UNIVERSITAT DE BARCELONA

TRABAJO FIN DE GRADO

Análisis de sentimiento de Robos y Seguridad en Twitter

Realizado por
Marta Ortiz Fernández

Para la obtención del título de
Grado en Ingeniería Informática

Dirigido por
Dr. Òscar Amoròs Hughet

Convocatoria de Junio, curso 2020/21

Agradecimientos

Quiero agradecer a Ana por apoyarme durante el trabajo final de grado, a sabiendas que pese a mis dudas a lo largo del proyecto, no ha dejado que me rinda y lo dé por imposible.

Quiero agradecer a mi abuela Soledad por las constantes llamadas para verificar como iba el proyecto. Y a mi abuela Pilar por el apoyo ofrecido.

También quiero agradecer a Simone y a Elvira por su idea. Porque de lo contrario no me hubiera decidido a hacer ningún proyecto. Y este final sería inexistente.

Resumen

Este proyecto busca desarrollar la manera de reagrupar de forma óptima los efectivos del cuerpo de seguridad y de policía en toda el área de España. Para ello se tiene en cuenta el sentimiento que posee la comunidad acerca de la seguridad. Este sentimiento se recoge en forma de comentarios hechos en la red social de Twitter. También se han utilizado los datos ofrecidos por el Ministerio de España (concretamente el portal estadístico de criminalidad) sobre los delitos de patrimonio donde se incluyen robos, hurtos y estafas para contrastarlos con el sentimiento de seguridad obtenido, a fin de poder evaluar la sensación de la comunidad en relación al número de casos ocurridos en el territorio.

Palabras clave: Cuerpo Policial, Robos, Seguridad, España, Twitter

Abstract

This proyect seeks to develop the way to optimally regroup the personnel of the security force and police throughout the area of Spain. This proyect takes into account the community's feeling about safety. The sentiment is collected in the form of comments made on the social network twitter. The data provided by the Ministry of Spain (specifically the statistical portal of criminalistic) on property crimes including theft and scams to contrast them with the feeling of the security obtained: in order to be able to assess the feeling of the community in relation to the number of cases that occurred in the territory.

Keywords: Police, Theft, Safety, Spain, Twitter

Índice general

Índice de cuadros

Índice de figuras

Índice de extractos de código

1. Introducción

1.1. Justificación

Este proyecto quiere conseguir que las personas se sientan seguras y dejen de lado ese sentimiento de inseguridad que han ido acumulando a raíz de este último año de pandemia. El enfoque está destinado a reubicar los agentes del Cuerpo Policial para llegar a optimizar el sistema ejecutivo. Para ello se va a emplear el uso de la red social de Twitter dando peso al sentimiento encontrado en los comentarios referentes a los robos y la seguridad de España.

1.2. Antecedentes

Los antecedentes parten de los datos que ofrece el portal estadístico de Criminalidad del Ministerio del Interior de España [2]. Se han recogido en dos cuadros 1.1 y 1.2 con los delitos anuales cometidos sobre el patrimonio. Estos datos van desde el año 2010 al año 2019.

Los delitos de patrimonio se agrupan de la siguiente manera:

1. Hurtos
2. Robos con fuerza. De los cuales tenemos:
   a) Robos con fuerza en el interior de vehículos
   b) Robos con fuerza en viviendas
   c) Robos con fuerza en establecimientos
3. Robos con violencia o intimidación. De los cuales tenemos:
   a) Robos con violencia en la vía pública
   b) Robos con violencia en viviendas
   c) Robos con violencia en establecimientos
4. Sustracción de vehículos
5. Estafas. Pudiendo hallar en estas también las estafas bancarias.
6. Daños
7. Daños contra la propiedad intelectual o industrial
8. Blanqueo de capitales
9. Otros englobados en el apartado del patrimonio.

En referente a los cuadros 1.1 y 1.2 que se listan a continuación se ha enumerado el texto recortado de la primera columna del cuadro de la siguiente manera:

1. con fuerza en las cosas
2. con violencia o intimidación
3. PI : propiedad intelectual/industrial
4. contra el patrimonio

Total Nacional	2010	2011	2012	2013	2014
Patrimonio	1.779.019	1.742.631	1.754.632	1.679.585	1.595.984
Hurtos	785.635	786.704	790.281	770.296	727.800
Robos (1)	443.772	414.961	405.939	381.777	344.875
Robos (2)	84.411	87.718	96.607	86.034	70.855
Sustracción de vehículos	65.948	60.061	55.197	48.855	43.206
Estafas	102.567	106.262	124.647	122.464	140.418
Daños	266.291	254.361	246.355	226.619	218.166
Contra la PI (3)	3.619	2.643	2.884	3.260	2.608
Blanqueo de capitales	182	171	199	243	230
Otros (4)	26.59	29.750	32.523	40.037	47.826

Cuadro 1.1: Tabla de la serie anual de los delitos de patrimonio en hechos conocidos ocurridos desde año 2010 al 2014. Fuente: Portal Estadístico de Criminalidad [2]

Total Nacional	2015	2016	2017	2018	2019
Patrimonio	1.573.983	1.572.967	1.593.930	1.664.242	1.707.144
Hurtos	715.469	711.507	712.398	706.072	700.453
Robos (1)	322.705	318.164	301.734	302.043	298.098
Robos (2)	64.581	62.952	61.763	60.295	65.874
Sustracción de vehículos	43.170	43.335	42.519	35.897	35.105
Estafas	165.267	179.718	214.595	289.182	327.616
Daños	215.519	214.709	214.246	213.815	219.424
Contra la PI (3)	2.047	1.889	1.622	2.205	1.807
Blanqueo de capitales	290	262	260	272	295
Otros (4)	44.935	40.431	44.793	54.461	58.472

Cuadro 1.2: Tabla de la serie anual de los delitos de patrimonio en hechos conocidos ocurridos desde año 2015 al 2019. Fuente: Portal Estadístico de Criminalidad [2]

Se puede observar, si se toma en cuenta el número de casos sobre el patrimonio, que ha habido un aumento de manera significativa desde el año 2016 al año 2019. Además el número total de casos recogidos en el año 2019 se asemeja al número total de casos recogidos en el año 2012. Esta semejanza ofrece la sensación de que hemos retrocedido en el tiempo, dejando que la delincuencia crezca. Ese total de **1.707.144** casos recogidos por toda España en el año 2019 se visualiza en la imagen 1.1.

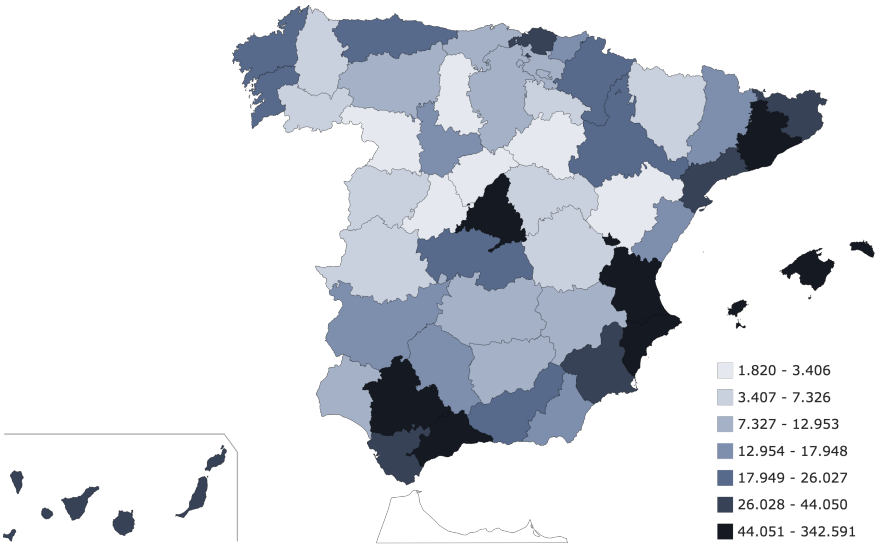

Figura 1.1: Mapa de los delitos de patrimonio en hechos conocidos ocurridos en el año 2019. Fuente: Portal Estadístico de Criminalidad [2]

1.3. Objetivos

Los objetivos se dividen en dos grupos: objetivo general y objetivos específicos.

1.3.1. Objetivo general

- Analizar el sentimiento de la comunidad sobre la seguridad y los robos acotando como zona de estudio el área de España para mejorar la asignación de los efectivos de las fuerzas y cuerpos del estado.

