Tarea Análisis Cluster y Análisis Discriminante

Master in Data Science & Bussines Analytics with R

Daniel Silva Gomes de Araújo

01/02/2023

Exercise 1:

- Seleccione una muestra de 1000 clientes para facilitar el coste computacioinal de esta tarea. Si lo desea fije una semilla para garantizar la reproducibilidad de la tarea.

```
## cargar datos
setwd("C:\\Users\\dgoma\\Downloads\\Tarea Clasificación y Discriminación")
rfm_data <- readRDS("rfm_data.RDS")
set.seed(15)
rfm_data <- rfm_data[sample(nrow(rfm_data), 1000), ]
str(rfm_data)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1000 obs. of  4 variables:
##  $ codigo_socio: chr  "id_0180500" "id_0912397" "id_0234108" "id_0765228" ...
##  $ frecuencia  : int  3 1 1 4 1 2 1 2 2 1 ...
##  $ monetario   : num  149.5 29 98.3 139 32.2 ...
##  $ actualidad  : num  941 1019 952 209 421 ...
```

```
# La variable frecuencia ya esta en numeros enteros.
# No es necesario la conversión.
```

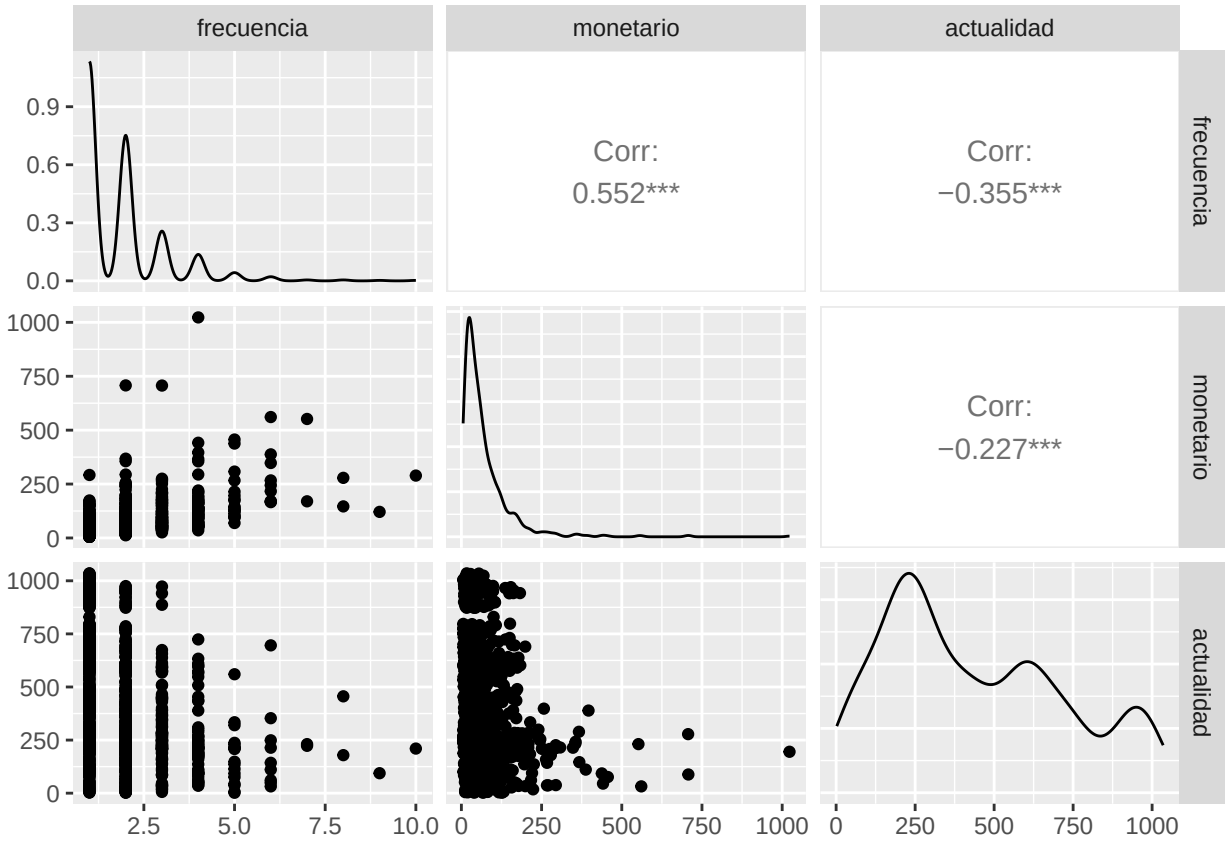
Exercise 2:

- Con muestra de clientes seleccionada realice una análisis exploratorio (EDA) de las variables.

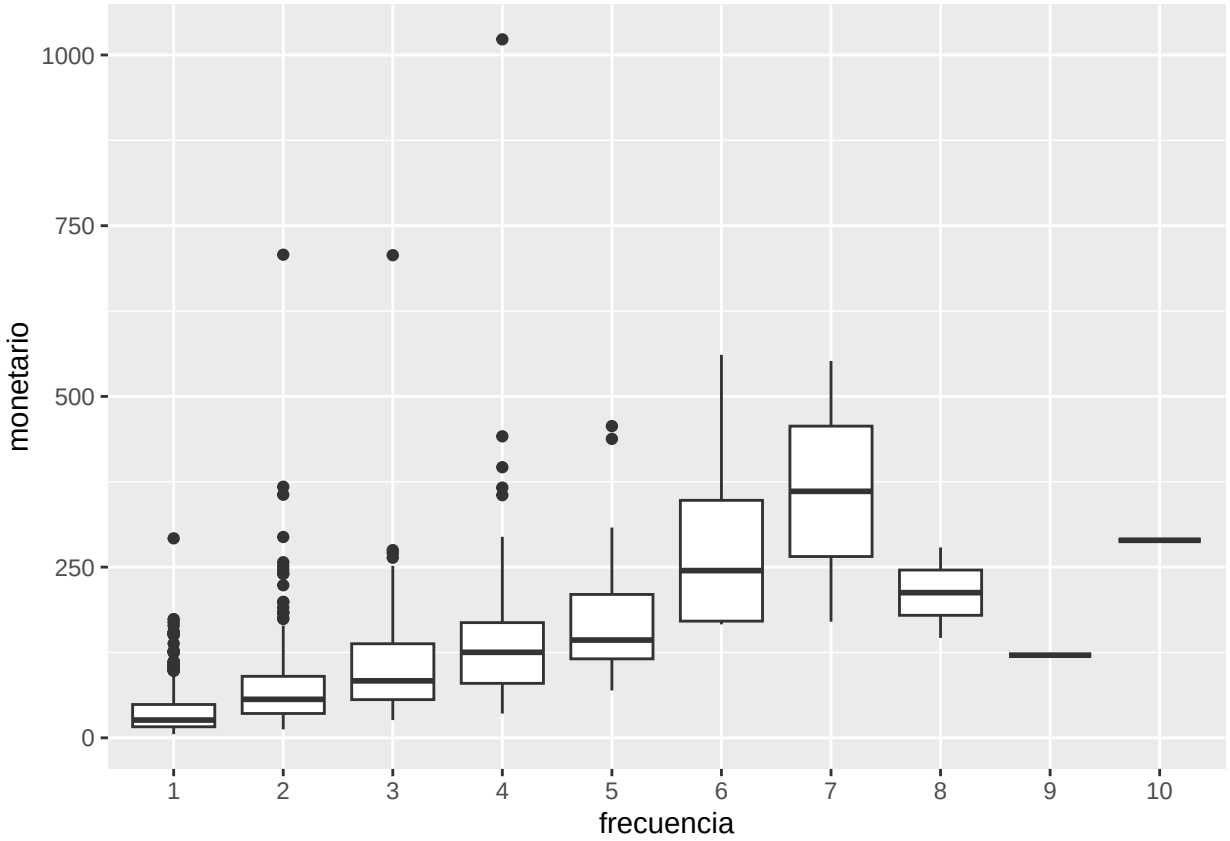
```
## cargar datos y paquetes
setwd("C:\\Users\\dgoma\\Downloads\\Tarea Clasificación y Discriminación")
rfm_data <- readRDS("rfm_data.RDS")
set.seed(15)
rfm_data <- rfm_data[sample(nrow(rfm_data), 1000), ]
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

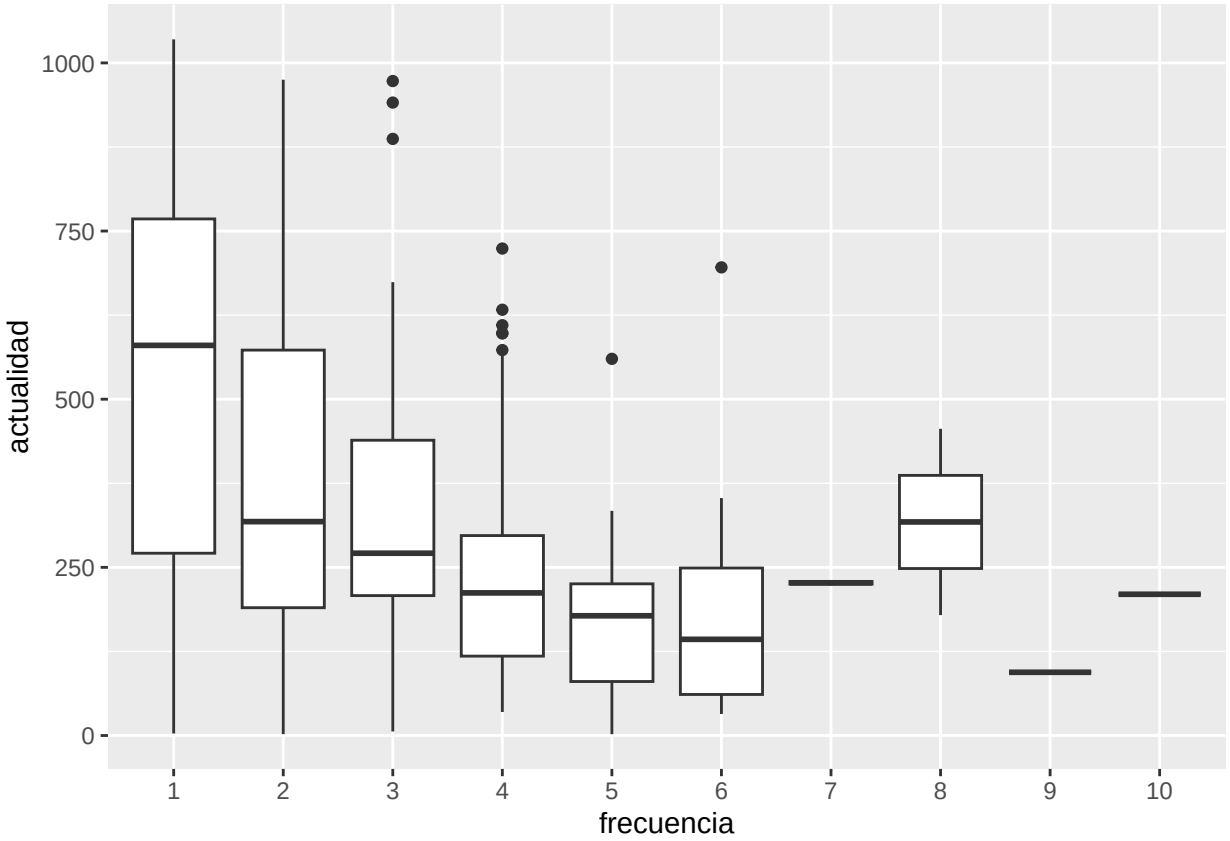
```
library(GGally)

