Ejercicio 1 de la Evaluación continua de Análisis Multivariante

Pérez Efremova, Daniel

Noviembre 2020

Índice

Índice de Códigos

Ejercicio 1

Visualización general de los datos

En este ejercicio vamos a considerar la base de datos EUROSEC, que contiene como variables los porcentajes de empleo en países europeos en los sectores: agricultura, minería, industria, energía, construcción, servicios industriales, finanzas, servicios, transportes y comunicaciones.

Código 1: Carga de datos y librerías

```
library(ggplot2)
library(latex2exp)
library(ggpubr)
library(dplyr)
library(gridExtra)
library(reshape2)
library(GGally)

datos =
 read.table('https://www.mhe.es/universidad/ciencias_matematicas/pena/
     home/fichero/eurosec.dat', header=FALSE)
# nombres de las variables
colnames(datos) =
c('agric', 'min', 'ind', 'eng', 'cons', 'si', 'fin','serv', 'transcom')
```

Primeramente visualizamos de forma general las distribuciones marginales a través de los histogramas (Figura 1) y los gráficos de dispersión (Figura 2) para ver cómo se distribuyen las variables y si existe algún tipo de relación entre ellas.

Código 2: Generación de histogramas y gráficos de dispersión

```
# histogramas
gg <- melt(datos)
plot1 = ggplot(gg, aes(x=value, fill=variable)) +
geom_histogram(bins=10)+
facet_wrap(variable~.)

# Graficos de dispersion
plot2 = ggpairs(datos, diag = list(continuous = "blankDiag"))
```

A primera vista en la Figura 1 no se aprecian mezclas de distribuciones salvo en la variable *agric*, que se corresponde con el sector agrícola, como es lógico en cada país la climatología y el nivel de desarrollo económico puede favorecer o no el desarrollo de la agricultura, dando lugar a mayor o menor número de empleos en el sector. Tampoco se aprecian distribuciones conocidas, salvo en el sector industrial, ya que al ser países occidentales era de esperar una distribución normal en el número de empleos en industria que es uno de los motores de la economía.

Por otro lado, en la Figura 2 no se aprecian correlaciones lineales significativas entre las componentes, salvo en sectores que se complementen entre sí, como por ejemplo transportes y comunicación (*transcom*) con servicios (*serv*).