1.3.2. Objetivos específicos

- Recoger de Twitter los comentarios relacionados con los robos, la seguridad y la delincuencia.
- Acotar la búsqueda a la región de España, englobando Canarias, Ceuta y Melilla.
- Analizar el texto de un tweet (280 caracteres) para descifrar el sentimiento de la comunidad sobre la seguridad.
- Mostrar los datos recogidos de los robos en el área de España acotada por su ciudad y número de tweets referidos a esa localización.
- Optimizar el programa para que con otros parámetros de búsqueda y selección de nombre de ficheros, pueda ejecutarse.

1.4. Método

Se va a llevar a cabo un trabajo de análisis sobre los comentarios en Twitter. Este trabajo tiene los siguientes pasos a seguir:

- Recoger los datos de la aplicación.
- Guardar los datos en un fichero.
- Analizar el sentimiento del texto.
- Mostrar los resultados del análisis obtenido en un mapa de España.
- Verificar si tiene peso el sentimiento obtenido con el número de robos cometidos en esa zona del mapa.

2. Propuesta

2.1. Introducción

En este apartado vamos a explicar:

- Las características sobre los usuarios, es decir, los participantes encontrados en Twitter.
- Las librerías que utilizamos a lo largo del proyecto.
- Las decisiones a tomar con respecto a dos variables del dataframe recogido de la API de Twitter.
- Los requerimientos a seguir durante la planificación.

2.1.1. Participantes

Los participantes son anónimos, señalados con un nick que encontramos en la variable *screen_name*, de escritura española y con una localización escrita por ellos mismos en la variable *location*.

2.1.2. Instrumentos

Los tweets han sido recogidos en lapsos de tres días mediante la función `search_tweets2` encontrada en la librería `rtweet` [5]. Con un máximo de 18.000 registros por búsqueda en cada una de las temáticas propuestas (robos, seguridad y delincuencia). Se han hecho dos búsquedas tomando como centro respectivamente, las coordenadas de la ciudad de Madrid y de Santa Cruz de Tenerife con un radio de 750 y 300 kilómetros para englobar la zona definida como España (Figura 2.1).

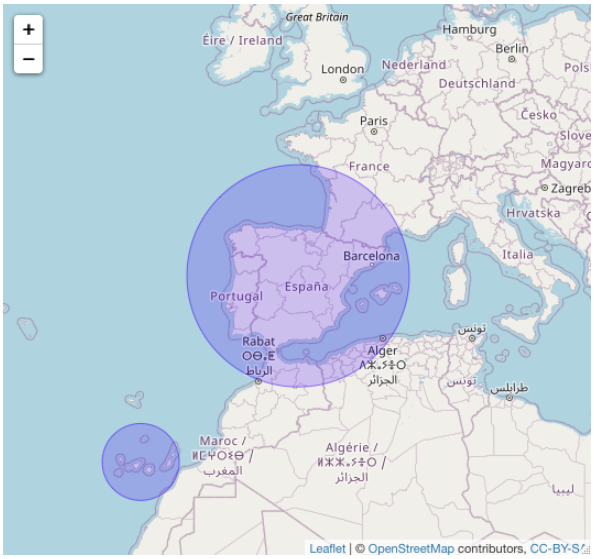

Figura 2.1: Mapa del área acotada de España

El texto de cada tweet se ha alojado en un corpus de la librería `quanteda` [1]. Se emplea el método **quanteda::corpus** que crea el corpus a partir del dataframe con todos los datos recogidos de los comentarios de Twitter.

La limpieza del texto se hace a través de la función **clean_and_generate_token** 2.1.2 del fichero `TextTwitter`. Se emplea la función **as.character** para recoger los textos del corpus.

```
clean_and_generate_tokens <- function(corpus){
  remove_stopwords <- c(stopwords("spanish")[stopwords("spanish")
     != "no"], "etc", "q", "i", "d") #I+D

  toks <- as.character(corpus) %>%
    char_tolower() %>%
    str_replace_all("\\(.*\\)","'") %>%
    tokens(remove_url = TRUE, remove_punct = TRUE, remove_symbols
       = TRUE, remove_numbers = TRUE, split_hyphens = TRUE) %>%
    tokens_remove(pattern = "#\\w+|@\\w+|[^[:alpha:]
       ]+|[a|e|i|o|u]{3,30}", valuetype = "regex") %>%
    tokens_remove(remove_stopwords) %>%
    tokens_wordstem()

  return(toks)
```

Extracto de código 2.1: Extracto de código de la función clean_and_generate_token

Este texto pasa a minúscula con el método **char_tolower**. Se eliminan aquellos comentarios que se hacen entre paréntesis con regex al usar la función **str_replace_all**. La función **tokens** hace la separación por palabras y elimina además los enlaces, la puntuación, los símbolos, los números y los guiones. Empleamos la función **tokens_remove** utilizando de nuevo regex para descartar etiquetas y menciones, los caracteres que no sean alfanuméricos y repeticiones de más de tres dígitos en las vocales. Nuevamente pasamos la función **tokens_remove** para eliminar las *stopwords* (definidas como palabras

vacías es el nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones exceptuando el término no) y aplicamos **tokens_wordstem** para hacer *stemming* (reduciendo todas las conjugaciones a la raíz) de las palabras.

Para hacer el análisis de sentimiento se utiliza el diccionario definido por Rachael Tatman [7]. Este diccionario hace referencia a una lista de palabras donde por cada palabra hay un tipo de sentimiento definido como positivo o negativo. Con el método **quanteda.sentiment::textstat_polarity** se pasa la matriz de características calculada al pasar el texto tokenizado por parámetro del método **dfm**, el diccionario anteriormente mencionado y la función a aplicar que por defecto es:

$$sentiment = \frac{log(positive)}{log(negative)}$$

Para hacer las visualizaciones se emplea la librería `ggplot` [10] comprendida en la colección `tidyverse` para los gráficos y la librería `leaflet` [3] para la visualización de mapas.

2.1.3. Análisis estadístico (o análisis de datos)

Se acotan los datos por lenguaje en español y por la variable *location* en lugar de *place_name* en la función 4.3.1 debido a los datos observados en las tablas 2.1 y 2.2.

Lenguaje	número de registros
Español(es)	5542
Indefinido(und)	30
Portugués(pt)	10
Catalán(ca)	9
Inglés(en)	4
Checo(cs)	1
Talago(tl)	1

Cuadro 2.1: Tabla de los registros por localización(*location*)

Lenguaje	número de registros
Español(es)	54
Indefinido(und)	3
Catalán(ca)	1

Cuadro 2.2: Tabla de los registros por localización(*place_name*)

Como se puede observar rápidamente, funciona mejor la variable *location* que la variable *place_name*. Además se toma únicamente los valores del lenguaje español para hacer el análisis, porque el porcentaje de registros de los otros lenguajes es del 1 % en contraposición al 99 % que constituye el lenguaje español.

$$Porcentaje = \frac{otros\ lenguajes}{total} = \frac{55}{5597} = 0{,}009826$$

2.2. Planificación

Definimos como estructura de planificación los siguientes puntos:

- Primeros pasos
- Base de Datos
- Código
- Memoria

2.2.1. Primeros pasos

En referente a los primeros pasos a seguir lo que necesitamos es instalar RCode y RStudio. Concretamente se ha utilizado la versión 1.1.463 de RStudio y la versión 3.6.3 de R. También se hace la planificación del proyecto, aduciendo la necesidad de enfocarnos primero en la recogida de información para proceder a continuación con la búsqueda del sentimiento en el código.

2.2.2. Base de Datos

Para la base de datos recogida en los ficheros de los directorios `databdd` y `databddout` es necesaria la autenticación de la API de Twitter recogida en el método **getToken** del fichero `SetPackages`.

```
twitter_token <- rtweet::rtweet_bot(api_key = "xxxx", api_secret
   = "xxxx", access_token = "xxxx", access_secret = "xxxx")

getToken <- function(){ return(twitter_token) }
```

Extracto de código 2.2: Extracto de código del fichero SetPackages.R

Se consigue entrando en el portal de desarrolladores [4]. Al acceder a la cuenta de Twitter te permite crear una aplicación donde se pide rellenar los siguientes campos:

- Nombre de la aplicación
- Descripción de la aplicación
- Enlace de la aplicación (si lo hay)
- Uso que se le quiere dar a la aplicación

Una vez rellenados los campos requeridos se podrá pedir las credenciales necesarias (claves y tokens) para recoger los datos de Twitter de la librería `rtweet`.