## ggpairs
ggpairs(rfm_data, columns = 2:4)
```



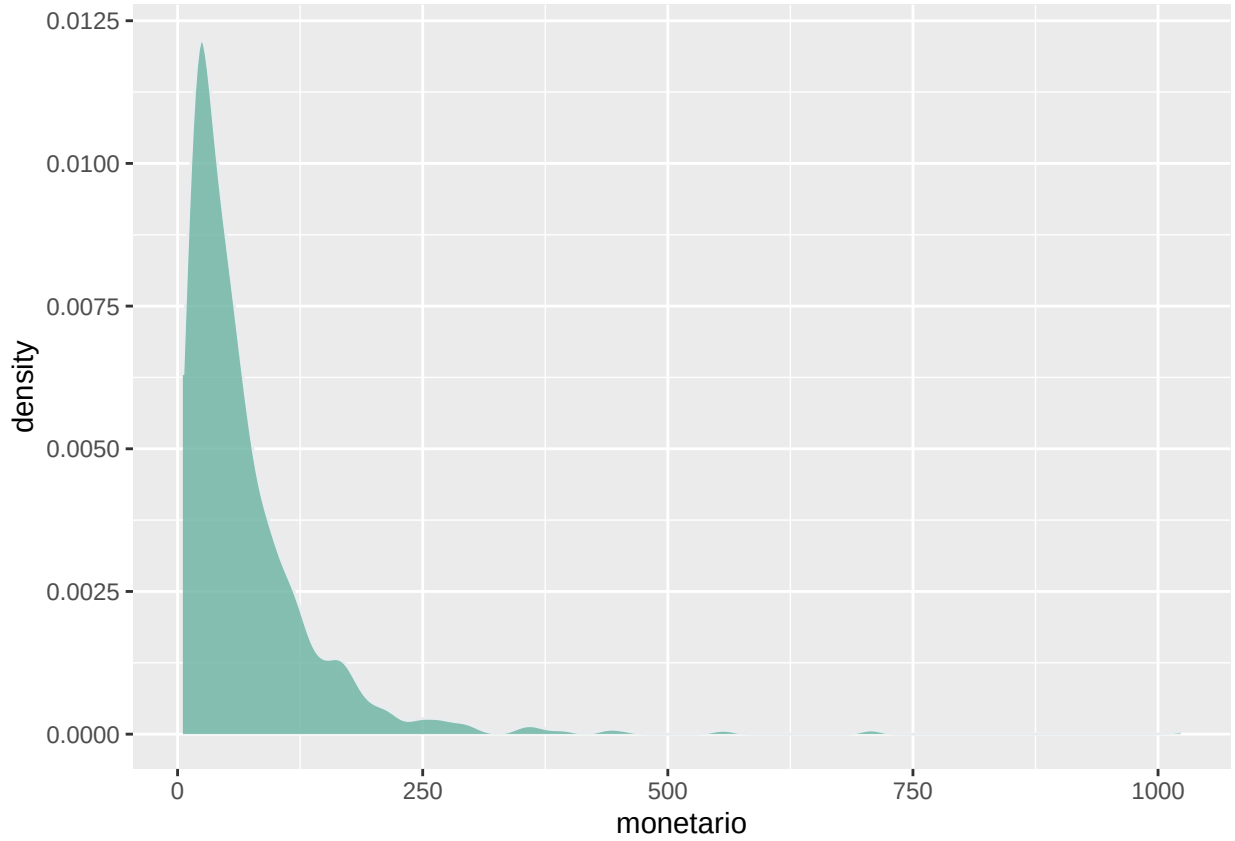

```
# boxplots frequencia-monetario, frecuencia-actualidad
rfm_data$frecuencia <- as.factor(rfm_data$frecuencia)
ggplot(rfm_data, aes(x=frecuencia, y=monetario)) + geom_boxplot()
```



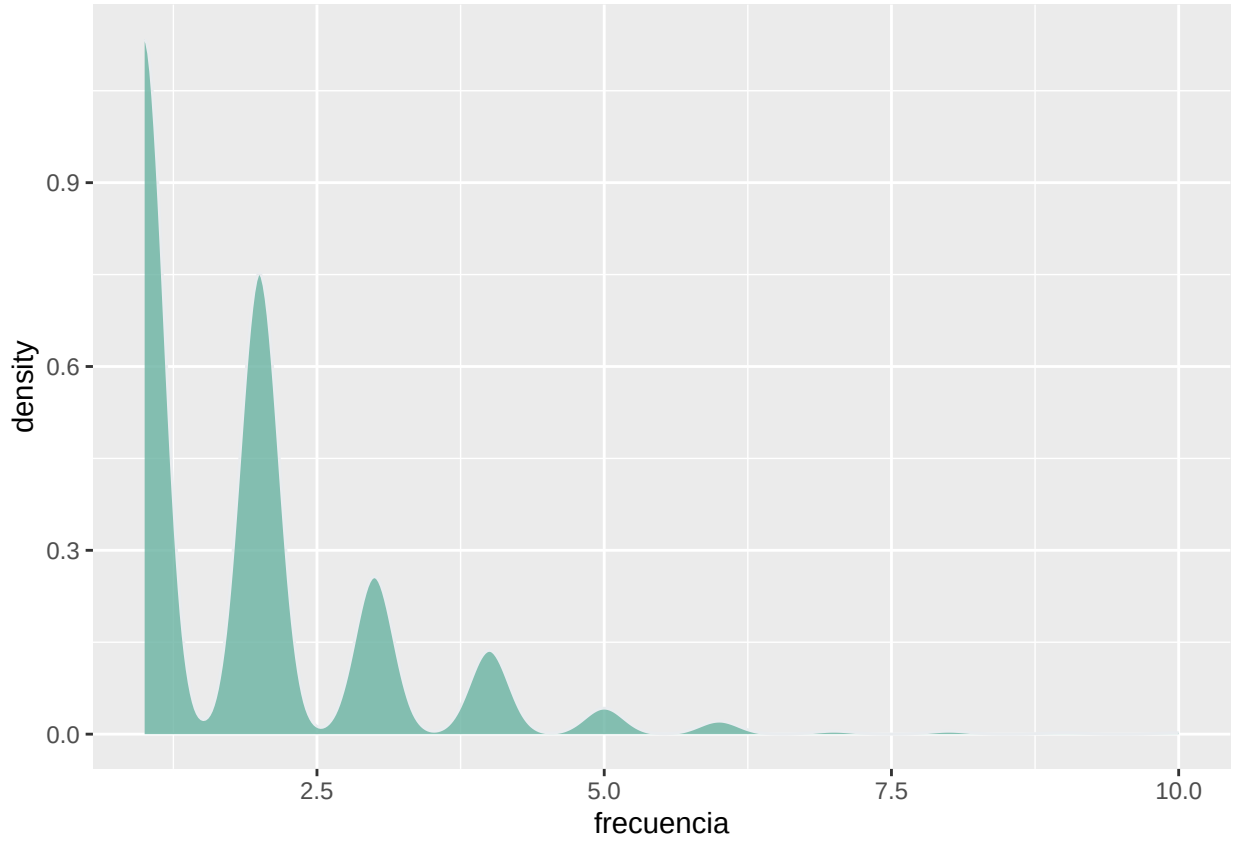

```
ggplot(rfm_data, aes(x=frecuencia, y=actualidad)) + geom_boxplot()
```



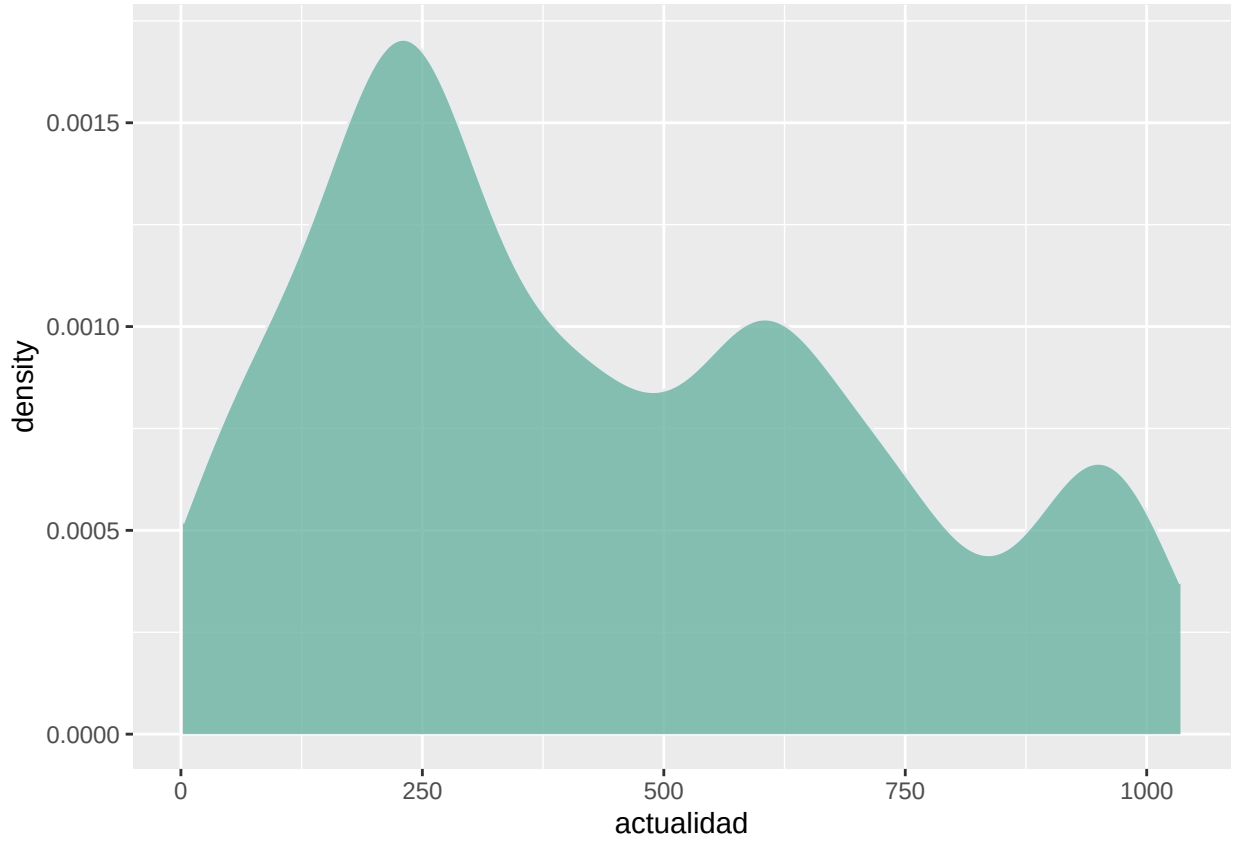

```
rfm_data$frecuencia <- as.numeric(rfm_data$frecuencia)

# densidad monetario frequencia, actualidad
ggplot(rfm_data, aes(x=monetario)) +
    geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8)
```




```
ggplot(rfm_data, aes(x=frecuencia)) +
    geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8)
```




```
ggplot(rfm_data, aes(x=actualidad)) +
    geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8)
```




```
# Para que no tengan mayor ponderación en la distancia aquellas variables
# con mayor variación y para que el ordenamiento de las distancias se man-
# tenga se recomienda tipificar las variables,
```

Exercise 3:

- Para determinar en cuantos grupos puede segmentar a sus clientes en función de las variables propuestas, lleve a cabo, como primera opción, un - análisis jerárquico aglomerativo, utilizando la distancia euclídea y el método de Ward.
- A la luz de los resultados obtenidos, ¿en cuantos grupos dividiría a los clientes?

```
## cargar datos y paquetes
setwd("C:\\Users\\dgoma\\Downloads\\Tarea Clasificación y Discriminación")
rfm_data <- readRDS("rfm_data.RDS")
set.seed(15)
rfm_data <- rfm_data[sample(nrow(rfm_data), 1000), ]
library(factoextra)
```

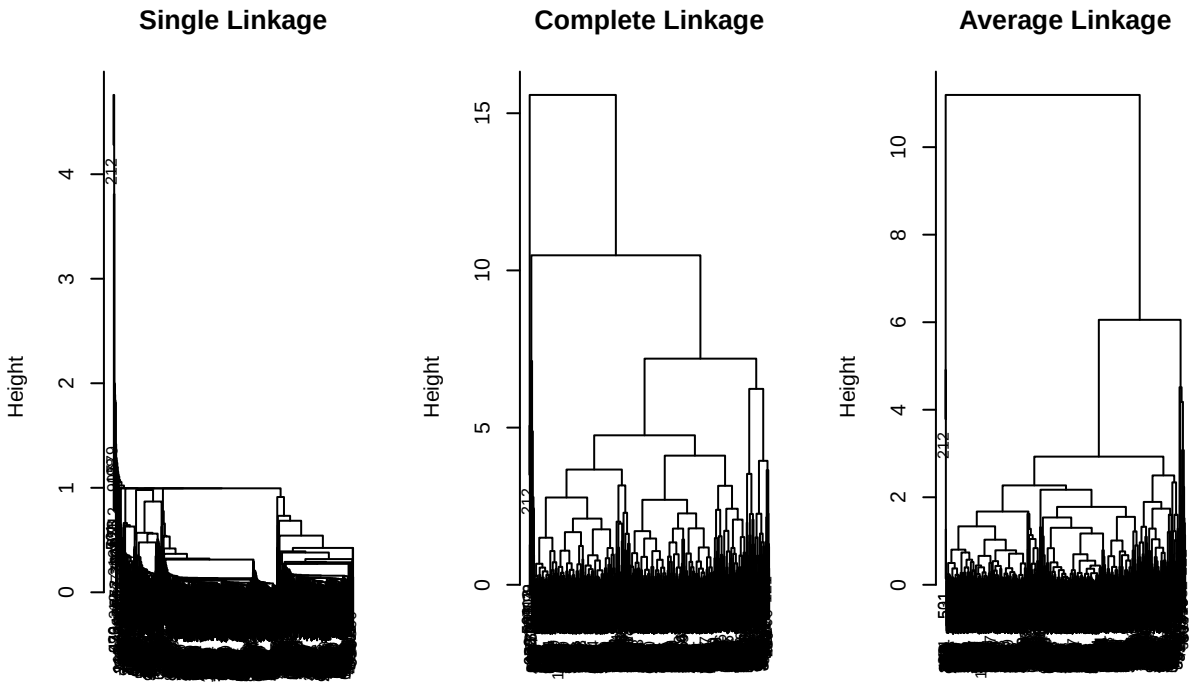
```
## Warning: package 'factoextra' was built under R version 4.2.2
```

```
## nueva tabla con datos tipificados
codigo_socio <- rfm_data$codigo_socio
frecuencia <- scale(rfm_data$frecuencia)
```

```
monetario <- scale(rfm_data$monetario)
actualidad <- scale(rfm_data$actualidad)
rfm_data <- data.frame(codigo_socio, frecuencia, monetario, actualidad)
x <- c("codigo_socio", "frecuencia", "monetario", "actualidad")
colnames(rfm_data) <- x

# distancia euclídea
d.euclidea <- dist(rfm_data, method = "euclidean")
```

```
## Warning in dist(rfm_data, method = "euclidean"): NAs introduzidos por coerção
```

```
hc.single <- hclust(d.euclidea, method = "single")
hc.complete <- hclust(d.euclidea, method = "complete")
hc.average <- hclust(d.euclidea, method = "average")
layout(matrix(1:3, ncol = 3))
plot(hc.single, main = "Single Linkage", sub = "", xlab = "", cex = 0.8)
plot(hc.complete, main = "Complete Linkage", sub = "", xlab = "", cex = 0.8)
plot(hc.average, main = "Average Linkage", sub = "", xlab = "", cex = 0.8)
```

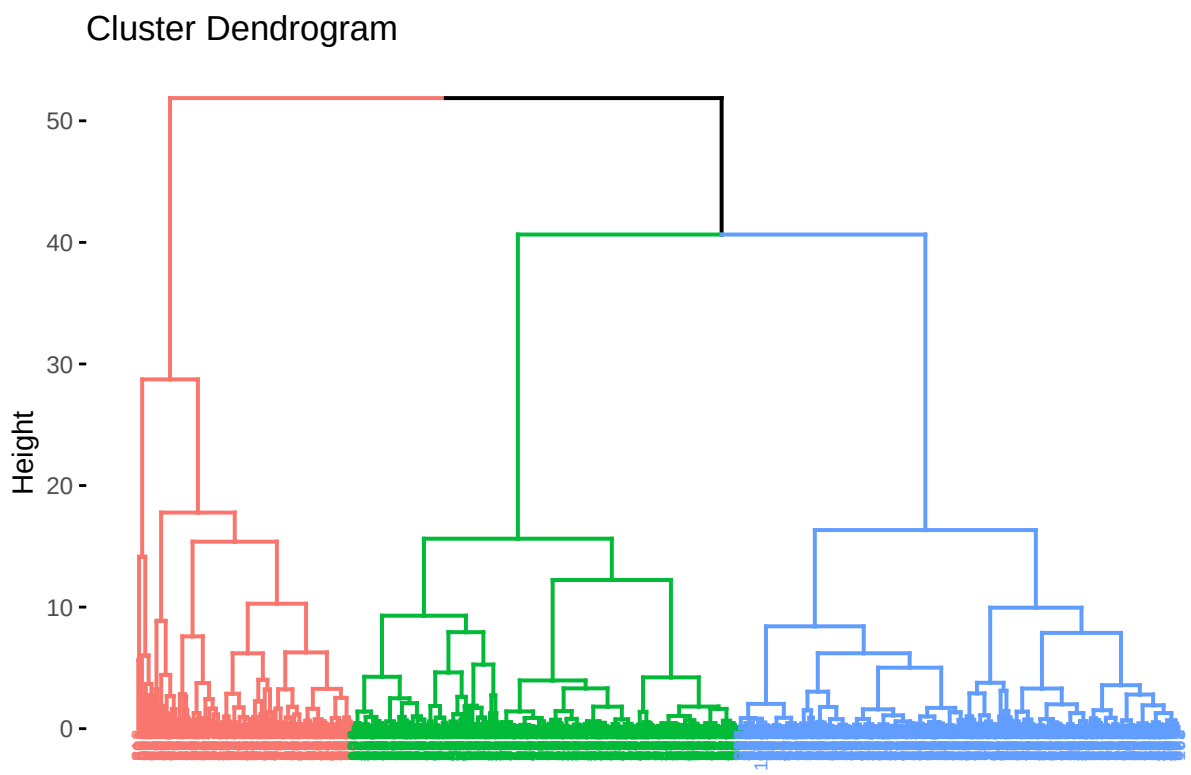



