

ANÁLISE DOS DADOS COM R

Ferramentas de Machine Learning

James R. Hunter, PhD

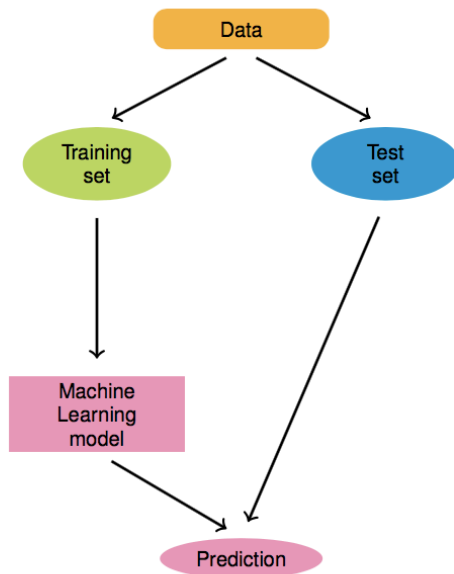
Retrovirologia, EPM, UNIFESP

2023-10-17



TIPOS DE *MACHINE LEARNING*

Supervised



Unsupervised

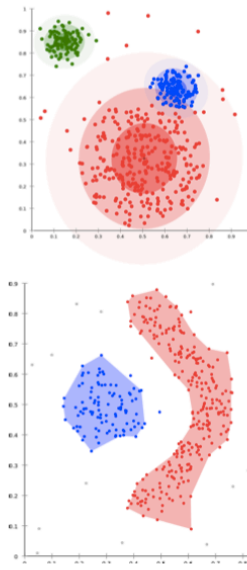


Image Source: Wikipedia

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

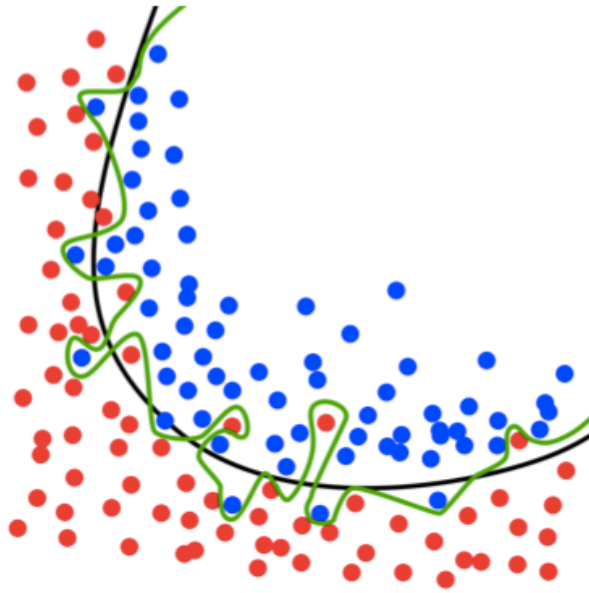


Image Source: Wikipedia

CARACTERÍSTICAS DOS MODELOS

- Covariáveis
 - Quantos são suficientes para construir um modelo?
 - Número insuficiente – modelo não descreve suficiente a condição
 - Número demais – overfitting

FORTELECER O MODELO

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

MACHINE LEARNING EM MODELAGEM BIOLÓGICA/MÉDICA

- Tipicamente, projetos com “big data”
- Modelo pode fornecer informação rapidamente e corretamente
 - Médicos podem usar a informação para desenhar tratamentos ou diagnósticos
- Aplicação para medicina personalizada de precisão
- Exemplo:
 - 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 entre 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 signifiquem 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”

VAMOS PÔR AS MÃOS NA MASSA

DADOS

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

```
1 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, 5, ...
$ father <dbl> 78.5, 78.5, 78.5, 78.5, 75.5, 75.5, 75.5, 75.5, 75.0, 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, 64.0, ...
$ sex    <fct> M, F, F, F, M, M, F, F, M, F, M, M, F, F, F, M, M, M, F, F, F, ...
$ height <dbl> 73.2, 69.2, 69.0, 69.0, 73.5, 72.5, 65.5, 65.5, 71.0, 68.0, 70.0, ...
$ nkids  <int> 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 5, 5, 5, 5, 5, 6, 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
- 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

```
1 pacman::p_load(caret, ggpubr, broom, nortest, janitor)
```

PROCESSO DE **caret**

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

```
1 Define sets of model parameter values to evaluate
2 for each parameter set do
3   for each resampling iteration do
4     Hold-out specific samples
5     [Optional] Pre-process the data
6     Fit the model on the remainder
7     Predict the hold-out samples
8   end
9   Calculate the average performance across hold-out predictions
10 end
11 Determine the optimal parameter set
12 Fit the final model to all the training data using the optimal parameter set
```

caret DIVISÃO DOS DADOS

- função `createDataPartition()`
 - Dar para função a variável dependente `galton$height`
 - Proporção (`p`) que você quer na amostra de treinamento (70%)
 - Pode ser entre 50% e 70%
 - Mais pode causar *overfitting*
 - Função retorna os índices dos casos do conjunto de treinamento
 - Argumento `list = FALSE`


