

Nombre: Luis Alejandro Baena  
Código: 1023628877

Nombre: Jesús David Cantellon Espitia  
Código: 1013457170



## Estadística II

### Taller I

---

## Resumen

*Este trabajo aborda dos aspectos fundamentales en el análisis estadístico multivariado: las pruebas para detectar normalidad multivariada y la detección de outliers en este contexto. En la primera sección, se exploran pruebas clásicas como la prueba de Shapiro-Wilk generalizada y el test de Mardia, centrándose en su aplicación y utilidad. Además, se mencionan aspectos prácticos como el código para implementar estas pruebas. En la segunda sección, se profundiza en la detección de outliers mediante el uso del diagrama de caja (boxplot), detallando sus partes y la interpretación de su distribución y se aborda el enfoque clásico y por DCM del método de Mahalanobis para detección de outliers, un método analítico muy ampliamente utilizado en la actualidad. El trabajo busca proporcionar una comprensión sólida de estos conceptos fundamentales y su aplicación práctica en el análisis de datos multivariados.*

## Índice

1	Pruebas para detectar normalidad multivariada . . . . .	2
1.1	Prueba de Shapiro Wilk Generalizada . . . . .	2
1.2	Test de Mardia . . . . .	2
1.3	Aplicación . . . . .	4
1.3.1	Código . . . . .	5
2	Detección de outliers en el contexto multivariado . . . . .	6
2.1	Diagrama de Caja (Boxplot) . . . . .	6
2.1.1	Partes de un boxplot . . . . .	6
2.1.2	Distribución del diagrama de caja . . . . .	7
2.1.3	Aplicación . . . . .	8
2.2	Detección de outliers multivariados por la distancia de Mahalanobis . . . . .	10
2.2.1	Distancia de Mahalanobis . . . . .	10
2.2.2	Método de Mahalanobis por estimación clásica de la matriz de covarianzas . . . . .	11
2.2.3	Método de Mahalanobis por estimación DCM de la matriz de covarianzas . . . . .	11
2.2.3.1	Estimador por Determinante de Covarianza Mínima (DCM) . . . . .	11
2.2.3.2	FAST-MCD . . . . .	12
2.2.3.3	Algoritmo FAST-MCD en R . . . . .	13
2.2.4	Ejemplos de Aplicación . . . . .	15
2.2.4.4	Ejemplo con $p = 2$ . . . . .	15
2.2.4.5	Efecto de Enmascaramiento . . . . .	22
2.2.4.6	Ejemplo con $p > 2$ . . . . .	23

# 1. Pruebas para detectar normalidad multivariada

En esta sección dedicada a las pruebas para detectar normalidad multivariada, se exploran diversas técnicas utilizadas para evaluar si un conjunto de datos multivariado sigue una distribución normal. La normalidad es un supuesto fundamental en muchos análisis estadísticos y modelos predictivos, por lo que su verificación es crucial para garantizar la validez de los resultados obtenidos. Se tienen métodos específicos para testear la normalidad multivariada, como la prueba de Mardia y la prueba de Shapiro Wilk Generalizada. Un entendimiento sólido de estas pruebas es fundamental para la correcta aplicación de técnicas estadísticas y el análisis adecuado de datos multivariados en una gran variedad de campos. Se tomará como nivel de significancia  $\alpha = 0,05$

## 1.1. Prueba de Shapiro Wilk Generalizada

Si  $X_1, \dots, X_n$  son vectores aleatorios idénticamente distribuidos en  $R^p$ ,  $p \geq 1$ . Si  $N^P(\mu, \Sigma)$  denota la densidad de una normal p-variada con vector de media  $\mu$  y matriz de covarianza  $\Sigma$ .

Si la hipótesis nula  $H_0 : X_1, \dots, X_n$  es una muestra de  $N^P(\mu, \Sigma)$  donde  $\mu$  y  $\Sigma$  son desconocidos, se propone la siguiente prueba estadística:

$$W^* = \frac{1}{p} \sum_{i=1}^p W_{z_i}$$

Donde  $W_{z_i}$  es el estadístico Shapiro-Wilk evaluado en la i-ésima coordenada de la observación transformada  $Z_{i1}, \dots, Z_{in}; i = 1, \dots, p$ . La prueba basada en  $W^*$  rechaza  $H_0$  en una prueba de tamaño  $\alpha$  si  $W^* < c_{a;n,p}$  donde  $c_{a;n,p}$  satisface la ecuación:

$$\alpha = P\{W^* < c_{a;n,p} / H_0 \text{ es verdadero}\}$$

## 1.2. Test de Mardia

Otro procedimiento para evaluar la normalidad multivariada es una generalización de la prueba univariada basada en las medidas de asimetría y curtosis. La prueba se debe a Mardia (1970). Sean  $\mathbf{y}$  y  $\mathbf{x}$  independientes y distribuida idénticamente con vector medio  $\mu$  y matriz de covarianza  $\Sigma$ . Entonces asimetría y la curtosis para poblaciones multivariadas son definidas por Mardia como:

$$\begin{aligned}\beta_{1,p} &= E[(\mathbf{y} - \mu)' \Sigma^{-1} (\mathbf{x} - \mu)]^3 \\ \beta_{2,p} &= E[(\mathbf{y} - \mu)' \Sigma^{-1} (\mathbf{x} - \mu)]^2\end{aligned}$$

Dado que los momentos centrales de tercer orden para la distribución normal multivariada son cero,  $\beta_{1,p} = 0$  cuando  $\mathbf{y}$  es  $N_p(\mu, \Sigma)$ . También se puede demostrar que si  $\mathbf{y}$  es  $N_p(\mu, \Sigma)$ , entonces:

$$\beta_{2,p} = p(p+2)$$

Para estimar  $\beta_{1,p}$  y  $\beta_{2,p}$  usando una muestra  $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_p$ , primero definimos:

$$g_{ij} = (\mathbf{y}_i - \bar{\mathbf{y}})' \hat{\Sigma}^{-1} (\mathbf{y}_j - \bar{\mathbf{y}})$$

donde  $\hat{\Sigma} = \sum_{i=1}^n \frac{(\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})'}{n}$  es el estimador de máxima verosimilitud. Entonces las estimaciones de  $\beta_{1,p}$  y  $\beta_{2,p}$  vienen dadas por:

$$b_{1,p} = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n g_{ij}^3$$

$$b_{2,p} = \frac{1}{n} \sum_{i=1}^n g_{ii}^2$$

Cuando  $n \geq 50$ , se encuentran disponibles las siguientes pruebas aproximadas. Para  $b_{1,p}$ , el estadístico:

$$z_1 = \frac{(p+1)(n+1)(n+3)}{6[(n+1)(p+1)-6]} b_{1,p} \quad (1)$$

es aproximadamente  $\chi^2$  (chi-cuadrado) con  $\frac{1}{6}p(p+1)(p+2)$  grados de libertad. Rechazamos la hipótesis de normalidad multivariada si  $z_1 \geq \chi_{0,05}^2$ . Con  $b_{2,p}$ , por otro lado, deseamos rechazar valores grandes (distribución demasiado puntiaguda) o valores pequeños (distribución demasiado plana).

Para los **2,5 %** puntos superiores de  $b_{2,p}$  utilice:

$$z_2 = \frac{b_{2,p} - p(p+2)}{\sqrt{8p(p+2)/n}} \quad (2)$$

que es aproximadamente  $N(0,1)$ . Para los **2,5 %** puntos inferiores tenemos dos casos:

1. cuando  $50 \leq n \leq 400$ , utilice:

$$z_2 = \frac{b_{2,p} - p(p+2)}{\sqrt{8p(p+2)/n}} \quad (3)$$

2. cuando  $n \geq 400$ , use  $z_2$  como está dado por (2)

En resumen:

$z_1$ ,  $z_2$  y  $z_3$  son estadísticas que se utilizan para evaluar si los valores calculados de  $b_{1,p}$  y  $b_{2,p}$  son significativamente diferentes de ciertos valores de referencia. Aquí está el significado de cada una:

- $z_1$ : Esta estadística se calcula para  $b_{1,p}$  y se utiliza para probar la hipótesis de que la distribución multivariada es normal. Se compara con un valor crítico de la distribución chi-cuadrado ( $\chi^2$ ) para determinar si se rechaza la normalidad multivariada.
- $z_2$ : Esta estadística se calcula para  $b_{2,p}$  y se utiliza para evaluar si la distribución es demasiado apuntada o demasiado plana. Se compara con valores críticos de la distribución normal estándar ( $N(0,1)$ ) para determinar si se rechaza la hipótesis de normalidad.
- $z_3$ : Esta estadística también se calcula para  $b_{2,p}$  y se utiliza para evaluar si la distribución es demasiado apuntada o demasiado plana, pero se maneja de manera diferente dependiendo del tamaño de la muestra ( $n$ ). Para muestras pequeñas o medianas ( $50 \leq n \leq 400$ ), se calcula de una manera, mientras que para muestras grandes ( $n \geq 400$ ), se utiliza la misma fórmula que  $z_2$ .

### 1.3. Aplicacion

En este caso, utilizamos el test de Mardia y Shapiro Wilk para nuestra aplicación. La hipótesis nula  $H_0$  establece que los datos siguen una distribución normal multivariada, mientras que la hipótesis alternativa  $H_1$  sostiene que los datos no siguen una distribución normal multivariada.

