Lista 07 - Fundamentos Estatísticos para Ciência dos Dados

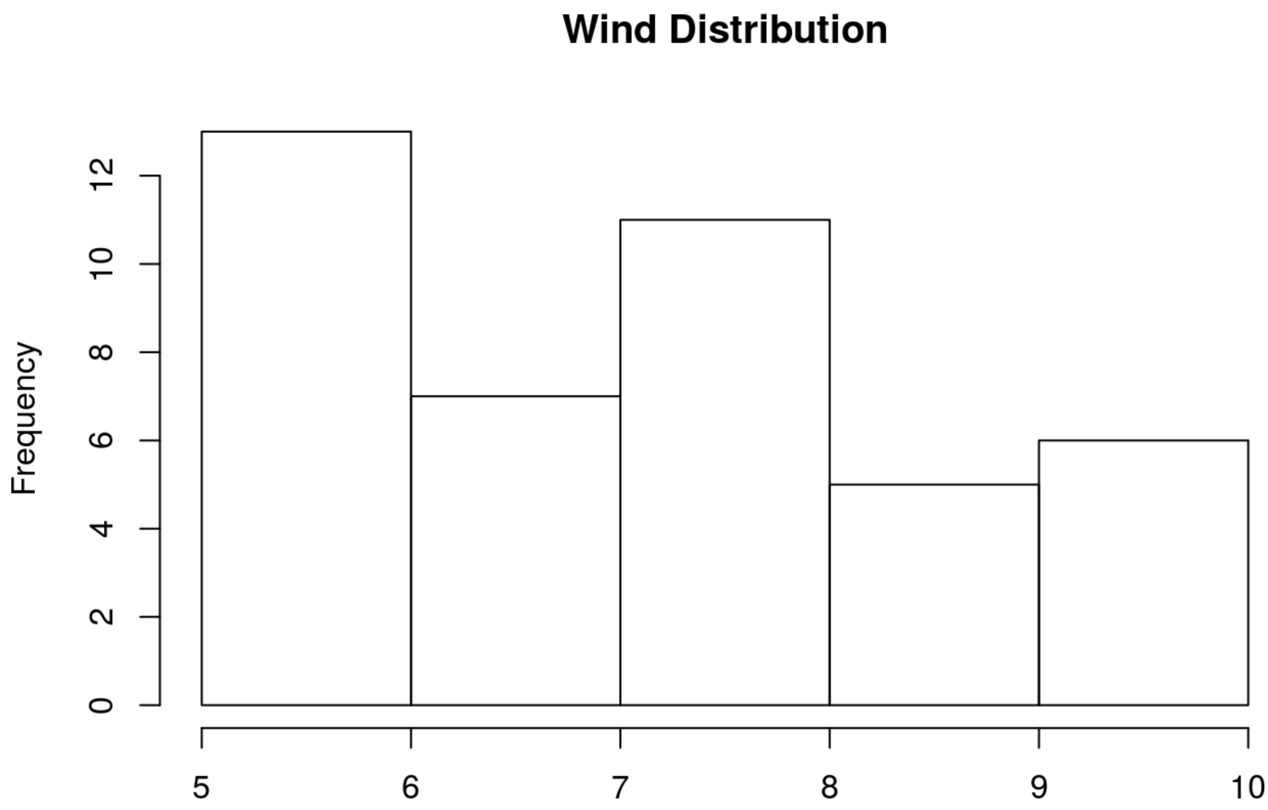
Nome: Vinícius de Oliveira Silva

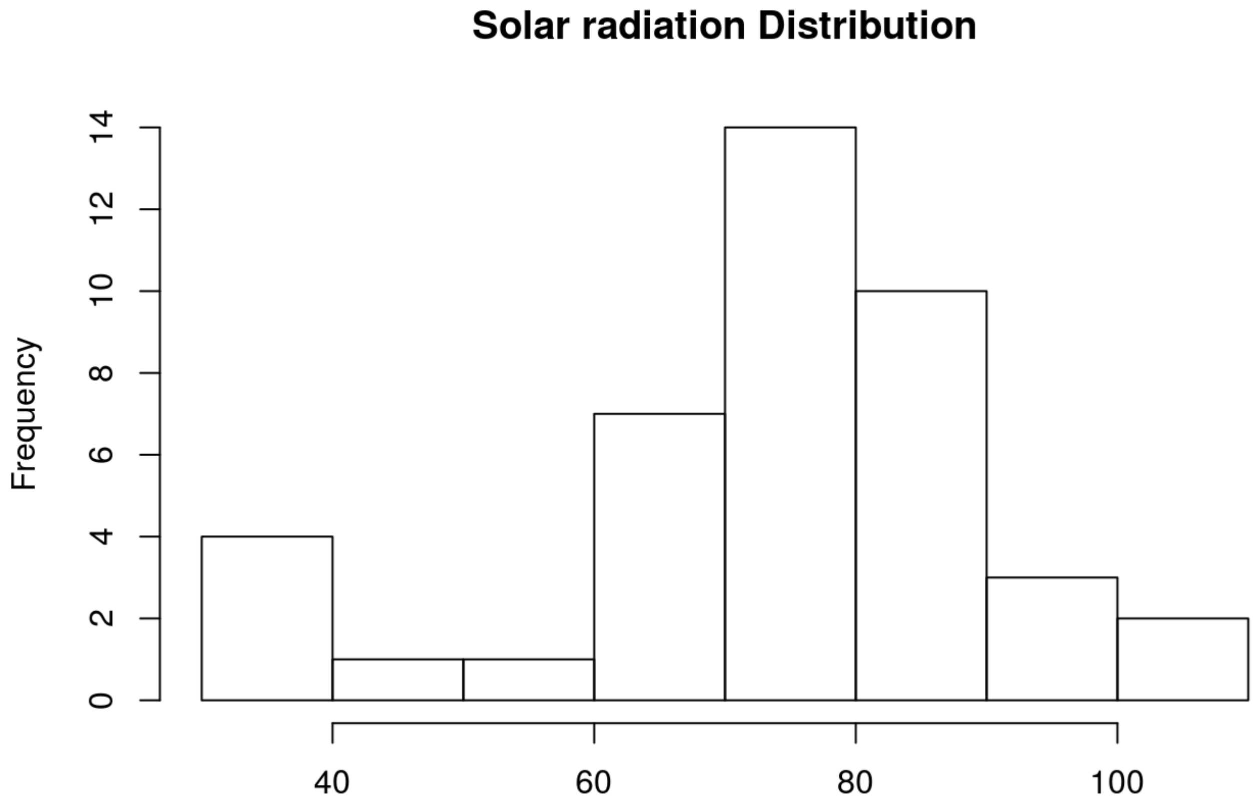
Matrícula: 2013007820

