Aula 4

James Hunter, Ph.D.

Professor, Retrovirologia, UNIFESP

29 de maio de 2020

Terminamos Ontem com um Modelo de Regressão

- Estudo de Sir Francis Galton sobre alturas nas famílias
- Começo da regressão
- Filhos mais altos que os pais?

Modelo e Resultados

```
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 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.424 on 463 degrees of freedom
## Multiple R-squared:  0.1531, Adjusted R-squared:  0.1513
## F-statistic: 83.72 on 1 and 463 DF,  p-value: < 2.2e-16
```

Coeficientes do Modelo

```
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

Produz a Equação

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

Analisar o Modelo

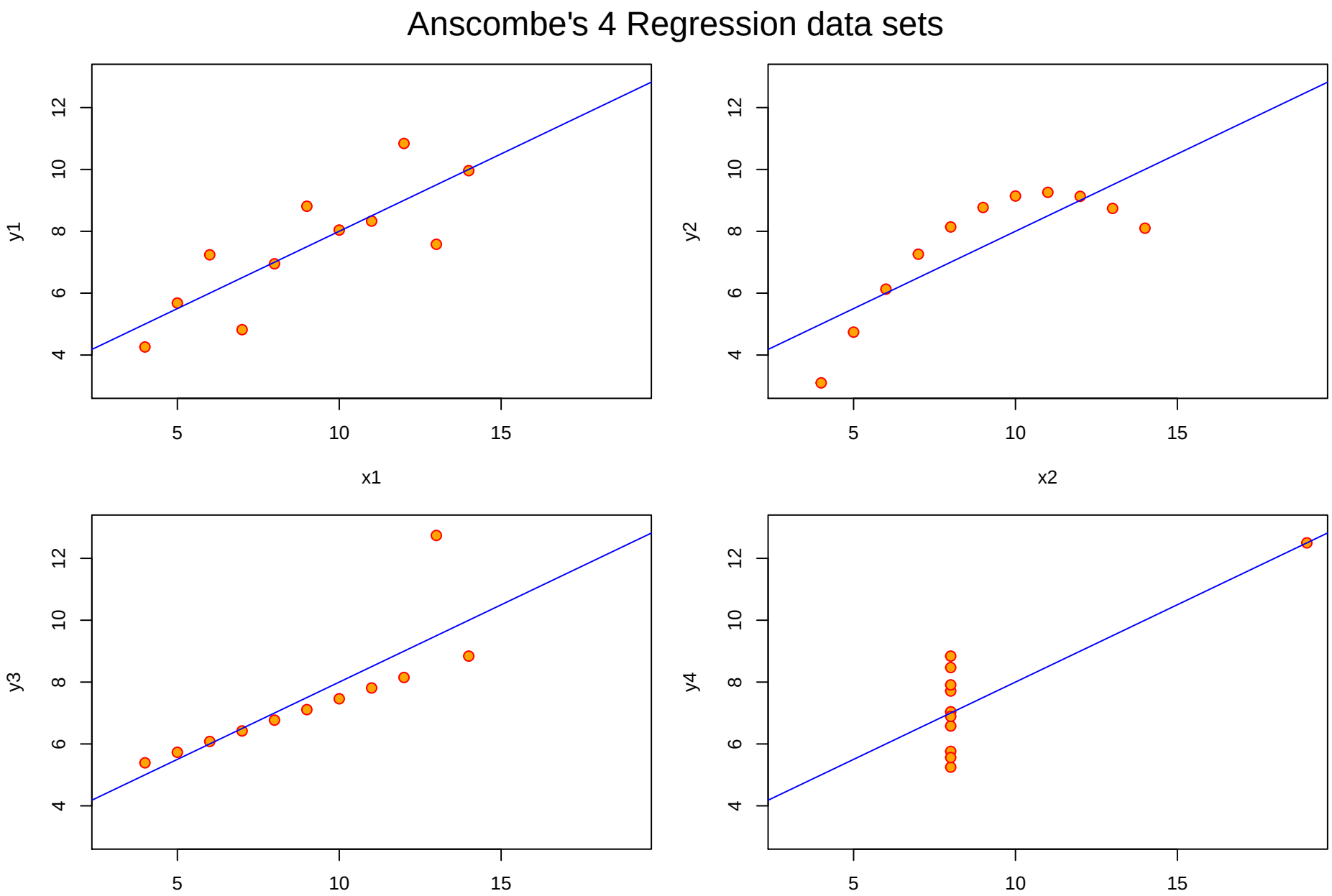
- Num curso sobre R, porque tanta atenção à estatística?
- Maioria dos analistas aplicam regressão - Ignorantes das exigências da técnica - Em situações inapropriadas - Erroneamente

Um exemplo da Dificuldade com Regressão

- 4 datasets
- Todos têm os mesmos valores de resumo e da regressão

Number of observations $(n) = 11$
Mean of the x's $(\bar{x}) = 9.0$
Mean of the y's $(\bar{y}) = 7.5$
Regression coefficient (b_1) of y on $x = 0.5$
Equation of regression line: $y = 3 + 0.5\,x$
Sum of squares of $x - \bar{x} = 110.0$
Regression sum of squares $= 27.50$ (1 d.f.)
Residual sum of squares of $y = 13.75$ (9 d.f.)
Estimated standard error of $b_1 = 0.118$
Multiple $R^2 = 0.667$

O Quarteto de Anscombe



O Que Significa o Modelo? Como Interpretar Ele?

- Temos que testar nossos modelos
- De qualquer técnica de análise
- Respeitamos as premissas?
- Os dados são em forma apropriada pela técnica de análise

Existe Relação Entre Variáveis Independente e Dependentes?

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

- Se β_1 (inclinação da linha) for 0, o que seria a equação?

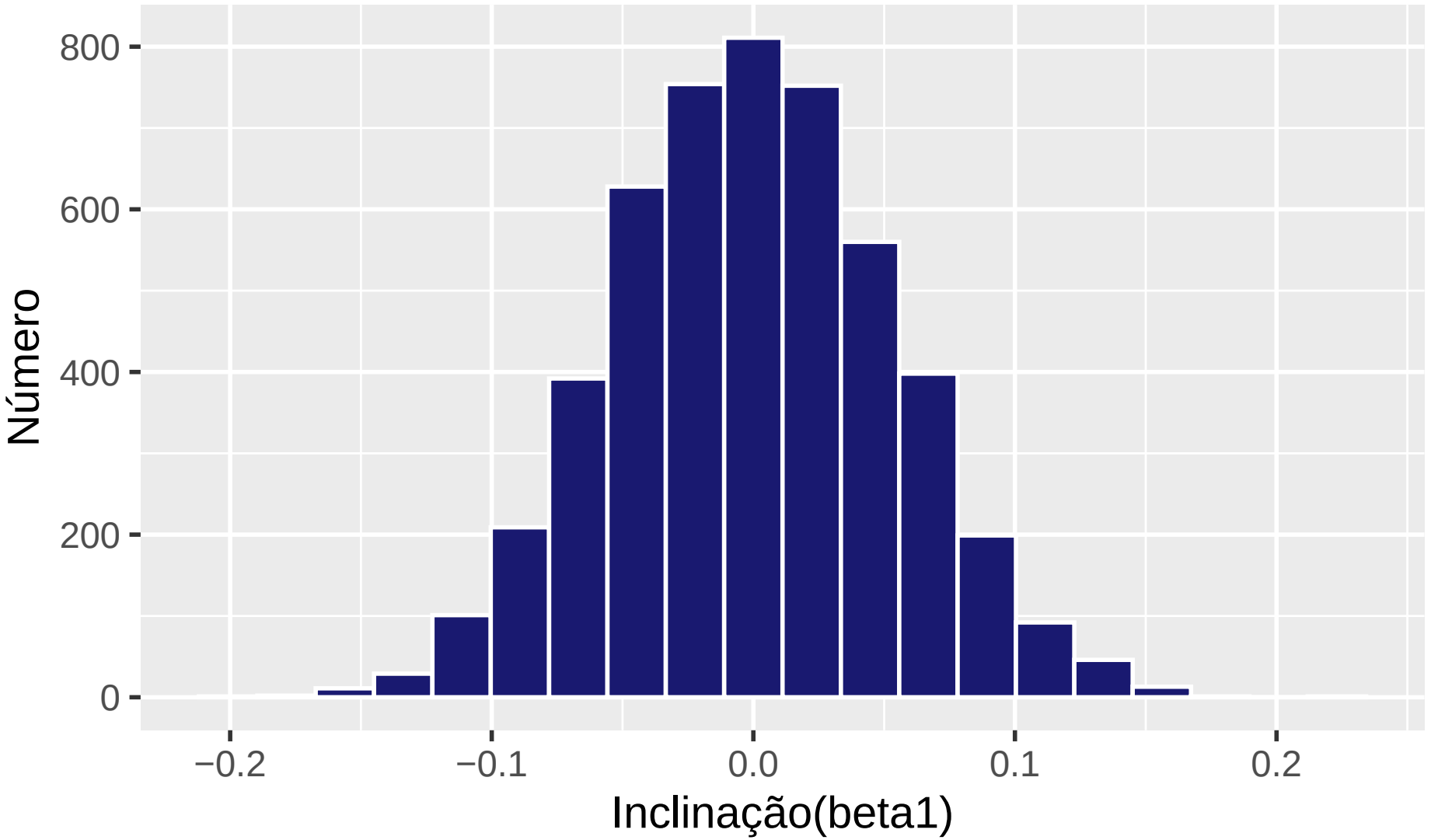
$$Y_i = \beta_0 + \epsilon_i$$

- X desaparece
- Relação entre Y e X não existe
  - Só tem intercepto e erro
- Faz possível teste eficiente de existência ou não de uma relação entre X e Y
- Cria uma hipótese nula de $H_0 : \beta_1 = 0$

Teste de Hipótese Nula

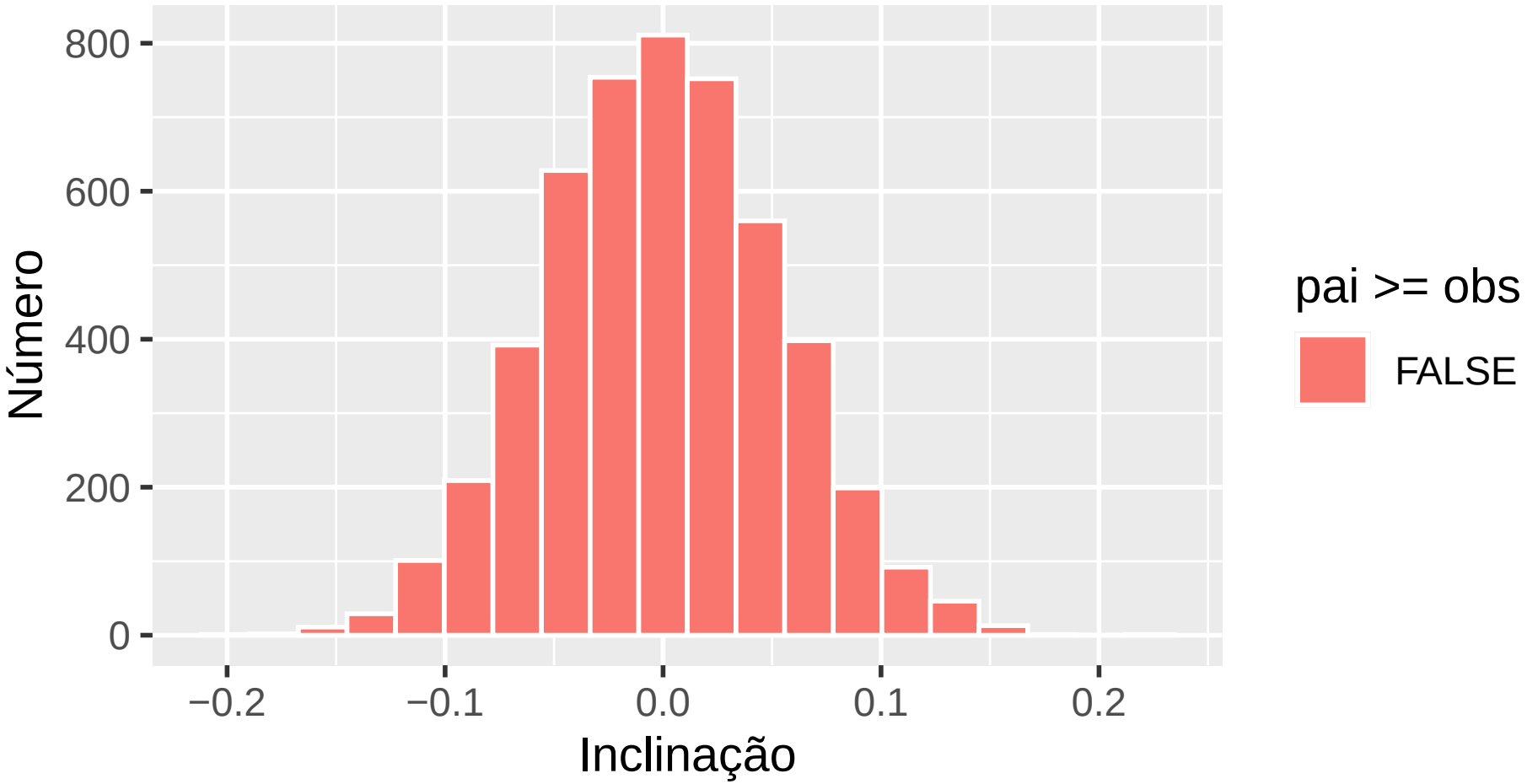
- Vamos fazer uma simulação de hipótese nula
- Se a nula é correta, qualquer altura do filho podia ter ocorrido com qualquer altura do pai.
- Podemos calcular o modelo de regressão 5.000 vezes com valores de todo a base de alturas dos filhos
- Como resultado, vamos focar nos valores da inclinação, β_1
- Depois, nós vamos comparar nosso valor de β_1 observado e ver onde cai na distribuição dos valores simulados

Histograma das Inclinação dos Modelos



Histograma com Valores Abaixo/Acima do Valor da Amostra

