

# Tipología y ciclo de vida de los datos

## Práctica 2

11 de junio de 2019

Violeta Nashielli Andrade Méndez

### 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.

### Competencias

En esta práctica se desarrollan las siguientes competencias del Master 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:

- 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.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.

### Índice de resolución

1. Descripción del dataset.
2. Integración y selección de los datos de interés a analizar.
3. Limpieza de los datos.
4. Análisis de los datos.
  - Selección de los grupos de datos que se quieren analizar/comparar.
  - Comprobación de la normalidad y homogeneidad de la varianza.
  - 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.
5. Representación de los resultados a partir de tablas y gráficas.
6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?
7. Código: Hay que adjuntar el código.

## 1. Descripción del Dataset

Para esta práctica se buscó una base de datos que fuera simple de interpretar, como actualmente trabajo en una empresa financiera, se buscaron bases de este tipo.

Buscando en las bases financieras de la página de "Kaggle", se encontró la base de "Lending Club Loan Data" <https://www.kaggle.com/wendykan/lending-club-loan-data>.

Esta base contiene información acerca de préstamos emitidos desde el 2007 hasta el 2015, la base contiene muchas variables que suenan interesantes y que podrían ayudarnos a responder muchas preguntas. Por ejemplo ¿Qué tipo de cliente tiene un estatus actual específico?

Las variables que podrían darnos mucha información acerca de los clientes son por ejemplo:

1. "loan amount". El monto del préstamo
2. "term" El número de periodos del préstamo, pueden ser 36 o 60 meses
3. "int\_rate" La tasa de interés asignada a ese préstamo
4. "emp\_title" El título de la persona que solicitó el préstamo
5. "home\_ownership" El estatus de la persona que solicitó el préstamo, con respecto a su vivienda (si es rentada, propia, o tiene una hipoteca)

Con estas variables podríamos realizar grupos de tipos de personas que piden un préstamo, o podríamos identificar para cada tipo de "home\_ownership" el monto de préstamo que solicitan. Las posibilidades de análisis con las variables que parece tener la base son muchas.

## 2. Integración y selección de los datos a utilizar

La base original (loan.csv) tiene 2,260,668 registros y 145 variables, por lo tanto se realizó una muestra aleatoria de la base y se escogieron 10 mil registros aleatorios. Por otro lado se realizó un análisis de las columnas de la base y se decidió trabajar con las más significativas y las que tuvieran más sentido para el análisis.

La descripción de cada columna se incluye en el archivo .xlsx anexo (LCDDataDictionary(Final)), en la hoja Esquema Final.

Las columnas en amarillo fueron eliminadas de la base, bien porque estaban vacías en su totalidad como id, member\_id y url; o bien porque no contenían descripción de la columna y no sabríamos interpretar los resultados.

Las columnas en verde, fueron las columnas elegidas para este análisis. También le cambié el nombre a las variables para que durante el análisis sea más fácil saber a que se refiere cada una de ellas. A continuación se da un resumen de las variables a utilizar:

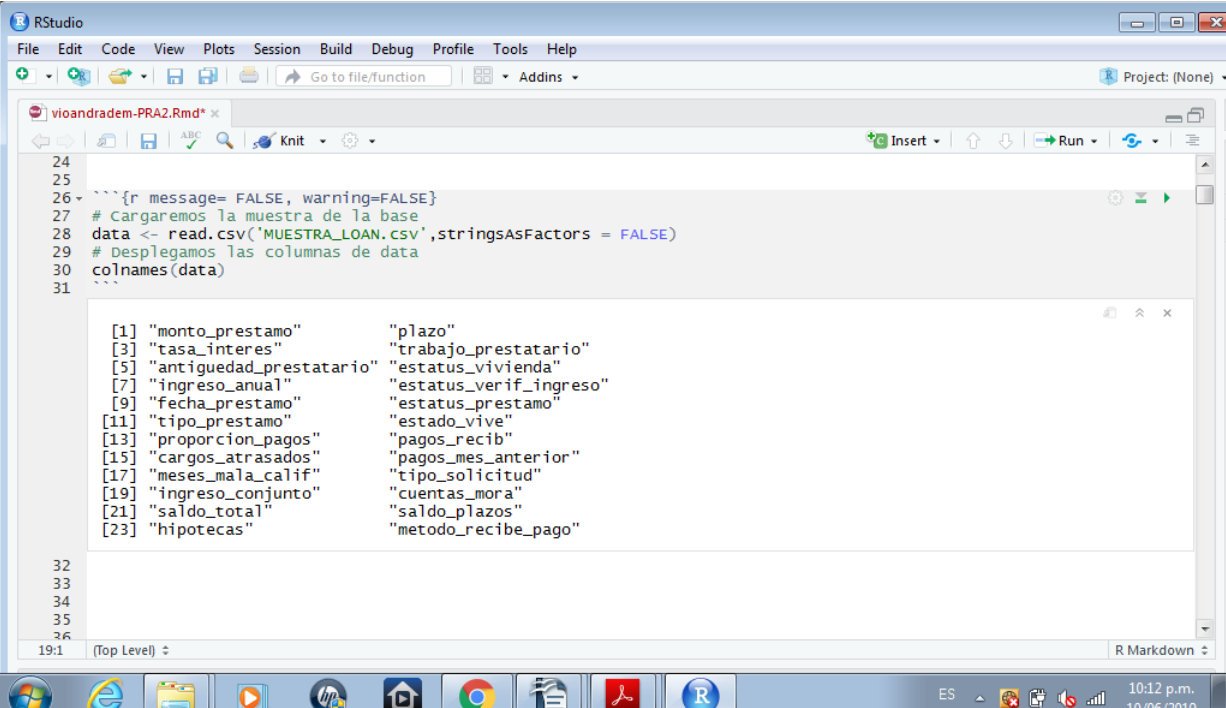
1. monto\_prestamo. Es el monto del préstamo solicitado.
2. plazo. Plazo del préstamo
3. tasa\_interes. Tasa del préstamo
4. trabajo\_prestatario. Título del trabajo del prestatario
5. antigüedad\_prestatario. Antigüedad en su trabajo.
6. estatus\_vivienda. Estatus de la vivienda del prestatario
7. ingreso\_anual. Ingreso anual del prestatario
8. estatus\_verif\_ingreso. Verificación del ingreso del prestatario
9. fecha\_prestamo. Fecha en que se dió el préstamo
10. estatus\_prestamo. Estatus actual de préstamo
11. tipo\_prestamo. Para que se utilizó el préstamo

12. estado\_vive. Estado en el que vive el prestatario
13. proporcion\_pagos. Relación entre pagos de préstamo y el ingreso
14. pagos\_recib. Pagos ya realizados
15. cargos\_atrasados. Cargos atrasados que ya se pagaron
16. pago\_mes\_anterior. Indica si se recibió el pago anterior
17. meses\_mala\_calif. Hace cuántos meses tuvo una mala calificación
18. tipo\_solicitud. Si realizó una solicitud en conjunto
19. ingreso\_conjunto. Ingreso del conjunto de las personas
20. cuentas\_mora. Número de cuentas que tiene el prestatario en mora
21. saldo\_total. Saldo total actual de las cuentas del prestatario
22. saldo\_plazos. Saldo actual de las cuentas a plazos
23. hipotecas. Número de hipotecas que tiene el prestatario
24. metodo\_recibe\_prestamo. Forma en que recibe el préstamo el prestatario

A la muestra aleatoria se le llamó MUESTRA\_LOAN y es la base con la que trabajaremos en esta práctica.

