Regressão

Fabio Carvalho Lima

14/11/2019

```
options(digits = 6)
library(tidyverse)
```

```
## -- Attaching packages ------------------------------ tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts --------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(HistData)
library(RColorBrewer)
library(ggExtra)
data("GaltonFamilies")
```

```
# split dataset between fathers and sons and mothers and daughters

set.seed(1989, sample.kind = "Rounding")
```

```
## Warning in set.seed(1989, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

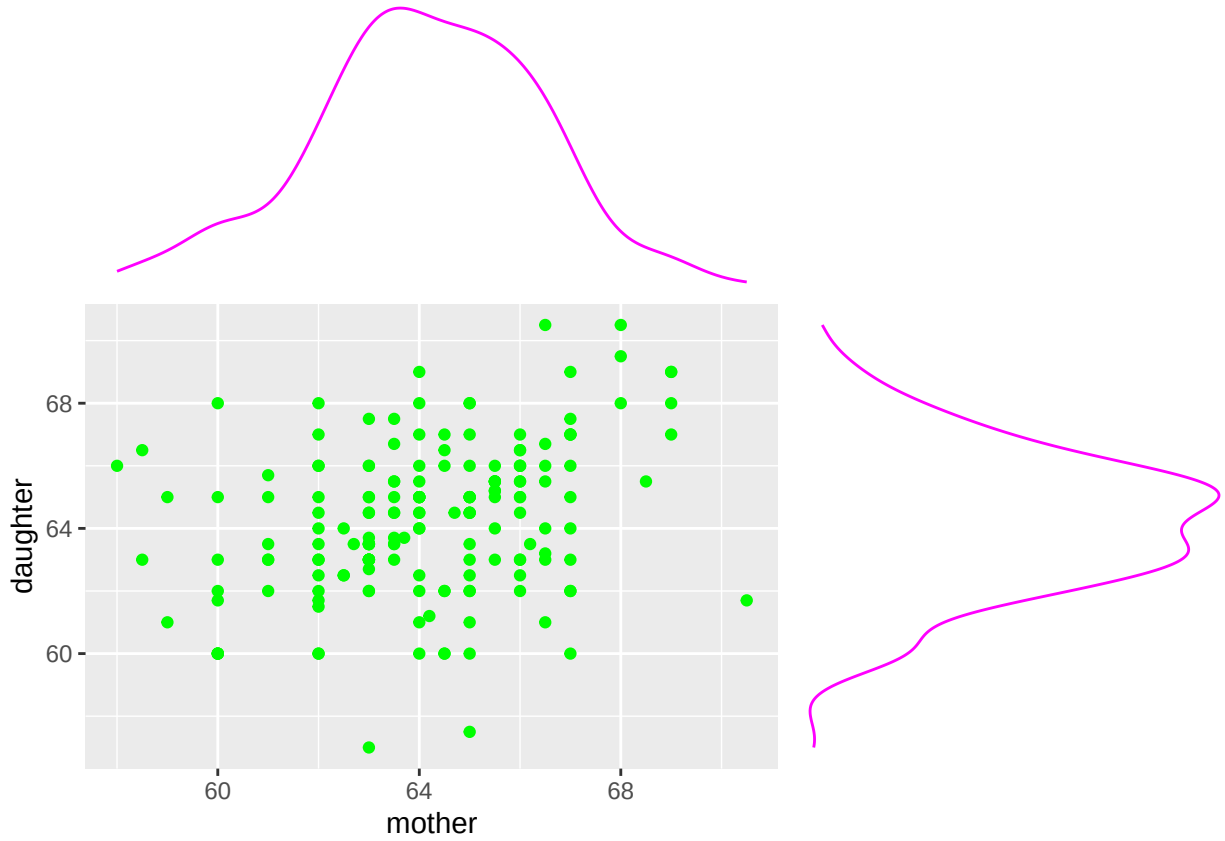
```
female_heights <- GaltonFamilies %>%
   filter(gender == "female") %>%
   group_by(family) %>%
   sample_n(1) %>%
   ungroup() %>%
select(mother, childHeight)   %>%
rename(daughter = childHeight)
```

Vamos fazer um resumo das mães e filhas, primeiro vamos verificar a distribuição desses dados.

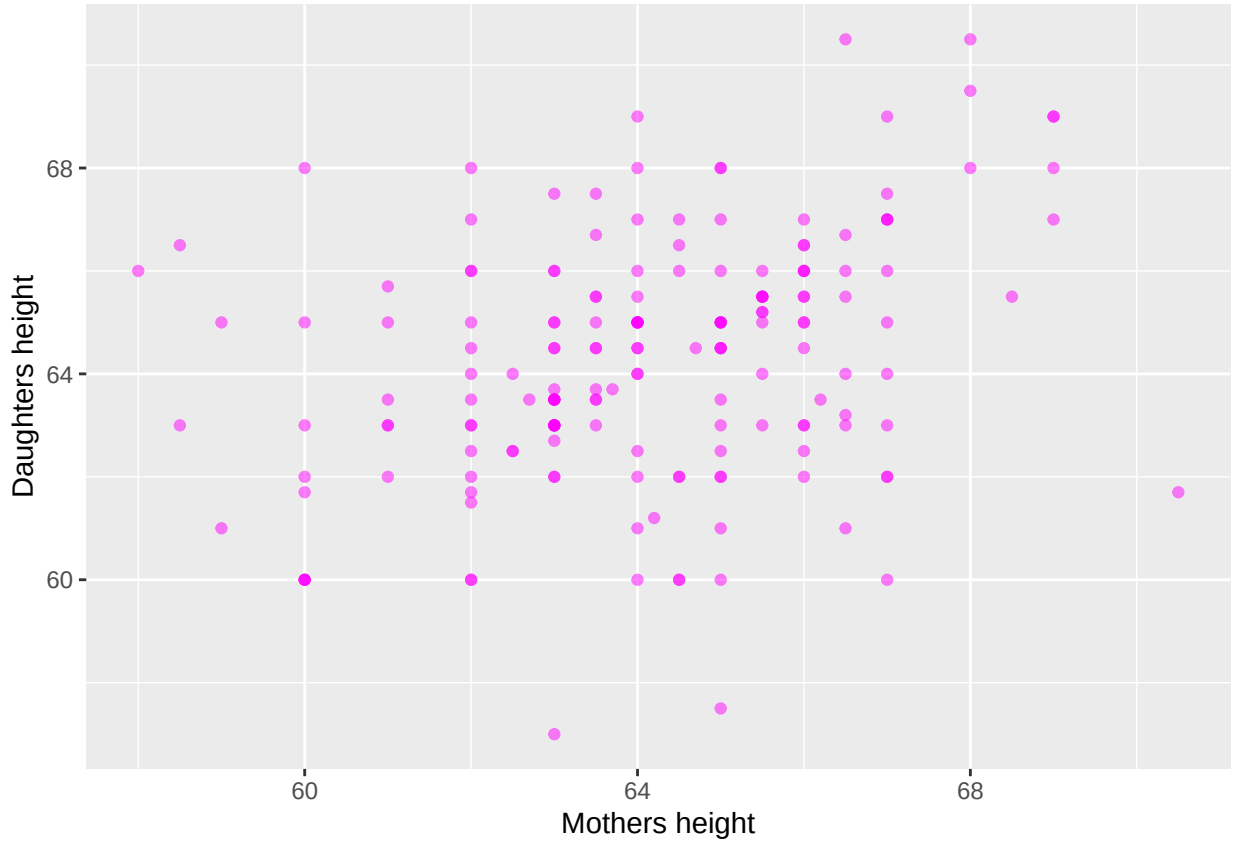
```
female_heights %>%
  summarize(mean(mother), sd(mother), mean(daughter), sd(daughter), r = cor(mother, daughter))
```

