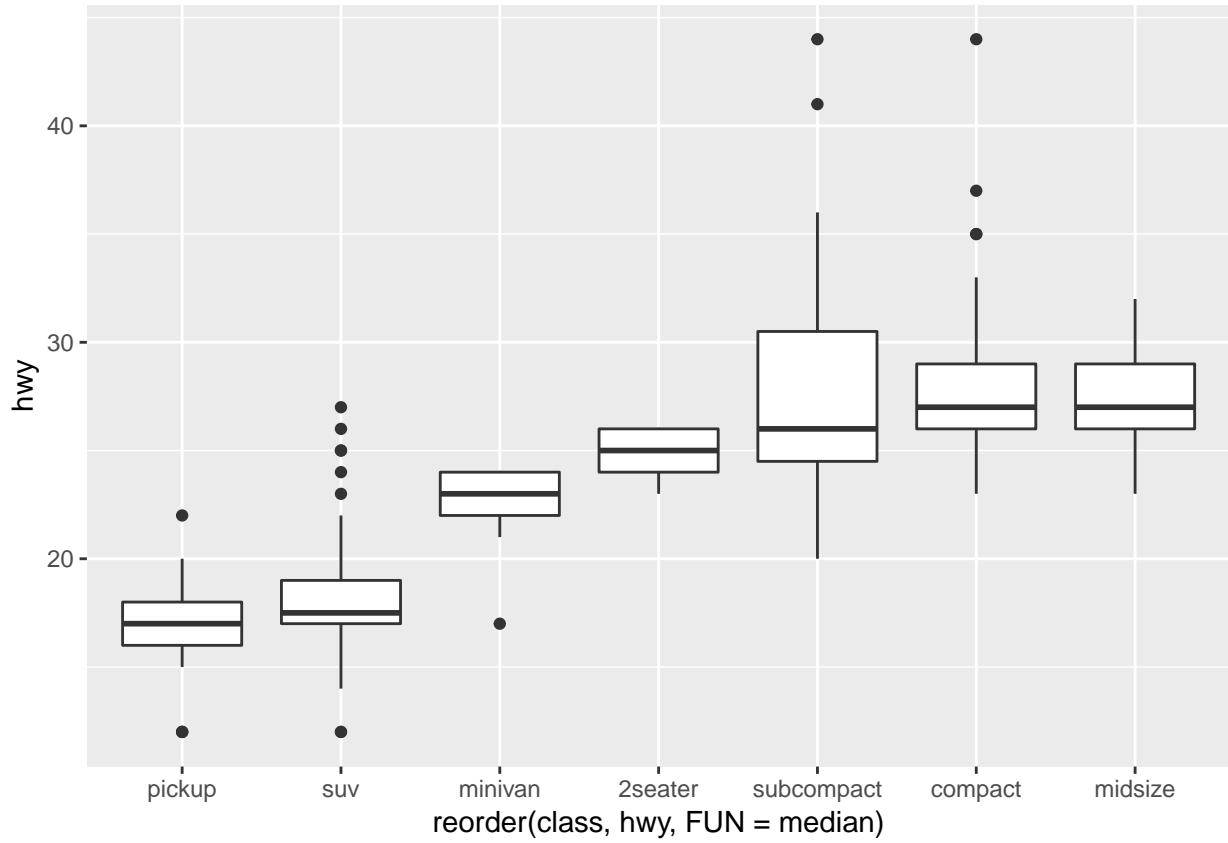
Variável class na base de dados mpg

```
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +  
  geom_boxplot()
```



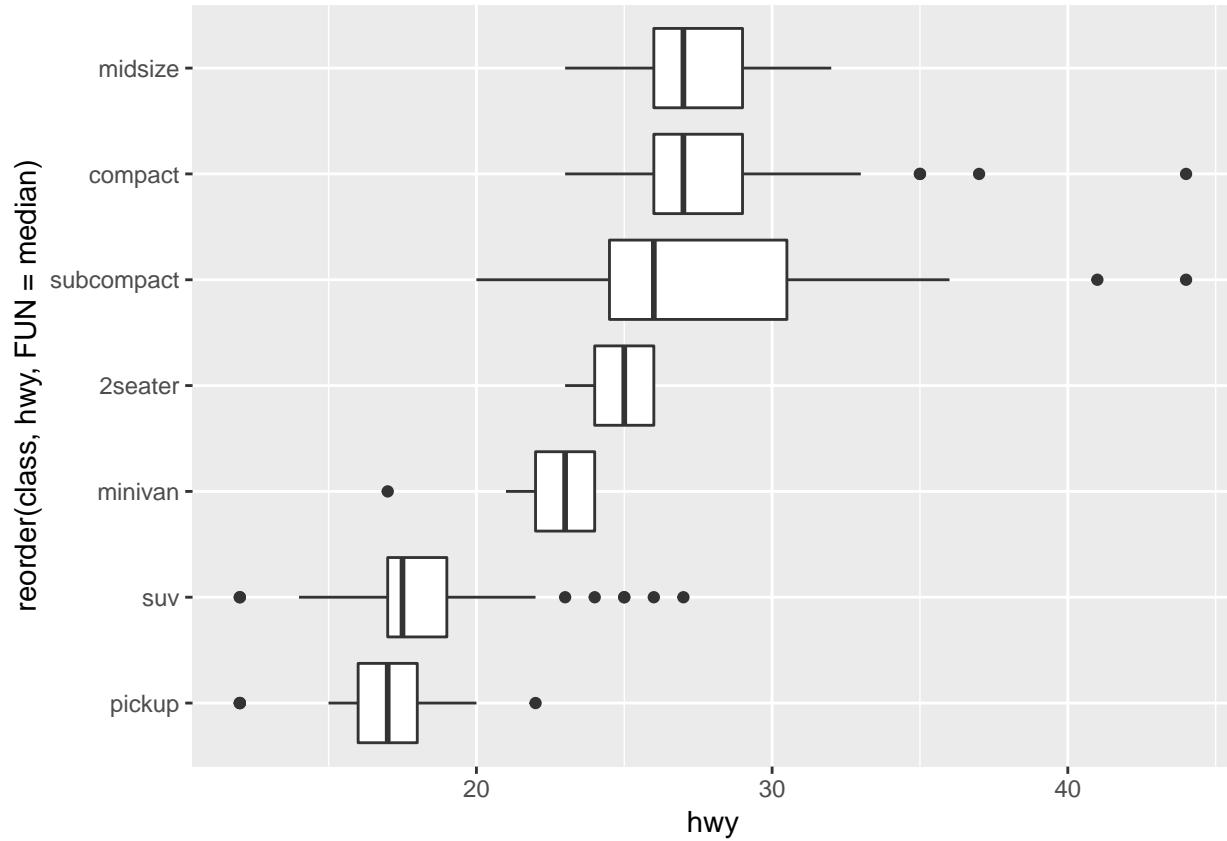
reordenando class com base no valor mediano de hwy:

```
ggplot(data = mpg) +  
  geom_boxplot(mapping = aes(x = reorder(class, hwy, FUN = median), y = hwy))
```