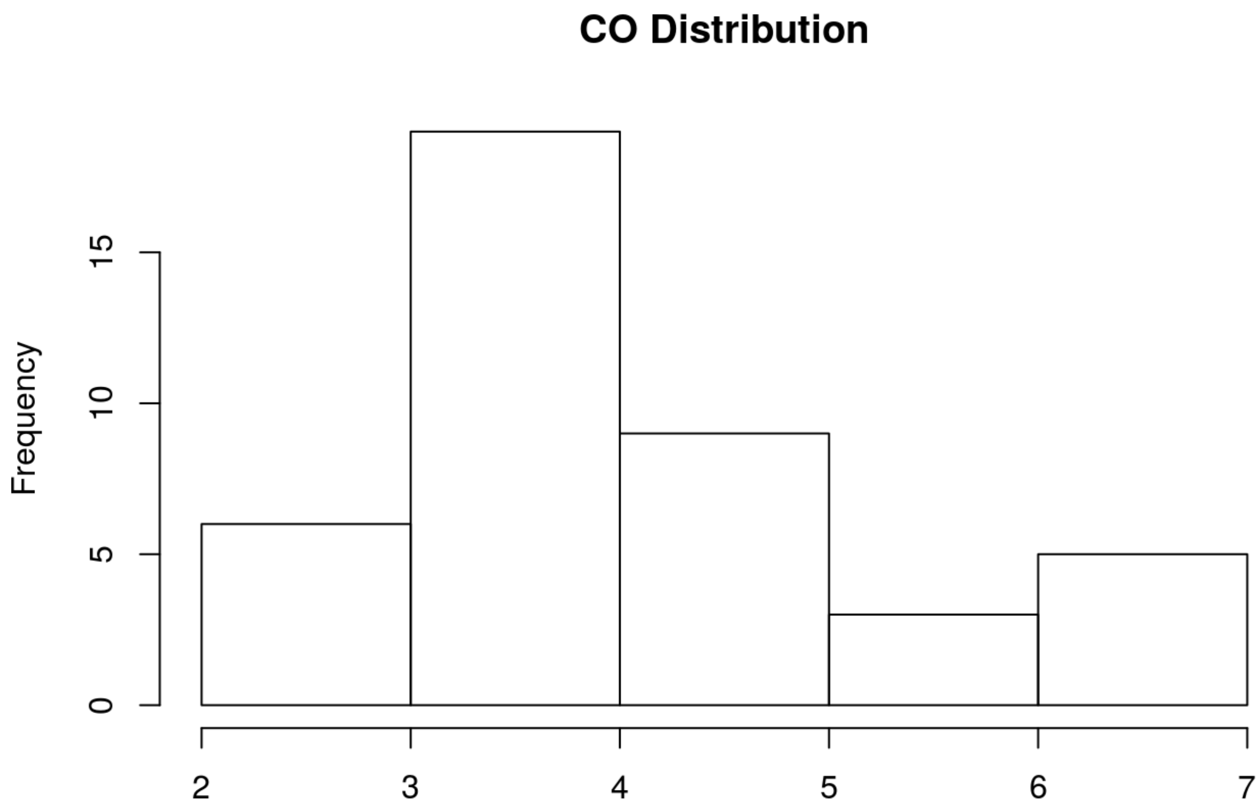
Exercícios Livro Johnson e Wichern

Questão 1.6

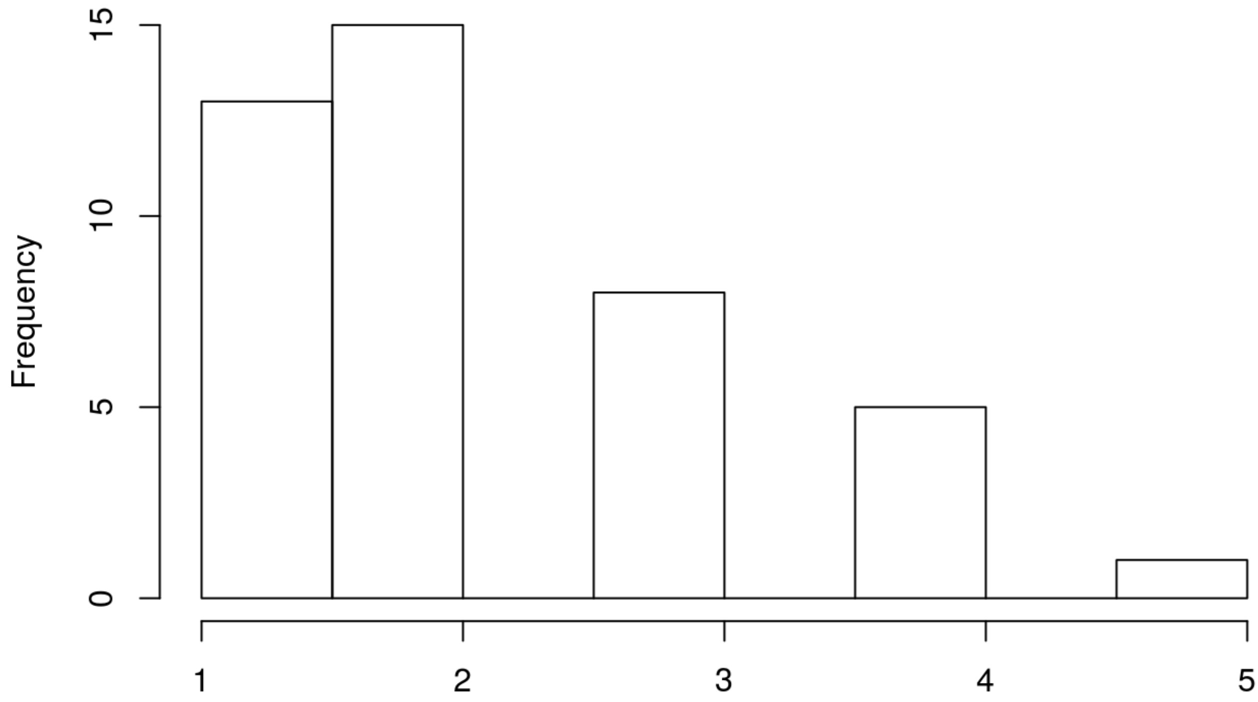
```
table = read.table("T1-5.dat", header=FALSE)


titles = c("Wind", "Solar radiation", "CO", "NO", "NO2", "O3", "HC")
tIndex <- 1

for (i in names(table)){
  hist(as.array(table[[i]]), main = paste(titles[tIndex], "Distribution"), xlab = "
")
  tIndex <- tIndex + 1
}
```