Vamos a usar el conjunto de datos iris que es un conjunto de datos integrado en R, que contiene mediciones de varias características de flores de iris.

**Formato:** Un marco de datos con 50 filas y 4 columnas. Las columnas son las siguientes:

- `Sepal.Length`: Valores de longitud del sépalo de las flores de iris.
- `Sepal.Width`: Valores de ancho del sépalo de las flores de iris.
- `Petal.Length`: Valores de longitud del pétalo de las flores de iris.
- `Petal.Width`: Valores de ancho del pétalo de las flores de iris.

$H_0 =$  Los datos siguen una distribución normal multivariada

$H_1 =$  Los datos NO siguen una distribución normal multivariada

```
> source("c:\\Users\\alejo\\OneDrive - Universidad EAFIT\\EAFIT\\Semestre5\\Es$  
  
Shapiro-Wilk normality test  
  
data:  Z  
W = 0.95878, p-value = 0.07906  
  
      Test      Statistic      p value Result  
1 Mardia Skewness 25.6643445196298 0.177185884467652 YES  
2 Mardia Kurtosis 1.29499223711605 0.195322907441935 YES  
3      MVN      <NA>      <NA>      YES
```

Figura 1: Resultados de los tests

Dado que el valor p obtenido del test de Shapiro Wilk (0.07) es mayor que el nivel de significancia establecido (0.05), no tenemos evidencia suficiente para rechazar la hipótesis nula. Además con el test Mardia se comprueba esto. En otras palabras, no podemos concluir que los datos no siguen una distribución normal multivariante a un nivel de confianza del 95 %.

### 1.3.1. Código

```
# Cargar bibliotecas necesarias para llevar a cabo las pruebas de normalidad multivariante
library(MVN) # Proporciona funciones para el análisis de normalidad multivariante
library(mvnormtest) # Proporciona diferentes pruebas de normalidad multivariante

# Cargar el conjunto de datos 'iris' incorporado en R
data(iris)

# Seleccionar las primeras 50 filas y las primeras 4 columnas del conjunto de datos 'iris',
# que corresponden a la especie 'setosa'
setosa <- iris[1:50, 1:4]

# Realizar la prueba de Shapiro-Wilk multivariante utilizando la función mshapiro.test()
# Se transpone el conjunto de datos 'setosa' utilizando la función t() para que los datos
# estén en el formato adecuado para la prueba
result1 <- mshapiro.test(t(setosa))

# Imprimir los resultados de la prueba de Shapiro-Wilk multivariante
print(result1)

# Realizar prueba de normalidad multivariante utilizando el método de Mardia
result2 <- mvn(setosa, mvnTest = "mardia")

# Imprimir los resultados de la prueba de normalidad multivariante
print(result2$multivariateNormality)
```

## Conclusiones

En resumen, en esta sección hemos explorado diversas técnicas para evaluar la normalidad multivariada, fundamental en muchos análisis estadísticos. Pruebas específicas como la de Mardia y la de Shapiro-Wilk Generalizada son cruciales para garantizar la validez de los resultados. Se estableció un nivel de significancia de  $\alpha = 0,05$  para evaluar la evidencia en contra de la normalidad multivariada en nuestros datos. Estas pruebas son esenciales para mejorar la confiabilidad y la interpretación de nuestros análisis estadísticos y modelos.

## 2. Detección de outliers en el contexto multivariado

El proceso de detección de outliers en el contexto multivariado implica identificar observaciones atípicas que se desvían significativamente de la estructura general de un conjunto de datos que contiene múltiples variables. Este enfoque es crucial para diversas aplicaciones en campos como la estadística, la minería de datos y el aprendizaje automático, donde la presencia de outliers puede distorsionar análisis posteriores y afectar la calidad de los modelos. En esta sección, se exploran diferentes técnicas y métodos utilizados para la detección de outliers multivariados, que van desde el boxplot para el caso univariado hasta la estimación clásica de la matriz de covarianza y la estimación usando el método del determinante de covarianza mínima.

### 2.1. Diagrama de Caja (Boxplot)

Cuando mostramos la distribución de datos de forma estandarizada utilizando 5 resúmenes: mínimo, Q1 (primer cuartil o cuartil inferior), mediana, Q3 (tercer cuartil o cuartil superior) y máximo, se denomina diagrama de caja (Boxplot). También se le denomina diagrama de caja y bigotes. Veamos a detalle qué es un Boxplot, sus aplicaciones y cómo dibujar diagramas de caja.

Utilizamos este tipo de gráfico para saber:

- Forma de distribución
- Valor central de la misma
- variabilidad de la misma

Un diagrama de caja es un gráfico que muestra datos de un resumen de cinco números que incluye una de las medidas de tendencia central. No muestra la distribución en particular tanto como lo hace un histograma. Pero se utiliza principalmente para indicar que una distribución está sesgada o no y si existen posibles observaciones inusuales (también llamados outliers) presentes en el conjunto de datos.

En palabras simples, podemos definir el diagrama de caja en términos de conceptos relacionados con la estadística descriptiva. Eso significa que el diagrama de caja o bigotes es un método utilizado para representar gráficamente grupos de datos numéricos a través de sus cuartiles. Estos también pueden tener algunas líneas que se extienden desde las cajas o bigotes, lo que indica la variabilidad fuera de los cuartiles inferior y superior, de ahí los términos diagrama de caja y bigotes y diagrama de caja y bigotes. Los valores atípicos se pueden indicar como puntos individuales.

#### 2.1.1. Partes de un boxplot

- **Minimum:** El valor mínimo en el conjunto de datos dado
- **Primer cuartil (Q1):** el primer cuartil es la mediana de la mitad inferior del conjunto de datos.
- **Mediana:** la mediana es el valor medio del conjunto de datos, que divide el conjunto de datos dado en dos partes iguales. La mediana se considera el segundo cuartil.
- **Tercer cuartil (Q3):** el tercer cuartil es la mediana de la mitad superior de los datos.
- **Máximo:** el valor máximo en el conjunto de datos dado

Además de estos cinco términos, los otros términos utilizados en el diagrama de caja son:

## El diagrama de caja y bigotes

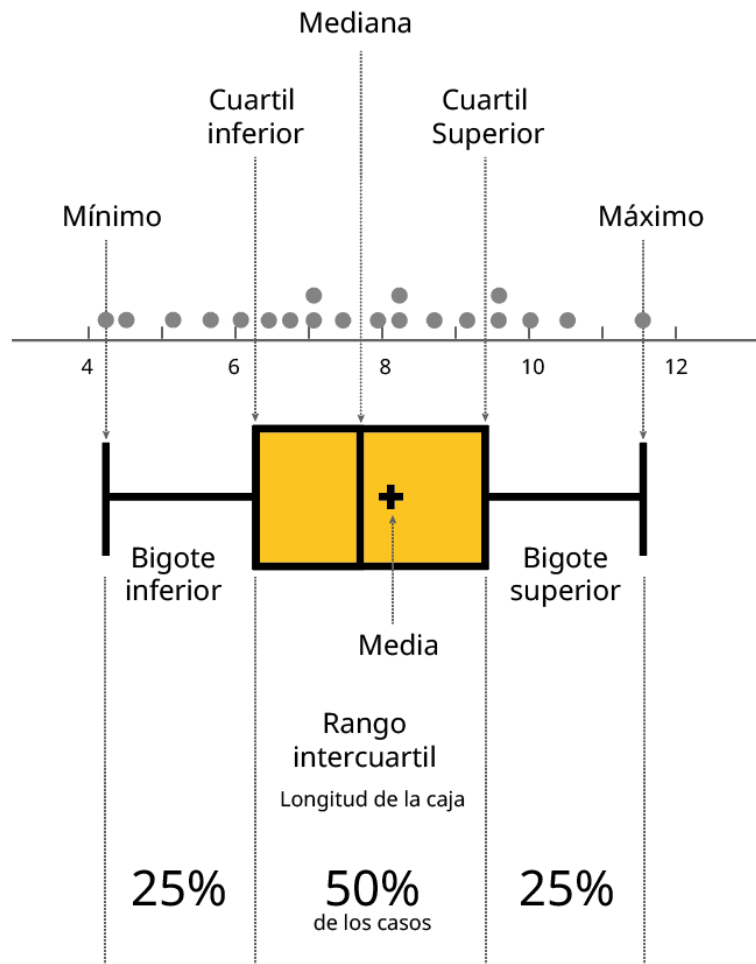


Figura 2: Partes de un Diagrama de Cajas y Bigotes

- **Rango intercuartil (IQR):** la diferencia entre el tercer cuartil y el primer cuartil se conoce como rango intercuartil. (es decir)  $IQR = Q3 - Q1$
- **Valor atípico (outlier):** los datos que se encuentran en el extremo izquierdo o derecho de los datos ordenados se prueban como valores atípicos. Los valores atípicos son mayores que  $Q3 + (1,5 \times IQR)$  o menores que  $Q1 - (1,5 \times IQR)$ .
- **Mediana:** la mediana es el valor medio del conjunto de datos, que divide el conjunto de datos dado en dos partes iguales. La mediana se considera el segundo cuartil.

### 2.1.2. Distribución del diagrama de caja

La distribución del diagrama de caja explicará qué tan estrechamente están agrupados los datos, cómo están sesgados y también la simetría de los datos.

