

# Computational Stats

*Tiago dos Santos*

*2018-10-04*

## Contents

<b>1</b>	<b>Lesson 1</b>	<b>1</b>
1.1	Start by creating a vector . . . . .	1
1.2	Now a Matrix! . . . . .	2
1.3	DataFrames . . . . .	4
1.4	Reading a Tab Separated File . . . . .	4
1.5	Generating data . . . . .	5
1.6	Getting insights . . . . .	5
1.7	Lists . . . . .	11
1.8	Functions . . . . .	11
<b>2</b>	<b>Lessons 2</b>	<b>12</b>
2.1	Random Variables and Vectors . . . . .	12

To use the lab computers, the access credentials are:

usr: enc pwd: Ecom\*2018

## 1 Lesson 1

```
x <- 3+5  
  
ls()  
  
## [1] "datasetsDir" "fig_basePath" "x"
```

### 1.1 Start by creating a vector

```
y <- c(2,5,9,8)  
  
y[1:3]  
  
## [1] 2 5 9  
  
y[c(1,3)]  
  
## [1] 2 9
```

#### 1.1.0.1 Get the elements 1,2,3 from the vector

```
y[1:3]  
  
## [1] 2 5 9
```

### 1.1.0.2 Get the elements 1,3 from the vector

```
y[c(1,3)]
```

```
## [1] 2 9
```

### 1.1.0.3 Get an array from 0 to 1, with a 0.001 step

```
y <- 1:1000/1000  
y <- seq(0,1,0.001)
```

### 1.1.0.4 Which values are lower than 0.008?

```
isValueLowerThan <- y < 0.008  
y[isValueLowerThan]
```

```
## [1] 0.000 0.001 0.002 0.003 0.004 0.005 0.006 0.007
```

```
idxs <- which(y<0.08)  
y[idxs]
```

```
## [1] 0.000 0.001 0.002 0.003 0.004 0.005 0.006 0.007 0.008 0.009 0.010  
## [12] 0.011 0.012 0.013 0.014 0.015 0.016 0.017 0.018 0.019 0.020 0.021  
## [23] 0.022 0.023 0.024 0.025 0.026 0.027 0.028 0.029 0.030 0.031 0.032  
## [34] 0.033 0.034 0.035 0.036 0.037 0.038 0.039 0.040 0.041 0.042 0.043  
## [45] 0.044 0.045 0.046 0.047 0.048 0.049 0.050 0.051 0.052 0.053 0.054  
## [56] 0.055 0.056 0.057 0.058 0.059 0.060 0.061 0.062 0.063 0.064 0.065  
## [67] 0.066 0.067 0.068 0.069 0.070 0.071 0.072 0.073 0.074 0.075 0.076  
## [78] 0.077 0.078 0.079
```

### 1.1.0.5 Creating objects by repetition

```
colors <- c("amarelo","verde","vermelho","azul")
```

```
rep(colors, 5)
```

```
## [1] "amarelo" "verde" "vermelho" "azul" "amarelo" "verde"  
## [7] "vermelho" "azul" "amarelo" "verde" "vermelho" "azul"  
## [13] "amarelo" "verde" "vermelho" "azul" "amarelo" "verde"  
## [19] "vermelho" "azul"
```

```
print("===")
```

```
## [1] "==="
```

```
rep(10,5)
```

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

## 1.2 Now a Matrix!

```
M <- matrix(1:9, ncol=3)  
M
```

```
##      [,1] [,2] [,3]  
## [1,]    1    4    7
```

```
## [2,] 2 5 8
## [3,] 3 6 9
```

Transposing the Matrix

```
t(M)
```

```
##      [,1] [,2] [,3]
## [1,] 1 2 3
## [2,] 4 5 6
## [3,] 7 8 9
```

Accessing the Matrix

```
M[1,2]
```

```
## [1] 4
```

```
M[1,]
```

```
## [1] 1 4 7
```

```
M[,2]
```

```
## [1] 4 5 6
```