Nas variáveis com nomes longos, geom_boxplot () funciona melhor girando 90 graus

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = reorder(class, hwy, FUN = median), y = hwy)) +
  coord_flip() ## coord_flip() e a função que permite girar o gráfico
```

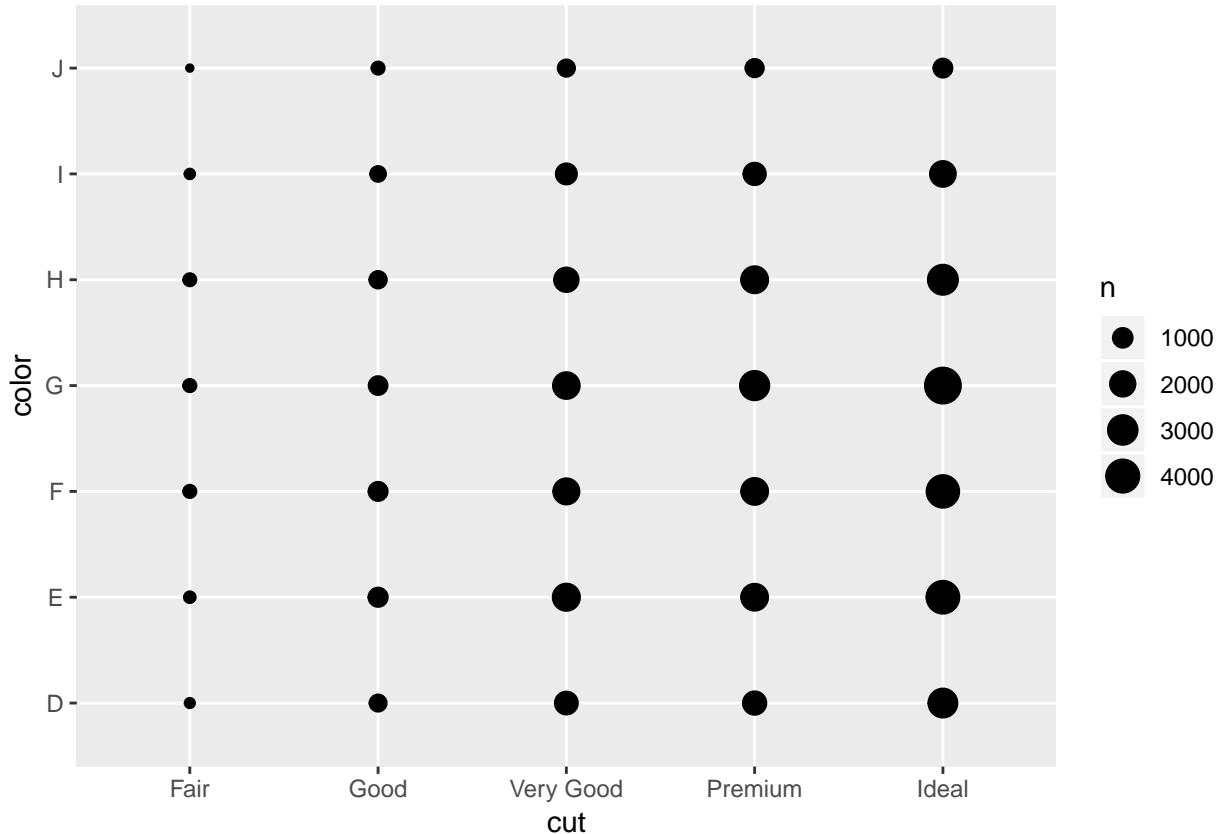


Exemplos CAP 7.5.2

Duas variáveis categóricas

Para visualizar a covariância entre 2 variáveis categóricas é preciso contar o número de observações para cada combinação. Uma forma de fazer isso é usando a função `geom_count()`:

```
ggplot(data = diamonds) +
  geom_count(mapping = aes(x = cut, y = color))
```



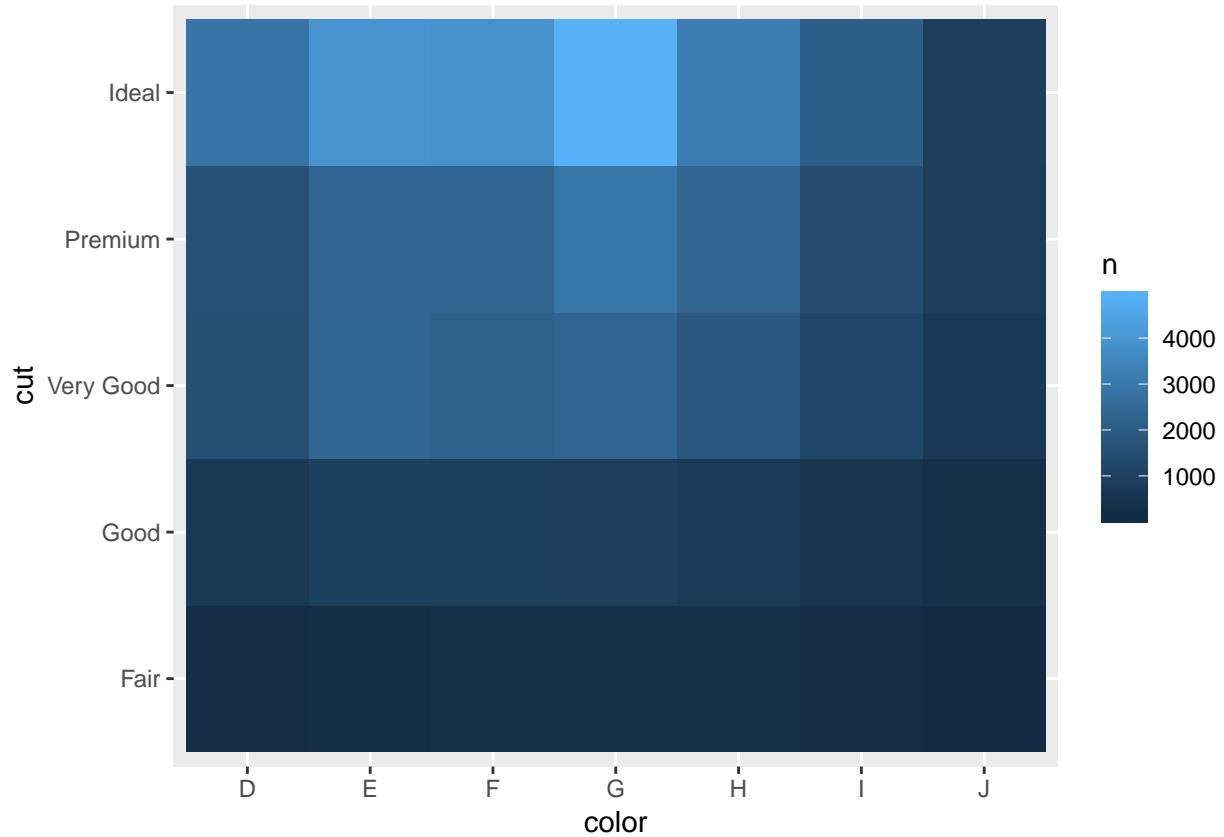
outra forma é usando a função dplyr para visualizar manualmente

