act_Matrices&Vectores

Carlos David Contreras Chacon

2022-10-05

Matrices y vectores aleatorios

1.

```
X = matrix(c(1, 6, 8, 4, 2, 3, 3, 6, 3), ncol = 3)
X
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    3
## [2,]    6    2    6
## [3,]    8    3    3
```

```
bX = X %*% c(1, 1, 1)
cX = X %*% c(1, 2, -3)
A = cbind(bX, cX)
A
```

```
##      [,1] [,2]
## [1,]    8    0
## [2,]   14   -8
## [3,]   14    5
```

a)

```
bX_mu = mean(bX)
cX_mu = mean(cX)

bX_var = var(bX[,1])
cX_var = var(cX)
cov = cov(bX[,1], cX[,1])
cat("Media de b'X:", bX_mu, "\n")
```

```
## Media de b'X: 12
```

```
cat("Media de c'X:", cX_mu, "\n", "\n")
```

```
## Media de c'X: -1
##
```

```
cat("Varianza de b'X:", bX_var, "\n")
```

```
## Varianza de b'X: 12
```

```
cat("Varianza de c'X:", cX_var, "\n", "\n")
```

```
## Varianza de c'X: 43
##
```

```
cat("Covarianza:", cov, "\n")
```

```
## Covarianza: -3
```

b)

```
covA = cov(A)
bX_det = det(covA)
cat("Determinante de cov_A(S):", bX_det, "\n")
```

```
## Determinante de cov_A(S): 507
```

c)

```
cat("Matriz de Varianzas-Covarianzas:", covA, "\n")
```

```
## Matriz de Varianzas-Covarianzas: 12 -3 -3 43
```

d)

```
e = eigen(covA)
e
```

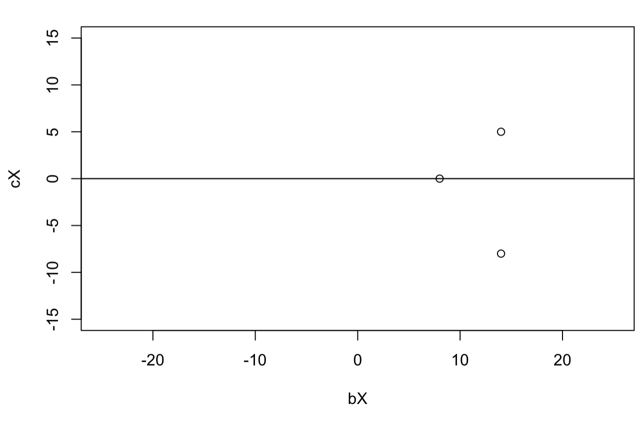
```
## eigen() decomposition
## $values
## [1] 43.28765 11.71235
##
## $vectors
##             [,1]        [,2]
## [1,] -0.09544671 -0.99543454
## [2,]  0.99543454 -0.09544671
```

e)

Son independientes puesto que ambos pesos en los coeficientes contribuyen para poder obtener Y.

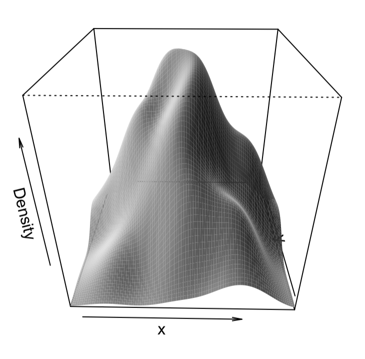
f)

```
plot(bX, cX, xlim = c(-25,25), ylim= c(-15,15))
x11 = seq(0, 100, 100)
x12 = e$vectors[1,1] / e$vectors[2,1] * x11
x21 = seq(0,100,100)
x22 = e$vectors[1,1] / e$vectors[1,2] * x21
abline(x11,x12)
abline(x21,x22)
```



