

01-Solution-Exercicios-Cap11-Parte1.R

rique

2020-12-07

```
# Obs: Caso tenha problemas com a acentuação, consulte este link:
# https://support.rstudio.com/hc/en-us/articles/200532197-Character-Encoding

# Configurando o diretório de trabalho
# Coloque entre aspas o diretório de trabalho que você está usando no seu computador
# Não use diretórios com espaço no nome
setwd("C:/FCD/BigDataRAzure/Cap12")
getwd()
```

```
## [1] "C:/FCD/BigDataRAzure/Cap12"
```

```
## Massa de dados aleatória
```

```
# Criando a massa de dados (apesar de aleatória, y possui
# uma relação com os dados de x)
```

```
x <- seq(0, 100)
```

```
y <- 2 * x + 35
```

```
# Imprimindo as variáveis
```

```
x
```

```
##      [1]      0      1      2      3      4      5      6      7      8      9     10     11     12     13     14     15     16     17
##     [19]     18     19     20     21     22     23     24     25     26     27     28     29     30     31     32     33     34     35
##     [37]     36     37     38     39     40     41     42     43     44     45     46     47     48     49     50     51     52     53
##     [55]     54     55     56     57     58     59     60     61     62     63     64     65     66     67     68     69     70     71
##     [73]     72     73     74     75     76     77     78     79     80     81     82     83     84     85     86     87     88     89
##     [91]     90     91     92     93     94     95     96     97     98     99    100
```

```
y
```

```