```
## Número de simulações com beta1 >= obs:  0
```



O Valor-p da Inclinação (β_1)

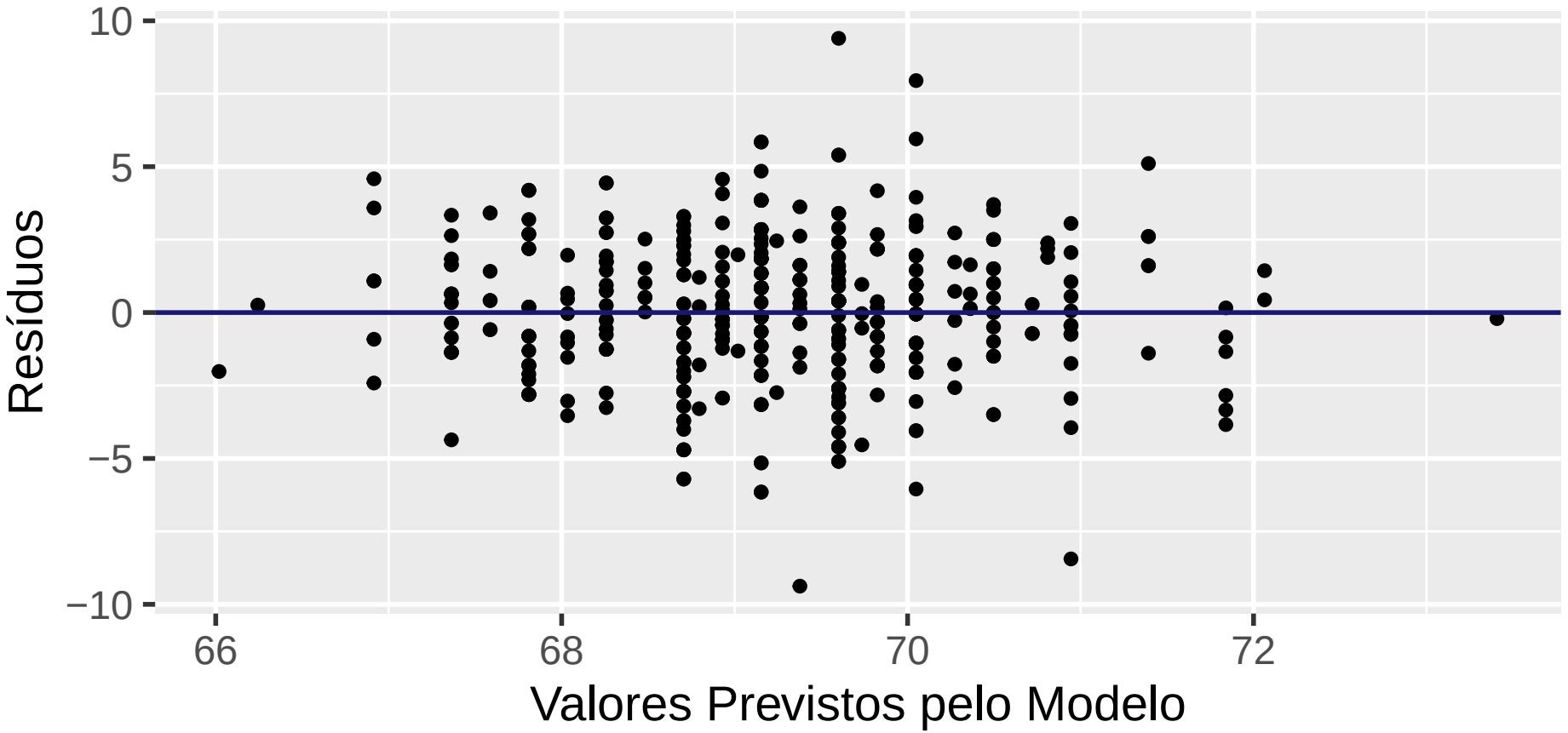
- Porque **nenhuma** das simulações produziu um valor superior ao observado (0.448)
  - Pode concluir que o valor-p deste teste é 0.
  - Não parece existir nenhuma chance que a inclinação $= 0$
- Assim, rejeitamos a hipótese nula e concluir que uma relação linear entre as alturas dos pais e filhos realmente existe.

Premissas de Regressão Linear

1. Todas as variáveis independentes devem ter a mesma variância - Gráfico de resíduo deve evitar padrões indo de esquerda até direta
2. Todas as observações, resíduos e variáveis independentes: todos devem ser independentes - Gráfico de resíduo não deve mostrar um padrão sinuoso
3. Resíduos têm uma distribuição perto a normal - Gráfico "qq" dos resíduos padronizados - Indica que as variáveis têm distribuição normal multivariada
4. Variáveis independentes devem evitar *multicollinearity* - Ter correlações altas entre elas

Gráfico de Resíduos

- Gráfico que mostra o valor previsto pelo modelo ("fitted value") vs. o resíduo
- Uso da função `broom::augment()`
  - Eficiente para extrair os valores utilizados nos testes dos modelos



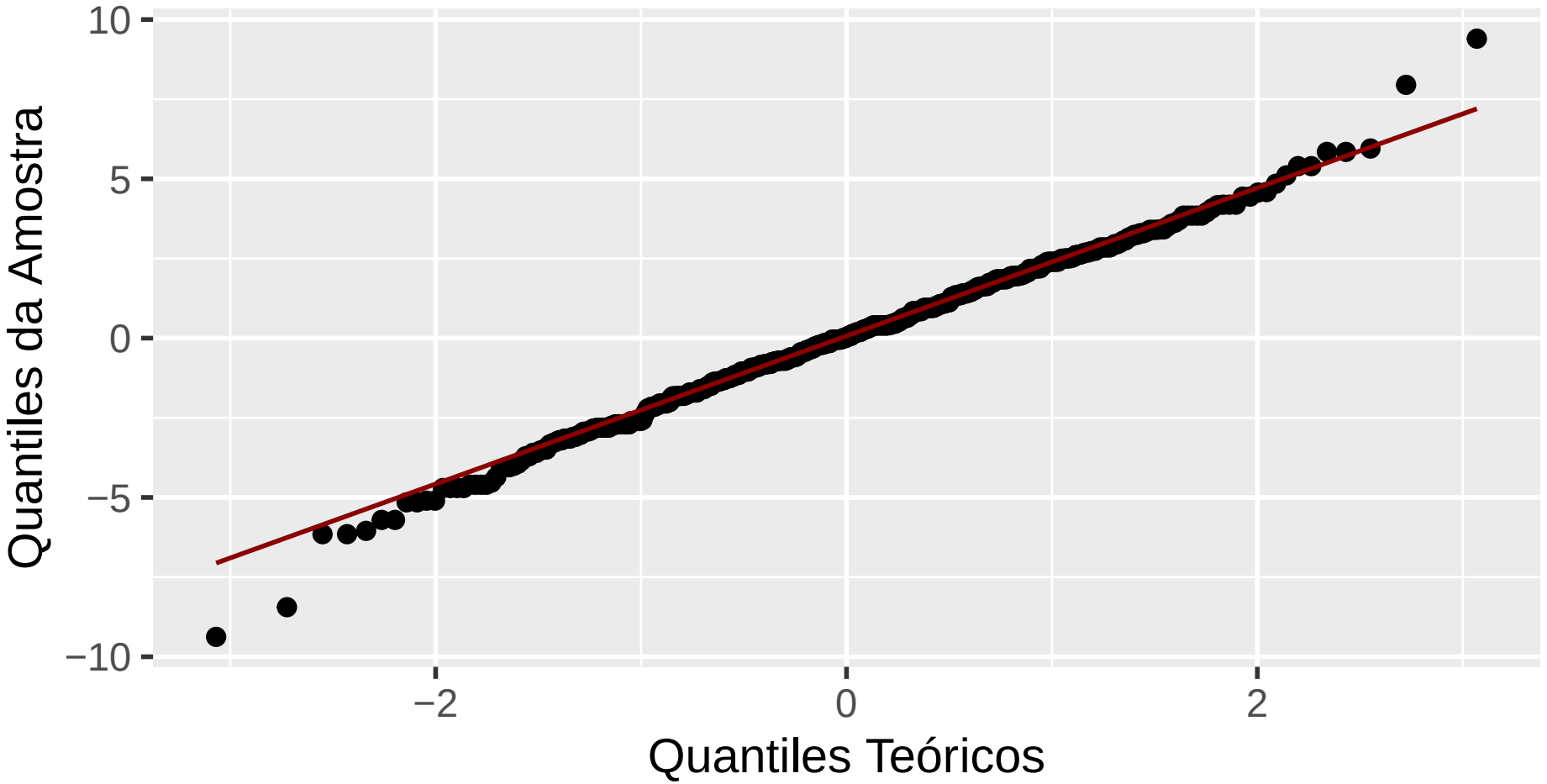
Importância dos Resíduos

- Pode usar os erros/resíduos para verificar se as premissas da regressão foram respeitadas
- Não devem mostrar um padrão linear

Gráfico Q-Q

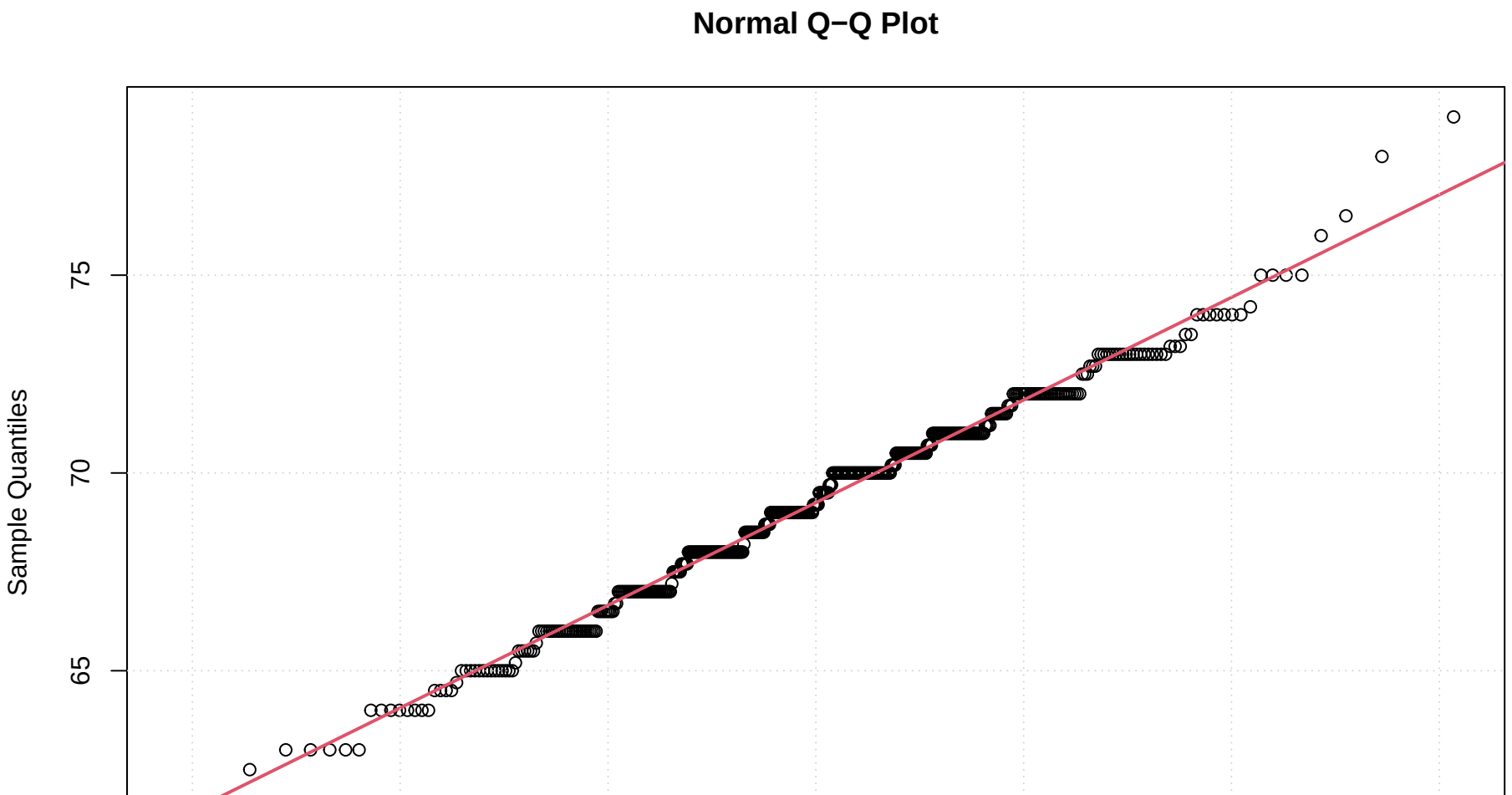
- Verifica a normalidade dos resíduos
  - Mais perto a uma linha reta, melhor o "fit" com uma distribuição normal

