Projeto nr. 01 - Altura dos filhos versus altura dos pais

Vanderlei Kleinschmidt

06/10/2020

Dados utilizados por Francis Galton na análise da altura dos filhos em comparação à altura dos pais. A famosa comparação de Galton das alturas de 928 filhos adultos com as de seus 205 pares de pais (pai e mãe). Quando os pais são mais altos que a média, seus filhos tendem a ser mais baixos (ou seja, mais próximos da média) e quando os pais são mais baixos que a média, seus filhos tendem a ser mais altos. Galton chamou isso de "regressão à mediocridade".

Variáveis: parent = altura do pai/mãe em polegadas (amplitude 64 - 73) child = altura da criança em polegadas (amplitude 61.7 - 73.7)

Fonte:

Galton, F., "Regression Towards Mediocrity in Hereditary Stature," Journal of the Anthropological Institute of Great Britain and Ireland, 15, 246-263, 1886.

Os dados estão disponíveis no pacote UsingR.

Pacotes carregados para a análise a seguir.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

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

```
library(UsingR)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: HistData

## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units

##
## Attaching package: 'UsingR'

## The following object is masked from 'package:survival':
##
##     cancer
```

```
library(ggpubr)
```

Como os dados estão estão em polegadas e no Brasil trabalhamos com centímetros, o primeiro passo é converter os dados. Uma polegada equivale a aproximadamente 2,54cm.

```
galton <- 2.54 * galton

head(galton, 15)
```

```
##      child parent
## 1  156.718 179.07
## 2  156.718 173.99
## 3  156.718 166.37
## 4  156.718 163.83
## 5  156.718 162.56
## 6  157.988 171.45
## 7  157.988 171.45
## 8  157.988 171.45
```

```
## 9  157.988 168.91
## 10 157.988 168.91
## 11 157.988 168.91
## 12 157.988 163.83
## 13 160.528 179.07
## 14 160.528 176.53
## 15 160.528 173.99
```

Estatísticas descritivas

Optei por dividir as estatísticas descritivas em dois grupos:

1 - Medidas de tendência central: média, mediana, moda, valores mínimo e máximo e amplitude; 2 - Medidas de dispersão: desvio padrão, variância e coeficiente de varição.

1 - Medidas de tendência central: média, mediana, moda, valores mínimo e máximo e amplitude:

```
summary(galton$child)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   156.7   168.1   173.2   172.9   178.3   187.2
```

```
summary(galton$parent)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   162.6   171.4   174.0   173.5   176.5   185.4
```

```
range(galton$child)
```

```
## [1] 156.718 187.198
```

```
diff(range(galton$child)) # amplitude
```

```
## [1] 30.48
```

```
range(galton$parent)
```

```
## [1] 162.56 185.42
```

```
diff(range(galton$parent)) # amplitude
```

```
## [1] 22.86
```

Criando função para obter a moda

```
obter_moda <- function(x) {
  uniqv <- unique(x)
  uniqv[which.max(tabulate(match(x, uniqv)))]
}
```

Obtendo a moda

```
moda_child <- obter_moda(galton$child)
moda_child
```

```
## [1] 175.768
```

```
moda_parent <- obter_moda(galton$parent)
moda_parent
```

```
## [1] 173.99
```

Contagem de vezes que a moda aparece no dataset

```
filter(galton, child == 175.768) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   167
```

```
filter(galton, parent == 173.99) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   219
```

A amplitude da altura dos filhos é maior do que a amplitude da altura dos pais. Enquanto a altura dos pais varia entre 162,6cm e 185,4cm, a dos filhos varia entre 156,7cm 187,2cm. A altura média dos filhos é de 172,9cm enquanto a altura média dos pais é ligeiramente maior, 173,5Ocm. Analisando a mediana, vemos que a mediana dos pais é quase um centímetro maior que a dos filhos enquanto a moda dos filhos é quase dois centímetros maior. Porém, a altura de 175,768cm (moda) aparece 167 vezes para os filhos enquanto a altura de 173,99cm (moda) aparece 219 vezes para os pais.