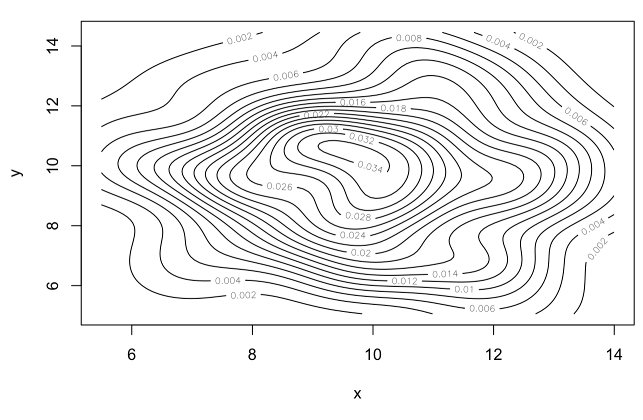
2.

```
library(MVN)
x = rnorm(100, 10, 2)
y = rnorm(100, 10, 2)
datos = data.frame(x,y)
mvn(datos, mvnTest = "hz", multivariatePlot = "persp")
```




```
## $multivariateNormality
##            Test        HZ   p value MVN
## 1 Henze-Zirkler 0.3575401 0.8869578 YES
##
## $univariateNormality
##               Test  Variable Statistic   p value Normality
## 1 Anderson-Darling     x        0.1940    0.8907    YES
## 2 Anderson-Darling     y        0.2806    0.6354    YES
##
## $Descriptives
##     n     Mean  Std.Dev    Median      Min      Max     25th     75th
## x 100 9.979267 1.843290 10.045121 5.501237 13.99113 8.726564 11.33220
## y 100 9.732361 2.023101  9.847664 5.060731 14.44824 8.400162 10.94706
##          Skew   Kurtosis
## x -0.16402519 -0.5185468
## y  0.05596624 -0.3573964
```

```
mvn(datos, mvnTest = "hz", multivariatePlot = "contour")
```




```
## $multivariateNormality
##            Test        HZ   p value MVN
## 1 Henze-Zirkler 0.3575401 0.8869578 YES
##
## $univariateNormality
##               Test  Variable Statistic   p value Normality
## 1 Anderson-Darling     x        0.1940    0.8907    YES
## 2 Anderson-Darling     y        0.2806    0.6354    YES
##
## $Descriptives
##     n     Mean  Std.Dev    Median      Min      Max     25th     75th
## x 100 9.979267 1.843290 10.045121 5.501237 13.99113 8.726564 11.33220
## y 100 9.732361 2.023101  9.847664 5.060731 14.44824 8.400162 10.94706
##          Skew   Kurtosis
## x -0.16402519 -0.5185468
## y  0.05596624 -0.3573964
```

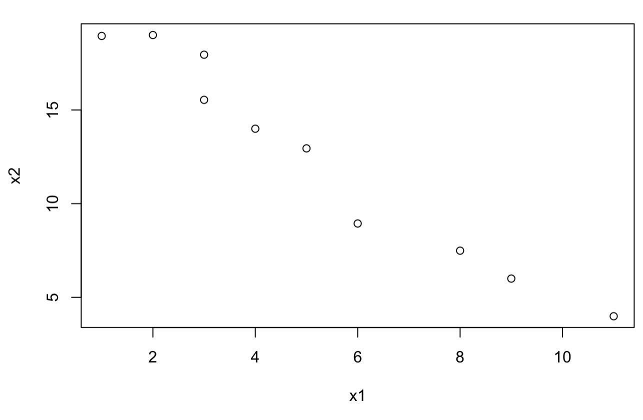

Viendo las funciones, tenemos un diagrama que ofrece una distribución de la probabilidad de una variable en la superficie de tres dimensioens. Podemos ver como se relacionan las variables entre si. Tambien hay una que muestra como se comporta la distribución en cuanto a una normal multivariable.