```
grqq <- ggplot(data = mods, aes(sample = .resid))
grqq <- grqq + stat_qq()
grqq <- grqq + stat_qq_line(color = "darkred")
grqq <- grqq + labs(x = "Quantiles Teóricos",
                    y = "Quantiles da Amostra")
```


Quantiles da Amostra
10
5
0
−5
−10
−2
0
2
Quantiles Teóricos

Gráficos Q-Q Também Disponível Diretamente em Base R

```
qqnorm(boys$height)
qqline(boys$height, col = 2, lwd = 2)
grid()
```



Teste-F das Variâncias do Modelo

- Teste-F é um teste que verifica que as variâncias das variáveis são perto de iguais
- Utiliza a Distribuição F
  - Tem 2 graus de liberdade como parâmetros
- Serve como um teste de significância total de um modelo
- Produzido pelo função `Summary` da função `lm`

Teste-F do Modelo das Alturas Pai-Filho

```
##
## 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 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.424 on 463 degrees of freedom
## Multiple R-squared:  0.1531, Adjusted R-squared:  0.1513
## F-statistic: 83.72 on 1 and 463 DF,  p-value: < 2.2e-16
```

Resumo de Soma dos Quadrados

- Soma Total de Quadrados

$$SST = \sum (y_i - \bar{y})^2$$

- Soma dos Quadrados dos Erros

$$SSE = \sum (y_i - \hat{y})^2$$

- Soma dos Quadrados de Regressão

$$SSR = \sum (\hat{y}_i - \bar{y})^2 = SST - SSE$$

R^2 – Coeficiente de Determinação

- Medida de quanto a linha de regressão explica a variância em Y
- Relação entre a SSR e a SST

$$R^2 = \frac{SSR}{SST}$$

- Calculado pelo `lm`
  - visível em `Summary`
- Varia entre 0 e 1
- $\sqrt{R^2} = r$ (coeficiente de correlação)

R^2

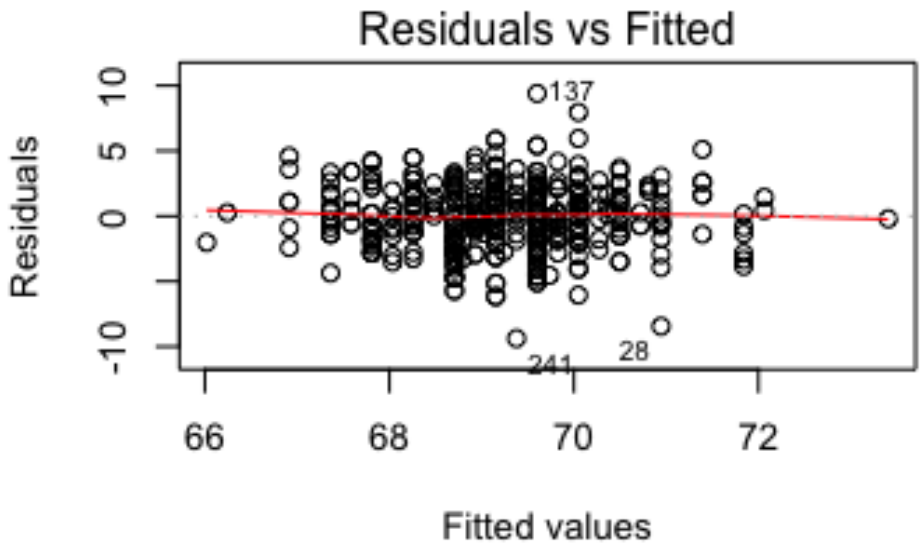
```
##
## Call:
## lm(formula = height ~ father, data = boys)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3774 -1.4968  0.0181  1.6375  9.3987
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 38.25891    3.38663   11.30   <2e-16 ***
## father       0.44775    0.04894    9.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.424 on 463 degrees of freedom
## Multiple R-squared:  0.1531, Adjusted R-squared:  0.1513
## F-statistic: 83.72 on 1 and 463 DF,  p-value: < 2.2e-16
```

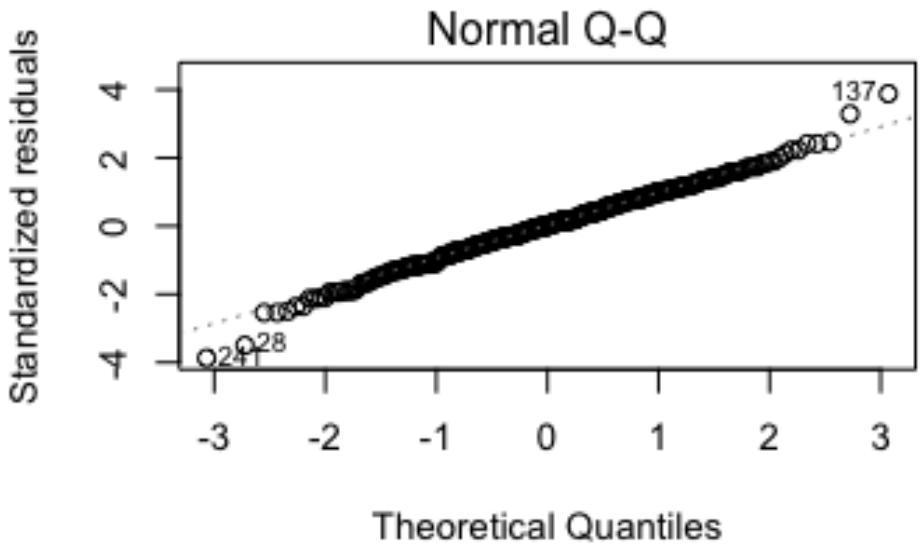
Significância de R^2

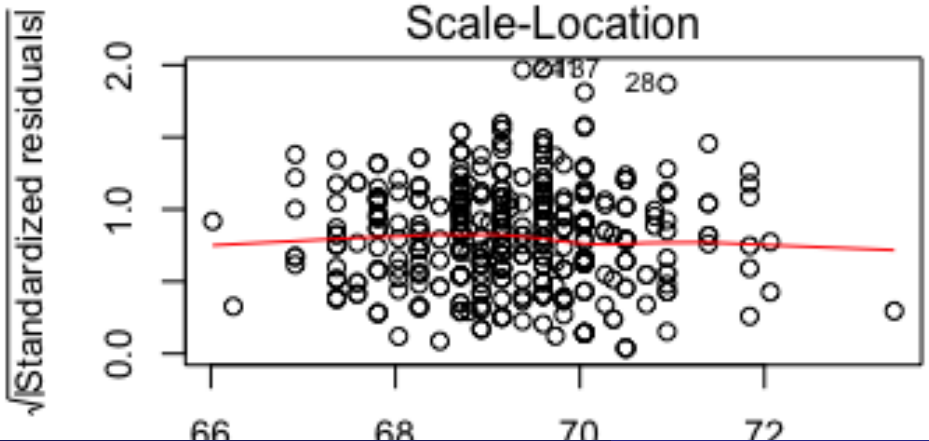
- Se 100% da variância ser explicado pela regressão
- $SSR = SST$
- $\therefore\ R^2 = SST/SST = 1$
- Variância completamente explicado pela regressão
- Em geral, o grau em que a regressão explica a variância no modelo

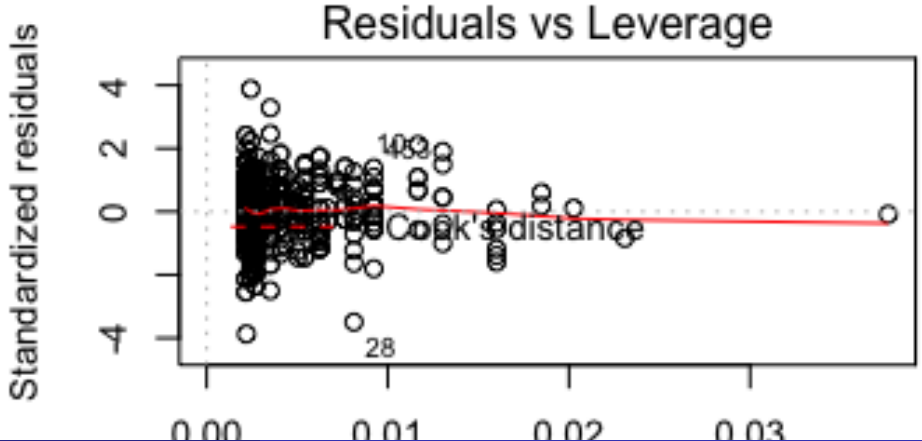
Dois Gráficos Mais Avançados

Função `plot` para Objetos `lm`

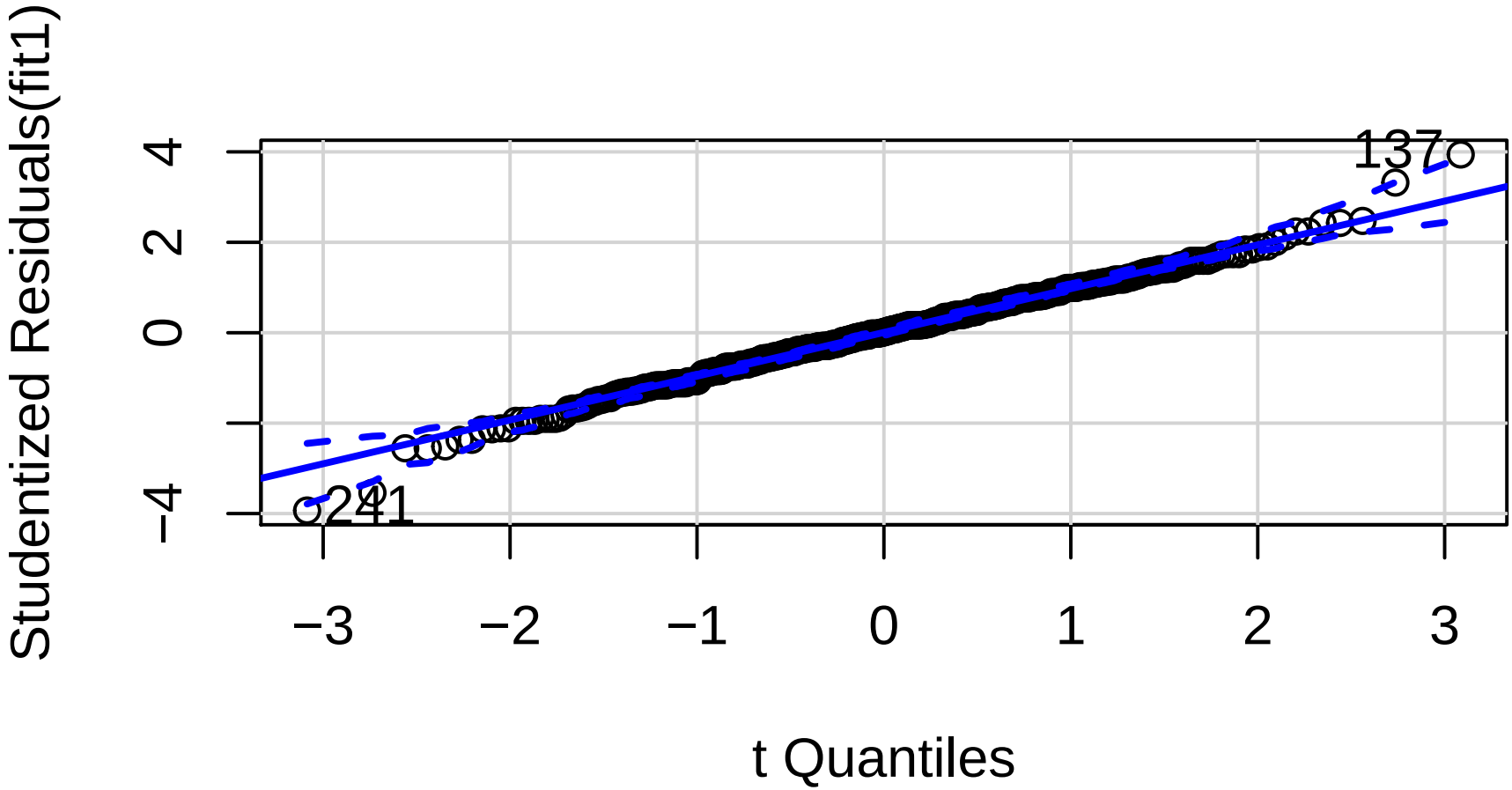








Função qqPlot() do Pacote car