Matrix Operation

```
M2 <- t(M)
```

```
M+M2 # valuewise add
```

```
##      [,1] [,2] [,3]
## [1,] 2 6 10
## [2,] 6 10 14
## [3,] 10 14 18
```

```
M*M2 # valuewise multiplication
```

```
##      [,1] [,2] [,3]
## [1,] 1 8 21
## [2,] 8 25 48
## [3,] 21 48 81
```

```
M%*%M2 # Matricial Multiplication
```

```
##      [,1] [,2] [,3]
## [1,] 66 78 90
## [2,] 78 93 108
## [3,] 90 108 126
```

### 1.2.0.1 Joining Matrixes

Matrix Operation

```
cbind(M,M2)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 1 4 7 1 2 3
## [2,] 2 5 8 4 5 6
## [3,] 3 6 9 7 8 9
```

```
rbind(M,M2)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
## [4,]    1    2    3
## [5,]    4    5    6
## [6,]    7    8    9
```

### 1.2.0.2 Inverting a matrix

```
#solve(M) # M must not be singular
```

## 1.3 DataFrames

```
y <- 1:10
y2 <- 11:20
y3 <- letters[1:10]

d1 <- data.frame(y,y2,y3)

d1
```

```
##      y y2 y3
## 1    1 11  a
## 2    2 12  b
## 3    3 13  c
## 4    4 14  d
## 5    5 15  e
## 6    6 16  f
## 7    7 17  g
## 8    8 18  h
## 9    9 19  i
## 10  10 20  j
```

## 1.4 Reading a Tab Separated File

```
emp <- read.table(file.path(datasetsDir,"empresas.txt"), header=F)
knitr::kable(head(emp))
```

V1	V2	V3	V4	V5
Soflor	2	5	10	3
Florinha	3	10	22	7
Flora	5	30	55	18
Floflo	2	5	12	4
Fazflor	3	15	28	8
Comercflor	2	10	18	5

```
dim(emp)
```

```
## [1] 40  5
```

```
names(emp) <- c("nome", "n.socios", "c.social", "vmm", "n.emp")
knitr::kable(head(emp))
```

nome	n.socios	c.social	vmm	n.emp
Soflor	2	5	10	3
Florinha	3	10	22	7
Flora	5	30	55	18
Floflo	2	5	12	4
Fazflor	3	15	28	8
Comercflor	2	10	18	5

```
emp$n.socios
```

```
## [1] 2 3 5 2 3 2 3 4 6 5 2 3 2 3 2 3 3 2 5 2 2 3 3 2 2 2 2 4 4 3 2 2 4 2 2
## [36] 2 3 3 3 2
```

```
emp[,2]
```

```
## [1] 2 3 5 2 3 2 3 4 6 5 2 3 2 3 2 3 3 2 5 2 2 3 3 2 2 2 2 4 4 3 2 2 4 2 2
## [36] 2 3 3 3 2
```

## 1.5 Generating data

```
set.seed(5)
```

```
emp$ant <- round(rnorm(dim(emp)[1], 10, 1))
```

## 1.6 Getting insights

```
summary(emp)
```

```
##          nome          n.socios          c.social          vmm
## Alecrim      : 1    Min.      :2.00    Min.      : 5.00    Min.      : 5.00
## Beijaflor    : 1    1st Qu.:2.00    1st Qu.: 5.00    1st Qu.: 11.00
## Caflor       : 1    Median :3.00    Median :10.00    Median : 19.00
## Comercflor   : 1    Mean     :2.85    Mean     :11.72    Mean      :24.48
## Cravinho     : 1    3rd Qu.:3.00    3rd Qu.:15.00    3rd Qu.: 31.00
## Cravo        : 1    Max.      :6.00    Max.      :50.00    Max.      :100.00
## (Other)      :34
##          n.emp          ant
## Min.      : 2.000    Min.      : 8
## 1st Qu.: 3.000    1st Qu.: 9
## Median : 5.500    Median :10
## Mean      : 6.225    Mean      :10
## 3rd Qu.: 9.000    3rd Qu.:11
## Max.      :18.000    Max.      :12
##
```

```
mean(emp$n.socios)
```

```
## [1] 2.85
```

```
sd(emp$n.socios)
```

```
## [1] 1.051251
```

```
tapply(emp$vmm, emp$n.emp, mean) # vmm mean by number of employees
```

```
##          2          3          4          5          6          7
##  8.714286 10.875000 12.666667 18.000000 22.000000 23.000000
##          8          9         10         11         14         15
## 28.000000 32.250000 45.000000 61.000000 45.000000 100.000000
##         16         18
## 55.000000 55.000000
```

```
tapply(emp$vmm, emp$n.emp, sd) # vmm sd by number of employees
```

```
##          2          3          4          5          6          7          8          9
## 2.627691 1.457738 1.154701 0.000000 1.414214 1.414214      NA 1.500000
##         10         11         14         15         16         18
##        NA 1.414214      NA      NA      NA      NA      NA
```