```
1 set.seed(42)
2 indice <- createDataPartition(galton$height, p = 0.70, list = FALSE)
3 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 (`-`)

```
1 train_data <- galton[indice, ]  
2 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 tenha deixado fora todos os folds
- Pode repetir o processo inteiro um número das vezes

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 DAS ALTURAS

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

[illegible]

```
1 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:

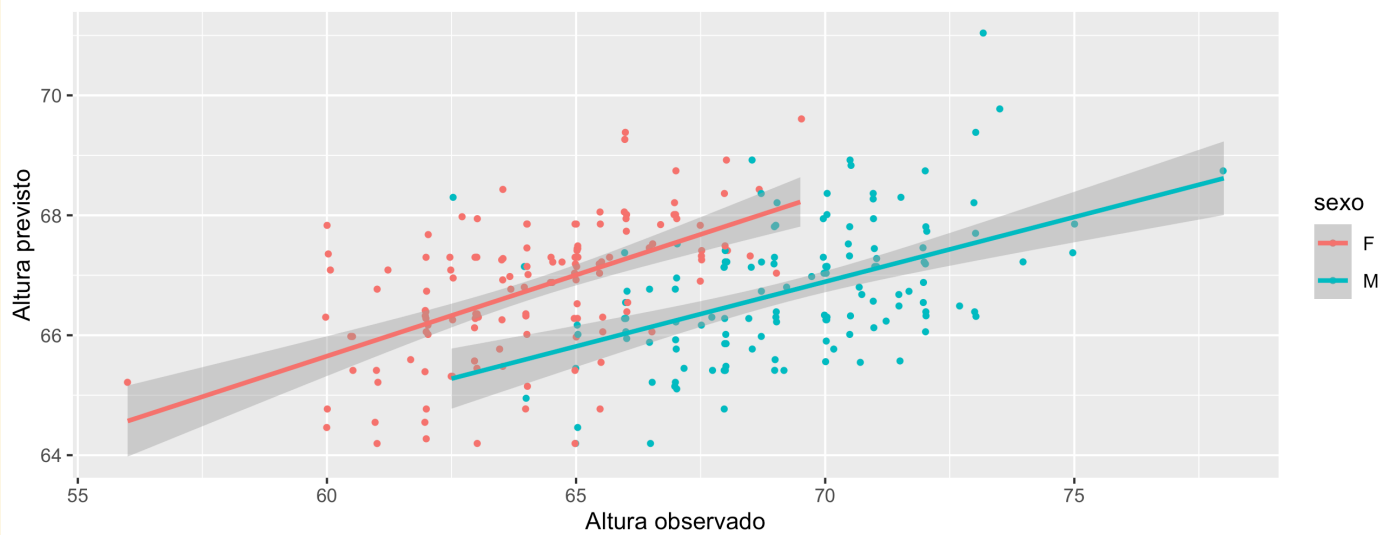
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	23.59851	5.08952	4.637	4.31e-06	***
father	0.37731	0.05589	6.751	3.34e-11	***
mother	0.26601	0.05870	4.532	7.00e-06	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

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

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)

```
1 pred <- predict(fit_pai_mae, test_data)
2 res <- tibble(pred = pred,
3               obs = test_data$height,
4               dif = obs - pred)
5 padrao_in <- 2
6 # teste de bom, ruim
7 res <- res %>%
8   mutate(bomruim = ifelse(abs(dif) <= padrao_in, "bom", "ruim"))
9 tabyl(res$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 categórica
- 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

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

INCLUIR **sex** NA REGRESSÃO

[illegible]