```
# método de Ward
hc.ward <- hcut(rfm_data,k = 3,hc_func = "hclust",hc_metric = "euclidean",
                hc_method = "ward.D2")
```

```
## Warning in stats::dist(x, method = method, ...): NAs introduzidos por coerção
```

```
fviz_dend(hc.ward, cex = 0.5, k = 3, color_labels_by_k = TRUE)
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
```




```
# En los gráficos queda claro que una división en 3 o 4 clusters sería más
# apropiada.
```

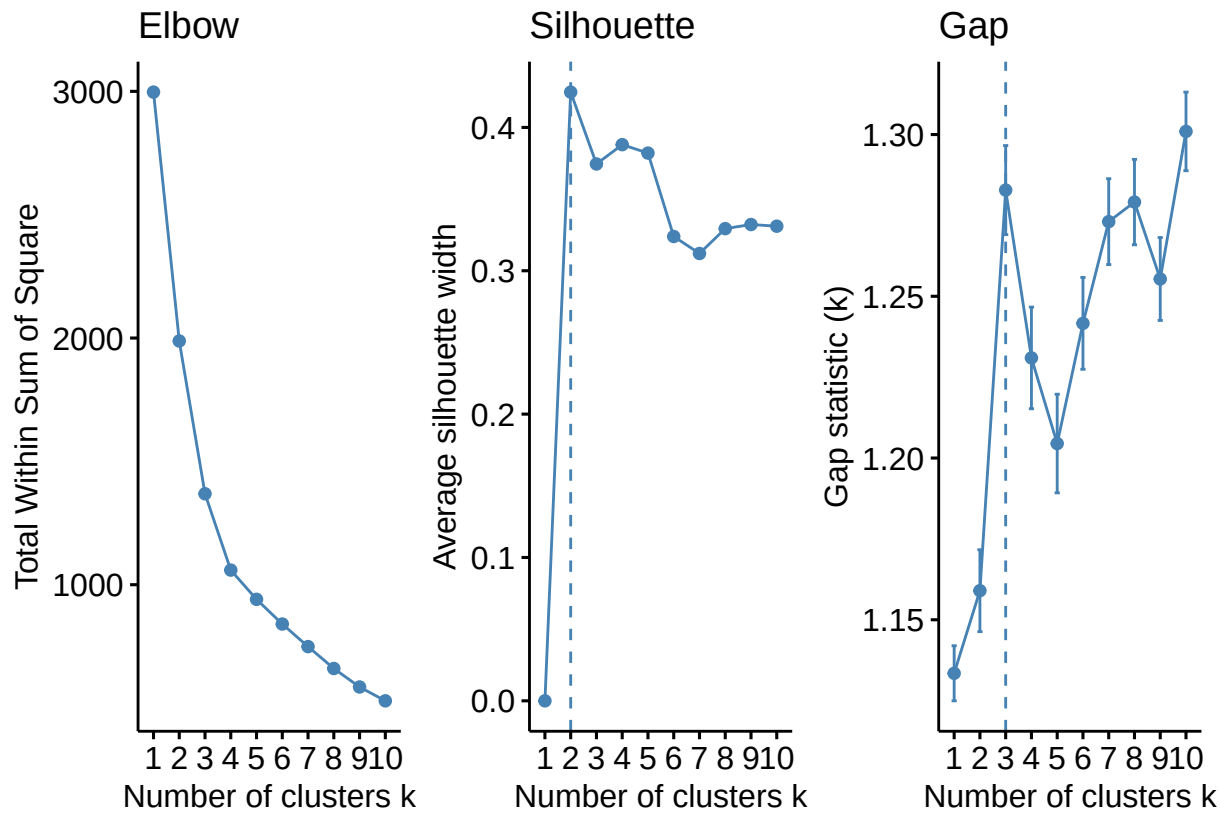
Exercise 4:

- Para la empresa es muy importante el número de segmentos en los que se dividen sus clientes para llevar a cabo acciones publicitarias y poder así incrementar sus beneficios.
- Compare los tres métodos heurísticos estudiados (Elbow, Silouette y GAP) para la determinación del número óptimo de clusters y, especifique, según su criterio, el número de optimo de segmentos.

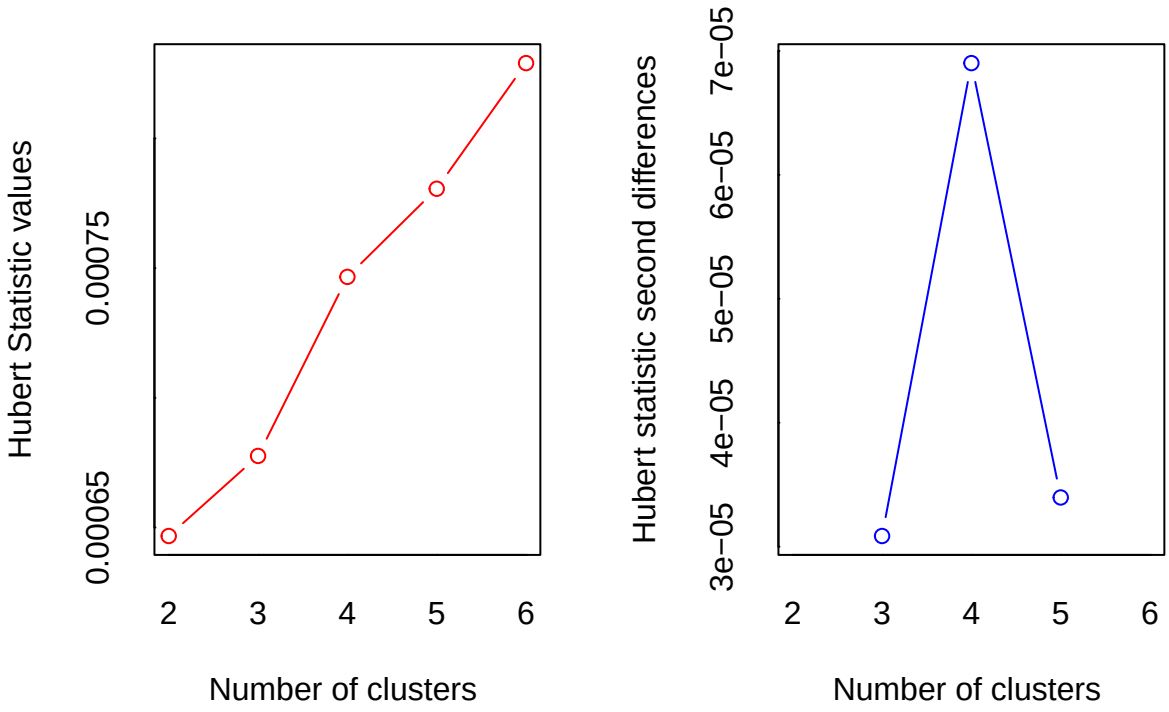
```
## cargar datos y paquetes
setwd("C:\\Users\\dgoma\\Downloads\\Tarea Clasificación y Discriminación")
rfm_data <- readRDS("rfm_data.RDS")
set.seed(15)
rfm_data <- rfm_data[sample(nrow(rfm_data), 1000), ]
library(NbClust)
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.2.2
```

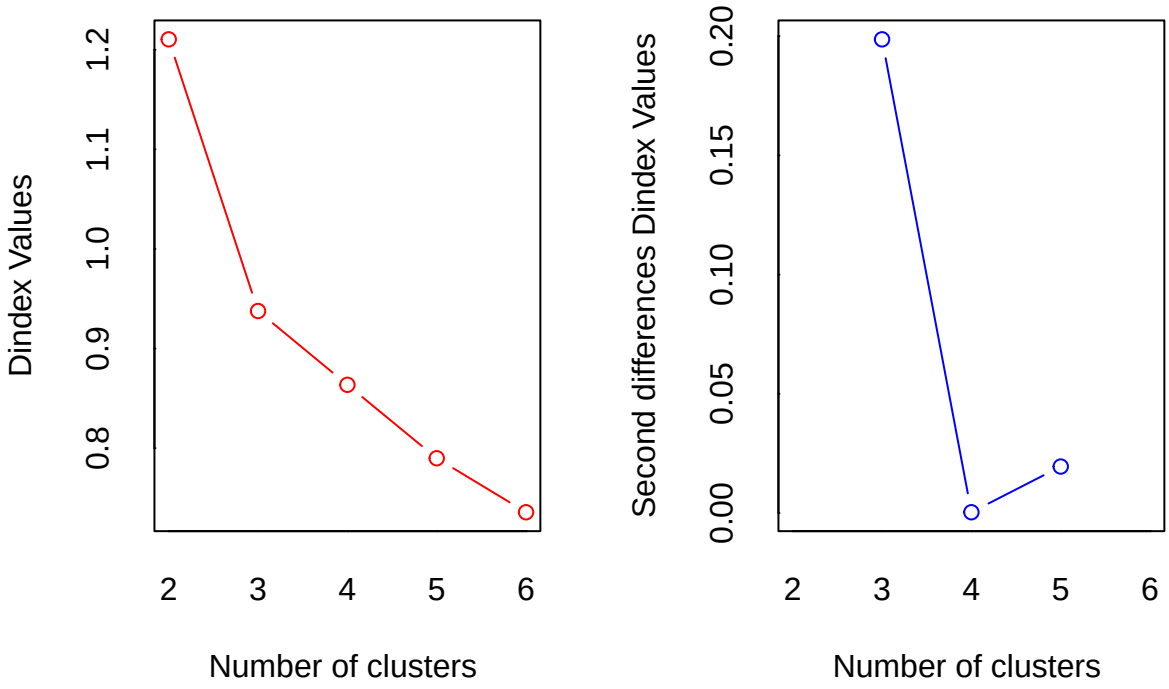
```r
# nueva tabla sin caracteres
frecuencia <- scale(rfm_data$frecuencia)
monetario <- scale(rfm_data$monetario)
actualidad <- scale(rfm_data$actualidad)
rfm_data <- data.frame(frecuencia, monetario, actualidad)
x <- c("frecuencia", "monetario", "actualidad")
colnames(rfm_data) <- x

# metodos Elbow, Silouette y GAP
p1 <- fviz_nbclust(rfm_data,
FUN = hcut, method = "wss",
k.max = 10) +
ggtitle("Elbow")
p2 <- fviz_nbclust(rfm_data,
FUN = hcut, method = "silhouette",
k.max = 10) +
ggtitle("Silhouette")
p3 <- fviz_nbclust(rfm_data,
FUN = hcut, method = "gap_stat",
k.max = 10) +
ggtitle("Gap")
p1 + p2 + p3
```



```r
# El gráfico de sedimentación y el criterio gap presentam un número óptimo
# de 3 clusters. Yo lo dividiría los clientes en 3 grupos.

# NbClust()
NbClust(data = rfm_data, distance = "euclidean", min.nc = 2, max.nc = 6, method = "kmeans", index = "al
```




```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 5 proposed 2 as the best number of clusters
## * 15 proposed 3 as the best number of clusters
## * 2 proposed 4 as the best number of clusters
## * 2 proposed 6 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *******************************************************************

## $All.index
##        KL       CH Hartigan      CCC     Scott   Marriot   TrCovW    TraceW
## 2 0.2532 534.7309 502.3734  -9.3497  974.5005 912192063 485573.0 1951.4228
## 3 1.5266 652.4842 104.5021  -3.8543 2187.0759 610455510 197746.4 1298.0235
## 4 5.1600 514.9006 179.8067 -10.0938 2767.5292 607356685 166076.9 1174.8770
## 5 0.3714 500.3401  78.4482  -8.4311 2983.4096 764730106 119788.3  995.2124
```

