

Métodos Multivariados: Tarea 3

Aldo Carmona, Diego Arellano, Mateo De La Roche, Victor Contreras

Ejercicio 1

Encuentra los estimados máximo-verosímiles del vector promedio μ y la matriz de covarianzas Σ de la muestra aleatoria X de una población normal bivariada.

La muestra aleatoria X es:

$$X = \begin{bmatrix} 3 & 6 \\ 4 & 4 \\ 5 & 7 \\ 4 & 7 \end{bmatrix}$$

Los estimados máximo-verosímiles son:

Para el vector promedio μ :

$$\mu = \begin{pmatrix} 4.0 \\ 6.0 \end{pmatrix}$$

Para la matriz de covarianzas Σ :

$$\Sigma = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})'$$

$$\Sigma = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & 2 \end{pmatrix}$$

Ejercicio 2

Sean (x_1, \dots, x_{20}) una muestra de una población $(N_6(\mu, \Sigma))$. Especificar cada una de las distribuciones completamente de las siguientes variables:

a. $(x_1 - \mu)' \Sigma^{-1} (x_1 - \mu)$

La variable $(x_1 - \mu)' \Sigma^{-1} (x_1 - \mu)$ es una forma cuadrática que, bajo la suposición de normalidad multivariada, sigue una distribución Chi-cuadrado con k grados de libertad, donde k es la dimensión del vector x_i . En este caso, $k = 6$ debido a que la muestra proviene de una distribución normal multivariada de dimensión 6.

b. \bar{x} y $\sqrt{n}(\bar{x} - \mu)$

El vector de medias de la muestra \bar{x} sigue una distribución normal multivariada $\mathcal{N}_6(\mu, \frac{1}{n}\Sigma)$. La variable $\sqrt{n}(\bar{x} - \mu)$, que es un escalamiento de \bar{x} , sigue una distribución normal multivariada $\mathcal{N}_6(0, \Sigma)$ debido al Teorema del Límite Central, que aplica incluso en el contexto multivariado.

c. $(n - 1)S$

La matriz S es la matriz de covarianzas de la muestra y, al multiplicarla por $(n - 1)$, obtenemos una variable que sigue una distribución Wishart con $n - 1$ grados de libertad y matriz de parámetro Σ . Esto se debe a que la distribución Wishart es la generalización de la distribución Chi-cuadrado para matrices de covarianzas muestrales.

d. $n(\bar{x} - \mu)' \Sigma^{-1} (\bar{x} - \mu)$ **(distribución aproximada)**

La variable $n(\bar{x} - \mu)' \Sigma^{-1} (\bar{x} - \mu)$ es una forma cuadrática basada en la media de la muestra. Bajo la suposición de normalidad multivariada, seguiría una distribución Chi-cuadrado con k grados de libertad si \bar{x} fuera la verdadera media poblacional. Sin embargo, como \bar{x} es una estimación, la distribución resultante es solo aproximadamente Chi-cuadrado y esta aproximación mejora con el tamaño de la muestra. En este caso, $k = 6$ ya que es la dimensión del vector \bar{x} .

Ejercicio 3

Harry Roberts, un naturalista del Departamento de Pesca de Alaska, estudia a los osos grizzly con el objetivo de mantener una población sana. Medidas de $n = 61$ osos dan los siguientes estimados de \bar{x} y S . Las variables son: peso (kg), longitud del cuerpo (cm), cuello (cm), circunferencia (cm), longitud de la cabeza (cm), ancho de la cabeza (cm):

```
n <- 61 # 61 osos
xbar <- c(95.52, 164.38, 55.69, 93.39, 17.98, 31.13)
S <- matrix(c(3266.46, 1343.97, 731.54, 1175.50, 162.68, 238.37,
```

```

1343.97, 721.91, 324.25, 537.35, 80.17, 117.73,
731.54, 324.25, 179.28, 281.17, 39.15, 56.80,
1175.50, 537.35, 281.17, 474.98, 63.73, 94.85,
162.68, 80.17, 39.15, 63.73, 9.95, 13.88,
238.37, 117.73, 56.80, 94.85, 13.88, 21.26),
nrow = 6)

```

- a. Obtener los intervalos de confianza simultáneos de 95 % para muestras grandes para las seis medias poblacionales de medidas corporales.

