Práctica 2: Limpieza y análisis de datos

Maite Gracia

30 de December, 2020

Contents

1 Descripción del dataset

Se ha decidido utilizar un dataset de la web Kaggle para la presente práctica. [**enlace**](https://www.kaggle.com/kemical/kickstarter-projects) 8 de la web Kickstarter. Kickstarter es una plataforma de micro mecenazgo, es decir, gente de todo el mundo ayuda a financiar las ideas y proyectos de pequeñas empresas o particulares.

En la web de Kickstarter se pueden encontrar miles de campañas que buscan financiación para desarrollar productos de todo tipo. Desde películas independientes, a juegos de mesa o ropa, peluches, libros etc. Cada una de estas campañas tendrá un periodo de tiempo en el que cualquiera podrá aportar dinero al proyecto y si se consigue llegar al límite de dinero requerido la campaña será fundada.

Yo personalmente utilicé Kickstarter hace unos años para lanzar una serie de productos lo cual es una de las razones por las que he elegido el presente dataset. El objetivo principal sería poder crear un modelo que predijera que probabilidad tiene cualquier tipo de producto de conseguir recaudar dinero mediante una campaña de Kickstarter antes de ser lanzado.

2 Integración y selección de los datos de interés a analizar

Las variables que componen el dataset son:

- ID: identificador interno de Kickstarter
- name: nombre del proyecto
- category: categoría específica en la que se encuentra el proyecto
- main_category: categoría principal de la campaña
- currency: divisa en la que se creó el proyecto
- deadline: fecha límite
- goal: cantidad de dinero que el creador necesita para completar el proyecto
- launched: fecha lanzamiento
- pledged: cantidad total aportada al proyecto
- state: condición en la que se encuentra el Proyecto (failed, successful, canceled, live, undefined)
- backers: total de mecenas.
- country: país en el que se encuentra el Proyecto.
- usd_pledged: conversión en dólares de la columna pledged hecha por Kickstarter
- usd_pledged_real: conversión en dólares de la columna pledged hecha a través de Fixer.io API
- usd_goal_real: conversión en dólares de la columna goal hecha a través de Fixer.io API

Antes de cargar el archivo en R se hace una inspección de los datos. Al tratarse de un archivo con extensión .csv, hay que cerciorarse del tipo de separador utilizado (en este caso la ",") y posteriormente se procede a su carga teniendo en cuenta el separador antes mencionado:

```
# Asignamos los datos del fichero cargado a una variable denominada dataSet
dataSet <- read.csv('../data/ks-projects-201801.csv')
nrow(dataSet)
```

```
## [1] 378661
```

```
names(dataSet)
```

```
##  [1] "ID"               "name"             "category"         "main_category"
##  [5] "currency"         "deadline"         "goal"             "launched"
##  [9] "pledged"          "state"            "backers"          "country"
## [13] "usd.pledged"      "usd_pledged_real" "usd_goal_real"
```

Vemos que el dataset original se compone de 378,661 muestras y 13 variables. Ya que se trata de una cantidad de muestras muy elevadas, se ha decidido aplicar una técnica para reducir la cantidad de estas, se empleará la técnica de muestreo aleatorio simple sin sustitución, es decir, se van a extraer 3000 muestras aleatorias del conjunto de datos, donde la probabilidad de escoger cada una de las muestras será la misma para todas, 1/378,661.

Para ello generaremos un fichero al que llamaremos sample_ks.csv que contendrá las 3000 muestras.

```
library(sampling)
indices <- sample( 1:nrow( dataSet ), 3000 )
dataSet <- dataSet[ indices, ]
```

A partir de ahora cuando se haga referencia al dataset, estaremos hablando del dataset que cotiene las 3000 muestras, no el dataset original.

3 Limpieza de los datos

```
# Muestra de las 5 primeras líneas del dataset completo
head(dataSet, 5)
```

```
##                ID                                                name
## 79159  1402662983      FUBAR PRESS/ 215 INK: FREE COMIC BOOK DAY 2012.
## 337068   78689199                  75 year commemorative flight in 2019
## 375859  985634609 Elijah Cross records "Revenge" (finally)! (Canceled)
## 252420  353862189                       Project: Steam World (Single)
## 315091  674867929                  Billy in Motion - a USC thesis film
##           category main_category currency   deadline    goal
## 79159  Comic Books        Comics      USD 2012-04-01    1200
## 337068      Flight    Technology      SEK 2015-10-04 1824000
## 375859  Indie Rock         Music      USD 2011-05-15    3500
## 252420       Music         Music      USD 2013-09-10    1100
## 315091      Shorts  Film & Video      USD 2012-10-25    6000
##                   launched pledged      state backers country usd.pledged
## 79159  2012-03-16 04:43:06 3512.01 successful     112      US     3512.01
## 337068 2015-09-07 10:57:27  190.00     failed       3      SE       22.52
## 375859 2011-04-05 09:20:58   90.00   canceled       4      US       90.00
## 252420 2013-07-12 23:33:30  433.00     failed       4      US      433.00
## 315091 2012-09-25 00:57:00 7547.00 successful      95      US     7547.00
##        usd_pledged_real usd_goal_real
## 79159           3512.01        1200.0
## 337068            23.29      223603.4
## 375859            90.00        3500.0
## 252420           433.00        1100.0
## 315091          7547.00        6000.0
```

```
# Análisis descriptivo del dataset
summary(dataSet)
```

```
##        ID                 name              category         main_category
##  Min.   :1.829e+05   Length:3000        Length:3000        Length:3000
##  1st Qu.:5.171e+08   Class :character   Class :character   Class :character
##  Median :1.049e+09   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1.051e+09
##  3rd Qu.:1.564e+09
##  Max.   :2.147e+09
##
##    currency           deadline              goal            launched
##  Length:3000        Length:3000        Min.   :       1   Length:3000
##  Class :character   Class :character   1st Qu.:    2000   Class :character
##  Mode  :character   Mode  :character   Median :    5000   Mode  :character
##                                        Mean   :   60740
```

```
##                                      3rd Qu.:   17000
##                                      Max.   :55000000
##
##     pledged           state              backers          country
##  Min.   :      0   Length:3000        Min.   :    0.0   Length:3000
##  1st Qu.:     40   Class :character   1st Qu.:    2.0   Class :character
##  Median :    704   Mode  :character   Median :   12.0   Mode  :character
##  Mean   :   8549                      Mean   :  101.9
##  3rd Qu.:   4332                      3rd Qu.:   57.0
##  Max.   :2152285                      Max.   :14971.0
##
##   usd.pledged        usd_pledged_real    usd_goal_real
##  Min.   :      0.0   Min.   :      0.0   Min.   :       1
##  1st Qu.:     20.5   1st Qu.:     40.0   1st Qu.:    2000
##  Median :    461.1   Median :    713.5   Median :    5000
##  Mean   :   6353.1   Mean   :   8301.5   Mean   :   58190
##  3rd Qu.:   3352.8   3rd Qu.:   4273.2   3rd Qu.:   15000
##  Max.   :1255444.0   Max.   :2152285.0   Max.   :55000000
##  NA's   :32
```

```
# Comprobamos si hay NA en el dataset original
sapply(dataSet, function(x) sum(is.na(x)))
```

```
##               ID             name         category    main_category
##                0                0                0                0
##         currency         deadline             goal         launched
##                0                0                0                0
##          pledged            state          backers          country
##                0                0                0                0
##      usd.pledged usd_pledged_real    usd_goal_real
##               32                0                0
```