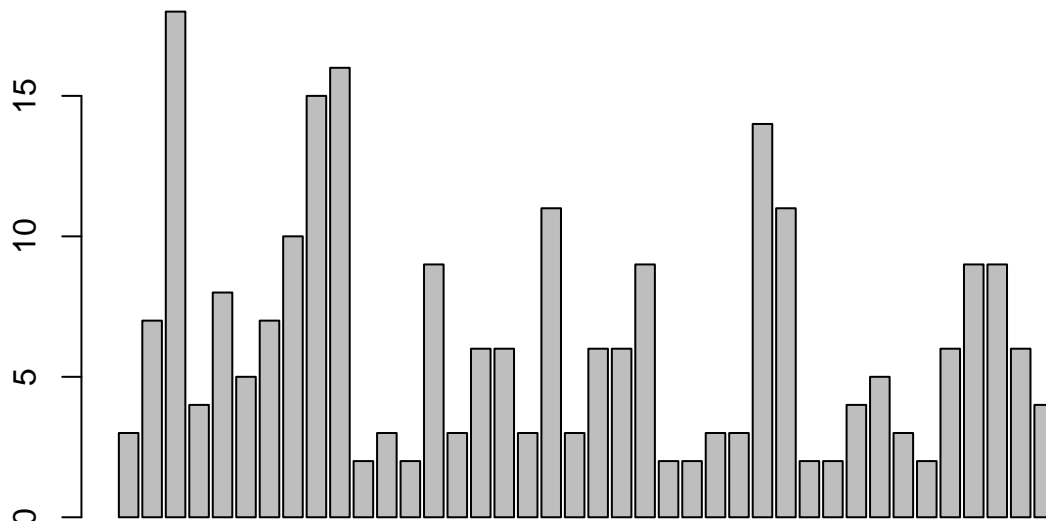
$$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$$

$$S^2 = \frac{1}{N} \sum_{i=1}^N (X_i - \bar{X})^2 \quad (1)$$

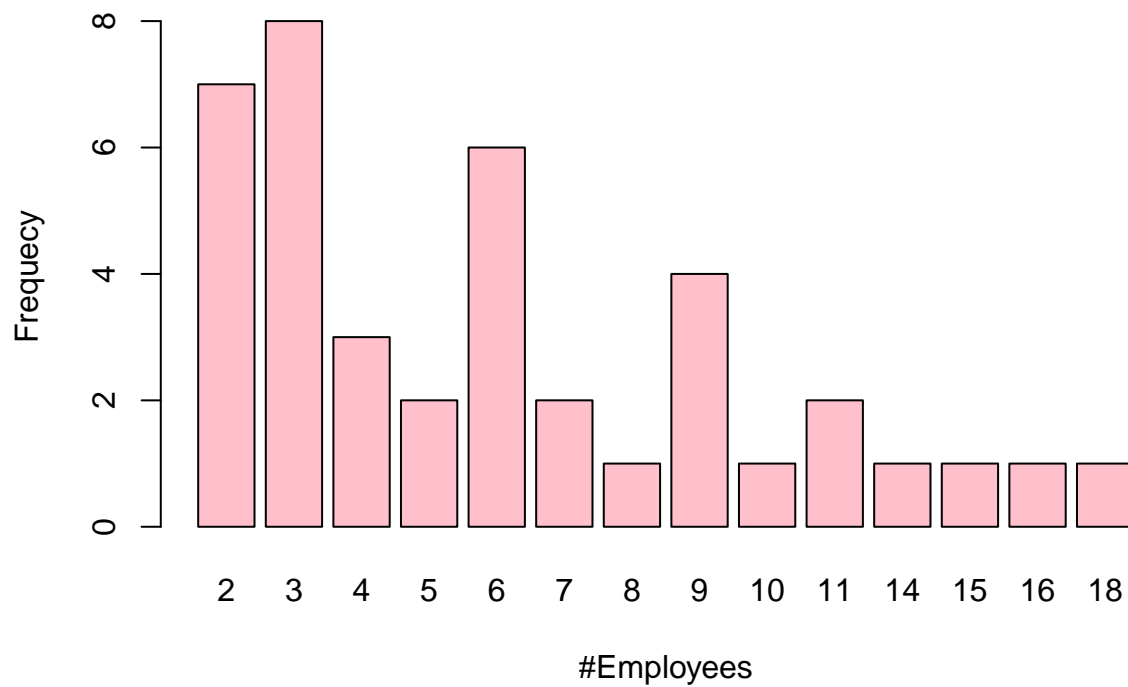
```
table(emp$n.emp) #first line are values, second line is frequency
```

```
##
##  2  3  4  5  6  7  8  9 10 11 14 15 16 18
##  7  8  3  2  6  2  1  4  1  2  1  1  1  1
```

```
barplot(emp$n.emp) # each company is a bin in x label, y is the number of employees
```