- **Sesgado positivo:** si la distancia desde la mediana al máximo es mayor que la distancia desde la mediana al mínimo, entonces el diagrama de caja está sesgado positivamente.

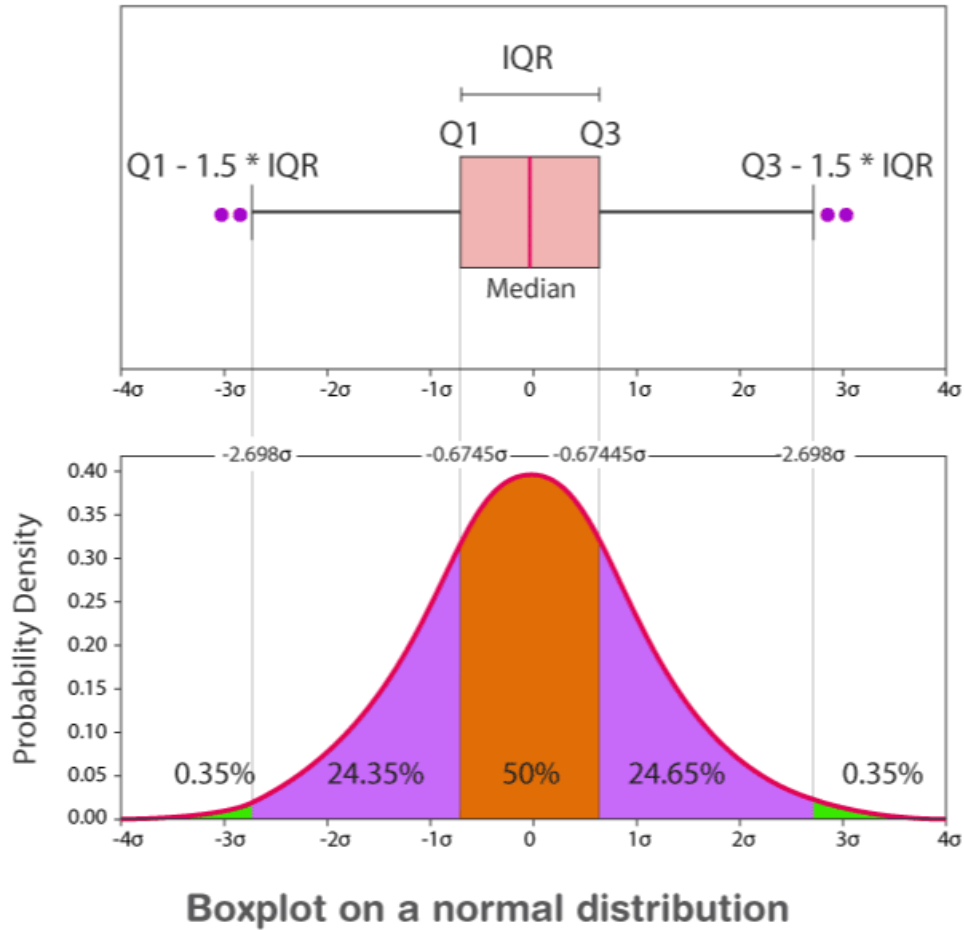


Figura 3: Boxplot de una distribución normal

- **Sesgado negativo:** si la distancia desde la mediana al mínimo es mayor que la distancia desde la mediana al máximo, entonces el diagrama de caja está sesgado negativamente.
- **Simétrico:** Se dice que el diagrama de caja es simétrico si la mediana es equidistante de los valores máximo y mínimo.

### 2.1.3. Aplicación

Para la aplicación del boxplot en la detección de outliers, se utilizará el dataset integrado de R "nh-temp", el cual contiene datos sobre la temperatura media anual en grados Fahrenheit en New Haven, Connecticut, desde 1912 hasta 1971.

Este conjunto de datos proporciona una oportunidad adecuada para ilustrar el uso del boxplot en la identificación de valores atípicos dentro de una serie temporal, permitiendo así una comprensión más profunda de la variabilidad y los patrones presentes en los datos climáticos a lo largo de un extenso período de tiempo. El análisis de outliers en este contexto puede revelar eventos climáticos extremos o anomalías significativas que podrían tener implicaciones importantes en la comprensión del cambio climático y la toma de decisiones en materia de gestión ambiental.



Boxplot de temperatura anual en grados Fahrenheit en New Haven, Connecticut, from 1912 to 1971.

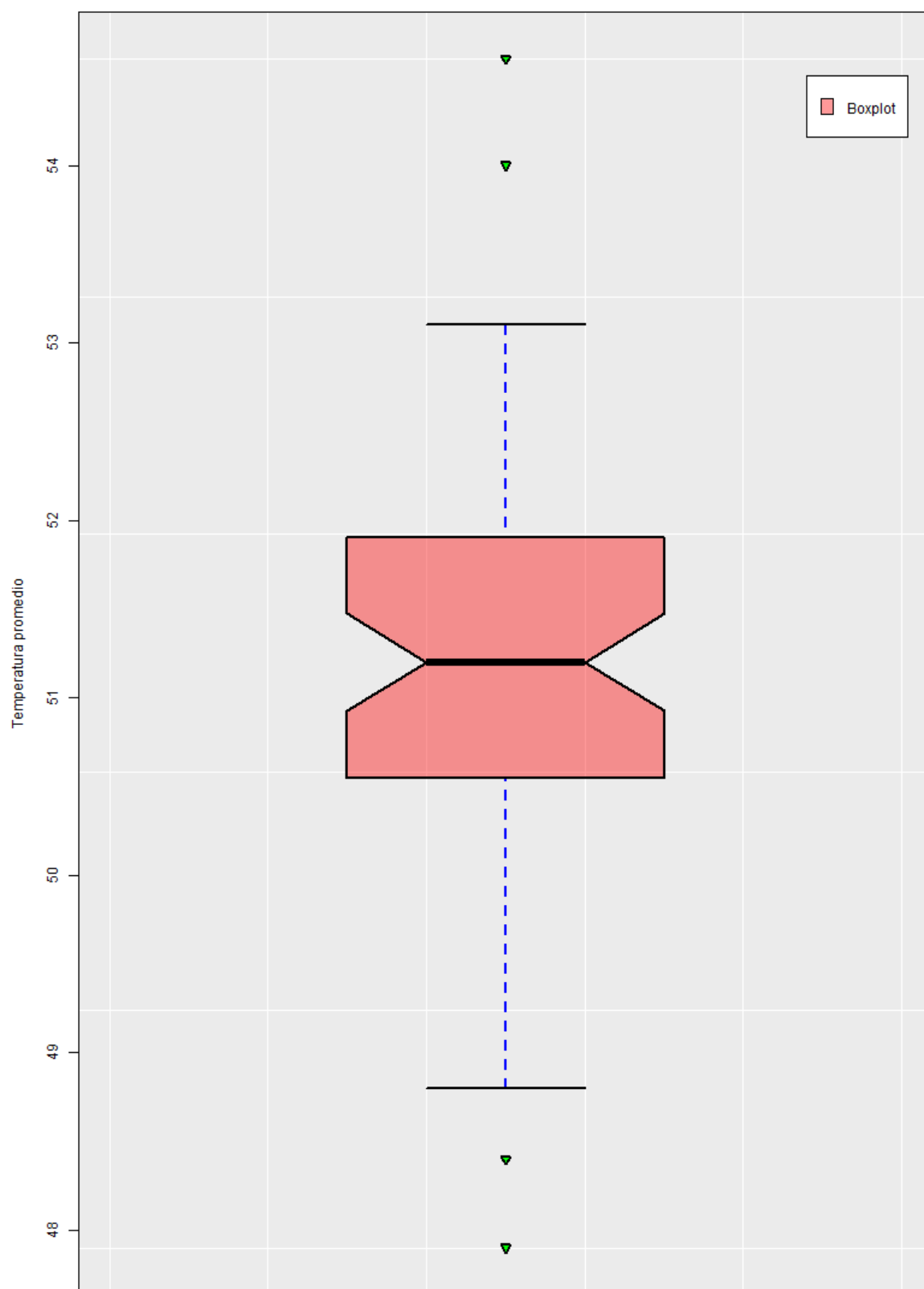


Figura 4: Boxplot del dataset "nhtemp"

## Código

```
data(nhtemp)

plot.new()

rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4],
     col = "#ebebeb")

# Añadimos un grid blanco
grid(nx = NULL, ny = NULL, col = "white", lty = 1,
     lwd = par("lwd"), equiloggs = TRUE)

# Boxplot
par(new = TRUE)
boxplot(nhtemp, # Datos
        horizontal = FALSE, # Horizontal o vertical
        lwd = 2, # Lines width
        col = rgb(1, 0, 0, alpha = 0.4), # Color
        ylab = "Temperatura promedio", # Etiqueta eje Y
        main = "Boxplot de temperatura anual en grados Fahrenheit en New Haven, Connecticut",
        notch = TRUE, # Añade intervalos de confianza para la mediana
        border = "black", # Color del borde del boxplot
        outpch = 25, # Símbolo para los outliers
        outbg = "green", # Color de los datos atípicos
        whiskcol = "blue", # Color de los bigotes
        whisklty = 2, # Tipo de línea para los bigotes
        lty = 1) # Tipo de línea (caja y mediana)

# Agregamos una leyenda
legend("topright", legend = "Boxplot", # Posición y título
      fill = rgb(1, 0, 0, alpha = 0.4), # Color
      inset = c(0.03, 0.05), # Cambiamos los márgenes
      bg = "white") # Color de fondo de la leyenda
```

## 2.2. Detección de outliers multivariados por la distancia de Mahalanobis

### 2.2.1. Distancia de Mahalanobis

Desde un punto de vista geométrico, la distancia Ecludiana entre dos puntos es la mas corta posible entre ellos. Un problema con la medida de la distancia Euclidiana es que no toma en cuenta la correlación entre variables altamente correlacionadas. En este caso, la distancia Euclidiana asigna ponderaciones iguales a tales variables, y ya que esas variables miden esencialmente la misma característica, entonces esta característica obtiene un peso adicional. En efecto, variables correlacionadas obtienen exceso de pesos en la distancia Euclidean.