2 - Medidas de dispersão: desvio padrão, variância e coeficiente de varição:

```
devPadPais = sd(galton$parent)
devPadFilhos = sd(galton$child)
varPais = var(galton$parent)
varFilhos = var(galton$child)
```

```
mediaPais = mean(galton$parent)
cvPais = (devPadPais/mediaPais)*100
mediaFilhos = mean(galton$child)
cvFilhos = (devPadFilhos/mediaFilhos)*100
```

Criando vetores e inserindo os dados

```
estatisticas = c("Desvio Padrão", "Variância", "Coeficiente de Variação")
child = c(round(devPadFilhos, digits = 2), round(varFilhos, digits = 2), round(cvFilhos, digits = 2))
parent = c(round(devPadPais, digits = 2), round(varPais, digits = 2), round(cvPais, digits = 2))
```

passo o nome dos vetores (sem precisar usar o "c")

```
dispersao = data.frame(estatisticas, child, parent)
dispersao
```

```
##              estatisticas child parent
## 1           Desvio Padrão   6.4   4.54
## 2               Variância  40.9  20.61
## 3 Coeficiente de Variação   3.7   2.62
```

Podemos também estimar uma medida da variância conjunta dos dados, através da covariância, e uma medida do grau de associação linear entre as variáveis, com o coeficiente de correlação.

Covariância

```
covar = cov(galton$parent, galton$child)
covar
```

```
## [1] 13.32007
```

Correlação

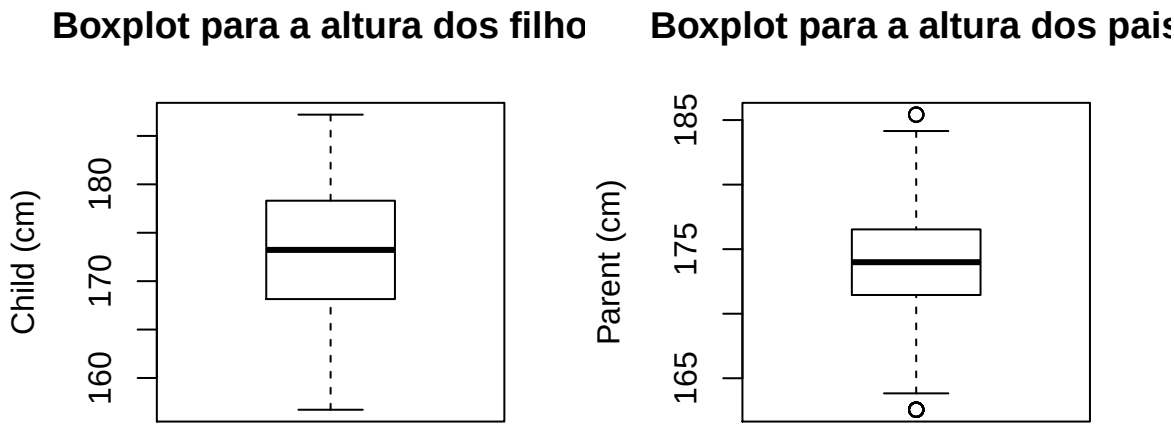
```
correl = cor(galton$parent, galton$child)
correl
```

```
## [1] 0.4587624
```

Analisando os resultados das medidas de dispersão podemos ver que a altura dos filhos tem uma variabilidade maior do que a dos pais em torno da média. O mesmo pode ser visto pelo coeficiente de variação, que é uma medida padronizada de dispersão. A covariância indica haver algum grau de variação conjunta positiva entre as varáveis, mas a medida de correlação, de 0,46, dá uma ideia de uma associação não muito forte entre elas.

Boxplot Quero comparar os dois boxplots lado a lado, por isso uso a função 'par'