Figura 1: Histograma de cada variable

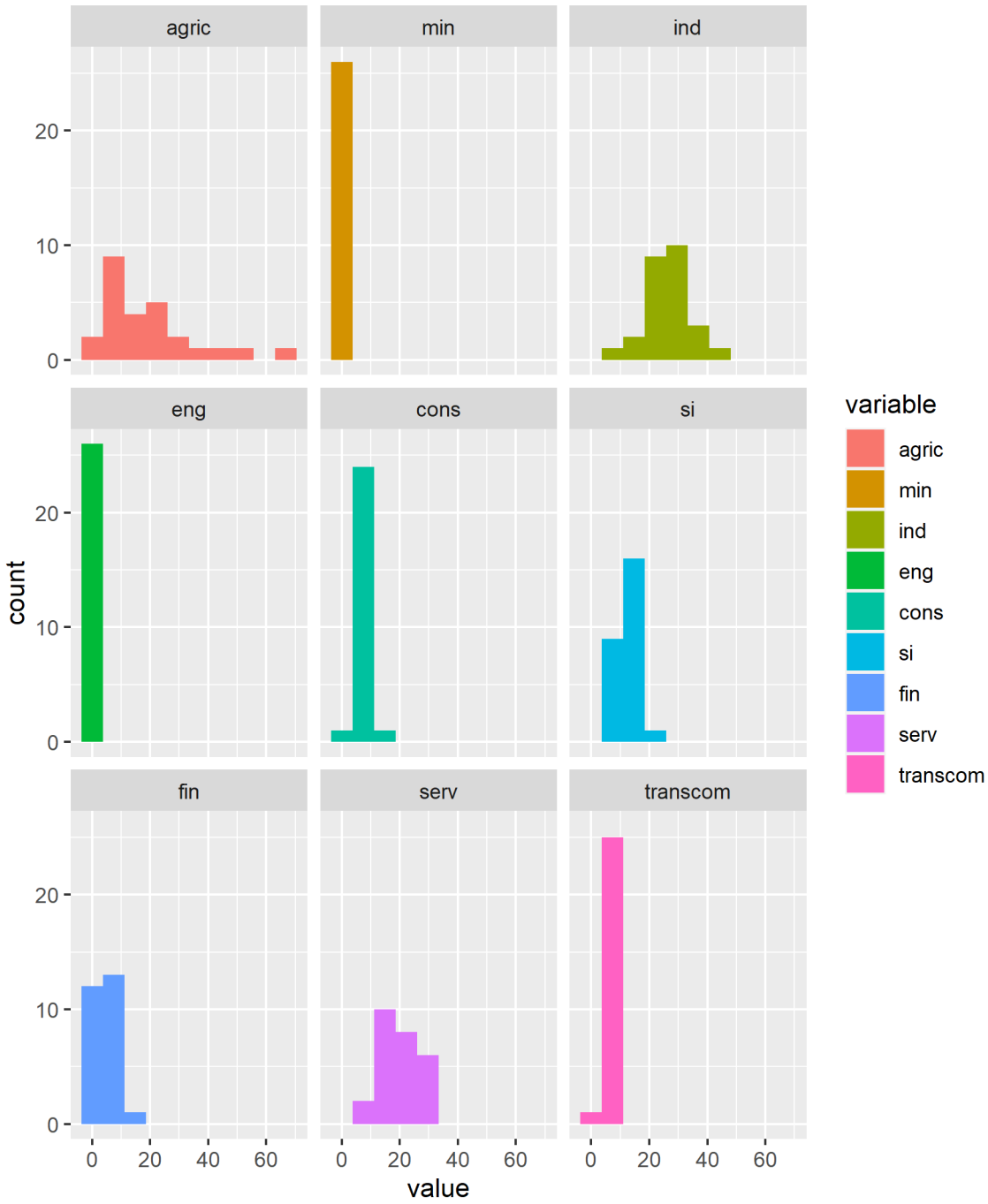
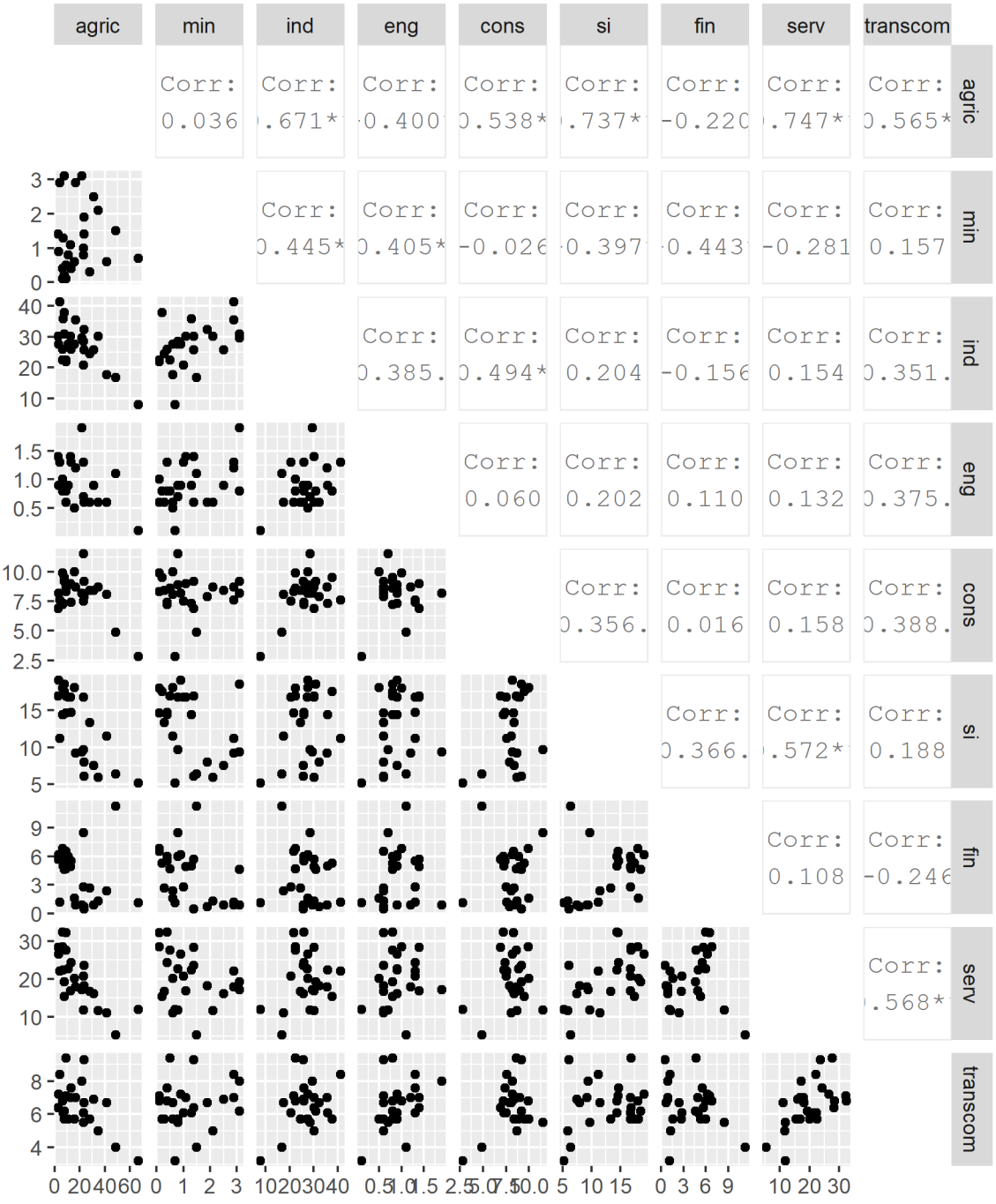