Esta base de datos se recoge a partir de dos llamadas al método **initBDD** del fichero Main. Cada tres días permite guardar un nuevo fichero `.rds` en el directorio `databdd` con la información referente a las etiquetas recogida.

```
RDS_dir <- "databdd"
coord <- c("40.36329,-3.69141,750km",
   "28.469648,-16.2540884,300km")
path_out_robos = "databddout/all_robos_unique.rds"
path_out_seguridad = "databddout/all_seguridad_unique.rds"
# Inicializamos la bdd de robos
initBDD(RDS_dir, coord, tags = c("#Robo","Robos"),
   pattern_file_name = "_robos_esp", path_out_robos)

# Inicializamos la bdd de seguridad
initBDD(RDS_dir, coord, tags = c("#Seguridad",
   "#Proteccion","#Delincuencia", "delitos"), pattern_file_name
   = "_seguridad_esp", path_out_seguridad)
```

Extracto de código 2.3: Extracto de código del fichero Main.R

2.2.3. Código

Para el código, como ya hemos pincelado anteriormente en los primeros pasos, emplearemos una versión 1.1.463 de RStudio y una versión 3.6.3 de R. ePara la visualización del mapa el código emplearemos la librería opencage [11] que permite codificar geográficamente con el nombre de un lugar o bien con la longitud y latitud del sitio de referencia. Es una librería gratuita que permite hacer 2500 consultas al día.

Figura 2.2: Imagen de RStudio

Figura 2.3: Imagen de R

Figura 2.4: Imagen de Opencage

2.2.4. Memoria

Para la memoria emplearemos `Overleaf`. `Overleaf` es un editor online que permite crear documentos en LaTeX.

Figura 2.5: Imagen de Overleaf

2.3. Presupuesto

Los costes se dividen en:

- Costes de Recursos Humanos
- Costes de Recursos Materiales
- Costes de Recursos Indirectos

2.3.1. Costes de Recursos Humanos

Recursos Humanos	Horas	Precio Unitario	Importe total
Programador	192.33	10 €	1923.3 €

Cuadro 2.3: Tabla del presupuesto en costes de Recursos Humanos

2.3.2. Costes de Recursos Materiales

Definimos los costes de Recursos Materiales como el Software y las herramientas de las que vamos a disponer a lo largo de todo el proyecto.

Recursos Materiales	Precio Unitario	Importe total
iMac (20 pulgadas, principios de 2008)	279 €	279 €
RStudio-1.1.463	995 €/año	82,9 €
Overleaf	28 €/mes	28 €
Opencage (Small)	90 €/mes	90 €
Dropbox	20 €/mes	20 €
	Total	499.9 €

Cuadro 2.4: Tabla del presupuesto en costes de Recursos Materiales

2.3.3. Costes de Recursos Indirectos

Recursos Indirectos	Precio Unitario	Importe total
Luz	56,3 €/mes	56,3 €
Fibra	42,99€/mes	42,99 €
	Total	99.29 €

Cuadro 2.5: Tabla del presupuesto en costes de Recursos Indirectos

2.3.4. Costes Totales

Calculamos los costes totales aunando los costes anteriores por las categorías indicadas además de incluir el precio del IVA.

Tipo de Recurso	Precio
Humano	1923.30 €
Material	499.90 €
Indirecto	99.29 €
Total	2522.49 €
IVA(%21)	529.72 €
Total con IVA	3052.21 €

Cuadro 2.6: Tabla del resumen del Presupuesto del proyecto

2.4. Conclusiones

Es bien sabido que una buena planificación sostiene el uso óptimo de los recursos de los que disponemos y ayuda a evitar pérdidas de tiempo en los planteamientos a seguir. Con estas librerías de uso gratuito se ha buscado reducir al máximo los costes del proyecto. Asumiendo la versión de bajo coste que se ha creído oportuno que pondría una empresa que llevara a cabo el trabajo.

3. Diseño del problema

3.1. Introducción

Recordamos que nuestro problema consta de verificar cuales son las zonas más afectadas de España a través de la recogida de datos de la aplicación de Twitter para optimizar la reagrupación de los efectivos de los cuerpos de seguridad. Para conseguirlo, necesitamos definir el tipo de lenguaje de programación que vamos a utilizar. Además de concretar los pasos que queremos que el programa realice con éxito.

Dichos pasos son:

- Recoger los datos de la aplicación Twitter
- Guardarlos los datos de la aplicación Twitter en un fichero
- Analizar el sentimiento del texto
- Mostrar los resultados del análisis obtenido

3.2. Diagrama de clases

Definimos el diagrama de clases poniendo todos los ficheros que tenemos en el directorio llamado R tal y como se muestra en la imagen 3.1.

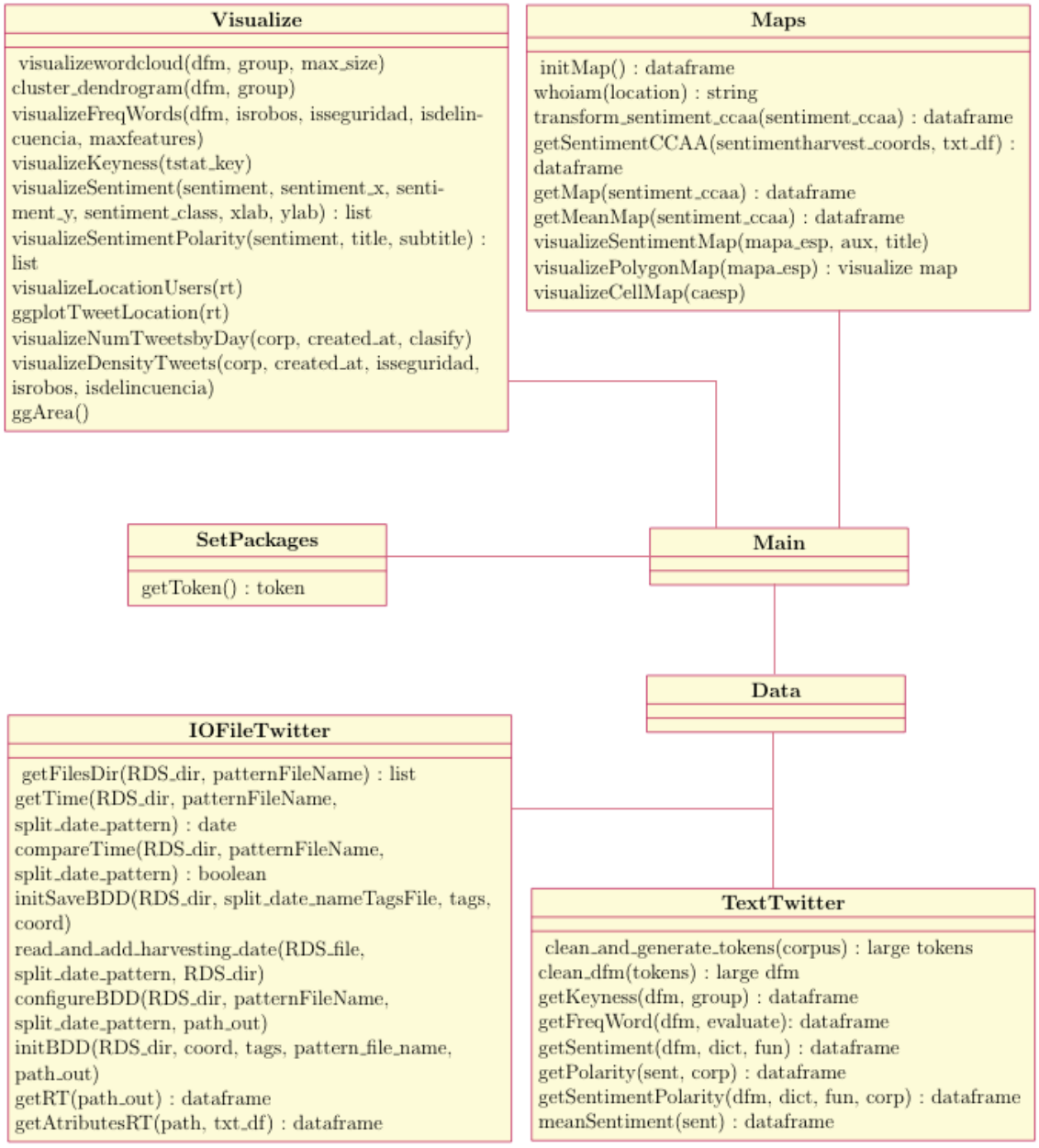

Figura 3.1: Imagen del diagrama de clases.

3.3. Desarrollo

3.3.1. Lenguaje de programación

Se eligió R en lugar de Python como lenguaje de programación debido a que, aunque R es más lento que el Python, R es más potente en análisis y visualización de datos.

