Tipología y ciclo de vida de los datos aula 2, Práctica 2: Limpieza y análisis de datos

Jokin Cuesta Arrillaga & Andrés Escamilla Baca

07/06/2022

Índice

---

1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

Este dataset contiene una versión del repositorio del dataset del repositorio UCI con las tarjetas de credito aprobadas en una entidad de crédito.

Este dataset puede ser utilizado para predecir que perfil de clientes pueden tener éxito a la hora de solicitar una tarjeta de crédito.

La variable dicotómica "Approved" puede predecirse con una combinación del resto de variables tales como genero, edad, ingresos,... A continuación se analizará en más detalle cada una de ellas.

2. Integración y selección de los datos de interés a analizar. Puede ser el resultado de adicionar diferentes datasets o una subselección útil de los datos originales, en base al objetivo que se quiera conseguir.

Vamos a hacer la lectura del archivo "CreditCardApprovals.csv" y analizaremos su información más relevante:

```
#LECTURA DEL FICHERO
ds<-read.csv("CreditCardApprovals.csv")
```

```
#OBTENEMOS LAS DIMENSIONES DEL CONJUNTO DE DATOS
dim(ds)
```

```
## [1] 690  16
```

```
#OBTENEMOS LA ESTRUCTURA DEL CONJUNTO DE DATOS
str(ds)
```

```
## 'data.frame':    690 obs. of  16 variables:
##  $ Gender        : int  1 0 0 1 1 1 1 0 1 1 ...
##  $ Age           : num  30.8 58.7 24.5 27.8 20.2 ...
##  $ Debt          : num  0 4.46 0.5 1.54 5.62 ...
##  $ Married       : int  1 1 1 1 1 1 1 1 0 0 ...
##  $ BankCustomer  : int  1 1 1 1 1 1 1 1 0 0 ...
##  $ Industry      : chr  "Industrials" "Materials" "Materials" "Industrials" ...
##  $ Ethnicity     : chr  "White" "Black" "Black" "White" ...
##  $ YearsEmployed : num  1.25 3.04 1.5 3.75 1.71 ...
##  $ PriorDefault  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Employed      : int  1 1 0 1 0 0 0 0 0 0 ...
##  $ CreditScore   : int  1 6 0 5 0 0 0 0 0 0 ...
##  $ DriversLicense: int  0 0 0 1 0 1 1 0 0 1 ...
##  $ Citizen       : chr  "ByBirth" "ByBirth" "ByBirth" "ByBirth" ...
##  $ ZipCode       : int  202 43 280 100 120 360 164 80 180 52 ...
##  $ Income        : int  0 560 824 3 0 0 31285 1349 314 1442 ...
##  $ Approved      : int  1 1 1 1 1 1 1 1 1 1 ...
```

Vemos que tenemos 690 registros con un total de 16 variables. Se describen a continuación cada una de las variables del dataset:

- Gender: género del individuo (0: mujer; 1: hombre)
- Age: edad del individuo.
- Debt: deuda pendiente.
- Married: estado civil (0: soltero o divorciado; 1: casado)
- BankCustomer: ¿Es cliente? (0: no tiene cuenta; 1: sí tiene cuenta)
- Industry: sector donde trabaja
- Ethnicity: etnia del cliente.

- YearsEmployed: años empleado en el trabajo actual.
- PriorDefault: impagos anteriores (0: no tiene impagos anteriores; 1: sí tiene impagos anteriores)
- Employed: ¿stá empleado actualmente? (0: no empleado; 1: sí empleado)
- CreditScore: scoring de crédito
- DriversLicense: ¿tiene carnet de conducir? (0: no tiene carnet; 1: sí tiene carnet)
- Citizen: tipo de ciudadanía
- ZipCode: código postal
- Income: ingresos
- Approved: ¿tarjeta de crédito aprobada? (0: no aprobada; 1: sí aprobada)

```
#OBTENEMOS LOS VALORES RESUMEN DE CADA TIPO DE VARIABLE
summary(ds)
```

```
##      Gender            Age              Debt           Married
##  Min.   :0.0000   Min.   :13.75   Min.   : 0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:22.67   1st Qu.: 1.000   1st Qu.:1.0000
##  Median :1.0000   Median :28.46   Median : 2.750   Median :1.0000
##  Mean   :0.6957   Mean   :31.48   Mean   : 4.759   Mean   :0.7609
##  3rd Qu.:1.0000   3rd Qu.:37.52   3rd Qu.: 7.207   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :80.25   Max.   :28.000   Max.   :1.0000
##                   NA's   :6
##   BankCustomer      Industry          Ethnicity         YearsEmployed
##  Min.   :0.0000   Length:690         Length:690         Min.   : 0.000
##  1st Qu.:1.0000   Class :character   Class :character   1st Qu.: 0.165
##  Median :1.0000   Mode  :character   Mode  :character   Median : 1.000
##  Mean   :0.7638                                        Mean   : 2.233
##  3rd Qu.:1.0000                                        3rd Qu.: 2.710
##  Max.   :1.0000                                        Max.   :28.500
##                                                        NA's   :5
##   PriorDefault       Employed       CreditScore   DriversLicense
##  Min.   :0.0000   Min.   :0.0000   Min.   : 0.0   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.0   1st Qu.:0.000
##  Median :1.0000   Median :0.0000   Median : 0.0   Median :0.000
##  Mean   :0.5232   Mean   :0.4275   Mean   : 2.4   Mean   :0.458
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 3.0   3rd Qu.:1.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :67.0   Max.   :1.000
##
##    Citizen            ZipCode           Income            Approved
##  Length:690         Min.   :   0.0   Min.   :     0.0   Min.   :0.0000
##  Class :character   1st Qu.:  60.0   1st Qu.:     0.0   1st Qu.:0.0000
##  Mode  :character   Median : 160.0   Median :     5.0   Median :0.0000
##                     Mean   : 180.5   Mean   :  1017.4   Mean   :0.4449
##                     3rd Qu.: 272.0   3rd Qu.:   395.5   3rd Qu.:1.0000
##                     Max.   :2000.0   Max.   :100000.0   Max.   :1.0000
##
```

La mayoría de las variables son numéricas (tanto int como float). También tenemos información sobre el sexo y etnia de cada cliente. También tenemos otras variables de tipo categórico. Por último, vemos que

muchas de las variables son de tipo dicotónico con valores 0 y 1 (los valores para cada número los hemos descrito anteriormente).

Por otro lado, vemos algunas variables que lo más seguro es que no las necesitemos y por lo tanto veremos si podría ser viable su eliminación ya que no aportan información de valor al análisis.