3.1 Normalización de los datos

Basándonos en la estadística descriptiva de la muestra y en la descripción de cada variable podemos ver que todas las variables menos ID son de tipo carácter. Para poder analizar de forma eficaz los datos haremos las siguientes conversiones:

- Variables category, main_category, currency y country van a convertirse a tipo factor para poder agrupar proyectos.

```
dataSet$category <- as.factor(dataSet$category)
dataSet$main_category <- as.factor(dataSet$main_category)
dataSet$currency <- as.factor(dataSet$currency)
dataSet$country <- as.factor(dataSet$country)

# Valores que toman las variables currency y country
unique(dataSet$currency)
```

```
##  [1] USD SEK GBP NZD EUR MXN CAD CHF DKK AUD HKD NOK SGD
## Levels: AUD CAD CHF DKK EUR GBP HKD MXN NOK NZD SEK SGD USD
```

```
unique(dataSet$country)
```

```
##  [1] US   SE   GB   NZ   BE   MX   CA   DE   IE   CH   N,0" DK   IT   AU   NL
## [16] ES   AT   FR   HK   NO   SG   LU
## 22 Levels: AT AU BE CA CH DE DK ES FR GB HK IE IT LU MX N,0" NL NO NZ SE ... US
```

Vemos que country tiene un carácter especial en algunos de los casos, vamos a sustituirlos por NA y más adelante imputaremos estos valores basándonos en la variable currency.

```
dataSet$country[dataSet$country == 'N,0"'] <- NA
```

- Las variables deadline y launched se convertirán a tipo Date.

```
dataSet$deadline <- as.Date(dataSet$deadline, '%Y-%m-%d')
dataSet$launched <- as.Date(dataSet$launched, '%Y-%m-%d')
```

- goal, pledged y usd.pledged van a pasar a ser tipo numérico.

```
dataSet$goal <- as.numeric(dataSet$goal)
dataSet$pledged <- as.numeric(dataSet$pledged)
dataSet$usd_pledged <- as.numeric(dataSet$usd_pledged)
dataSet$usd_pledged_real <- as.numeric(dataSet$usd_pledged_real)
dataSet$usd_goal_real <- as.numeric(dataSet$usd_goal_real)
```

- La variable state, como ya se ha explicado, detalla el estado en el que acabó o estaba en ese momento la campaña. Vemos que hay 5 estados failed, successful, canceled, suspended y undefined. Ya que undefined no está detallado que significa, se ha decidido añadir una nueva columna status, que contendrá dos valores, 0 si el proyecto no ha sido fundado y 1 si el proyecto ha recaudado los fondos suficientes.

```
dataSet['status'] <- as.factor(ifelse(dataSet$pledged > dataSet$goal, 1, 0))
```

- Se va a añadir una columna nueva euros_pledged que contendrá la conversión de usd_pledged_real a euros. Se utilizará la conversión 1€ = 1.23$ a 19 de diciembre.

```
dataSet['euros_pledged'] <- as.numeric(
  format(as.numeric(dataSet$usd_pledged_real)/1.23), nsmall = 1)
```

- Se va a añadir una columna nueva euros_goal que contendrá la conversión de goal a euros. Se utilizará la conversión 1€ = 1.23$ a 19 de diciembre.

```
dataSet['euros_goal'] <- round(as.numeric(
  format(as.numeric(dataSet$goal)/1.23), nsmall = 1), 2)
```

- Se va a añadir una columna nueva proyect_length de tipo numérico, que contendrá el total de días que el proyecto ha estado abierto a financiación. Esta nueva columna será resultado de la diferencia entre la columna deadline y launched.

```
dataSet['proyect_length'] <- as.numeric(dataSet$deadline - dataSet$launched)
```

```
# Muestra set de datos
head(dataSet, 5)
```

```
##                ID                                                name
## 79159  1402662983      FUBAR PRESS/ 215 INK: FREE COMIC BOOK DAY 2012.
## 337068   78689199                 75 year commemorative flight in 2019
## 375859  985634609 Elijah Cross records "Revenge" (finally)! (Canceled)
## 252420  353862189                       Project: Steam World (Single)
## 315091  674867929                  Billy in Motion - a USC thesis film
##           category main_category currency   deadline    goal   launched pledged
## 79159  Comic Books        Comics      USD 2012-04-01    1200 2012-03-16 3512.01
## 337068      Flight    Technology      SEK 2015-10-04 1824000 2015-09-07  190.00
## 375859  Indie Rock         Music      USD 2011-05-15    3500 2011-04-05   90.00
## 252420       Music         Music      USD 2013-09-10    1100 2013-07-12  433.00
## 315091      Shorts  Film & Video      USD 2012-10-25    6000 2012-09-25 7547.00
##             state backers country usd.pledged usd_pledged_real usd_goal_real
```

```
## 79159  successful     112      US     3512.01          3512.01        1200.0
## 337068     failed       3      SE       22.52            23.29      223603.4
## 375859   canceled       4      US       90.00            90.00        3500.0
## 252420     failed       4      US      433.00           433.00        1100.0
## 315091 successful      95      US     7547.00          7547.00        6000.0
##        usd_pledged status euros_pledged euros_goal proyect_length
## 79159      3512.01      1    2855.29300     975.61             16
## 337068       23.29      0      18.93496 1482927.00             27
## 375859       90.00      0      73.17073    2845.53             40
## 252420      433.00      0     352.03250     894.31             60
## 315091     7547.00      1    6135.77200    4878.05             30
```

```
summary(dataSet)
```

