Análisis Multivariado

Ana María Sánchez

2024-08-15

Carga inicial de los datos

```
library(MultBiplotR)

# Cargar los datos Protein
data("Protein")
str(Protein)
```

```
## 'data.frame':    25 obs. of  11 variables:
##  $ Comunist         : Factor w/ 2 levels "No","Yes": 2 1 1 2 2 1 2 1 1
1 ...
##  $ Region           : Factor w/ 3 levels "North","Center",..: 3 2 2 3
2 1 2 1 2 3 ...
##  $ Red_Meat         : num  10.1 8.9 13.5 7.8 9.7 10.6 8.4 9.5 18 10.2
...
##  $ White_Meat       : num  1.4 14 9.3 6 11.4 10.8 11.6 4.9 9.9 3 ...
##  $ Eggs             : num  0.5 4.3 4.1 1.6 2.8 3.7 3.7 2.7 3.3 2.8 ...
##  $ Milk             : num  8.9 19.9 17.5 8.3 12.5 25 11.1 33.7 19.5
17.6 ...
##  $ Fish             : num  0.2 2.1 4.5 1.2 2 9.9 5.4 5.8 5.7 5.9 ...
##  $ Cereal           : num  42.3 28 26.6 56.7 34.3 21.9 24.6 26.3 28.1
41.7 ...
##  $ Starch           : num  0.6 3.6 5.7 1.1 5 4.8 6.5 5.1 4.8 2.2 ...
##  $ Nuts             : num  5.5 1.3 2.1 3.7 1.1 0.7 0.8 1 2.4 7.8 ...
##  $ Fruits_Vegetables: num  1.7 4.3 4 4.2 4 2.4 3.6 1.4 6.5 6.5 ...
```

a) Cálculo del vector de medias y matriz de covarianzas

```
# Vector de medias
numeric_columns <- sapply(Protein, is.numeric)
mean_vector <- colMeans(Protein[, numeric_columns])
mean_vector
```

```
##          Red_Meat        White_Meat              Eggs
Milk
##             9.828             7.896             2.936
17.112
##              Fish           Cereal           Starch
Nuts
##             4.284            32.248             4.276
3.072
## Fruits_Vegetables
##             4.136
```

```r
numeric_data <- Protein[, sapply(Protein, is.numeric)]

# Calcular la matriz de covarianzas
cov_matrix <- cov(numeric_data)
cov_matrix
```

```
##                      Red_Meat White_Meat        Eggs       Milk
Fish
## Red_Meat           11.2029333   1.891783  2.19061667  11.960900
0.6942167
## White_Meat          1.8917833  13.646233  2.56140000   7.388383  -
2.9413167
## Eggs                2.1906167   2.561400  1.24906667   4.570383
0.2493500
## Milk               11.9609000   7.388383  4.57038333  50.486933
3.3335333
## Fish                0.6942167  -2.941317  0.24935000   3.333533
11.5772333
## Cereal            -18.3622333 -16.776050 -8.73846667 -46.221850 -
19.5758667
## Starch              0.7407000   1.894067  0.82590000   2.582383
2.2454333
## Nuts               -2.3225167  -4.657617 -1.24228333  -8.762983  -
0.9942167
## Fruits_Vegetables  -0.4481333  -0.408600 -0.09176667  -5.234200
1.6335167
##                         Cereal     Starch       Nuts Fruits_Vegetables
## Red_Meat          -18.3622333  0.7407000 -2.3225167       -0.44813333
## White_Meat        -16.7760500  1.8940667 -4.6576167       -0.40860000
## Eggs               -8.7384667  0.8259000 -1.2422833       -0.09176667
## Milk              -46.2218500  2.5823833 -8.7629833       -5.23420000
## Fish              -19.5758667  2.2454333 -0.9942167        1.63351667
## Cereal            120.4459333 -9.5633833 14.1868167        0.92153333
## Starch             -9.5633833  2.6702333 -1.5390333        0.24881667
## Nuts               14.1868167 -1.5390333  3.9429333        1.34313333
## Fruits_Vegetables   0.9215333  0.2488167  1.3431333        3.25406667
```