3.3.2. Recoger los datos de la aplicación Twitter

Para recoger los datos de las búsquedas hechas Twitter, se escogió la librería `rtweet` [5]. Para emplearla es necesario tener una clave o una cuenta de Twitter como se estipula en el apartado de Propuesta 2. Los datos que recoge son de una ventana de 6 a 9 días atrás. Esta librería posee una función `search_tweets` que devuelve 91 variables en un dataframe. Además se le añade una columna extra con la variable *harvest_date* mediante el método **read_and_add_harvesting_date** [6] con la intención de guardar los ficheros con la fecha definida. Y se le introduce la variable *clasify* que clasifica en función de la variable *query* el tipo de tweet encontrado.

3.3.3. Guardar los datos de la aplicación Twitter en un fichero

Los datos son guardados en formato `.rds` en lugar de `.csv`. Se hizo la comparativa de estos dos ficheros:

```
system.time(readRDS("otherfiles/localizaciones.rds"))
system.time(read.csv("otherfiles/localizaciones.csv"))
```

Extracto de código 3.1: Extracto de código de la comparativa RDS y CSV

Observando en la tabla 3.1 que evidentemente `.rds` es inferior al `.csv`. Pues el primero es un fichero de texto en binario mientras que el segundo es un fichero de texto plano.

tipo fichero	tiempo usuario	tiempo sistema	tiempo transcurrido
rds	0.028	0.001	0.057
csv	0,338	0.005	0.390

Cuadro 3.1: Tabla comparativa de tiempo entre un fichero `.csv` y un `.rds`

Definimos como **tweet** al comentario de un usuario hecho en Twitter. Y **retweet** como la copia que ha hecho un usuario sobre un comentario hecho en Twitter.

Los datos guardados contienen la estructura de un dataframe con 13 columnas. Estas columnas pertenecen a las siguientes variables:

- *user_id*: identificador del usuario que escribió el tweet.
- *status_id*: identificador del tweet.
- *created_at*: fecha en la que fue publicado el tweet.
- *screen_name*: nombre del usuario que escribió el tweet.
- *text*: contenido del tweet.
- *is_retweet*: booleano que determina si el tweet es original o ha sido retweeteado.
- *favorite_count*: número de veces que ha gustado el tweet.
- *retweet_count*: número de veces que ha sido retweetado el tweet.
- *lang*: idioma en el que está escrito el texto del tweet.
- *place_name*: ubicación desde la que ha escrito el usuario el tweet.
- *location*: lugar escrito por el usuario desde dónde se ha publicado el tweet.
- *query*: consulta con la que se recogió el tweet. Puede ser: #Robo, Robos, #Seguridad, #Delincuencia y delitos.
- *harvest_date*: fecha en la que fue recogido el tweet y guardado en la base de datos.

3.3.4. Analizar el sentimiento del texto

Para el sentimiento del texto se necesita un diccionario [7] y adecuar los caracteres en un corpus definido por la librería `quanteda` [1]. El propio texto fue limpiado tal como se ha explicado en el apartado de instrumentos 2.1.2 del capítulo dos.

Cuando terminamos por ejecutar la función **clean_and_generate_tokens** 2.1.2 nos devuelve una estructura large tokens donde cada tweet está dividido en palabras. Esta subsecuencia de palabras tiene el nombre de n-grama. Y dependiendo del número de palabras en las que se divide se denomina unigrama, esto es divido por palabras, o bien bigrama, divido por parejas de palabras.

3.3.5. Mostrar los resultados del análisis obtenido

Para mostrar el análisis obtenido habían dos problemáticas a tener en cuenta:

- La primera era la petición de introducir una tarjeta de crédito para trabajar con la geolocalización de `Google Maps`. Por lo que se buscó una alternativa mediante la librería `opencage` [11] que permitía en el formato gratuito 2500 peticiones por día empleando una clave.

- La segunda problemática fue encontrada en la visualización del mapa. Utilizando la librería `leaflet` [3], que otorgaba un OpenStreetMap de forma totalmente gratuita.

3.4. Conclusiones

Las mejores librerías que no tengan restricciones en el uso son de pago. Por consiguiente, es necesario tener en mente el número de peticiones que uno desea hacer en la aplicación para no sobrecargar los límites gratuitos ofrecidos. Además, para mayor seguimiento del sentimiento encontrado en la aplicación, se podría haber definido un diccionario de palabras con los pesos establecidos. Sin embargo, por límites en el tiempo se ha dejado esa funcionalidad de lado.

4. Implementación

4.1. Introducción

En este capítulo explicaremos de manera muy superficial la estructura de los ficheros del proyecto en R que hemos construido.

4.2. Directorios

El proyecto TFG se divide en los siguientes directorios:

- **CNIG_MAPS**: contiene dos mapas de España. Tal como indica el nombre de ambos directorios: *Provincias* y *CCAA* (Comunidades Autónomas).
- **databdd**: directorio donde se guardan los ficheros `.rds` con las peticiones hechas mediante la función search_tweets de la librería `rtweet`. Cada uno de los ficheros tiene en el nombre la fecha en la que fue creado seguido del nombre *robos_esp* o *seguridad_esp*.
- **databddout**: directorio donde se unifican todos los ficheros de databdd en dos: *all_robos_unique* y *all_seguridad_unique*. El primero posee todos los ficheros definidos como *robos_esp* y el segundo todos los definidos como *seguridad_esp*.
- **images**: contiene todas las imágenes de los mostreos que se han ido haciendo para el testeo.
- **otherfiles**: contiene el diccionario en formato `.txt` (*negative_words_es* y *positive_words_es*) y todas las localizaciones definidas en el área de España (*localizaciones*).
- **R**: directorio en el que está alojado todo el código en `.R`.
- **renv**: librería `renv` [9] que administra las rutas de la biblioteca y aísla las dependencias del proyecto.

4.3. Código Relevante

Dentro del directorio **R** encontramos:

- `Data.R`: fichero que carga los ficheros: `DatafileTwitter.R`, `IOFileTwitter.R` y `TextTwitter.R`.
- `IOFileTwitter.R`: fichero que contiene los métodos de lectura y escritura de la base de datos de Twitter.
- `Main.R`: fichero dónde se ejecuta el programa.

- `MapsSpain.R`: fichero donde se carga el mapa de España.
- `SetPackages.R`: fichero que hace la carga de las librerías que vamos a utilizar en el programa y lleva a término la autenticación de Twitter.
- `TextTwitter.R`: fichero que realiza los cálculos sobre el corpus. Este es definido como el conjunto de textos hallados en los tweets.
- `Visualize.R`: fichero que contiene los métodos de visualización de los datos.

4.3.1. IOFileTwitter

Hacemos especial mención a los métodos **initSaveBDD** y **getAtributesRT**.

En **initSaveBDD** hacemos las llamadas a la API de Twitter. Las coordenadas están definidas en un `string` que contiene tres variables: latitud, longitud y kilómetros.

```
initSaveBDD <- function(RDS_dir, split_date_nameTagsFile, tags,
   coord){
  tagsbd <- search_tweets2(tags, n=18000, geocode = coord[1],
     retryonratelimit=TRUE, token = getToken())
  tagscanarias <- search_tweets2(tags, n=18000, geocode =
     coord[2], retryonratelimit=TRUE, token = getToken())
  both_df <- rbind(tagsbd, tagscanarias)
  saveRDS(object = both_df, file = file.path(RDS_dir,
     paste0(Sys.Date(), split_date_nameTagsFile)))
}
```

Extracto de código 4.1: Extracto de código de la función initSaveBDD

En **getAtributesRT** pedimos el fichero de datos de la Twitter en el que escogemos los tweets en idioma español, hacemos la limpieza de la variable *location*, filtramos por el listado con todos los nombres de población, provincia y comunidad autónoma habidos en España [8] y seleccionamos 13 variables de las 93.

```
getAtributesRT <- function(path, txt_df){
  rt <- getRT(path) %>%
    filter(stringr::str_detect(lang, "es")) %>%
    mutate(location =
       stringr::str_remove_all(stringr::str_to_title(location),
       "[^[:alnum:],' ]+| $|^ ")) %>%
    filter(location %in% txt_df$Poblacion | location %in%
       txt_df$Provincia | location %in% txt_df$Comunidad) %>%
    select(user_id, status_id, created_at, screen_name,
       text,is_retweet, favorite_count, retweet_count, lang,
       place_name, location, query, harvest_date)
  return(rt)
}
```

Extracto de código 4.2: Extracto de código de la función getAtributesRT

4.4. Pruebas

Para el análisis empleamos métodos de la librería `quanteda`:

- La función `textplot_wordcloud` de la librería `quanteda.textplots` muestra una nube de palabras con el tamaño adecuado a la frecuencia que poseen las palabras. La función `textstat_keyness` de la librería `quanteda.textstats` calcula en base a un objetivo las semejanzas y diferencias con la clave ofrecida y sus frecuencias.
- La función `textstat_dist` de la librería `quanteda.textstats` nos ofrece la posibilidad de confeccionar un dendrograma en base a la distancia que hay entre los diversos textos almacenados en los ficheros `.rds`. Por defecto emplea la distancia euclidiana para el cálculo y para observar la semejanza entre ambos textos, el método de correlación.
- La función `textstat_polarity` de la librería `quanteda.sentiment` nos ofrece tres maneras predefinidas de calcular el sentimiento de un texto:
  - sent_logic:
  $$sentiment = log(\frac{positive}{negative})$$
  - sent_abspropdiff:
  $$sentiment = \frac{positive - negative}{Total_{features}}$$
  - sent_relpropdiff:
  $$sentiment = \frac{positive - negative}{positive + negative}$$

4.5. Conclusiones

En este capítulo concluimos que la complicación observada en el lenguaje R reside en el conjunto de librerías distintas que dispone para hacer una misma función.

