Universitat Oberta de Catalunya

Patricia Lázaro Tello
plazarotello@uoc.edu

PRA1: Selección y preprocesado de un conjunto de datos

Patricia Lázaro Tello

Índice general

Patricia Lázaro Tello
plazarotello@uoc.edu

1 Introducción

Kickstarter es una plataforma de financiación colectiva para proyectos creativos — música, arte, tecnología, videojuegos… — fundada en 2009. A finales de 2012 el modelo de financiación alternativo que proponía la plataforma ganó mucha popularidad y tracción, y aunque a día de hoy ese fervor ha disminuido, la financiación colectiva y este tipo de plataformas se han normalizado y permanecen relevantes.

Kickstarter es la plataforma de financiación colectiva más popular actualmente, pero compite en su nicho de mercado con otras como Indiegogo o GoFundMe. Esta plataforma se utiliza para financiar proyectos (ideas con un objetivo claro y que eventualmente son completadas).

En este trabajo se propone un proyecto de minería de datos sobre los proyectos que han intentado financiarse a través de **Kickstarter** entre el 12 de agosto y el 16 de septiembre de 2021.

El análisis del éxito o fracaso de proyectos en la fase de financiación colectiva es un tema de estudio interesante debido a:

- La posibilidad de orientar un proyecto a un público objetivo u otro en **Kickstarter** para maximizar las posibilidades de ser financiado con éxito.
- Decidir el modelo de financiación (o la plataforma de finanaciación colectiva) del proyecto de antemano y ahorrar costes de tiempo.

2 Objetivos del proyecto

El proyecto objeto de estudio de este trabajo consiste en el **estudio de las características de los proyectos financiados con éxito en Kickstarter**.

Su objetivo es averiguar si un proyecto va a ser financiado con éxito a partir las características del mismo; por lo tanto, se trata de un **problema de clasificación**. Se estudiará cuál de los siguientes modelos es más adecuado para el caso de estudio:

- **Regresión logística**
- **Árboles de decisión**
- **Algoritmo k-NN**
- **Redes neuronales**

A este respecto, el problema de clasificación es **binario** (el proyecto es o no financia-

do con éxito), y se prevee que **no sea equilibrado** (hay muchos más proyectos no financiados que financiados con éxito).

Debido a que se asume que las clases no están equilibradas, definir una medida de cumplimiento del objetivo resulta más difícil que en casos de clasificación binaria equilibrada. Se procede a definir las medidas que se utilizarán:

$$F_2 = (1 + 2^2) \times \frac{precision \times recall}{(2^2 \times precision) + recall}$$

$$FM = \sqrt{\frac{TP}{TP + FP} \times \frac{TP}{TP + FN}}$$

$$Jaccard = \frac{TP}{TP + FP + FN}$$

Tanto el denominado F2-Score (F_2), como el índice de Fowlkes-Mallows (FM en este documento; representa la media geométrica de precisión y *recall*) y el índice de Jaccard ($Jaccard$) son **medidas de evaluación de sistemas de clasificación binaria no equilibrados**.

Estas medidas de evaluación son necesarias para no aceptar modelos no entrenados, como por ejemplo un modelo que clasificara todos los proyectos como fallidos; en ese caso, la precisión sería muy alta, ya que clasificaría correctamente la mayoría de los proyectos, pero el *recall* sería muy bajo al no clasificar correctamente ninguno de los proyectos de éxito.

Con estas medidas se intenta balancear precisión y *recall*; para el proyecto objeto de estudio de este trabajo **es más importante el *recall*** (que los proyectos de éxito se clasifiquen como tal) que la precisión (se acepta que algún proyecto fallido pueda ser considerado de éxito).

El objetivo del proyecto de minería de datos se alcanzará cuando el modelo de clasificación elegido obtenga una medida de $F_2 > 0.75$, $FM > 0.75$ y $Jaccard > 0.75$.

Los datos para el estudio de los casos de éxito y fracaso de proyectos en Kickstarter se obtienen de un *scraper* de Web Robots ejecutado mensualmente (más detalles en la bibliografía).

El trabajo se va a realizar utilizando solo los datos correspondiente al 2021 (hasta el 16 de septiembre), pero se podría aplicar el proyecto de minería de datos sobre el conjunto de datos global, que se remonta hasta 2014.

Patricia Lázaro Tello
plazarotello@uoc.edu

3 Carga de la base de datos

Se procede a cargar en memoria el conjunto de datos con el que se va a trabajar. Este *dataset* tiene 2 particularidades: cada mes corresponde con una carpeta, y está a su vez contiene varios ficheros CSV, que tienen los datos de los proyectos de **Kickstarter** minados por el bot de Web Robots ese mes.

```
csv.files <- dir(dir.path, full.names=TRUE, recursive=TRUE, pattern='.csv')
csv.files <- sort(csv.files, decreasing=TRUE)
raw <- as.data.frame(data.table::rbindlist(lapply(csv.files, read.csv,
                                                  encoding='UTF-8'),
                                           fill=TRUE))
raw.dim <- dim(raw)
```

Se han cargado en memoria 327 ficheros CSV, que corresponden con 1.186.347 proyectos, cada uno de ellos con 39 atributos.

---

4 Análisis inicial

A continuación se mostrarán los datos más relevantes de los atributos de cada proyecto. Al haber 39 atributos (demasiados para mostrar resúmenes de cada uno de ellos sin filtrar), se va a realizar un cribado de variables según la utilidad que se observe para el problema a resolver, basándose en los nombres de las columnas y un análisis visual inicial.

```
cat(colnames(raw), sep=' | ', fill=60)
```

```
## backers_count | blurb | category |
## converted_pledged_amount | country |
## country_displayable_name | created_at | creator |
## currency | currency_symbol | currency_trailing_code |
## current_currency | deadline | disable_communication |
## friends | fx_rate | goal | id | is_backing | is_starrable |
## is_starred | launched_at | location | name | permissions |
## photo | pledged | profile | slug | source_url | spotlight |
## staff_pick | state | state_changed_at | static_usd_rate |
## urls | usd_exchange_rate | usd_pledged | usd_type
```

- **backers_count**: número de personas que han financiado el proyecto.
- **blurb**: pequeña descripción del proyecto.

- **category**: categoría del proyecto en formato JSON.
- **converted_pledged_amount**: cantidad de dinero que se ha recaudado, convertida a dólares americanos.
- **country**: país en que se realiza el proyecto.
- **country_displayable_name**: nombre largo del país.
- **created_at**: fecha en que se creó el proyecto en Kickstarter.
- **creator**: creador del proyecto en formato JSON.
- **currency**: moneda en que se financia el proyecto.
- **currency_symbol**: símbolo de la moneda en que se financia el proyecto.
- **currency_trailing_code**: formato de la cantidad de dinero (si lleva separadores para miles, etc).
- **current_currency**: moneda en que se ha explorado el proyecto.
- **deadline**: fecha en que se cierra la campaña de financiación en Kickstarter.
- **disable_communication**: si el proyecto tiene deshabilitado un tablón de anuncios de actualizaciones del proyecto.
- **friends**: campo vacío.
- **fx_rate**: ratio de conversión de la moneda local a dólares americanos.
- **goal**: objetivo de financiación en moneda local.
- **id** identificador único del proyecto.
- **is_backing**: si el proyecto se encuentra en medio de la campaña de financiación.
- **is_starrable**: si los usuarios de la página pueden guardar este proyecto (añadir a sus favoritos).
- **is_starred**: si el proyecto ha sido guardado por el *scraper* como favorito.
- **launched_at**: fecha en que se lanzó la campaña de financiación.
- **location**: localización (ciudad, región…) del proyecto, en formato JSON.
- **name**: nombre del proyecto.
- **permissions**: campo vacío.
- **photo**: link a la fotografía de portada del proyecto, en formato JSON.
- **pledged**: cantidad de dinero recaudada por el proyecto, en moneda local.
- **profile**: perfil del proyecto en formato JSON.
- **slug**: final de la url del proyecto en Kickstarter.
- **source_url**: url desde la que el *scraper* se encontró el proyecto.
- **spotlight**: si el proyecto ha utilizado la herramienta de creación de página *Spotlight*. Sólo disponible después de una financiación exitosa.
- **staff_pick**: si el proyecto ha sido elegido por Kickstarter como uno de sus favoritos.
- **state**: el estado del proyecto en Kickstarter.
- **state_changed_at**: fecha en que el estado del proyecto cambió en Kickstarter.

- **static_usd_rate**: ratio de cambio de divisa que se utiliza por defecto.
- **urls**: urls del proyecto, en formato JSON.
- **usd_exchange_rate**: ratio de cambio de divisa que se aplicó.
- **usd_pledged**: cantidad de dinero recaudada finalmente, en dólares americanos.
- **usd_type**: tipo de dólar americano.

De la descripción de las variables se obtiene información interesante. Los datos se encuentran en varias divisas (dólares americanos y la divisa local de cada país); para poder hacer una clasificación adecuada se requerirá convertir los datos a una única divisa. Por conveniencia, se utilizará el dólar americano (USD).

Además, hay varios campos que se encuentran en formato JSON y que podrían contener información interesante. Será necesario analizar estos campos en detalle para obtener la mayor cantidad de información interesante para la clasificación.

---

5 Selección de características

A continuación se eliminan campos que contienen información innecesaria: **usd_type**, **urls**, **permissions**, **fx_rate**, **currency current_currency**, **currency_trailing_code**, **currency_symbol**, **country_displayable_name**, **converted_pledged_amount**, **friends**, **spotlight**, **source_url** y **pledged**.

```
dropped.unused.cols <- raw
dropped.unused.cols$usd_type <- NULL
dropped.unused.cols$urls <- NULL
dropped.unused.cols$permissions <- NULL
dropped.unused.cols$fx_rate <- NULL
dropped.unused.cols$currency <- NULL
dropped.unused.cols$current_currency <- NULL
dropped.unused.cols$currency_trailing_code <- NULL
dropped.unused.cols$currency_symbol <- NULL
dropped.unused.cols$country_displayable_name <- NULL
dropped.unused.cols$converted_pledged_amount <- NULL
dropped.unused.cols$friends <- NULL
dropped.unused.cols$spotlight <- NULL
dropped.unused.cols$source_url <- NULL
dropped.unused.cols$pledged <- NULL
dropped.unused.cols.dim <- dim(dropped.unused.cols)
```

Patricia Lázaro Tello
plazarotello@uoc.edu

Tras el cribado inicial de variables se obtienen 1.186.347 observaciones de 25 atributos. Se procede a explorar la estructura y naturaleza de las variables consideradas inicialmente útiles con la finalidad de eliminar atributos redudantes o sin utilidad.

```
attribute.summary <- function(fn.df, fn.col, n=3){
  print.values <- sapply(fn.df[1:n, fn.col], gsub, pattern='(.{80})',
                         replacement='\\1\n')
  print.summary <- summary(fn.df[fn.col])

  col.name <- ifelse(is.character(fn.col), fn.col, colnames(fn.df)[fn.col])

  cat('[*] Dataframe: ', deparse(substitute(fn.df)),
      ' \t Atributo: ', col.name,
      '\n', sep='', fill=60)
  cat('Resumen de las características:\n')
  cat(print.summary, fill=60)
  cat('\n')
  cat('Ejemplos de uso:\n')
  cat(print.values, fill=60, sep='\t')
}

possible.values <- function(fn.df, fn.col){
  cat('\nValores posibles:')
  table(fn.df[fn.col])
}
```

```
attribute.summary(dropped.unused.cols, 'id')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: id
##
## Resumen de las características:
## Min.   :     18520   1st Qu.: 536339130
## Median :1072735957   Mean   :1072940776
## 3rd Qu.:1609069961   Max.   :2147476221
##
## Ejemplos de uso:
## 838575496    147947813   293869602
```

Este atributo no tiene valor a nivel informativo, pero es importante reservarlo para eliminar posibles registros redundantes. Se procede a realizar dicha tarea:

```
cat(dropped.unused.cols.dim[1], ' observaciones de proyectos\n',
    length(unique(dropped.unused.cols$id)), ' proyectos únicos\n', sep='')
```

```
## 1186347 observaciones de proyectos
## 210319 proyectos únicos
```

```
dropped.unused.cols <- dropped.unused.cols %>%
  dplyr::distinct(id, .keep_all = TRUE)

dropped.unused.cols.dim <- dim(dropped.unused.cols)
```

```
attribute.summary(dropped.unused.cols, 'backers_count')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: backers_count
##
##
## Resumen de las características:
## Min.    :     0   1st Qu.:     4   Median :    26
## Mean    :   154   3rd Qu.:    90   Max.    :105857
##
## Ejemplos de uso:
## 100  918 75
```

El uso de este atributo es autoexplicativo: el número de personas que financian un proyecto posiblemente tenga un impacto alto en si la campaña de financiación de dicho proyecto es exitosa o no.

Se utilizará en el análisis inicial de forma descriptiva, ya que a la hora de determinar si un futuro proyecto se financiará exitosamente no se poseerán todavía el número de personas que va a financiarlo.

```
attribute.summary(dropped.unused.cols, 'blurb')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: blurb
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## Combines polyphonic voice generation and lush reverb, delivering a beautiful atm
## ospheric presence that blends in with your guitar tone.
## Software patching goes hardware! Create your own modular synthesiser by programm
## ing Patchblocks with our software editor.
## the open source musical instrument platform, with interchangeable necks and buil
## t-in effects.
```

Se podría aplicar análisis de sentimientos o análisis de textos a la descripción del proyecto; sin embargo, esto sale fuera del alcance de la asignatura. Una métrica sencilla

de calcular que quizás pueda influenciar en el éxito de la campaña de financiación es el número de palabras que contiene la descripción.

```
attribute.summary(dropped.unused.cols, 'category', n=1)
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: category
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## {"id":339,"name":"Sound","analytics_name":"Sound","slug":"technology/sound","pos
## ition":12,"parent_id":16,"parent_name":"Technology","color":6526716,"urls":{"web
## ":{"discover":"http://www.kickstarter.com/discover/categories/technology/sound"}
## }}
```

Observando la estructura del texto en formato JSON, se intuye que se puede extraer el nombre de la categoría (`parent_name`) y de la subcategoría (`name`) a la que pertenece el proyecto.

```
attribute.summary(dropped.unused.cols, 'country')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: country
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## NZ   GB  US
```

```
attribute.summary(dropped.unused.cols, 'location', n=2)
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: location
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## {"id":28676724,"name":"Christchurch Central","slug":"christchurch-central-christ
## church-nz","short_name":"Christchurch Central, NZ","displayable_name":"Christchu
## rch Central, NZ","localized_name":"Christchurch Central","country":"NZ","state":
## "Canterbury","type":"Suburb","is_root":false,"expanded_country":"New Zealand","u
## rls":{"web":{"discover":"https://www.kickstarter.com/discover/places/christchurc
## h-central-christchurch-nz","location":"https://www.kickstarter.com/locations/chr
## istchurch-central-christchurch-nz"},"api":{"nearby_projects":"https://api.kickst
## arter.com/v1/discover?signature=1631827295.e1aafb06c401f5d15fa3fb7cda1ad48e9b131
```

```
## 075&woe_id=28676724"}}}
## {"id":44544,"name":"Belfast","slug":"belfast-belfast-northern-ireland","short_na
## me":"Belfast, UK","displayable_name":"Belfast, UK","localized_name":"Belfast","c
## ountry":"GB","state":"Northern Ireland","type":"Town","is_root":false,"expanded_
## country":"United Kingdom","urls":{"web":{"discover":"https://www.kickstarter.com
## /discover/places/belfast-belfast-northern-ireland","location":"https://www.kicks
## tarter.com/locations/belfast-belfast-northern-ireland"},"api":{"nearby_projects"
## :"https://api.kickstarter.com/v1/discover?signature=1631820384.736f6b73273de921f
## e82fcfb3a7f0581c0615095&woe_id=44544"}}}
```

Analizando los 2 atributos relacionados con la localización geográfica del proyecto, se observa que el atributo `country` se encuentra redundado en el atributo `location`, que contiene más información adicional (`name` para el nombre de la ciudad, `state` para el estado de EEUU al que pertenece la ciudad, y `expanded_country` para el país al que pertenece). Se procede a eliminar `country` por ser redundante.

```
dropped.unused.cols$country <- NULL
dropped.unused.cols.dim <- dim(dropped.unused.cols)
```

```
attribute.summary(dropped.unused.cols, 'created_at')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: created_at
##
## Resumen de las características:
## Min.   :1240366270   1st Qu.:1423672580
## Median :1483313402   Mean   :1485613319
## 3rd Qu.:1558124072   Max.   :1631711869
##
## Ejemplos de uso:
## 1382412214   1381837548  1353724582
```

```
attribute.summary(dropped.unused.cols, 'deadline')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: deadline
##
## Resumen de las características:
## Min.   :1242467940   1st Qu.:1429992658
## Median :1490025922   Mean   :1492756736
## 3rd Qu.:1565609412   Max.   :1636940051
##
## Ejemplos de uso:
## 1386889680   1386194726  1377799927
```

```
attribute.summary(dropped.unused.cols, 'launched_at')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: launched_at
```

```
##
## Resumen de las características:
## Min.   :1240673781   1st Qu.:1427134760
## Median :1487206922   Mean   :1489886122
## 3rd Qu.:1562625871   Max.   :1631755648
##
## Ejemplos de uso:
## 1384297680   1383602726  1376417527
```

```
attribute.summary(dropped.unused.cols, 'state_changed_at')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo:
## state_changed_at
##
## Resumen de las características:
## Min.   :1242468025   1st Qu.:1429877494
## Median :1489941842   Mean   :1492563297
## 3rd Qu.:1565528190   Max.   :1631755649
##
## Ejemplos de uso:
## 1386889680   1386194729  1377799927
```

Las fechas de creación, lanzamiento y finalización de la campaña, y de cambio de estado del proyecto, pueden resultar útiles para obtener atributos derivados como el tiempo de preparación de la campaña de financiación, o período de financiación más exitoso.

