Lista 02 - Fundamentos Estatísticos para Ciência dos Dados

1-

```
media = function (x){
  if(any(x < 0)){
    return("Existem componentes negativas no vetor")
  }
  return(prod(x)^(1/length(x)))
}
```

2-

a)

```
media = function (x){
  if(any(x < 0)){
    return("Existem componentes negativas no vetor")
  }
  return(prod(x)^(1/length(x)))
}

set.seed(123)
data <- data.frame(matrix(rnorm(10000, mean=3), ncol=25, dimnames=list(NULL, paste("X",
1:25, sep="."))))

for (coli in 1:(dim(data)[2])){
  mediaColi = media(data[, coli])
  print( paste("Coluna", coli, ":", mediaColi))
}
```

Resultado:

```
[1] "Coluna 1 : 2.84566675391185"
[1] "Coluna 2 : 2.79089727564161"
[1] "Coluna 3 : 2.85129600596228"
[1] "Coluna 4 : 2.83280065008546"
[1] "Coluna 5 : Existem componentes negativas no vetor"
[1] "Coluna 6 : 2.74610302260878"
[1] "Coluna 7 : 2.78441864034139"
[1] "Coluna 8 : 2.77856728388433"
[1] "Coluna 9 : Existem componentes negativas no vetor"
[1] "Coluna 10 : 2.79284012672326"
[1] "Coluna 11 : 2.74597719635928"
[1] "Coluna 12 : Existem componentes negativas no vetor"
[1] "Coluna 13 : Existem componentes negativas no vetor"
[1] "Coluna 14 : Existem componentes negativas no vetor"
[1] "Coluna 15 : Existem componentes negativas no vetor"
[1] "Coluna 16 : Existem componentes negativas no vetor"
[1] "Coluna 17 : 2.80916442992269"
[1] "Coluna 18 : Existem componentes negativas no vetor"
[1] "Coluna 19 : Existem componentes negativas no vetor"
[1] "Coluna 20 : Existem componentes negativas no vetor"
[1] "Coluna 21 : 2.8616279040828"
[1] "Coluna 22 : Existem componentes negativas no vetor"
[1] "Coluna 23 : 2.74122971837727"
[1] "Coluna 24 : 2.76990055065242"
[1] "Coluna 25 : Existem componentes negativas no vetor"
```

b)

```
set.seed(123)
data <- data.frame(matrix(rnorm(10000, mean=3), ncol=25, dimnames=list(NULL, paste("X",
1:25, sep="."))))

for (coli in 1:(dim(data)[2])){
  mediaColi = sd(data[, coli])
  print( paste("Coluna", coli, ":", mediaColi))
}
```

Resultado:

```
[1] "Coluna 1 : 0.969015457538878"
[1] "Coluna 2 : 0.994285794505739"
[1] "Coluna 3 : 1.02652443405254"
[1] "Coluna 4 : 0.953598474217803"
[1] "Coluna 5 : 1.05951878110894"
[1] "Coluna 6 : 0.924258964941437"
[1] "Coluna 7 : 0.987593057079863"
[1] "Coluna 8 : 1.02302630511393"
[1] "Coluna 9 : 1.012879775481"
[1] "Coluna 10 : 0.976722604513245"
[1] "Coluna 11 : 1.00152902840031"
[1] "Coluna 12 : 0.987552919778121"
[1] "Coluna 13 : 0.95515358975177"
[1] "Coluna 14 : 1.01615998899861"
[1] "Coluna 15 : 1.02154419914103"
[1] "Coluna 16 : 1.05155483290154"
[1] "Coluna 17 : 0.954212689556275"
[1] "Coluna 18 : 0.952107481473251"
[1] "Coluna 19 : 0.999002882644834"
[1] "Coluna 20 : 1.02700381562964"
[1] "Coluna 21 : 1.0145608351813"
[1] "Coluna 22 : 1.04158507867094"
[1] "Coluna 23 : 1.04952980187319"
[1] "Coluna 24 : 0.981679298932684"
[1] "Coluna 25 : 0.982459914304845"
```

c)

```
set.seed(123)
data <- data.frame(matrix(rnorm(10000, mean=3), ncol=25, dimnames=list(NULL, paste("X",
1:25, sep="."))))

for (line_i in 1:(dim(data)[1])){
  print( paste("Linha", line_i, ":", sum(data[line_i, ])))
}
```