Para cada una de las seis variables, el intervalo de confianza del 95% se calcula como $\bar{x} \pm t_{\frac{\alpha}{2}, n-1} \times \frac{S}{\sqrt{n}}$, donde \bar{x} es la media muestral, S es la desviación estándar de la muestra, n es el tamaño de la muestra, y $t_{\frac{\alpha}{2}, n-1}$ es el valor crítico de la distribución t de Student con $n - 1$ grados de libertad.

```

S_sqrt <- sqrt(diag(S)) # Convertimos la varianza en desviación estándar

# Intervalos de confianza simultáneos de 95%
alpha <- 0.05
t_critical <- qt(1 - alpha/2, df=n-1) # Valor crítico de t para dos colas
cat("Valor crítico con t de dos colas: ", t_critical)

```

Valor crítico con t de dos colas: 2.000298

```

# Cálculos de los intervalos de confianza
CI <- sapply(1:length(xbar), function(i) {
  se <- S_sqrt[i] / sqrt(n)
  lower_bound <- xbar[i] - t_critical * se
  upper_bound <- xbar[i] + t_critical * se
  return(c(lower_bound, upper_bound))
})
CI <- t(CI) # Transponemos para obtener una matriz más legible
CI

```

```

      [,1]      [,2]
[1,] 80.88245 110.15755
[2,] 157.49869 171.26131
[3,] 52.26078 59.11922
[4,] 87.80829 98.97171
[5,] 17.17213 18.78787
[6,] 29.94910 32.31090

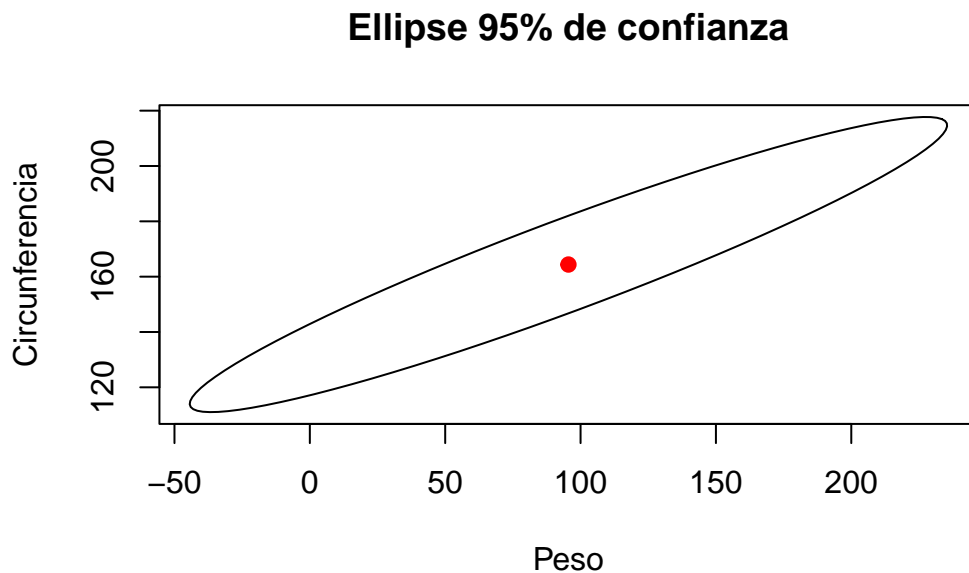
```

- b. Obtener la elipse de confianza simultánea del 95 % de grandes muestras para el peso medio y la circunferencia media.

```
# Seleccionamos las covarianzas para el peso y la circunferencia del matriz S original
cov_matrix <- S[c(1, 4), c(1, 4)]

# Calculamos la elipse de confianza para el 95%
ellipse_data <- ellipse(cov_matrix, centre = xbar[1:2], level = 0.95)

# Graficamos la elipse
plot(ellipse_data, type = 'l', xlab = 'Peso', ylab = 'Circunferencia',
     main = 'Ellipse 95% de confianza')
points(xbar[1], xbar[2], pch = 19, col = 'red') # Añadir el punto central
```



- c. Obtener los intervalos de confianza Bonferronizados del 95 % para las seis medias en la parte (a).

```
# Intervalos de confianza Bonferronizados del 95%
alpha_bonferroni <- alpha / length(xbar)
t_critical_bonferroni <- qt(1 - alpha_bonferroni/2, df=n-1)

CI_bonferroni <- sapply(1:length(xbar), function(i) {
```

```

    se <- S[i] / sqrt(n)
    lower_bound <- xbar[i] - t_critical_bonferroni * se
    upper_bound <- xbar[i] + t_critical_bonferroni * se
    return(c(lower_bound, upper_bound))
  })
CI_bonferroni <- t(CI_bonferroni)

for (i in 1:nrow(CI_bonferroni)) {
  cat(sprintf("Variable %d: Intervalo de confianza del 95%% Bonferronizado: [%f, %f]\n",
              i, CI_bonferroni[i, 1], CI_bonferroni[i, 2]))
}

```

```

Variable 1: Intervalo de confianza del 95% Bonferronizado: [-1045.635066, 1236.675066]
Variable 2: Intervalo de confianza del 95% Bonferronizado: [-305.143023, 633.903023]
Variable 3: Intervalo de confianza del 95% Bonferronizado: [-199.877366, 311.257366]
Variable 4: Intervalo de confianza del 95% Bonferronizado: [-317.277138, 504.057138]
Variable 5: Intervalo de confianza del 95% Bonferronizado: [-38.853118, 74.813118]
Variable 6: Intervalo de confianza del 95% Bonferronizado: [-52.145819, 114.405819]

```

- d. Referirse a pa parte (b). Construir el rectángulo de confianza Bonferroni de 95 % para el peso medio y la circunferencia media usando $m = 6$. Comparar este rectángulo con la elipse de confianza en la parte (b).