### 3. Limpieza de los datos

Lo primero que se realizará es cargar la base de datos MUESTRA\_LOAN:



```

24
25
26 {r message= FALSE, warning=FALSE}
27 # Cargaremos la muestra de la base
28 data <- read.csv('MUESTRA_LOAN.csv', stringsAsFactors = FALSE)
29 # Desplegamos las columnas de data
30 colnames(data)
31

```

[1]	"monto_prestamo"	"plazo"
[3]	"tasa_interes"	"trabajo_prestatario"
[5]	"antiguedad_prestatario"	"estatus_vivienda"
[7]	"ingreso_anual"	"estatus_verif_ingreso"
[9]	"fecha_prestamo"	"estatus_prestamo"
[11]	"tipo_prestamo"	"estado_vive"
[13]	"proporcion_pagos"	"pagos_recib"
[15]	"cargos_atrasados"	"pagos_mes_anterior"
[17]	"meses_mala_calif"	"tipo_solicitud"
[19]	"ingreso_conjunto"	"cuentas_mora"
[21]	"saldo_total"	"saldo_plazos"
[23]	"hipotecas"	"metodo_recibe_pago"

Se eliminaron los dos id que contenía la base original, esto se hizo porque ambos venían completamente vacíos. Por esta razón añadiremos un consecutivo que servirá como id para esta nueva base, se llamará id.

Se muestra la forma de la base:

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ - Go to file/function Addins Project: (None)

vioandradem-PRA2.Rmd x
ABC Knit Insert Run

$ monto_prestamo : int 10450 14000 11500 11000 4800 5000 30000 35000 23000 32000 ...
$ plazo          : chr "36 months" "36 months" "36 months" "36 months" ...
$ tasa_interes   : num 18.94 14.47 8.19 6.46 18.94 ...
$ trabajo_prestatario : chr "Project Manager" "Rad tech" "President" "School Readiness Teacher" ...
$ antiguedad_prestatario : chr "10+ years" "10+ years" "10+ years" "3 years" ...
$ estatus_vivienda : chr "MORTGAGE" "MORTGAGE" "MORTGAGE" "ANY" ...
$ ingreso_anual   : num 70000 36000 173370 45000 28000 ...
$ estatus_verif_ingreso : chr "Source verified" "Not verified" "Not verified" "Not verified" ...
$ fecha_prestamo  : chr "01-12-2018" "01-12-2018" "01-12-2018" "01-12-2018" ...
$ estatus_prestamo : chr "Current" "Current" "Current" "Current" ...
$ tipo_prestamo   : chr "Debt consolidation" "Credit card refinancing" "Credit card refinancing" "Credit card refinancing" ...
card refinancing ...
$ estado_vive     : chr "TN" "NE" "MN" "MN" ...
$ proporcion_pagos : num 27.4 28.1 25.3 21.7 12.6 ...
$ pagos_recib     : num 377 946 374 668 316 ...
$ cargos_atrasados : num 0 0 0 0 0 0 0 0 ...
$ pagos_mes_anterior : chr "01-02-2019" "01-02-2019" "01-02-2019" "01-02-2019" ...
$ meses_mala_calif : int NA 28 NA NA NA NA NA 46 NA NA ...
$ tipo_solicitud  : chr "Individual" "Individual" "Individual" "Individual" ...
$ ingreso_conjunto : num NA NA NA NA NA NA 107000 NA NA NA ...
$ cuentas_mora     : int 0 0 0 0 0 0 0 0 ...
$ saldo_total      : int 410636 40699 509992 91997 9887 195476 219255 57866 16223 63815 ...
$ saldo_plazos     : int 6660 34579 46707 80176 0 37602 22255 21465 938 5233 ...
$ hipotecas        : int 1 0 2 0 0 1 4 2 0 3 ...
$ metodo_recibe_pago : chr "Cash" "Cash" "Cash" "DirectPay" ...

30:1 (Top Level) R Markdown
Console

```

Ahora comenzaremos con la exploración y la limpieza de los datos. Empezamos con identificar si es que existen valores nulos en alguna de las variables elegidas:

```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ - Go to file/function Addins Project: (None)

vioandradem-PRA2.Rmd x
ABC Knit Insert Run

42 Ahora comenzaremos con la exploración y la limpieza de los datos. Empezamos con identificar si es que existen
valores nulos en alguna de las variables elegidas:
43 {r}
44 colSums(is.na(data))
45 colSums(data=="")
46

monto_prestamo      plazo      tasa_interes
0                    0
trabajo_prestatario antiguedad_prestatario estatus_vivienda
0                    0
ingreso_anual        estatus_verif_ingreso fecha_prestamo
0                    0
estatus_prestamo     tipo_prestamo      estado_vive
0                    0
proporcion_pagos     pagos_recib        cargos_atrasados
8                    0
pagos_mes_anterior   meses_mala_calif    tipo_solicitud
0                    7348
ingreso_conjunto     cuentas_mora        saldo_total
9423                 0
saldo_plazos         hipotecas          metodo_recibe_pago
3753                 190
id
0