Resultado:

```
[1] "Linha 1 : 71.7336405015799"
[1] "Linha 2 : 70.6036050491132"
[1] "Linha 3 : 73.3534154136219"
[1] "Linha 4 : 73.9685209348089"
[1] "Linha 5 : 71.6746029306237"
```

… O resultado é exibido para as demais linhas

d)

```
set.seed(123)
data <- data.frame(matrix(rnorm(10000, mean=3), ncol=25, dimnames=list(NULL, paste("X",
1:25, sep="."))))
selected <- data[which(data$X.1 > 3 & data$X.20<3), ]
print(nrow(selected))
```

Foram selecionadas 102 linhas

e)

```
set.seed(123)
data <- data.frame(matrix(rnorm(10000, mean=3), ncol=25, dimnames=list(NULL, paste("X",
1:25, sep="."))))

x = c()
for(i in 1:(dim(data)[2])){
  x[i] = paste("Var", i, sep="")
}
colnames(data) = x
print(colnames(data))
```

Resultado:

```
> print(colnames(data))
 [1] "Var1"  "Var2"  "Var3"  "Var4"  "Var5"  "Var6"  "Var7"  "Var8"  "Var9"
[10] "Var10" "Var11" "Var12" "Var13" "Var14" "Var15" "Var16" "Var17" "Var18"
[19] "Var19" "Var20" "Var21" "Var22" "Var23" "Var24" "Var25"
```

3-

a)

```
?iris
```

Resultado:

iris {datasets} R Documentation

Edgar Anderson's Iris Data

Description

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.

Usage

```
iris
iris3
```

Format

`iris` is a data frame with 150 cases (rows) and 5 variables (columns) named `Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`, and `Species`.

`iris3` gives the same data arranged as a 3-dimensional array of size 50 by 4 by 3, as represented by S-PLUS. The first dimension gives the case number within the species subsample, the second the measurements with names `Sepal L.`, `Sepal W.`, `Petal L.`, and `Petal W.`, and the third the species.

Source

Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, **7**, Part II, 179–188.

The data were collected by Anderson, Edgar (1935). The irises of the Gaspe Peninsula, *Bulletin of the American Iris Society*, **59**, 2–5.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole. (has `iris3` as `iris`.)

See Also

`matplot` some examples of which use `iris`.