El rectángulo de confianza Bonferroni se traza utilizando los intervalos de confianza Bonferronizados para el peso medio y la circunferencia media, proporcionando una región de confianza que es más conservadora que la elipse de confianza.

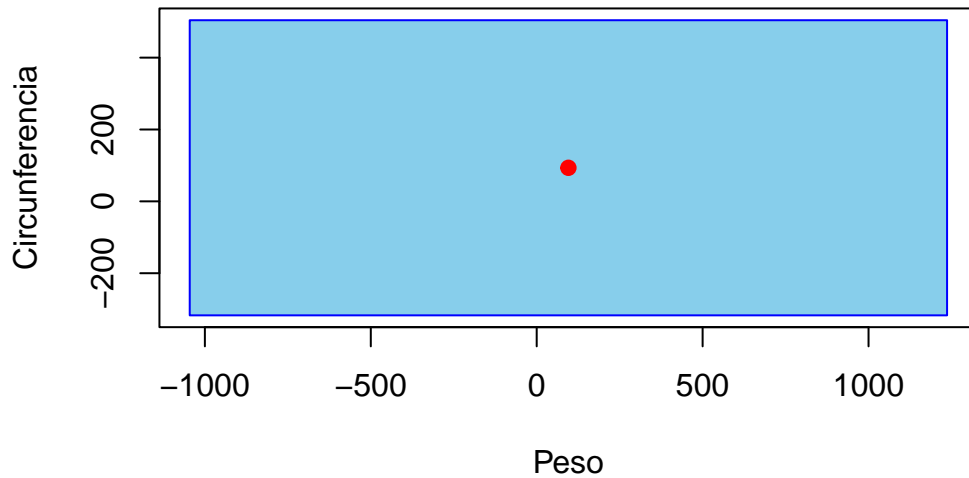
```

# Intervalos de confianza Bonferronizados para el peso medio y la circunferencia media
CI_bonferroni_peso <- CI_bonferroni[1, ]
CI_bonferroni_circunferencia <- CI_bonferroni[4, ]

# Dibuja el rectángulo de confianza Bonferroni
plot(NULL, xlim=c(min(CI_bonferroni_peso), max(CI_bonferroni_peso)),
      ylim=c(min(CI_bonferroni_circunferencia),
              max(CI_bonferroni_circunferencia)),
      xlab="Peso", ylab="Circunferencia",
      main="Rectángulo de Confianza Bonferroni")
rect(CI_bonferroni_peso[1], CI_bonferroni_circunferencia[1],
     CI_bonferroni_peso[2], CI_bonferroni_circunferencia[2],
     col='skyblue', border='blue')
points(xbar[1], xbar[4], pch=19, col='red')

```

Rectángulo de Confianza Bonferroni



- e. Obtener el intervalo de confianza de 95 % de Bonferroni para la diferencia del ancho medio de la cabeza y el largo medio de la cabeza usando $m = 6 + 1 = 7$ para considerar esta afirmación así como para cada media individual.

Para la diferencia entre el ancho medio y el largo medio de la cabeza, el intervalo de confianza de Bonferroni se calcula como sigue:

$$(\bar{x}_{ancho} - \bar{x}_{largo}) \pm t_{\frac{\alpha}{2m}, n-1} \times S_{diferencia}$$

donde $S_{diferencia}$ es la desviación estándar de la diferencia entre las dos medias, n es el tamaño de la muestra, m es el número total de pruebas o comparaciones, y $t_{\frac{\alpha}{2m}, n-1}$ es el valor crítico de la distribución t de Student con $n - 1$ grados de libertad ajustado para m comparaciones.

```
# Asumiendo que la sexta y quinta columnas de xbar y S corresponden
# al ancho y largo de la cabeza, respectivamente
xbar_ancho <- xbar[6]
xbar_largo <- xbar[5]
S_ancho <- S[5]
S_largo <- S[6]

# Calculamos la desviación estándar de la diferencia
S_diferencia <- sqrt((S_ancho^2 + S_largo^2) / n)
```

```

# Intervalo de confianza Bonferroni para la diferencia
alpha_bonferroni_diferencia <- alpha / 7
t_critical_bonferroni_diferencia <- qt(1 - alpha_bonferroni_diferencia/2, df=n-1)

# Límites del intervalo de confianza para la diferencia
lower_bound_diferencia <-
  (xbar_ancho - xbar_largo) - t_critical_bonferroni_diferencia * S_diferencia
upper_bound_diferencia <-
  (xbar_ancho - xbar_largo) + t_critical_bonferroni_diferencia * S_diferencia

# Imprimir el intervalo de confianza para la diferencia
cat(sprintf("Intervalo de confianza del 95%% Bonferronizado para la diferencia entre
  \nel ancho y largo de la cabeza: [%f, %f]\n",
    lower_bound_diferencia, upper_bound_diferencia))