```
## 6 0.3334 447.0704 277.8757 -10.6060 3367.8932 749706970 115680.1  922.4817
##   Friedman  Rubin Cindex     DB Silhouette   Duda  Pseudot2   Beale Ratkowsky
## 2   2.1290 1.5358 0.1245 1.2436     0.4404 0.5561  685.7698  1.3574    0.4006
## 3   4.7196 2.3089 0.0954 1.0789     0.3954 1.0465  -21.6006 -0.0755    0.4318
## 4   7.6188 2.5509 0.0893 1.2599     0.2887 3.3186 -168.3799 -1.1852    0.3856
## 5   7.1714 3.0114 0.0805 1.2005     0.3327 2.0699 -188.6610 -0.8746    0.3642
## 6   9.4055 3.2488 0.0768 1.1272     0.3057 2.0917 -173.2751 -0.8786    0.3382
##       Ball Ptbiserial    Frey McClain   Dunn Hubert SDindex Dindex   SDbw
## 2 975.7114     0.5210  1.2498  0.2312 0.0068  6e-04  3.1549 1.2104 1.5818
## 3 432.6745     0.5151  2.6387  0.6883 0.0012  7e-04  2.8126 0.9377 1.2335
## 4 293.7192     0.4327  0.5085  1.1699 0.0038  7e-04  3.2267 0.8636 0.9610
## 5 199.0425     0.4196  0.9599  1.4505 0.0030  8e-04  3.2160 0.7899 0.9945
## 6 153.7470     0.3969 -0.3752  1.7137 0.0023  8e-04  3.2662 0.7357 0.7951
##
## $All.CriticalValues
##   CritValue_Duda CritValue_PseudoT2 Fvalue_Beale
## 2         0.7092           352.2720        0.254
## 3         0.6714           237.8193        1.000
## 4         0.6549           127.0091        1.000
## 5         0.6140           229.4370        1.000
## 6         0.5538           267.4856        1.000
##
## $Best.nc
##                    KL       CH Hartigan     CCC    Scott   Marriot   TrCovW
## Number_clusters 4.00   3.0000   3.0000  3.0000    3.000         3      3.0
## Value_Index     5.16 652.4842 397.8713 -3.8543 1212.575 298637728 287826.6
##                    TraceW Friedman   Rubin Cindex     DB Silhouette   Duda
## Number_clusters   3.0000   4.0000  3.0000 6.0000 3.0000     2.0000 3.0000
## Value_Index     530.2527   2.8992 -0.5311 0.0768 1.0789     0.4404 1.0465
##                  PseudoT2  Beale Ratkowsky     Ball PtBiserial   Frey McClain
## Number_clusters   3.0000 2.0000    3.0000   3.0000      2.000 3.0000  2.0000
## Value_Index     -21.6006 1.3574    0.4318 543.0369      0.521 2.6387  0.2312
##                    Dunn Hubert SDindex Dindex   SDbw
## Number_clusters 2.0000      0  3.0000      0 6.0000
## Value_Index     0.0068      0  2.8126      0 0.7951
##
## $Best.partition
##    [1] 1 1 1 2 3 3 3 3 3 1 3 1 1 1 1 3 3 1 1 3 2 3 3 3 2 1 3 1 3 1 2 1 1 1 3 1 2
##   [38] 3 1 3 1 3 3 3 3 1 1 3 2 2 3 1 3 1 2 1 2 3 3 1 3 1 3 1 3 3 1 3 3 3 3 1 1 3
##   [75] 1 1 2 3 3 3 3 3 1 1 3 2 3 1 3 2 1 3 3 2 3 1 3 2 3 3 1 1 3 1 2 1 1 3 3 3 3
##  [112] 3 3 1 2 3 3 1 2 3 2 3 3 3 1 3 2 3 3 1 1 3 1 3 1 2 2 1 1 1 3 1 3 3 1 3 3 1
##  [149] 3 3 1 2 1 1 3 2 2 3 1 1 3 3 3 3 2 2 3 3 3 3 3 2 3 1 3 3 3 3 2 3 1 3 1 3 3
##  [186] 3 1 1 1 1 1 3 3 1 2 1 3 2 1 3 2 1 1 1 3 3 1 3 3 2 3 2 3 3 1 1 1 1 3 3 2 3
##  [223] 1 3 2 1 1 1 2 1 3 2 1 3 3 3 3 1 2 1 3 3 3 1 3 3 1 3 3 3 3 3 1 2 3 3 3 1 1
##  [260] 2 2 3 3 3 1 1 3 3 3 3 3 1 1 1 3 3 3 3 3 1 1 3 3 1 3 1 1 3 1 3 3 3 3 2 1 3
##  [297] 3 3 1 3 2 3 1 2 3 2 3 3 3 1 3 2 3 2 3 1 1 3 3 3 3 1 3 1 3 2 1 3 2 3 3 1 2
##  [334] 3 3 1 1 3 3 1 3 1 1 1 3 3 3 3 1 3 3 3 2 3 3 3 1 1 3 3 1 1 1 1 2 2 3 1 3 1
##  [371] 3 3 3 3 1 1 3 1 2 1 3 2 1 1 1 3 1 1 1 1 1 3 3 3 1 3 1 1 3 1 3 3 3 1 1 3 3
##  [408] 1 3 2 1 3 2 1 3 1 3 2 3 1 2 1 3 1 1 1 1 2 1 3 2 1 1 3 1 1 3 3 1 1 3 1 3 3
##  [445] 3 3 3 3 3 1 1 3 1 3 3 3 3 3 3 3 3 3 1 3 3 1 3 1 3 3 3 1 1 1 1 3 2 3 2 2 3
##  [482] 1 1 1 3 1 1 2 3 2 3 3 1 3 3 3 1 2 3 3 2 3 3 3 3 1 1 3 3 3 1 2 1 1 3 1 3 1
##  [519] 1 2 3 3 3 3 1 3 3 3 1 3 2 1 3 3 3 1 3 1 3 3 3 1 1 1 3 1 3 1 1 3 1 3 1 1 2
##  [556] 1 1 3 1 1 1 1 1 3 3 1 2 3 3 3 3 1 2 1 3 1 1 2 2 2 3 1 3 1 3 1 3 3 3 2 1 3
##  [593] 1 1 3 3 3 3 3 1 2 1 1 2 1 3 3 1 1 2 2 1 3 3 1 1 2 3 3 3 3 3 3 1 1 1 1 2 3
```

```
##  [630] 3 1 1 3 3 3 3 1 1 2 3 2 1 3 3 3 3 1 1 3 1 1 3 2 3 3 1 3 3 3 1 3 2 3 1 3 1
##  [667] 2 2 3 2 3 1 2 1 1 3 3 1 3 1 3 3 3 3 3 3 3 1 3 3 3 2 1 3 3 1 2 1 3 2 1 3 1
##  [704] 3 2 3 1 1 1 2 3 2 3 1 1 1 1 1 2 1 3 3 3 1 3 3 1 3 1 3 3 1 3 1 1 1 3 1 3 3
##  [741] 1 3 1 2 3 1 1 3 3 1 3 3 3 1 3 2 2 2 3 1 3 2 1 3 2 1 2 3 3 3 3 2 1 3 1 2 3
##  [778] 1 2 2 3 2 1 1 3 1 1 1 3 1 1 3 3 1 1 3 3 1 1 3 1 2 3 1 3 3 3 3 1 3 3 3 3 3
##  [815] 3 1 1 3 3 3 1 3 3 3 1 3 2 1 2 1 1 3 3 1 3 3 3 3 1 2 3 3 3 3 3 3 3 1 2 1 3
##  [852] 3 3 3 2 1 2 3 3 1 2 1 3 1 3 3 1 1 1 1 1 3 3 1 2 1 3 3 1 3 1 3 3 3 1 1 1 3
##  [889] 1 1 3 1 3 3 3 3 3 3 1 3 2 3 3 3 3 1 2 3 1 2 3 3 3 1 2 1 2 1 2 2 3 1 1 1 3
##  [926] 3 3 3 3 1 1 2 3 3 3 1 1 3 3 2 3 3 2 3 1 3 1 3 3 1 1 2 3 3 1 3 1 1 2 1 1 1
##  [963] 3 1 1 1 1 2 3 3 1 2 2 1 3 3 3 3 3 1 3 2 3 1 3 2 1 1 2 1 3 2 3 3 3 1 1 3 3
## [1000] 1
```

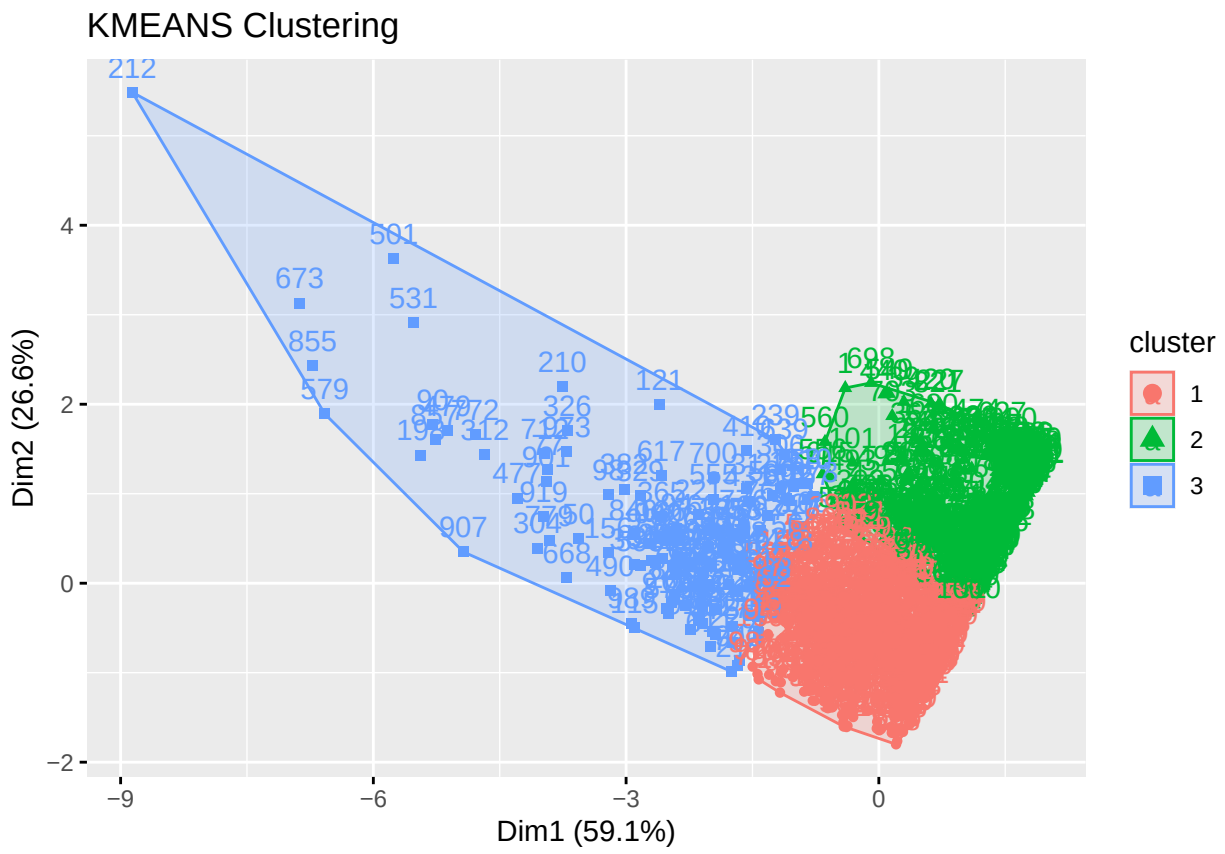
```r
# Metodo k-means: Según la regla de la mayoría, el mejor número de
# clusters es 3, confirmando la conclusión dada previamente.
```

Exercise 5:

• En la empresa están contentos con los resultados que ha obtenido pero desean hacer un k-means, la técnica que vienen utilizando desde hace tiempo para llevar a cabo el modelo RFM. Por ello, le piden que, en base al número de clusters optimo obtenido anteriormente lleve a cabo un k-means.

• Posteriormente, incluya al datastet original (a la muestra de 1000 clientes que seleccionó en el Ejercicio 1) la nueva variable que especifica el grupo al que pertenece cada cliente. Llame a esta variable segmento.

```r
## cargar datos y paquetes
setwd("C:\\Users\\dgoma\\Downloads\\Tarea Clasificación y Discriminación")
rfm_data <- readRDS("rfm_data.RDS")
set.seed(15)
rfm_data <- rfm_data[sample(nrow(rfm_data), 1000), ]
library(factoextra)

## nueva tabla sin caracteres
frecuencia <- scale(rfm_data$frecuencia)
monetario <- scale(rfm_data$monetario)
actualidad <- scale(rfm_data$actualidad)
rfm_data <- data.frame(frecuencia, monetario, actualidad)
x <- c("frecuencia", "monetario", "actualidad")
colnames(rfm_data) <- x

## k-means
kmeans_tic <- eclust(rfm_data, "kmeans", k = 3)
```




```
## nueva tabla con variable segmento
segmento <- kmeans_tic$cluster
frecuencia <- scale(rfm_data$frecuencia)
monetario <- scale(rfm_data$monetario)
actualidad <- scale(rfm_data$actualidad)
rfm_data <- data.frame(frecuencia, monetario, actualidad, segmento)
x <- c("frecuencia", "monetario", "actualidad", "segmento")
colnames(rfm_data) <- x
head(rfm_data)
```

```
##   frecuencia  monetario  actualidad segmento
## 1  0.9725149  1.0154060  1.77261991        2
## 2 -0.7502750 -0.5168230  2.04782433        2
## 3 -0.7502750  0.3639862  1.81143079        2
## 4  1.8339098  0.8811293 -0.81006774        3
## 5 -0.7502750 -0.4761331 -0.06207624        1
## 6  0.1111199 -0.1320491 -0.87004819        1
```

Exercise 6:

- El Marketing Manager quiere analizar los grupos obtenidos y para ello le pide un descriptivo básico de cada grupo en función de las variables (monetario, frecuencia, actualidad). De esta forma, podrá ver en que grupo están los mejores clientes, los que solo compran una vez, los que hace mucho que no compran, los que gastan más dinero, etc.

• Además, le pide que interprete los resultados y asigne un nombre (informativo pero corto) o un acrónimo, a cada segmento obtenido.

• En el departamento de marketing están muy contentos con usted y quieren que siga trabajando como científico de datos. Ahora tienen otro reto para proponerle. Con estos mismos datos, llevar a cabo un análisis discriminante (AD). El objetivo es poder determinar si el segmento o cluster al que pertenecería cada cliente para poder asignarle una campaña de marketing lo más específica posible.

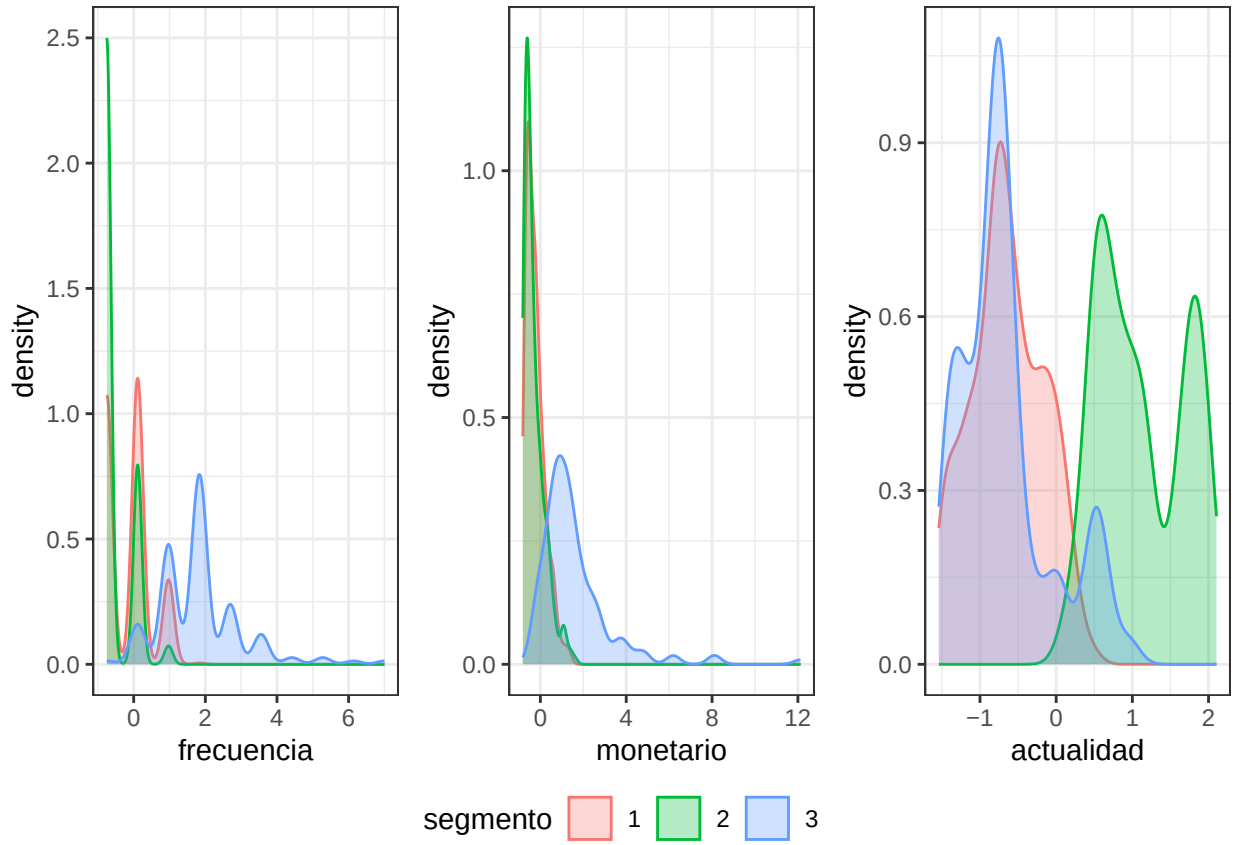
```
## cargar datos y paquetes
setwd("C:\\Users\\dgoma\\Downloads\\Tarea Clasificación y Discriminación")
rfm_data <- readRDS("rfm_data.RDS")
set.seed(15)
rfm_data <- rfm_data[sample(nrow(rfm_data), 1000), ]
library(factoextra)
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.2.2
```