El cambio de estado se utilizará en el análisis inicial de forma descriptiva, ya que a la hora de determinar si un futuro proyecto se financiará exitosamente no se poseerá todavía datos sobre la recaudación.

Las fechas se encuentra en formato UNIX epoch y habrán de ser convertidas en fases posteriores.

```
attribute.summary(dropped.unused.cols, 'creator', n=1)
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: creator
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## {"id":556577677,"name":"Flux Effects // Michael Weavers","is_registered":null,"i
## s_email_verified":null,"chosen_currency":null,"is_superbacker":null,"avatar":{"t
## humb":"https://ksr-ugc.imgix.net/assets/008/082/911/93f2bfaa72b180d60b420a094d44
## 0e44_original.jpg?ixlib=rb-4.0.2&w=40&h=40&fit=crop&v=1461501566&auto=format&fra
```

```
## me=1&q=92&s=0e7e2415700b23b8c54ae64e0d5751b8","small":"https://ksr-ugc.imgix.net
## /assets/008/082/911/93f2bfaa72b180d60b420a094d440e44_original.jpg?ixlib=rb-4.0.2
## &w=80&h=80&fit=crop&v=1461501566&auto=format&frame=1&q=92&s=44a38d6aa58dbd5b5d03
## 658c0a843c5e","medium":"https://ksr-ugc.imgix.net/assets/008/082/911/93f2bfaa72b
## 180d60b420a094d440e44_original.jpg?ixlib=rb-4.0.2&w=160&h=160&fit=crop&v=1461501
## 566&auto=format&frame=1&q=92&s=49fc40f44ec8cbbf8e3cd775ac77356a"},"urls":{"web":
## {"user":"https://www.kickstarter.com/profile/556577677"},"api":{"user":"https://
## api.kickstarter.com/v1/users/556577677?signature=1631848957.fa04627c4dc86fd5ef22
## 0b1a3eba8e9c2a3870e3"}}}
```

De los datos del creador se puede obtener información útil, como el atributo `is_superbacker`, que da información sobre si el usuario creador del proyecto tiene experiencia en el rol de patrocinador de otros proyectos (25 proyectos apoyados y 10 USD contribuidos en el último año).

Además, gracias a los identificadores (`id`) se puede saber también la experiencia que tiene el creador en la realización de campañas de financiación colectiva.

```
attribute.summary(dropped.unused.cols, 'disable_communication')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo:
## disable_communication
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## false    false   false
```

```
possible.values(dropped.unused.cols, 'disable_communication')
```

```
##
## Valores posibles:

##
##  false
## 210319
```

Se observa que todos los proyectos tienen la característica del tablón de anuncio activada; se concluye que la variable no aporta información relevante y se procede a eliminarla:

```
dropped.unused.cols$disable_communication <- NULL
dropped.unused.cols.dim <- dim(dropped.unused.cols)
```

```
attribute.summary(dropped.unused.cols, 'goal')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: goal
##
## Resumen de las características:
## Min.   :        0   1st Qu.:     1500   Median :     5000
## Mean   :    55939   3rd Qu.:    15000   Max.   :100000000
##
## Ejemplos de uso:
## 7000 10000   18000
```

A priori se puede asumir que la cantidad de dinero necesaria para financiar el proyecto (`goal`) es una característica importante para predecir el éxito o fracaso de la campaña de financiación. Esta variable, al estar en la divisa local, habrá que covnertirla a dólares americanos.

```
attribute.summary(dropped.unused.cols, 'is_backing')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: is_backing
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## NA   NA  NA
```

```
possible.values(dropped.unused.cols, 'is_backing')
```

```
##
## Valores posibles:

##
##         false   true
## 122554    141      1
```

Esta variable no contiene información útil; por lo tanto se procede a eliminarla:

```
dropped.unused.cols$is_backing <- NULL
dropped.unused.cols.dim <- dim(dropped.unused.cols)
```

```
attribute.summary(dropped.unused.cols, 'name')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: name
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
```

```
##
## Ejemplos de uso:
## Liquid Ambience // A boutique atmospheric guitar effect
## Patchblocks - programmable mini synth modules
## kitar, the evolving instrument
```

El nombre del proyecto, como sucede con su descripción, puede ser fuente de información para un análisis de texto o sentimiento; en este caso, se utilizará la misma métrica del número de palabras para comprobar si un nombre largo o corto tiene influencia en el éxito de la campaña.

```
attribute.summary(dropped.unused.cols, 'photo', n=1)
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: photo
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## {"key":"assets/011/595/301/16a59433ff3828eb5f05de1ccef2c818_original.jpg","full"
## :"https://ksr-ugc.imgix.net/assets/011/595/301/16a59433ff3828eb5f05de1ccef2c818_
## original.jpg?ixlib=rb-4.0.2&crop=faces&w=560&h=315&fit=crop&v=1463684983&auto=fo
## rmat&frame=1&q=92&s=77607016d544d789a5192a824eb85a83","ed":"https://ksr-ugc.imgi
## x.net/assets/011/595/301/16a59433ff3828eb5f05de1ccef2c818_original.jpg?ixlib=rb-
## 4.0.2&crop=faces&w=352&h=198&fit=crop&v=1463684983&auto=format&frame=1&q=92&s=b2
## d25a9df2ded0548c2a3cf1c5d474a7","med":"https://ksr-ugc.imgix.net/assets/011/595/
## 301/16a59433ff3828eb5f05de1ccef2c818_original.jpg?ixlib=rb-4.0.2&crop=faces&w=27
## 2&h=153&fit=crop&v=1463684983&auto=format&frame=1&q=92&s=a1bb5fd4a6fabcba95e7798
## 4df9daf5c","little":"https://ksr-ugc.imgix.net/assets/011/595/301/16a59433ff3828
## eb5f05de1ccef2c818_original.jpg?ixlib=rb-4.0.2&crop=faces&w=208&h=117&fit=crop&v
## =1463684983&auto=format&frame=1&q=92&s=1fb77c59041a018c30a3626a10c74812","small"
## :"https://ksr-ugc.imgix.net/assets/011/595/301/16a59433ff3828eb5f05de1ccef2c818_
## original.jpg?ixlib=rb-4.0.2&crop=faces&w=160&h=90&fit=crop&v=1463684983&auto=for
## mat&frame=1&q=92&s=1dae990899595dbf95ffe6ac16907ad0","thumb":"https://ksr-ugc.im
## gix.net/assets/011/595/301/16a59433ff3828eb5f05de1ccef2c818_original.jpg?ixlib=r
## b-4.0.2&crop=faces&w=48&h=27&fit=crop&v=1463684983&auto=format&frame=1&q=92&s=41
## d687e41dafe4109c54a417a59fab61","1024x576":"https://ksr-ugc.imgix.net/assets/011
## /595/301/16a59433ff3828eb5f05de1ccef2c818_original.jpg?ixlib=rb-4.0.2&crop=faces
## &w=1024&h=576&fit=crop&v=1463684983&auto=format&frame=1&q=92&s=ae08ae30f1b413b00
## a1cc627dd9fe854","1536x864":"https://ksr-ugc.imgix.net/assets/011/595/301/16a594
## 33ff3828eb5f05de1ccef2c818_original.jpg?ixlib=rb-4.0.2&crop=faces&w=1552&h=873&f
## it=crop&v=1463684983&auto=format&frame=1&q=92&s=37b64133ce1c93d74d38c29408d7aaf3
## "}
```

Al no llevar a cabo un análisis de imágenes — de nuevo, este análisis se encuentra fuera del alcance de la asignatura —, se puede analizar únicamente si el uso de una

imagen para el proyecto es un factor de éxito o no.

```
attribute.summary(dropped.unused.cols, 'profile', n=1)
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: profile
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## {"id":745762,"project_id":745762,"state":"inactive","state_changed_at":142591584
## 1,"name":null,"blurb":null,"background_color":null,"text_color":null,"link_backg
## round_color":null,"link_text_color":null,"link_text":null,"link_url":null,"show_
## feature_image":false,"background_image_opacity":0.8,"should_show_feature_image_s
## ection":true,"feature_image_attributes":{"image_urls":{"default":"https://ksr-ug
## c.imgix.net/assets/011/595/301/16a59433ff3828eb5f05de1ccef2c818_original.jpg?ixl
## ib=rb-4.0.2&crop=faces&w=1552&h=873&fit=crop&v=1463684983&auto=format&frame=1&q=
## 92&s=37b64133ce1c93d74d38c29408d7aaf3","baseball_card":"https://ksr-ugc.imgix.ne
## t/assets/011/595/301/16a59433ff3828eb5f05de1ccef2c818_original.jpg?ixlib=rb-4.0.
## 2&crop=faces&w=560&h=315&fit=crop&v=1463684983&auto=format&frame=1&q=92&s=776070
## 16d544d789a5192a824eb85a83"}}}
```

No se observa información útil del perfil en formato JSON; se procede a eliminar esta variable.

```
dropped.unused.cols$profile <- NULL
dropped.unused.cols.dim <- dim(dropped.unused.cols)
```

```
attribute.summary(dropped.unused.cols, 'slug')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: slug
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## liquid-ambience-a-boutique-atmospheric-guitar-effe
## patchblocks-programmable-mini-synth-modules
## kitar-the-evolving-instrument
```

Un buen *slug* es importante para el SEO del proyecto; puede tener un impacto relevante en el éxito de su campaña de financiación.

```
attribute.summary(dropped.unused.cols, 'staff_pick')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: staff_pick
##
```

```
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## true true    true
```

Ser elegido por el equipo de Kickstarter como uno de sus proyectos favoritos probablemente sea un factor de éxito de las campañas de Kickstarter. Será necesario convertirlo a valores lógicos previamente.

```
attribute.summary(dropped.unused.cols, 'state')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: state
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## successful   successful  successful
```

Esta variable representa las diferentes clases del problema de clasificación, y es la variable que se desea predecir. Será necesario convertir este atributo a éxito/fracaso (1 o 0).

```
attribute.summary(dropped.unused.cols, 'usd_exchange_rate')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo:
## usd_exchange_rate
##
## Resumen de las características:
## Min.   :0   1st Qu.:1   Median :1   Mean   :1   3rd Qu.:1
## Max.   :2   NA's   :10195
##
## Ejemplos de uso:
## 0.82682006   1.63884186  1
```

Es una variable intermedia que se utilizará para calcular la cantidad de dinero objetivo que se pretende recaudar en cada proyecto. Como hay muchos valores nulos, se observa la variable `static_usd_rate` para ver si se puede completar la información:

```
attribute.summary(dropped.unused.cols, 'static_usd_rate')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo:
## static_usd_rate
##
## Resumen de las características:
```

```
## Min.   :0.01   1st Qu.:1.00   Median :1.00   Mean   :1.00
## 3rd Qu.:1.00   Max.   :1.72
##
## Ejemplos de uso:
## 0.82865695   1.59244797  1
```

La variable `static_usd_rate` se puede utilizar para completa los valores de `usd_exchange_rate` en el caso de valores nulos.

```
attribute.summary(dropped.unused.cols, 'usd_pledged')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: usd_pledged
##
## Resumen de las características:
## Min.   :       0   1st Qu.:     101   Median :    1500
## Mean   :   14070   3rd Qu.:    6559   Max.   :11385449
##
## Ejemplos de uso:
## 24224.12861935   106838.146455765    19024.69
```

Es la cantidad de dinero que se ha recaudado, convertido a dólares americanos. Será una variable útil para comprobar si se han sobrepasado los objetivos de financiación iniciales o lo lejos que se ha quedado el proyecto de ser financiado.

Se utilizará en el análisis inicial de forma descriptiva, ya que a la hora de determinar si un futuro proyecto se financiará exitosamente no se poseerá todavía información sobre la cantidad de dinero recaudada. Esta variable podría ser también la variable objetivo a predecir según el planteamiento del proyecto de minería.

```
attribute.summary(dropped.unused.cols, 'is_starrable')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: is_starrable
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## false    false   false
```

```
possible.values(dropped.unused.cols, 'is_starrable')
```

```
##
## Valores posibles:

##
##  false   true
```

```
## 203498   6821
```

La posibilidad de aparecer en la página principal de la plataforma parece a priori una característica a tener en cuenta a la hora de predecir el éxito o fracaso de una campaña de financiación. La variable habrá de ser convertida a valores lógicos.

```
attribute.summary(dropped.unused.cols, 'is_starred')
```

```
## [*] Dataframe: dropped.unused.cols    Atributo: is_starred
##
## Resumen de las características:
## Length:210319      Class :character   Mode  :character
##
## Ejemplos de uso:
## NA   NA  NA
```

```
possible.values(dropped.unused.cols, 'is_starred')
```

```
##
## Valores posibles:

##
##         false   true
## 122554    140      2
```

La variable no contiene información útil; por tanto se procede a su eliminación.

```
dropped.unused.cols$is_starred <- NULL
dropped.unused.cols.dim <- dim(dropped.unused.cols)
```

---

6 Limpieza de datos

A continuación, se procede a observar si hay valores nulos o vacíos en los registros:

```
colSums(is.na(dropped.unused.cols) | dropped.unused.cols == '')
```

```
##     backers_count             blurb          category        created_at
##                 0                 2                 0                 0
##           creator          deadline              goal                id
##                 0                 0                 0                 0
##      is_starrable       launched_at          location              name
##                 0                 0               213                 0
##             photo              slug       staff_pick             state
```

```
##                0                0                0                0
##  state_changed_at   static_usd_rate usd_exchange_rate       usd_pledged
##                0                0            10195                0
```

Solo se acepta que `blurb` pueda tener valores nulos o vacíos, ya que una descripción vacía es correcta. Se procede a rellenar los valores nulos `usd_exchange_rate` con la variable `static_usd_rate`, eliminar datos nulos o vacíos del resto de campos, cambiar los posibles nulos a cadenas vacías en `blurb` y eliminar registros con el mismo `id`:

```
sifted.data <- dropped.unused.cols %>%
  replace_na(list(blurb='')) %>%
  dplyr::mutate(usd_exchange_rate = dplyr::coalesce(usd_exchange_rate,
                                                    static_usd_rate)) %>%
  dplyr::select(-static_usd_rate) %>%
  dplyr::filter_all(all_vars(!is.na(.))) %>%
  dplyr::filter_at(vars(-blurb), all_vars(.!='')) %>%
  dplyr::distinct(id, .keep_all = TRUE)

sifted.data.dim <- dim(sifted.data)

colSums(is.na(sifted.data) | sifted.data == '')
```

```
##     backers_count             blurb          category        created_at
##                 0                 2                 0                 0
##           creator          deadline              goal                id
##                 0                 0                 0                 0
##      is_starrable       launched_at          location              name
##                 0                 0                 0                 0
##             photo              slug       staff_pick             state
##                 0                 0                 0                 0
##  state_changed_at usd_exchange_rate       usd_pledged
##                 0                 0                 0
```

Tras un segundo cribado observando los datos restantes, se obtienen 210.106 registros con 19 atributos cada uno de ellos.

7 Extracción de información de los datos

Se procede a realizar las transformaciones planteadas en la sección anterior para cada atributo. Esto incluye recuento de palabras, conversiones de tipos de valores y extracción de información de texto en formato JSON.