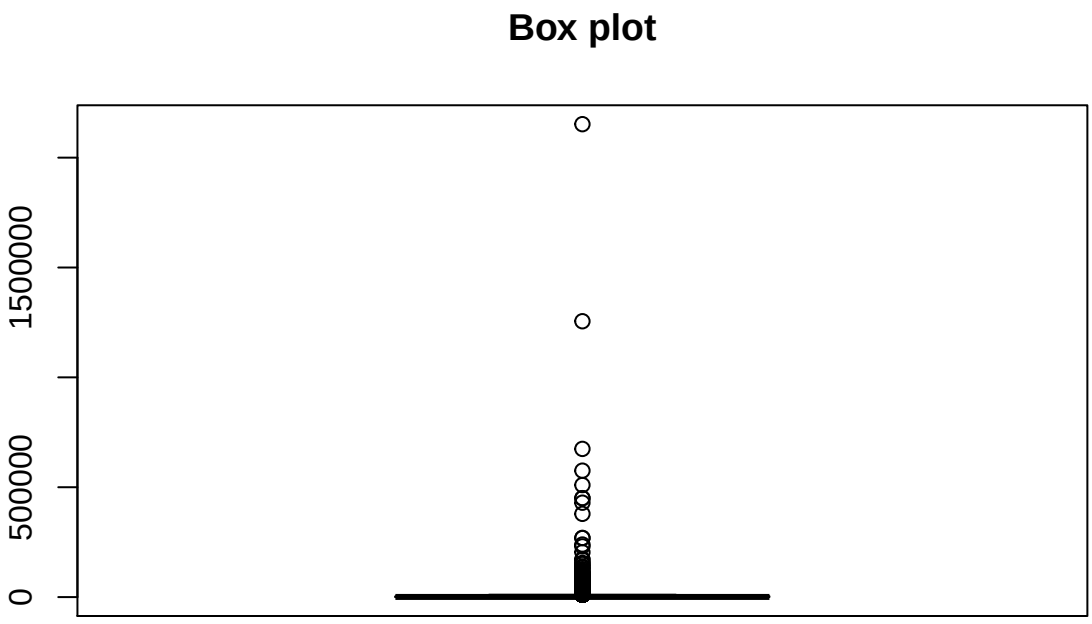
```
##        ID                 name                    category
##  Min.   :1.829e+05   Length:3000        Product Design: 174
##  1st Qu.:5.171e+08   Class :character   Music         : 143
##  Median :1.049e+09   Mode  :character   Documentary   : 133
##  Mean   :1.051e+09                      Tabletop Games: 104
##  3rd Qu.:1.564e+09                      Food          : 103
##  Max.   :2.147e+09                      Shorts        :  90
##                                         (Other)       :2253
##       main_category    currency       deadline                goal
##  Film & Video:507   USD    :2369   Min.   :2009-07-01   Min.   :       1
##  Music       :443   GBP    : 258   1st Qu.:2013-06-13   1st Qu.:    2000
##  Publishing  :319   EUR    : 141   Median :2014-12-31   Median :    5000
##  Technology  :273   CAD    : 107   Mean   :2014-10-22   Mean   :   60740
##  Games       :254   AUD    :  55   3rd Qu.:2016-04-18   3rd Qu.:   17000
##  Design      :234   MXN    :  16   Max.   :2018-02-27   Max.   :55000000
##  (Other)     :970   (Other):  54
##     launched             pledged           state              backers
##  Min.   :2009-04-30   Min.   :      0   Length:3000        Min.   :    0.0
##  1st Qu.:2013-05-09   1st Qu.:     40   Class :character   1st Qu.:    2.0
##  Median :2014-11-24   Median :    704   Mode  :character   Median :   12.0
##  Mean   :2014-09-18   Mean   :   8549                      Mean   :  101.9
##  3rd Qu.:2016-03-15   3rd Qu.:   4332                      3rd Qu.:   57.0
##  Max.   :2018-01-01   Max.   :2152285                      Max.   :14971.0
##
##     country      usd.pledged        usd_pledged_real    usd_goal_real
##  US     :2344   Min.   :      0.0   Min.   :      0.0   Min.   :       1
##  GB     : 257   1st Qu.:     20.5   1st Qu.:     40.0   1st Qu.:    2000
##  CA     : 106   Median :    461.1   Median :    713.5   Median :    5000
##  AU     :  55   Mean   :   6353.1   Mean   :   8301.5   Mean   :   58190
##  DE     :  39   3rd Qu.:   3352.8   3rd Qu.:   4273.2   3rd Qu.:   15000
##  (Other): 167   Max.   :1255444.0   Max.   :2152285.0   Max.   :55000000
##  NA's   :  32   NA's   :32
##   usd_pledged        status   euros_pledged         euros_goal
##  Min.   :      0.0   0:1918   Min.   :      0.0   Min.   :       1
##  1st Qu.:     40.0   1:1082   1st Qu.:     32.5   1st Qu.:    1626
##  Median :    713.5            Median :    580.1   Median :    4065
##  Mean   :   8301.5            Mean   :   6749.1   Mean   :   49382
##  3rd Qu.:   4273.2            3rd Qu.:   3474.1   3rd Qu.:   13821
##  Max.   :2152285.0            Max.   :1749825.0   Max.   :44715450
##
##  proyect_length
```

```
##  Min.   : 1.00
##  1st Qu.:30.00
##  Median :30.00
##  Mean   :34.11
##  3rd Qu.:36.00
##  Max.   :91.00
##
```

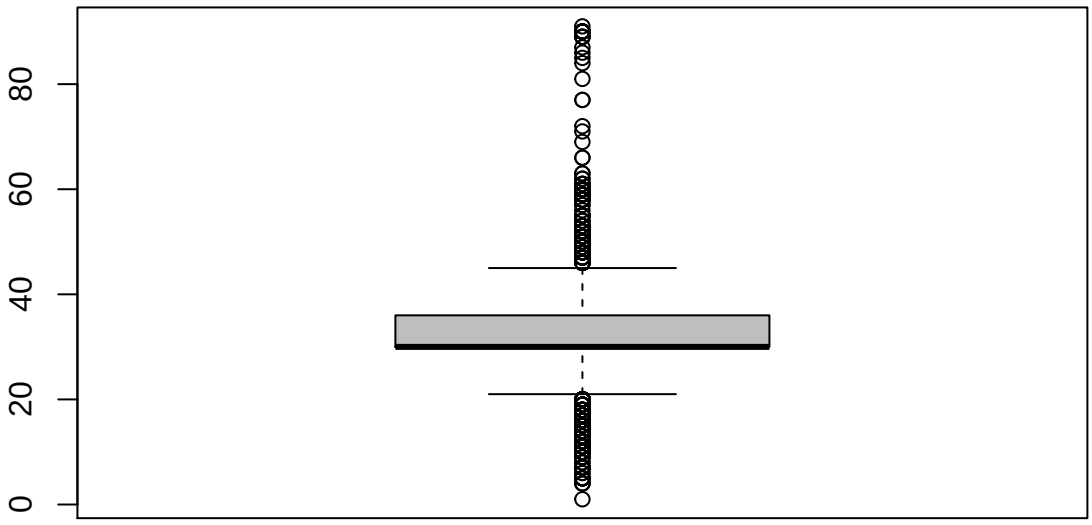
3.2 Valores atípicos

Volviendo a la estadística descriptiva vemos que la diferencia entre la media y el máximo y mínimo valor de muestras de la variable pledged y proyect_length es bastante significativa, lo que puede indicar la presencia de outliers. Vamos a comprobar si tenemos outliers mediante diagrama de cajas.

```
# Importamos la librería ggplot2
library(ggplot2)

# Diagrama de cajas para la variable pledged y proyect_length
boxplot(dataSet$pledged, main="Box plot", col="gray")
```




```
boxplot(dataSet$proyect_length, main="Box plot", col="gray")
```



Vemos que ambas variables tienen valores extremos por lo que vamos a analizar para determinar cómo proceder con ellos.

```
tail(sort(dataSet$pledged), 10)
```

```
##  [1]  268964.3  378935.0  429598.0  449969.0  450333.8  509977.2  575377.1
##  [8]  674425.2 1255444.0 2152285.0
```

```
tail(sort(dataSet$proyect_length), 10)
```

```
##  [1] 90 90 90 90 90 90 90 90 90 91
```

Vemos que hay bastante diferencia entre la media de la variable pledged y los valores más altos, pero haciendo un poco de investigación online se ha encontrado que algún proyecto ha llegado a recaudar más de 20,000,000$, [**enlace**](https://www.marketwatch.com/story/10-kickstarter-products-that-raised-the-most-money-2017-06-22-10883052). Por este motivo se ha decidido aceptar dichos outliers y tratarlos como datos válidos.