```

Intervalo de confianza del 95% Bonferronizado para la diferencia entre
el ancho y largo de la cabeza: [-89.774083, 116.074083]

Ejercicio 4

```

# Carga de librerías

# Definición del conjunto de datos
datos <- matrix(c(4, 4, NA, 8, NA, 3, 5, NA), nrow = 4, ncol = 3)

```

Warning in matrix(c(4, 4, NA, 8, NA, 3, 5, NA), nrow = 4, ncol = 3): data
length [8] is not a sub-multiple or multiple of the number of columns [3]

```

# Imprimir datos
print(datos)

```

	[,1]	[,2]	[,3]
[1,]	4	NA	4
[2,]	4	3	4
[3,]	NA	5	NA
[4,]	8	NA	8

```
# idvars informs the columns that are identification variables
a.out <- amelia(datos, m = 10, idvars=3)
```

Warning in amelia.prep(x = x, m = m, idvars = idvars, empri = empri, ts = ts, :
You have a small number of observations, relative to the number, of variables
in the imputation model. Consider removing some variables, or reducing the
order of time polynomials to reduce the number of parameters.

```
-- Imputation 1 --
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49
```

```
-- Imputation 2 --
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14
```

```
-- Imputation 3 --
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30 31
```

```
-- Imputation 4 --
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14
```

```
-- Imputation 5 --
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
```

```
-- Imputation 6 --
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
```

The resulting variance matrix was not invertible. Please check your data for highly collinearity.

```
-- Imputation 7 --
```



```

1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49

```

```
-- Imputation 8 --
```

```
1  2  3  4  5  6  7  8  9 10 11 12 13 14
```

```
-- Imputation 9 --
```

```
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25
```

```
-- Imputation 10 --
```

```
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
```

```
a.out$imputations
```

```
$imp1
```

	[,1]	[,2]	[,3]
[1,]	4.00000	3.0496	4
[2,]	4.00000	3.0000	4
[3,]	4.01325	5.0000	NA
[4,]	8.00000	607.8434	8

```
$imp2
```

	[,1]	[,2]	[,3]
[1,]	4.000000	5.006176	4
[2,]	4.000000	3.000000	4
[3,]	2.502437	5.000000	NA
[4,]	8.000000	5.020013	8

```
$imp3
```

	[,1]	[,2]	[,3]
[1,]	4.000000	3.002192	4
[2,]	4.000000	3.000000	4
[3,]	7.385742	5.000000	NA
[4,]	8.000000	5.381218	8

\$imp4

	[,1]	[,2]	[,3]
[1,]	4.00000	2.996498	4
[2,]	4.00000	3.000000	4
[3,]	12.01671	5.000000	NA
[4,]	8.00000	3.999233	8

\$imp5

	[,1]	[,2]	[,3]
[1,]	4.000000	4.960315	4
[2,]	4.000000	3.000000	4
[3,]	6.787551	5.000000	NA
[4,]	8.000000	5.016360	8

\$imp6

[1] NA

\$imp7

	[,1]	[,2]	[,3]
[1,]	4.000000	3.016706	4
[2,]	4.000000	3.000000	4
[3,]	4.012938	5.000000	NA
[4,]	8.000000	607.931863	8

\$imp8

	[,1]	[,2]	[,3]
[1,]	4.00000	3.007159	4
[2,]	4.00000	3.000000	4
[3,]	12.06352	5.000000	NA
[4,]	8.00000	4.028998	8

\$imp9

	[,1]	[,2]	[,3]
[1,]	4.00000	3.006977	4
[2,]	4.00000	3.000000	4
[3,]	8.30061	5.000000	NA
[4,]	8.00000	4.861853	8

\$imp10

	[,1]	[,2]	[,3]
[1,]	4.00000	3.02767	4
[2,]	4.00000	3.00000	4

```
[3,] 12.03317 5.00000 NA
[4,]  8.00000 4.01685  8
```

```
attr("class")
[1] "mi" "list"
```

Ejercicio 4 (Versión 2)