5. Pruebas

5.1. Introducción

En este capítulo mostraremos las imágenes sobre los diversos estudios que se han realizado en formato de gráfica y visualización de mapas de España con el sentimiento encontrado por provincia.

5.2. Volumen de los datos

En las figuras 5.1 y 5.2 se muestra el número de tweets encontrados con respecto a la densidad y al volumen de los mensajes en función del tiempo. Observamos en el primer gráfico 5.1 que el número de tweets relacionados con la delincuencia es muy superior a los robos y a la seguridad. Sin embargo, en el segundo 5.2 en relación a la densidad, se observa que la delincuencia va en aumento, los robos descienden y la seguridad se mantiene constante.

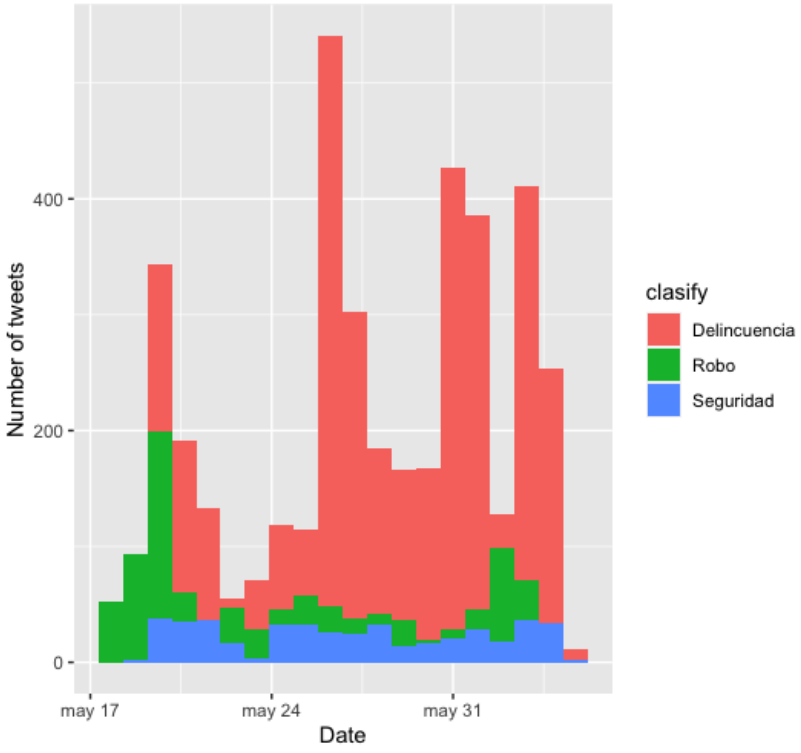

Figura 5.1: Imagen del volumen de Tweets

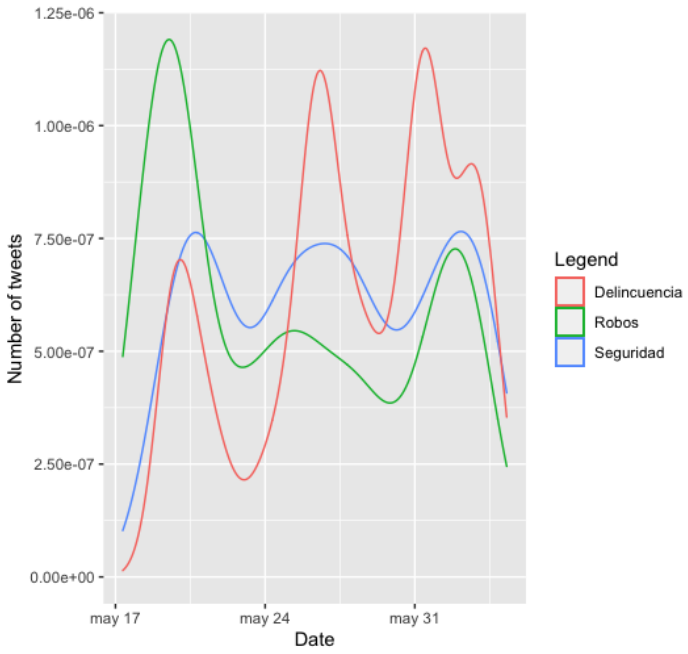

Figura 5.2: Imagen de la densidad del volumen de los tweets

5.3. Unigramas

Definimos como unigramas como al conjunto de textos dividido por palabras.

5.3.1. Nube de palabras

En la figura 5.3 podemos ver una clasificación por nube de palabras separadas por colores cada una de las clases. Marrón para robos, naranja para seguridad y gris para delincuencia. Además, observamos en la figura 5.3 que hay mayoría de palabras en la clasificación por Seguridad, con un peso notable (debido al tamaño) en la clasificación Robo y con una menor incidencia en la clasificación Delincuencia.

Figura 5.3: Imagen de la nube de palabras del conjunto compuesto por unigramas

5.3.2. Dendrograma

Un dendrograma es un tipo de representación que clasifica los datos en función de la distancia, agrupándolos por categorías, es decir, en ramas de un árbol. En la figura 5.4 observamos un dendrograma sobre las tres clasificaciones: robos, seguridad y delincuencia. Contra todo pronóstico, atisbamos que el parecido en base a la distancia euclidiana entre las tres clasificaciones es mayor entre Seguridad y Robo que entre Robo y Delincuencia.

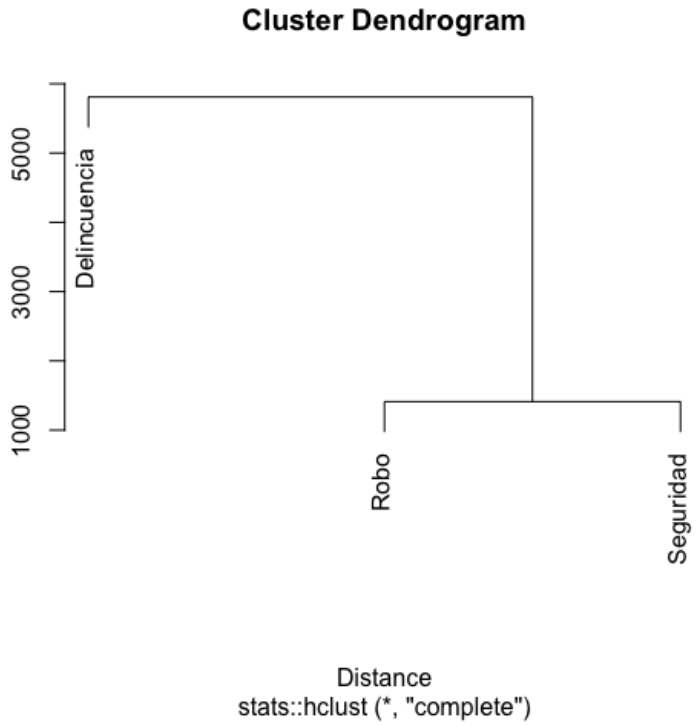

Figura 5.4: Imagen del dendrograma de la clasificaciones Robo, Delincuencia y Seguridad

5.3.3. Frecuencia de palabras

En la figura 5.5 visualizamos las veinte palabras que aparecen con mayor frecuencia en las clasificaciones de Robo, Seguridad y Delincuencia. En la figura 5.6 se muestra la comparativa de las frecuencias de palabras relacionadas con la clasificación Seguridad contra las clasificaciones Robos y Delincuencia. Encontrando en azul las palabras del *target*, es decir, de la clasificación seguridad y en gris aquellas que no aparecen y por tanto es improbable que pertenezcan a la clasificación definida como *target*.