```
diamonds %>%
  count(color, cut)
```

```
## # A tibble: 35 x 3
##   color    cut     n
##   <ord> <ord> <int>
## 1 D      Fair    163
## 2 D      Good    662
## 3 D      Very Good 1513
## 4 D      Premium 1603
## 5 D      Ideal   2834
## 6 E      Fair    224
## 7 E      Good    933
## 8 E      Very Good 2400
## 9 E      Premium 2337
## 10 E     Ideal   3903
## # ... with 25 more rows
```

Visualizando com geom_tile() e a estética de preenchimento: basicamente unimos os dois comandos

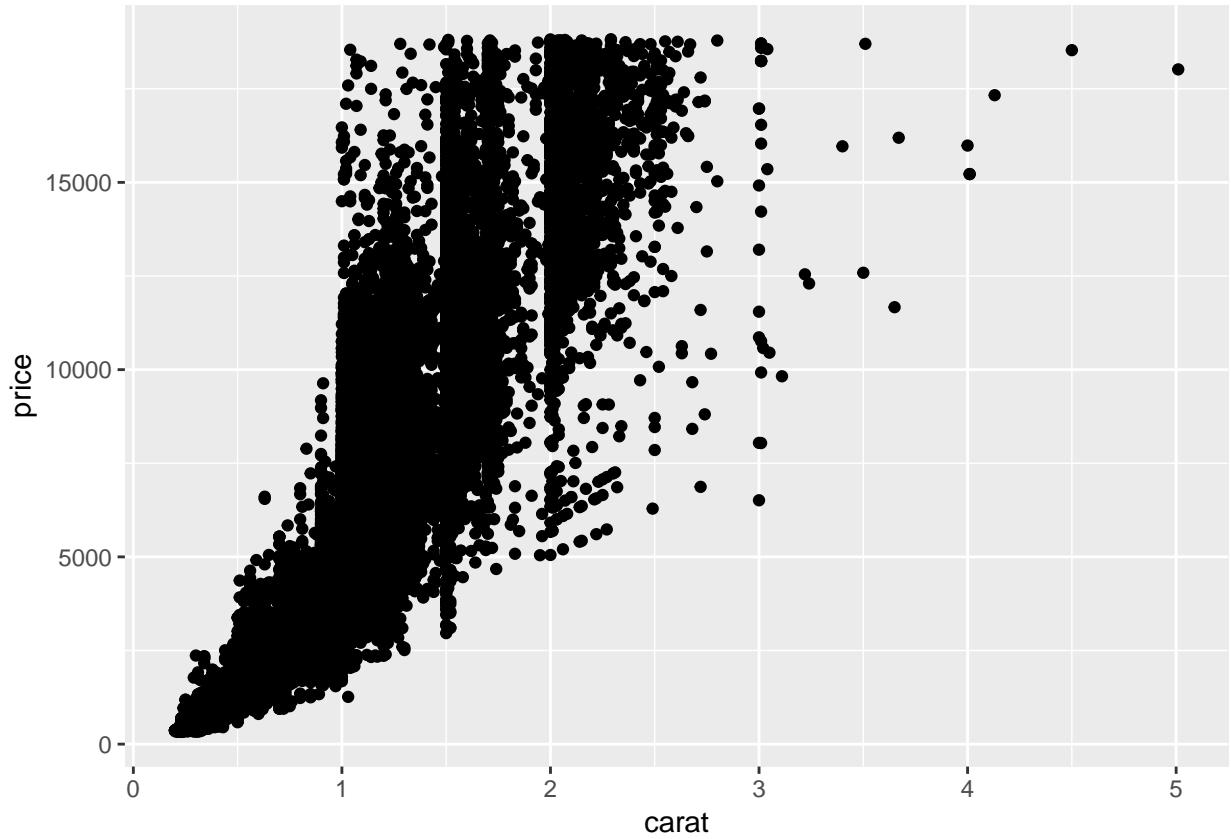
```
diamonds %>%
  count(color, cut) %>%
  ggplot(mapping = aes(x = color, y = cut)) +
  geom_tile(mapping = aes(fill = n)) ## basicamente unimos os dois comandos
```



Exemplos CAP 7.5.3

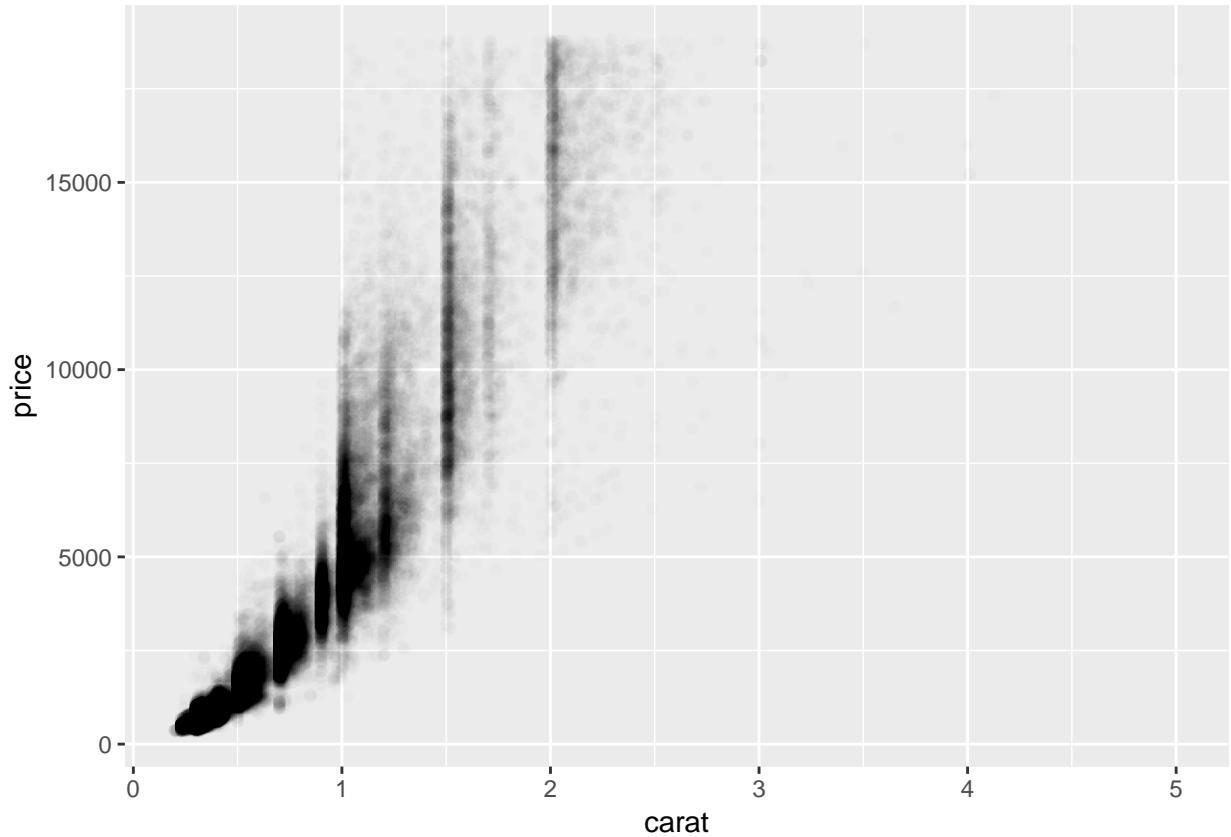
Observando duas variáveis contínuas usando a função geom_point()