3.

```
n = 10 #Cantidad de datos
x1 = c(1,2,3,3,4,5,6,8,9,11) #Años
x2 = c(18.95, 19.00, 17.95, 15.54, 14.00, 12.95, 8.94, 7.49, 6.00, 3.99) #Precio
X = cbind(x1,x2) #DF
Xbar = colMeans(X) #Media
S = cov(X) # Covarianza
Sinv = solve(S)
```

a)

```
plot(x1, x2) # Diagrama de dispersión
```



b)

Es visible que conforme x1 crece, x2 disminuye por lo que la covarianza es negativa.

c)

```
d = mahalanobis(X, colMeans(X), cov(X)) #Distancia de Mahalanobis
d
```

```
##  [1] 1.8753045 2.0203262 2.9009088 0.7352659 0.3105192 0.0176162 3.7329012
##  [8] 0.8165401 1.3753379 4.2152799
```

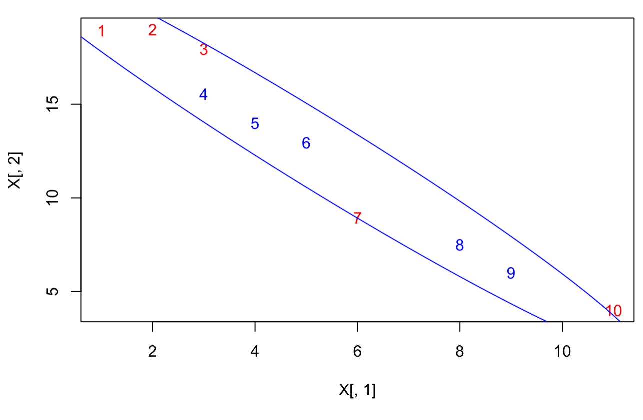
d)

```
library(ellipse)
```

```
##
## Attaching package: 'ellipse'
```

```
## The following object is masked from 'package:graphics':
##
##     pairs
```

```
p = 2
elps = t(t(ellipse(S, level=0.85, npoints=1000))+Xbar)
plot(X[,1],X[,2],type="n")
index = d < qchisq(0.5,df=p)
text(X[,1][index],X[,2][index],(1:n)[index],col="blue")
text(X[,1][!index],X[,2][!index],(1:n)[!index],col="red")
lines(elps,col="blue")
```

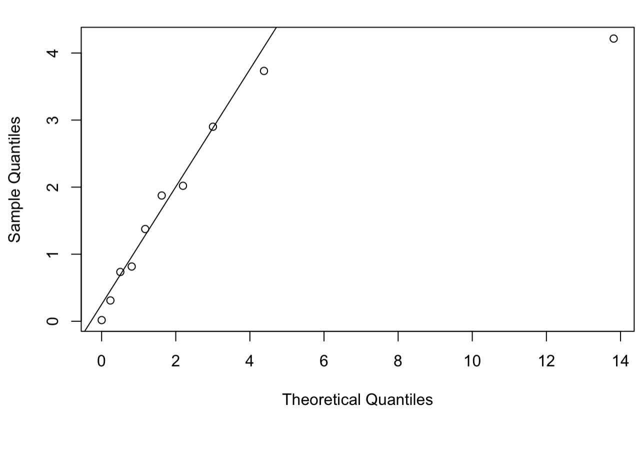


e)

```
names(d) = 1:10
sort(d)
```

```
##         6         5         4         8         9         1         2         3
## 0.0176162 0.3105192 0.7352659 0.8165401 1.3753379 1.8753045 2.0203262 2.9009088
##         7        10
## 3.7329012 4.2152799
```

```
qqplot(qchisq(ppoints(500),df=p), d, main="",
xlab="Theoretical Quantiles", ylab="Sample Quantiles")
qqline(d,distribution=function(x){qchisq(x,df=p)})
```



f)

Concluyendo con los resultados de la parte b y c, los datos son aproximadamente normales en dos dimensiones. Se puede observar como los datos se aproximan a la linea teórica de una normal.