```
## # A tibble: 1 x 5
##   `mean(mother)` `sd(mother)` `mean(daughter)` `sd(daughter)`     r
##            <dbl>        <dbl>            <dbl>          <dbl> <dbl>
## 1           64.1         2.29             64.3           2.39 0.325
```

```
p <- female_heights %>%
     ggplot(aes(x = mother, y = daughter)) +
     geom_point(color = 'green') +
     theme(legend.position = 'none')

p1 <- ggMarginal(p, type ='density', color = 'magenta', size = 1.5)
p1
```




```
female_heights %>% ggplot(aes(mother, daughter)) +
                   geom_point(alpha = 0.5, color = 'magenta') +
                   labs(x = 'Mothers height', y = 'Daughters height')
```



O coeficiente de correlação é definido por uma lista de pares, como a média de padronizada dos valores:

```
x <- female_heights$mother
y <- female_heights$daughter

rho <- mean(scale(x)*scale(y))
rho
```

```
## [1] 0.322676
```

```
correlation <- cor(x, y, method = 'pearson')
correlation
```

```
## [1] 0.32452
```

Nas aplicações mais gerais de data science, nós observamos que os dados tem variações aleatórias. Por exemplo, em muitos casos, nós não observamos os dados de toda a população de interesse, e sim uma amostra aleatória. Assim como a média e o desvio padrão amostrais, a correlação amostral é usada para estimar a correlação populacional. Isso implica que a correlação calculada é um resumo de uma variável aleatória.

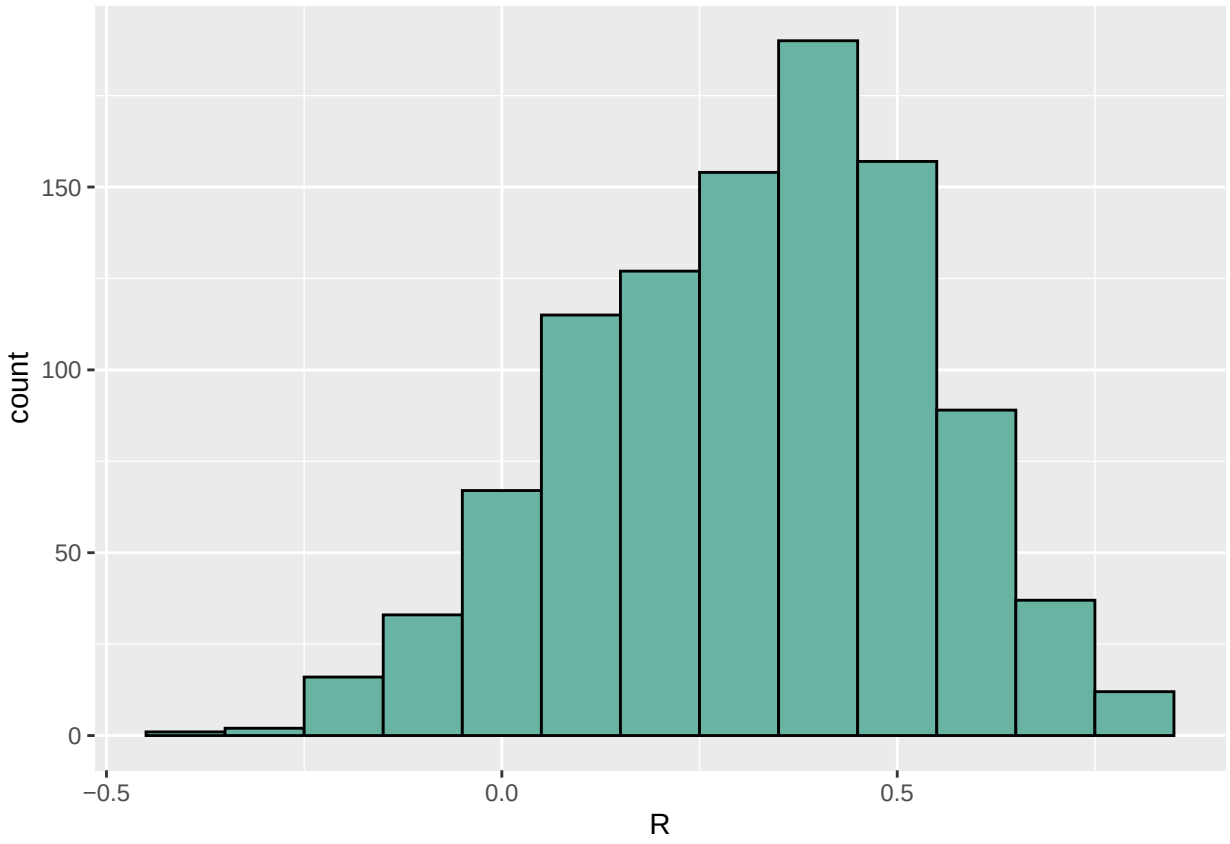
```
R <- sample_n(female_heights, 25, replace = TRUE) %>%
  summarize(r = cor(mother, daughter), n = n())
R
```