```
library(MVN)
```

```
## Warning: package 'MVN' was built under R version 4.2.2
```

```
## nueva tabla con variable segmento
frecuencia <- scale(rfm_data$frecuencia)
monetario <- scale(rfm_data$monetario)
actualidad <- scale(rfm_data$actualidad)
segmento <- kmeans_tic$cluster
rfm_data <- data.frame(frecuencia, monetario, actualidad, segmento)
x <- c("frecuencia", "monetario", "actualidad", "segmento")
colnames(rfm_data) <- x

## descriptivo basico de la variable segmento
# La variable segmento es un factor
rfm_data$segmento <- as.factor(rfm_data$segmento)
p1 <- ggplot(data = rfm_data, aes(x = frecuencia, fill = segmento, colour = segmento)) +
  geom_density(alpha = 0.3) +
  theme_bw()
p2 <- ggplot(data = rfm_data, aes(x = monetario, fill = segmento, colour = segmento)) +
  geom_density(alpha = 0.3) +
  theme_bw()
p3 <- ggplot(data = rfm_data, aes(x = actualidad, fill = segmento, colour = segmento)) +
  geom_density(alpha = 0.3) +
  theme_bw()
ggarrange(p1, p2, p3, ncol = 3, nrow = 1, common.legend = TRUE, legend = "bottom")
```




```
rfm_data$segmento <- as.numeric(rfm_data$segmento)
# Classificación
# 1 - Clientes regulares (regular)
# 2 - Clientes que llevan más tiempo sin comprar (ocasional)
# 3 - Clientes frecuentes, gastan más dinero (frecuente)

## nueva tabla con los nombres
setwd("C:\\Users\\dgoma\\Downloads\\Tarea Clasificación y Discriminación")
rfm_data <- readRDS("rfm_data.RDS")
set.seed(15)
rfm_data <- rfm_data[sample(nrow(rfm_data), 1000), ]
frecuencia <- rfm_data$frecuencia
monetario <- rfm_data$monetario
actualidad <- rfm_data$actualidad
segmento_nombre <- kmeans_tic$cluster
segmento_nombre[segmento_nombre == "1"] <- "regular"
segmento_nombre[segmento_nombre == "2"] <- "ocasional"
segmento_nombre[segmento_nombre == "3"] <- "frecuente"
segmento_nombre <- as.factor(segmento_nombre)
rfm_data <- data.frame(segmento_nombre, frecuencia, monetario, actualidad)
x <- c("segmento_nombre", "frecuencia", "monetario", "actualidad")
colnames(rfm_data) <- x
head(rfm_data)
```

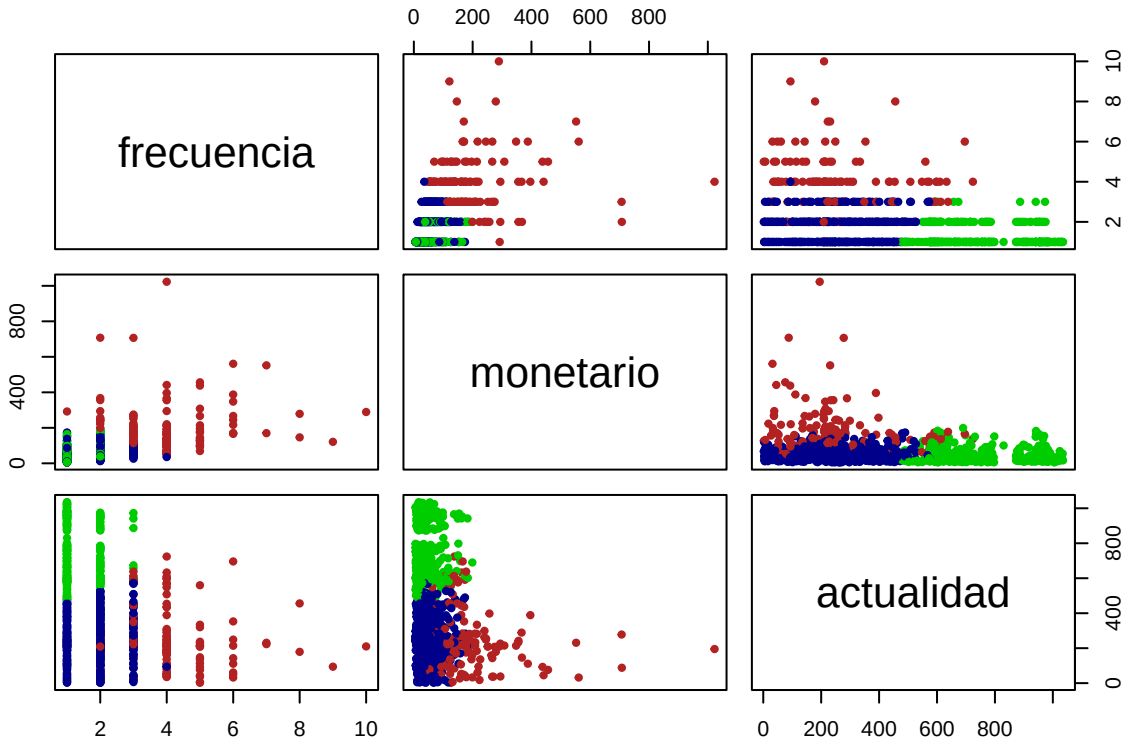
```
##   segmento_nombre frecuencia monetario actualidad
```

```
## 1       ocasional           3    149.52        941
## 2       ocasional           1     29.02       1019
## 3       ocasional           1     98.29        952
## 4       frecuente           4    138.96        209
## 5         regular           1     32.22        421
## 6         regular           2     59.28        192
```

```
str(rfm_data)
```

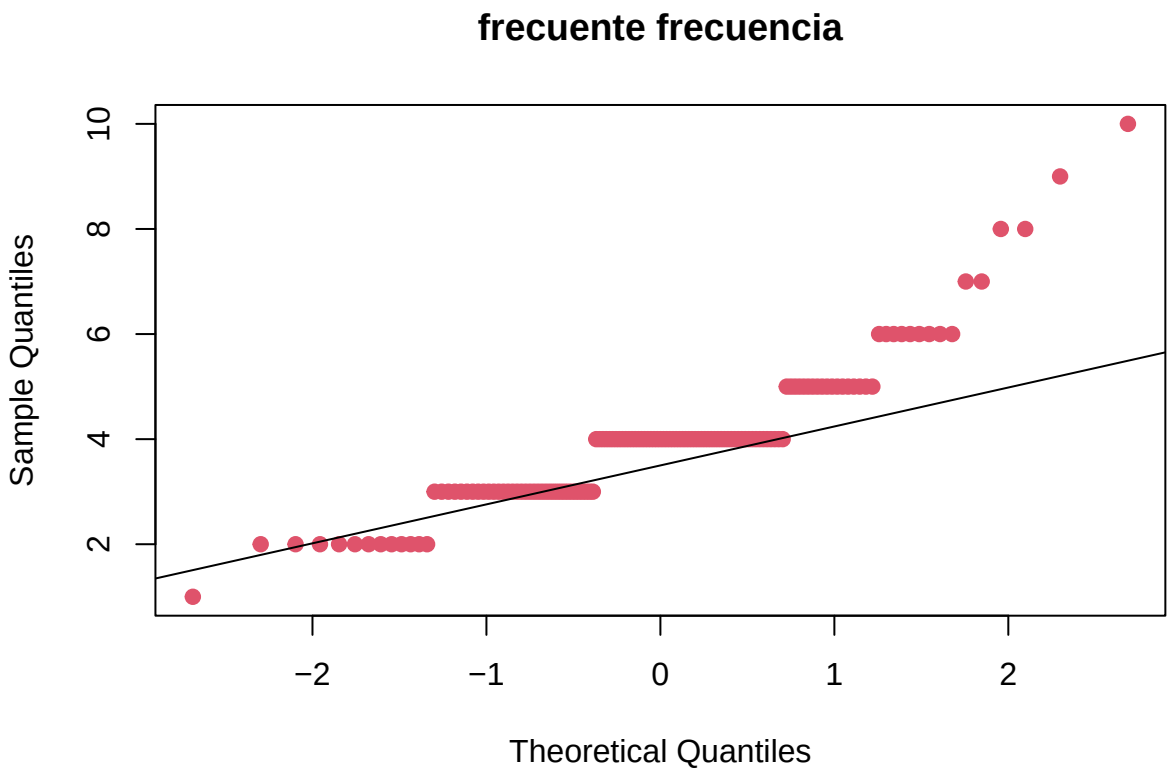
```
## 'data.frame':    1000 obs. of  4 variables:
##  $ segmento_nombre: Factor w/ 3 levels "frecuente","ocasional",..: 2 2 2 1 3 3 3 3 3 2 ...
##  $ frecuencia     : int  3 1 1 4 1 2 1 2 2 1 ...
##  $ monetario      : num  149.5 29 98.3 139 32.2 ...
##  $ actualidad     : num  941 1019 952 209 421 ...
```

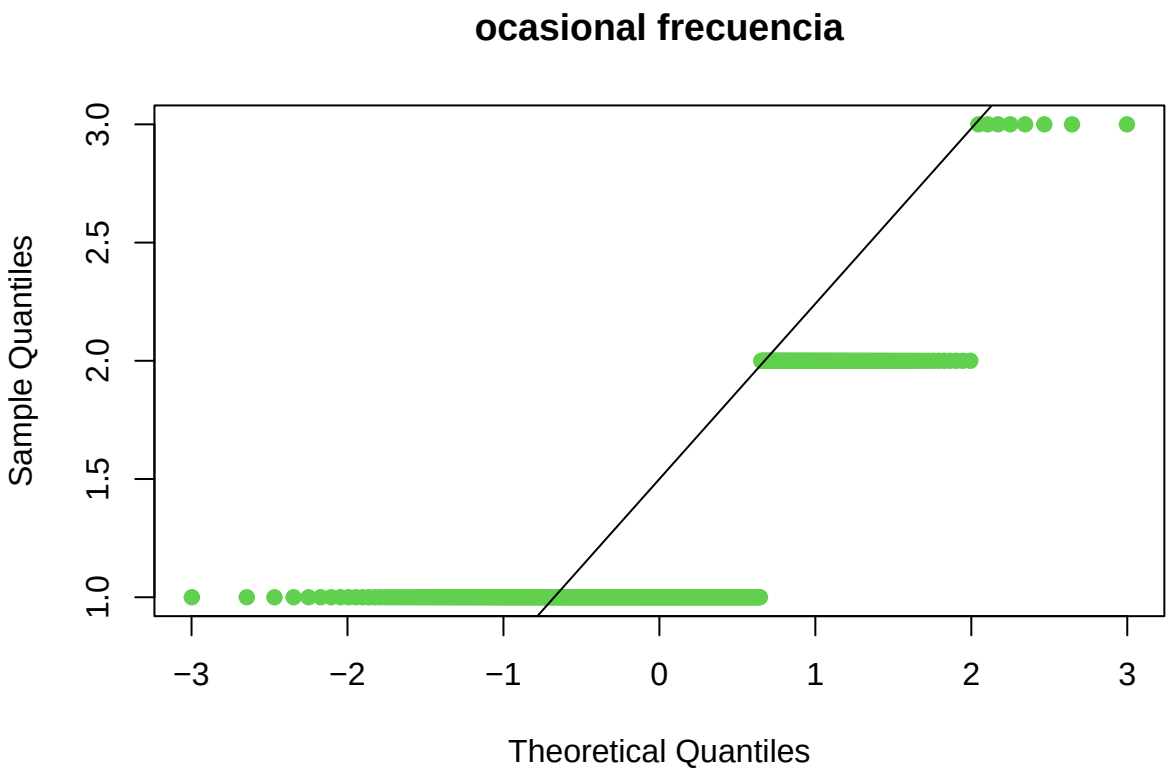
```
## análisis discriminante: diagrama de dispersion
pairs(x = rfm_data[, -1], col = c("firebrick", "green3", "darkblue")[rfm_data$segmento_nombre], pch = 2
```

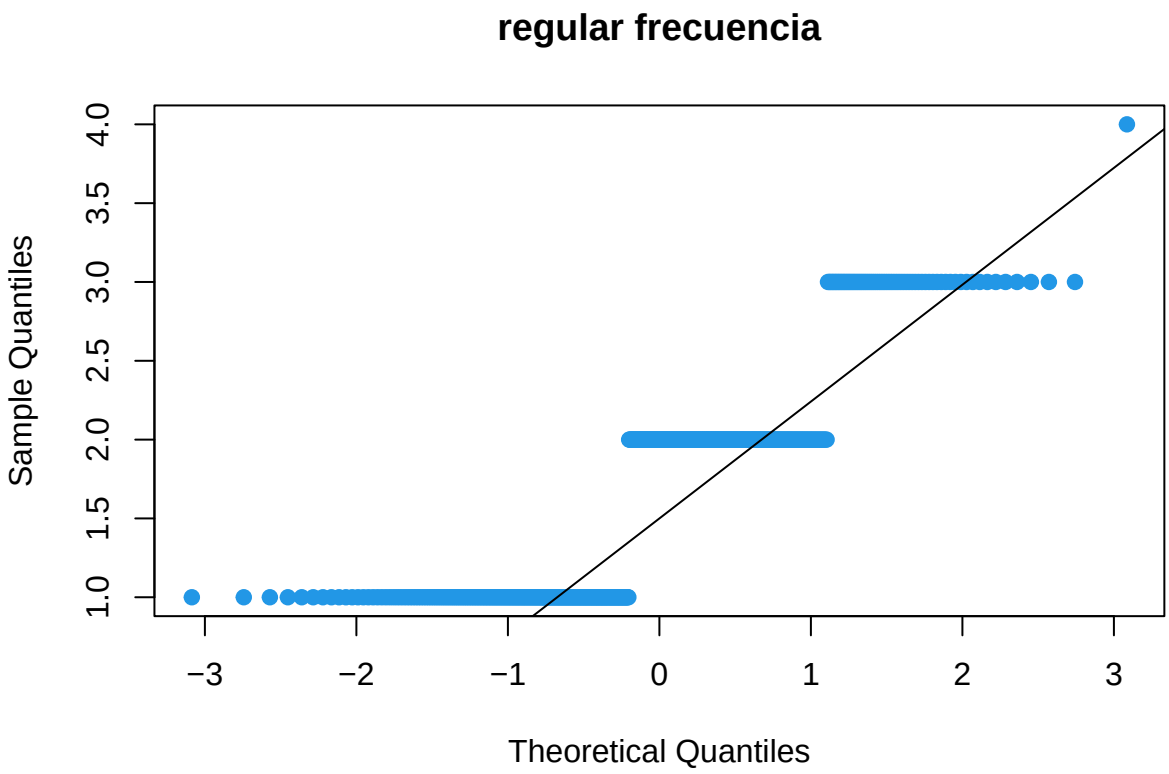


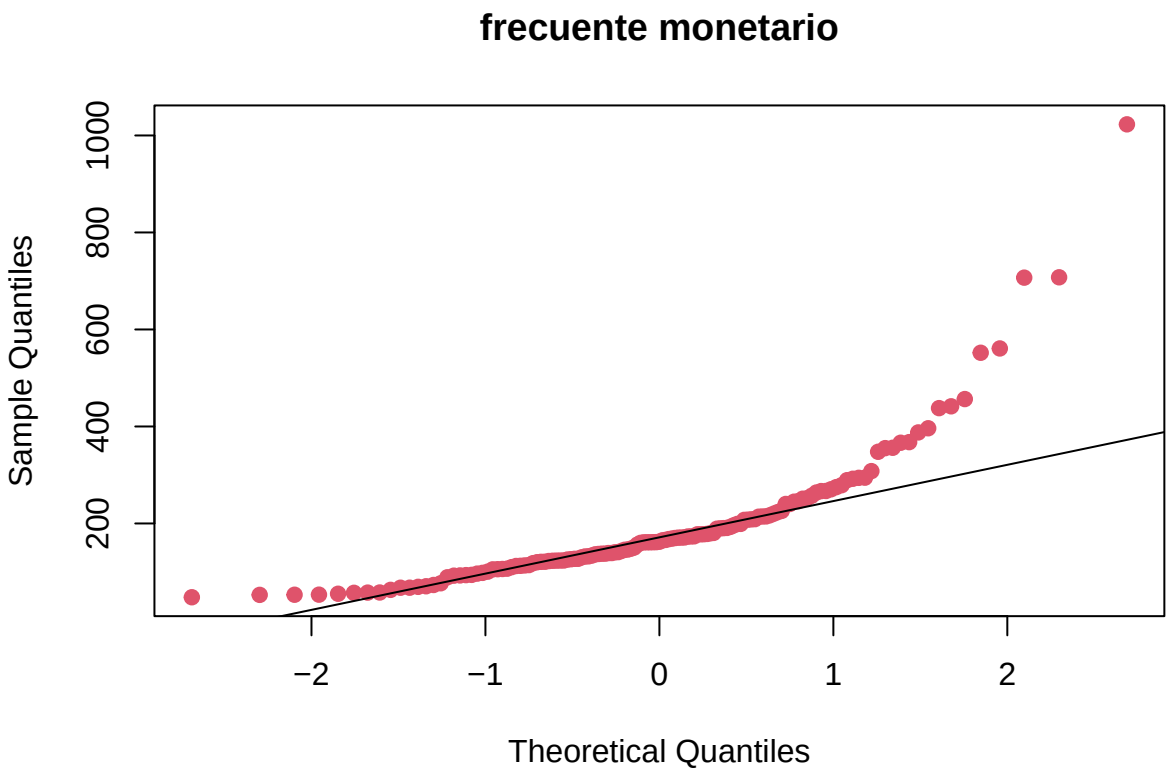