Por otro lado, para la variable proyect_length vemos que el valor máximo de esta son 16,739 días, lo que corresponde a más de 3 años, implicaría que un proyecto ha estado recaudando fondos durante todo ese tiempo. Haciendo un poco de investigación sobre la normativa de Kickstarter, se ha encontrado [**enlace**](https://help.kickstarter.com/hc/en-us/articles/115005128434-What-is-the-maximum-project-duration-#:~:text=Projects%20on%20Kickstarter%20can%20last,at%2030%20days%20or%20less.) que, hoy en día, la duración máxima por proyecto es de 60 días. También en este otro artículo de explica que hasta el año 2011 la duración máxima era de 90 días [**enlace**](https://www.kickstarter.com/blog/shortening-the-maximum-project-length).

```
outliersDays <- tail(sort(dataSet$proyect_length), 7)
outliersDays
```

```
## [1] 90 90 90 90 90 90 91
```

```
indexes <- which(dataSet$proyect_length %in% outliersDays)
indexes
```

```
##  [1]  234  376  762 1156 1388 1588 1619 1866 2058 2426 2527 2604 2679 2700
```

```
dataSet$launched[indexes]
```

```
##  [1] "2011-01-11" "2010-10-14" "2010-04-21" "2010-07-18" "2010-09-30"
```

```
##  [6] "2011-05-25" "2010-03-09" "2010-07-01" "2011-04-08" "2011-04-22"
## [11] "2010-04-27" "2011-04-13" "2010-04-21" "2011-05-05"
```

Al comprobar el valor launchdate para cada una de las muestras en las que se encontraban los outliers vemos que la fecha es 1970-01-01 01:00:00, se han recogido mal al guardar los datos, de ahí que salgan valores tan extremos para proyect_length.

Por todo ello se ha decidido que cualquier duración significativamente mayor de 90 días se va a tratar como outlier y se reemplazará por NA para posteriormente imputarlo con un valor de 90.

```
# Se reemplazan los valores en las posiciones indexes por NA
dataSet$proyect_length[indexes] <- NA
```

Muestra de todos las variables y sus valores NA's.

```
# Comprobamos si quedan NA's
sapply(dataSet, function(x) sum(is.na(x)))
```

```
##               ID             name         category    main_category
##                0                0                0                0
##         currency         deadline             goal         launched
##                0                0                0                0
##          pledged            state          backers          country
##                0                0                0               32
##      usd.pledged usd_pledged_real    usd_goal_real      usd_pledged
##               32                0                0                0
##           status    euros_pledged       euros_goal   proyect_length
##                0                0                0               14
```

3.3 Imputación de valores

- Cómo se ha mencionado anteriormente los valores NA de la variable proyect_length se van a reemplazar por 90 ya que es el máximo número de días que un proyecto puede estar recaudando dinero.

```
dataSet$proyect_length[indexes] <- 90
```

- Imputación de valores perdidos para la variable country.

```
idx <- which(is.na(dataSet$country))
# encontrar las combinaciones únicas de country y currency pero no cuando country
# es NA
uniques <- unique(dataSet[c('country', 'currency')])
uniques <- uniques[!is.na(uniques$country),]

# reemplazar los NA's de country con los valores únicos asociados con currency
na.country <- which(is.na(dataSet$country))
na.currency <- dataSet$currency[na.country]
dataSet$country[idx] <- uniques$country[match(na.currency, uniques$currency)]
```

Por último comprobamos si quedan NA's en los datos.

```
# Comprobamos si quedan NAs
sapply(dataSet, function(x) sum(is.na(x)))
```

```
##               ID             name         category    main_category
##                0                0                0                0
##         currency         deadline             goal         launched
##                0                0                0                0
##          pledged            state          backers          country
```

```
##                0                0                0                0
##      usd.pledged usd_pledged_real    usd_goal_real      usd_pledged
##               32                0                0                0
##           status    euros_pledged       euros_goal   proyect_length
##                0                0                0                0
```

3.4 Selección de datos

A continuación, vamos a detallar que atributos hemos descartado y cuales hemos decidido sean imprescindibles para el análisis:

- Se ha decidido borrar del dataset la columna de usd_pledged, esta representa la conversión a dólares por parte de Kickstarter del atributo pledged, pero se han descubierto bastantes inconsistencias. El creador del dataset, por este mismo motivo, decidió incluir un nuevo atributo con una conversión más precisa de pledged, que es la que vamos a usar.
- También se ha decidido descartar la variable usd_goal_pledged porque no resulta significativa para el estudio.
- Por otro lado, vamos a prescindir de la variable state. Como se ha mencionado anteriormente, un Kickstarter es satisfactorio si el proyecto consigue recaudar el dinero marcado como objetivo en el tiempo estimado, por lo que no es necesario para nuestro estudio si dicho proyecto se ha cancelado, o se ha suspendido o sigue activo. Se puede dar el caso por ejemplo que un proyecto llegue al objetivo económico marcado dentro de tiempo, pero el organizador, por cualquier motivo decida suspenderlo. En ese caso el proyecto aparecerá como cancelado, pero desde el punto de vista del objetivo del proyecto, la recaudación ha sido satisfactoria.

```r
# Quitar columnas usd_pledged y usd_goal_pledged del daataset
drops <- c('usd.pledged', 'usd_goal_pledged', 'state')
dataSet <- dataSet[ , !(names(dataSet) %in% drops)]
# Anális descriptivo del dataset limpio
summary(dataSet)
```

```
##        ID                 name                    category
##  Min.   :1.829e+05   Length:3000        Product Design: 174
##  1st Qu.:5.171e+08   Class :character   Music         : 143
##  Median :1.049e+09   Mode  :character   Documentary   : 133
##  Mean   :1.051e+09                      Tabletop Games: 104
##  3rd Qu.:1.564e+09                      Food          : 103
##  Max.   :2.147e+09                      Shorts        :  90
##                                         (Other)       :2253
##       main_category    currency       deadline                goal
##  Film & Video:507   USD    :2369   Min.   :2009-07-01   Min.   :       1
##  Music       :443   GBP    : 258   1st Qu.:2013-06-13   1st Qu.:    2000
##  Publishing  :319   EUR    : 141   Median :2014-12-31   Median :    5000
##  Technology  :273   CAD    : 107   Mean   :2014-10-22   Mean   :   60740
##  Games       :254   AUD    :  55   3rd Qu.:2016-04-18   3rd Qu.:   17000
##  Design      :234   MXN    :  16   Max.   :2018-02-27   Max.   :55000000
##  (Other)     :970   (Other):  54
##     launched            pledged           backers           country
##  Min.   :2009-04-30   Min.   :      0   Min.   :    0.0   US     :2369
##  1st Qu.:2013-05-09   1st Qu.:     40   1st Qu.:    2.0   GB     : 258
##  Median :2014-11-24   Median :    704   Median :   12.0   CA     : 107
##  Mean   :2014-09-18   Mean   :   8549   Mean   :  101.9   AU     :  55
##  3rd Qu.:2016-03-15   3rd Qu.:   4332   3rd Qu.:   57.0   DE     :  39
##  Max.   :2018-01-01   Max.   :2152285   Max.   :14971.0   IT     :  23
```

```
##                                                          (Other): 149
##  usd_pledged_real    usd_goal_real       usd_pledged        status
##  Min.   :      0.0   Min.   :       1   Min.   :      0.0   0:1918
##  1st Qu.:     40.0   1st Qu.:    2000   1st Qu.:     40.0   1:1082
##  Median :    713.5   Median :    5000   Median :    713.5
##  Mean   :   8301.5   Mean   :   58190   Mean   :   8301.5
##  3rd Qu.:   4273.2   3rd Qu.:   15000   3rd Qu.:   4273.2
##  Max.   :2152285.0   Max.   :55000000   Max.   :2152285.0
##
##  euros_pledged         euros_goal       proyect_length
##  Min.   :      0.0   Min.   :       1   Min.   : 1.00
##  1st Qu.:     32.5   1st Qu.:    1626   1st Qu.:30.00
##  Median :    580.1   Median :    4065   Median :30.00
##  Mean   :   6749.1   Mean   :   49382   Mean   :34.11
##  3rd Qu.:   3474.1   3rd Qu.:   13821   3rd Qu.:36.00
##  Max.   :1749825.0   Max.   :44715450   Max.   :90.00
##
```