Encontramos que en la figura 5.5 las dos palabras que mayor frecuencia tienen en la clasificación Delincuencia con la palabra *delito* y *no*. Para Robos son *robo* y *delito* y para Seguridad *seguridad* y *no*. Por otro lado, también observamos que el número de veces que aparece en la clasificación Seguridad es minúsculo en comparación a Robos y Delincuencia. En la figura 5.6 contrastamos las palabras que hay en la clasificación Seguridad contra las clasificaciones de Robo y Delincuencia. Atisbamos que la predicción sobre la seguridad es alta para las palabras como *información* o *internacional* e improbable para las palabras como *delitos* o *condenados*.

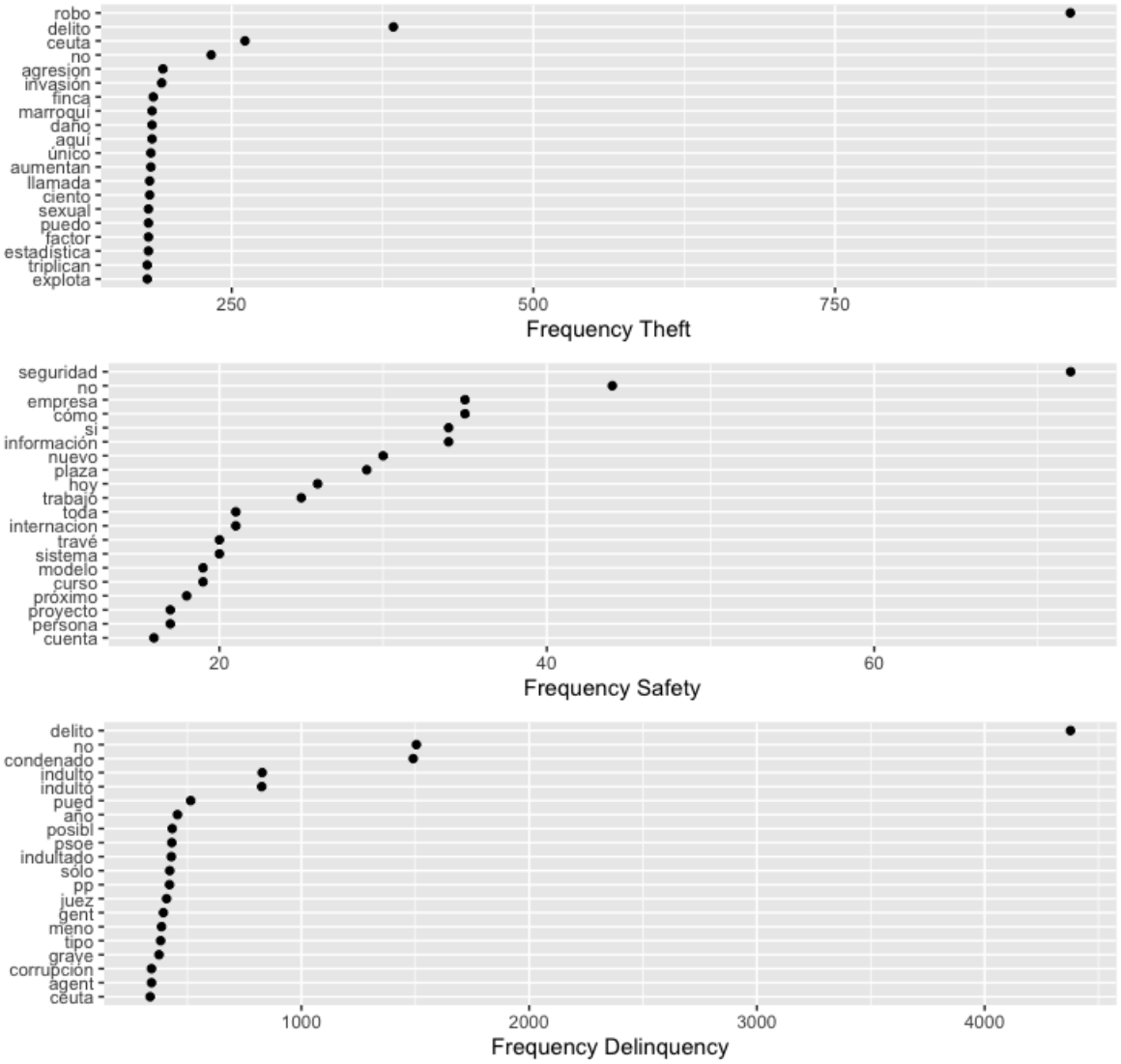

Figura 5.5: Imagen de la frecuencia de palabras compuesta por unigramas dividida por cada una de las clasificaciones hechas

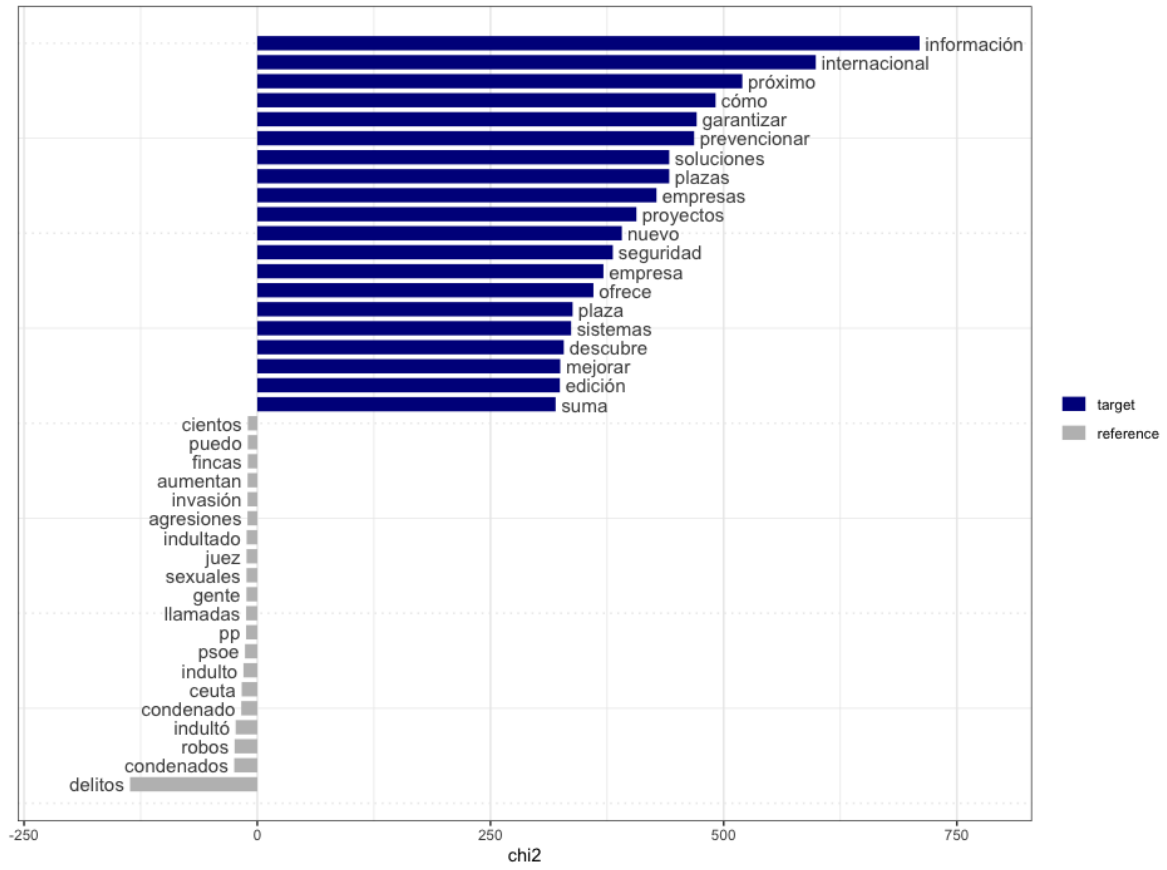

Figura 5.6: Imagen de la comparativa de unigramas de la clase Seguridad

5.3.4. Análisis de sentimiento

En la figura 5.7 hacemos un cálculo del sentimiento encontrado en cada tweet en función del día que fue creado. Distinguimos además el tipo de clasificación del tweet. Cada gráfico constituye a un tipo de cálculo diferente en relación a los positivos y negativos encontrados al contrastar el diccionario con el texto.

En la figura 5.8 calculamos la media por día de los datos encontrados en la figura 5.7 y nos damos cuenta que sentimiento positivo solo es la clasificación Seguridad. Tras los valores encontrados en la figura 5.8, agrupamos los datos en función de si son positivos, negativos o neutrales. Y observamos en el gráfico circular de la 5.9, en el que se contabiliza por la polaridad pintando de naranja los tweets positivos, de amarillo los neutrales y de verde los positivos, agrupados por las distintas clasificaciones.