43:7 Chunk 4 R Markdown
Console

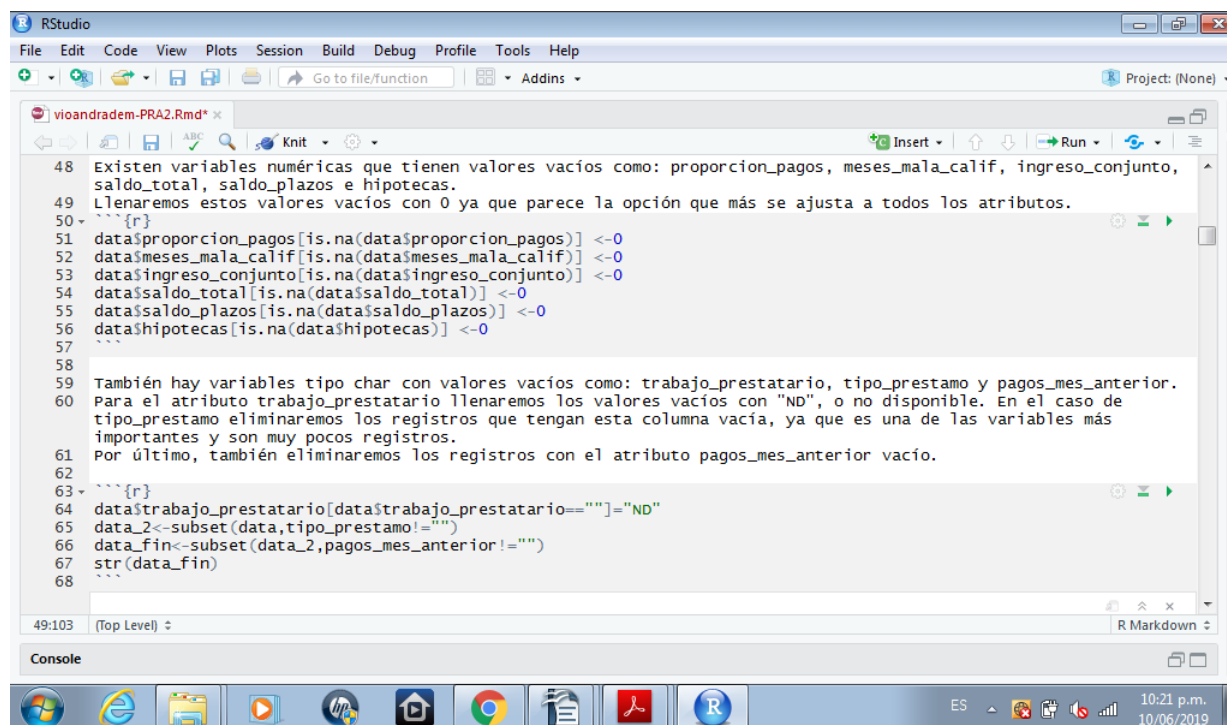
```

Existen variables numéricas que tienen valores vacíos como: `proporcion_pagos`, `meses_mala_calif`, `ingreso_conjunto`, `saldo_total`, `saldo_plazos` e `hipotecas`.  
Llenaremos estos valores vacíos con 0 ya que parece la opción que más se ajusta a todos los atributos.

También hay variables tipo char con valores vacíos como: `trabajo_prestatario`, `tipo_prestamo` y `pagos_mes_anterior`.

Para el atributo `trabajo_prestatario` llenaremos los valores vacíos con "ND", o no disponible. En el caso de `tipo_prestamo` eliminaremos los registros que tengan esta columna vacía, ya que es una de las variables más importantes y son muy pocos registros.

Por último, también eliminaremos los registros con el atributo `pagos_mes_anterior` vacío.

The image is a screenshot of the RStudio interface. The main window displays an R script file named 'vioandradem-PRA2.Rmd'. The script contains two main sections of code. The first section, starting at line 48, is a comment explaining that several numeric variables have NA values and will be replaced with 0. This is followed by a code block (lines 51-56) that uses the `data$variable[is.na(data$variable)] <- 0` pattern for `proporcion_pagos`, `meses_mala_calif`, `ingreso_conjunto`, `saldo_total`, `saldo_plazos`, and `hipotecas`. The second section, starting at line 59, is a comment explaining that character variables have NA values and will be replaced with 'ND' or removed. This is followed by a code block (lines 64-67) that first replaces NA values in `trabajo_prestatario` with 'ND', then subsets the data to remove rows where `tipo_prestamo` is NA, and finally subsets the data to remove rows where `pagos_mes_anterior` is NA. The console at the bottom is empty. The Windows taskbar at the very bottom shows the date as 10/06/2019 and the time as 10:21 p.m.

Se eliminaron 134 registros de la base muestra y nos quedan 9866 registros.  
A continuación discretizamos algunas variables:

The screenshot shows the RStudio interface with a script editor containing the following code:

```

70 Se eliminaron 134 registros de la base muestra y nos quedan 9866 registros.
71 A continuación veremos que variables sería posible discretizar.
72
73 ```{r}
74 apply(data_fin,2, function(x) length(unique(x)))
75

```

The output of the `apply` function is displayed in a text window, showing the number of unique values for each variable in the dataset:

Variable	Number of Unique Values
monto_prestamo	871
plazo	2
tasa_interes	413
trabajo_prestatario	5521
antiguedad_prestatario	12
estatus_vivienda	5
ingreso_anual	1410
estatus_verif_ingreso	3
fecha_prestamo	125
estatus_prestamo	8
tipo_prestamo	559
estado_vive	50
proporcion_pagos	3411
pagos_recib	9743
cargos_atrasados	277
pagos_mes_anterior	106
meses_mala_calif	100
tipo_solicitud	2
ingreso_conjunto	296
cuentas_mora	4
saldo_total	9372
saldo_plazos	5271
hipotecas	15
metodo_recibe_pago	2
id	9866

Discretizamos las variables de hasta 8 clases, con excepción de la variable `cuentas_mora`; esto es porque es una variable en la que el número es importante.

The screenshot shows the RStudio interface with a script editor containing the following code:

```

76
77 Discretizamos las variables de hasta 8 clases, con excepción de la variable cuentas_mora; esto es porque es una
78 variable en la que el número es importante.
79 ```{r}
80 # Discretizamos las variables con hasta 8 clases
81 cols<-c("plazo","estatus_vivienda","estatus_verif_ingreso","estatus_prestamo",
82 "tipo_solicitud","metodo_recibe_pago")
83 for (i in cols){
84   data_fin[,i] <- as.factor(data_fin[,i])
85 }
86 # observamos los resultados
87 str(data_fin)

```

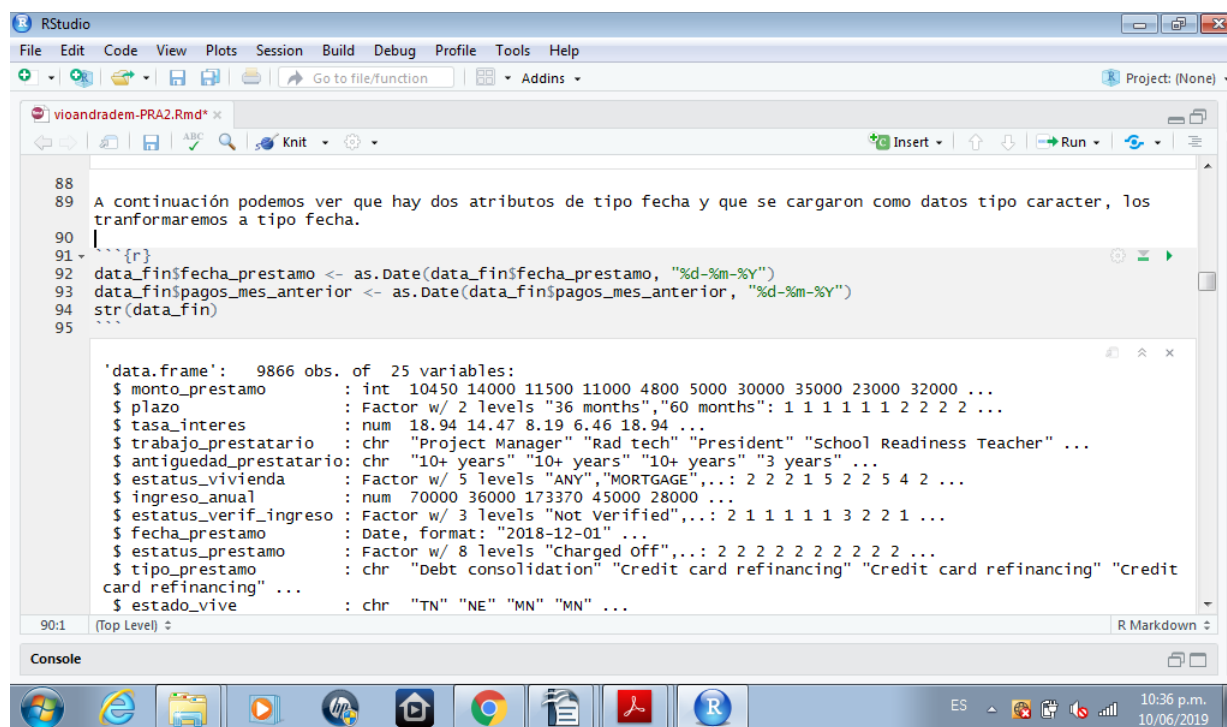
The output of the `str` function is displayed in a text window, showing the structure of the dataset after discretization:

```

'data.frame': 9866 obs. of 25 variables:
 $ monto_prestamo : int 10450 14000 11500 11000 4800 5000 30000 35000 23000 32000 ...
 $ plazo          : Factor w/ 2 levels "36 months","60 months": 1 1 1 1 1 2 2 2 2 ...
 $ tasa_interes   : num 18.94 14.47 8.19 6.46 18.94 ...
 $ trabajo_prestatario: chr "Project Manager" "Rad tech" "President" "School Readiness Teacher" ...
 $ antiguedad_prestatario: chr "10+ years" "10+ years" "10+ years" "3 years" ...
 $ estatus_vivienda : Factor w/ 5 levels "ANY","MORTGAGE",...: 2 2 2 1 5 2 2 5 4 2 ...
 $ ingreso_anual   : num 70000 36000 173370 45000 28000 ...
 $ estatus_verif_ingreso : Factor w/ 3 levels "Not Verified",...: 2 1 1 1 1 1 3 2 2 1 ...
 $ fecha_prestamo  : chr "01-12-2018" "01-12-2018" "01-12-2018" "01-12-2018" ...

```

A continuación podemos ver que hay dos atributos de tipo fecha y que se cargaron como datos tipo caracter, los transformaremos a tipo fecha.



The screenshot shows the RStudio interface. The script editor contains the following R code:

```
88  
89 A continuación podemos ver que hay dos atributos de tipo fecha y que se cargaron como datos tipo caracter, los  
90 transformaremos a tipo fecha.  
91  
92 {r}  
93 data_fin$fecha_prestamo <- as.Date(data_fin$fecha_prestamo, "%d-%m-%Y")  
94 data_fin$pagos_mes_anterior <- as.Date(data_fin$pagos_mes_anterior, "%d-%m-%Y")  
95 str(data_fin)
```

The console output shows the structure of the data frame:

```
'data.frame': 9866 obs. of 25 variables:  
 $ monto_prestamo : int 10450 14000 11500 11000 4800 5000 30000 35000 23000 32000 ...  
 $ plazo : Factor w/ 2 levels "36 months","60 months": 1 1 1 1 1 1 2 2 2 2 ...  
 $ tasa_interes : num 18.94 14.47 8.19 6.46 18.94 ...  
 $ trabajo_prestatario : chr "Project Manager" "Rad tech" "President" "School Readiness Teacher" ...  
 $ antiguedad_prestatario: chr "10+ years" "10+ years" "10+ years" "3 years" ...  
 $ estatus_vivienda : Factor w/ 5 levels "ANY","MORTGAGE",...: 2 2 2 1 5 2 2 5 4 2 ...  
 $ ingreso_anual : num 70000 36000 173370 45000 28000 ...  
 $ estatus_verif_ingreso : Factor w/ 3 levels "Not Verified",...: 2 1 1 1 1 1 3 2 2 1 ...  
 $ fecha_prestamo : Date, format: "2018-12-01" ...  
 $ estatus_prestamo : Factor w/ 8 levels "Charged Off",...: 2 2 2 2 2 2 2 2 2 ...  
 $ tipo_prestamo : chr "Debt consolidation" "Credit card refinancing" "Credit card refinancing" "Credit  
card refinancing" ...  
 $ estado_vive : chr "TN" "NE" "MN" "MN" ...
```

## 4. Análisis de los datos

## 5. Representación gráfica de los resultados

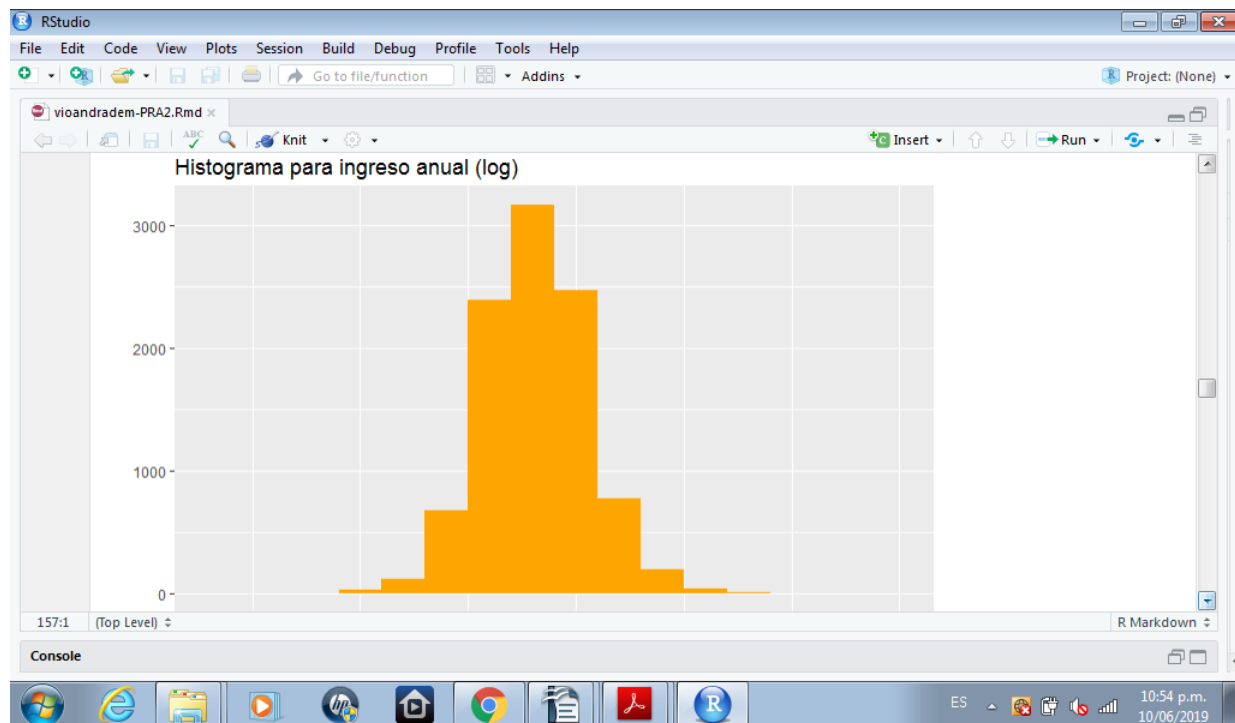
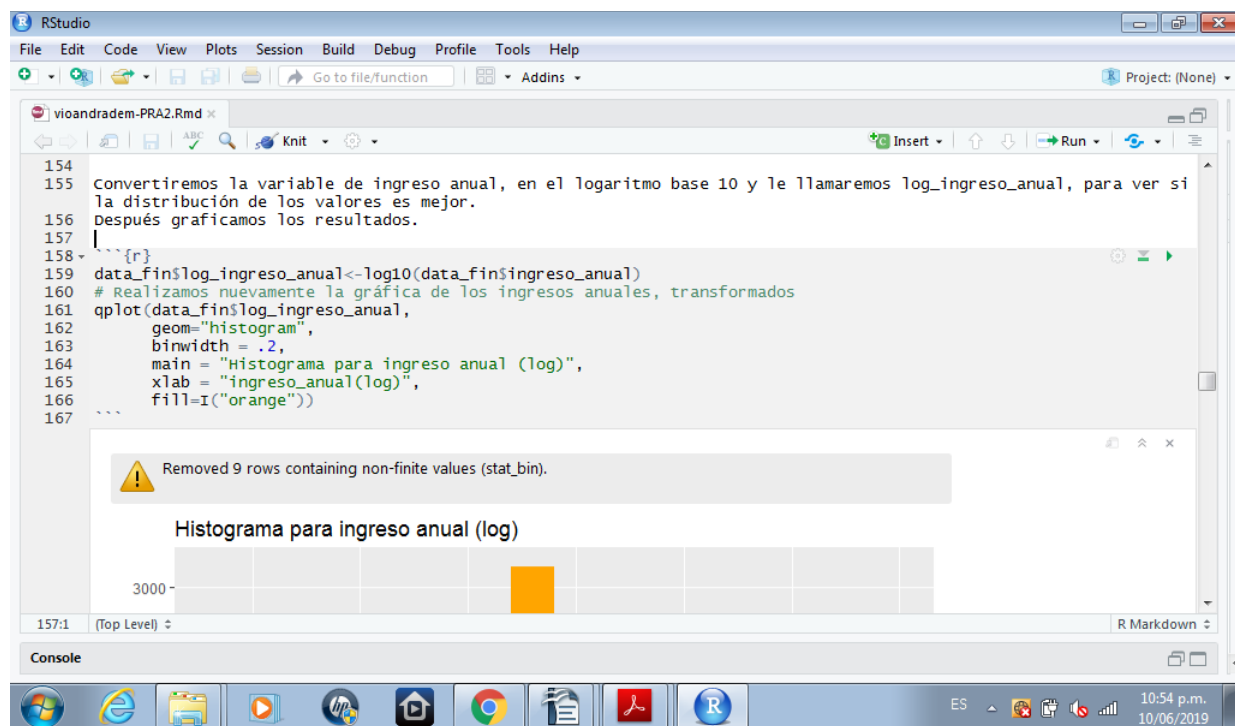
Realizaremos a continuación algunas gráficas que junto con el resumen anterior, nos ayuden a darnos una idea de como son los datos numéricos más importantes de la base:







Convertiremos la variable de ingreso anual, en el logaritmo base 10 y le llamaremos log\_ingreso\_anual, para ver si la distribución de los valores es mejor. Después graficamos los resultados.



Al convertir la variable se muestra una nota, de que 9 valores fueron eliminados, debido a que salen del rango.

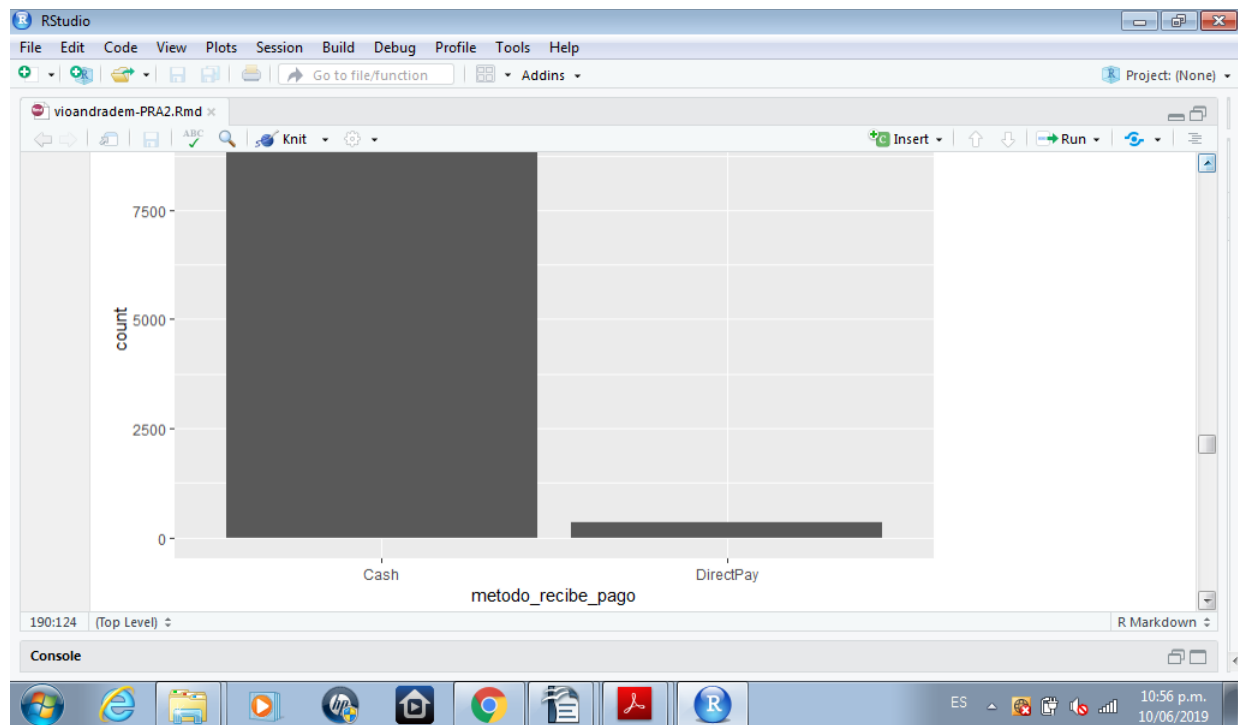
De cualquier forma, utilizaremos esta variable transformada para continuar con el análisis, ya que tiene una mejor distribución que la variable original. Solamente hay que tener en cuenta que una vez realizado el análisis se tiene que transformar inversamente la variable para que tenga sentido el resultado final.

Ahora realizaremos el análisis de las variables categóricas.

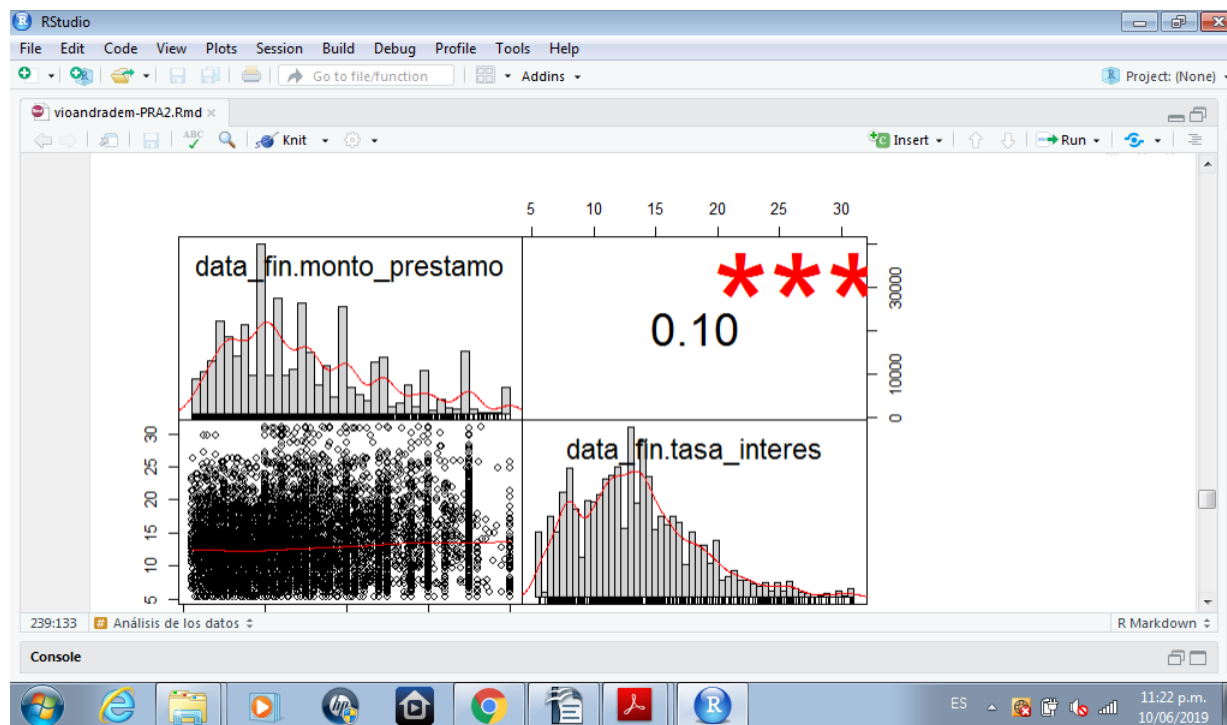
Podemos ver con estas tablas las siguientes afirmaciones:

1. La mayoría de los préstamos son a 36 meses de plazo
2. Los estatus más comunes en el caso de la vivienda son: MORTGAGE y RENT
3. En el caso del estatus de la verificación del ingreso, la distribución en las 3 clases es muy parecida
4. Los estatus del préstamo más comunes son: Current y Fully Paid.
5. Hay muy pocos préstamos que están retrasados (121)

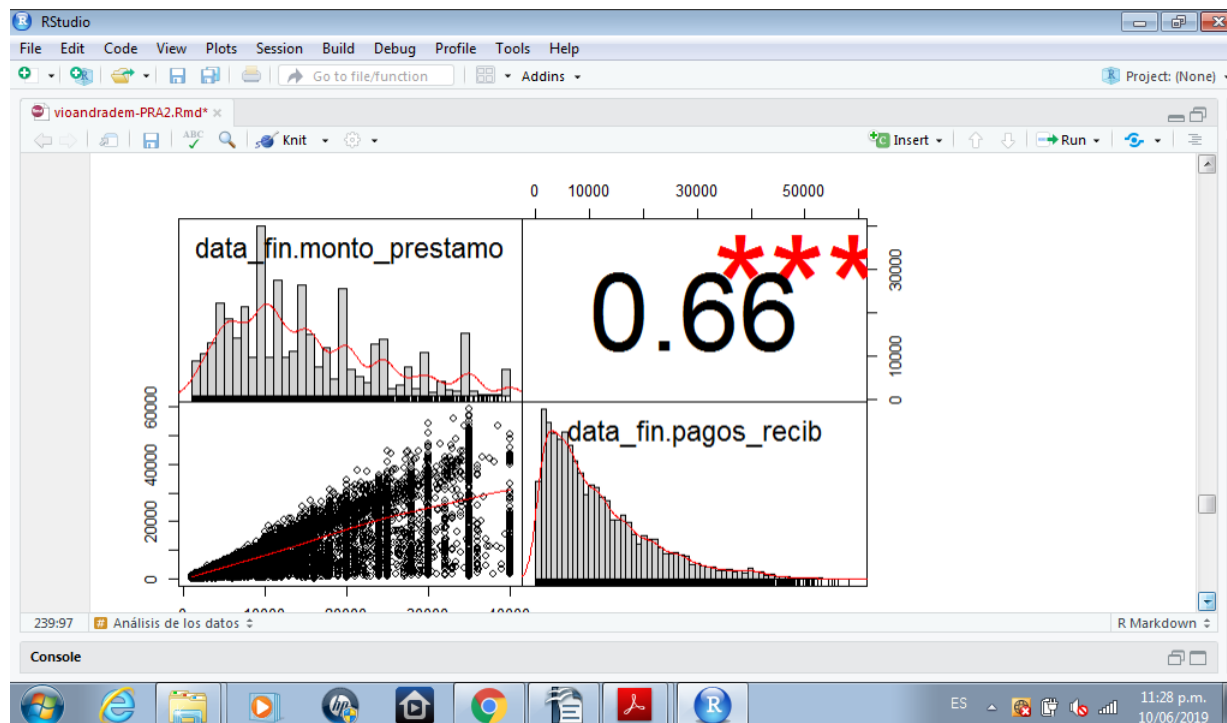
Veamos ahora algunos graficos adicionales. El primero nos muestra el numero de préstamos que están a 36 meses y a 60 meses.



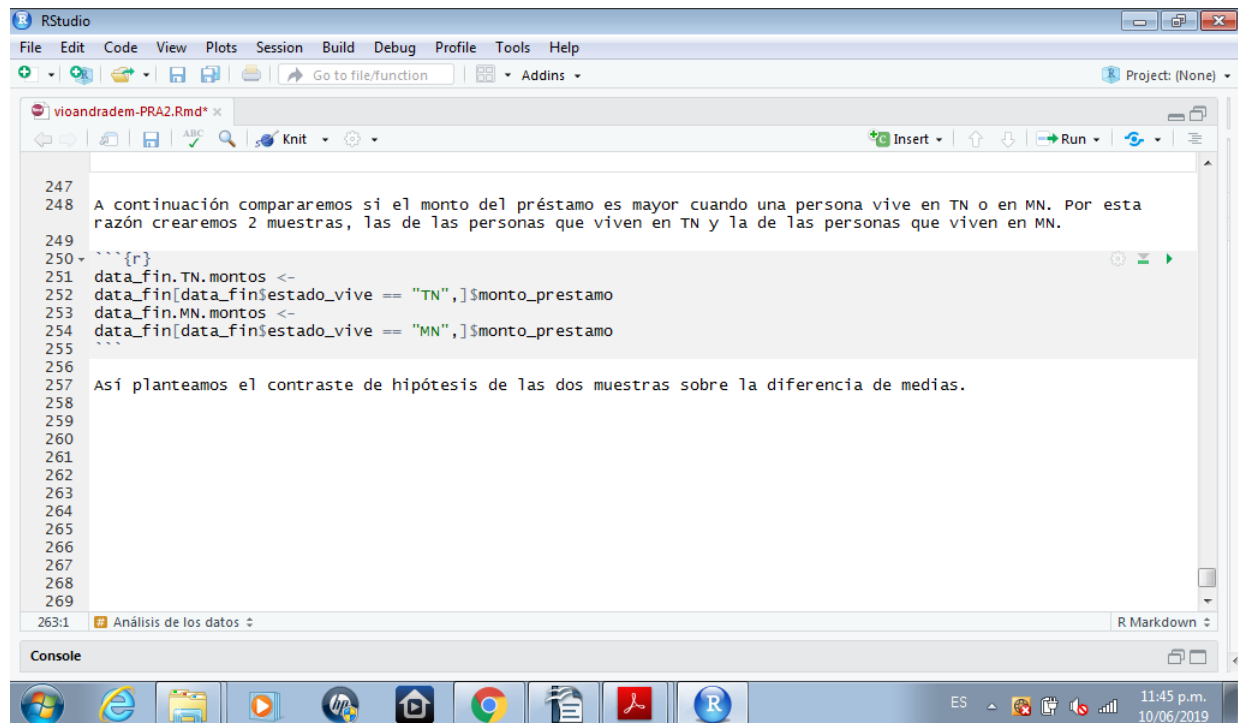
Realizaremos un análisis de correlación entre las variables monto\_prestamo y la tasa de interés.



También realizaremos un análisis de correlación entre las variables monto\_prestamo y pagos\_recib:

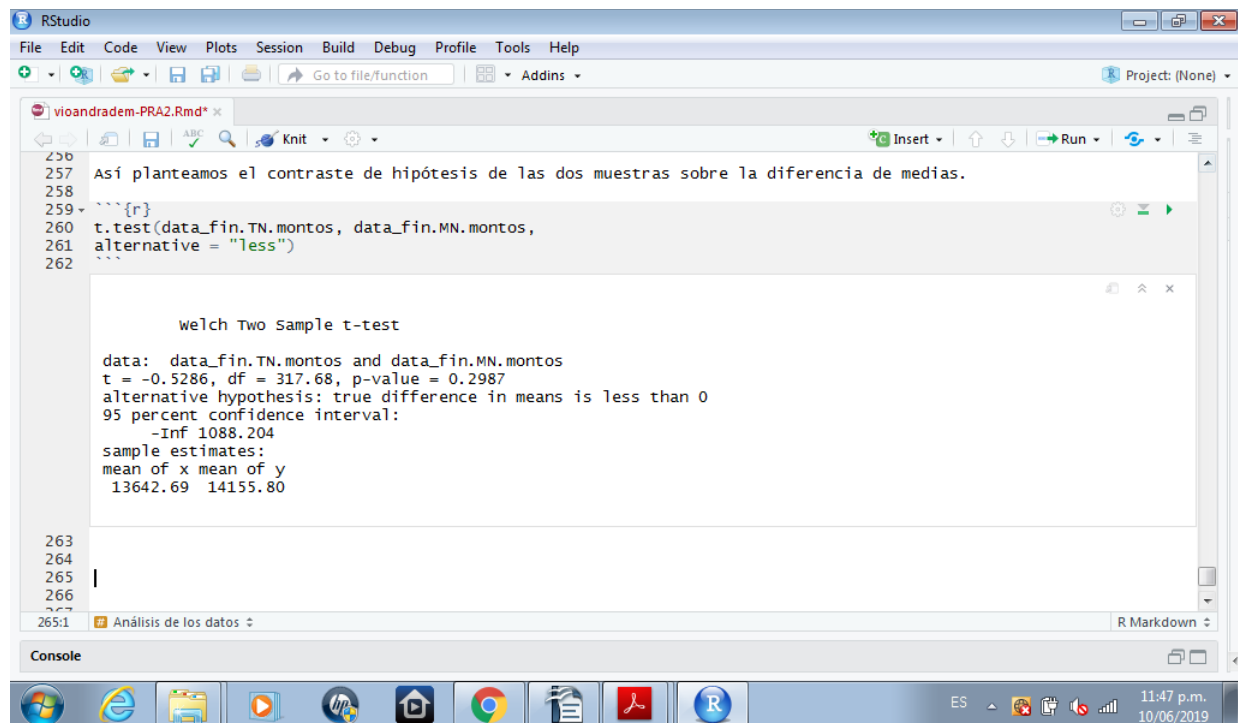


A continuación compararemos si el monto del préstamo es mayor cuando una persona vive en TN o en MN. Por esta razón crearemos 2 muestras, las de las personas que viven en TN y la de las personas que viven en MN.



The screenshot shows the RStudio interface with a script editor containing R code. The code filters a dataset into two groups based on the 'estado\_vive' variable: 'TN' and 'MN'. It then assigns the 'monto\_prestamo' values to two separate vectors, 'data\_fin.TN.montos' and 'data\_fin.MN.montos'. A comment in Spanish explains the purpose of creating these two samples.