```
ggplot(data = diamonds) +  
  geom_point(mapping = aes(x = carat, y = price))
```



Usando a estética alfa para adicionar transparência

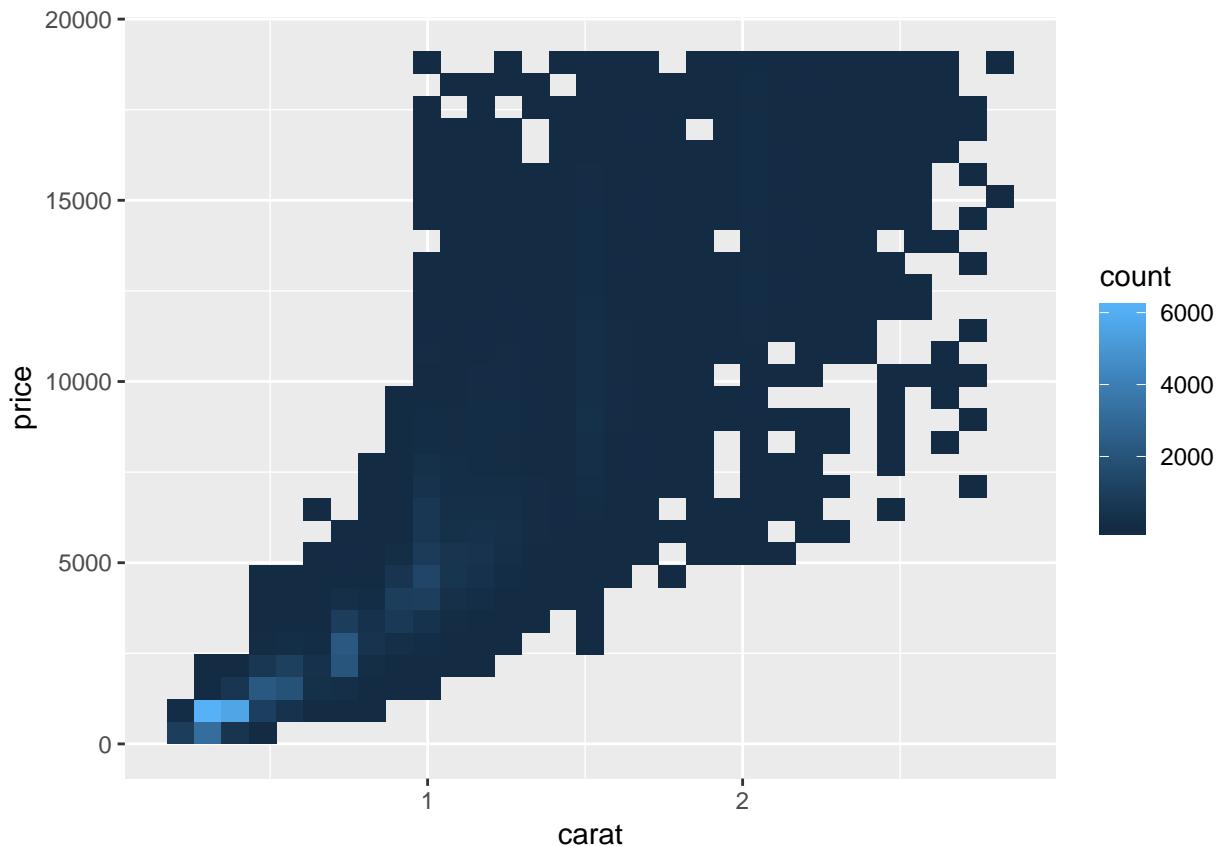
```
ggplot(data = diamonds) +  
  geom_point(mapping = aes(x = carat, y = price), alpha = 1 / 100)
```



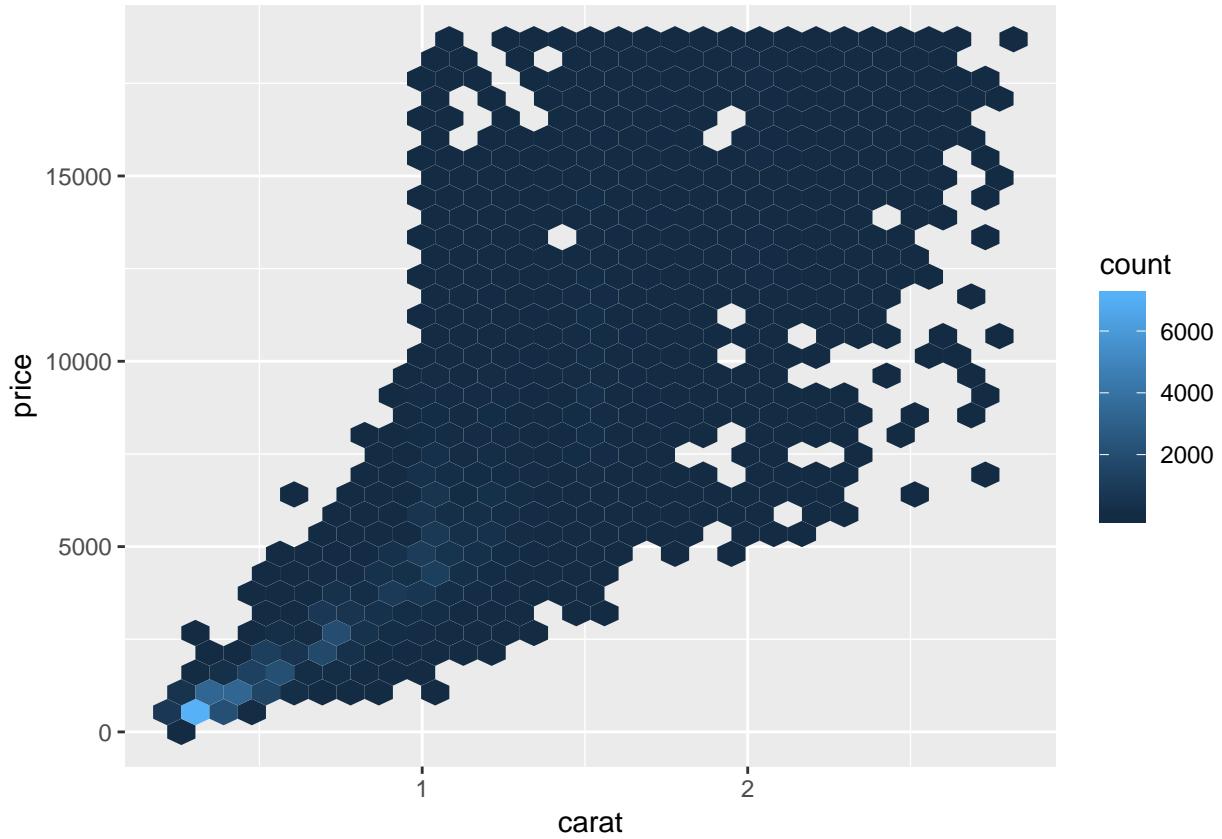
```
# novo comando acrescentado = alpha = 1 / 100)
```

Usando geom_bin2d () e geom_hex () para dividir as coordenadas em bins 2d

```
ggplot(data = smaller) +  
  geom_bin2d(mapping = aes(x = carat, y = price))
```

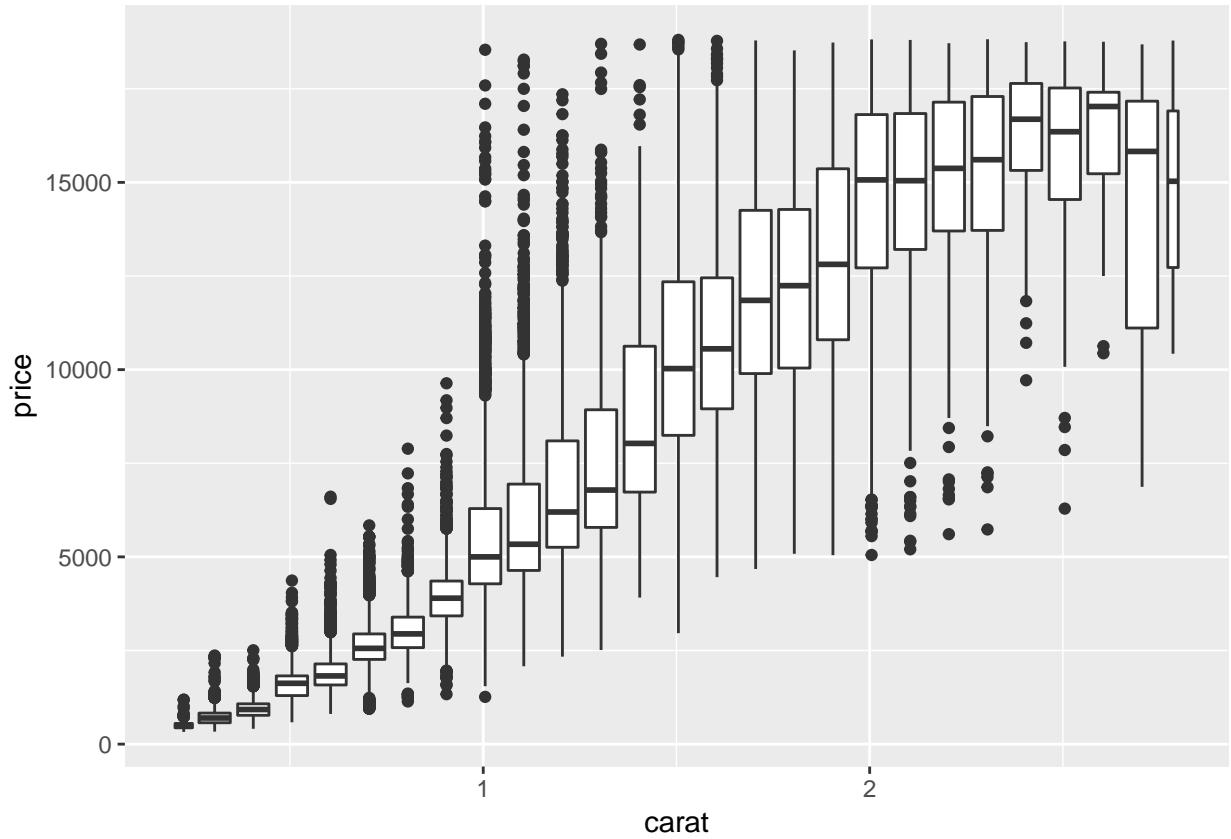