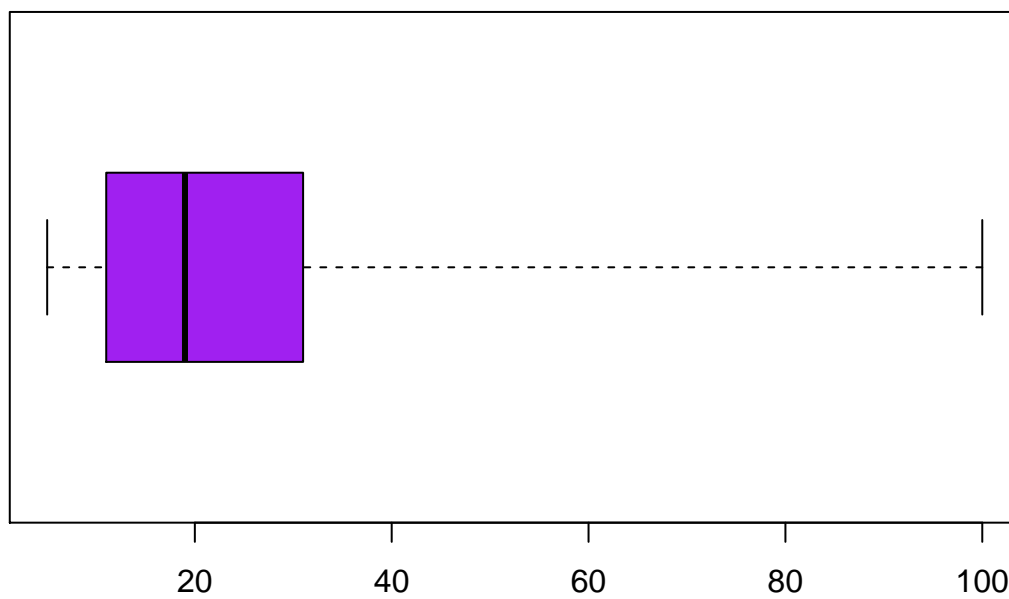


```
barplot(table(emp$n.emp), xlab="#Employees", ylab="Frequency", col="pink")
```