```
247  
248 A continuación compararemos si el monto del préstamo es mayor cuando una persona vive en TN o en MN. Por esta  
249 razón crearemos 2 muestras, las de las personas que viven en TN y la de las personas que viven en MN.  
250  
251 ```{r}  
252 data_fin.TN.montos <-  
253 data_fin[data_fin$estado_vive == "TN",]$monto_prestamo  
254 data_fin.MN.montos <-  
255 data_fin[data_fin$estado_vive == "MN",]$monto_prestamo  
256  
257 Así planteamos el contraste de hipótesis de las dos muestras sobre la diferencia de medias.  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
263:1 Análisis de los datos
```



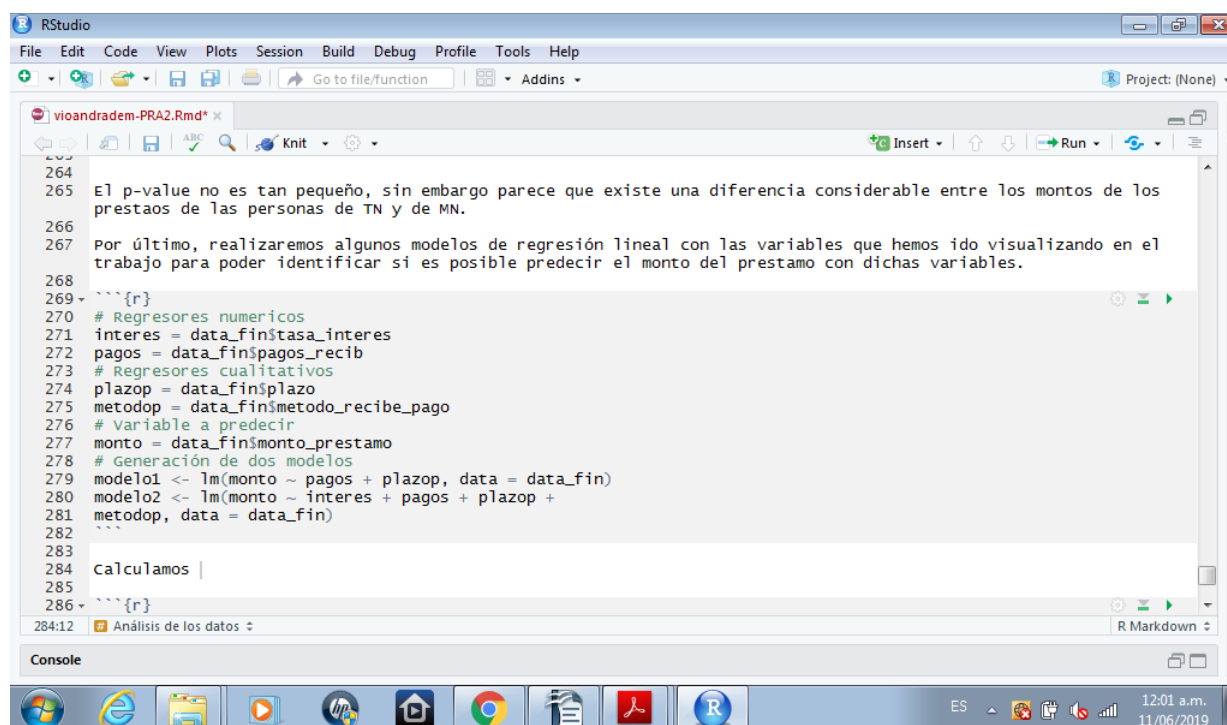
The screenshot shows the RStudio interface with the same script editor. The code now performs a Welch Two Sample t-test on the two vectors created in the previous step. The output of the test is displayed in a text box, showing the t-statistic, degrees of freedom, p-value, and a 95% confidence interval.