```
ggplot(data = smaller) +
  geom_hex(mapping = aes(x = carat, y = price))
```



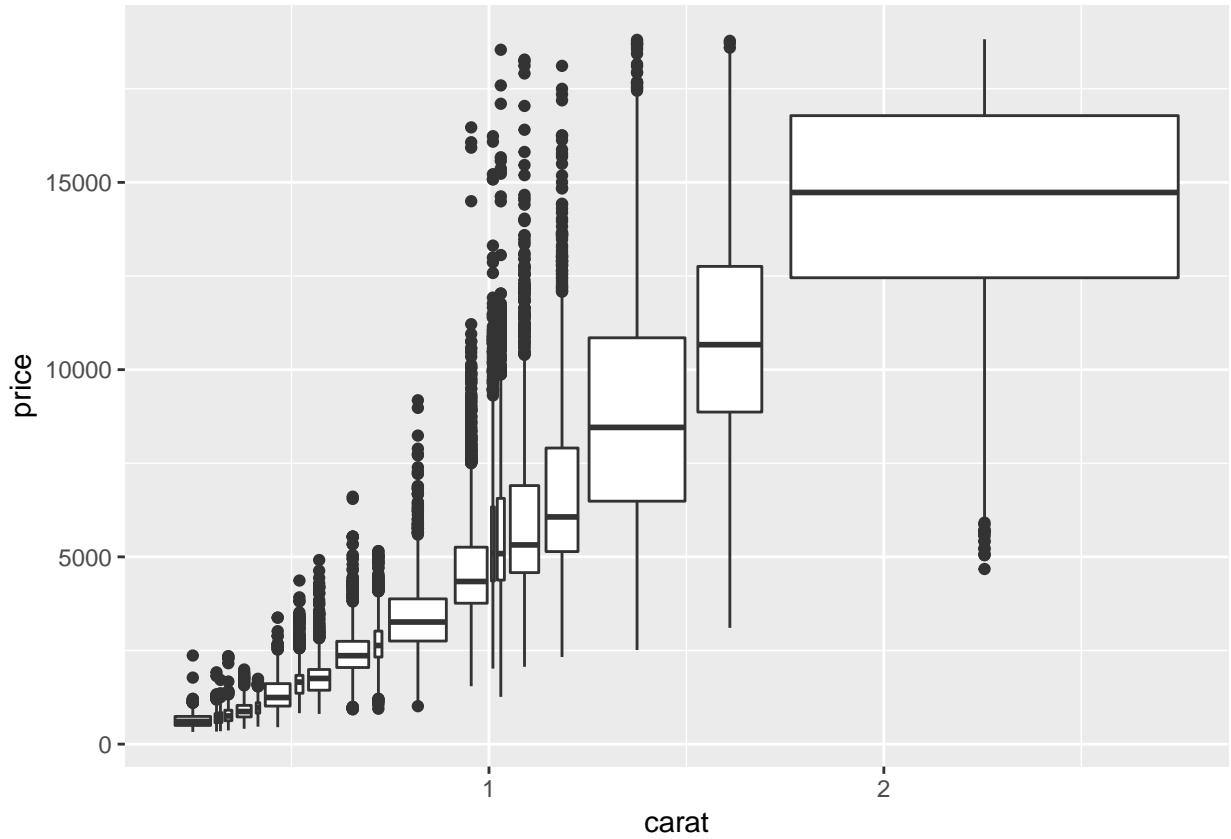
Usando boxplot

```
ggplot(data = smaller, mapping = aes(x = carat, y = price)) +  
  geom_boxplot(mapping = aes(group = cut_width(carat, 0.1)))
```



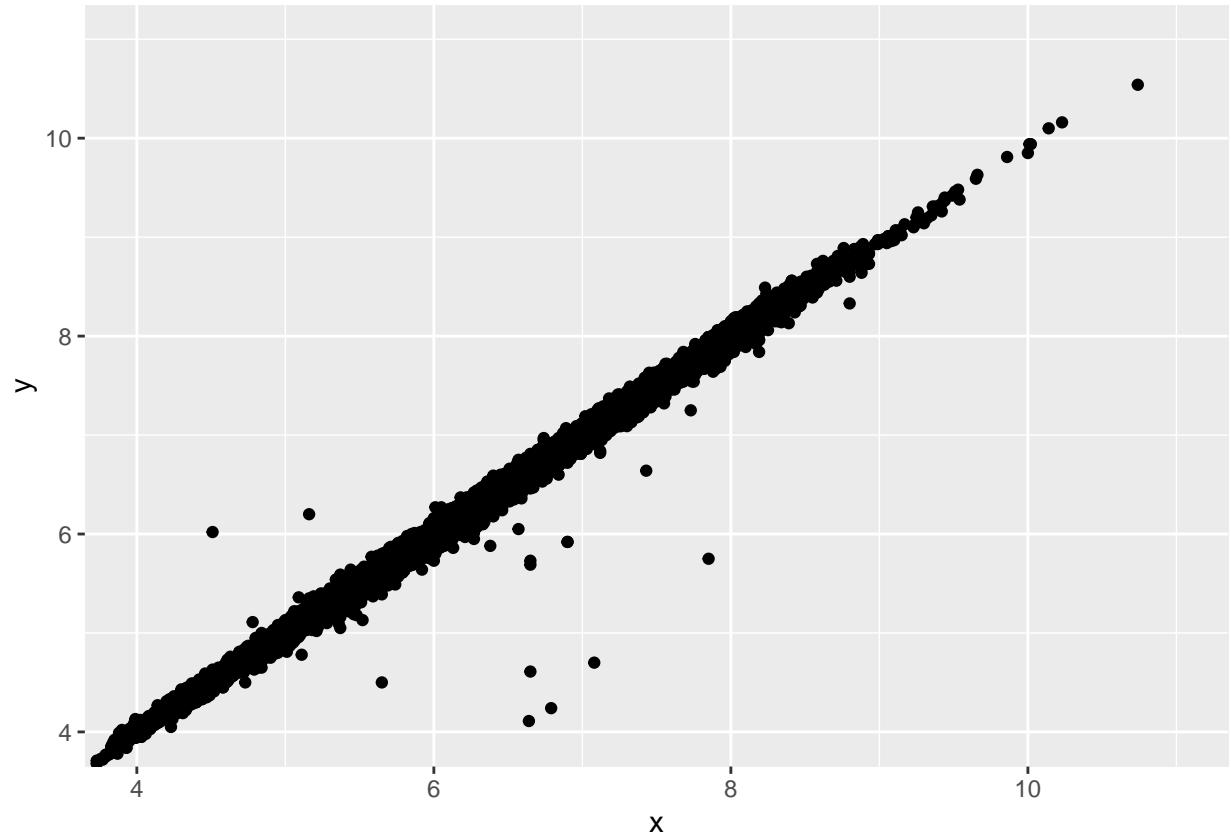
Usando `cut_number()` para exibir aproximadamente o mesmo número de pontos em cada caixa.

