Hands-on R - Regressão Linear Simples

Rodrigo Heldt - rodrigoheldt@gmail.com

IMED - HOPEAD - Nov 13, 2017

Pré-processamento dos dados para análises posteriores

1. Limpando os objetos do ambiente

```
rm(list=ls())
```

2. Carregando (e instalando) pacotes de funções necessários

```
# install.packages('caTools')
# Esse comando deve ser rodado apenas na primeira vez que você usar o pacote para
# instalá-lo em seu computador. Após instalado, a cada vez que for utilizá-lo basta
# carregar o pacote usando a função library

library(caTools)
```

4. Importando a base de dados de um arquivo csv

```
# Usar a funcao read.csv

base <- read.csv("C:/Users/Rodrigo/Dropbox/Hands-on R Workshop/Parte 2 - Pre-processamento dos dados/Da
                 sep = ",")
```

4. Codificando variáveis categóricas

```
# Usar a função as.factor para que o R transforme as variáveis desejadas na classe tipo
# fator (categórico)

base$Mulher = as.factor(base$Mulher)
base$Educacao = as.factor(base$Educacao)
base$PartTime = as.factor(base$PartTime)
```

5. Definindo variáveis numéricas

```
# Usar a função as.numeric para que o R transforme as variáveis desejadas na classe tipo
# numeric (número)

base$SalarioPorHora = as.numeric(base$SalarioPorHora)
base$LogSalarioPorHora = as.numeric(base$LogSalarioPorHora)
base$Idade = as.numeric(base$Idade)
```

6. Divindindo a base em uma parte para treinamento dos modelos e outra para teste das previsões do modelo treinado

```
set.seed(123)
split = sample.split(base$SalarioPorHora, SplitRatio = 0.8)
baseTrain = subset(base, split == TRUE)
baseTest = subset(base, split == FALSE)
```

Regressão Linear Simples

0 Carregar pacotes

```
#install.packages("ftsa")
library(ftsa)
```

1. Ajustando um modelo de regressão linear simples usando a base de treinamento

1.1 Modelo: SalarioPorHora = alpha + Idade

```
reg.simples = lm(formula = SalarioPorHora ~ Idade,
                 data = baseTrain)
summary(reg.simples)
```

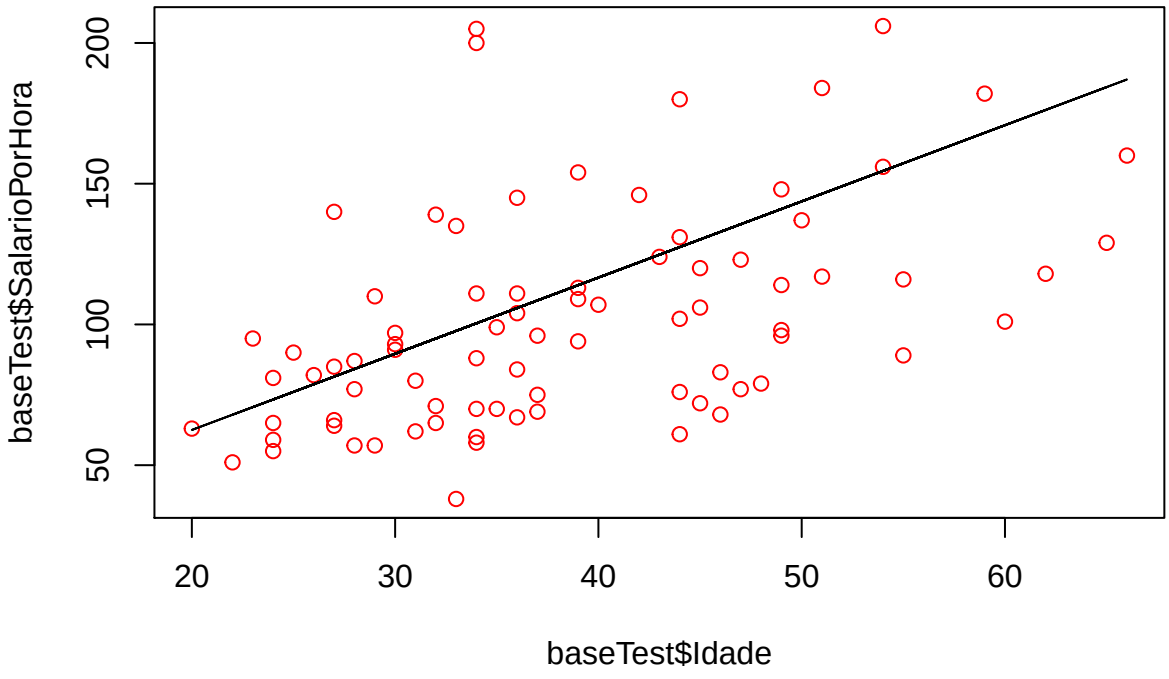
```
##
## Call:
## lm(formula = SalarioPorHora ~ Idade, data = baseTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -91.960 -34.030  -7.117  23.108 242.988
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.3777     9.3312   0.898     0.37
## Idade         2.7068     0.2237  12.102   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.56 on 417 degrees of freedom
## Multiple R-squared:  0.2599, Adjusted R-squared:  0.2582
## F-statistic: 146.5 on 1 and 417 DF,  p-value: < 2.2e-16
```

2. Prevendo os valores para a base de teste a partir dos modelos ajustes

```
## Prevendo os valores do SalárioPorHora para a base de teste
y_pred_simples = predict(reg.simples, newdata = baseTest)
```

3. Verificando a performance das previsões realizadas

```
## Visualizar os resultados da previsão para a base de teste
plot(x = baseTest$Idade, y = baseTest$SalarioPorHora, col = "Red")
lines(x = baseTest$Idade, y = y_pred_simples)
```




```
## Verificando medidas de precisão das previsões
# Mean Absolut Error
mae <- error(true = baseTest$SalarioPorHora, forecast = y_pred_simples, method = "mae")
mae
```

```
## [1] 28.79873
```

```
# Mean Absolut Percentage Error
mape <- error(true = baseTest$SalarioPorHora, forecast = y_pred_simples, method = "mape")
mape
```

```
## [1] 31.97533
```

```
# Root Mean Square Error
rmse <- error(true = baseTest$SalarioPorHora, forecast = y_pred_simples, method = "rmse")
rmse
```

```
## [1] 36.80986
```

4. Exercício

E1 - Ajuste uma regressão simples com a variável dependente LogSalarioPorHora e a variável independente Mulher na base de treinamento

E2 - Faça a previsão do LogSalarioPorHora a partir da variavel Mulher na base de teste

E3 - Cheque o mean absolut error (mae) entre os valores reais e os valores previstos de SalarioPorHora