```
1 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

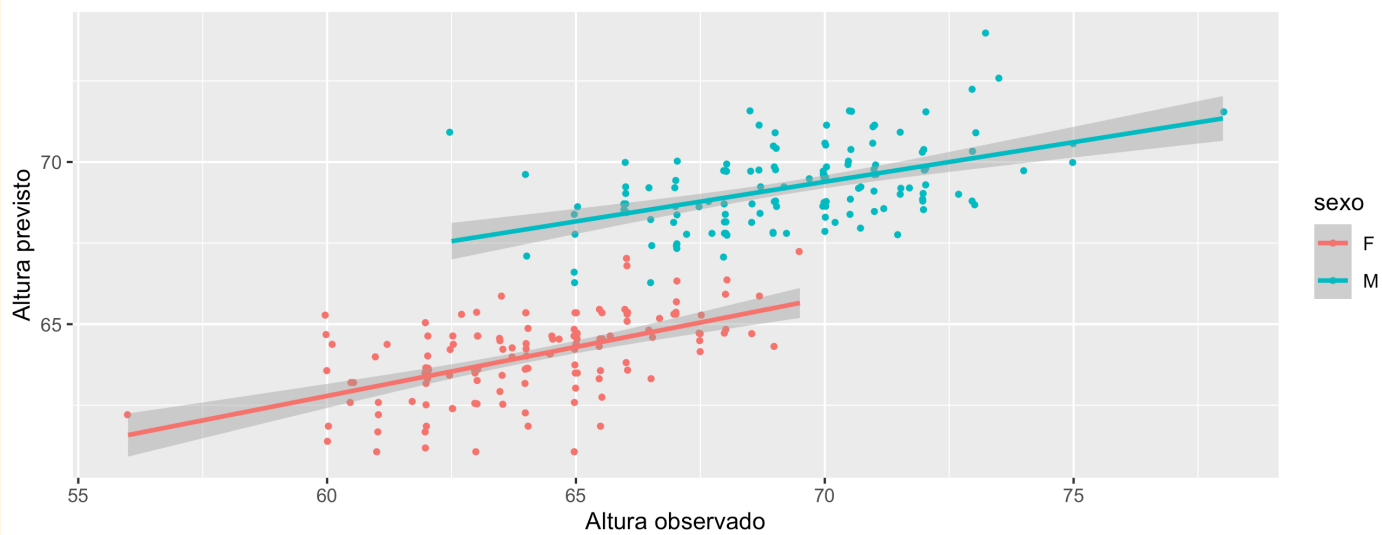
Coefficients:

	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

DESEMPENHO DO MODELO

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



QUANTA PRECISÃO TEVE O MODELO?

```
1 pred <- predict(fit_pms, test_data)
2 res_pms <- tibble(pred = pred,
3                   obs = test_data$height,
4                   dif = obs - pred)
5 padrao_in <- 2
6 # teste de bom, ruim
7 res_pms <- res_pms %>%
8   mutate(bomruim = ifelse(abs(dif) <= padrao_in, "bom", "ruim"))
9 tabyl(res_pms$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 característica

varImp () FUNÇÃO EM caret

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

```
1 varImp(fit_pms)
```

```
lm variable importance
```

	Overall
sexM	100.00
father	13.55
mother	0.00

EXEMPLO - 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

O QUE ESTE ENSINA?

- Expectativa de vida (*lifeExp*) - variável dependente
 - Medida por país
- Nossa hipótese é que expectativa de vida depende em:
 - O ano pesquisado (1952 - 2007 a cada cinco anos)
 - Com a passagem do tempo (aumento no ano), expectativa de vida aumenta naturalmente
 - Produto Interno Bruto (*GDP*) por capita

Expectativa de vida como medida da saúde dos países aumenta com base no bem-estar económico da população. Tornou-se melhor desde a década de 1950.

QUESTÃO FILOSÓFICA

- Objectivo dos modelos de *Machine Learning*: previsão precisa
 - As sutilezas de obedecer a todas as suposições e testes de hipóteses estatísticas não são tão importantes
- Objectivo dos modelos estatísticos: relacionar os dados da amostra para uma verdade maior sobre uma população
 - Suposições, testes de hipóteses, intervalos de confiança são todos muito importantes

HIPÓTESES NULAS E ALTERNATIVAS

- Se construimos um modelo estatístico, temos que estabelecer primeiro uma hipótese nula
- H_0 : Expectativa de vida não varia por causa das variáveis independentes (só tem intercepto no modelo)
 - $H_0 : Y_i = b_0 + \epsilon_i$
- H_1 : Expectativa de vida tem relação com pelo menos uma das duas covariáveis

$$H_1 : Y_i = \left(\sum_{k=1}^K b_k X_{ik} \right) + b_0 + \epsilon_i$$

CARREGAR gapminder

```
1 gm <- gapminder::gapminder %>%  
2   janitor::clean_names()  
3 glimpse(gm)
```

Rows: 1,704

Columns: 6

\$ country <fct> "Afghanistan", "Afghanistan", "Afghanistan",
"Afghanistan",...

\$ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia,
Asia,...

\$ year <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992,
1997,...

\$ life_exp <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854,
40....

\$ pop <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372,
1...

\$ gdp_percap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811,
786.1134,...

ESTADÍSTICAS DESCRIPTIVAS

```
1 gm %>%
2   select(year, pop, gdp_percap) %>%
3   mutate(pop = log10(pop)) %>%
4   descr(stats = c("mean", "sd", "min", "q1", "med", "q3", "max", "iqr", "cv"),
5         transpose = TRUE)
```

Descriptive Statistics

gm

N: 1704

	Mean	Std.Dev	Min	Q1	Median	Q3	Max	IQR	CV
gdp_percap	7215.33	9857.45	241.17	1201.92	3531.85	9325.86	113523.13	8123.40	1.37
pop	6.85	0.70	4.78	6.45	6.85	7.29	9.12	0.85	0.10
year	1979.50	17.27	1952.00	1964.50	1979.50	1994.50	2007.00	27.50	0.01

