# Trabajo Integrador
# Introducción a la Ciencia de Datos

Ignacio Aguilera Martos

22 de Diciembre de 2019

## Índice

# 1. Análisis Exploratorio de los Datos

En esta primera sección vamos a hacer una exploración de los datos intentando extraer algunas conclusiones de los mismos y poder conocer más en profundidad la información que nos arrojan las variables. Esta sección va a estar dividida en dos subsecciones, una para el conjunto de regresión y otra para el conjunto de clasificación.

## 1.1. Conjunto de Regresión

El conjunto de datos del que dispongo para realizar regresión es el conjunto “treasury”. Vamos a analizar el conjunto de datos.

En primer lugar el conjunto de datos dispone de 16 variables numéricas y 1049 observaciones como podemos observar con el siguiente código.

```r
# En primer lugar vamos a visualizar el dataset de regresión.

library(foreign)
dataset_regresion<-read.arff("../..//DATOS/Datasets Regresion/treasury/treasury.dat")
dataset_regresion

# Ahora vamos a ver el número de variables y el tipo de cada una.
cat("El número de variables es: ", length(colnames(dataset_regresion)), "\n")
cat("El tipo de las variables es:\n")
for(i in 1:length(colnames(dataset_regresion))){
  cat("\t",colnames(dataset_regresion)[i], ": ", class(dataset_regresion[i][[1]]), "\n")
}
```

```
1Y-CMaturityRate :  numeric
30Y-CMortgageRate :  numeric
3M-Rate-AuctionAverage :  numeric
3M-Rate-SecondaryMarket :  numeric
3Y-CMaturityRate :  numeric
5Y-CMaturityRate :  numeric
bankCredit :  numeric
currency :  numeric
demandDeposits :  numeric
federalFunds :  numeric
moneyStock :  numeric
checkableDeposits :  numeric
loansLeases :  numeric
savingsDeposits :  numeric
tradeCurrencies :  numeric
1MonthCDRate :  numeric
```

Figura 1: Tipos de las variables y código para obtenerlos

Como podemos ver el conjunto dispone de 16 variables de las cuales todas son numéricas. Esto es lógico pues el conjunto está destinado para el problema de regresión y además nos facilita el trabajo.

Para poder seguir analizando el conjunto vamos a hacer un estudio pormenorizado de las variables del mismo en función a una serie de estadísticos básicos.

Estos estadísticos que voy a emplear son: media, mediana, moda, desviación típica, mínimo, máximo, curtosis y asimetría.

En este conjunto podemos dividir las variables en dos grupos. El primero de ellos corresponde a las variables de entrada que son las que nos van a permitir obtener resultados sobre la salida y el segundo grupo es la propia salida esperada del sistema.

Vamos a empezar a analizar en primer lugar la salida.

### 1.1.1. Estudio de los estadísticos

**Salida**

La variable de salida es la que tiene por nombre 1MonthCDRate. Los estadísticos que nos arroja esta variable son los siguientes:

```
1MonthCDRate :
        Media:  7.521945
        Mediana:  6.61
        Desviación típica:  3.377216
        Moda:  5.56
        Kurtosis:  1.733596
        Asimetría:  1.32818
        Mínimo:  3.02
        Máximo:  20.76
```

Figura 2: Estadísticos de la variable de salida.

Podemos observar que la media de la salida es aproximadamente de 7,5 y su desviación típica de 3,3, esto nos indica que la mayoría de los datos (el 95 % en caso de estar ante una distribución normal) van a estar dentro del intervalo $[0{,}9, 14, 1]$. Aún así podemos ver mediante el mínimo y el máximo que los datos se van a mover en el intervalo $[3{,}02, 20{,}76]$. Esto parece que nos va a indicar que deberíamos de tener una cola más pesada a la izquierda de la distribución y una cola más alargada a la derecha de la misma. Este hecho viene corroborado también por la mediana. Vemos que es 6,61 (menor que la media) con lo que nos está diciendo que el 50 % de los datos van a estar en el intervalo $[3{,}02, 6{,}61]$ y por tanto proporcionalmente esta cola será más pesada.

Otra forma que tenemos de comprobar lo que hemos dicho es mediante el coeficiente de asimetría. Al ser positivo nos está indicando que la distribución es asimétrica a la derecha, cuestión que ya sabemos y hemos razonado.

La curtosis nos está dando una indicación de cómo de puntiaguda o achatada es la distribución. En este caso el coeficiente es positivo, por lo que la distribución será más puntiaguda. Esto nos está indicando que vamos a tener una mayor concentración de datos entorno a la media.

**Variable 1: 1Y-CMaturityRate**

Los estadísticos que nos arroja esta variable son:

```
1Y-CMaturityRate :
        Media:  97.35363
        Mediana:  92.526
        Desviación típica:  14.47144
        Moda:  86.878
        Kurtosis:  0.2769317
        Asimetría:  1.10812
        Mínimo:  77.055
        Máximo:  142.645
```

Figura 3: Estadísticos de la variable 1 1Y_CMaturityRate

En primer lugar podemos observar que la media es de 97,35363 y la mediana es ligeramente inferior. Esto nos vuelve a indicar que vamos a tener una cola más corta a la izquierda y una más larga a la derecha, es decir, vamos a tener más valores a la derecha de la distribución. Podemos corroborar este hecho también comprobando el mínimo y el máximo. Como podemos ver el máximo está mucho más alejado de la media que el mínimo por lo que podemos intuir que la distribución va a ser más alargada por ese lado.

De igual forma si miramos el coeficiente de asimetría vemos que es positivo lo que nos está indicando la asimetría de la cola derecha.

La curtosis es positiva pero no muy lejana del cero, por lo que tendrá la forma de una normal pero un poco más puntiaguda.

**Variable 2: 30Y-CMortgageRate**

Los estadísticos que nos arroja esta variable son:

```
30Y-CMortgageRate :
        Media:  7.543937
        Mediana:  6.71
        Desviación típica:  3.105787
        Moda:  5.59
        Kurtosis:  0.4963418
        Asimetría:  1.026759
        Mínimo:  3.02
        Máximo:  17.15
```

Figura 4: Estadísticos de la variable 2 30Y_CMortgageRate

Podemos ver que la media es 7,543937 y la mediana 6,71 lo que de nuevo nos hace sospechar que la cola derecha es más alargada. Esto lo podemos ver (y podemos decir ya que la cola va a ser bastante alargada) con el máximo y el mínimo. Sabemos que el 95 % de los datos debería de estar en el intervalo $[media - 2 * stdv, media + 2 * stdv]$ pero la realidad es que observamos que la cola derecha es más larga. Asimismo vemos que la moda es aún menor que la mediana con lo que corroboramos que la cola izquierda es más pesada y la derecha más alargada.

El coeficiente de asimetría nos termina de corroborar lo que estamos infiriendo pues al ser positivo nos indica una asimetría en la cola derecha.

En cuanto a la curtosis podemos ver que es más puntiaguda que una normal.

**Variable 3: 3M-Rate-AuctionAverage**

Los estadísticos que nos arroja esta variable son:

```
3M-Rate-AuctionAverage :
        Media:  10.40085
        Mediana:  9.9
        Desviación típica:  2.958872
        Moda:  10.4
        Kurtosis:  -0.2401306
        Asimetría:  0.826309
        Mínimo:  6.49
        Máximo:  18.63
```

Figura 5: Estadísticos de la variable 3 3M_Rate_AuctionAverage

Esta variable tiene como media 10,40085 y mediana 9,9. En esta variable no vemos una diferencia tan grande por lo que a priori podemos pensar que no es muy asimétrica. La desviación típica es de 2,958872 por lo que (en caso de que fuese una normal) el 95 % de los datos se van a encontrar en el intervalo [4,483106, 16,318594]. Vemos que el mínimo es 6,49 con lo que podemos entender que la cola izquierda es más corta mientras que el máximo es 18,63 con lo que la cola derecha va a ser algo más alargada.

Si observamos los coeficiente de asimetría y curtosis podemos ver que el coeficiente de asimetría es positivo lo que nos indica que la distribución es asimétrica a la derecha y la curtosis es ligeramente negativa con lo que podemos decir que la distribución será algo más achatada que una distribución normal.

**Variable 4: 3M-Rate-SecondaryMarket**

Los estadísticos que nos arroja esta variable son:

```
3M-Rate-SecondaryMarket :
        Media:  6.85122
        Mediana:  5.81
        Desviación típica:  2.954287
        Moda:  5.12
        Kurtosis:  1.147822
        Asimetría:  1.206275
        Mínimo:  2.67
        Máximo:  16.75
```

Figura 6: Estadísticos de la variable 4 3M_Rate_SecondaryMarket

Podemos observar que el comportamiento de esta variable es el mismo que las anteriores y que va a tener una cola a la derecha más alargada. Esto es comprobable por todas las razones expuestas en las secciones anteriores.

**Variable 5: 3Y-CMaturityRate**

Los estadísticos que nos arroja esta variable son:

```
3Y-CMaturityRate :
        Media:  6.829342
        Mediana:  5.77
        Desviación típica:  2.942284
        Moda:  5
        Kurtosis:  1.14164
        Asimetría:  1.199487
        Mínimo:  2.69
        Máximo:  16.76
```

Figura 7: Estadísticos de la variable 5 3Y_CMaturityRate

De igual forma los estadísticos de esta variable nos están arrojando la misma información que para el resto de variables con una cola derecha muy alargada. Podemos corroborar esta información por lo mismo que hemos dicho antes (mediana, media, máximo y minino) además de coeficiente de asimetría que al ser positivo nos indica que es asimétrica a la derecha.

Por otro lado, al igual que en los casos previos, la curtosis nos indica que la distribución es más puntiaguda o apuntada que una distribución normal.

**Variable 6: 5Y-CMaturityRate**

Los estadísticos que nos arroja esta variable son:

```
5Y-CMaturityRate :
        Media:  8.117378
        Mediana:  7.44
        Desviación típica:  2.88388
        Moda:  6.53
        Kurtosis:  -0.08284863
        Asimetría:  0.8626431
        Mínimo:  4.09
        Máximo:  16.47
```

Figura 8: Estadísticos de la variable 6 5Y_CMaturityRate

En esta variable observamos también una cola derecha más alargada pero podemos observar que aquí no es una cola tan alargada como en los casos anteriores.

Si comprobamos la media y la mediana observamos que la mediana es menor que la media por lo que ya nos está apuntando a la asimetría pero si vemos la desviación típica y calculamos el intervalo en el que deberían estar el 95 % de los datos ($[2{,}349618, 13, 885138]$) podemos percibir que la cola izquierda es más corta (pues el mínimo es 4,09) y la cola derecha más alargada pues el máximo llega hasta 16,47.

Podemos decir que la asimetría derecha se manifiesta pero con menor intensidad como podemos percibir por el coeficiente de asimetría, pues aunque es positivo, es menor que en otros casos.

En cuanto a la curtosis podemos ver que es ligeramente negativa pero muy cercana al cero por lo que podemos decir que la distribución esta ligerísimamente achatada con respecto a una normal aunque visualmente probablemente no pudiéramos distinguirlo.

**Variable 7: bankCredit**

Los estadísticos que nos arroja esta variable son:

```
bankCredit :
        Media:  8.359104
        Mediana:  7.76
        Desviación típica:  2.766248
        Moda:  6.63
        Kurtosis:  -0.2687357
        Asimetría:  0.8139356
        Mínimo:  4.17
        Máximo:  16.13
```

Figura 9: Estadísticos de la variable 7 bankCredit

En este caso podemos ver que la mediana es más pequeña que la media por lo que podemos pensar de nuevo en que esta variable tiene una cola derecha más alargada. Si calculamos el intervalo en el que deberían de estar el 95 % de los datos ($[2{,}826608, 13{,}8916]$) podemos apreciar esa tendencia a una cola derecha más pesada.

Aún así, si comprobamos el coeficiente de asimetría, podemos ver que se manifiesta la asimetría derecha pero de forma ligera como en la variable anterior pues el coeficiente no es tan grande como en otras variables que ya hemos analizado.

En cuanto a la curtosis tenemos una curtosis negativa lo que nos indica que la distribución está algo más achatada que su correspondiente distribución normal.

**Variable 8: currency**

Los estadísticos que nos arroja esta variable son:

```
currency :
        Media:  2639.677
        Mediana:  2616.1
        Desviación típica:  1010.521
        Moda:  1287.7
        Kurtosis:  -0.9085405
        Asimetría:  0.3116162
        Mínimo:  1130.9
        Máximo:  4809.2
```

Figura 10: Estadísticos de la variable 8 currency

En este caso la mediana también es algo más pequeña que la media, pero si comprobamos el rango de valores de la variable ($[1130{,}9, 4809{,}2, ]$) observamos que la diferencia no es tan significativa por lo que no podemos decir tan rápido que tenemos una cola derecha más alargada. Como vemos la desviación típica es de 1010,521 por lo que el intervalo que debe contener el 95 % de los datos es $[618{,}635, 4660, 719]$.

Con esta información podemos apuntar que debe existir una ligera asimetría en la cola derecha porque el mínimo es mayor que el mínimo que nos da el intervalo del 95 % de los datos y el máximo es algo mayor que el máximo del intervalo del 95 %.

Para contrastar esta información tenemos el coeficiente de asimetría que, al ser positivo, nos dice que la distribución presenta asimetría derecha pero podemos ver que es mucho más pequeño que en el resto de variables con lo que la asimetría no es tan pronunciada.

En cuanto a la curtosis podemos ver que es negativa por lo que la distribución es más achatada que una distribución normal.

### Variable 9: demandDeposits

Los estadísticos que nos arroja esta variable son:

```
demandDeposits :
        Media:  256.8477
        Mediana:  224.4
        Desviación típica:  114.5754
        Moda:  120.1
        Kurtosis:  -0.9246782
        Asimetría:  0.5228948
        Mínimo:  105.6
        Máximo:  533
```

Figura 11: Estadísticos de la variable 9 demandDeposits

Como podemos observar tenemos que la mediana es menor que la media. Volvemos a tener una situación que nos lleva a pensar que tenemos una cola derecha más alargada que la izquierda.

Si calculamos el intervalo en el que debe estar el 95 % de los datos ($[27{,}6969, 485{,}9985]$) podemos observar que el máximo es algo más grande que el máximo de dicho intervalo e igualmente ocurre con el mínimo con lo que tenemos un desplazamiento de los datos hacia la derecha.

El hecho viene refrendado por el coeficiente de asimetría que al ser positivo nos indica dicha asimetría derecha.

En cuanto a la curtosis al tener una curtosis negativa estamos ante una distribución más achatada que en el caso de una normal.

### Variable 10: federalFunds

Los estadísticos que nos arroja esta variable son:

```
federalFunds :
        Media:  308.1154
        Mediana:  287.7
        Desviación típica:  59.80509
        Moda:  277.9
        Kurtosis:  -1.40515
        Asimetría:  0.3380899
        Mínimo:  225.8
        Máximo:  412.1
```

Figura 12: Estadísticos de la variable 10 federalFunds

Podemos observar en esta variable el mismo comportamiento que venimos destacando del resto. Tenemos que la mediana es más pequeña que la media lo que nos indica que la cola derecha de la distribución debe ser algo más alargada.

El intervalo en el que el 95 % de los datos debe caer es $[188{,}50522, 427{,}72558]$. Podemos ver que el máximo de este intervalo es ligeramente más grande que el máximo y el mínimo es ligeramente más grande que el del intervalo del 95 % por lo que podemos decir que si existe una asimetría derecha esta no es muy pronunciada.

Podemos contrastar lo que estamos afirmando por el coeficiente de asimetría que es ligeramente positivo, por lo que se confirma lo que estamos razonando de que tenemos una ligera asimetría derecha.

En cuanto a la curtosis tenemos que es negativa y grande por lo que será muy achatada con respecto a una normal de mismos parámetros.

**Variable 11: moneyStock**

Los estadísticos que nos arroja esta variable son:

```
moneyStock :
        Media:  7.549495
        Mediana:  6.64
        Desviación típica:  3.538662
        Moda:  5.45
        Kurtosis:  1.779054
        Asimetría:  1.341457
        Mínimo:  2.86
        Máximo:  20.06
```

Figura 13: Estadísticos de la variable 11 moneyStock

Podemos repetir exactamente el mismo razonamiento que hemos hecho anteriormente para argumentar la asimetría derecha, pero en este caso al comprobar el coeficiente de asimetría podemos ver que es mucho más pronunciada.

En cuanto a la curtosis tenemos el comportamiento opuesto al de la variable anterior teniendo que la distribución es significativamente más puntiaguda o apuntada que su normal de mismos parámetros.

**Variable 12: moneyStock**

Los estadísticos que nos arroja esta variable son:

```
checkableDeposits :
        Media:  813.3304
        Mediana:  796
        Desviación típica:  258.6885
        Moda:  1145.9
        Kurtosis:  -1.401232
        Asimetría:  -0.1719616
        Mínimo:  381.1
        Máximo:  1154.1
```

Figura 14: Estadísticos de la variable 12 checkableDeposits

Este es el primer ejemplo en el que podemos ver como la cola más alargada no va a ser la cola derecha sino la izquierda. En este caso, al igual que los anteriores, podemos ver que la mediana es menor que la media (aunque no mucho proporcionalmente al rango de valores que se toman). Esto podría indicarnos que la cola derecha continúa siendo algo más alargada que la izquierda, pero en este caso podemos apreciar que la moda (el valor más frecuente) es significativamente más grande que la media, por lo que no podemos intuir un comportamiento a priori. Tenemos información contradictoria, una parte nos dice que debemos tener la cola derecha más alargada y la otra nos está diciendo que la izquierda es la que debería ser más alargada.

Para poder estudiar esta situación compleja recurrimos al coeficiente de asimetría. Como podemos ver en este caso es negativo pero ligeramente. Tal y como podíamos pensar la situación está razonablemente equilibrada aunque mostrando una ligera asimetría izquierda.

En cuanto a la curtosis tenemos que es negativa y grande, por lo que estamos ante una distribución mucho más achatada que una distribución normal de mismos parámetros.

**Variable 13: savingsDeposits**

Los estadísticos que nos arroja esta variable son:

```
savingsDeposits :
        Media:  1959.122
        Mediana:  2023.9
        Desviación típica:  720.5311
        Moda:  2112.7
        Kurtosis:  -0.8147294
        Asimetría:  0.3023447
        Mínimo:  868.1
        Máximo:  3550.3
```

Figura 15: Estadísticos de la variable 13 savingsDeposits

El caso de esta variable es singular también. Podemos ver que la mediana es mayor que la media, la moda también por lo que podríamos pensar que la cola más alargada es la izquierda.

Por contra si miramos el intervalo minino y máximo podemos ver que lo más probable es que se extienda más la cola derecha pues el máximo es más grande proporcionalmente a la media que el mínimo.

Para contrastar la información miramos el coeficiente de asimetría y observamos que, aunque es pequeño, es positivo. Esto nos indica que la distribución es asimétrica derecha.

Si analizamos la curtosis podemos ver que es negativa lo que nos indica que la distribución es más achatada que una normal de mismos parámetros.

**Variable 14: tradeCurrencies**

Los estadísticos que nos arroja esta variable son:

```
tradeCurrencies :
        Media:  954.6694
        Mediana:  947.9
        Desviación típica:  372.2925
        Moda:  343.9
        Kurtosis:  -0.2597352
        Asimetría:  -0.1331363
        Mínimo:  175.6
        Máximo:  1758.1
```

Figura 16: Estadísticos de la variable 14 tradeCurrencies

En cuanto a esta variable podemos ver que la media es mayor (ligeramente) que la mediana y la moda es mucho menor que la media. Por contra tenemos que el mínimo es más significativo que el máximo en cuanto al intervalo de valores se refiere. En ese caso a priori no podemos decir nada.

Si miramos el coeficiente de asimetría podemos contrastar esta información pues es muy cercano a cero y en este caso ligeramente negativo por lo que si podemos decir algo es que es ligeramente asimétrica la distribución a la izquierda.

La curtosis es ligeramente negativa por lo que podemos decir que es un poco más achatada que su normal asociada.

Cabe decir tras todo este estudio de estadísticos de las variables que podríamos hacer transformaciones que modifiquen nuestras variables para que cumplan una distribución normal. Por ejemplo es sabido que si tenemos distribuciones con asimetría derecha podemos solucionarlo en la mayoría de casos con una transformación logarítmica. Esto puede esperar hasta que probemos los modelos y comprobemos si es necesario.

### 1.1.2. Estudio de la correlación de las variables

En esta sección vamos a estudiar la correlación entre variables que no son de salida y de dichas variables con la de salida para intentar ver cuales van a ser las más relevantes para nuestro estudio o si hay alguna variable que podamos quitar.

En primer lugar vamos a hacer un estudio de la correlación entre las variables. Nuestro objetivo va a ser obtener aquellas que tienen alta correlación con otras variables, es decir, obtener aquellas variables que son explicadas por otras pues estas las podremos quitar.

Por supuesto todo este estudio habrá que contrastarlo sobre los resultados de la regresión, pero podemos hacer hipótesis previas al ajuste de los modelos.

El código que voy a emplear para el estudio es el siguiente:

```r
obtainCorrelated<-function(varIndex, dataset, threshold=0.9){
  combinations<-combn(varIndex,2)

  correlations<-vector("numeric", dim(combinations)[2])

  for(i in 1:dim(combinations)[2]){
    pair<-combinations[,i]
    correlations[i]<-cor(dataset[,pair[1]], dataset[,pair[2]], method = c("pearson", "kendall", "spearman"))
  }

  for(i in 1:length(correlations)){
    if(abs(correlations[i])>threshold){
      cat("La pareja de variables: ", combinations[,i][1], ", ", combinations[,i][2], " tiene una correlación: ", correlations[i], "\n")
    }
  }
}
```

Figura 17: Código para el estudio de la correlación entre variables.

Para poder eliminar una variable de forma que estemos seguros de que no quitamos información debemos exigir que la correlación entre variables sea alta. En este caso como se puede ver en el código solamente vamos a mostrar las parejas de variables que presenten una correlación alta, en concreto que en valor absoluto sea mayor a 0,9.

Vamos a ver las parejas de variables que cumplen esta condición.

```
La pareja de variables:  2 ,  3  tiene una correlación:  0.9367248
La pareja de variables:  2 ,  4  tiene una correlación:  0.9864352
La pareja de variables:  2 ,  5  tiene una correlación:  0.9874777
La pareja de variables:  2 ,  6  tiene una correlación:  0.9849371
La pareja de variables:  2 ,  7  tiene una correlación:  0.9668119
La pareja de variables:  2 ,  11  tiene una correlación:  0.9692879
La pareja de variables:  3 ,  6  tiene una correlación:  0.9717959
La pareja de variables:  3 ,  7  tiene una correlación:  0.9806363
La pareja de variables:  3 ,  12  tiene una correlación:  -0.9201255
La pareja de variables:  4 ,  5  tiene una correlación:  0.9977408
La pareja de variables:  4 ,  6  tiene una correlación:  0.9527171
La pareja de variables:  4 ,  7  tiene una correlación:  0.928719
La pareja de variables:  4 ,  11  tiene una correlación:  0.9848922
La pareja de variables:  5 ,  6  tiene una correlación:  0.9536684
La pareja de variables:  5 ,  7  tiene una correlación:  0.9295766
La pareja de variables:  5 ,  11  tiene una correlación:  0.9861269
La pareja de variables:  6 ,  7  tiene una correlación:  0.9955387
La pareja de variables:  6 ,  11  tiene una correlación:  0.9300853
La pareja de variables:  7 ,  11  tiene una correlación:  0.904041
La pareja de variables:  8 ,  9  tiene una correlación:  0.99126
La pareja de variables:  8 ,  12  tiene una correlación:  0.9340624
La pareja de variables:  8 ,  14  tiene una correlación:  0.9977185
La pareja de variables:  8 ,  15  tiene una correlación:  0.9531331
La pareja de variables:  9 ,  10  tiene una correlación:  0.9059379
La pareja de variables:  9 ,  12  tiene una correlación:  0.9252762
La pareja de variables:  9 ,  14  tiene una correlación:  0.9826211
La pareja de variables:  9 ,  15  tiene una correlación:  0.9319984
La pareja de variables:  10 ,  12  tiene una correlación:  0.937445
La pareja de variables:  12 ,  13  tiene una correlación:  0.9610653
La pareja de variables:  12 ,  14  tiene una correlación:  0.9189515
La pareja de variables:  12 ,  15  tiene una correlación:  0.9123665
La pareja de variables:  14 ,  15  tiene una correlación:  0.9502578
```

Figura 18: Primer filtrado de la correlación entre variables

Podemos observar claramente que la variable 2 tiene una correlación muy alta con otras variables, por lo que podemos eliminarla al ser explicada por las variables 3,4,5,6,7 y 11.

```
La pareja de variables:  3 ,  6  tiene una correlación:  0.9717959
La pareja de variables:  3 ,  7  tiene una correlación:  0.9806363
La pareja de variables:  3 ,  12  tiene una correlación:  -0.9201255
La pareja de variables:  4 ,  5  tiene una correlación:  0.9977408
La pareja de variables:  4 ,  6  tiene una correlación:  0.9527171
La pareja de variables:  4 ,  7  tiene una correlación:  0.928719
La pareja de variables:  4 ,  11  tiene una correlación:  0.9848922
La pareja de variables:  5 ,  6  tiene una correlación:  0.9536684
La pareja de variables:  5 ,  7  tiene una correlación:  0.9295766
La pareja de variables:  5 ,  11  tiene una correlación:  0.9861269
La pareja de variables:  6 ,  7  tiene una correlación:  0.9955387
La pareja de variables:  6 ,  11  tiene una correlación:  0.9300853
La pareja de variables:  7 ,  11  tiene una correlación:  0.904041
La pareja de variables:  8 ,  9  tiene una correlación:  0.99126
La pareja de variables:  8 ,  12  tiene una correlación:  0.9340624
La pareja de variables:  8 ,  14  tiene una correlación:  0.9977185
La pareja de variables:  8 ,  15  tiene una correlación:  0.9531331
La pareja de variables:  9 ,  10  tiene una correlación:  0.9059379
La pareja de variables:  9 ,  12  tiene una correlación:  0.9252762
La pareja de variables:  9 ,  14  tiene una correlación:  0.9826211
La pareja de variables:  9 ,  15  tiene una correlación:  0.9319984
La pareja de variables:  10 ,  12  tiene una correlación:  0.937445
La pareja de variables:  12 ,  13  tiene una correlación:  0.9610653
La pareja de variables:  12 ,  14  tiene una correlación:  0.9189515
La pareja de variables:  12 ,  15  tiene una correlación:  0.9123665
La pareja de variables:  14 ,  15  tiene una correlación:  0.9502578
```

Figura 19: Correlación entre variables después de eliminar la segunda.

Podemos ver a su vez que la variable 9 tiene una correlación muy alta con las variables 10, 12, 14 y 15 por lo que podemos eliminarla.

```
La pareja de variables:  3 ,  6  tiene una correlación:  0.9717959
La pareja de variables:  3 ,  7  tiene una correlación:  0.9806363
La pareja de variables:  3 ,  12  tiene una correlación:  -0.9201255
La pareja de variables:  4 ,  5  tiene una correlación:  0.9977408
La pareja de variables:  4 ,  6  tiene una correlación:  0.9527171
La pareja de variables:  4 ,  7  tiene una correlación:  0.928719
La pareja de variables:  4 ,  11  tiene una correlación:  0.9848922
La pareja de variables:  5 ,  6  tiene una correlación:  0.9536684
La pareja de variables:  5 ,  7  tiene una correlación:  0.9295766
La pareja de variables:  5 ,  11  tiene una correlación:  0.9861269
La pareja de variables:  6 ,  7  tiene una correlación:  0.9955387
La pareja de variables:  6 ,  11  tiene una correlación:  0.9300853
La pareja de variables:  7 ,  11  tiene una correlación:  0.904041
La pareja de variables:  8 ,  12  tiene una correlación:  0.9340624
La pareja de variables:  8 ,  14  tiene una correlación:  0.9977185
La pareja de variables:  8 ,  15  tiene una correlación:  0.9531331
La pareja de variables:  10 ,  12  tiene una correlación:  0.937445
La pareja de variables:  12 ,  13  tiene una correlación:  0.9610653
La pareja de variables:  12 ,  14  tiene una correlación:  0.9189515
La pareja de variables:  12 ,  15  tiene una correlación:  0.9123665
La pareja de variables:  14 ,  15  tiene una correlación:  0.9502578
```

Figura 20: Correlación entre variables después de eliminar la segunda y la novena.

La variable 4 podemos ver que tiene alta correlación con las variables 5,6,7 y 11 por lo que podemos quitarla también.

```
La pareja de variables:  3 ,  6  tiene una correlación:  0.9717959
La pareja de variables:  3 ,  7  tiene una correlación:  0.9806363
La pareja de variables:  3 ,  12  tiene una correlación:  -0.9201255
La pareja de variables:  5 ,  6  tiene una correlación:  0.9536684
La pareja de variables:  5 ,  7  tiene una correlación:  0.9295766
La pareja de variables:  5 ,  11  tiene una correlación:  0.9861269
La pareja de variables:  6 ,  7  tiene una correlación:  0.9955387
La pareja de variables:  6 ,  11  tiene una correlación:  0.9300853
La pareja de variables:  7 ,  11  tiene una correlación:  0.904041
La pareja de variables:  8 ,  12  tiene una correlación:  0.9340624
La pareja de variables:  8 ,  14  tiene una correlación:  0.9977185
La pareja de variables:  8 ,  15  tiene una correlación:  0.9531331
La pareja de variables:  10 ,  12  tiene una correlación:  0.937445
La pareja de variables:  12 ,  13  tiene una correlación:  0.9610653
La pareja de variables:  12 ,  14  tiene una correlación:  0.9189515
La pareja de variables:  12 ,  15  tiene una correlación:  0.9123665
La pareja de variables:  14 ,  15  tiene una correlación:  0.9502578
```

Figura 21: Correlación entre variables después de eliminar la segunda, la novena y la cuarta.

La tercera variable sigue manteniendo una alta correlación con las variables 6,7 y 12, por lo que podemos eliminarla.

```
La pareja de variables:  5 ,  6  tiene una correlación:  0.9536684
La pareja de variables:  5 ,  7  tiene una correlación:  0.9295766
La pareja de variables:  5 ,  11  tiene una correlación:  0.9861269
La pareja de variables:  6 ,  7  tiene una correlación:  0.9955387
La pareja de variables:  6 ,  11  tiene una correlación:  0.9300853
La pareja de variables:  7 ,  11  tiene una correlación:  0.904041
La pareja de variables:  8 ,  12  tiene una correlación:  0.9340624
La pareja de variables:  8 ,  14  tiene una correlación:  0.9977185
La pareja de variables:  8 ,  15  tiene una correlación:  0.9531331
La pareja de variables:  10 ,  12  tiene una correlación:  0.937445
La pareja de variables:  12 ,  13  tiene una correlación:  0.9610653
La pareja de variables:  12 ,  14  tiene una correlación:  0.9189515
La pareja de variables:  12 ,  15  tiene una correlación:  0.9123665
La pareja de variables:  14 ,  15  tiene una correlación:  0.9502578
```

Figura 22: Correlación entre variables después de eliminar la segunda, la novena, la cuarta y la tercera.

Podemos ver que la variable 12 y 5 tienen una alta correlación con otras tres que además no se comparten por lo que podemos eliminar también ambas variables.

```
La pareja de variables:  6 ,  7  tiene una correlación:  0.9955387
La pareja de variables:  6 ,  11  tiene una correlación:  0.9300853
La pareja de variables:  7 ,  11  tiene una correlación:  0.904041
La pareja de variables:  8 ,  14  tiene una correlación:  0.9977185
La pareja de variables:  8 ,  15  tiene una correlación:  0.9531331
La pareja de variables:  14 ,  15  tiene una correlación:  0.9502578
```

Figura 23: Correlación entre variables después de eliminar la segunda, la novena, la cuarta, la tercera, la duodécima y la quinta.

Podemos observar que la variable 6 y 8 tienen alta correlación con otras dos variables no compartidas, por lo que podemos eliminar ambas.

```
La pareja de variables:  7 ,  11  tiene una correlación:  0.904041
La pareja de variables:  14 ,  15  tiene una correlación:  0.9502578
```

Figura 24: Correlación entre variables después de eliminar la segunda, la novena, la cuarta, la tercera, la duodécima, la quinta, la sexta y la octava.

Finalmente nos hemos quedado con dos parejas únicamente por lo que ya no está tan claro la eliminación de dichas variables. Vamos a parar por tanto el proceso de eliminación en este punto.

Veamos la correlación entre variables que nos queda con las que hemos decidido mantener.

```
La pareja de variables:  1 ,  7  tiene una correlación:  0.6250722
La pareja de variables:  1 ,  10  tiene una correlación:  -0.6408934
La pareja de variables:  1 ,  11  tiene una correlación:  0.4560397
La pareja de variables:  1 ,  13  tiene una correlación:  -0.7161397
La pareja de variables:  1 ,  14  tiene una correlación:  -0.5672793
La pareja de variables:  1 ,  15  tiene una correlación:  -0.4763829
La pareja de variables:  7 ,  10  tiene una correlación:  -0.8168239
La pareja de variables:  7 ,  11  tiene una correlación:  0.904041
La pareja de variables:  7 ,  13  tiene una correlación:  -0.862284
La pareja de variables:  7 ,  14  tiene una correlación:  -0.8273161
La pareja de variables:  7 ,  15  tiene una correlación:  -0.8651072
La pareja de variables:  10 ,  11  tiene una correlación:  -0.7072023
La pareja de variables:  10 ,  13  tiene una correlación:  0.8731687
La pareja de variables:  10 ,  14  tiene una correlación:  0.8663811
La pareja de variables:  10 ,  15  tiene una correlación:  0.8259606
La pareja de variables:  11 ,  13  tiene una correlación:  -0.8347681
La pareja de variables:  11 ,  14  tiene una correlación:  -0.7063516
La pareja de variables:  11 ,  15  tiene una correlación:  -0.8060014
La pareja de variables:  13 ,  14  tiene una correlación:  0.7864516
La pareja de variables:  13 ,  15  tiene una correlación:  0.8127696
La pareja de variables:  14 ,  15  tiene una correlación:  0.9502578
```

Figura 25: Correlación entre variables después de hacer la limpieza de variables.

Finalmente por tanto nos hemos quedado con las variables 1,7,10,11,13,14 y 15.

Sobre estas variables merece la pena estudiar su correlación con la variable de salida para ver que seguimos teniendo variables que explican el comportamiento de la salida.

```
La correlación de la variable  1  con la salida es de:  0.4507399
La correlación de la variable  7  con la salida es de:  0.9106219
La correlación de la variable  10  con la salida es de:  -0.7021663
La correlación de la variable  11  con la salida es de:  0.9946502
La correlación de la variable  13  con la salida es de:  -0.8280948
La correlación de la variable  14  con la salida es de:  -0.693953
La correlación de la variable  15  con la salida es de:  -0.7931364
```

Figura 26: Correlación de las variables no eliminadas con la variable de salida.

Podemos ver que las variables 7, 11, 13 y 15 tienen una correlación en valor absoluto mayor a 0,75 con lo que estas variables nos van a resultar de mucho interés a la hora de realizar un modelo de regresión lineal.

Todas estas hipótesis serán contrastadas en la sección de regresión al ajustar los modelos.

### 1.1.3. Valores perdidos

En cuanto a los valores perdidos vamos a comprobar en primer lugar si tenemos valores NA.

```
> which(is.na(dataset_regresion))
integer(0)
```

Figura 27: Código y resultados para comprobar si tenemos valores perdidos.

Como podemos comprobar no tenemos ningún valor perdido.

### 1.1.4. Outliers

Vamos a comprobar si tenemos outliers en nuestro conjunto de datos.

En primer lugar vamos a ver un pairplot de las variables para ver si podemos distinguir algo visualmente.

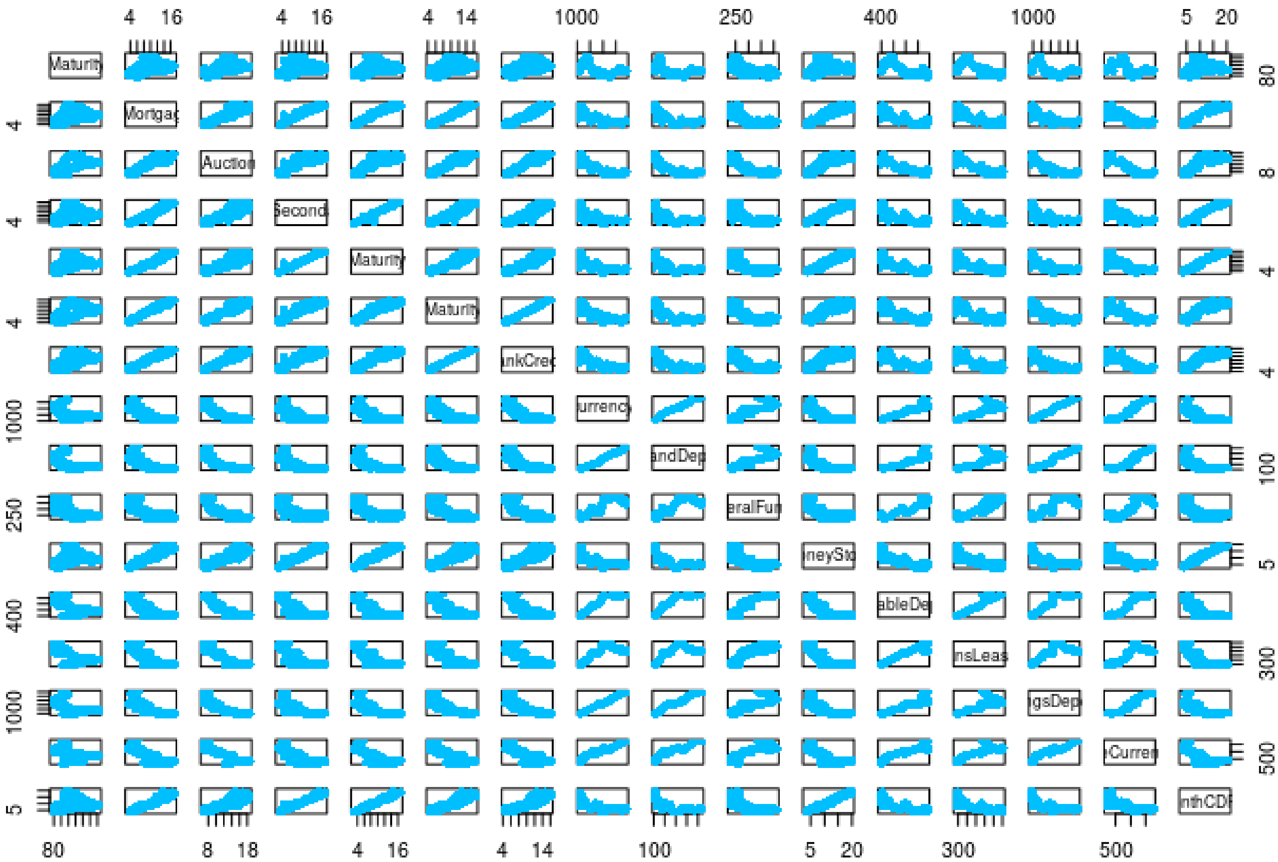

Figura 28: Scatterplot de todas las variables dos a dos

Visualmente con tantas variables no podemos destacar ninguna anomalía. Vamos a centrarnos solamente en las variables que hemos decidido quedarnos del estudio previo.

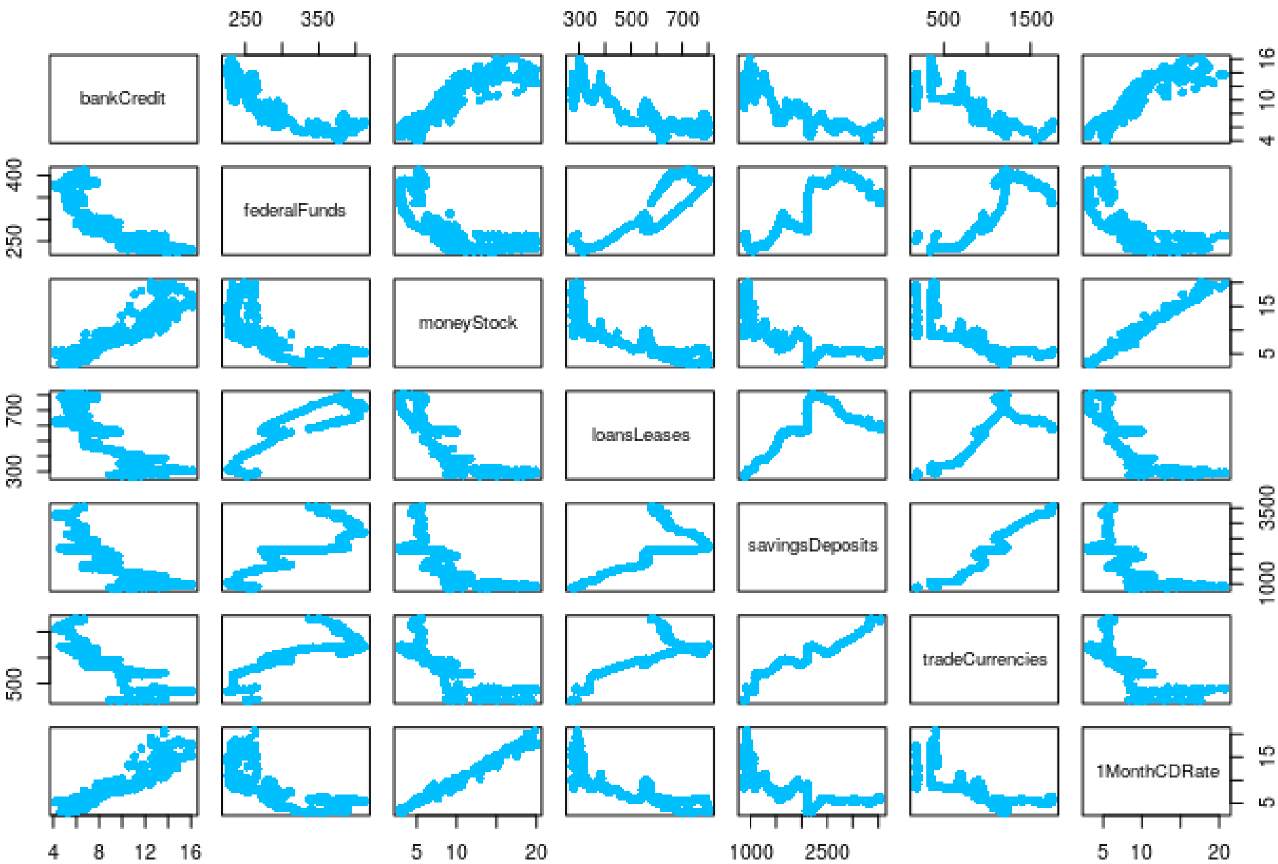


Figura 29: Scatterplot de las variables seleccionadas

Como podemos ver no podemos destacar ningún outlier en el conjunto de datos de forma visual.

De paso podemos ver que hemos hecho una selección de variables adecuada pues podemos observar una clara relación lineal con la salida.

Vamos a ver ahora un boxplot de todas las variables.

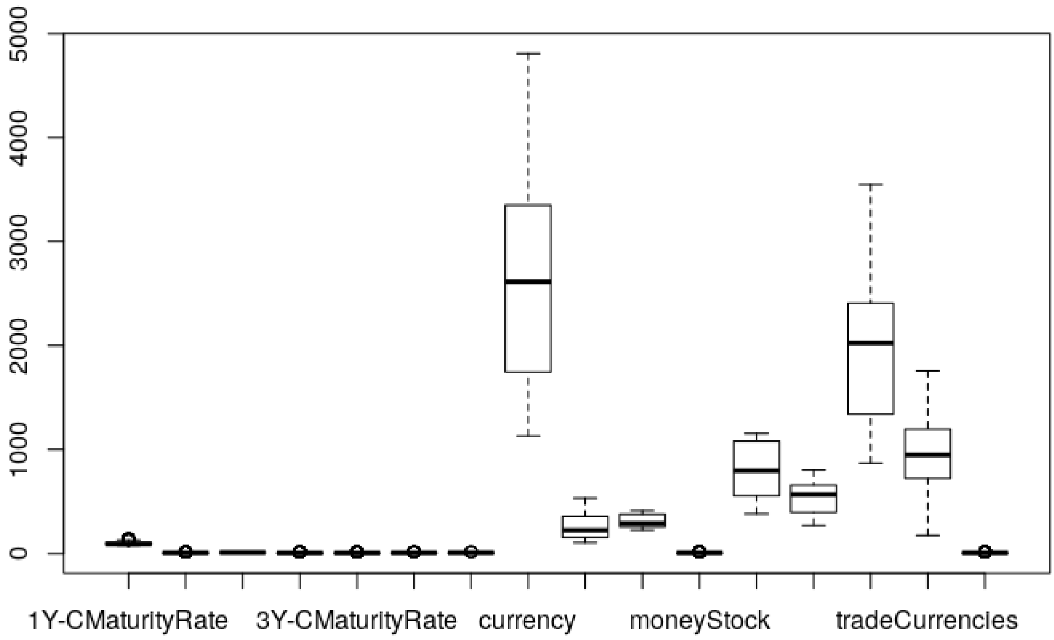


Figura 30: Boxplot de las todas las variables

Podemos ver que por la diferencia de escalas no todas las variables son visibles. En las que podemos observar de forma adecuada podemos ver que no hay outliers que nos destaquen fuera del rango intercuartil. Por tanto podemos eliminarlas del boxplot para reducir la escala y poder ver el resto de variables.

Para el siguiente boxplot vamos a quitar las variables 8, 9, 10, 12, 13, 14 y 15.

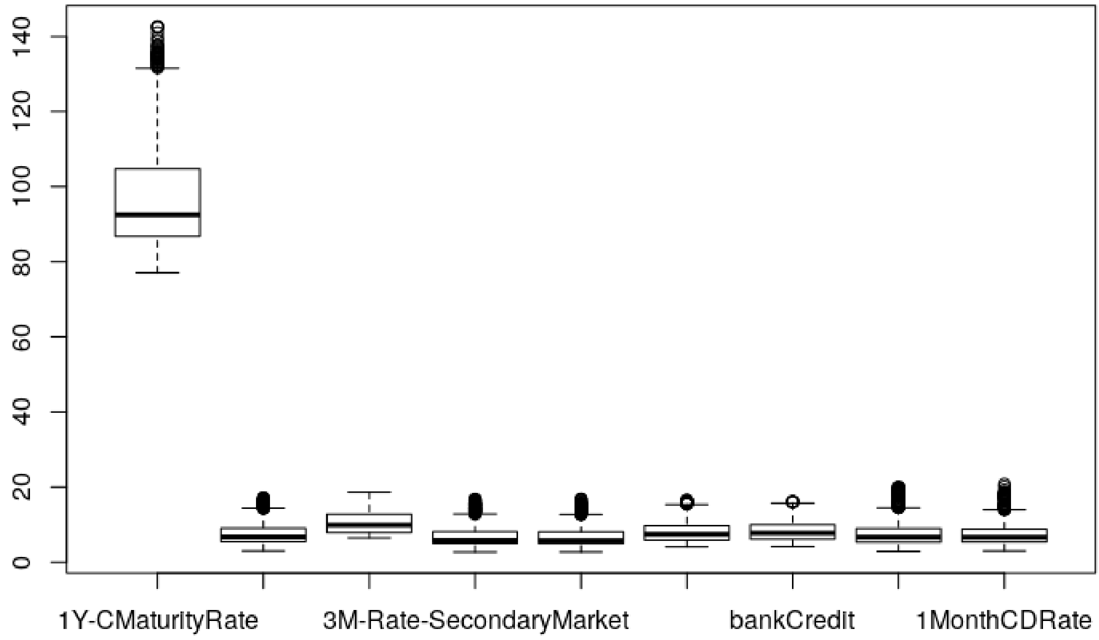


Figura 31: Boxplot quitando las variables 8, 9, 10, 12, 13, 14 y 15

Podemos observar que la primera variable tiene una escala mayor que el resto, por lo que para poder continuar la tendremos que quitar. Si observamos sus valores podemos ver que tenemos algunas anomalías por encima. Al tener tantos valores por encima no podemos decir de forma tan clara que estos valores son anómalos pues podría ser un comportamiento esperado de la variable y por tanto a priori no debemos eliminar dichos valores.

Eliminamos además la variable 1 para continuar con el estudio.

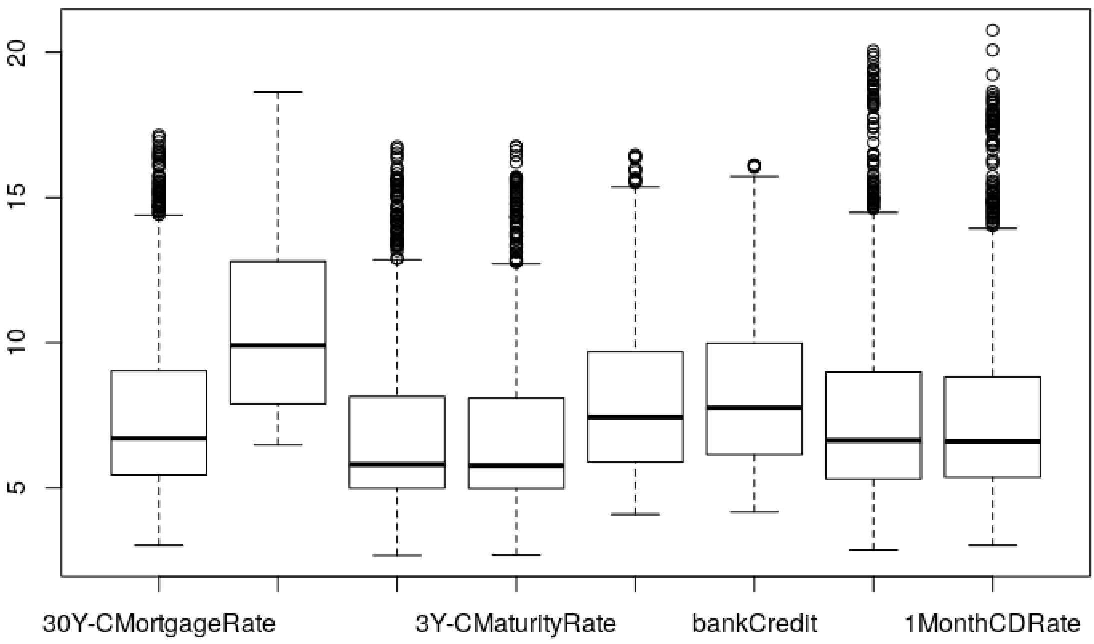


Figura 32: Boxplot quitando las variables 1, 8, 9, 10, 12, 13, 14 y 15

Podemos observar que en este último boxplot tenemos muchas más anomalías, de hecho, todas las variables poseen valores fuera del rango intercuartil menos para la segunda variable.

Podemos observar que la concentración de valores fuera de rango es muy grande por lo que no debemos eliminar dichos valores. Además este estudio es de todas las variables y no de las que hemos seleccionado para quedarnos. Veamos su boxplot.

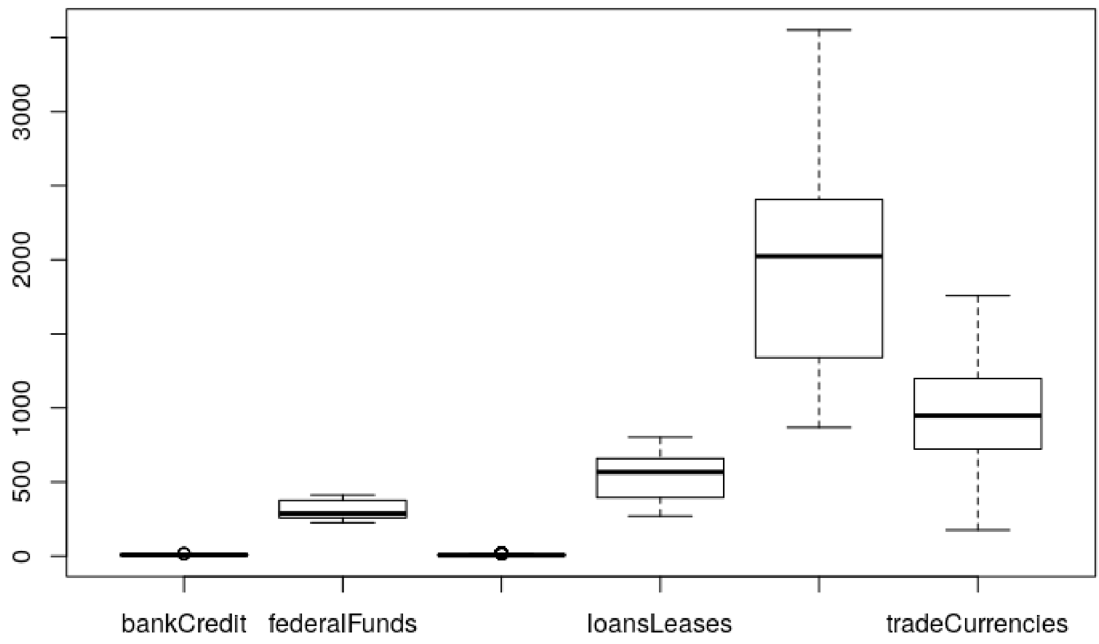


Figura 33: Boxplot manteniendo las variables seleccionadas.

Podemos ver que en las variables seleccionadas tenemos dos que poseen anomalías, vamos a estudiarlas en un boxplot aislado.

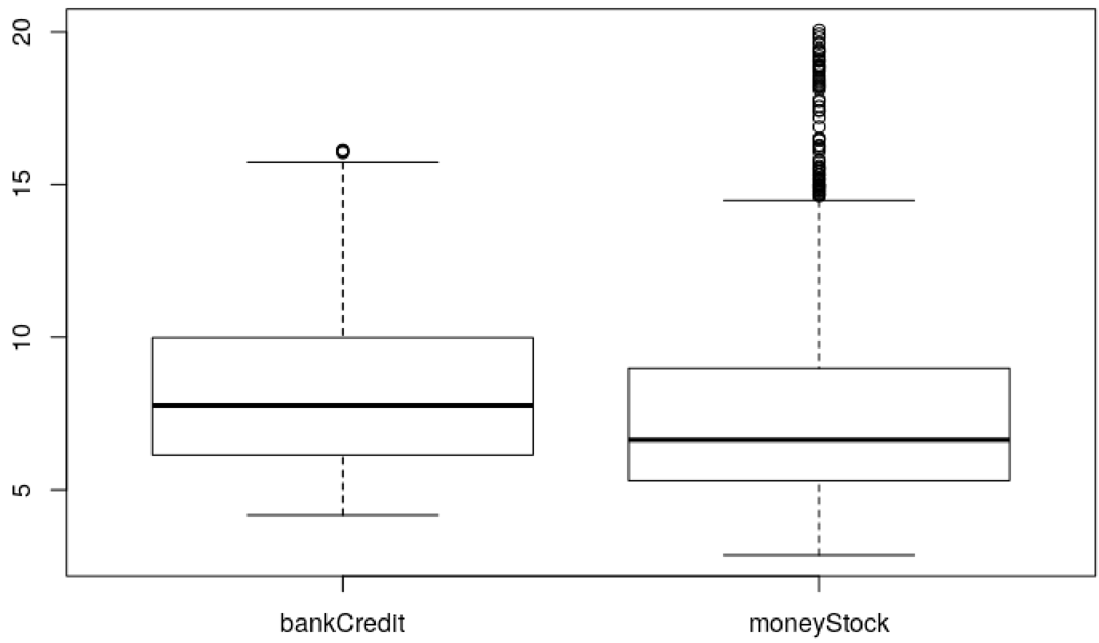


Figura 34: Boxplot de las variables que parecen tener anomalías.

Como podemos observar tenemos que la variable moneyStock tiene valores anómalos pero muy densos por lo que no debemos quitarlos. En el caso de bankCredit vamos a estudiar su scatterplot.

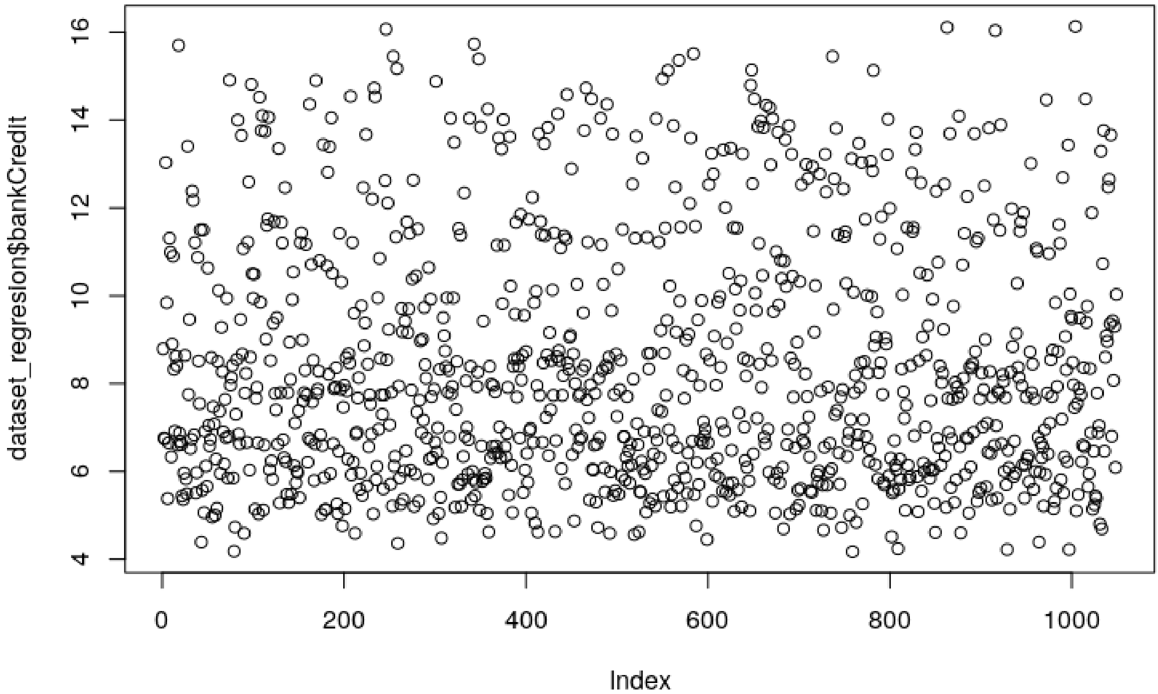


Figura 35: ScatterPlot de la variable bankCredit

Podemos ver que las anomalías que nos encontramos no son tal pues es un conjunto distribuido de forma casi uniforme.

### 1.1.5. Distribución de las variables

Vamos a ver en primer lugar unos histogramas de las variables.

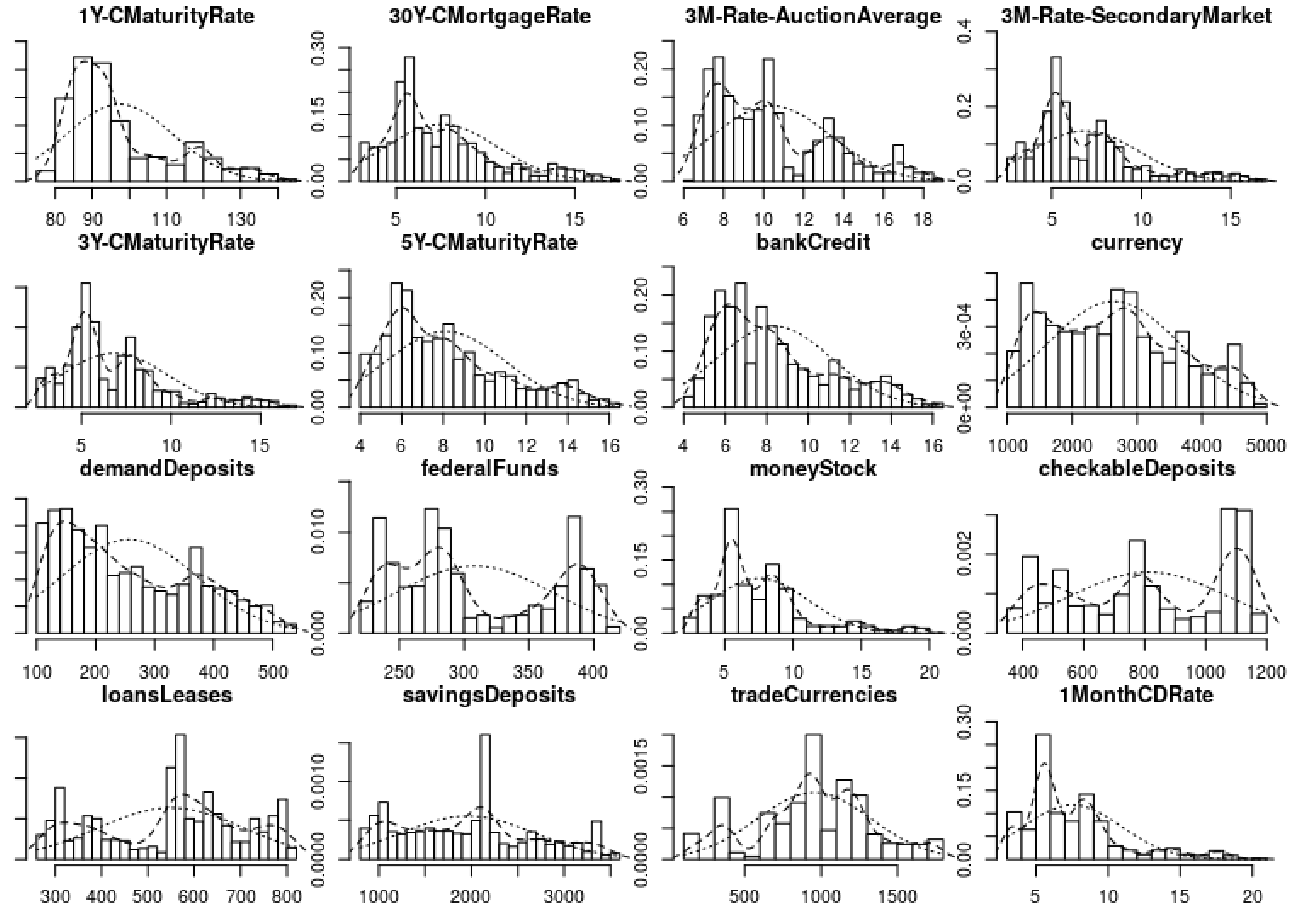


Figura 36: Histograma de todas las variables

Como podemos ver hemos acertado en el estudio previo de los estadísticos y podemos corroborar que la mayoría de variables tienen la cola derecha de su distribución más alargada.

Podemos ver que las distribuciones no se parecen para nada a una normal de forma visual, aunque para poder estar seguros vamos a hacer un test de normalidad.

Para esto vamos a utilizar el test de Wilcoxon. Veamos los resultados.

```
Test de normalidad para la variable  1
P-valor:  3.404098e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 37: Test de normalidad para la variable 1.

```
Test de normalidad para la variable  2
P-valor:  3.401871e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 38: Test de normalidad para la variable 2.

```
Test de normalidad para la variable  3
P-valor:  3.401848e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 39: Test de normalidad para la variable 3.

```
Test de normalidad para la variable  4
P-valor:  3.400497e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 40: Test de normalidad para la variable 4.

```
Test de normalidad para la variable  5
P-valor:  3.400346e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 41: Test de normalidad para la variable 5.

```
Test de normalidad para la variable  6
P-valor:  3.402266e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 42: Test de normalidad para la variable 6.

```
Test de normalidad para la variable  7
P-valor:  3.402179e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 43: Test de normalidad para la variable 7.

```
Test de normalidad para la variable  8
P-valor:  3.404102e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 44: Test de normalidad para la variable 8.

```
Test de normalidad para la variable  9
P-valor:  3.403861e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 45: Test de normalidad para la variable 9.

```
Test de normalidad para la variable  10
P-valor:  3.402683e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 46: Test de normalidad para la variable 10.

```
Test de normalidad para la variable  11
P-valor:  3.40101e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 47: Test de normalidad para la variable 11.

```
Test de normalidad para la variable  12
P-valor:  3.403832e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 48: Test de normalidad para la variable 12.

```
Test de normalidad para la variable  13
P-valor:  3.403666e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 49: Test de normalidad para la variable 13.

```
Test de normalidad para la variable  14
P-valor:  3.404037e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 50: Test de normalidad para la variable 14.

```
Test de normalidad para la variable  15
P-valor:  3.403915e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 51: Test de normalidad para la variable 15.

```
Test de normalidad para la variable  16
P-valor:  3.395458e-173
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 52: Test de normalidad para la variable 16.

Como podemos ver por los resultados del test podemos rechazar en todos los casos la hipótesis nula por lo que podemos decir que ninguna de las variables sigue una distribución normal, tal y como hemos podido ver de forma gráfica.

## 1.2. Conjunto de Clasificación

El conjunto con el que vamos a atacar el problema de clasificación es el conjunto de datos "heart". Vamos a analizar este conjunto de datos antes de abordar el problema.

El problema dispone de 14 variables y 270 observaciones, siendo la última la clase a la que pertenece cada instancia.

Vamos a ver los tipos de las variables.

```
Age :  integer
Sex :  integer
ChestPainType :  integer
RestBloodPressure :  integer
SerumCholestoral :  integer
FastingBloodSugar :  integer
ResElectrocardiographic :  integer
MaxHeartRate :  integer
ExerciseInduced :  integer
Oldpeak :  numeric
Slope :  integer
MajorVessels :  integer
Thal :  integer
Class :  integer
```

Figura 53: Tipos de las variables.

Podemos observar que todas las variables son de tipo integer, es decir, de tipo entero y que hay una variable de tipo numeric, o lo que es lo mismo, de tipo real.

Vamos a realizar el estudio del conjunto completo dividiéndolo en dos partes, la variable que nos indica la clase y el resto de variables.

### 1.2.1. Estudio de los estadísticos

#### Salida

Vamos a ver los estadísticos que nos da la variable de salida o lo que es lo mismo, la clase asociada a cada instancia.

```
Class :
        Media:  1.444444
        Mediana:  1
        Desviación típica:  0.4978268
        Moda:  1
        Kurtosis:  -1.957763
        Asimetría:  0.2223657
        Mínimo:  1
        Máximo:  2
```

Figura 54: Estadísticos de la variable de salida.

Para poder analizar los estadísticos de esta variable tenemos que tener en cuenta que es una variable que sólo tiene dos posibles valores: 1 y 2.

La media es 1,444444 por lo que podemos decir que seguramente encontraremos un número mayor de instancias de la clase 1 que de la clase 2, pues si fuera al contrario la media debería ser más grande que 1,5. Este hecho es también contrastable por el valor de la moda, o lo que es lo mismo, el valor más frecuente. La moda es 1 por lo que ya sabemos que hay más instancias de la clase 1 que de la 2. En este caso estudiar la distribución carece de sentido por tomar únicamente dos valores, por lo que no estudiaremos la asimetría ni la curtosis.

**Variable 1: Age**

Vamos a estudiar los estadísticos que nos arroja la variable.

```
Age :
        Media:  54.43333
        Mediana:  55
        Desviación típica:  9.109067
        Moda:  54
        Kurtosis:  -0.574982
        Asimetría:  -0.1618018
        Mínimo:  29
        Máximo:  77
```

Figura 55: Estadísticos de la variable 1.

En primer lugar cabe decir que esta variable es de tipo entero. Podemos ver que el intervalo en el que se mueven los valores es $[29, 77]$.

Como podemos ver la media, la mediana y la moda están muy próximas entre sí por lo que la distribución debe estar centrada. Este hecho se puede contrastar con el coeficiente de asimetría que aunque es negativo está muy cercano a 0 lo que nos indica que la distribución es prácticamente simétrica.

En cuanto a la curtosis podemos ver que es negativa por lo que la distribución es más achatada que su normal homóloga en parámetros.

**Variable 2: Sex**

Vamos a estudiar los estadísticos que nos arroja la variable.

```
Sex :
        Media:  0.6777778
        Mediana:  1
        Desviación típica:  0.4681954
        Moda:  1
        Kurtosis:  -1.432815
        Asimetría:  -0.7566044
        Mínimo:  0
        Máximo:  1
```

Figura 56: Estadísticos de la variable 2.

Esta variable es de tipo entero y sólo puede tomar dos valores: 0 y 1, siendo cada uno de los números el sexo masculino o femenino. Podemos ver que la media es 0,6777778 por lo que hay más valores de 1 que de 0 y por tanto debe haber un género predominante sobre el otro dentro de los datos.

Al ser una variable que sólo puede tomar dos valores no tiene sentido que estudiemos la distribución de la variable.

**Variable 3: ChestPainType**

Vamos a estudiar los estadísticos que nos arroja la variable.

```
ChestPainType :
        Media:  3.174074
        Mediana:  3
        Desviación típica:  0.95009
        Moda:  4
        Kurtosis:  -0.33309
        Asimetría:  -0.8690273
        Mínimo:  1
        Máximo:  4
```

Figura 57: Estadísticos de la variable 3.

Esta variable es de tipo entero y toma valores en el intervalo $[1, 4]$ indicando el tipo de dolor de pecho que posee el paciente. La media y la mediana no tienen sentido en este caso pues no son fácilmente interpretables. Al no ser algo binario no podemos establecer en qué grado aparece un valor sobre otro.

Lo que si nos puede ser útil de analizar es la moda. Podemos ver que toma valor 4, o lo que es lo mismo, el dolor de pecho más típico es aquel que va asociado con el número 4.

Algo que sí podemos decir sobre esta variable es que está desplazada hacia la derecha y debería de tener una cola más pesada a la derecha y más alargada a la izquierda. Esto lo podemos intuir pues la desviación típica es cercana a 1 y por tanto el intervalo que contiene el 95 % de los datos debería de ser $[1, 5]$ aproximadamente, por lo que vemos que está desplazada a la derecha con respecto al intervalo $[1, 4]$.

Este dato es comprobable también por el coeficiente de asimetría que es negativo y por tanto nos indica que la distribución es asimétrica a la izquierda. En cuanto a la curtosis podemos ver que es negativa y por tanto la distribución debe ser algo más achatada que una distribución normal de mismos parámetros.

**Variable 4: RestBloodPressure**

Vamos a estudiar los estadísticos que nos arroja la variable.

```
RestBloodPressure :
        Media:  131.3444
        Mediana:  130
        Desviación típica:  17.86161
        Moda:  120
        Kurtosis:  0.8552338
        Asimetría:  0.7146087
        Mínimo:  94
        Máximo:  200
```

Figura 58: Estadísticos de la variable 4.

Esta es también una variable de tipo entero. Podemos ver que toma valores en el intervalo

[94, 200]. La media y la mediana están muy próximas entre sí por lo que podemos pensar que la distribución está centrada. Por contra si miramos la moda es de 120 por lo que podemos intuir una asimetría en la distribución.

Si miramos el coeficiente de asimetría tenemos que es positivo por lo que la distribución es asimétrica a la derecha. Esto nos indica que la cola de la derecha es algo más alargada que la de la izquierda. En cuanto a la curtosis podemos ver que es positiva por lo que la distribución es más apuntada que su distribución normal asociada de mismos parámetros.

### Variable 5: SerumCholestoral

Vamos a estudiar los estadísticos que nos arroja la variable.

```
SerumCholestoral :
        Media:  249.6593
        Mediana:  245
        Desviación típica:  51.68624
        Moda:  234
        Kurtosis:  4.725721
        Asimetría:  1.170601
        Mínimo:  126
        Máximo:  564
```

Figura 59: Estadísticos de la variable 5.

Esta variable también es de tipo entero. El intervalo en el que toma valores es [126, 564]. Si observamos la media y la mediana están próximas, por contra la moda es significativamente menor que ambas por lo que nos puede dar a intuir una asimetría derecha con una cola de la distribución algo más alargada.

Este hecho viene dado por el coeficiente de asimetría. Como podemos ver es positivo y además lo suficientemente grande como para que podamos decir que la asimetría es notable. Este hecho puede no ser tan sencillo de deducir a partir de los datos porque la variable es de tipo entero. Esto puede hacer que haya algunos valores más frecuentes que otros y por tanto la asimetría no sea tan evidente a través de los estadísticos básicos.

Otro valor muy significativo es la curtosis que podemos ver que es positiva y muy grande, lo que nos está indicando que la distribución es extremadamente puntiaguda y por tanto podemos pensar que el argumento anterior (hay valores mucho más frecuentes que otros) es algo a tener en cuenta.

### Variable 6: FastingBloodSugar

Vamos a estudiar los estadísticos que nos arroja la variable.

```
FastingBloodSugar :
        Media:  0.1481481
        Mediana:  0
        Desviación típica:  0.3559065
        Moda:  0
        Kurtosis:  1.887507
        Asimetría:  1.969892
        Mínimo:  0
        Máximo:  1
```

Figura 60: Estadísticos de la variable 6.

La variable FastingBloodSugar es de tipo entero y sólo puede tomar dos valores: 0 y 1. Si vemos la media tenemos que es 0,1481481 por lo que podemos decir que hay muchas más instancias que toman valor 0 frente a las que toman valor 1. El equilibrio, es decir si hubiera el mismo número de 0 que de 1, sería 0,5 por lo que podemos ver el desequilibrio de valores. La moda es 0 también por lo que corroboramos que este es el valor más frecuente.

En cuanto a la curtosis y asimetría no nos aportan más información que la ya razonada pues estamos ante una variable binaria.

**Variable 7: ResElectrocardiographic**

Vamos a estudiar los estadísticos que nos arroja la variable.

```
ResElectrocardiographic :
        Media:  1.022222
        Mediana:  2
        Desviación típica:  0.9978912
        Moda:  2
        Kurtosis:  -1.998017
        Asimetría:  -0.04420775
        Mínimo:  0
        Máximo:  2
```

Figura 61: Estadísticos de la variable 7.

Esta variable es también de tipo entero y podemos ver que toma valores en el intervalo $[0, 2]$ por lo que sólo puede tomar los valores 0, 1 y 2.La media es cercana a 1 por lo que pueden pasar dos cosas con los valores de esta variable: la primera es que los valores que toman las 3 variables estén equilibrados y la segunda es que las ocurrencias de 0 y 2 estén equilibradas.

La moda es 2, cosa que podemos ver pues la media es ligeramente superior a 1. Como tenemos una variable muy simple (sólo toma tres valores) el coeficiente de asimetría no nos da más información que la media, pudiendo corroborar que la distribución es simétrica. En cuanto a la curtosis podemos ver que es mucho más achatada que su distribución normal asociada, por lo que podemos intuir que habrá menos ocurrencias del valor 1 frente al 0 y 2.

**Variable 8: MaxHeartRate**

Vamos a estudiar los estadísticos que nos arroja la variable.

```
MaxHeartRate :
        Media:  149.6778
        Mediana:  153.5
        Desviación típica:  23.16572
        Moda:  162
        Kurtosis:  -0.1445826
        Asimetría:  -0.5218874
        Mínimo:  71
        Máximo:  202
```

Figura 62: Estadísticos de la variable 8.

La variable MaxHeartRate es de tipo entero. Podemos ver que toma valores en el intervalo $[71, 202]$. La media y la mediana están razonablemente cercanas pero la moda es sustancialmente mayor a ambas, lo que nos está indicando que la cola izquierda debe ser algo más alargada que la derecha. Esto podríamos intentar razonarlo también calculando el intervalo en el que deberíamos encontrar el 95 % de los datos ($[103{,}34636, 196{,}00924]$) pero en este caso no nos otorga mucha información.

El coeficiente de asimetría es negativo lo que nos indica una asimetría a la izquierda. La curtosis es negativa pero muy cercana a 0 por lo que podemos decir que la distribución estará ligeramente achatada.

### Variable 9: ExerciseInduced

Vamos a estudiar los estadísticos que nos arroja la variable.

```
ExerciseInduced :
        Media:  0.3296296
        Mediana:  0
        Desviación típica:  0.4709516
        Moda:  0
        Kurtosis:  -1.485858
        Asimetría:  0.7208357
        Mínimo:  0
        Máximo:  1
```

Figura 63: Estadísticos de la variable 9.

Esta variable es también de tipo entero y podemos ver que toma valores en el intervalo $[0, 1]$ por lo que es binaria. La media es menor que 0,5 lo que nos indica que hay más ocurrencias del valor 0 que del valor 1, cuestión que se refuerza al ver que la mediana y la moda son 0.

En cuanto a la curtosis y el coeficiente de asimetría carecen de sentido al no aportar mayor información en una variable binaria.

### Variable 10: Oldpeak

Vamos a estudiar los estadísticos que nos arroja la variable.

```
Oldpeak :
        Media:  8.9
        Mediana:  4
        Desviación típica:  11.00394
        Moda:  0
        Kurtosis:  2.748951
        Asimetría:  1.552755
        Mínimo:  0
        Máximo:  62
```

Figura 64: Estadísticos de la variable 10.

Esta es la única variable del conjunto de datos que es de tipo real. Toma valores dentro del intervalo $[0, 62]$ y podemos ver que la media es 8,9. La mediana es tan solo 4 por lo que podemos intuir que la cola derecha será muy alargada comparado con la izquierda. En cuanto a la moda podemos ver que el valor más frecuente es 0.

Si estudiamos la curtosis y asimetría podemos ver que la distribución es muy asimétrica a la derecha, lo que sustenta el razonamiento de la cola derecha más alargada. La curtosis es positiva y muy grande por lo que podemos ver que la distribución es mucho más puntiaguda o apuntada que la distribución de una normal de mismos parámetros.

**Variable 11: Slope**

Vamos a estudiar los estadísticos que nos arroja la variable.

```
Slope :
        Media:  1.585185
        Mediana:  2
        Desviación típica:  0.6143898
        Moda:  1
        Kurtosis:  -0.6351525
        Asimetría:  0.5371309
        Mínimo:  1
        Máximo:  3
```

Figura 65: Estadísticos de la variable 11.

La variable Slope es de tipo entero y toma valores dentro del intervalo $[1, 3]$, por lo que sólo puede tomar 3 valores distintos: 1, 2, y 3. La media es 1,585185 lo que nos indica que hay una descompensación hacia el 1 teniendo más ocurrencias de éste valor.

Este hecho lo podemos contrastar con el valor de la moda que es 1. En cuanto a la mediana podemos ver que es 2 lo que nos hace pensar que aproximadamente puede haber el mismo número de ocurrencias del valor 1 que de los valores 2 y 3 juntos.

El coeficiente de asimetría soporta estos razonamientos, pues es positivo indicándonos que la cola derecha es más alargada que la izquierda. En cuanto a la curtosis podemos ver que es negativa indicando que la distribución es más achatada que una normal de mismos parámetros.

**Variable 12: MajorVessels**

Vamos a estudiar los estadísticos que nos arroja la variable.

```
MajorVessels :
        Media:  0.6703704
        Mediana:  0
        Desviación típica:  0.9438964
        Moda:  0
        Kurtosis:  0.2464212
        Asimetría:  1.19648
        Mínimo:  0
        Máximo:  3
```

Figura 66: Estadísticos de la variable 12.

Esta variable es de tipo entero y toma valores en el intervalo $[0, 3]$, por lo que sólo puede tomar 4 valores distintos. Podemos ver que la media está entre 0 y 1 lo que nos lleva a pensar que debe haber más ocurrencias de valores 0 que del resto. La moda y la mediana corroboran esta suposición.

La asimetría es positiva, lo que nos indica que la cola derecha debe ser más alargada que la izquierda. Por otro lado la curtosis es positiva pero no muy grande por lo que la distribución es ligeramente más apuntada que una normal de mismos parámetros.

**Variable 13: Thal**

Vamos a estudiar los estadísticos que nos arroja la variable.

```
Thal :
        Media:  4.696296
        Mediana:  3
        Desviación típica:  1.940659
        Moda:  3
        Kurtosis:  -1.895968
        Asimetría:  0.284084
        Mínimo:  3
        Máximo:  7
```

Figura 67: Estadísticos de la variable 13.

La variable Thal es de tipo entero también y toma valores en el intervalo $[3, 7]$. La media es 4,696296 y la mediana y la moda 3. Esto nos deja pensar que la distribución será aproximadamente simétrica pues no se percibe por estos estadísticos un desbalanceo acusado. El coeficiente de asimetría, aunque positivo, posee un valor muy cercano a cero. Por otro lado la curtosis es negativa y por tanto es más achatada que su distribución normal homóloga.

### 1.2.2. Estudio de la correlación de las variables

Vamos a estudiar la correlación entre variables por si pudiéramos encontrar, como en el caso del conjunto de regresión, variables altamente correladas que se puedan eliminar.

Veamos la correlación entre las variables:

```
La pareja de variables:  1 ,  2  tiene una correlación:  -0.09440069
La pareja de variables:  1 ,  3  tiene una correlación:  0.09691976
La pareja de variables:  1 ,  4  tiene una correlación:  0.2730528
La pareja de variables:  1 ,  5  tiene una correlación:  0.2200563
La pareja de variables:  1 ,  6  tiene una correlación:  0.123458
La pareja de variables:  1 ,  7  tiene una correlación:  0.128171
La pareja de variables:  1 ,  8  tiene una correlación:  -0.4022154
La pareja de variables:  1 ,  9  tiene una correlación:  0.09829655
La pareja de variables:  1 ,  10  tiene una correlación:  0.1803817
La pareja de variables:  1 ,  11  tiene una correlación:  0.1597736
La pareja de variables:  1 ,  12  tiene una correlación:  0.3560806
La pareja de variables:  1 ,  13  tiene una correlación:  0.1060998
La pareja de variables:  2 ,  3  tiene una correlación:  0.03463555
La pareja de variables:  2 ,  4  tiene una correlación:  -0.06269339
La pareja de variables:  2 ,  5  tiene una correlación:  -0.2016475
La pareja de variables:  2 ,  6  tiene una correlación:  0.04213967
La pareja de variables:  2 ,  7  tiene una correlación:  0.03925345
La pareja de variables:  2 ,  8  tiene una correlación:  -0.07610146
La pareja de variables:  2 ,  9  tiene una correlación:  0.1800218
La pareja de variables:  2 ,  10  tiene una correlación:  0.1178308
La pareja de variables:  2 ,  11  tiene una correlación:  0.05054483
La pareja de variables:  2 ,  12  tiene una correlación:  0.08682993
La pareja de variables:  2 ,  13  tiene una correlación:  0.3910464
La pareja de variables:  3 ,  4  tiene una correlación:  -0.04319613
La pareja de variables:  3 ,  5  tiene una correlación:  0.09046515
La pareja de variables:  3 ,  6  tiene una correlación:  -0.09853685
La pareja de variables:  3 ,  7  tiene una correlación:  0.07432523
La pareja de variables:  3 ,  8  tiene una correlación:  -0.317682
La pareja de variables:  3 ,  9  tiene una correlación:  0.3531598
La pareja de variables:  3 ,  10  tiene una correlación:  0.09838843
La pareja de variables:  3 ,  11  tiene una correlación:  0.1368997
La pareja de variables:  3 ,  12  tiene una correlación:  0.2258895
La pareja de variables:  3 ,  13  tiene una correlación:  0.2626587
La pareja de variables:  4 ,  5  tiene una correlación:  0.1730192
La pareja de variables:  4 ,  6  tiene una correlación:  0.155681
La pareja de variables:  4 ,  7  tiene una correlación:  0.1161575
La pareja de variables:  4 ,  8  tiene una correlación:  -0.03913566
La pareja de variables:  4 ,  9  tiene una correlación:  0.08279264
La pareja de variables:  4 ,  10  tiene una correlación:  0.1780036
La pareja de variables:  4 ,  11  tiene una correlación:  0.142472
La pareja de variables:  4 ,  12  tiene una correlación:  0.08569741
La pareja de variables:  4 ,  13  tiene una correlación:  0.1320451
```

Figura 68: Correlación entre las variables sin contar la de clase.

```
La pareja de variables:  5 ,  6  tiene una correlación:  0.02518594
La pareja de variables:  5 ,  7  tiene una correlación:  0.1676516
La pareja de variables:  5 ,  8  tiene una correlación:  -0.01873919
La pareja de variables:  5 ,  9  tiene una correlación:  0.07824253
La pareja de variables:  5 ,  10  tiene una correlación:  -0.0008314035
La pareja de variables:  5 ,  11  tiene una correlación:  -0.005755285
La pareja de variables:  5 ,  12  tiene una correlación:  0.1265415
La pareja de variables:  5 ,  13  tiene una correlación:  0.02883608
La pareja de variables:  6 ,  7  tiene una correlación:  0.05349879
La pareja de variables:  6 ,  8  tiene una correlación:  0.02249417
La pareja de variables:  6 ,  9  tiene una correlación:  -0.004107162
La pareja de variables:  6 ,  10  tiene una correlación:  -0.05980043
La pareja de variables:  6 ,  11  tiene una correlación:  0.04407599
La pareja de variables:  6 ,  12  tiene una correlación:  0.1237744
La pareja de variables:  6 ,  13  tiene una correlación:  0.04923748
La pareja de variables:  7 ,  8  tiene una correlación:  -0.07462755
La pareja de variables:  7 ,  9  tiene una correlación:  0.09509836
La pareja de variables:  7 ,  10  tiene una correlación:  0.06723504
La pareja de variables:  7 ,  11  tiene una correlación:  0.1606143
La pareja de variables:  7 ,  12  tiene una correlación:  0.1143682
La pareja de variables:  7 ,  13  tiene una correlación:  0.007337215
La pareja de variables:  8 ,  9  tiene una correlación:  -0.3807186
La pareja de variables:  8 ,  10  tiene una correlación:  -0.2790604
La pareja de variables:  8 ,  11  tiene una correlación:  -0.3868469
La pareja de variables:  8 ,  12  tiene una correlación:  -0.2653328
La pareja de variables:  8 ,  13  tiene una correlación:  -0.2533969
La pareja de variables:  9 ,  10  tiene una correlación:  0.252431
La pareja de variables:  9 ,  11  tiene una correlación:  0.2559084
La pareja de variables:  9 ,  12  tiene una correlación:  0.1533474
La pareja de variables:  9 ,  13  tiene una correlación:  0.3214491
La pareja de variables:  10 ,  11  tiene una correlación:  0.5261103
La pareja de variables:  10 ,  12  tiene una correlación:  0.1628854
La pareja de variables:  10 ,  13  tiene una correlación:  0.2614343
La pareja de variables:  11 ,  12  tiene una correlación:  0.1094977
La pareja de variables:  11 ,  13  tiene una correlación:  0.2836777
La pareja de variables:  12 ,  13  tiene una correlación:  0.2556481
```

Figura 69: Correlación entre las variables sin contar la de clase.

Como podemos observar no hay ninguna variable altamente correlada con otra, por lo que no podemos simplificar el conjunto de datos.

Este hecho puede venir de que la mayoría de variables son de tipo entero con muy pocos valores a tomar.

Veamos ahora la correlación entre las variables con la variable de clase.

```
La correlación de la variable  1  con la salida es de:  0.2123222
La correlación de la variable  2  con la salida es de:  0.2977208
La correlación de la variable  3  con la salida es de:  0.4174362
La correlación de la variable  4  con la salida es de:  0.1553827
La correlación de la variable  5  con la salida es de:  0.1180205
La correlación de la variable  6  con la salida es de:  -0.01631883
La correlación de la variable  7  con la salida es de:  0.1820908
La correlación de la variable  8  con la salida es de:  -0.418514
La correlación de la variable  9  con la salida es de:  0.4193027
La correlación de la variable  10  con la salida es de:  0.3393059
La correlación de la variable  11  con la salida es de:  0.337616
La correlación de la variable  12  con la salida es de:  0.4553365
La correlación de la variable  13  con la salida es de:  0.5250203
```

Figura 70: Correlación entre las variables con la de clase.

Podemos ver que no hay variables con un alto grado de correlación con la salida. Tendremos que comprobar con el ajuste de los modelos de clasificación el desempeño que obtenemos.

### 1.2.3. Valores perdidos

Vamos a comprobar si nuestro conjunto de clasificación tiene algún valor perdido o estamos ante un conjunto limpio de missing values como en el caso de regresión.

```
> which(is.na(dataset_clasificacion))
integer(0)
```

Figura 71: Valores perdidos en el conjunto de clasificación.

Como podemos ver tenemos un conjunto sin valores perdidos, por lo que no tenemos que hacer mayores disquisiciones en este terreno.

### 1.2.4. Outliers

Vamos a observar primero un pairplot de todas las variables.

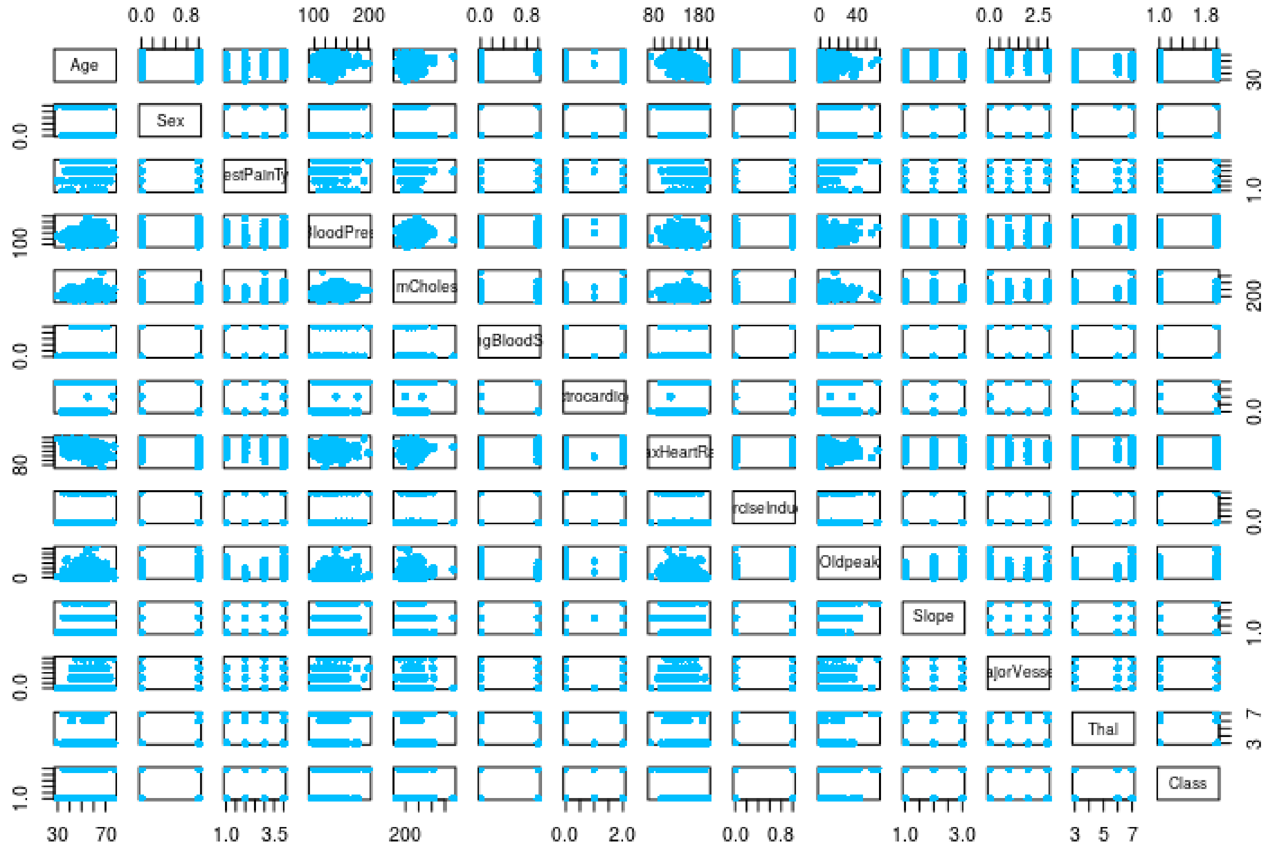


Figura 72: Pairplot de todas las variables.

En este pairplot no podemos ver demasiada información pero sí podemos intuir que hay dos tipos de variables: las que producen nubes de puntos y las que producen líneas separadas.

Vamos a verlas por separado.

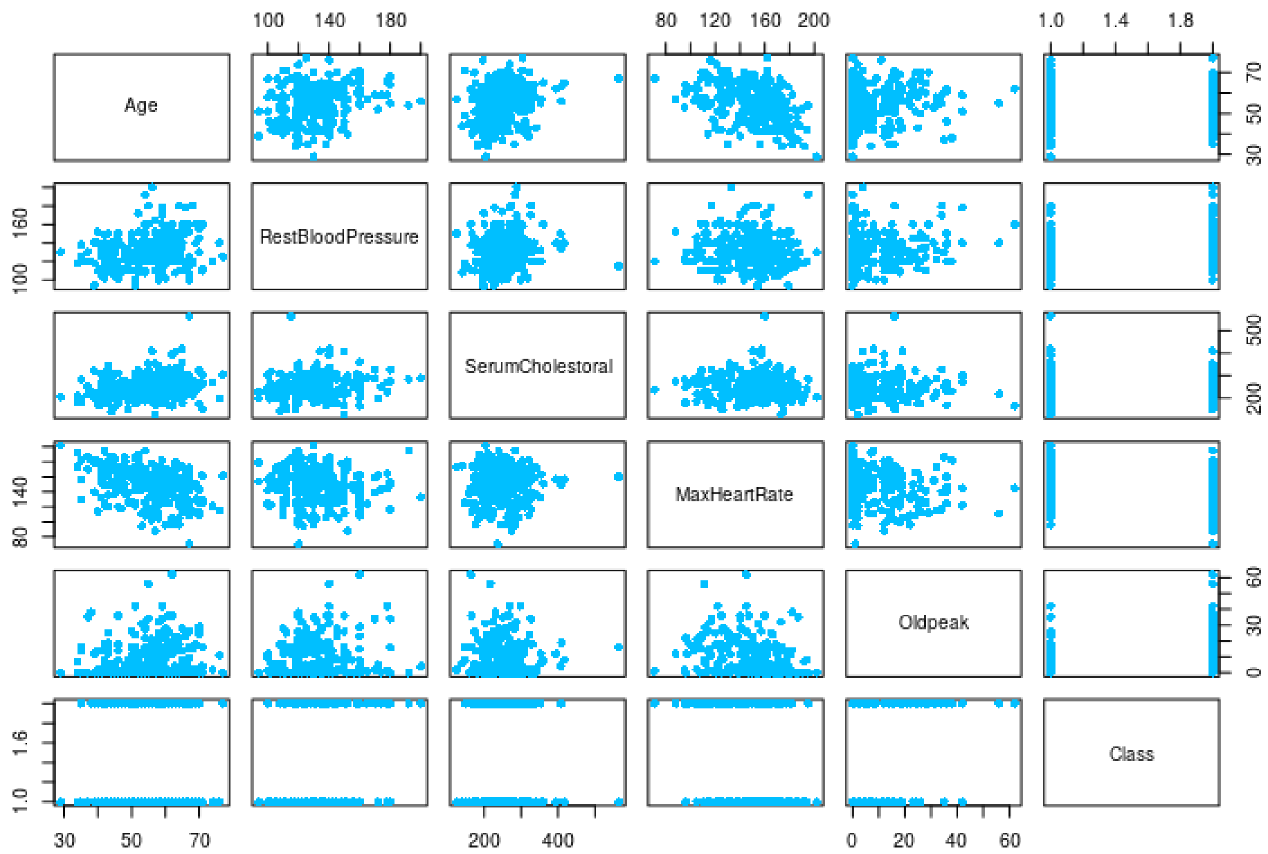


Figura 73: Pairplot de las variables que producen nubes de puntos.

Aquí podemos observar las variables. Podemos ver que hay algunos puntos anómalos pero no tienen por qué representar un problema en un principio. Otra cosa que podemos ver es que estas variables no forman dos clúster separados, cosa que puede ser conflictiva al ajustar los modelos.

Vamos a ver el pairplot del resto de variables.

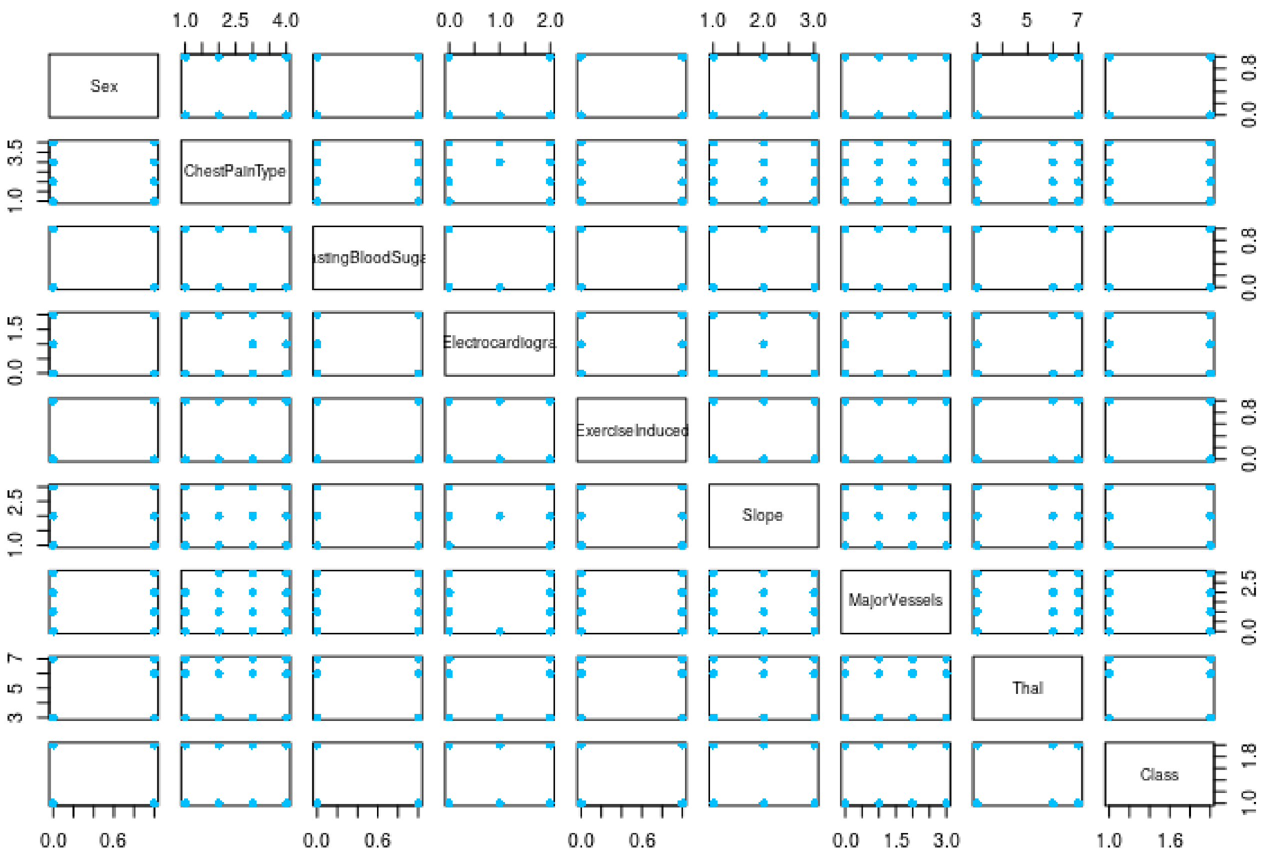


Figura 74: Pairplot de las variables que forman líneas

De estos gráficos no podemos sacar demasiada información.

Vamos a hacer ahora un boxplot de todas las variables.

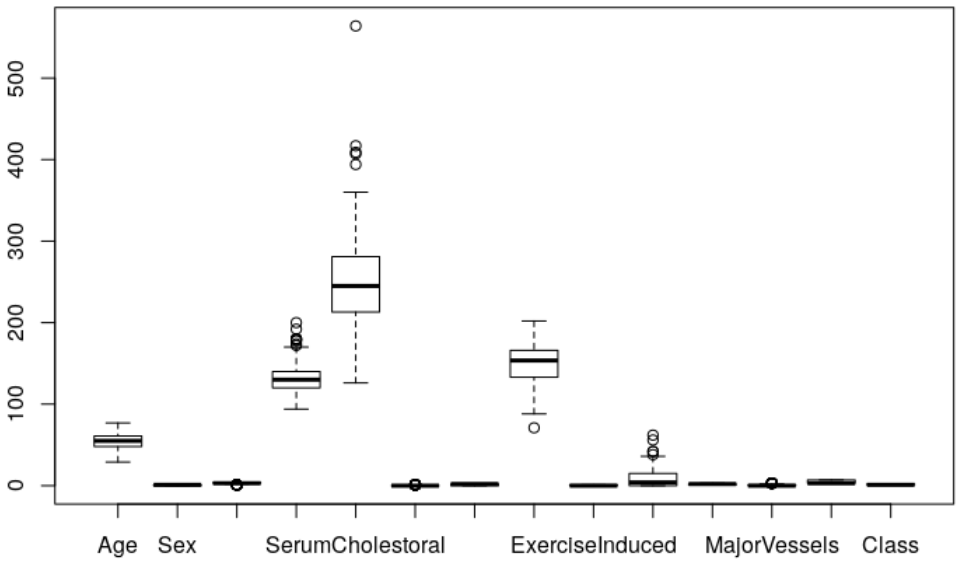


Figura 75: Boxplot de todas las variables.

Como podemos ver tenemos algunas anomalías o valores fuera de rango en las variables 4, 5, 6, 8 y 10.

Vamos a quitarlas para poder ver el resto.

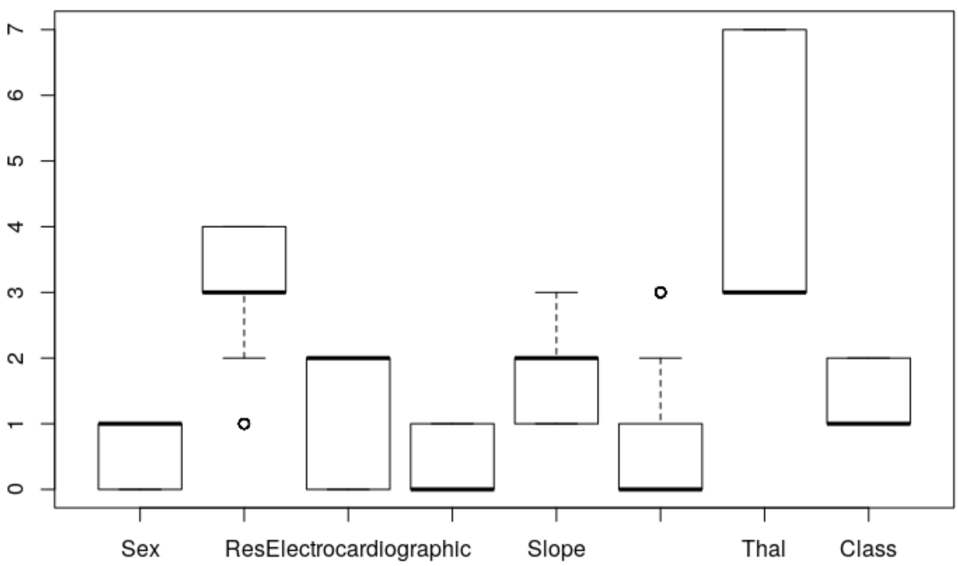


Figura 76: Boxplot eliminando variables.

Como podemos ver tenemos algunos valores que están fuera de rango y tendremos que estudiar si esto afecta a la clasificación.

Veamos también los boxplot dividiendo los valores por clases.

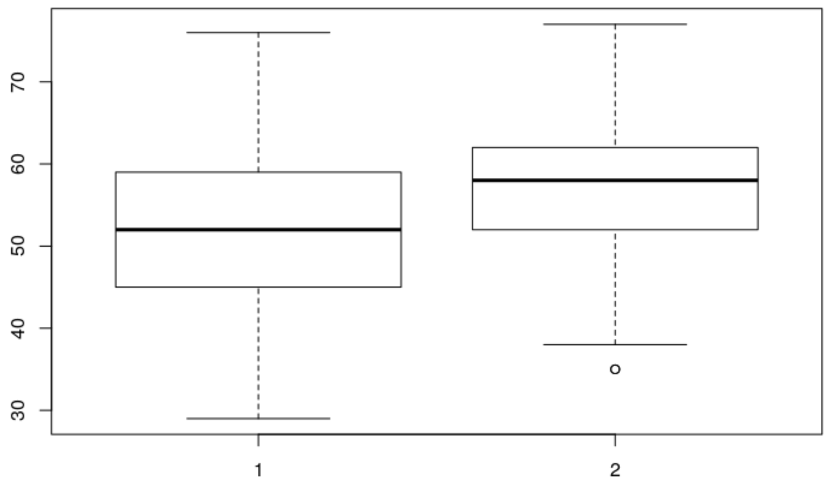


Figura 77: Boxplot de la variable 1.

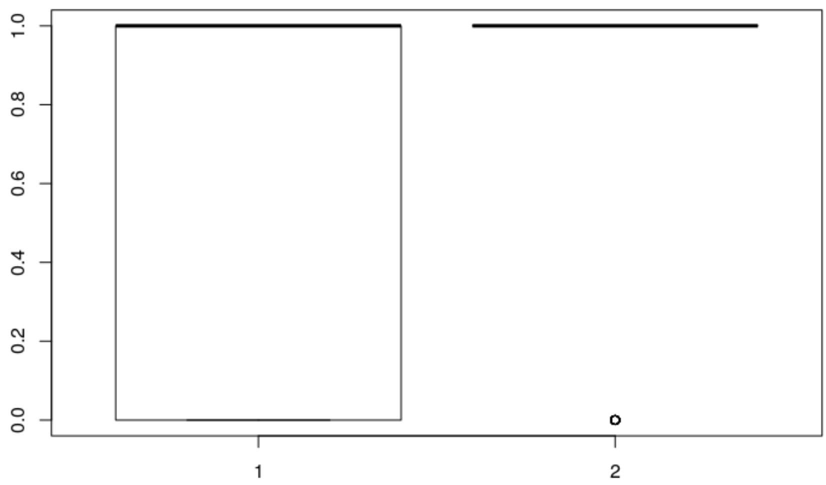


Figura 78: Boxplot de la variable 2.

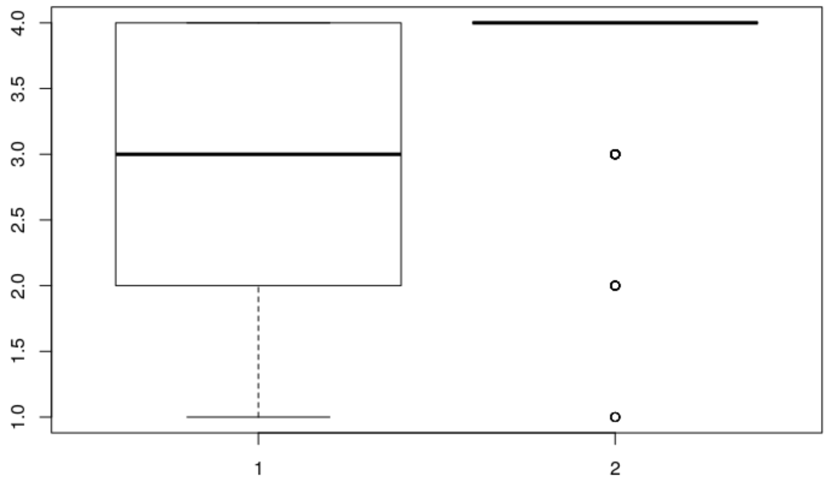


Figura 79: Boxplot de la variable 3.

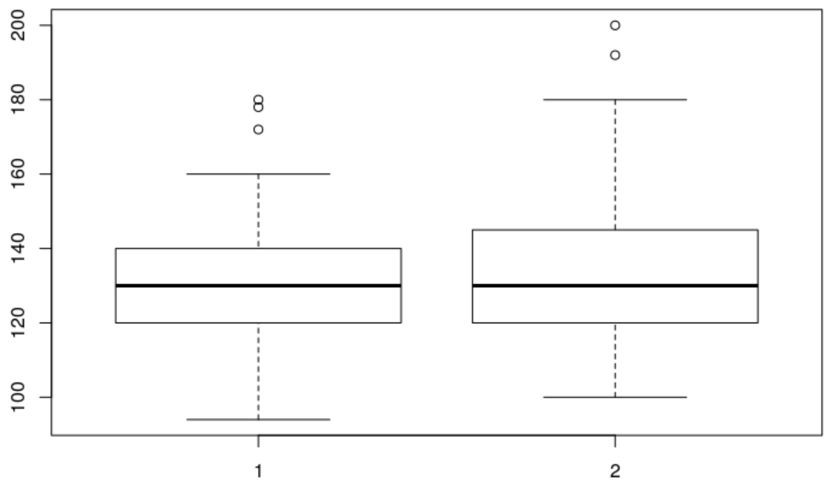


Figura 80: Boxplot de la variable 4.

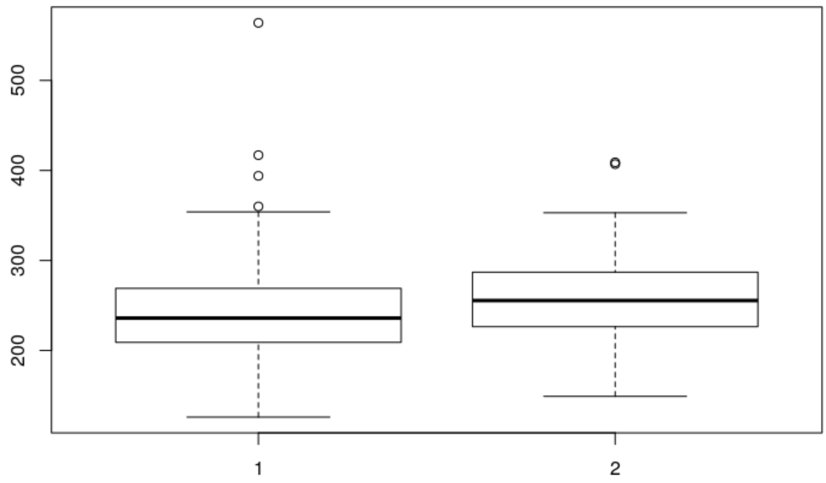


Figura 81: Boxplot de la variable 5.

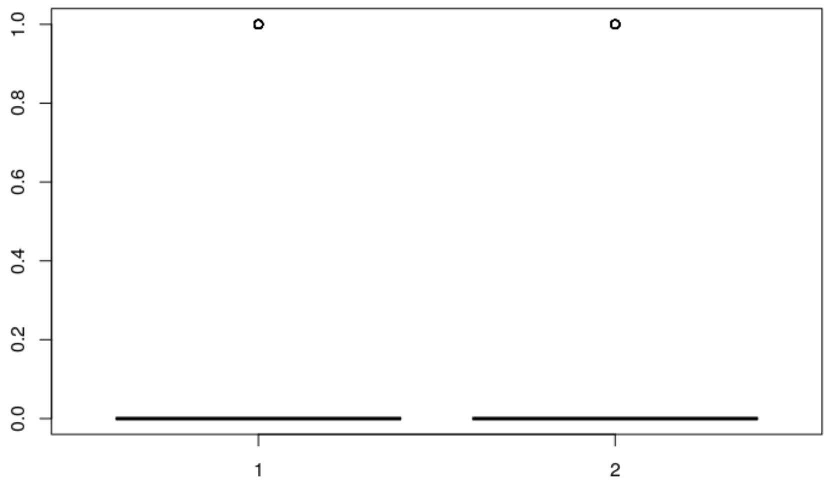


Figura 82: Boxplot de la variable 6.

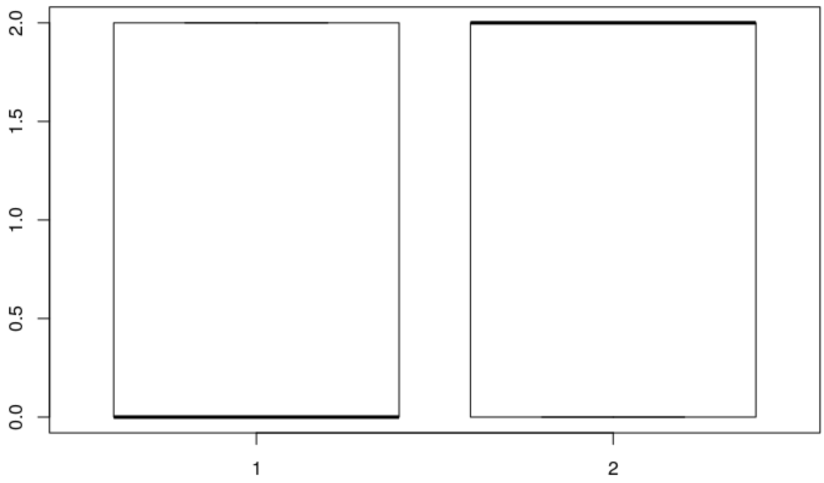


Figura 83: Boxplot de la variable 7.

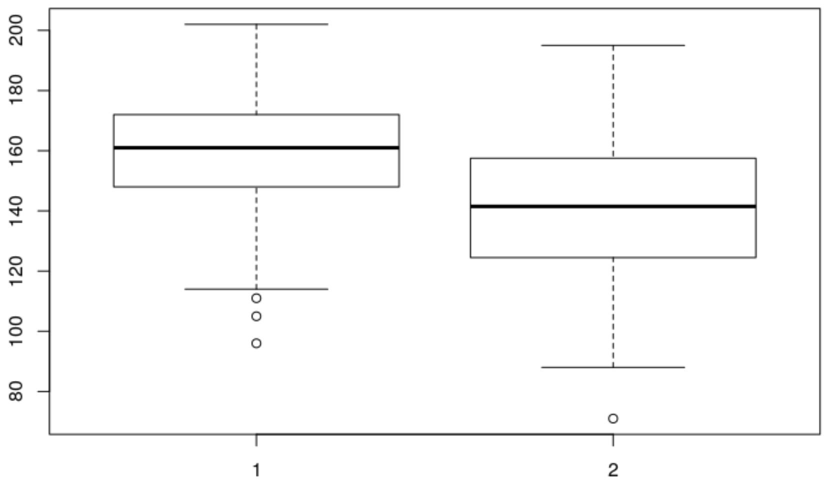


Figura 84: Boxplot de la variable 8.

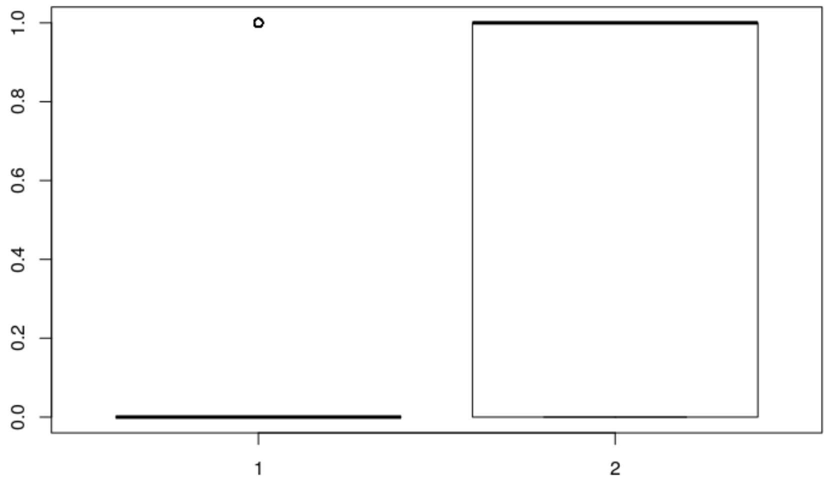


Figura 85: Boxplot de la variable 9.

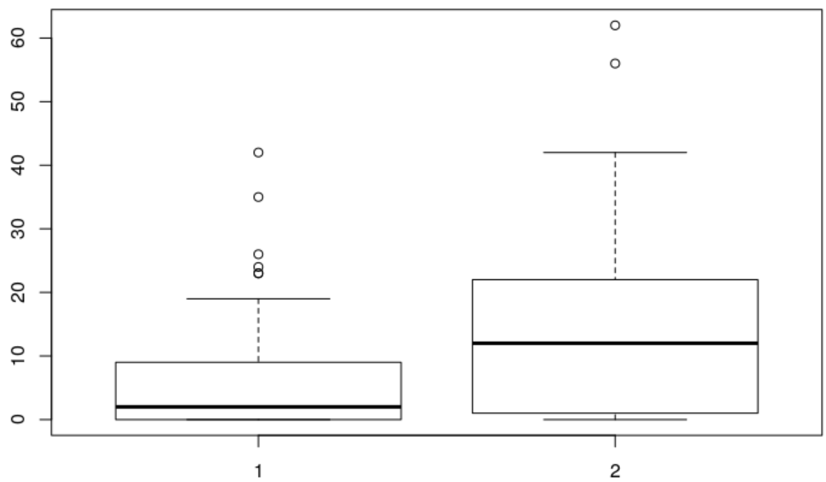


Figura 86: Boxplot de la variable 10.

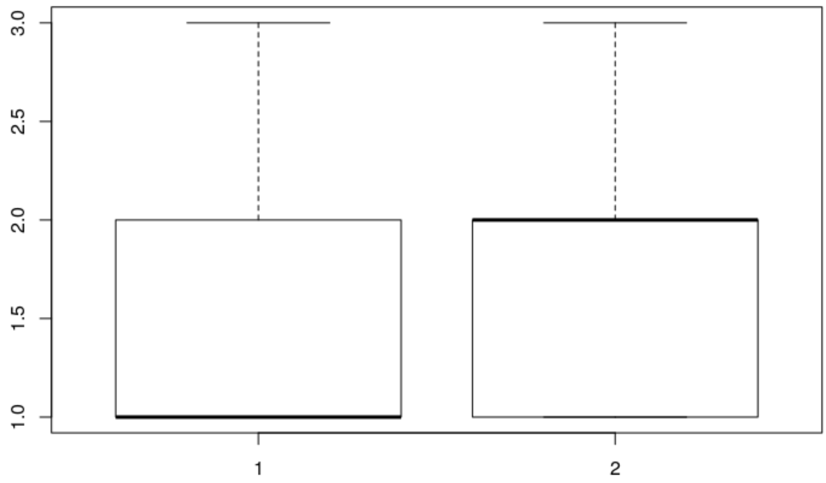


Figura 87: Boxplot de la variable 11.

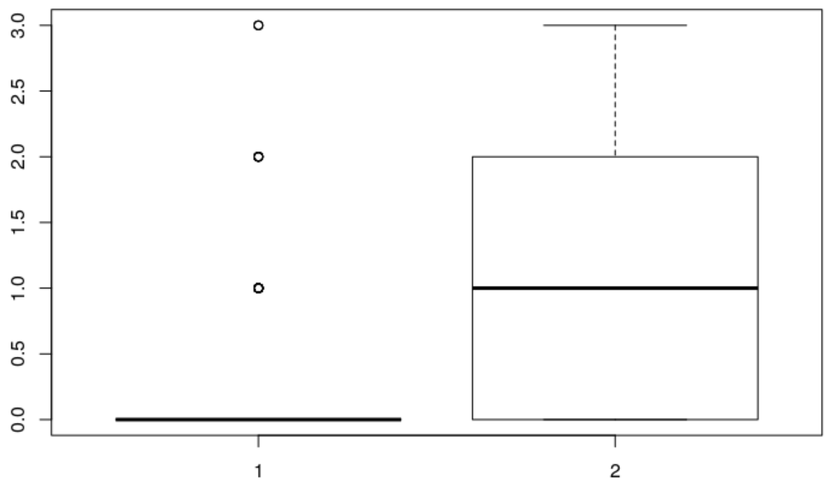


Figura 88: Boxplot de la variable 12.

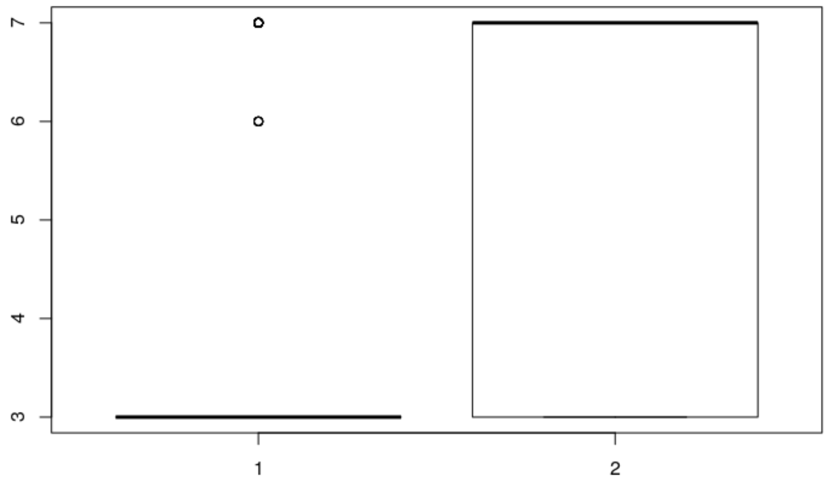


Figura 89: Boxplot de la variable 13.

En este problema no es tan sencillo decir que un valor es anómalo frente al resto. Al ser la mayoría de variables de tipo entero y con pocos valores simplemente puede que se esté dando el caso de que tengamos una concentración de valores mayor de uno de los posibles valores frente al resto y esto haga que los demás se consideren anómalos en el boxplot por cómo resulta el cálculo del rango intercuartil.

### 1.2.5. Distribución de las variables

Vamos a estudiar la distribución de las variables con un histograma de todas las variables.

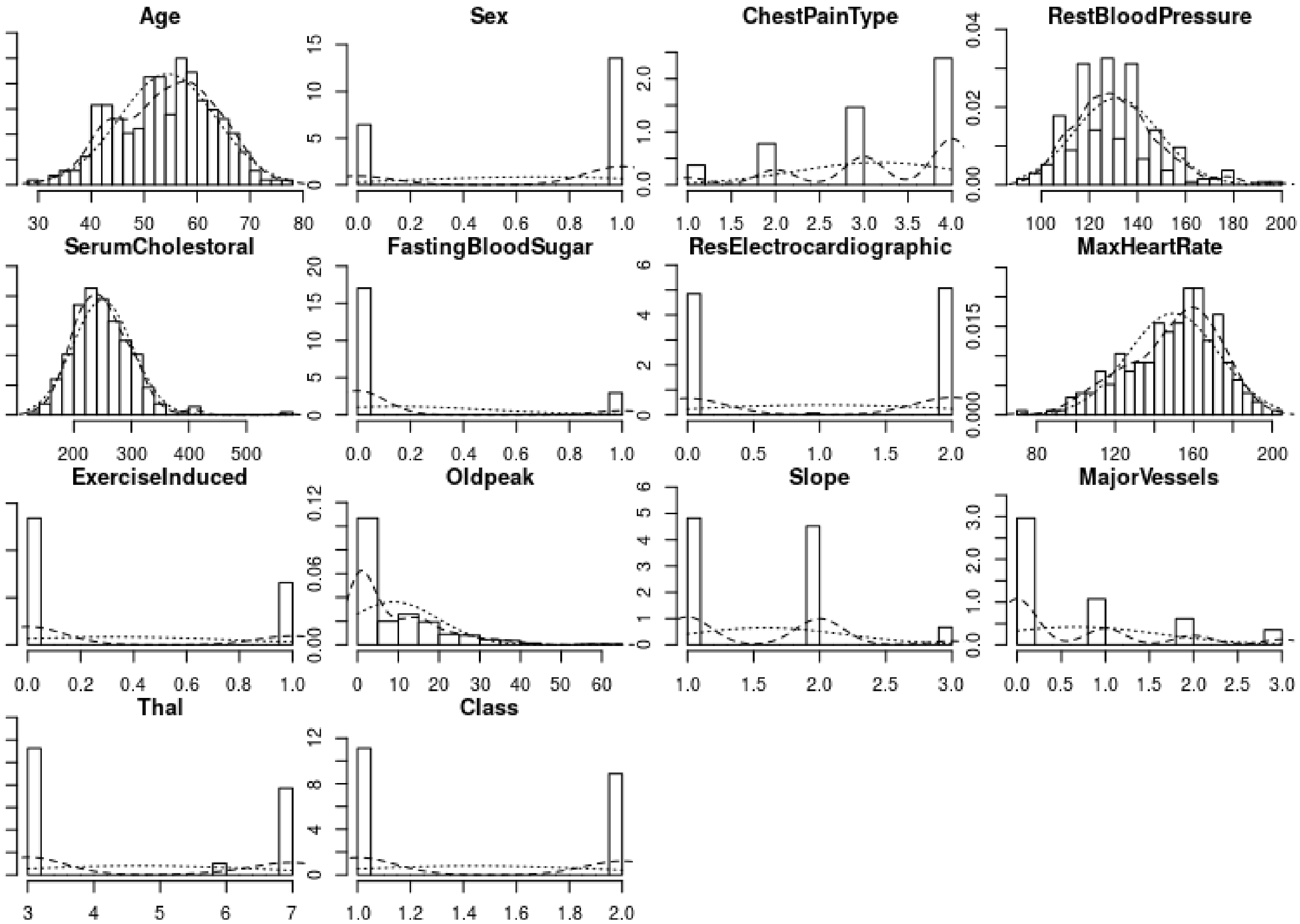


Figura 90: Histograma de todas las variables.

En primer lugar cabe decir que los histogramas de las variables que toman muy pocos valores carecen de sentido pues van a salir degenerados. Este es el caso de las variables Sex, ChestPainType, FastingBloodSugar, etc.

En un estudio visual podemos ver que hay algunas variables que puede que tengan una distribución normal. Vamos a ver si podemos decir algo acerca de esto con un test de normalidad.

```
Test de normalidad para la variable  1
P-valor:  4.79724e-46
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 91: Test de normalidad para la variable 1.

```
Test de normalidad para la variable  2
P-valor:  1.084361e-41
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 92: Test de normalidad para la variable 2.

```
Test de normalidad para la variable  3
P-valor:  1.31613e-47
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 93: Test de normalidad para la variable 3.

```
Test de normalidad para la variable  4
P-valor:  4.289576e-46
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 94: Test de normalidad para la variable 4.

```
Test de normalidad para la variable  5
P-valor:  4.943515e-46
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 95: Test de normalidad para la variable 5.

```
Test de normalidad para la variable  6
P-valor:  2.669615e-10
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 96: Test de normalidad para la variable 6.

```
Test de normalidad para la variable  7
P-valor:  1.172643e-31
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 97: Test de normalidad para la variable 7.

```
Test de normalidad para la variable  8
P-valor:  4.913854e-46
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 98: Test de normalidad para la variable 8.

```
Test de normalidad para la variable  9
P-valor:  4.037819e-21
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 99: Test de normalidad para la variable 9.

```
Test de normalidad para la variable  10
P-valor:  3.894157e-32
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 100: Test de normalidad para la variable 10.

```
Test de normalidad para la variable  11
P-valor:  2.13e-48
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 101: Test de normalidad para la variable 11.

```
Test de normalidad para la variable  12
P-valor:  1.285625e-20
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 102: Test de normalidad para la variable 12.

```
Test de normalidad para la variable  13
P-valor:  8.71151e-49
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 103: Test de normalidad para la variable 13.

```
Test de normalidad para la variable  14
P-valor:  4.424602e-49
Como es menor a 0.05 se rechaza la hipótesis nula, no sigue una normal.
```

Figura 104: Test de normalidad para la variable 14.

Como podemos comprobar, en todos los casos tenemos p-valores más pequeños que 0,05 por lo que en todos los casos rechazamos la hipótesis nula y por tanto descartamos que ninguna de las variables siga una distribución normal.

# 2. Regresión: Treasury

En esta sección vamos a abordar el problema de regresión dado por el conjunto de datos Treasury.

Ya hemos analizado previamente las características del conjunto de datos y hemos hecho hipótesis de variables que podríamos quitar. Vamos a ver en esta sección si dichas hipótesis estaban fundamentadas o no.

## 2.1. Regresión lineal simple

En esta sección se va a resolver el primero de los apartados del problema de regresión. Para ello lo que se ha realizado es generar todos los modelos de regresión posibles con una sola variable para el problema dado.

Como tenemos 16 variables en total en el conjunto de datos tendremos por tanto 15 modelos al final. De entre estos se nos ha pedido que escojamos los 5 mejores. Vamos a ver los resultados que los modelos nos otorgan y vamos a escoger en consecuencia los 5 mejores modelos.

```
Call:
lm(formula = `1MonthCDRate` ~ `1Y-CMaturityRate`)

Residuals:
    Min      1Q  Median      3Q     Max
-4.1512 -1.9110 -0.8291  1.2034 13.2902

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        -2.718650   0.633652   -4.29 1.95e-05 ***
`1Y-CMaturityRate`  0.105190   0.006438   16.34  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.016 on 1047 degrees of freedom
Multiple R-squared:  0.2032,    Adjusted R-squared:  0.2024
F-statistic:   267 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 105: Regresor con la variable 1Y-CMaturityRate.

```
Call:
lm(formula = `1MonthCDRate` ~ `30Y-CMortgageRate`)

Residuals:
    Min      1Q  Median      3Q     Max
-2.1273 -0.3047 -0.0005  0.2845  5.0385

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         -0.494992   0.058094  -8.521   <2e-16 ***
`30Y-CMortgageRate`  1.062699   0.007121 149.227   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.716 on 1047 degrees of freedom
Multiple R-squared:  0.9551,    Adjusted R-squared:  0.9551
F-statistic: 2.227e+04 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 106: Regresor con la variable 30Y-CMortgageRate.

```
Call:
lm(formula = `1MonthCDRate` ~ `3M-Rate-AuctionAverage`)

Residuals:
    Min      1Q  Median      3Q     Max
-4.0282 -1.1657 -0.1724  0.9523  8.6332

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)               -3.0062     0.1762  -17.06   <2e-16 ***
`3M-Rate-AuctionAverage`   1.0122     0.0163   62.11   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.561 on 1047 degrees of freedom
Multiple R-squared:  0.7865,    Adjusted R-squared:  0.7863
F-statistic:  3857 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 107: Regresor con la variable 3M-Rate-AuctionAverage.

```
Call:
lm(formula = `1MonthCDRate` ~ `3M-Rate-SecondaryMarket`)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.6394 -0.2313 -0.0473  0.1691  4.4752 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)               -0.23439    0.03655  -6.412 2.17e-10 ***
`3M-Rate-SecondaryMarket`  1.13211    0.00490 231.057  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4686 on 1047 degrees of freedom
Multiple R-squared:  0.9808,	Adjusted R-squared:  0.9807 
F-statistic: 5.339e+04 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 108: Regresor con la variable 3M-Rate-SecondaryMarket.

```
Call:
lm(formula = `1MonthCDRate` ~ `3Y-CMaturityRate`)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.09409 -0.20977 -0.04589  0.16718  2.83967 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        -0.252144   0.033839  -7.451 1.93e-13 ***
`3Y-CMaturityRate`  1.138336   0.004551 250.137  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4335 on 1047 degrees of freedom
Multiple R-squared:  0.9835,	Adjusted R-squared:  0.9835 
F-statistic: 6.257e+04 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 109: Regresor con la variable 3Y-CMaturityRate.

```
Call:
lm(formula = `1MonthCDRate` ~ `5Y-CMaturityRate`)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.7461 -0.7861 -0.0870  0.5551  6.5906 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)        -1.39320    0.10820  -12.88   <2e-16 ***
`5Y-CMaturityRate`  1.09828    0.01256   87.44   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.173 on 1047 degrees of freedom
Multiple R-squared:  0.8796,	Adjusted R-squared:  0.8794 
F-statistic:  7646 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 110: Regresor con la variable 5Y-CMaturityRate.

```
Call:
lm(formula = `1MonthCDRate` ~ bankCredit)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.9522 -1.0144 -0.0736  0.7275  7.3114 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.77126    0.13728   -12.9   <2e-16 ***
bankCredit   1.11175    0.01559    71.3   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.396 on 1047 degrees of freedom
Multiple R-squared:  0.8292,	Adjusted R-squared:  0.8291 
F-statistic:  5084 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 111: Regresor con la variable bankCredit.

```
Call:
lm(formula = `1MonthCDRate` ~ currency)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.7892 -1.7480  0.0812  1.3679  9.8951 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.376e+01  2.064e-01   66.65   <2e-16 ***
currency    -2.363e-03  7.304e-05  -32.35   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.389 on 1047 degrees of freedom
Multiple R-squared:  0.4999,	Adjusted R-squared:  0.4994 
F-statistic:  1046 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 112: Regresor con la variable currency.

```
Call:
lm(formula = `1MonthCDRate` ~ demandDeposits)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.8930 -1.7384  0.0472  1.2405 10.4037 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    12.6578912  0.1882159   67.25   <2e-16 ***
demandDeposits -0.0199961  0.0006693  -29.88   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.482 on 1047 degrees of freedom
Multiple R-squared:  0.4602,	Adjusted R-squared:  0.4597 
F-statistic: 892.6 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 113: Regresor con la variable demandDeposits.

```
Call:
lm(formula = `1MonthCDRate` ~ federalFunds)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.0218 -1.6325 -0.3797  1.3448 11.4452 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  19.739213   0.390006   50.61   <2e-16 ***
federalFunds -0.039652   0.001243  -31.91   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.406 on 1047 degrees of freedom
Multiple R-squared:  0.493,	Adjusted R-squared:  0.4926 
F-statistic:  1018 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 114: Regresor con la variable federalFunds.

```
Call:
lm(formula = `1MonthCDRate` ~ moneyStock)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.35798 -0.11996 -0.01642  0.10068  2.01779 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.355430   0.025401   13.99   <2e-16 ***
moneyStock  0.949271   0.003047  311.56   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.349 on 1047 degrees of freedom
Multiple R-squared:  0.9893,	Adjusted R-squared:  0.9893 
F-statistic: 9.707e+04 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 115: Regresor con la variable moneyStock.

```
Call:
lm(formula = `1MonthCDRate` ~ checkableDeposits)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.6772 -1.5990  0.3663  0.9614  8.9735 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)       16.1087917  0.2025477   79.53   <2e-16 ***
checkableDeposits -0.0105576  0.0002373  -44.48   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.988 on 1047 degrees of freedom
Multiple R-squared:  0.654,	Adjusted R-squared:  0.6537 
F-statistic:  1979 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 116: Regresor con la variable checkableDeposits.

```
Call:
lm(formula = `1MonthCDRate` ~ loansLeases)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.0805 -1.4078 -0.3461  1.0526  8.6286 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 17.3090553  0.2129478   81.28   <2e-16 ***
loansLeases -0.0178049  0.0003725  -47.80   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.894 on 1047 degrees of freedom
Multiple R-squared:  0.6857,	Adjusted R-squared:  0.6854 
F-statistic:  2285 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 117: Regresor con la variable loansLeases.

```
Call:
lm(formula = `1MonthCDRate` ~ savingsDeposits)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.0089 -1.6994  0.0811  1.4772  9.8744 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     13.8942658  0.2177022   63.82   <2e-16 ***
savingsDeposits -0.0032526  0.0001043  -31.19   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.433 on 1047 degrees of freedom
Multiple R-squared:  0.4816,	Adjusted R-squared:  0.4811 
F-statistic: 972.6 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 118: Regresor con la variable savingsDeposits.

```
Call:
lm(formula = `1MonthCDRate` ~ tradeCurrencies)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.7712 -1.3262 -0.3366  1.0867  9.2458 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     14.3906585  0.1749509   82.25   <2e-16 ***
tradeCurrencies -0.0071949  0.0001707  -42.14   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.058 on 1047 degrees of freedom
Multiple R-squared:  0.6291,	Adjusted R-squared:  0.6287 
F-statistic:  1776 on 1 and 1047 DF,  p-value: < 2.2e-16
```

Figura 119: Regresor con la variable tradeCurrencies.

Para poder discernir los modelos que mejor desempeño tienen vamos a analizar los valores de $R^2$ ajustado.

Según los valores que hemos obtenido los 5 mejores modelos son:

| **Variable** | $R^2$ **ajustado** |
|---|---|
| moneyStock | 0.9893 |
| 3Y-CMaturityRate | 0.9835 |
| 3M-Rate-SecondaryMarket | 0.9807 |
| 30Y-CMortgageRate | 0.9551 |
| 5Y-CMaturityRate | 0.8794 |

Aquí podemos intuir que estas variables van a ser extremadamente importantes en el problema de regresión pues por ellas solas obtienen ya un buen resultado.

Vamos a ver las gráficas que obtienen y su línea predicha.

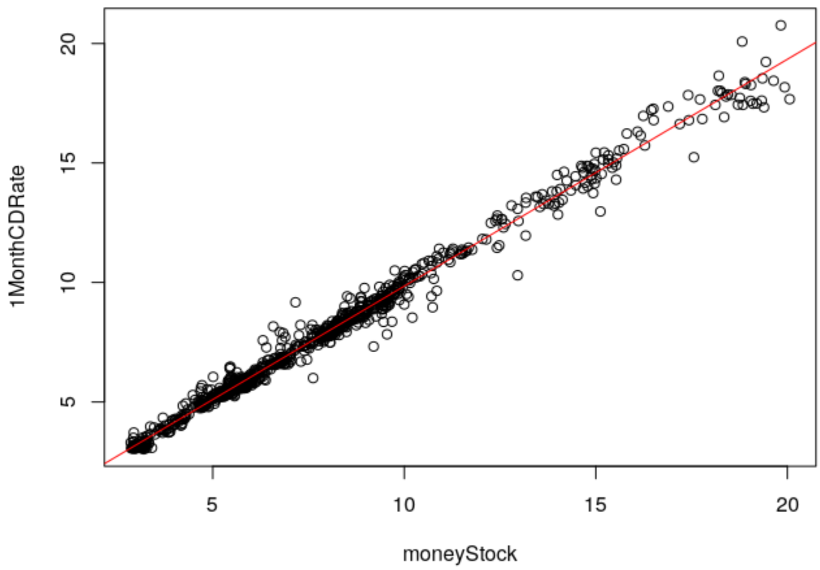


Figura 120: Conjunto de datos y la línea obtenida por el modelo con la variable moneyStock.

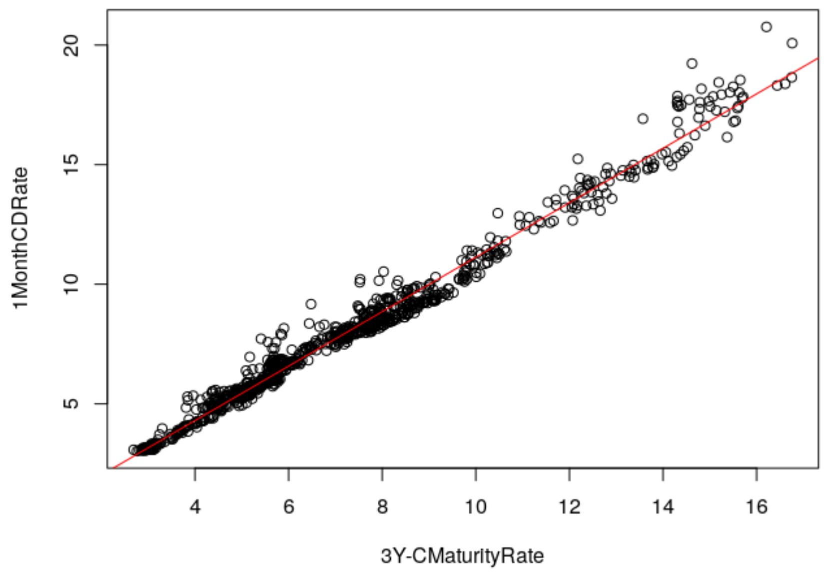


Figura 121: Conjunto de datos y la línea obtenida por el modelo con la variable 3Y-CMaturityRate.

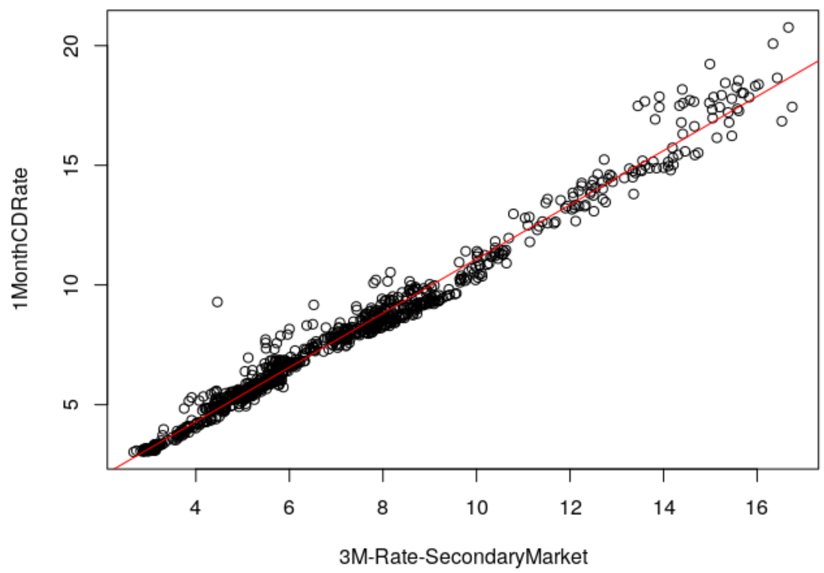


Figura 122: Conjunto de datos y la línea obtenida por el modelo con la variable 3M-Rate-SecondaryMarket.

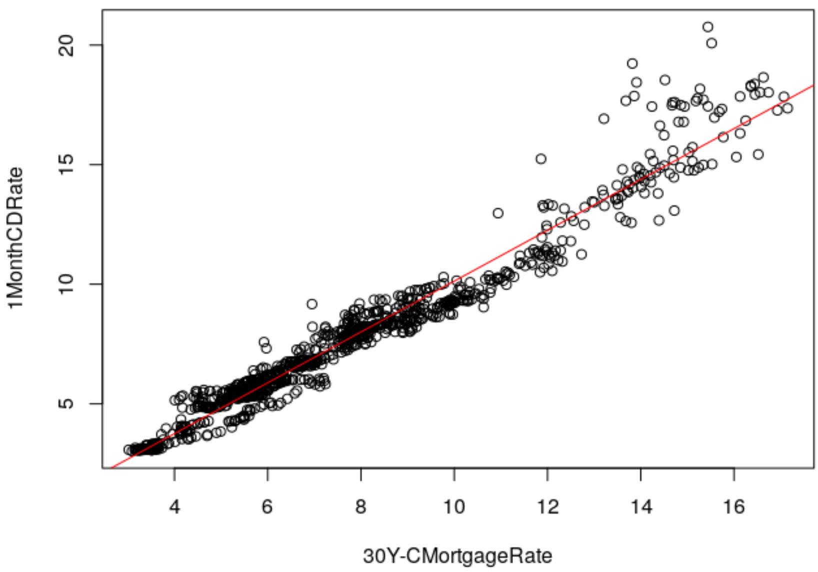


Figura 123: Conjunto de datos y la línea obtenida por el modelo con la variable 30Y-CMortgageRate.

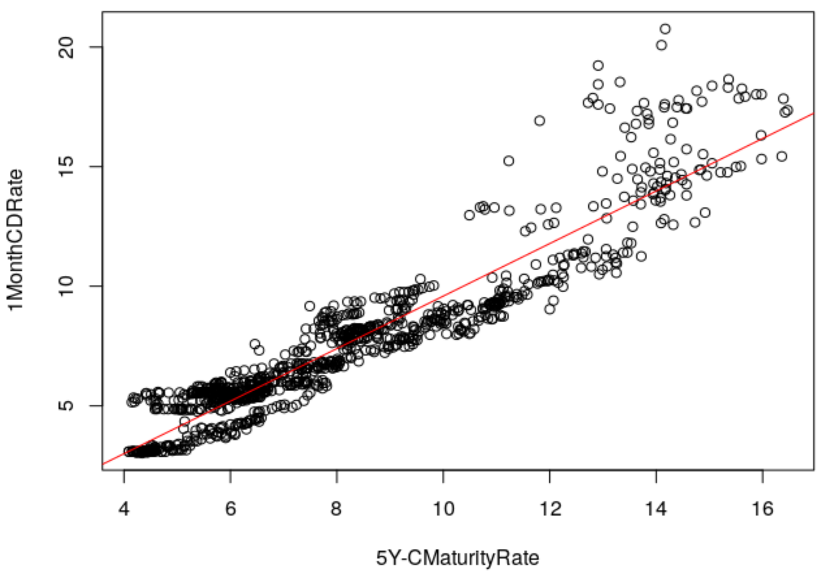


Figura 124: Conjunto de datos y la línea obtenida por el modelo con la variable 5Y-CMaturityRate.

Como podemos observar el ajuste es muy bueno en las 4 primeras variables y en la segunda algo peor sobre todo cuanto mayores son los valores de la variable de salida.

El error cometido por los modelos coincide en la ordenación del $R^2$ ajustado.

Los modelos de una sola variable ya nos proporcionan unos resultados realmente buenos teniendo el mejor de ellos un $R^2$ ajustado de 0,9893, lo que nos va a dar un margen de mejora realmente escaso. Aún así vamos a continuar con modelos más complejos para ver si podemos mejorar este resultado.

## 2.2. Regresión lineal múltiple

En este apartado vamos a estudiar los modelos de regresión lineal múltiple. Para empezar esta sección vamos a ir de más a menos empezando con un modelo que emplee todas las variable y luego iremos eliminando las variables de menor calidad hasta obtener un modelo simple con un buen rendimiento.

```
Call:
lm(formula = `1MonthCDRate` ~ ., data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.15400 -0.11350 -0.02425  0.06447  1.98636 

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)                0.3888300  0.2922598   1.330  0.18367    
`1Y-CMaturityRate`        -0.0078034  0.0010164  -7.677 3.75e-14 ***
`30Y-CMortgageRate`        0.1872664  0.0625423   2.994  0.00282 ** 
`3M-Rate-AuctionAverage`  -0.0177122  0.0212776  -0.832  0.40536    
`3M-Rate-SecondaryMarket`  0.0471672  0.0384214   1.228  0.21987    
`3Y-CMaturityRate`         0.2464712  0.0478963   5.146 3.18e-07 ***
`5Y-CMaturityRate`         0.1558061  0.1131354   1.377  0.16876    
bankCredit                -0.1422313  0.0770943  -1.845  0.06534 .  
currency                   0.0017686  0.0004687   3.773  0.00017 ***
demandDeposits             0.0191436  0.0103577   1.848  0.06485 .  
federalFunds               0.0001295  0.0005897   0.220  0.82618    
moneyStock                 0.5811069  0.0158907  36.569  < 2e-16 ***
checkableDeposits         -0.0260884  0.0103044  -2.532  0.01150 *  
loansLeases                0.0254454  0.0103785   2.452  0.01438 *  
savingsDeposits           -0.0013883  0.0004364  -3.181  0.00151 ** 
tradeCurrencies            0.0005457  0.0001040   5.247 1.88e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2377 on 1033 degrees of freedom
Multiple R-squared:  0.9951,	Adjusted R-squared:  0.995 
F-statistic: 1.403e+04 on 15 and 1033 DF,  p-value: < 2.2e-16
```

Figura 125: Resultados para el modelo con todas las variables.

Como podemos ver tenemos varias variables que proporcionan malos resultados o que no aportan información como la variable federalFunds.

En cuanto al $R^2$ ajustado podemos ver que es 0,995, lo que nos está indicando que el modelo es de mucha calidad en cuanto al ajuste, pero no podemos olvidar que el mejor resultado de los modelos de una única variable fue 0,9893.

Vamos a eliminar las variables de mala calidad para el ajuste como por ejemplo la variable federalFunds.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds, data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.15241 -0.11335 -0.02378  0.06315  1.98494 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)               0.4302850  0.2230517   1.929 0.053994 .  
`1Y-CMaturityRate`       -0.0077566  0.0009934  -7.808 1.42e-14 ***
`30Y-CMortgageRate`       0.1888123  0.0621165   3.040 0.002428 ** 
`3M-Rate-AuctionAverage` -0.0192017  0.0201587  -0.953 0.341053    
`3M-Rate-SecondaryMarket` 0.0474560  0.0383813   1.236 0.216577    
`3Y-CMaturityRate`        0.2442259  0.0467713   5.222 2.14e-07 ***
`5Y-CMaturityRate`        0.1602087  0.1112946   1.440 0.150311    
bankCredit               -0.1469584  0.0739957  -1.986 0.047293 *  
currency                  0.0017634  0.0004679   3.769 0.000173 ***
demandDeposits            0.0192211  0.0103469   1.858 0.063499 .  
moneyStock                0.5817900  0.0155762  37.351  < 2e-16 ***
checkableDeposits        -0.0260428  0.0102975  -2.529 0.011585 *  
loansLeases               0.0254230  0.0103732   2.451 0.014417 *  
savingsDeposits          -0.0013965  0.0004346  -3.213 0.001352 ** 
tradeCurrencies           0.0005401  0.0001008   5.357 1.04e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2376 on 1034 degrees of freedom
Multiple R-squared:  0.9951,    Adjusted R-squared:  0.9951 
F-statistic: 1.505e+04 on 14 and 1034 DF,  p-value: < 2.2e-16
```

Figura 126: Resultados eliminando la variable federalFunds.

Podemos ver que incluso el $R^2$ ajustado ha subido ligeramente. Esto no es porque el modelo haya mejorado su regresión eliminando una variable, sino porque el $R^2$ ajustado se calcula también en base al número de variables que participan en el modelo y al eliminar una podemos haber producido el aumento.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds - `3M-Rate-AuctionAverage`,
    data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.16771 -0.11356 -0.02274  0.06095  1.98373 

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)                3.068e-01  1.815e-01   1.690 0.091262 .  
`1Y-CMaturityRate`        -7.759e-03  9.934e-04  -7.811 1.39e-14 ***
`30Y-CMortgageRate`        1.990e-01  6.118e-02   3.253 0.001180 ** 
`3M-Rate-SecondaryMarket`  4.156e-02  3.788e-02   1.097 0.272802    
`3Y-CMaturityRate`         2.586e-01  4.428e-02   5.840 6.97e-09 ***
`5Y-CMaturityRate`         1.377e-01  1.088e-01   1.266 0.205739    
bankCredit                -1.503e-01  7.391e-02  -2.034 0.042252 *  
currency                   1.684e-03  4.604e-04   3.658 0.000267 ***
demandDeposits             1.873e-02  1.033e-02   1.812 0.070236 .  
moneyStock                 5.747e-01  1.370e-02  41.938  < 2e-16 ***
checkableDeposits         -2.542e-02  1.028e-02  -2.474 0.013521 *  
loansLeases                2.487e-02  1.036e-02   2.401 0.016523 *  
savingsDeposits           -1.305e-03  4.239e-04  -3.079 0.002128 ** 
tradeCurrencies            5.556e-04  9.951e-05   5.583 3.02e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2376 on 1035 degrees of freedom
Multiple R-squared:  0.9951,	Adjusted R-squared:  0.9951 
F-statistic: 1.621e+04 on 13 and 1035 DF,  p-value: < 2.2e-16
```

Figura 127: Resultados eliminando la variable federalFunds y 3M-Rate-AuctionAverage.

Podemos ver que el valor de $R^2$ ajustado no ha variado al eliminar la variable 3M-Rate-AuctionAverage. Vamos a continuar eliminando la variable 3M-Rate-SecondaryMarket.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds - `3M-Rate-AuctionAverage` -
    `3M-Rate-SecondaryMarket`, data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max
-1.15221 -0.11383 -0.02355  0.06132  1.98401

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          3.109e-01  1.815e-01   1.713 0.087022 .
`1Y-CMaturityRate`  -7.765e-03  9.934e-04  -7.816 1.34e-14 ***
`30Y-CMortgageRate`  2.067e-01  6.079e-02   3.401 0.000698 ***
`3Y-CMaturityRate`   2.943e-01  2.997e-02   9.821  < 2e-16 ***
`5Y-CMaturityRate`   1.287e-01  1.085e-01   1.187 0.235619
bankCredit          -1.450e-01  7.376e-02  -1.966 0.049552 *
currency             1.688e-03  4.604e-04   3.667 0.000258 ***
demandDeposits       1.870e-02  1.033e-02   1.810 0.070620 .
moneyStock           5.759e-01  1.366e-02  42.154  < 2e-16 ***
checkableDeposits   -2.541e-02  1.028e-02  -2.473 0.013571 *
loansLeases          2.485e-02  1.036e-02   2.399 0.016601 *
savingsDeposits     -1.305e-03  4.240e-04  -3.079 0.002134 **
tradeCurrencies      5.475e-04  9.925e-05   5.516 4.37e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2376 on 1036 degrees of freedom
Multiple R-squared:  0.9951,    Adjusted R-squared:  0.9951
F-statistic: 1.756e+04 on 12 and 1036 DF,  p-value: < 2.2e-16
```

Figura 128: Resultados eliminando la variable federalFunds, 3M-Rate-AuctionAverage y 3M-Rate-SecondaryMarket.

Seguimos sin bajar en la bondad del ajuste, por lo que sabemos que estamos yendo en la buena dirección. Vamos a eliminar ahora la variable 5Y-CMaturityRate.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds - `3M-Rate-AuctionAverage` -
    `3M-Rate-SecondaryMarket` - `5Y-CMaturityRate`, data = dataset_regresion)

Residuals:
    Min      1Q  Median      3Q     Max
-1.1827 -0.1114 -0.0232  0.0627  1.9784

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          2.409e-01  1.717e-01   1.403 0.160860
`1Y-CMaturityRate`  -7.480e-03  9.642e-04  -7.758 2.06e-14 ***
`30Y-CMortgageRate`  2.675e-01  3.280e-02   8.154 1.01e-15 ***
`3Y-CMaturityRate`   2.781e-01  2.669e-02  10.423  < 2e-16 ***
bankCredit          -6.094e-02  2.051e-02  -2.972 0.003031 **
currency             1.530e-03  4.408e-04   3.472 0.000539 ***
demandDeposits       1.857e-02  1.034e-02   1.797 0.072700 .
moneyStock           5.767e-01  1.365e-02  42.242  < 2e-16 ***
checkableDeposits   -2.501e-02  1.027e-02  -2.434 0.015104 *
loansLeases          2.448e-02  1.035e-02   2.364 0.018249 *
savingsDeposits     -1.133e-03  3.984e-04  -2.844 0.004547 **
tradeCurrencies      5.656e-04  9.808e-05   5.766 1.07e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2377 on 1037 degrees of freedom
Multiple R-squared:  0.9951,    Adjusted R-squared:  0.995
F-statistic: 1.915e+04 on 11 and 1037 DF,  p-value: < 2.2e-16
```

Figura 129: Resultados eliminando la variable federalFunds, 3M-Rate-AuctionAverage, 3M-Rate-SecondaryMarket y 5Y-CMaturityRate.

Como podemos ver hemos bajado ligeramente la bondad del ajuste al eliminar la última variable, pero la reducción ha sido tan baja que no es apreciable. Seguimos teniendo una variable que no está dando demasiado peso en el modelo y por tanto podríamos pensar en eliminarla. Vamos a ver qué ocurre cuando eliminamos la variable demandDeposits.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds - `3M-Rate-AuctionAverage` -
    `3M-Rate-SecondaryMarket` - `5Y-CMaturityRate` - demandDeposits,
    data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max
-1.17101 -0.11467 -0.02595  0.05937  1.98654

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          2.410e-01  1.718e-01   1.402 0.161139
`1Y-CMaturityRate`  -7.412e-03  9.645e-04  -7.685 3.52e-14 ***
`30Y-CMortgageRate`  2.724e-01  3.272e-02   8.324 2.66e-16 ***
`3Y-CMaturityRate`   2.790e-01  2.671e-02  10.447  < 2e-16 ***
bankCredit          -6.566e-02  2.036e-02  -3.225 0.001298 **
currency             1.663e-03  4.350e-04   3.824 0.000139 ***
moneyStock           5.747e-01  1.362e-02  42.186  < 2e-16 ***
checkableDeposits   -6.687e-03  1.259e-03  -5.310 1.34e-07 ***
loansLeases          6.003e-03  1.197e-03   5.017 6.17e-07 ***
savingsDeposits     -1.300e-03  3.878e-04  -3.353 0.000827 ***
tradeCurrencies      5.740e-04  9.808e-05   5.853 6.48e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2379 on 1038 degrees of freedom
Multiple R-squared:  0.9951,    Adjusted R-squared:  0.995
F-statistic: 2.101e+04 on 10 and 1038 DF,  p-value: < 2.2e-16
```

Figura 130: Resultados eliminando la variable federalFunds, 3M-Rate-AuctionAverage, 3M-Rate-SecondaryMarket, 5Y-CMaturityRate y demandDeposits.

Vemos que seguimos manteniendo el ajuste. Vamos a ver si eliminamos ahora bankCredit.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds - `3M-Rate-AuctionAverage` -
    `3M-Rate-SecondaryMarket` - `5Y-CMaturityRate` - demandDeposits -
    bankCredit, data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max
-1.16389 -0.11514 -0.02423  0.06393  1.96616

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          6.178e-02  1.634e-01   0.378 0.705358
`1Y-CMaturityRate`  -8.226e-03  9.351e-04  -8.797  < 2e-16 ***
`30Y-CMortgageRate`  1.841e-01  1.801e-02  10.224  < 2e-16 ***
`3Y-CMaturityRate`   3.122e-01  2.476e-02  12.610  < 2e-16 ***
currency             1.661e-03  4.369e-04   3.801 0.000153 ***
moneyStock           5.835e-01  1.341e-02  43.513  < 2e-16 ***
checkableDeposits   -6.674e-03  1.265e-03  -5.276 1.61e-07 ***
loansLeases          5.975e-03  1.202e-03   4.971 7.77e-07 ***
savingsDeposits     -1.239e-03  3.890e-04  -3.186 0.001486 **
tradeCurrencies      5.476e-04  9.818e-05   5.578 3.10e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.239 on 1039 degrees of freedom
Multiple R-squared:  0.995,     Adjusted R-squared:  0.995
F-statistic: 2.314e+04 on 9 and 1039 DF,  p-value: < 2.2e-16
```

Figura 131: Resultados eliminando la variable federalFunds, 3M-Rate-AuctionAverage, 3M-Rate-SecondaryMarket, 5Y-CMaturityRate, demandDeposits y bankCredit.

Podemos ver que ahora la variable savingsDeposits nos está dando un ajuste un poco peor que el resto. Como no hemos reducido el rendimiento del modelo podemos probar a eliminar esta variable para ver si empeoramos el ajuste.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds - `3M-Rate-AuctionAverage` -
    `3M-Rate-SecondaryMarket` - `5Y-CMaturityRate` - demandDeposits -
    bankCredit - savingsDeposits, data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max
-1.16939 -0.11925 -0.03015  0.06452  1.94999

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          8.313e-02  1.639e-01   0.507 0.612196
`1Y-CMaturityRate`  -8.234e-03  9.392e-04  -8.767  < 2e-16 ***
`30Y-CMortgageRate`  1.712e-01  1.762e-02   9.714  < 2e-16 ***
`3Y-CMaturityRate`   3.197e-01  2.475e-02  12.917  < 2e-16 ***
currency             2.926e-04  8.095e-05   3.615 0.000315 ***
moneyStock           5.858e-01  1.345e-02  43.551  < 2e-16 ***
checkableDeposits   -3.061e-03  5.629e-04  -5.438 6.72e-08 ***
loansLeases          2.644e-03  5.955e-04   4.441 9.92e-06 ***
tradeCurrencies      6.356e-04  9.463e-05   6.717 3.05e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.24 on 1040 degrees of freedom
Multiple R-squared:  0.995,     Adjusted R-squared:  0.9949
F-statistic: 2.58e+04 on 8 and 1040 DF,  p-value: < 2.2e-16
```

Figura 132: Resultados eliminando la variable federalFunds, 3M-Rate-AuctionAverage, 3M-Rate-SecondaryMarket, 5Y-CMaturityRate, demandDeposits, bankCredit y savingsDeposits.

Podemos ver que hemos bajado un poco en el ajuste pero hemos reducido considerablemente la complejidad del modelo eliminando muchas variables. Podemos intentar eliminar alguna variable más aunque ya estamos en p-valores muy bajos, por lo que es probable que al eliminar alguna variable más ya si reduzcamos el desempeño del modelo de forma significativa.

Si comparamos los p-valores de las variables la siguiente candidata a ser eliminada del modelo sería currency. Vamos a ver los resultados que obtenemos si la eliminamos.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds - `3M-Rate-AuctionAverage` -
    `3M-Rate-SecondaryMarket` - `5Y-CMaturityRate` - demandDeposits -
    bankCredit - savingsDeposits - currency, data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max
-1.18717 -0.12020 -0.02665  0.06467  1.99030

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          3.062e-01  1.527e-01   2.005  0.04526 *
`1Y-CMaturityRate`  -9.362e-03  8.910e-04 -10.507  < 2e-16 ***
`30Y-CMortgageRate`  1.620e-01  1.754e-02   9.237  < 2e-16 ***
`3Y-CMaturityRate`   3.254e-01  2.485e-02  13.093  < 2e-16 ***
moneyStock           5.893e-01  1.349e-02  43.674  < 2e-16 ***
checkableDeposits   -1.167e-03  2.066e-04  -5.646 2.12e-08 ***
loansLeases          6.971e-04  2.552e-04   2.732  0.00641 **
tradeCurrencies      8.379e-04  7.673e-05  10.920  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2414 on 1041 degrees of freedom
Multiple R-squared:  0.9949,    Adjusted R-squared:  0.9949
F-statistic: 2.915e+04 on 7 and 1041 DF,  p-value: < 2.2e-16
```

Figura 133: Resultados eliminando la variable federalFunds, 3M-Rate-AuctionAverage, 3M-Rate-SecondaryMarket, 5Y-CMaturityRate, demandDeposits, bankCredit, savingsDeposits y currency.

Seguimos teniendo un modelo con buenos resultados y ahora podemos ver que la variable que empeora el resultado del modelo debería ser loansLeases, por lo que podemos eliminarla y ver si obtenemos resultados mucho peores o mantenemos la línea de ajuste.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds - `3M-Rate-AuctionAverage` -
    `3M-Rate-SecondaryMarket` - `5Y-CMaturityRate` - demandDeposits -
    bankCredit - savingsDeposits - currency - loansLeases, data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max
-1.23796 -0.11955 -0.02504  0.06760  1.98934

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          4.992e-01  1.358e-01   3.675  0.00025 ***
`1Y-CMaturityRate`  -9.566e-03  8.906e-04 -10.741  < 2e-16 ***
`30Y-CMortgageRate`  1.591e-01  1.756e-02   9.062  < 2e-16 ***
`3Y-CMaturityRate`   3.266e-01  2.492e-02  13.105  < 2e-16 ***
moneyStock           5.831e-01  1.335e-02  43.696  < 2e-16 ***
checkableDeposits   -6.808e-04  1.056e-04  -6.450 1.71e-10 ***
tradeCurrencies      7.064e-04  5.993e-05  11.787  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2422 on 1042 degrees of freedom
Multiple R-squared:  0.9949,	Adjusted R-squared:  0.9949
F-statistic: 3.38e+04 on 6 and 1042 DF,  p-value: < 2.2e-16
```

Figura 134: Resultados eliminando la variable federalFunds, 3M-Rate-AuctionAverage, 3M-Rate-SecondaryMarket, 5Y-CMaturityRate, demandDeposits, bankCredit, savingsDeposits, currency y loansLeases.

El ajuste no ha empeorado y de nuevo tenemos una situación en la que no podemos eliminar a priori ninguna de las variables. La que tiene un p-valor más grande es checkableDeposits.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds - `3M-Rate-AuctionAverage` -
    `3M-Rate-SecondaryMarket` - `5Y-CMaturityRate` - demandDeposits -
    bankCredit - savingsDeposits - currency - loansLeases - checkableDeposits,
    data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max
-1.24106 -0.12090 -0.02788  0.06900  2.02145

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -1.909e-01  8.531e-02  -2.237   0.0255 *
`1Y-CMaturityRate`  -5.586e-03  6.546e-04  -8.533  < 2e-16 ***
`30Y-CMortgageRate`  1.447e-01  1.775e-02   8.150 1.03e-15 ***
`3Y-CMaturityRate`   3.327e-01  2.538e-02  13.106  < 2e-16 ***
moneyStock           5.976e-01  1.341e-02  44.576  < 2e-16 ***
tradeCurrencies      3.991e-04  3.706e-05  10.770  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2468 on 1043 degrees of freedom
Multiple R-squared:  0.9947,	Adjusted R-squared:  0.9947
F-statistic: 3.903e+04 on 5 and 1043 DF,  p-value: < 2.2e-16
```

Figura 135: Resultados eliminando la variable federalFunds, 3M-Rate-AuctionAverage, 3M-Rate-SecondaryMarket, 5Y-CMaturityRate, demandDeposits, bankCredit, savingsDeposits, currency, loansLeases y checkableDeposits.

El ajuste no ha empeorado casi el modelo por lo que vamos a continuar reduciendo el problema. Podemos ver que de todas las variables iniciales, las que explican el problema y lo resuelven son muchas menos. Vamos a eliminar la variable 30Y-CMortgageRate.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds - `3M-Rate-AuctionAverage` -
    `3M-Rate-SecondaryMarket` - `5Y-CMaturityRate` - demandDeposits -
    bankCredit - savingsDeposits - currency - loansLeases - checkableDeposits -
    `30Y-CMortgageRate`, data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max
-1.25244 -0.12692 -0.04326  0.07020  2.00489

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        -1.404e-01  8.771e-02  -1.600     0.11
`1Y-CMaturityRate` -3.810e-03  6.363e-04  -5.987 2.93e-09 ***
`3Y-CMaturityRate`  4.930e-01  1.655e-02  29.791  < 2e-16 ***
moneyStock          5.787e-01  1.361e-02  42.514  < 2e-16 ***
tradeCurrencies     3.122e-04  3.658e-05   8.534  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2545 on 1044 degrees of freedom
Multiple R-squared:  0.9943,    Adjusted R-squared:  0.9943
F-statistic: 4.589e+04 on 4 and 1044 DF,  p-value: < 2.2e-16
```

Figura 136: Resultados eliminando la variable federalFunds, 3M-Rate-AuctionAverage, 3M-Rate-SecondaryMarket, 5Y-CMaturityRate, demandDeposits, bankCredit, savingsDeposits, currency, loansLeases, checkableDeposits y 30Y-CMortgageRate.

Estamos empeorando en el ajuste de forma muy ligera pero merece la pena por la tremenda simplificación del problema. Como podemos ver todavía tenemos las dos variables que mejor ajuste han producido por sí solas.

Vamos a eliminar la variable con un p-valor mayor, que en este caso es 1Y-CMaturityRate.

```
Call:
lm(formula = `1MonthCDRate` ~ . - federalFunds - `3M-Rate-AuctionAverage` -
    `3M-Rate-SecondaryMarket` - `5Y-CMaturityRate` - demandDeposits -
    bankCredit - savingsDeposits - currency - loansLeases - checkableDeposits -
    `30Y-CMortgageRate` - `1Y-CMaturityRate`, data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max
-1.23890 -0.12477 -0.03665  0.07720  2.05442

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        -5.098e-01  6.337e-02  -8.044 2.35e-15 ***
`3Y-CMaturityRate`  4.734e-01  1.649e-02  28.708  < 2e-16 ***
moneyStock          5.910e-01  1.368e-02  43.221  < 2e-16 ***
tradeCurrencies     3.527e-04  3.655e-05   9.652  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2587 on 1045 degrees of freedom
Multiple R-squared:  0.9942,    Adjusted R-squared:  0.9941
F-statistic: 5.92e+04 on 3 and 1045 DF,  p-value: < 2.2e-16
```

Figura 137: Resultados eliminando la variable federalFunds, 3M-Rate-AuctionAverage, 3M-Rate-SecondaryMarket, 5Y-CMaturityRate, demandDeposits, bankCredit, savingsDeposits, currency, loansLeases, checkableDeposits, 30Y-CMortgageRate y 1Y-CMaturityRate.

Hemos obtenido un modelo mucho más reducido que el inicial que tiene un $R^2$ ajustado de 0,9942, lo cual es más que lo que hemos obtenido en la mejor de las regresiones lineales simples. Hemos disminuido un poco el rendimiento, pues inicialmente partíamos de un modelo que tenía un $R^2$ ajustado de 0,9951 pero he decidido sacrificar un poco de rendimiento (hemos bajado sólo 0,001 en el $R^2$ ajustado) por simplificar enormemente el modelo.

Claramente el modelo obtenido es superior al modelo simple pues mejoramos el rendimiento únicamente empleando dos variables más. Recordemos que el mejor rendimiento lo obtuvimos con la variable moneyStock que nos dio un $R^2$ ajustado de 0,9893 y hemos subido el rendimiento a 0,9942. Si queremos primar la simplicidad del modelo frente a un mejor rendimiento está claro que el modelo de una única variable es suficientemente bueno. Por contra si queremos un poco más de rendimiento a costa de aumentar ligeramente la complejidad de la solución podemos optar por el modelo de regresión múltiple de 3 variables que hemos generado.

Recordemos que en la fase de EDA hicimos suposiciones sobre las variables que podíamos eliminar. En el estudio exploratorio concluimos que a priori parecían interesantes las variables 7,10,11,13,14 y 15. Veamos el desempeño de dicho modelo.

```
Call:
lm(formula = `1MonthCDRate` ~ `1Y-CMaturityRate` + `3M-Rate-SecondaryMarket` +
    `3Y-CMaturityRate` + currency + federalFunds + savingsDeposits +
    tradeCurrencies, data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max
-1.15477 -0.22895 -0.03515  0.14017  2.64796

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                0.2002195  0.2427939   0.825  0.40976
`1Y-CMaturityRate`        -0.0085030  0.0012390  -6.863 1.16e-11 ***
`3M-Rate-SecondaryMarket`  0.2125098  0.0648213   3.278  0.00108 **
`3Y-CMaturityRate`         0.9498319  0.0647961  14.659  < 2e-16 ***
currency                  -0.0011629  0.0002806  -4.144 3.69e-05 ***
federalFunds               0.0001122  0.0006700   0.167  0.86707
savingsDeposits            0.0015656  0.0003486   4.492 7.85e-06 ***
tradeCurrencies            0.0001831  0.0001568   1.167  0.24339
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4127 on 1041 degrees of freedom
Multiple R-squared:  0.9852,    Adjusted R-squared:  0.9851
F-statistic:  9877 on 7 and 1041 DF,  p-value: < 2.2e-16
```

Figura 138: Resultados con las variables escogidas en el EDA.

Podemos ver que el rendimiento del modelo es incluso peor que el que nos proporciona la variable moneyStock por sí sola por lo que este modelo no es una buena opción.

En cuanto a interacciones no lineales podemos probar con el modelo simple de 3 variables que hemos obtenido haciendo algunas transformaciones no lineales, pero debemos tener presente que es complicado obtener resultados mucho mejores que los que ya tenemos.

```
Call:
lm(formula = `1MonthCDRate` ~ `3Y-CMaturityRate` + moneyStock +
    tradeCurrencies + `3Y-CMaturityRate` * `3Y-CMaturityRate` +
    moneyStock * moneyStock + tradeCurrencies * tradeCurrencies +
    `3Y-CMaturityRate` * moneyStock + `3Y-CMaturityRate` * tradeCurrencies +
    moneyStock * tradeCurrencies, data = dataset_regresion)

Residuals:
     Min       1Q   Median       3Q      Max
-1.01003 -0.12307 -0.04249  0.06876  2.05743

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                       -2.035e-02  1.554e-01  -0.131  0.89578
`3Y-CMaturityRate`                 5.207e-01  3.235e-02  16.095  < 2e-16 ***
moneyStock                         4.560e-01  2.896e-02  15.743  < 2e-16 ***
tradeCurrencies                    1.556e-04  1.059e-04   1.469  0.14221
`3Y-CMaturityRate`:moneyStock      4.339e-03  1.069e-03   4.058 5.32e-05 ***
`3Y-CMaturityRate`:tradeCurrencies -1.517e-04  4.679e-05  -3.243  0.00122 **
moneyStock:tradeCurrencies         1.599e-04  3.967e-05   4.030 5.99e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2552 on 1042 degrees of freedom
Multiple R-squared:  0.9943,	Adjusted R-squared:  0.9943
F-statistic: 3.042e+04 on 6 and 1042 DF,  p-value: < 2.2e-16
```

Figura 139: Regresión con interacciones no lineales.

Como podemos ver por la fórmula que estamos aplicando estamos considerando todas las posibles interacciones cuadráticas que implican las 3 variables consideradas en el último modelo. Como podemos ver no obtenemos una mejora significativa, pues mejoramos el $R^2$ ajustado solamente 0,0001 por lo tanto no merece la pena.

## 2.3. Algoritmo k-NN para regresión

En esta sección vamos a emplear el algoritmo KNN para regresión. Para ello vamos a aplicarlo primero con todas las variables, después con la que mejor resultado nos ha dado en los modelos de regresión simple y posteriormente con la mejor fórmula del modelo de regresión múltiple.

Para comparar los algoritmos vamos a emplear el error en predicción en train y en test por cada fold y en media. A priori además no sabemos el valor que debemos emplear para K, por lo que vamos a hacer un estudio al respecto para elegir bien dicho valor variándolo entre 3 y 55 a saltos de 2.


Figura 140: Error en test del modelo KNN con todas las variables.

Podemos ver claramente que el valor de K para el que se obtiene el menor error es para $K = 5$ donde obtenemos de media 0,04609969 de error en los folds. Si estudiamos el error por fold vemos que tenemos los siguientes números:

| **Folds** | **Error** |
|---|---|
| 1 | 0.07921098 |
| 2 | 0.05298160 |
| 3 | 0.02544653 |
| 4 | 0.04156857 |
| 5 | 0.03129077 |

En cuanto al error en el train para cada valor de K obtenemos la siguiente gráfica:

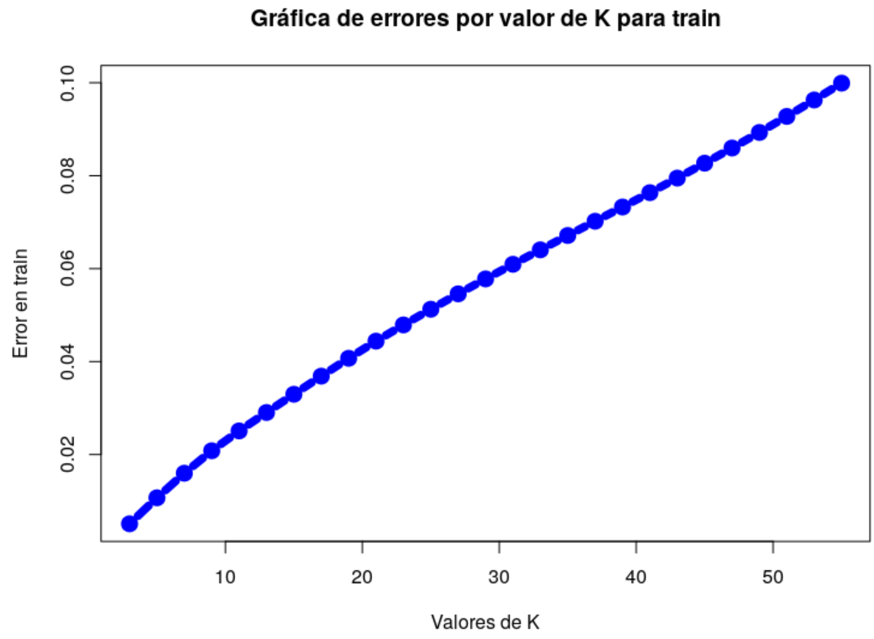


Figura 141: Error en train del modelo KNN con todas las variables.

Podemos ver que el menor error se comete con $K = 3$ donde obtenemos un error medio de 0,00509912 en la validación cruzada. El error cometido por cada fold es:

| **Fold** | **Error** |
|---|---|
| 1 | 0.005395550 |
| 2 | 0.004238368 |
| 3 | 0.004765470 |
| 4 | 0.005374334 |
| 5 | 0.005721879 |

Vamos a ver ahora el error que se comete empleando la mejor variable de los modelos lineales simples.


Figura 142: Error en test del modelo KNN con la variable moneyStock.

Podemos ver en esta gráfica que el modelo con una sola variable es muy dependiente del valor de K. El mejor valor que podemos tomar para este parámetro es 21 ya que es el que menos error nos da en media en la validación cruzada. El error cometido en media en el test es 0,1237461 y por cada fold:

| **Fold** | **Error** |
|---|---|
| 1 | 0.15658084 |
| 2 | 0.16779980 |
| 3 | 0.10178984 |
| 4 | 0.10885809 |
| 5 | 0.08370183 |

Veamos el error en el train:


Figura 143: Error en train del modelo KNN con la variable moneyStock.

Como podemos ver en el caso del train el mejor resultado de error se obtiene para el valor de $K = 3$. En este caso se comete un error de 0,06193327. El error por folds es:

| **Fold** | **Error** |
|---|---|
| 1 | 0.05902647 |
| 2 | 0.05254069 |
| 3 | 0.06674650 |
| 4 | 0.06400127 |
| 5 | 0.06735141 |

Vamos a ver ahora el desempeño que tiene el modelo KNN con las tres variables que tomamos finalmente en el estudio de regresión múltiple.


Figura 144: Error en test del modelo KNN con las variables 3Y-CMaturityRate, moneyStock y tradeCurrencies.

Podemos ver que el menor valor de error se produce con el valor $K = 9$ siendo dicho error en media 0,05378016. Si analizamos el error por fold podemos ver que obtenemos los siguientes resultados:

| **Fold** | **Error** |
|---|---|
| 1 | 0.08178329 |
| 2 | 0.06936034 |
| 3 | 0.03370024 |
| 4 | 0.04640869 |
| 5 | 0.03764825 |

Veamos ahora el mejor valor de K según el error cometido en el train y comparemos finalmente los modelos.

El error cometido en media en el train según el valor de K es el siguiente:

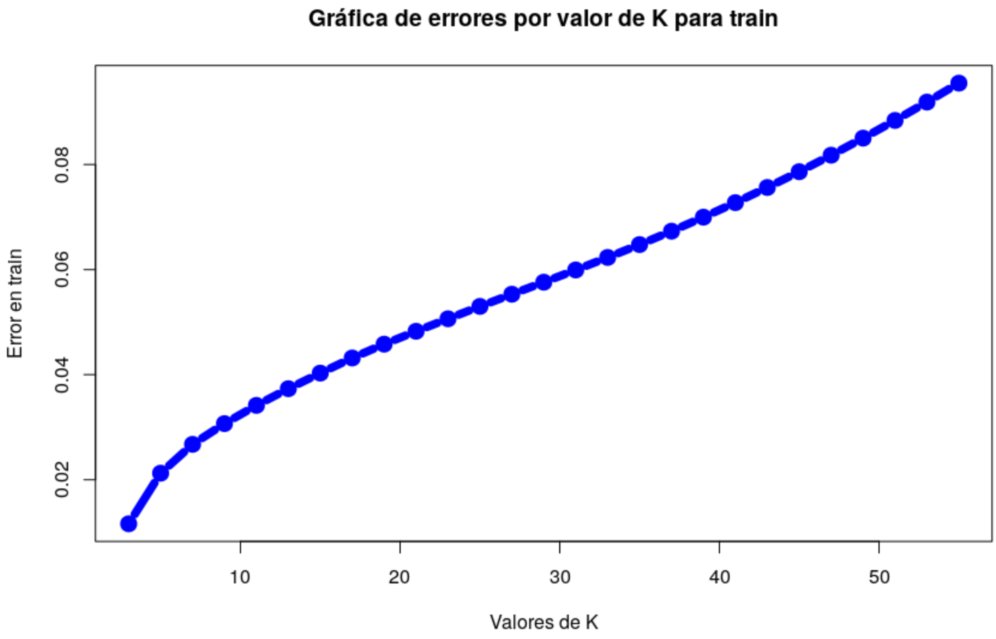


Figura 145: Error en train del modelo KNN con las variables 3Y-CMaturityRate, moneyStock y tradeCurrencies.

Podemos ver que el menor error se comete para el valor de K igual a 3, donde se comete en media un error de 0,01161164. Si analizamos el error por folds obtenemos los siguientes resultados:

| **Fold** | **Error** |
|---|---|
| 1 | 0.00975913 |
| 2 | 0.01180357 |
| 3 | 0.01163929 |
| 4 | 0.01197810 |
| 5 | 0.01287809 |

Vamos a comparar por tanto los modelos obtenidos. Para esta comparación vamos a tanto el error por folds en el test como el error medio cometido en los mismos. Se nos podría plantear en esta sección el dilema de si emplear el error en train o en test para hacer la comparativa o ambos incluso. Desde un punto de vista objetivo debemos usar el error cometido sobre el test. Nosotros estamos suponiendo que disponemos de datos que conocemos (los del train) y estamos entrenando el modelo con ellos para luego evaluarlo con unos datos nuevos simulando que son datos desconocidos para nosotros como si nos llegasen en una aplicación real del modelo. Por tanto lo más consistente es evaluarlo empleando el error cometido en el test.

Veamos la siguiente tabla:

| **Folds** | **Error modelo todas** | **Error moneyStock** | **Error modelo tres variables** |
|---|---|---|---|
| 1 | 0.07921098 | 0.15658084 | 0.08178329 |
| 2 | 0.05298160 | 0.16779980 | 0.06936034 |
| 3 | 0.02544653 | 0.10178984 | 0.03370024 |
| 4 | 0.04156857 | 0.10885809 | 0.04640869 |
| 5 | 0.03129077 | 0.08370783 | 0.03764825 |
| **Error medio** | 0.04609969 | 0.1237461 | 0.05378016 |

Podemos comprobar que el peor modelo en cuanto a error cometido es el modelo que emplea una única variable para el ajuste con KNN. Este modelo comete un error mayor en todos los folds comparado con los otros dos modelos del KNN, lo cual se hace evidente también en el error medio como era de esperar.

Entre el modelo de tres variables y el modelo que hace uso de todas podemos ver que el modelo que comete un error menor es el modelo que emplea todas las variables. Esto es algo lógico pues lo vimos también en el ajuste de regresión lineal múltiple que con todas las variables obteníamos un buen resultado. En ese caso primamos la simplicidad que ganábamos frente a una pérdida de rendimiento menor. Si observamos la diferencia de errores cometidos es de 0,00768047 en el error medio, lo cual es prácticamente insignificante y por tanto es muy lógico pensar que el modelo más simple de tres variables es una mejor opción que el modelo que hace uso de todas.

## 2.4. Test de significancia entre algoritmos

Queremos ver ahora si los algoritmos son significativamente diferentes por los resultados que nos han arrojado. Para ello primero vamos a hacer una comparativa entre los algoritmos KNN y regresión lineal múltiple con la configuración de todas las variables.

Para empezar vamos a hacer la comparativa entre el modelo de regresión lineal múltiple y el modelo de KNN. Para hacer esta comparativa vamos a emplear el test de Wilcoxon que nos va a decir si las diferencias entre algoritmos son significativas. Para ello disponemos de dos archivos CSV que contienen los resultados de un modelo de regresión lineal, un KNN y un M5P para distintos conjuntos de datos. En esta hoja se han introducido los errores cometidos por los mejores modelos conseguidos para regresión lineal y para KNN en el conjunto treasury que es el que hemos estudiado en este trabajo.

Con esto hecho podemos realizar el test de Wilcoxon entre los modelos de regresión lineal y KNN.

| **Datos** | $R^+$ | $R^-$ | **P-value** |
|---|---|---|---|
| **Test** | 92 | 79 | 0.7987061 |
| **Train** | 10 | 161 | 0.000328064 |

Como podemos ver por el p-valor las diferencias que encontramos en el test no son significativas, es decir, los algoritmos no son estadísticamente diferentes. En cuanto al train podemos ver que sí existe dicha diferencia y por tanto son significativamente diferentes. De hecho podemos decir que con un 99,9671936 % de probabilidad los algoritmos son diferentes.

Creo que igual que hemos comentado antes la comparativa justa es en el test, pues es el conocimiento que nos viene nuevo al modelo y con el que no se ha entrenado. Lo que sí podemos decir por los resultados del train es que, al ser significativamente diferentes y ser el mejor modelo el KNN podemos decir que KNN se ajusta muy bien al conjunto de train y por tanto podemos estar ante un posible sobreajuste ya que dicha significancia no se mantiene en el test.

Como en test no hemos visto diferencias significativas vamos a aplicar el test de Friedman entre los tres modelos para ver si entre los tres existen diferencias.

| Datos | P-value |
|---|---|
| **Test** | 0.06948 |
| **Train** | 3.843e-05 |

Podemos ver que el p-valor en ambos casos es pequeño. En el caso de train vemos que claramente la diferencia es significativa. Esto es algo lógico pues si teníamos ya dicha diferencia significativa en train entre KNN y el modelo lineal esto se va a mantener en el Friedman.

En el caso de test podemos ver que tenemos un valor ligeramente superior a 0,05. Normalmente esta es la cota a partir de la cual decimos que se puede rechazar la hipótesis nula por lo que estamos en el umbral de poder decir que la diferencia entre los algoritmos es significativa. Habiendo analizado el comportamiento previo entre KNN y el modelo lineal y sabiendo que no existen diferencias entre ellos podemos deducir que esta diferencia la está marcando el algoritmo M5P.

Para poder comprobar esto vamos a realizar el test post-hoc de Holm. Veamos los resultados en train y test.

| Modelos | 1 | 2 |
|---|---|---|
| **2** | 0.0067 | - |
| **3** | 0.0039 | 0.0067 |

Cuadro 1: Test post-hoc Holm en train

| Modelos | 1 | 2 |
|---|---|---|
| **2** | 0.97 | - |
| **3** | 0.27 | 0.27 |

Cuadro 2: Test post-hoc Holm en test

En este test ya podemos ver que en train la diferencia significativa es entre todos ellos, mientras que en test tenemos que los modelos KNN y regresión lineal no tienen diferencias significativas con un p-valor muy alto tenemos que con el modelo M5P tampoco existen dichas diferencias significativas pero con un pvalor menor.

Por tanto podemos sacar dos conclusiones de todos estos tests. La primera de ellas es que no existe una diferencia significativa entre los mismos cuando los aplicamos al conjunto de test. La segunda es que es bastante probable que tengamos sobreajuste en los modelos ya que en train si que obtenemos que las diferencias son bastante significativas.

# 3. Clasificación: heart

Vamos ahora a estudiar el problema de clasificación. Ya hemos visto en el EDA el problema que tenemos con las variables ya que todas son categóricas y la mayoría binarias lo cual no nos da mucha información y podemos intuir que la clasificación no va a funcionar demasiado bien.

## 3.1. Algoritmo KNN

Vamos a probar en esta sección el uso del algoritmo KNN para el problema dado. A priori no sabemos cuál será la mejor configuración del valor de K, por lo que tendremos que hacer primero un estudio del mismo.

Para evaluar el desempeño del modelo se ha seguido un esquema de validación cruzada con 10 folds. De todos los folds se obtiene un accuracy del modelo entrenando con el correspondiente porcentaje y prediciendo sobre el resto. Con esto al final obtenemos 10 porcentajes de acierto de los que podemos hacer la media. Para evaluar el mejor valor de K vamos a emplear esa media. Veamos los resultados para train y test:

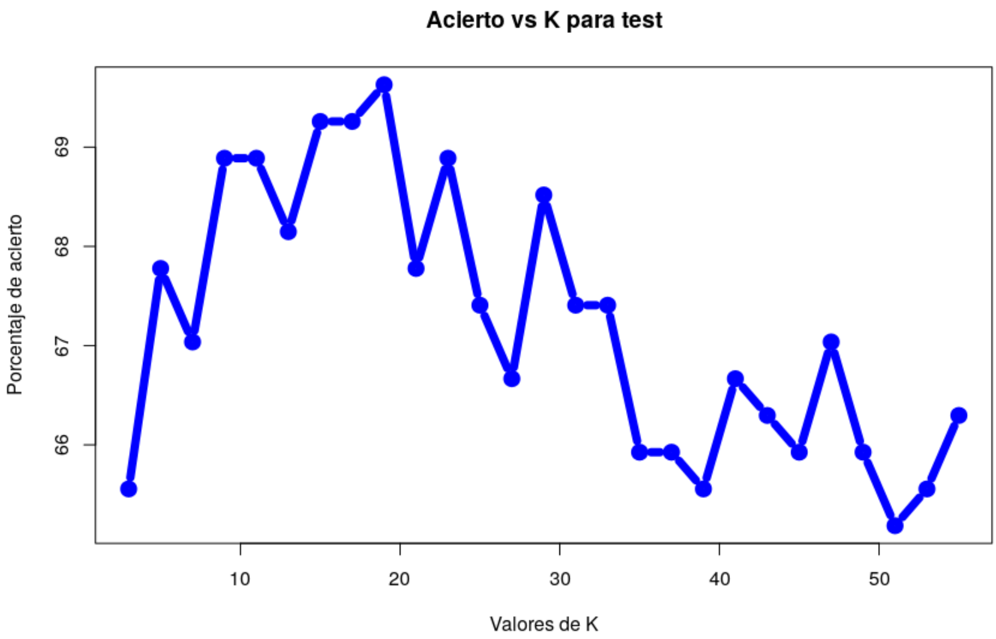


Figura 146: Porcentaje de acierto frente al valor de K para test.

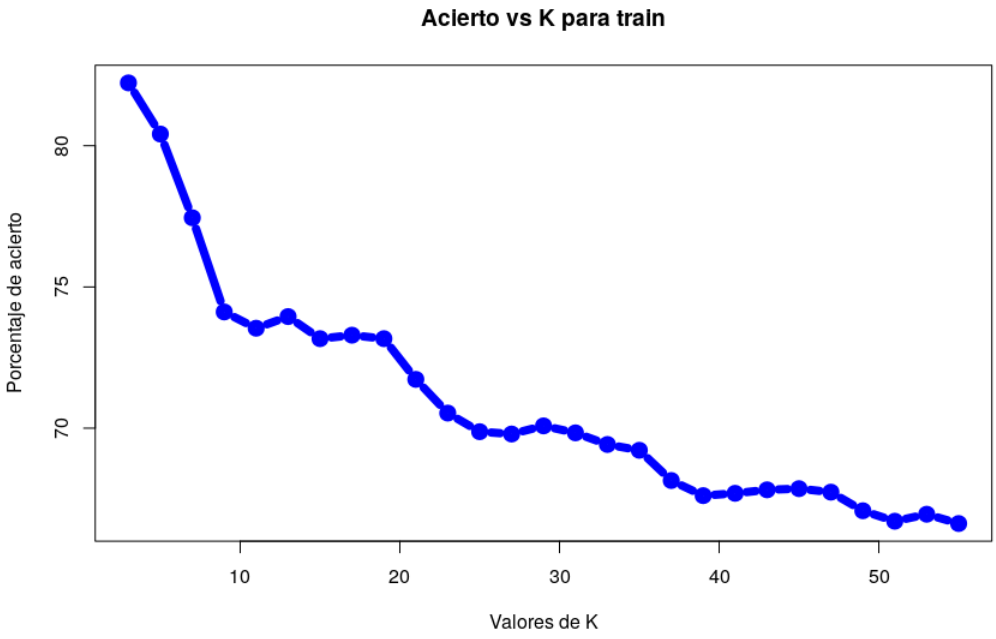


Figura 147: Porcentaje de acierto frente al valor de K para train.

Podemos ver que en el caso de train el porcentaje de acierto disminuye a medida que aumentamos el valor de K, por lo que $K = 3$ es el mejor de los valores para el train. Este comportamiento es lógico ya que a menor valor K más peso tendrá la instancia que posee la clase real que debe tener el dato.

En el caso de test podemos ver que se produce una cierta fluctuación de los aciertos. Debemos fijarnos y hacer notar que la oscilación no es demasiado grande, ya que el rango de valores es pequeño. En concreto para el test se obtiene el mejor rendimiento con $K = 19$ donde se obtiene un rendimiento del 69,62963 %. En el caso del train con $K = 3$ el rendimiento que se obtuvo fue de un 82,22222 %.

El rendimiento en train lo podemos usar para estudiar el sobreajuste del modelo para el valor de K elegido en el test. Como hemos dicho antes el mejor valor que se produce de acierto en el test se produce cuando K vale 19. Con este valor de K obtiene el modelo en train un porcentaje de acierto de 73,16872 %. Este valor no nos indica un sobreajuste puesto que el rendimiento del modelo no es excesivamente bueno.

Podemos plantearnos la pregunta de por qué nuestro modelo no funciona bien para este conjunto de datos. Para poder responder a esta pregunta vamos a hacer uso de la técnica de reducción de dimensionalidad T-SNE que nos va a proporcionar una reducción de los datos a dos dimensiones, con lo que vamos a poder ver cómo de solapadas están ambas clases.

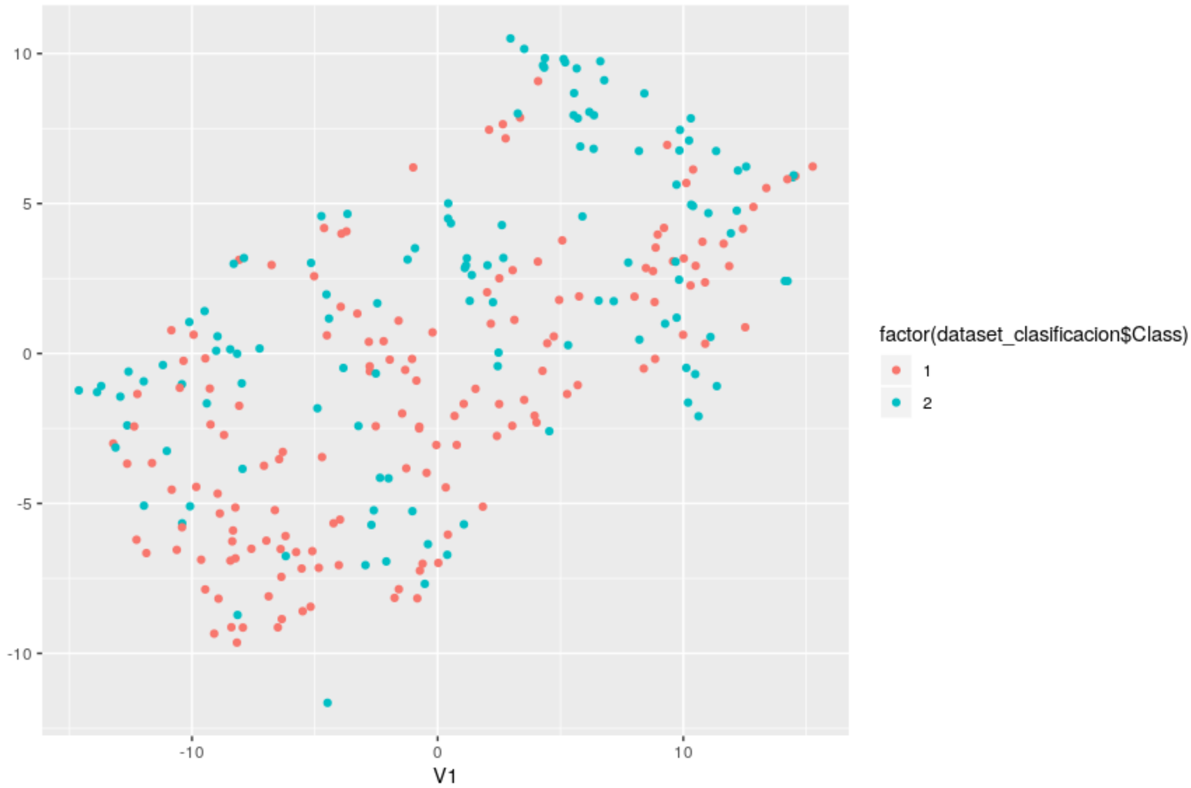


Figura 148: Reducción de dimensionalidad para visualización con T-SNE.

Debemos tener en cuenta que esta reducción de dimensionalidad no tiene por qué ser muy precisa en la visualización que nos está dando puesto que la mayoría de las variables son categóricas (y binarias) por lo que la información de la que disponemos no es especialmente buena.

Aún así podemos ver que el problema parece ser que las clases están muy pegadas e incluso mezcladas. En las siguientes secciones detallaremos un poco más este punto pues podremos dar algo más de información con LDA y QDA.

Para poder tener unos resultados consistentes que podamos comparar posteriormente con el resto de los modelos vamos a obtener el error en test y train con el valor de $K = 19$.

| **Fold** | **Train** | **Test** |
|---|---|---|
| **1** | 72.83951 | 66.66667 |
| **2** | 72.83951 | 70.37037 |
| **3** | 74.89712 | 62.96296 |
| **4** | 72.83951 | 77.77778 |
| **5** | 70.37037 | 81.48148 |
| **6** | 74.4856 | 70.37037 |
| **7** | 73.66255 | 66.66667 |
| **8** | 74.07407 | 59.25926 |
| **9** | 72.83951 | 74.07407 |
| **10** | 72.83951 | 66.66667 |
| **Acierto Medio** | 73.16872 | 69.62963 |

Cuadro 3: Resultados para KNN.

## 3.2. Algoritmo LDA

Vamos a estudiar en esta sección el comportamiento del algoritmo LDA sobre el conjunto de datos que tenemos. Recordemos que el fundamento de LDA es trazar una recta que divida bien los datos, por lo que debemos tener en cuenta que si nuestro conjunto no es separable linealmente o si no tiene una frontera que pueda ser separada por una línea no obtendremos un resultado bueno.

La metodología seguida es la misma que en el caso de KNN. Vamos a tener 10-FCV por lo que vamos a obtener 10 medidas de accuracy para el modelo tanto para train como para test.

Veamos los resultados para train y test en media y por fold:

| **Fold** | **Train** | **Test** |
|---|---|---|
| **1** | 85.59671 | 85.18519 |
| **2** | 86.41975 | 85.18519 |
| **3** | 85.59671 | 85.18519 |
| **4** | 86.00823 | 85.18519 |
| **5** | 85.18519 | 85.18519 |
| **6** | 84.77366 | 88.88889 |
| **7** | 85.18519 | 96.2963 |
| **8** | 86.41975 | 77.77778 |
| **9** | 88.06584 | 62.96296 |
| **10** | 84.36214 | 96.2963 |
| **Acierto Medio** | 85.76132 | 84.81481 |

Cuadro 4: Tabla de resultados para LDA.

Podemos ver que el resultado es mucho mejor que el obtenido por KNN en clasificación por lo que estamos ante un mejor modelo. No sólo supera al KNN en el resultado en media, sino en cada uno de los folds.

Podemos ver que LDA obtiene un resultado mejor que KNN como hemos mencionado. Si queremos intentar intuir un poco las fronteras de ambas clases podemos pensar que están muy juntas. Esto puede provocar que KNN se confunda ya que cada punto va a tener cerca puntos de su clase y de la contraria, pero no implica que la frontera no sea separable como LDA nos está demostrando.

## 3.3. Algoritmo QDA

Ahora vamos a probar el algoritmo QDA sobre nuestros datos. Con esta ejecución vamos a obtener algo más de información sobre las clases y su separación. Si obtenemos mejores resultados sabremos que la división de las clases no es puramente lineal y necesitamos de términos de mayor orden. Por contra si obtenemos los mismos o peores podremos deducir que la frontera está más definida a través del orden lineal.

Veamos la tabla de resultados:

| Fold | Train | Test |
|---|---|---|
| **1** | 88.88889 | 77.77778 |
| **2** | 86.41975 | 92.59259 |
| **3** | 87.65432 | 85.18519 |
| **4** | 87.2428 | 74.07407 |
| **5** | 87.2428 | 85.18519 |
| **6** | 87.65432 | 85.18519 |
| **7** | 88.06584 | 88.88889 |
| **8** | 87.2428 | 77.77778 |
| **9** | 90.12346 | 66.66667 |
| **10** | 87.2428 | 96.2963 |
| **Acierto Medio** | 87.77778 | 82.96296 |

Cuadro 5: Tabla de resultados para QDA.

Si comparamos el acierto medio en test de QDA y LDA podemos ver que LDA es ligeramente superior. Por contra QDA ajusta mejor en train. Esto nos puede estar indicando que este modelo sobreajusta, pues aunque es mejor en train que LDA luego no tiene mejor desempeño en test.

Si hacemos el análisis por folds podemos ver que no está tan claro pues tenemos folds de LDA que tienen mejor acierto y folds de QDA que tienen mejor acierto, por tanto deberemos esperar hasta el test estadístico para ver si existen diferencias significativas. De momento en el acierto medio podemos ver que LDA es mejor para el test.

Si hablamos de las fronteras del problema podemos ver que no hemos mejorado, por lo que las fronteras no son cuadráticas y se ajustan más a un modelo lineal.

## 3.4. Comparativa de algoritmos

Para hacer esta comparativa vamos a emplear distintos datos. En primer lugar vamos a comparar el acierto medio de los modelos, así como el acierto por folds y su desviación típica.

Para que sea una comparativa más detallada y estadísticamente fundamentada vamos a emplear los tests de Wilcoxon y post-hoc de Holm para estudiar si las diferencias entre algoritmos es significativa o no.

Para empezar vamos a hacer el test de Wilcoxon por parejas entre todos los modelos. Se han hecho los tests entre KNN y LDA, KNN y QDA y LDA y QDA. Veamos las tablas:

| **Test** | **KNN vs LDA** | **LDA vs QDA** | **KNN vs QDA** |
|---|---|---|---|
| $R^+$ | 0 | 6.5 | 0 |
| $R^-$ | 55 | 14.5 | 55 |
| **p-value** | 0.00588927 | 0.461838 | 0.001853125 |
| **Train** | **KNN vs LDA** | **LDA vs QDA** | **KNN vs QDA** |
| $R^+$ | 0 | 45 | 0 |
| $R^-$ | 55 | 0 | 55 |
| **p-value** | 0.001953125 | 0.009151689 | 0.00588927 |

Cuadro 6: Tabla de tests de Wilcoxon.

Este test se ha hecho empleando los datos de los 10 folds para poder tener unas medidas algo más completas que el hecho de comparar entre sí la media del acierto en los mismos.

En la tabla podemos ver que hay dos partes: train y test. Si queremos estudiar el rendimiento de los modelos vamos a comparar los resultados en test y posteriormente veremos train para estudiar el sobreajuste.

Podemos ver en test que no hay diferencias significativas entre LDA y QDA como hemos podido intuir de secciones anteriores. En cambio entre LDA y KNN y QDA y KNN hay diferencias significativas. Como hemos visto el rendimiento de KNN es sensiblemente menor que el que obtiene LDA y QDA y esto se ve reflejado en estos tests entre los resultados de los modelos.

En el caso del train podemos ver que todos los algoritmos tienen diferencia significativas con el resto de los mismos. Es claro que LDA y QDA ajustan mejor en train que KNN pero estos resultados nos hacen pensar que este mejor ajuste no sólo se produce en test sino en train también a favor de LDA. Por tanto es posible que LDA o QDA sobreajusten.

Vamos a ver ahora el test múltiple:

| **Train** | **1** | **2** |
|---|---|---|
| **2** | 0.059 | - |
| **3** | 0.0118 | 0.0118 |
| **Test** | **1** | **2** |
| **2** | 0.0118 | - |
| **3** | 0.0059 | 0.3387 |

Cuadro 7: Resultados del test post-hoc de Holm.

El test post-hoc de Holm nos está dando la misma información que teníamos para test, esto es, todos los algoritmos son significativamente diferentes. En el caso de train tenemos un resultado que no se comparte con lo que teníamos previamente. Podemos ver que en train QDA es significativamente diferente de KNN y LDA pero KNN y LDA no lo son entre sí o no podemos afirmar nada de ellos pues su p-valor no es menor que 0,05. Aún así la diferencia es muy pequeña y puede haberse producido por el error que acumula el test múltiple a medida que aumentamos el número de algoritmos a probar.

Ya hemos comparado en las secciones previas el acierto medio de cada modelo. Otro estadístico interesante es la desviación típica del acierto entre folds. Los resultados son los siguientes:

| **Test** | **Desviación Típica** |
|---|---|
| **KNN** | 6.716776 |
| **LDA** | 9.474856 |
| **QDA** | 8.936771 |
| **Train** | **Desviación Típica** |
| **KNN** | 1.254225 |
| **LDA** | 1.048284 |
| **QDA** | 1.045588 |

Cuadro 8: Desviaciones típicas en los aciertos de los modelos.

Este dato nos proporciona información acerca de la robustez del ajuste del modelo, a menor desviación típica más consitente es el resultado y por tanto es algo bueno para el modelo. Como podemos comprobar en test el que menos oscila es el algoritmo KNN, seguido de QDA y después LDA. Esto nos puede estar hablando de un sobreajuste de LDA y QDA muy acusado que beneficia en el funcionamiento de algunos folds pero no en el caso general y por tanto en media superan a KNN.

En el caso de train vemos que los resultados son muy parecidos entre sí y por tanto no tenemos mucha más información de la que no dispusiéramos con anterioridad.

# 4. Código

## 4.1. EDA

```r
# ##########################################################################################
# ##########################################################################################
##### ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ #####
##### DATASET REGRESION #####
##### ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ #####
# ##########################################################################################
# ##########################################################################################

library(foreign)
library(dplyr)
dataset_regresion<-read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_
    Regresion/treasury/treasury.dat")
dataset_regresion

# Ahora vamos a ver el numero de variables y el tipo de cada una.
cat("El numero de variables es: ", length(colnames(dataset_regresion)), "\n")
cat("El tipo de las variables es:\n")
for(i in 1:length(colnames(dataset_regresion))){
cat("\t",colnames(dataset_regresion)[i], ": ", class(dataset_regresion[i][[1]]), "\n")
}

# Como podemos ver todas las variables son de tipo numerico
# Ahora vamos a calcular una serie de estadisticos interesantes para cada variable.
library(e1071)

# Funcion para calcular la moda
calculaModa <- function(v) {
uniqv <- unique(v)
uniqv[which.max(tabulate(match(v, uniqv)))]
}

# Funcion que nos da los estadisticos de una columna del dataset
calculaEstadisticos<-function(col){
estadisticos<-list(media=NA, mediana=NA, stdv=NA, moda=NA, curtosis=NA, asimetria=NA, minimo=NA, maximo=NA)
estadisticos$media<-mean(col, na.rm=TRUE)
estadisticos$mediana<-median(col, na.rm=TRUE)
estadisticos$stdv<-sd(col, na.rm = TRUE)
estadisticos$moda<-calculaModa(col)
estadisticos$kurtosis<-kurtosis(col, na.rm=TRUE)
estadisticos$asimetria<-skewness(col, na.rm=TRUE)
estadisticos$minimo<-min(col, na.rm=TRUE)
estadisticos$maximo<-max(col, na.rm=TRUE)
return(estadisticos)
}

# Imprimimos bien formateados los estadisticos
cat("\n\n\n", "Estadisticos de las columnas:\n")
for(i in 1:length(colnames(dataset_regresion))){
cat("\t", colnames(dataset_regresion)[i], ":", "\n")
estadisticos<-calculaEstadisticos(dataset_regresion[,i])
cat("\t\t", "Media: ", estadisticos$media, "\n")
cat("\t\t", "Mediana: ", estadisticos$mediana, "\n")
cat("\t\t", "Desviacion tipica: ", estadisticos$stdv, "\n")
cat("\t\t", "Moda: ", estadisticos$moda, "\n")
cat("\t\t", "Kurtosis: ", estadisticos$kurtosis, "\n")
cat("\t\t", "Asimetria: ", estadisticos$asimetria, "\n")
cat("\t\t", "Minimo: ", estadisticos$minimo, "\n")
cat("\t\t", "Maximo: ", estadisticos$maximo, "\n")
}

#########################################################################
## Estudio de correlacion de las variables ##
#########################################################################

# Funcion que obtiene las variables cuya correlacion supere threshold
```

```
obtainCorrelated<-function(varIndex, dataset, threshold=0.9){
combinations<-combn(varIndex,2)

correlations<-vector("numeric", dim(combinations)[2])

for(i in 1:dim(combinations)[2]){
pair<-combinations[,i]
correlations[i]<-cor(dataset[,pair[1]], dataset[,pair[2]], method = c("pearson", "kendall", "spearman"))
}

for(i in 1:length(correlations)){
if(abs(correlations[i])>threshold){
cat("La pareja de variables: ", combinations[,i][1], ", ", combinations[,i][2], " tiene una correlacion: ", correlations[i], "\n")
}
}
}

obtainCorrelated(1:(length(dataset_regresion)-1), dataset_regresion)

# Vemos que la variable 2 esta altamente correlada con el resto, la quitamos y repetimos
obtainCorrelated(c(1,3:(length(dataset_regresion)-1)), dataset_regresion)

# La variable 9 podemos quitarla tambien
obtainCorrelated(c(1,3:8,10:(length(dataset_regresion)-1)), dataset_regresion)

# La 4 podemos quitarla tambien
obtainCorrelated(c(1,3,5:8,10:(length(dataset_regresion)-1)), dataset_regresion)

# La 3 podemos quitarla tambien
obtainCorrelated(c(1,5:8,10:(length(dataset_regresion)-1)), dataset_regresion)

# La 12 podemos quitarla
obtainCorrelated(c(1,5:8,10:11,13:(length(dataset_regresion)-1)), dataset_regresion)

# La 5 podemos quitarla
obtainCorrelated(c(1,6:8,10:11,13:(length(dataset_regresion)-1)), dataset_regresion)

# La 6 la podemos quitar
obtainCorrelated(c(1,7:8,10:11,13:(length(dataset_regresion)-1)), dataset_regresion)

# La 8 podemos quitarla
obtainCorrelated(c(1,7,10:11,13:(length(dataset_regresion)-1)), dataset_regresion)

# Finalmente nos hemos quedado con las caracteristicas 1,7,10,11,13,14,15
obtainCorrelated(c(1,7,10:11,13:(length(dataset_regresion)-1)), dataset_regresion, -1)
# Ahora merece la pena que veamos la correlacion de las variables con la de salida


# Funcion que obtiene la correlacion de las variables con las de salida
corrSalida<-function(var,dataset){
for(v in var){
correlation<-cor(dataset[,v], dataset[,length(dataset)], method=c("pearson", "kendall", "spearman"))
cat("La correlacion de la variable ", v, " con la salida es de: ", correlation, "\n")
}
}

var<-c(1,7,10,11,13,14,15)
corrSalida(var, dataset_regresion)

# Viendo la salida parece logico que quitemos la variable 1 porque tiene una correlacion baja
var<-c(7,10,11,13,14,15)
corrSalida(var, dataset_regresion)

#####################################################################
## Valores perdidos (Missing values) ##
#####################################################################

# Vamos a ver si tenemos valores perdidos
which(is.na(dataset_regresion))

# No tenemos valores perdidos

#####################################################################
## Outliers ##
#####################################################################

# Pairplot
pairs(dataset_regresion, pch=16, col="deepskyblue")

var<-c(7,10,11,13,14,15)
pairs(dataset_regresion[,c(var,16)], pch=16, col="deepskyblue")

boxplot(dataset_regresion)
boxplot(dataset_regresion[,c(-8,-9,-10,-14,-12,-15,-13)])
boxplot(dataset_regresion[,c(-1,-8,-9,-10,-12,-15,-13,-14)])
boxplot(dataset_regresion[,var])
boxplot(dataset_regresion[,var][,c(1,3)])
plot(dataset_regresion$bankCredit)

#####################################################################
## Distribucion de las variables ##
#####################################################################
library(plyr)
library(psych)
multi.hist(dataset_regresion)

# Test de normalidad para una variable
normalityTest<-function(dataset, var=-1){
if(var==-1){
for(i in 1:length(dataset)){
cat("Test de normalidad para la variable ", i, "\n")
pvalue<-wilcox.test(dataset[,i])$p.value
cat("P-valor: ", pvalue, "\n")
```

```r
if(pvalue<0.05){
cat("Como es menor a 0.05 se rechaza la hipotesis nula, no sigue una normal.")
}
else{
cat("Como es mayor a 0.05 no podemos rechazar la hipotesis nula y por tanto no podemos afirmar nada.")
}
cat("\n\n")
}
}
else{
for(i in var){
cat("Test de normalidad para la variable ", i, "\n")
pvalue<-wilcox.test(dataset[,i])$p.value
cat("P-valor: ", pvalue, "\n")
if(pvalue<0.05){
cat("Como es menor a 0.05 se rechaza la hipotesis nula, no sigue una normal.")
}
else{
cat("Como es mayor a 0.05 no podemos rechazar la hipotesis nula y por tanto no podemos afirmar nada.")
}
cat("\n\n")
}
}
}

normalityTest(dataset_regresion)


# ##############################################################################################
# ##############################################################################################
##### ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ #####
##### DATASET CLASIFICACION #####
##### ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ #####
# ##############################################################################################
# ##############################################################################################

dataset_clasificacion<-read.csv("/home/nacheteam/MEGA/Master/Introduccion a la ciencia de datos/Trabajo Integrador/DATOS/Datasets Clasificacion/heart/heart.dat", header=FALSE, comment.char = "@")
colnames(dataset_clasificacion)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
dataset_clasificacion

# Ahora vamos a ver el numero de variables y el tipo de cada una.
cat("El numero de variables es: ", length(colnames(dataset_clasificacion)), "\n")
cat("El tipo de las variables es:\n")
for(i in 1:length(colnames(dataset_clasificacion))){
cat("\t",colnames(dataset_clasificacion)[i], ": ", class(dataset_clasificacion[i][[1]]), "\n")
}

# Ahora vamos a calcular los estadisticos de las variables

cat("\n\n\n", "Estadisticos de las columnas:\n")
for(i in 1:length(colnames(dataset_clasificacion))){
cat("\t", colnames(dataset_clasificacion)[i], ":", "\n")
estadisticos<-calculaEstadisticos(dataset_clasificacion[,i])
cat("\t\t", "Media: ", estadisticos$media, "\n")
cat("\t\t", "Mediana: ", estadisticos$mediana, "\n")
cat("\t\t", "Desviacion tipica: ", estadisticos$stdv, "\n")
cat("\t\t", "Moda: ", estadisticos$moda, "\n")
cat("\t\t", "Kurtosis: ", estadisticos$kurtosis, "\n")
cat("\t\t", "Asimetria: ", estadisticos$asimetria, "\n")
cat("\t\t", "Minimo: ", estadisticos$minimo, "\n")
cat("\t\t", "Maximo: ", estadisticos$maximo, "\n")
}

##################################################################
## Estudio de correlacion de las variables ##
##################################################################

obtainCorrelated(1:(length(dataset_clasificacion)-1), dataset_clasificacion, threshold = -1)

# Viendo la salida parece logico que quitemos la variable 1 porque tiene una correlacion baja
corrSalida(1:(length(dataset_clasificacion)-1), dataset_clasificacion)

##################################################################
## Valores perdidos (Missing values) ##
##################################################################

# Vamos a ver si tenemos valores perdidos
which(is.na(dataset_clasificacion))

# No tenemos valores perdidos

##################################################################
## Outliers ##
##################################################################

pairs(dataset_clasificacion, pch=16, col="deepskyblue")

# Vamos a quitar las variables que no son muy interesantes
var<-c(-2,-3,-6,-7,-9,-11,-12,-13)
pairs(dataset_clasificacion[,var], pch=16, col="deepskyblue")
pairs(dataset_clasificacion[,c(-var, length(dataset_clasificacion))], pch=16, col="deepskyblue")

boxplot(dataset_clasificacion)
```

```
boxplot(dataset_clasificacion[,c(−1,−4,−5,−6,−8,−10)])

boxplot(dataset_clasificacion[,1]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,2]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,3]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,4]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,5]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,6]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,7]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,8]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,9]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,10]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,11]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,12]~dataset_clasificacion[,14], data=dataset_clasificacion)
boxplot(dataset_clasificacion[,13]~dataset_clasificacion[,14], data=dataset_clasificacion)


######################################################################
## Distribucion de las variables ##
######################################################################
library(plyr)
library(psych)
multi.hist(dataset_clasificacion)

normalityTest(dataset_clasificacion)
```

## 4.2. Regresión

```
library(foreign)
library(dplyr)
dataset_regresion<−read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_
    ↪ Regresion/treasury/treasury.dat")
dataset_regresion

colnames(dataset_regresion)
attach(dataset_regresion)

######################################################################
## Apartado 1 ##
######################################################################

# Vamos a hacer un modelo por variable
modelo.1Y_CMaturiyRate<−lm(`1MonthCDRate`~`1Y−CMaturityRate`)
modelo.30Y_CMortgageRate<−lm(`1MonthCDRate`~`30Y−CMortgageRate`)
modelo.3M_Rate_AuctionAverage<−lm(`1MonthCDRate`~`3M−Rate−AuctionAverage`)
modelo.3M_Rate_SecondaryMarket<−lm(`1MonthCDRate`~`3M−Rate−SecondaryMarket`)
modelo.3Y_CMaturityRate<−lm(`1MonthCDRate`~`3Y−CMaturityRate`)
modelo.5Y_CMaturityRate<−lm(`1MonthCDRate`~`5Y−CMaturityRate`)
modelo.bankCredit<−lm(`1MonthCDRate`~bankCredit)
modelo.currency<−lm(`1MonthCDRate`~currency)
modelo.demandDeposits<−lm(`1MonthCDRate`~demandDeposits)
modelo.federalFunds<−lm(`1MonthCDRate`~federalFunds)
modelo.moneyStock<−lm(`1MonthCDRate`~moneyStock)
modelo.checkableDeposits<−lm(`1MonthCDRate`~checkableDeposits)
modelo.loansLeases<−lm(`1MonthCDRate`~loansLeases)
modelo.savingsDeposits<−lm(`1MonthCDRate`~savingsDeposits)
modelo.tradeCurrencies<−lm(`1MonthCDRate`~tradeCurrencies)

# Pintamos y hacemos summary de cada modelo

# Modelo 1 0.2024 de R2 ajustado
plot(`1MonthCDRate`~`1Y−CMaturityRate`, dataset_regresion)
abline(modelo.1Y_CMaturiyRate, col="red")
summary(modelo.1Y_CMaturiyRate)

# Modelo 2 0.9551 de R2 ajustado
plot(`1MonthCDRate`~`30Y−CMortgageRate`, dataset_regresion)
abline(modelo.30Y_CMortgageRate, col="red")
summary(modelo.30Y_CMortgageRate)

# Modelo 3 0.7863 de R2 ajustado
plot(`1MonthCDRate`~`3M−Rate−AuctionAverage`, dataset_regresion)
abline(modelo.3M_Rate_AuctionAverage, col="red")
summary(modelo.3M_Rate_AuctionAverage)

# Modelo 4 0.9807 de R2 ajustado
plot(`1MonthCDRate`~`3M−Rate−SecondaryMarket`, dataset_regresion)
abline(modelo.3M_Rate_SecondaryMarket, col="red")
summary(modelo.3M_Rate_SecondaryMarket)

# Modelo 5 0.9835 de R2 ajustado
plot(`1MonthCDRate`~`3Y−CMaturityRate`, dataset_regresion)
abline(modelo.3Y_CMaturityRate, col="red")
summary(modelo.3Y_CMaturityRate)

# Modelo 6 0.8794 de R2 ajustado
plot(`1MonthCDRate`~`5Y−CMaturityRate`, dataset_regresion)
abline(modelo.5Y_CMaturityRate, col="red")
summary(modelo.5Y_CMaturityRate)

# Modelo 7 0.8291 de R2 ajustado
plot(`1MonthCDRate`~bankCredit, dataset_regresion)
abline(modelo.bankCredit, col="red")
summary(modelo.bankCredit)

# Modelo 8 0.4994 de R2 ajustado
```

```
plot(`1MonthCDRate`~currency, dataset_regresion)
abline(modelo.currency, col="red")
summary(modelo.currency)

# Modelo 9 0.4597 de R2 ajustado
plot(`1MonthCDRate`~demandDeposits, dataset_regresion)
abline(modelo.demandDeposits, col="red")
summary(modelo.demandDeposits)

# Modelo 10 0.4926 de R2 ajustado
plot(`1MonthCDRate`~federalFunds, dataset_regresion)
abline(modelo.federalFunds, col="red")
summary(modelo.federalFunds)

# Modelo 11 0.9893 de R2 ajustado
plot(`1MonthCDRate`~moneyStock, dataset_regresion)
abline(modelo.moneyStock, col="red")
summary(modelo.moneyStock)

# Modelo 12 0.6537 de R2 ajustado
plot(`1MonthCDRate`~checkableDeposits, dataset_regresion)
abline(modelo.checkableDeposits, col="red")
summary(modelo.checkableDeposits)

# Modelo 13 0.6854 de R2 ajustado
plot(`1MonthCDRate`~loansLeases, dataset_regresion)
abline(modelo.loansLeases, col="red")
summary(modelo.loansLeases)

# Modelo 14 0.4811 de R2 ajustado
plot(`1MonthCDRate`~savingsDeposits, dataset_regresion)
abline(modelo.savingsDeposits, col="red")
summary(modelo.savingsDeposits)

# Modelo 15 0.6287 de R2 ajustado
plot(`1MonthCDRate`~tradeCurrencies, dataset_regresion)
abline(modelo.tradeCurrencies, col="red")
summary(modelo.tradeCurrencies)

# Tabla de R2 ajustado
# Modelo1: 0.2024
# Modelo2: 0.9551
# Modelo3: 0.7863
# Modelo4: 0.9807
# Modelo5: 0.9835
# Modelo6: 0.8794
# Modelo7: 0.8291
# Modelo8: 0.4994
# Modelo9: 0.4597
# Modelo10: 0.4926
# Modelo11: 0.9893 ~
# Modelo12: 0.6537
# Modelo13: 0.6854
# Modelo14: 0.4811
# Modelo15: 0.6287

# Nos quedamos con los 5 mejores modelos.
modelo.moneyStock
modelo.3Y_CMaturityRate
modelo.3M_Rate_SecondaryMarket
modelo.30Y_CMortgageRate
modelo.5Y_CMaturityRate

########################################################################
## Apartado 2 ##
########################################################################

modelo1<-lm(`1MonthCDRate`~., dataset_regresion)
summary(modelo1)

# Quitamos federalFunds

modelo2<-lm(`1MonthCDRate`~.-federalFunds, dataset_regresion)
summary(modelo2)

# Quitamos 3M-Rate-AuctionAverage
modelo3<-lm(`1MonthCDRate`~.-federalFunds-`3M-Rate-AuctionAverage`, dataset_regresion)
summary(modelo3)

# Quitamos 3M-Rate-SecondaryMarket
modelo4<-lm(`1MonthCDRate`~.-federalFunds-`3M-Rate-AuctionAverage`-`3M-Rate-SecondaryMarket`, dataset_regresion)
summary(modelo4)

# Quitamos 5Y-CMaturiryRate
modelo5<-lm(`1MonthCDRate`~.-federalFunds-`3M-Rate-AuctionAverage`-`3M-Rate-SecondaryMarket`-`5Y-CMaturityRate`, dataset_
    ↪ regresion)
summary(modelo5)

# Vamos a probar a quitar demandDeposits
modelo6<-lm(`1MonthCDRate`~.-federalFunds-`3M-Rate-AuctionAverage`-`3M-Rate-SecondaryMarket`-`5Y-CMaturityRate`-
    ↪ demandDeposits, dataset_regresion)
summary(modelo6)

# Vamos a probar a quitar bankCredit
modelo7<-lm(`1MonthCDRate`~.-federalFunds-`3M-Rate-AuctionAverage`-`3M-Rate-SecondaryMarket`-`5Y-CMaturityRate`-
    ↪ demandDeposits-bankCredit, dataset_regresion)
summary(modelo7)

# Vamos a probar a quitar savingsDeposits
modelo8<-lm(`1MonthCDRate`~.-federalFunds-`3M-Rate-AuctionAverage`-`3M-Rate-SecondaryMarket`-`5Y-CMaturityRate`-
    ↪ demandDeposits-bankCredit-savingsDeposits, dataset_regresion)
summary(modelo8)
```

```
# Vamos a probar a quitar currency
modelo9<-lm(`1MonthCDRate`~.-federalFunds-`3M-Rate-AuctionAverage`-`3M-Rate-SecondaryMarket`-`5Y-CMaturityRate`-
    ↪ demandDeposits-bankCredit-savingsDeposits-currency, dataset_regresion)
summary(modelo9)

# Vamos a probar a quitar loansLeases
modelo10<-lm(`1MonthCDRate`~.-federalFunds-`3M-Rate-AuctionAverage`-`3M-Rate-SecondaryMarket`-`5Y-CMaturityRate`-
    ↪ demandDeposits-bankCredit-savingsDeposits-currency-loansLeases, dataset_regresion)
summary(modelo10)

# Vamos a probar a quitar checkableDeposits
modelo11<-lm(`1MonthCDRate`~.-federalFunds-`3M-Rate-AuctionAverage`-`3M-Rate-SecondaryMarket`-`5Y-CMaturityRate`-
    ↪ demandDeposits-bankCredit-savingsDeposits-currency-loansLeases-checkableDeposits, dataset_regresion)
summary(modelo11)

# Vamos a probar a quitar 30Y-CMortgageRate
modelo12<-lm(`1MonthCDRate`~.-federalFunds-`3M-Rate-AuctionAverage`-`3M-Rate-SecondaryMarket`-`5Y-CMaturityRate`-
    ↪ demandDeposits-bankCredit-savingsDeposits-currency-loansLeases-checkableDeposits-`30Y-CMortgageRate`, dataset_regresion)
summary(modelo12)

# Vamos a probar a quitar 1Y-CMaturityRate
modelo13<-lm(`1MonthCDRate`~.-federalFunds-`3M-Rate-AuctionAverage`-`3M-Rate-SecondaryMarket`-`5Y-CMaturityRate`-
    ↪ demandDeposits-bankCredit-savingsDeposits-currency-loansLeases-checkableDeposits-`30Y-CMortgageRate`-`1Y-CMaturityRate`,
    ↪ dataset_regresion)
summary(modelo13)

# Vamos a comprobar que resultados obtiene el modelo con las variables que decidimos quedarnos en el EDA
var<-c(7,10,11,13,14,15)
names(dataset_regresion)[var]
modelo14<-lm(`1MonthCDRate`~`1Y-CMaturityRate`+`3M-Rate-SecondaryMarket`+`3Y-CMaturityRate`+currency+federalFunds+
    ↪ savingsDeposits+tradeCurrencies, dataset_regresion)
summary(modelo14)

# Quitamos las que tienen poca significancia
modelo15<-lm(`1MonthCDRate`~`1Y-CMaturityRate`+`3M-Rate-SecondaryMarket`+`3Y-CMaturityRate`+currency+savingsDeposits, dataset_
    ↪ regresion)
summary(modelo15)

# Si quisieramos un modelo mas simple podemos quitar 3M-Rate-SecondaryMarket
modelo16<-lm(`1MonthCDRate`~`1Y-CMaturityRate`+`3Y-CMaturityRate`+currency+savingsDeposits, dataset_regresion)
summary(modelo16)

# Finalmente nos hemos quedado con 4.
# El modelo6 es el mejor y el modelo 9 es mas simple con resultados muy parecidos. Merece la pena el ultimo.

# Vamos a probar una dependencia cuadratica sobre estas variables aunque el resultado ya es lo suficientemente bueno.
modelo17<-lm(`1MonthCDRate`~`3Y-CMaturityRate`+moneyStock+tradeCurrencies+`3Y-CMaturityRate`*`3Y-CMaturityRate`+moneyStock*
    ↪ moneyStock+tradeCurrencies*tradeCurrencies+`3Y-CMaturityRate`*moneyStock+`3Y-CMaturityRate`*tradeCurrencies+moneyStock*
    ↪ tradeCurrencies, dataset_regresion)
summary(modelo17)
# No hay ninguna mejora

######################################################################
## Apartado 3 ##
######################################################################

# Funcion para leer los folds
readFolds<-function(){
fold1.train<-read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Regresion/
    ↪ treasury/treasury-5-1tra.dat")
fold1.test<-read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Regresion/
    ↪ treasury/treasury-5-1tst.dat")

fold2.train<-read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Regresion/
    ↪ treasury/treasury-5-2tra.dat")
fold2.test<-read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Regresion/
    ↪ treasury/treasury-5-2tst.dat")

fold3.train<-read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Regresion/
    ↪ treasury/treasury-5-3tra.dat")
fold3.test<-read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Regresion/
    ↪ treasury/treasury-5-3tst.dat")

fold4.train<-read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Regresion/
    ↪ treasury/treasury-5-4tra.dat")
fold4.test<-read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Regresion/
    ↪ treasury/treasury-5-4tst.dat")

fold5.train<-read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Regresion/
    ↪ treasury/treasury-5-5tra.dat")
fold5.test<-read.arff("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Regresion/
    ↪ treasury/treasury-5-5tst.dat")

folds<-list(fold1=list(train=fold1.train, test=fold1.test),
fold2=list(train=fold2.train, test=fold2.test),
fold3=list(train=fold3.train, test=fold3.test),
fold4=list(train=fold4.train, test=fold4.test),
fold5=list(train=fold5.train, test=fold5.test))

return(folds)
}

folds<-readFolds()

library(kknn)

# Funcion que ejecuta knn sobre test o train con el valor de k introducido y la formula pasada
ejecutaKNN<-function(formula, folds, k, tt="test"){
errores<-vector(mode = "numeric", length = 5)
for(i in 1:5){
if(tt=="test"){
modelo.knn<-kknn(formula, folds[[i]]$train, folds[[i]]$test, k=k)
yprime=modelo.knn$fitted.values
```

```
errores[i]<-sum(abs(folds[[i]]$test$`1MonthCDRate`-yprime)^2)/length(yprime)
}
else{
modelo.knn<-kknn(formula, folds[[i]]$train, folds[[i]]$train, k=k)
yprime=modelo.knn$fitted.values
errores[i]<-sum(abs(folds[[i]]$train$`1MonthCDRate`-yprime)^2)/length(yprime)
}
}
return(list(errores=errores, media.errores=mean(errores)))
}

# Todas las variables con k desde 3 hasta 55 a saltos de 2
resultados<-sapply(seq(3,55,2), ejecutaKNN, formula=`1MonthCDRate`~., folds=folds, tt="test")
plot(seq(3,55,2), resultados[2,], type="b", col="blue", lwd=7, xlab="Valores de K", ylab="Error en test", main = "Grafica de errores por valor
    ↪ de K para test")
resultados[,2]

resultados<-sapply(seq(3,55,2), ejecutaKNN, formula=`1MonthCDRate`~., folds=folds, tt="train")
plot(seq(3,55,2), resultados[2,], type="b", col="blue", lwd=7, xlab="Valores de K", ylab="Error en train", main = "Grafica de errores por valor
    ↪ de K para train")
resultados[,1]

# Ahora vamos a probar con las variables del mejor modelo de una variable

resultados<-sapply(seq(3,55,2), ejecutaKNN, formula=`1MonthCDRate`~moneyStock, folds=folds, tt="test")
plot(seq(3,55,2), resultados[2,], type="b", col="blue", lwd=7, xlab="Valores de K", ylab="Error en test", main = "Grafica de errores por valor
    ↪ de K para test")
resultados[,10]

resultados<-sapply(seq(3,55,2), ejecutaKNN, formula=`1MonthCDRate`~moneyStock, folds=folds, tt="train")
plot(seq(3,55,2), resultados[2,], type="b", col="blue", lwd=7, xlab="Valores de K", ylab="Error en train", main = "Grafica de errores por valor
    ↪ de K para train")
resultados[,1]

# Ahora vamos a probar con el mejor modelo lineal de 3 variables

resultados<-sapply(seq(3,55,2), ejecutaKNN, formula=`1MonthCDRate`~`3Y-CMaturityRate`+moneyStock+tradeCurrencies, folds=folds, tt="
    ↪ test")
plot(seq(3,55,2), resultados[2,], type="b", col="blue", lwd=7, xlab="Valores de K", ylab="Error en test", main = "Grafica de errores por valor
    ↪ de K para test")
resultados[,4]

resultados<-sapply(seq(3,55,2), ejecutaKNN, formula=`1MonthCDRate`~`3Y-CMaturityRate`+moneyStock+tradeCurrencies, folds=folds, tt="
    ↪ train")
plot(seq(3,55,2), resultados[2,], type="b", col="blue", lwd=7, xlab="Valores de K", ylab="Error en train", main = "Grafica de errores por valor
    ↪ de K para train")
resultados[,1]

##########################################################################
## Apartado 4 ##
##########################################################################

# Funcion que ejecuta LM y obtiene los errores
ejecutaLM<-function(formula, folds, tt="test"){
errores<-vector(mode = "numeric", length = 5)
for(i in 1:5){
if(tt=="test"){
modelo<-lm(formula, folds[[i]]$train)
yprime=predict(modelo,folds[[i]]$test)
errores[i]<-sum(abs(folds[[i]]$test$`1MonthCDRate`-yprime)^2)/length(yprime)
}
else{
modelo<-lm(formula, folds[[i]]$train)
yprime=predict(modelo,folds[[i]]$train)
errores[i]<-sum(abs(folds[[i]]$train$`1MonthCDRate`-yprime)^2)/length(yprime)
}
}
return(list(errores=errores, media.errores=mean(errores)))
}

ejecutaLM(`1MonthCDRate`~moneyStock+tradeCurrencies+`3Y-CMaturityRate`, folds, tt="test")
ejecutaLM(`1MonthCDRate`~moneyStock+tradeCurrencies+`3Y-CMaturityRate`, folds, tt="train")

ejecutaKNN(formula = `1MonthCDRate`~. ,folds = folds, k=5, tt="test")
ejecutaKNN(formula = `1MonthCDRate`~. ,folds = folds, k=5, tt="train")

ejecutaLM(`1MonthCDRate`~., folds, tt="test")
ejecutaLM(`1MonthCDRate`~., folds, tt="train")

# Leemos los resultados de test
resultadoststst <- read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Tablas_de_
    ↪ resultados_para_tests_estadisticos/regr_test_alumnos.csv")
tablatst <- cbind(resultadoststst[,2:dim(resultadoststst)[2]])
colnames(tablatst) <- names(resultadoststst)[2:dim(resultadoststst)[2]]
rownames(tablatst) <- resultadoststst[,1]

# Leemos los resultados de train
resultadostra <- read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Tablas_de_
    ↪ resultados_para_tests_estadisticos/regr_train_alumnos.csv")
tablatra <- cbind(resultadostra[,2:dim(resultadostra)[2]])
colnames(tablatra) <- names(resultadostra)[2:dim(resultadostra)[2]]
rownames(tablatra) <- resultadostra[,1]

# Calculamos las diferencias para test
difs <- (tablatst[,1] - tablatst[,2]) / tablatst[,1]
wilc_1_2 <- cbind(ifelse (difs<0, abs(difs)+0.1, 0+0.1), ifelse (difs>0, abs(difs)+0.1, 0+0.1))
colnames(wilc_1_2) <- c(colnames(tablatst)[1], colnames(tablatst)[2])
head(wilc_1_2)

# Hacemos el test
LMvsKNNtst <- wilcox.test(wilc_1_2[,1], wilc_1_2[,2], alternative = "two.sided", paired=TRUE)
Rmas <- LMvsKNNtst$statistic
pvalue <- LMvsKNNtst$p.value
```

```
LMvsKNNtst <- wilcox.test(wilc_1_2[,2], wilc_1_2[,1], alternative = "two.sided", paired=TRUE)
Rmenos <- LMvsKNNtst$statistic
Rmas
Rmenos
pvalue

# Calculamos las diferencias para train
difs <- (tablatra[,1] - tablatra[,2]) / tablatra[,1]
wilc_1_2 <- cbind(ifelse (difs<0, abs(difs)+0.1, 0+0.1), ifelse (difs>0, abs(difs)+0.1, 0+0.1))
colnames(wilc_1_2) <- c(colnames(tablatra)[1], colnames(tablatra)[2])
head(wilc_1_2)

# Hacemos el test
LMvsKNNtra <- wilcox.test(wilc_1_2[,1], wilc_1_2[,2], alternative = "two.sided", paired=TRUE)
Rmas <- LMvsKNNtra$statistic
pvalue <- LMvsKNNtra$p.value
LMvsKNNtra <- wilcox.test(wilc_1_2[,2], wilc_1_2[,1], alternative = "two.sided", paired=TRUE)
Rmenos <- LMvsKNNtra$statistic
Rmas
Rmenos
pvalue

# Test de friedman
friedman.test(as.matrix(tablatra))
friedman.test(as.matrix(tablatst))

# Test post-hoc Holm
tam <-dim(tablatra)
groups <-rep(1:tam[2], each=tam[1])
pairwise.wilcox.test(as.matrix(tablatra), groups, p.adjust= "holm", paired = T)

tam <-dim(tablatst)
groups <-rep(1:tam[2], each=tam[1])
pairwise.wilcox.test(as.matrix(tablatst), groups, p.adjust= "holm", paired = T)
```

## 4.3. Clasificación

```
library(foreign)
library(dplyr)
dataset_clasificacion<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_
    ↪ Clasificacion/heart/heart.dat", header=FALSE, comment.char = "@")
colnames(dataset_clasificacion)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
dataset_clasificacion

colnames(dataset_clasificacion)
attach(dataset_clasificacion)

library(caret)
library(class)

library(MASS)
library(ISLR)

#########################################################################
## Apartado 1 ##
#########################################################################

# Leemos los folds
readFolds<-function(){
fold1.train<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion
    ↪ /heart/heart-10-1tra.dat", header=FALSE, comment.char = "@")
fold1.test<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion/
    ↪ heart/heart-10-1tst.dat", header=FALSE, comment.char = "@")
colnames(fold1.train)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
colnames(fold1.test)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")

fold2.train<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion
    ↪ /heart/heart-10-2tra.dat", header=FALSE, comment.char = "@")
fold2.test<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion/
    ↪ heart/heart-10-2tst.dat", header=FALSE, comment.char = "@")
colnames(fold2.train)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
colnames(fold2.test)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")

fold3.train<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion
    ↪ /heart/heart-10-3tra.dat", header=FALSE, comment.char = "@")
fold3.test<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion/
    ↪ heart/heart-10-3tst.dat", header=FALSE, comment.char = "@")
colnames(fold3.train)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
colnames(fold3.test)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")

fold4.train<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion
    ↪ /heart/heart-10-4tra.dat", header=FALSE, comment.char = "@")
fold4.test<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion/
    ↪ heart/heart-10-4tst.dat", header=FALSE, comment.char = "@")
colnames(fold4.train)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
colnames(fold4.test)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
```

```
fold5.train<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion
    ↪ /heart/heart-10-5tra.dat", header=FALSE, comment.char = "@")
fold5.test<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion/
    ↪ heart/heart-10-5tst.dat", header=FALSE, comment.char = "@")
colnames(fold5.train)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
colnames(fold5.test)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")

fold6.train<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion
    ↪ /heart/heart-10-6tra.dat", header=FALSE, comment.char = "@")
fold6.test<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion/
    ↪ heart/heart-10-6tst.dat", header=FALSE, comment.char = "@")
colnames(fold6.train)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
colnames(fold6.test)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")

fold7.train<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion
    ↪ /heart/heart-10-7tra.dat", header=FALSE, comment.char = "@")
fold7.test<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion/
    ↪ heart/heart-10-7tst.dat", header=FALSE, comment.char = "@")
colnames(fold7.train)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
colnames(fold7.test)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")

fold8.train<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion
    ↪ /heart/heart-10-8tra.dat", header=FALSE, comment.char = "@")
fold8.test<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion/
    ↪ heart/heart-10-8tst.dat", header=FALSE, comment.char = "@")
colnames(fold8.train)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
colnames(fold8.test)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")

fold9.train<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion
    ↪ /heart/heart-10-9tra.dat", header=FALSE, comment.char = "@")
fold9.test<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion/
    ↪ heart/heart-10-9tst.dat", header=FALSE, comment.char = "@")
colnames(fold9.train)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
colnames(fold9.test)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")

fold10.train<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_
    ↪ Clasificacion/heart/heart-10-10tra.dat", header=FALSE, comment.char = "@")
fold10.test<-read.csv("/home/nacheteam/MEGA/Master/Introduccion_a_la_ciencia_de_datos/Trabajo_Integrador/DATOS/Datasets_Clasificacion
    ↪ /heart/heart-10-10tst.dat", header=FALSE, comment.char = "@")
colnames(fold10.train)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")
colnames(fold10.test)<-c("Age", "Sex", "ChestPainType", "RestBloodPressure", "SerumCholestoral", "FastingBloodSugar", "
    ↪ ResElectrocardiographic", "MaxHeartRate", "ExerciseInduced", "Oldpeak", "Slope", "MajorVessels", "Thal", "Class")

folds<-list(fold1=list(train=fold1.train, test=fold1.test),
fold2=list(train=fold2.train, test=fold2.test),
fold3=list(train=fold3.train, test=fold3.test),
fold4=list(train=fold4.train, test=fold4.test),
fold5=list(train=fold5.train, test=fold5.test),
fold6=list(train=fold6.train, test=fold6.test),
fold7=list(train=fold7.train, test=fold7.test),
fold8=list(train=fold8.train, test=fold8.test),
fold9=list(train=fold9.train, test=fold9.test),
fold10=list(train=fold10.train, test=fold10.test))

return(folds)
}

folds<-readFolds()

# Funcion que ejecuta KNN sobre test o train con el valor de k introducido
ejecutaKNN<-function(k, type="test"){
cat("Valor de k: ", k, ", tipo: ", type)
acc<-0
aciertos<-vector(mode="numeric", length = 10)
for(i in 1:10){
test<-folds[[i]]$test
train<-folds[[i]]$train
if(type=="test"){
pred <- knn(train = train[,-length(train)], test = test[,-length(test)], cl = train[,length(train)], k=k)

cat("\nFold ", i, ": ", mean(pred==test[,length(test)])*100, " %")
aciertos[i]<-mean(pred==test[,length(test)])*100
acc<-acc+mean(pred==test[,length(test)])*100
#print(table(pred,test[,length(test)]))
}
else{
pred <- knn(train = train[,-length(train)], test = train[,-length(train)], cl = train[,length(train)], k=k)

cat("\nFold ", i, ": ", mean(pred==train[,length(train)])*100, " %")
aciertos[i]<-mean(pred==train[,length(train)])*100
acc<-acc+mean(pred==train[,length(train)])*100
#print(table(pred,test[,length(test)]))
}
}
acc<-acc/10
cat("\nAccuracy medio: ", acc)
return(list(aciertos=aciertos, acierto.medio=acc, desviacion=sd(aciertos)))
}

aciertos.knn.test<-sapply(seq(3,55,2), ejecutaKNN, type="test")
plot(seq(3,55,2), aciertos.knn.test[2,], type="b", col="blue", lwd=7, xlab="Valores de K", ylab="Porcentaje de acierto", main = "Acierto vs K 
    ↪ para test")
```

```
aciertos.knn.test[,9]

aciertos.knn.train<-sapply(seq(3,55,2), ejecutaKNN, type="train")
plot(seq(3,55,2), aciertos.knn.train[2,], type="b", col="blue", lwd=7, xlab="Valores de K", ylab="Porcentaje de acierto", main = "Acierto vs K 
    ↪ para train")
aciertos.knn.train[,1]
aciertos.knn.train[,9]

# Utilizamos TSNE para reducir los datos y pintarlos por clases
library(Rtsne)
reduced<-data.frame(Rtsne(dataset_clasificacion[,-1], dims=2))

p1 <- ggplot(as.data.frame(reduced$Y), aes(x=V1, y=V2)) + geom_point(aes(col=factor(dataset_clasificacion$Class))) + theme(axis.title.y =
    ↪ element_blank())
p1

ejecutaKNN(k=19, type="test")
ejecutaKNN(k=19, type="train")

##########################################################################
## Apartado 2 ##
##########################################################################

# Vamos a comprobar si nuestras variables cumplen la precondicion de que son normales
# Sabemos ya por el EDA que no

# Todas las variables.

# Prediciendo sobre el test
mean_score<-0
aciertos.lda.test<-vector(mode = "numeric", length = 10)
for(i in 1:10){
test<-folds[[i]]$test
train<-folds[[i]]$train
model<-lda(Class~.,data=train)

lda.pred <- predict(model,test)
cat("\nFold ", i, ":")
print(table(lda.pred$class,test$Class))
print(mean(lda.pred$class==test$Class))
aciertos.lda.test[i]<-mean(lda.pred$class==test$Class)
mean_score<-mean_score+mean(lda.pred$class==test$Class)
}
mean_score<-mean_score/10
cat("\nDesviacion tipica: ", sd(aciertos.lda.test)*100)
cat("\nScore final: ", mean_score)

# Prediciendo sobre el train
mean_score<-0
aciertos.lda.train<-vector(mode="numeric", length = 10)
for(i in 1:10){
test<-folds[[i]]$test
train<-folds[[i]]$train
model<-lda(Class~.,data=train)

lda.pred <- predict(model,train)
cat("\nFold ", i, ":")
print(table(lda.pred$class,train$Class))
print(mean(lda.pred$class==train$Class))
aciertos.lda.train[i]<-mean(lda.pred$class==train$Class)
mean_score<-mean_score+mean(lda.pred$class==train$Class)
}
mean_score<-mean_score/10
cat("\nDesviacion tipica: ", sd(aciertos.lda.train)*100)
cat("\nScore final: ", mean_score)

##########################################################################
## Apartado 3 ##
##########################################################################

# Prediciendo sobre el test
mean_score<-0
aciertos.qda.test<-vector(mode="numeric", length = 10)
for(i in 1:10){
test<-folds[[i]]$test
train<-folds[[i]]$train
model<-qda(Class~.,data=train)

lda.pred <- predict(model,test)
cat("\nFold ", i, ":")
print(table(lda.pred$class,test$Class))
print(mean(lda.pred$class==test$Class))
aciertos.qda.test[i]<-mean(lda.pred$class==test$Class)
mean_score<-mean_score+mean(lda.pred$class==test$Class)
}
mean_score<-mean_score/10
cat("\nDesviacion tipica: ", sd(aciertos.qda.test)*100)
cat("\nScore final: ", mean_score)

# Prediciendo sobre el train
mean_score<-0
aciertos.qda.train<-vector(mode="numeric", length = 10)
for(i in 1:10){
test<-folds[[i]]$test
train<-folds[[i]]$train
model<-qda(Class~.,data=train)

lda.pred <- predict(model,train)
cat("\nFold ", i, ":")
print(table(lda.pred$class,train$Class))
print(mean(lda.pred$class==train$Class))
aciertos.qda.train[i]<-mean(lda.pred$class==train$Class)
mean_score<-mean_score+mean(lda.pred$class==train$Class)
```

```r
}
mean_score<-mean_score/10
cat("\nDesviacion_tipica: ", sd(aciertos.qda.train)*100)
cat("\nScore_final: ", mean_score)

##########################################################################
## Apartado 4 ##
##########################################################################

computeWilcoxon<-function(model1, model2){
# Leemos los resultados de test
resultadostst <- data.frame(list(knn=aciertos.knn.test[,9]$aciertos, lda=aciertos.lda.test, qda=aciertos.qda.test))
tablatst <- cbind(resultadostst[,1:dim(resultadostst)[2]])
colnames(tablatst) <- names(resultadostst)[1:dim(resultadostst)[2]]

# Leemos los resultados de train
resultadostra <- data.frame(list(knn=aciertos.knn.train[,9]$aciertos, lda=aciertos.lda.train, qda=aciertos.qda.train))
tablatra <- cbind(resultadostra[,1:dim(resultadostra)[2]])
colnames(tablatra) <- names(resultadostra)[1:dim(resultadostra)[2]]

# Calculamos las diferencias para test
difs <- (tablatst[,model1] - tablatst[,model2]) / tablatst[,model1]
wilc_1_2 <- cbind(ifelse (difs<0, abs(difs)+0.1, 0+0.1), ifelse (difs>0, abs(difs)+0.1, 0+0.1))
colnames(wilc_1_2) <- c(colnames(tablatst)[model1], colnames(tablatst)[model2])
head(wilc_1_2)

# Hacemos el test
LMvsKNNtst <- wilcox.test(wilc_1_2[,1], wilc_1_2[,2], alternative = "two.sided", paired=TRUE)
Rmas.test <- LMvsKNNtst$statistic
pvalue.test <- LMvsKNNtst$p.value
LMvsKNNtst <- wilcox.test(wilc_1_2[,2], wilc_1_2[,1], alternative = "two.sided", paired=TRUE)
Rmenos.test <- LMvsKNNtst$statistic

# Calculamos las diferencias para train
difs <- (tablatra[,model1] - tablatra[,model2]) / tablatra[,model1]
wilc_1_2 <- cbind(ifelse (difs<0, abs(difs)+0.1, 0+0.1), ifelse (difs>0, abs(difs)+0.1, 0+0.1))
colnames(wilc_1_2) <- c(colnames(tablatra)[model1], colnames(tablatra)[model2])
head(wilc_1_2)

# Hacemos el test
LMvsKNNtra <- wilcox.test(wilc_1_2[,1], wilc_1_2[,2], alternative = "two.sided", paired=TRUE)
Rmas.train <- LMvsKNNtra$statistic
pvalue.train <- LMvsKNNtra$p.value
LMvsKNNtra <- wilcox.test(wilc_1_2[,2], wilc_1_2[,1], alternative = "two.sided", paired=TRUE)
Rmenos.train <- LMvsKNNtra$statistic
return(list(test=list(Rmas=Rmas.test, Rmenos=Rmenos.test, pvalue=pvalue.test), train=list(Rmas=Rmas.train, Rmenos=Rmenos.train, pvalue=
    ↪ pvalue.train)))
}

computeWilcoxon(1,2)
computeWilcoxon(2,3)
computeWilcoxon(1,3)

# Test de friedman
friedman.test(as.matrix(tablatra))
friedman.test(as.matrix(tablatst))

# Test post-hoc Holm
tam <-dim(tablatra)
groups <-rep(1:tam[2], each=tam[1])
pairwise.wilcox.test(as.matrix(tablatra), groups, p.adjust= "holm", paired = T)

tam <-dim(tablatst)
groups <-rep(1:tam[2], each=tam[1])
pairwise.wilcox.test(as.matrix(tablatst), groups, p.adjust= "holm", paired = T)
```