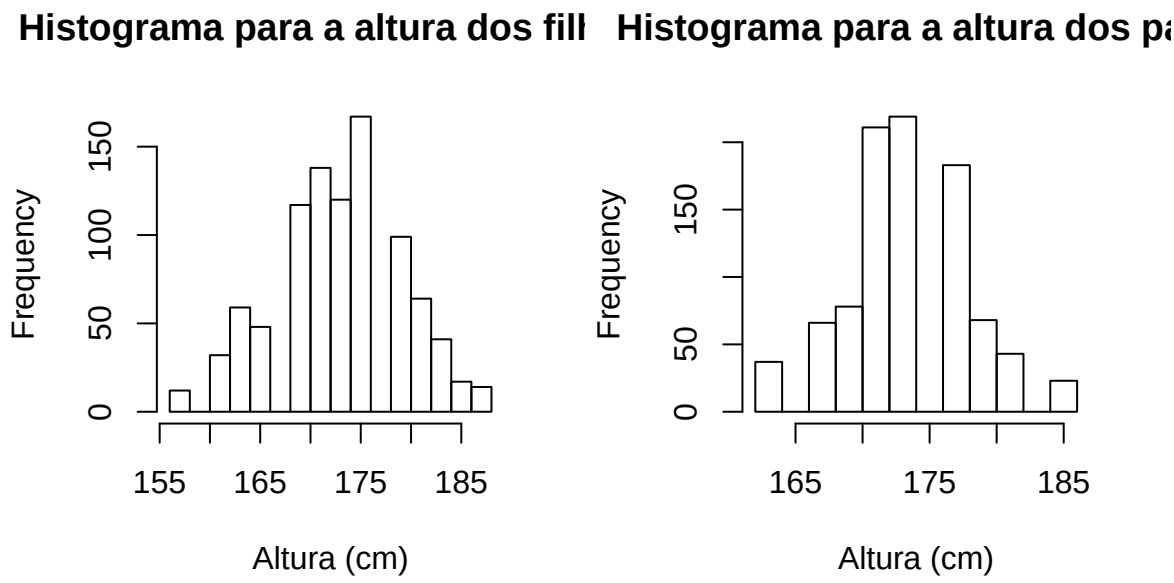
```
par(mfrow = c(1, 2), oma = c(4, 1, 1, 1))
boxplot(galton$child, main = "Boxplot para a altura dos filhos", ylab = "Child (cm)")
boxplot(galton$parent, main = "Boxplot para a altura dos pais", ylab = "Parent (cm)")
```



Não conseguimos identificar dados discrepantes na variável altura dos filhos, mas no caso da altura dos pais parece haver valores extremos tanto na borda superior quanto inferior. Não vamos nos preocupar com esses outliers por enquanto, acredito que eles não irão afetar o resultado do modelo que irei estimar na sequência.

Histograma

```
par(mfrow = c(1, 2), oma = c(4, 1, 1, 1))
hist(galton$child, main = "Histograma para a altura dos filhos", xlab = "Altura (cm)")
hist(galton$parent, main = "Histograma para a altura dos pais", xlab = "Altura (cm)")
```



O histograma mostra novamente os dados extremos da altura dos pais. Ele mostra também como os dados são mais dispersos no caso da distribuição da altura dos filhoes, comparativamente à distribuição da altura dos pais, como se pode observar pelas medidas de dispersão.

Quero analisar os dados em forma de categoria, para poder definir uma estatura relativa para cada variável.

```
table(galton$child)
```

```
##
## 156.718 157.988 160.528 163.068 165.608 168.148 170.688 173.228 175.768 178.308
##       5       7      32      59      48     117     138     120     167      99
## 180.848 183.388 185.928 187.198
##      64      41      17      14
```

```
table(galton$parent)
```

```
##
## 162.56 163.83 166.37 168.91 171.45 173.99 176.53 179.07 181.61 184.15 185.42
##     14     23     66     78    211    219    183     68     43     19      4
```

Calculando a proporção de cada categoria

```
model_table1 <- table(galton$child)
model_table2 <- table(galton$parent)
prop.table(model_table1) #retorna a proporção % de filhos em cada categoria de altura
```