```
1 paste("Correlation Coefficient (year x life):", with(gm, round(cor(life_exp, year), 3)))
```

[1] "Correlation Coefficient (year x life): 0.436"

```
1 paste("Correlation Coefficient (life x gdp):", with(gm, round(cor(life_exp, gdp_percap), 3)))
```

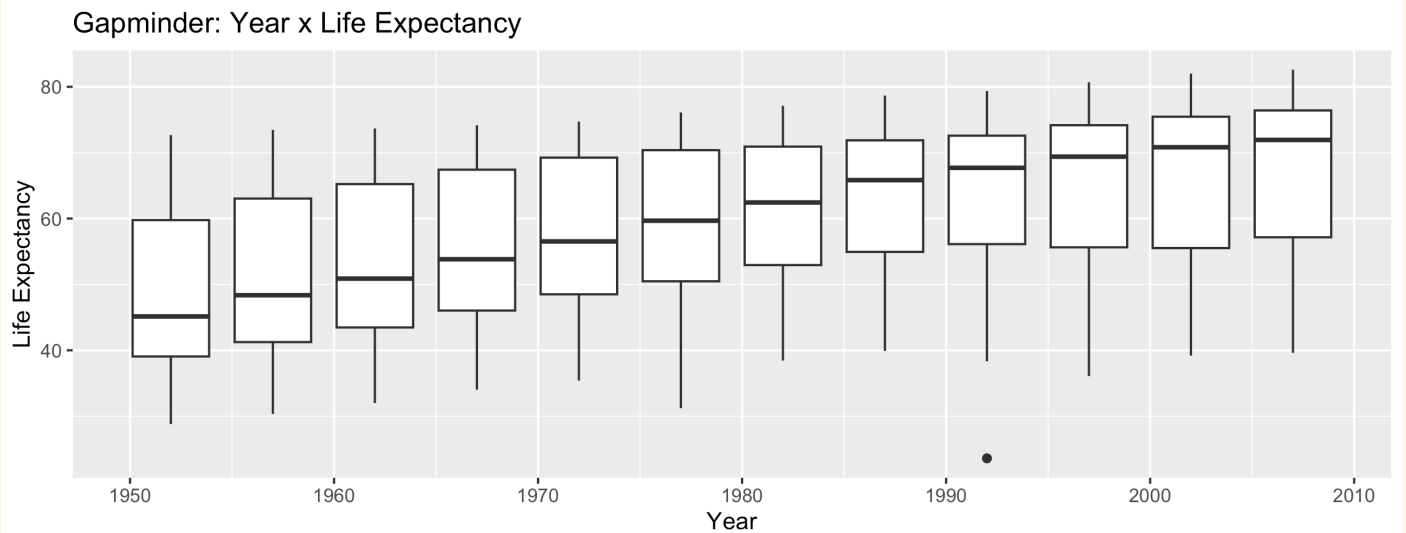
[1] "Correlation Coefficient (life x gdp): 0.584"

```
1 paste("Correlation Coefficient (gdp x life):", with(gm, round(cor(gdp_percap, year), 3)))
```

[1] "Correlation Coefficient (gdp x life): 0.227"

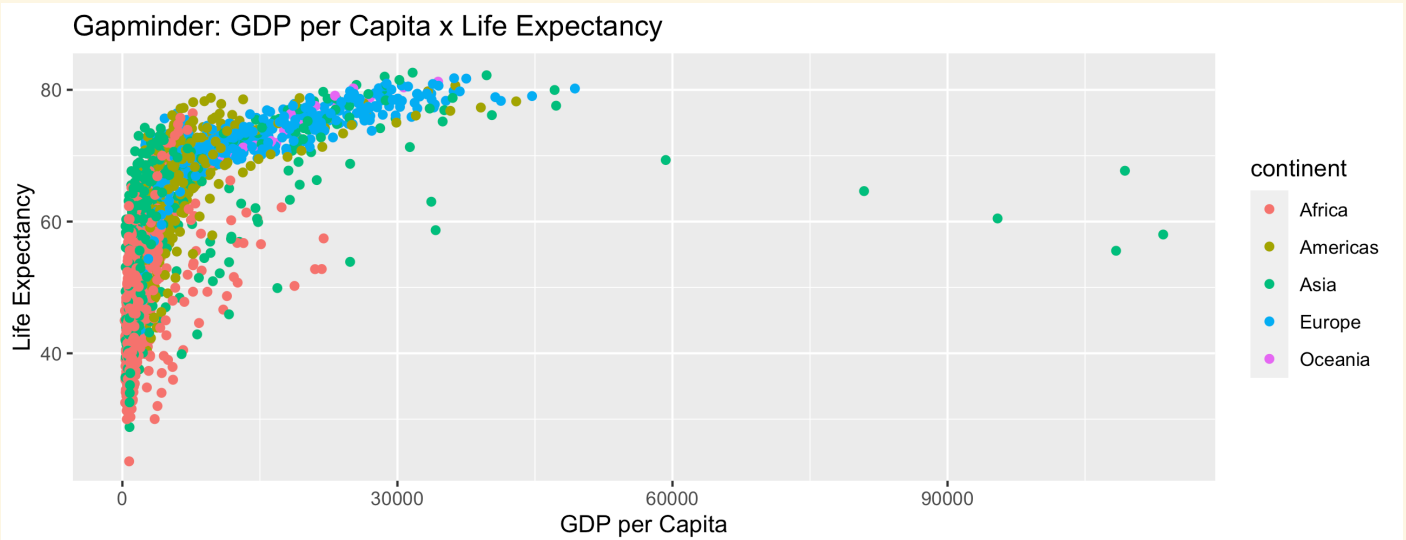
BOXPLOT: life_exp x year