3. Limpieza de los datos.

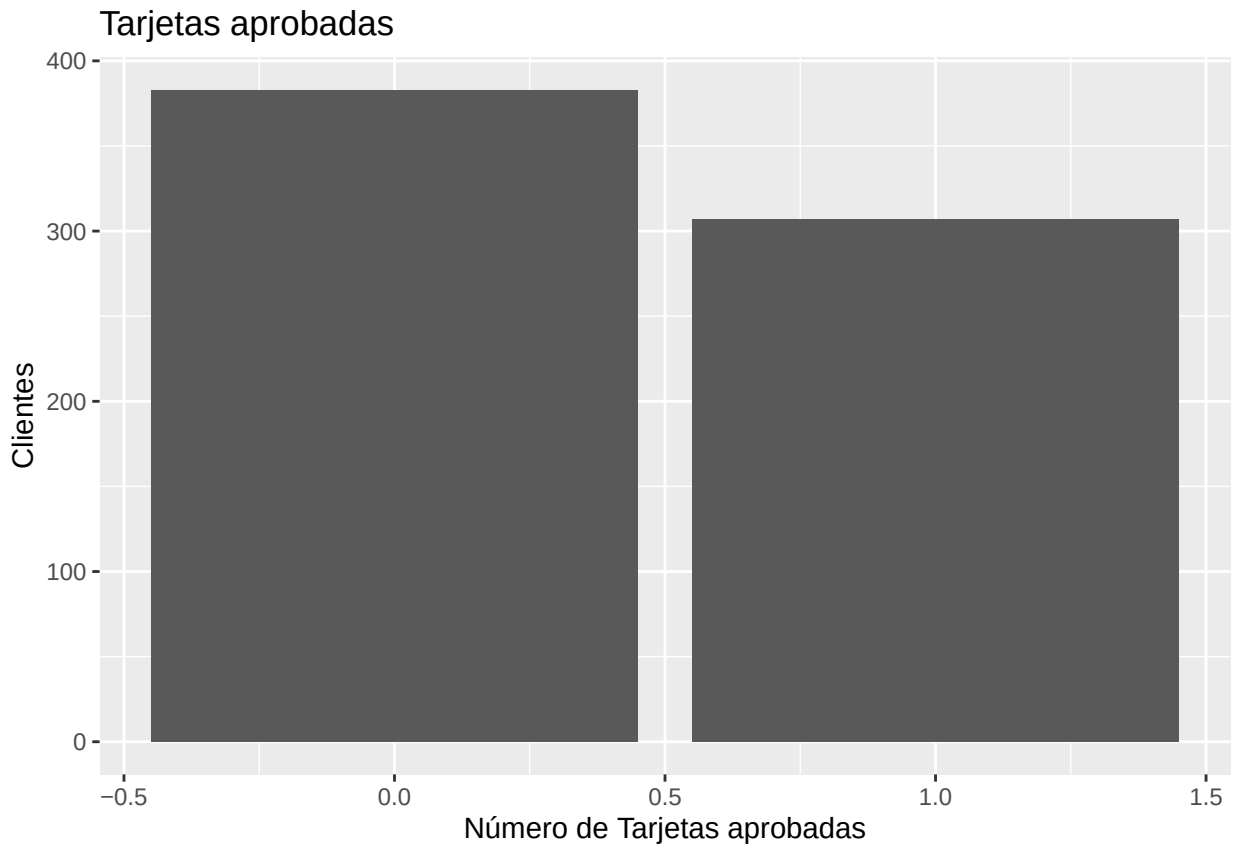
Para tener un conocimiento mayor sobre los datos vamos a utilizar algunas herramientas de visualización como son ggplot2 o grid. Vamos a visualizar la variable más importante del dataset ("Approved") y que combinaremos con el resto de variables para sacar conclusiones:

```
if(!require("ggplot2")) install.packages("ggplot2"); library("ggplot2")
```

```
## Loading required package: ggplot2
```

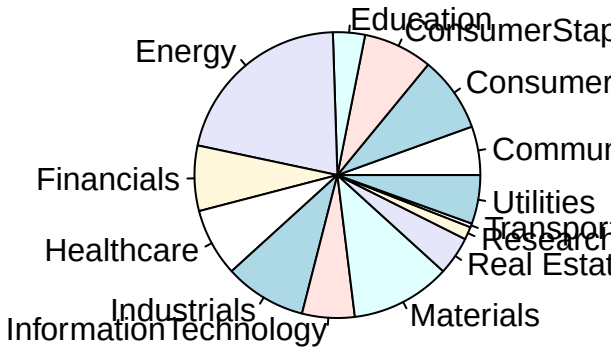
```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

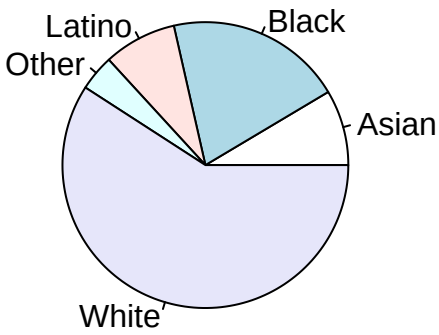
```
plotbyApproved<-ggplot(ds,aes(Approved))+geom_bar() +labs(x="Número de Tarjetas aprobadas",
y="Clientes")+ guides(fill=guide_legend(title=""))+
scale_fill_manual(values=c("blue","#008000"))+ggtitle("Tarjetas aprobadas")
plotbyApproved
```



También vamos a visualizar las principales variables categóricas:

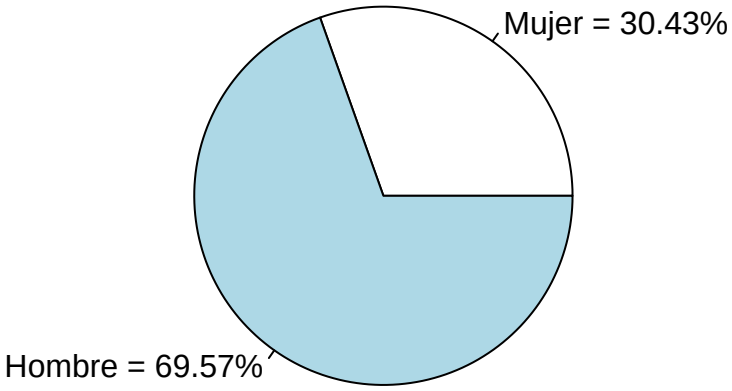
```
par(mfrow=c(1,2))
count_industry = table(ds$Industry)
count_ethnicity = table(ds$Ethnicity)
pie(count_industry)
pie(count_ethnicity)
```





Por último, visualizaremos el porcentaje de hombres y mujeres entre los clientes:

```
count_ds = table(ds$Gender)
lab_ds1 <- c("Mujer", "Hombre")
lab_ds2 <- paste0(lab_ds1, " = ", round(100 * count_ds/sum(count_ds), 2),"%")
pie(count_ds, labels = lab_ds2)
```



3.1. ¿Los datos contienen ceros o elementos vacíos? Gestiona cada uno de estos casos.

Vamos a comprobar si existen variables con elementos vacios:

```
colSums(is.na(ds))
```

```
##         Gender            Age           Debt        Married   BankCustomer
##              0              6              0              0              0
##       Industry      Ethnicity  YearsEmployed   PriorDefault       Employed
##              0              0              5              0              0
##    CreditScore DriversLicense        Citizen        ZipCode         Income
##              0              0              0              0              0
##       Approved
##              0
```

Vemos que las variables "Age" y "YearsEmployed" contienen varios elementos vacios. Vamos a proceder a su eliminación:

```
ds <- ds[!is.na(ds$Age),]
ds <- ds[!is.na(ds$YearsEmployed),]
```

Comprobamos de nuevo si existen valores vacios y vemos que ya se han eliminado los registros que los contenían.

```
colSums(is.na(ds))
```

```
##         Gender            Age           Debt        Married   BankCustomer
##              0              0              0              0              0
##       Industry      Ethnicity  YearsEmployed   PriorDefault       Employed
##              0              0              0              0              0
##    CreditScore DriversLicense        Citizen        ZipCode         Income
##              0              0              0              0              0
##       Approved
##              0
```

Se va a proceder a borrar las variables "Citizen" y "ZipCode" ya que no son relevantes para el análisis que se pretende ejecutar.

```
del <- c("Citizen", "ZipCode")
ds <- ds[ , !(names(ds) %in% del)]
```

3.2. Identifica y gestiona los valores extremos.

Se identifican 3 variables que muestran valores que en principio podrían ser extremos.

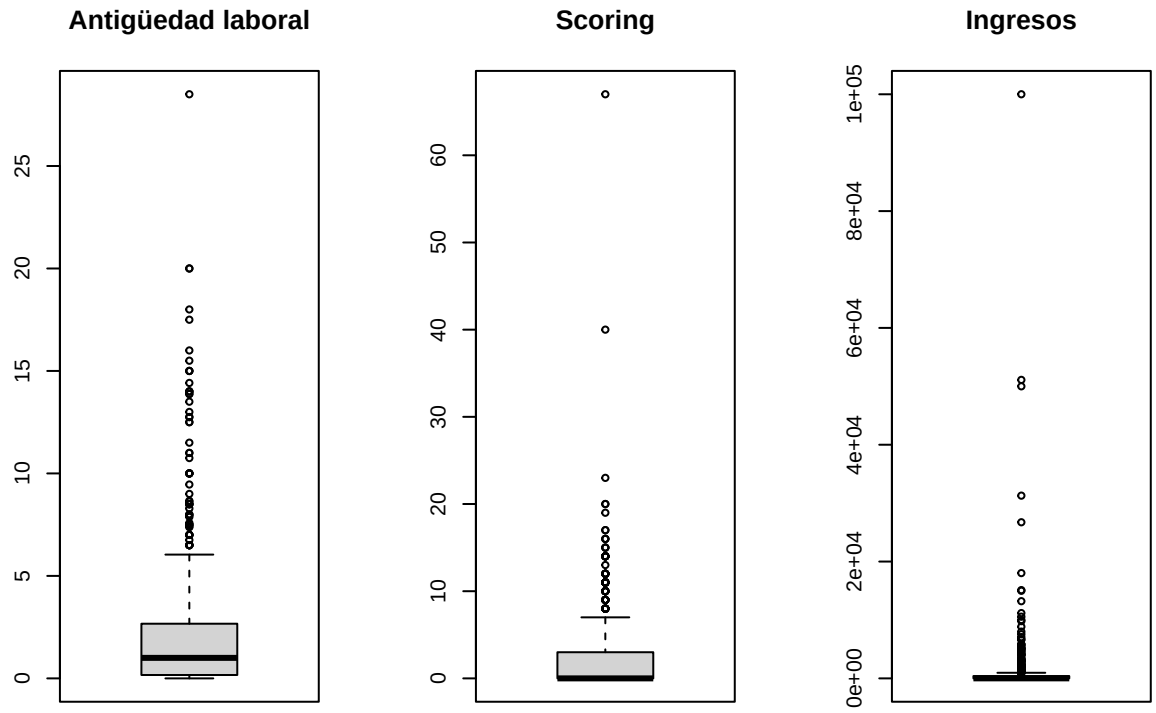
```
summary(ds[c("YearsEmployed","CreditScore","Income")])
```

```
##  YearsEmployed     CreditScore        Income
##  Min.   : 0.000   Min.   : 0.00   Min.   :     0
##  1st Qu.: 0.165   1st Qu.: 0.00   1st Qu.:     0
##  Median : 1.000   Median : 0.00   Median :     4
##  Mean   : 2.228   Mean   : 2.37   Mean   :  1009
##  3rd Qu.: 2.667   3rd Qu.: 3.00   3rd Qu.:   395
##  Max.   :28.500   Max.   :67.00   Max.   :100000
```

Vamos a visualizar la distribución de estos valores:

```
par(mfrow=c(1,3))

#Edad
boxplot(ds$YearsEmployed, main="Antigüedad laboral")
boxplot(ds$CreditScore, main="Scoring")
boxplot(ds$Income, main="Ingresos")
```




```
x<-boxplot.stats(ds$YearsEmployed)$out
idx <- which( ds$YearsEmployed %in% x)
sort(ds$YearsEmployed[idx])
```

```
##  [1]  6.500  6.500  6.500  6.500  6.500  6.500  6.750  7.000  7.000  7.000
## [11]  7.000  7.000  7.375  7.415  7.500  7.500  7.500  7.585  7.875  7.960
## [21]  8.000  8.000  8.290  8.500  8.500  8.500  8.500  8.625  8.665  9.000
## [31]  9.460 10.000 10.000 10.000 10.000 10.000 10.750 11.000 11.000 11.500
## [41] 12.500 12.500 12.750 12.750 13.000 13.500 13.875 13.875 14.000 14.000
## [51] 14.000 14.415 15.000 15.000 15.000 15.500 16.000 17.500 18.000 20.000
## [61] 20.000 28.500
```

```
x<-boxplot.stats(ds$CreditScore)$out
idx <- which( ds$CreditScore %in% x)
sort(ds$CreditScore[idx])
```

```
##  [1]  8  8  8  8  8  8  8  8  8  8  9  9  9  9  9  9  9  9  9 10 10 10 10 10
## [26] 10 10 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 12 12 12 12
## [51] 12 12 12 13 14 14 14 14 14 14 14 14 15 15 15 16 16 16 17 17 19 20 20 23 40
## [76] 67
```

```
x<-boxplot.stats(ds$Income)$out
idx <- which( ds$Income %in% x)
sort(ds$Income[idx])
```

```
##   [1]    990   1000   1000   1000   1000   1000   1000   1000   1000   1000
##  [11]   1000   1004   1058   1062   1065   1097   1110   1187   1200   1200
##  [21]   1208   1210   1210   1212   1236   1249   1260   1270   1300   1332
##  [31]   1344   1349   1391   1400   1430   1442   1465   1583   1602   1655
##  [41]   1704   1950   2000   2000   2000   2010   2028   2072   2079   2100
##  [51]   2197   2200   2206   2279   2283   2300   2384   2503   2510   2690
##  [61]   2803   2954   3000   3000   3000   3000   3065   3065   3257   3290
##  [71]   3376   3552   4000   4000   4000   4071   4159   4208   4500   4607
##  [81]   4700   5000   5000   5000   5124   5200   5298   5552   5777   5800
##  [91]   5860   6590   6700   7059   7544   8000   8851   9800  10000  10561
## [101]  11177  13212  15000  15108  18027  26726  31285  50000  51100 100000
```

Vamos a sustituir los "outliers" por la media de cada variable:

```
ds$YearsEmployed[ds$YearsEmployed > 18.000] <- mean(ds$YearsEmployed)
ds$CreditScore[ds$CreditScore > 25] <- mean(ds$CreditScore)
ds$Income[ds$Income > 20000] <- mean(ds$Income)
```

4. Análisis de los datos.

4.1. Selección de los grupos de datos que se quieren analizar/comparar (p. e., si se van a comparar grupos de datos, ¿cuáles son estos grupos y qué tipo de análisis se van a aplicar?)

A continuación, crearemos los grupos dentro de nuestro conjunto de datos que utilizaremos para analizar y comparar debido a su interés. Se dividirá el dataset por etnias, géneros y edad.

```
#Por etnia
ds.White <- ds[ds$Ethnicity == "White",]
ds.Black <- ds[ds$Ethnicity == "Black",]
ds.Asian <- ds[ds$Ethnicity == "Asian",]
ds.Latino <- ds[ds$Ethnicity == "Latino",]
ds.Other <- ds[ds$Ethnicity == "Other",]

#Por género
ds.Male <- ds[ds$Gender == 0,]
ds.Female <-ds[ds$Gender == 1,]

#Por edad
ds.joven <-ds[ds$Age < 0, ]
ds.adulto <-ds[ds$Age > 30 & ds$Age < 50,]
ds.mayor <-ds[ds$Age > 50,]
```

4.2. Comprobación de la normalidad y homogeneidad de la varianza.

Utilizaremos la prueba de normalidad de Anderson-Darling para la comprobación de la normalidad.

En breves palabras, si en la prueba se obtiene un p-valor superior al nivel de significación prefijado alpha = 0, 05, se considera que una variable en cuestión sigue una distribución normal.

Para ello, utilizaremos la librería nortest.

```
if(!require(nortest)){
  install.packages('nortest', repos='http://cran.us.r-project.org')
  library(nortest)
}
```

```
## Loading required package: nortest
```

```
alpha = 0.05
col.names = colnames(ds)
for (i in 1:ncol(ds)) {
  if (i == 1) cat("Variables que no siguen una distribución normal:\n")
  if (is.integer(ds[,i]) | is.numeric(ds[,i])) {
    p_val = ad.test(ds[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      # Format output
      if (i < ncol(ds) - 1) cat(", ")
      if (i %% 3 == 0) cat("\n")
    }
  }
}
```

```
## Variables que no siguen una distribución normal:
## Gender, Age, Debt,
## Married, BankCustomer, YearsEmployed, PriorDefault,
## Employed, CreditScore, DriversLicense,
## IncomeApproved
```

Para comprobar la homogeneidad de la varianza realizaremos un test de Fligner-Killen. En este caso, estudiaremos los grupos Male y Female, para ver si sus variables 'Approved' siguen la misma varianza o no. En el siguiente test, la hipótesis nula consiste en que ambas varianzas son iguales.

```
fligner.test(x = list(ds.Male$Approved,ds.Female$Approved))
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  list(ds.Male$Approved, ds.Female$Approved)
## Fligner-Killeen:med chi-squared = 0.34085, df = 1, p-value = 0.5593
```

Observamos que conseguimos un p-valor superior a 0,05, por lo que podemos rechazar la hipótesis nula. Entonces, concluimos que que las varianzas de ambas muestras son homogéneas.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

4.3.1. Análisis de correlación

Antes de nada, vamos a realizar un matriz de correlación para ver en qué grado influye cada variable en el valor objetivo ('Approved').

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("Correlación", "p-value")

# Calcular el coeficiente de correlación para cada variable cuantitativa
# con respecto al campo "precio"
for (i in 1:(ncol(ds) - 1)) {
  if (is.integer(ds[,i]) | is.numeric(ds[,i])) {
    spearman_test = cor.test(ds[,i],
                             ds[,length(ds)],
                             method = "spearman",exact=FALSE)
    corr_coef = spearman_test$estimate
    p_val = spearman_test$p.value
    # Add row to matrix
    pair = matrix(ncol = 2, nrow = 1)
    pair[1][1] = corr_coef
    pair[2][1] = p_val
    corr_matrix <- rbind(corr_matrix, pair)
    rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(ds)[i]
  }
}

print(corr_matrix)
```

```
##                Correlación       p-value
## Gender         -0.02242154  5.597257e-01
## Age             0.15781528  3.617760e-05
## Debt            0.19179072  4.773022e-07
## Married         0.18655590  9.816274e-07
## BankCustomer    0.19506190  3.010750e-07
## YearsEmployed   0.37552109  3.647293e-24
## PriorDefault    0.72112437 5.269994e-110
## Employed        0.45231063  1.512190e-35
## CreditScore     0.50271558  9.053725e-45
## DriversLicense  0.03376241  3.797279e-01
## Income          0.28107148  8.554681e-14
```

Sabemos que las variables más correlacionadas con el Approved son las que están cerca de los valores -1 y +1. Por ello, la variable más relevante es PriorDefault. Se muestra también los valores p para saber el peso estadístico de la correlación obtenida.

4.3.2 Análisis de clustering

A continuación vamos a efectuar un análisis cluster para ver si podemos ayudar a la entidad de crédito a dividir sus clientes en grupos homogeneos, de manera que puedan realizar acciones comerciales adaptadas a cada grupo.

En primer lugar establecemos un subgrupo con las variables “Age”, “Debt”, “YearsEmployed”, “CreditScore” e “Income”.

```
x <- cbind(ds[,2:3],ds[8],ds[11],ds[13])
dim(x)
```

```
## [1] 679   5
```

Cargamos el paquete cluster:

```
if (!require('cluster')) install.packages('cluster')
```

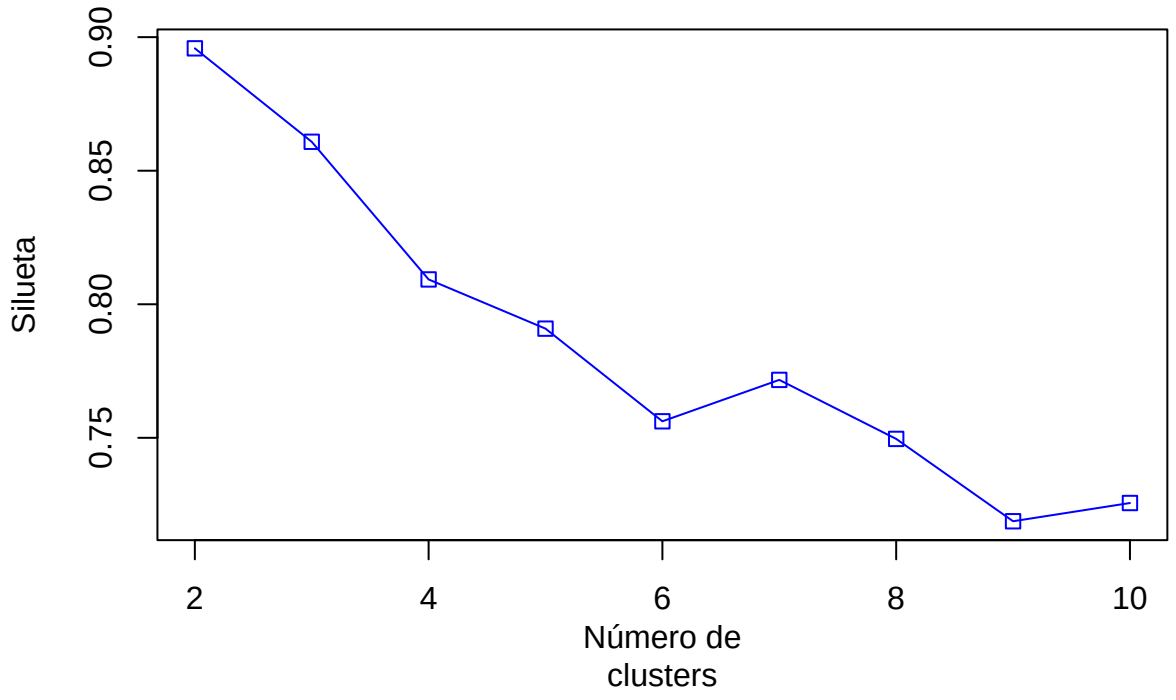
```
## Loading required package: cluster
```

```
library(cluster)
```

Vamos a aplicar el algoritmo k-means con varios valores de clústers (desde 2 hasta un máximo de 10). La función silhouette devuelve para cada muestra el clúster dónde ha sido asignado, el clúster vecino y el valor de la silueta. Por lo tanto, calculando la media de la tercera columna podemos obtener una estimación de la calidad del agrupamiento.

```
d <- daisy(x)
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
fit <- kmeans(x, i)
y_cluster <- fit$cluster
sk <- silhouette(y_cluster, d)
resultados[i] <- mean(sk[,3])
}
```

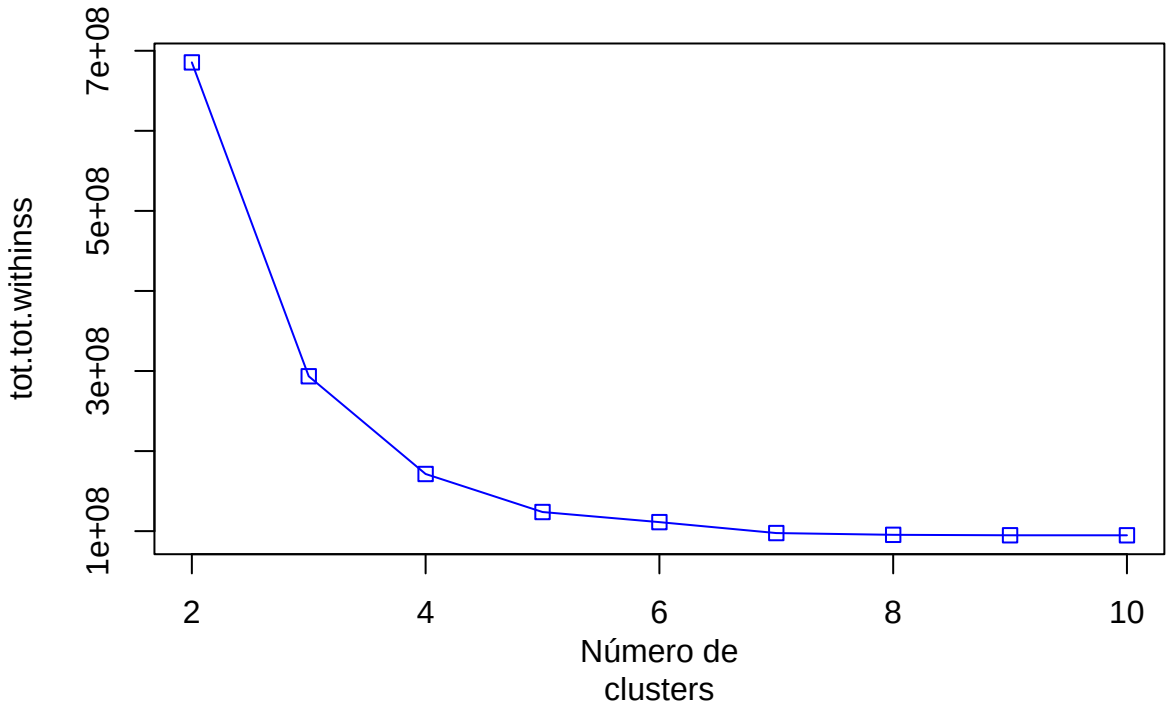
```
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Número de
clusters",ylab="Silueta")
```



Vemos que desde el cluster 2 hasta el 4 ocurre una gran caida en el gráfico. Esto nos está indicando que es con k=4 donde la mejora es más significativa.

Otra forma de evaluar cual es el mejor número de clústers es considerar el mejor modelo a aquel que ofrece la menor suma de los cuadrados de las distancias de los puntos de cada grupo con respecto a su centro (withinss), con la mayor separación entre centros de grupos (betweenss). Una manera común de hacer la selección del número de clústers consiste en aplicar el método elbow (codo), que no es más que la selección del número de clústers en base a la inspección de la gráfica que se obtiene al iterar con el mismo conjunto de datos para distintos valores del número de clústers. Se seleccionará el valor que se encuentra en el codo de la curva.

```
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
fit <- kmeans(x, i)
resultados[i] <- fit$tot.withinss
}
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Número de
clusters",ylab="tot.tot.withinss")
```

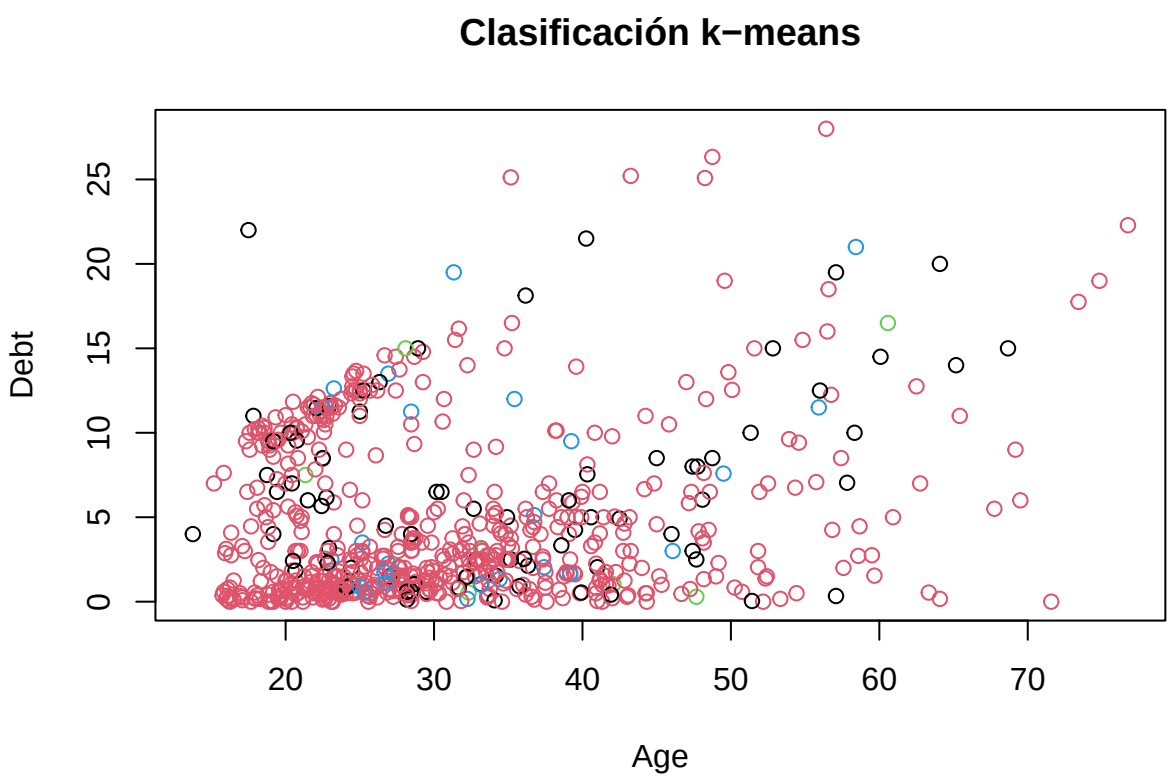


En este caso también vemos que el número óptimo de clústers podrían ser 3 o 4, ya que es a partir de donde la curva empieza a estabilizarse. Nosotros optaremos por quedarnos con k=4 que coincide con el método anterior.

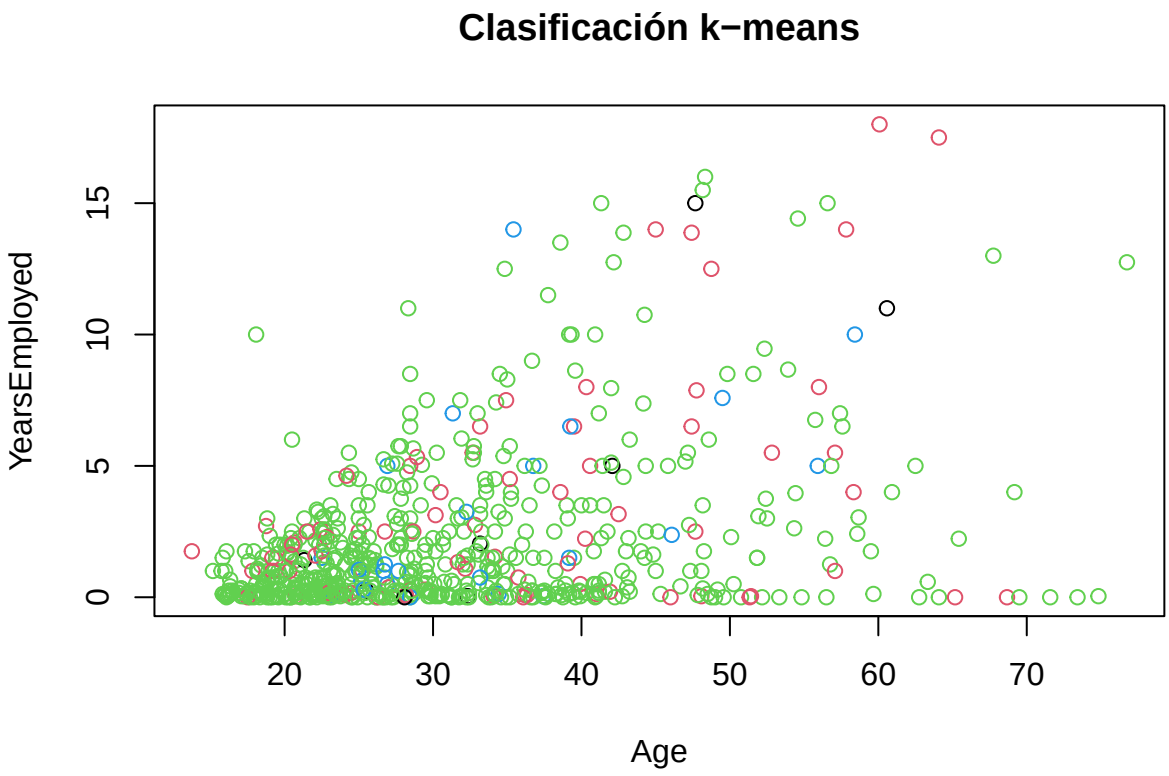
En el caso que estamos estudiando, la entidad de crédito lo que pretende es clasificar a los clientes en 4 grupos: 1) Clientes con potencial nulo de hacer negocio 2) clientes que muestran un bajo potencial, 3) clientes con potencial medio, 4) clientes prioritarios (alto potencial de contratación). Por ello, vamos a ver

cómo se comporta kmeans en el caso de pedirle 4 clústers. Para eso comparamos visualmente los campos dos a dos. Empezamos con el primer par y continuamos para tener una imagen visual para ver como se relacionan cada una de ellas:

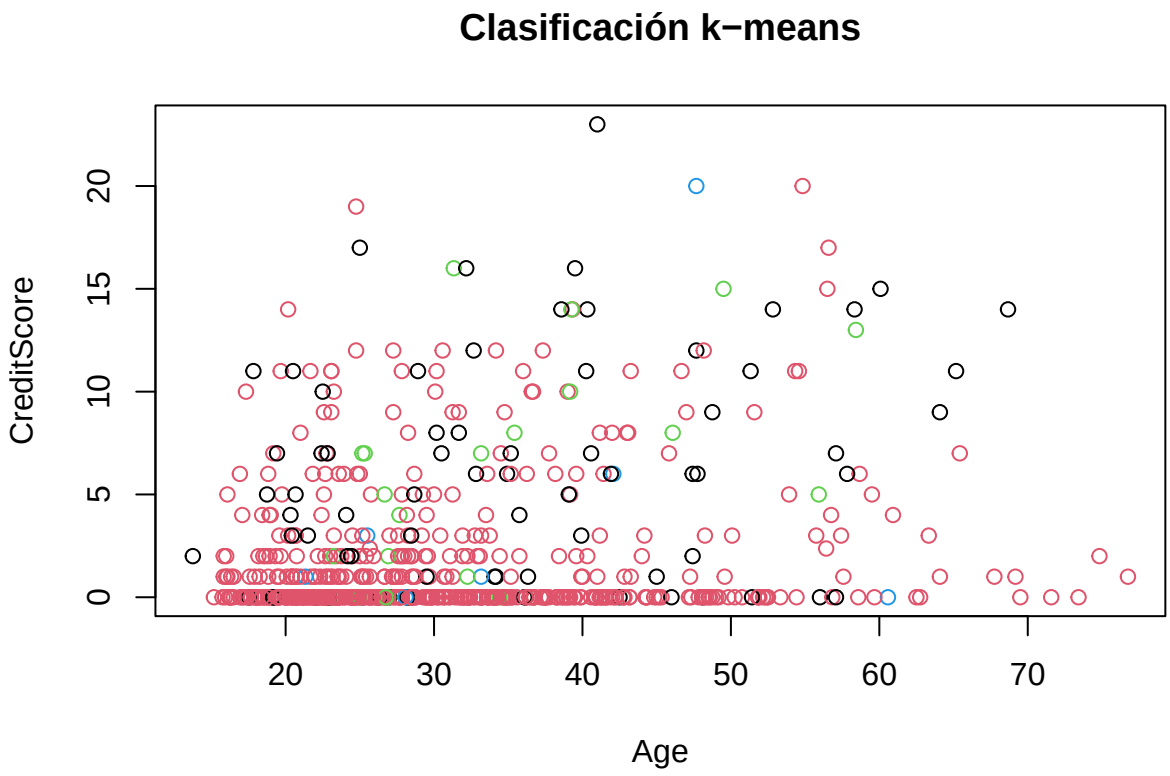
```
ds4clusters <- kmeans(x, 4)
# Debt y Age
plot(x[c(1,2)], col=ds4clusters$cluster, main="Clasificación k-means")
```



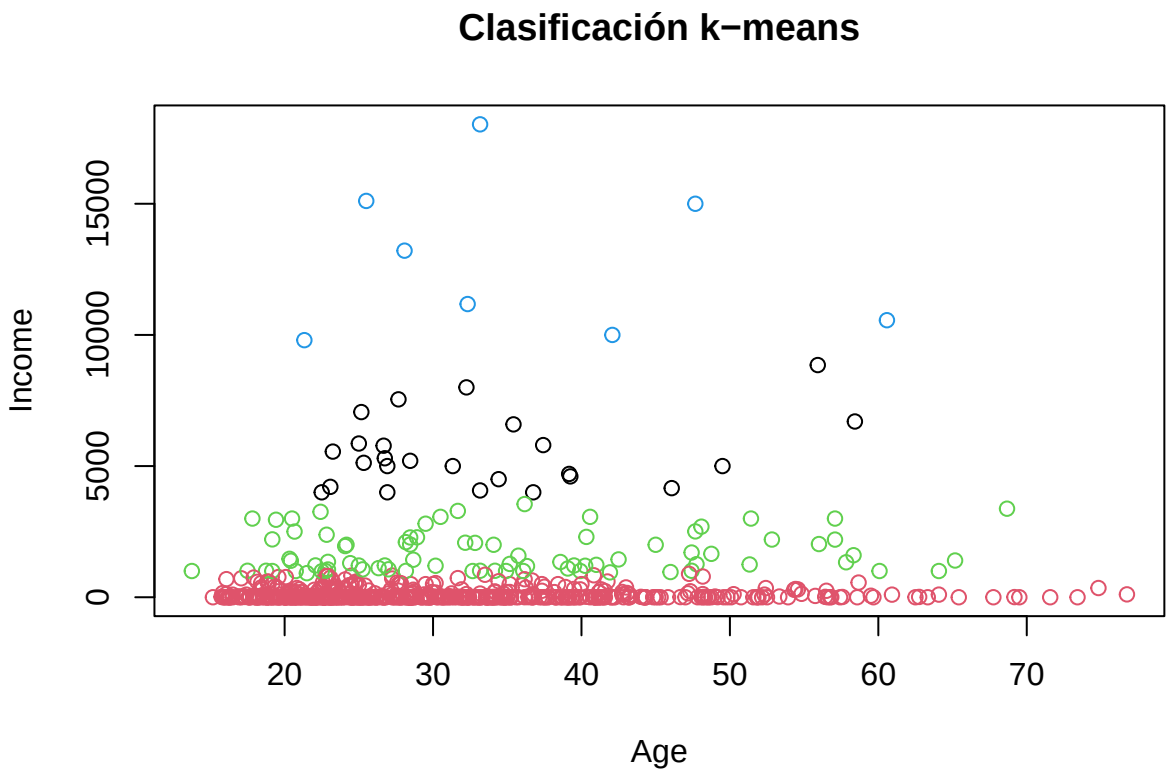

```
ds4clusters <- kmeans(x, 4)
# YearsEmployed y Age
plot(x[c(1,3)], col=ds4clusters$cluster, main="Clasificación k-means")
```



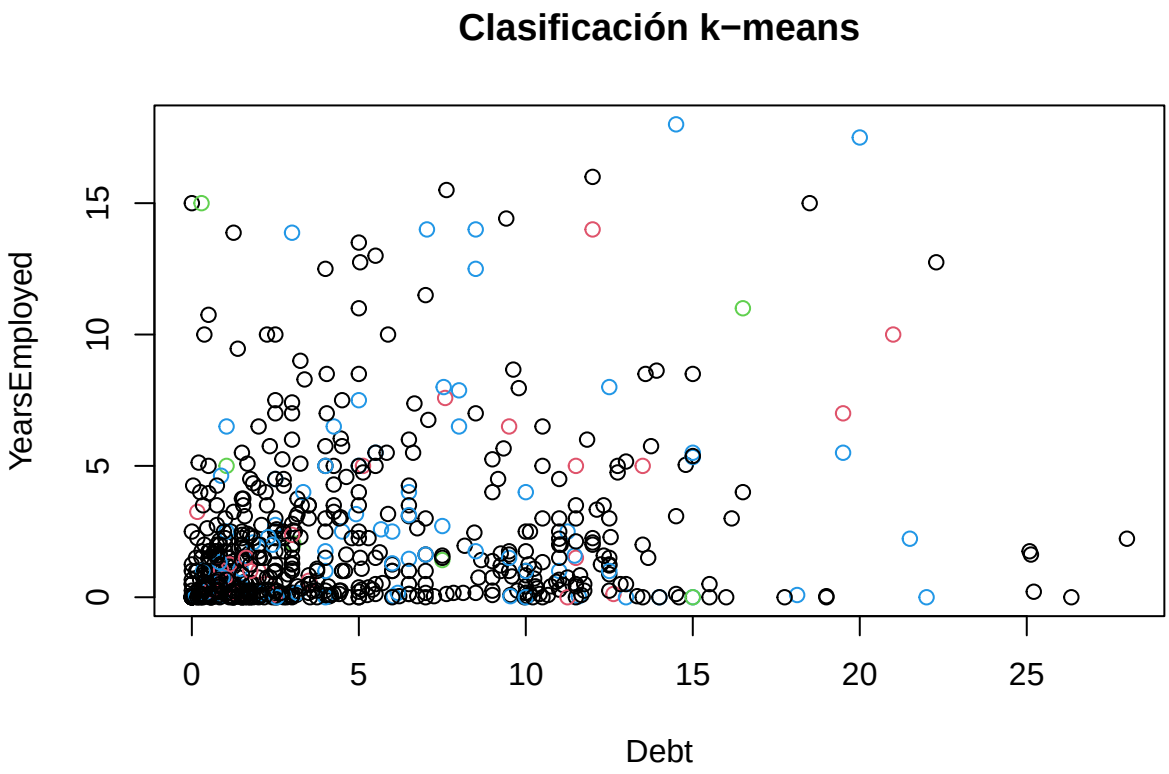

```
ds4clusters <- kmeans(x, 4)
# CreditScore y Age
plot(x[c(1,4)], col=ds4clusters$cluster, main="Clasificación k-means")
```



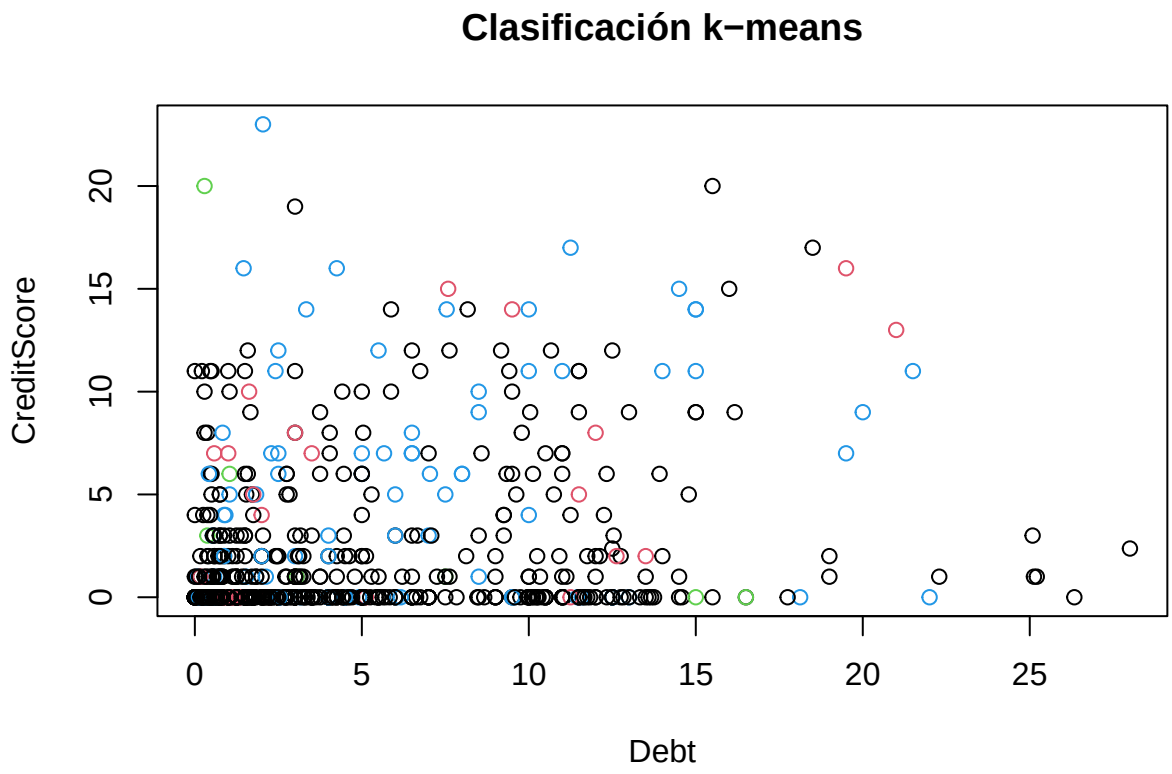

```
ds4clusters <- kmeans(x, 4)
# Income y Age
plot(x[c(1,5)], col=ds4clusters$cluster, main="Clasificación k-means")
```



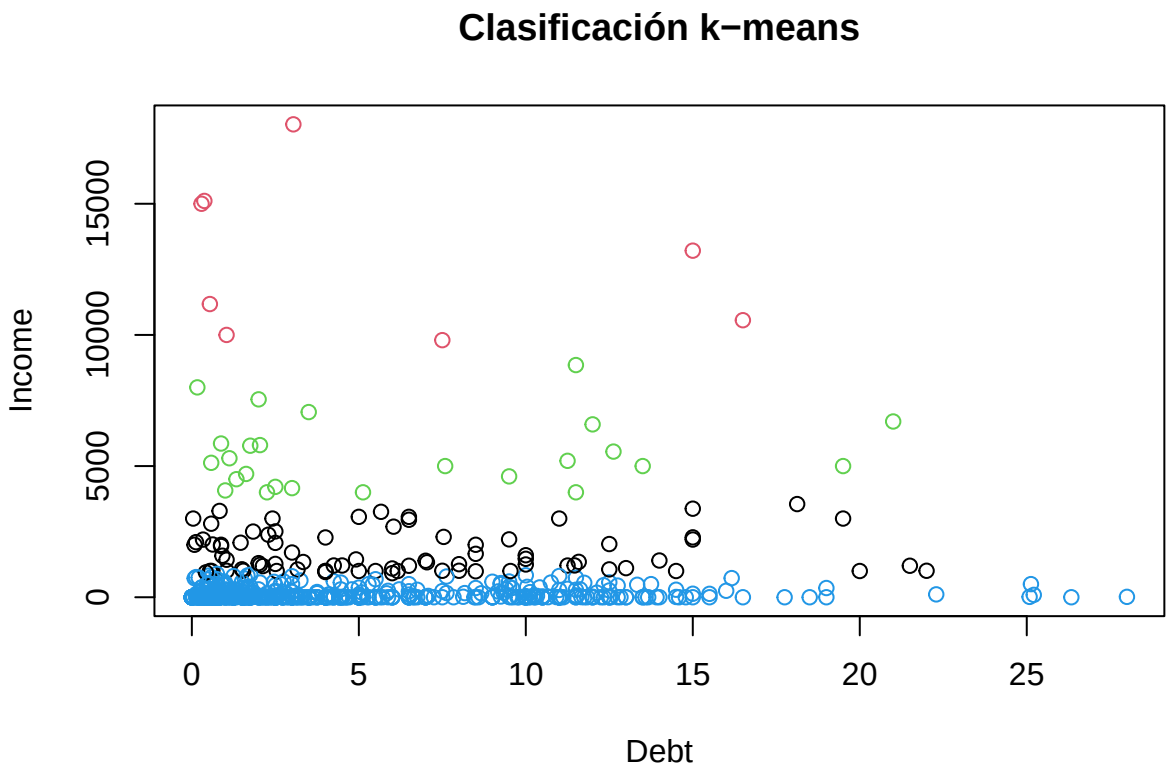

```
ds4clusters <- kmeans(x, 4)
# YearsEmployed y Debt
plot(x[c(2,3)], col=ds4clusters$cluster, main="Clasificación k-means")
```



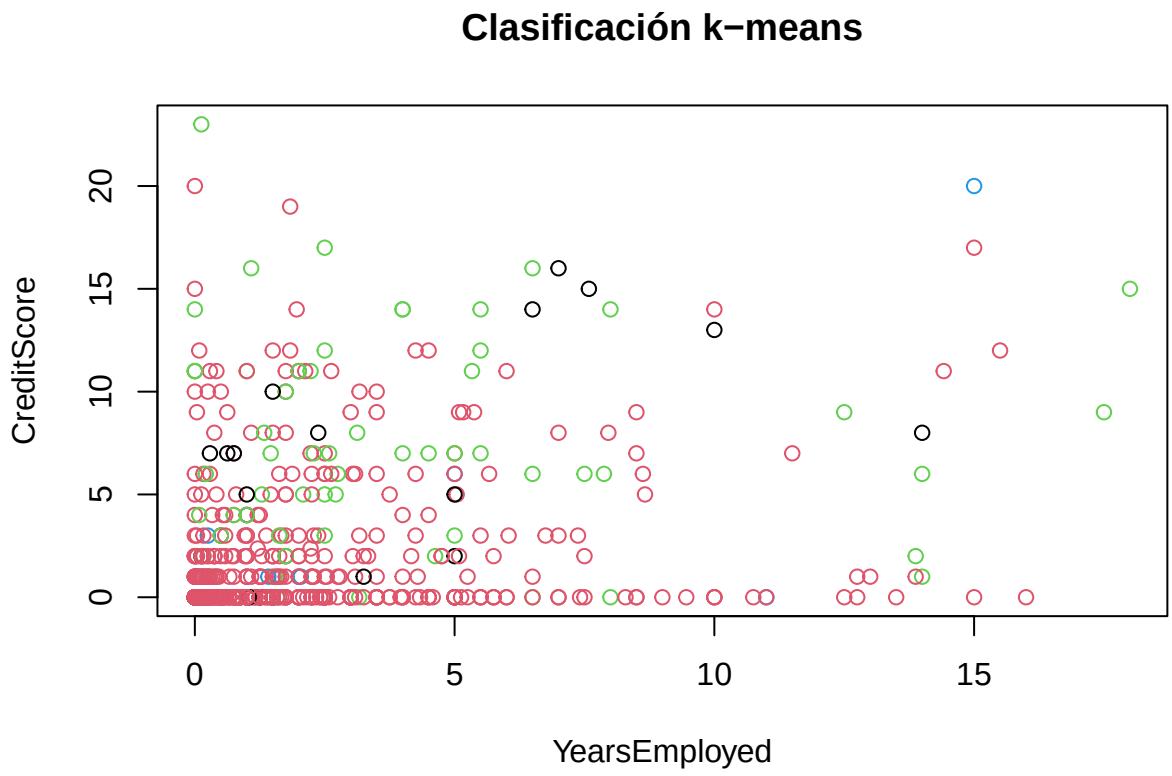

```
ds4clusters <- kmeans(x, 4)
# CreditScore y Debt
plot(x[c(2,4)], col=ds4clusters$cluster, main="Clasificación k-means")
```




```
ds4clusters <- kmeans(x, 4)
# Income y Debt
plot(x[c(2,5)], col=ds4clusters$cluster, main="Clasificación k-means")
```

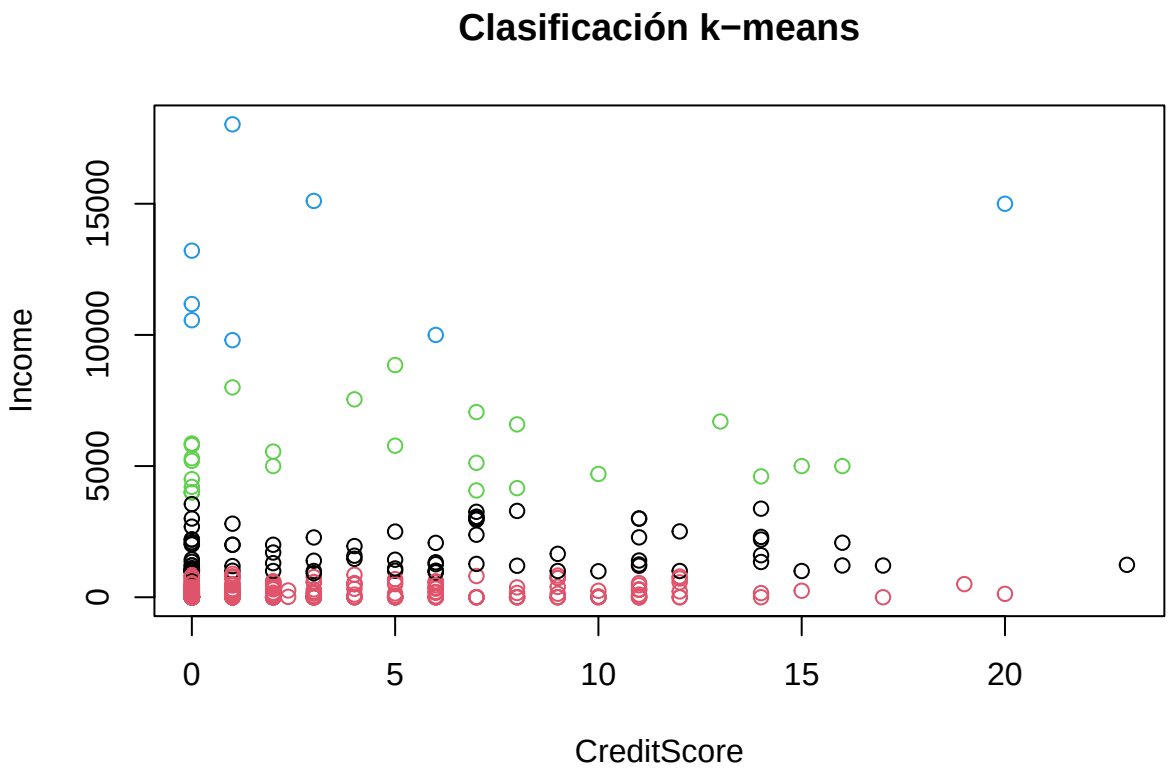



```
ds4clusters <- kmeans(x, 4)
# CreditScore y YearsEmployed
plot(x[c(3,4)], col=ds4clusters$cluster, main="Clasificación k-means")
```




```
ds4clusters <- kmeans(x, 4)
# Income y YearsEmployed
plot(x[c(3,5)], col=ds4clusters$cluster, main="Clasificación k-means")
```


```
ds4clusters <- kmeans(x, 4)
# Income y CreditScore
plot(x[c(4,5)], col=ds4clusters$cluster, main="Clasificación k-means")
```



Vemos que la mayoría de los pares de variables analizadas no nos ayudan a establecer grupos de clientes. Tan solo en el caso de la variable Income combinada con el resto nos dan una muestra de cómo podriamos clasificar a los clientes según los 4 grupos que le gustaría a la empresa para potenciar sus ventas.

5. Representación de los resultados a partir de tablas y gráficas.

A lo largo de la práctica hemos ido introduciendo las gráficas necesarias para la comprensión de la actividad, por lo que no vemos la necesidad de concentrar todas las representaciones en un punto de la práctica

6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

A partir de los resultados obtenidos en este trabajo, hemos conseguido realizar desde un dataset obtenido de un banco de datos (Kaggle) realizar la limpieza y análisis de datos. En primer lugar, hemos justificado la elección del dataset, aludiendo a su gran interés en el mundo financial.

En el segundo apartado hemos analizado todas las variables de las que disponemos para realizar posteriormente las diferentes análisis. Posteriormente, hemos realizado el preprocesamiento de los datos, trabajando con los valores nulos y outliers. Para estos últimos, hemos decidido sustituirlos con la media de los datos.

En el cuarto apartado, nos hemos sumergido en la estadística, primeramente separando los grupos de datos que son interesantes de analizar (género, etnia y edad). Después, hemos comprobado la normalidad y la

homogeneidad de las varianzas concluyendo las variables que siguen y no siguen la normalidad con la prueba de normalidad de Anderson-Darling. Hemos concluido también que los grupos de datos de Male y Female siguen la misma varianza para la variable Approved mediante el Fligner-Killeen test.

Finalmente, hemos realizado análisis de los datos como son la correlación de las variables con el valor objetivo (Approved), concluyendo que la que más influye es PriorDefault. Después, hemos realizado un ejercicio de clustering para poder agrupar los datos con el método de k-means, observando que que la mayoría de los pares de variables analizadas no nos ayudan a establecer grupos de clientes. Tan solo en el caso de la variable Income combinada con el resto nos dan una muestra de cómo podríamos clasificar a los clientes según los 4 grupos que le gustaría a la empresa para potenciar sus ventas.

7. Código: Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos..

El código utilizado se incluye junto a las respuestas en el archivo entregado.

8. Contribuciones y firmas.

Contribuciones	Firma
Investigación previa	J.C., A.E.
Redacción de las respuestas	J.C., A.E.
Desarrollo código	J.C., A.E.