Práctica 2: Limpieza y análisis de datos

Table of Contents

```
# getwd()
```

Presentación

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas. Para hacer esta práctica tendréis que trabajar en grupos de 2 personas. Tendréis que entregar un solo archivo con el enlace Github donde se encuentren las soluciones incluyendo los nombres de los componentes del equipo. Podéis utilizar la Wiki de Github para describir vuestro equipo y los diferentes archivos que corresponden a vuestra entrega. Cada miembro del equipo tendrá que contribuir con su usuario Github. Aunque no se trata del mismo enunciado, los siguientes ejemplos de ediciones anteriores os pueden servir como guía:

● Ejemplo: https://github.com/Bengis/nba-gap-cleaning
● Ejemplo complejo (archivo adjunto).

Competencias

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science:

● Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.

● Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

Objetivos

Los objetivos concretos de esta práctica son:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.

- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

Descripción de la Práctica a realizar

El objetivo de esta actividad será el tratamiento de un dataset, que puede ser el creado en la práctica 1 o bien cualquier dataset libre disponible en Kaggle. Algunos ejemplos de dataset con los que podéis trabajar son:

- Red Wine Quality
- Titanic: Machine Learning from Disaster

El último ejemplo corresponde a una competición activa de Kaggle de manera que, opcionalmente, podéis aprovechar el trabajo realizado durante la práctica para entrar en esta competición. Siguiendo las principales etapas de un proyecto analítico, las diferentes tareas a realizar (y justificar) son las siguientes:

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

El dataset elegido se llama ‘Credit Card customers’ y puede descargarse desde la plataforma Kaggle. Su importancia radica en que trata de resolver un frecuente problema de negocios. Este conjunto de datos contiene la información de una agencia bancaria, donde el administrador desea conocer el o los motivos por los cuales están perdiendo clientes, así como prevenir el así llamado ‘Credit Card Churning’.

Habiendo identificado los clientes que potencialmente desean prescindir de algún servicio bancario, la agencia podría anticipar estrategias para lograr que dichos clientes se queden por más tiempo.

Este dataset contiene información sobre 10,127 clientes. Entre los datos descriptores se encuentran 23 variables que son:

0 CLIENTNUM: de tipo entero y sin valores nulos
1 Attrition_Flag: de tipo entero y sin valores nulos
2 Customer_Age: de tipo entero y sin valores nulos
3 Gender: de tipo objeto y sin valores nulos
4 Dependent_count: de tipo entero y sin valores nulos
5 Education_Level: de tipo objeto y sin valores nulos
6 Marital_Status: de tipo objeto y sin valores nulos
7 Income_Category: de tipo objeto y sin valores nulos
8 Card_Category: de tipo objeto y sin valores nulos
9 Months_on_book: de tipo entero y sin valores nulos
10 Total_Relationship_Count: de tipo entero y sin valores nulos
11 Months_Inactive_12_mon: de tipo entero y sin valores nulos
12 Contacts_Count_12_mon: de tipo entero y sin valores nulos
13 Credit_Limit: de tipo flotante y sin valores nulos
14 Total_Revolving_Bal: de tipo entero y sin valores nulos
15 Avg_Open_To_Buy: de tipo flotante y sin valores nulos
16 Total_Amt_Chng_Q4_Q1: de tipo flotante y sin valores nulos
17 Total_Trans_Amt: de tipo entero y sin valores nulos
18 Total_Trans_Ct: de tipo entero y sin valores nulos
19 Total_Ct_Chng_Q4_Q1: de tipo flotante y sin valores nulos
20 Avg_Utilization_Ratio: de tipo flotante y sin valores nulos
21 Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_1: de tipo objeto y sin valores nulos
22 Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dependent_count_Education_Level_Months_Inactive_12_mon_2: de tipo flotante y sin valores nulos

Este dataset se encuentra completamente en el idioma inglés y se trabajará como tal sin hacer traducciones de variables o información, sin embargo se describen en la tabla siguiente aquellas que se tomarán en cuenta para los análisis posteriores:

Variable	Descripción

“CLIENTNUM”	Numero de cliente: Identificador único del cliente titular de la cuenta
“Customer_Age”	Variable demográfica: edad del cliente en años
“Dependent_count”	Variable demográfica: número de dependientes
“Marital_Status”	Variable demográfica: Casado, Soltero, Divorciado, Desconocido
“Card_Category”	Variable de producto: tipo de tarjeta (azul, plata, oro, platino)
“Total_Relationship_Count”	Número total de productos en poder del cliente
“Months_Inactive_12_mon”	Número de meses inactivos en los últimos 12 meses
“Contacts_Count_12_mon”	Número de contactos en los últimos 12 meses
“Total_Revolving_Bal”	Saldo rotatorio total en la tarjeta de crédito
“Total_Amt_Chng_Q4_Q1”	Cambio en el monto de la transacción (Q4 sobre Q1)
“Total_Trans_Ct”	Recuento total de transacciones (últimos 12 meses)
“Avg_Utilization_Ratio”	Índice de utilización promedio de la tarjeta
“Attrition_Flag”	Variable de evento interno (actividad del cliente): si la cuenta está cerrada, entonces 1 más 0
“Gender”	Variable demográfica: M = Hombre, F = Mujer
“Education_Level”	Variable demográfica: calificación educativa del titular de la cuenta (ejemplo: escuela secundaria, graduado universitario, etc.)
“Income_Category”	Variable demográfica: categoría de ingresos anuales del titular de la cuenta (<$ 40K, $ 40K - 60K, $ 60K - $ 80K, $ 80K- $ 120K,> $ 120K, Desconocido)
“Months_on_book”	Periodo de relación con el banco
“Credit_Limit”	Límite de crédito en la tarjeta de crédito

“Avg_Open_To_Buy”	Línea de crédito abierta para comprar (promedio de los últimos 12 meses)
“Total_Trans_Amt”	Monto total de la transacción (últimos 12 meses)
“Total_Ct_Chng_Q4_Q1”	Cambio en el recuento de transacciones (Q4 sobre Q1)

Las modificaciones realizadas (preprocesamiento) se muestran a continuación.

Integración y selección de los datos de interés a analizar

Carga del archivo

```
#getwd()
```

Se procede a abrir el archivo formato .csv y examinar el tipo de datos con los que R ha interpretado cada variable.

```
df <- data.frame(read.csv("BankChurners.csv", header=TRUE))
attach(df)
head(df)
```

```
##   CLIENTNUM    Attrition_Flag Customer_Age Gender Dependent_count
## 1 768805383 Existing Customer           45      M               3
## 2 818770008 Existing Customer           49      F               5
## 3 713982108 Existing Customer           51      M               3
## 4 769911858 Existing Customer           40      F               4
## 5 709106358 Existing Customer           40      M               3
## 6 713061558 Existing Customer           44      M               2
##   Education_Level Marital_Status Income_Category Card_Category
Months_on_book
## 1     High School        Married     $60K - $80K          Blue
39
## 2        Graduate         Single  Less than $40K          Blue
44
## 3        Graduate        Married    $80K - $120K          Blue
36
## 4     High School        Unknown  Less than $40K          Blue
34
## 5      Uneducated        Married     $60K - $80K          Blue
21
## 6        Graduate        Married     $40K - $60K          Blue
36
##   Total_Relationship_Count Months_Inactive_12_mon Contacts_Count_12_mon
```

```
## 1                         5                       1                       3
## 2                         6                       1                       2
## 3                         4                       1                       0
## 4                         3                       4                       1
## 5                         5                       1                       0
## 6                         3                       1                       2
##   Credit_Limit Total_Revolving_Bal Avg_Open_To_Buy Total_Amt_Chng_Q4_Q1
## 1        12691                 777           11914                 1.335
## 2         8256                 864            7392                 1.541
## 3         3418                   0            3418                 2.594
## 4         3313                2517             796                 1.405
## 5         4716                   0            4716                 2.175
## 6         4010                1247            2763                 1.376
##   Total_Trans_Amt Total_Trans_Ct Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio
## 1            1144             42                1.625                 61.00
## 2            1291             33                3.714                105.00
## 3            1887             20                2.333                  0.00
## 4            1171             20                2.333                  0.76
## 5             816             28                2.500                  0.00
## 6            1088             24              846.000                311.00
##
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dep
endent_count_Education_Level_Months_Inactive_12_mon_1
## 1
9,34E-01
## 2
5,69E-01
## 3
2,11E-01
## 4
13.366
## 5
2,17E-01
## 6
5,51E-01
##
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dep
endent_count_Education_Level_Months_Inactive_12_mon_2
## 1
99.991
## 2
99.994
## 3
99.998
## 4
99.987
## 5
99.998
## 6
99.994
```

Análisis del archivo

A continuación se muestra un resumen del contenido del dataframe:

```
summary(df)
```

```
##    CLIENTNUM         Attrition_Flag      Customer_Age      Gender
##  Min.   :708082083   Length:10127       Min.   :26.00   Length:10127
##  1st Qu.:713036770   Class :character   1st Qu.:41.00   Class :character
##  Median :717926358   Mode  :character   Median :46.00   Mode  :character
##  Mean   :739177606                      Mean   :46.33
##  3rd Qu.:773143533                      3rd Qu.:52.00
##  Max.   :828343083                      Max.   :73.00
##  Dependent_count Education_Level    Marital_Status     Income_Category
##  Min.   :0.000   Length:10127       Length:10127       Length:10127
##  1st Qu.:1.000   Class :character   Class :character   Class :character
##  Median :2.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2.346
##  3rd Qu.:3.000
##  Max.   :5.000
##  Card_Category      Months_on_book  Total_Relationship_Count
##  Length:10127       Min.   :13.00   Min.   :1.000
##  Class :character   1st Qu.:31.00   1st Qu.:3.000
##  Mode  :character   Median :36.00   Median :4.000
##                     Mean   :35.93   Mean   :3.813
##                     3rd Qu.:40.00   3rd Qu.:5.000
##                     Max.   :56.00   Max.   :6.000
##  Months_Inactive_12_mon Contacts_Count_12_mon  Credit_Limit
##  Min.   :0.000          Min.   :0.000         Min.   : 1438
##  1st Qu.:2.000          1st Qu.:2.000         1st Qu.: 2555
##  Median :2.000          Median :2.000         Median : 4549
##  Mean   :2.341          Mean   :2.455         Mean   : 8632
##  3rd Qu.:3.000          3rd Qu.:3.000         3rd Qu.:11068
##  Max.   :6.000          Max.   :6.000         Max.   :34516
##  Total_Revolving_Bal Avg_Open_To_Buy Total_Amt_Chng_Q4_Q1 Total_Trans_Amt
##  Min.   :   0        Min.   :    3   Min.   :  0.0        Min.   :  510
##  1st Qu.: 359        1st Qu.: 1324   1st Qu.:484.0        1st Qu.: 2156
##  Median :1276        Median : 3474   Median :674.0        Median : 3899
##  Mean   :1163        Mean   : 7469   Mean   :574.7        Mean   : 4404
##  3rd Qu.:1784        3rd Qu.: 9859   3rd Qu.:792.0        3rd Qu.: 4741
##  Max.   :2517        Max.   :34516   Max.   :999.0        Max.   :18484
##  Total_Trans_Ct   Total_Ct_Chng_Q4_Q1 Avg_Utilization_Ratio
##  Min.   : 10.00   Min.   :  0.0       Min.   :  0.00
##  1st Qu.: 45.00   1st Qu.:316.0       1st Qu.:  0.06
##  Median : 67.00   Median :625.0       Median :132.00
##  Mean   : 64.86   Mean   :518.5       Mean   :247.33
##  3rd Qu.: 81.00   3rd Qu.:761.0       3rd Qu.:463.00
##  Max.   :139.00   Max.   :981.0       Max.   :999.00
##
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dep
```

```
endent_count_Education_Level_Months_Inactive_12_mon_1
##  Length:10127
##  Class :character
##  Mode  :character
##
##
##
##
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dep
endent_count_Education_Level_Months_Inactive_12_mon_2
##  Min.   :  0.01
##  1st Qu.: 99.97
##  Median : 99.98
##  Mean   :132.85
##  3rd Qu.: 99.99
##  Max.   :997.00
```

Se revisa la estructura del dataframe:

```
str(df)
```

```
## 'data.frame':    10127 obs. of  23 variables:
##  $ CLIENTNUM
: int  768805383 818770008 713982108 769911858 709106358 713061558 810347208
818906208 710930508 719661558 ...
##  $ Attrition_Flag
: chr  "Existing Customer" "Existing Customer" "Existing Customer" "Existing
Customer" ...
##  $ Customer_Age
: int  45 49 51 40 40 44 51 32 37 48 ...
##  $ Gender
: chr  "M" "F" "M" "F" ...
##  $ Dependent_count
: int  3 5 3 4 3 2 4 0 3 2 ...
##  $ Education_Level
: chr  "High School" "Graduate" "Graduate" "High School" ...
##  $ Marital_Status
: chr  "Married" "Single" "Married" "Unknown" ...
##  $ Income_Category
: chr  "$60K - $80K" "Less than $40K" "$80K - $120K" "Less than $40K" ...
##  $ Card_Category
: chr  "Blue" "Blue" "Blue" "Blue" ...
##  $ Months_on_book
: int  39 44 36 34 21 36 46 27 36 36 ...
##  $ Total_Relationship_Count
: int  5 6 4 3 5 3 6 2 5 6 ...
##  $ Months_Inactive_12_mon
: int  1 1 1 4 1 1 1 2 2 3 ...
##  $ Contacts_Count_12_mon
: int  3 2 0 1 0 2 3 2 0 3 ...
```

```
##  $ Credit_Limit
: num  12691 8256 3418 3313 4716 ...
##  $ Total_Revolving_Bal
: int  777 864 0 2517 0 1247 2264 1396 2517 1677 ...
##  $ Avg_Open_To_Buy
: num  11914 7392 3418 796 4716 ...
##  $ Total_Amt_Chng_Q4_Q1
: num  1.33 1.54 2.59 1.4 2.17 ...
##  $ Total_Trans_Amt
: int  1144 1291 1887 1171 816 1088 1330 1538 1350 1441 ...
##  $ Total_Trans_Ct
: int  42 33 20 20 28 24 31 36 24 32 ...
##  $ Total_Ct_Chng_Q4_Q1
: num  1.62 3.71 2.33 2.33 2.5 ...
##  $ Avg_Utilization_Ratio
: num  61 105 0 0.76 0 311 66 48 113 144 ...
##  $
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dep
endent_count_Education_Level_Months_Inactive_12_mon_1: chr  "9,34E-01"
"5,69E-01" "2,11E-01" "13.366" ...
##  $
Naive_Bayes_Classifier_Attrition_Flag_Card_Category_Contacts_Count_12_mon_Dep
endent_count_Education_Level_Months_Inactive_12_mon_2: num  100 100 100 100
100 ...
```

Selección de los datos de interés

Para el presente proyecto se utilizarán todas las variables presentes en el juego de datos a excepción de las dos últimas puesto que no se aplicará el análisis Naive Bayes.

```
df <- df[,-(22:23)]
names(df)
```

```
##  [1] "CLIENTNUM"                "Attrition_Flag"
##  [3] "Customer_Age"             "Gender"
##  [5] "Dependent_count"          "Education_Level"
##  [7] "Marital_Status"           "Income_Category"
##  [9] "Card_Category"            "Months_on_book"
## [11] "Total_Relationship_Count" "Months_Inactive_12_mon"
## [13] "Contacts_Count_12_mon"    "Credit_Limit"
## [15] "Total_Revolving_Bal"      "Avg_Open_To_Buy"
## [17] "Total_Amt_Chng_Q4_Q1"     "Total_Trans_Amt"
## [19] "Total_Trans_Ct"           "Total_Ct_Chng_Q4_Q1"
## [21] "Avg_Utilization_Ratio"
```

Análisis gráfico descriptivo

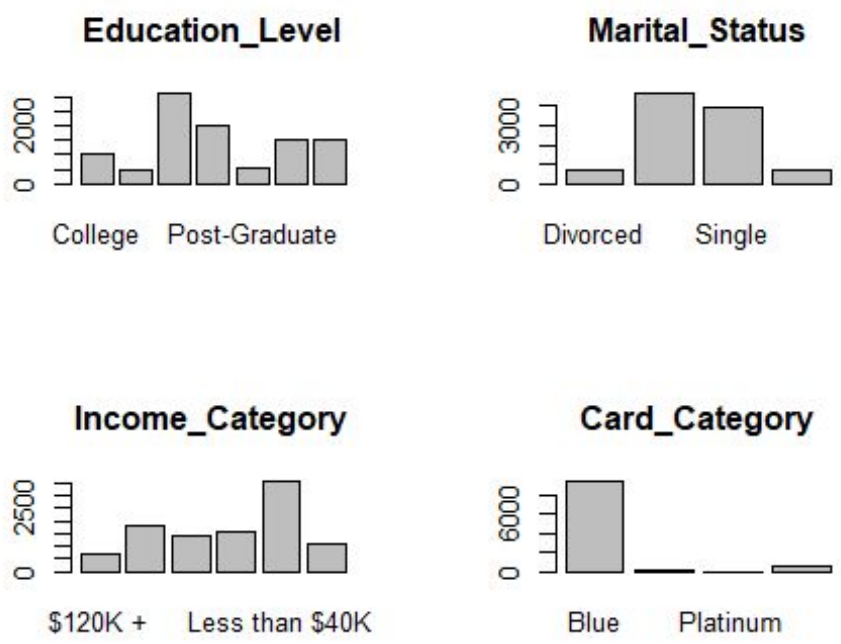
Se elaboran gráficas por variable para tener un primer acercamiento y conocer más sobre su comportamiento:

```r
sapply(df,class)
```

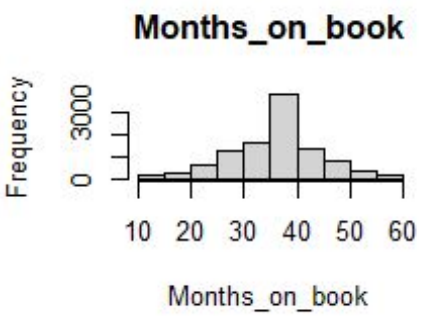
```
##                 CLIENTNUM           Attrition_Flag            Customer_Age
##                 "integer"              "character"               "integer"
##                    Gender          Dependent_count          Education_Level
##               "character"                 "integer"               "character"
##            Marital_Status          Income_Category           Card_Category
##               "character"              "character"               "character"
##            Months_on_book Total_Relationship_Count   Months_Inactive_12_mon
##                 "integer"                 "integer"                 "integer"
##     Contacts_Count_12_mon             Credit_Limit      Total_Revolving_Bal
##                 "integer"                 "numeric"                 "integer"
##           Avg_Open_To_Buy     Total_Amt_Chng_Q4_Q1          Total_Trans_Amt
##                 "numeric"                 "numeric"                 "integer"
##            Total_Trans_Ct      Total_Ct_Chng_Q4_Q1    Avg_Utilization_Ratio
##                 "integer"                 "numeric"                 "numeric"
```

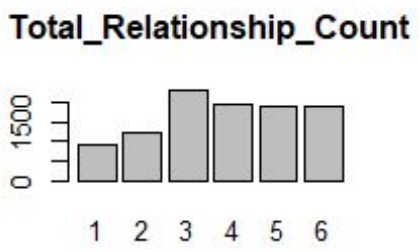
```r
library(ggplot2)

par(mfrow=c(2,2))

#p1 <- plot(CLIENTNUM, main="CLIENTNUM")
p2 <- plot(as.factor(Attrition_Flag), main="Attrition_Flag")
p3 <- hist(Customer_Age, main="Customer_Age")
p4 <- plot(as.factor(Gender), main="Gender")
p5 <- hist(Dependent_count, main="Dependent_count")
```

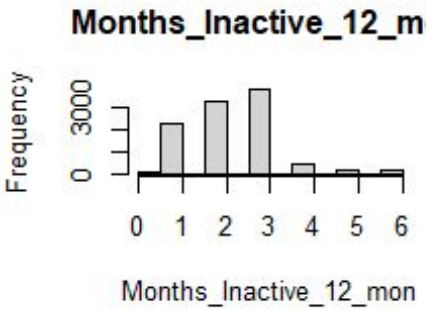
```r
p6 <- plot(as.factor(Education_Level), main="Education_Level")
p7 <- plot(as.factor(Marital_Status), main="Marital_Status")
p8 <- plot(as.factor(Income_Category), main="Income_Category")
p9 <- plot(as.factor(Card_Category), main="Card_Category")
```

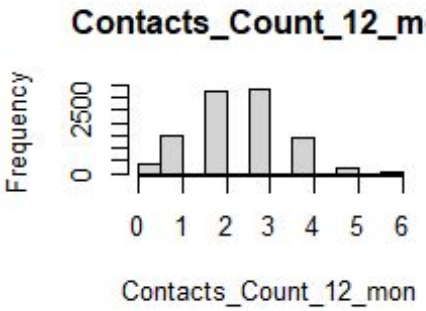



```r
p10 <- hist(Months_on_book, main="Months_on_book")
p11 <- plot(as.factor(Total_Relationship_Count),
main="Total_Relationship_Count")
p12 <- hist(Months_Inactive_12_mon, main="Months_Inactive_12_mon")
p13 <- hist(Contacts_Count_12_mon, main="Contacts_Count_12_mon")
```

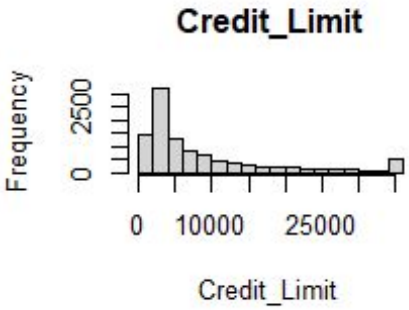


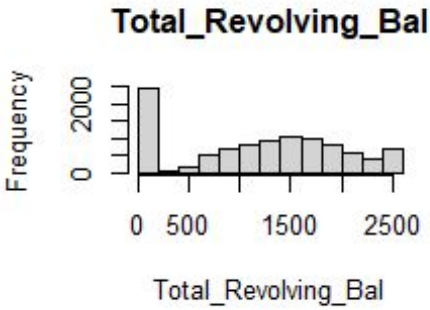


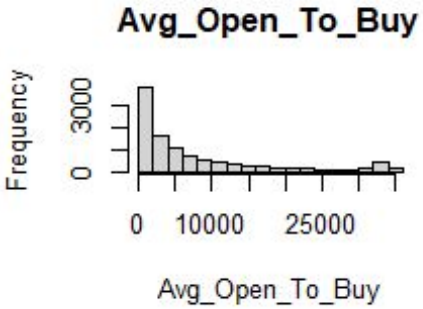


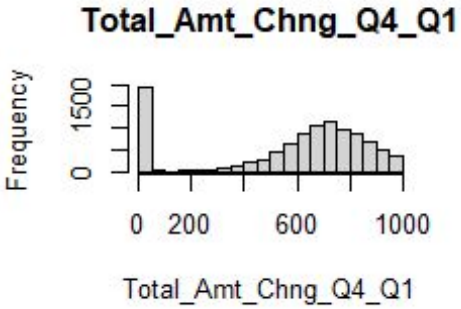



```
p14 <- hist(Credit_Limit, main="Credit_Limit")
p15 <- hist(Total_Revolving_Bal, main="Total_Revolving_Bal")
p16 <- hist(Avg_Open_To_Buy, main="Avg_Open_To_Buy")
p17 <- hist(Total_Amt_Chng_Q4_Q1, main="Total_Amt_Chng_Q4_Q1")
```

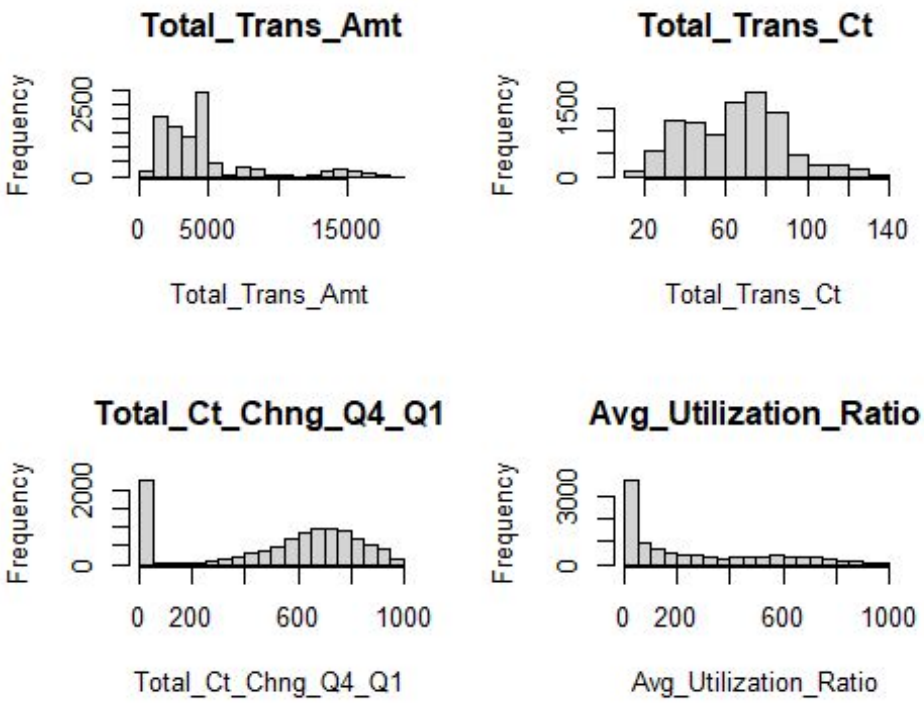









```
p18 <- hist(Total_Trans_Amt, main="Total_Trans_Amt")
p19 <- hist(Total_Trans_Ct, main="Total_Trans_Ct")
p20 <- hist(Total_Ct_Chng_Q4_Q1, main="Total_Ct_Chng_Q4_Q1")
p21 <- hist(Avg_Utilization_Ratio, main="Avg_Utilization_Ratio")
```



Los gráficos generados nos permiten explorar los datos de forma preliminar. Algunos de los insights que se pueden extraer son:

- Variables cuantitativas: presentan una alta dispersión en sus datos.
- Variables cuanlitativas: se puede observar que el banco en cuestión presenta una mayor cantidad de clientes con cuentas activas que inactivas, existen más mujeres que hombres, la mayoría están solteros o casados con estudios terminados y que usan la tarjeta de crédito categoría “Blue”.

Limpieza de los datos

El dataset procesado se denomina ‘BankChurners_clean.csv’ y se encuentra alojado en este mismo repositorio.

Normalización de los datos cuantitativos y cualitativos

Estas normalizaciones tienen como objetivo uniformizar los formatos. En este caso no es necesario normalizarlos. Sin embargo, los valores perdidos o valores extremos (de haberlos), se tratarán más adelante.

Se detallan los tipos de datos por variable:

```
sapply(df, function(x) class(x))
```

```
##                 CLIENTNUM           Attrition_Flag             Customer_Age
##                 "integer"              "character"                "integer"
##                    Gender          Dependent_count          Education_Level
##               "character"                "integer"               "character"
##            Marital_Status          Income_Category            Card_Category
##               "character"                "character"               "character"
##            Months_on_book Total_Relationship_Count   Months_Inactive_12_mon
##                 "integer"                "integer"                "integer"
##    Contacts_Count_12_mon             Credit_Limit      Total_Revolving_Bal
##                 "integer"                "numeric"                "integer"
##          Avg_Open_To_Buy     Total_Amt_Chng_Q4_Q1          Total_Trans_Amt
##                 "numeric"                "numeric"                "integer"
##           Total_Trans_Ct      Total_Ct_Chng_Q4_Q1    Avg_Utilization_Ratio
##                 "integer"                "numeric"                "numeric"
```

¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Este dataset en particular presenta una completitud del 100% como se demuestra a continuación:

```
sapply(df, function(x) sum(is.na(x)))
```

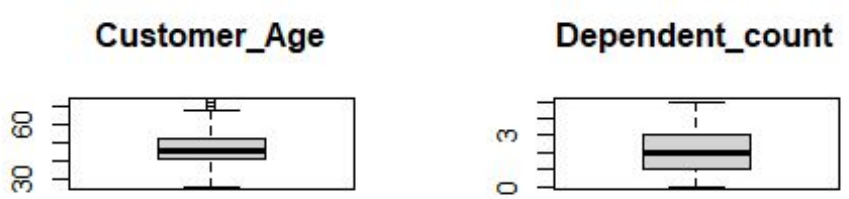
```
##                 CLIENTNUM           Attrition_Flag             Customer_Age
##                         0                        0                        0
##                    Gender          Dependent_count          Education_Level
##                         0                        0                        0
##            Marital_Status          Income_Category            Card_Category
##                         0                        0                        0
##            Months_on_book Total_Relationship_Count   Months_Inactive_12_mon
##                         0                        0                        0
##    Contacts_Count_12_mon             Credit_Limit      Total_Revolving_Bal
##                         0                        0                        0
##          Avg_Open_To_Buy     Total_Amt_Chng_Q4_Q1          Total_Trans_Amt
##                         0                        0                        0
##           Total_Trans_Ct      Total_Ct_Chng_Q4_Q1    Avg_Utilization_Ratio
##                         0                        0                        0
```

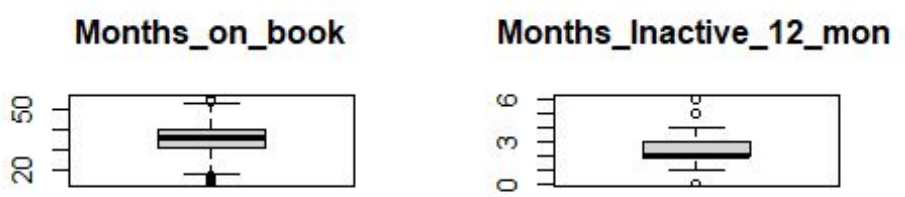
En el caso de haber encontrado elementos vacíos o nulos es conveniente conocer la fuente de los datos antes de proceder a eliminarlos ya que se puede estar perdiendo valiosa información. Lo que se hubiera hecho en este caso es una imputación de valores vacíos/nulos y, en medida de lo posible, evitar la eliminación de algún valor.

Identificación y tratamiento de valores extremos.

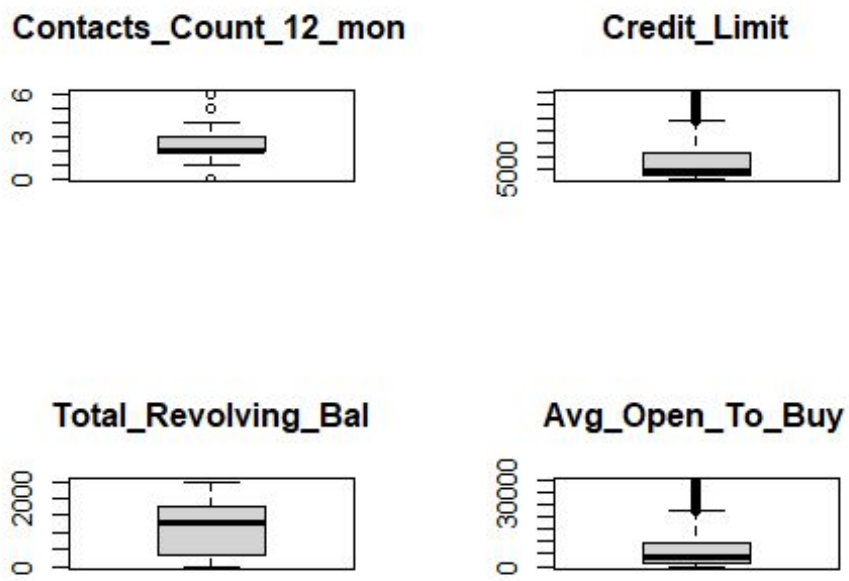
Se procede a hacer la exploración de outliers con el diagráma de caja y bigote, para posteriormente de acuerdo a un análisis por cada variable extraer e imputar los valores con la mediana de aquellas variables donde se haya considerado meritorio hacerlo.

```
par(mfrow=c(2,2))

#p1 <- boxplot(CLIENTNUM, main="CLIENTNUM")
p3 <- boxplot(Customer_Age, main="Customer_Age")
p5 <- boxplot(Dependent_count, main="Dependent_count")
p10 <- boxplot(Months_on_book, main="Months_on_book")
p12 <- boxplot(Months_Inactive_12_mon, main="Months_Inactive_12_mon")
```

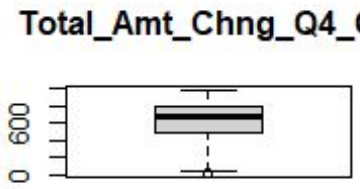


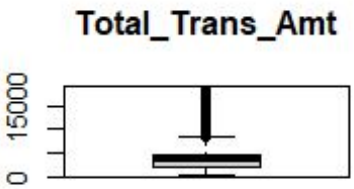


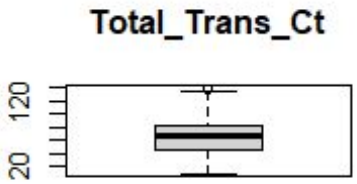

```
p13 <- boxplot(Contacts_Count_12_mon, main="Contacts_Count_12_mon")
p14 <- boxplot(Credit_Limit, main="Credit_Limit")
p15 <- boxplot(Total_Revolving_Bal, main="Total_Revolving_Bal")
p16 <- boxplot(Avg_Open_To_Buy, main="Avg_Open_To_Buy")
```

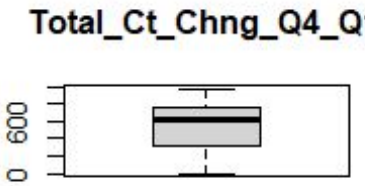



```
p17 <- boxplot(Total_Amt_Chng_Q4_Q1, main="Total_Amt_Chng_Q4_Q1")
p18 <- boxplot(Total_Trans_Amt, main="Total_Trans_Amt")
p19 <- boxplot(Total_Trans_Ct, main="Total_Trans_Ct")
p20 <- boxplot(Total_Ct_Chng_Q4_Q1, main="Total_Ct_Chng_Q4_Q1")
```










```
p21 <- boxplot(Avg_Utilization_Ratio, main="Avg_Utilization_Ratio")
```

De los diagramas se puede extraer la siguiente información:

- Simetría de la distribución de los datos.
- Detectar la presencia de valores atípicos o outliers.
- Ver cómo es la dispersión de los puntos con la mediana, los percentiles 25 y 75 y los valores máximos y mínimos.

Al revisar los diagramas generados se encuentra que:

La distribución no es simétrica para la mayoría de variables, y es simétrica en los casos de las variables: “Customer_Age”, “Dependent_count” y “Months_on_book”.

Para la mayoría de variables se ha optado por mantener los valores originales con excepción de aquellos outliers encontrados para las variables: “”Customer_Age” y Total_Trans_Ct”.

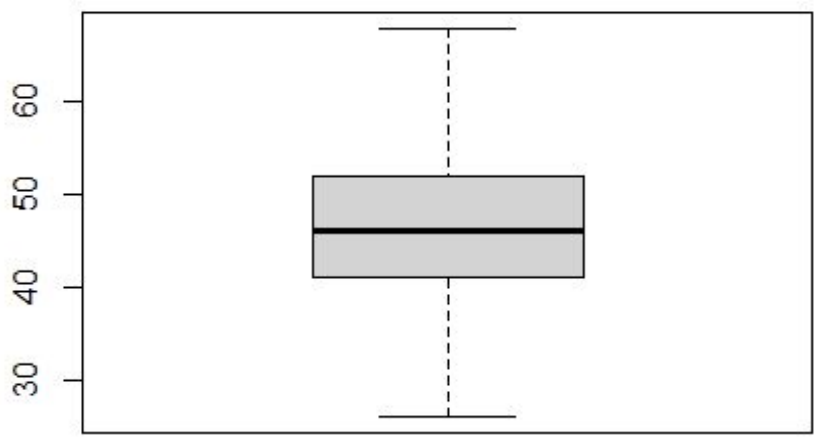
A continuación se detalla un análisis variable por variable:

Customer_Age

```
x<-boxplot.stats(Customer_Age)$out
idx <- which(Customer_Age %in% x)
ca <- df$Customer_Age[idx] #Valores atípicos
min(ca)

## [1] 70

max(ca)

## [1] 73

length(ca)

## [1] 2

ca

## [1] 73 70
```

En este caso, es un poco extraño que únicamente 2 de 10 mil clientes superen los 70 años y por ello se puede considerar que se trata de valores que probablemente se ingresaron o calcularon erróneamente. Si se tuviera la fecha de nacimiento sería fácil comprobar la edad real de ese par de clientes, sin embargo al desconocer ese dato se procede a imputar.

```
df$Customer_Age[df$Customer_Age>69] <- NA #Dejamos en NA solo los más
extremos
df$Customer_Age[idx]

## [1] NA NA

boxplot(df$Customer_Age)
```




```
idx <- which(is.na(df$Customer_Age))
length(idx) #número de valores perdidos
```

```
## [1] 2
```

```
for (i in 1:length(idx)){
index <- idx[i]
df[index,]$Customer_Age <- median(df$Customer_Age, na.rm=TRUE ) #imputación
}
df$Customer_Age[idx] #mostramos el resultado
```

```
## [1] 46 46
```

Months_on_book

```
x<-boxplot.stats(Months_on_book)$out
idx <- which(Months_on_book %in% x)
mob <- df$Months_on_book[idx] #Valores atípicos
min(mob)
```

```
## [1] 13
```

```
max(mob)
```

```
## [1] 56
```

```
length(mob)
```

```
## [1] 386
```

Como se observa, la variable 'Months_on_book' tiene 386 outliers. Sin embargo, dado que representa el período de pertenencia con la entidad financiera en meses va a ser normal el encontrar clientes muy antiguos (56 meses) o muy recientes (13 meses) por lo que no se los eliminará del dataset.

Months_Inactive_12_mon

```
x<-boxplot.stats(Months_Inactive_12_mon)$out
idx <- which(Months_Inactive_12_mon %in% x)
mi <- df$Months_Inactive_12_mon[idx] #Valores atípicos
min(mi)
```

```
## [1] 0
```

```
max(mi)
```

```
## [1] 6
```

```
length(mi)
```

```
## [1] 331
```

La variable 'Months_Inactive_12_mon' tiene 331 outliers. Sin embargo, dado que pueden haber clientes que han permanecido inactivos entre 0-6 meses, este podría ser un parámetro a considerar para saber si el cliente está por suspender o no algún servicio bancario. Por lo tanto, no se los eliminarán del dataset.

Contacts_Count_12_mon

```
x<-boxplot.stats(Contacts_Count_12_mon)$out
idx <- which(Contacts_Count_12_mon %in% x)
cc <- df$Contacts_Count_12_mon[idx] #Valores atípicos
min(cc)
```

```
## [1] 0
```

```
max(cc)
```

```
## [1] 6
```

```
length(cc)
```

```
## [1] 629
```

La variable 'Contacts_Count_12_mon' presenta 629 outliers pero dado el contexto de que pueden haber clientes con los cuales no se ha contactado entre 0-6 meses, este podría ser un parámetro a considerar para saber si el cliente está por suspender o no algún servicio bancario. Por lo tanto, no se los eliminarán del dataset.

Credit_Limit

```
x<-boxplot.stats(Credit_Limit)$out
idx <- which(Credit_Limit %in% x)
cl <- df$Credit_Limit[idx] #Valores atípicos
min(cl)

## [1] 23848

max(cl)

## [1] 34516

length(cl)

## [1] 984
```

La variable 'Credit_Limit' presenta 984 outliers pero dado que el límite de crédito puede variar dependiendo de muchos factores para a un cliente en específico se opta por mantenerlos.

Avg_Open_To_Buy

```
x<-boxplot.stats(Avg_Open_To_Buy)$out
idx <- which(Avg_Open_To_Buy %in% x)
avg <- df$Avg_Open_To_Buy[idx] #Valores atípicos
min(avg)

## [1] 22664

max(avg)

## [1] 34516

length(avg)

## [1] 963
```

Dado que la variable 'Avg_Open_To_Buy' presenta 963 outliers y representa línea de crédito abierta para compra se mantienen los valores.

Total_Amt_Chng_Q4_Q1

```
x<-boxplot.stats(Total_Amt_Chng_Q4_Q1)$out
idx <- which(Total_Amt_Chng_Q4_Q1 %in% x)
tac <- df$Total_Amt_Chng_Q4_Q1[idx] #Valores atípicos
min(tac)

## [1] 0

max(tac)

## [1] 18

length(tac)

## [1] 1954
```

La variable 'Total_Amt_Chng_Q4_Q1' tiene 1954 outliers y muestra los cambios transaccionales que pueden ser muy variables dependiendo del perfil de cada cliente, así que se opta por mantenerlos.

Total_Trans_Amt

```
x<-boxplot.stats(Total_Trans_Amt)$out
idx <- which(Total_Trans_Amt %in% x)
tta <- df$Total_Trans_Amt[idx] #Valores atípicos
min(tta)

## [1] 8620

max(tta)

## [1] 18484

length(tta)

## [1] 896
```

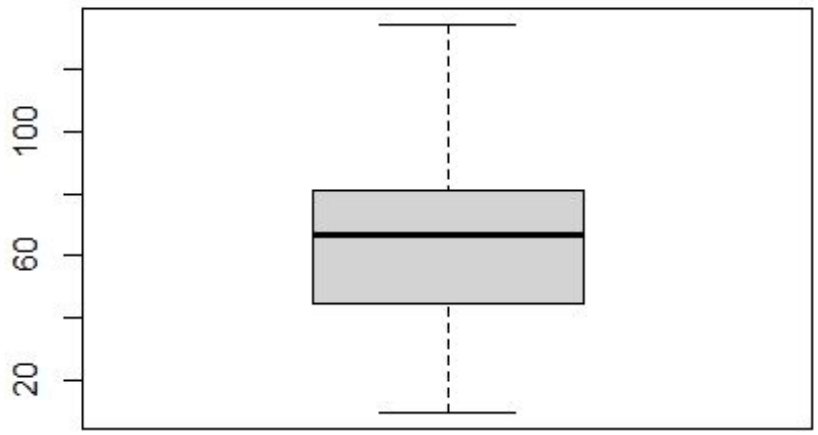
La variable 'Total_Trans_Amt' tiene 896 outliers y muestra la totalidad de cantidad de transacciones que pueden ser muy variables de persona a persona, así que se opta por mantenerlos.

Total_Trans_Ct

```
x<-boxplot.stats(Total_Trans_Ct)$out
idx <- which(Total_Trans_Ct %in% x)
ttc <- df$Total_Trans_Ct[idx] #Valores atípicos
min(ttc)

## [1] 138

max(ttc)

## [1] 139

length(ttc)

## [1] 2
```

La variable 'Total_Trans_Ct' tiene 2 outliers. Es extraño que existan únicamente dos clientes que durante los últimos 12 meses hayan realizado casi 140 transacciones. Por ello se opta por imputar esos datos.

```
df$Total_Trans_Ct[df$Total_Trans_Ct>137] <- NA #Dejamos en NA solo los más
extremos
df$Total_Trans_Ct[idx]

## [1] NA NA

boxplot(df$Total_Trans_Ct)
```




```
idx <- which(is.na(df$Total_Trans_Ct))
length(idx) #número de valores perdidos

## [1] 2

for (i in 1:length(idx)){
index <- idx[i]
df[index,]$Total_Trans_Ct <- median(df$Total_Trans_Ct, na.rm=TRUE )
#imputación
}
df$Total_Trans_Ct[idx] #mostramos el resultado

## [1] 67 67
```

Exportación de los datos preprocesados

```
# Exportación de los datos limpios en formato .csv
write.csv(df, "BankChurners_clean.csv")
```

Análisis de los datos.

Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

```
library(MASS)
library(dplyr)
library(plyr)
```

Para este estudio se tomarán algunas variables de tipo categórico para observar si se comportan de una forma distinta ante la variable que deseeamos predecir. Antes de ello se eliminará la variable CLIENTUM de nuestro conjunto de datos.

```
data=subset(df, select=-c(CLIENTNUM))
colnames(data)
```

```
##  [1] "Attrition_Flag"           "Customer_Age"
##  [3] "Gender"                   "Dependent_count"
##  [5] "Education_Level"          "Marital_Status"
##  [7] "Income_Category"          "Card_Category"
##  [9] "Months_on_book"           "Total_Relationship_Count"
## [11] "Months_Inactive_12_mon"   "Contacts_Count_12_mon"
## [13] "Credit_Limit"             "Total_Revolving_Bal"
## [15] "Avg_Open_To_Buy"          "Total_Amt_Chng_Q4_Q1"
## [17] "Total_Trans_Amt"          "Total_Trans_Ct"
## [19] "Total_Ct_Chng_Q4_Q1"      "Avg_Utilization_Ratio"
```

En este caso nos interesarán los grupos correspondientes al género, el nivel de educación y estado marital.

```
#Selección de grupos.
#Agrupación por genero.
data.femenino<-data[data$Gender=="F",]
data.masculino<-data[data$Gender=="M",]
#Agrupación por nivel de educación
data.HighSchool<-data[data$Education_Level=="High School",]
data.Graduate<-data[data$Education_Level=="Graduate",]
data.Uneducated<-data[data$Education_Level=="Uneducated",]
data.Unknown<-data[data$Education_Level=="Unknown",]
data.College<-data[data$Education_Level=="College",]
data.Postgraduate<-data[data$Education_Level=="Post-Graduate",]
data.Doctorate<-data[data$Education_Level=="Doctorate",]
#Agrupación por estado marital
data.Married<-data[data$Marital_Status=="Married",]
data.Single<-data[data$Marital_Status=="Single",]
data.UnknownStatus<-data[data$Marital_Status=="Unknown",]
data.Divorced<-data[data$Marital_Status=="Divorced",]
```

Por último dado que nuestra variable a predecir se encuentra con los valores “Existing Customer” y “Attrited Customer” se procede a cambiarla con notación binaria. Es decir, “Attrited Customer” tomará el valor 1 y “Existing Customer” tomará el valor 0.

```
#Se convierten a dummy
data$Attrition_Flag<-ifelse(data$Attrition_Flag=="Attrited Customer",1,0)
```

Comprobación de la normalidad y homogeneidad de la varianza.

Es de principal importancia dependiendo del modelo a aplicar, el realizar la correspondiente comprobación de normalidad y homogeneidad de la varianza en nuestro conjunto de datos. Por lo cual, para realizar la comprobación de nuestras variables sobre una población normalmente distribuida, se utilizará la prueba de normalidad de Anderson-Darling.

Teniendo en cuenta un $\alpha = 0.05$, si obtenemos un p-valor superior al alpha mencionado anteriormente, entonces no rechazaremos nuestra hipótesis nula y los valores vendrán de una distribución/población normal.

```
library(nortest)
col.names=colnames(data)
alpha = 0.05

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

```
## Variables que no siguen una distribución normal:
## Attrition_Flag
## Customer_Age
## Dependent_count
## Months_on_book
## Total_Relationship_Count
## Months_Inactive_12_mon
## Contacts_Count_12_mon
## Credit_Limit
## Total_Revolving_Bal
## Avg_Open_To_Buy
## Total_Amt_Chng_Q4_Q1
## Total_Trans_Amt
```

```
## Total_Trans_Ct
## Total_Ct_Chng_Q4_Q1Avg_Utilization_Ratio
```

Posterior a ello, realizamos la comprobación sobre la homogeneidad de las varianzas con el test de Fligner.Killen. Para nuestro caso, estudiaremos la homogeneidad en cuanto a los grupos conformados por las siguientes variables: Género, nivel educativo, estado marital, categoría de ingresos y categoría de la tarjeta. Recordemos que la hipótesis nula consiste en que las varianzas de los grupos son iguales.

```
fligner.test(Attrition_Flag~Gender, data=data)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Attrition_Flag by Gender
## Fligner-Killeen:med chi-squared = 14.067, df = 1, p-value = 0.0001764

fligner.test(Attrition_Flag~Education_Level, data=data)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Attrition_Flag by Education_Level
## Fligner-Killeen:med chi-squared = 12.51, df = 6, p-value = 0.05151

fligner.test(Attrition_Flag~Marital_Status, data=data)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Attrition_Flag by Marital_Status
## Fligner-Killeen:med chi-squared = 6.0555, df = 3, p-value = 0.1089

fligner.test(Attrition_Flag~Income_Category, data=data)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Attrition_Flag by Income_Category
## Fligner-Killeen:med chi-squared = 12.831, df = 5, p-value = 0.02502

fligner.test(Attrition_Flag~Card_Category, data=data)

##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Attrition_Flag by Card_Category
## Fligner-Killeen:med chi-squared = 2.234, df = 3, p-value = 0.5253
```

Teniendo en cuenta los resultados anteriores, solamente podemos no rechazar la hipótesis nula para las variables: nivel educativo, estado marital y categoría de la tarjeta. Dado que se obtienen que el valor-p asociado a la prueba es mayor que el alpha establecido.

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

Correlaciones

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")
# Calcular el coeficiente de correlación para cada variable cuantitativa
# con respecto al campo "precio"
for (i in 1:(ncol(data) - 1)) {
  if (is.integer(data[[i]]) | is.numeric(data[[i]])) {
    spearman_test = cor.test(data[[i]],
                             data[[1]],
                             method = "spearman")
    corr_coef = spearman_test$estimate
    p_val = spearman_test$p.value
    # Add row to matrix
    pair = matrix(ncol = 2, nrow = 1)
    pair[1][1] = corr_coef
    pair[2][1] = p_val
    corr_matrix <- rbind(corr_matrix, pair)
    rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(data)[i]
  }
}
```

```
## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
```

```
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[i]], data[[1]], method = "spearman"):
Cannot
## compute exact p-value with ties
```

```
corr_matrix
```

```
##                             estimate       p-value
## Attrition_Flag            1.00000000  0.000000e+00
## Customer_Age              0.01766624  7.544887e-02
## Dependent_count           0.02098332  3.472192e-02
## Months_on_book            0.01529958  1.236719e-01
## Total_Relationship_Count -0.14967404  8.092007e-52
## Months_Inactive_12_mon    0.17183886  5.792309e-68
## Contacts_Count_12_mon     0.18903770  4.123362e-82
## Credit_Limit             -0.05090987  2.961133e-07
## Total_Revolving_Bal      -0.24055101 2.921366e-133
## Avg_Open_To_Buy           0.02750028  5.646767e-03
## Total_Amt_Chng_Q4_Q1     -0.06430376  9.357322e-11
## Total_Trans_Amt          -0.22378218 3.915150e-115
```

```
## Total_Trans_Ct           -0.37608761  0.000000e+00
## Total_Ct_Chng_Q4_Q1      -0.20240148  4.225131e-94
```

Los valores que estamos observando anteriormente de correlación, no conllevan una fuerte correlación teniendo en cuenta nuestra variable objetivo, ya que por lo general están variando entre -0.37 y 0.18, por lo cual se pueden clasificar dichos valores como una correlación débil.

Contraste de hipótesis

Para ello se utilizará la prueba de Fisher, la cual es un test exacto cuando se quiere observar si existe asociación entre dos variables de tipo cualitativo, es decir, si las proporciones de una variable son diferentes dependiendo del valor que adquiera la otra variable.

H0: Las variables son independientes por lo que una variable no varia entre los distintos niveles de la otra variable.

H1: Las variables son dependientes, una variable varia entre los distintos niveles de la otra variable.

```
tabla <- table(data$Gender, data$Attrition_Flag)
tabla
```

```
##
##        0    1
##   F 4428  930
##   M 4072  697
```

Test de Fisher.

```
fisher.test(x = tabla, alternative = "two.sided")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  tabla
## p-value = 0.0001825
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.7310913 0.9082957
## sample estimates:
## odds ratio
##  0.8150034
```

Teniendo en cuenta el valor p observado, es igual a 0.0001825 es menor que el nivel de significancia 0.05, rechazamos la hipótesis nula. Es decir, las variables de Attrition_Flag y Gender son dependientes. Este pequeño análisis nos ayudará a futuro para ingresar las variables en el modelo de regresión logística.

Modelo de regresión logística

Se plantea un primer modelo de regresión logística teniendo en cuenta que nuestra variable objetivo es binaria. Además se traen variables de interés en las cuales trataremos de predecir la estancia de los clientes dentro de la empresa. Entre dichas variables encontramos por ejemplo: edad del usuario, género, si tiene personas que dependen del usuario, nivel educativo, estado marital, categoría de ingresos, categoría de la tarjeta entre otros.

```
logisticModelFull <- glm(Attrition_Flag ~
Customer_Age+Gender+Dependent_count+
                           Education_Level+Marital_Status+Income_Category+
                           Card_Category+Months_on_book+
                           Total_Relationship_Count+Months_Inactive_12_mon
                         +Contacts_Count_12_mon+Total_Revolving_Bal+
                           Total_Trans_Amt+Total_Trans_Ct,
                         family = "binomial",data)
summary(logisticModelFull)
```

```
##
## Call:
## glm(formula = Attrition_Flag ~ Customer_Age + Gender + Dependent_count +
##     Education_Level + Marital_Status + Income_Category + Card_Category +
##     Months_on_book + Total_Relationship_Count + Months_Inactive_12_mon +
##     Contacts_Count_12_mon + Total_Revolving_Bal + Total_Trans_Amt +
##     Total_Trans_Ct, family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9222  -0.3986  -0.1880  -0.0749   3.5550
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    4.409e+00  4.167e-01  10.581  < 2e-16 ***
## Customer_Age                  -5.469e-03  7.310e-03  -0.748 0.454402
## GenderM                       -8.969e-01  1.402e-01  -6.399 1.56e-10 ***
## Dependent_count                1.324e-01  2.859e-02   4.633 3.61e-06 ***
## Education_LevelDoctorate       4.139e-01  1.968e-01   2.104 0.035405 *
## Education_LevelGraduate        2.698e-02  1.335e-01   0.202 0.839918
## Education_LevelHigh School    -1.604e-02  1.427e-01  -0.112 0.910500
## Education_LevelPost-Graduate   3.408e-01  1.947e-01   1.751 0.080008 .
## Education_LevelUneducated      3.180e-02  1.505e-01   0.211 0.832666
## Education_LevelUnknown         1.405e-01  1.480e-01   0.949 0.342391
## Marital_StatusMarried         -5.162e-01  1.459e-01  -3.537 0.000405 ***
## Marital_StatusSingle           1.069e-01  1.467e-01   0.728 0.466352
## Marital_StatusUnknown          1.694e-02  1.872e-01   0.091 0.927862
## Income_Category$40K - $60K    -6.645e-01  1.800e-01  -3.692 0.000222 ***
## Income_Category$60K - $80K    -5.414e-01  1.662e-01  -3.257 0.001126 **
## Income_Category$80K - $120K   -2.987e-01  1.598e-01  -1.869 0.061574 .
## Income_CategoryLess than $40K -5.332e-01  1.946e-01  -2.740 0.006143 **
```

```
## Income_CategoryUnknown        -6.816e-01  2.176e-01  -3.132 0.001737 **
## Card_CategoryGold              7.818e-01  3.264e-01   2.395 0.016601 *
## Card_CategoryPlatinum          7.040e-01  6.613e-01   1.065 0.287042
## Card_CategorySilver            1.279e-01  1.665e-01   0.768 0.442323
## Months_on_book                -4.760e-03  7.322e-03  -0.650 0.515651
## Total_Relationship_Count      -4.624e-01  2.628e-02 -17.595  < 2e-16 ***
## Months_Inactive_12_mon         4.991e-01  3.644e-02  13.695  < 2e-16 ***
## Contacts_Count_12_mon          5.363e-01  3.491e-02  15.361  < 2e-16 ***
## Total_Revolving_Bal           -1.001e-03  4.449e-05 -22.501  < 2e-16 ***
## Total_Trans_Amt                4.418e-04  2.218e-05  19.919  < 2e-16 ***
## Total_Trans_Ct                -1.185e-01  3.517e-03 -33.695  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8927.2  on 10126  degrees of freedom
## Residual deviance: 5102.4  on 10099  degrees of freedom
## AIC: 5158.4
##
## Number of Fisher Scoring iterations: 6
```

Dentro del resumen de los resultados del modelo obtenemos que las siguientes variables son estadísticamente signficativas: Gender, Dependent_count, Educational_Level, Marital_Status, Income_Category,Card_Category, Total_Relationship_Count, Months_Inactive_12_mon, Contacts_Count_12_mon, Total_Revolving_Bal, Total_Trans_Amt y Total_Trans_Ct.

Sin embargo, utilizaremos la función stepAIC de R, la cual realiza la selección del modelo paso a paso por el criterio de información de Akaike (AIC), el cual es una medida de calidad relativa de un modelo estadístico.

```
logisticModelNew <-  stepAIC(logisticModelFull, trace = 0)
summary(logisticModelNew)
```

```
##
## Call:
## glm(formula = Attrition_Flag ~ Customer_Age + Gender + Dependent_count +
##     Marital_Status + Income_Category + Card_Category +
Total_Relationship_Count +
##     Months_Inactive_12_mon + Contacts_Count_12_mon + Total_Revolving_Bal +
##     Total_Trans_Amt + Total_Trans_Ct, family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9538  -0.3956  -0.1889  -0.0761   3.5570
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   4.490e+00  4.033e-01  11.133  < 2e-16 ***
```

```
## Customer_Age                   -9.203e-03  4.520e-03  -2.036 0.041754 *
## GenderM                        -8.999e-01  1.400e-01  -6.426 1.31e-10 ***
## Dependent_count                 1.320e-01  2.854e-02   4.625 3.74e-06 ***
## Marital_StatusMarried          -5.190e-01  1.456e-01  -3.565 0.000364 ***
## Marital_StatusSingle            1.039e-01  1.464e-01   0.710 0.477799
## Marital_StatusUnknown           8.944e-03  1.867e-01   0.048 0.961799
## Income_Category$40K - $60K     -6.608e-01  1.797e-01  -3.678 0.000235 ***
## Income_Category$60K - $80K     -5.346e-01  1.661e-01  -3.219 0.001287 **
## Income_Category$80K - $120K    -2.966e-01  1.596e-01  -1.859 0.063098 .
## Income_CategoryLess than $40K -5.312e-01  1.943e-01  -2.734 0.006264 **
## Income_CategoryUnknown         -6.698e-01  2.175e-01  -3.080 0.002071 **
## Card_CategoryGold               7.772e-01  3.272e-01   2.376 0.017525 *
## Card_CategoryPlatinum           7.858e-01  6.609e-01   1.189 0.234449
## Card_CategorySilver             1.240e-01  1.662e-01   0.746 0.455575
## Total_Relationship_Count       -4.635e-01  2.625e-02 -17.656  < 2e-16 ***
## Months_Inactive_12_mon          4.948e-01  3.620e-02  13.670  < 2e-16 ***
## Contacts_Count_12_mon           5.372e-01  3.488e-02  15.401  < 2e-16 ***
## Total_Revolving_Bal            -1.001e-03  4.441e-05 -22.538  < 2e-16 ***
## Total_Trans_Amt                 4.416e-04  2.216e-05  19.932  < 2e-16 ***
## Total_Trans_Ct                 -1.184e-01  3.514e-03 -33.689  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8927.2  on 10126  degrees of freedom
## Residual deviance: 5113.0  on 10106  degrees of freedom
## AIC: 5155
##
## Number of Fisher Scoring iterations: 6
```

Finalmente mediante esta función se elige el modelo con las siguientes variables: Customer_Age, Gender, Dependent_count, Marital_Status, Income_Category, Card_Category, Total_Relationship_Count, Months_Inactive_12_mon, Contacts_Count_12_mon, Total_Revolving_Bal, Total_Trans_Amt, Total_Trans_Ct.

Validación cruzada

Por otra parte realizaremos un acercamiento con validación cruzada con el modelo anteriormente visto, y observar si se realizan algunos cambios dentro del modelo, en particular se utilizará el método Validación cruzada de K-fold. Recordando que este método evalua el rendimiento del modelo en diferentes subconjuntos de los datos de entrenamiento y luego calcula la tasa de error de predicción promedio. En la práctica, normalmente se realiza una validación cruzada de k veces utilizando $k = 5$ o $k = 10$, por lo cual utilizaremos $k = 10$.

Primero, se decide realizar una división de los datos en los conjuntos de datos de entrenamiento y de prueba. En este caso se genera una variable aleatoria que toma valores 1 y 0 gracias a que proviene de una distribución binomial con una probabilidad de 0.66.

```r
library(caret)
set.seed(27345)

data$isTrain <- rbinom(nrow(data),1,0.66)
train <-  data %>% filter(data$isTrain =="1")
test <- data %>%  filter(data$isTrain == "0")
train_control <- trainControl(method = "cv", number = 10)
model <- train(Attrition_Flag ~ Customer_Age + Gender + Dependent_count +
                 Marital_Status + Income_Category + Card_Category +
Total_Relationship_Count +
                 Months_Inactive_12_mon + Contacts_Count_12_mon +
Total_Revolving_Bal +
                 Total_Trans_Amt + Total_Trans_Ct,
               data = train,
               trControl = train_control,
               method = "glm",
               family=binomial())

## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to
do
## classification? If so, use a 2 level factor as your outcome column.

summary(model)

##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8534  -0.4067  -0.1933  -0.0801   3.4678
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      4.605e+00  4.870e-01   9.457  < 2e-16 ***
## Customer_Age                    -1.309e-02  5.511e-03  -2.375 0.017526 *
## GenderM                         -6.875e-01  1.639e-01  -4.195 2.73e-05 ***
## Dependent_count                  1.436e-01  3.499e-02   4.105 4.05e-05 ***
## Marital_StatusMarried           -5.538e-01  1.747e-01  -3.171 0.001519 **
## Marital_StatusSingle             4.502e-02  1.754e-01   0.257 0.797391
## Marital_StatusUnknown           -8.831e-02  2.258e-01  -0.391 0.695691
## `Income_Category$40K - $60K`    -6.204e-01  2.148e-01  -2.888 0.003871 **
## `Income_Category$60K - $80K`    -7.223e-01  2.025e-01  -3.567 0.000362 ***
## `Income_Category$80K - $120K`   -5.346e-01  1.929e-01  -2.772 0.005579 **
## `Income_CategoryLess than $40K` -4.703e-01  2.300e-01  -2.045 0.040890 *
## Income_CategoryUnknown          -7.046e-01  2.601e-01  -2.709 0.006747 **
## Card_CategoryGold                5.179e-01  4.146e-01   1.249 0.211686
## Card_CategoryPlatinum            3.926e-01  8.026e-01   0.489 0.624742
## Card_CategorySilver              1.356e-01  2.022e-01   0.670 0.502631
```

```
## Total_Relationship_Count        -4.285e-01  3.183e-02 -13.460  < 2e-16 ***
## Months_Inactive_12_mon           4.797e-01  4.407e-02  10.885  < 2e-16 ***
## Contacts_Count_12_mon            5.059e-01  4.245e-02  11.919  < 2e-16 ***
## Total_Revolving_Bal             -9.920e-04  5.433e-05 -18.260  < 2e-16 ***
## Total_Trans_Amt                  4.576e-04  2.674e-05  17.115  < 2e-16 ***
## Total_Trans_Ct                  -1.187e-01  4.286e-03 -27.697  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5886.8  on 6657  degrees of freedom
## Residual deviance: 3418.5  on 6637  degrees of freedom
## AIC: 3460.5
##
## Number of Fisher Scoring iterations: 6
```

Sin embargo, observamos que en este acercamiento se encuentra que Card_Category no es significativo estadisticamente en ninguno de sus niveles por lo cual volvemos a plantear el modelo sin dicha categoría.

```
library(caret)
set.seed(27345)

data$isTrain <- rbinom(nrow(data),1,0.66)
train <-  data %>% filter(data$isTrain =="1")
test <- data %>%  filter(data$isTrain == "0")
train_control <- trainControl(method = "cv", number = 10)
model <- train(Attrition_Flag ~ Customer_Age + Gender + Dependent_count +
                 Marital_Status + Income_Category + Total_Relationship_Count
+
                 Months_Inactive_12_mon + Contacts_Count_12_mon +
Total_Revolving_Bal +
                 Total_Trans_Amt + Total_Trans_Ct,
               data = train,
               trControl = train_control,
               method = "glm",
               family=binomial())

## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to
do
## classification? If so, use a 2 level factor as your outcome column.

summary(model)

##
## Call:
## NULL
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -2.6688  -0.4076  -0.1935  -0.0800   3.4660
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     4.615e+00  4.867e-01   9.483  < 2e-16 ***
## Customer_Age                   -1.295e-02  5.504e-03  -2.353 0.018648 *
## GenderM                        -6.877e-01  1.638e-01  -4.199 2.69e-05 ***
## Dependent_count                 1.439e-01  3.498e-02   4.115 3.88e-05 ***
## Marital_StatusMarried          -5.622e-01  1.746e-01  -3.220 0.001280 **
## Marital_StatusSingle            4.266e-02  1.754e-01   0.243 0.807870
## Marital_StatusUnknown          -8.391e-02  2.258e-01  -0.372 0.710163
## `Income_Category$40K - $60K`   -6.281e-01  2.149e-01  -2.923 0.003466 **
## `Income_Category$60K - $80K`   -7.253e-01  2.027e-01  -3.578 0.000347 ***
## `Income_Category$80K - $120K`  -5.370e-01  1.930e-01  -2.782 0.005397 **
## `Income_CategoryLess than $40K` -4.803e-01  2.300e-01  -2.088 0.036796 *
## Income_CategoryUnknown         -7.114e-01  2.600e-01  -2.736 0.006225 **
## Total_Relationship_Count       -4.292e-01  3.183e-02 -13.484  < 2e-16 ***
## Months_Inactive_12_mon          4.786e-01  4.403e-02  10.869  < 2e-16 ***
## Contacts_Count_12_mon           5.060e-01  4.242e-02  11.927  < 2e-16 ***
## Total_Revolving_Bal            -9.895e-04  5.424e-05 -18.242  < 2e-16 ***
## Total_Trans_Amt                 4.609e-04  2.653e-05  17.375  < 2e-16 ***
## Total_Trans_Ct                 -1.187e-01  4.281e-03 -27.740  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5886.8  on 6657  degrees of freedom
## Residual deviance: 3420.5  on 6640  degrees of freedom
## AIC: 3456.5
##
## Number of Fisher Scoring iterations: 6
```

Algo que podemos observar teniendo en cuenta la validación cruzada junto con el modelo anteriormente propuesto es el valor del Criterio de Akaike que toma. Por ejemplo para el modelo *LogisticModelNew* tenemos un valor de $AIC = 5155$ mientras que para el modelo con validación cruzada se tiene un valor de $AIC = 3456.5$. Esto podría llegar a ser un indicador, para tener en cuenta una métodología (K-Fold) sobre otra.

Representación de los resultados a partir de tablas y gráficas.

Modelo de regresión lineal

Interpretacion de coeficientes (odds)

```
odds <-  coef(logisticModelNew) %>% exp() %>% round(2)
odds
```

```
##                    (Intercept)                   Customer_Age
##                          89.15                           0.99
##                         GenderM                Dependent_count
##                           0.41                           1.14
##          Marital_StatusMarried           Marital_StatusSingle
##                           0.60                           1.11
##          Marital_StatusUnknown    Income_Category$40K - $60K
##                           1.01                           0.52
##    Income_Category$60K - $80K   Income_Category$80K - $120K
##                           0.59                           0.74
## Income_CategoryLess than $40K        Income_CategoryUnknown
##                           0.59                           0.51
##              Card_CategoryGold         Card_CategoryPlatinum
##                           2.18                           2.19
##            Card_CategorySilver      Total_Relationship_Count
##                           1.13                           0.63
##         Months_Inactive_12_mon         Contacts_Count_12_mon
##                           1.64                           1.71
##            Total_Revolving_Bal               Total_Trans_Amt
##                           1.00                           1.00
##                 Total_Trans_Ct
##                           0.89
```

Una breve interpretación de los resultados obtenidos con los coeficientes u "odds", puede ser la siguiente:

- Los usuarios que se encuentran solteros aumentan la tasa de abandono en un 11%.
- Dentro de los usuarios que tienen diferentes categoráis de tarjetas se tiene que aquellos que tienen la categoría platino aumentan la tasa de abandono en un 119%.
- Una relación inversa que observamos la tasa de abandono es un un 59% menor si el usario es de género masculino comparado con las usuarias femeninas.

Predicciones

Se realiza el modelo escogido anteriomente con el conjunto de datos de entrenamiento, y se realizan las predicciones con la función predict con el conjunto de datos de prueba.

```
LogisticTrainNew <- glm(formula = Attrition_Flag ~ Customer_Age + Gender +
Dependent_count +
                          Marital_Status + Income_Category + Card_Category +
Total_Relationship_Count +
                          Months_Inactive_12_mon + Contacts_Count_12_mon +
Total_Revolving_Bal +
                          Total_Trans_Amt + Total_Trans_Ct, family =
"binomial", data = train)

#prediciton
test$predictNew <- predict(LogisticTrainNew , type = "response" , newdata =
test)
```

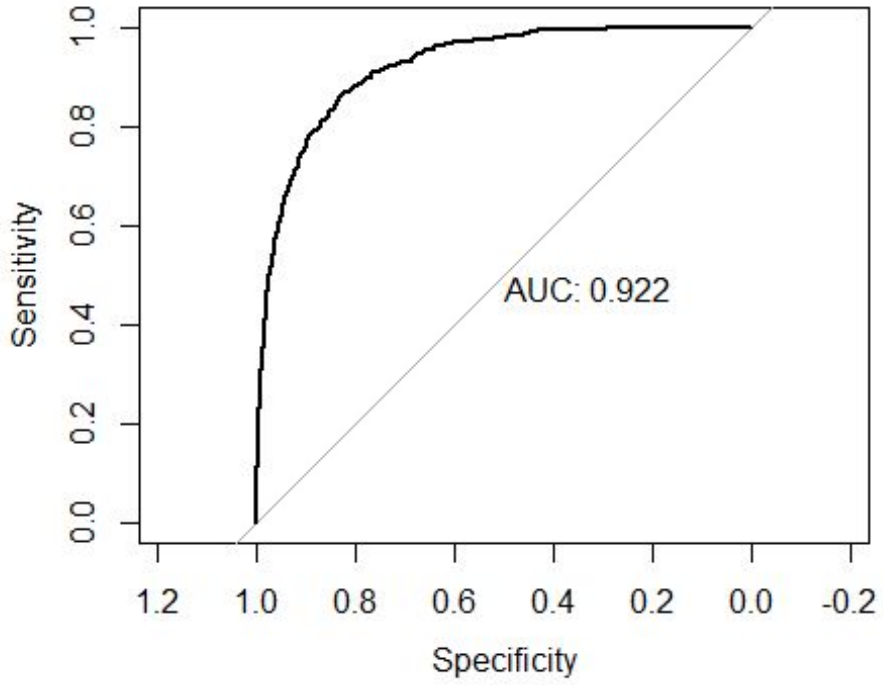
Para revisar que tan bien el modelo logístico predice los valores del conjunto de prueba se procede a realizar la matriz de confusión.

```
library(readr)
library(ggplot2)
library(boot)
library(e1071)
predicciones=as.factor(ifelse(test = test$predictNew > 0.35, yes = 1, no =
0))
observaciones=as.factor(test$Attrition_Flag)
matriz<-confusionMatrix(observaciones, predicciones)
matriz
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2694  223
##          1  163  389
##
##                Accuracy : 0.8887
##                  95% CI : (0.8778, 0.899)
##     No Information Rate : 0.8236
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6017
##
##  Mcnemar's Test P-Value : 0.002673
##
##             Sensitivity : 0.9429
##             Specificity : 0.6356
##          Pos Pred Value : 0.9236
##          Neg Pred Value : 0.7047
##              Prevalence : 0.8236
##          Detection Rate : 0.7766
##    Detection Prevalence : 0.8409
##       Balanced Accuracy : 0.7893
##
##        'Positive' Class : 0
##
```

Como podemos observar, el modelo de reresión logística planteado nos da un nivel de exactitud de 88.87%, lo cual nos puede indicar la falta de variables exogenas que nos ayuden a modelar nuestro datos actuales.

```
library(pROC)
test_prob = predict(logisticModelNew, newdata = test, type = "response")
test_roc = roc(test$Attrition_Flag ~ test_prob, plot = TRUE, print.auc =
TRUE)
```

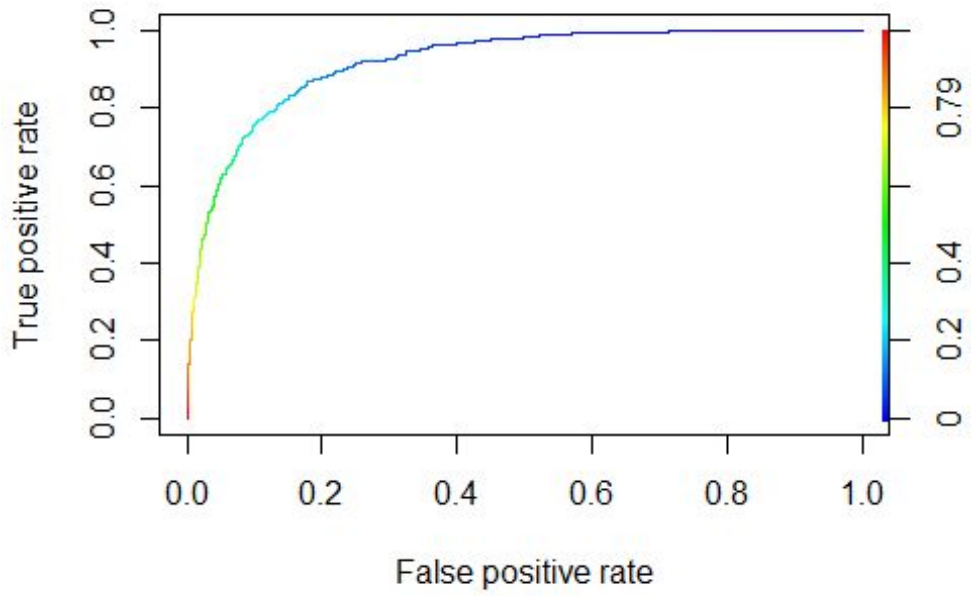


Por otra parte se realiza la grafíca ROC en la cual contiene también el valor de AUC (más conocida como el area bajo la curva ROC), hay que recordar que este valor varia entre 0 y 1, donde un modelo cuyas predicciones son un 100% incorrectas tiene un AUC de 0.0, otras donde sus predicciones son de un 100% entonces su valor de AUC asociado es de 1.0. En nuestro caso, obtuvimos un valor de AUC igual a 0.922 lo cual nos indica que las predicciones que se están haciendo se desvian un poco pero están cerca de pertenecer a un modelo "perfecto".

Modelo de regresión lineal con validación cruzada

Para el modelo de regresión logística con validación cruzada se realizan las correspondientes predicciones para el conjunto de datos de test y generamos la curva ROC.

```
library(ggplot2)
library(ROCR)

predict0 <- predict(model, type = 'raw',test)
ROCRpred0 <- prediction(as.numeric(predict0),as.numeric(test$Attrition_Flag))
ROCRperf0<- performance(ROCRpred0, 'tpr', 'fpr')
plot(ROCRperf0, colorize=TRUE, text.adj=c(-0.2,1.7))
```



Por otra parte, comparando las predicciones generadas por el método de validación cruzada no varian comparado con el modelo anteriormente presenteado, al igual que los valores de precisión, sensibilidad y especificidad.

```
predicciones2=as.factor(ifelse(test = predict0 > 0.35, yes = 1, no = 0))
observaciones2=as.factor(test$Attrition_Flag)
matriz2<-confusionMatrix(observaciones2, predicciones2)
matriz2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2693  224
##          1  165  387
##
##                Accuracy : 0.8879
##                  95% CI : (0.8769, 0.8982)
##     No Information Rate : 0.8239
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5984
##
##  Mcnemar's Test P-Value : 0.003275
##
##             Sensitivity : 0.9423
```

```
##             Specificity : 0.6334
##          Pos Pred Value : 0.9232
##          Neg Pred Value : 0.7011
##              Prevalence : 0.8239
##          Detection Rate : 0.7763
##    Detection Prevalence : 0.8409
##       Balanced Accuracy : 0.7878
##
##        'Positive' Class : 0
##
```

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

Teniendo en cuenta los resultados por ejemplo de la exactitud y la precisión presentada dentro del código se recomendaría el poder utilizar nuevas variables que nos ayuden a encontrar relaciones escondidas dentro de los datos y generen un mejor modelo con una exactitud y precisión más alta. Sin embargo los resultados actuales que se tienen del modelo de regresión logístico son bastante buenos teniendo en cuenta los datos y la cantidad de datos que se obtuvieron para la realización de la práctica. Otra opción que tendíamos a futuro, sería plantear otros tipos de modelos que se comparen con el modelo actual, tal como lo puede ser un árbol de decisión, clústeres e inclusive redes neuronales para manejar las relaciones complejas que no se pueden detectar utilizando otros modelos anteriormente mencionados.

Por otra parte, respondiendo a nuestra pregunta planteada y a partir de los resultados obtenidos mediante el modelo de regresión logistica podemos modelar y predecir a cierto nivel la deserción de los clientes dentro de la empresa.

Código: Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

Project Link:https://github.com/wickedlexie/AnalisisPRA2

Recursos

Los siguientes recursos son de utilidad para la realización de la práctica:

● Calvo M., Subirats L., Pérez D. (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC.
● Megan Squire (2015). Clean Data. Packt Publishing Ltd.

● Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
● Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.
● Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.
● Wes McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc.
● Tutorial de Github: https://guides.github.com/activities/hello-world.

Criterios de valoración

Todos los apartados son obligatorios. La ponderación de los ejercicios es la siguiente:

Los apartados 1, 2 y 6 valen 0,5 puntos.
Los apartados 3, 5 y 7 valen 2 puntos.
El apartado 4 vale 2,5 puntos.

Se valorará la idoneidad de las respuestas, que deberán ser claras y completas. Las diferentes etapas deberán justificarse y acompañarse del código correspondiente. También se valorará la síntesis y claridad, a través del uso de comentarios, del código resultante, así como la calidad de los datos finales analizados.

Formato y fecha de entrega

Durante la semana del 21 al 25 de diciembre el grupo podrá entregar al profesor una entrega parcial opcional. Esta entrega parcial es muy recomendable para recibir asesoramiento sobre la práctica y verificar que la dirección tomada es la correcta. Se entregarán comentarios a los estudiantes que hayan efectuado la entrega parcial pero no contará para la nota de la práctica.

En la entrega parcial los estudiantes deberán entregar por correo electrónico, al profesor encargado del aula, el enlace al repositorio Github con el que hayan avanzado.

En referente a la entrega final, hay que entregar un único fichero que contenga el enlace Github, el cual no se podrá modificar posteriormente a la fecha de entrega, donde haya:

1. Una Wiki con los nombres de los componentes del grupo y una descripción de los ficheros.
2. Un documento PDF con las respuestas a las preguntas y los nombres de los componentes del grupo. Además, al final del documento, deberá aparecer la siguiente tabla de contribuciones al trabajo, la cual debe firmar cada integrante del grupo con sus iniciales. Las iniciales representan la confirmación de que el integrante ha participado en dicho apartado. Todos los integrantes deben participar en cada apartado, por lo que, idealmente, los apartados deberían estar firmados por todos los integrantes.

Contribuciones	**Firma**

Investigación previa	DFDO (dfdavila)	Mónica Alexandra Gómez Martínez
Redacción de las respuestas	DFDO (dfdavila)	Mónica Alexandra Gómez Martínez
Desarrollo código	DFDO (dfdavila)	Mónica Alexandra Gómez Martínez

3. Una carpeta con el código generado para analizar los datos.
4. El fichero CSV con los datos originales.
5. El fichero CSV con los datos finales analizados.

Este documento de entrega final de la Práctica 2 se debe entregar en el espacio de Entrega y Registro de AC del aula antes de las 23:59 del día 5 de enero. No se aceptarán entregas fuera de plazo.