```
1 ggplot(gm, aes(x = year, y = life_exp, group = year)) +  
2   geom_boxplot() +  
3   labs(title = "Gapminder: Year x Life Expectancy", x = "Year", y = "Life E
```



SCATTERPLOT: life_exp x gdp_percap

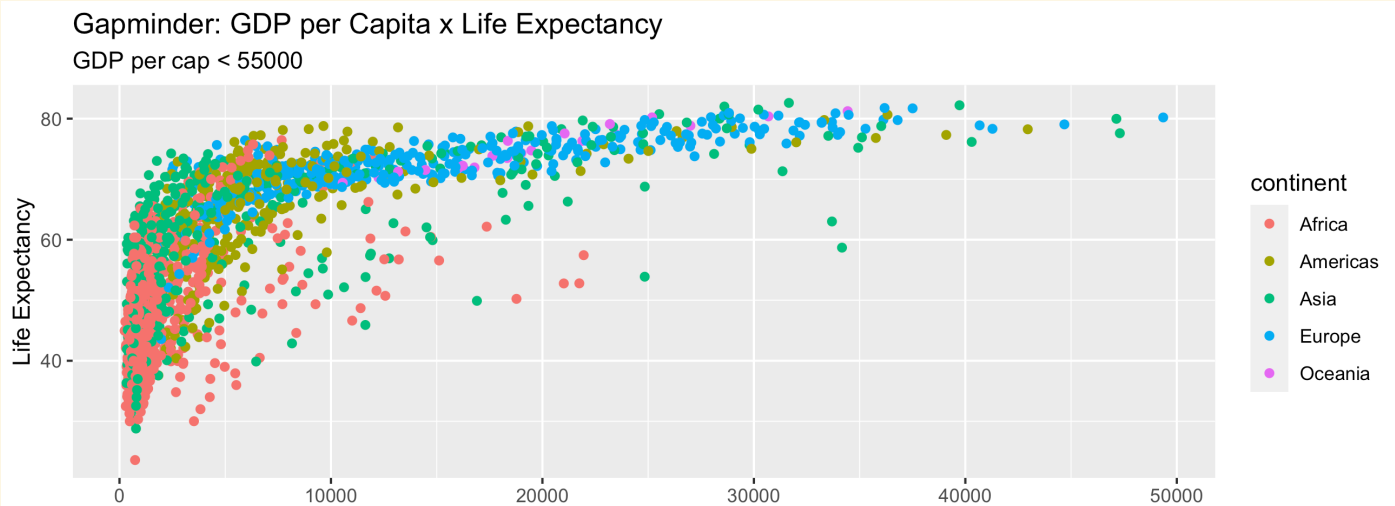
```
1 ggplot(gm, aes(x = gdp_percap, y = life_exp, color = continent)) +  
2   geom_point() +  
3   labs(title = "Gapminder: GDP per Capita x Life Expectancy", x = "GDP per
```



SCATTERPLOT: life_exp x gdp_percap - 2

- Livre-se do PIB per capita muito alto para ver a massa com mais clareza

```
1 gm %>%  
2   filter(gdp_percap < 55000) %>%  
3   ggplot(aes(x = gdp_percap, y = life_exp, color = continent)) +  
4     geom_point() +  
5     labs(title = "Gapminder: GDP per Capita x Life Expectancy", x = "GDP per Capita", y = "Life Expectancy",
```



INICIALIZAR **caret** E OS PACOTES RELACIONADOS

```
1 pacman::p_load(caret, tidyverse, broom, nortest, janitor)
```

ESTABELECEER OS CONJUNTOS DE TREINAMENTO E TESTE

```
1 set.seed = 42
2 index <- createDataPartition(gm$life_exp, p = 0.7, list = FALSE)
3 head(index[, 1], 25)
```

```
[1] 1 3 5 6 8 9 11 12 13 14 15 16 17 18 19 20 21 22 23 25 26 27 28 30
31
```

```
1 gm_train <- gm[index, ]
2 gm_test <- gm[-index, ]
```


PLANO PARA VALIDAÇÃO CRUZADA

- 142 países: divide dados em 10 *folds*
 - 14.2 países por *fold*
- Repetições de validação cruzada
 - Continuar com 10 repetições que usamos antes

train PARA CONSTRUIR MODELO

[illegible]