Dados los datos $X = \begin{pmatrix} 3 & 6 & 0 \\ 4 & 4 & 3 \\ NA & 8 & 3 \\ 5 & NA & NA \end{pmatrix}$ con componentes faltantes, usar el algoritmo EM

para estimar μ y Σ . Determinar los estimados iniciales, e iterar para encontrar los primeros estimados revisados.

```
X1 <- c(3,4,NA,5)
X2 <- c(6,4,8,NA)
X3 <- c(0,3,3,NA)
X <- data.frame(X1,X2,X3)
n <- nrow(X)

mu_est <- colMeans(X, na.rm = TRUE)

for (col in names(X)) {
  X[[col]][is.na(X[[col]])] <- mu_est[which(names(X) == col)]
}

var_est <- (n-1)/n * var(X)

for(i in (1:5)) {

  x_31 <- mu_est[1] + var_est[1, 2:3] %*%
    solve(var_est[2:3, 2:3]) %*%
    t(X[3, 2:3] - mu_est[2:3])

  x_31x_32 <- x_31*X[3,2]

  x_31x_33 <- x_31*X[3,3]

  x2_31 <- var_est[1,1] - var_est[1, 2:3] %*%
    solve(var_est[2:3, 2:3]) %*%
```

```

var_est[2:3,1] + x_31**2

x_42_43 <- mu_est[2:3] + var_est[1, 2:3] %*%
  solve(var_est[1,1]) %*%
  t(X[4, 1] - mu_est[1])

x_41x_42_43 <- x_42_43 * X[4,1]

x2_42_43 <- var_est[2:3,2:3] - var_est[1, 2:3] %*%
  solve(var_est[1,1]) %*%
  var_est[2:3,1] + x_42_43 %*%
  t(x_42_43)

T1 <- c(X[1,1]+X[2,1]+x_31+X[4, 1], X[1,2]+X[2,2]+X[3,2]+x_42_43[1],
        X[1,3]+X[2,3]+X[3,3]+x_42_43[2])

x_1_var <- X[1,1]**2 + X[2,1]**2 + x2_31 + X[4,1]**2

x_2_var <- X[1,2]**2 + X[2,2]**2 + X[3,2]**2 + x2_42_43[1,1]

x_3_var <- X[1,3]**2 + X[2,3]**2 + X[3,3]**2 + x2_42_43[2,2]

x_12_cov <- X[1,1]*X[1,2] + X[2,1]*X[2,2] + x_31x_32 + x_41x_42_43[1,1]

x_13_cov <- X[1,1]*X[1,3] + X[2,1]*X[2,3] + x_31x_33 + x_41x_42_43[2,1]

x_23_cov <- X[1,2]*X[1,3] + X[2,2]*X[2,3] + X[3,2]*X[3,3] + x2_42_43[1,2]

T2 <- matrix(c(x_1_var, x_12_cov, x_13_cov,
               x_12_cov, x_2_var, x_23_cov,
               x_13_cov, x_23_cov, x_3_var), nrow = 3, ncol = 3)

mu_est <- 1/n * (T1)
var_est <- 1/n * T2 - mu_est%*%t(mu_est)
}

```

Ejercicio 5

Este ejercicio se refiere al ave conocida como milano picogarfo

```
X <- read.delim("https://raw.githubusercontent.com/jvega68/EA3/master/datos/J&W/T5-12.DAT"
                sep = "", header = F)
str(X)
```

```
'data.frame':  45 obs. of  2 variables:
 $ V1: int  191 197 208 180 180 188 210 196 191 179 ...
 $ V2: int  284 285 288 273 275 280 283 288 271 257 ...
```

- a. Encuentra y bosqueja la elipse de 95% de confianza para las medias poblacionales μ_1 y μ_2 . Supongan que se sabe que $\mu_1 = 190\text{mm}$ y $\mu_2 = 275\text{mm}$ para machos milanos picogarfios. ¿Son valores plausibles para la longitud media de la cola y longitud media del ala para los pájaros hembras? Explicar.

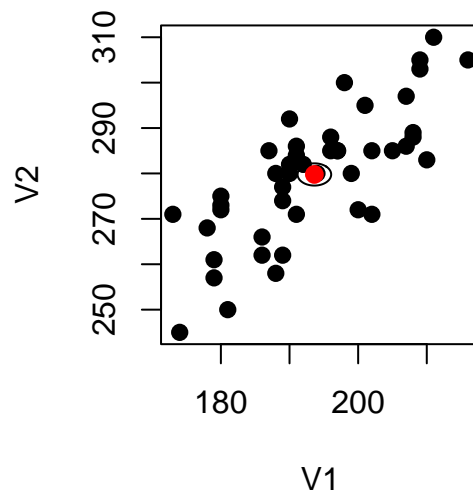
```
set.seed(1)

x_bar <- colMeans(X)

S <- var(X)

library(ellipse)
par(pty = "s")
plot(X, pch = 16, cex = 1.2,
     main= "Datos del Milano picogarfio")
lines(ellipse(0, centre = x_bar, t = sqrt(qchisq(0.95,2))))
points(x_bar[1],x_bar[2], col = "red", cex = 1.3, pch = 16)
```

Datos del Milano picogarfio



No son valores plausibles para las medias de las hembras, ya que las medias de los machos quedan fuera del intervalo de confianza, por lo que se esperaría que que las medias de las hembras lo compensarían.