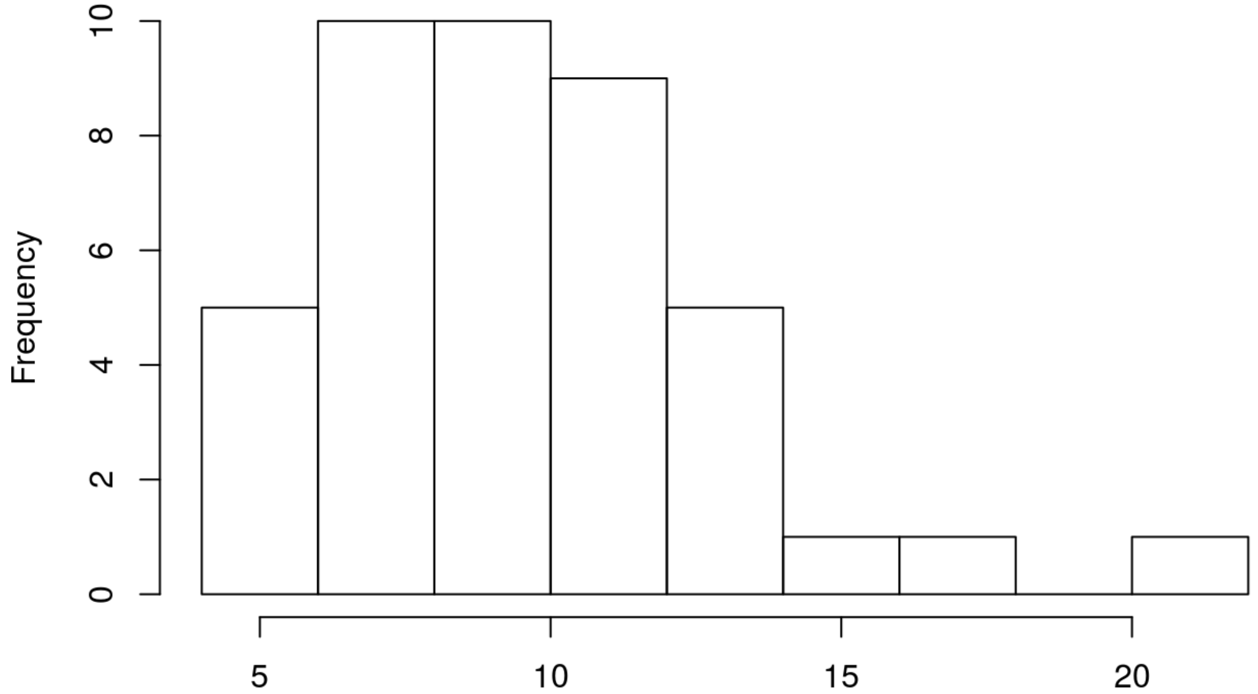

Solar radiation Distribution
Frequency
0
2
4
6
8
10
12
14
40
60
80
100

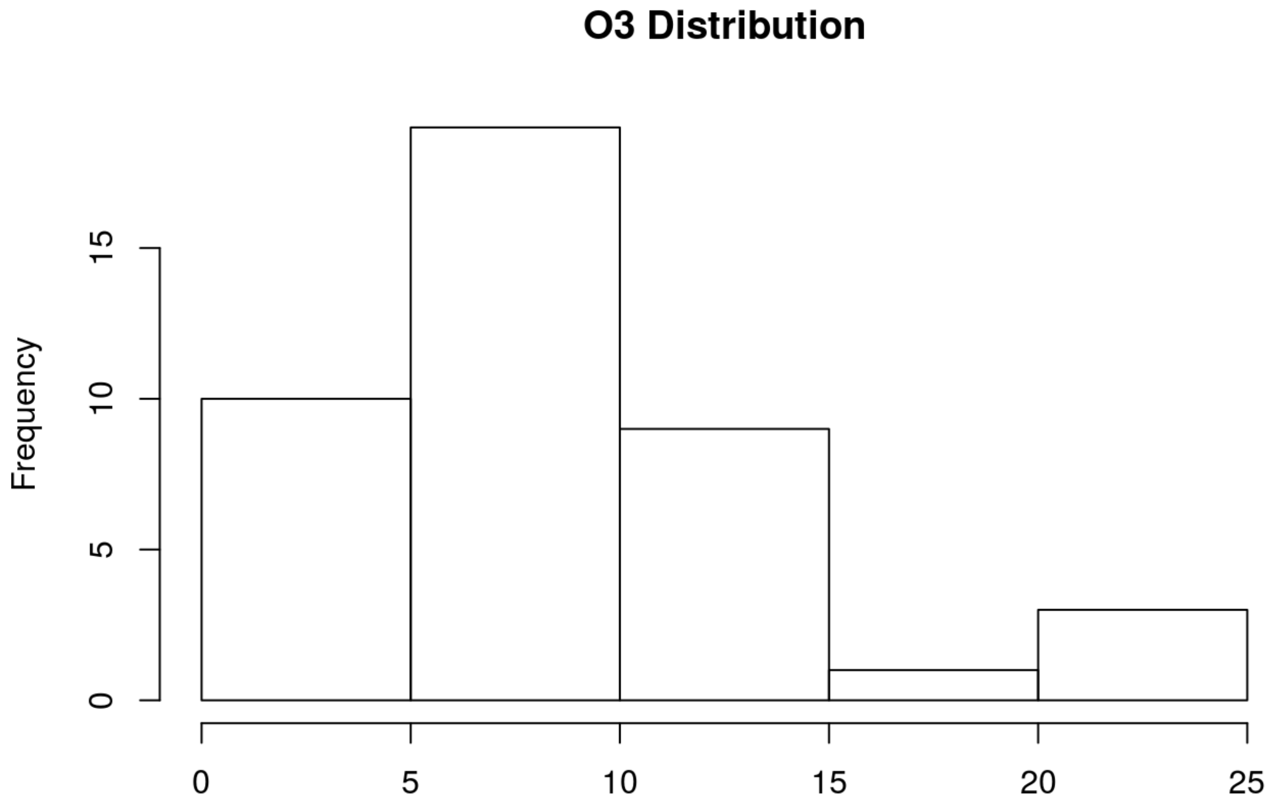

CO Distribution
Frequency
0
5
10
15
2
3
4
5
6
7

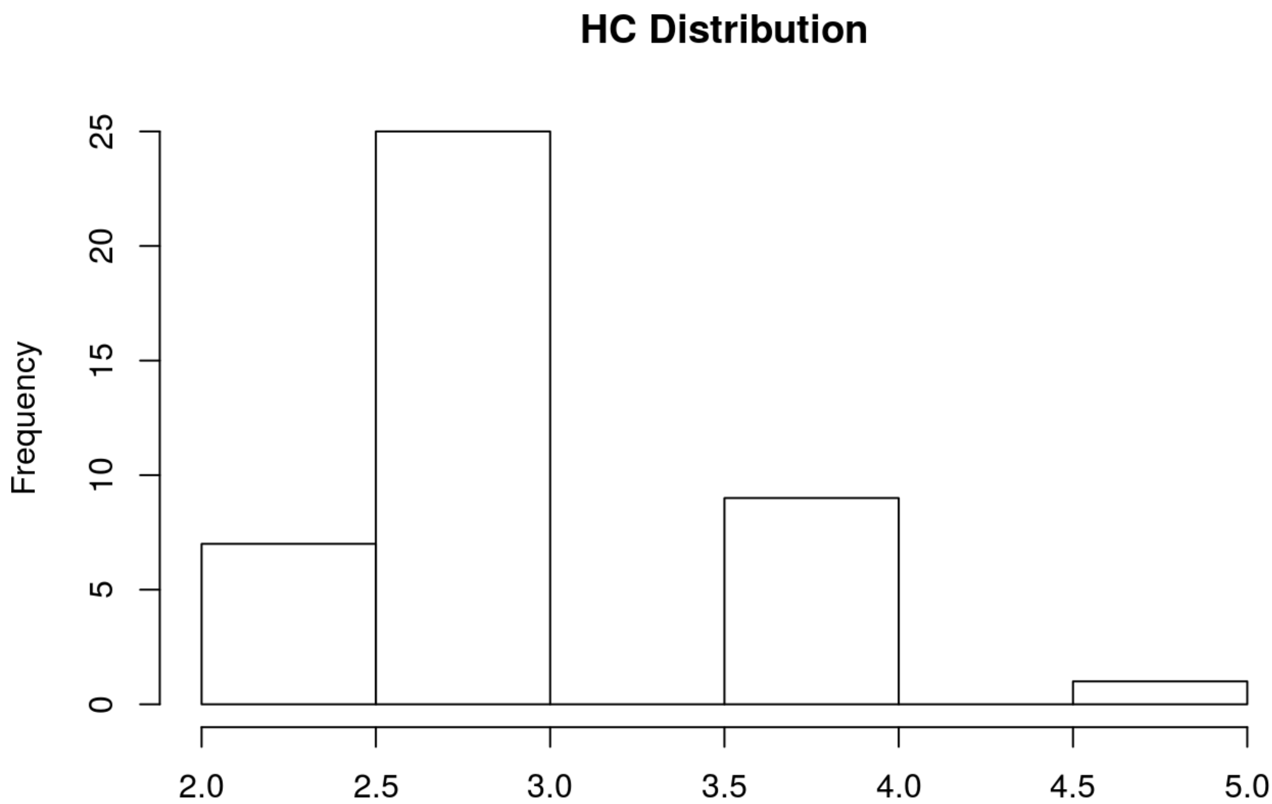
NO Distribution



NO2 Distribution




O3 Distribution
Frequency
0
5
10
15
0
5
10
15
20
25

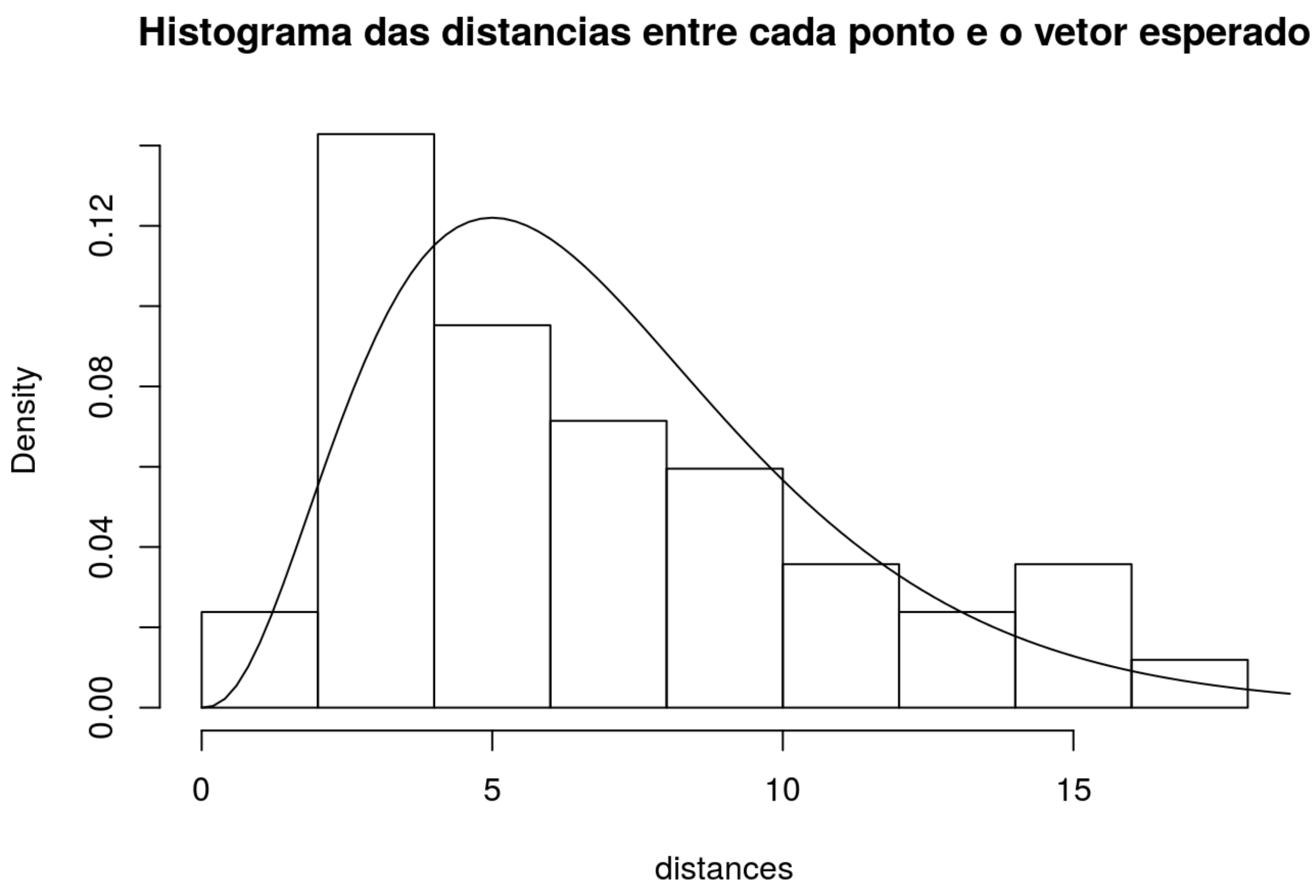

HC Distribution
Frequency
0
5
10
15
20
25
2.0
2.5
3.0
3.5
4.0
4.5
5.0

Definindo a função estatística $d^2(\mathbf{y}, \boldsymbol{\mu})$

```
#Gerando um vetor com as variancias de cada uma das 7 v.a's:
variances <- numeric(length = 7)
counter <- 1
for(i in names(table)){

  variances[counter] <- var(table[[i]])
  counter <- counter+1
}

#Computando a matriz de covariancia