Figura 2: Matriz de gráficos de dispersión y matriz de correlaciones



Medidas de posición central, variabilidad, simetría y curtosis

Ahora vamos a calcular las medias, desviaciones estándar y coeficientes de asimetría, curtosis y de variación de Pearson para cada variable. Esto podemos hacerlo sencillamente con el siguiente código, donde hemos operado de forma matricial con la función *apply* de R.

Código 3: Cálculo de los estadísticos descriptivos

```
restar_media = function(x){

        x - mean(x) # a cada elemento de un vector le resta su media
}

sumar_potencias3 = function(x){

        sum(x**3) # funcion para calcular la suma de un vector al cubo
}

sumar_potencias4 = function(x){

        sum(x**4) # funcion para calcular la suma de un vector a la
             cuarta
}

n = dim(datos)[1]  # numero de observaciones

medias = colMeans(datos)  # vector de medias
desv_tipicas = apply(datos, 2, sd)*((n-1)/n)  # vector de desviaciones
     tipicas

# codigo para calcular los coficientes de asimetria y curtosis

datos_centrados = apply(datos, 2, restar_media)  # a cada valor de una
     columna le resta su media

sumas_cubos = apply(datos_centrados, 2, sumar_potencias3)  # suma de
     cada columna al cubo
sumas_cuartas = apply(datos_centrados, 2, sumar_potencias4)  # suma de
     cada columna al cubo

asimetria = (1/n)*sumas_cubos/(desv_tipicas**3)
curtosis = (1/n)*sumas_cuartas/(desv_tipicas**4) + 1

# codigo para calcular los coeficientes de variacion

coef_variacion = desv_tipicas/abs(medias)
```

Por sencillez en la lectura vemos en la siguiente tabla los resultados.

Tabla 1: Análisis descriptivo de las medias

	agric	min	ind	eng	cons	si	fin	serv	transcom
Medias	19.13	1.25	27.00	0.90	8.16	12.95	4.00	20.02	6.54
D. Estándar	14.94	0.93	6.73	0.36	1.58	4.39	2.69	6.56	1.33
Coef. asimetría	1.46	0.78	-0.47	0.52	-1.35	-0.35	0.58	-0.014	-0.09
Coef. curtosis	6.07	3.46	5.13	4.76	7.73	2.79	4.05	3.67	4.81
Coef. variación	0.78	0.74	0.24	0.39	0.19	0.33	0.67	0.32	0.20

Vemos que las medias son muy distintas, concluyendo que en promedio los sectores que mayor empleo concentran son: industrial (*ind*), servicios (*serv*) y agricultura (*agri*). Era de esperar puesto que son los sectores tradicionales de la economía y la base para el correcto desarrollo del resto.