7.1 Número de palabras

Se procede a crear sendos atributos de recuento de palabras para el nombre, el *slug* y la descripción del proyecto.

```
sifted.data$blurb <- unlist(lapply(sifted.data$blurb, str_count,
                                   pattern='\\S+'))
sifted.data$slug <- unlist(lapply(sifted.data$slug, str_count,
                                  pattern='[^-]+'))
sifted.data$name <- unlist(lapply(sifted.data$name, str_count, pattern='\\S+'))

sifted.data %>% select(id, name, slug, blurb) %>% head(n=5)
```

```
##           id name slug blurb
## 1  838575496    8    7    19
## 2  147947813    6    5    16
## 3  293869602    4    4    12
## 4 1801284145    4    5    13
## 5  681994346    7    6    12
```

7.2 Conversión a fechas

Se procede convertir las variables de creación, lanzamiento, finalización y cambio de estado del proyecto al tipo fecha, siguiendo el estándar UNIX epoch.

```
sifted.data$created_at <- anytime(sifted.data$created_at)
sifted.data$launched_at <- anytime(sifted.data$launched_at)
sifted.data$state_changed_at <- anytime(sifted.data$state_changed_at)
sifted.data$deadline <- anytime(sifted.data$deadline)

sifted.data %>% select(id, created_at, launched_at, state_changed_at,
                       deadline) %>% head(n=5)
```

```
##           id          created_at         launched_at    state_changed_at
## 1  838575496 2013-10-22 05:23:34 2013-11-13 00:08:00 2013-12-13 00:08:00
## 2  147947813 2013-10-15 13:45:48 2013-11-04 23:05:26 2013-12-04 23:05:29
```

```
## 3  293869602 2012-11-24 03:36:22 2013-08-13 20:12:07 2013-08-29 20:12:07
## 4 1801284145 2013-05-09 13:15:49 2013-07-29 09:36:13 2013-09-03 00:59:01
## 5  681994346 2013-06-04 02:28:45 2013-07-09 07:01:25 2013-08-08 07:01:25
##              deadline
## 1 2013-12-13 00:08:00
## 2 2013-12-04 23:05:26
## 3 2013-08-29 20:12:07
## 4 2013-09-03 00:59:00
## 5 2013-08-08 07:01:25
```

7.3 Conversión a valores lógicos

Se procede a convertir las variables `is_starrable` y `staff_pick`, inicialmente formateadas como cadenas de caracteres, a valores lógicos.

En primer lugar, es necesario estudiar los valores que toma cada variable para poder decidir cómo convertir cada valor en un valor lógico.

```
unique(sifted.data$is_starrable)
```

```
## [1] "false" "true"
```

```
unique(sifted.data$staff_pick)
```

```
## [1] "true"  "false"
```

Ambas variables toman únicamente los valores `false` y `true`, cuya asignación resulta directa a sus correspondientes valores lógicos.

```
sifted.data <- sifted.data %>%
  dplyr::mutate(is_starrable = str_trim(tolower(is_starrable), side='both'),
                staff_pick = str_trim(tolower(staff_pick), side='both')) %>%
  dplyr::mutate(is_starrable = (is_starrable == 'true'),
                staff_pick = (staff_pick == 'true'))

sifted.data %>% select(id, is_starrable, staff_pick) %>% head(n=5)
```

```
##           id is_starrable staff_pick
## 1  838575496        FALSE       TRUE
## 2  147947813        FALSE       TRUE
## 3  293869602        FALSE       TRUE
## 4 1801284145        FALSE       TRUE
## 5  681994346        FALSE       TRUE
```

A continuación se procede a observar los valores que toma la variable `state`, que es

la que informa de si el proyecto ha superado con éxito la campaña de financiación colectiva o no.

```
sifted.data <- sifted.data %>%
  dplyr::mutate(state = str_trim(tolower(state), side='both'))
table(sifted.data$state)
```

```
##
##   canceled     failed       live successful
##       9620      77776       6925     115785
```

Los estados `failed`y `successful` se corresponden con las clases de fracaso y éxito respectivamente. Se tomará que los proyectos cuya campaña ha sido cancelada antes de la fecha de finalización (`canceled`) también suponen casos de fracaso. No se tomarán en cuenta los casos en que la campaña de financiación sigue en curso (`live`).

Una consideración importante a tener en cuenta es la posible correlación entre los proyectos con campaña de financiación en curso (`live`) y el atributo `is_starrable`, que hace referencia a si el proyecto puede guardarse en favoritos. Antes de eliminar estos registros, se analizar la distribución de `is_starrable` según el estado del proyecto.

```
sifted.data %>% dplyr::count(state, is_starrable)
```

```
##        state is_starrable      n
## 1   canceled        FALSE   9620
## 2     failed        FALSE  77776
## 3       live        FALSE    104
## 4       live         TRUE   6821
## 5 successful        FALSE 115785
```

Todos los proyectos que pueden ser guardados en favoritos tienen una campaña de financiación en curso; por tanto, se puede eliminar la variable `is_starrable` al eliminar los registros de estos proyectos, ya que solo contendrá valores falsos.

```
sifted.data <- sifted.data %>%
  dplyr::mutate(state = replace(state, state=='canceled', 'failed')) %>%
  dplyr::mutate(state = replace(state, state=='live', NA)) %>%
  dplyr::filter_all(all_vars(!is.na(.))) %>%
  dplyr::mutate(success= state=='successful') %>%
  dplyr::select(-state)
sifted.data.dim <- dim(sifted.data)

table(sifted.data$success)
```

```
##
##  FALSE   TRUE
##  87396 115785
```

```
table(sifted.data$is_starrable)
```

```
##
##  FALSE
## 203181
```

```
sifted.data$is_starrable <- NULL
```

7.4 Cambio de divisas

El atributo `goal`, que representa la cantidad de dinero necesaria para cumplir el objetivo de financiación de la campaña, se encuentra en la divisa local del país de origen. Este atributo hay que multiplicarlo por el ratio de cambio de divisa para transformarlo a dólares americanos (USD).

```
sifted.data <- sifted.data %>%
  dplyr::mutate(goal = goal * usd_exchange_rate) %>%
  dplyr::select(-usd_exchange_rate)
sifted.data.dim <- dim(sifted.data)
```

7.5 Extracción de información de JSON

Se procede a extraer información de las variables que almacenan textos en formato JSON: `category`, `creator` y `location`. Para ello, se podría utilizar una librería de *parseo* de texto en formato JSON, como puede ser `tidyjson` o `jsonlite` pero como la información a obtener es muy simple y se encuentra en el primer nivel del JSON, se creará una función para extraerla.

```
extract.json.attr <- function(json, attr.name, type='string'){
  pattern.1.level <- switch(
    type,
    string=paste('(\\"', attr.name, '\\"\\:\\"([^,{]+)\\")', sep=''),
    num=paste('(\\"', attr.name, '\\"\\:([^,{]+))', sep=''))
  tmp <- unlist(str_match(json, pattern=pattern.1.level))
  return(tmp[length(tmp)])
}
```

7.5.1 Categoría y subcategoría

De la variable que codifica la información de la categoría se va a obtener la categoría y subcategoría a la que pertenece el proyecto. Esta información se encuentra guardada en los campos `parent_name` y `name` respectivamente.

```
sifted.data$subcategory <- sifted.data$category %>%
  lapply(., extract.json.attr, attr.name='name', type='string') %>%
  unlist
sifted.data$category <- sifted.data$category %>%
  lapply(., extract.json.attr, attr.name='parent_name', type='string') %>%
  unlist

sum(is.na(sifted.data$category))
```

```
## [1] 9143
```

```
sum(is.na(sifted.data$subcategory))
```

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

Hay algunos proyectos que no tienen categoría, aunque sí subcategoría. Se asume entonces que algunos proyectos han sido creados en una categoría, sin seleccionar subcategoría, y por tanto se ha asignado la categoría a la mayor granularidad posible del campo. Se procede a copiar el valor de la subcategoría a la categoría para estos casos:

```
sifted.data <- sifted.data %>%
  mutate(category= coalesce(category, subcategory))
sum(is.na(sifted.data$category))
```

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

```
sifted.data.dim <- dim(sifted.data)
```

7.5.2 Experiencia de los creadores

De la información del creador se pretende obtener varios atributos directos y derivados: por un lado, el atributo `is_superbacker` que ofrece información de la experiencia que tiene el creador como patrocinador de otros proyectos, y a través del identificador del creador se puede obtener su número de proyectos con campañas de financiación lanzadas.

```
sifted.data$superbacker <- sifted.data$creator %>%
  lapply(., extract.json.attr, attr.name='is_superbacker', type='num') %>%
  unlist
sifted.data$creator <- sifted.data$creator %>%
  lapply(., extract.json.attr, attr.name='id', type='num') %>%
  unlist

attribute.summary(sifted.data, 'superbacker')
```

```
## [*] Dataframe: sifted.data    Atributo: superbacker
##
## Resumen de las características:
## Length:203181      Class :character   Mode  :character
##
## Ejemplos de uso:
## null null    null
```

```
possible.values(sifted.data, 'superbacker')
```

```
##
## Valores posibles:

##
##   null
## 203181
```

```
attribute.summary(sifted.data, 'creator')
```

```
## [*] Dataframe: sifted.data    Atributo: creator
##
## Resumen de las características:
## Length:203181      Class :character   Mode  :character
##
## Ejemplos de uso:
## 556577677    2012099678  837684442
```

```
sum(is.na(sifted.data$creator))
```

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

Se observa que el atributo `superbacker` no contiene información útil; se procederá a su eliminación. Además, el atributo `creator` es el identificador numérico del usuario creador del proyecto, pero lo interesante del atributo es conocer la cantidad de proyectos que ha realizado el creador, no su identificador. Se procede a contar los identificadores repetidos.

```
sifted.data$superbacker <- NULL
sifted.data <- sifted.data %>% mutate(creator=as.numeric(creator))

creator.freq <- as.data.frame(table(sifted.data$creator)) %>%
  transmute(creator = as.numeric(as.character(Var1)), freq = Freq)

sifted.data <- dplyr::left_join(x=sifted.data, y=creator.freq, by='creator')
sifted.data$creator <- sifted.data$freq
sifted.data$freq <- NULL

attribute.summary(sifted.data, 'creator')
```

```
## [*] Dataframe: sifted.data    Atributo: creator
##
## Resumen de las características:
## Min.   : 1   1st Qu.: 1   Median : 1   Mean   : 2
## 3rd Qu.: 1   Max.   :60
##
## Ejemplos de uso:
## 1    2   1
```

```
sum(is.na(sifted.data$creator))
```

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

```
sifted.data.dim <- dim(sifted.data)
```

La variable `creator` hace referencia al número de proyectos (contando el observado en la fila) que el usuario que creador ha desarrollado también. La experiencia de un usuario en la plataforma puede tener relevancia a la hora de asegurar el éxito de la campaña de financiación de un proyecto.

7.5.3 Datos de localización

La variable `location` tiene información de la ciudad (atributo `name`), región (`state`) y país (`country`) al que pertenece el proyecto. Con esta información se puede investigar si el lugar de origen del proyecto tiene relación con el éxito de su campaña de financiación.

```
sifted.data$city <- sifted.data$location %>%
  lapply(., extract.json.attr, attr.name='name', type='string') %>%
  unlist
sifted.data$region <- sifted.data$location %>%
```

```
  lapply(., extract.json.attr, attr.name='state', type='string') %>%
  unlist
sifted.data$expanded.country <- sifted.data$location %>%
  lapply(., extract.json.attr, attr.name='expanded_country', type='string') %>%
  unlist
sifted.data$country <- sifted.data$location %>%
  lapply(., extract.json.attr, attr.name='country', type='string') %>%
  unlist

sifted.data$location <- NULL

attribute.summary(sifted.data, 'city')
```

```
## [*] Dataframe: sifted.data    Atributo: city
##
## Resumen de las características:
## Length:203181      Class :character   Mode  :character
##
## Ejemplos de uso:
## Christchurch Central Belfast Los Angeles
```

```
sum(is.na(sifted.data$city))
```

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

```
attribute.summary(sifted.data, 'region')
```

```
## [*] Dataframe: sifted.data    Atributo: region
##
## Resumen de las características:
## Length:203181      Class :character   Mode  :character
##
## Ejemplos de uso:
## Canterbury   Northern Ireland    CA
```

```
sum(is.na(sifted.data$region))
```

```
## [1] 55
```

```
attribute.summary(sifted.data, 'country')
```

```
## [*] Dataframe: sifted.data    Atributo: country
##
## Resumen de las características:
## Length:203181      Class :character   Mode  :character
```

```
##
## Ejemplos de uso:
## NZ   GB  US
```

```
sum(is.na(sifted.data$country))
```

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

Existen 55 registros sin región asociada; se procede a eliminar dichos registros.

```
sifted.data <- sifted.data %>% dplyr::filter_all(all_vars(!is.na(.)))
sifted.data.dim <- dim(sifted.data)
```

7.5.4 Fotografía del proyecto

La variable `photo` contiene información de la fotografía del proyecto a distintas resoluciones. Esta información podría ser estudiada mediante técnicas de análisis fotográfico, que sobrepasan el alcance de la asignatura; para este trabajo se utilizará únicamente si existe o no fotografía en el proyecto.

```
sifted.data$photo <- sifted.data$photo %>%
  lapply(., function(item) nchar(extract.json.attr(
    json=item, attr.name='key', type='string'))>0) %>%
  unlist %>% replace_na(FALSE)
```

Tras la extracción de información de las variables en bruto del *dataset*, se obtiene un total de 203.094 observaciones con 21 atributos en cada una de ellas.

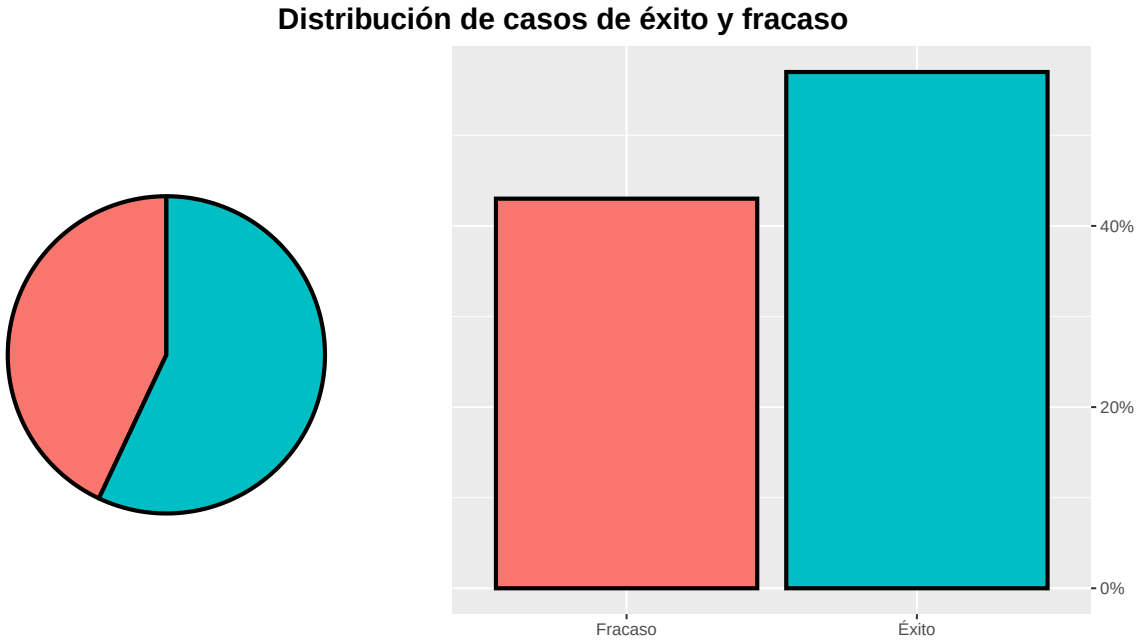
```
clean.data <- sifted.data
clean.data.dim <- dim(clean.data)
```

8 Distribución de los datos

Se procede a examinar la distribución de los datos; para ello se utilizarán *violin plots* en las variables continuas, lo que permitirá observar la distribución y densidad de los datos a la vez, y *bar plots* para las variables discretas.

