

## Aula 3b

James Hunter, Ph.D.

Professor, Retrovirologia, UNIFESP

28 de maio de 2020

- Simulação é uma técnica central da ciência dos dados
- for loops
- números aleatórios em R: `runif()` e `rnorm()`
- mais usos para os verbos de `dplyr`

# Probabilidade e Distribuições dos Dados

- Não vou repetir um curso de estatística
- Mas, só rever alguns conceitos básicos

- Probabilidade é um número sem unidade entre 0 e 1
- São limites absolutos

- Queremos que as vendas camiseta variem entre .15 e .40 do publico
  - ▶ Como fazer?
  - ▶ Vendas tornará variável aleatória
- Distribuições formais dão um conceito de como as variáveis aleatórias são distribuídas em realidade
- Podemos usar uma distribuição para definir os valores das variáveis aleatórias

# Distribuições Contínuas Mais em Uso

- ① Uniforme
- ② Normal (Gaussiana)

# Distribuições Teóricas e Distribuições Empíricas

- A distribuição segue os dados
  - ▶ Os dados fazem uma distribuição
- Distribuições teóricas
  - ▶ Os valores de demanda vão seguir a equação da distribuição

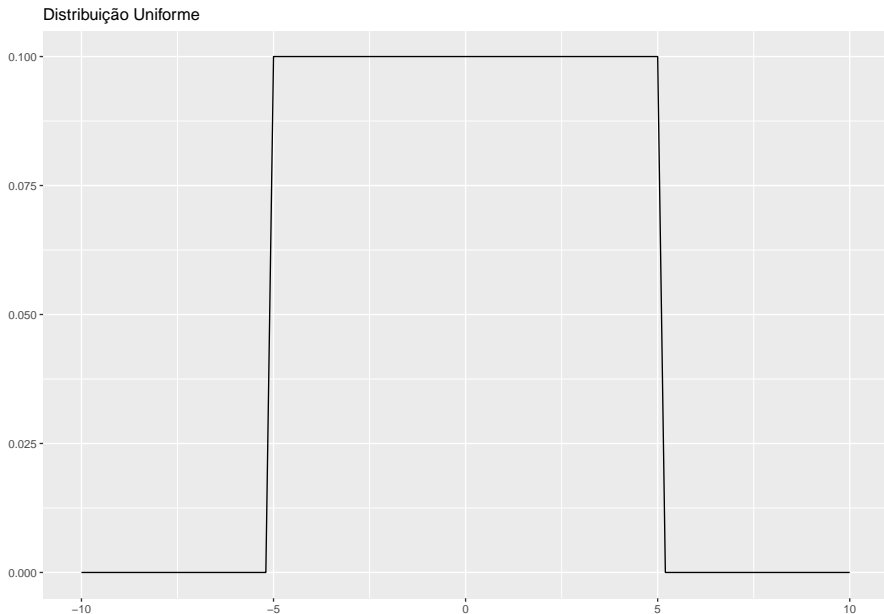
# Distribuição Uniforme

- Todos os valores num intervalo tem chance igual de aparecer
- Intervalo definido por um limite inferior e limite superior
- Fora do intervalo, probabilidade de um número ser selecionado é 0

$$p(x) = \begin{cases} \frac{1}{b-a} & \text{para } a \leq x \leq b \\ 0 & \text{para qualquer outro valor} \end{cases}$$



# Gráfico da Distribuição Uniforme



# Valores Aleatórios da Distribuição Uniforme

```
set.seed(42)  
runif(10, min = 0, max = 10)
```

```
## [1] 9.148060 9.370754 2.861395 8.304476 6.417455 5.190959  
## [9] 6.569923 7.050648
```

# Não São Números Inteiros

- Para fazer esses números números inteiros
  - ▶ Precisa usar `floor` e modificar os limites para permitir a gama completa que quer
  - ▶ aqui  $\text{min} = 8$ ,  $\text{max} = 12$