##      [1]     35     37     39     41     43     45     47     49     51     53     55     57     59     61     63     65     67     69
##     [19]     71     73     75     77     79     81     83     85     87     89     91     93     95     97     99    101    103    105
##     [37]    107    109    111    113    115    117    119    121    123    125    127    129    131    133    135    137    139    141
##     [55]    143    145    147    149    151    153    155    157    159    161    163    165    167    169    171    173    175    177
##     [73]    179    181    183    185    187    189    191    193    195    197    199    201    203    205    207    209    211    213
##     [91]    215    217    219    221    223    225    227    229    231    233    235
```

```
# Gerando uma distribuição normal
```

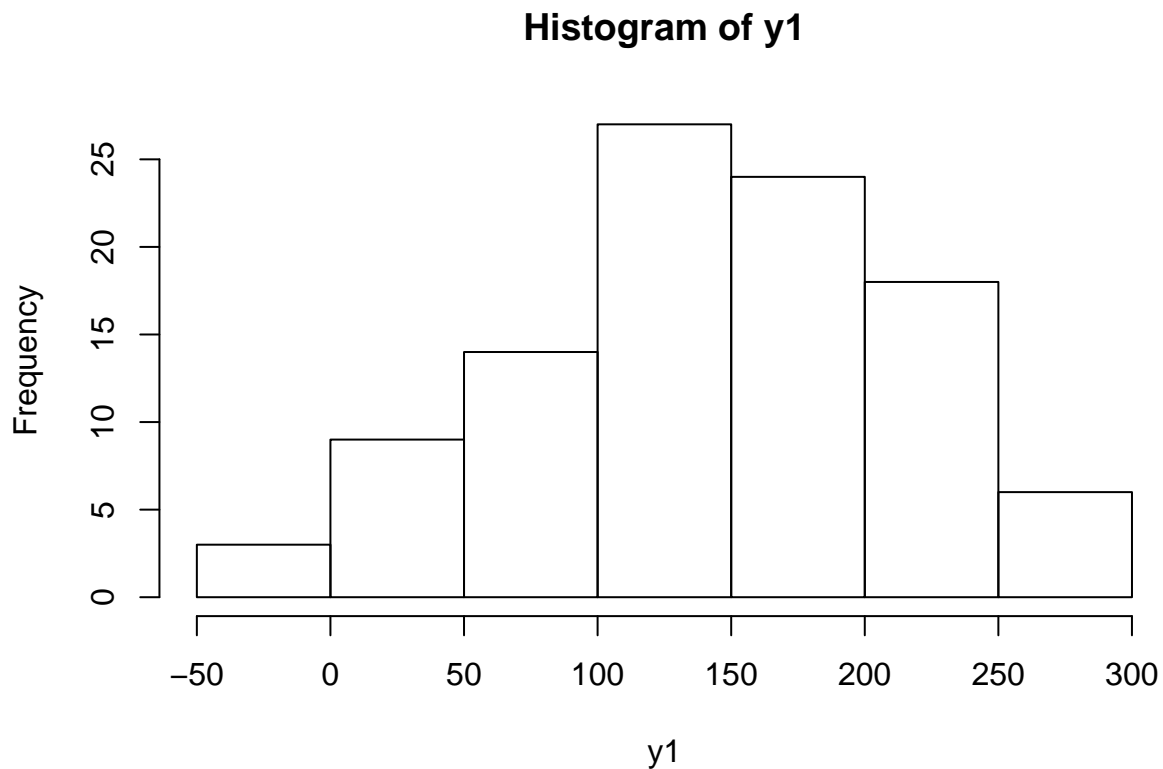
```
y1 <- y + rnorm(101, 0, 50)
```

```
y1
```

```
##      [1]  69.8405340 -0.6709913  58.2081806  48.6880533 132.6382660  66.8881369
##      [7] 138.8792980  22.6425226  52.0384176  38.0842553  22.1209055  97.5531200
##     [13] 107.1112903 132.3868187 127.9094764  61.5104424  31.5248620 -23.0045645
##     [19]   7.6486443  22.1144065  -6.1815118  56.5383745 110.6328906   2.8512481
```

```
## [25] 51.9047605 107.5402248 150.0935267 122.4582218 220.5588120 99.8270958
## [31] 145.8353274 124.5171944 103.1308910 84.3583945 188.0115992 94.6250314
## [37] 42.0192975 187.5887058 110.4354948 98.0119791 145.6760005 155.6999534
## [43] 127.6244596 165.6863411 104.0210200 144.2964108 156.8589660 221.6772333