8.1 Casos de éxito y fracaso

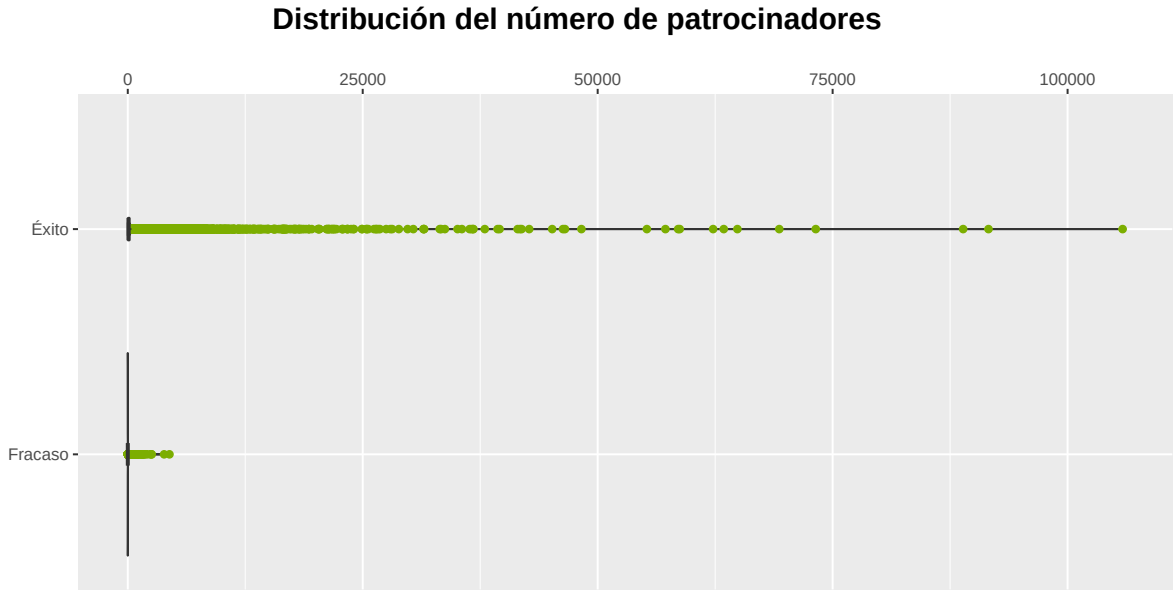
```
success.props <-as.data.frame(table(clean.data$success)) %>%
  dplyr::mutate(key=Var1, value=Freq) %>%
  dplyr::select(key, value)

annotate_figure(top=text_grob(face='bold', size=16,
  label='Distribución de casos de éxito y fracaso'),
  ggarrange(nrow=1, ncol=2, align='v', widths=c(1,1.5),
    ggplot(data=success.props, mapping=aes(x='', y=value, fill=key)) +
      geom_col(color='black', size=1.05) +
      coord_polar(theta='y') + theme_void() +
      theme(legend.position='none') + no.axis.x + no.axis.y +
      xlab('') + ylab(''),

    ggplot(data=clean.data, mapping=aes(x=success, fill=success)) +
      geom_bar(aes(y = (..count..)/sum(..count..)), color='black',
               size=1.05) +
      xlab('') + ylab('') +
      scale_y_continuous(position='right', labels=percent) +
      scale_x_discrete(labels=function(label)
        ifelse(label, 'Éxito', 'Fracaso')) +
      theme(legend.position='none')
      ))
```



El número de casos de éxito y fracaso se encuentra equilibrado: existen 115.744 casos de éxito y 87.350 casos de fracaso de la campaña de financiación.

8.2 Número de patrocinadores

```
ggplot(data=clean.data, mapping=aes(x=success, y=backers_count)) +
  geom_violin(fill=default.color.main) +
  xlab('') + ylab('') +
  scale_y_continuous(position='right') +
  scale_x_discrete(labels=function(label) ifelse(label, 'Éxito', 'Fracaso')) +
  geom_boxplot(width=0.1, outlier.color=default.color.terciary,
               fill=default.color.secondary) +
  coord_flip() +
  ggtitle('Distribución del número de patrocinadores') +
  title.centered
```




```
summary(clean.data$backers_count[clean.data$success == TRUE])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1      33      73     265     175  105857
```

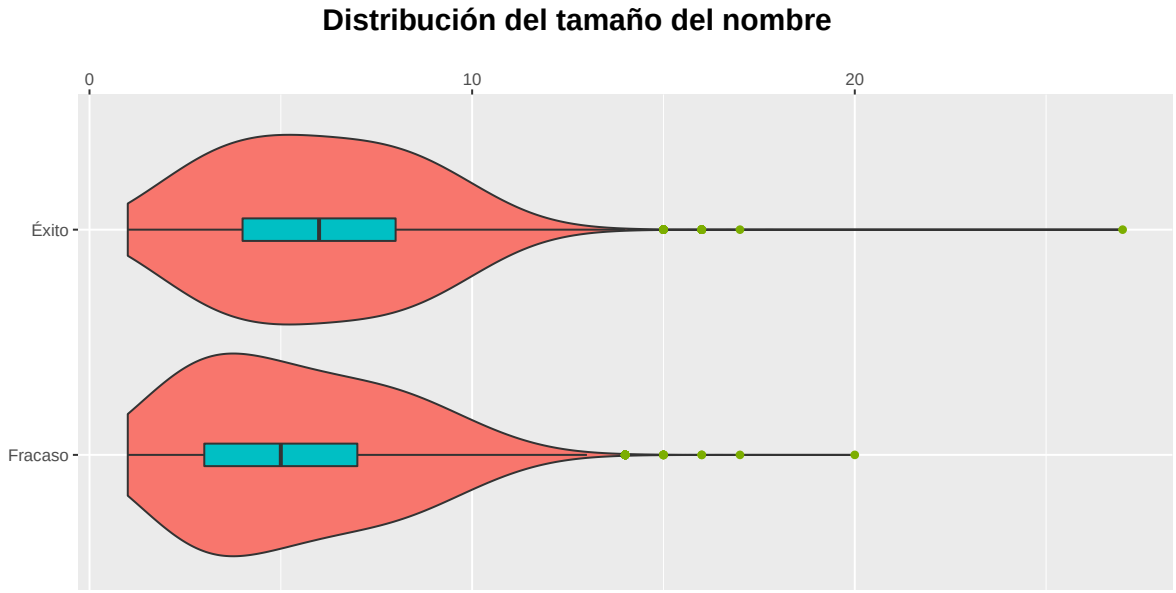
```
summary(clean.data$backers_count[clean.data$success == FALSE])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       1       3      12       9    4435
```

Los proyectos financiados con éxito tienen de media más patrocinadores que los que fracasan en su campaña.

8.3 Extensión del nombre, *slug* y descripción

```
ggplot(data=clean.data, mapping=aes(x=success, y=name)) +
  geom_violin(fill=default.color.main, adjust=4) +
  xlab('') + ylab('') +
  scale_y_continuous(position='right') +
  scale_x_discrete(labels=function(label) ifelse(label, 'Éxito', 'Fracaso')) +
  geom_boxplot(width=0.1, outlier.color=default.color.terciary,
               fill=default.color.secondary) +
  coord_flip() +
  ggtitle('Distribución del tamaño del nombre') +
  title.centered
```



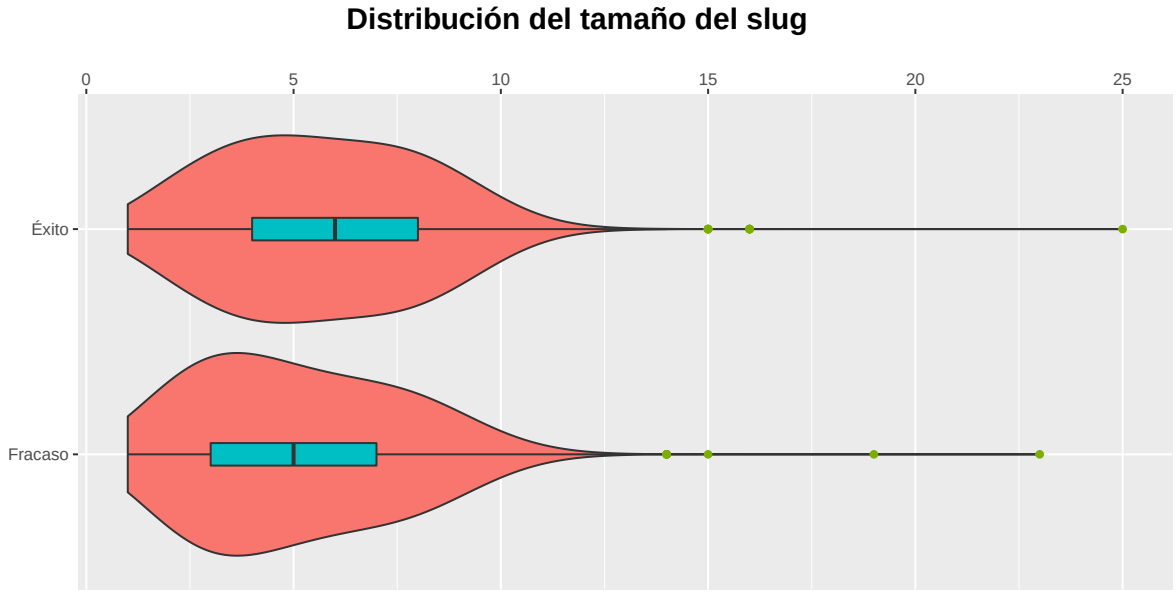

```
summary(clean.data$name[clean.data$success == TRUE])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       4       6       6       8      27
```

```
summary(clean.data$name[clean.data$success == FALSE])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     3.0     5.0     5.3     7.0    20.0
```

```
ggplot(data=clean.data, mapping=aes(x=success, y=slug)) +
  geom_violin(fill=default.color.main, adjust=4) +
  xlab('') + ylab('') +
  scale_y_continuous(position='right') +
  scale_x_discrete(labels=function(label) ifelse(label, 'Éxito', 'Fracaso')) +
  geom_boxplot(width=0.1, outlier.color=default.color.terciary,
               fill=default.color.secondary) +
  coord_flip() +
  ggtitle('Distribución del tamaño del slug') +
  title.centered
```




```
summary(clean.data$slug[clean.data$success == TRUE])
```

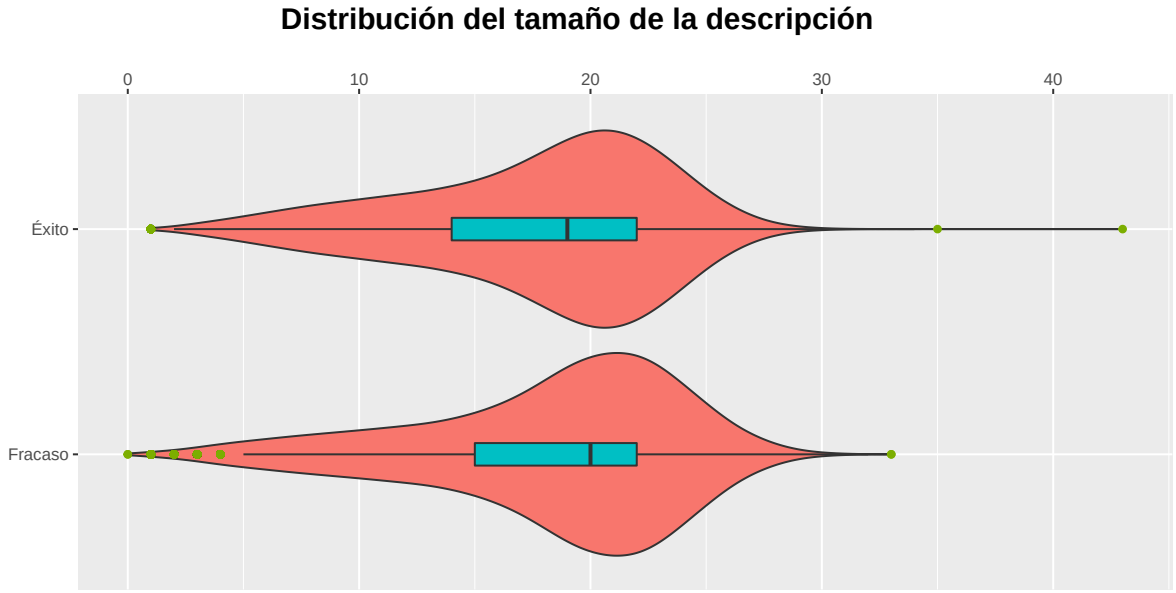
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     4.0     6.0     5.7     8.0    25.0
```

```
summary(clean.data$slug[clean.data$success == FALSE])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     3.0     5.0     5.1     7.0    23.0
```

La distribución de valores de la extensión del nombre y el *slug* es similar para los casos de éxito y fracaso. No parecen tener relevancia a la hora de determinar si la financiación de un proyecto tendrá o no éxito.

```
ggplot(data=clean.data, mapping=aes(x=success, y=blurb)) +
  geom_violin(fill=default.color.main, adjust=2) +
  xlab('') + ylab('') +
  scale_y_continuous(position='right') +
  scale_x_discrete(labels=function(label) ifelse(label, 'Éxito', 'Fracaso')) +
  geom_boxplot(width=0.1, outlier.color=default.color.terciary,
               fill=default.color.secondary) +
  coord_flip() +
  ggtitle('Distribución del tamaño de la descripción') +
  title.centered
```




```
summary(clean.data$blurb[clean.data$success == TRUE])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1      14      19      18      22      43
```

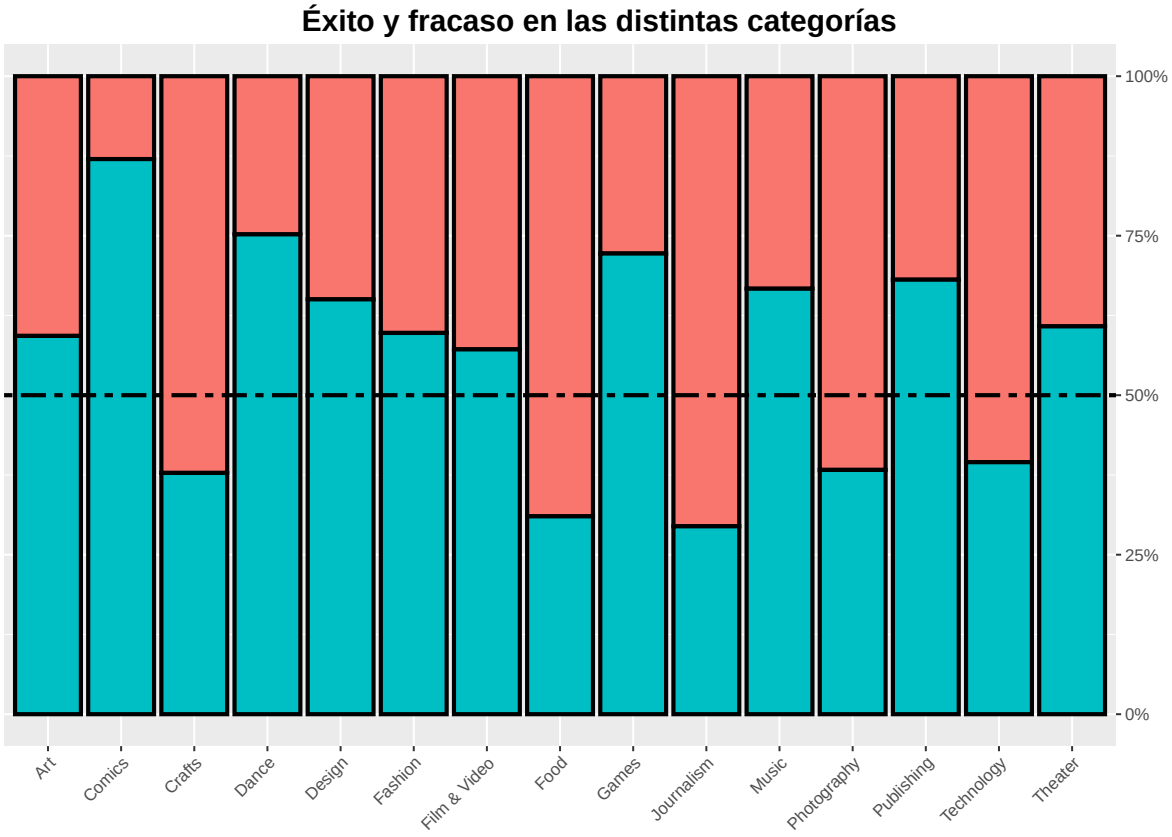
```
summary(clean.data$blurb[clean.data$success == FALSE])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0      15      20      18      22      33
```

Todos los proyectos que tienen éxito en su campaña de financiación comparten el hecho de tener una descripción. El resto de estadísticas — medias, cuartiles y valores extremos — se encuentran en rangos muy similares, con densidad muy parecidas. En base al análisis visual, se puede concluir que el número de palabras de la descripción del proyecto no es por sí solo muy relevante para su éxito.

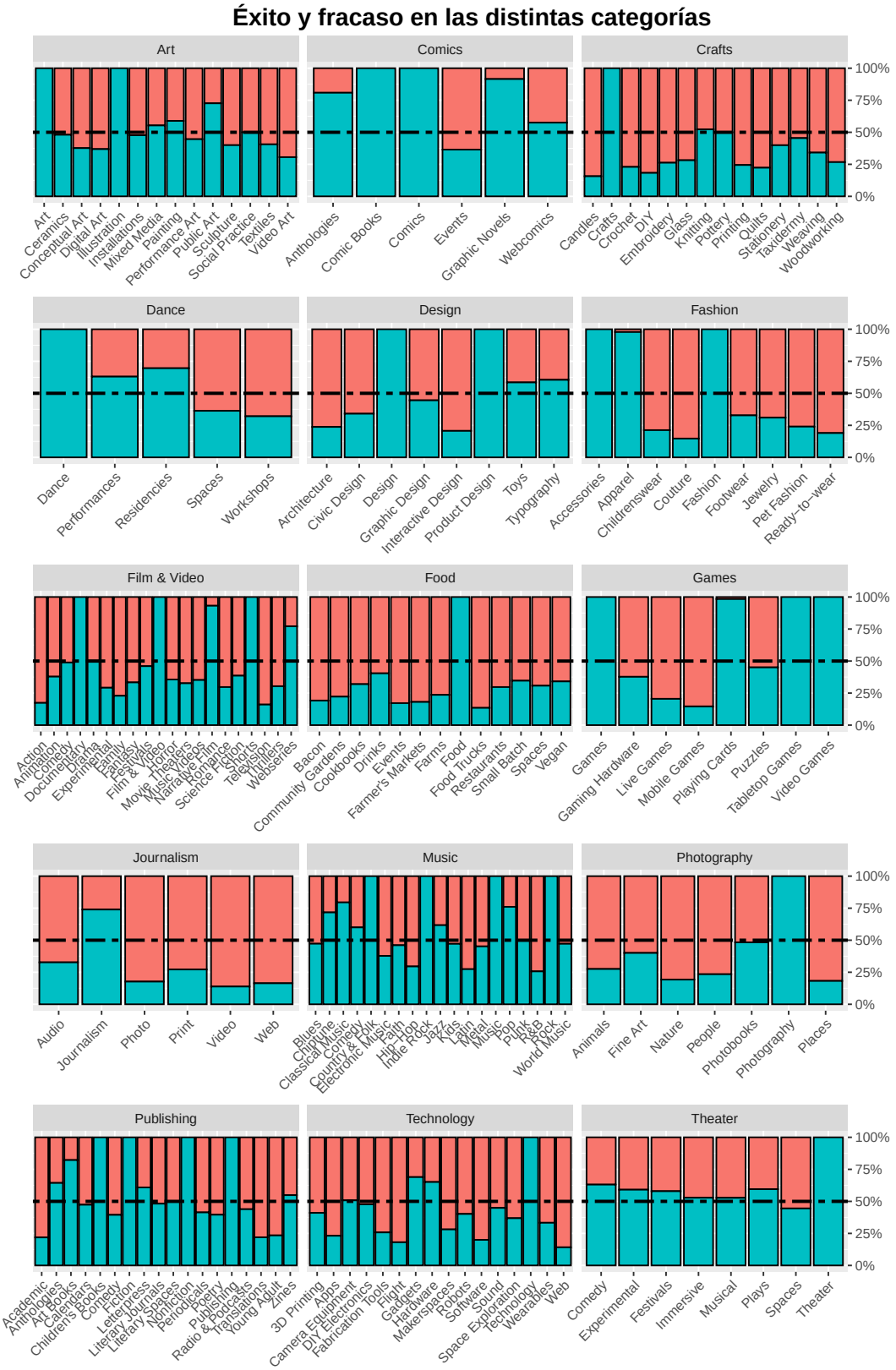
8.4 Categoría y subcategoría del proyecto