Por otro lado, las desviaciones típicas no son en general grandes salvo en *agric*, que es especialmente grande en comparación al resto.

Los coeficientes de asimetría y curtosis son todos moderadamente cercanos a cero y cercanos a 3 o 4 respectivamente, indicando una simetría moderada, salvo en dos casos. En el caso de *agric*, vemos que la asimetría está muy marcada hacia la derecha con una alta curtosis, esto nos está indicando que pueden existir algunos países cuyo número de empleos dedicados a la agricultura sea muy alto en comparación a la media y quizá deberían estudiarse por separado. Por otro lado, la variable *cons* muestra una curtosis entre 7 y 8, con asimetría hacia la izquierda, lo que indica una clara presencia de *outliers*, que deben estudiarse por separado, y que están contaminando la descripción del resto de datos.

Una vez confirmada la presencia de outliers, debemos emplear algunas medidas de dispersión y centralidad robustas, como por ejemplo la mediana, la MEDA y el coeficiente de variación robusto, que es el cociente meda/mediana.

Código 4: Cálculo de los estadísticos robustos

```
calcular_MEDA = function(x){

        median(abs(x-median(x)))
}

medianas = apply(datos, 2, median)
medas = apply(datos, 2, calcular_MEDA)
c_robusto = medas/medianas
```

Vemos en la Tabla 2 el resultado. Vemos que el análisis robusto es relativamente

Tabla 2: Análisis descriptivo de las medias

	agric	min	ind	eng	cons	si	fin	serv	transcom
mediana	14.45	0.95	27.55	0.85	8.35	14.4	4.65	19.65	6.7
meda	8.4	0.55	3.15	0.25	0.8	3.4	2.05	4.1	0.75
meda/mediana	0.58	0.57	0.11	0.29	0.09	0.23	0.44	0.20	0.11

similar al de la Tabla 1, confirmando que las ligeras diferencias entre ambos análisis se deben en gran medida a la existencia de valores atípicos, sobretodo en *agric*.

Valores atípicos

Para encontrar los valores atípicos (cuya existencia hemos pronosticado anteriormente) usaremos la distancia de Mahalanobis al vector de medias y daremos

fuerza a nuestras conclusiones con la proyección de los datos sobre las direcciones de máxima y mínima curtosis.

Código 5: Cálculo de las ditancias de Mahalanobis y las proyecciones de máxima y mínima curtosis (librería *Pursuit* de R)

```
library(Pursuit) # libreria que contiene el metodo del TB

S = cov(datos[,1:9])  #matriz de covarianzas de los datos
distancias_mah = c()  # vector que almaena las distancias

for(i in 1:n){  # iteramos sobre las filas de datos calculando la
    distancia Mah.

        # usamos los datos centrados calculados anteriormente
        d = t(matrix(datos_centrados[i,1:9]))%*%inv(S)matrix%*%(matrix
            (datos_centrados[i, 1:9]))
        distancias_mah = append(distancias_mah, d)
}

datos$dist_mah = sqrt(distancias_mah)  # guardamos el resultado en el
    dataframe
qplot(datos$dist_mah, bins=10, xlim=c(min(datos$dist_mah), max(datos$
    dist_mah))) #histograma de distancias

# calculamos las proyecciones de maxima y minima curtosis

Res <- PP_Optimizer(data = datos[1:9], findex = 'kurtosismax',
optmethod = "GTSA", dimproj = 1, sphere = FALSE)

Res2 <- PP_Optimizer(data = datos[1:9], findex = 'kurtosismin',
optmethod = "GTSA", dimproj = 1, sphere = FALSE)

datos$pmaxk = Res$proj.data  # proyeccion en la de maxima curtosis
datos$pmink = Res2$proj.data # proyeccion en la de minima curtosis

# grafico de las proyecciones
ggplot(datos, aes(x= pmink, y= pmaxk, label=c(1:n)))+
geom_point() +
geom_text(aes(label=ifelse(pmaxk>26,as.character(c(1:n)),'')),hjust=0,
    vjust=0)
```

Vemos en la Figura 3 que el histograma de las distancia nos muestra observaciones a una distancia muy alejada del resto, tomaremos el valor máximo $d_m = 3{,}5$ para filtrar los datos y comprobar qué observaciones están a una distancia $d \geq d_m$