b) Media de los datos por región

```r
# Calcular la media de las variables numéricas por región
mean_by_region <- aggregate(Protein[, numeric_columns],
                            by = list(Region = Protein$Region),
                            FUN = mean)

# Mostrar el resultado
mean_by_region
```

```
##   Region Red_Meat White_Meat     Eggs     Milk     Fish   Cereal
Starch
## 1  North  9.85000    7.05000 3.150000 26.67500 8.225000 22.67500
4.5500
```

```
## 2 Center 11.17692   10.40769 3.546154 18.34615 3.130769 28.94615
5.0000
## 3  South  7.62500    4.23750 1.837500 10.32500 4.187500 42.40000
2.9625
##       Nuts Fruits_Vegetables
## 1 1.175000          2.125000
## 2 2.246154          4.207692
## 3 5.362500          5.025000
```

Hay diferencias claras en las preferencias alimentarias entre las regiones. El Norte tiende a consumir más leche y pescado, mientras que el Centro consume más carne (tanto roja como blanca). El Sur muestra consistentemente un consumo más bajo en todas las categorías, lo que podría reflejar diferencias en la dieta tradicional o en la disponibilidad de estos productos. Las diferencias en el consumo de estos productos proteícos pueden estar influidas por factores como la cultura, la economía, la geografía y la disponibilidad de productos.

c) Gráfico de estrellas y caras de chernoff

```r
# Gráfico de estrellas
Sys.setlocale("LC_ALL", "es_ES.UTF-8")
```

```
## [1] "LC_COLLATE=es_ES.UTF-8;LC_CTYPE=es_ES.UTF-
8;LC_MONETARY=es_ES.UTF-8;LC_NUMERIC=C;LC_TIME=es_ES.UTF-8"
```

```r
stars(Protein[, numeric_columns],
      labels = rownames(Protein),
      main = "Gráfico de Estrellas para el Consumo de Proteínas en
Europa",
      full = TRUE, # Las estrellas se llenan completamente (en lugar de
una mitad)
      flip.labels = FALSE, # No se invierten las etiquetas
      draw.segments = TRUE) # Segmentos visibles entre variables
```

co de Estrellas para el Consumo de Proteínas en Eu


```r
# Instalar y cargar el paquete aplpack para las Caras de Chernoff
if (!require(aplpack)) install.packages("aplpack", dependencies = TRUE)

## Cargando paquete requerido: aplpack

library(aplpack)

# Gráfico de caras de Chernoff
faces(Protein[, numeric_columns],
      labels = rownames(Protein),
      main = "Caras de Chernoff para el Consumo de Proteínas en Europa")
```


```
## effect of variables:
##  modified item       Var
##  "height of face   " "Red_Meat"
##  "width of face    " "White_Meat"
##  "structure of face" "Eggs"
##  "height of mouth  " "Milk"
##  "width of mouth   " "Fish"
##  "smiling          " "Cereal"
##  "height of eyes   " "Starch"
##  "width of eyes    " "Nuts"
##  "height of hair   " "Fruits_Vegetables"
##  "width of hair    "  "Red_Meat"
##  "style of hair    "  "White_Meat"
##  "height of nose   "  "Eggs"
##  "width of nose    "  "Milk"
##  "width of ear     "  "Fish"
##  "height of ear    "  "Cereal"
```