3.5 Exportación de los datos limpios

Una vez el procesado de los datos ha finalizado, se genera un archivo csv con nombre "ks-projects-201801_clean.csv", que contendrá el dataset con 3000 muestras limpias.

```
# Exportación de los datos limpios en .csv
# write.csv(dataSet, '../data/ks-projects-201801_sample_clean.csv')

dataSet <- read.csv('../data/ks-projects-201801_sample_clean.csv')
attach(dataSet)
```

4 Análisis de los datos

4.1 Selección de los grupos de datos a analizar

De todo el conjunto de datos, se han seleccionado los siguientes atributos para poder ser analizados creyendo que son estos los que aportarán más valor al análisis posterior:

- main_category: recoge las 15 principales categorías presentes.

```
unique(main_category)
```

```
##  [1] "Technology"   "Music"        "Food"         "Art"          "Film & Video"
##  [6] "Games"        "Publishing"   "Design"       "Fashion"      "Crafts"
## [11] "Comics"       "Theater"      "Photography"  "Journalism"   "Dance"
```

- proyect_length: tiempo de duración de cada proyecto, expresado en días.
- euros_pledged: conversión a € de la variable usd_pledged_real.
- country: país en el que se publicó el proyecto Kickstarter. La variable currency representa la moneda de dicho país por lo que nos resulta redundante.
- backers: cantidad total de mecenas del proyecto.
- status: estado del proyecto, 0 no ha conseguido el objetivo, 1 si lo ha conseguido.

4.2 Normalidad y homocedasticidad

A la hora de identificar los métodos de análisis más adecuados se debe conocer antes las características de los datos, por ejemplo, si estos siguen una distribución normal o si presentan homocedasticidad. Por ello vamos a comprobar que las variables numéricas elegidas siguen una distribución normal o presentan homogeneidad de la varianza.

- Test de normalidad

Se va a utilizar el test Shapiro-Wilk, asumiendo un intervalo de confianza del 95%. Esto quiere decir que si el p-valor es menor o igual que el nivel de significancia con un valor de 0.05, entonces podemos rechazar la presunción de normalidad, es decir, la variable no sigue una distribución normal.

```
shapiro.test(proyect_length)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  proyect_length
## W = 0.83726, p-value < 2.2e-16
```

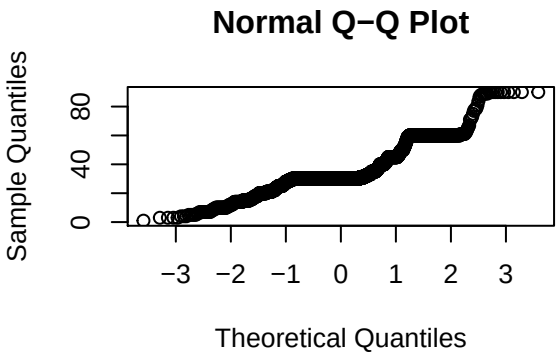
```
shapiro.test(euros_pledged)
```

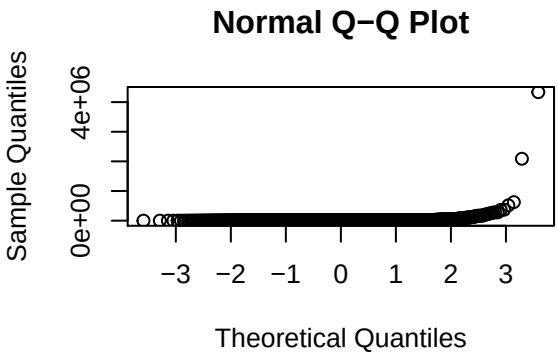
```
##
##  Shapiro-Wilk normality test
##
## data:  euros_pledged
## W = 0.042348, p-value < 2.2e-16
```

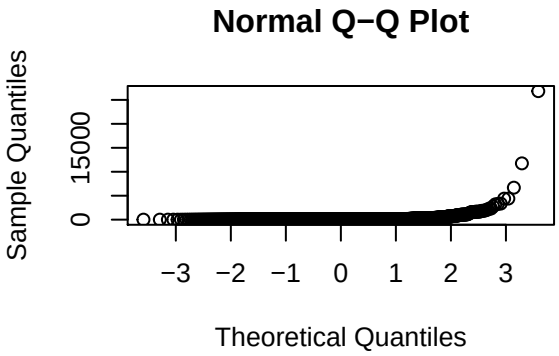
```
shapiro.test(backers)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  backers
## W = 0.096508, p-value < 2.2e-16
```

```
# Representación de la distribución
par(mfrow=c(2,2))
qqnorm(proyect_length)
qqnorm(euros_pledged)
qqnorm(backers)
```







Se puede apreciar que los datos no siguen una distribución normal ya que en el total de las comprobaciones el p-valor del Test de Shapiro-Wilk el p-value $<$ 2.2e-16 rechazando dicha distribución normal.

- Test de homocedasticidad

Ya que hemos comprobado que nuestros datos no siguen una distribución normal (p-value $<$ 2.2e-16 en todos los casos), para el test de homocedasticidad tendremos utilizaremos el de Fligner-Killeen. La hipótesis nula asume la igualdad de varianzas, por lo que p-values inferiores al nivel de significancia (0.05), indicarán heterocedasticidad.

Para ello comprobaremos distintos grupos de datos entre sí:

```
fligner.test(proyect_length ~ euros_pledged, data = dataSet)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  proyect_length by euros_pledged
## Fligner-Killeen:med chi-squared = 1463.2, df = 1954, p-value = 1
```

```
fligner.test(euros_pledged ~ backers, data = dataSet)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  euros_pledged by backers
## Fligner-Killeen:med chi-squared = 1973.4, df = 380, p-value < 2.2e-16
```

```
fligner.test(proyect_length ~ backers, data = dataSet)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
```

```
##
## data:  proyect_length by backers
## Fligner-Killeen:med chi-squared = 320.81, df = 380, p-value = 0.9876
```

De este análisis podemos observar dos casos, para las variables proyect_length-euros_pledged y proyect_length-backers el test de Fligner-Killen da un p-value mayor que 0.05 (1 y 0.942 respectivamente), por lo que se asume homocedasticidad.

Por otro lado, la prueba para euros_pledged y backers se resuelve con un p-value < 2.2e-16, por lo que en este caso si se puede rechazar la hipótesis nula de homocedasticidad y se concluye que la variable euros_pledged presenta varianzas estadísticamente diferentes para los diferentes grupos de backers.

4.3 Pruebas estadísticas

4.3.1 Contraste de hipótesis de dos muestras

Para comprobar si existe relación entre las variables status y main_category, es decir, proyecto fundado exitosamente y tipo categoría vamos a aplicar el test no paramétrico khi cuadrado mediante la función chist.test().