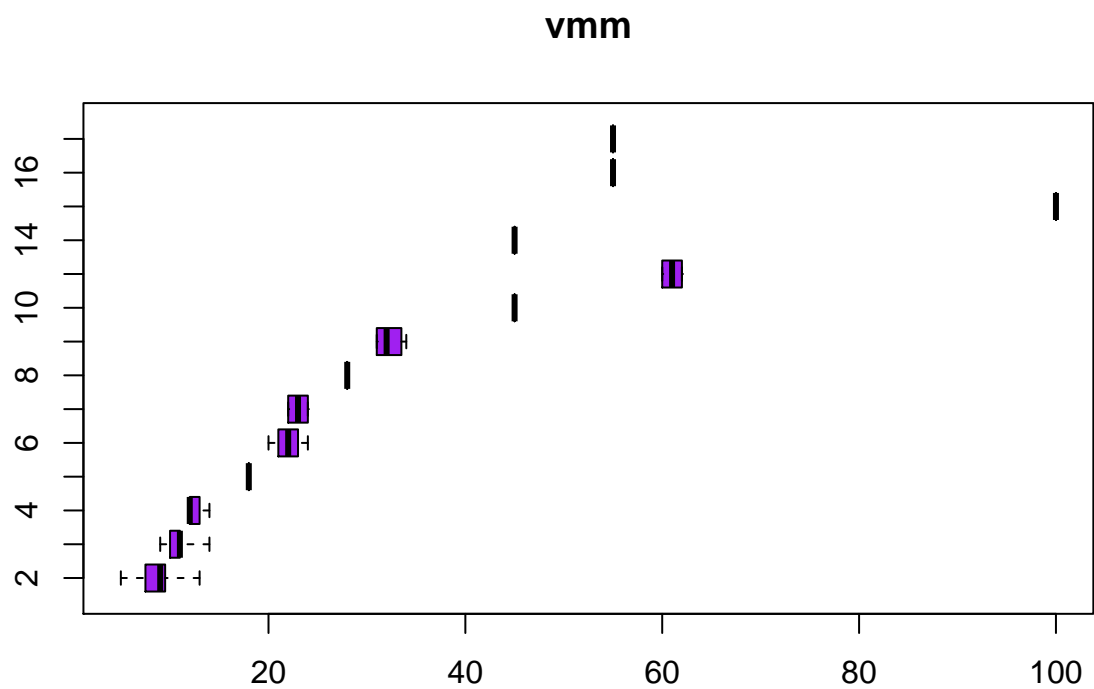


```
boxplot(emp$vmm,range=0,col="purple",horizontal=T,main="vmm")
```

**vmm**

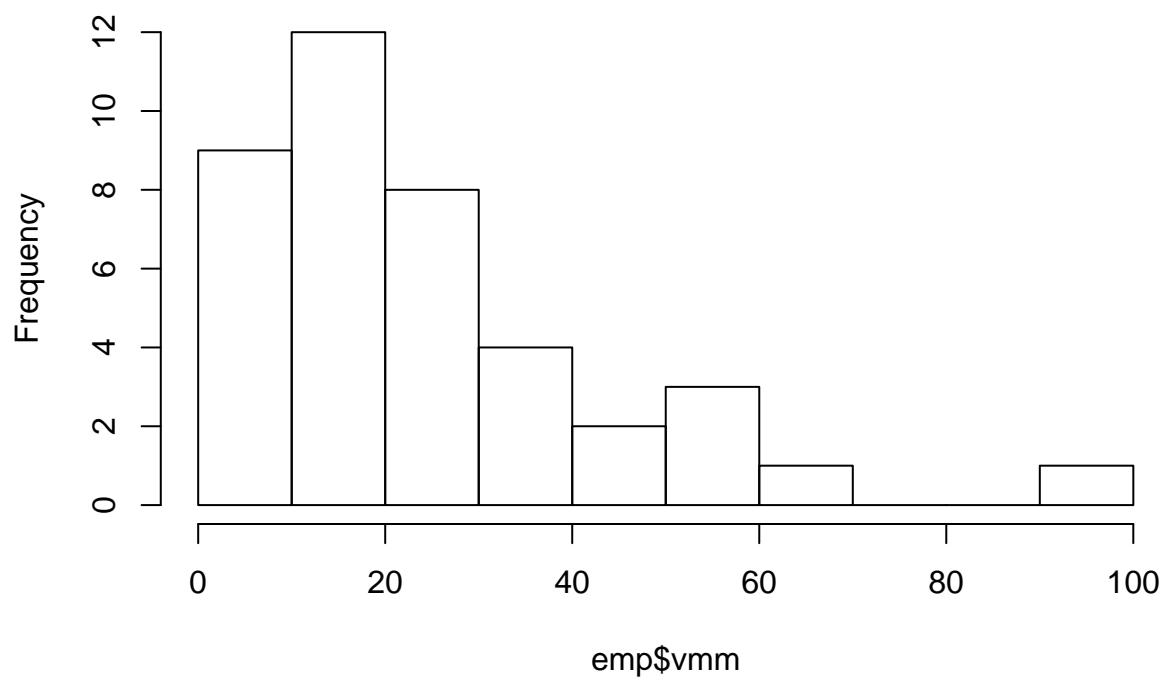


```
boxplot(emp$vmm ~ emp$n.emp,range=0,col="purple",horizontal=T,main="vmm")
```



```
hist(emp$vmm)
```