Código 6: Comprobación de las observaciones con mayor distancia de Mahalanobis

```
print(datos[datos$dist_mah >= 3.5, 1:10])  # paises con distancia Mah
    mayor que 3.5

[1]  agric min  ind eng cons   si  fin serv transcom dist_mah
7     7.7 3.1 30.8 0.8  9.2 18.5  4.6 19.2       6.2 4.200297
15   22.9 0.8 28.5 0.7 11.5  9.7  8.5 11.8       5.5 3.696938
18   66.8 0.7  7.9 0.1  2.8  5.2  1.1 11.9       3.2 4.249545
26   48.7 1.5 16.8 1.1  4.9  6.4 11.3  5.3       4.0 4.314171
```

Vemos que estos valores atípicos se corresponden con los países 7, 15, 18 y 26. Para complementar el análisis podemos comprobar que efectivamente son valores atípicos mirando las proyecciones sobre las direcciones de máxima y mínima curtosis.

Figura 3: Histograma de las distancias de Mahalanobis al vector de medias

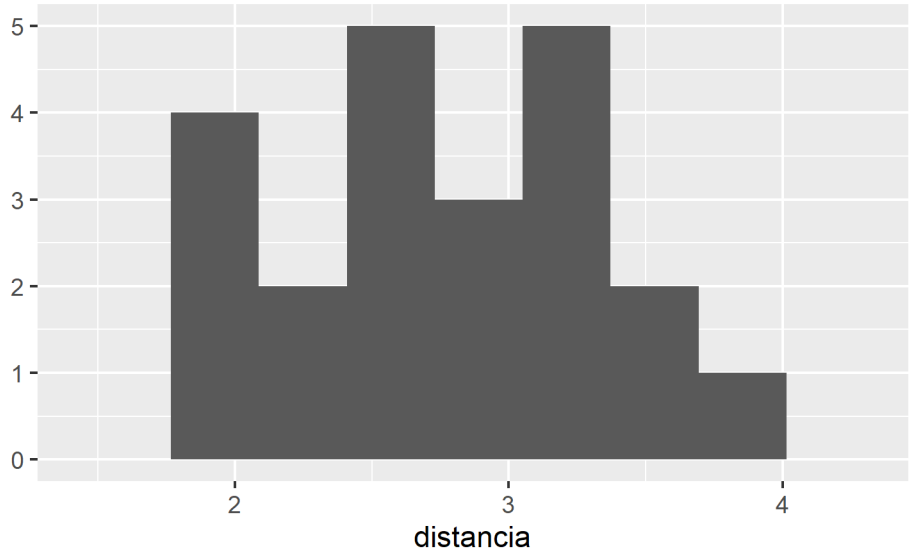
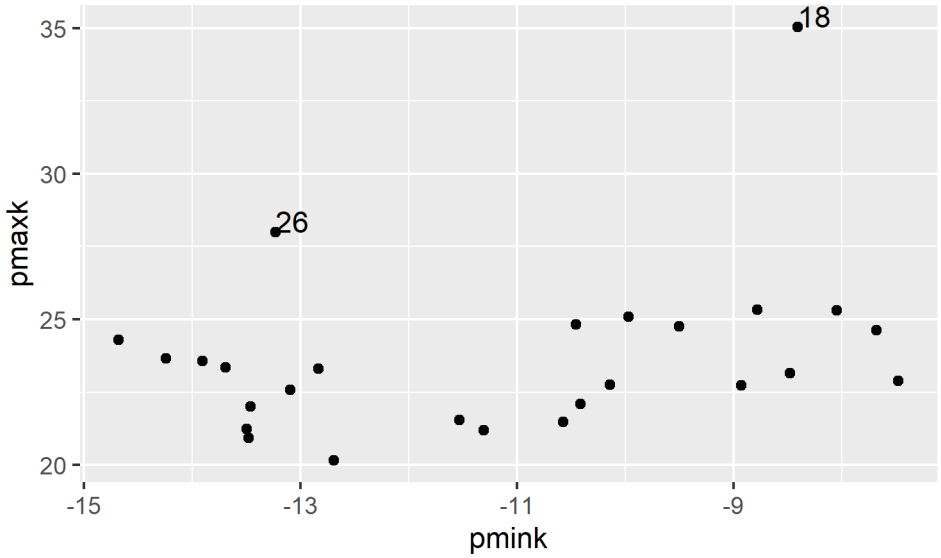