d) Comprobación de normalidad en los datos

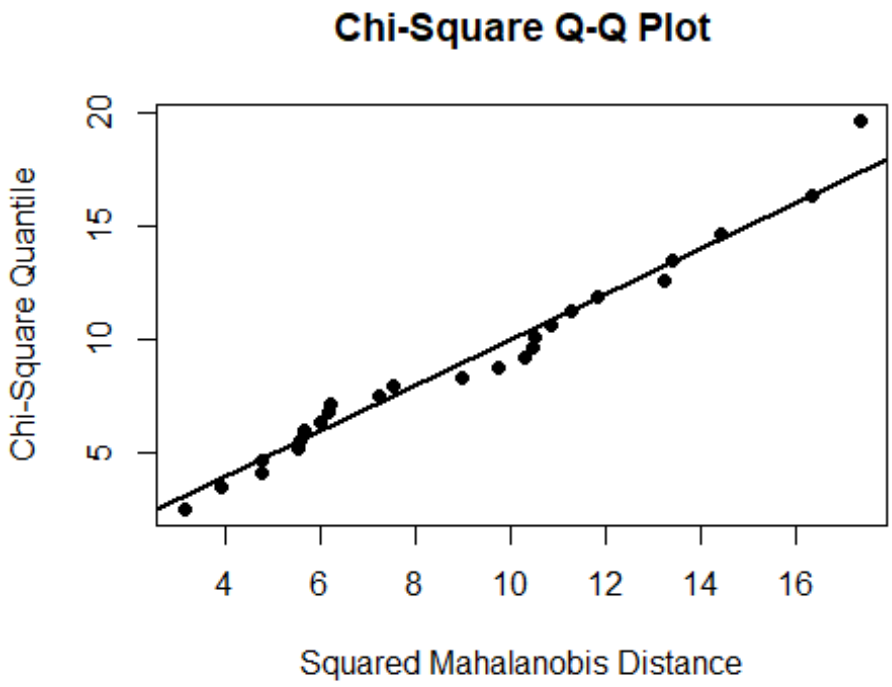
```
# Instalar y cargar el paquete MVN para pruebas de normalidad
multivariada
if (!require(MVN)) install.packages("MVN", dependencies = TRUE)

## Cargando paquete requerido: MVN
```

```r
library(MVN)

# Seleccionar solo las columnas numéricas
numeric_data <- Protein[, sapply(Protein, is.numeric)]

# Gráfico Q-Q de normalidad multivariada
mvn_result <- mvn(data = numeric_data, mvnTest = "royston",
multivariatePlot = "qq")
```



e) Comprobación del supuesto de normalidad en los datos con Mardia

```r
library(MVN)

# Seleccionar solo las columnas numéricas del dataset Protein
numeric_data <- Protein[, sapply(Protein, is.numeric)]

# Realizar la prueba de Mardia
mardia_test <- mvn(data = numeric_data, mvnTest = "mardia")

# Mostrar los resultados de la prueba
print(mardia_test$multivariateNormality)
```

```
##              Test          Statistic           p value Result
## 1 Mardia Skewness   168.086605971262  0.41858425807236    YES
## 2 Mardia Kurtosis -0.523571666842164 0.600576492382074    YES
## 3             MVN               <NA>              <NA>    YES
```