```
ggplot(data=clean.data, mapping=aes(x=category, fill=success)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), color='black',
           size=1.05, position='fill') +
  geom_hline(yintercept=0.5, color='black', linetype='twodash', size=1.05) +
  scale_y_continuous(position='right', labels=percent) +
  xlab('') + ylab('') + theme(legend.position='none') +
  theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1)) +
  ggtitle('Éxito y fracaso en las distintas categorías') + title.centered
```



Los proyectos de cómics, danza y videojuegos son los que más se financian en la plataforma **Kickstarter**, mientras que los relacionados con comida o periodismo son lo que menos éxito tienen.

```
ggplot(data=clean.data, mapping=aes(x=subcategory, fill=success)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), color='black',
           position='fill') +
  facet_wrap(. ~ category, scales='free_x', nrow=5, ncol=3) +
  geom_hline(yintercept=0.5, color='black', linetype='twodash', size=1.025) +
  scale_y_continuous(position='right', labels=percent) +
  xlab('') + ylab('') + theme(legend.position='none') +
  theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1)) +
  theme(plot.margin = unit(c(0.5,0,0,1), "cm")) +
  ggtitle('Éxito y fracaso en las distintas categorías') + title.centered
```


Éxito y fracaso en las distintas categorías
Art
Art
Ceramics
Conceptual Art
Digital Art
Illustration
Installations
Mixed Media
Painting
Performance Art
Public Art
Sculpture
Social Practice
Textiles
Video Art
Comics
Anthologies
Comic Books
Comics
Events
Graphic Novels
Webcomics
Crafts
Candles
Crafts
Crochet
DIY
Embroidery
Glass
Knitting
Pottery
Printing
Quilts
Stationery
Taxidermy
Weaving
Woodworking
Dance
Dance
Performances
Residencies
Spaces
Workshops
Design
Architecture
Civic Design
Design
Graphic Design
Interactive Design
Product Design
Toys
Typography
Fashion
Accessories
Apparel
Childrenswear
Couture
Fashion
Footwear
Jewelry
Pet Fashion
Ready-to-wear
Film & Video
Action
Animation
Comedy
Documentary
Drama
Experimental
Family
Fantasy
Festivals
Film & Video
Horror
Movie Theaters
Music Video
Narrative Film
Romance
Science Fiction
Shorts
Television
Thrillers
Webseries
Food
Bacon
Community Gardens
Cookbooks
Drinks
Events
Farmer's Markets
Farms
Food
Food Trucks
Restaurants
Small Batch
Spaces
Vegan
Games
Games
Gaming Hardware
Live Games
Mobile Games
Playing Cards
Puzzles
Tabletop Games
Video Games
Journalism
Audio
Journalism
Photo
Print
Video
Web
Music
Blues
Chiptune
Classical Music
Comedy
Country & Folk
Electronic Music
Faith
Hip Hop
Indie Rock
Jazz
Kids
Latin
Metal
Music
Pop
Punk
R&B
Rock
World Music
Photography
Animals
Fine Art
Nature
People
Photobooks
Photography
Places
Publishing
Academic
Anthologies
Art Books
Calendars
Children's Books
Comedy
Fiction
Letterpress
Literary Journals
Literary Spaces
Nonfiction
Periodicals
Poetry
Publishing
Radio & Podcasts
Translations
Young Adult
Zines
Technology
3D Printing
Apps
Camera Equipment
DIY Electronics
Fabrication Tools
Flight
Gadgets
Hardware
Makerspaces
Robots
Software
Sound
Space Exploration
Technology
Wearables
Web
Theater
Comedy
Experimental
Festivals
Immersive
Musical
Plays
Spaces
Theater
100%
75%
50%
25%
0%

Se observa que hay ciertas subcategorías cuyos proyectos alcanzan sus objetivos de financiación consistentemente. Por ejemplo, dentro de la categoría de publicaciones, los libros infantiles y los géneros de ficción y no ficción no tienen un solo proyecto que no haya sido financiado.

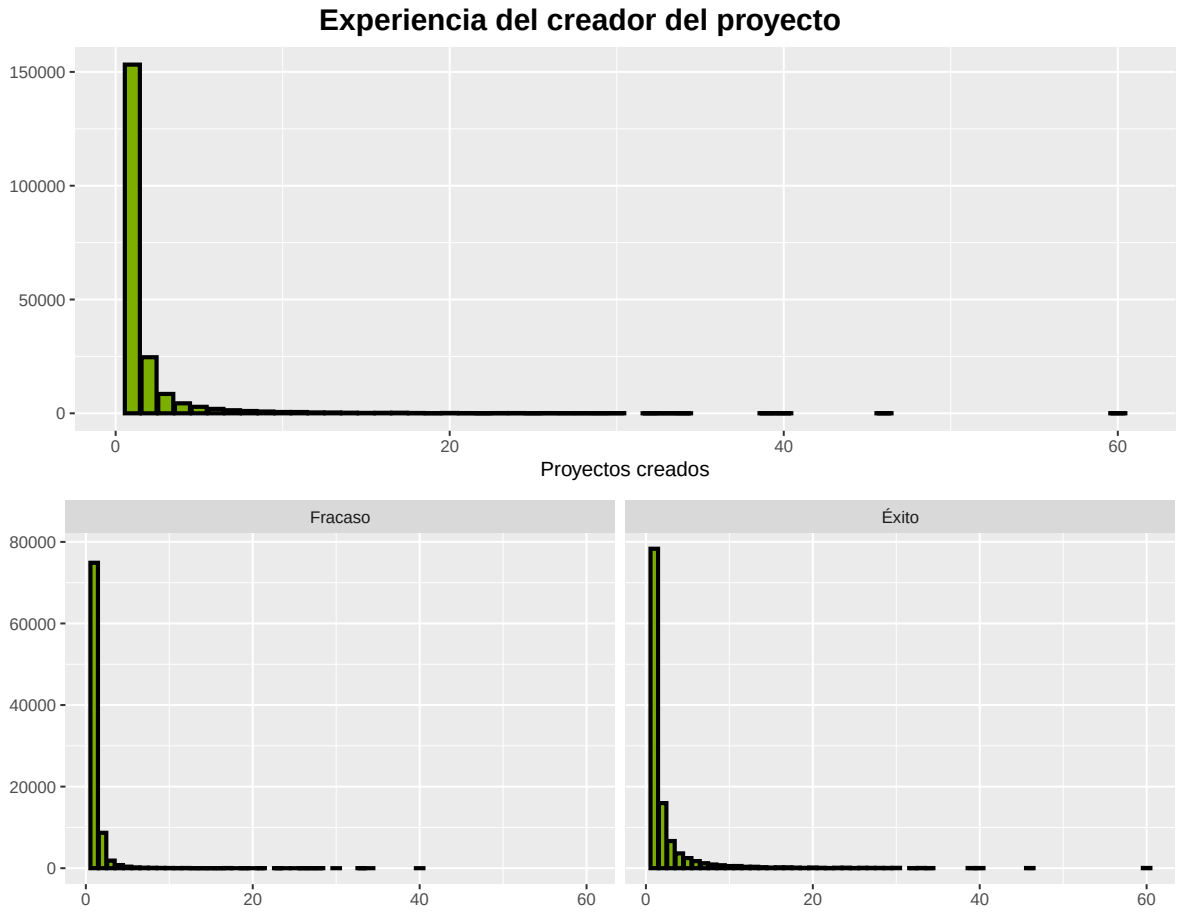
Como datos interesantes a destacar, los proyectos de la categoría de cómics son los más consistentes en cuanto a su éxito de financiación: los patrocinadores no consideran el género del cómic como determinante a la hora de invertir en él. Además, los eventos de cómics no resultan nada interesante.

Se puede observar que no incluir una subcategoría en el proyecto da los mejores resultados para casi todas las categorías, salvo en el caso de periodismo.

Con estos datos se puede obtener un perfil de los patrocinadores que utilizan **Kickstarter**, aunque este no es el objetivo del trabajo.

8.5 Experiencia del creador

```
annotate_figure(
  top=text_grob(face='bold', size=16, 'Experiencia del creador del proyecto'),
  ggarrange(nrow=2, ncol=1,
    ggplot(data=clean.data, mapping=aes(x=creator)) +
      geom_bar(size=1.05, color='black', fill=default.color.terciary) +
      xlab('Proyectos creados') + ylab(''),

    ggplot(data=clean.data, mapping=aes(x=creator)) +
      geom_bar(size=1.05, color='black', fill=default.color.terciary) +
      xlab('') + ylab('') + facet_wrap(.~success, labeller=success.labeller)
    ))
```




```
attribute.summary(clean.data, 'creator')
```

```
## [*] Dataframe: clean.data     Atributo: creator
##
## Resumen de las características:
## Min.   : 1   1st Qu.: 1   Median : 1   Mean   : 2
## 3rd Qu.: 1   Max.   :60
##
## Ejemplos de uso:
## 1    2   1
```

```
summary(clean.data$creator[clean.data$success])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       2       2      60
```

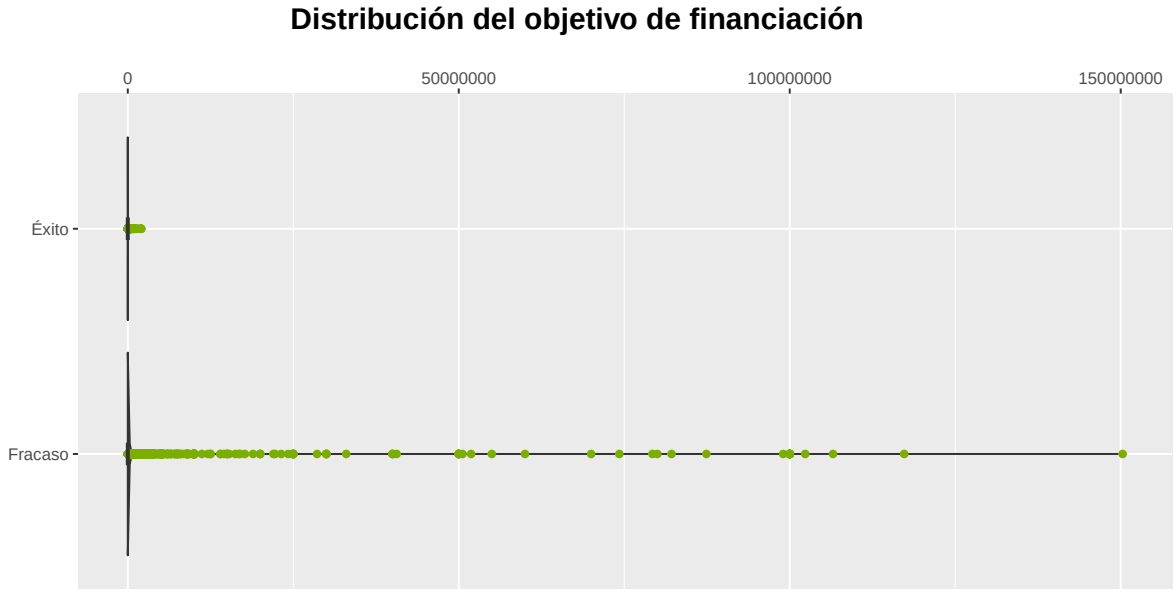
```
summary(clean.data$creator[!clean.data$success])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1      40
```

La mayoría de los usuarios creadores de proyectos en **Kickstarter** son novatos (solo han publicado un único proyecto). Analizando la distribución en los casos de éxito y fracaso, se observa que haber publicado varios proyectos en la plataforma podría tener cierto peso a la hora de determinar si un proyecto supera la campaña de financiación.

8.6 Objetivo de financiación y recaudación obtenida

```
ggplot(data=clean.data, mapping=aes(x=success, y=goal)) +
  geom_violin(fill=default.color.main) +
  xlab('') + ylab('') +
  scale_y_continuous(position='right') +
  scale_x_discrete(labels=function(label) ifelse(label, 'Éxito', 'Fracaso')) +
  geom_boxplot(width=0.1, outlier.color=default.color.terciary,
               fill=default.color.secondary) +
  coord_flip() +
  ggtitle('Distribución del objetivo de financiación') +
  title.centered
```