```
## [1] 137 241
```

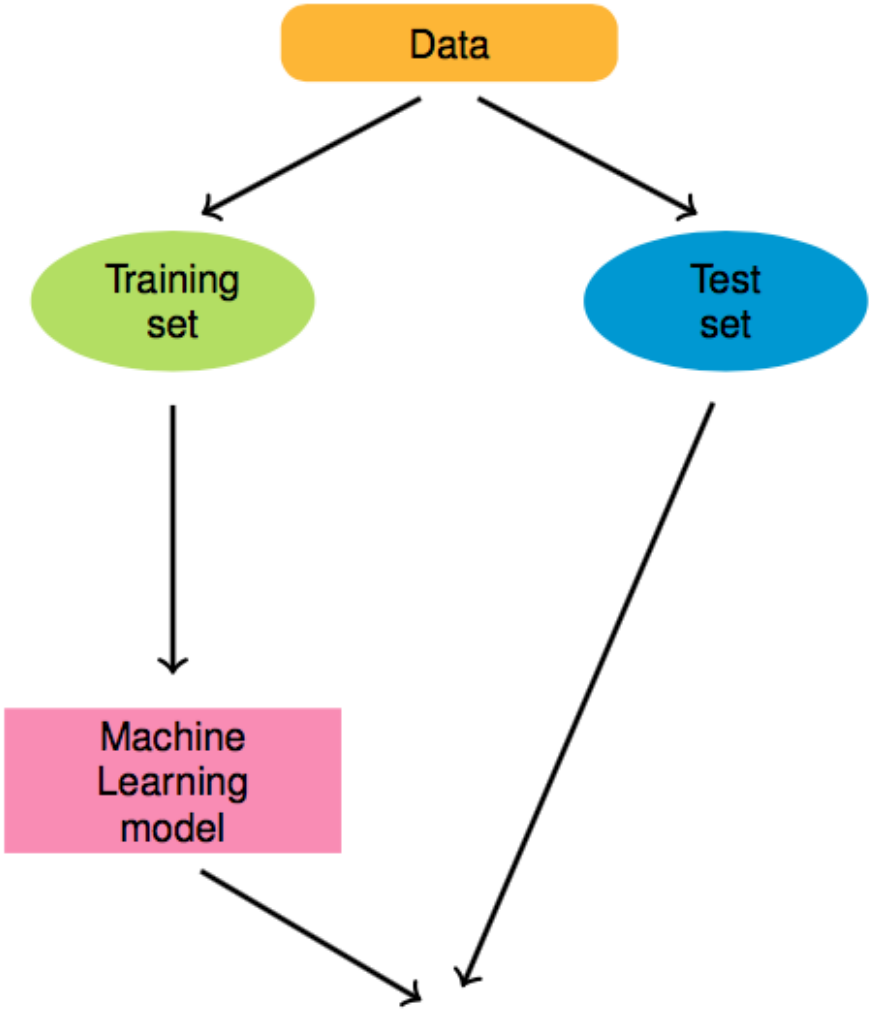
Regressão Linear Múltipla - *MLR*

- Regressão quando tem mais de 1 variável independente
  - Mais de 1 covariado
- Mudança na Equação do Modelo de Regressão

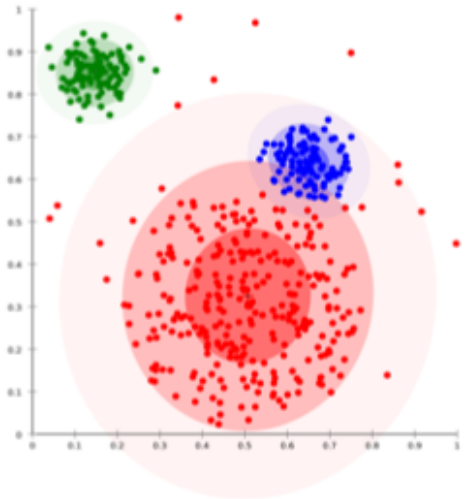
$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i + ... + \beta_k X_i + \epsilon_i$$

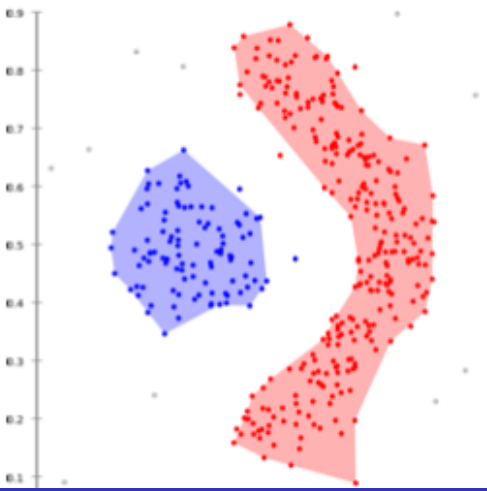
Tipos de *Machine Learning*

Supervised



Unsupervised





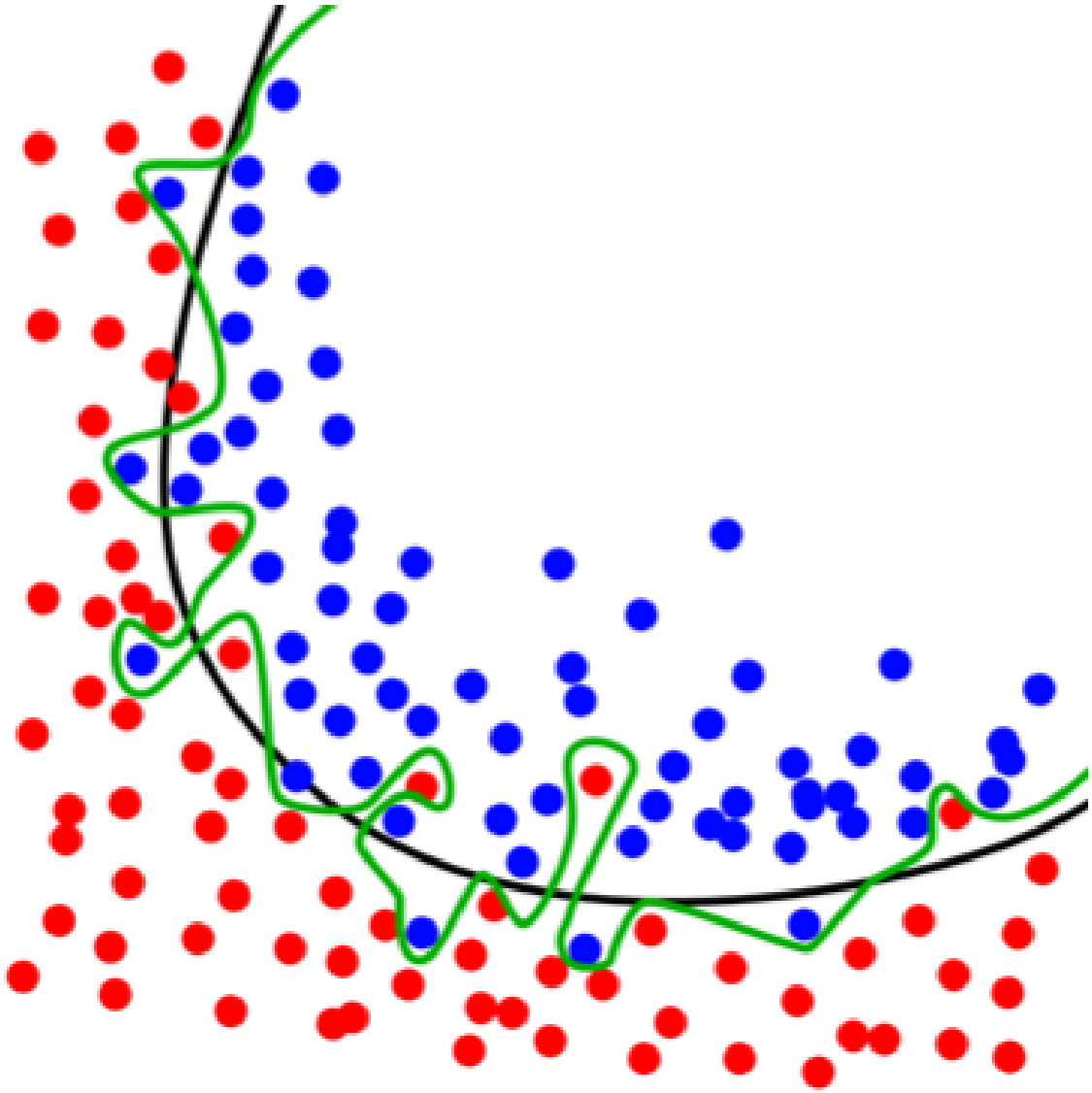
Supervisionado x Não-Supervisiado

- Supervisionado – tem variável dependente (y) - Pode ter uma ou mais de uma covariáveis - Sub-divisão
  - Regressão
  - Classificação
- Não-Supervisionado – não tem variável dependente - Variás variáveis independentes - Procurar relações entre elas - Ex: análise de clusters

Treinamento x Testagem dos Modelos

- Divisão dos data frames em partes separadas
- Quer evitar *overfitting*
- **NUNCA, JAMAIS, USE OS MESMOS CASOS PARA TESTES QUE VOCÊ USOU PARA TREINAMENTO**

Overfitting



Características dos Modelos

- Covariáveis
  - Quantos são suficientes para construir um modelo?
  - Número insuficiente – modelo não descreve suficiente a condição
  - Número demais – overfitting
    - Modelo *kitchen sink*
    - Joga tudo dento e espera ter um resultado bom

Fortelecer o Modelo

- Bootstrapping
- k-fold Cross Validation
  - Tirar uma parte (*fold*) do grupo de treinamento
    - Treinar o resto do grupo de treinamento
    - Testar o modelo com os casos do *fold*
  - Faça o mesmo com os outros folds
  - Use como modelo final aquele que mostra melhor desempenho

Fonte Muito Útil para *Machine Learning*

- Dr. Sharin Glander, Univ. de Münster, Alemanha
  - Webinar excelente
  - "Building meaningful machine learning models for disease prediction"
  - https://shiring.github.io/machine_learning/2017/03/31/webinar_code

Machine Learning em Modelagem de Negócios

- Tipicamente, projetos com “big data”
- Modelo pode fornecer informação rapidamente e corretamente
  - Executivos podem usar a informação para elaborar estratégias
- Aplicação: entender o que o consumidor quer
- Exemplo: (de medicina)
  - Diagnostico de câncer de mama com ajuda de modelo informatizado

Podemos Ter Confiança nos Modelos de Machine Learning?

- Algoritmos de ML modelam interações de alto grau enter as variáveis
- Interpretação dos resultados de ML pode ser difícil
- A "caixa preta" dos algoritmos de ML escondem como eles fazem escolhas
  - Em alguns algoritmos (e.g. redes neurais)
- Assim, *precisamos modelos que significam algo* para os
  - Arquitetos
  - Usadores
- "Meaningful Models"

O Que Faz um Modelo um "Meaningful Model"

- Poder generalizar baseado no modelo
- Responde à pergunta original
- . . . com suficiente precisão para ser confiável
- Grau de precisão depende do problema

Covariáveis – *Features*

- As variáveis independentes
- Variáveis para treinar o modelo
- Selecionar as variáveis certas – **crucial**
- Mais features não necessariamente bom
  - Perigo de “overfitting”

Mãos na Massa

Dados

- Continuar com os dados de `galton`
- Expandir a análise para incluir altura da mãe