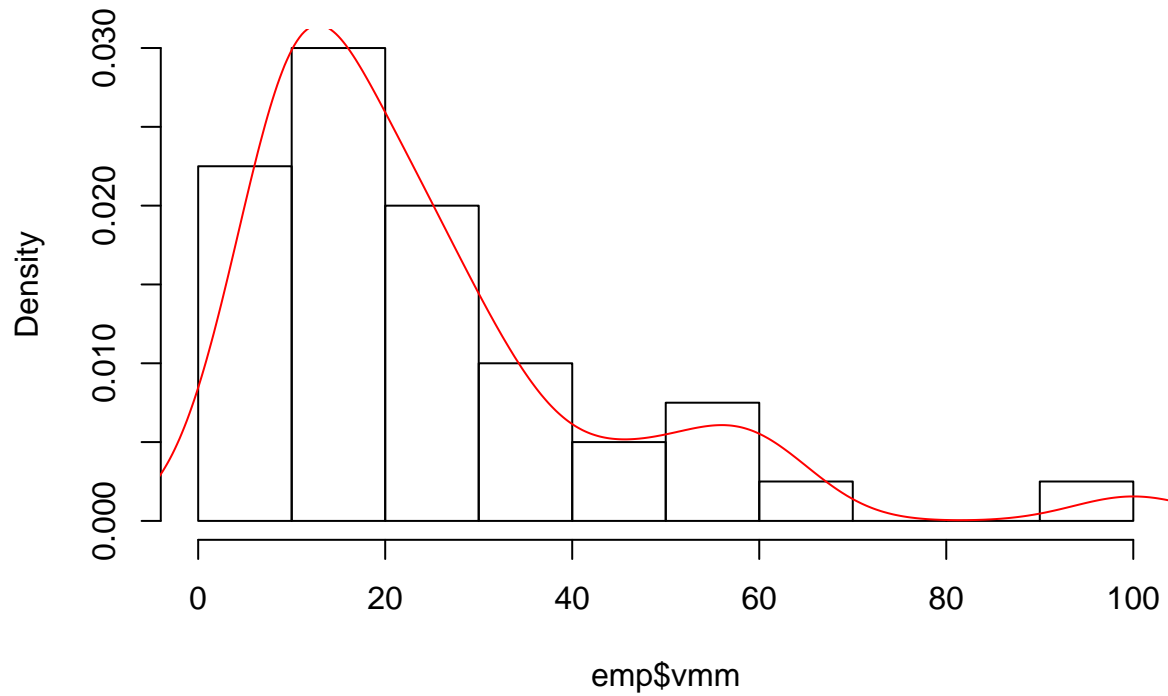
### Histogram of emp\$vmm



```
hist(emp$vmm, freq=F)
lines(density(emp$vmm), col=2)
```