- b. Construir los intervalos simultáneos T^2 de 95 % para μ_1 y μ_2 y los intervalos de Bonferroni de 95 % de confianza para las mismas cantidades. Comparar los dos conjuntos de intervalos. ¿Qué ventaja, si la hay tienen los intervalos T^2 sobre los intervalos Bonferroni?

Intervalos simultáneos T^2

```
n <- nrow(X)
p <- ncol(X)

c1 <- p*(n-1)/(n-p)*qf(.05,p,n-p,lower.tail = F)
for(i in 1:2)print(x_bar[i] + c(-1,1)*sqrt(c1*diag(S)[i]/n ))
```

```
[1] 189.4217 197.8227
```

```
[1] 274.2564 285.2992
```

Intervalos de Bonferroni

```
c3 <- qt(.05/(2*p), n-1, lower.tail = F)
for(i in 1:2) print(x_bar[i] + c(-1,1)*c3*sqrt(diag(S)[i]/n))
```

```
[1] 189.8216 197.4229
```

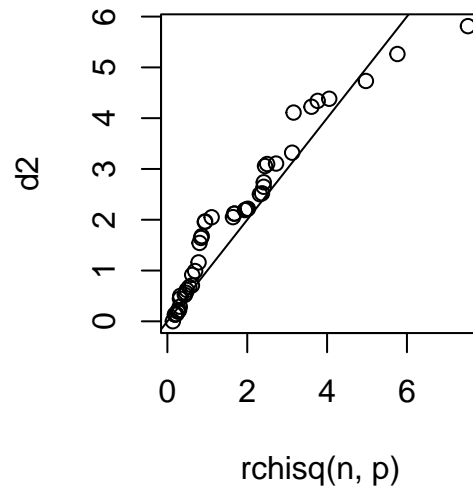
```
[1] 274.7819 284.7736
```

Sabemos que los intervalos simultáneos son más anchos que los de Bonferroni para satisfacer la restricción de nivel de confianza 95% dado que toma en cuenta todas las dependencias entre las variables. Si deseamos intervalos más anchos, usaríamos los simultáneos.

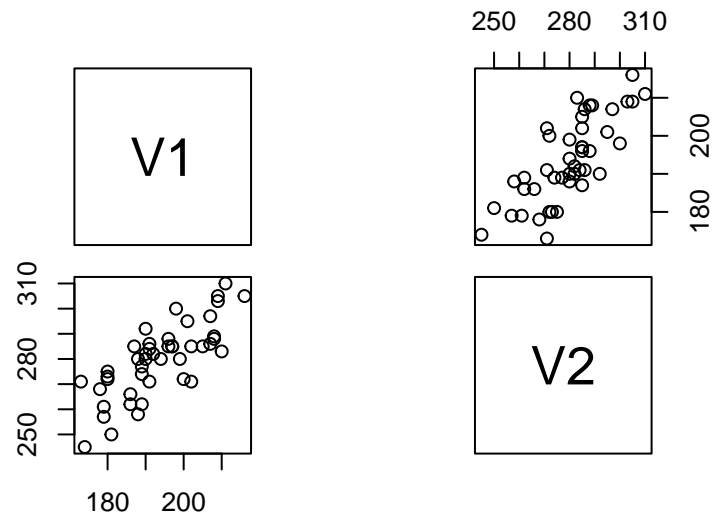
- c. Es la distribución normal bivariada un modelo poblacional viable? Explicar con referencia a qq-plots y un diagrama de dispersión.

```
par(mfrow = c(1,1), pty = "s")

qqplot(rchisq(n,p),mahalanobis(X, center = x_bar, cov = S), ylab="d2")
abline(a=0, b=1)
```



```
pairs(X)
```



La parte superior del qq-plot se aleja significativamente de la línea recta, por lo que no consideraría que la distribución normal bivariada es un modelo viable.