## [49] 93.7600604 139.9929336 123.6947450 115.6879548 155.2683960 120.1375953
## [55] 146.8614772 105.6504217 114.3879853 231.2823770 159.8089436 97.9449816
## [61] 157.5543288 138.0280587 146.8099094 185.1129521 158.4948068 223.2862171
## [67] 187.8701388 102.9404532 157.0489013 184.7600278 224.1715434 184.6599162
## [73] 219.8098884 165.5609809 217.9304388 240.0289793 268.6280880 211.1273013
## [79] 196.3668313 170.7351600 254.6413530 133.8246129 220.8719737 164.3062975
## [85] 234.5416212 280.9991849 260.4270097 240.4807178 166.5011086 189.6624045
## [91] 287.6324723 246.0772397 215.2327177 189.9079513 299.9936194 201.2127323
## [97] 168.2906793 222.0210683 211.8543632 188.1967652 240.3246918
```

```
hist(y1)
```



```
# Crie um plot do relacionamento de x e y1
plot(x, y1, pch = 19, xlab = 'X', ylab = 'Y')

# Crie um modelo de regressão para as duas variáveis x e y1
modelo <- lm(y1 ~ x)
modelo
```

```
##
## Call:
## lm(formula = y1 ~ x)
##
```

```
## Coefficients:
## (Intercept)      x
##      45.781      1.954

class(modelo)

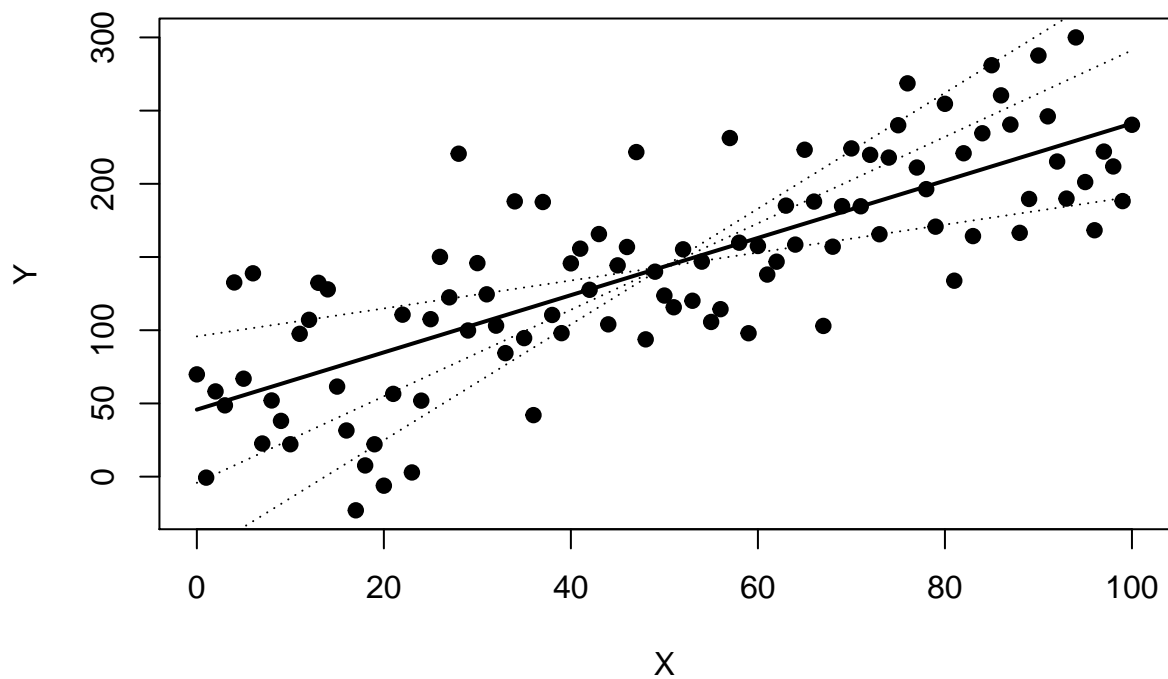
## [1] "lm"

# Capture os coeficientes
a <- modelo$coefficients[1]
b <- modelo$coefficients[2]

# Fórmula de Regressão
y2 <- a + b*x

# Visualize a linha de regressão
lines(x, y2, lwd = 2)

# Simulando outras possíveis linhas de regressão
y3 <- (y2[51]-50*(b-1))+(b-1)*x
y4 <- (y2[51]-50*(b+1))+(b+1)*x
y5 <- (y2[51]-50*(b+2))+(b+2)*x
lines(x,y3,lty=3)
lines(x,y4,lty=3)
lines(x,y5,lty=3)
```



```
## Pesquisa sobre idade e tempo de reação
```