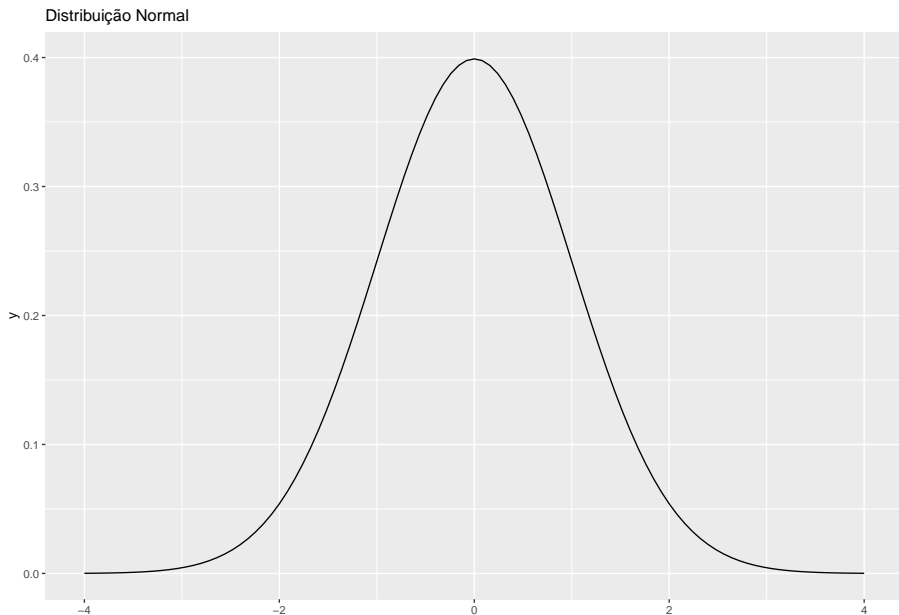
```
set.seed(42)
floor(runif(10, min = 8, max = 12.999))
```

```
## [1] 12 12 9 12 11 10 11 8 11 11
```

- A famosa curva de sino
- Teorema de Limite Central
- Especifica não com limites, mas com **parâmetros**
  - ▶ **média** ( $\mu$ )
  - ▶ **desvio padrão** ( $\sigma$ )

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

# Gráfico da Distribuição Normal



# Números de Compras com a Distribuição Normal

- Compras em ano1 com público de 4.000 (10 amostras)

```
set.seed(42)
publico <- 4000
interval <- seq(from = 0.15, to = 0.4, by = 0.001)
floor(publico * rnorm(10, mean = mean(interval),
                     sd = sd(interval)))
```

```
## [1] 1498 936 1205 1283 1217 1069 1538 1072 1686 1081
```

- Limites não são fixos

# Quando Usar Cada Um

- **Uniforme** (`unif`)
  - Quando você quer que todos os números têm uma chance igual de ser selecionados
- **Normal** (`norm`)
  - ▶ Quando você sabe que os valores são distribuídos normalmente
  - ▶ Quando você tem um grande número de valores que não têm outra distribuição conhecida
- Teorema de Limite Centrale

# Duas Outras Distribuições Importantes

- Ambas *discretas*
- Famílias de distribuições



- Probabilidade de número de sucessos numa sequência de  $n$  tentativas
- Ex., jogar uma moeda
- Quando você tem só dois resultados possíveis
- TRUE/FALSE
- Distribuição `binom` em R
- Testes de Proporção

- Probabilidade de uma série dos eventos ocorrer num certo período de tempo
- Pode generalizar o uso para a contagem dos eventos ou objetos
- Ex., você quer saber quantas vendas são feitas por hora
- Um parâmetro:  $\lambda$  - Número esperado de ocorrências
- Distribuição pois em R

# Porque Distribuições São Importantes

- Descrevem curvas teóricas que usamos para comparar aos dados
- “*Machine Learning is Glorified Multidimensional Curve Fitting*” - Michael Levitt, Prêmio Nobel de Química
- Testes de estatística são a comparação dos dados amostrais às curvas das distribuições
- Avalia qual é a probabilidade ( $p$ ) que os dados espelham perfeitamente a distribuição
- Se  $p$  for abaixo de um certo valor  $\alpha$ , consideramos os dados significativos
- *Significativo* – não suportam uma hipótese arbitrária (“hipótese nula”)

# Este Não É Uma Aula de Estatística

- Pulamos diretamente aos modelos

- Tirar conclusões sobre uma população baseado numa amostra
- Mesma ideia atrás de inferência em estatística
- Modelos de *machine learning* como grandes modelos estatísticos
- Também existem modelos de simulação
  - ▶ Replicar com matemática um processo cujas dimensões e regras são bem compreendidas