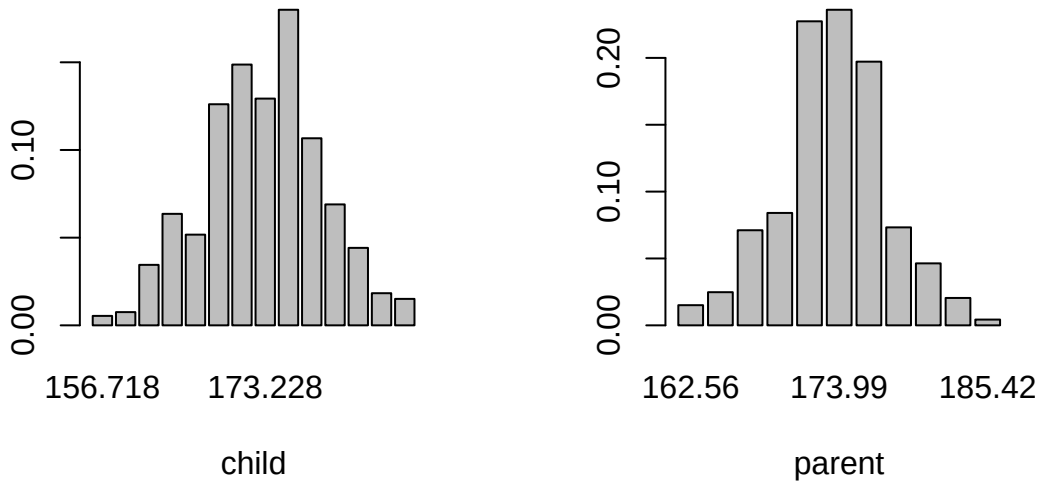
```
##
##     156.718     157.988     160.528     163.068     165.608     168.148
## 0.005387931 0.007543103 0.034482759 0.063577586 0.051724138 0.126077586
##     170.688     173.228     175.768     178.308     180.848     183.388
## 0.148706897 0.129310345 0.179956897 0.106681034 0.068965517 0.044181034
##     185.928     187.198
## 0.018318966 0.015086207
```

```
prop.table(model_table2) #retorna a proporção % de pais em cada categoria de altura
```

```
##
##      162.56      163.83      166.37      168.91      171.45      173.99
## 0.015086207 0.024784483 0.071120690 0.084051724 0.227370690 0.235991379
##      176.53      179.07      181.61      184.15      185.42
## 0.197198276 0.073275862 0.046336207 0.020474138 0.004310345
```

Graficamente talvez seja melhor para entender a diferença entre as séries

```
par(mfrow = c(1, 2), oma = c(4, 1, 1, 1))
barplot(prop.table(table(galton$child)), xlab = "child")
barplot(prop.table(table(galton$parent)), xlab = "parent")
```



Vou criar duas novas colunas (variáveis categóricas), onde eu classifico os indivíduos em baixo, mediano, alto e muito alto, usando como critério os quartis definidos anteriormente nas estatísticas descritivas, que se mostraram mais adequados. Porém, eu poderia também usar a "prop.table" pra isso, mas não quero fazer isso nesse momento.