```
summary(clean.data$goal[clean.data$success])
```

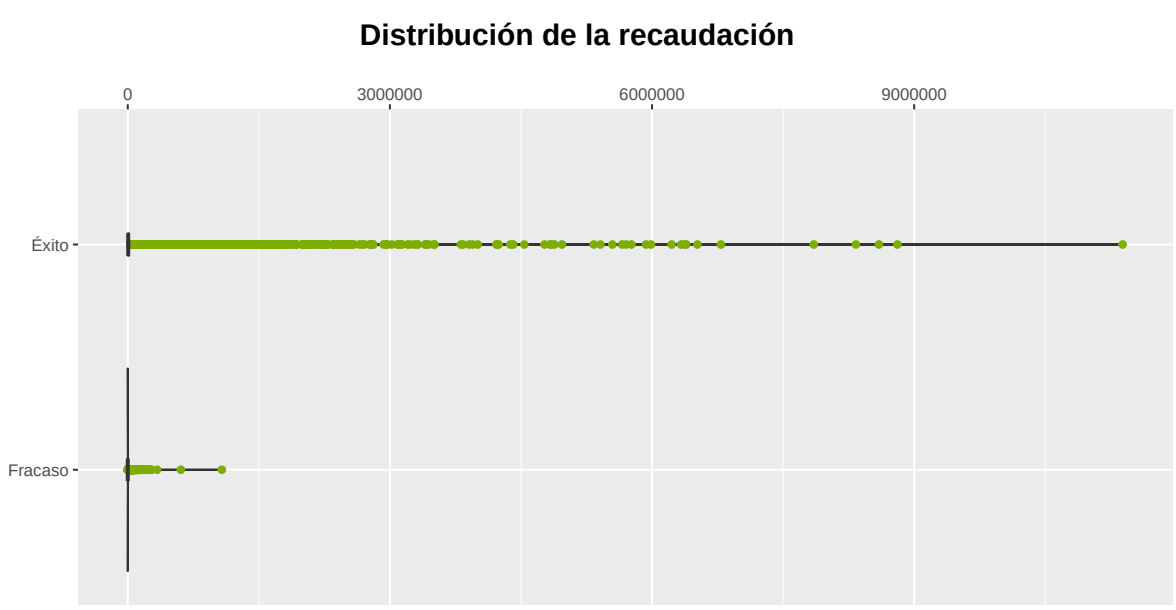
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    1000    3385    8765    9085 2000000
```

```
summary(clean.data$goal[!clean.data$success])
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##         1      2401      7000     86186     22512 150289382
```

Se observa que las cantidades de dólares americanos establecidas como objetivo de financiación son significativamente menores en los proyectos que se han financiado con éxito. Así, se puede inferir que los proyectos de menor envergadura y que necesitan menos dinero para ser llevados a cabo tienen más posibilidades de triunfar.

```
ggplot(data=clean.data, mapping=aes(x=success, y=usd_pledged)) +
  geom_violin(fill=default.color.main) +
  xlab('') + ylab('') +
  scale_y_continuous(position='right') +
  scale_x_discrete(labels=function(label) ifelse(label, 'Éxito', 'Fracaso')) +
  geom_boxplot(width=0.1, outlier.color=default.color.terciary,
               fill=default.color.secondary) +
  coord_flip() +
  ggtitle('Distribución de la recaudación') +
  title.centered
```




```
summary(clean.data$usd_pledged[clean.data$success])
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##         1      1829      5050     24158     13289 11385449
```

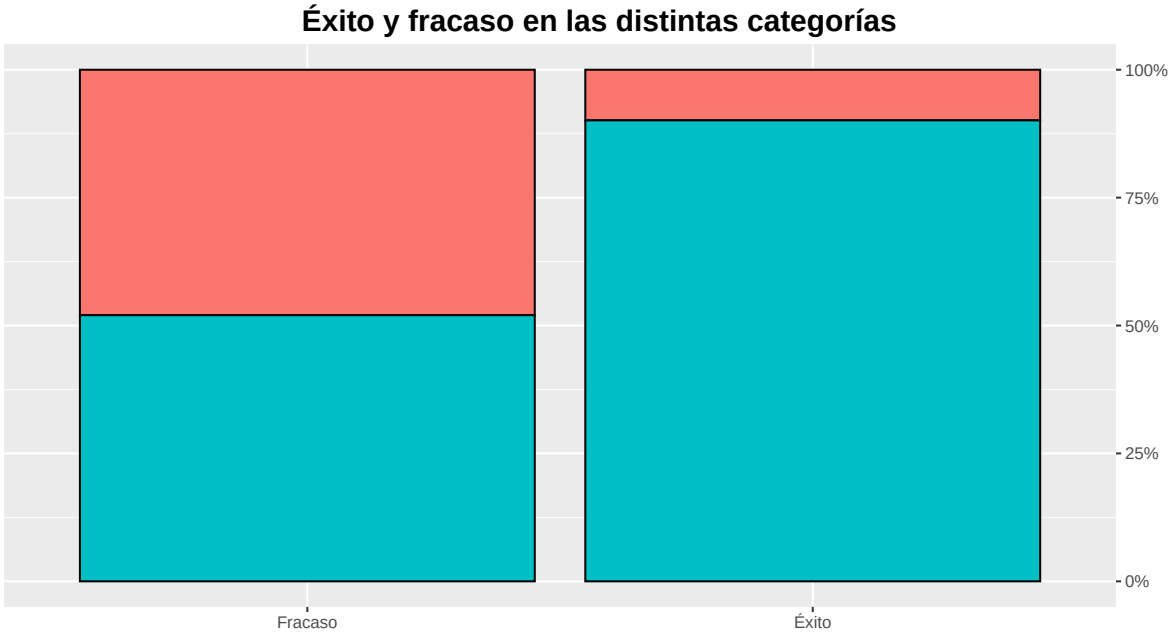
```
summary(clean.data$usd_pledged[!clean.data$success])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       1      56    1158     445 1076751
```

Como era de esperar, los proyectos que han sido financiados con éxito han recaudado de media mucho más que los proyectos que han fracasado en su campaña.

8.7 Impacto de ser elegido por el equipo de Kickstarter

```
ggplot(data=clean.data, mapping=aes(x=staff_pick, fill=success)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), color='black',
           position='fill') +
  scale_y_continuous(position='right', labels=percent) +
  xlab('') + ylab('') + theme(legend.position='none') +
  scale_x_discrete(labels=function(label) ifelse(label, 'Éxito', 'Fracaso')) +
  ggtitle('Éxito y fracaso en las distintas categorías') + title.centered
```



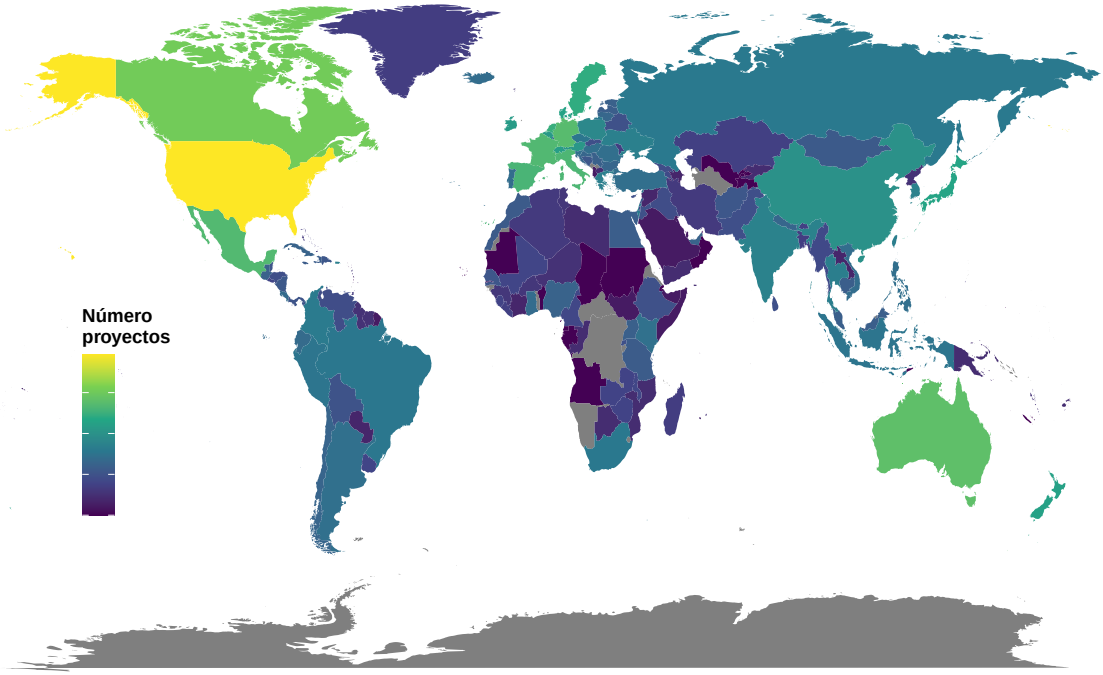
Se observa que ser escogido por el equipo desarrollador de la plataforma de financiación supone un gran impacto para cumplir el objetivo de financiación: la mayoría de los proyectos de éxito habían sido escogidos.

Por otro lado, tampoco se trata de una medida definitiva: la mitad de los proyectos que fracasan en la fase de financiación habían sido escogidos.

8.8 Países de origen de los proyectos

```
world.map <- map_data('world')
data("iso3166")

location.data <- clean.data %>% dplyr::select(country) %>%
  count(country, name='count') %>%
  full_join(., iso3166, by=c('country'='a2')) %>%
  dplyr::select(-a3, -ISOname, -sovereignty, -country) %>%
  left_join(., world.map, by=c('mapname'='region')) %>%
  dplyr::select(-subregion) %>%
  replace_na(list(count=0)) %>%
  dplyr::filter_all(all_vars(!is.na(.)))
location.data <- location.data[order(location.data$order),]

ggplot(location.data, aes(long, lat)) +
  geom_polygon(aes(group=group, fill=log(count))) +
  scale_fill_viridis_c() + theme_void() +
  title.centered + xlab('') + ylab('') + labs(fill='Número\nproyectos') +
  theme(legend.position=c(0.15, 0.4), legend.text=element_blank(),
        legend.title=element_text(face='bold', size=10, hjust=0.0)) +
  theme(plot.margin=unit(c(0,0,-1.0,0), 'cm')) +
  ggtitle('Distribución de los proyectos por países')
```

Distribución de los proyectos por países



El número de proyectos se ecuentra en escala logarítmica.

En el mapa se observa que Estados Unidos es, sin duda, el país que aglutina la mayoría de los proyectos que intentan financiarse a través de **Kickstarter**, seguido de Canadá y México.

Destaca Australia, que se sitúa como generador de proyectos de **Kickstarter** por delante de los países europeos; y África y Oriente Medio como las regiones que menos utilizan esta plataforma.

9 Preprocesado y gestión de características

Respecto a las fechas de creación, lanzamiento, *deadline* y cambio de estado de los proyectos, se pueden obtener datos útiles como tiempo de preparación de la campaña, periodo de recaudación o tiempo en que un proyecto termina de financiarse (con un resultado positivo o negativo).

- El **tiempo de preparación de la campaña** hace referencia a la diferencia de tiempo entre que se crea el proyecto hasta que se lanza la campaña de financiación.
- El **periodo de recaudación** es la diferencia entre la fecha de finalización de la campaña y el inicio de la misma.
- El **tiempo en que un proyecto termina su campaña de financiación** corresponde con la diferencia entre el final de la campaña y la fecha en que cambia el estado de la financiación, ya sea porque se cierra el periodo antes o porque se espera a que se cierre automáticamente al llegar a la *deadline*.

```
clean.data <- clean.data %>% dplyr::mutate(
  prep.period=as.integer(difftime(launched_at, created_at, units='days')),
  collection.period=as.integer(difftime(deadline, launched_at, units='days')),
  finish.period=as.integer(difftime(deadline, state_changed_at,
                                    units='days')))
clean.data$finish.period <- unlist(sapply(clean.data$finish.period,
                                          function(item) max(item, 0)))

attribute.summary(clean.data, 'prep.period')
```

```
## [*] Dataframe: clean.data     Atributo: prep.period
##
## Resumen de las características:
## Min.    :    0   1st Qu.:   3   Median :  11   Mean    :  48
## 3rd Qu.:  37   Max.    :3318
##
## Ejemplos de uso:
## 21   20  262
```

```
attribute.summary(clean.data, 'collection.period')
```

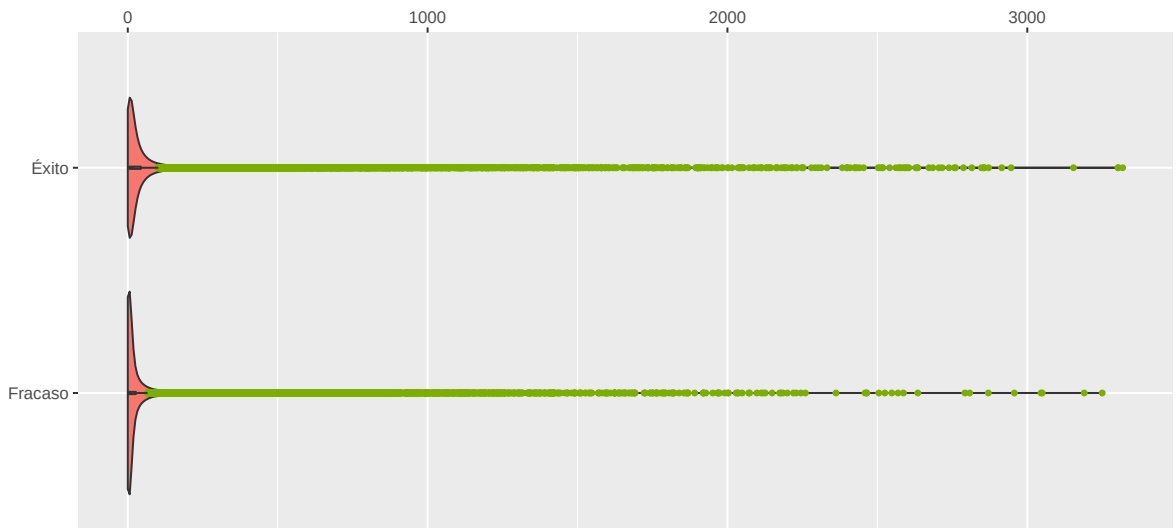
```
## [*] Dataframe: clean.data     Atributo: collection.period
##
## Resumen de las características:
## Min.    :  1   1st Qu.: 29   Median : 30   Mean    : 33
```

```
## 3rd Qu.: 35   Max.   :120
##
## Ejemplos de uso:
## 30   30  16
```

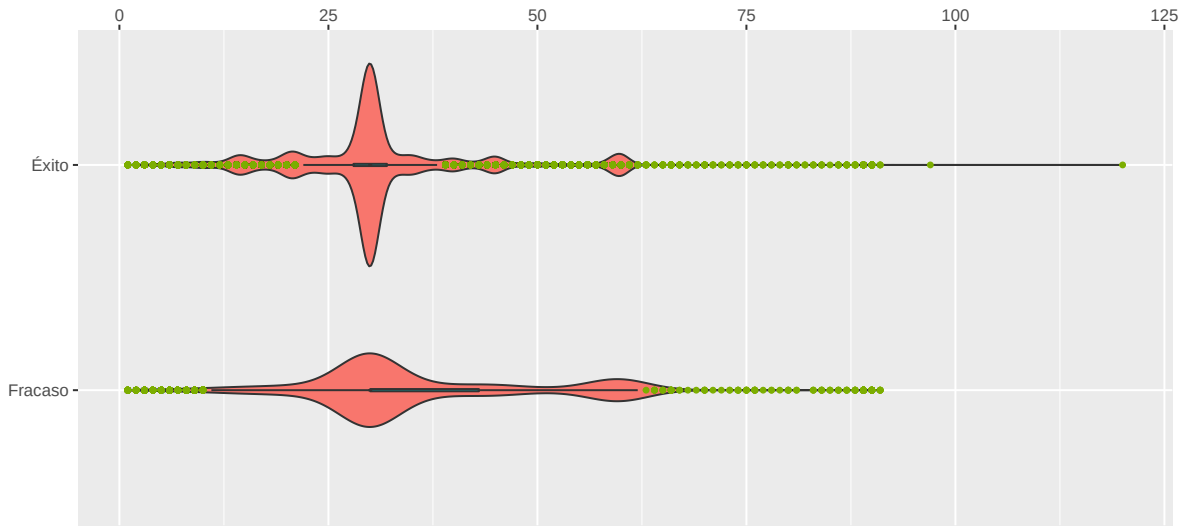
```
attribute.summary(clean.data, 'finish.period')
```

```
## [*] Dataframe: clean.data     Atributo: finish.period
##
## Resumen de las características:
## Min.   : 0   1st Qu.: 0   Median : 0   Mean   : 1
## 3rd Qu.: 0   Max.   :85
##
## Ejemplos de uso:
## 0    0   0
```

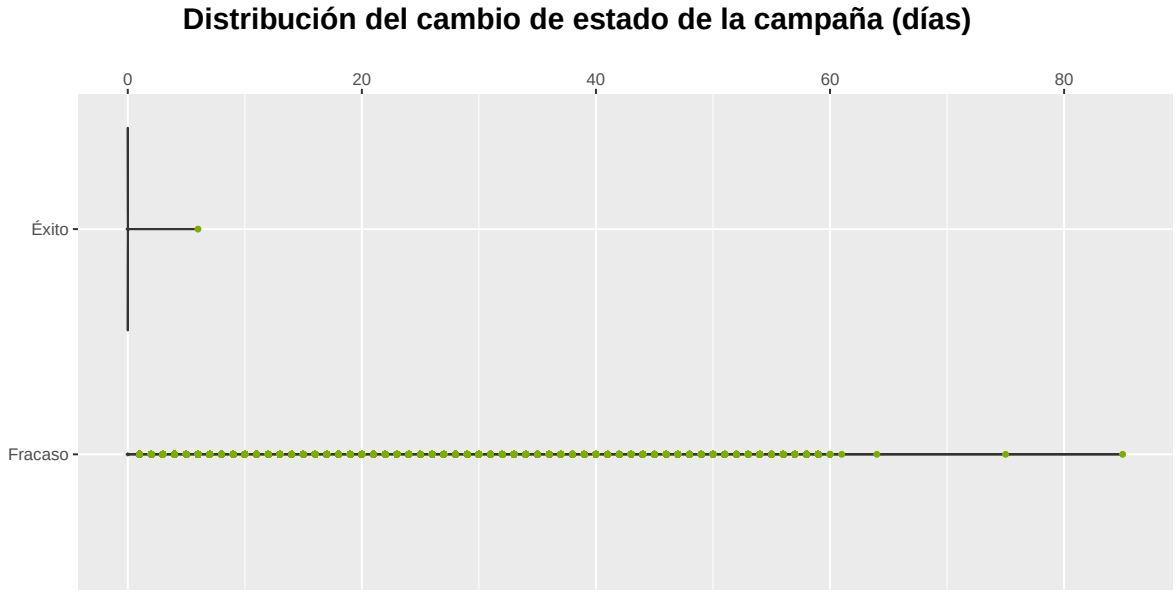
```
ggplot(data=clean.data, mapping=aes(x=success, y=prep.period)) +
  geom_violin(fill=default.color.main, adjust=4) +
  xlab('') + ylab('') +
  scale_y_continuous(position='right') +
  scale_x_discrete(labels=function(label) ifelse(label, 'Éxito', 'Fracaso')) +
  geom_boxplot(width=0.01, outlier.color=default.color.terciary,
               fill=default.color.secondary, outlier.size=1.05) +
  coord_flip() +
  ggtitle('Distribución del periodo de preparación (días)') +
  title.centered
```



```
ggplot(data=clean.data, mapping=aes(x=success, y=collection.period)) +
  geom_violin(fill=default.color.main, adjust=4) +
  xlab('') + ylab('') +
  scale_y_continuous(position='right') +
  scale_x_discrete(labels=function(label) ifelse(label, 'Éxito', 'Fracaso')) +
  geom_boxplot(width=0.01, outlier.color=default.color.terciary,
               fill=default.color.secondary, outlier.size=1.05) +
  coord_flip() +
  ggtitle('Distribución del periodo de recaudación (días)') +
  title.centered
```




```
ggplot(data=clean.data, mapping=aes(x=success, y=finish.period)) +
  geom_violin(fill=default.color.main, adjust=4) +
  xlab('') + ylab('') +
  scale_y_continuous(position='right') +
  scale_x_discrete(labels=function(label) ifelse(label, 'Éxito', 'Fracaso')) +
  geom_boxplot(width=0.01, outlier.color=default.color.terciary,
               fill=default.color.secondary, outlier.size=1.05) +
  coord_flip() +
  ggtitle('Distribución del cambio de estado de la campaña (días)') +
  title.centered
```



Tanto proyectos exitosos como fallidos invierten cantidades de tiempo similares para preparar sus campañas de financiación. Esto sugiere que existe un periodo de preparación antes de la creación del proyecto en la plataforma, obviamente no contemplado por la misma. Otra interpretación es que no se invierte en preparar la campaña.

En cualquier caso, una distribución tan similar en casos de éxito y fracaso no proporciona mucha información analizable a simple vista de las causas de dicho éxito o fracaso.

En cuanto al período de recaudación, los casos fallidos de financiación parecen alargarse más en el tiempo, llegando a 60 días con más facilidad que los casos de éxito, que se ciñen de media a 30 días.

Respecto a la distribución del cambio de estado de la campaña, se observa que los proyectos de éxito suelen esperar a la *deadline* para terminar el periodo de recaudación, mientras que los proyectos con financiación fracasada suelen cerrar el periodo de recaudación antes (cancelando la campaña).

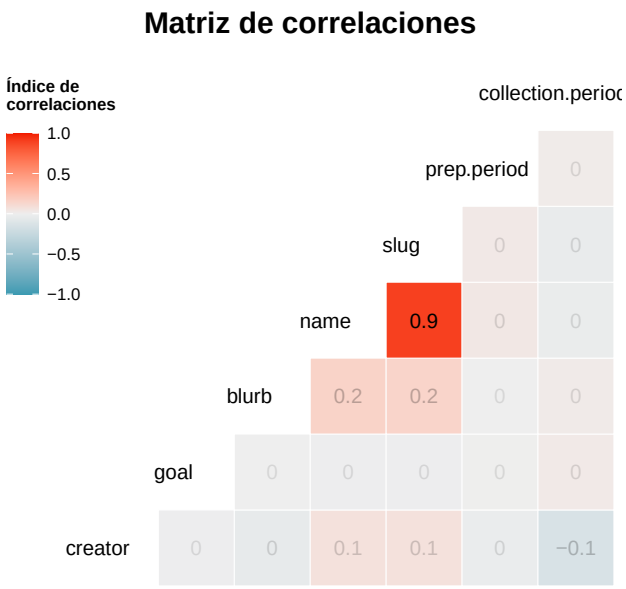
9.1 Correlaciones entre variables numéricas

A continuación, se procede a explorar si existen correlaciones entre las variables numéricas que se van a utilizar en el modelo de clasificación.

```
corr.cols <- c('creator', 'goal', 'blurb', 'name', 'slug', 'prep.period',
               'collection.period')
```

Patricia Lázaro Tello
plazarotello@uoc.edu

```
ggcorr(data=clean.data[, corr.cols], label=TRUE, label_alpha=TRUE,
       nudge_x=-0.3, layout.exp=1, legend.position=c(0.1, 0.75),
       name='Índice de\ncorrelaciones\n') +
  theme(legend.title=element_text(face='bold', vjust=-1.0)) +
  ggtitle('Matriz de correlaciones') + title.centered
```



Como era de esperar, el número de palabras en el nombre y en el *slug* están muy correlados: esto se debe a que el *slug* no es más que el nombre del proyecto formateado para su uso en URLs. Se procede a eliminar el *slug* del conjunto de datos.

```
clean.data$slug <- NULL
clean.data.dim <- dim(clean.data)
```

En el resto de variables analizadas no se aprecian correlaciones significativas.

10 Exploración del conjunto de datos

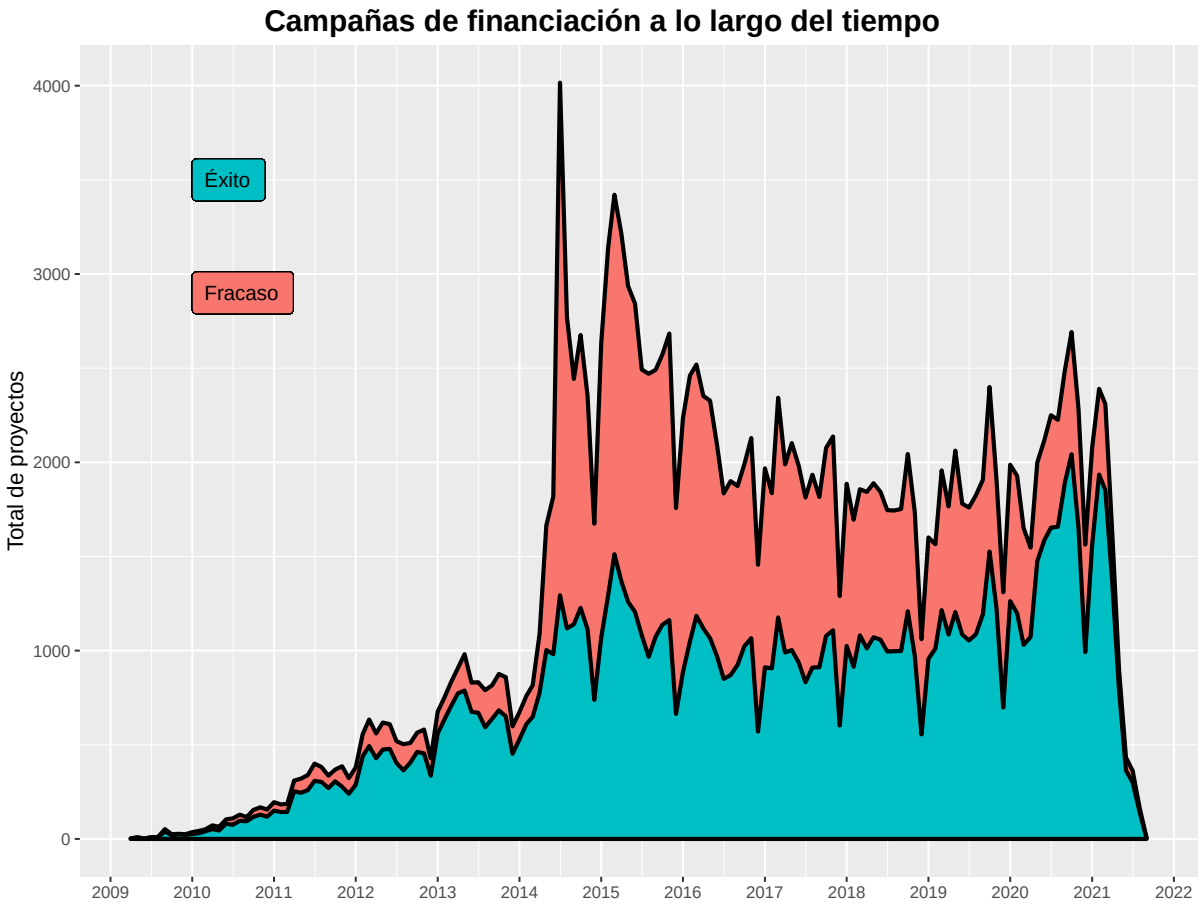
Se procede a visualizar los datos de acuerdo a varios atributos simultáneamente con la intención de obtener más información acerca de los casos de éxito y fracaso en las campañas de financiación.

10.1 Campañas lanzadas

Se desea analizar si el número de campañas de financiación lanzadas cada día afecta al éxito de las mismas. A priori se puede argumentar que cuantas más campañas estén activas a la vez en **Kickstarter**, existirá más competencia por la atención del patrocinador.

```
launched.campaigns <- clean.data %>% dplyr::select(success, launched_at)
launched.campaigns$launched_at <- as.POSIXct(trunc(
  launched.campaigns$launched_at, 'months'))

launched.campaigns <-
  launched.campaigns[order(launched.campaigns$launched_at),] %>%
  group_by(launched_at, success) %>%
  add_count(launched_at, success, name='projects') %>%
  distinct(launched_at, success, projects) %>%
  group_by(launched_at) %>%
  mutate(total=sum(projects)) %>%
  filter(success==TRUE) %>%
  group_by(launched_at)

ggplot(data=launched.campaigns, mapping=aes(x=launched_at)) +
  geom_ribbon(aes(ymin=projects, ymax=total), fill=default.color.main,
              outline.type='upper', color='black', size=1.05) +
  geom_ribbon(aes(ymin=0, ymax=projects), fill=default.color.secondary,
              outline.type='both', color='black', size=1.05) +
  geom_label(x=as.POSIXct('2010-01-01'), y=3500, label='Éxito', color='black',
             fill=default.color.secondary, hjust=0.0,
             label.padding = unit(0.5, 'lines')) +
  geom_label(x=as.POSIXct('2010-01-01'), y=2900, label='Fracaso',
             color='black', hjust=0.0, fill=default.color.main,
             label.padding = unit(0.5, 'lines')) +
  ggtitle('Campañas de financiación a lo largo del tiempo') + title.centered +
  xlab('') + ylab('Total de proyectos') +
```

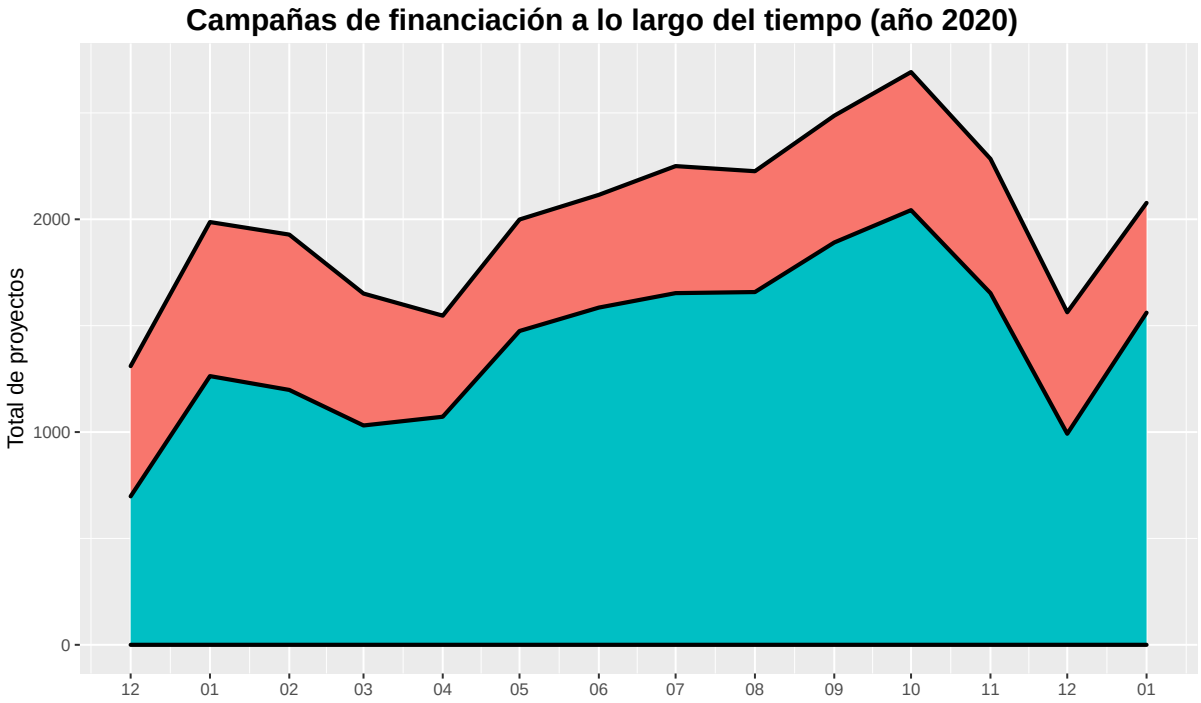
```
  scale_x_datetime(date_breaks = "1 year", date_labels = "%Y")
```



Según se observa en la gráfica, a partir de cierto umbral, cuantas más campañas se lanzan al día peores resultados producen. También se aprecia un patrón repetido en anualmente; se procede a analizar únicamente los datos de 2020 (hasta enero de 2021 incluido) para obtener más información del patrón.

```
campaigns.2020 <- launched.campaigns %>%
  filter(launched_at >= as.POSIXct('2019-12-01') &
           launched_at <= as.POSIXct('2021-01-01'))
ggplot(data=campaigns.2020, mapping=aes(x=launched_at)) +
  geom_ribbon(aes(ymin=projects, ymax=total), fill=default.color.main,
              outline.type='upper', color='black', size=1.05) +
  geom_ribbon(aes(ymin=0, ymax=projects), fill=default.color.secondary,
              outline.type='both', color='black', size=1.05) +
  geom_label(x=as.POSIXct('2010-01-01'), y=3500, label='Éxito', color='black',
```

```
            fill=default.color.secondary, hjust=0.0,
            label.padding = unit(0.5, 'lines')) +
  geom_label(x=as.POSIXct('2010-01-01'), y=2900, label='Fracaso',
            color='black', hjust=0.0, fill=default.color.main,
            label.padding = unit(0.5, 'lines')) +
  ggtitle('Campañas de financiación a lo largo del tiempo (año 2020)') +
  title.centered + xlab('') + ylab('Total de proyectos') +
  scale_x_datetime(date_breaks = "1 month", date_labels = "%m")
```



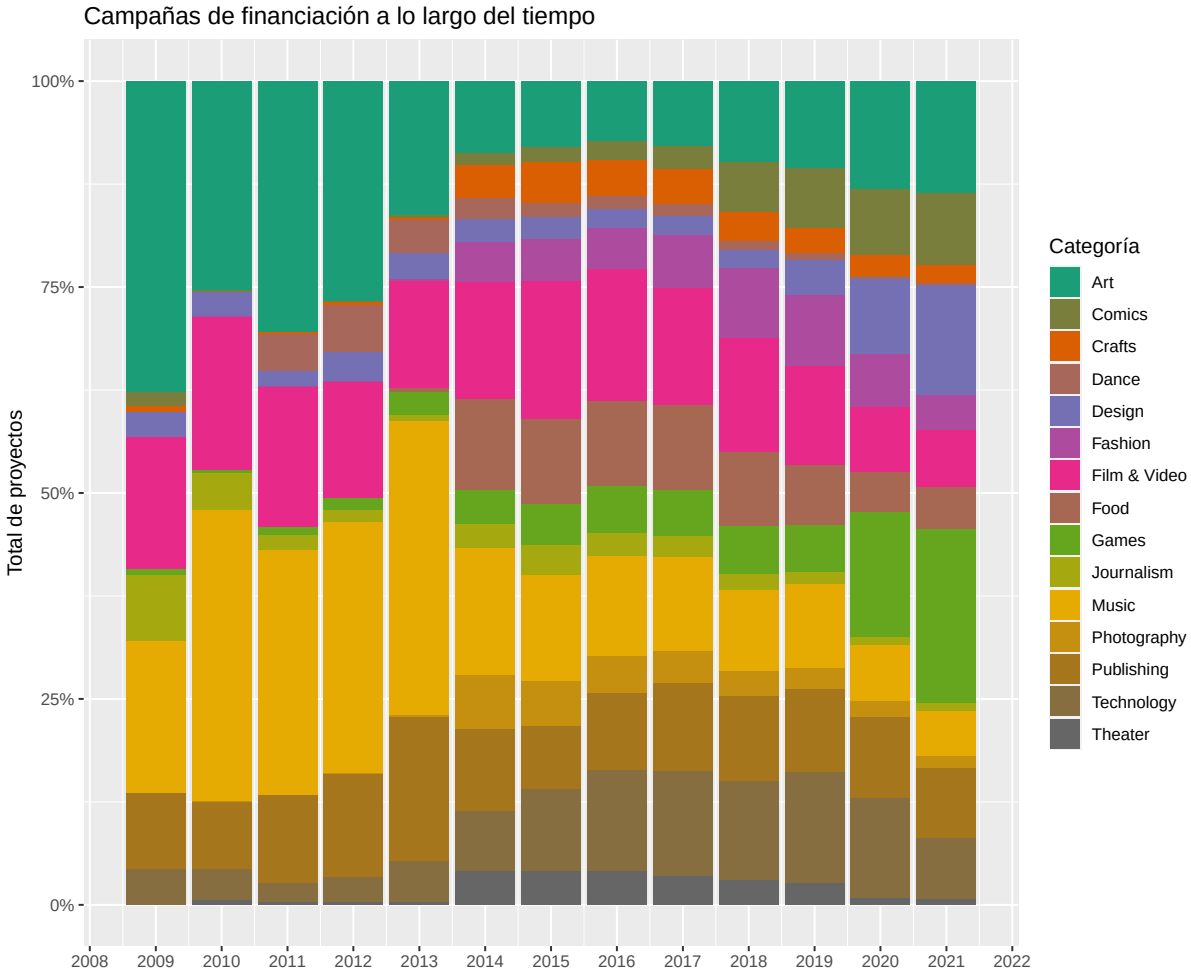
Se observa que diciembre es el mes en que menos proyectos comienzan su campaña de financiación. Tiene sentido ya que coincide con el día de acción de gracias y *black friday* al inicio del mes y con la navidad al final, eventos en los que se gasta mucho dinero en regalos y compras.

También se ve que los meses de septiembre y octubre son los más populares para lanzar campañas de financiación, coincidiendo con el final del verano y el comienzo de las clases para los niños.

10.2 Popularidad de las categorías a lo largo del tiempo

A continuación se propone analizar la cantidad de proyectos que se han creado para cada categoría a lo largo de los años. Así, se observa cómo ha variado la popularidad de cada categoría con el tiempo.

```
popular.categories <- clean.data %>% dplyr::select(category, launched_at)
popular.categories$launched_at <- as.POSIXct(trunc(
  popular.categories$launched_at, 'years'))

popular.categories <-
  popular.categories[order(popular.categories$launched_at),] %>%
  group_by(launched_at, category) %>%
  add_count(launched_at, category, name='projects') %>%
  distinct(launched_at, category, projects) %>%
  group_by(launched_at) %>%
  mutate(total=sum(projects)) %>%
  mutate(pctg=projects/total)

cat.count <- length(unique(popular.categories$category))
categories.palette <- colorRampPalette(brewer.pal(8, "Dark2"))(cat.count)

ggplot(data=popular.categories, mapping=aes(x=launched_at, y=pctg,
                                            fill=category)) +
  geom_bar(position='fill', stat='identity') +
  scale_fill_manual(values=categories.palette) +
  ggtitle('Campañas de financiación a lo largo del tiempo') +
  xlab('') + ylab('Total de proyectos') +
  scale_x_datetime(date_breaks = "1 year", date_labels = "%Y") +
  scale_y_continuous(position='left', labels=percent) +
  labs(fill='Categoría') + guides(fill=guide_legend(ncol=1, byrow=FALSE))
```



Analizando la gráfica anterior, se obseva que los proyectos de juegos y diseño son los que más están creciendo en los últimos años; la categoría de arte también está viendo un repunte en su popularidad después de llegar a su puntuación más baja en 2016.

Los proyectos de música sobre todo, y en menor medida los proyectos de películas y vídeos, están en delive después de haber obtenido buenas puntuaciones durante los primeros años recogidos en el *dataset*.

11 Construcción del conjunto de datos limpio

Para la creación del conjunto de datos listo para entrenar un modelo de clasificación — ya sea una regresión logística, un árbol de decisión, un algoritmo k-NN o una red neuronal — es necesario proceder de la siguiente manera:

1. Seleccionar las características que formarán parte del conjunto de datos. Han de ser solo datos que se puedan obtener antes de lanzar la campaña (por ejemplo, el dinero recaudado no entra en el conjunto de datos).

2. Codificar las variables no numéricas.

- Como en un paso posterior se realizará `PCAMix`, no se va a aplicar *one-hot encoding* sobre las variables categóricas (que produciría una explosión de dimensionalidad (*curse of dimensionality*)).

3. Normalizar las variables al rango $[0.0, 1.0]$.

- Este paso no es necesario para un árbol de decisión, pero la normalización no afecta al resultado que se obtiene.
- No es necesario aplicar normalización a las variables categóricas codificadas porque ya se encuentran en el rango deseado.

4. Si hubiera demasiadas variables, se aplicaría PCA para reducir la dimensionalidad.

11.1 Selección de características

```
final.dataset <- clean.data %>%
  select(name, blurb, photo, creator, category,
         collection.period, launched_at, goal, country, success) %>%
  rename(launch.month = launched_at, words.name = name, words.blurb = blurb,
         created.projects = creator, photo.exists = photo) %>%
  mutate(launch.month = format(launch.month, format='%B'))

final.dataset.dim <- dim(final.dataset)
```

Se ha elegido incluir el mes de lanzamiento de la campaña de financiación en el *dataset* final debido al patrón encontrado al analizar los casos de éxito y fracaso a lo largo del tiempo.

Se ha incluido además el objetivo de financiación (`goal`) y el periodo (días) en que la recaudación estará disponible (`collection.period`).

11.2 Codificación de variables

Para las variables lógicas, se aplicará $TRUE = 1$ y $FALSE = 0$.

```
final.dataset <- final.dataset %>%
  mutate(photo.exists=ifelse(photo.exists, 1, 0),
         success=ifelse(success, 1, 0))
```

11.3 Normalización de variables numéricas

A continuación se procede a normalizar las variables numéricas `words.name`, `words.blurb`, `created.projects`, `goal` y `collection.period` para que se encuentren en el rango deseado $[0.0, 1.0]$. Se utilizará la **normalización por la diferencia**:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

```
diff.normalization <- function(x){
  x.min <- min(x); x.max <- max(x)
  return((x - x.min)/(x.max - x.min))
}
```

```
final.dataset$words.name <- diff.normalization(final.dataset$words.name)
final.dataset$words.blurb <- diff.normalization(final.dataset$words.blurb)
final.dataset$created.projects <-
  diff.normalization(final.dataset$created.projects)
final.dataset$collection.period <-
  diff.normalization(final.dataset$collection.period)
final.dataset$goal <- diff.normalization(final.dataset$goal)
```

11.4 Reducción de dimensionalidad

El siguiente paso para conseguir un conjunto de datos aceptable para entrenar un modelo de clasificación es reducir la dimensionalidad. Como se tienen variables categóricas, se va a utilizar `PCAmix`, que sirve para reducir la dimensionalidad utilizando variables numéricas y categóricas al mismo tiempo.

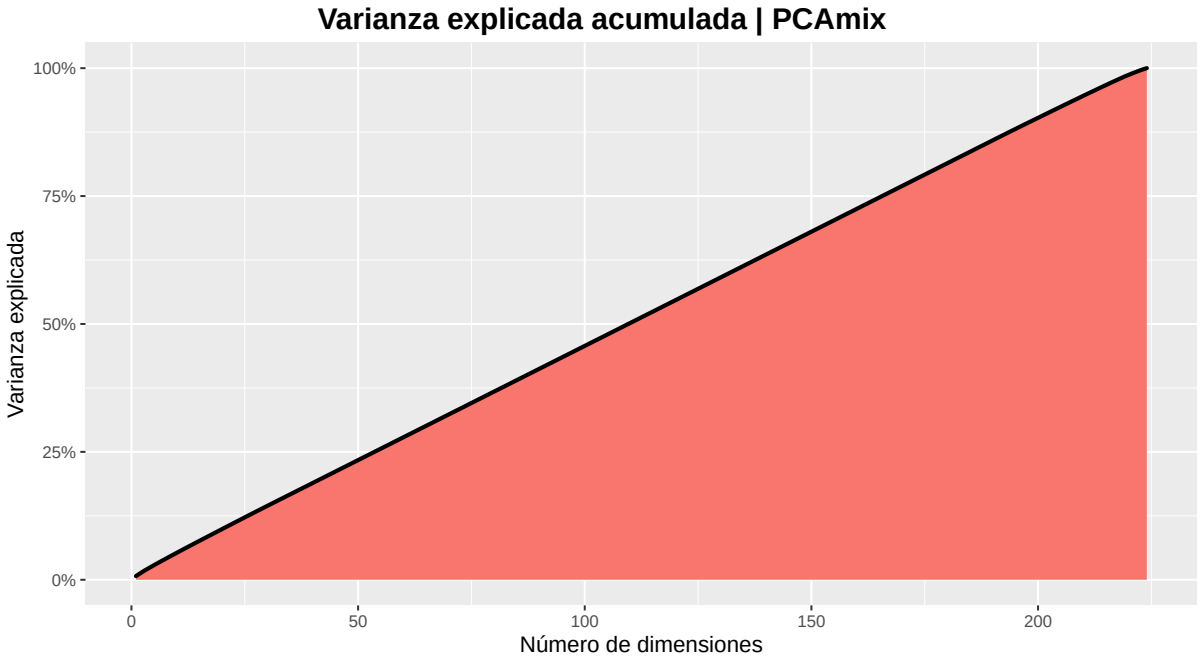
```
ds.split <- splitmix(final.dataset)
ds.pcamix <- PCAmix(X.quanti=ds.split$X.quanti, X.quali=ds.split$X.quali,
                    rename.level=TRUE, graph=FALSE)
tail(ds.pcamix$eig, n=5)
```

```
##         Eigenvalue Proportion Cumulative
## dim 220       0.83       0.37         99
## dim 221       0.81       0.36         99
## dim 222       0.75       0.33         99
## dim 223       0.73       0.33        100
## dim 224       0.63       0.28        100
```

Como era de esperar, la aplicación de PCA mixto — es decir, PCA para variables numéricas y MCA (*Multiple Correspondence Analysis*) sobre las variables categóricas — ha resultado en una explosión de dimensionalidad. Esto se debe a que MCA ha llevado a cabo *one-hot encoding*; en el caso de este *dataset*, la variable categoría explota en 15 variables, una por cada categoría, por ejemplo.

Se obtienen 224 dimensiones que explican el 100% de la varianza. Como son demasiadas dimensiones, se decide cortar por la primera dimensión cuya varianza acumulada explique el 75% o más de la varianza.

```
ds.pcamix.eigen <- as.data.frame(ds.pcamix$eig)

ggplot(data=ds.pcamix.eigen, mapping=aes(x=1:dim(ds.pcamix.eigen)[1],
                                         y=Cumulative)) +
  geom_area(fill=default.color.main, color='black', size=1.05) +
  scale_y_continuous(labels=scales::percent_format(scale = 1)) +
  title.centered + xlab('Número de dimensiones') +
  ylab('Varianza explicada') +
  ggtitle('Varianza explicada acumulada | PCAmix')
```




```
ds.pcamix.eigen <- ds.pcamix.eigen %>%
  filter(round(Cumulative) <= 75)
tail(ds.pcamix.eigen)
```

```
##         Eigenvalue Proportion Cumulative
## dim 161          1       0.45         73
## dim 162          1       0.45         73
## dim 163          1       0.45         74
## dim 164          1       0.45         74
## dim 165          1       0.45         75
## dim 166          1       0.45         75
```

Se decide quedarse con 166 dimensiones. Se aplica `PCAmix` otra vez, especificando el número de dimensiones a guardar:

```
ds.pcamix <- PCAmix(X.quanti=ds.split$X.quanti, X.quali=ds.split$X.quali,
                    rename.level=TRUE, graph=FALSE,
                    ndim=dim(ds.pcamix.eigen)[1])

pca.components <- data.frame(ds.pcamix$ind$coord)

pca.dims <- colnames(pca.components)[which(startsWith(
  colnames(pca.components), 'dim.'))]

col.normalization <- function(df, column){
```

```
  return(diff.normalization(df[,column]))
}

pca.dataset <- final.dataset %>%
  dplyr::select(success) %>%
  bind_cols(data.frame(sapply(pca.dims, col.normalization, df=pca.components)))
pca.dataset.dim <- dim(pca.dataset)
```

Para el árbol de decisión expandir las variables categóricas o no no supone ningún cambio a la hora de entrenar el modelo ni en el resultado del entrenamiento; por tanto se decide guardar el *dataset* completo con las variables sin expandir ni reducir su dimensionalidad, así como el conjunto de datos expandido y reducido.

```
write.csv(final.dataset, 'kickstarter-clean-compact.csv', row.names=FALSE,
          fileEncoding='UTF-8')
write.csv(pca.dataset, 'kickstarter-clean-expanded.csv', row.names=FALSE,
          fileEncoding='UTF-8')
saveRDS(object=ds.pcamix, file='kickstarter-pcamix.rds')
```

Se han guardado un total de 203.094 registros en 2 ficheros: el fichero compacto tiene 10 atributos, mientras que el fichero expandido tiene 167 variables.

A continuación se presentan algunas estadísticas interesantes de la volumetría del *dataset*:

Fase	Observaciones	Atributos
raw	1.186.347	39
criba inicial	210.319	20
criba avanzada	203.094	21
limpieza	203.094	23
final (compacto)	203.094	10
final(expandido)	203.094	167

El *dataset* original y los *datasets* finales (compacto y expandido), así como el objeto `PCAmix` necesario para transformar nuevas observaciones a la entrada de los algoritmos de clasificación, se encuentran en este kaggle.

Patricia Lázaro Tello
plazarotello@uoc.edu

Bibliografía

Jobs Admin. 2017. «Creating & Visualizing Neural Network in R». https://www.analyticsvidhya.com/blog/2017/09/creating-visualizing-neural-network-in-r/.

Kickstarter. 2009. «Kickstarter». https://www.kickstarter.com.

Navlani, Avinash. 2019. «Neural Network Models in R». https://www.datacamp.com/community/tutorials/neural-network-models-r.

Web Robots. 2021. «Kickstarter Datasets». https://webrobots.io/kickstarter-datasets/.

Wikipedia. 2021a. «Fowlkes-Mallows index». https://en.wikipedia.org/wiki/Fowlkes%E2%80%93Mallows_index.

———. 2021b. «Jaccard index». https://en.wikipedia.org/wiki/Jaccard_index.

———. 2021c. «F-score». https://en.wikipedia.org/wiki/F-score#F%CE%B2.

Yudha Wijaya, Cornellius. 2020. «5 Must-Know Dimensionality Reduction Techniques via Prince». https://towardsdatascience.com/5-must-know-dimensionality-reduction-techniques-via-prince-e6ffb27e55d1.