```
## análisis discriminante: q-q plot
for (k in 2:4) {
  j0 <- names(rfm_data)[k]
  x0 <- seq(min(rfm_data[, k]), max(rfm_data[, k]), le = 50)
  for (i in 1:3) {
    i0 <- levels(rfm_data$segmento_nombre)[i]
```

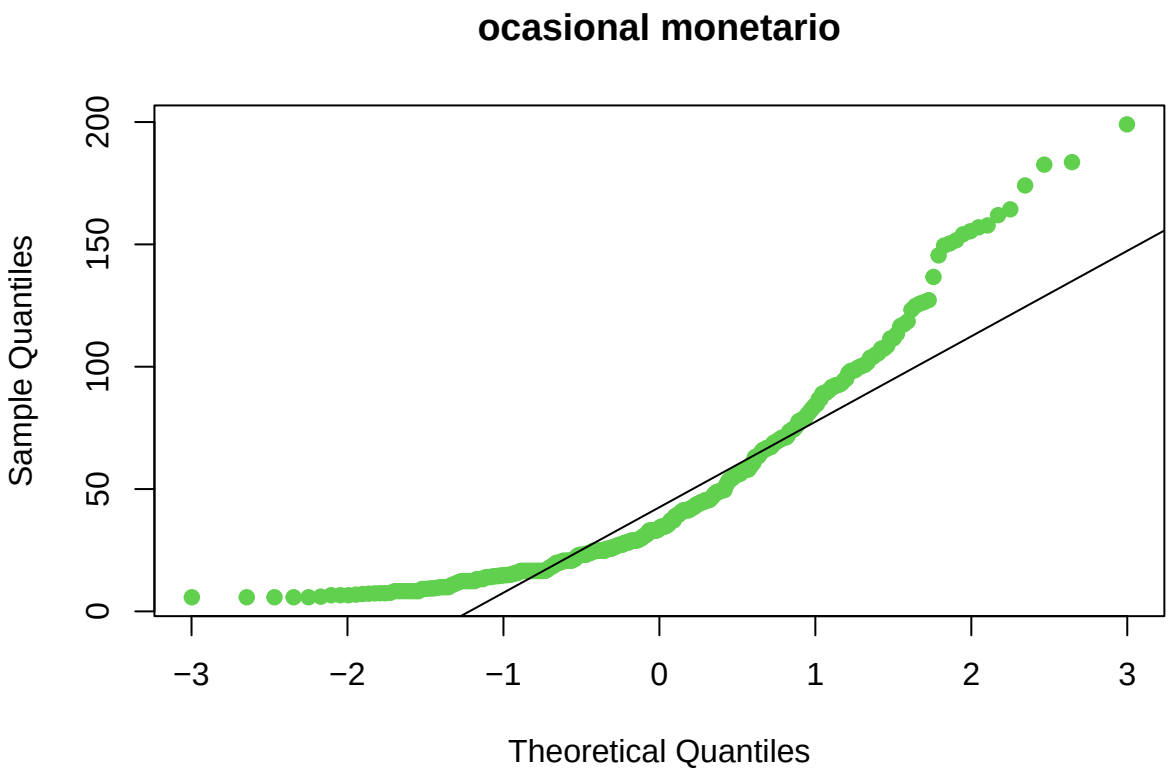
```
    x <- rfm_data[rfm_data$segmento_nombre == i0, j0]
    qqnorm(x, main = paste(i0, j0), pch = 19, col = i + 1)
    qqline(x)}}
```

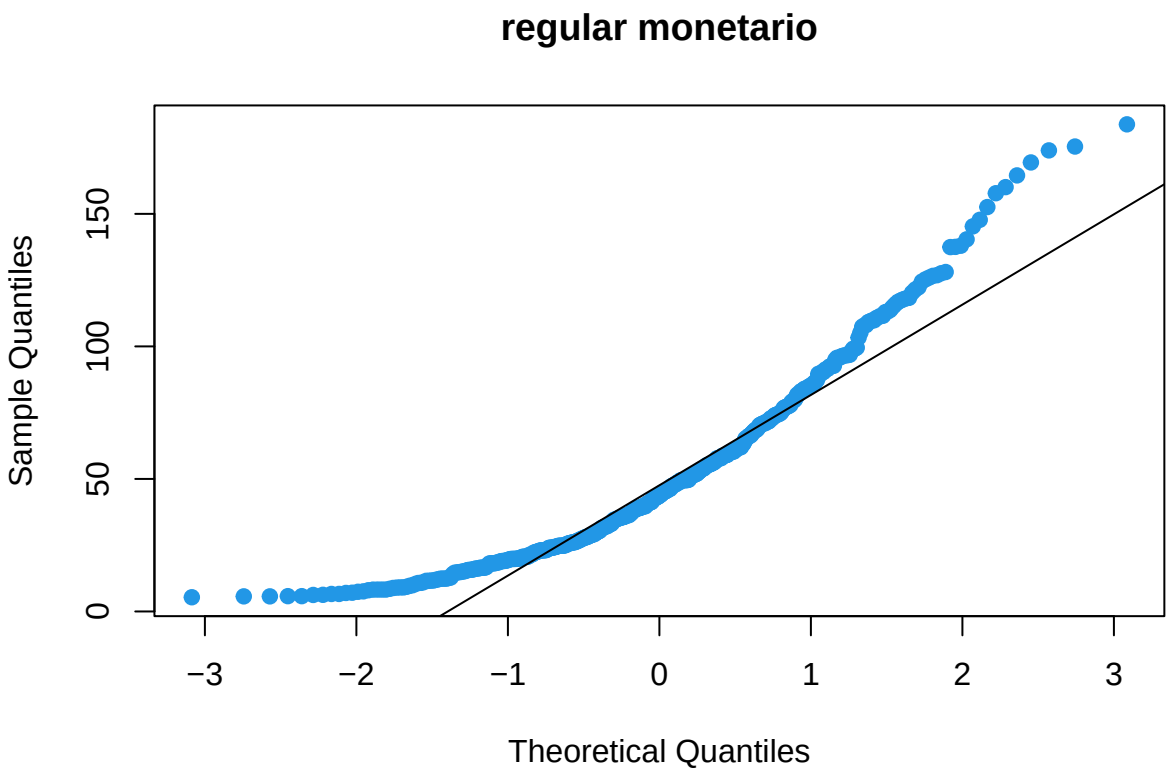


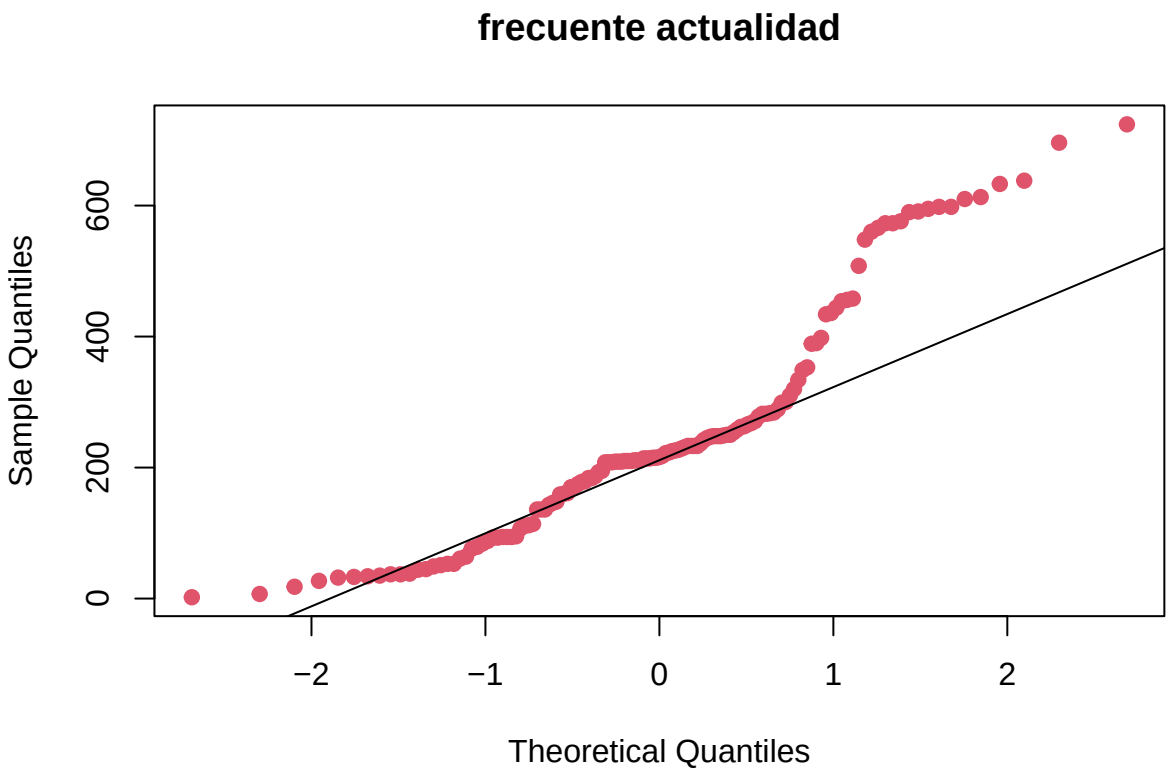

ocasional frecuencia
Sample Quantiles
1.0
1.5
2.0
2.5
3.0
−3
−2
−1
0
1
2
3
Theoretical Quantiles

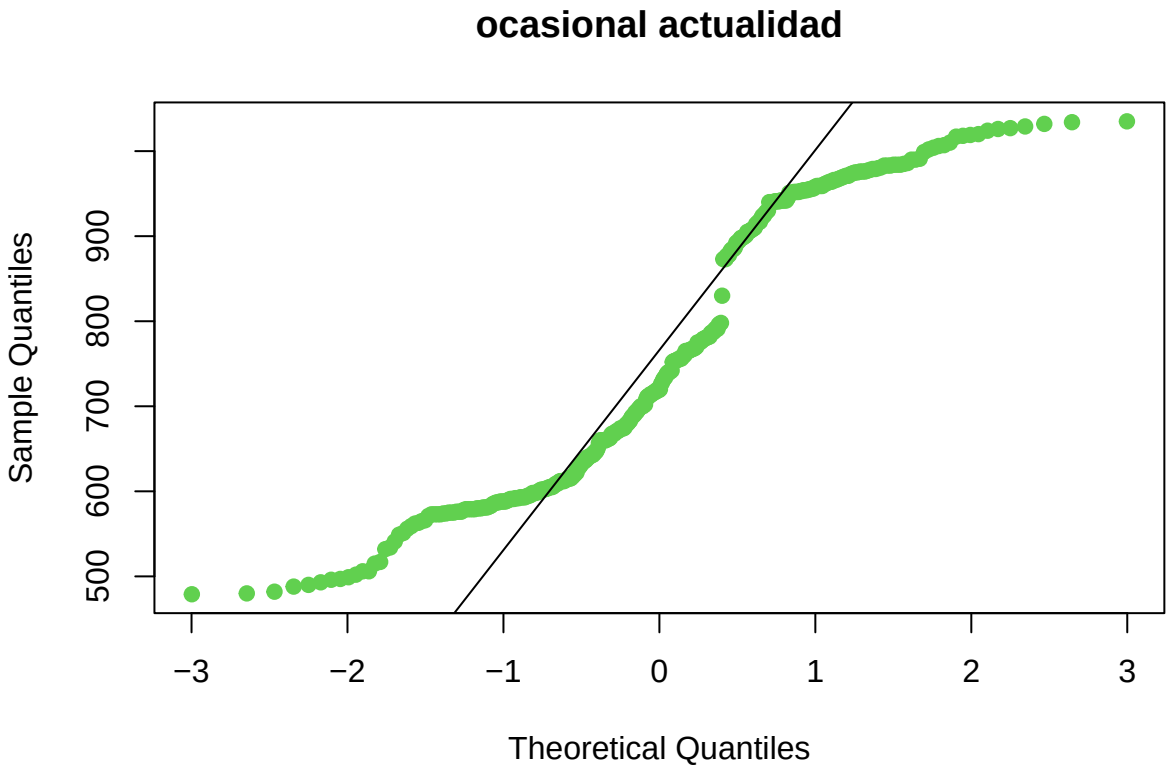

regular frecuencia
Sample Quantiles
1.0
1.5
2.0
2.5
3.0
3.5
4.0
−3
−2
−1
0
1
2
3
Theoretical Quantiles

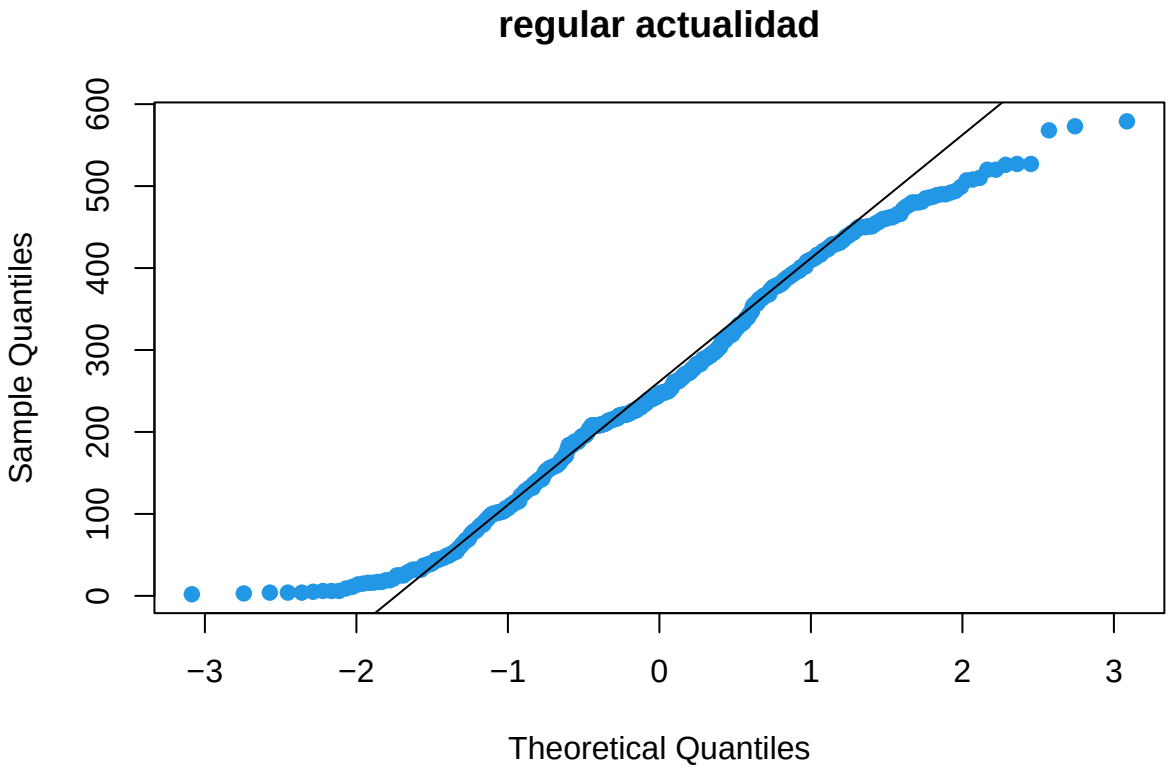

frecuente monetario
Sample Quantiles
200
400
600
800
1000
−2
−1
0
1
2
Theoretical Quantiles


ocasional monetario
Sample Quantiles
0
50
100
150
200
−3
−2
−1
0
1
2
3
Theoretical Quantiles

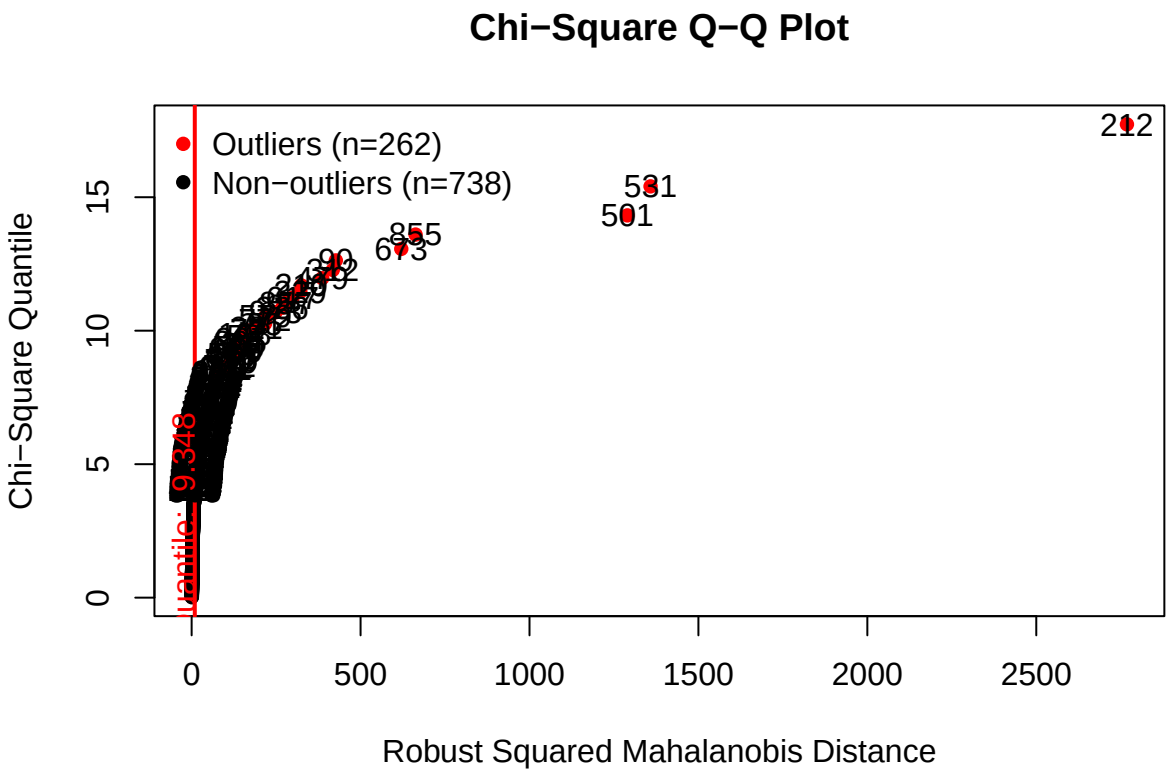

regular monetario
Sample Quantiles
0
50
100
150
−3
−2
−1
0
1
2
3
Theoretical Quantiles


frecuente actualidad
Sample Quantiles
0
200
400
600
−2
−1
0
1
2
Theoretical Quantiles


ocasional actualidad
Sample Quantiles
500
600
700
800
900
−3
−2
−1
0
1
2
3
Theoretical Quantiles