Figura 4: Proyecciones en las direcciones de máxima y mínima curtosis



Vemos en la Figura 4 que al proyectar los datos sobre las direcciones de máxima y mínima curtosis podemos distinguir claramente dos de los outliers (y análoga-

mente proyectando sobre las segundas o terceras direcciones de máxima/mínima curtosis podemos ir detectando el resto visualmente). Se observan dos unidades muestrales muy alejadas del resto, con valor de proyección en la dirección de curtosis máxima mayores que 26. Vemos en el código siguiente que efectivamente estos valores se corresponden con los países 18 y 26.

Código 7: Valores de la proyección en la dirección de máxima curtosis frente a la dirección

```
print(datos[datos$pmaxk > 26, 1:12])

   agric  min  ind eng cons  si  fin serv transcom dist_mah pmaxk  pmink
18  66.8  0.7  7.9 0.1  2.8 5.2  1.1 11.9    3.2    4.249 35.030 -8.408
26  48.7  1.5 16.8 1.1  4.9 6.4 11.3  5.3    4.0    4.314 28.005 -13.228
```

Podemos concluir que existen cuatro países alejados de la media, en los casos de los países 18 y 26 probablemente se deba a que tienen una gran porcentaje de empleos dedicados a la agricultura, en el país 7 por que tiene un alto porcentaje de empleos en industria y servicios industriales respecto a la media y por último, el país 15 se muestra en general moderadamente lejos de todas las medias marginales.

Ejercicio 2

Para este ejercicio hacemos uso de las técnicas de inferencia multivariante asumiendo que si los datos no provienen de una distribución normal, el tamaño muestral es suficientemente grande como para asumir normalidad.

Dado que no tenemos ningún vector μ de medias poblacional con que plantear un test y sacar conclusiones, vamos a plantear distintos tests de igualdad de vector de medias para $\mu \in [0,2] \times [3,5] \times [2,4] = C$. Para esto, planteamos los contrastes:

$$H_0 : \mu_{ijk} = \bar{X}$$
$$H_1 : \mu_{ijk} \neq \bar{X}$$

con $\mu_{ijk} = (i,j,k)$ y escogeremos los valores i,j,k formando particiones de tamaño 0,25 de los intervalos que conforman el cubo C, lo que se conoce como un *linspace*, para formar finalmente una rejilla de valores tridimensional que es una aproximación discreta del cubo C.

A través de los distintos test podremos aproximar una región del espacio R^3 donde está contenido el vector de medias poblacional.

Como sabemos, el estadístico de contraste es

$$T^2_{ijk} = (n-1)(\mu_{ijk} - \bar{X})' S^{-1} (\mu_{ijk} - \bar{X})$$

que puede ponerse segun la F de Snedecor:

$$F_{ijk} = \frac{n-p}{p(n-1)} T^2_{ijk}$$

De manera que en cada test rechazamos la hipótesis nula si $F_{ijk} > F_{(p,n-p)}(\alpha)$ y el pvalor del test viene dado por $\rho_{ijk} = P\{F_{ijk} > F_{(p,n-p)}\}$.

Vemos en el siguiente código como construir el experimento y la visualización tridimensional.