Un enfoque alternativo es escalar la contribución de las variables individuales al valor de la distancia según la variabilidad de cada variable. Este enfoque es considerado por la distancia de Mahalanobis, que ha sido desarrollada como una medida estadística por el estadístico indio Prasanta Chandra Mahalanobis. La distancia de Mahalanobis tiene grandes aplicaciones en el campo de la estadística multivariada. Difiere de

la distancia Euclidiana en que tiene en cuenta las correlaciones entre variables. Es una métrica invariante a la escala y provee una medida de distancia entre un punto  $\mathbf{x} \in \mathbf{R}^p$  generado de una distribución de probabilidad  $p$ -variada  $f_{\mathbf{x}}(\cdot)$  y la media  $\boldsymbol{\mu} = \mathbf{E}(\mathbf{X})$  de la distribución. Asuma que  $f_{\mathbf{x}}(\cdot)$  tiene momentos de segundo orden finitos y denotamos  $\boldsymbol{\Sigma} = \mathbf{E}(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T$  como la matriz de covarianzas. Entonces, la distancia de Mahalanobis es

$$D(\mathbf{X}, \boldsymbol{\mu}) = \sqrt{(\mathbf{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{X} - \boldsymbol{\mu})}.$$

Si la matriz de covarianzas es la matriz identidad, la distancia de Mahalanobis se reduce a la distancia Euclidea.

### 2.2.2. Método de Mahalanobis por estimación clásica de la matriz de covarianzas

Supongamos que  $\mathbf{X}$  es un vector  $p$ -dimensional que tiene distribución normal multivariada, es decir,  $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ . La distancia de Mahalanobis al cuadrado  $D^2(\mathbf{X}, \boldsymbol{\mu})$  es distribuida como una variable aleatoria  $\chi^2$  con  $p$  grados de libertad. Este método usa las estimaciones de la distancia de Mahalanobis, al introducir la media muestral multivariada  $\bar{\mathbf{X}}$  y las estimaciones de la matriz de covarianza  $\mathbf{S}$  para la media desconocida  $\boldsymbol{\mu}$  y la matriz de covarianza  $\boldsymbol{\Sigma}$ , y etiquetando como valor atípico cualquier observación que tenga una distancia al cuadrado de Mahalanobis  $d^2(\mathbf{X}, \bar{\mathbf{X}})$  por encima de una cuartil predefinido de la distribución  $\chi^2$  con  $p$  grados de libertad.

### 2.2.3. Método de Mahalanobis por estimación DCM de la matriz de covarianzas

El método anterior es un poco problemático porque todo depende de la suposición de normalidad (aunque este problema podremos ignorarlo generalmente si tenemos suficiente evidencia que prueba nuestra hipótesis de normalidad) y las estimaciones de los parámetros son particularmente sensibles a los valores atípicos. Por lo tanto, es importante considerar alternativas robustas a estos estimadores para calcular distancias de Mahalanobis robustas. El estimador mas ampliamente usado de este tipo es el estimador por Determinante de Covarianza Minima (que en ahora en adelante lo denotaremos con las cifras DCM) del cual hablaremos a continuación.

#### 2.2.3.1 Estimador por Determinante de Covarianza Minima (DCM)

El objetivo del DCM, como su nombre lo indica, es encontrar  $h$  observaciones (de las  $n$  observaciones totales), donde  $n/2 \leq h < n$ , cuya matrix de covarianzas clasica tenga el determinante mas pequeño. La estimación DCM de la ubicación es entonces el promedio de estos  $h$  puntos, y la estimación DCM de la dispersión es su matriz estimada clasica de covarianza. Su valor de *break-down* es  $(n - h)/n$ , es decir, este es el porcentaje máximo de datos atípicos que un estimador puede manejar antes de proporcionar resultados no confiables; osea que, cuanto mayor sea el valor de *break-down*, más robusto es el estimador ante datos atípicos

Hay diversas maneras de calcular este estimador: La forma mas natural es tomar todos los subconjuntos posibles de tamaño  $h$  de la muestra de  $n$  elementos, calcular el determinante de la matrix de covarianzas clasica de cada subconjunto, tomar el que genere el determinante mas pequeño y luego usar ese subconjunto para obtener las estimaciones de la esperanza multivariada y la matriz de covarianzas. Aunque este algoritmo sea sencillo, es computacionalmente costoso debido a que tiene que calcular  $\frac{n!}{h!(n-h)!}$  determinantes de matrices  $p \times p$  y despues hallar el mínimo de estos resultados, pensemos en  $n = 1000$ ,  $h = 500$  y  $p = 3$  que resulta en número de operaciones de un orden de  $2,7 \times 10^{299}$ , que en el peor de los casos le tomaría aproximadamente  $9 \times 10^{290}$  años en terminar.

Rousseeuw y Driessen desarrollaron un rapido algoritmo para el estimador DCM que lo llamaron *FAST-MCD*. Este algoritmo puede tratar con  $n = 50000$ ,  $h = 25015$  y  $p = 30$  en un tiempo de 15 minutos, es bastante rapido. A continuación indagaremos en él:

### 2.2.3.2 FAST-MCD

1. El  $h$  por defecto es  $[(n + p + 1)/2]$ , donde  $[x]$  indica que tomamos la parte entera de  $x$ . Pero podemos escoger cualquier entero  $h$  con  $[(n + p + 1)/2] \leq h \leq n$ . El programa entonces reporta del valor de *break-down* del estimador DCM como  $(n - h + 1)/n$ . Si se está seguro que el conjunto de datos contiene menos del 25 % de contaminación, que usualmente es el caso, un buen compromiso entre valor de *break-down* y eficiencia estadística se obtiene estableciendo  $h = [0,75n]$ .
2. Si  $h = n$ , entonces la estimación de la tendencia central del DCM  $T$  es el promedio of conjunto de datos entero, y la estimación de la dispersión del DCM  $S$  es su matriz de covarianza. Informamos de esto y paramos.
3. Si  $p = 1$  (datos univariados), calculamos las estimaciones del DCM  $(T, S)$  por el algoritmo exacto de Rousseeuw and Leroy en tiempo  $O(n \log n)$ , luego paramos.
4. De aquí en adelante,  $h < n$  y  $p \geq 2$ . Si  $n$  es pequeña (por ejemplo,  $n \leq 600$ ), entonces
  - Repetir (decir) 500 veces:
    - Construir un subconjunto  $H_1$  de tamaño  $h$  siguiendo el siguiente procedimiento: Extrae un subconjunto  $J$  de tamaño  $p + 1$ , y luego calcula  $T_0 := \text{promedio}(J)$  y  $S_0 := \text{covarianza}(J)$ . [Si  $\det(S_0) = 0$ , entonces extiende  $J$  añadiendo otra observación aleatoria, y continua añadiendo observaciones hasta que  $\det(S_0) > 0$ .] Luego calculamos las distancias  $d_0^2(i) := (X_i - T_0)^T S_0^{-1} (X_i - T_0)$  para  $i = 1, \dots, n$ . Ordenarlos en  $d_0(\pi(1)) \leq \dots \leq d_0(\pi(n))$  y colocarlos en  $H_1 := \{\pi(1), \dots, \pi(h)\}$ ; es decir, después de calcular las distancias entre las observaciones y el vector medio ponderado, se ordenan en secuencia ascendente y se almacenan en el conjunto  $H_1$ .
    - Denotamos el siguiente procedimiento como *C-step*: Dado un subconjunto  $H_{old}$  de tamaño  $h$  o el par  $(T_{old}, S_{old})$ , hacemos: Calcular las distancias  $d_{old}(i)$  para  $i = 1, \dots, n$ . Ordenar estas distancias, con lo que se obtiene una permutación  $\pi$  para la que  $d_{old}(\pi(1)) \leq d_{old}(\pi(2)) \leq \dots \leq d_{old}(\pi(n))$ . Poner  $H_{new} := \{\pi(1), \pi(2), \dots, \pi(h)\}$ . Calcular  $T_{new} := \text{promedio}(H_{new})$  y  $S_{new} := \text{covarianza}(H_{new})$ . Realizamos dos *C-step*.
  - Para los 10 resultados con el mas bajo  $\det(S_3)$ :
    - Realizar *C-step* hasta la convergencia.
  - Reportar la solución  $(T, S)$  con el mas bajo  $\det(S)$ .
5. Si  $n$  es mas grande (por ejemplo,  $n > 600$ ), entonces
  - Construir cinco subconjuntos aleatorios disjuntos de tamaño  $n_{sub}$  (por ejemplo, cinco subconjuntos de tamaño  $n_{sub} = 300$ );
  - Dentro de cada subconjunto, repetir  $500/5 = 100$  veces:
    - Construir un subconjunto inicial  $H_1$  de tamaño  $h_{sub} = [n_{sub}(h/n)]$ ;
    - Realizar dos *C-step*, usando  $n_{sub}$  y  $h_{sub}$ ;
    - Mantener los 10 mejores resultados  $(T_{sub}, S_{sub})$ ;
  - Agrupar los subconjuntos, obteniendo el conjunto fusionado (por ejemplo, de tamaño  $n_{merged} = 1500$ );