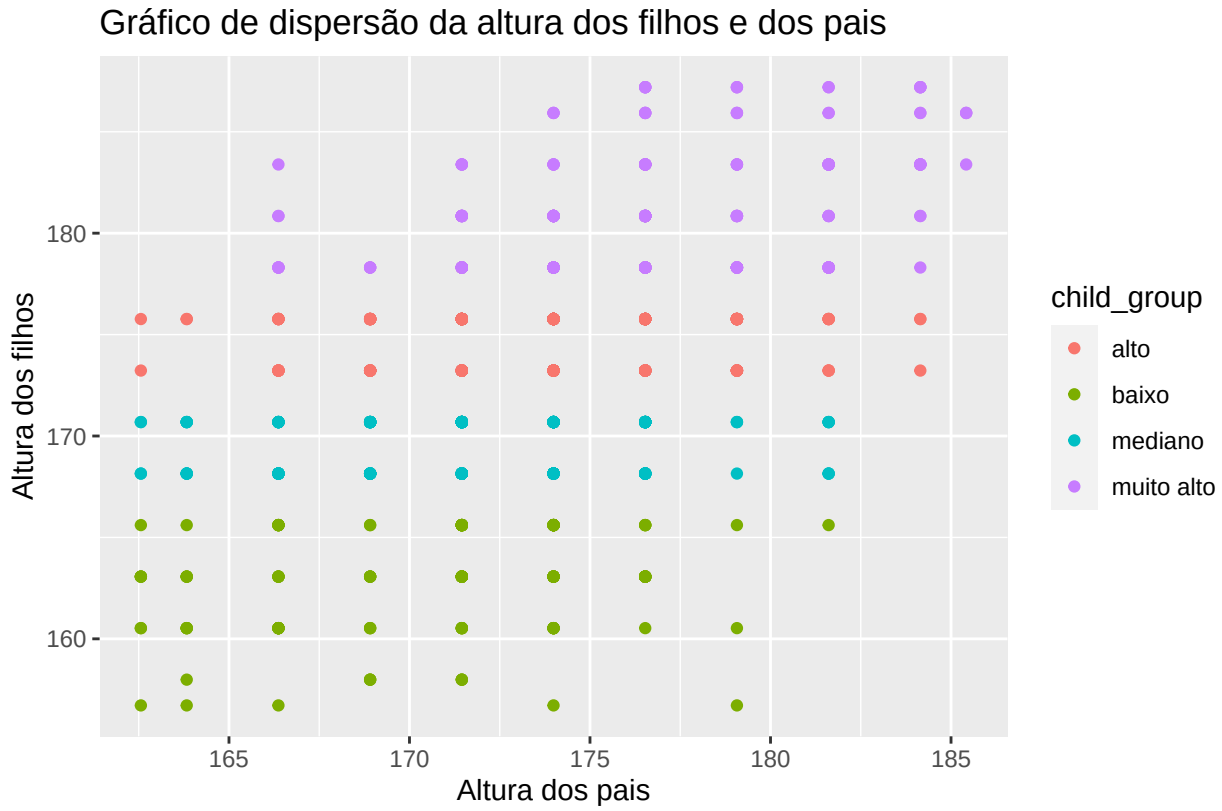
```r
group_child <- function(child){
  if (child >= 156 & child <= 168){
    return('baixo')
  }else if(child > 168 & child <= 173){
    return('mediano')
  }else if (child > 173 & child <= 178){
    return('alto')
  }else if (child > 178){
    return('muito alto')
  }
}

group_parent <- function(parent){
  if (parent >= 162 & parent <= 171){
    return('baixo')
  }else if(parent > 171 & parent <= 174){
    return('mediano')
  }else if (parent > 174 & parent <= 176){
    return('alto')
  }else if (parent > 176){
    return('muito alto')
  }
}


# ajusta essas transformações no dataset
galton$child_group <- sapply(galton$child, group_child)
galton$child_group <- as.factor(galton$child_group)

galton$parent_group <- sapply(galton$parent, group_parent)
galton$parent_group <- as.factor(galton$parent_group)

head(galton, 15)
```

```
##      child parent child_group parent_group
## 1  156.718 179.07       baixo   muito alto
## 2  156.718 173.99       baixo      mediano
## 3  156.718 166.37       baixo        baixo
## 4  156.718 163.83       baixo        baixo
## 5  156.718 162.56       baixo        baixo
## 6  157.988 171.45       baixo      mediano
## 7  157.988 171.45       baixo      mediano
## 8  157.988 171.45       baixo      mediano
## 9  157.988 168.91       baixo        baixo
## 10 157.988 168.91       baixo        baixo
## 11 157.988 168.91       baixo        baixo
## 12 157.988 163.83       baixo        baixo
## 13 160.528 179.07       baixo   muito alto
## 14 160.528 176.53       baixo   muito alto
## 15 160.528 173.99       baixo      mediano
```