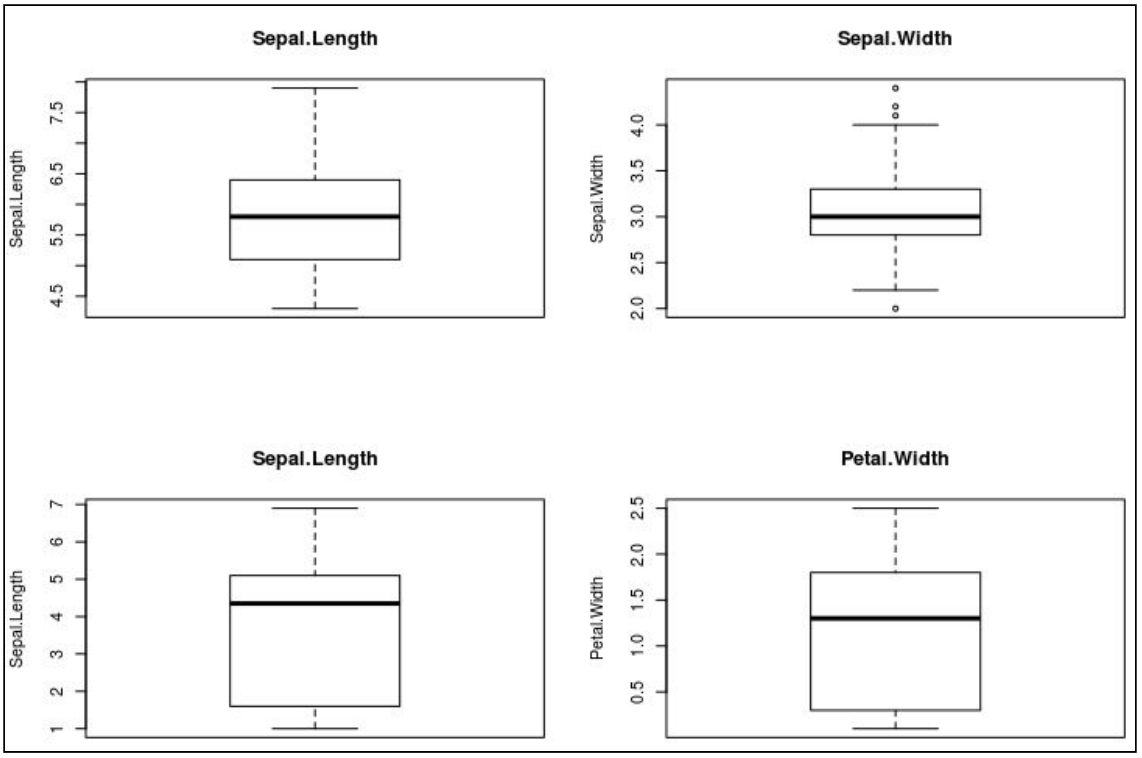
Examples

```
dni3 <- dimnames(iris3)
ii <- data.frame(matrix(aperm(iris3, c(1,3,2)), ncol = 4,
                        dimnames = list(NULL, sub(" L.",".Length",
                                                  sub(" W.",".Width", dni3[[2]])))),
    Species = gl(3, 50, labels = sub("S", "s", sub("V", "v", dni3[[3]])))
all.equal(ii, iris) # TRUE
```

b)

```
flowers = read.csv("iris.csv", header = T)
par(mfrow=c(3,2))
boxplot(flowers$Sepal.Length, main="Sepal.Length", ylab="Sepal.Length")
boxplot(flowers$Sepal.Width, main = "Sepal.Width", ylab="Sepal.Width")
boxplot(flowers$Petal.Length, main="Sepal.Length", ylab="Sepal.Length")
boxplot(flowers$Petal.Width, main="Petal.Width", ylab="Petal.Width")
```