- En el conjunto fusionado, repetir para cada una de las 50 soluciones  $(T_{sub}, S_{sub})$ :
    - Realizar dos *C-Step*, usando  $n_{merged}$  y  $h_{merged} = [n_{merged}(h/n)]$ ;
    - Mantener los 10 mejores resultados  $(T_{merged}, S_{merged})$ ;
  - En el conjunto de datos completo, repetir para los  $m_{full}$  mejores resultados:
    - Dar muchos *C-step*, usando  $n$  y  $h$ ;
    - Mantener el mejor resultado final  $(T_{full}, S_{full})$ . (Aquí,  $m_{full}$  y el número de *C-step*, preferiblemente hasta la convergencia, depende de que tan grande sea el conjunto de datos).
6. Para obtener consistencia cuando los datos provienen de una distribución normal multivariada, nosotros designamos  $T_{DCM} = T_{full}$  y

$$S_{DCM} = \frac{med_i d_{(T_{full}, S_{full})}^2(i)}{\chi_{p,0,5}^2} S_{full}.$$

7. Una estimación reponderada de un paso se obtiene mediante

$$T_1 = \frac{\sum_{i=1}^n w_i X_i}{\sum_{i=1}^n w_i}$$

$$S_1 = \frac{\sum_{i=1}^n w_i (X_i - T_1)(X_i - T_1)^T}{\sum_{i=1}^n w_i - 1},$$

donde  $w_i = 1$  si  $d_{(T_{DCM}, S_{DCM})}(i) \leq \sqrt{\chi_{p,0,975}^2}$  y  $w_i = 0$  en otro caso.

### 2.2.3.3 Algoritmo FAST-MCD en R

La librería `robustbase` de *R* provee una función llamada `covMcd` que utiliza el algoritmo *FAST-MCD* ya anteriormente visto. Esta función contiene los siguiente parámetros por defecto: `covMcd(x, cor = FALSE, raw.only = FALSE, alpha = , nsamp = , nmini = , kmini = , scalefn = , maxcsteps = , initHsets = NULL, save.hsets = FALSE, names = TRUE, seed = , tolSolve = , trace = , use.correction = , wgtFUN = , control = rrcov.control())`. A continuación se detallará cada uno:

- **x**: Matriz o marco de datos.
- **cor**: Booleano indicando si se debe incluir una matriz de correlación en el resultado.
- **raw.only**: Booleano indicando si solo se debe devolver la estimación “cruda”, es decir, sin realizar un paso de ponderación.
- **alpha**: Parámetro numérico que controla el tamaño de los subconjuntos sobre los cuales se minimiza el determinante. Los valores permitidos están entre **0,5** y **1**, con un valor predeterminado de **0,5**.
- **nsamp**: Número de subconjuntos utilizados para estimaciones iniciales o “best”, “exact” o “deterministic”. El valor predeterminado es `nsamp = 500`. Para `nsamp = “best”`, se realiza una enumeración exhaustiva, siempre y cuando el número de pruebas no exceda los **100,000** (`= nLarge`). Para “exact”, se intentará una enumeración exhaustiva sin importar cuántas muestras se necesiten. En este caso, puede aparecer un mensaje de advertencia indicando que el cálculo puede llevar mucho tiempo. Para “deterministic”, se calcula el *MCD determinista*; como propuesto por Hubert et al. (2012), comienza desde las  $h$  observaciones más centrales de seis estimadores (deterministas). El “MCD determinista” es una versión del método MCD que se caracteriza por su determinismo en la selección de los subconjuntos de datos.

- `nmini`, `kmini`: Parámetros que controlan la división de los datos en subconjuntos iniciales.
- `scalefn`: Función para calcular una estimación robusta de escala en el MCD determinístico.
- `maxcsteps`: Número máximo de pasos de concentración en el MCD determinístico.
- `initHsets`: Matriz de subconjuntos iniciales de observaciones.
- `save.hsets`: Booleano indicando si se deben devolver los subconjuntos iniciales.
- `names`: Booleano indicando si se deben incluir nombres en los resultados.
- `seed`: Semilla inicial para el generador de números aleatorios.
- `tolSolve`: Tolerancia numérica para la inversión de la matriz de covarianza.
- `trace`: Booleano indicando si se deben imprimir resultados intermedios.
- `use.correction`: Booleano indicando si se deben usar factores de corrección para muestras finitas.
- `wgtFUN`: Cadena de caracteres o función para calcular los pesos en el paso de ponderación.
- `control`: Lista con opciones de estimación, que puede incluir los parámetros anteriores.

La función `covMcd` retorna un objeto de la clase `mcd` que es básicamente una lista con componentes:

- `center`: Estimación final de la ubicación.
- `cov`: Estimación final de la dispersión.
- `cor`: Estimación final de la matriz de correlación (solo si `cor = TRUE`).
- `crit`: Valor del criterio, es decir, el logaritmo del determinante.
- `best`: El mejor subconjunto encontrado y utilizado para calcular las estimaciones crudas.
- `mah`: Distancias de Mahalanobis de las observaciones utilizando la estimación final de la ubicación y dispersión.
- `mcd.wt`: Pesos de las observaciones utilizando la estimación final de la ubicación y dispersión.
- `cnp2`: Un vector de longitud dos que contiene el factor de corrección de consistencia y el factor de corrección de muestra finita de la estimación final de la matriz de covarianza.
- `raw.center`: Estimación bruta (sin ponderar) de la ubicación.
- `raw.cov`: Estimación bruta (sin ponderar) de la dispersión.
- `raw.mah`: Distancias de Mahalanobis de las observaciones basadas en la estimación bruta de la ubicación y dispersión.
- `raw.weights`: Pesos de las observaciones basados en la estimación bruta de la ubicación y dispersión.
- `raw.cnp2`: Un vector de longitud dos que contiene el factor de corrección de consistencia y el factor de corrección de muestra finita de la estimación bruta de la matriz de covarianza.
- `X`: Los datos de entrada como una matriz numérica, sin NA.
- `n.obs`: Número total de observaciones.

- **alpha**: El tamaño de los subconjuntos sobre los cuales se minimiza el determinante.
- **quan**: El número de observaciones sobre las cuales se basa el MCD.
- **method**: Cadena de caracteres que nombra el método ('Minimum Covariance Determinant'), comenzando con 'Deterministic' cuando **nsamp**='deterministic'.
- **iBest**: Índices del 1 al 6 que denotan cuál de los (seis) subconjuntos iniciales lleva al mejor conjunto encontrado.
- **n.csteps**: Para cada uno de los subconjuntos iniciales, el número de pasos de concentración ejecutados hasta la convergencia.
- **call**: La llamada utilizada.

## 2.2.4. Ejemplos de Aplicación

### 2.2.4.4 Ejemplo con $p = 2$

Para este ejemplo usaremos la base de datos `weight-height.csv` que contiene **10000** observaciones o registros que contienen 3 variables: *Gender* (Cadena con valores "Male" o "Female", indicando si el individuo es Masculino o Femenino, respectivamente), *Height* (Número real indicando la altura del individuo en pulgadas) y *Weight* (Número real indicando el peso corporal del individuo en libras). Nos interesa las variables *Height* y *Weight*.

Antes de comenzar a detectar outliers, debemos hacer una evaluación de normalidad a los datos ya que el método de detección de outliers por la distancia de Mahalanobis funciona bastante bien preferiblemente para los datos que sigan una distribución normal multivariada:

```
# Importamos la base de datos weight-height.csv en la variable conjunto_datos_Pi2
conjunto_datos_Pi2 <- read.csv("C:/Users/David Espitia/Desktop/weight-height.csv")

# Establecemos semilla aleatoria para reproductividad
set.seed(123)

# Importamos las librerías que contienen las funciones de evaluación de normalidad
library(mvnormtest)
library(MVN)

# Tomamos las columnas de interés 'Height' y 'Weight'
conjunto_datos_Pi2_interes <- conjunto_datos_Pi2[c('Height', 'Weight')]

# Definimos algunas variables de interes
n <- length(conjunto_datos_Pi2_interes[,1])
p <- ncol(conjunto_datos_Pi2_interes)

# Luego, generamos una secuencia de 5000 índices aleatorios de nuestro dataframe
# para extraer un subconjunto de nuestra base de datos. Esto debido a que en R,
# el tamaño maximo de la muestra con la que puede tratar mshapiro.test es 5000.
indices_aleatorios <- sample(1:n, 5000)

# Finalmente, extraemos las filas correspondientes a esos índices
conjunto_datos_test_norm <- conjunto_datos_Pi2_interes[indices_aleatorios, ]
```

```
# Hacemos la prueba al conjunto_datos_test_norm
resultado_normalidad <- mshapiro.test(t(conjunto_datos_test_norm))
print(resultado_normalidad)
```

Obtenemos la siguiente salida de código:

```
shapiro-wilk normality test

data:  Z
W = 0.99959, p-value = 0.4033
```

Figura 5: Resultado de Test de Normalidad de weight-height.csv

Dado que el valor  $p$  obtenido del test de Shapiro Wilk (**0,4033**) es mayor que el nivel de significancia establecido (**0,05**), no tenemos evidencia suficiente a un nivel de confianza del 95 % para rechazar la hipótesis nula de que los datos provienen de una distribución normal multivariada.