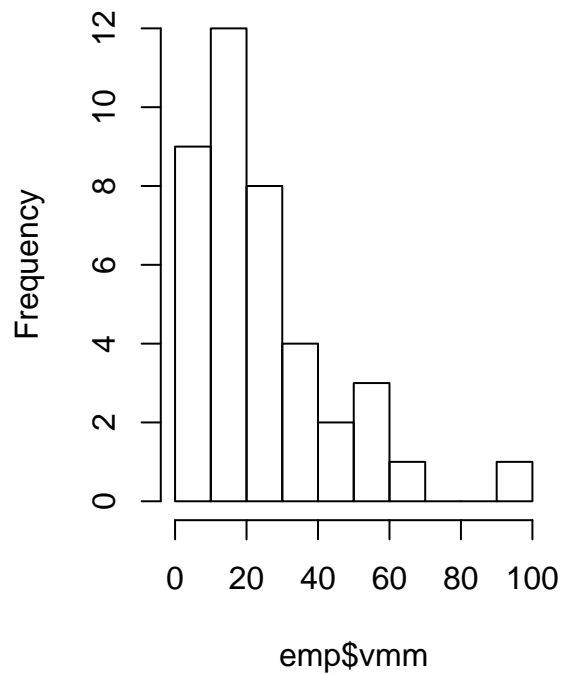


**Histogram of emp\$vmm**

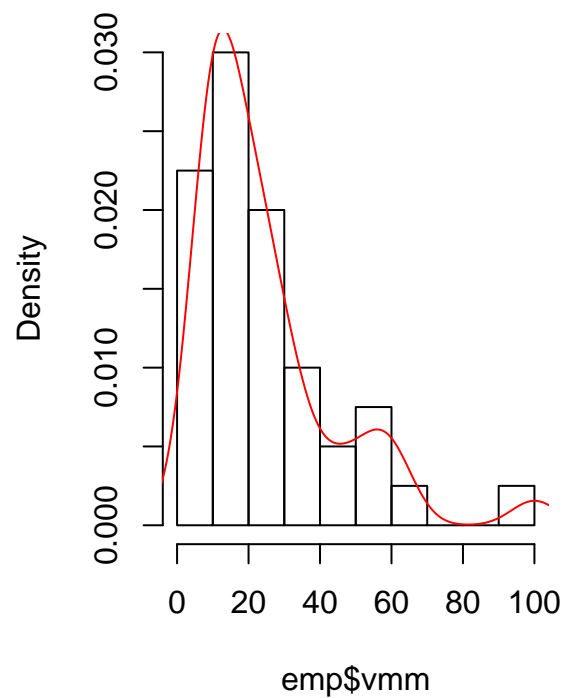


```
par(mfrow=c(1,2))  
hist(emp$vmm)  
  
hist(emp$vmm, freq=F)  
lines(density(emp$vmm), col=2)
```