```
1 summary(fit_gm_1)
```

Call:

```
lm(formula = .outcome ~ ., data = dat)
```

Residuals:

Min	1Q	Median	3Q	Max
-64.225	-6.986	1.251	7.754	19.841

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-435.89173435	32.86049457	-13.27	<2e-16 ***
year	0.24780454	0.01662278	14.91	<2e-16 ***
gdp_percap	0.00066402	0.00002878	23.07	<2e-16 ***

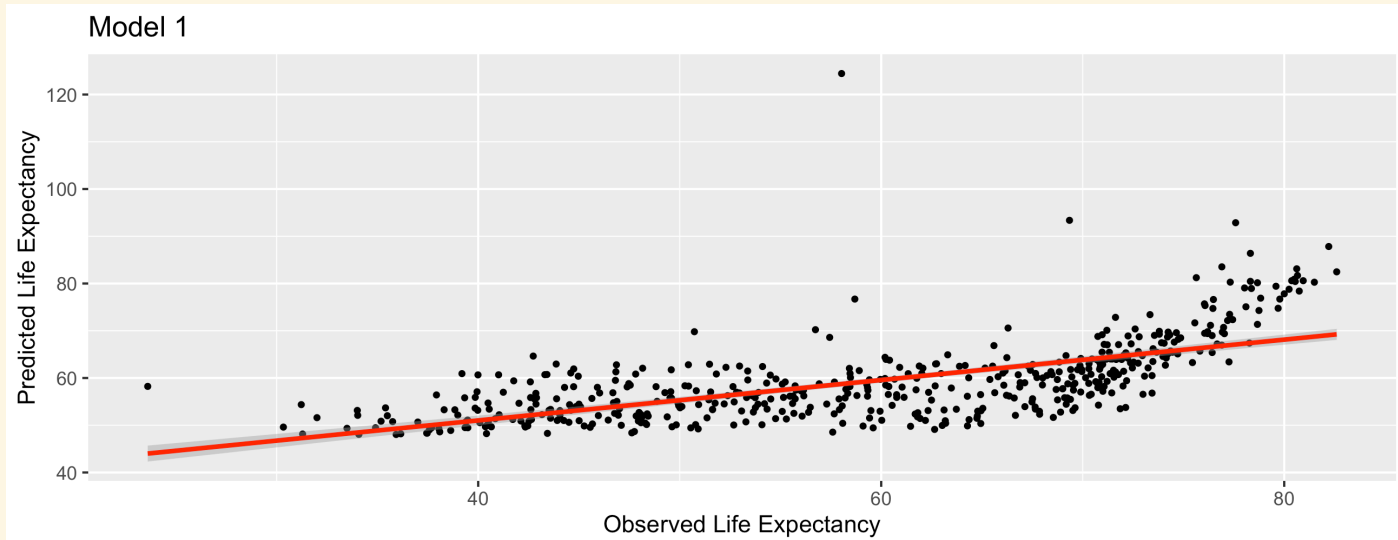
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

DESEMPENHO DO MODELO

- Aplicar o modelo para os dados de teste
 - Até este ponto, modelo não viu os dados de teste