# Estatística vs. *Machine Learning*

- Testes estatísticos tendem de ser mais simples para aplicar e analisar
- Modelos de *machine learning* podem formar estruturas e previsões mais sofisticadas
- *Machine learning* mais certo que os modelos estatísticos?
  - ▶ Estudo de 2018 diz que não têm resultados melhores<sup>1</sup>
- Decisão para usar um modelo de *machine learning* depende de:
  - ▶ Necessidade
  - ▶ Sofisticação do modelo para estudo
  - ▶ Tamanho do amostra
  - ▶ Habilidade e experiência do pesquisador com o algoritmo de *machine learning*

---

<sup>1</sup>Makridakis S, Spiliotis E, Assimakopoulos V. Statistical and Machine Learning forecasting methods: Concerns and ways forward. PLoS ONE. 2018 Mar 27;13(3):e0194889.

- Existe uma variável dependente
  - ▶ **Supervisionado**
    - ★ Supervisão por causa que o resultado do modelo pode ser avaliado em termos da realidade dos resultados observados
  - ▶ 2 subtipos
    - ★ **Classificação** - Colocar cada caso em um grupo baseado no valores das variáveis independentes
    - ★ **Regressão** - Determinar um valor de uma combinação das variáveis independentes
- Não existe uma variável dependente
  - ▶ **Não-supervisionado**
    - ★ Explorar a estrutura dos casos e tentar agrupar eles em *clusters* dos casos
    - ★ **Análise dos Clusters**

# Regressão Linear Simples



- Termo vem de eugenismo (*eugenics*) de Sir Francis Galton.
- Estudou alturas de famílias
- Observou que crianças de pais altos tendiam de ser mais baixas de que os pais e crianças de pais baixos tendiam de ser mais altas
- Chamou a tendência **regressão à média**
- Usaremos esses dados clássicos

# Método de Mínimos Quadrados

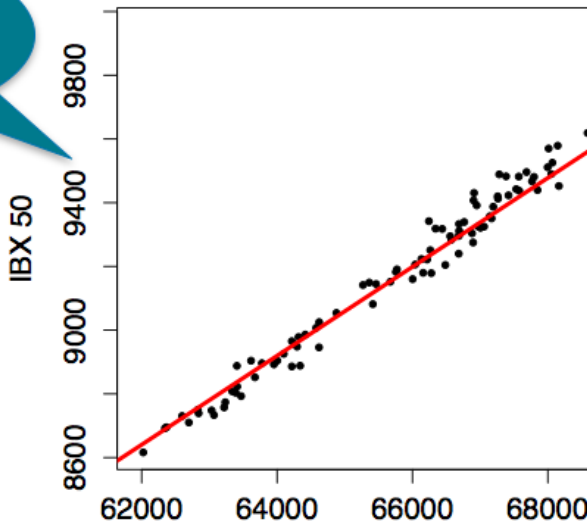
- Solucionamos com o método *Mínimos Quadrados*
  - ▶ Inventado por Carl Friedrich Gauss (1777 - 1855)
  - ▶ Método minimiza as divergências entre os valores lineares previstos e os valores dos dados
  - ▶ Consegue o melhor relação entre a variável de resultado e as variáveis prognósticas
- Por enquanto, vamos restringir o modelo para forma linear
  - ▶ Outras formas existem

*Prever um resultado numa variável dependente baseado em uma ou mais variáveis independentes*

- Uma – regressão linear *simples*
- Mais – regressão linear *múltipla*

**Variável  
Dependente**

**IBOVESPA e IBX 50 e**



$$y = \beta_0 + \beta_1 x$$

- $\beta_1$  = inclinação da linha (*slope*)
- $\beta_0$  = intercepto (onde cruza o eixo  $y$ )
- Os dois parâmetros da regressão
- Com estes parâmetros, Mínimos Quadrados acha a reta que melhor prevê o valor da variável dependente dado o valor de independente

# “Melhor” Quer Dizer “Bom”?

- Apesar de ser a melhor maneira de prever  $y$ , possível que não descreve bem  $y$
- **Bom** depende dos dados
- **Melhor** depende do método

# Equação de Regressão

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

- $Y_i$  = valor de variável dependente
- $\beta_0$  = intercepto
- $\beta_1$  = inclinação da reta de regressão
- $X_i$  = valor da variável independente
- $\epsilon_i$  = termo de erro de cada caso