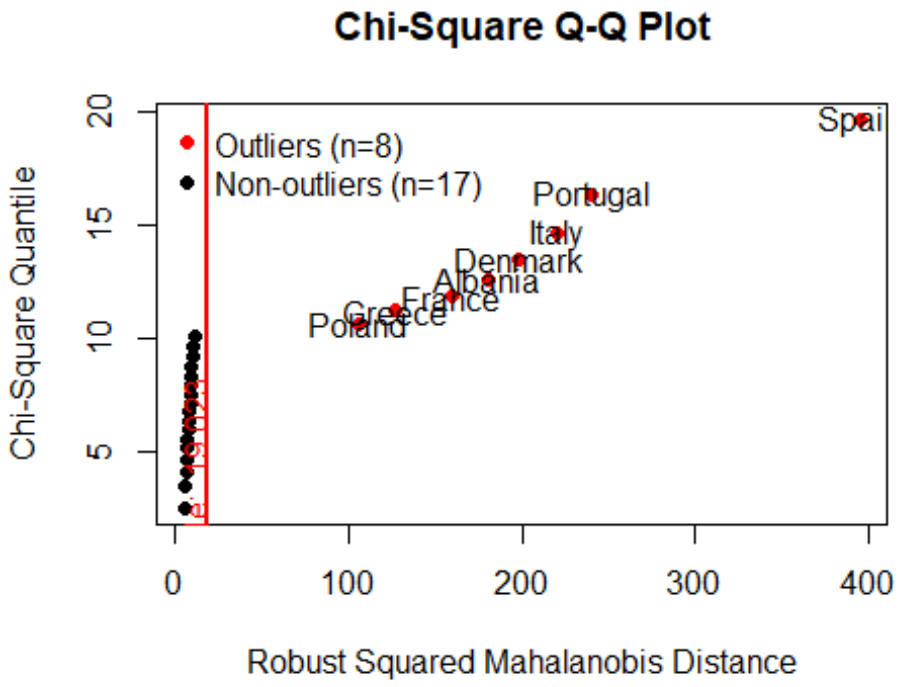
Con los resultados obtenidos, bajo la prueba de Mardia, los datos no muestran desviaciones significativas de la normalidad multivariada. En otras palabras, no hay evidencia suficiente para rechazar la hipótesis de que los datos provienen de una población normal multivariada.

f) Detección de outliers en los datos

```
library(MVN)

# Seleccionar solo las columnas numéricas del dataset Protein
numeric_data <- Protein[, sapply(Protein, is.numeric)]

# Calcular las distancias de Mahalanobis y realizar la prueba de outliers multivariados
outliers_test <- mvn(data = numeric_data, multivariateOutlierMethod = "quan")
```



g) Pruebas de hipótesis sobre igualdad de medias de forma univariada y multivariada

- Univariado

```
# Cargar los datos
data("Protein", package = "MultBiplotR")
numeric_data <- Protein[, sapply(Protein, is.numeric)]

# Definir las medias hipotéticas
mu0 <- c(9, 7, 2, 15, 5, 30, 4, 3, 4)
```

```r
# Inicializar una lista para guardar los resultados
t_test_results <- list()

# Realizar pruebas t para cada variable
for (i in seq_along(numeric_data)) {
  var_name <- colnames(numeric_data)[i]
  t_test <- t.test(numeric_data[[i]], mu = mu0[i])
  t_test_results[[var_name]] <- t_test
}

# Mostrar resultados de las pruebas t
for (var_name in names(t_test_results)) {
  cat("Variable:", var_name, "\n")
  print(t_test_results[[var_name]])
  cat("\n\n")
}
```

```
## Variable: Red_Meat
##
##  One Sample t-test
##
## data:  numeric_data[[i]]
## t = 1.2369, df = 24, p-value = 0.2281
## alternative hypothesis: true mean is not equal to 9
## 95 percent confidence interval:
##   8.446394 11.209606
## sample estimates:
## mean of x
##     9.828
##
##
##
## Variable: White_Meat
##
##  One Sample t-test
##
## data:  numeric_data[[i]]
## t = 1.2128, df = 24, p-value = 0.237
## alternative hypothesis: true mean is not equal to 7
## 95 percent confidence interval:
##  6.371158 9.420842
## sample estimates:
## mean of x
##     7.896
##
##
##
## Variable: Eggs
##
```

```
##  One Sample t-test
##
## data:  numeric_data[[i]]
## t = 4.1875, df = 24, p-value = 0.0003278
## alternative hypothesis: true mean is not equal to 2
## 95 percent confidence interval:
##  2.474671 3.397329
## sample estimates:
## mean of x
##     2.936
##
##
##
## Variable: Milk
##
##  One Sample t-test
##
## data:  numeric_data[[i]]
## t = 1.4862, df = 24, p-value = 0.1502
## alternative hypothesis: true mean is not equal to 15
## 95 percent confidence interval:
##  14.17903 20.04497
## sample estimates:
## mean of x
##    17.112
##
##
##
## Variable: Fish
##
##  One Sample t-test
##
## data:  numeric_data[[i]]
## t = -1.0522, df = 24, p-value = 0.3032
## alternative hypothesis: true mean is not equal to 5
## 95 percent confidence interval:
##  2.879503 5.688497
## sample estimates:
## mean of x
##     4.284
##
##
##
## Variable: Cereal
##
##  One Sample t-test
##
## data:  numeric_data[[i]]
## t = 1.0242, df = 24, p-value = 0.316
## alternative hypothesis: true mean is not equal to 30
```

```
## 95 percent confidence interval:
##  27.71783 36.77817
## sample estimates:
## mean of x
##    32.248
##
##
##
## Variable: Starch
##
##  One Sample t-test
##
## data:  numeric_data[[i]]
## t = 0.84451, df = 24, p-value = 0.4067
## alternative hypothesis: true mean is not equal to 4
## 95 percent confidence interval:
##  3.601483 4.950517
## sample estimates:
## mean of x
##     4.276
##
##
##
## Variable: Nuts
##
##  One Sample t-test
##
## data:  numeric_data[[i]]
## t = 0.1813, df = 24, p-value = 0.8577
## alternative hypothesis: true mean is not equal to 3
## 95 percent confidence interval:
##  2.252351 3.891649
## sample estimates:
## mean of x
##     3.072
##
##
##
## Variable: Fruits_Vegetables
##
##  One Sample t-test
##
## data:  numeric_data[[i]]
## t = 0.37696, df = 24, p-value = 0.7095
## alternative hypothesis: true mean is not equal to 4
## 95 percent confidence interval:
##  3.391385 4.880615
## sample estimates:
## mean of x
##     4.136
```