```
glimpse(galton)
```

```
## Rows: 898
## Columns: 6
## $ family <fct> 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, ...
## $ father <dbl> 78.5, 78.5, 78.5, 78.5, 75.5, 75.5, 75.5, 75.5, 75.0, 75.0, ...
## $ mother <dbl> 67.0, 67.0, 67.0, 67.0, 66.5, 66.5, 66.5, 66.5, 64.0, 64.0, ...
## $ sex    <fct> M, F, F, F, M, M, F, F, M, F, M, M, F, F, F, M, M, M, F, F, ...
## $ height <dbl> 73.2, 69.2, 69.0, 69.0, 73.5, 72.5, 65.5, 65.5, 71.0, 68.0, ...
## $ nkids  <int> 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, ...
```

Método Novo Que Seguimos

- Método de *Machine Learning*
- Seguir a metodologia do pacote `caret`
- Passo 1
- Dividir os casos em 2 grupos: treinamento, testes
- Divisão aleatória
- Treinar o modelo com o grupo de treinamento
  - Deixar os casos de testes ao lado durante treinamento
- Depois testar as previsões do modelo com os valores do grupo de testes
- Objetivo: fazer previsões corretas
  - Mais importante que a elegância do modelo

Carregar Pacotes Necessários para Este Método

- `caret` : *Classification And REgression Training*
- `ggpubr`: gráficos
- `broom` : funções para mostrar e comparar os modelos
- `nortest`: testes de normalidade estatística
- `janitor`: ajuda com tabelas

Processo de caret

- Fornece um *workflow* eficiente para problemas de regressão e classificação
- Modelos construídos com a função `train`

```
Define sets of model parameter values to evaluate
for each parameter set do
    for each resampling iteration do
        Hold–out specific samples
        [Optional] Pre–process the data
        Fit the model on the remainder
        Predict the hold–out samples
    end
    Calculate the average performance across hold–out
end
```

```r
set.seed(42)
indice <- createDataPartition(galton$height, p = 0.70, list = FALSE)
head(indice[, 1], 25)
```

```
##  [1]  2  3  4  6  7  8  9 13 14 15 17 18 20 21 23 24 25 26 27 28 29 30 31 33 34
```

Criar `train_data` e `test_data`

- **VSS** lembre da virgula depois do `indice`
  - Por quê?
- Para `test_data`, você quer os dados que **NÃO** são de `train_data`
  - Assim, precisa usar o sinal de menos (-)

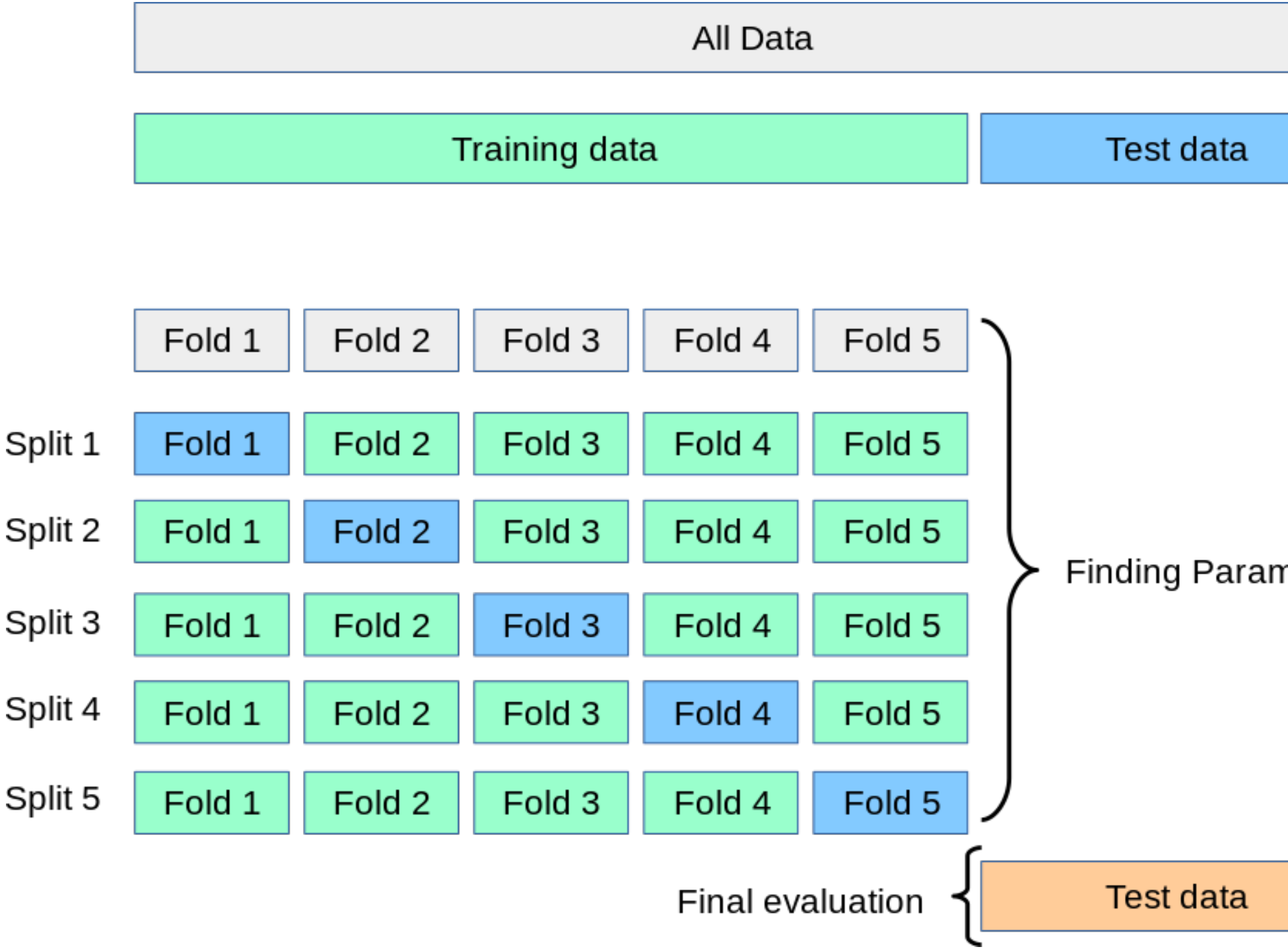
```
train_data <- galton[indice, ]
test_data <- galton[-indice, ]
```

Validação Cruzada (*Cross-Validation*)

- Validação do cálculo dos parâmetros do modelo utilizando pedaços dos casos cada repetição
- Evita necessidade de dividir o conjunto em 3 grupos (treinamento, validação, testes)
- Relacionado ao processo de *bootstrap* - reamostragem
- `caret` seleciona o modelo que tem o melhor desempenho

Processo de *k-fold* Validação Cruzada

- Dividir o grupo dos casos de treinamento em k subgrupos iguais
- Treinar o modelo com $k - 1$ dos folds
- Software testa este modelo com os casos do fold deixado fora e avalia desempenho (precisão)
- Repetir até tinha deixado fora todos os folds
- Pode repetir o processo inteiro um número das vezes


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Split 2
Split 3
Split 4
Split 5
Finding Param
Final evaluation
Test data

Pre-Processamento

- Se tiver traços das variáveis muito não normais
- Pode reduzir a não-normalidade das curvas com
  - Centralização (subtrair a média do valor)
  - Normalização (dividir valor centralizado por des. padrão)
- `caret` oferece essas opções

train() Modelo

- caret::train() é a função que determina os parâmetros do modelo da regressão

```
fit_pai_mae <- caret::train(height ~ father + mother,
                    method = "lm",
                    data = train_data,
                    trControl = trainControl(method = "repeatedcv",
                                             number = 5,
                                             repeats = 10,
                                             savePredictions = "none",
                                             verboseIter = FALSE))
```

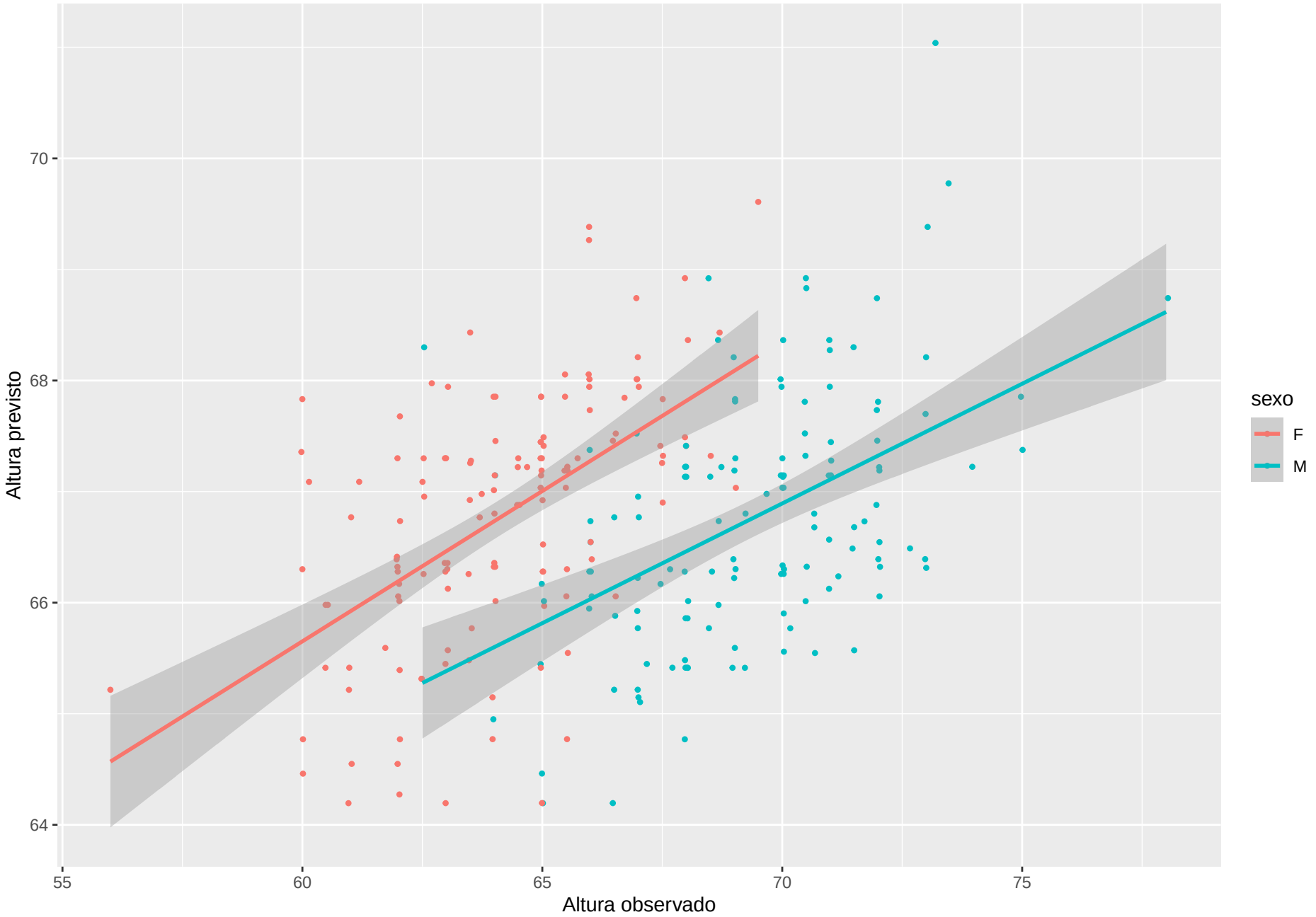
```r
summary(fit_pai_mae)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.480 -2.740 -0.179  2.807 11.699
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.59851    5.08952   4.637 4.31e-06 ***
## father       0.37731    0.05589   6.751 3.34e-11 ***
## mother       0.26601    0.05870   4.532 7.00e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.404 on 628 degrees of freedom
## Multiple R-squared:  0.1052, Adjusted R-squared:  0.1023
## F-statistic:  36.9 on 2 and 628 DF,  p-value: 7.022e-16
```

Como Foi o Desempenho do Modelo?

- Aplicar o modelo aos dados do conjunto de `test_data`
  - Até agora, o modelo não tinha visto esses dados
  - Indica o que pode fazer com qualquer dados que mede a mesma fenômeno
  - `predict` calcula os valores previstos usando os parâmetros do modelo