Las hipótesis nula y alternativa quedarían de la siguiente manera:

- Hipótesis nula, H0: el éxito del proyecto y la categoría en la que se encuentre son variables independientes.
- Hipótesis alternativa, H1: existe relación entre la categoría en la que se encuentra un proyecto y el éxito de este.

```
chisqTable <- table( status, main_category )
chisq.test(chisqTable, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  chisqTable
## X-squared = 135.19, df = 14, p-value < 2.2e-16
```

Vemos que el p-value resultante del test es < 2.2e-16 por lo que podemos rechazar la hipótesis nula y afirmar, con un 95% de confianza que existe relación significativa entre la categoría en la que se encuentra un proyecto y su éxito o no.

4.3.2 Modelo regresión logística simple

En este primero modelo de regresión logística simple se quiere analizar la probabilidad de que un proyecto consiga recaudar los fondos propuestos en base a la longitud establecida.

Como se ha comentado anteriormente, Kickstarter decidió disminuir la duración máxima de los proyectos a 60 días y de hecho recomiendan configurar la campaña para que dure 30 días o menos, ya que la probabilidad de éxito disminuye conforme se alarga el proyecto. Es por ello por lo que queremos comprobar si esto es cierto mediante este modelo.

```
# Ajuste de un modelo logístico
modelLogisticSimple <- glm(status ~ proyect_length, family = "binomial")
summary(modelLogisticSimple)
```

```
##
## Call:
## glm(formula = status ~ proyect_length, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -1.2365  -0.9691  -0.8443   1.4011   2.0008
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.160265   0.113582   1.411    0.158
## proyect_length -0.022407   0.003239  -6.918  4.6e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3906.5  on 2999  degrees of freedom
## Residual deviance: 3855.1  on 2998  degrees of freedom
## AIC: 3859.1
##
## Number of Fisher Scoring iterations: 4
```

```
confint(object = modelLogisticSimple, level = 0.95 )
```

```
##                       2.5 %     97.5 %
## (Intercept)    -0.06126099  0.3841249
## proyect_length -0.02882939 -0.0161265
```

4.3.2.1 Predicciones Vamos a comparar probabilidad de que un proyecto sea exitoso cuando la duración de este es de 25 días y cuando es de 45 días y 60 días.

```
# Predicción para 25 días
round(predict(modelLogisticSimple, data.frame(proyect_length = 25), type="response"), 2)
```

```
##   1
## 0.4
```

```
# Predicción para 45 días
round(predict(modelLogisticSimple, data.frame(proyect_length = 45), type="response"), 2)
```

```
##   1
## 0.3
```

```
# Predicción para 60 días
round(predict(modelLogisticSimple, data.frame(proyect_length = 60), type="response"), 2)
```

```
##    1
## 0.23
```

Podemos dar por cierto la afirmación de que a maa2ás largo el proyecto menos probabilidad de éxito hay ya que hemos obtenido que, para los proyectos de 25 días hay un 39% de probabilidad, para los de 45 días un 30% y para los de 60 días un 23%, se aprecia la tendencia a la baja.

4.3.3 Modelo de regresión logística múltiple

Vamos a crear un primer modelo predictivo de regresión logística para predecir la expectativa de que un proyecto sea exitoso antes de lanzarlo. Para ello tendremos como variable respuesta status, y como variables explicativas usaremos: main_category, proyect_length, backers y euros_goal.

Vamos a especificar el nivel base de referencia para la variable cualitativa:

- Para la variable main_category, la categoría 'Dance'.

```r
status <- as.factor(status)
main_category <- as.factor(main_category)
class(euros_goal)
```

```
## [1] "numeric"
```

```r
# Nivel de referencia
main_category <- relevel(main_category, ref = 'Dance')

modelLogistic = glm(formula = status ~ main_category + proyect_length +
                    backers + euros_goal, family = binomial(link = logit))
summary(modelLogistic)
```

```
##
## Call:
## glm(formula = status ~ main_category + proyect_length + backers +
##     euros_goal, family = binomial(link = logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.0874  -0.5554  -0.0774   0.2408   6.5723
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                1.144e+00  5.915e-01   1.933 0.053199 .
## main_categoryArt          -1.564e+00  5.932e-01  -2.637 0.008373 **
## main_categoryComics       -1.866e+00  6.550e-01  -2.849 0.004389 **
## main_categoryCrafts       -2.513e+00  6.996e-01  -3.593 0.000327 ***
## main_categoryDesign       -1.831e+00  6.099e-01  -3.002 0.002686 **
## main_categoryFashion      -1.988e+00  6.098e-01  -3.260 0.001113 **
## main_categoryFilm & Video -1.308e+00  5.814e-01  -2.250 0.024446 *
## main_categoryFood         -1.917e+00  6.290e-01  -3.048 0.002307 **
## main_categoryGames        -3.246e+00  6.398e-01  -5.074 3.89e-07 ***
## main_categoryJournalism   -2.208e+00  7.710e-01  -2.864 0.004185 **
## main_categoryMusic        -1.141e+00  5.825e-01  -1.958 0.050185 .
## main_categoryPhotography  -1.563e+00  6.437e-01  -2.428 0.015162 *
## main_categoryPublishing   -1.946e+00  5.933e-01  -3.280 0.001039 **
## main_categoryTechnology   -2.341e+00  6.470e-01  -3.618 0.000297 ***
## main_categoryTheater      -9.959e-01  6.265e-01  -1.590 0.111941
## proyect_length            -1.623e-02  4.676e-03  -3.472 0.000517 ***
## backers                    4.698e-02  2.155e-03  21.806  < 2e-16 ***
## euros_goal                -2.282e-04  1.389e-05 -16.432  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3906.5  on 2999  degrees of freedom
## Residual deviance: 1947.0  on 2982  degrees of freedom
## AIC: 1983
##
## Number of Fisher Scoring iterations: 11
```

```r
# Odds Ratio
exp(coefficients(modelLogistic))
```

```
##               (Intercept)          main_categoryArt       main_categoryComics
##                3.13799049                0.20925829                0.15475588
##       main_categoryCrafts       main_categoryDesign      main_categoryFashion
##                0.08099729                0.16030910                0.13696670
## main_categoryFilm & Video         main_categoryFood        main_categoryGames
##                0.27031614                0.14706335                0.03891524
##   main_categoryJournalism        main_categoryMusic  main_categoryPhotography
##                0.10992788                0.31956235                0.20944202
##   main_categoryPublishing   main_categoryTechnology      main_categoryTheater
##                0.14286434                0.09622286                0.36939710
##            proyect_length                   backers                euros_goal
##                0.98389732                1.04810405                0.99977181
```

```
exp(confint.default(modelLogistic, level = 0.95))
```

```
##                                2.5 %     97.5 %
## (Intercept)                0.98435499 10.0034890
## main_categoryArt           0.06542029  0.6693494
## main_categoryComics        0.04286705  0.5586898
## main_categoryCrafts        0.02055741  0.3191337
## main_categoryDesign        0.04850739  0.5297957
## main_categoryFashion       0.04145493  0.4525366
## main_categoryFilm & Video 0.08649463  0.8448017
## main_categoryFood          0.04286511  0.5045510
## main_categoryGames         0.01110586  0.1363601
## main_categoryJournalism    0.02425832  0.4981441
## main_categoryMusic         0.10202628  1.0009195
## main_categoryPhotography   0.05930819  0.7396273
## main_categoryPublishing    0.04466118  0.4570013
## main_categoryTechnology    0.02707183  0.3420101
## main_categoryTheater       0.10819136  1.2612303
## proyect_length             0.97492153  0.9929558
## backers                    1.04368732  1.0525395
## euros_goal                 0.99974460  0.9997990
```

Observando el sumario del modelo podemos decir que las variables backers, proyect_length y euros_goal son significativas al tener p-values menores que el nivel de significancia 0.05 (< 2e-16 y 2.47e-08 respectivamente). La variable proyect_length, como ya sabíamos afecta de forma negativa al éxito del proyecto. Por otra parte, la variable backers afecta a al éxito del proyecto de manera positiva 0.0469. La variable euros_goal (objetivo económico marcado al que se tiene que llegar antes de que se termine la campaña), afecta de forma negativa a razón de 0.0002.