```
## # A tibble: 1 x 2
##       r     n
##   <dbl> <int>
## 1 0.332    25
```

R é uma variável aleatória. Nós podemos rodar a simulação de Monte Carlo para ver a sua distribuição.

```
B <- 10^3
n <- 25
R <- replicate(B, {
  sample_n(female_heights, n, replace = TRUE) %>%
    summarize(r = cor(mother, daughter)) %>%
    pull(r)
})
```

```
ggplot(data = NULL, aes(x = R)) +
  geom_histogram(binwidth = 0.1, fill = "#69b3a2", color = 'black')
```



Nós podemos ver que o valor esperado de R é o do a correlação populacional = 0.3187 e que tem um erro padrão de valor:

o erro do desvio padrão dos valores que a correlação pode assumir = 0.215815

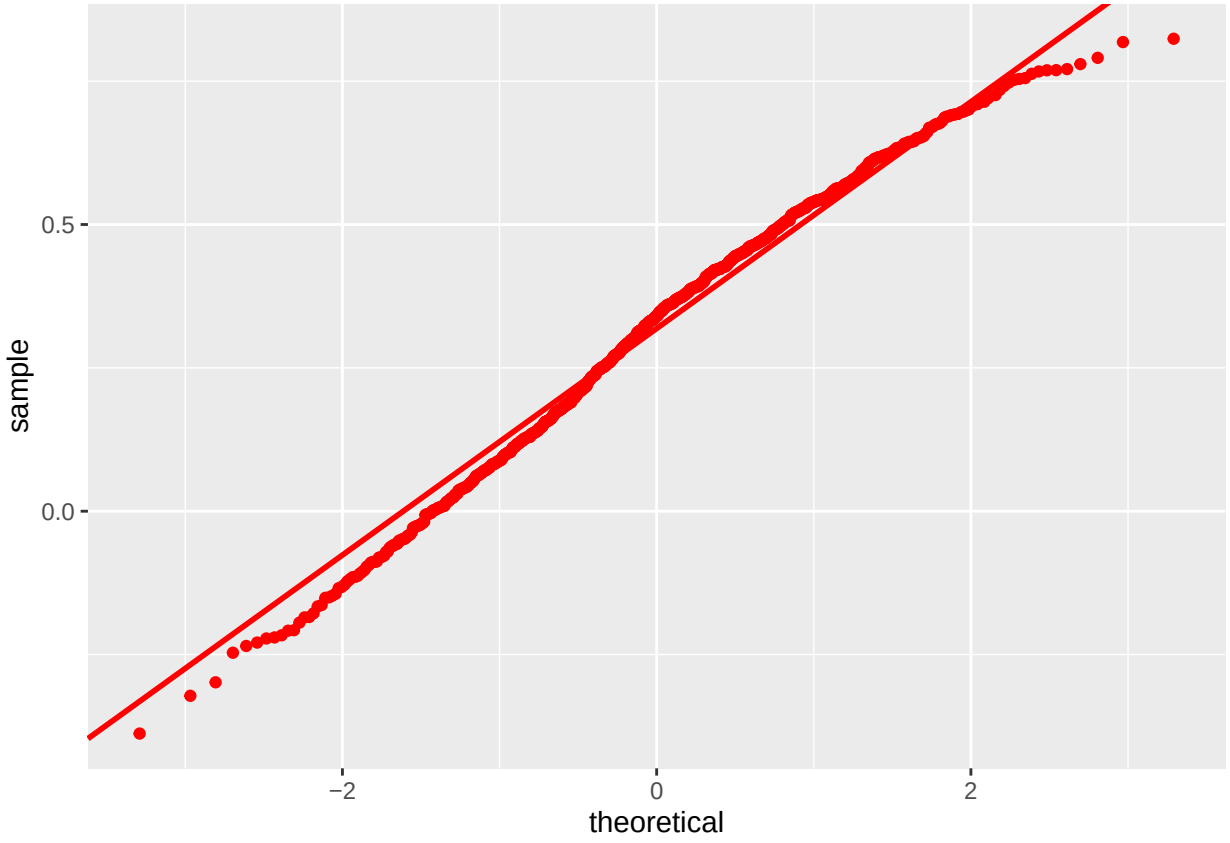
Lembremos que qdo interpretarmos correlações, que elas derivam de amostras então suas estimativas tem incertezas. Como a correlação amostral tem uma média *iid*, o teorema central do limite também se aplica. Portanto, para grandes valores de n, a distribuição de R é aproximadamente normal com valor esperado de ρ. O desvio padrão que é complexo de se derivar pode ser calculado por:

$$\sqrt{\frac{1-r^2}{n-2}}$$

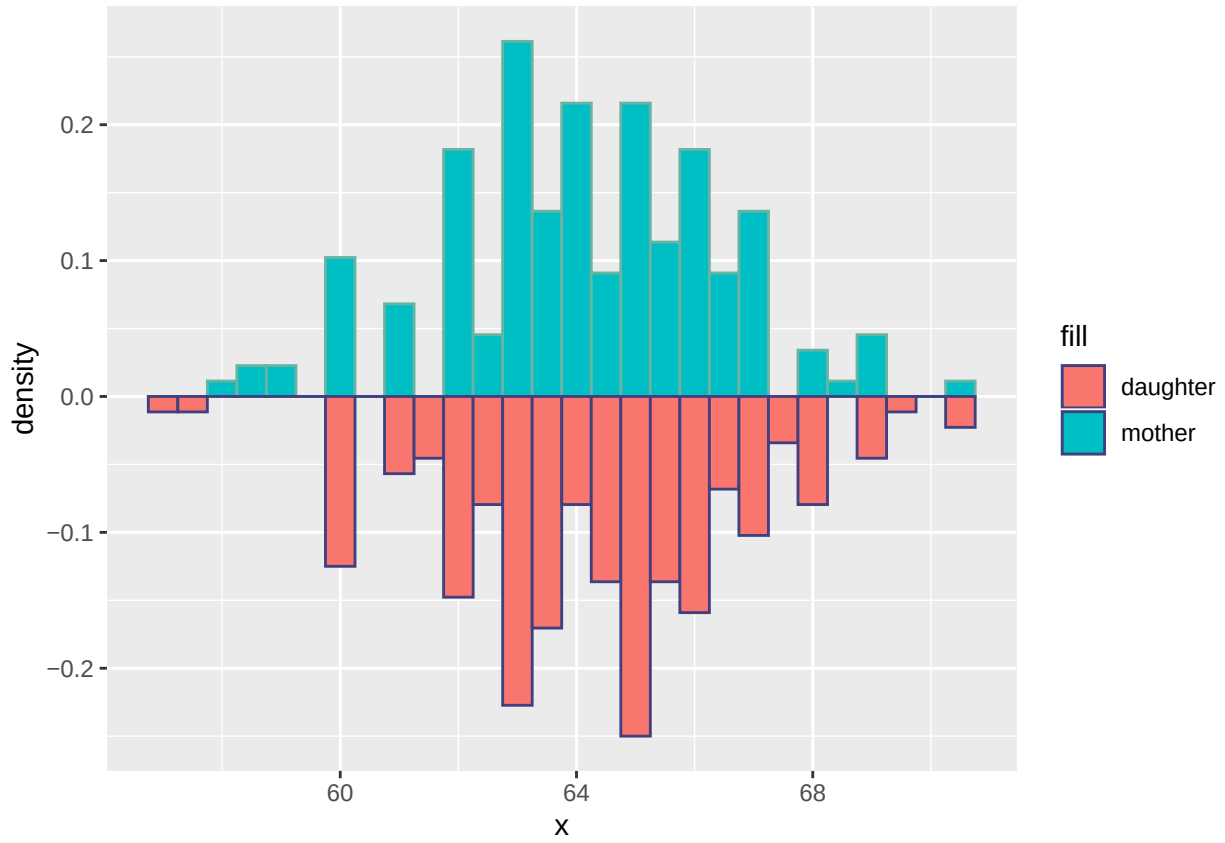
No nosso exemplo, anterior n = 25 não parece ser grande o suficiente para fazer aproximações boas:

```
data.frame(R) %>%
  ggplot(aes(sample = R)) +
```

```
  stat_qq(color = 'red') +
  geom_abline(color = 'red',size = 1, intercept = mean(R), slope = sqrt((1 - mean(R)^2)/(n - 2)))
```




```
p <- female_heights %>%
     ggplot(aes(x = x)) +
     geom_histogram(binwidth = 0.5, aes(x = mother, y = ..density.., fill = "mother"), color = "#69b3a2
     geom_histogram(binwidth = 0.5, aes(x = daughter, y = -..density.., fill = "daughter"), color = "#4

p
```




```r
sum(female_heights$mother == 60)
```

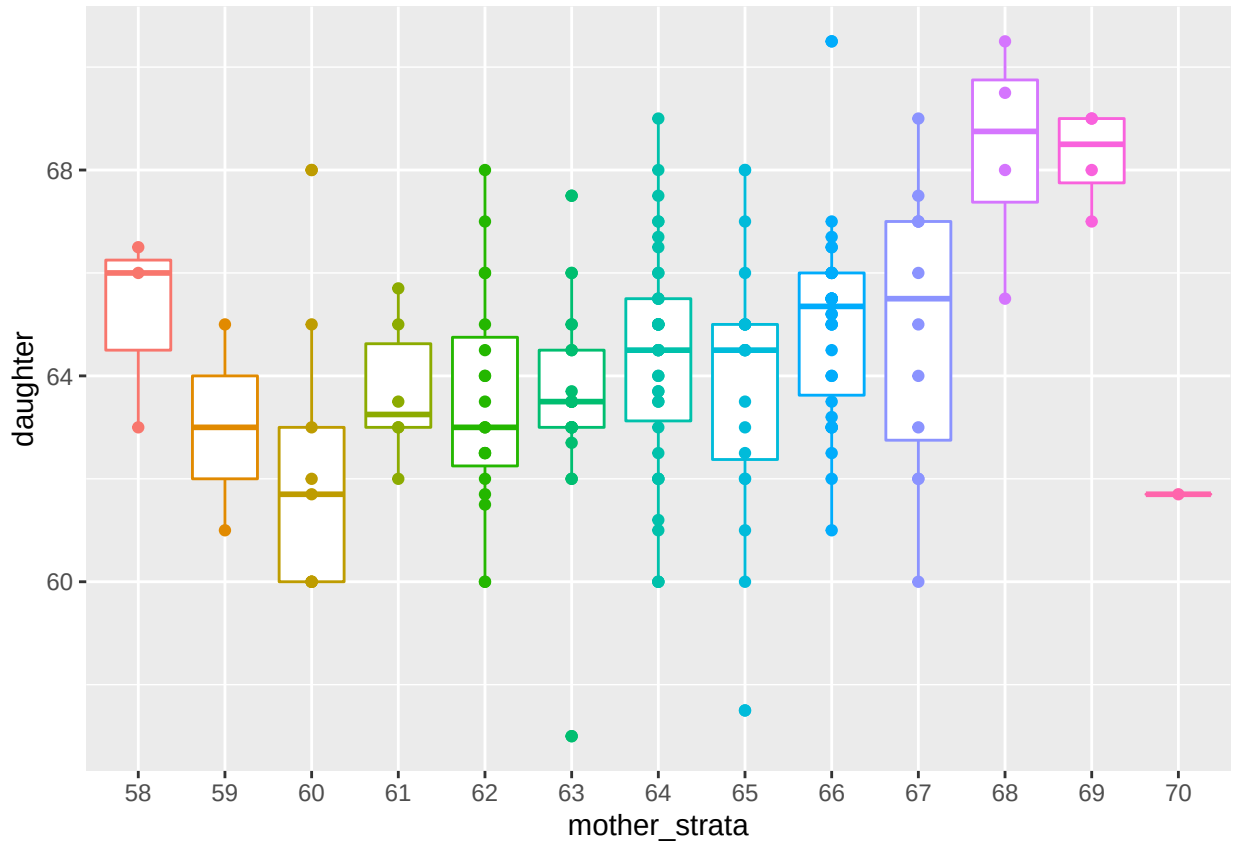
```
## [1] 9
```

Temos `r sum(female_heights$mother == 64)`, mães que tem exatamente 60 polegadas de altura. Condicionando as espectativas e definindo estratos com valores similares de altura, iguais a 60 inches.

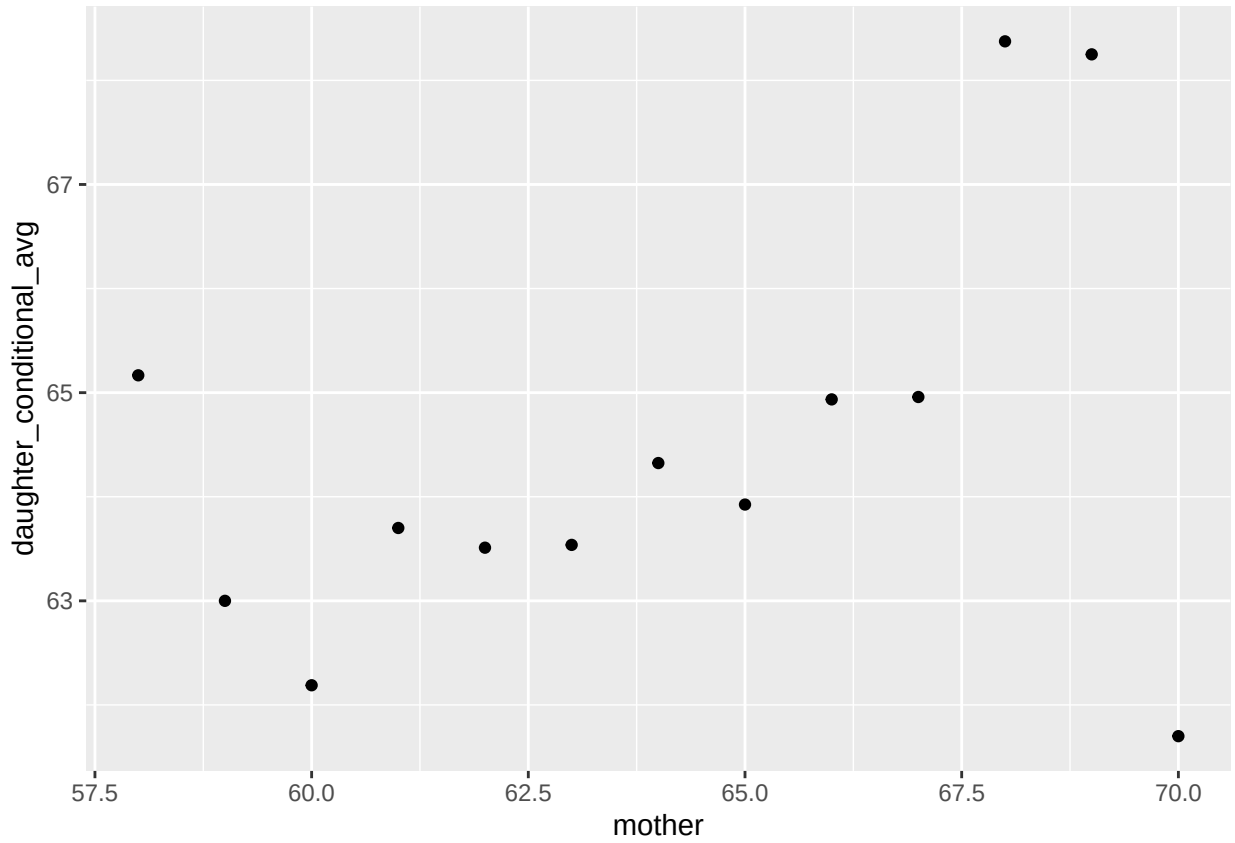
```r
conditional_avg <- female_heights %>%
  filter(round(mother) == 60) %>%
  summarize(avg = mean(daughter)) %>%
  pull(avg)