Ejercicio 6

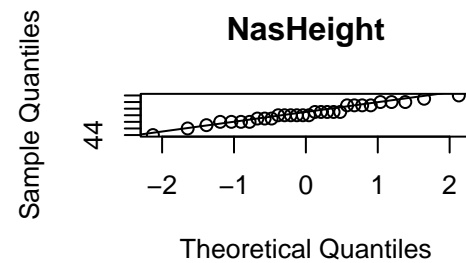
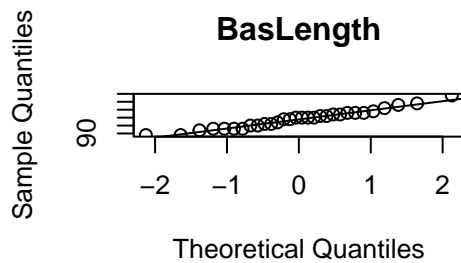
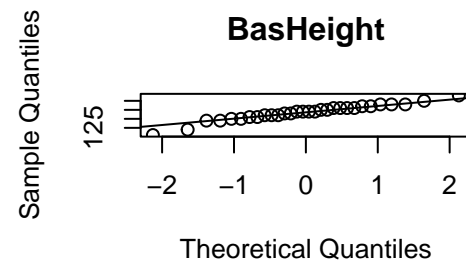
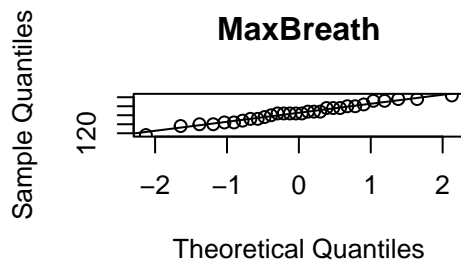
Consideren las 30 observaciones de cráneos de hombres egipcios para el primer periodo de tiempo:

```
X <- read.delim("https://raw.githubusercontent.com/jvega68/EA3/master/datos/J&W/T6-13.DAT"
                sep = "", header = F)
names(X) <- c("MaxBreath", "BasHeight", "BasLength", "NasHeight", "TimePeriod")
Y <- subset(X, TimePeriod==1)
Y$TimePeriod <- NULL
str(Y)
```

```
'data.frame':  30 obs. of  4 variables:
 $ MaxBreath: int  131 125 131 119 136 138 139 125 131 134 ...
 $ BasHeight: int  138 131 132 132 143 137 130 136 134 134 ...
 $ BasLength: int  89 92 99 96 100 89 108 93 102 99 ...
 $ NasHeight: int  49 48 50 44 54 56 48 48 51 51 ...
```

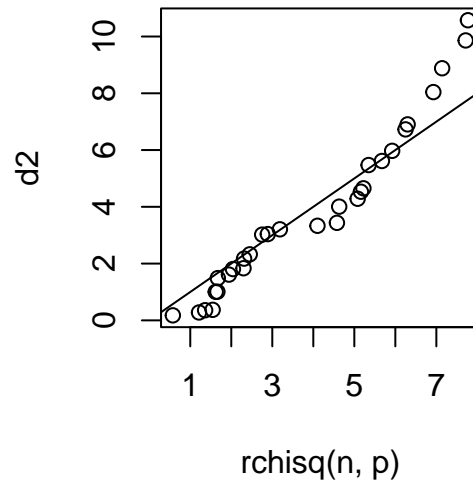

- a. Construir qq-plots de las distribuciones marginales para las cuatro variables. También construir la gráfica ji-cuadrada de las distribución multivariada. ¿Los datos se ven normales multivariados? Explicar.

```
par(mfrow = c(2,2))
for(i in 1:4) {
  qqnorm(Y[,i],main = names(Y)[i])
  qqline(Y[,i])
}
```



```
n <- nrow(Y)
p <- ncol(Y)

par(mfrow = c(1,1), pty = "s")
set.seed(1)
qqplot(rchisq(n,p),mahalanobis(Y, center = colMeans(Y), cov = var(Y)),ylab="d2")
abline(a=0, b=1)
```



Los datos parecen no desviarse tanto de la línea recta, por lo que podemos concluir que los datos se ven normales multivariados.

- b. Construir los intervalos de Bonferroni del 95% para cada una de las variables. También los simultáneos T^2 correspondientes y comparar.

```
ybar <- colMeans(Y)
S <- var(Y)

# Bonferroni
print("Intervalos de Bonferroni 95%")
```

```
[1] "Intervalos de Bonferroni 95%"
```

```
c3 <- qt(.05/(2*p), n-1, lower.tail = F)
for(i in 1:4) {
  print(ybar[i] + c(-1,1)*c3*sqrt(diag(S)[i]/n))
}
```

```
[1] 128.8727 133.8607
```

```
[1] 131.427 135.773
```

```
[1] 96.30548 102.02785
[1] 49.18965 51.87702
```

```
# Simultáneos  $T^2$ 
print("Intervalos simultáneos  $T^2$  95%")
```

```
[1] "Intervalos simultáneos  $T^2$  95%"
```

```
c1 <- p*(n-1)/(n-p)*qf(.05,p,n-p,lower.tail = F)
for(i in 1:4) {
  print(ybar[i] + c(-1,1)*sqrt(c1*diag(S)[i]/n ))
}
```

```
[1] 128.0909 134.6425
[1] 130.7458 136.4542
[1] 95.40858 102.92475
[1] 48.76844 52.29822
```

Nótese que los intervalos simultáneos son más anchos que los Bonferronizados porque toman en cuenta todas las posibles dependencias entre las variables, por lo que los de Bonferroni son más precisos.