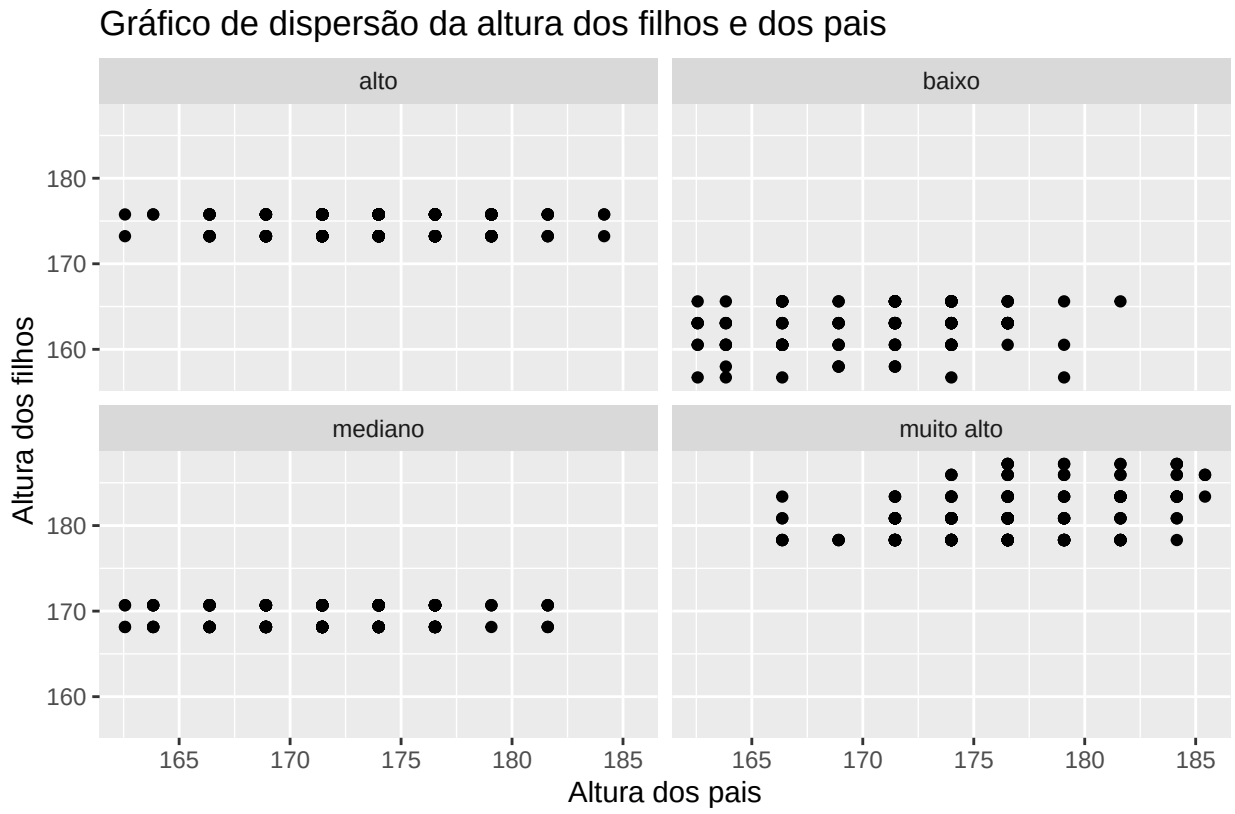
Nos gráficos gerados a partir da categorização é possível observar que a medida que a altura dos pais aumenta, a altura dos filhos também aumenta, mas não na mesma proporção. Isso corrobora com a tese de Galton

sobre a regressão à mediocridade, ou seja, a tendência dos filhos terem altura mais próximas da média.

```
ggplot(data = galton) +
  geom_point(mapping = aes(x = parent, y = child, color = child_group)) +
  labs(title="Gráfico de dispersão da altura dos filhos e dos pais",
       y="Altura dos filhos",x="Altura dos pais", caption="")
```



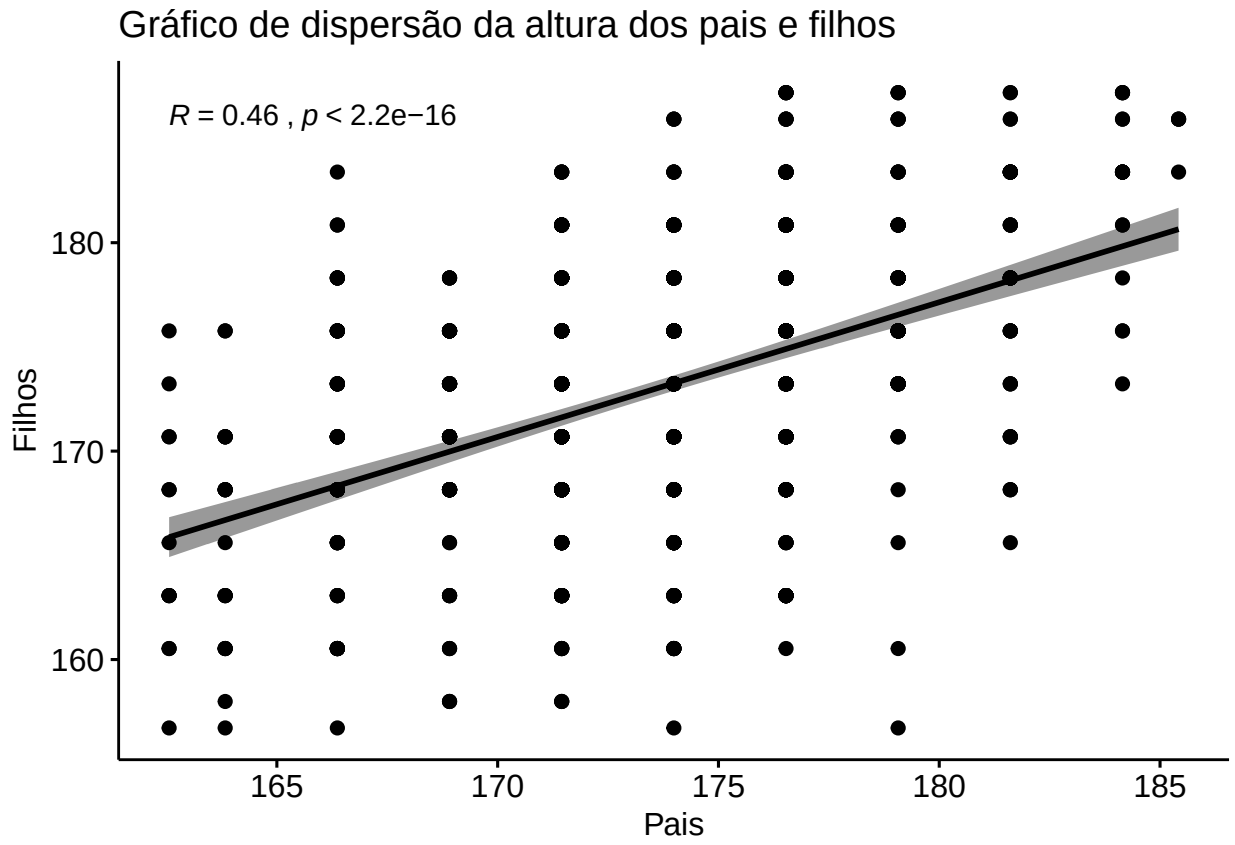

```
ggplot(data = galton) +
  geom_point(mapping = aes(x = parent, y = child)) +
  facet_wrap(~child_group) +
  labs(title="Gráfico de dispersão da altura dos filhos e dos pais",
       y="Altura dos filhos",x="Altura dos pais", caption="")
```