```r
# previsões
prv <- predict(fit_pai_mae, test_data)
# comparar para preços observados
gg_pai_mae_1 <- data.frame(obs = test_data$height,
                           previs = prv,
                           sexo = test_data$sex) %>%
  ggplot(aes(x = obs, y = previs, color = sexo)) +
    geom_jitter(shape = 20) +
    geom_smooth(method = "lm") +
    labs(x = "Altura observado", y = "Altura previsto")
```


Altura previsto
70
68
66
64
55
60
65
70
75
Altura observado
sexo
F
M

Quanta Precisão Teve o Modelo?

- Olhar a diferença entre os valores verdadeiros (observados) e os valores previstos pelo modelo
- Quantas dessas diferenças foram menores que um padrão razoável (? 2 polegadas)

```
pred <- predict(fit_pai_mae, test_data)
res <- tibble(pred = pred,
                  obs = test_data$height,
                  dif = obs - pred)
padrao_in <- 2
# teste de bom, ruim
res <- res %>%
  mutate(bomruim = ifelse(abs(dif) <= padrao_in, "bom", "ruim"))
janitor::tabyl(res$bomruim) %>% adorn_pct_formatting()
```

```
##  res$bomruim   n percent
##          bom  95   35.6%
##         ruim 172   64.4%
```

Modelo Não É Bom

- Precisão muito baixo
  - 36% dentro do padrão de 2 polegadas
- R^2 muito baixo (0.1023)
  - Só 10% da variância no modelo explicada pelas variáveis

Podemos Fazer Melhor?

- Gênero pode ter um efeito
- Gênero é uma variável catégorica
- Regressão compara as distribuições dos números
- Mas pode incluir variáveis categóricas

Inclusão das Variáveis Categóricas em Regressão

- Dividir a variável em "*dummy*" variáveis
  - 1 variável *dummy* para cada nível da variável categórica menos o primeiro nível
  - Se tiver 3 níveis (`alto`, `medio`, `baixo`), o sistema criaria 2 novas variáveis
    - `medio` e `baixo`
    - `alto` seria um valor de referência que representa o caso quando nenhum dos outros níveis está presente

```r
notas <- tibble(x = rep(c("alto", "media", "baixo"), 3),
                y = c(3, 2, 1, 3, 2, 1, 7, 5, 2))
summary(lm(y ~ x, data = notas))
```

```
##
## Call:
## lm(formula = y ~ x, data = notas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3333 -1.0000 -0.3333  0.6667  2.6667
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.3333     0.9813   4.416  0.00449 **
## xbaixo       -3.0000     1.3878  -2.162  0.07390 .
## xmedia       -1.3333     1.3878  -0.961  0.37377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.7 on 6 degrees of freedom
## Multiple R-squared:  0.4388, Adjusted R-squared:  0.2518
## F-statistic: 2.346 on 2 and 6 DF,  p-value: 0.1767
```

Incluir sex na Regressão

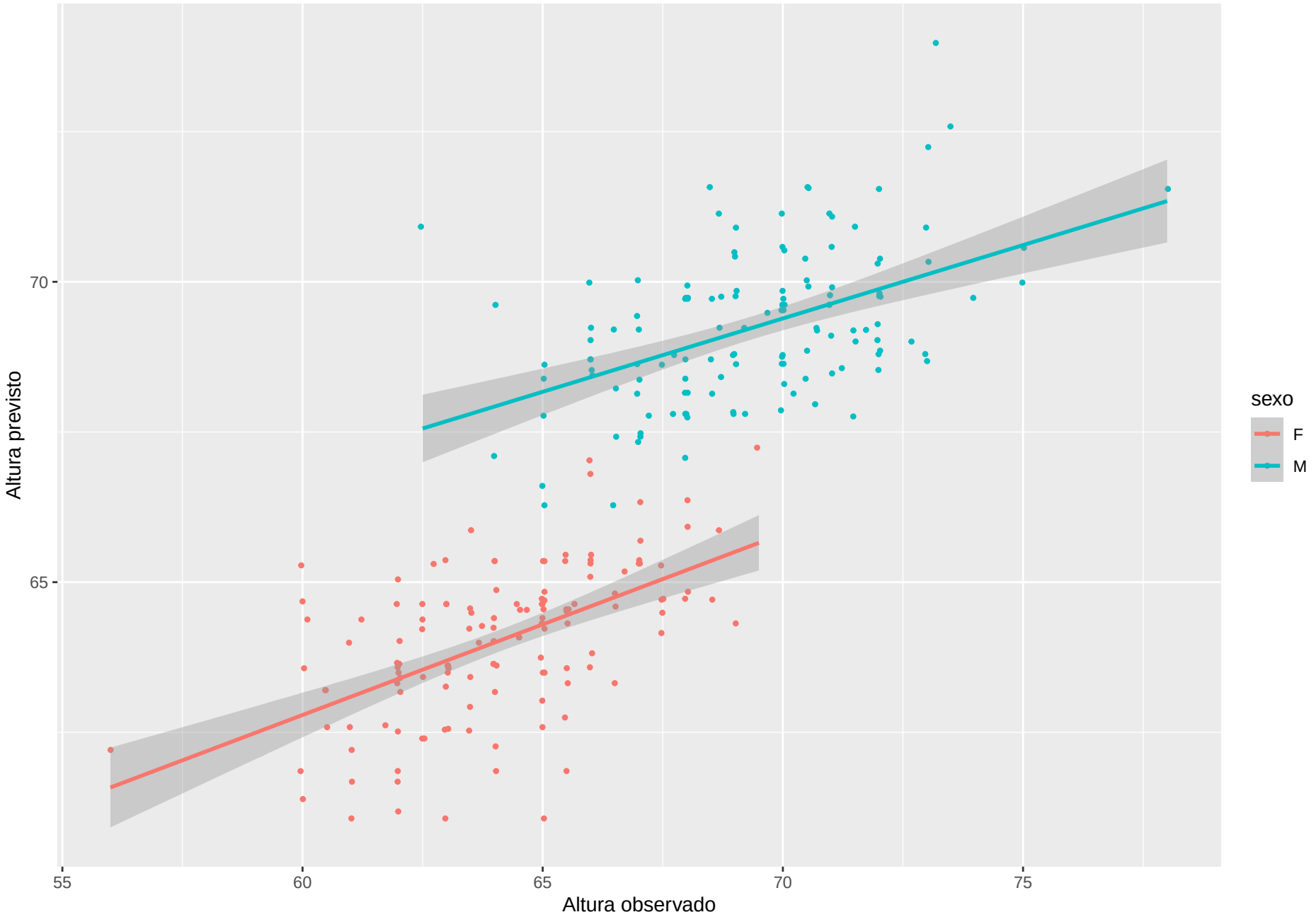
```
fit_pms <- caret::train(height ~ father + mother + sex,
                        method = "lm",
                        data = train_data,
                        trControl = trainControl(method = "repeatedcv",
                                                 number = 5,
                                                 repeats = 10,
                                                 savePredictions = "none",
                                                 verboseIter = FALSE))
```

```r
summary(fit_pms)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4833 -1.5274  0.0932  1.5369  9.1510
##
## Coefficients:
##             Estimate Std. Error t value   Pr(>|t|)
## (Intercept) 15.05115    3.29308   4.571 0.00000586 ***
## father       0.40976    0.03604  11.369    < 2e-16 ***
## mother       0.32157    0.03788   8.489    < 2e-16 ***
## sexM         5.21288    0.17527  29.742    < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.194 on 627 degrees of freedom
## Multiple R-squared:  0.6288, Adjusted R-squared:  0.627
## F-statistic: 354.1 on 3 and 627 DF,  p-value: < 2.2e-16
```

Desempenho do Modelo

```
# previsões
prv <- predict(fit_pms, test_data)
# comparar para preços observados
gg_pms_1 <- data.frame(obs = test_data$height,
                       previs = prv,
                       sexo = test_data$sex) %>%
  ggplot(aes(x = obs, y = previs, color = sexo)) +
    geom_jitter(shape = 20) +
    geom_smooth(method = "lm") +
    labs(x = "Altura observado", y = "Altura previsto")
```


70
65
Altura previsto
55
60
65
70
75
Altura observado
sexo
F
M

Quanta Precisão Teve o Modelo?

```
pred <- predict(fit_pms, test_data)
res_pms <- tibble(pred = pred,
                  obs = test_data$height,
                  dif = obs - pred)
padrao_in <- 2
# teste de bom, ruim
res_pms <- res_pms %>%
  mutate(bomruim = ifelse(abs(dif) <= padrao_in, "bom", "ruim"))
tabyl(res_pms$bomruim) %>% adorn_pct_formatting()
```

```
##  res_pms$bomruim   n percent
##              bom 183   68.5%
##             ruim  84   31.5%
```

Resultado

- Modelo consegue prever 69% das alturas dentro da padrão
  - Dobro do modelo anterior
- R^2 aumentou a 0.627 (muito)
- Gênero tem um papel importante na determinação das alturas das crianças
  - O modelo inclui esta caraterística

varImp() Função em caret

- Função avalia a importância relativa das variáveis no modelo
- Mais importante - 100%
- Menos importante - 0%
- Nosso modelo 2