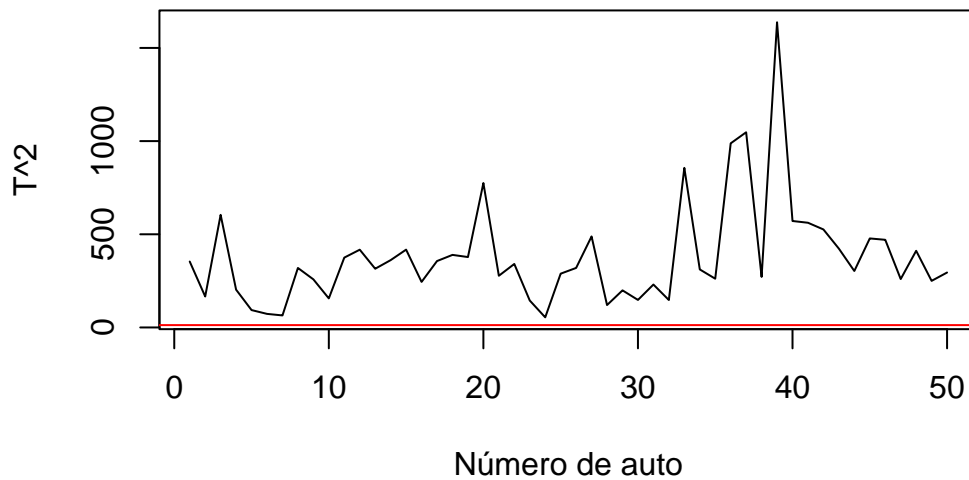
Ejercicio 7

```
X <- read.delim("https://raw.githubusercontent.com/jvega68/EA3/master/datos/J&W/T5-14.dat",
               sep = "", header = F)
first30 <- head(X, 30)
```

```
# estimamos  $\mu$  y  $S$  con 30 observaciones
n <- nrow(X)
p <- ncol(X)
mu <- colMeans(first30)
S <- var(first30)
```

```
T2 <- n*mahalanobis(X, mu, S)
plot(T2, type="l", main="Gráfica  $T^2$  para los 50 autos",
     xlab="Número de auto", ylab=" $T^2$ ")
abline(h=qchisq(0.95, p), col="red") # Límite crítico del 95%
```

Gráfica T^2 para los 50 autos



```
# Obtener los autos con valores de  $T^2$  por encima del límite crítico
autos_preocupantes <- which(T2 > qchisq(0.95, p))
```

Ejercicio 8

```
# cargamos los datos
X <- read.delim("https://raw.githubusercontent.com/jvega68/EA3/master/datos/J&W/T3-2.DAT",
               sep = ",", header = F)

# Calcular la media y la matriz de covarianza
media <- colMeans(X)
S <- (var(X))

# Realizar una transformación de los datos
X_transformed <- t(solve(chol(cov_matrix)) %*% t(X))

# Calcular la región de confianza del 95% para el vector de media
n <- nrow(X)
p <- ncol(X)
alpha <- 0.05
```

```

z_value <- qnorm(1 - alpha/2)
conf_interval <- matrix(0, nrow = 2, ncol = 2)
for (i in 1:p) {
  lower_bound <- media[i] - z_value * sqrt(1/(n-1)) * sqrt(cov_matrix[i,i])
  upper_bound <- media[i] + z_value * sqrt(1/(n-1)) * sqrt(cov_matrix[i,i])
  conf_interval[i, ] <- c(lower_bound, upper_bound)
}
conf_interval

```

```

      [,1]      [,2]
[1,] -13.44552 32.28552
[2,]  10.55273 27.99127

```

```

# Calcular los intervalos Bonferroni ajustados del 95%
bonferroni_factor <- qt(1 - alpha/(2*p), df = n - 1)
bonferroni_interval <- matrix(0, nrow = 2, ncol = 2)
for (i in 1:p) {
  lower_bound <- media[i] - bonferroni_factor * sqrt(1/(n-1)) * sqrt(cov_matrix[i,i])
  upper_bound <- media[i] + bonferroni_factor * sqrt(1/(n-1)) * sqrt(cov_matrix[i,i])
  bonferroni_interval[i, ] <- c(lower_bound, upper_bound)
}
bonferroni_interval

```

```

      [,1]      [,2]
[1,] -18.47353 37.31353
[2,]   8.63541 29.90859

```

```

# método del laboratorio
c1 <- p*(n-1)/(n-p)*qf(.05,p,n-p,lower.tail = F)
for(i in 1:length(media))print(media[i] + c(-1,1)*sqrt(c1*diag(S)[i]/n ))

```

```

[1]  7.410227 11.429773
[1] 15.05537 23.48863

```

```

# bonferronizados (método del lab)
c3 <- qt(.05/(2*p), n-1, lower.tail = F)
for(i in 1:length(media)) print(media[i] + c(-1,1)*c3*sqrt(diag(S)[i]/n))

```

[1] 7.621906 11.218094
[1] 15.49949 23.04451