**Histogram of emp\$vmm**

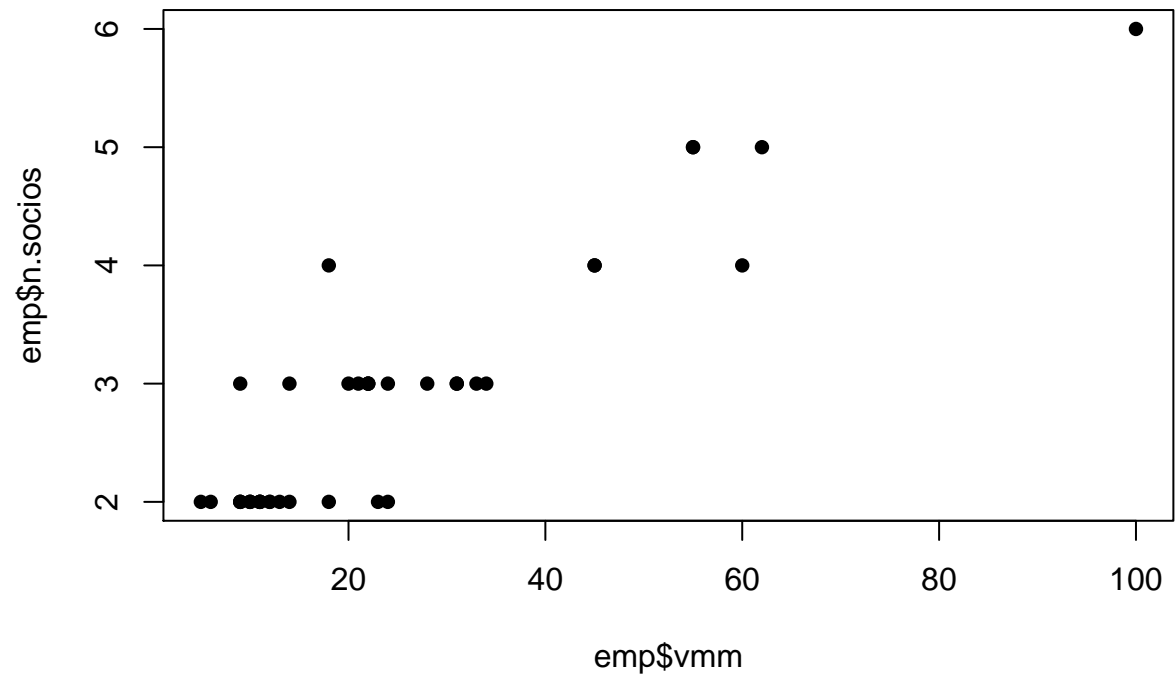


**Histogram of emp\$vmm**

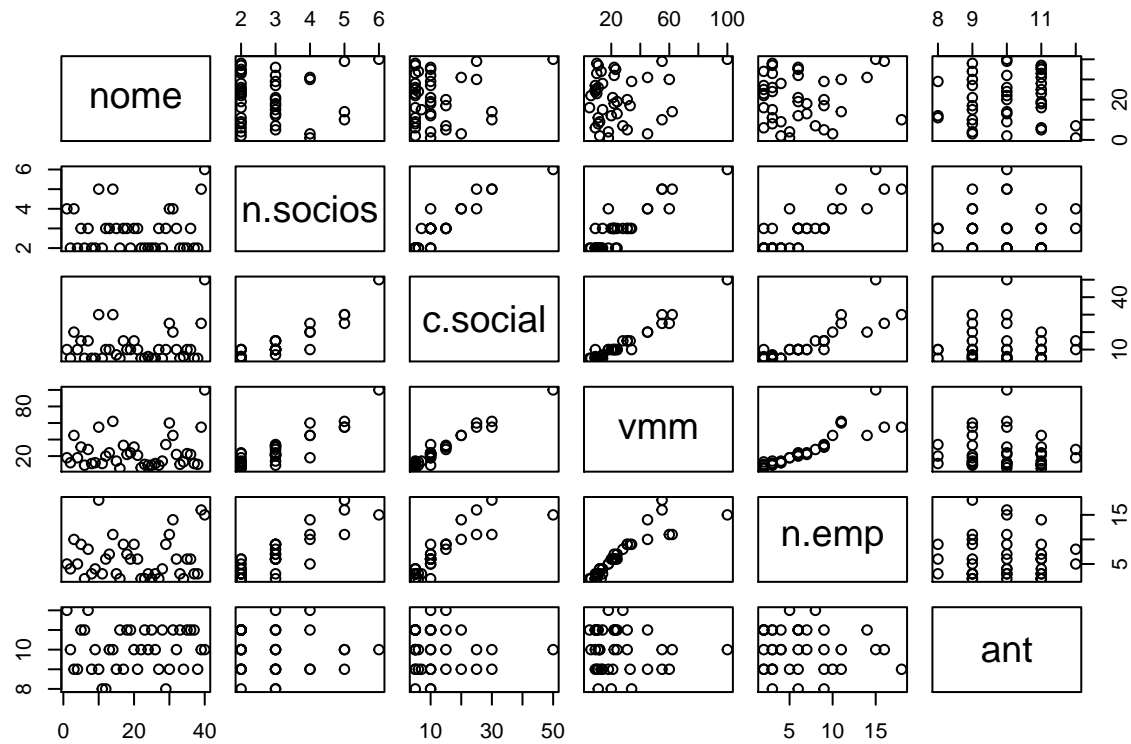


```
par(mfrow=c(1,1))
```

```
plot(emp$vmm,emp$n.socios,pch=16)
```



```
plot(emp)
```



## 1.7 Lists

```
uma.lista <- list(
  um.vector=1:10,
  uma.palavra="olá",
  uma.matrix=M,
  outra.lista=list(
    a="flor",
    b=rep(3,5)
  )
)

uma.lista["um.vector"]

## $um.vector
## [1] 1 2 3 4 5 6 7 8 9 10

uma.lista$um.vector

## [1] 1 2 3 4 5 6 7 8 9 10

uma.lista[1]

## $um.vector
## [1] 1 2 3 4 5 6 7 8 9 10
```

## 1.8 Functions

```
desconto <- function(price, discount=25){
  #Discount is a number between 0 and 100
  #calcula o desconto de um preço
  newPrice <- price*(1-discount/100)
  discount <- price - newPrice
  list(
    novo.preco=newPrice,
    desconto=discount)
}

desconto(1000,20)

## $novo.preco
## [1] 800
##
## $desconto
## [1] 200

desconto(1000,25)

## $novo.preco
## [1] 750
##
## $desconto
## [1] 250
```

This is how you function

## 2 Lessons 2

### 2.1 Random Variables and Vectors

#### 2.1.1 Elements of probability

A random variable  $\mathbf{X}$  is a function that takes an event space and return a value:

$$X : \Omega \rightarrow \mathbb{R}$$

#### 2.1.2 Expected value

1+1