# Equação de Regressão - Estimação

$$\hat{Y}_i = b_0 + b_1 X_i + e_i$$

- $\hat{Y}_i$  = valor de variável dependente (estimado)
- $b_0$  = intercepto
- $b_1$  = inclinação da reta de regressão
- $X_i$  = valor da variável independente
- $e_i$  = termo de erro de cada caso



# Termo de Erro ( $\epsilon$ )

- Também chamado **resíduo**
- Responsável pela variabilidade em  $y$  que a reta não consegue explicar

- Faz o cálculo que minimiza o quadrado da soma dos erros
- Erros = resíduos = diferenças entre o valor *observado* e o valor *esperado*

$$\min \sum (y_i - \hat{y}_i)^2$$

- $y_i$  = valor observado da variável dependente
- $\hat{y}_i$  = valor estimado da variável dependente

# Basta de Teoria – Exemplo

- A base de dados de Galton sobre altura nas famílias
- Pergunta é se filhos/as são mais altos ou mais baixos de que os pais
- Mediu 898 filhos/as em 197 famílias
- Base de dados originais (em papel) fica na University College, London (UCL)

```
galton <- readRDS(here::here("galton.rds"))  
str(galton)
```

```
## 'data.frame':    898 obs. of  6 variables:  
## $ family: Factor w/ 197 levels "1","10","100",...: 1 1 1 1  
## $ father: num  78.5 78.5 78.5 78.5 75.5 75.5 75.5 75.5 75  
## $ mother: num  67 67 67 67 66.5 66.5 66.5 66.5 64 64 ...  
## $ sex    : Factor w/ 2 levels "F","M": 2 1 1 1 2 2 1 1 2 1  
## $ height: num  73.2 69.2 69 69 73.5 72.5 65.5 65.5 71 68  
## $ nkids  : int   4 4 4 4 4 4 4 4 2 2 ...
```

- height, father, mother todos medem altura em polegadas

## Foco em Pais e Filhos

```
boys <- galton %>%
  filter(sex == "M") %>%
  select(-family, -mother, -sex, -nkids)
glimpse(boys)
```

```
## Rows: 465
```

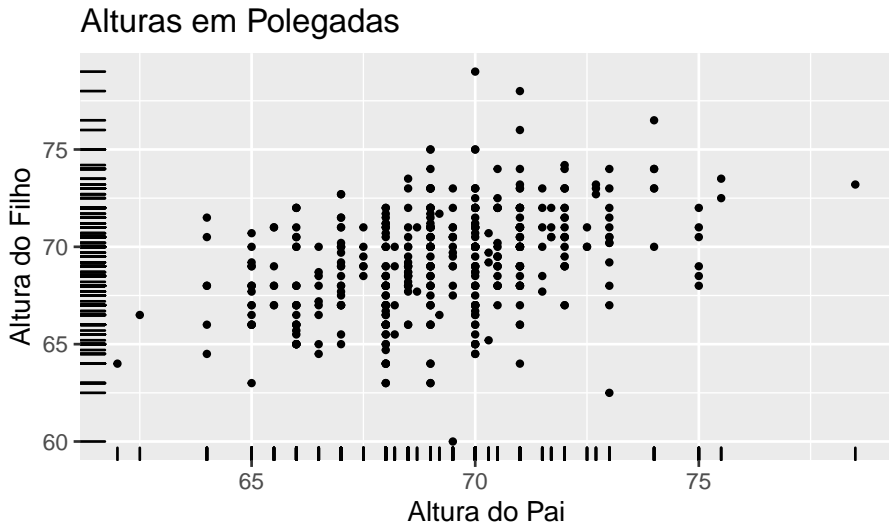
```
## Columns: 2
```

```
## $ father <dbl> 78.5, 75.5, 75.5, 75.0, 75.0, 75.0, 75.0, 75
```