```
ggplot(data = smaller, mapping = aes(x = carat, y = price)) +  
  geom_boxplot(mapping = aes(group = cut_number(carat, 20)))
```



A grande sacada é que graficos bidimensionais revelam outliers que não são visíveis em parcelas unidimensionais. Observe:

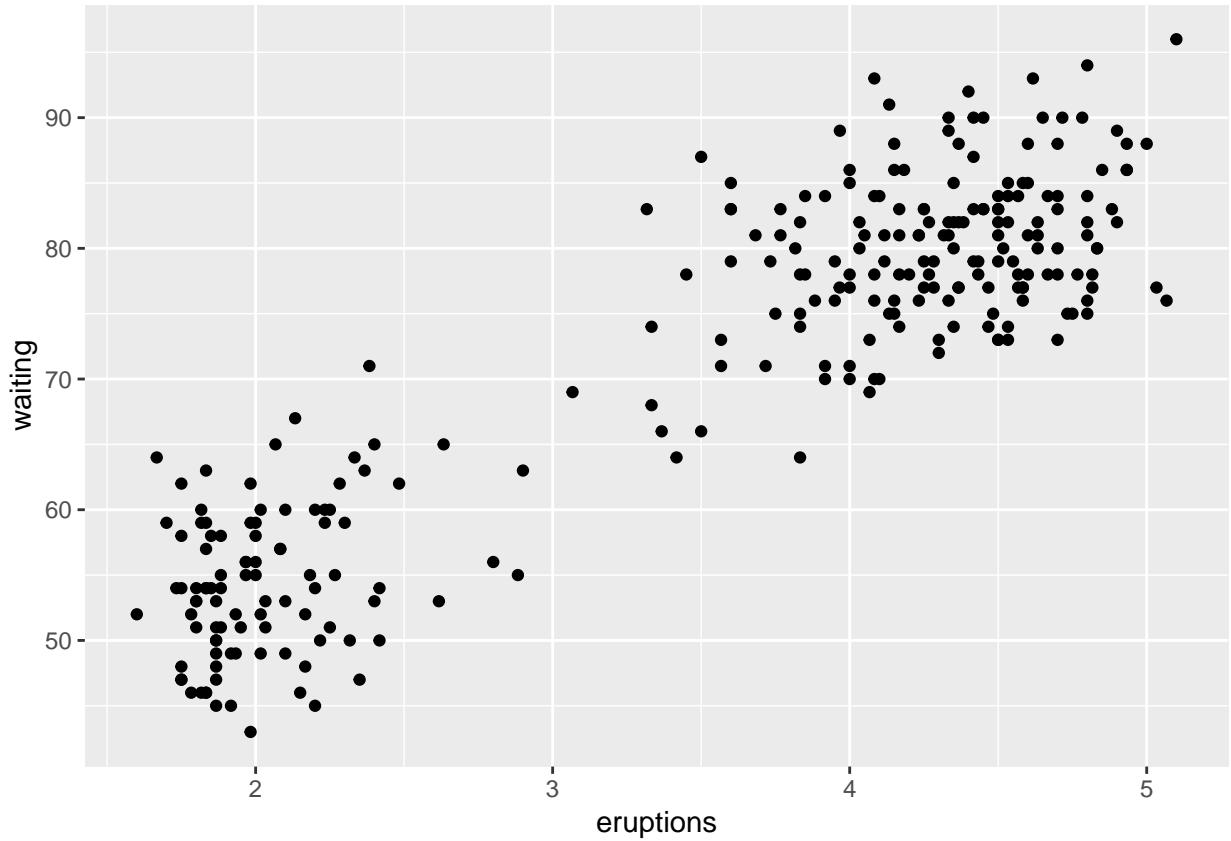
```
ggplot(data = diamonds) +
  geom_point(mapping = aes(x = x, y = y)) +
  coord_cartesian(xlim = c(4, 11), ylim = c(4, 11))
```



Exemplos CAP 7.6 Padrões nos fornecem pistas sobre relacionamentos das variáveis no banco.

```
# Retornando a base de dados sobre vulcoes, podemos utilizar graficos de  
# dispersao para observar padroes de comportamento
```

```
ggplot(data = faithful) +  
  geom_point(mapping = aes(x = eruptions, y = waiting))
```



Podemos usar o pacote modelr para extrair padrões dos dados

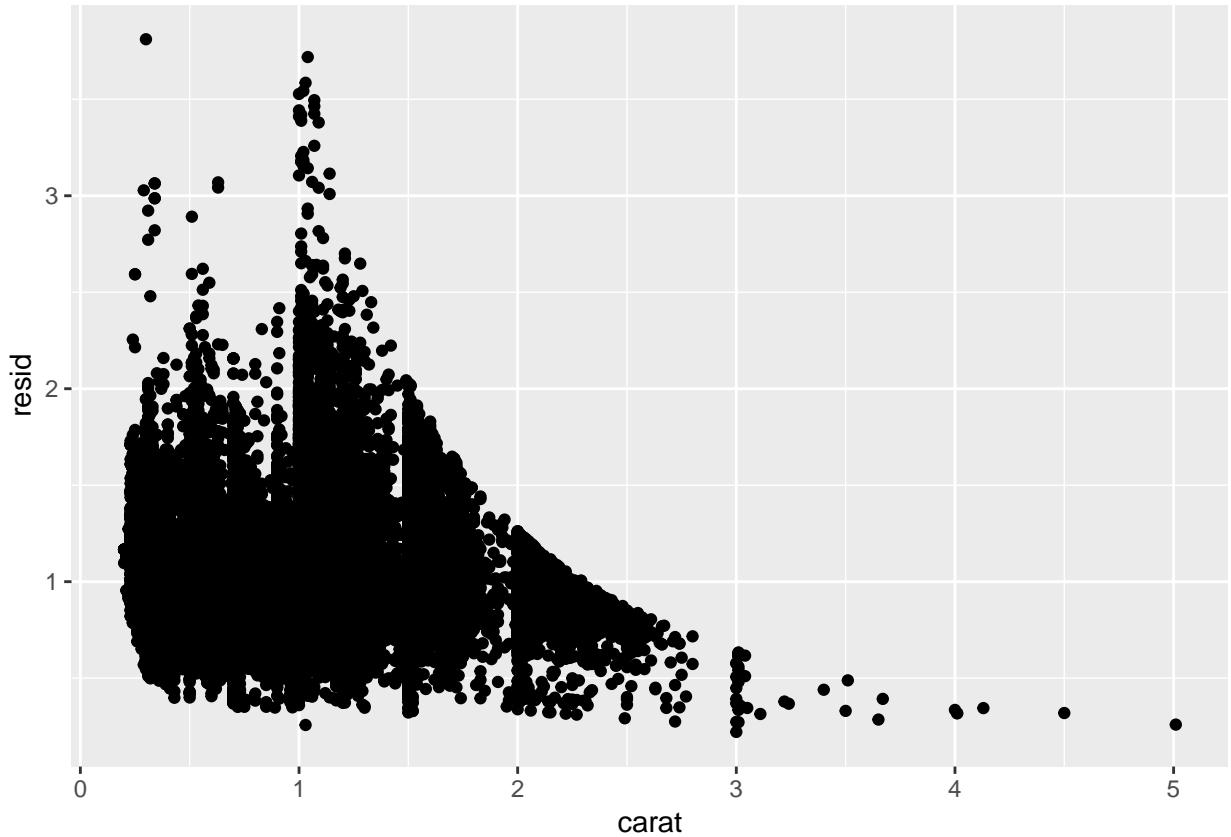
Vamos retornar agora para a base de diamantes