```
1 ## predictions
2 pred_1 <- predict(fit_gm_1, gm_test)
3 ## compare
4 gm_pred_1 <- data.frame(obs = gm_test$life_exp,
5                          preds = pred_1)
6 gm_1_plot <- ggplot(gm_pred_1, aes(x = obs, y = preds)) +
7   geom_jitter(shape = 20) +
8   geom_smooth(method = "lm", color = "red")+
9   labs(title = "Model 1", x = "Observed Life Expectancy", y = "Predicted Li
```

1 gm_1_plot



PRECISÃO DE MODELO 1

- Estabelecer um padrão de precisão
 - Valor previsto 2 anos \pm expectativa de vida observada

$$abs(observado - previsto) \leq 2$$

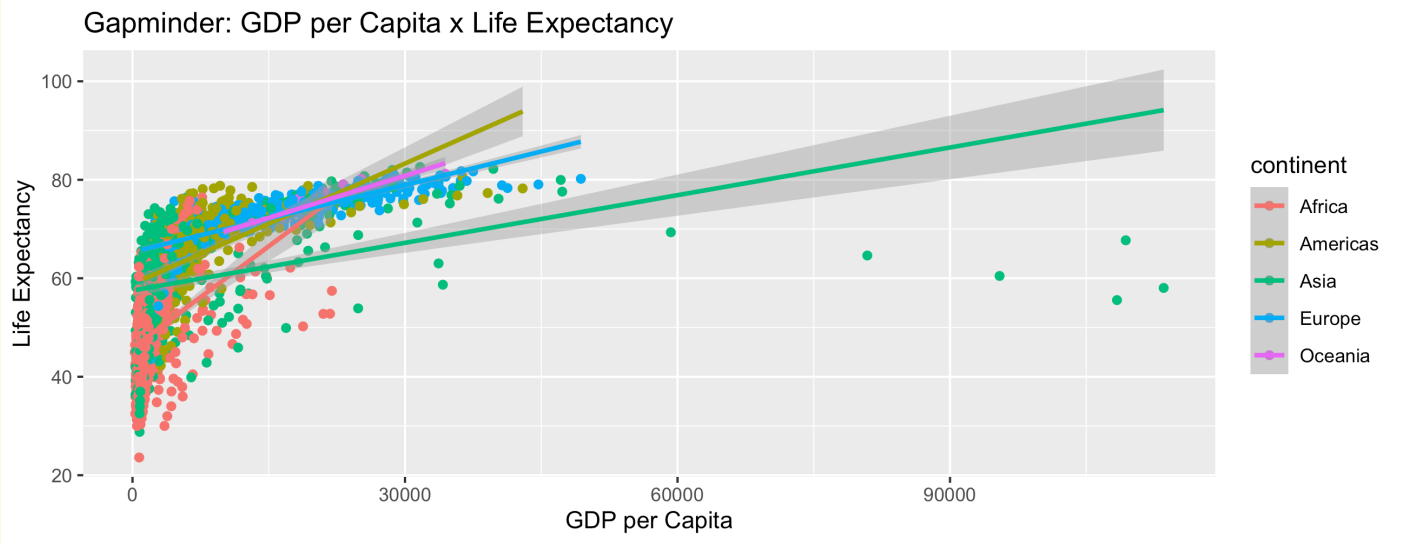
```
1 # calculate difference between predicted and observed
2 gm_pred_1 <- gm_pred_1 %>%
3   mutate(dif = obs - preds,
4           goodbad = ifelse(abs(dif) <= 2, "good", "bad"))
5 tabyl(gm_pred_1$goodbad) %>% adorn_pct_formatting()
```

```
gm_pred_1$goodbad    n percent
      bad 238      46.9%
      good 270      53.1%
```

ESTE MODELO - CONCLUSÃO

- Captura apenas 47% da variação na expectativa de vida
- Continentes parecem de ter um efeito
 - Todos tem inclinações diferentes
- Acrescentar `continent` como uma variável no modelo
- Precisão de 58% não muita boa
- Outliers estranhos e alguns valores de idade previsões de `life_exp` acima de 120 anos

GRÁFICO DOS CONTINENTES COMO FATOR



MODELO # 2 – TRÊS COVARIÁVEIS

[illegible]

```
1 summary(fit_gm_2)
```

Call:

```
lm(formula = .outcome ~ ., data = dat)
```

Residuals:

Min	1Q	Median	3Q	Max
-25.8105	-4.1216	0.2014	4.3880	19.9893

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-523.9697580	23.3526788	-22.44	<2e-16 ***
year	0.2890737	0.0118065	24.48	<2e-16 ***
gdp_percap	0.0002886	0.0000233	12.39	<2e-16 ***
continentAmericas	14.2457830	0.5760049	24.73	<2e-16 ***
continentAsia	9.7975727	0.5376687	18.22	<2e-16 ***
continentEurope	19.5029043	0.6128062	31.83	<2e-16 ***
continentOceania	21.0420867	1.7663765	11.91	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.788 on 1189 degrees of freedom

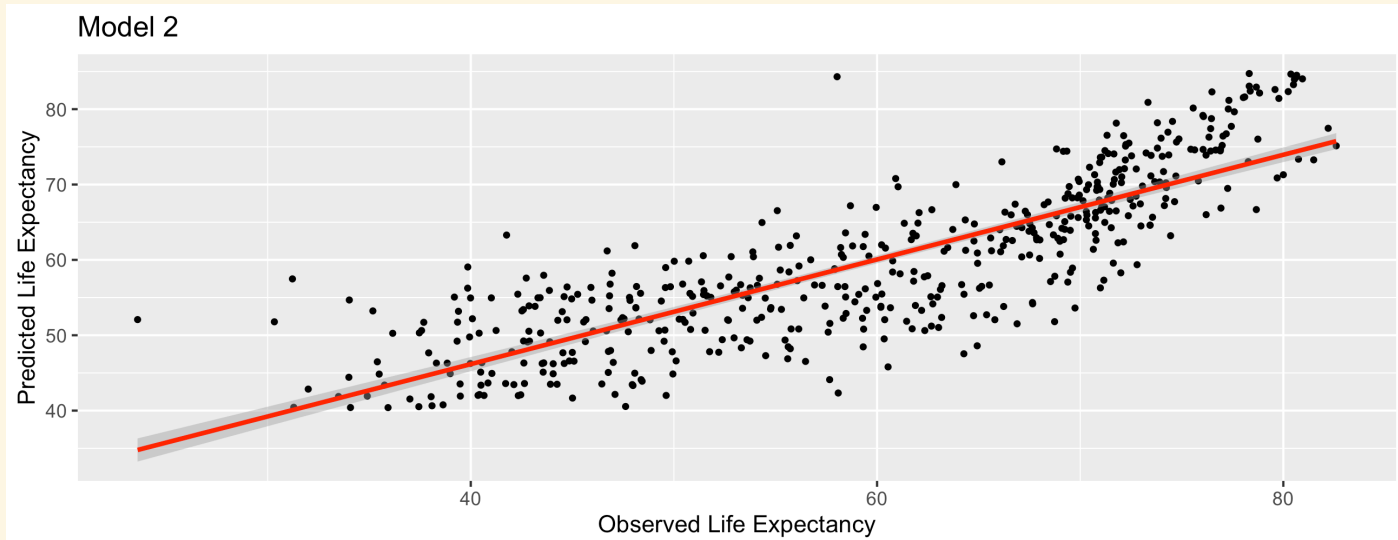
Multiple R-squared: 0.7256, Adjusted R-squared: 0.7243

F-statistic: 524.1 on 6 and 1189 DF, p-value: < 2.2e-16

COMO NÓS SAÍMOS (DESTA VEZ)?