```
## análisis discriminante: estudio de outliers
outliers <- mvn(data = rfm_data[, -1], mvnTest = "hz", multivariateOutlierMethod = "quan")
```

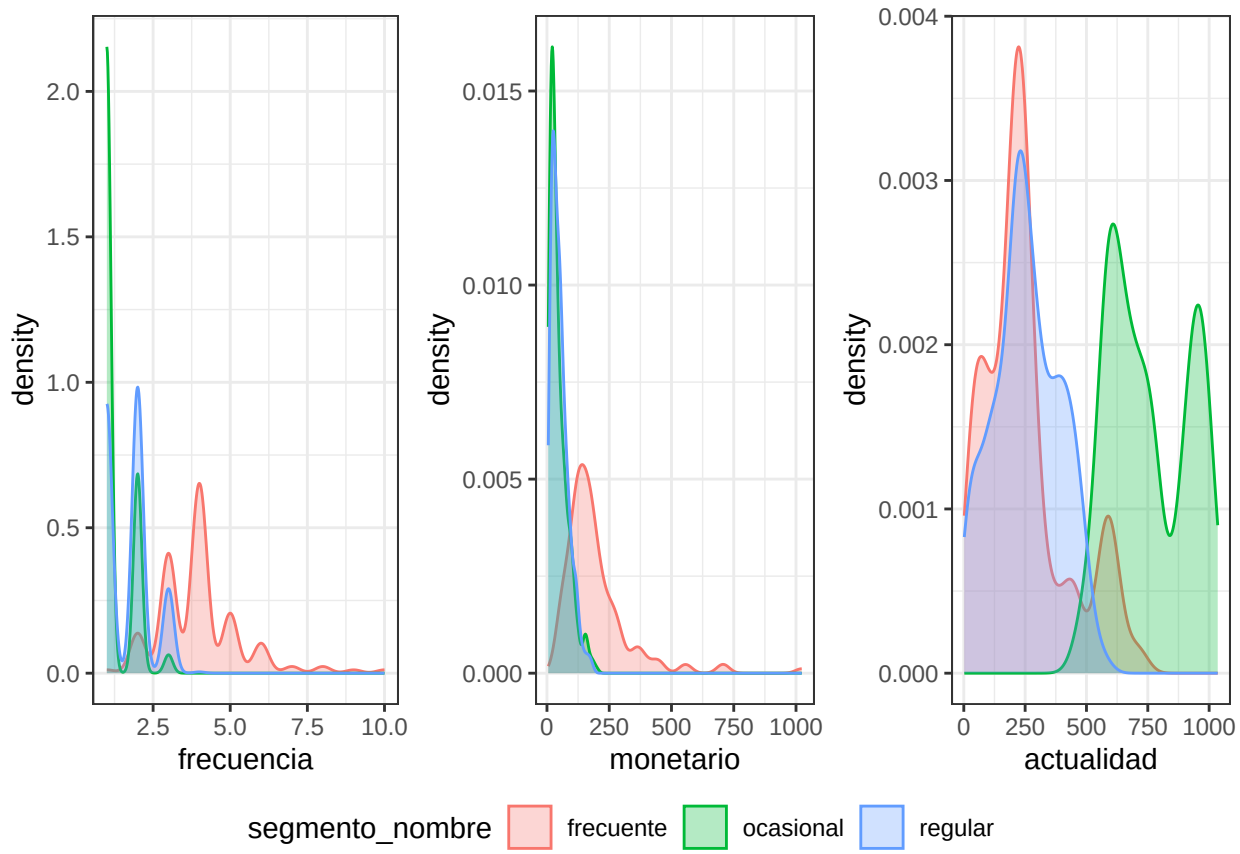


Exercise 7:

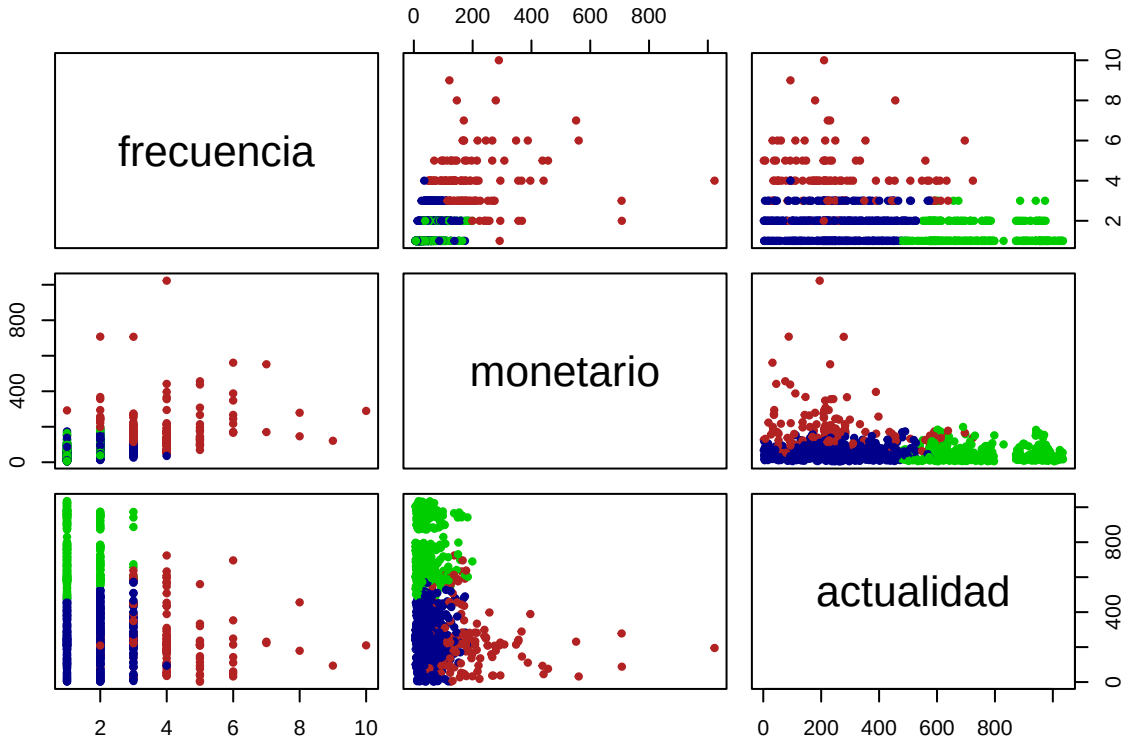
• Realice unos gráficos exploratorios para verificar los supuestos del AD vitos en teoría. Puede ayudarse de: los gráficos de densidad, representando las tres variables por segmento o grupo obtenido en el AC, la matriz de diagramas de dispersión, el análisis de outliers.