En el gráfico 5.9 se observa que indistintamente del método de análisis de sentimiento que empleemos no influye en el resultado final. Por lo que nos decantaremos con la función lógica.

En la figura 5.10 visualizamos el sentimiento lógico calculado y las coordenadas obtenidas. Donde observamos muy superficialmente que la mayoría de provincias son azules, y por tanto pertenecen a la clasificación Seguridad. Mientras que Galicia y Cataluña tienen un ligero tinte verde (clasificación Robos), Extremadura, Canarias, Baleares, Cantabria y La Rioja de naranja (clasificación Delincuencia).

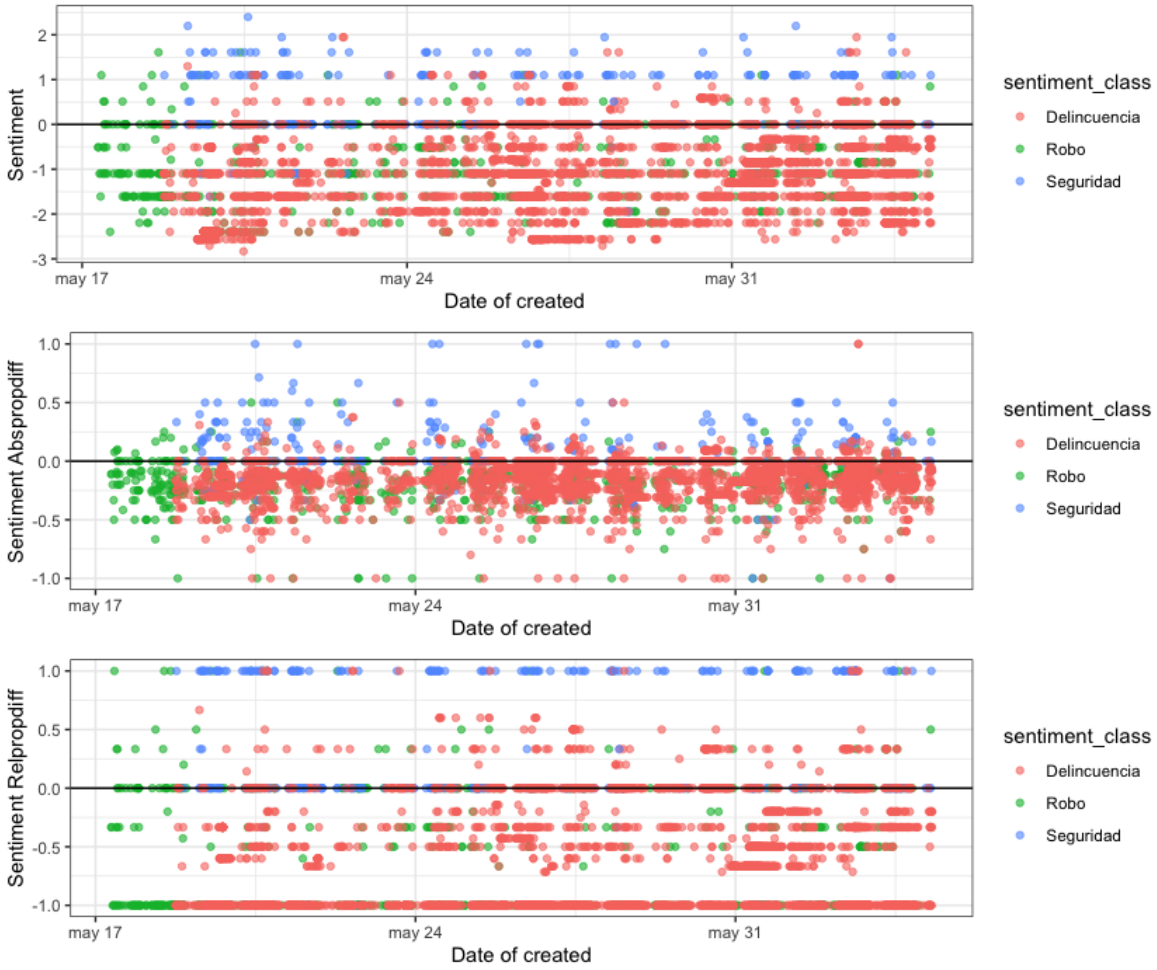

Figura 5.7: Imagen del cálculo del sentimiento de unigramas en función del método Lógico, Abspropdiff y Relpropdiff

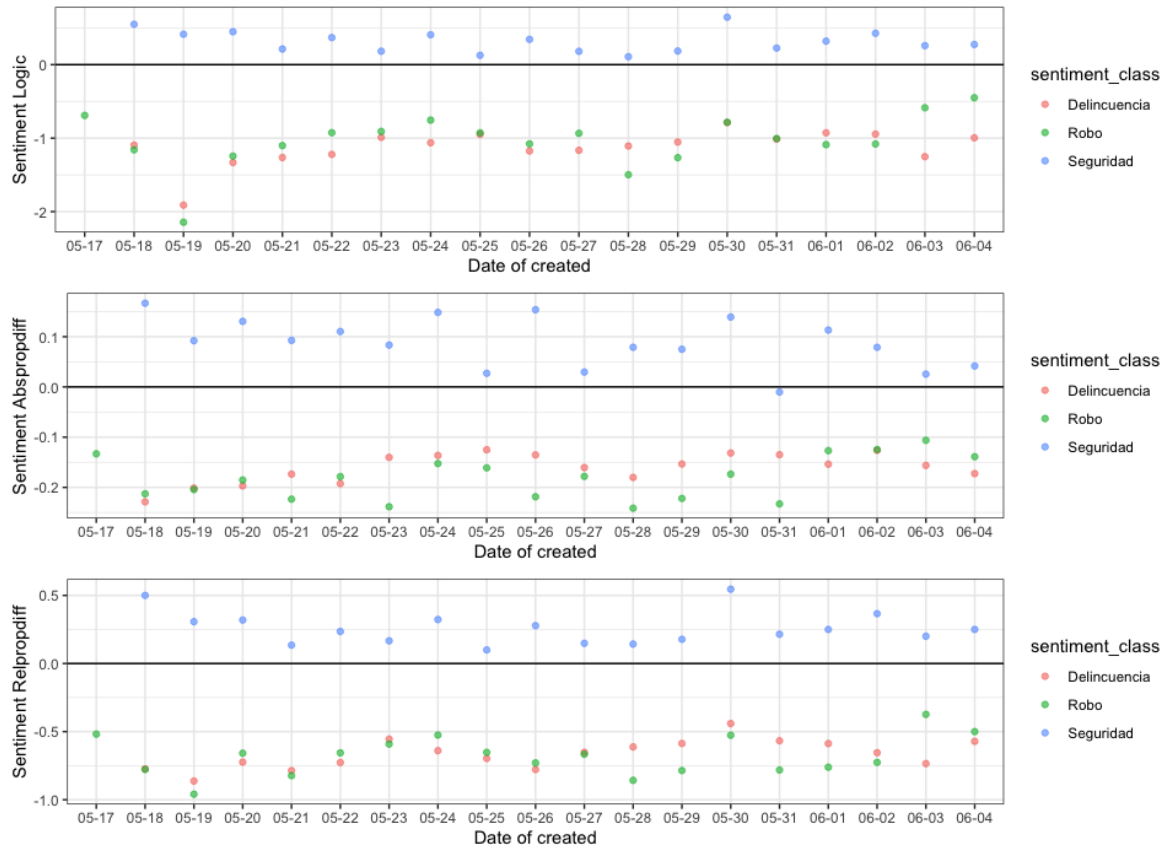

Figura 5.8: Imagen del cálculo de la media del sentimiento de unigramas en función del método Lógico, Abspropdiff y Relpropdiff