En cuanto a la variable main_category, hemos definido antes que el nivel de referencia es 'Dance', si nos fijamos en los OR (odds-ratio) vemos que para el resto las categorías el OR está por debajo de la unidad, lo que indica que es menos probable que un proyecto sea éxitos si pertenece a cualquier otra categoría que si pertenece a Dance.

Cuanto más se aleja el valor del odds-ratio de la unidad, más fuerte es la relación entre la variable dependiente e independiente, por ello vemos que los más alejados de la unidad corresponden a categoría Games (0.038). Si calculamos la inversa tenemos que para Technology $1/0.038 = 26.31$, esto se podría interpretar como, si el proyecto pertenece a la categoría Dance, los odds de éxito son 26.31 veces mayor que si pertenece a la categoría Games.

Basándonos en el intervalo de confianza de las tres variables independientes podemos decir que la que más impacto tiene sobre el éxito del proyecto es main_category, es decir, la categoría en la que encuentra el proyecto. Y dentro de main_category, podemos decir que la categoría que influye de manera más positiva al éxito de un proyecto es Dance.

A modo de nota, se ha probado a introducir la variable country, pero empeoraba el modelo considerablemente, el AIC pasaba a superar los 3700. Además de que todos los p-values para las distintas variables dummy creadas para representar los distintos países estaban por encima de 0.05.

4.3.3.1 Comparación de las predicciones con las observaciones

Para conocer un poco más del modelo crearemos la matriz de confusión y veremos qué porcentaje de observaciones de entrenamiento es capaz de clasificar correctamente el modelo.

```
predicciones <- ifelse(test = modelLogistic$fitted.values > 0.5, yes = 1, no = 0)
# predicciones
matriz_confusion <- table(modelLogistic$model$status, predicciones,
                          dnn = c("observaciones", "predicciones"))
matriz_confusion
```

```
##              predicciones
## observaciones    0    1
##             0 1854   78
##             1  232  836
```

El modelo es capaz de clasificar correctamente ((1854 + 836) / (1854 + 836 + 232 + 78)) = 0.8966, es decir, el 89.66% de las observaciones de entrenamiento, por lo que se puede considerar bueno.

4.3.3.2 Predicción

A continuación, vamos a aplicar nuestro modelo para predecir que probabilidad tiene de ser exitoso un proyecto que pertenece a la categoría Music, cuya duración va a ser de 30 y cuyo objetivo monetario serán 10,000€. Ya que el número de mecenas no se sabe hasta que el proyecto no ha finalizado, vamos a calcularlo para 5 valores distintos (20, 40, 60, 80, 100).

```
predict(modelLogistic, data.frame(main_category = 'Dance', proyect_length = 30,
                                  backers = seq(from = 20, to = 100, by = 20),
                                  euros_goal = 10000), type ="response")
```

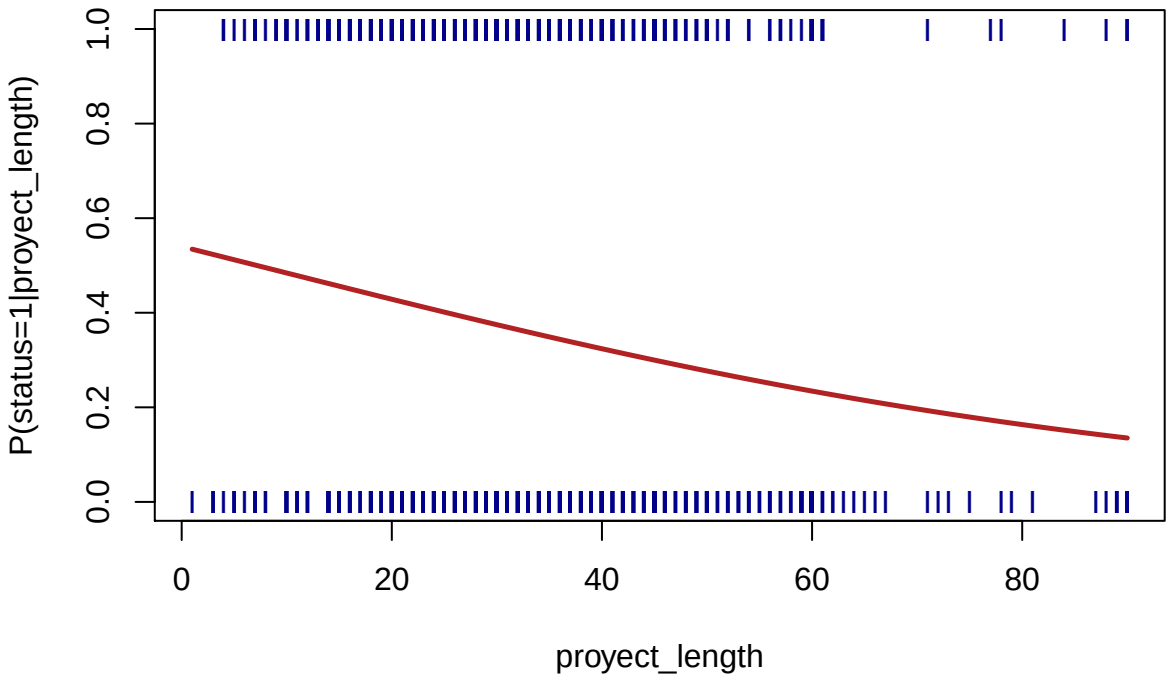
```
##         1         2         3         4         5
## 0.3349416 0.5630965 0.7673479 0.8940745 0.9557530
```

La probabilidad, expresada en porcentaje para los valores mencionados anteriormente iría desde 33.49% si hay 20 backers a un 95.57% si los backers son 100.

5 Representación de los resultados

- Visualización predictiva de la variable del éxito de un proyecto en base a la variable proyect_length

```
dataSet$status <- as.character(dataSet$status)
dataSet$status <- as.numeric(dataSet$status)

plot(status ~ proyect_length, dataSet, col = "darkblue",
     main = "Modelo regresión logística",
     ylab = "P(status=1|proyect_length)",
     xlab = "proyect_length", pch = "I")

curve(predict(modelLogisticSimple, data.frame(proyect_length = x), type = "response"),
      col = "firebrick", lwd = 2.5, add = TRUE)
```