#--------------------------------------------------------------------------------
-----
covMatrix <- diag(variances)

i<-1
j<-1

while(i<=7){
  j<-1
  while (j<=7) {
    if(i!=j){
      covMatrix[i,j]<-sqrt(variances[i])*sqrt(variances[j])*cor(table[i], table[j])
    }
    j<-j+1
  }
  i<-i+1
}

#--------------------------------------------------------------------------------
-----

#Computando o vetor esperado μ
#--------------------------------------------------------------------------------
-----
mu <- as.numeric(apply(table, MARGIN=2, FUN=mean))
#--------------------------------------------------------------------------------
-----

#Definindo a funcao:
d2 <- function(y){
  return (t(y-mu)%*%solve(covMatrix)%*%(y-mu))

}
```

Calculando a distancia de cada ponto ao vetor esperado, gerando o histograma das distancias e sobrepondo a curva chi-squared:

```
distances <- apply(table, MARGIN = 1, FUN = d2)
hist(distances, probability = TRUE, main="Histograma das distancias entre cada pont
o e o vetor esperado")
x <-rchisq(nrow(table), 7)
plot( function(x) dchisq(x, df=7), from=0, to=20, add=TRUE)
```



Questão 2.7

a)

Os autovalores da matriz são:

```
## [1] 10  5
```

Os autovetores são:

```
##            [,1]       [,2]
## [1,] -0.8944272 -0.4472136
## [2,]  0.4472136 -0.8944272
```

b)

Sabemos que a decomposição espectral de uma matriz A é dada por PDP^T, onde P é a matriz formada pelos autovetores de A e D é uma matriz em que os elementos da diagonal são os autovetores de A e os demais são 0s. Desta forma, para mostrarmos a decomposição espectral da matriz, basta que apresentemos a matriz P, a matriz D e a matriz P^T.

Matriz P:

```
##             [,1]       [,2]
## [1,] -0.8944272 -0.4472136
## [2,]  0.4472136 -0.8944272
```

Matriz D:

```
##      [,1] [,2]
## [1,]   10    0
## [2,]    0    5
```

Matriz P^T:

```
##             [,1]       [,2]
## [1,] -0.8944272  0.4472136
## [2,] -0.4472136 -0.8944272
```

c)

Para encontrarmos a inversa da matriz A, fazemos:

```
matrixAInverse <-solve(matrixA)
print(matrixAInverse)
```

```
##      [,1] [,2]
## [1,] 0.12 0.04
## [2,] 0.04 0.18
```

d)

Os autovetores de A^{-1} são:

```
##            [,1]       [,2]
## [1,] 0.4472136 -0.8944272
## [2,] 0.8944272  0.4472136
```

Os autovalores de A^{-1} são:

```
## [1] 0.2 0.1
```

Questão 2.18

??