conditional_avg
```

```
## [1] 62.1889
```

```r
female_heights %>% mutate(mother_strata = factor(round(mother))) %>%
  ggplot(aes(mother_strata, daughter, color = mother_strata)) +
  geom_boxplot() +
  geom_point() + scale_color_discrete(guide = FALSE)
```




```
female_heights %>%
  mutate(mother = round(mother)) %>%
  group_by(mother) %>%
  summarize(daughter_conditional_avg = mean(daughter)) %>%
  ggplot(aes(mother, daughter_conditional_avg)) +
  geom_point()
```




```
# calculate values to plot regression line on original data

mu_x <- mean(female_heights$mother)
mu_y <- mean(female_heights$daughter)
s_x <- sd(female_heights$mother)
s_y <- sd(female_heights$daughter)
r <- cor(female_heights$mother, female_heights$daughter)
m <- r * s_y/s_x # slope
b <- mu_y - m*mu_x # intercept b

# outra maneira de calcular a correlação entre x e y
# do meu livro de machine learning R

ro <- cov(female_heights$mother, female_heights$daughter)/(s_x*s_y)
c(r, ro)
```

```
## [1] 0.32452 0.32452
```

```
c(m, b)
```

```
## [1]  0.339386 42.517012
```

```
var_rho <- (r^2)*100
var_rho
```

```
## [1] 10.5313
```

```
r*s_y/s_x
```

```
## [1] 0.339386
```

```
y <- b + m*60
y
```

```
## [1] 62.8801
```

O altura esperada da filha, pode ser calculada usando a seguinte equação:

$y = b + mx$ com a inclinação $m = \rho\frac{\sigma_y}{\sigma_x}$ e a intersecção $b = \mu_y - m\mu_x$