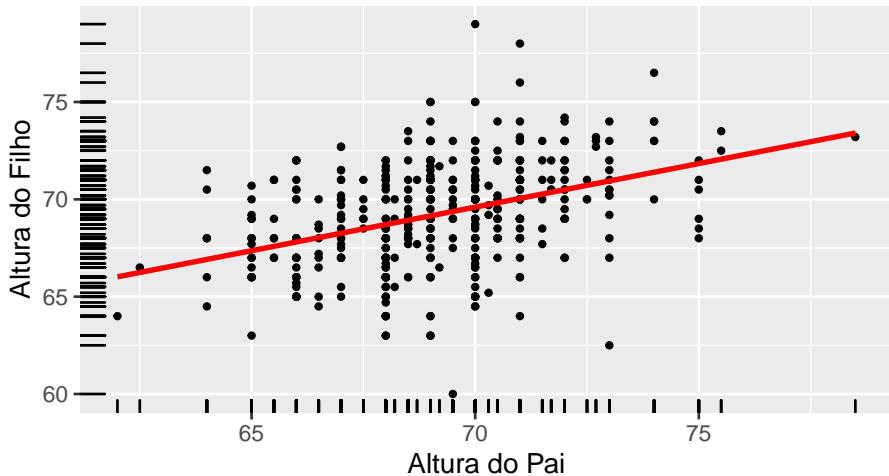
```
## $ height <dbl> 73.2, 73.5, 72.5, 71.0, 70.5, 68.5, 72.0, 69
```

- father é a variável independente
- height é a variável dependente
- Queremos ver se a altura do pai prevê a altura do filho

# Pai/Filho – Gráfico de Dispersão



## Alturas em Polegadas



# O Que Podemos Dizer Agora?

- **Parece** que mais altos os pais, mais altos os filhos
- Vamos olhar nas estatísticas descritivas das 2 variáveis
  - ▶ mais correlação

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.  
## Please use a list of either functions or lambdas:  
##  
##   # Simple named list:  
##   list(mean = mean, median = median)  
##  
##   # Auto named with `tibble::lst()`:  
##   tibble::lst(mean, median)  
##  
##   # Using lambdas  
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last_warnings()` to see where this warning
```

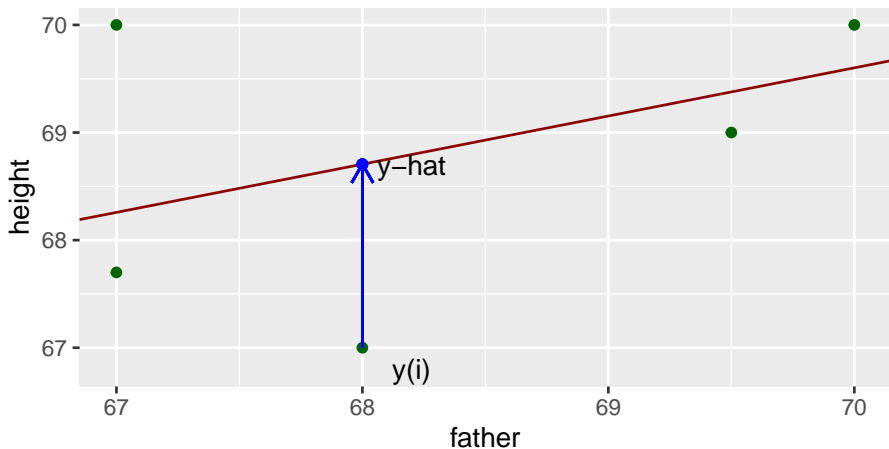


# O Que É a “Correlação”?

- *Coeficiente de Correlação* mede o grau da associação linear entre 2 variáveis
- Sempre cai entre -1 e +1
  - ▶ -1 significa uma relação perfeitamente inversa (quando  $x$  sobe,  $y$  desce pela mesma proporção)
  - ▶ 0 significa que não existe uma relação linear entre as 2 variáveis
  - ▶ +1 significa uma relação perfeitamente positiva (quando  $x$  sobe,  $y$  sobe pela mesma proporção)
- V.S.S: quando tem correlação positiva, tem inclinação da linha de tendência positiva, e vice versa

# Para Calcular a Linha de Regressão – O Que Queremos?

- Uma linha que minimiza a diferença entre  $y_i$  e  $\hat{y}$
- Precisamos trabalhar com o quadrado da diferença
  - ▶ para não ter uma soma de 0

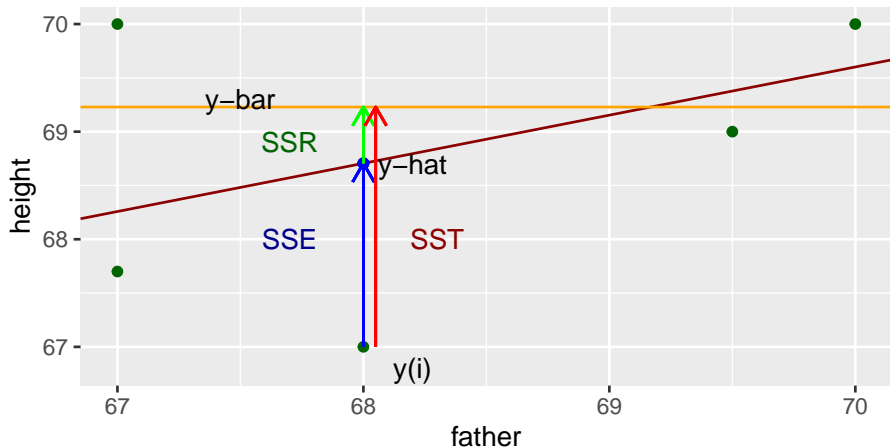


# SSE – Um Componente do Soma de Quadrados (SST)

- $SST = SSE + SSR$
- SST – Total
- SSE – Relacionados aos Erros/Resíduos
- SSR – Relacionados/Explicados pela regressão

# SST – O Que Representa?

- A variância total é a diferença entre o valor do modelo para cada valor de  $X$  e a média dos valores da variável dependente ( $\hat{y}$ )



# Soma dos Quadrados

- Referimos a esse soma dos quadrados que queremos minimizar como **SSE**
  - ▶ Error sum of squares
- SSE como componente da soma dos quadrados total
  - ▶ SSE -- soma dos quadrados relacionados ao resíduo
  - ▶ SSR -- soma dos quadrados relacionados a regressão
- Expressão de SSE

$$SSE = \sum_{i=1}^n (y_i - \hat{y})^2$$

$$SSE = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$$

# Para Determinar a Formula para $\beta_0$ e $\beta_1$

- Para minimizar a SSE (determinar a linha mais eficiente), precisamos usar cálculo
- Fazer a derivativo parcial com respeito a  $\beta_0$  e  $\beta_1$

$$\frac{\partial}{\partial \beta_0} SSE = \frac{\partial}{\partial \beta_1} SSE = 0$$

- Chamadas as equações normais
- Confiamos nos softwares para calcular os parâmetros da equação

- Função `lm` (“linear model”)
- `lm(formula, data, subset, weights, na.action, method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL, offset, ...)`
- Os importantes são `formula`, `data`, `subset`, `weights`, `na.action`
- `formula`: onde mostra quais variáveis você está modelando
  - ▶ Variável dependente vem primeiro
  - ▶ Separada da independente(s) por “ ~ ”
  - ▶ Para os boys: `height ~ father`
  - ▶ `data`: data frame ou tibble que contem as variáveis
  - ▶ `subset`, `weights`: parâmetros que permitem que você customizar tratamento das variáveis
  - ▶ `na.action`: como vai tratar os dados missing na base de dados

# Função Aplicada aos Pais e Filhos

- Função `lm` produz uma lista de 12 itens em um formato especial