Código 8: Generador de una aproximación de la región C mediante un rejilla, cálculo del p-valor de los test y generación de un gráfico interactivo de la región.

```
saltos = 0.25  # tamano de las particiones

# definimos ahora la rejilla de valores
x_1 = seq(from=0, to=2, by=saltos)
x_2 = seq(from=3, to=5, by=saltos)
x_3 = seq(from=2, to=4, by=saltos)

#distintas lista para almacenar los estadisticos de contraste y pvalores
lista_F = c()
lista_pvalores = c()
lista_T = c()

# matriz que contiene los mu_ijk
medias_pob = matrix(ncol=3)

for(i in x_1){
  for(j in x_2){
        for(k in x_3){

            T2 = (c(i,j,k)-media)%*% %inv(S)%*% %(c(i,j,k)-media)*(n-1)
            lista_T = append(lista_T, T2)

            F = ((n-p)/((n-1)*p))*T2
            pvalor = pf(F, 3, 57, lower.tail=FALSE)

            lista_F = append(lista_F, F)
            lista_pvalores = append(lista_pvalores, pvalor)
            medias_pob = rbind(medias_pob, c(i,j,k))

                }
        }
}

#borramos la primera fila de NA al definir la matriz de resultados
medias_pob = medias_pob[2:(length(x_1)*length(x_2)*length(x_3) +1), 1:3]

# definimos el dataframe resultado de las operaciones
grid = cbind(medias_pob, lista_pvalores)
grid = cbind(grid, lista_T)
grid = cbind(grid, lista_F)

colnames(grid) = c('x_1', 'x_2', 'x_3', 'pvalores', 'T', 'F')
grid = data.frame(grid)

# vemos los primeros 5 autovalores mas altos con sus respectivos
    vectores y estadisticos
print(grid[order(grid$pvalores, decreasing=TRUE),][1:5,])

# creamos la visualizacion con plotly (es interactiva)

fig <- plot_ly(grid, x = ~x_1, y = ~x_2, z = ~x_3,
marker = list(color = ~pvalores, colorscale = c('#FFE1A1', '#683531'),
    showscale = TRUE))

fig <- fig %>% add_markers()

fig
```

```
[1]   x_1  x_2  x_3 pvalores       T         F
       1 4.00 3.00 1.0000000 0.0000000 0.0000000
       1 4.00 2.75 0.8752386 0.7137097 0.2298387
       1 4.00 3.25 0.8752386 0.7137097 0.2298387
       1 3.75 3.00 0.7383743 1.3084677 0.4213710
       1 4.25 3.00 0.7383743 1.3084677 0.4213710
```

Vemos que los pvalores más altos se concentran alrededor del vector de medias muestral $\bar{X} = (1, 4, 3)$, donde el estadístico de T^2 toma valores entorno a 0 y 1. Sin embargo vemos que al variar la componente x_1, esta no aparece como uno de los pvalores más altos, esto se debe a que la matriz de covarianzas

$$S = \begin{pmatrix} 2 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 6 \end{pmatrix}$$

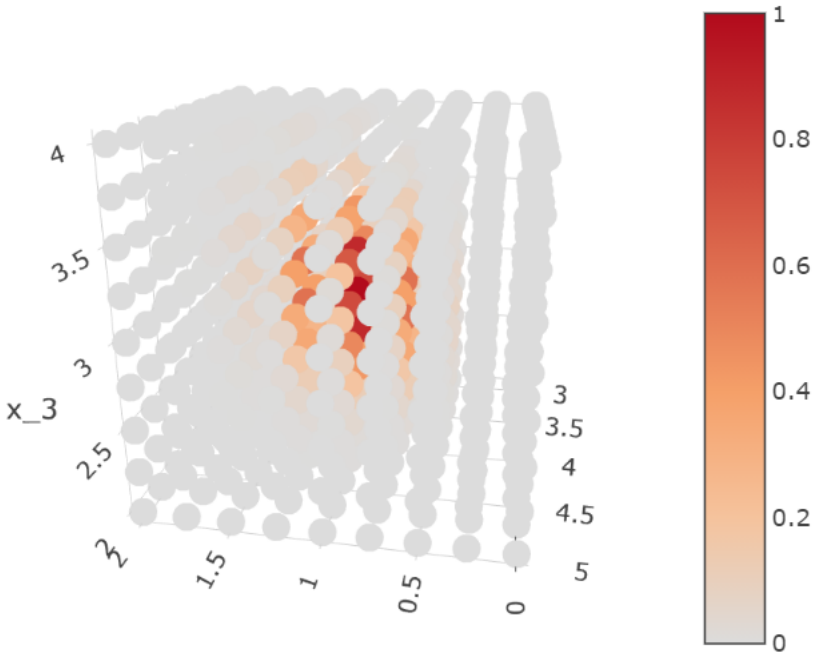
nos dice que la varianza de x_1 es muy baja, por lo que al variar μ en la dirección $(1, 0, 0)$ los pvalores decrecen más rápidamente, de manera que no aparecen en el conjunto de los 5 pvalores más grandes. Visualizamos en la Figura 5 el resultado, donde vemos los posibles valores de μ coloreados por su pvalor durante las iteraciones.