```

# Criando os dados
Idade <- c(9,13,14,21,15,18,20,8,14,23,16,21,10,12,20,
          9,13,5,15,21)

Tempo <- c(17.87,13.75,12.72,6.98,11.01,10.48,10.19,19.11,
           12.72,0.45,10.67,1.59,14.91,14.14,9.40,16.23,
           12.74,20.64,12.34,6.44)

# Crie um Gráfico de Dispersão (ScatterPlot)
plot(Idade, Tempo,
     xlab = 'Idade',
     ylab = 'Tempo de Reação')

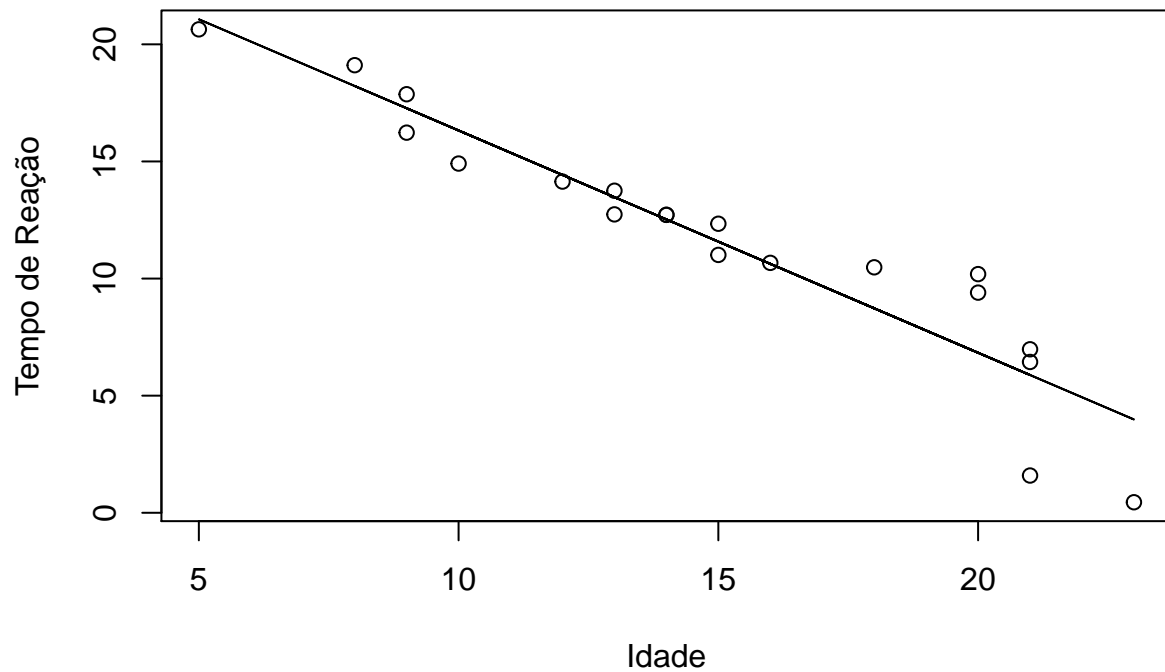
# Crie um modelo de regressão
modelo <- lm(Tempo ~ Idade)
modelo

##
## Call:
## lm(formula = Tempo ~ Idade)
##
## Coefficients:
## (Intercept)      Idade
##    25.8134    -0.9491

# Calcule a reta de regressão
y <- a + b*x
reta <- 25.8134 - 0.9491 * Idade

# Crie o gráfico da reta
lines(Idade,reta)

```



```
# Relação entre altura e peso
```

```
# Criando os dados
```

```
alturas = c(176, 154, 138, 196, 132, 176, 181, 169, 150, 175)
```

```
pesos = c(82, 49, 53, 112, 47, 69, 77, 71, 62, 78)
```

```
plot(alturas, pesos, pch = 16, cex = 1.3, col = "blue",  
     main = "Altura x Peso",  
     ylab = "Peso Corporal (kg)",  
     xlab = "Altura (cm)")
```

```
# Crie o modelo de regressão
```

```
modelo <- lm(pesos ~ alturas)
```

```
# Visualizando o modelo
```

```
modelo
```

```
##
```

```
## Call:
```

```
## lm(formula = pesos ~ alturas)
```

```
##
```

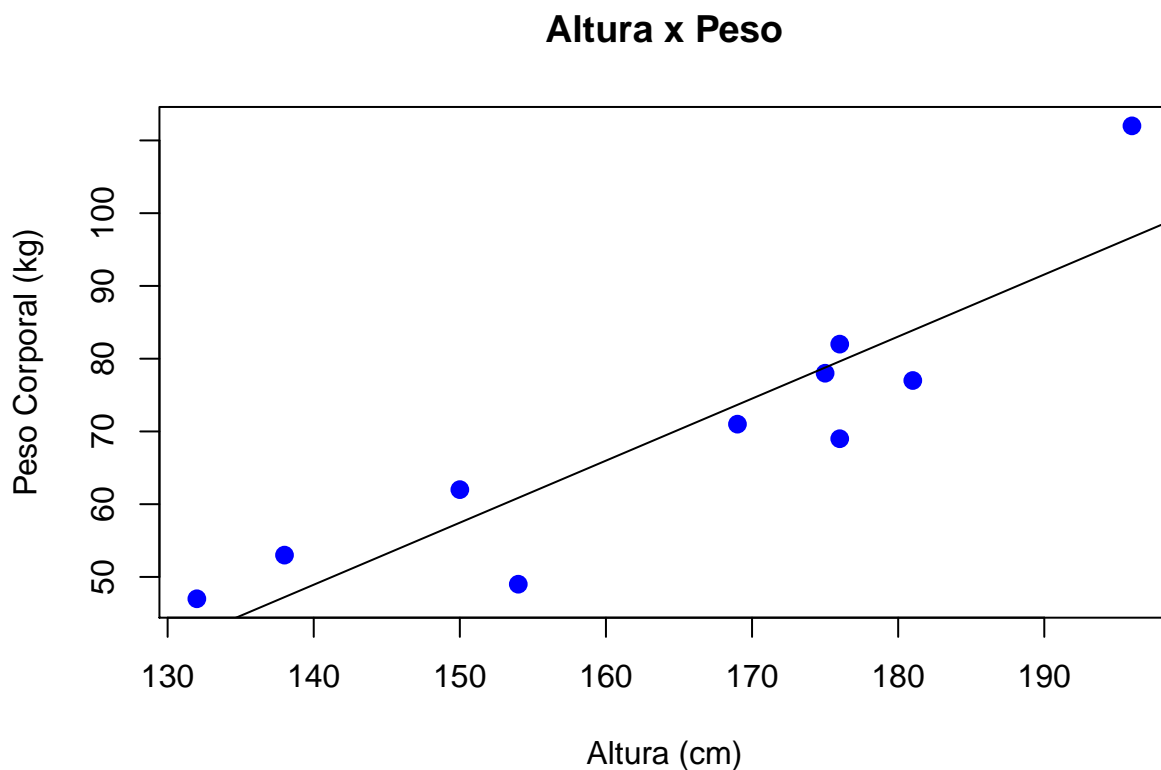
```
## Coefficients:
```

```
## (Intercept)      alturas
```

```
##      -70.4627      0.8528
```

```
summary(modelo)
```

```