```
## cargar datos y paquetes
setwd("C:\\Users\\dgoma\\Downloads\\Tarea Clasificación y Discriminación")
rfm_data <- readRDS("rfm_data.RDS")
set.seed(15)
rfm_data <- rfm_data[sample(nrow(rfm_data), 1000), ]
frecuencia <- rfm_data$frecuencia
monetario <- rfm_data$monetario
actualidad <- rfm_data$actualidad
segmento_nombre <- kmeans_tic$cluster
segmento_nombre[segmento_nombre == "1"] <- "regular"
segmento_nombre[segmento_nombre == "2"] <- "ocasional"
segmento_nombre[segmento_nombre == "3"] <- "frecuente"
segmento_nombre <- as.factor(segmento_nombre)
rfm_data <- data.frame(segmento_nombre, frecuencia, monetario, actualidad)
x <- c("segmento_nombre", "frecuencia", "monetario", "actualidad")
colnames(rfm_data) <- x
library(ggpubr)
library(MVN)
```

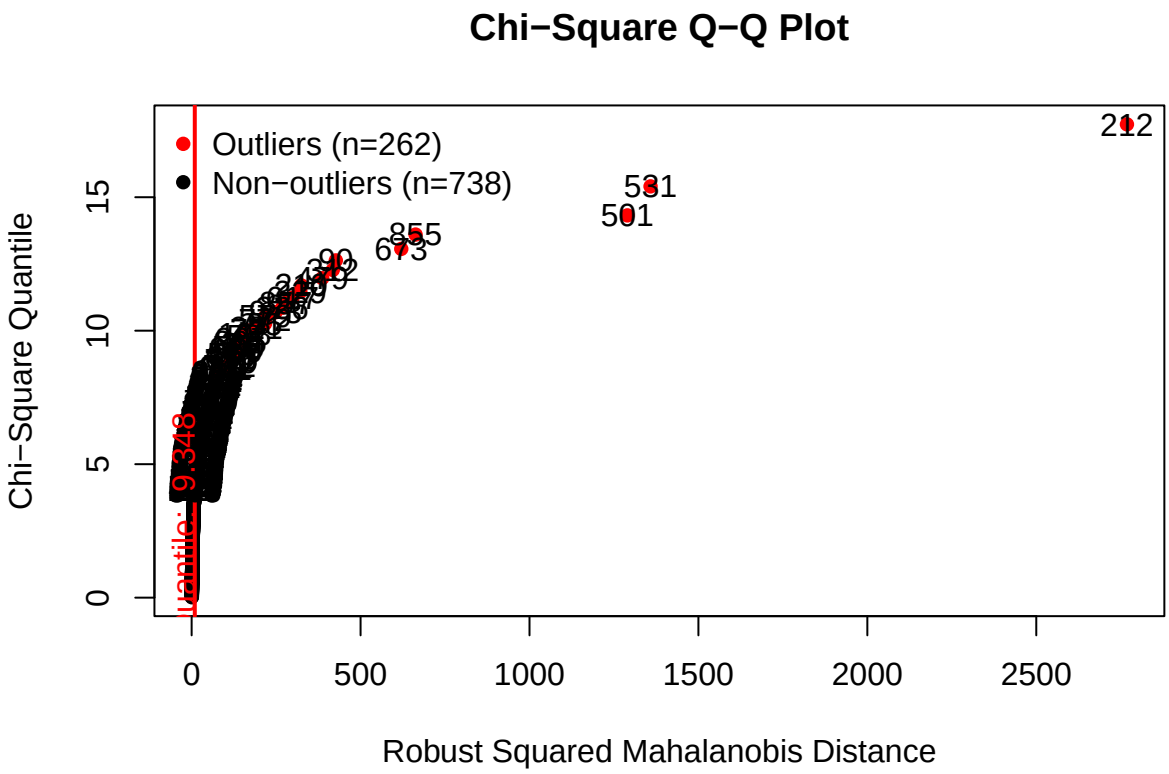
```r
## los gráficos de densidad, representando las tres variables por segmento o grupo obtenido en el AC
p1 <- ggplot(data = rfm_data, aes(x = frecuencia, fill = segmento_nombre, colour = segmento_nombre)) +
  geom_density(alpha = 0.3) +
  theme_bw()
p2 <- ggplot(data = rfm_data, aes(x = monetario, fill = segmento_nombre, colour = segmento_nombre)) +
  geom_density(alpha = 0.3) +
  theme_bw()
p3 <- ggplot(data = rfm_data, aes(x = actualidad, fill = segmento_nombre, colour = segmento_nombre)) +
  geom_density(alpha = 0.3) +
  theme_bw()
ggarrange(p1, p2, p3, ncol = 3, nrow = 1, common.legend = TRUE, legend = "bottom")
```




```r
## la matriz de diagramas de dispersión
pairs(x = rfm_data[, -1], col = c("firebrick", "green3", "darkblue")[rfm_data$segmento_nombre], pch = 2
```




```
## el análisis de outliers
outliers <- mvn(data = rfm_data[, -1], mvnTest = "hz", multivariateOutlierMethod = "quan")
```



Exercise 8:

• Para llevar a cabo el AD y poder comprobar su bondad posteriormente, divida el conjunto de datos en dos. Uno para el entrenamiento (train) de la función lineal discriminante y otro para el estudio de las predicciones (test).

```
## cargar datos y paquetes
setwd("C:\\Users\\dgoma\\Downloads\\Tarea Clasificación y Discriminación")
rfm_data <- readRDS("rfm_data.RDS")
set.seed(15)
rfm_data <- rfm_data[sample(nrow(rfm_data), 1000), ]
frecuencia <- rfm_data$frecuencia
monetario <- rfm_data$monetario
actualidad <- rfm_data$actualidad
segmento_nombre <- kmeans_tic$cluster
segmento_nombre[segmento_nombre == "1"] <- "regular"
segmento_nombre[segmento_nombre == "2"] <- "ocasional"
segmento_nombre[segmento_nombre == "3"] <- "frecuente"
segmento_nombre <- as.factor(segmento_nombre)
rfm_data <- data.frame(segmento_nombre, frecuencia, monetario, actualidad)
x <- c("segmento_nombre", "frecuencia", "monetario", "actualidad")
colnames(rfm_data) <- x
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.2
```

```r
## dividir el conjunto de datos en dos
training_samples <- rfm_data$segmento_nombre |>
  createDataPartition(p = 0.8, list = FALSE) # training (80%) y test (20%)
train_data <- rfm_data[training_samples, ]
test_data <- rfm_data[-training_samples, ]
# Sí, conviene normalizar los datos

## normalización de datos
preproc_param <- train_data |>
  preProcess(method = c("center", "scale"))
train_transformed <- preproc_param |> predict(train_data)
test_transformed <- preproc_param |> predict(test_data)
```

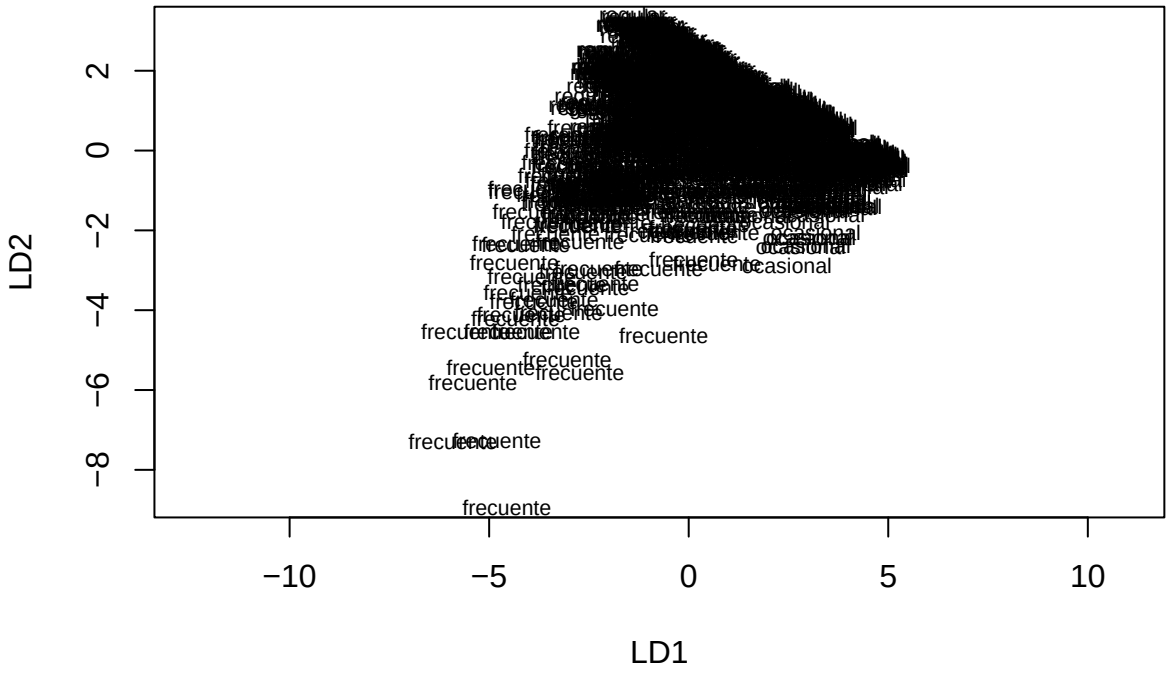
Exercise 9:

- Obtenga la función o funciones lineales discriminantes que mejor separan a sus clientes y represente los resultados gráficamente.

```r
## cargar paquetes
library(MASS)
library(caret)
# library(klaR)

## funciones lineales discriminantes que mejor separan a sus clientes
model_lda <- lda(segmento_nombre ~ frecuencia + monetario + actualidad, data = rfm_data)
model_lda
```

```
## Call:
## lda(segmento_nombre ~ frecuencia + monetario + actualidad, data = rfm_data)
##
## Prior probabilities of groups:
## frecuente ocasional   regular
##     0.139     0.368     0.493
##
## Group means:
##           frecuencia monetario actualidad
## frecuente   3.985612 194.75295   249.2662
## ocasional   1.279891  46.95041   754.7310
## regular     1.716024  51.35174   255.9939
##
## Coefficients of linear discriminants:
##                      LD1          LD2
## frecuencia -0.550321433 -0.821043942
## monetario  -0.002545694 -0.009057447
## actualidad  0.005688989 -0.003585961
##
## Proportion of trace:
##    LD1    LD2
## 0.7219 0.2781
```

```r
# LD1 = (-0.550321433 * frecuencia) + (-0.002545694 * monetario)  + (0.005688989 * actualidad)
# LD2 = (-0.821043942 * frecuencia) + (-0.009057447 * monetario)  + (-0.003585961 * actualidad)

## representación de los resultados gráficamente
plot(model_lda)
```




```r
# partimat(segmento_nombre ~ frecuencia + monetario + actualidad, data = rfm_data, method = "lda", prec
```

Exercise 10:

• Para finalizar y poder presentar sus resultados al Marketing Manager, lleve a cabo una predicción, tanto con el conjunto de train como con el de test y obtenga la matriz de confusión para poder determinar la preción del modelo en ambos conjuntos de datos (train y test).

```r
## cargar datos y paquete
setwd("C:\\Users\\dgoma\\Downloads\\Tarea Clasificación y Discriminación")
rfm_data <- readRDS("rfm_data.RDS")
set.seed(15)
rfm_data <- rfm_data[sample(nrow(rfm_data), 1000), ]
frecuencia <- rfm_data$frecuencia
monetario <- rfm_data$monetario
actualidad <- rfm_data$actualidad
segmento_nombre <- kmeans_tic$cluster
segmento_nombre[segmento_nombre == "1"] <- "regular"
```

```r
segmento_nombre[segmento_nombre == "2"] <- "ocasional"
segmento_nombre[segmento_nombre == "3"] <- "frecuente"
segmento_nombre <- as.factor(segmento_nombre)
rfm_data <- data.frame(segmento_nombre, frecuencia, monetario, actualidad)
x <- c("segmento_nombre", "frecuencia", "monetario", "actualidad")
colnames(rfm_data) <- x
set.seed(15)
training_samples <- rfm_data$segmento_nombre |>
  createDataPartition(p = 0.8, list = FALSE) # training (80%) y test (20%)
train_data <- rfm_data[training_samples, ]
test_data <- rfm_data[-training_samples, ]
preproc_param <- train_data |>
  preProcess(method = c("center", "scale"))
train_transformed <- preproc_param |> predict(train_data)
test_transformed <- preproc_param |> predict(test_data)
model_lda <- lda(segmento_nombre ~ frecuencia + monetario + actualidad, data = rfm_data)
library(caret)
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.2
```

```
## Warning: package 'carData' was built under R version 4.2.2
```

```r
## predicción -  train - matriz de confusión
p1 <- predict(model_lda, train_transformed)$class
tab <- table(Predicted = p1, Actual = train_transformed$segmento_nombre)
tab
```

```
##            Actual
## Predicted   frecuente ocasional regular
##   frecuente         4         0       0
##   ocasional         0         0       0
##   regular         108       295     395
```

```r
sum(diag(tab)) / sum(tab)
```

```
## [1] 0.4975062
```

```r
# Precisión del modelo: 49,75%

## predicción -  test - matriz de confusión
p2 <- predict(model_lda, test_transformed)$class
tab1 <- table(Predicted = p2, Actual = test_transformed$segmento_nombre)
tab1
```

```
##            Actual
## Predicted   frecuente ocasional regular
##   frecuente         0         0       0
##   ocasional         0         0       0
##   regular          27        73      98
```

```r
sum(diag(tab1)) / sum(tab1)
```

```
## [1] 0.4949495
```

```r
# Precisión del modelo: 49,49%
```