En esencia, lo que deducimos del código 8 y la Figura 5 es que podemos acotar una región del espacio donde los pvalores de los test de igualdad de medias son significativamente altos como para no rechazar que el verdadero valor del vector de medias poblacional μ es μ_{ijk}. Al no tener una generalización natural del concepto de intervalo de confianza en análisis multivariante, esta podría ser una aproximación simple al problema.

Concluimos que el verdadero valor de la media poblacional μ debe encontrarse en una región del espacio *próxima* a $(1, 4, 3)$, admitiéndose que el vector puede variar en mayor medida a lo largo de los ejes x_2 y x_3 que a lo largo de x_1.

NOTA: Es posible que un cubo no sea la mejor figura para intentar construir la *región de confianza*, y que, en un espacio euclidiano, resulte más familiar usar una esfera, pero, por sencillez en la exposición y construcción, he usado un cubo.

Figura 5: Mapa de calor con los distintos valores de μ (μ_{ijk}) en el cubo C y su correspondiente pvalor en los test.



Ejercicio 3

Para este ejercicio vamos a simular 1000 experimentos de observación de $n = 100$ unidades muestrales de una población normal tridimensional con media $\mu = (1, 4, 3)$ y matriz de covarianzas

$$S = \begin{pmatrix} 2 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 6 \end{pmatrix}$$

es decir, usaremos los datos del ejercicio anterior como ejemplo.

Supondremos que existen dos grupos, tal que $n_1 = 60$ y $n_2 = 40$, evidentemente ambos pertenecen a la misma población, pero vamos a ver como se comporta el cociente de verosimilitudes contrastando en cada experimento

$$H_0 : \mu = \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$

En este contraste, una vez calculada la matriz de covarianzas de todos los datos sin agrupar S y la suma de las matrices de covarianzas por grupos, S_w, se puede rechazar H_0 cuando el valor del estadístico de contraste

$$\lambda = nlog\left(\frac{det(S)}{det(S_w)}\right)$$

sea suficientemente grande comparado con el percentil α de una χ^2_g, donde $g = p(G-1)$, siendo p la cantidad de variables y G el número de grupos.

En nuestro caso, $p = 3$ y $G = 2$, luego $g = 3$, así que tras la simulación, los valores de λ obtenidos deberían tener una distribución muy similar a la de una χ^2_3. El código para la simulación es el siguiente.

Código 9: Experimento para estudiar la distribución asintótica del cociente de verosimilitudes

```
library(MASS) # libreria para generar vectores aleatorios
library(ggplot2)

# datos para generar las observaciones

s <- matrix(c(2,2,1,2,5,2,1,2,6),3,3) # covarianzas poblacional
mu = c(1,4,3) # vector de medias poblacional
G = 2 # numero de grupos
n1 = 60; n2 = 40; n = n1+n2 # tamano de muestra y grupos
p = 3 # numero de variables
g = p*(G-1) # grados de libertad chi

lambdas = c() # vector que contiene los estadisticos de los test

for (i in 1:1000){

        # generamos las dos muestras aleatorias
        set.seed(i)
```

```
        grupo1 = mvrnorm(n=n1, mu, s)

        set.seed(1000+i)
        grupo2 = mvrnorm(n=n2, mu, s)

        # muestra total sin agrupar
        obs = rbind(grupo1, grupo2)


        # matrices de covarianzas
        s1 = var(grupo1)*((n1-1)/n1) #grupo 1
        s2 = var(grupo2)*((n2-1)/n2) #grupo 2
        st = var(obs)*((n-1)/n) #covarianza de todas las observaciones

        #calculo de la matriz w

        sw = (s1*n1 + s2*n2)/n

        # calculo del estadistico de contraste
        lambdas[i] = n*log(det(st)/det(sw))


}

# hacemos un grafico de la densidad de los valores y una chi teorica
f = ggplot() +
geom_density(aes(lambdas),color='black')+
stat_function(fun = dchisq, args = list(df = g), color='red')+
labs(x='', y='densidad')
f
```

En la Figura 6 vemos que efectivamente la densidad de los valores del estadístico de contraste (negro) es aproximadamente la misma que la de una χ^2_3 (rojo), como pronosticaba la teoría.

Figura 6: Comparación de la densidad de una χ^2_3 con la obtenida en la simulación