```
varImp(fit_pms)
```

```
## lm variable importance
##
##        Overall
## sexM    100.00
## father   13.55
## mother    0.00
```

Regressão Linear - Pensamentos Finais

- Fizemos só as técnicas básicas
- Existem muitas extensões que preservam a caráter linear
  - Polinomial - permite curvas mais complexas mas no formato linear $Y = \beta_0 + \beta_1 X + \beta_1 X^2 + \beta_1 X^3 + ... + \epsilon$
  - *Stepwise* - acrescentar variáveis múltiplas 1 por vez para testar quantos produzem o melhor modelo
  - Ridge, Lasso, etc., etc.

Exemplo Final da Regressão - `gapminder`

- Pacote R derivado do site https://www.gapminder.org/
- Monitora condições socio-economicas no mundo
- Fruto das pesquisas do Hans Rosling e família
- Eles acham que pobreza no mundo pode ser eliminada por 2030
- Assiste o video:
  https://www.gapminder.org/videos/dont-panic-end-poverty/
- Empolgante!

Tidymodel

- Novo metapacote como `tidyverse` com pacotes que executam todos os passos de `caret`
- Mesmo autor que o `caret`
- Ainda estou aprendendo como utilizar
- Vale algum investimento
- Mesmo passos que `caret` mas com diferente nomes e mais detalhes possíveis

Tipos de Técnicas de Classificação

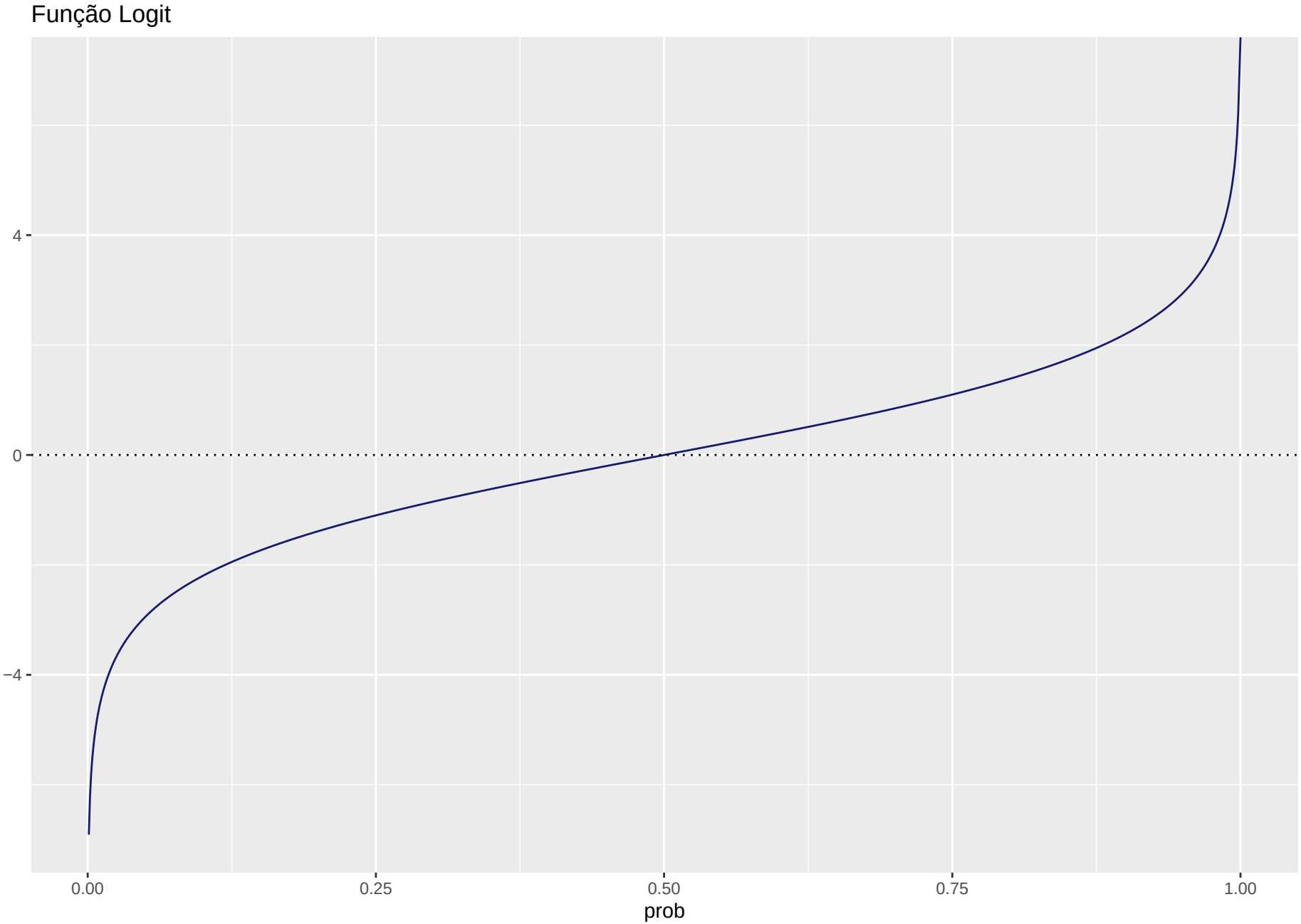
Regressão Logística

- Tipo usado freqüentemente
- Extensão do conceito básico da regressão linear
  - como regressão polinomial, *stepwise*
- Variável dependente (`Y`) agora é **binomial**
  - Tem 2 estados:
    - `TRUE`; `FALSE`
    - 1 ; 0
    - “comprar” ; “não comprar”
    - “passou no teste” ; “reprovou no teste”
- As variáveis independentes podem ser numéricas ou categóricas

Função `logit`

- *log-odds*
- *odds* de um evento $= p/(1-p)$
  - probabilidade do evento dividido pela probabilidade que não acontecerá
- **logit** é o logaritmo natural dos odds

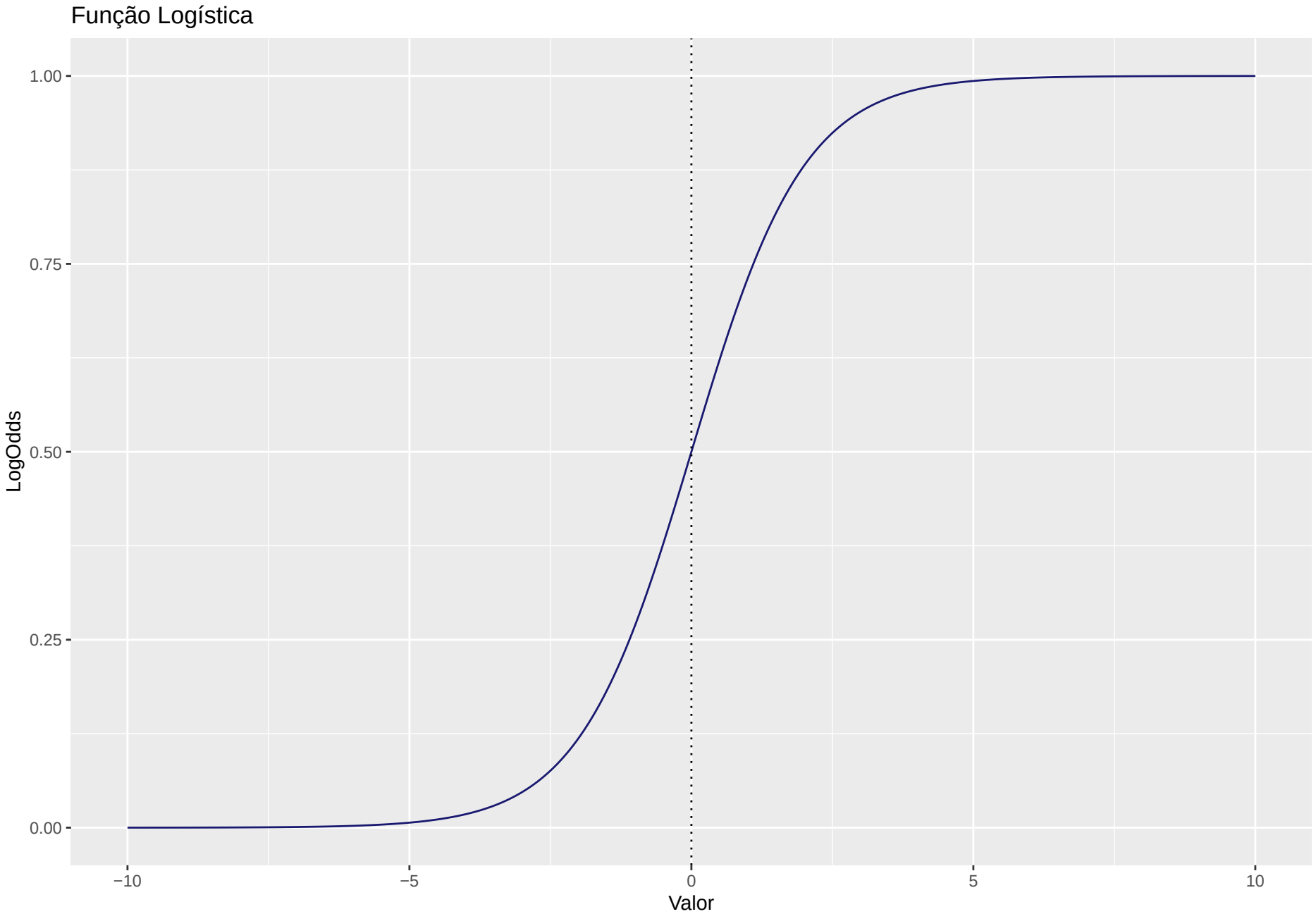
$$logit(p) = \frac{p}{1-p}$$


Função Logit
4
0
−4
0.00
0.25
0.50
0.75
1.00
prob

Função Logística

- Aplicamos função para as variáveis independentes (X)
  - Resultado: Variável dependente fica no intervalo entre 0 e 1
    - ★ intervalo de probabilidades
- Função **Logística**
  - Inverso de **logit**
  - Aplicável a qualquer número

$$logit^{-1}(x) = \frac{1}{1 + e^{-x}}$$


Função Logística
1.00
0.75
0.50
0.25
0.00
LogOdds
−10
−5
0
5
10
Valor

Comparar Regressão Linear com Regressão Logística

- Regressão Linear (usando notação de matrizes)

$$y = X\beta + \epsilon_i$$

- Regressão Logística

$$p(y_i = 1) = logit^{-1}(X_i\beta) + \epsilon_i$$

Modelos Lineares Gerais (General Linear Models)

- Regressão logística faz parte dessa classe dos modelos: **GLM**
- Eles manipulam os matrizes diferente do modelo linear simples
- Outros modelos GLM: poisson (dados de contagem)
- Output seria semelhante com o output do regressão simples

Exemplo Simples

- Estudo de 100 pacientes que têm ou não têm doença cardíaca coronária (CHD)
- Estudo interessado na relação entre a idade do paciente e a CHD
- Dados vêm de Hosmer & Lemeshow, *Applied Logistic Regression* (2a Ed.)
  - No arquivo `chdage.csv`

Carregar os Dados

```
chdage <- read_csv(here::here("chdage.csv")) %>%
  mutate(chd = factor(chd)) %>%
  mutate(chd = fct_recode(chd, negativo = "0", positivo = "1"))
glimpse(chdage)
```

```
## Rows: 100
## Columns: 3
## $ id    <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18...
## $ idade <dbl> 20, 23, 24, 25, 25, 26, 26, 28, 28, 29, 30, 30, 30, 30, 30, 3...
## $ chd   <fct> negativo, negativo, negativo, negativo, positivo, negativo, n...
```

Analise Básica Exploratória

```
chdage %>%
  select(idade) %>%
  descr(stats = c("mean", "sd", "min", "q1", "med", "q3",
                  "max", "iqr", "cv"))
```

```
## 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 was generated.

## Descriptive Statistics
## chdage$idade
## N: 100
##
##                 idade
## ------------- -------
##          Mean   44.38
##       Std.Dev   11.72
##           Min   20.00
##            Q1   34.50
##        Median   44.00
##            Q3   55.00
##           Max   69.00
##           IQR   20.25
##            CV    0.26
```

Boxplot da Idade