```
fit1 <- lm(height ~ father, data = boys)
summary(fit1)
```

```
##
```

```
## Call:
```

```
## lm(formula = height ~ father, data = boys)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -9.3774 -1.4968  0.0181  1.6375  9.3987
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 38.25891    3.38663   11.30   <2e-16 ***
## father      0.44775    0.04894    9.15   <2e-16 ***
```

```
## ---
```



# O Que Diz o Modelo

$$\hat{y} = 38.259 + 0.448x$$

- Se o pai tivesse 0 altura, o filho teria 38.259 polegadas de altura
  - ▶ Não faz sentido prático, mas estabelece a base para calculo de altura
  - ▶ Para cada polegada incremental da altura do pai, o filho seria 0.448 polegadas mais alto

# Extrair os Valores dos Coeficientes

## 1 Usar broom::tidy

```
broom::tidy(fit1) %>% knitr::kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	38.2589122	3.3866340	11.297032	0
father	0.4477479	0.0489353	9.149788	0

## 2 Usar coef

```
coef(fit1)
```

```
## (Intercept)      father  
## 38.2589122    0.4477479
```

# Previsões de Novos Valores

- Pode usar o modelo para prever novos valores da altura dos filhos
- Usar `broom::augment`

```
fit1 %>% broom::augment(newdata = tibble(father = 72))
```

```
## # A tibble: 1 x 3
##   father .fitted .se.fit
##   <dbl>   <dbl>   <dbl>
## 1      72    70.5    0.178
```