Ahora bien, ya que tenemos mucha evidencia de la normalidad de los datos, procedemos a hacer la detección de outliers considerando los 2 métodos: Por estimación clásica de la matriz de covarianzas y por estimación DCM de la misma. Comenzaremos haciendo la detección por estimación clásica:

### Estimación clásica de la matriz de covarianzas y vector de medias

Primero, calculamos la matriz de covarianzas muestral y el vector de medias por el método de estimación clásica:

```
# METODO ESTIMACION CLASICA

# Calcula la matriz de covarianza muestral por el método clasico
matriz_cov_e1_2 <- cov(conjunto_datos_Pi2_interes)

# Calcula el vector de medias por el método clasico
vector_medias_e1_2 <- colMeans(conjunto_datos_Pi2_interes)
```



```

> matriz_cov_e1_2
      Height      weight
Height  14.80347  114.2427
weight 114.24266 1030.9519
> vector_medias_e1_2
      Height      weight
66.36756  161.44036

```

Figura 6: Matriz de covarianzas y vector de media por el método clásico - Ejemplo  $p=2$

Despues usaremos esto para calcular la distancia de Mahalanobis estimada para cada registro en nuestro conjunto de datos para despues etiquetarla como outlier si es mayor al percentil **95** de la distribución  $\chi^2_{p=2}$  (que se justificará inmediatamente a continuación):

```
# Ciclo en los registros
```

```
# Definimos nuestro umbral de desición
```

```
corte1 <- qchisq(0.95,p)
```

```
# Declaramos una lista que contendrá los indices de los registros outliers de nuestro conjunto de
```

```
indices_outliers <- list()
```

```
# Recorremos cada registro para hacerle la evaluación
```

```
for (i in 1:n){
```

```
  d2 <- as.matrix(conjunto_datos_Pi2_interes[i,] - vector_medias_e1_2)%*%as.matrix(solve(matriz_c
```

```
  if (d2 > corte1){
```

```
    indices_outliers <- c(indices_outliers, i)
```

```
  }
```

```
}
```

```
print(indices_outliers)
```

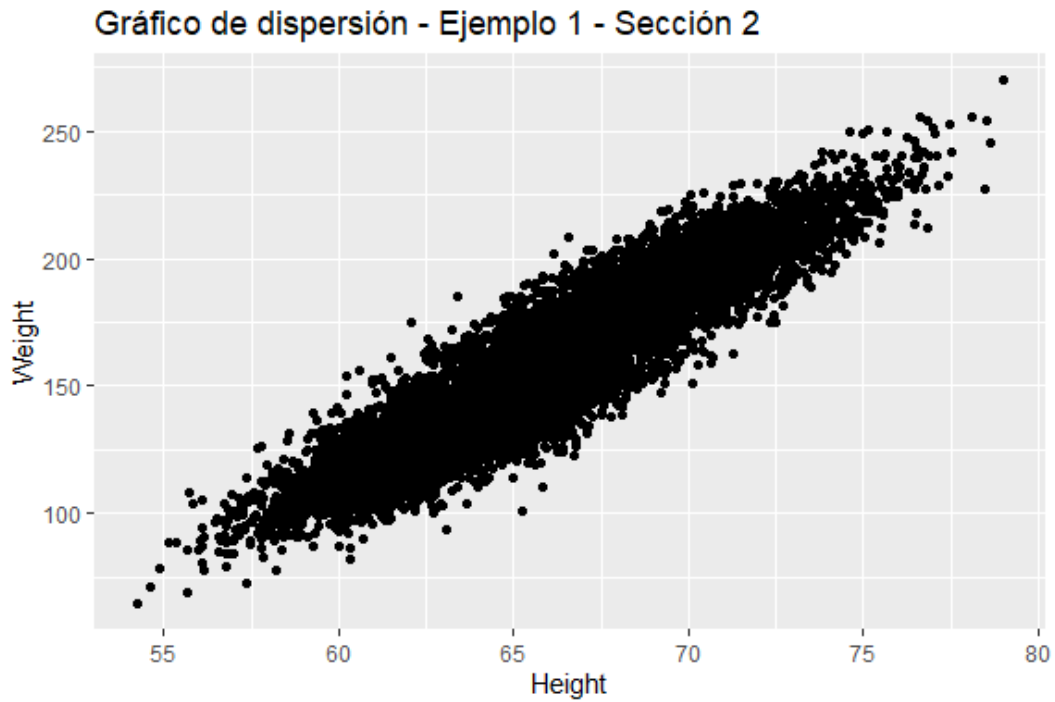


Figura 7: Grafico de dispersión Weight-Height

Observemos en la Figura 7 el gráfico de dispersión de los datos que los mismos tienen un comportamiento bastante compacto y no tiene muchos puntos con posiciones bastantes extrañas, por lo tanto escogimos el percentil 95 debido a que se sospecha que no hay bastantes outliers y queremos ser muy conservadores en la detección, principalmente debido al contexto de los datos (ya que son peso y estatura). A continuación observemos los resultados, una grafica de dispersión con los outliers pintados de rojo y un análisis:

ESTADISTICA 2.R* x		
indices_outliers x		
conjunto_datos_Pi2_interes x		
Show Attributes		
Name	Type	Value
indices_outliers	list [380]	List of length 380
[[1]]	integer [1]	1
[[2]]	integer [1]	18
[[3]]	integer [1]	63
[[4]]	integer [1]	79
[[5]]	integer [1]	83
[[6]]	integer [1]	87
[[7]]	integer [1]	160
[[8]]	integer [1]	161
[[9]]	integer [1]	164
[[10]]	integer [1]	184
[[11]]	integer [1]	191

Figura 8: Registros outliers por método clasico - Ejemplo  $p = 2$

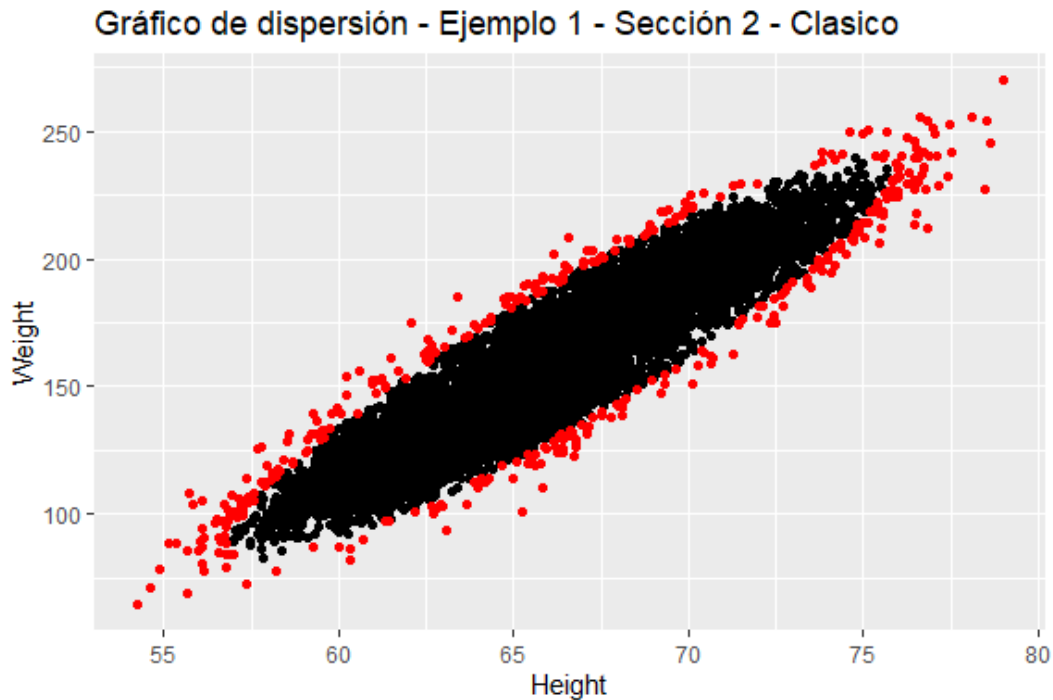


Figura 9: Grafico de dispersión de los datos - Método clasico -  $p = 2$

Este método por estimación clasica de la matriz de covarianzas y vector de medias nos detectó **380** outliers, siendo la mayoría de estos personas con Indices de Masa Corporal muy bajos como **15,44876** (este el minimo IMC de los outliers) o muy altos tales como **33,02813** (este el maximo IMC de los outliers). Geométricamente observamos que los outliers estan por fuera de una elipse umbral que determina si una

observación es un outlier o no. Ahora, apliquemos la estimación por DCM:

### Estimación DCM de la matriz de covarianzas y vector de medias

Este proceso es análogo al anterior: Calculamos la matriz de covarianzas muestral y el vector de medias por el método de estimación DCM:

```
# METODO ESTIMACION DCM

# Importamos la librería robustbase que contiene la función covMcd
library(robustbase)

# Hacemos la estimación por DCM usando covMcd
DCM1 <- covMcd(conjunto_datos_Pi2_interes)

# Calcula la matriz de covarianza muestral por el método DCM
matriz_cov_e1Mcd_2 <- DCM1$cov

# Calcula el vector de medias por el método DCM
vector_medias_e1Mcd_2 <- DCM1$center
```

```
> matriz_cov_e1Mcd_2
      Height      Weight
Height 15.46915 123.6566
Weight 123.65664 1126.2323
> vector_medias_e1Mcd_2
      Height      Weight
66.38602 161.47294
```

Figura 10: Matriz de covarianzas y vector de media por el método DCM - Ejemplo  $p = 2$

Observamos que este método estima una covarianza entre *Height* y *Weight* con valor mayor que el método clásico y una varianza mayor para cada variable; también aumentó la media de cada variable en esta estimación. Ahora, usaremos esto para calcular la distancia de Mahalanobis estimada para cada registro en nuestro conjunto de datos para después etiquetarla como outlier si es mayor al percentil **95** de la distribución  $\chi^2_{p=2}$  (usamos este umbral por la misma razón anteriormente explicada):

```
# Ciclo en los registros
```

```
# Declaramos una lista que contendrá los índices de los registros outliers de nuestro conjunto de datos
indices_outliers_DCM <- list()
```

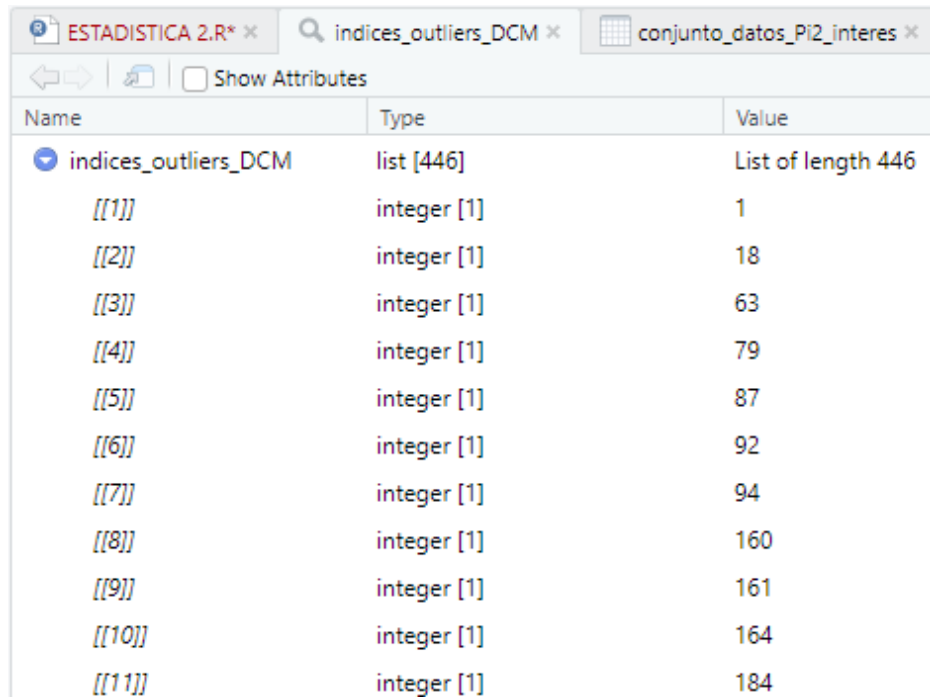
```

# Recorremos cada registro para hacerle la evaluación
for (i in 1:n){
  d2 <- as.matrix(conjunto_datos_Pi2_interes[i,] - vector_medias_e1Mcd_2)%*%as.matrix(solve(matri
  if (d2 > corte1){
    indices_outliers_DCM <- c(indices_outliers_DCM, i)
  }
}

print(indices_outliers_DCM)

```

A continuación observemos los resultados, una grafica de dispersión con los outliers pintados de rojo y un análisis:



Name	Type	Value
indices_outliers_DCM	list [446]	List of length 446
[[1]]	integer [1]	1
[[2]]	integer [1]	18
[[3]]	integer [1]	63
[[4]]	integer [1]	79
[[5]]	integer [1]	87
[[6]]	integer [1]	92
[[7]]	integer [1]	94
[[8]]	integer [1]	160
[[9]]	integer [1]	161
[[10]]	integer [1]	164
[[11]]	integer [1]	184

Figura 11: Registros outliers por método DCM - Ejemplo  $p = 2$

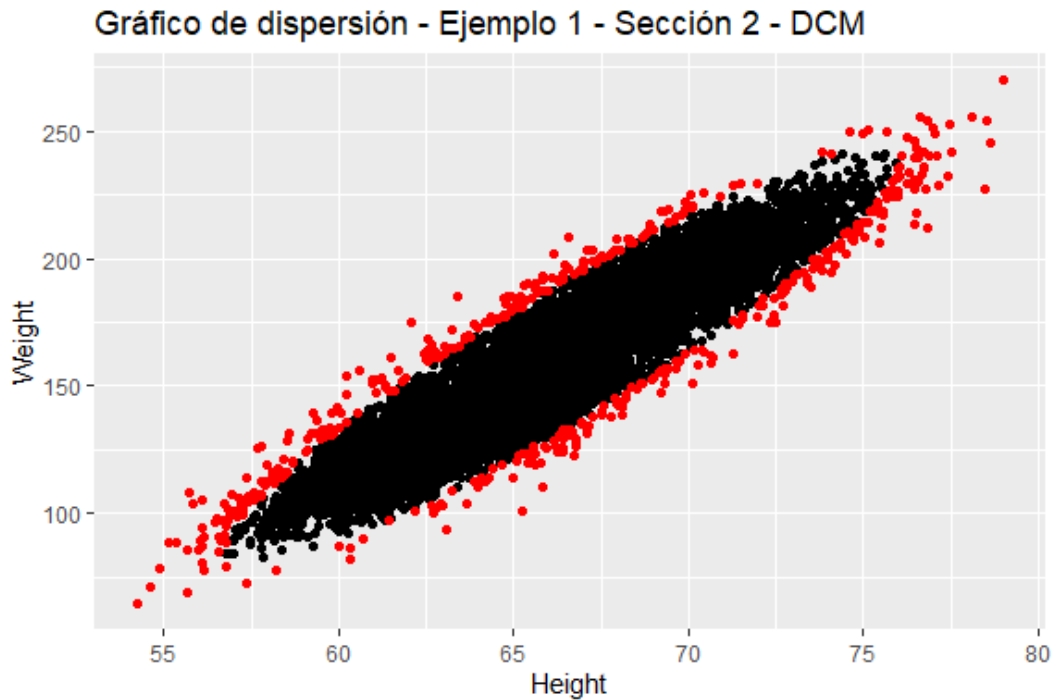


Figura 12: Grafico de dispersión de los datos - DCM -  $p=2$

Este método por estimación DCM de la matriz de covarianzas y vector de medias nos detectó **446** outliers, **66** outliers mas que el método clasico, siendo la mayoría de estos también personas con Indices de Masa Corporal muy bajos y altos siendo los indices mínimo y maximo los mismos que los detectados por el método anterior. La razón por la que el método DCM detectó mas outliers es por el efecto de enmascaramiento:

#### 2.2.4.5 Efecto de Enmascaramiento

Se dice que un valor atípico enmascara a un segundo cercano si este último puede ser considerado un valor atípico por sí mismo, pero ya no si se considera junto con el primero. De manera equivalente, después de la eliminación de un valor atípico, otra instancia puede surgir como un valor atípico. El enmascaramiento se produce cuando un grupo de puntos periféricos sesga las estimaciones de la media y la covarianza hacia él, y la distancia resultante del punto periférico a la media es pequeña.

El enmascaramiento se puede resolver mediante el uso de estimaciones robustas del centroide y la matriz de covarianza, que por definición se ven menos afectadas por los valores atípicos. Es menos probable que los puntos periféricos entren en el cálculo de las estadísticas robustas, por lo que no podrán influir en los parámetros utilizados en la distancia de Mahalanobis. Algunos estimadores robustos del centroide y de la matriz de covarianza incluyen el elipsoide de volumen mínimo (MVE) y el ya estudiado determinante de covarianza mínima (DCM).

Por estas razones, al usar estimadores robustos del vector de medias y de la matriz de covarianzas se detectan mas outliers, porque al no verse influidos por los valores atipicos, el método por Mahalanobis es mas preciso al calcular la distancia sin sesgos y por tanto ningún outlier enmascara a otro provocando que se puedan detectar mas outliers.