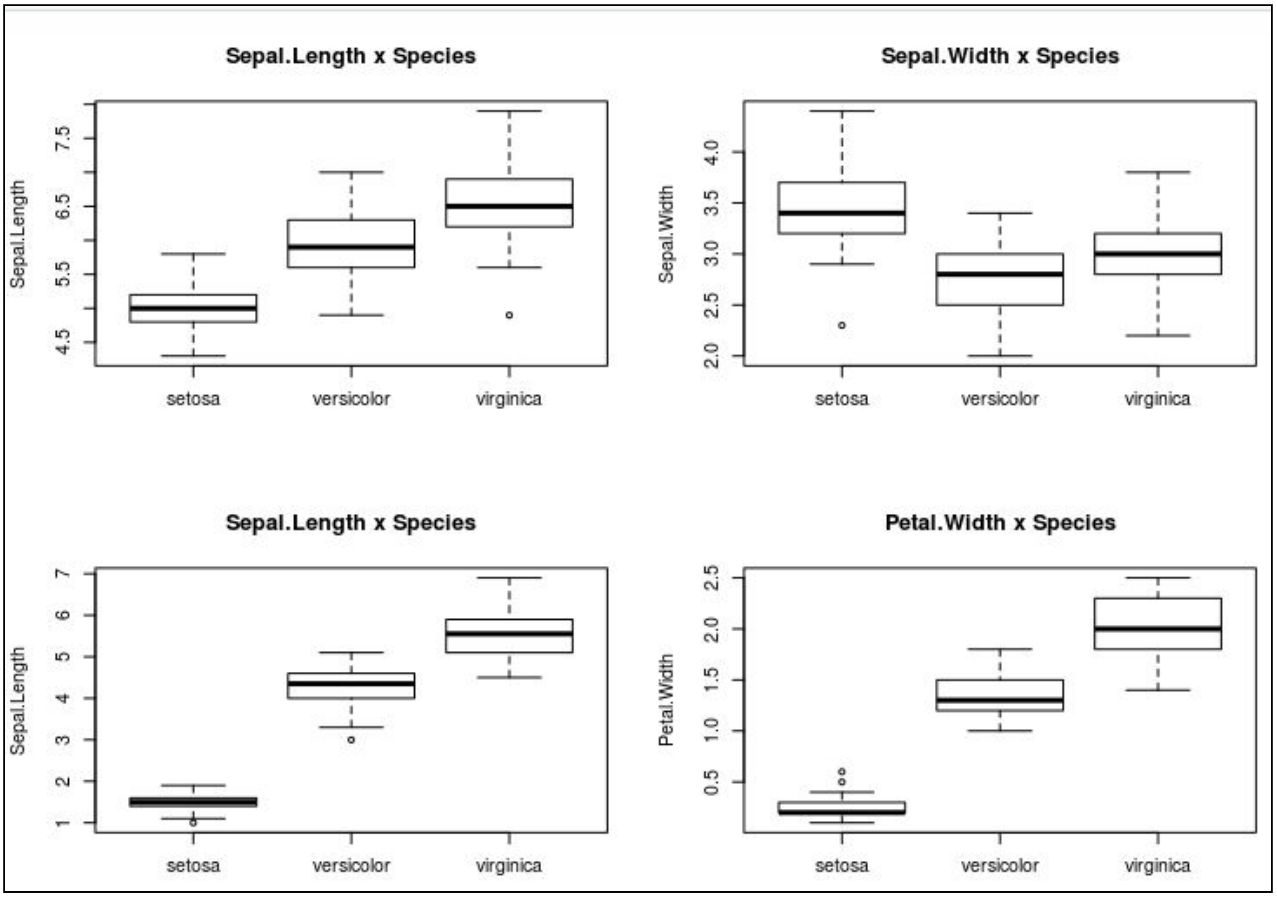
Resultado:



c)

```
flowers = read.csv("iris.csv", header = T)
par(mfrow=c(3,2))
boxplot(flowers$Sepal.Length ~ flowers$Species, main="Sepal.Length x Species",
ylab="Sepal.Length")
boxplot(flowers$Sepal.Width ~ flowers$Species, main = "Sepal.Width x Species",
ylab="Sepal.Width")
boxplot(flowers$Petal.Length ~ flowers$Species, main="Sepal.Length x Species",
ylab="Sepal.Length")
boxplot(flowers$Petal.Width ~ flowers$Species, main="Petal.Width x Species",
ylab="Petal.Width")
```

Resultado:


Sepal.Length x Species
Sepal.Length
4.5
5.5
6.5
7.5
setosa
versicolor
virginica
Sepal.Width x Species
Sepal.Width
2.0
2.5
3.0
3.5
4.0
setosa
versicolor
virginica
Sepal.Length x Species
Sepal.Length
1
2
3
4
5
6
7
setosa
versicolor
virginica
Petal.Width x Species
Petal.Width
0.5
1.0
1.5
2.0
2.5
setosa
versicolor
virginica

4-

a)

```
?attitude
```

Resultado:

The Chatterjee–Price Attitude Data

Description

From a survey of the clerical employees of a large financial organization, the data are aggregated from the questionnaires of the approximately 35 employees for each of 30 (randomly selected) departments. The numbers give the percent proportion of favourable responses to seven questions in each department.