O gráfico de disperãoa abaixo mostra que à medida em que a média da altura dos pais aumenta, a média da altura dos filhos almenta também, indicando a possibilidade de termos uma correlação positiva entre essas duas variáveis.

```
ggscatter(galton, x = "parent", y = "child",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "Pais", ylab = "Filhos", title = "Gráfico de dispersão da altura dos pais e filhos")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Entrando na reta final, Vou dividir a amostra em dois grupos, sendo 70% para treinamento e 30% para teste. A partir daí eu treino o modelo e verifico a sua acurácia ou capacidade de prever a altura dos filhos, tendo como base apenas a altura dos pais (uma espécie de transferência de altura, ou hereditariedade).

```
linhas <- sample(1:nrow(galton), 0.7 * nrow(galton))
dados_treino <- galton[linhas,]
dados_teste <- galton[-linhas,]

head(dados_treino, 15)
```

```
##       child parent child_group parent_group
## 517 173.228 168.91        alto        baixo
## 704 178.308 181.61  muito alto   muito alto
## 758 178.308 173.99  muito alto      mediano
## 271 170.688 181.61     mediano   muito alto
## 155 168.148 179.07     mediano   muito alto
## 463 173.228 173.99        alto      mediano
## 393 170.688 166.37     mediano        baixo
## 239 168.148 168.91     mediano        baixo
## 138 165.608 171.45       baixo      mediano
## 248 168.148 168.91     mediano        baixo
## 326 170.688 173.99     mediano      mediano
## 708 178.308 179.07  muito alto   muito alto
## 732 178.308 176.53  muito alto   muito alto
## 490 173.228 171.45        alto      mediano
## 492 173.228 171.45        alto      mediano
```

```r
head(dados_teste, 15)
```

```
##      child parent child_group parent_group
## 1  156.718 179.07       baixo   muito alto
## 9  157.988 168.91       baixo        baixo
## 15 160.528 173.99       baixo      mediano
## 18 160.528 173.99       baixo      mediano
## 19 160.528 173.99       baixo      mediano
## 22 160.528 171.45       baixo      mediano
## 28 160.528 168.91       baixo        baixo
## 29 160.528 168.91       baixo        baixo
## 32 160.528 166.37       baixo        baixo
## 35 160.528 166.37       baixo        baixo
## 37 160.528 166.37       baixo        baixo
## 41 160.528 163.83       baixo        baixo
## 43 160.528 162.56       baixo        baixo
## 54 163.068 176.53       baixo   muito alto
## 55 163.068 176.53       baixo   muito alto
```

Eu gosto de analisar o resultado de um modelo estimado, mesmo que seja com dados de treino. Isso me ajuda a verificar se o modelo está atendendo àquilo que se espera dele, a priori. No caso deste estudo, o que eu espero encontrar é uma relação positiva e significativa entre a variável dependente, child, e a variável explicativa, parent. Isso significa que o coeficiente estimado "parent" tem que ter sinal positivo e tem que ser estatisticamente significativo. Para treinar o modelo de regressão eu uso a função "lm".