##
## Call:
## lm(formula = pesos ~ alturas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.8746  -5.8428   0.7893   4.8001  15.3061
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -70.4627    24.0148  -2.934 0.018878 *
## alturas      0.8528     0.1448   5.889 0.000366 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.854 on 8 degrees of freedom
## Multiple R-squared:  0.8126, Adjusted R-squared:  0.7891
## F-statistic: 34.68 on 1 and 8 DF,  p-value: 0.0003662
# Gere a linha de regressão
abline(-70.4627, 0.8528)
```



```
# Faça as previsões de pesos com base na nova lista de alturas
alturas2 = data.frame(c(179, 152, 134, 197, 131, 178, 185, 162, 155, 172))
previsao <- predict(modelo, alturas2)
```

```
previsao
```

```
##          1          2          3          4          5          6          7          8
## 79.63709 60.87462 47.22918 96.69388 42.11214 79.63709 83.90129 73.66721
##          9         10
## 57.46326 78.78425
```

```
# Plot
```

```
plot(alturas, pesos, pch = 16, cex = 1.3,
     col = "blue",
     main = "Altura x Peso",
     ylab = "Peso (kg)",
     xlab = "Altura (cm)")
```

```
# Construindo a linha de regressão
```

```
abline(lm(pesos ~ alturas))
```

```
# Obtendo o tamanho de uma das amostras de dados
```

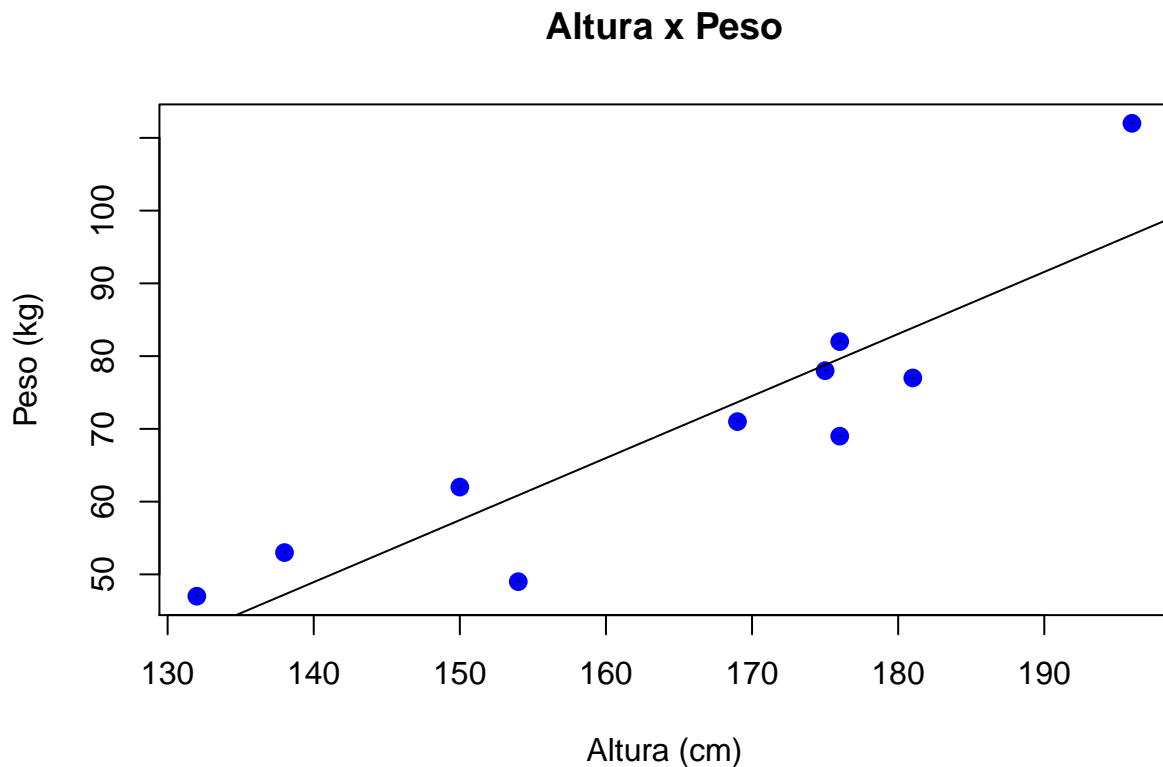
```
num <- length(alturas)
```

```
num
```

```
## [1] 10
```

```
# Gerando um gráfico com os valores residuais
```

```
for (k in 1: num)
  lines(c(alturas[k], alturas[k]),
        c(pesos[k], pesos[k]))
```



```
# Gerando gráficos com a distribuição dos resíduos
par(mfrow = c(2,2))
plot(modelo)
```