```
1 ## predictions
2 pred_2 <- predict(fit_gm_2, gm_test)
3 ## compare
4 gm_pred_2 <- data.frame(obs = gm_test$life_exp,
5                          preds = pred_2)
6 gm_2_plot <- ggplot(gm_pred_2, aes(x = obs, y = preds)) +
7   geom_jitter(shape = 20) +
8   geom_smooth(method = "lm", color = "red")+
9   labs(title = "Model 2", x = "Observed Life Expectancy", y = "Predicted Li
```

```
1 gm_2_plot
```



PRECISÃO DE MODELO 2

- Estabelecer um padrão de precisão
 - Valor previsto 2 anos \pm expectativa de vida observada

$$abs(observado - previsto) \leq 2$$

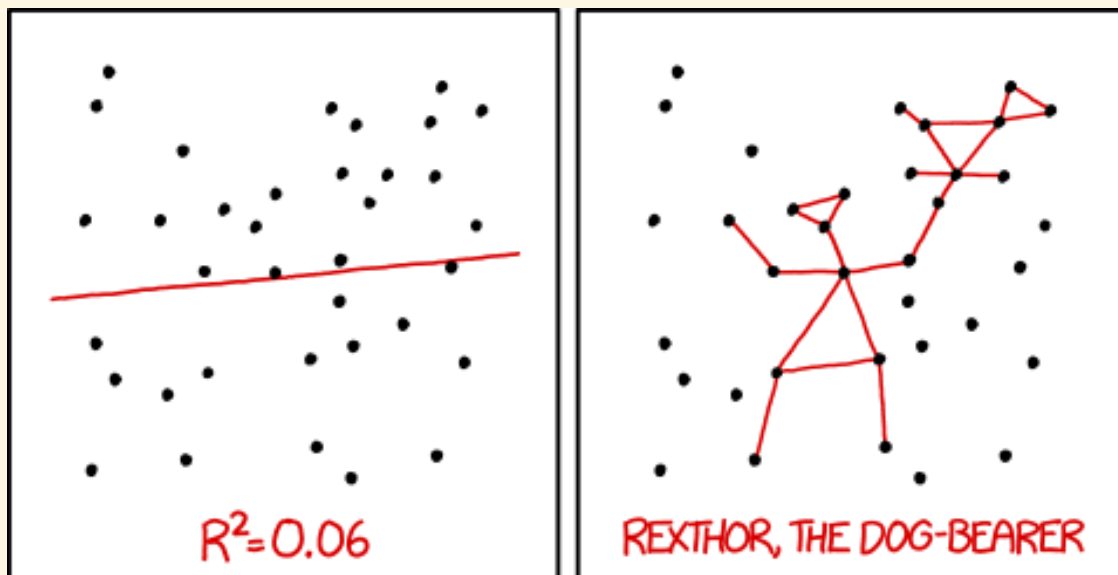
```
1 # calculate difference between predicted and observed
2 gm_pred_2 <- gm_pred_2 %>%
3   mutate(dif = obs - preds,
4          goodbad = ifelse(abs(dif) <= 2, "good", "bad"))
5 tabyl(gm_pred_2$goodbad) %>% adorn_pct_formatting()
```

```
gm_pred_2$goodbad    n percent
      bad 196    38.6%
      good 312    61.4%
```

MODELO 2 - CONCLUSÃO

- R^2 melhor
- Gráfico mostra uma tendência de precisão mais clara (agora 63%)
- Continentes parecem ser fator importante
 - Reflete intuição

PERIGO DE INTERPRETAÇÃO DE R^2 BAIXO



I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER TO GUESS THE DIRECTION OF THE CORRELATION FROM THE SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.