La mayoría de los resultados sugieren que no hay evidencia estadística para afirmar que las medias observadas de consumo de estos alimentos difieren significativamente de las medias hipotéticas probadas, excepto en el caso de los huevos, donde sí se observa una diferencia significativa.

- Multivariado

```
num_vars <- ncol(numeric_data)
cat("Número de variables en numeric_data:", num_vars, "\n")

## Número de variables en numeric_data: 9

mu_0 <- c(9, 7, 2, 15, 5, 30, 4, 3, 4)

# Convertir mu_0 a un vector columna
mu_0 <- matrix(mu_0, ncol = 1)

# Función para calcular el estadístico T² de Hotelling
T2 <- function(X, mu, n) {
  Xbarra <- colMeans(X)
  Xbarra <- matrix(Xbarra, ncol = 1) # Convertir a vector columna
  S <- cov(X)
  InvS <- solve(S)
  DifMed <- Xbarra - mu
  T2 <- n * t(DifMed) %*% InvS %*% DifMed
  return(T2)
}

# Número de variables y tamaño de la muestra
p <- num_vars
n <- nrow(numeric_data)

# Calcular el estadístico T² de Hotelling
T2_statistic <- T2(numeric_data, mu_0, n)
cat("Hotelling's T² statistic:", T2_statistic, "\n")

## Hotelling's T² statistic: 64.84185

# Calcular el valor crítico de la distribución F
qf <- qf(0.10, p, n - p, lower.tail = FALSE)
V <- (((n - 1) * p) / (n - p)) * qf
cat("Critical value from F-distribution:", qf, "\n")

## Critical value from F-distribution: 2.055331

cat("Scaled critical value V:", V, "\n")

## Scaled critical value V: 27.74696
```

El valor del estadístico T^2 (64.84185) es mucho mayor que el valor crítico escalado V (27.74696). Esto indica que hay evidencia suficiente para rechazar la hipótesis nula y

concluir que las medias muestrales son significativamente diferentes de las medias hipotéticas, contrario a los resultados arrojados por las pruebas univariadas.