Usage

```
attitude
```

Format

A data frame with 30 observations on 7 variables. The first column are the short names from the reference, the second one the variable names in the data frame:

Y	rating	numeric Overall rating
X[1]	complaints	numeric Handling of employee complaints
X[2]	privileges	numeric Does not allow special privileges
X[3]	learning	numeric Opportunity to learn
X[4]	raises	numeric Raises based on performance
X[5]	critical	numeric Too critical
X[6]	advancel	numeric Advancement

Source

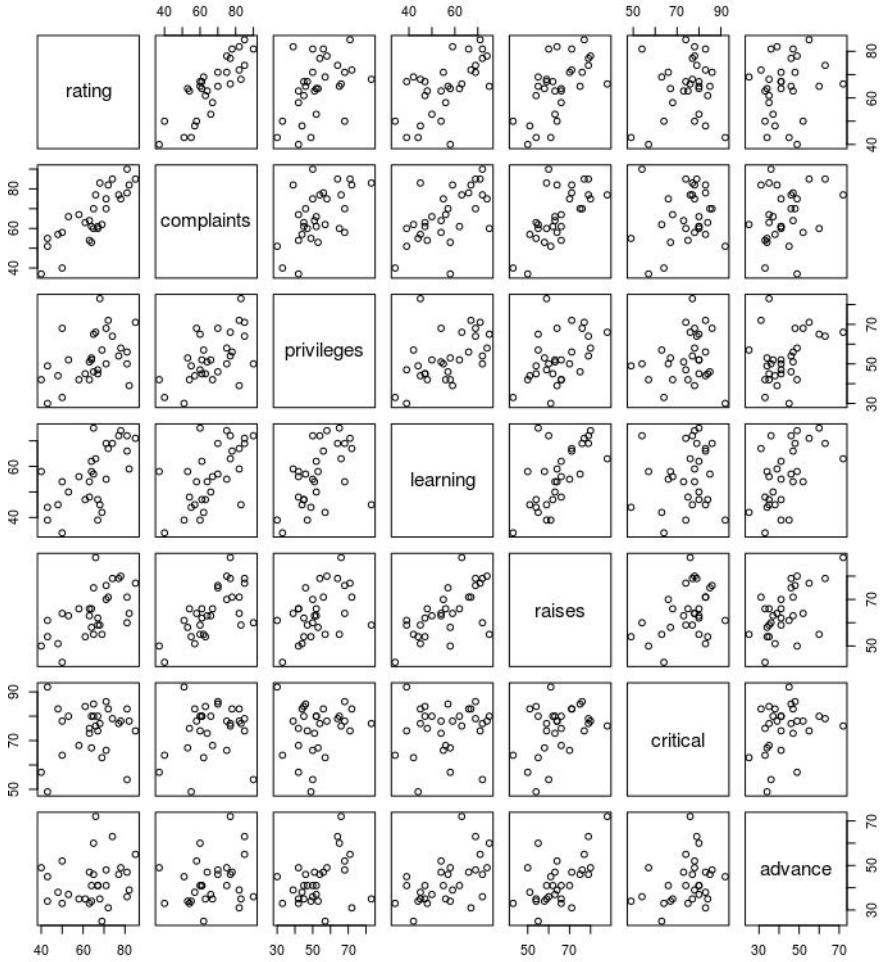
Chatterjee, S. and Price, B. (1977) *Regression Analysis by Example*. New York: Wiley. (Section 3.7, p.68ff of 2nd ed.(1991).)

Examples

```
require(stats); require(graphics)
pairs(attitude, main = "attitude data")
summary(attitude)
summary(fm1 <- lm(rating ~ ., data = attitude))
opar <- par(mfrow = c(2, 2), oma = c(0, 0, 1.1, 0),
            mar = c(4.1, 4.1, 2.1, 1.1))
plot(fm1)
summary(fm2 <- lm(rating ~ complaints, data = attitude))
plot(fm2)
par(opar)
```

b)

Resultado:



c)

```
att = read.csv("attitude.csv", header = T)

for (coli in 1:(dim(att)[2])){
  mediaColi = mean(att[, coli])
  print( paste("Coluna", coli, ":", mediaColi))
}
```