```
library(modelr)

mod <- lm(log(price) ~ log(carat), data = diamonds)

diamonds2.6 <- diamonds %>%
  add_residuals(mod) %>%
  mutate(resid = exp(resid))

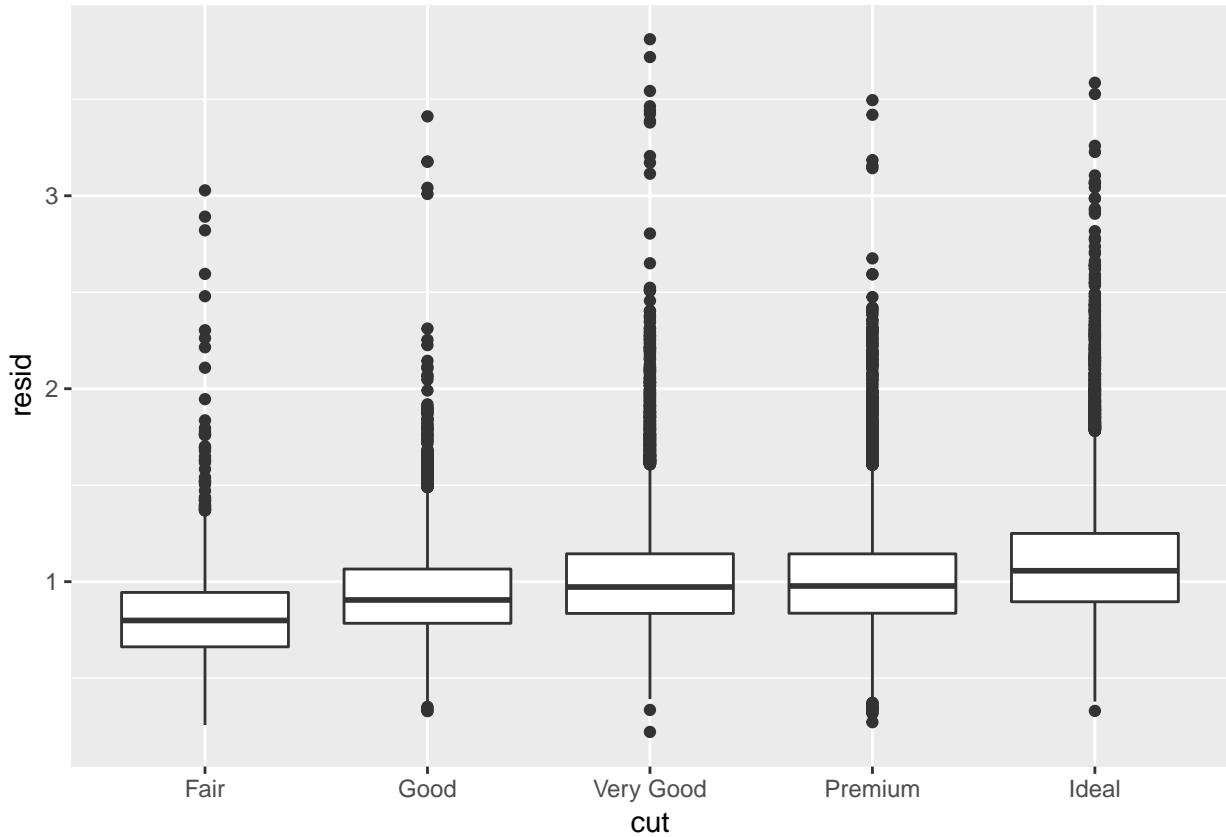
ggplot(data = diamonds2.6) +
  geom_point(mapping = aes(x = carat, y = resid))
```



O código se ajustou a um modelo que prevê preço de quilate e, em seguida, calcula os resíduos. Os resíduos nos dão uma visão do preço do diamante, uma vez que o efeito do quilate foi removido.

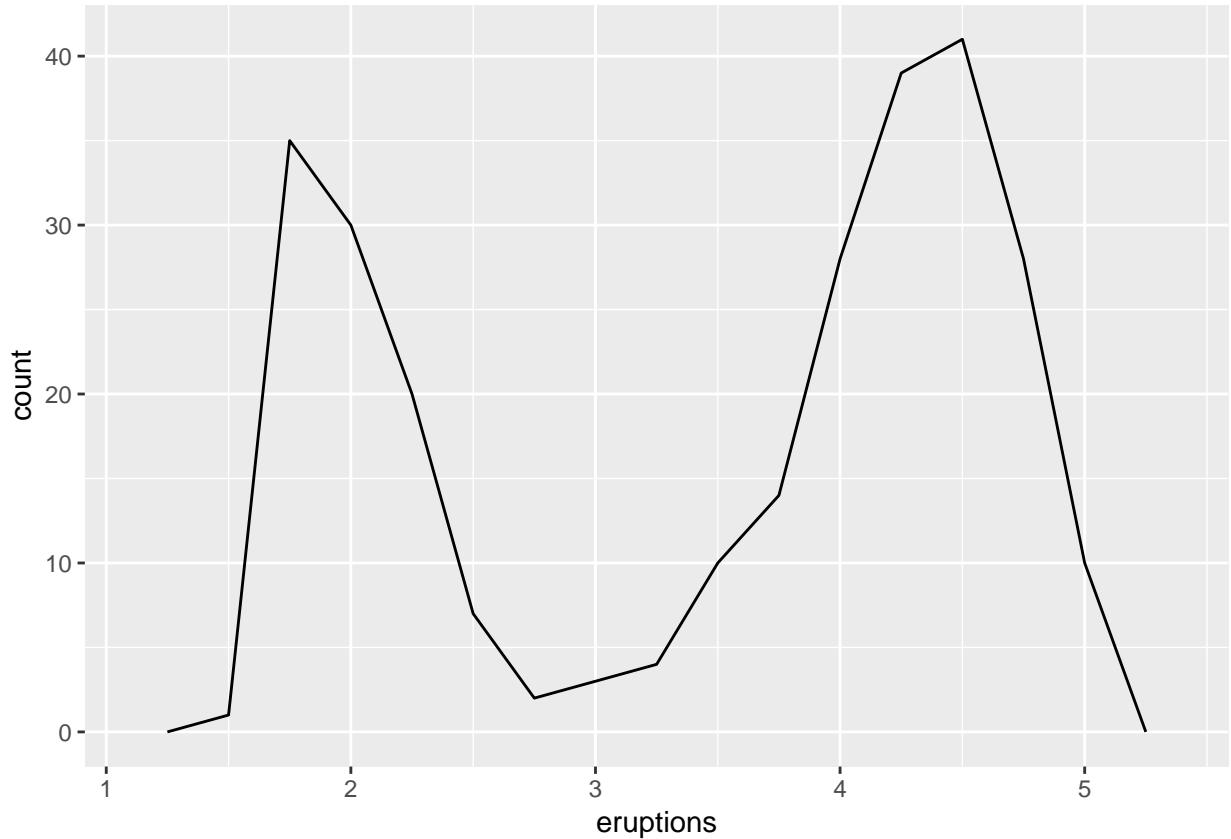
Agora vamos observar o gráfico desse novo modelo