#### 2.2.4.6 Ejemplo con $p > 2$

Para este ejemplo tomaremos la base de datos `apple_quality.csv`. Según la fuente de donde los tomamos, este conjunto de datos contiene **4001** registros de información sobre varios atributos de un conjunto de frutas, proporcionando información sobre sus características. Contiene las siguientes variables:

- *A\_id* (ID): Identificador único para cada fruta
- *Size* (Numérico): Tamaño de la fruta
- *Weight* (Numérico): Peso de la fruta
- *Sweetness* (Numérico): Grado de dulzura de la fruta
- *Crunchiness* (Numérico): Textura que indica la crujencia de la fruta
- *Juiciness* (Numérico): Nivel de jugosidad de la fruta
- *Ripeness* (Numérico): Etapa de madurez de la fruta
- *Acidity* (Numérico): Nivel de acidez de la fruta
- *Quality* (Cadena): Calidad general de la fruta

Nos interesa las variables numéricas, por lo tanto solo trabajaremos con ellas ignorando *A\_id* y *Quality*. No se mostrará los códigos debido a que son similares a los del ejemplo con  $p = 2$ , pero se mostrará los resultados correspondientes (Cabe aclarar que se usó el como umbral de desición otra vez el percentil 95 de la distribución  $\chi^2_{p=7}$ ):

## Estimación clásica de la matriz de covarianzas y vector de medias

```
> matriz_covarianzas_clasico
      size      weight Sweetness Crunchiness Juiciness Ripeness Acidity
size      3.71741031 -0.52742168 -1.2165989  0.4594249 -0.07031204 -0.4870700 0.79835708
weight    -0.52742168  2.56802937 -0.4803813 -0.2155353 -0.28539619 -0.7323950 0.05550836
Sweetness -1.21659886 -0.48038131  3.7769616 -0.1023734  0.35968955 -0.9974091 0.35269854
Crunchiness 0.45942493 -0.21553529 -0.1023734  1.9677278 -0.70294385 -0.5310836 0.20704506
Juiciness  -0.07031204 -0.28539619  0.3596895 -0.7029439  3.72600278 -0.3514847 1.01311691
Ripeness   -0.48707002 -0.73239504 -0.9974091 -0.5310836 -0.35148472  3.5134757 -0.80166686
Acidity    0.79835708  0.05550836  0.3526985  0.2070451  1.01311691 -0.8016669 4.45323794
> vector_medias_clasico
      size      weight Sweetness Crunchiness Juiciness Ripeness Acidity
-0.5030146 -0.9895465 -0.4704785  0.9854779  0.5121180  0.4982774 0.0768773
```

Figura 13: Matriz de covarianzas y vector de medias por método clásico - Ejemplo  $p > 2$

Name	Type	Value
indices_outliers_clasico_a...	list [295]	List of length 295
[[1]]	integer [1]	9
[[2]]	integer [1]	20
[[3]]	integer [1]	21
[[4]]	integer [1]	22
[[5]]	integer [1]	45
[[6]]	integer [1]	66
[[7]]	integer [1]	72
[[8]]	integer [1]	77
[[9]]	integer [1]	84
[[10]]	integer [1]	113
[[11]]	integer [1]	147
[[12]]	integer [1]	162
[[13]]	integer [1]	186
[[14]]	integer [1]	208
[[15]]	integer [1]	213
[[16]]	integer [1]	224
[[17]]	integer [1]	228
[[18]]	integer [1]	245
[[19]]	integer [1]	248
[[20]]	integer [1]	252
[[21]]	integer [1]	254
[[22]]	integer [1]	261
[[23]]	integer [1]	270

Figura 14: Registros outliers por método clásico - Ejemplo  $p > 2$



## Estimación DCM de la matriz de covarianzas y vector de medias

```
> matriz_covarianzas_DCM
      Size      weight Sweetness Crunchiness Juiciness Ripeness Acidity
Size      3.2749154 -0.29545635 -1.1590405  0.6535023 -0.39720280 -0.5331287  0.6967551
weight    -0.2954563  2.13753998 -0.5888812 -0.2501553  0.03473885 -0.7615063  0.4145775
Sweetness -1.1590405 -0.58888118  3.7640704 -0.1659199  0.57447522 -0.9292683  0.1868033
Crunchiness 0.6535023 -0.25015528 -0.1659199  1.6022271 -0.33532596 -0.5769999  0.5177680
Juiciness  -0.3972028  0.03473885  0.5744752 -0.3353260  3.10800440 -0.1679948  0.3963053
Ripeness   -0.5331287 -0.76150625 -0.9292683 -0.5769999 -0.16799483  3.2612881 -0.7072949
Acidity    0.6967551  0.41457751  0.1868033  0.5177680  0.39630532 -0.7072949  3.7453794
> vector_medias_DCM
      Size      weight Sweetness Crunchiness Juiciness Ripeness Acidity
-0.6137470 -0.8661067 -0.4490120  1.0784886  0.2699588  0.5993714 -0.1687464
```

Figura 15: Matriz de covarianzas y vector de medias por método DCM - Ejemplo  $p > 2$

ESTADISTICA 2.R\* x

Untitled1\* x

indices\_outliers\_DCM\_apple x

←

→

📄

☐ Show Attributes

Name	Type	Value
indices_outliers_DCM_ap...	list [544]	List of length 544
[[1]]	integer [1]	4
[[2]]	integer [1]	7
[[3]]	integer [1]	9
[[4]]	integer [1]	18
[[5]]	integer [1]	20
[[6]]	integer [1]	21
[[7]]	integer [1]	22
[[8]]	integer [1]	34
[[9]]	integer [1]	40
[[10]]	integer [1]	45
[[11]]	integer [1]	55
[[12]]	integer [1]	66
[[13]]	integer [1]	67
[[14]]	integer [1]	72
[[15]]	integer [1]	77
[[16]]	integer [1]	84
[[17]]	integer [1]	113
[[18]]	integer [1]	114
[[19]]	integer [1]	115
[[20]]	integer [1]	131
[[21]]	integer [1]	146
[[22]]	integer [1]	147
[[23]]	integer [1]	162

Figura 16: Registros outliers por método DCM - Ejemplo  $p > 2$

Observamos que las estimaciones del vector de medias y las covarianzas entre variables diferentes cambian bastante en ambos métodos, siendo la varianza de cada variable el valor que menos cambia de un método a otro. Notamos que el método clásico detectó **295** outliers mientras que el método DCM detectó **544**, es decir **249** mas outliers. Esto era de esperarse debido al efecto de enmascaramiento que provoca los estimadores clásicos de la matriz de covarianzas y el vector de medias al no ser muy robustos, pues, muchos outliers quedan indetectables al usar los estimadores clásicos, pero con las estimaciones DCM se evita efectivamente este problema y la limpieza de datos se logra con éxito.

## Conclusión

En el presente trabajo se abordaron y aplicaron los test de hipótesis de Shapiro-Wilk y Mardia para poner a prueba si un conjunto de datos provienen de una distribución normal multivariada; se colocaron en práctica con los datos integrados en R “iris” tomando la especie “setosa” y se obtuvo como resultado que hay suficiente evidencia para no rechazar la hipótesis nula. Estos tests muy importantes en la actualidad debido a que muchos procesos exigen la normalidad de los datos. Además se indagó profundamente en la detección de datos atípicos usando el método de Boxplot aplicado a “nh-temp” (integrado en R) y la medida de distancia estadística de Mahalanobis explorando dos estimaciones para esta última: Estimación de la matriz de covarianzas y vector de medias por el método clásico o por el método de Determinante de Covarianza Mínima. Se hicieron 2 limpiezas de datos con datasets diferentes, uno con dimensionalidad 2 y otro con dimensionalidad 7 y se comprobó que efectivamente el método DCM detecta mas outliers debido a que es mas resistente al efecto del enmascaramiento. La detección de outliers es crucial para evitar sesgos y ruido en los datos que son bastante usados para entrenamientos de redes neuronales o clustering.

## Referencias

- [1] Edgar Acuna y Caroline Rodriguez. «A Meta Analysis Study of Outlier Detection Methods in Classification». En: (2004).
- [2] Jaime Carlos Porras Cerron. *Vista de comparación de pruebas de normalidad multivariada*. URL: [https://revistas.lamolina.edu.pe/index.php/acu/article/view/483/pdf\\_21](https://revistas.lamolina.edu.pe/index.php/acu/article/view/483/pdf_21).
- [3] *Conjunto de datos IRIS*. URL: <https://search.r-project.org/CRAN/refmans/MVTests/html/iris.html>.
- [4] *Distribución normal multivariada - Wikipedia*. URL: [https://es.wikipedia.org/wiki/Distribuci%C3%B3n\\_normal\\_multivariada](https://es.wikipedia.org/wiki/Distribuci%C3%B3n_normal_multivariada).
- [5] Hamid Ghorbani. «Mahalanobis Distance and Its Application for Detecting Multivariate Outliers». En: *Facta Universitatis. Mathematics and Informatics* (2019). ISSN: 0352-9665.
- [6] *Libreria MVN*. URL: <https://cran.r-project.org/web/packages/MVN/MVN.pdf>.
- [7] *mshapiro.test en R*. URL: <https://www.rdocumentation.org/packages/RVAideMemoire/versions/0.9-83-7/topics/mshapiro.test>.
- [8] *MVN in R*. URL: <https://cran.r-project.org/web/packages/MVN/vignettes/MVN.html>.
- [9] ALVIN C. RENCHER. *Methods of Multivariate Analysis*. Second edition. ISBN: 0-471-41889-7. 2002. URL: <https://onlinelibrary.wiley.com/doi/book/10.1002/9781118391686>.
- [10] Peter J. Rousseeuw y Karel Van Driessen Driessen. «A Fast Algorithm for the Minimum Covariance Determinant Estimator». En: *Technometrics* 41.3 (1999), págs. 212-223. DOI: 10.1080/00401706.1999.10485670.
- [11] Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. URL: <https://egrcc.github.io/docs/math/all-of-statistics.pdf>.