```
256  
257 Así planteamos el contraste de hipótesis de las dos muestras sobre la diferencia de medias.  
258  
259 ```{r}  
260 t.test(data_fin.TN.montos, data_fin.MN.montos,  
261 alternative = "less")  
262  
263  
264  
265  
266  
267  
263:1 Análisis de los datos
```

wech Two sample t-test

data: data\_fin.TN.montos and data\_fin.MN.montos  
t = -0.5286, df = 317.68, p-value = 0.2987  
alternative hypothesis: true difference in means is less than 0  
95 percent confidence interval:  
-Inf 1088.204  
sample estimates:  
mean of x mean of y  
13642.69 14155.80

Por último, realizaremos algunos modelos de regresión lineal con las variables que hemos ido visualizando en el trabajo para poder identificar si es posible predecir el monto del préstamo con dichas variables.

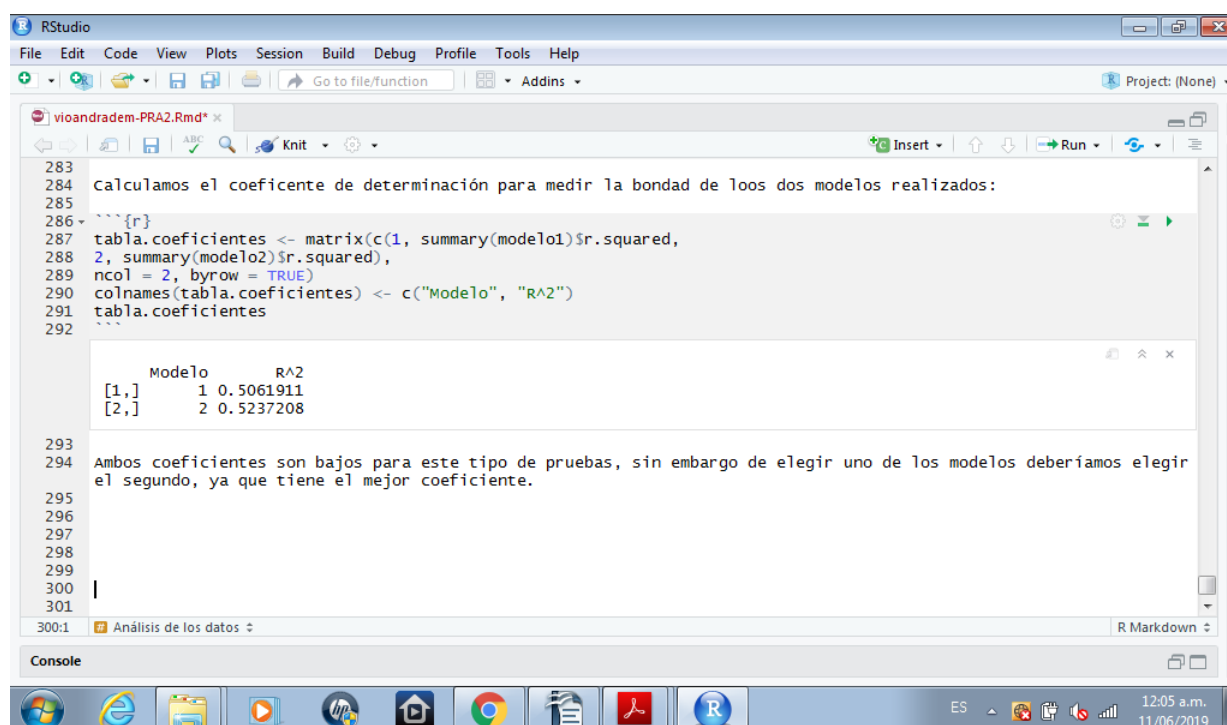


The screenshot shows the RStudio interface with a script editor containing R code. The code defines two linear regression models, 'modelo1' and 'modelo2', using the 'lm' function. 'modelo1' uses 'pagos' and 'plazop' as predictors, while 'modelo2' uses 'interes', 'pagos', and 'plazop'. The response variable for both is 'monto'. The code is written in Spanish and includes comments in both languages.

```

264
265 el p-value no es tan pequeño, sin embargo parece que existe una diferencia considerable entre los montos de los
    prestaos de las personas de TN y de MN.
266
267 Por último, realizaremos algunos modelos de regresión lineal con las variables que hemos ido visualizando en el
    trabajo para poder identificar si es posible predecir el monto del préstamo con dichas variables.
268
269 ```{r}
270 # Regresores numericos
271 interes = data_fin$interes
272 pagos = data_fin$pagos_recib
273 # Regresores cualitativos
274 plazop = data_fin$plazo
275 metodop = data_fin$metodo_recibe_pago
276 # Variable a predecir
277 monto = data_fin$monto_prestamo
278 # Generación de dos modelos
279 modelo1 <- lm(monto ~ pagos + plazop, data = data_fin)
280 modelo2 <- lm(monto ~ interes + pagos + plazop +
281 metodop, data = data_fin)
282
283
284 Calculamos |
285
286 ```{r}
284:12  Análisis de los datos
  
```

Calculamos el coeficiente de determinación para medir la bondad de los dos modelos realizados:



The screenshot shows the RStudio interface with a script editor. The code calculates the coefficient of determination (R-squared) for the two models. It uses the 'summary' function to get the R-squared values and then creates a matrix 'tabla.coeficientes' to store them. The output of the matrix is displayed in a window. The code is written in Spanish and includes comments in both languages.

```

283
284 Calculamos el coeficiente de determinación para medir la bondad de los dos modelos realizados:
285
286 ```{r}
287 tabla.coeficientes <- matrix(c(1, summary(modelo1)$r.squared,
288 2, summary(modelo2)$r.squared),
289 ncol = 2, byrow = TRUE)
290 colnames(tabla.coeficientes) <- c("Modelo", "R^2")
291 tabla.coeficientes
292
293
294 Ambos coeficientes son bajos para este tipo de pruebas, sin embargo de elegir uno de los modelos deberíamos elegir
    el segundo, ya que tiene el mejor coeficiente.
295
296
297
298
299
300
301
300:1  Análisis de los datos
  
```

Modelo	R <sup>2</sup>
[1,]	1 0.5061911
[2,]	2 0.5237208

## 6. Conclusiones

Con la realización de esta práctica pudimos observar varias cosas acerca de las etapas principales de un proyecto analítico. Por ejemplo la importancia de elegir una base de datos que tenga sentido para nosotros y que se pueda interpretar fácilmente.

Comenzamos la práctica identificando si es que había valores nulos o vacíos, a continuación hicimos algunos cambios de variables por variables categóricas, limpiamos fechas y eliminamos columnas vacías. Después analizamos algunas variables para identificar su comportamiento, esto se realizó en su mayoría con gráficas para poder visualizar las características de las variables.

Por último realizamos pruebas estadísticas de la base de datos y elegimos una variable objetivo para predecir su valor con respecto a las otras variables.

Si bien estas pruebas no fueron de gran ayuda, se puede identificar que con más análisis de las variables y con más pruebas se podrían encontrar modelos más avanzados que nos permitirán lograr el objetivo en algún momento.