```
ggplot(data = diamonds2.6) +  
  geom_boxplot(mapping = aes(x = cut, y = resid))
```

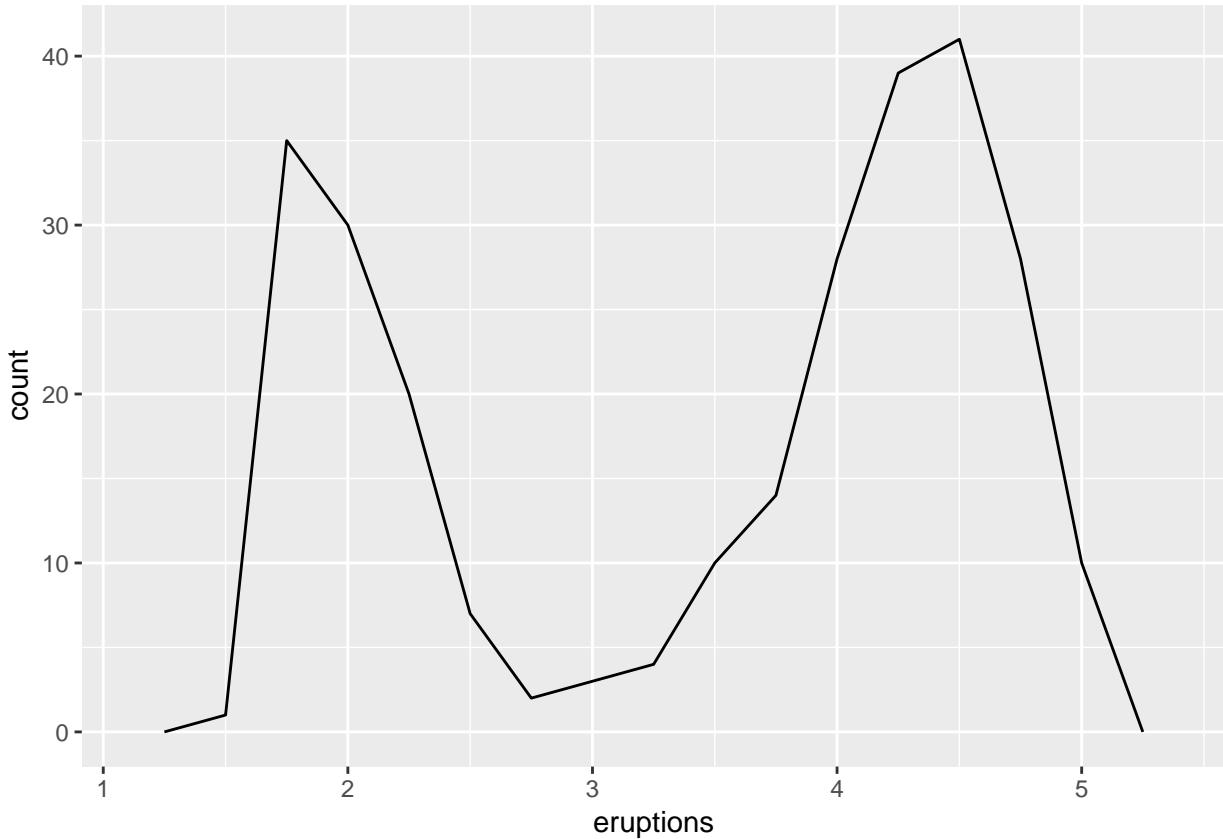


Exemplos CAP 7.7 Existem formas mais práticas de chamar os comandos e funções. Por exemplo

```
# com o tempo, ao invés de solicitar o comando assim:
ggplot(data = faithful, mapping = aes(x = eruptions)) +
  geom_freqpoly(binwidth = 0.25) ## exemplo 1
```



```
# podemos chamar o gráfico assim:  
ggplot(faithful, aes(eruptions)) +  
  geom_freqpoly(binwidth = 0.25) ## exemplo 2
```



EXERCÍCIO 10 ** Com os dados disponibilizados na plataforma, reproduza os resultados do livro de Kellstedt (2013) utilizando o código apresentado nos slides da aula.**

```
# selecionando diretório
setwd("C:/Users/Carol/Desktop/cadeiras-mestrado/análise de dados - davi moreira/listas/lista 05")

# carregar arquivo
load("vote_growth_usa.Rdata")
# modelo de regressão bivariada : estimando o modelo
reg.10 <- lm(Vote ~ Growth, data = bd)

summary(reg.10)

##
## Call:
## lm(formula = Vote ~ Growth, data = bd)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -8.1968 -3.7667 -0.7972  3.1294 10.0107 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 51.5082    0.8569  60.110 < 2e-16 ***
## Growth      0.6249    0.1577   3.962  0.00039 ***  
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.955 on 32 degrees of freedom
## Multiple R-squared:  0.3291, Adjusted R-squared:  0.3081
## F-statistic:  15.7 on 1 and 32 DF,  p-value: 0.0003898

```

```
confint(reg.10)
```

```

##               2.5 %    97.5 %
## (Intercept) 49.7627069 53.2536139
## Growth      0.3036193  0.9461963

```

EXERCÍCIO 11

**Com os dados e as variáveis do exercício 10, realize uma análise de regressão considerando apenas o período de 1876 a 1932. Apresente os resultados e os compare quanto ao modelo completo do exercício 10 em relação a:

- a) Significância estatística dos resultados
- b) Intervalo de confiança para beta
- c) Medidas de ajuste do modelo

```
#solicitando base de dados
```

```
setwd("C:/Users/Carol/Desktop/cadeiras-mestrado/análise de dados - davi moreira/listas/lista 06")
load("vote_growth_usa.Rdata")
```

```
# renomeando o rotulo
vote.growth <- bd
```

```
# filtrando para o periodo pedido
vote.growth.filtro <- vote.growth[1:15, c("Year", "Growth", "Vote")]
```

```
# solicitando nova regressao
reg.11 <- lm(Vote ~ Growth, data = vote.growth.filtro)
```

```
confint(reg.11)
```

```

##               2.5 %    97.5 %
## (Intercept) 48.66440604 55.305605
## Growth      0.02230689  1.044796

```

```
summary(reg.11)
```

```

##
## Call:
## lm(formula = Vote ~ Growth, data = vote.growth.filtro)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.7209 -3.5931  0.5013  3.1253  9.3127
##
```

```

## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 51.9850    1.5371 33.821 4.66e-14 ***
## Growth      0.5336    0.2366  2.255   0.042 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 5.638 on 13 degrees of freedom
## Multiple R-squared:  0.2811, Adjusted R-squared:  0.2258
## F-statistic: 5.083 on 1 and 13 DF,  p-value: 0.04205

```

- Em relação a significância estatística o modelo completo da questão 10 apresenta um p valor menor que o do modelo filtrado entre os anos 1876-1932. Logo, o modelo da questão 11 é mais próximo do padrão estatístico de 0,5. É moderadamente mais significativo estatisticamente. No entanto, todos os dois modelos estão dentro do intervalo do p-valor e são significativos
- Em relação ao intervalo de confiança o modelo completo da questão 10 apresenta uma amplitude de intervalo de confiança a menor que o do modelo filtrado entre os anos 1876 - 1932.
- Em relação a medida de ajuste de modelo, no modelo completo da questão 10 o valor do ajuste é de 0,31, enquanto no modelo filtrado, o valor do ajuste é de 0,23. No modelo da questão 10 o erro padrão foi estimado em 4,955. Já no modelo filtrado foi estimado em 5,638. Como o erro padrão foi menor no modelo completo da questão 10, é preferível usar este modelo para explicar a influência do crescimento do econômico no voto do partido da situação.