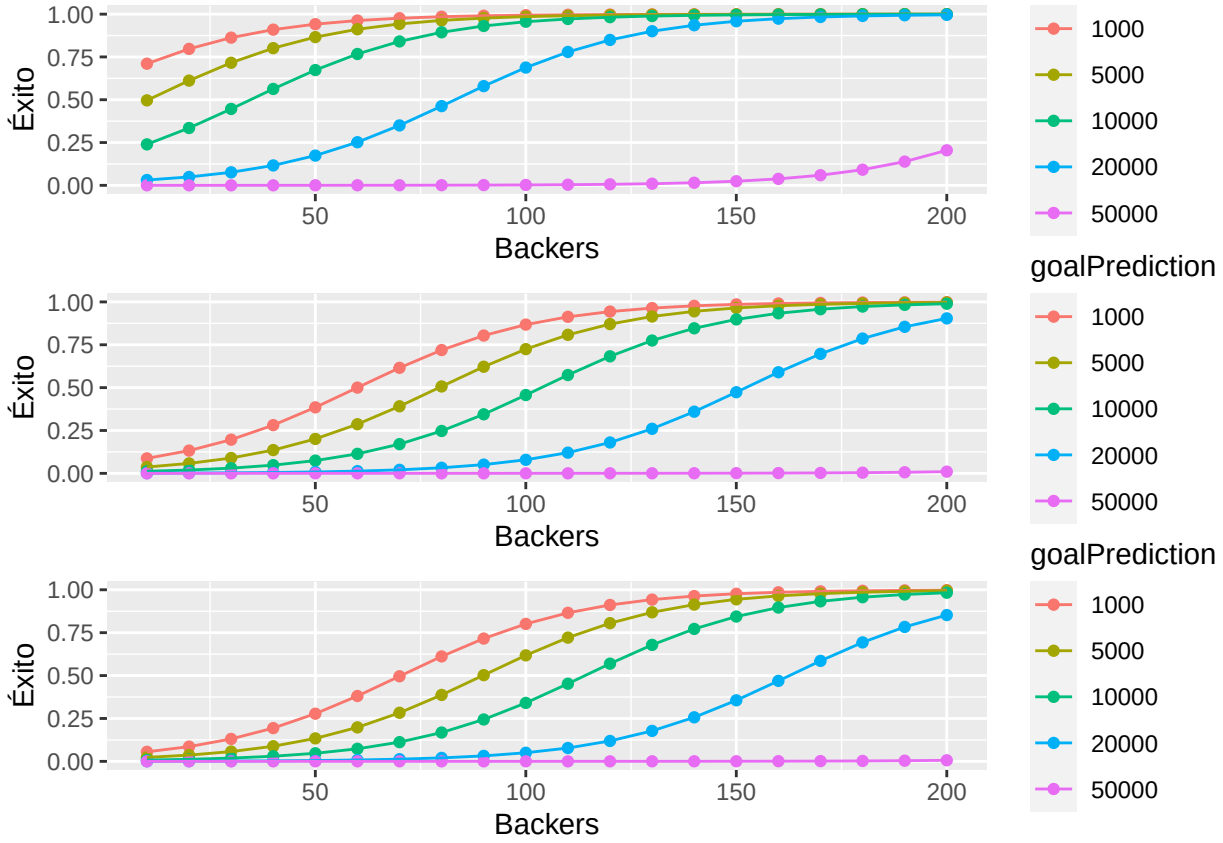
- Visualización predictiva de la variable del éxito de un proyecto en base a las variables main_category, proyect_length, backers y euros_goal. La variable main_category será Dance, la longitud del proyecto será 30 días. Luego la variable backers tomará valores de 10 a 150 de 10 en 10 y por último para la variable euros goal se han propuesto los valores 1000, 5000, 10000, 20000 y 50000.

```
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```
newdata <- data.frame(proyect_length = c(rep(30, 100)),
                      backers = rep(seq(from = 10, to = 200, by = 10), 5),
                      main_category=c(rep('Dance', 100)),
                      euros_goal=c(rep(1000, 20), rep(5000, 20), rep(10000, 20),
                                   rep(20000, 20), rep(50000, 20)))

successPrediction <- predict(modelLogistic, newdata, type="response")
goalPrediction <- as.factor(c(rep(1000, 20), rep(5000, 20), rep(10000, 20),
                    rep(20000, 20), rep(50000, 20)))
backersTotal <- rep(seq(from = 10, to = 200, by = 10), 5)
predictionDataFrame <- data.frame(goalPrediction, successPrediction, backersTotal)

# Gráfica para la categoría Dance
plot1 <- ggplot(predictionDataFrame, aes(x = backersTotal, y = successPrediction,
                                col = goalPrediction))+geom_point()+geom_line()+
                                ylab('Éxito') + xlab('Backers')

# Gráfica para la categoría Games
newdata$main_category <- c(rep('Games', 100))
successPrediction <- predict(modelLogistic, newdata, type="response")
predictionDataFrame <- data.frame(goalPrediction, successPrediction, backersTotal)
plot2 <- ggplot(predictionDataFrame, aes(x = backersTotal, y = successPrediction,
```

```
                                col = goalPrediction))+geom_point()+geom_line()+
                                ylab('Éxito') + xlab('Backers')

# Gráfica para longitud de proyecto 60 días
newdata$proyect_length <- c(rep(60, 100))
successPrediction <- predict(modelLogistic, newdata, type="response")
predictionDataFrame <- data.frame(goalPrediction, successPrediction, backersTotal)
plot3 <- ggplot(predictionDataFrame, aes(x = backersTotal, y = successPrediction,
                                col = goalPrediction))+geom_point()+geom_line()+
                                ylab('Éxito') + xlab('Backers')
grid.arrange(plot1, plot2, plot3, ncol = 1)
```



Podemos observar como a medida que aumenta la cantidad de mecenas la probabilidad de que el proyecto consiga recaudar el goal propuesta aumenta. Dependiendo de dicho goal, se necesitarán más backers o menos para que la probabilidad aumenta más deprisa. Por ejemplo, si el goal propuesto son 5000€, con apenas 50 backers la probabilidad llegaría al 87%, sin embargo, si el goal son 10000, con los 50 backers de antes la probabilidad estaría en torno al 70%. Vemos también que cuando el objetivo está muy por encima de la media de la variable euros_goal (26,803) la curva de subida no es tan pronunciada.

Al cambiar la categoría observamos que la probabilidad de éxito disminuye, es decir, se necesitan más mecenas por proyecto para que el proyecto sea exitoso. Y lo mismo ocurre si la duración del proyecto se alarga hasta los 60 días.

6 Conclusiones

Se ha conseguido crear un modelo que predice que probabilidad tiene un proyecto de conseguir el objetivo monetario propuesto en base a la categoría en la que se incluye el proyecto, la longitud de este, el goal fijado

y los mecenas. Cierto es que los mecenas del proyecto es imposible conocerlos de antemano, pero se pueden estimar ciertos valores que nos ayudarán a saber qué cantidad de backers necesitaremos para que nuestro proyecto sea exitoso.

Claramente la categoría a la que pertenece el mismo es un factor que influye en gran medida al igual que el goal propuesto. Se puede ver en las gráficas que a partir de cierto goal la probabilidad disminuye muchísimo.

De entrada, se puede pensar que a más días dure el proyecto mejor, más probabilidad de conseguir mecenas y llegar al objetivo, pero claramente, mediante los modelos y las graa2áficas se puede observar que esto no es así, sino todo lo contrario.

7 Agradecimientos

En primer lugar, agradecer y reconocer el trabajo de Mickaël Mouillé **[enalce]**(https://www.kaggle.com/kemical), creador del dataset por trabajo para recolectar datos durante tantos años y publicarlos para el uso público.

También agradecer a todas aquellas personas que han publicado sus dudas sobre el dataset para beneficio de todos.

8 Tabla de contribuciones

Contribuciones	Firma
Investigación previa	M.G.
Redacción de las respuestas	M.G.
Desarrollo código	M.G.