Resultado:

```
[1] "Coluna 1 : 64.6333333333333"
[1] "Coluna 2 : 66.6"
[1] "Coluna 3 : 53.1333333333333"
[1] "Coluna 4 : 56.3666666666667"
[1] "Coluna 5 : 64.6333333333333"
[1] "Coluna 6 : 74.7666666666667"
[1] "Coluna 7 : 42.9333333333333"
```

d)

```
att = read.csv("attitude.csv", header = T)
att$complaints <- cut(att$complaints, breaks = c(0,60,80,100), labels = c("bad", "okay",
"good") )
print(att$complaints)
```

Resultado:

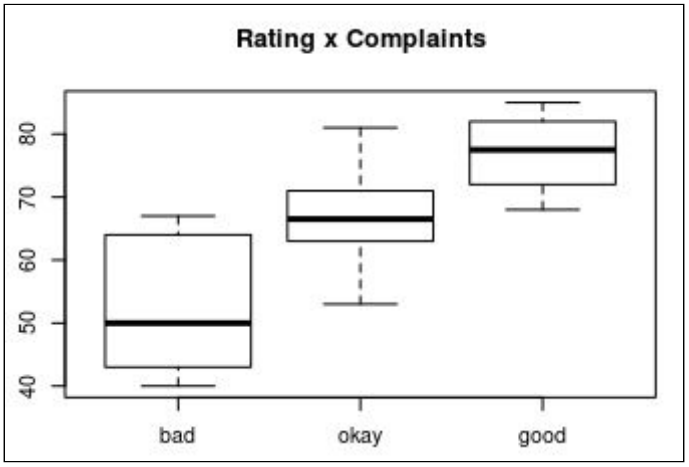
```
 [1] bad  okay okay okay okay bad  okay okay good okay bad  bad  okay good okay good good bad  okay bad  bad  okay okay bad  bad  okay okay bad
[29] good good
Levels: bad okay good
```

e)

```
att = read.csv("attitude.csv", header = T)

att$complaints <- cut(att$complaints, breaks = c(0,60,80,100), labels = c("bad", "okay",
"good") )
boxplot(att$rating ~ att$complaints, main = "Rating x Complaints")
```

Resultado:

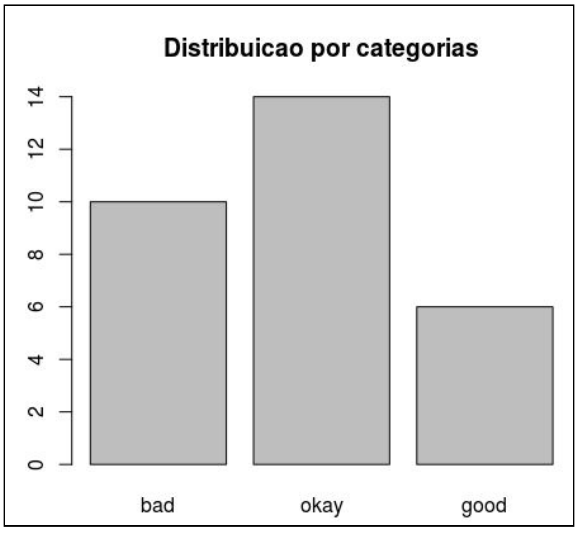


f)

```
att = read.csv("attitude.csv", header = T)

att$complaints <- cut(att$complaints, breaks = c(0,60,80,100), labels = c("bad", "okay",
"good") )
barplot(table(att$complaints), main = "Distribuicao por categorias")
```

Resultado:



g)

```
att = read.csv("attitude.csv", header = T)
par(mfrow=c(1,2))
att$complaints <- cut(att$complaints, breaks = c(0,60,80,100), labels = c("Ruim",
"Normal", "Bom") )
boxplot(att$rating ~ att$complaints, main = "Rating x Complaints")
barplot(table(att$complaints), main = "Distribuicao por categorias")
```

Resultado: