Práctica 2: Limpieza y análisis de datos

Concesión de préstamos bancarios

Germán Yepes Calero & Mario Martínez García

Contents

1. Descripción del dataset.

La práctica consiste en el tratamiento de un conjunto de datos con el objetivo de aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis para la extracción de conclusiones.

El dataset recoge una serie de solicitudes de préstamos hipotecarios realizados a la compañía Dream Housing Finance. Nuestro objetivo a lo largo de la práctica será automatizar el proceso de eligibilidad de préstamos (en tiempo real) en función de los detalles del cliente proporcionados al completar el formulario de solicitud en línea. Los detalles suministrados en el registro serán los siguientes:

- Gender: género del cliente, masculino o femenino (Male/Female). Variable categórica.
- Married: estado civil del cliente, casado o no casado. Variable categórica.
- Dependents: número de personas dependientes del cliente, número de hijos o tutelados de los que se hace cargo (0, 1, 2, +3). Variable categórica.
- Education: grado de educación del cliente, graduado universitario o no (Graduate, Not Graduate). Variable categórica.
- Self_Employed: si el cliente es trabajador por cuenta propia o no, autónomo (Yes, No). Variable categórica.
- ApplicantIncome: ingresos mensuales del cliente, salario mensual. Variable numérica.
- CoapplicantIncome: ingresos mensuales del coaplicante. Variable numérica.
- LoanAmount: monto del préstamo, cantidad de dinero solicitada en préstamo. Los valores de la variable cantidades irán multiplicados por 1000 para calcular la cantidad real solicitada. Variable numérica.
- Loan_Amount_Term: plazo de devolución del préstamo, número de mensualidades en las que se completará la devolución del préstamo. Los plazos van desde el medio año hasta los 20 años. Variable categórica.
- Credit_History: historial crediticio, informe en el que se recogen los antecedentes financieros del cliente. Indica si el historial crediticio está limpio o aparece algún punto negativo en el mismo (1: historial positivo, 0: historial negativo). Variable categórica binaria.
- Property_Area: área urbana donde se sitúa la propiedad, zona rural, zona semiurbana o zona urbana (Rural, Semiurban, Urban). Variable categórica.
- Loan_Status:estado del préstamo, indica si el préstamos ha sido conferido o rechazado por la institución bancaria. Variable categórica y clasificatoria.

En definitiva tendremos 13 variables, 12 de las cuales contendrán información sobre los solicitantes del préstamo y actuarán como atributos independientes para predecir el resultado de la solicitud de préstamo. La última de las variables, el estado del préstamo (Loan_Status), nos indicará la decisión final que tomó la compañía con respecto a la concesión del préstamo. Por tanto, actuará como variable dependiente y será la que intentaremos predecir a partir de diversas herramientas estadísticas.

Los datos han sido extraídos de la página web *kaggle.com*, concretamente del dataset titulado Bank_loan, cuyo enlace es: https://www.kaggle.com/madhansing/bank-loan2. Entre el conjuntos de datos que ofrecía el dataset encontramos 3 archivos csv:

- madhantr.csv: contendrá los detalles de un subconjunto del proceso de elegibilidad para préstamos de clientes (614 para ser exactos), y revelará si son elegibles o no para el monto del préstamo estipulado. En caso afirmativo, los empleados de la empresa se dirigirán específicamente a estos clientes con "Loan_Status" positivo y procederán a tramitar el resto de condiciones del préstamo.
- madhante.csv: contendrá información similar al archivo anterior pero no revelará el estado del préstamo (Loan_Status) para ninguno de los 367 clientes que incluye. El objetivo principal del dataset será predecir correctamente los resultados para los clientes de este subconjunto.
- sample_submission_49d68Cx.csv: constará de una única columna que nos indicará el estado del préstamo para los clientes del archivo anterior. Estará consituido por el mismo número de filas, 367. Será útil en algoritmos de clasificación, permitiéndonos comprobar la precisión de nuestro modelo.

Dado que la práctica sigue un enfoque más estadístico que clasificatorio, agruparemos en un solo dataset las tres archivos, generando un dataset con un mayor número de muestras. Aunque en la creación del modelo final volveremos a dividir el modelo en datos de entrenamiento y test, la agrupación del conjunto de datos en un único fichero nos ayudará a que el previo estudio estadístico genere modelos más robustos y acordes a la realidad. El dataset final contará con un total de 981 registros, es decir, 981 solicitudes de préstamos hipotecarios, todos con el estado del préstamo (aceptado/rechazado) incluido.

A partir del dataset trataremos de encontrar tendencias en la elegibilidad de préstamos que nos indiquen que condiciones propician que un préstamo sea concedido o no. La aplicación principal del problema sería la creación de un "primer filtro" en el proceso de elegibilidad que permita descartar las solicitudes menos aptas, reduciendo el número de solicitudes a las que deberían enfrentarse el personal humano de la organización. La respuesta sería generada una vez el cliente rellena el formulario de solicitud en línea. Además, no solo eliminaría aquellas opciones poco viables, sino que también proporcionaría una valoración preliminar de la probabilidad de que dicho préstamo se lleve a cabo.

2. Integración y selección de los datos de interés a analizar.

Como comentamos en la descripción del dataset, unificaremos en una única base de datos los tres archivos correspondientes a las solicitudes de préstamos hipotecarios. También eliminaremos la primera columna donde se indica el código identificativo de cada préstamo (Por ejemplo, *LP001116*) pues no será relevante en nuestros análisis posteriores. El resto de atributos sí serán tomados en cuenta e integrados en el dataset final. Mostraremos un ejemplo de las primeras líneas del dataset.

```
#Cargamos los datos de entrenamiento
loan_train = read.csv("madfhantr.csv", header = TRUE)
#Cargamos los datos del test
loan_test = read.csv("madhante.csv", header = TRUE)
#Cargamos los resultados sobre la adjudicación del préstamo de los datos del test
loan_test_result = read.csv("sample_submission_49d68Cx.csv", header = TRUE)
#Añadimos una columna con los resultados de la adjudicación sobre los datos del test
loan_test["Loan_Status"]=loan_test_result["Loan_Status"]
#Unimos los datos de entrenamiento y de test
loans = rbind(loan_train, loan_test)
#Eliminamos la primera columna que nos muestra el código identificativo
#Yloans = loan_train
loans <- loans[, -1]

#Mostramos los primeros elementos
pandoc.table(head(loans))
```

Table 1: Table continues below

Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome
Male	No	0	Graduate	No	5849
Male	Yes	1	Graduate	No	4583
Male	Yes	0	Graduate	Yes	3000
Male	Yes	0	Not Graduate	No	2583
Male	No	0	Graduate	No	6000
Male	Yes	2	Graduate	Yes	5417

Table 2: Table continues below

CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	NA	360	1
1508	128	360	1
0	66	360	1
2358	120	360	1
0	141	360	1
4196	267	360	1

Property_Area	Loan_Status
Urban	Y
Rural	N
Urban	Y

Property_Area	Loan_Status
Urban	Y
Urban	Y
Urban	Y

Comprobamos la dimensión del dataset para ver si los tres conjuntos de datos han sido agrupados correctamente.

```
#Dimensión de los datos
dim(loans)
```

```
## [1] 981  12
```

El dataset constará de 981 filas (981 solicitudes de préstamo) y 12 columnas (12 atributos, contando el estado del préstamo). A continuación, identificaremos la clase de cada uno de los atributos (categóricos-factor, numéricos-numeric, enteros-integer...) y procederemos a cambiar a aquellas clases que consideremos oportunas para el análisis.

```
#Mostramos la clase de cada uno de los atributos de nuestra función
sapply(loans, function(x) class(x))
```

```
##            Gender           Married        Dependents         Education
##          "factor"          "factor"          "factor"          "factor"
##     Self_Employed   ApplicantIncome CoapplicantIncome        LoanAmount
##          "factor"         "integer"         "numeric"         "integer"
##  Loan_Amount_Term    Credit_History     Property_Area       Loan_Status
##         "integer"         "integer"          "factor"          "factor"
```

```
#Transformamos "ApplicantIncome" de integer a numeric
loans$ApplicantIncome = as.numeric(loans$ApplicantIncome)

#Transformamos "LoanAmount" de integer a numeric
loans$LoanAmount = as.numeric(loans$LoanAmount)

#Transformamos"Credit_History" de integer a factor
loans$Credit_History = as.factor(loans$Credit_History)

#Volvemos a mostrar las clases corregidas
sapply(loans, function(x) class(x))
```

```
##            Gender           Married        Dependents         Education
##          "factor"          "factor"          "factor"          "factor"
##     Self_Employed   ApplicantIncome CoapplicantIncome        LoanAmount
##          "factor"         "numeric"         "numeric"         "numeric"
##  Loan_Amount_Term    Credit_History     Property_Area       Loan_Status
##         "integer"          "factor"          "factor"          "factor"
```

Los cambios de tipo de atributo realizados han sido tres:

- Atributo ApplicantIncome: *integer* → *numeric*

- Atributo LoanAmount: $integer \rightarrow numeric$
- Atributo Credit_History: $integer \rightarrow factor$

Con esto queda concluida la integración y selección de los datos de interés.

3. Limpieza de los datos.

En el presente apartado llevaremos a cabo las tareas de limpieza propias de cualquier proyecto de ciencia de datos. El objetivo será preparar la base de datos para los procedimientos de análisis posteriores.

3.1. Gestión de elementos vacíos, elementos nulos y datos que contienen ceros.

Comenzaremos las tarea de limpieza comprobando la existencia de registros inusuales, como datos que contienen ceros o elementos vacíos. Primero identificaremos aquellos registros que contengan elementos vacíos, es decir, que estén en blanco. Para facilitar el proceso de limpieza sustituiremos estos valores vacíos por valores nulos ("NA").

```
#Número de datos que continen ceros por campo
sapply(loans, function(x) sum(x == ""))
```

```
##            Gender           Married        Dependents         Education
##                24                 3                25                 0
##     Self_Employed   ApplicantIncome CoapplicantIncome        LoanAmount
##                55                 0                 0                NA
##  Loan_Amount_Term    Credit_History     Property_Area       Loan_Status
##                NA                NA                 0                 0
```

```
loans$Gender[loans$Gender==""]=NA
loans$Married[loans$Married==""]=NA
loans$Dependents[loans$Dependents==""]=NA
loans$Self_Employed[loans$Self_Employed==""]=NA
```

A continuación, obtendremos el números de elementos nulos que aparecen en cada uno de los atributos.

```
#Número de valores desconocidos por campo
sapply(loans, function(x) sum(is.na(x)))
```

```
##            Gender           Married        Dependents         Education
##                24                 3                25                 0
##     Self_Employed   ApplicantIncome CoapplicantIncome        LoanAmount
##                55                 0                 0                27
##  Loan_Amount_Term    Credit_History     Property_Area       Loan_Status
##                20                79                 0                 0
```

También comprobaremos si existen datos que contengan ceros que puedan significar errores en el dataset.

```
#Número de datos que continen ceros por campo
sapply(loans, function(x) sum(x == "0"))
```

```
##            Gender           Married        Dependents         Education
##                NA                NA                NA                 0
##     Self_Employed   ApplicantIncome CoapplicantIncome        LoanAmount
##                NA                 2               429                NA
##  Loan_Amount_Term    Credit_History     Property_Area       Loan_Status
##                NA                NA                 0                 0
```

Solo aparecen ceros en los atributos ApplicantIncome y CoapplicantIncome, los cuales indican las ganancias mensuales del solicitante del préstamo y de su co-solicitante. El cero en este caso significa que el cliente no dispone de ingresos mensuales, lo que puede perfectamente ocurrir en la realidad. Por tanto, no descartaremos ninguno de los datos que contienen ceros.

Veamos que medidas podemos tomar para eliminar los valores nulos de los distintos atributos. Para los valores nulos presentes en los atributos Loan_Amount_Term y Married, 20 y 3 respectivamente, eliminaremos las filas correspondientes. Al no existir un método adecuado para establecer valores para estos atributos, decidimos borrar los registros que incluyen datos desconocidos en estos campos. Además, la eliminación de 23 líneas no afecta en gran medida al tamaño de nuestra muestra.

```
suppressWarnings(suppressMessages(library(VIM)))

#Eliminar las filas con valores nulos sobre el campo "Loan_Amount_Term" (20)
loans <- loans[-which(is.na(loans["Loan_Amount_Term"])),]
#Eliminar las filas con valores nulos sobre el campo "Married" (3)
loans <- loans[-which(is.na(loans["Married"])),]
```

Para los valores nulos presentes en los atributos LoanAmount, Credit_History y Dependents; 27, 79 y 25, respectivamente, emplearemos un método de imputación de valores basado en la similitud o diferencia entre los registros: la imputación basada en k vecinos más próximos (en inglés, kNN-imputation).

La elección de esta alternativa se realiza bajo la hipótesis de que nuestros registros guardan cierta relación. A la hora de establecer relaciones entre los tres atributos y el resto hemos considerado que:

- LoanAmount: puede ser calculada en función del número de dependientes (Dependents), los salarios mensuales de los solicitantes (ApplicantIncome y CoapplicantIncome), los plazos de pago (Loan_Amount_Term) y la zona en la que se sitúa la propiedad (Property_Area). En función de si estos atributos toman unos valores u otros, el valor más frecuente para el monto del préstamo será el que sustituya al valor vacío.
- Credit_History: puede ser calculado en función del estado civil (Married), la educación recibida (Education), los salarios mensuales de los solicitantes (ApplicantIncome y CoapplicantIncome) y el estado del préstamo (Loan_Status).
- Dependents: puede ser calculado según el estado civil (Married).

Aunque la imputación no sea del todo precisa en algunos casos, es mejor trabajar con datos "aproximados" que con los propios elementos vacíos, ya que obtendremos análisis con menor margen de error.

```
#Imputación de los valores mediante la función kNN() del paquete VIM
loans$LoanAmount <- kNN(loans)$LoanAmount
loans$Credit_History <- kNN(loans)$Credit_History
loans$Dependents <- kNN(loans)$Dependents
```

Por último, para los valores nulos presentes en los atributos Gender y Self_Employed, 24 y 55 respectivamente, los sustituiremos directamente por la moda de cada uno de los atributos. En el caso del género (Gender), tendremos que un 80% de solicitantes tienen sexo masculino, por lo que supondremos que los registros vacíos también se corresponderán a este sexo.

En el caso del tipo de trabajo (Self_Employed), tendremos que un 90% de los solicitantes no trabajan por cuenta propia, por lo que supondremos que los registros vacíos también se corresponderán a este tipo de trabajadores.

```r
#Imputación de los valores mediante la moda
loans$Gender[is.na(loans$Gender) == TRUE] = "Male"
loans$Self_Employed[is.na(loans$Self_Employed) == TRUE] = "No"
```

Finalmente, eliminaremos de los atributos categóricos la clase vacía ("") que R crea directamente cuando existen valores vacíos en ellos. Comprobamos que realizados todos los cambios nuestro dataset no presenta elementos erróneos.

```r
#Reestructuración de los factores sobre los siguientes campos
loans$Gender<- as.factor(as.character(loans$Gender))
loans$Married<- as.factor(as.character(loans$Married))
loans$Dependents<- as.factor(as.character(loans$Dependents))
loans$Self_Employed<- as.factor(as.character(loans$Self_Employed))
#Comprobamos las modificaciones realizadas
sapply(loans, function(x) sum(is.na(x)))
```

```
##             Gender            Married         Dependents          Education
##                  0                  0                  0                  0
##      Self_Employed    ApplicantIncome CoapplicantIncome         LoanAmount
##                  0                  0                  0                  0
##   Loan_Amount_Term     Credit_History      Property_Area        Loan_Status
##                  0                  0                  0                  0
```

Para conocer mejor como han quedado distribuidos los datos en cada uno de nuestros atributos haremos un resumen estadístico de los mismos.

```r
#Realizamos un sumario
pandoc.table(summary(loans))
```

Table 4: Table continues below

Gender	Married	Dependents	Education	Self_Employed
Female:178	No :341	0 :547	Graduate :748	No :841
Male :780	Yes:617	1 :158	Not Graduate:210	Yes:117
NA	NA	2 :164	NA	NA
NA	NA	3+: 89	NA	NA
NA	NA	NA	NA	NA
NA	NA	NA	NA	NA

Table 5: Table continues below

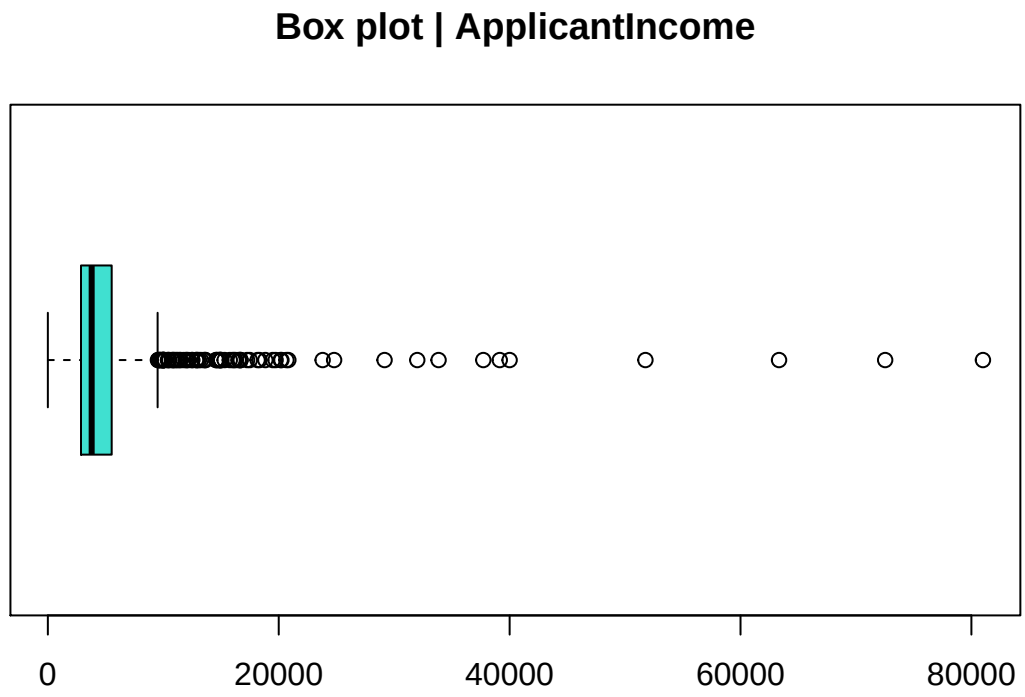
ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term
Min. : 0	Min. : 0	Min. : 9.0	Min. : 6.0
1st Qu.: 2874	1st Qu.: 0	1st Qu.:100.0	1st Qu.:360.0
Median : 3796	Median : 1094	Median :126.5	Median :360.0
Mean : 5193	Mean : 1597	Mean :142.5	Mean :342.1
3rd Qu.: 5526	3rd Qu.: 2335	3rd Qu.:161.8	3rd Qu.:360.0
Max. :81000	Max. :41667	Max. :700.0	Max. :480.0

Credit_History	Property_Area	Loan_Status
0:145	Rural :284	N:547
1:813	Semiurban:343	Y:411
NA	Urban :331	NA
NA	NA	NA
NA	NA	NA
NA	NA	NA

3.2. Identificación y tratamiento de valores extremos.

En este apartado nos centraremos en los valores extremos, aquellos valores que presenten desviaciones respecto al valor esperado de la distribución de nuestros datos. El estudio de los valores extremos se realizará únicamente sobre las variables numéricas. Una vez los identifiquemos a partir de diagramas de caja ("Boxplot"), nos plantearemos que es conveniente hacer con los mismos. El objetivo final será evitar que estos datos inusuales alteren nuestros análisis, obteniendo resultados que no se corresponden con la realidad.

```
#Estudiamos los valores extremos sobre ApplicantIncome
boxplot(loans$ApplicantIncome, main="Box plot | ApplicantIncome", col="turquoise", horizontal = TRUE)
```




```
#Mostamos los valores extremos sobre ApplicantIncome
boxplot.stats(loans$ApplicantIncome)$out
```

```
##  [1] 12841  9560 12500 11500 10750 13650 11417 14583 10408 23803 10513 20166
```

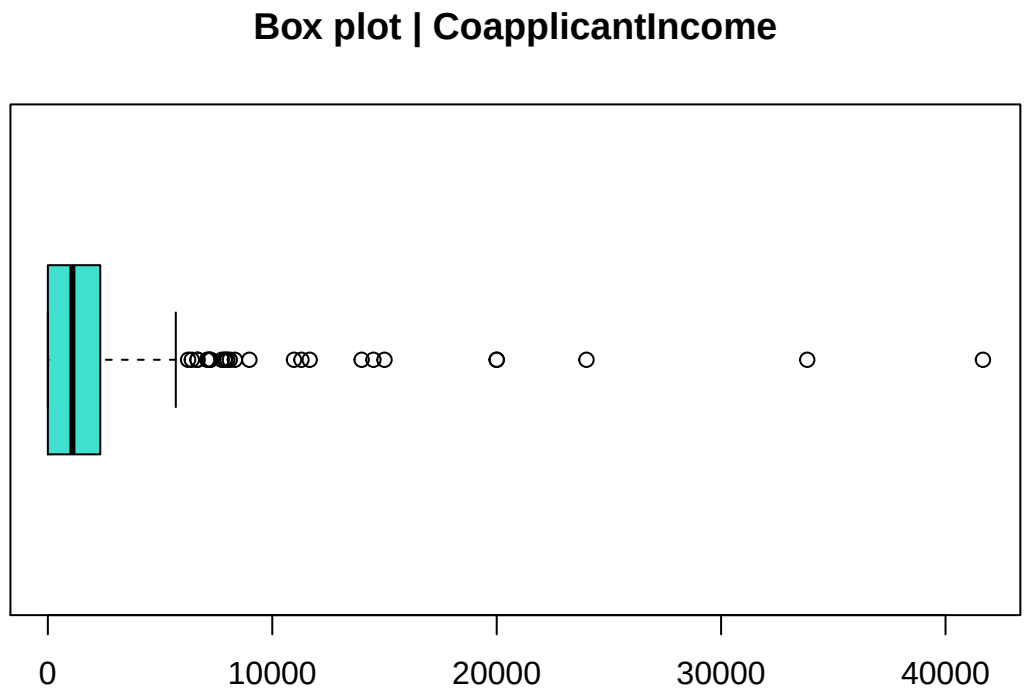
```
## [13] 14999 11757 14866 10000 39999  9538 51763 33846 39147 12000 11000 10000
## [25]  9703 16250 14683 11146 14583 20667 20233 10000 15000 63337  9833 19730
## [37] 15759 81000 14880 12876 10416 37719 16692 16525 16667 10833 18333 17263
## [49] 20833 13262 17500 11250 18165 10139 19484 16666 16120  9963 12000 13633
## [61] 12173 72529 13518  9719 12500 32000 10890 12941 15312 13083 10000 14911
## [73] 10000 18840 24797 29167 10000 14987 16000  9699
```

```r
#Calculamos el total de valores extremos sobre ApplicantIncome
length(boxplot.stats(loans$ApplicantIncome)$out)
```

```
## [1] 80
```

Podemos apreciar una gran cantidad de valores extremos por encima de los valores habituales en el atributo ApplicantIncome. Pese a que la mayoría de salarios mensuales de los solicitantes se situarán entre 0 y 10000, tendremos salarios de hasta 81000, y un número importante de salarios entre los 10000 y 20000. Tendremos un total de 80 valores outliers para este atributo. Aunque estos salarios puedan parecer desorbitados, bien podrían ser datos reales, por tanto, conservaremos estos valores extremos pues representarán un grupo de clientes que enriquecerán la varianza de nuestro dataset.

```r
#Estudiamos los valores extremos sobre CoapplicantIncome
boxplot(loans$CoapplicantIncome, main="Box plot | CoapplicantIncome", col="turquoise", horizontal = TRU
```




```r
#Mostamos los valores extremos sobre CoapplicantIncome
boxplot.stats(loans$CoapplicantIncome)$out
```

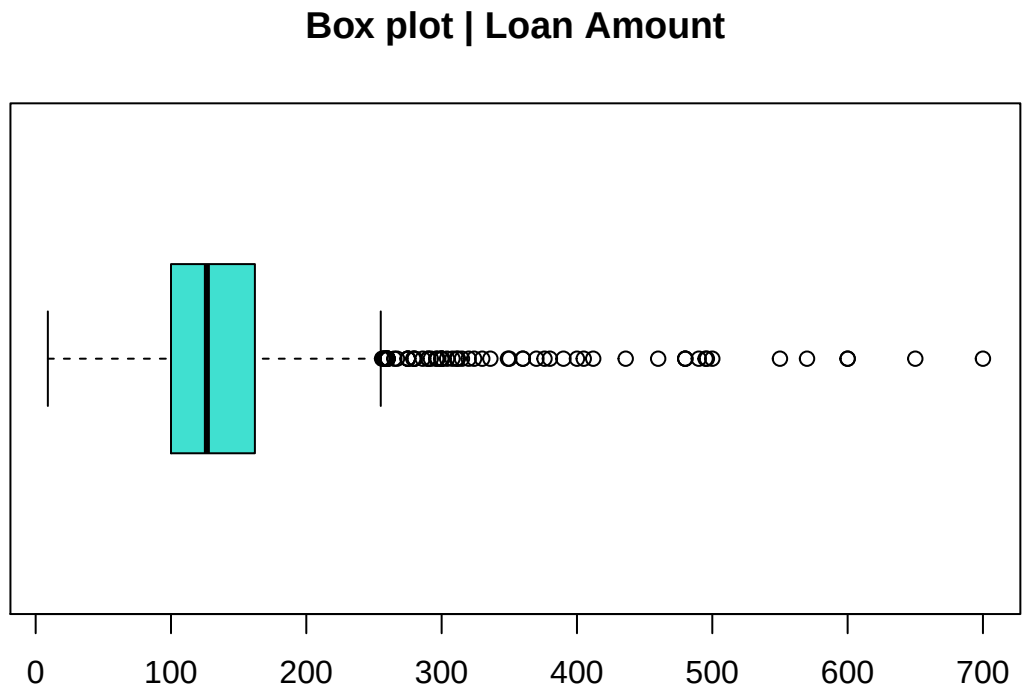
```
##  [1] 10968  8106  7210  8980  7750 11300  7250  7101  6250  7873 20000 20000
## [13]  8333  6667  6666  7166 33837 41667  7916 24000  8000  6414 14507 13983
## [25] 11666 15000
```

```
#Calculamos el total de valores extremos sobre CoapplicantIncome
length(boxplot.stats(loans$CoapplicantIncome)$out)
```

```
## [1] 26
```

Para el atributo CoapplicantIncome presenciamos un comportamiento parecido con salarios menores. La mayoría de salarios mensuales de los solicitantes se situarán entre 0 y 8000, apareciendo salarios de hasta 42000, y un número importante de salarios entre los 8000 y 10000. Tendremos un total de 26 valores outliers para este atributo. Al igual que con el atributo anterior, estos valores extremos permanecerán sin modificaciones en el análisis

```
#Estudiamos los valores extremos sobre Loan Amount
boxplot(loans$LoanAmount, main="Box plot | Loan Amount", col="turquoise", horizontal = TRUE)
```




```
#Mostamos los valores extremos sobre LoanAmount
boxplot.stats(loans$LoanAmount)$out
```

```
##  [1] 267 349 315 320 286 258 312 265 259 370 650 290 600 275 700 495 260 280 279
## [20] 304 330 436 257 480 300 376 490 259 308 570 380 296 275 360 405 500 480 311
## [39] 480 400 324 260 600 258 275 292 260 350 496 280 300 290 275 360 257 390 256
## [58] 300 550 260 336 412 460 297 300 260
```

```r
#Calculamos el total de valores extremos sobre LoanAmount
length(boxplot.stats(loans$LoanAmount)$out)
```

```
## [1] 66
```

El diagrama de caja del atributo Loan Amount nos muestra un total de 66 valores extremos. En la mayoría de préstamos se solicitarán cantidades de entre 100000 y 250000. Tendremos un número de préstamos situados entre los 250000 y los 400000, llegando a alcanzar incluso los 700000. Vemos que estos valores no representan cantidades muy alejadas de la realidad cuando hablamos de préstamos inmobiliarios. Dicho esto, emplearemos todos los valores extremos extraídos en el análisis.

Para concluir la limpieza de los datos, crearemos un nuevo fichero csv donde guardaremos el dataset modificado. Será en esta versión del dataset sobre la que ejecutaremos las diversas pruebas estadísticas en busca de conclusiones.

```r
#Creamos un fichero csv con los datos limpios
write.csv(loans, "Loans_clean.csv")
```

4. Análisis de los datos.

En la siguiente sección someteremos a nuestro dataset a una serie de análisis estadísticos con el objetivo de extraer conclusiones que nos ayuden a resolver el problema planteado. Los modelos girarán principalmente entorno a la concesión o no concesión de préstamo (Loan_Status), permitiéndonos crear ese modelo de clasificación de elegibilidad de las solicitudes.

Primero, analizaremos las correlaciones entre las variables cuantitativas, comprobando si existe relación entre las mismas y si aparecen asociaciones con la concesión del préstamo. Para aprender más sobre que elementos influyen en la eligibilidad del préstamo realizaremos diversos contrastes de hipótesis, estableciendo que condiciones favorecen que se conceda (salarios altos de los solicitantes, duración y cantidad del préstamo...)

A continuación, nos centraremos en los atributos cualitativos, creando tablas de contingencia que nos permitan discernir cuando es más probable que se garantice la elegibilidad del préstamo. En este caso compararemos las categorías de cada atributo (generalmente binarios) y veremos cuales propician una mejor aceptación del préstamo por parte del banco (género, estado civil, autónomo...).

Por último, desarrollaremos un conjunto de modelos de regresión en los que introduciremos distintas combinaciones de variables, quedándonos con el modelo que mejor represente nuestro dataset. A partir de este modelo, realizaremos una serie de predicciones que nos permitan averiguar cúal es la probabilidad aproximada de que la solicitud sea concedida en función de distintos parámetros.

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

Iniciamos las tareas de análisis seleccionando aquellos grupos de datos que procederemos a estudiar. En esta fase estableceremos un esquema de los procedimientos que llevaremos a cabo según el tipo de datos con el que estamos tratando. El parámetro que estudiaremos en función del resto de atributos será la variable dependiente, el estado del préstamo (Loan_Status).

Comenzamos escogiendo los grupos que queremos analizar y comparar.

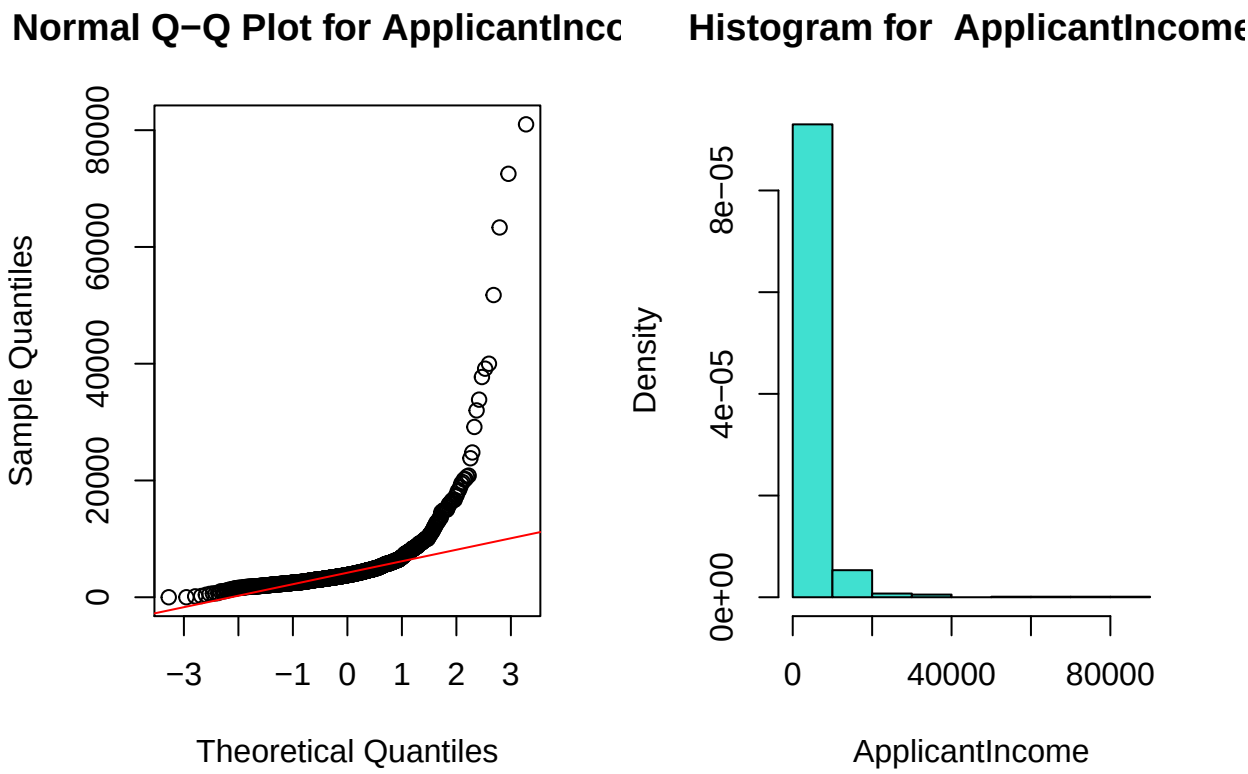
```
#Grupo estado del préstamo
loans.Loan_Status_Y <- loans[loans$Loan_Status == "Y",]
loans.Loan_Status_N <- loans[loans$Loan_Status == "N",]
```

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

Procedemos a comprobar la normalidad de nuestros atributos, comparando su similitud con una distribución normal ideal que tratase de explicar el total de la varianza. Realizaremos el test shapiro.test para cada atributo numérico, así como la representación de su histograma y una gráfica QQplot de cuantiles que nos ayude a interpretar los resultados.

Empezamos comprobando la normalidad del atributo ApplicantIncome, ganacias mensuales del solicitante.

```
par(mfrow=c(1,2))
#par(mar = c(5, 5, 2, 2))
i = "ApplicantIncome"
qqnorm(loans[,i], main = paste("Normal Q-Q Plot for", i))
qqline(loans[,i], col="red")
hist(loans[,i],  main=paste("Histogram for ", i),  xlab=i, freq = FALSE,  col="turquoise")
```



El gráfico QQplot compara los residuos del modelo con los valores de una variable que se distribuye normalmente. De esta forma, nos permite observar la proximidad entre la distribución del conjunto de datos con respecto a la distribución ideal evaluada en el modelo. La recta roja representa el modelo evaluado ideal y los puntos los datos del conjunto. Como podemos apreciar, los salarios más bajos se sitúan en la línea o muy próximos a la misma. Sin embargo, para salarios altos tendremos puntos que se alejan de la recta, siendo imposibles de explicar a partir de una distribución normal. En el histograma es fácil apreciar como la distribución del atributo se aleja de una distribución normal.

Del análisis de las gráficas podemos concluir que los valores bajos y centrales parecen seguir una distribución normal pero que los altos no se ajustan bien a una distribución de este tipo. Veamos numéricamente como queda la normalidad de nuestra variable.

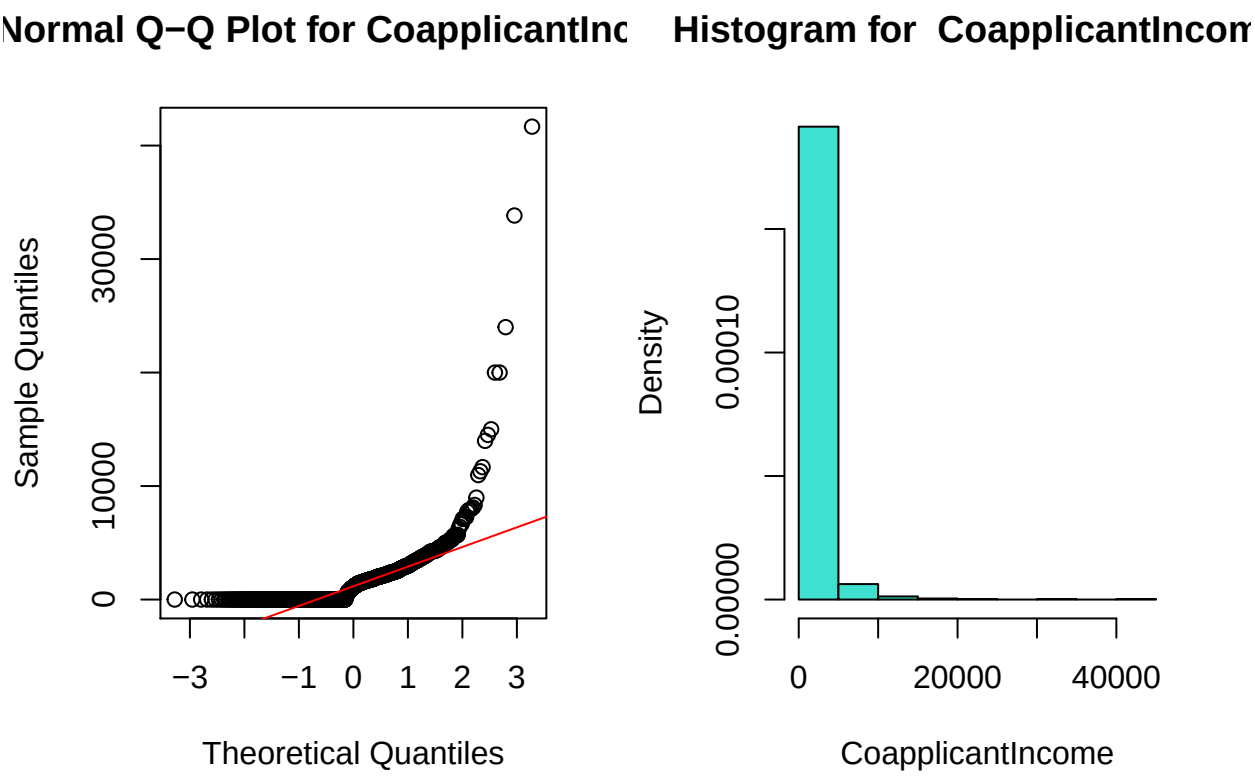
```
shapiro.test(loans[,i])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  loans[, i]
## W = 0.45833, p-value < 2.2e-16
```

Obtenemos un p-valor inferior al nivel de significación que imponemos en nuestro contraste, 0.05. Por tanto, rechazamos la hipótesis nula y aceptamos la hipótesis alternativa. De este modo, no podemos suponer que existe relación entre la variable ApplicantIncome y una distribución normal.

Proseguimos analizando la normalidad de la variable CoapplicantIncome.

```r
par(mfrow=c(1,2))
i = "CoapplicantIncome"
qqnorm(loans[,i], main = paste("Normal Q-Q Plot for", i))
qqline(loans[,i], col="red")
hist(loans[,i],  main=paste("Histogram for ", i),  xlab=i, freq = FALSE,  col="turquoise")
```



Obtenemos resultados muy similares, aunque en este caso ni siquiera los salarios bajos se ajustan de manera correcta a la distribución normal, situándose fuera de la recta. Los valores altos siguen presentando grandes desviaciones con respecto a la normalidad establecida. El histograma solidifica lo visto, no podemos aproximar de forma correcta la distribución de la variable CoapplicantIncome a una distribución normal. Comprobemos el resultado numérico aplicando el test shapiro.

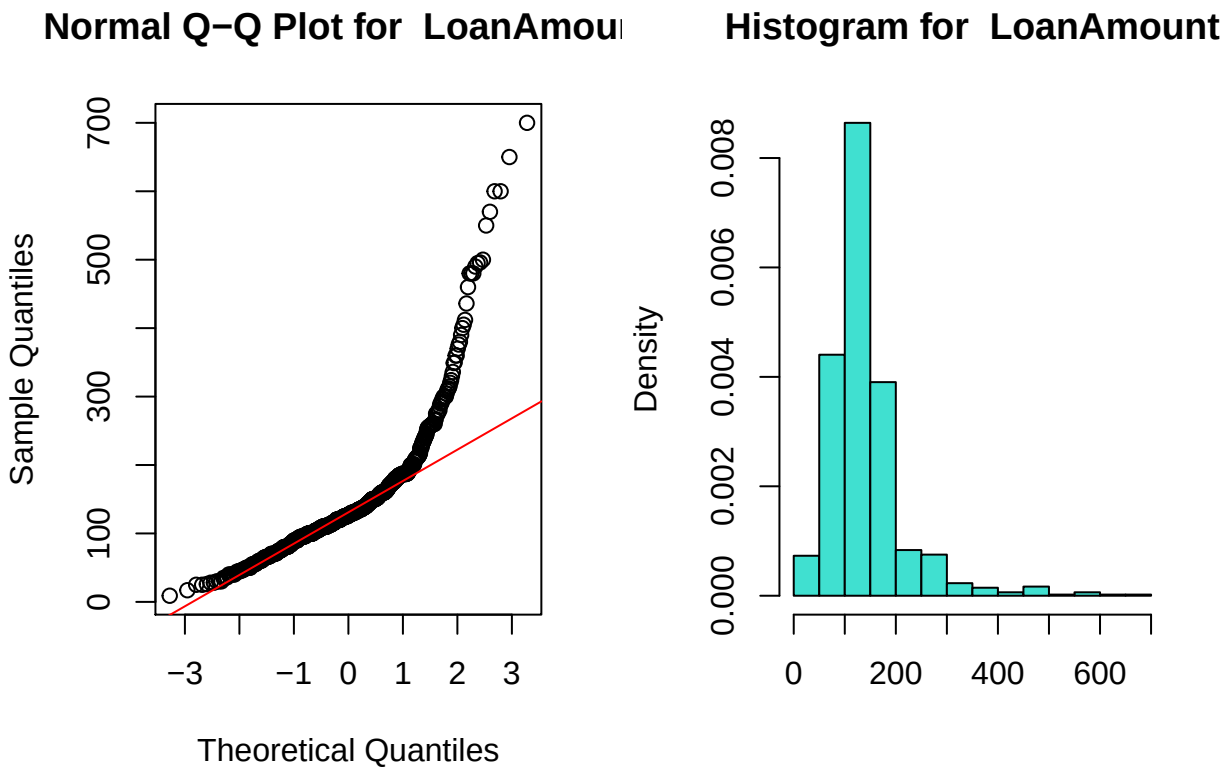
```r
shapiro.test(loans[,i])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  loans[, i]
## W = 0.5192, p-value < 2.2e-16
```

Obtenemos un p-valor inferior al nivel de significación que imponemos en nuestro contraste, 0.05. Por tanto, rechazamos la hipótesis nula y aceptamos la hipótesis alternativa. De este modo, no podemos suponer que existe relación entre la variable CoapplicantIncome y una distribución normal.

Procedemos del mismo modo con el atributo LoanAmount, cantidad solicitada en el préstamo.

```
par(mfrow=c(1,2))
i = "LoanAmount"
qqnorm(loans[,i],main = paste("Normal Q-Q Plot for ", i))
qqline(loans[,i],col="red")
hist(loans[,i],  main=paste("Histogram for ", i),  xlab=colnames(loans)[i], freq = FALSE, col="turquois
```



Pese a que los montos de préstamo bajos y medios quedan bien representados en la recta, las cantidades más altas vuelven a impedir que podamos considerar la distribución normal del atributo LoanAmount. Podemos apreciar también este problema en el histograma, con una cola derecha que no se corresponde con la cola izquierda, condición necesaria en una distribución normal. Veamos que resultados numéricos arroja el test sobre estos datos.

```
shapiro.test(loans[,i])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  loans[, i]
## W = 0.7779, p-value < 2.2e-16
```

Obtenemos un p-valor inferior al nivel de significación que imponemos en nuestro contraste, 0.05. Por tanto, rechazamos la hipótesis nula y aceptamos la hipótesis alternativa. De este modo, no podemos suponer que existe relación entre la variable LoanAmount y una distribución normal.

Una vez realizado las comprobaciones correspondientes de normalidad analizaremos la homogeneidad de la varianza en las variables cuantitativas mediante el test de Fligner-Killeen. Se trata de un test no paramétrico

que compara las varianzas basándose en la mediana. Es una alternativa cuando no se cumple la condición de normalidad en las muestras, como es el caso.

En función de los resultados arrojados, decidiremos si aplicar pruebas por contraste de hipótesis paramétricas como la prueba t de Student, o pruebas no paramétricas como la prueba Mann-Whitney. Para los casos en los que las varianzas de dichas variables permanecen constantes a lo largo del rango observado (resultado del test positivo, p-valor > 0.05) optaremos por test de t de Student. Para el resto de casos, emplearemos el test de Mann-Whitney.

Además, podremos concluir si existe una diferencia entre las variaciones en la muestra, es decir, si la aceptación de elegibilidad del préstamo varía según los valores tomados en estas variables.

```
loans["Loan_Status_numeric"] = lapply(loans["Loan_Status"], as.numeric)
fligner.test(Loan_Status_numeric ~ ApplicantIncome, data = loans)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Loan_Status_numeric by ApplicantIncome
## Fligner-Killeen:med chi-squared = 532.31, df = 731, p-value = 1
```

Obtenemos un p-valor superior al nivel de significación que imponemos en nuestro contraste, 0.05. Por tanto, no rechazamos la hipótesis nula y podemos suponer que se cumple la homogeneidad de la varianza en la variable "ApplicantIncome".

```
fligner.test(Loan_Status_numeric ~ CoapplicantIncome, data = loans)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Loan_Status_numeric by CoapplicantIncome
## Fligner-Killeen:med chi-squared = 265.02, df = 425, p-value = 1
```

Obtenemos un p-valor igual a uno. Por tanto, no rechazamos la hipótesis nula y podemos suponer que se cumple la homogeneidad de la varianza en la variable "CopplicantIncome".

```
fligner.test(Loan_Status_numeric ~ LoanAmount, data = loans)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Loan_Status_numeric by LoanAmount
## Fligner-Killeen:med chi-squared = 155.68, df = 231, p-value = 1
```

Obtenemos un p-valor superior al nivel de significación que imponemos en nuestro contraste, 0.05. Por tanto, no rechazamos la hipótesis nula y podemos suponer que se cumple la homogeneidad de la varianza en la variable "LoanAmount".

```
fligner.test(Loan_Status_numeric ~ Loan_Amount_Term, data = loans)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Loan_Status_numeric by Loan_Amount_Term
## Fligner-Killeen:med chi-squared = 8.8898, df = 11, p-value = 0.6321
```

Obtenemos un p-valor superior al nivel de significación que imponemos en nuestro contraste, 0.05. Por tanto, no rechazamos la hipótesis nula y podemos suponer que se cumple la homogeneidad de la varianza en la variable "Loan_Amount_Term".

La existencia o no existencia de homogeneidad entre las varianzas nos indicará que atributos permiten optimizar la labor de clasificación de elegibilidad de los préstamos. Con aquellos atributos que muestren varianzas homogéneas no seremos capaces de discernir entre la aceptación o rechazo del préstamo. En cambio, con los atributos que muestran diferencias en las varianzas sí podremos identificar en mayor medida el estado de concesión del préstamo.

Los resultados nos muestran que todas las variables numéricas presentan homocedasticidad, es decir, igualdad de varianzas entre los grupos que se han comparado. Por tanto, a la hora de realizar contrastes de hipótesis sobre estas variables aplicaremos test paramétricos como t de Student.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos.

A lo largo del siguiente apartado intentaremos establecer qué variables tendrán más peso sobre nuestro modelo final, o dicho de otra forma, qué variables están más relacionadas con la que consideramos como nuestra variable clasificadora "Loan_Status". Este intenso estudio sobre cada una de las variables se realizará con el objetivo de construir un modelo capaz de predecir la concesión o denegación del préstamo a partir de unas determinadas variables de entrada. Con el fin de conocer qué variables tendrán más peso sobre nuestro modelo, trataremos por separado las variables cuantitativas y las variables cualitativas del fichero de datos.

4.3.1. Variables Cuantitativas

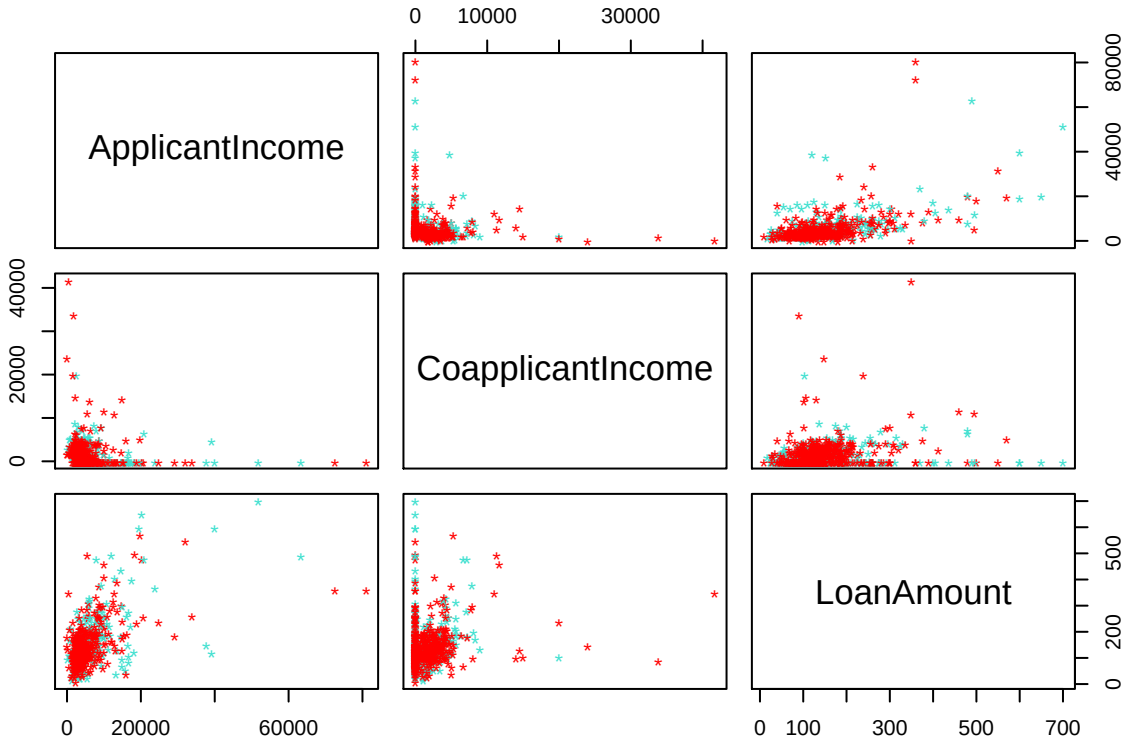
Las variables involucradas en este estudio son aquellas que han sido categorizadas como numéricas: "ApplicantIncome", "CoapplicantIncome", "LoanAmount" y "Loan_Amount_Term". Al tratarse de variables continuas realizaremos un gráfico de dispersión con el fin de ver la distribución de los datos así como las posibles correlaciones. Por otro lado, con el objetivo de comparar las características sobre la concesión o denegación del préstamo en una variable concreta, recurriremos al contraste de hipótesis.

Estudio de las correlaciones.

Mediante esta gráfica de dispersión podremos ver a simple vista si existe algún tipo de relación apreciable entre alguna pareja de datos numéricos. Se seleccionarán únicamente las variables "ApplicantIncome", "CoapplicantIncome" y "LoanAmount". Aunque "Loan_Amount_Term" es una variable numérica no se tendrá en cuenta sobre el gráfico de dispersión al disponer de únicamente 12 elementos distintos.

```
#Seleccionamos las variables cuantitavas sobre las que queremos ver su distribución
reduced=loans[c("ApplicantIncome", "CoapplicantIncome", "LoanAmount","Loan_Status")]

#Graficamos los distintos pares de variables
plot(reduced[1:3], pch='*',col=c("red", "turquoise")[unclass(reduced[,4])])
```



Aunque el gráfico de dispersión no nos aporta evidencias de relaciones entre los datos, calcularemos el coeficiente de correlación para obtener más información. Cabe destacar que el color rojo representa los préstamos rechazados y el color azul los préstamos concedidos.

Al no tener los datos distribuidos de forma normal, se utilizará el coeficiente de correlación de "Spearman" que además será robusto ante los valores outliers que hemos observado sobre los boxplot de los apartados anteriores.

```
#Mostramos las correlaciones de spearman de los diferentes pares de variables
cor(loans$ApplicantIncome, loans$CoapplicantIncome, method = "spearman")
```

```
## [1] -0.3378607
```

```
cor(loans$ApplicantIncome, loans$LoanAmount, method = "spearman")
```

```
## [1] 0.4766704
```

```
cor(loans$CoapplicantIncome, loans$LoanAmount, method = "spearman")
```

```
## [1] 0.2325266
```

Los coeficientes de correlación de "Spearman" entre los pares de variables presentes en el gráfico de dispersión no nos indicarán ninguna relación relevante entre los datos. Podremos ver la existencia de una ligera tendencia negativa entre "ApplicantIncome" y "CoapplicantIncome", indicándonos que a mayores ganancias

del solicitante, menores serán las ganancias del co-solicitante. Por otra parte, sobre los otros dos pares de variables tendremos asociaciones positivas, es decir: a mayores ganancias del solicitante o del co-solicitante, mayor será la cantidad del préstamo.

Cabe destacar que estas relaciones son de muy poco peso y no podremos considerarlas como relevantes sobre el estudio. Veremos como es la correlación entre nuestra variable clasificadora "Loan_Status" y las variables numéricas.

```
#Estudiamos la correlación de spearman entre nuestra variable predictora "Loan_Status" y las variables
cor(loans$Loan_Status_numeric, loans$ApplicantIncome, method = "spearman")
```

```
## [1] 0.01768154
```

```
cor(loans$Loan_Status_numeric, loans$CoapplicantIncome, method = "spearman")
```

```
## [1] 0.01548683
```

```
cor(loans$Loan_Status_numeric, loans$LoanAmount, method = "spearman")
```

```
## [1] -0.009766163
```

```
cor(loans$Loan_Status_numeric, loans$Loan_Amount_Term , method = "spearman")
```

```
## [1] -0.01930511
```

Obtenemos valores incluso más bajos que sobre los pares de variables anteriores. Únicamente podremos extraer una ligera relación positiva entre las variables "Loan_Status"-"ApplicantIncome" y "Loan_Status"-"CoapplicantIncome". Sobre los pares de variables "Loan_Status"-"LoanAmount" y "Loan_Status"-"Loan_Amount_Term" se consigue una ligera relación negativa.

Tras ver como las variables cuantitativas apenas muestran relación entre ellas, pasaremos a realizar contrastes de hipótesis que nos permitan conseguir evidencias sobre los datos. A pesar de que nuestros datos no se encuentran distribuidos de forma normal, las muestras utilizadas sobre cada uno de los contrastes serán lo suficientemente grandes (n>30), por lo que la distribución de la media muestral será aproximadamente una normal. Por otra parte, cada una de nuestras variables numéricas es homocedástica como vimos mediante el test no paramétrico de Fligner Killen. Al ser homocedástica y cumplirse el teorema del límite central, podremos utilizar un método paramétrico en nuestros contrastes como el t.test.

Cabe destacar que sobre todos los contrastes de hipótesis tomaremos $\alpha = 0.05$. Realizaremos 4 contrastes de hipótesis unilaterales que nos determinen sobre cada variable cuantitativa si existe una diferencia significativa entre la aceptación y el rechazo del préstamo y de haberla, a quién beneficiaría.

Contraste entre las ganancias medias de los solicitantes que reciben el préstamo y los solicitantes que no lo reciben.

Sobre este contraste veremos si las ganancias medias de los solicitantes que reciben el préstamo son significativamente superiores a las ganancias medias de los solicitantes que no reciben el préstamo. Realizaremos un contraste unilateral.

$$\begin{cases} H_0 : \mu_{App.Income.Yes} = \mu_{App.Income.No} \\ H_1 : \mu_{App.Income.Yes} > \mu_{App.Income.No} \end{cases}$$

```r
#Clasificamos las ganancias de los solicitantes que si reciben el préstamo de los que no
loans.Loan_Status_Y.apincome<- loans[loans$Loan_Status == "Y",]$ApplicantIncome
loans.Loan_Status_N.apincome<- loans[loans$Loan_Status == "N",]$ApplicantIncome

#Realizamos el t.test
t.test(loans.Loan_Status_Y.apincome, loans.Loan_Status_N.apincome, alternative = "greater", conf.level =
```

```
##
##  Welch Two Sample t-test
##
## data:  loans.Loan_Status_Y.apincome and loans.Loan_Status_N.apincome
## t = 0.97789, df = 871.86, p-value = 0.1642
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -251.9144       Inf
## sample estimates:
## mean of x mean of y
##  5403.569  5035.181
```

El p-valor obtenido es superior al nivel de significancia establecido de 0.05, por lo tanto no rechazaremos la hipótesis nula y consideraremos que las ganancias medias de los solicitantes que reciben el préstamo son muy similares a las ganancias medias de los solicitantes que no reciben el préstamo.

Cabe destacar que las medias calculadas por el test sobre ambos grupos muestran una leve diferencia de 400 a pesar de no rechazar la hipótesis nula.

Contraste entre las ganancias medias de los co-solicitantes que reciben el préstamos y los co-solicitantes que no lo reciben.

Intentaremos determinar si las ganancias medias de los co-solicitantes que reciben el préstamo son significativamente superiores a las ganancias medias de los co-solicitantes que no reciben el préstamo. Realizaremos un contraste unilateral.

$$\begin{cases} H_0 : \mu_{CoApp.Income.Yes} = \mu_{CoApp.Income.No} \\ H_1 : \mu_{CoApp.Income.Yes} > \mu_{CoApp.Income.No} \end{cases}$$

```r
#Clasificamos las ganancias de los co-solicitantes que si reciben el préstamo de los que no
loans.Loan_Status_Y.coapincome<- loans[loans$Loan_Status == "Y",]$CoapplicantIncome
loans.Loan_Status_N.coapincome<- loans[loans$Loan_Status == "N",]$CoapplicantIncome

#Realizamos el t.test
t.test(loans.Loan_Status_Y.coapincome, loans.Loan_Status_N.coapincome, alternative = "greater", conf.le
```

```
##
##  Welch Two Sample t-test
##
## data:  loans.Loan_Status_Y.coapincome and loans.Loan_Status_N.coapincome
## t = -1.0004, df = 915.43, p-value = 0.8413
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -442.9459       Inf
## sample estimates:
## mean of x mean of y
##  1501.272  1668.680
```

En este caso, dado que el p-valor es superior al nivel de significancia (0.05) no se dispondrá de las evidencias suficientes para considerar la existencia de una diferencia considerable entre los co-solicitantes de los préstamos. Por ello, no rechazaremos la hipótesis nula donde establecíamos que las ganancias medias de los co-solicitantes que reciben el préstamo son similares a las ganancias medias de los co-solicitantes que no reciben el préstamo.

Contraste entre la cantidad media de los préstamos aceptados y los rechazados.

En este caso tomaremos la cantidad del préstamo como variable cuantitativa e intentaremos comprobar si la cantidad media de los préstamos aceptados es significativamente superior a la cantidad media de los préstamos rechazados. Al igual que sobre los dos anteriores realizaremos un contraste unilateral.

$$\begin{cases} H_0 : \mu_{L.Amount.Yes} = \mu_{L.Amount.No} \\ H_1 : \mu_{L.Amount.Yes} > \mu_{L.Amount.No} \end{cases}$$

```
#Clasificamos la cantidad de los préstamos aceptados y los rechazados
loans.Loan_Status_Y.loanamount<- loans[loans$Loan_Status == "Y",]$LoanAmount
loans.Loan_Status_N.loanamount<- loans[loans$Loan_Status == "N",]$LoanAmount

#Realizamos el t.test
t.test(loans.Loan_Status_Y.loanamount, loans.Loan_Status_N.loanamount, alternative = "greater", conf.lev
```

```
##
##  Welch Two Sample t-test
##
## data:  loans.Loan_Status_Y.loanamount and loans.Loan_Status_N.loanamount
## t = 0.52426, df = 783.9, p-value = 0.3001
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -5.819122       Inf
## sample estimates:
## mean of x mean of y
##  144.0633  141.3455
```

Sobre esta variable tampoco tendremos evidencias suficientes que nos lleven a valorar la posibilidad de diferencias entre la cantidad media de los préstamos aceptados y rechazados.

Contraste entre el tiempo de devolución medio de los préstamos aceptados y los rechazados.

Sobre el último contraste veremos si el tiempo de devolución medio de los préstamos aceptados es significativamente superior al tiempo de devolución medio de los préstamos rechazados.

$$\begin{cases} H_0 : \mu_{L.AmountTerm.Yes} = \mu_{L.AmountTerm.No} \\ H_1 : \mu_{L.AmountTerm.Yes} < \mu_{L.AmountTerm.No} \end{cases}$$

```
#Clasificamos el tiempo de devolución de los préstamos aceptados y los rechazados
loans.Loan_Status_Y.loanamountterm<- loans[loans$Loan_Status == "Y",]$Loan_Amount_Term
loans.Loan_Status_N.loanamountterm<- loans[loans$Loan_Status == "N",]$Loan_Amount_Term

#Realizamos el t.test
t.test(loans.Loan_Status_Y.loanamountterm, loans.Loan_Status_N.loanamountterm, alternative = "less", co
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  loans.Loan_Status_Y.loanamountterm and loans.Loan_Status_N.loanamountterm
## t = -0.50426, df = 907.37, p-value = 0.3071
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 4.807713
## sample estimates:
## mean of x mean of y
##  340.9343  343.0567
```

Del mismo modo que en los constrastes anteriores, al obtener un p-valor superior al nivel de significancia (0.05), no rechazaremos la hipótesis nula, por lo que no podremos afirmar que el tiempo medio de devolución de los préstamos aceptados es superior al de los rechazados.

4.3.2. Variables Cualitativas

Tras analizar detenidamente las variables cuantitativas pasaremos a estudiar las relaciones entre las variables cualitativas y nuestra variable clasificadora "Loan_Status". El objetivo de este apartado es determinar la existencia de una relación significante entre una variable cualitativa y la variable clasificadora. Para ello, recurriremos a mostrar la distribución de la variable a estudiar mediante un diagrama de barras y calcularemos su relación mediante el test chi-cuadrado. Sobre cada uno de los apartados se tomarán las siguientes hipótesis.

$$\begin{cases} H_0 : No\ \ existe\ \ relación\ \ entre\ \ las\ \ variables \\ H_1 : Existe\ \ relación\ \ entre\ \ las\ \ variables \end{cases}$$

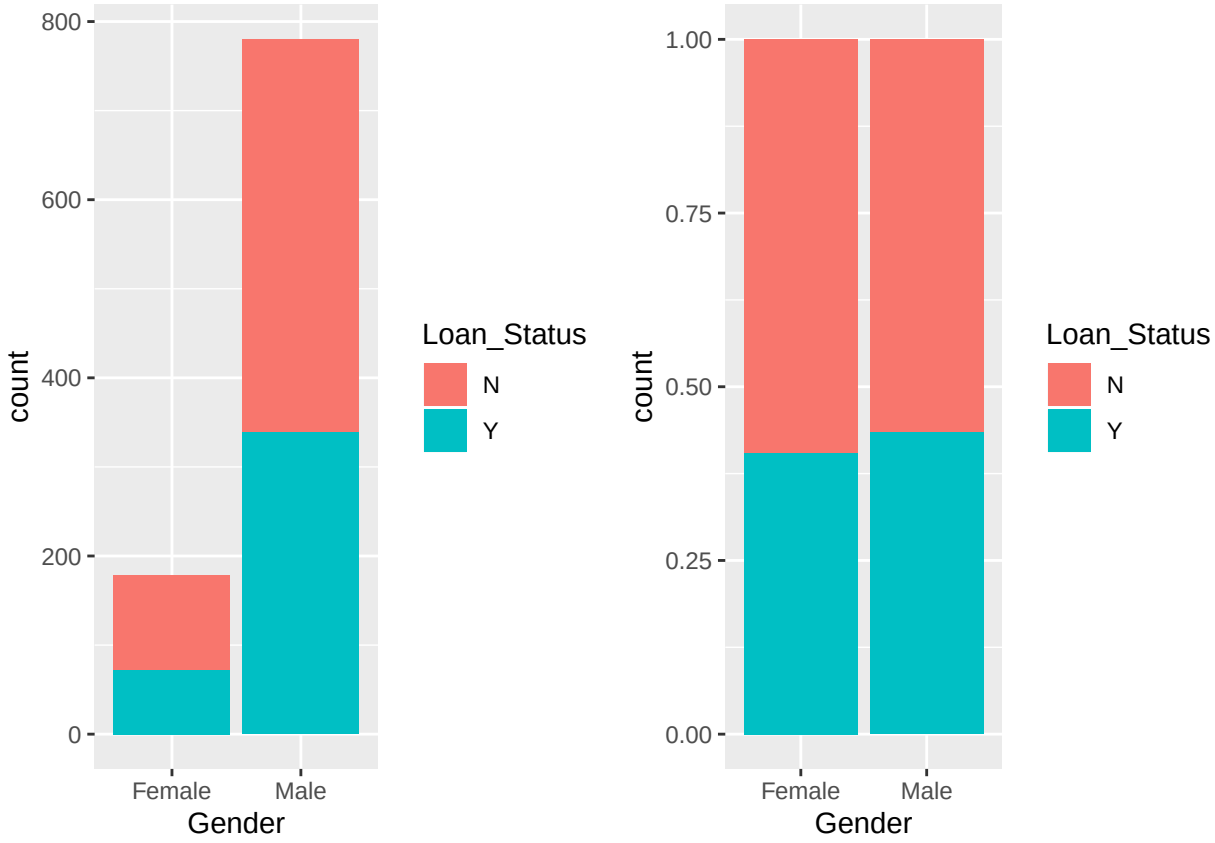
Una vez determinada la forma de actuación sobre este tipo de variables pasaremos a calcular la relación que obtenemos sobre cada variable cualitativa. De obtener relación entre las variables calcularemos el odds ratio con el fin de conocer la intensidad de la relación. El odds ratio es una medida de asociación que indica la fortaleza de relación entre dos variables. Los odds ratio oscilan entre 0 e infinito de la siguiente forma:

- Odd Ratio < 1. Tendremos una asociación negativa y nos encontraremos ante un factor de protección.
- Odd Ratio $= 1$. Indicará ausencia de asociación entre variables.
- Odd Ratio > 1. Conseguiremos una asociación positiva entre las variables, nos encontraremos ante un factor de riesgo.

¿Existe relación entre la concesión del préstamo y el género del solicitante?

Representaremos el diagrama de barras sin normalizar y normalizado. En color rojo se mostrarán los préstamos que han sido rechazados y en color azul, los aceptados.

```
#Mostramos la distrubución de la variable "Gender"
gender=ggplot(loans,aes(x=Gender,fill=Loan_Status))+geom_bar()
gendernorm=ggplot(loans,aes(x=Gender,fill=Loan_Status))+geom_bar(position="fill")
plot_grid(gender, gendernorm)
```



Sobre el diagrama de frecuencias podemos observar una gran diferencia entre hombres y mujeres, donde las mujeres representan aproximadamente 1/4 del total de los hombres. Por otra parte, si nos fijamos sobre el diagrama de barras normalizado, vemos como a simple vista apenas se observa diferencia entre la aceptación y el rechazo del préstamo entre hombres y mujeres.

Construimos una tabla de contigencia sobre la que realizar el test chi-cuadrado y lo aplicamos.

```
#Tabla de contingencia entre "Loan_Status" y "Gender"
Status_Gender<-table(loans$Loan_Status, loans$Gender)
Status_Gender
```

/	Female	Male
N	106	441
Y	72	339

```
#Realizamos el chisq.test
chisq.test(Status_Gender, 95)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  Status_Gender
## X-squared = 0.42085, df = 1, p-value = 0.5165
```

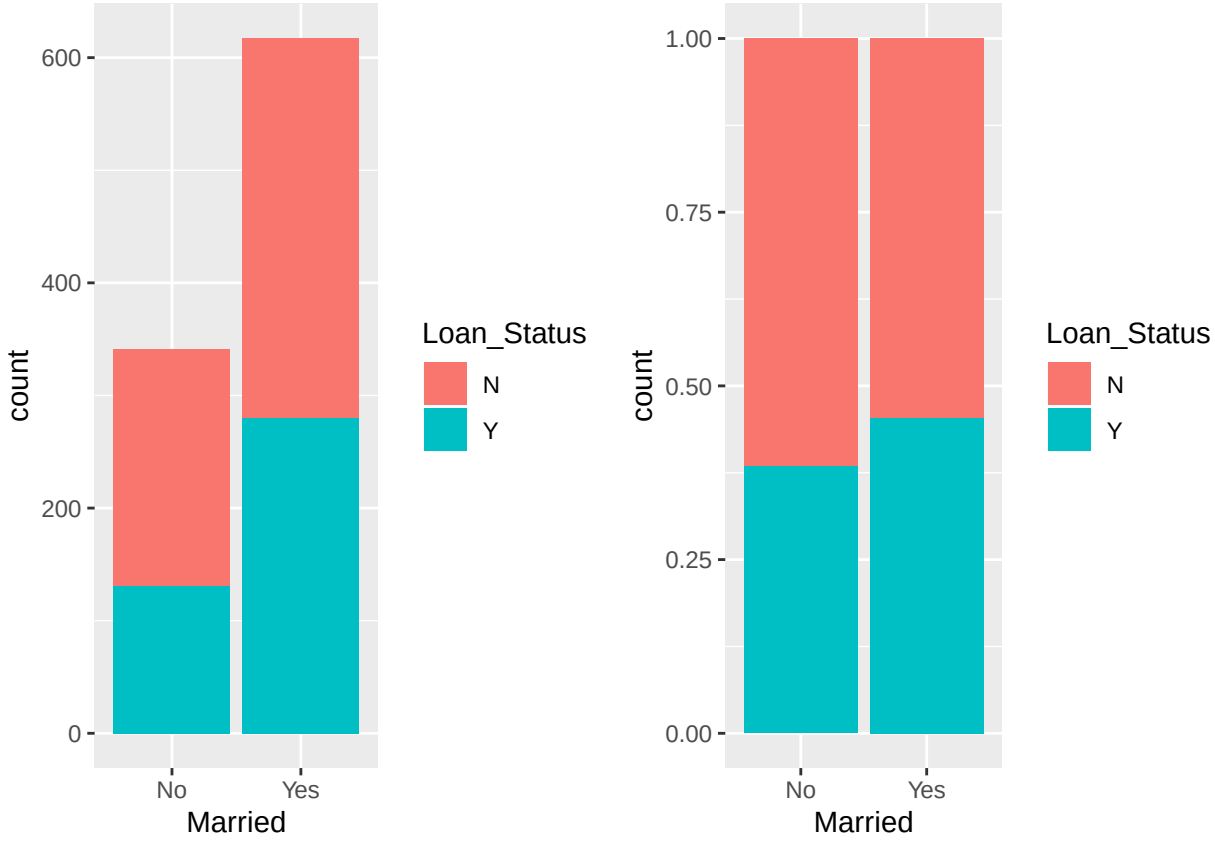
Obtenemos un p-valor superior al nivel de significancia (0.05) indicándonos que no podemos rechazar la

hipótesis nula y que por tanto no tendremos una relación apreciable entre la concesión del préstamo y el género del solicitante. Las conclusiones extraídas sobre el análisis del diagrama de barras normalizado coinciden con el test.

¿Existe relación entre la concesión del préstamo y el hecho de que el solicitante esté casado?

Representamos gráficamente los diagramas de barras para hacernos una idea sobre la distribución de la variable.

```
#Mostramos la distrubución de la variable "Married"
married=ggplot(loans,aes(x=Married,fill=Loan_Status))+geom_bar()
marriednorm=ggplot(loans,aes(x=Married,fill=Loan_Status))+geom_bar(position="fill")
plot_grid(married, marriednorm)
```



Los solicitantes casados predominan sobre los no casados como podemos ver sobre el diagrama de barras de frecuencias. Por otra parte, en el diagrama de barras normalizado observamos como los casados tienen un porcentaje ligeramente superior sobre la aceptación de préstamos que los no casados. Veremos si esta diferencia es significativa mediante el test chi-cuadrado.

Calculamos la tabla de contingencia entre "Loan_Status" y "Married" con la que realizaremos el test.

```
#Tabla de contingencia entre "Loan_Status" y "Married"
Status_Married<-table(loans$Loan_Status, loans$Married)
Status_Married
```

/	No	Yes
N	210	337
Y	131	280

```
#Realizamos el chisq.test
chisq.test(Status_Married, 95)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  Status_Married
## X-squared = 4.0689, df = 1, p-value = 0.04368
```

El p-valor es ligeramente inferior al nivel de significancia (0.05) por lo que rechazaremos la hipótesis nula y aceptaremos la hipótesis alternativa en la que se consideraba que las variables "Loan_Status" y "Married" se encuentran relacionadas. Con el objetivo de conocer cómo de fuerte es esta relación recurriremos al cálculo de la odds ratio.

```
#Calculamos la odd ratio
odds.ratio(Status_Married)
```

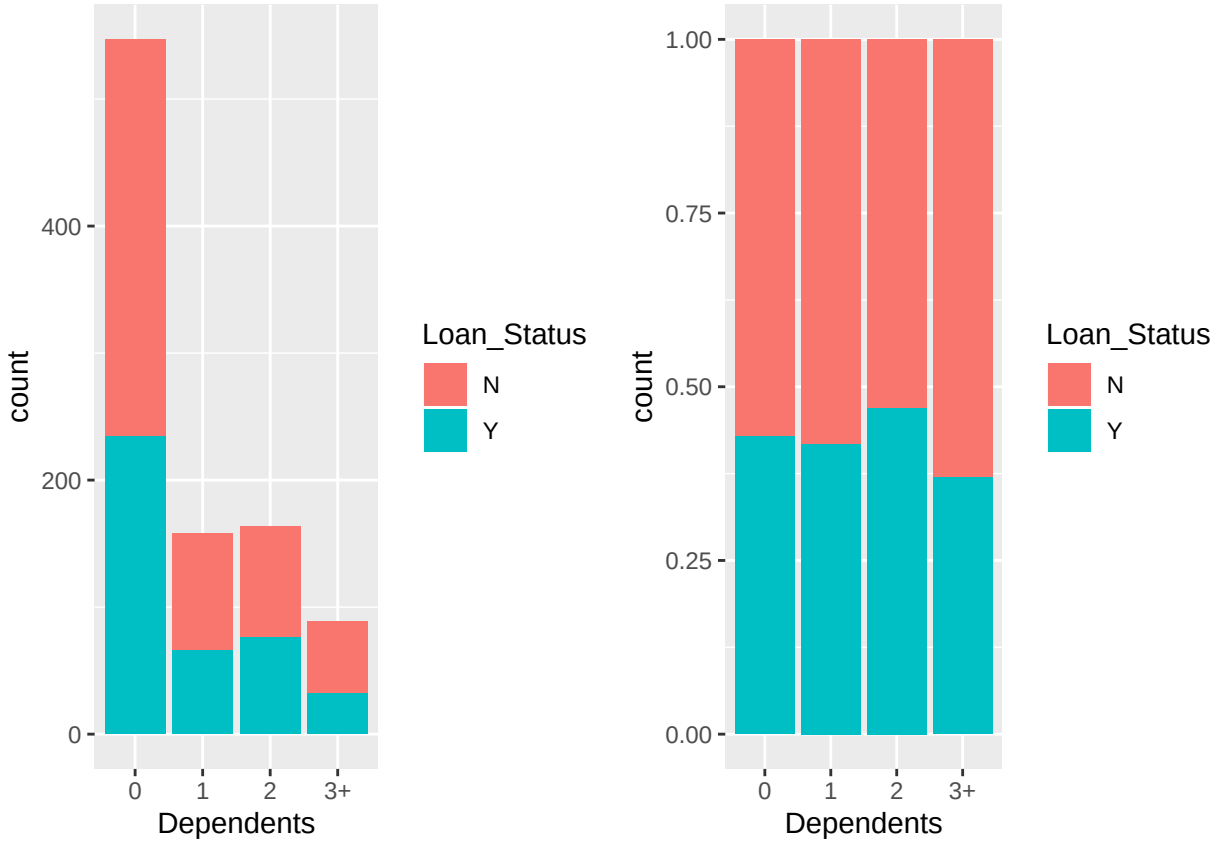
	OR	2.5 %	97.5 %	p
Fisher's test	1.331507	1.008341	1.761463	0.0408403

Al obtener un odds ratio superior a 1 vemos como el hecho de estar casado es un factor de riesgo sobre la concesión del préstamo. Más concretamente tendremos que la posibilidad de que te concedan el préstamo es 1.33 veces superior estando casado que no estándolo.

¿Existe relación entre la concesión del préstamo y el número de componentes de la unidad solicitante?

Representamos los diagramas de barras.

```
#Mostramos la distrubución de la variable "Dependents"
dependents=ggplot(loans,aes(x=Dependents,fill=Loan_Status))+geom_bar()
dependentsnorm=ggplot(loans,aes(x=Dependents,fill=Loan_Status))+geom_bar(position="fill")
plot_grid(dependents, dependentsnorm)
```



El número de solicitantes con cero componentes en la unidad familiar es sustancialmente superior al resto. Sin embargo, sobre la gráfica normalizada podemos apreciar como las diferencias entre las concesiones de los préstamos apenas sufren modificaciones. Calculamos la tabla de contingencia y aplicamos el test chi-cuadrado.

```
#Tabla de contingencia entre "Loan_Status" y "Dependents"
Status_Dependents<-table(loans$Loan_Status, loans$Dependents)
Status_Dependents
```

/	0	1	2	3+
N	312	92	87	56
Y	235	66	77	33

```
#Realizamos el chisq.test
chisq.test(Status_Dependents, 95)
```

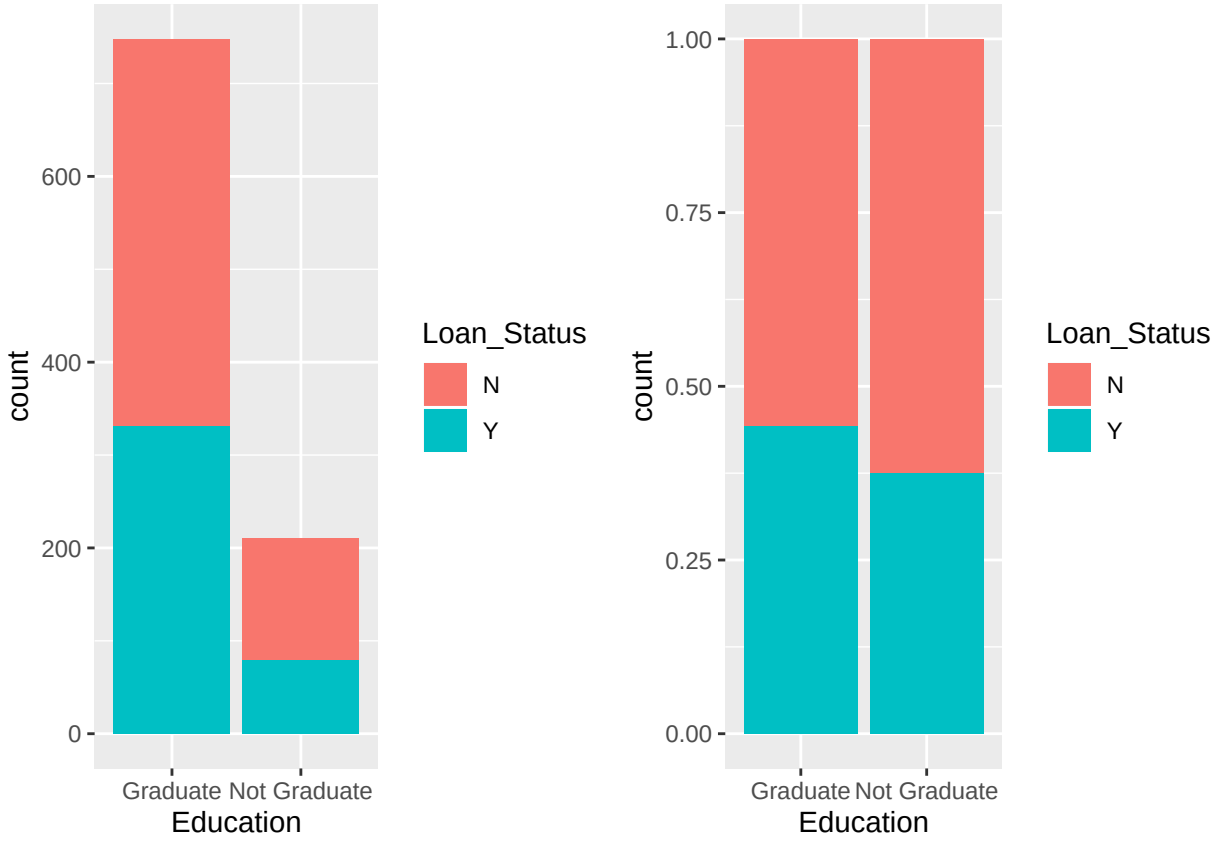
```
##
##  Pearson's Chi-squared test
##
## data:  Status_Dependents
## X-squared = 2.4129, df = 3, p-value = 0.4912
```

El p-valor es superior al nivel de significancia y no rechazaremos la hipótesis nula que implicaba la no relación entre las variables sometidas a estudio, en nuestro caso "Loan_Status" y "Dependents".

¿Existe relación entre la concesión del préstamo y los estudios del solicitante?

Representamos mediante diagramas de barras la distribución de los estudios de los solicitantes presentes en el fichero.

```
#Mostramos la distrubución de la variable "Education"
education=ggplot(loans,aes(x=Education,fill=Loan_Status))+geom_bar()
educationnorm=ggplot(loans,aes(x=Education,fill=Loan_Status))+geom_bar(position="fill")
plot_grid(education, educationnorm)
```



El número de graduados es muy superior al número de solicitantes no graduados. Además, los solicitantes graduados tienen un porcentaje ligeramente superior sobre el número de préstamos concedidos. Veremos si mediante el test chi-cuadrado la diferencia obtenida resulta significativa.

Calculamos la tabla de contingencia para aplicar el test.

```
#Tabla de contingencia entre "Loan_Status" y "Education"
loans$Education=factor(loans$Education, levels=c( "Not Graduate", "Graduate"),  labels=c("Not Graduate"
Status_Education<-table(loans$Loan_Status, loans$Education)
Status_Education
```

/	Not Graduate	Graduate
N	131	416
Y	79	332

```r
#Realizamos el chisq.test
chisq.test(Status_Education, 95)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  Status_Education
## X-squared = 2.7942, df = 1, p-value = 0.0946
```

Sobre este test, el p-valor es ligeramente superior al nivel de significancia, no rechazaremos la hipótesis nula que implicaba la no relación entre las variables "Loan_Status" y "Education". A pesar de esto, el p-valor es muy próximo a 0.05 y se encuentra cerca de ser aceptada la hipótesis alternativa por lo que calcularemos la odds ratio para ver el tipo de relación.

```r
#Calculamos la odd ratio
odds.ratio(Status_Education)
```

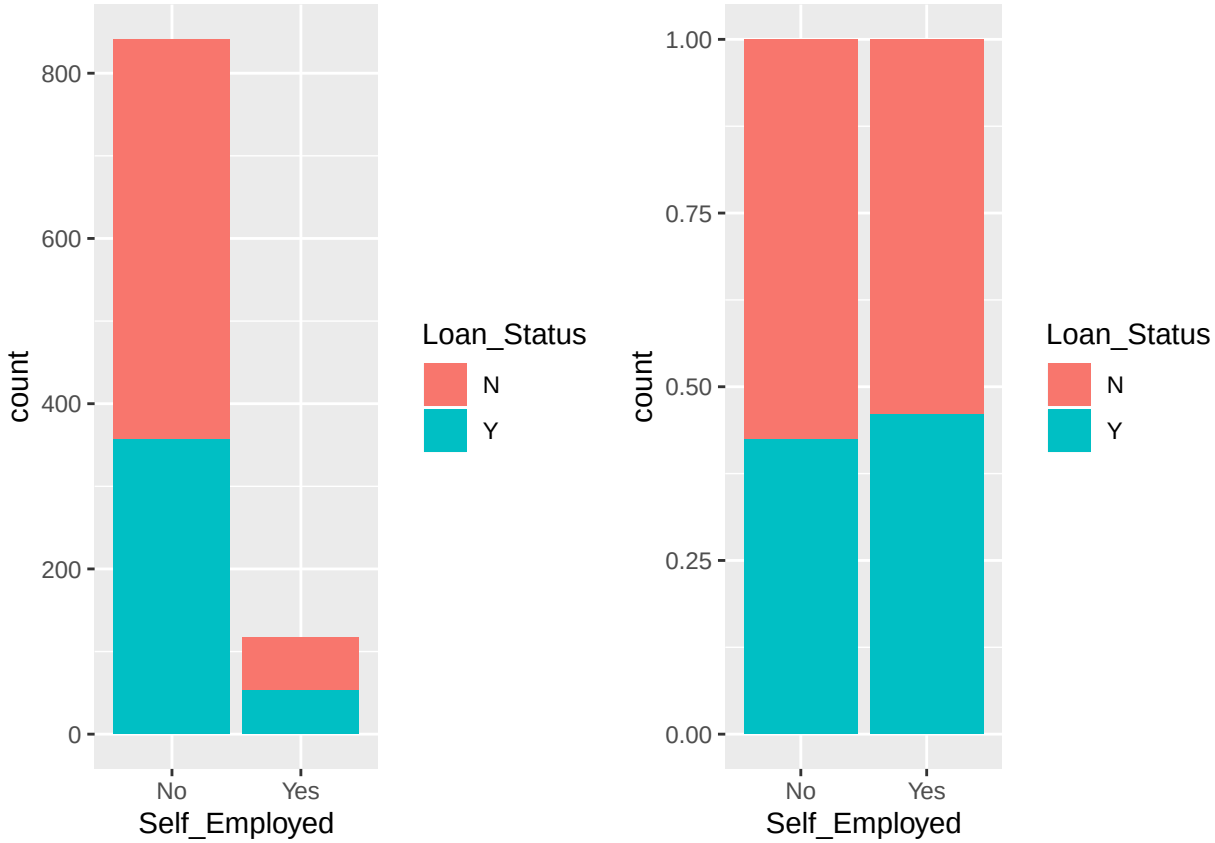
	OR	2.5 %	97.5 %	p
Fisher's test	1.323006	0.9561328	1.838419	0.083091

Al ser superior a 1 nos encontraremos ante un factor de riesgo. Este valor nos indica que las posibilidades de concesión del préstamo son 1.32 veces superiores estando graduado que no estándolo.

¿Existe relación entre la concesión del préstamo y el hecho de que el solicitante sea autónomo?

Representamos la distribución de la variable "Self_Employed".

```r
#Mostramos la distrubución de la variable "Self_Employed"
selfemployed=ggplot(loans,aes(x=Self_Employed,fill=Loan_Status))+geom_bar()
selfemployednorm=ggplot(loans,aes(x=Self_Employed,fill=Loan_Status))+geom_bar(position="fill")
plot_grid(selfemployed, selfemployednorm)
```



La gran mayoría de los solicitantes no son autónomos y, a priori, la diferencia entre los autónomos y los no autónomos a los que se les conceden el préstamo es mínima. Veremos si el test chi-cuadrado nos indica lo mismo.

Calculamos la tabla de contingencia y aplicamos el test chi-cuadrado.

```
#Tabla de contingencia entre "Loan_Status" y "Self_Employed"
Status_SelfEmployed<-table(loans$Loan_Status, loans$Self_Employed)
Status_SelfEmployed
```

/	No	Yes
N	484	63
Y	357	54

```
#Realizamos el chisq.test
chisq.test(Status_SelfEmployed, 95)
```

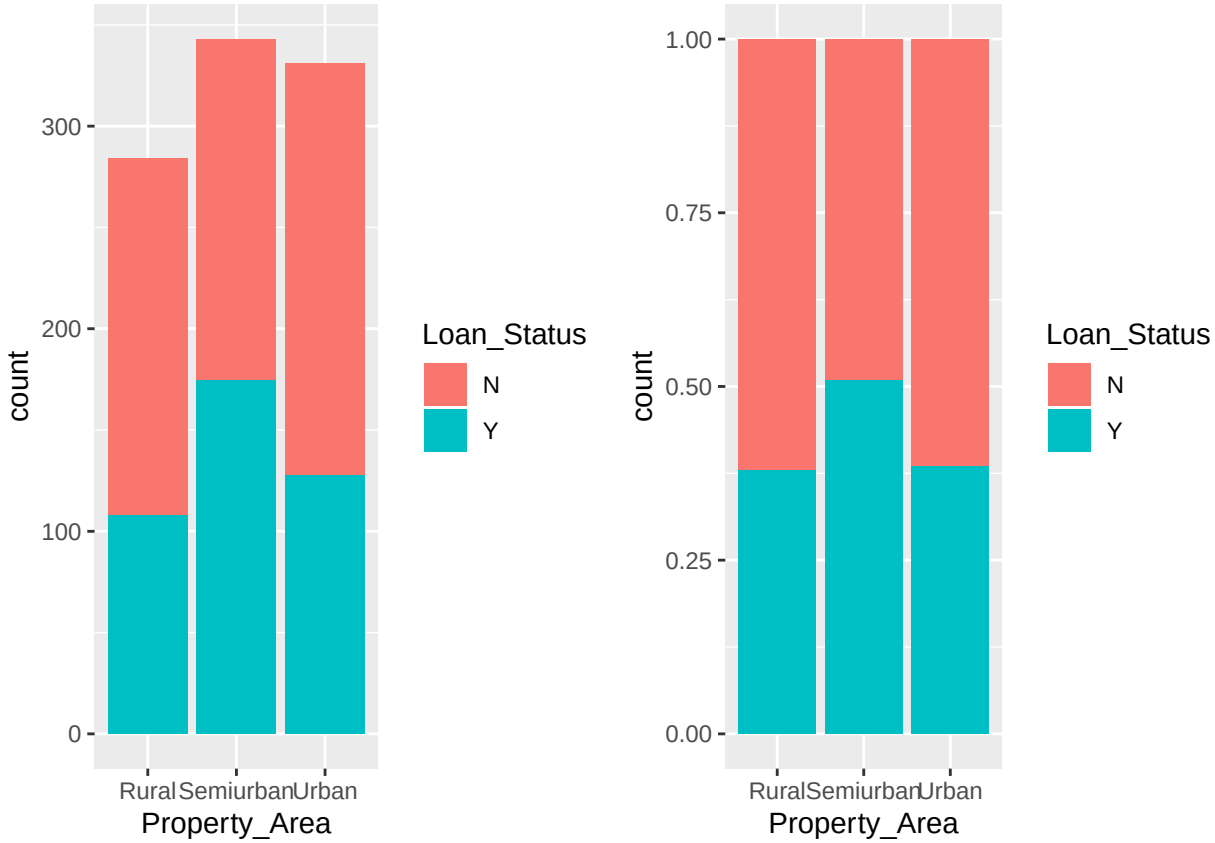
```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  Status_SelfEmployed
## X-squared = 0.43409, df = 1, p-value = 0.51
```

La hipótesis nula no será rechazada al obtener un p-valor superior al nivel de significancia, es decir, no podremos afirmar relación entre las variables "Loan_Status" y "Self_Employed".

¿Existe relación entre la concesión del préstamos y el tipo de propiedad?

Mediante un diagrama de barras mostramos la distribución de la variable "Property_Area" sobre el fichero de datos.

```
#Mostramos la distrubución de la variable "Property_Area"
propertyarea=ggplot(loans,aes(x=Property_Area,fill=Loan_Status))+geom_bar()
propertyarea_norm=ggplot(loans,aes(x=Property_Area,fill=Loan_Status))+geom_bar(position="fill")
plot_grid(propertyarea, propertyarea_norm)
```



Las áreas semiurbanas son las que más solicitudes de préstamos reciben, seguidas muy de cerca por las áreas urbanas y tras ellas, las rurales. Sobre el diagrama de barras normalizado vemos como la concesión de préstamos sobre áreas semiurbanas es significativamente superior a las áreas rurales o urbanas que mantienen un porcentaje muy similar.

Calculamos la tabla de contingencia para posteriormente aplicar el test chi-cuadrado.

```
#Tabla de contingencia entre "Loan_Status" y "Property_Area"
Status_PropertyArea<-table(loans$Loan_Status, loans$Property_Area)
Status_PropertyArea
```

/	Rural	Semiurban	Urban
N	176	168	203
Y	108	175	128

```r
#Realizamos el chisq.test
chisq.test(Status_PropertyArea, 95)
```

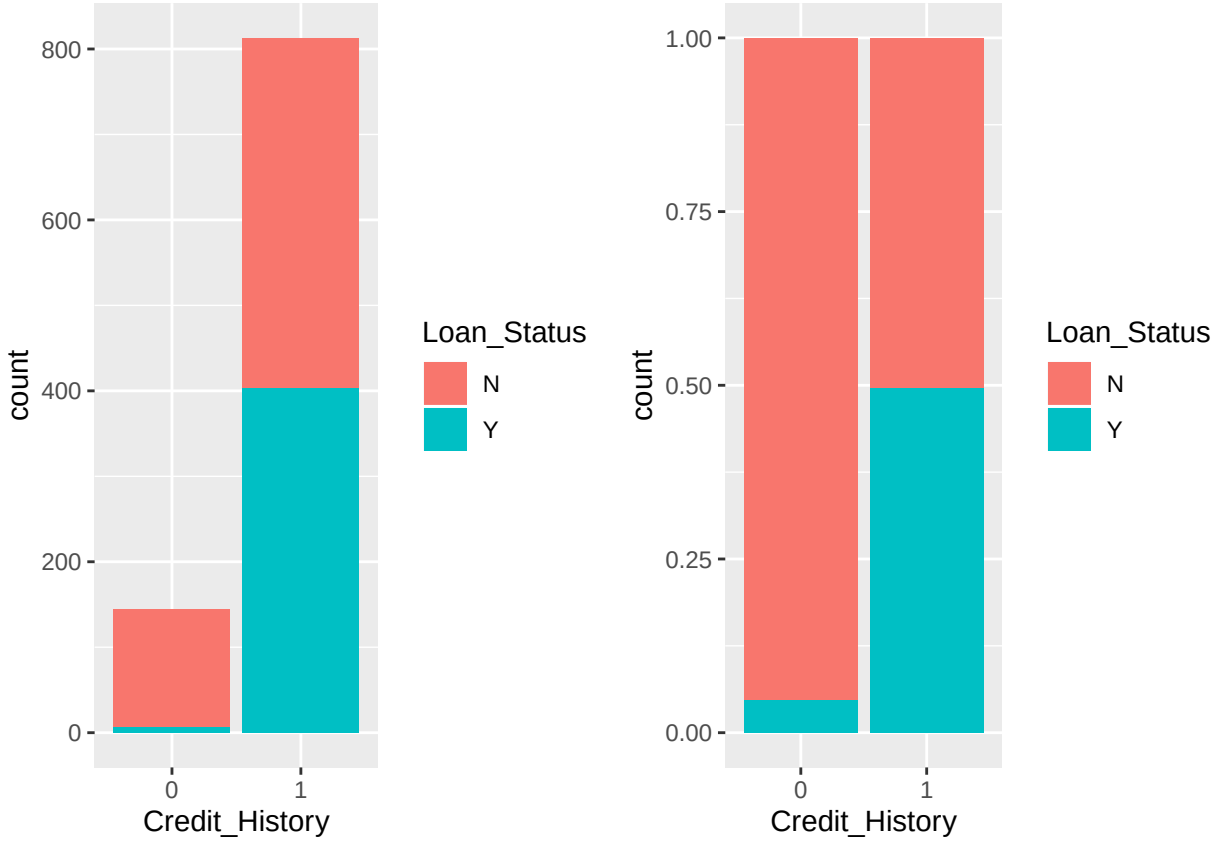
```
##
##  Pearson's Chi-squared test
##
## data:  Status_PropertyArea
## X-squared = 14.402, df = 2, p-value = 0.0007459
```

El p-valor es inferior al nivel de significancia por tanto, rechazaremos la hipótesis nula y aceptaremos la alternativa, donde afirmábamos la relación entre las variables "Loan_Status" y "Property_Area". En este caso, no podremos calcular la odds ratio ya que esta medida de asociación trabaja sobre dos variables dicotómicas divididas, que permiten crear una tabla de contingencia 2x2 a partir de la cual obtener el odds ratio.

¿Existe relación entre la concesión del préstamo y el historial crediticio?

Mostramos la distribución del historial crediticio mediante un diagrama de barras.

```r
#Mostramos la distrubución de la variable "Credit_History"
credit=ggplot(loans,aes(x=Credit_History,fill=Loan_Status))+geom_bar()
credit_norm=ggplot(loans,aes(x=Credit_History,fill=Loan_Status))+geom_bar(position="fill")
plot_grid(credit, credit_norm)
```



La mayoría de los solicitantes tienen un historial crediticio positivo. Además, sobre el diagrama de barras normalizado observamos que tener un historial crediticio positivo aumenta drásticamente tus posibilidades

de que te concedan el préstamo, más concretamente, tienes un 50% de posibilidades de que te concedan el préstamo. Sin embargo, un historial crediticio negativo implica menos de un 10% de posibilidades de que te concedan el préstamo.

Creamos la tabla de contingencia para aplicar el test chi-cuadrado.

```
#Tabla de contingencia entre "Loan_Status" y "Credit_History"
Status_CreditHistory<-table(loans$Loan_Status, loans$Credit_History)
Status_CreditHistory
```

/	0	1
N	138	409
Y	7	404

```
#Realizamos el chisq.test
chisq.test(Status_CreditHistory, 95)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  Status_CreditHistory
## X-squared = 99.29, df = 1, p-value < 2.2e-16
```

Como era de esperar, la hipótesis alternativa será aceptada al tener un p-valor inferior al nivel de significancia. De esta forma, afirmamos la relación entre las variables "Loan_Status" y "Credit_History". Según el diagrama de barras hemos visto una gran relación entre ellas, calcularemos la odd ratio para conocer el grado de esta relación.

```
#Calculamos la odd ratio
odds.ratio(Status_CreditHistory)
```

	OR	2.5 %	97.5 %	p
Fisher's test	19.43179	9.026455	49.8997	0

El valor del odds ratio es superior a 1, encontrándonos ante un valor de riesgo. Tendremos que la posibilidad de que te concedan el préstamo es 19.43 veces superior si posees un historial crediticio positivo que negativo.

4.4. Modelo logístico

4.4.1. Creación del modelo.

El primer paso en la creación del modelo es la separación de los datos en un conjunto de entrenamiento (train) y un conjunto de prueba (test). El primer conjunto nos permitirá, como su nombre indica, entrenar nuestro modelo y ajustar los parámetros definidos por las variables independientes con el objetivo de clasificar correctamente la variable dependiente Loan_Status. Con el segundo conjunto podremos comprobar la precisión del modelo generado, prediciendo la concesión de préstamos según los valores de la variables aleatorias. Este segundo conjunto no dispondrá de la columna Loan_Status, la cual será utilizada posteriormente para extraer la precisión del modelo.

Comenzamos dividiendo nuestro dataset en los dos conjuntos indicados. Seleccionamos una semilla ("seed") específica que nos permita controlar la aleatoriedad del sample y así poder reproducir el mismo modelo. En el conjunto de entrenamiento incluiremos el 75% de nuestros registros, constituyendo el 25% restante el conjunto de test.

```
#Definimos la semilla
set.seed(101)
#Nos deshacemos de la última columna, en la que aparecía Loan_Status representado numéricamente
loans=loans[,-13]
#Dividimos aleatoriamente el dataset en un grupo con el 75% y otro con el 25% restante.
sample = sample.split(loans, SplitRatio = .75)
#Asignamos el conjunto de entrenamiento al 75%
train = subset(loans, sample == TRUE)
#Asignamos el conjunto de prueba al 25% restante
test  = subset(loans, sample == FALSE)
```

Tal y como se planteó en los objetivos de la práctica, resultará de gran interés poder predecir los resultados sobre la concesión del préstamo a partir de las variables más influyentes. Este proceso de predicción podría ayudar a la empresa a la hora de clasificar aquellos clientes con más posibilidades de obtener un préstamo, con el fin de agilizar dichas concesiones y que el tiempo de espera desde la solicitud al resultado final se minimice lo máximo posible.

Nuestro objetivo fundamental del modelo es conseguir una gran capacidad predictora sobre la variable "Loan_Status", para satisfacer estas condiciones recurriremos a modelos de regresión logística. Crearemos varios modelos donde la variable dependiente ("Loan_Status") será siempre la misma y realizaremos modificaciones sobre las variables independientes. La elección de unas variables independientes u otras se basa principalmente en la relación obtenida sobre la variable "Loan_Status", calculada en los apartados anteriores, es decir, tomaremos las variables más influyentes sobre "Loan_Status". Los modelos creados serán los siguientes.

- *Modelo 1* : Consideraremos únicamente una variable predictora o independiente "Credit History". Esta variable es la que presenta la relación más fuerte con "Loan Status"
- *Modelo 2* : Además de "Credit_History" añadiríamos otra variable relacionada con "Loan_Status", "Married".
- *Modelo 3* : Tendremos como variables a "Credit_History", "Married" y "Education".
- *Modelo 4* : Tendremos como variables a "Credit_History", "Married", "Education" y "Property_Area".
- *Modelo 5* : Tendremos como variables a "Credit_History", "Married", "Education", "Property_Area" y "ApplicantIncome".

```
#Modelos de regresión logística
mlog1.glm <- glm(Loan_Status ~ Credit_History, train, family = "binomial")
mlog2.glm <- glm(Loan_Status ~ Credit_History + Married, train, family = "binomial")
mlog3.glm <- glm(Loan_Status ~ Credit_History + Married
                 + Education, train, family = "binomial")
mlog4.glm <- glm(Loan_Status ~ Credit_History + Married + Education
                 + Property_Area, train, family = "binomial")
mlog5.glm <- glm(Loan_Status ~ Credit_History + Married + Education
                 + Property_Area + ApplicantIncome, train, family = "binomial")


#rsq(mlog6.glm)
```

```
tabla.coeficientes <- matrix(c( 1, extractAIC(mlog1.glm )[2],
                                2, extractAIC(mlog2.glm )[2],
                                3, extractAIC(mlog3.glm )[2],
                                4, extractAIC(mlog4.glm )[2],
                                5, extractAIC(mlog5.glm )[2]),
                                ncol = 2, byrow = TRUE)


colnames(tabla.coeficientes) <- c("Modelo", "AIC")
tabla.coeficientes
```

Modelo	AIC
1	905.0882
2	906.0910
3	904.7792
4	895.0202
5	896.2122

Para comparar cada uno de los modelos lo mejor es recurrir al denominado "Criterio de Información de Akaike", cuya mínimización nos indicará un mejor ajuste del modelo. Cabe destacar que el AIC, nos permite comparar modelos pero no nos aporta información alguna al tratarlos por separado. Podemos apreciar como el mejor modelo es el cuarto, aunque seguido muy de cerca por el quinto, por lo que podríamos optar por cualquiera de los dos. En nuestro caso, optaremos por conservar la variable que nos indica las ganancias del solicitante con el fin de tener una variable numérica que sirva de soporte sobre nuestro modelo.

4.4.2. Precisión del modelo

A continuación, calcularemos la precisión del modelo elegido, mlog5.glm. Para ello, extraeremos del conjunto de prueba las columnas que se utilizan como variables en el modelo y se predecirá el estado de concesión que el modelo considera para cada registro. Finalmente, calcularemos la precisión del modelo a partir de los datos observados y esperados.

```
#Seleccionamos las variables que utilizamos en nuestro modelo sobre el subconjunto test
test_model= test[, c(2,4,6,10,11)]
#Guardamos el resultado de la adjudicación del préstamo del subconjunto test
test_class= test[,12]
#Predecimos los resultados de la adjudicación a partir del test_model
pred_prob=predict(mlog5.glm, test_model, type="response",se.fit = FALSE)
pred=pred_prob

#Clasificamos como "Y" las variables con una probabilidad superior al 0.5
pred[pred > 0.5]="Y"
#Clasificamos como "N" las variables con una probabilidad inferior al 0.5
pred[pred <= 0.5]="N"

#Convertimos a factor nuestras predicciones
pred = as.factor(pred)

#Calculamos la tabla de contingencia para las predicciones
prediction_table = table(pred, test_class)
prediction_table
```

pred/test_class	N	Y
N	109	53
Y	34	42

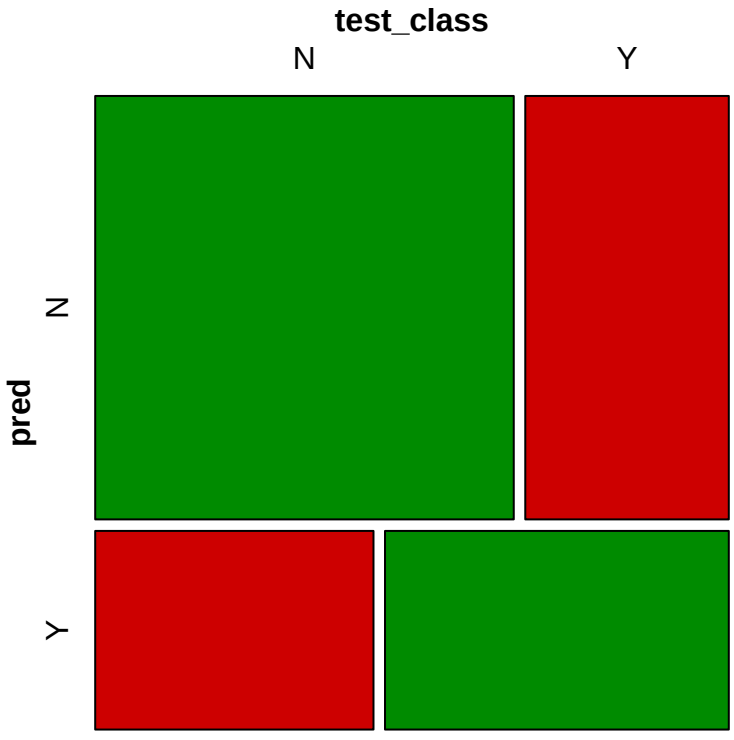
La matriz de confusión nos muestra que el modelo clasifica bien 109 solicitudes rechazadas y 42 solicitudes aceptadas, pero no lleva a cabo una clasificación correcta de 34 casos rechazados y 53 casos aceptados. Veamos cuál es la precisión del modelo a partir de la matriz.

```
#Mostramos la precisón
confusionMatrix(prediction_table)$overall[1]
```

```
##  Accuracy
## 0.6344538
```

Obtenemos una precisión del 63.5% para nuestro modelo logístico, es decir, el modelo conseguirá predecir correctamente 63 solicitudes de cada 100 que lleguen a las oficinas de la organización. Veamos de manera más gráfica este resultado.

```
#Representamos la tabla de contingencia para las predicciones
mosaic(prediction_table , shade = T, colorize = T,
       gp = gpar(fill = matrix(c("green4", "red3", "red3", "green4"), 2, 2)))
```



Como vemos, pese a que un porcentaje considerable de las solicitudes serán clasificadas correctamente por el método, especialmente para aquellas solicitudes con tendencia a ser rechazadas, el modelo sigue sin clasificar

correctamente un gran número de solicitudes. Además, es en las solicitudes aceptadas en las que el modelo tiende a equivocarse, siendo estas las que finalmente generan ingresos en la organización (la devolución de los intereses préstamo). Por tanto, concluimos que el modelo no será capaz realizar las predicciones por si mismo de manera independiente, y será necesaria la presencia de un empleado a la hora de tomar la decisión definitiva.

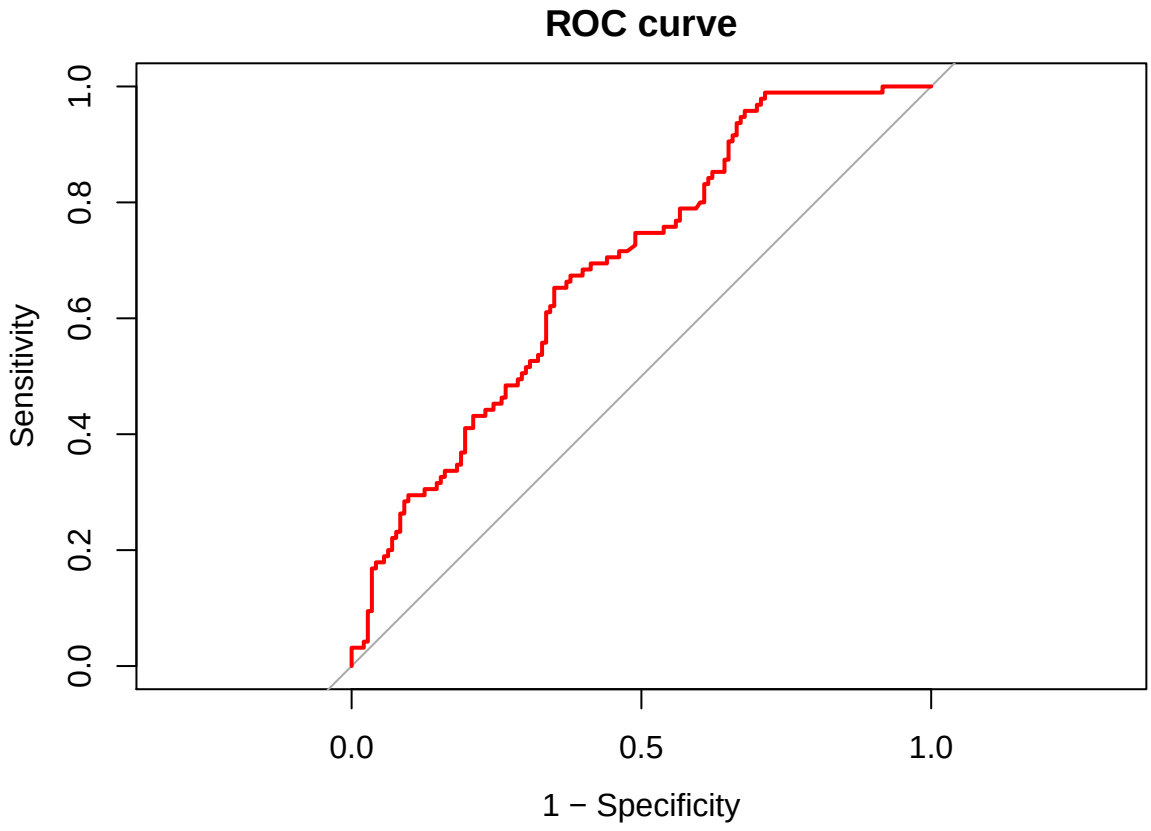
4.4.3. Bondad de ajuste

Con el objetivo de evaluar la bondad de ajuste en la regresión logística, podremos recurrir a la curva ROC, un gráfico que nos muestra la relación entre la sensibilidad y 1 menos la especificidad. Cada punto sobre la curva corresponderá a un nivel de umbral de discriminación en la matriz de confusión, es decir, se construirán todas las matrices cambiando dicho umbral desde el 1% hasta el 99%, calculándose la sensibilidad y 1 menos la especificidad, que serán representadas en la gráfica.

```
#Calculamos la curva Roc
resRoc <- roc(test_class ~ pred_prob)
resRoc$auc
```

```
## Area under the curve: 0.6885
```

```
#Representación de la curva Roc
plot(resRoc, legacy.axes = TRUE, col='red', main= 'ROC curve')
```



El mejor modelo será aquel que presente una curva lo más cercana posible a la esquina superior izquierda, por el contrario, un modelo no discriminante tendrá una curva ROC plana muy cercana a la diagonal. La

regla objetiva para comparar las curvas ROC es el área encontrada bajo la curva o también llamada AUROC. En nuestro caso vemos como el modelo se aproxima levemente a la esquina superior izquierda, alejándose de la diagonal y presenta el siguiente valor de AUROC:

AUROC = 0.6885

En nuestro caso, no contaremos con un ajuste logarítmico de gran calidad, pero estos son los resultados que nos evidencia el modelo. Aunque no se obtiene un modelo muy preciso debido a las pocas relaciones encontradas entre las variables, puede servir de ayuda a los prestamistas para agilizar los procesos de selección sobre personas candidatas a préstamos.

4.4.4. Predicciones del modelo

Realizaremos una serie de predicciones para ver el comportamiento final de nuestro modelo

- Predicción positiva (Loan_Status = Y)

Se seleccionarán resultados a priori favorables para la concesión del préstamo y prediciremos la probabilidad de dicha concesión.

```
newdata = data.frame(Credit_History="1", Married="Yes", Education="Graduate", Property_Area="Semiurban"
predict(mlog5.glm, newdata, type="response")
```

```
##         1
## 0.7525506
```

Con un historial crediticio positivo, estando casado, graduado, con unas ganancias de 50000€ y una propiedad semiurbana, la probabilidad de que te concedan el préstamo es de alrededor del 73%.

- Predicción negativa (Loan_Status = N)

Se seleccionarán resultados a priori desfavorables para la concesión del préstamo y prediciremos la probabilidad de dicha concesión.

```
newdata = data.frame(Credit_History="0", Married="No", Education="Not Graduate", Property_Area="Rural",
predict(mlog5.glm, newdata, type="response")
```

```
##          1
## 0.03608757
```

Con un historial crediticio negativo, no estando casado, sin graduar, con unas ganancias de 20000€ y una propiedad rural, la probabilidad de que te concedan el préstamo es de alrededordel 3%.

4.5. Random Forest

Estudiaremos el comportamiento de los datos sobre otro modelo con el fin de contrastar si los datos son lo suficientemente explicativos. Optaremos por el método de Random Forest, un método que combina una gran cantidad de árboles de decisión independientes probados sobre conjuntos aleatorios con igual distribución. A partir de este comando se crearán versiones diferentes del conjunto de entrenamiento usando muestro con reemplazo, tal y como define el método bagging.

```r
#Seleccionamos el test de nuestro modelo sin incluir la variable clasificadora
test_model_RF=test[,-12]

#Ajustamos el modelo con nuestro conjunto de entrenamiento
RF_model <- randomForest(Loan_Status~., data=train)

#Calculamos las predicciones del modelo
RF_pred = predict(RF_model, test_model_RF)

#Calculamos la tabla de contingencia para las predicciones
prediction_table_RF = table(RF_pred, test_class)
prediction_table_RF
```

RF_pred/test_class	N	Y
N	100	36
Y	43	59

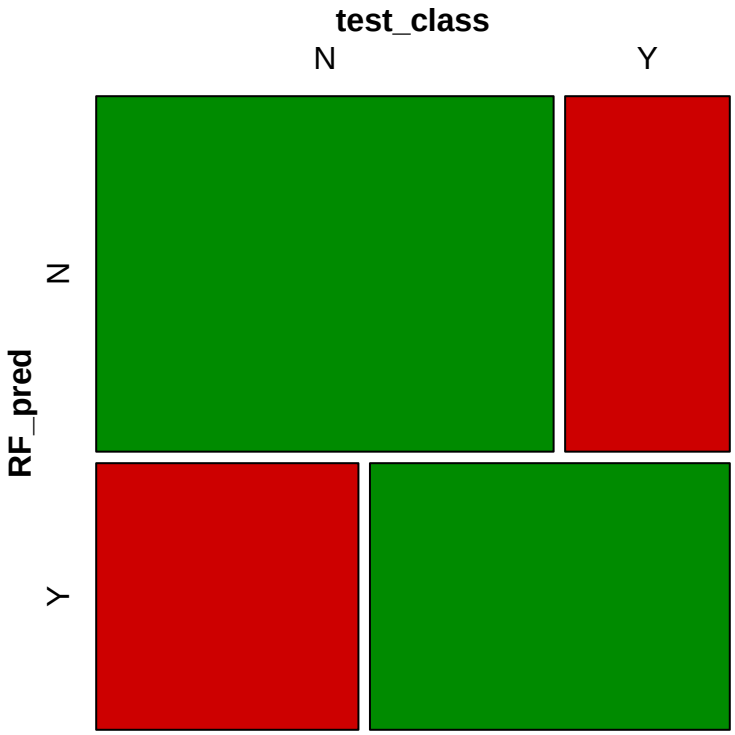
La matriz de confusión nos muestra que el modelo clasifica bien 103 solicitudes rechazadas y 56 solicitudes aceptadas, pero no lleva a cabo una clasificación correcta de 40 casos rechazados y 39 casos aceptados. Veamos cuál es la precisión del modelo a partir de la matriz.

```r
#Mostramos la precisión
confusionMatrix(prediction_table_RF)$overall[1]
```

```
##  Accuracy
## 0.6680672
```

Obtenemos una precisión del 66.8% para nuestro modelo logístico, es decir, el modelo conseguirá predecir correctamente 67 solicitudes de cada 100 que lleguen a las oficinas de la organización. Veamos de manera más gráfica este resultado.

```r
#Representamos la tabla de contingencia para las predicciones
mosaic(prediction_table_RF , shade = T, colorize = T,
       gp = gpar(fill = matrix(c("green4", "red3", "red3", "green4"), 2, 2)))
```



Como vemos, obtenemos una distribución muy parecida a la del modelo logístico. Pese a que un porcentaje considerable de las solicitudes serán clasificadas correctamente por el método, especialmente para aquellas solicitudes con tendencia a ser rechazadas, el modelo sigue sin clasificar correctamente un gran número de solicitudes.

5. Resolución del problema. Conclusiones finales.

Tras un extenso análisis sobre las variables cualitativas y cuantitativas presentes en el dataset, obtenemos algunos resultados un tanto sorprendentes. Las variables cuantitativas, aquellas que a priori podríamos pensar como importantes sobre la construcción del modelo, apenas han mostrado relación sobre la concesión del préstamo. Por ejemplo, las ganancias del solicitante no muestran una relación apreciable, algo bastante curioso. Esta independencia entre las variables cuantitativas y la concesión del acuerdo puede deberse en parte a los diversos trámites burocráticos a los que está asociado un préstamo que hacen que no dependa exclusivamente de las ganancias que posee un solicitante, sino también de otros parámetros externos que puedan asegurar la devolución del mismo ante situaciones extremas. De esta forma, encontramos unas variables cuantitativas tratadas en función de diversos parámetros externos que nos impiden establecer relaciones con la concesión final del préstamo.

Por otro lado, las variables cualitativas son las que presentarán una mayor influencia sobre la concesión del préstamo. Encontramos como "Married" y "Education" presentarán una relación similar con un odds ratio que ronda el 1.3 en ambos casos. Además, "Property Area" también presentará una buena relación debido al aumento de concesiones que se produce sobre las áreas semiurbanas. La variable más destacada es sin lugar a dudas el "Historial Crediticio", con un odds ratio de 19.43. Podemos decir que es la piedra angular de nuestro sistema.

Una vez obtenidas las variables más influyentes, se configuran varios modelos de regresión logística en función de nuestra variable categórica "Loan Status". El modelo elegido es capaz de predecir la probabilidad de concesión del préstamo en función del historial crediticio, el hecho de estar casado, la educación, la propiedad y las ganancias del solicitante. A pesar de que las ganancias del solicitante tenían una relación baja sobre la concesión del préstamo, serán consideradas sobre el modelo final con el fin de tener una variable numérica que acote nuestro modelo.

Los resultados arrojados sobre la precisión a la que sometimos a nuestro modelo logístico, alrededor de un 63.5%, nos hicieron plantearnos la creación de un nuevo modelo de clasificación más sofisticado. En nuestro caso escogimos un modelo RandomForest para analizar si la inexactitud se debe al modelo utilizado, o si se debe más bien a unos datos que no son suficientemente explicativos. El modelo generó una clasificación con precisión del 68%, algo mejor que el modelo logísitico, pero no lo suficiente como para ser considerado óptimo para nuestras necesidades. Por tanto, concluimos que son los datos los que carecen de tendencias y distribuciones relevantes a la hora de automatizar el proceso de concesión de préstamos.

Pese a todo, logramos satisfacer el objetivo fundamental propuesto al comienzo de la práctica. El modelo creado es capaz de actuar como un "primer filtro" que sirve de ayuda a los prestamistas para descartar las solicitudes menos aptas y dar prioridad a las que posean una mejor predisposición. Además, el modelo no solo clasificará las solicitudes, sino que también proporcionará valoración preliminar de la probabilidad de que dicho préstamo sea concedido.

6. Bibliografía

- Subirats, L., Pérez, D.O. & Calvo, M. (2019, septiembre). *Introducción a la limpieza y análisis de los datos*. Barcelona: FUOC. Material proporcionado por la UOC.
- https://www.kaggle.com/madhansing/bank-loan2 consultado por última vez el 06/06/2020.

7. Contribuciones

Contribuciones	Firma
Investigación previa	GYC, MMG
Redacción de las respuestas	GYC, MMG
Desarrollo de código	GYC, MMG