```r
modelo1 <- lm(child ~ parent, data = dados_treino)
summary(modelo1)
```

```
##
## Call:
## lm(formula = child ~ parent, data = dados_treino)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.6815  -3.5697   0.6521   4.0503  12.5285
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 58.19617    8.30401   7.008 6.08e-12 ***
## parent       0.66213    0.04787  13.832  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.582 on 647 degrees of freedom
## Multiple R-squared:  0.2282, Adjusted R-squared:  0.227
## F-statistic: 191.3 on 1 and 647 DF,  p-value: < 2.2e-16
```

Uma vez que o modelo tenha sido treinado, podemos ver que o sinal dos coeficientes estimados estão de acordo com o que se esperava. Ou seja, o intercepto é positivo, até mesmo porque não faria sentido ele ser negativo. Ele representa a soma de todas as variáveis que de alguma forma influenciam a altura dos filhos de forma significativa, mas que não estamos considerando neste modelo. Em termos numéricos, sem considerar a altura dos pais, os filhos terão pelo menos 57cm de altura. É estranho pensar assim, mas podemos pensar

também que os pais não são capazes de influenciar pelo menos 57cm da altura dos filhos! O coeficiente angular, determina a capacidade de transferência da altura dos pais para os filhos. Para cada centímetro a mais de altura dos pais, os filhos terão 0,67cm a mais de altura. Note que não é proporcional, mas é positivo. Já tínhamos visto acima, pelas outras estatísticas, que haveria uma relação positiva entre a altura dos pais e dos filhos, mas que essa relação não seria muito forte. Isso significa que existem outros fatores que afetam a altura dos filhos, e que não é apenas a hereditariedade. Do ponto de vista da significância estatística, os dois coeficientes estimados são estatisticamente significativos, ao nível de 1% de significância estatística. A probabilidade de se cometer um erro do tipo 1 é praticamente nulo. Olhando agora o R quadrado, vemos que o nosso modelo é capaz de explicar apenas 22,32% da altura dos filhos. Como dito anteriormente, há outras variáveis que não foram consideradas neste estudo e que afetam de forma significativa a variável dependente. Em termos práticos, a capacidade preditiva do modelo é bem fraca!

Vamos prosseguir com a análise, apenas a título de exercício, tendo em vista a qualidade dos resultados apresentados.

```
previsoes <- predict(modelo1, dados_teste)
summary(previsoes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   165.8   171.7   173.4   173.2   175.1   181.0
```

```
summary(galton$child)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   156.7   168.1   173.2   172.9   178.3   187.2
```

Vamos dar uma olhada rápida no resumo estatístico dos resultados previstos comparativamente com os dados de treino. O modelo super estima os valores mínimos e subestima os valores máximos da série de dados.

Tanto a média quanto a mediana são previstos de forma quase precisa, o que é bem interessante.

Visualizando os valores previstos e observados

```
resultados <- cbind(previsoes, dados_teste$child)
colnames(resultados) <- c('Previsto','Real')
resultados <- as.data.frame(resultados)
min(resultados)
```

```
## [1] 156.718
```

```
max(resultados)
```

```
## [1] 187.198
```

```
head(resultados, 15)
```

```
##    Previsto    Real
## 1  176.7631 156.718
## 9  170.0359 157.988
## 15 173.3995 160.528
## 18 173.3995 160.528
## 19 173.3995 160.528
```

```
## 22 171.7177 160.528
## 28 170.0359 160.528
## 29 170.0359 160.528
## 32 168.3541 160.528
## 35 168.3541 160.528
## 37 168.3541 160.528
## 41 166.6723 160.528
## 43 165.8314 160.528
## 54 175.0813 163.068
## 55 175.0813 163.068
```