```
chdbox <- ggplot(data = chdage, aes(x = chd, y = idade, group = chd))
chdbox <- chdbox + geom_boxplot()
chdbox
```

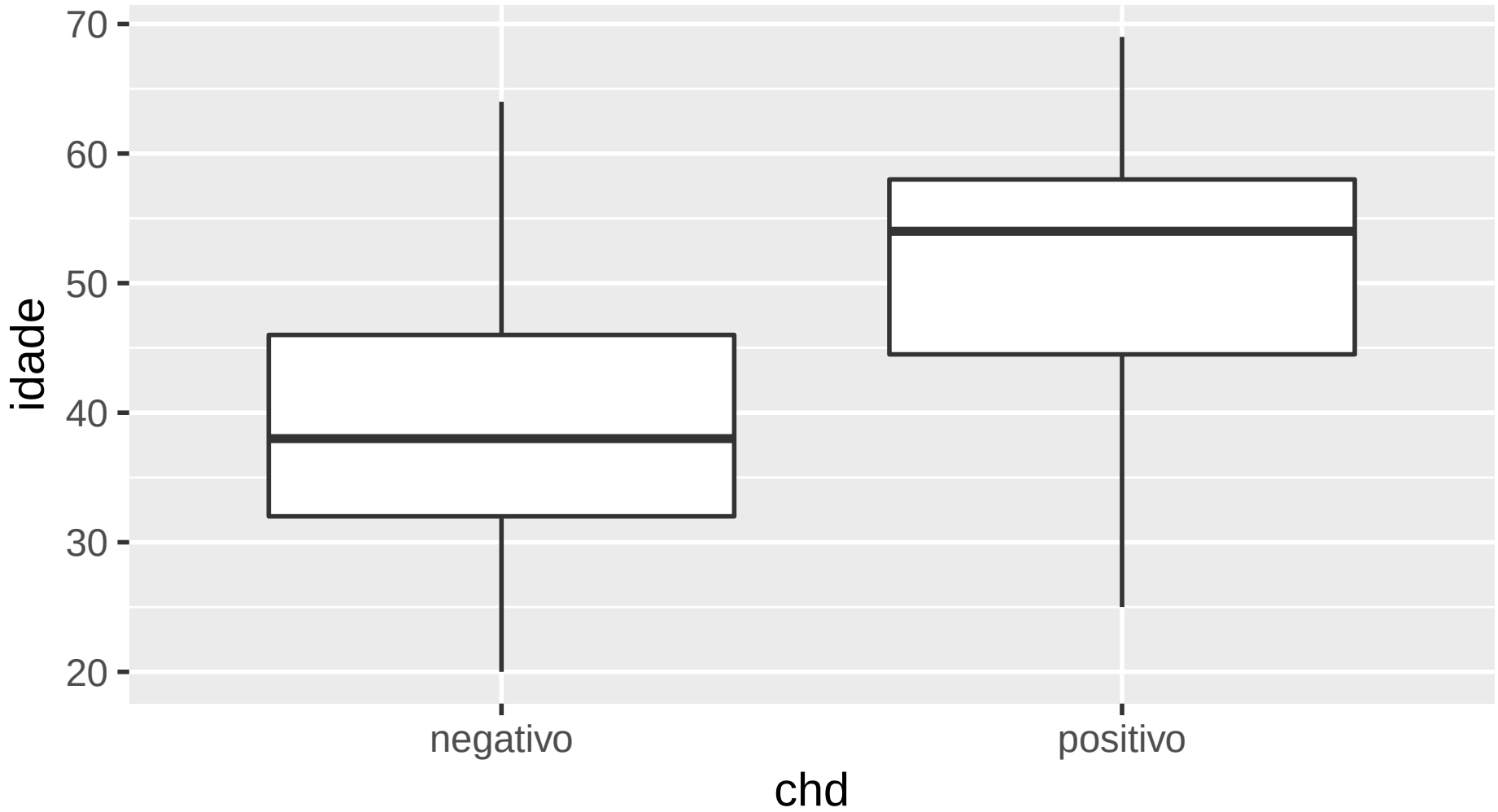
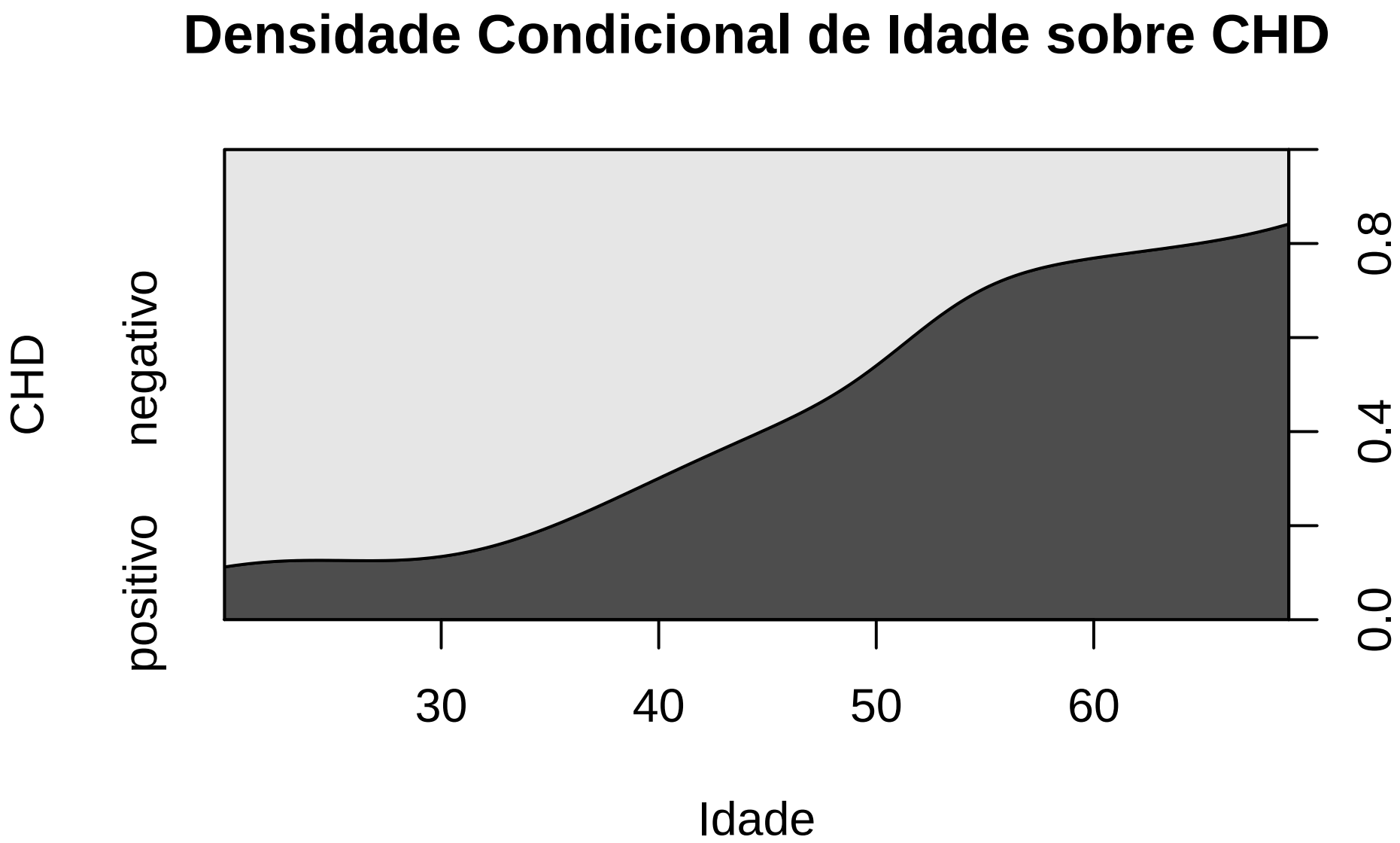


Gráfico de Densidade Condicional

- Também útil para entender como idade muda nas 2 categorias de CHD
- Mostra o número daqueles com a doença (chd $= 1$) para todos as idades
  - Numa forma continua

```r
cdplot(factor(chd) ~ idade, data = chdage,
       main = "Densidade Condicional de Idade sobre CHD",
       xlab = "Idade", ylab = "CHD")
```



Modelo

- Como o pacote `lm`, `glm` usa o formato de formula para especificar o modelo
  - variável dependente ~ variáveis independentes
  - variáveis independentes separados com +
- Fonte dos dados (`data =`)
- `Family` dos modelos (neste caso, `binomial`)
- Função `link` (neste caso, `logit`)

```
chdfit1 <- glm(chd ~ idade, data = chdage,
               family = binomial(link = "logit"))
```

Resultados

- Obter os resultados como no `lm`, com `summary`
- Também podemos olhar nos coeficientes com um gráfico chamada `coefplot`
- Vem de pacote de mesmo nome

Coeficientes do Modelo

```
summary(chdfit1)
```

```
##
## Call:
## glm(formula = chd ~ idade, family = binomial(link = "logit"),
##     data = chdage)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9718  -0.8456  -0.4576   0.8253   2.2859
##
## Coefficients:
##             Estimate Std. Error z value   Pr(>|z|)
## (Intercept) -5.30945    1.13365  -4.683 0.00000282 ***
## idade        0.11092    0.02406   4.610 0.00000402 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 136.66  on 99  degrees of freedom
## Residual deviance: 107.35  on 98  degrees of freedom
## AIC: 111.35
##
## Number of Fisher Scoring iterations: 4
```

Entender os Coeficientes

- Parecido com o que nós conhecemos da regressão linear
- Os coeficientes em si representam os *log odds* que o resultado $Y = 1$.
- Para entender os coeficientes melhor, precisa calcular o *logit inverso*
- Este põe os coeficientes no intervalo entre 0 e 1
  - ou seja, probabilidade

Logit Inverso

```
invlogit <- function(x) {
   1/(1 + exp(-x))
}
invlogit(chdfit1$coefficients[2])
```

```
##     idade
## 0.5277019
```

- Assim, podemos interpretar os resultados como probabilidades
- Com uma probabilidade acima de 50%, podemos dizer que idade tem uma relação positiva com a ocorrência de CHD

Desvio e AIC

- 2a parte dos resultados são os equivalentes de R^2, medidas de qualidade do modelo
- Invés da variância, com `glm` falamos de desvio
- Queremos minimizar o *desvio residual*
- AIC = Akaike's Information Criterion (aqui = 111.3530927)
- AIC útil para comparar modelos
  - Nota menor melhor

Este Modelo

- Desvio Residual = 107.3530927
- AIC = 111.3530927

Segundo Modelo para Comparação

- Modelo com Idade categórica – grupos de idade
- Esperança que podemos entender melhor as probabilidades relacionados aos grupos de idade mais específicos
  - Idosos mais propensos a CHD?
- Vamos usar `recode` do pacote `car`
  - Mais flexível que `recode` de `dplyr`

Grupos de Idade

```
chdage$idgrp <- car::Recode(chdage$idade, "20:29 = '20-29'; 30:34 = '30-34';
                            35:39 = '35-39'; 40:44 = '40-44'; 45:49 = '45-49';
                            50:54 = '50-54'; 55:59 = '55-59'; 60:69 = '60-69'",
                            as.factor = TRUE)
```

Modelo de Grupos

```
chdfit2 <- glm(chd ~  idgrp, data = chdage,
               family = binomial(link = "logit"))
```

Resultados

```
summary(chdfit2)
```

```
##
## Call:
## glm(formula = chd ~ idgrp, family = binomial(link = "logit"),
##     data = chdage)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7941  -0.9005  -0.4590   0.7325   2.1460
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.1972     1.0540  -2.085  0.03710 *
## idgrp30-34    0.3254     1.2992   0.250  0.80221
## idgrp35-39    1.0986     1.2471   0.881  0.37837
## idgrp40-44    1.5041     1.1878   1.266  0.20543
## idgrp45-49    2.0431     1.1918   1.714  0.08649 .
## idgrp50-54    2.7081     1.2823   2.112  0.03470 *
## idgrp55-59    3.3759     1.1991   2.815  0.00487 **
## idgrp60-69    3.5835     1.3175   2.720  0.00653 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 136.66  on 99  degrees of freedom
## Residual deviance: 107.96  on 92  degrees of freedom
## AIC: 123.96
##
## Number of Fisher Scoring iterations: 4
```

Idosos Têm Alta Probabilidade de CHD

```
invlogit(coef(chdfit2)[5:8])
```

```
## idgrp45-49 idgrp50-54 idgrp55-59 idgrp60-69
##  0.8852459  0.9375000  0.9669421  0.9729730
```

Qual modelo parece melhor?

- Modelo 1 – Idade Numérica
  - Desvio Residual = 107.3530927
  - AIC = 111.3530927
- Modelo 2 – Idade Categórica
  - Desvio Residual = 107.9614654
  - AIC = 123.9614654
- AIC melhor no modelo numérico
- Mas, modelo categórico oferece mais informação sobre grupos de idade de interesse

Alternativas – Técnicas de Classificação

- Arvores de Decisão
  - Vêm em vários sabores em `caret` e `tidymodels`
  - `rpart` - *recursive partitioning* - determinar uma arvore que cria uma classificação da variável dependente
  - `randomForest` - criar um monte de arvores (tb., `ranger`)
    - ★ tem bons resultados
- Principal Components Analysis (PCA) * Ajuda para reduzir o número de covariados a um número razoável * Controlar o problema de *kitchen sink*
- Primo perto da técnica de análise de fatores

Como Usar O Que Aprendemos

- Dei para vocês uma introdução a R e como usar ele para solucionar problemas
- Como em qualquer situação de aprendizagem à distância, vocês precisam fazer o trabalho
  - Vocês conhecem onde estão uma variedades de recursos que facilitam o uso de R
  - Aproveitem deles
- Mais que você usa R, mais fácil seria
- Se tiver duvidas, pergunte!
  - Nos foruns
  - Nos sites
  - Não esqueça Dr. Google!
- Eu uso R approx. 10 anos e faço cursos e acho novos livros todos os anos

Um Último Recurso

- Curso: STAT545 de Universidade de British Columbia
- Professor: Jenny Bryan
- No formato de um livro eletrônico
- https://stat545.com/index.html

Obrigado!