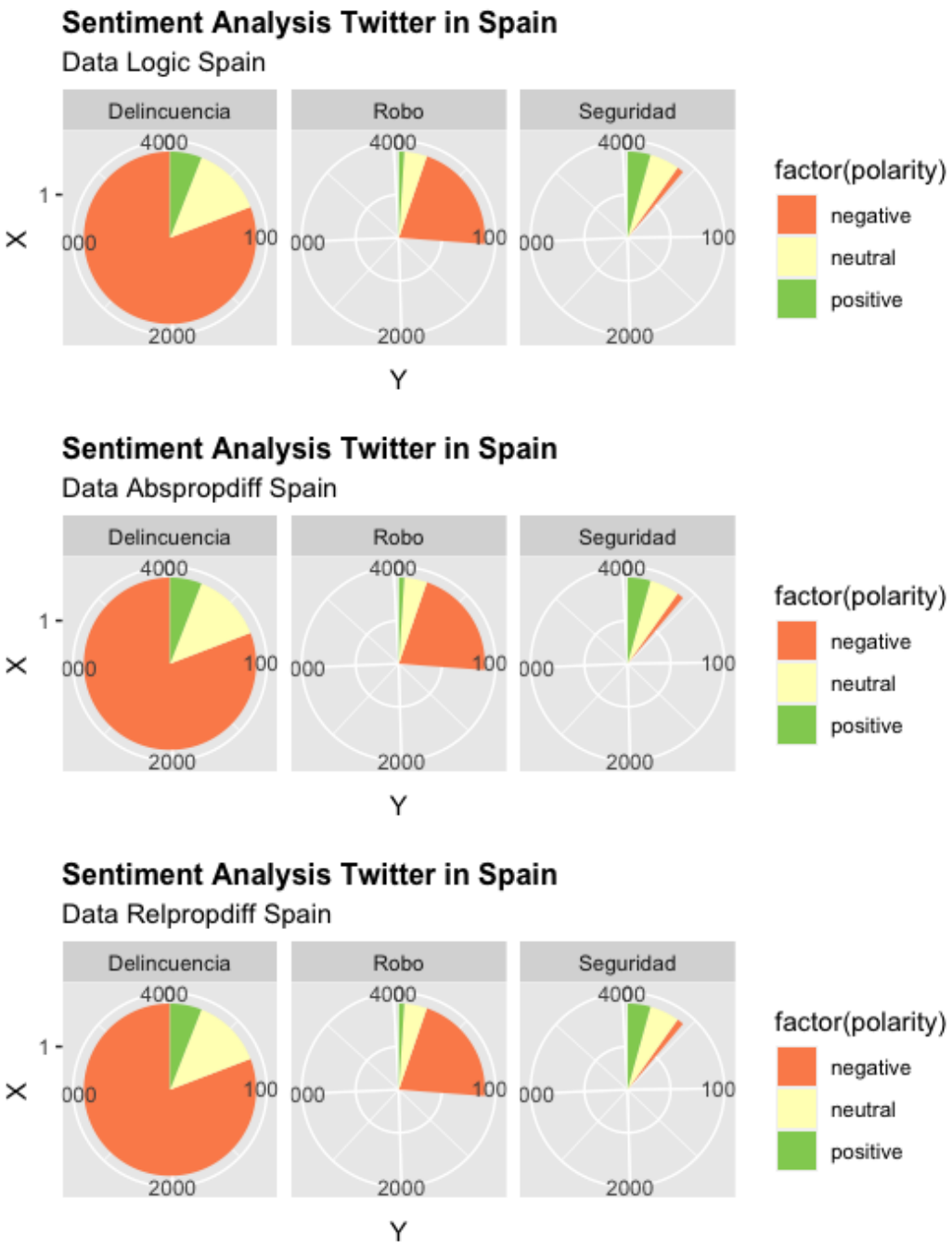

Figura 5.9: Gráfico circular de unigramas. Agrupa los tweets en función de su polaridad y su clasificación

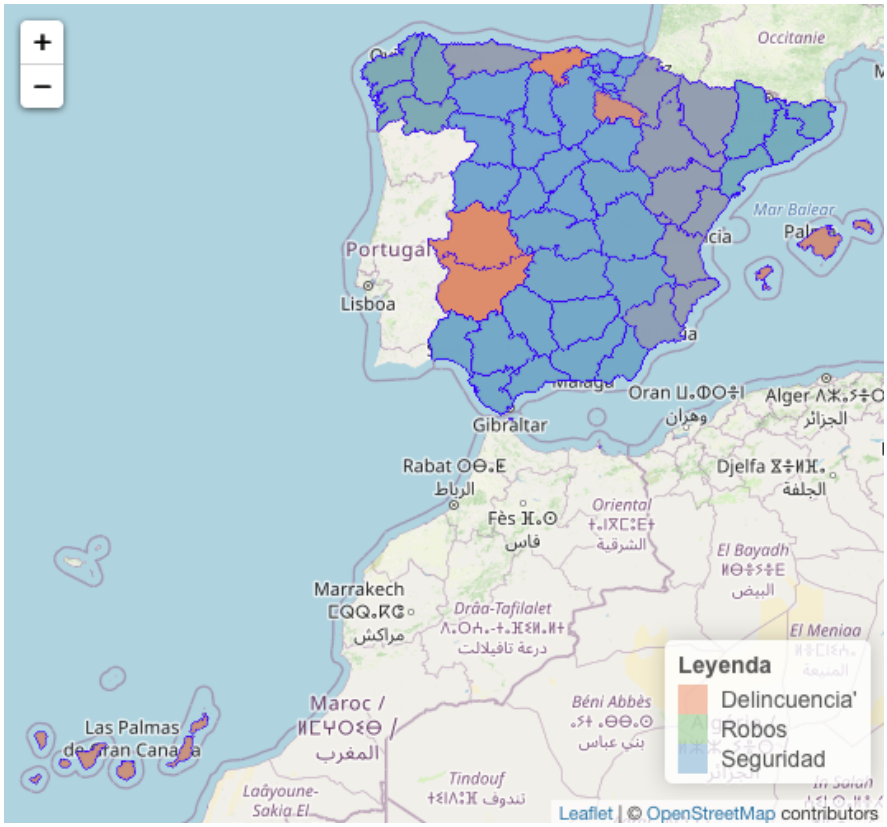

Figura 5.10: Mapa de España. Compuesto por una separación de unigramas y definido por el número de tweets que pertenece a clase Robos, Delincuencia y Seguridad

5.4. Bigramas

Definimos como bigramas como al conjunto de textos dividido por parejas de palabras.

5.4.1. Nube de palabras

En la figura 5.11 podemos ver una clasificación por nube de parejas de palabras separadas por colores cada una de las clases. Marrón para robos, naranja para seguridad y gris para delincuencia. Observamos en la figura 5.11 que hay mayoría de palabras en la clasificación por Seguridad. Con un peso notable (debido al tamaño) en contraposición a la clasificación Robo y la clasificación Delincuencia.

Figura 5.11: Imagen de la nube de palabras compuesta por bigrams

5.4.2. Frecuencia de palabras

En la figura 5.12 visualizamos las quince parejas de palabras que aparecen con mayor frecuencia en las clasificaciones de Robo, Seguridad y Delincuencia. En la figura 5.13 se muestra la comparativa de las frecuencias de parejas de palabras relacionadas con la clasificación Seguridad contra las clasificaciones Robos y Delincuencia. Encontrando en azul las palabras del *target*, es decir, de la clasificación seguridad y en gris aquellas que no aparecen y por tanto es improbable que pertenezcan a la clasificación definida como *target*.

Sin embargo, al otear las frecuencias de las parejas de palabras de la figura 5.12, comprobamos que la máxima se encuentra en el conjunto que compone *delitos graves*. Tras un estudio de las palabras, encontramos ese conjunto como el de la clasificación Robo en un tamaño inferior al que debiera. Concretamos que puede deberse a una mala visualización de los conjuntos de palabras. En la figura 5.13 volvemos a encontrar que la pareja más utilizada para la clasificación Seguridad es *información internacional* mientras que la menos predicha es *cuerpos fuerzas*.

5.4.3. Análisis de sentimiento

En 5.14 hacemos un cálculo del sentimiento encontrado en cada tweet en función del día que fue creado. Distinguimos además el tipo de clasificación del tweet.

En la figura 5.15 calculamos la media por día de los datos encontrados en la figura 5.14.

Observamos en el gráfico circular de la 5.16, en el que se contabiliza por la polaridad pintando de naranja los tweets positivos, de amarillo los neutrales y de verde los positivos, agrupados por las distintas clasificaciones.

En la figura 5.17 visualizamos el sentimiento lógico calculado y las coordenadas obtenidas.

En esta ocasión para calcular el sentimiento del conjunto de palabras tenemos en cuenta los términos *no* y *nunca* para modificar el sentimiento reflejado en la pareja de palabras. Razón por la que observamos un repunte interesante de positivos en la figura 5.14. Al calcular la media de la figura 5.14, no obstante, observamos que la única clasificación positiva en la figura 5.15 es la de Seguridad. Si miramos el gráfico 5.16 y lo comparamos con el 5.9 no encontramos modificación alguna en el número de positivos, neutrales y negativos analizados en el conjunto de parejas de palabras. Por otro lado, la visualización en el mapa 5.17 se ve modificada por el conjunto de colores. Encontrando que en Aragón y Valencia hay un repunte de Delincuencia y en Murcia se aprecia el verde de la clasificación de Robos.

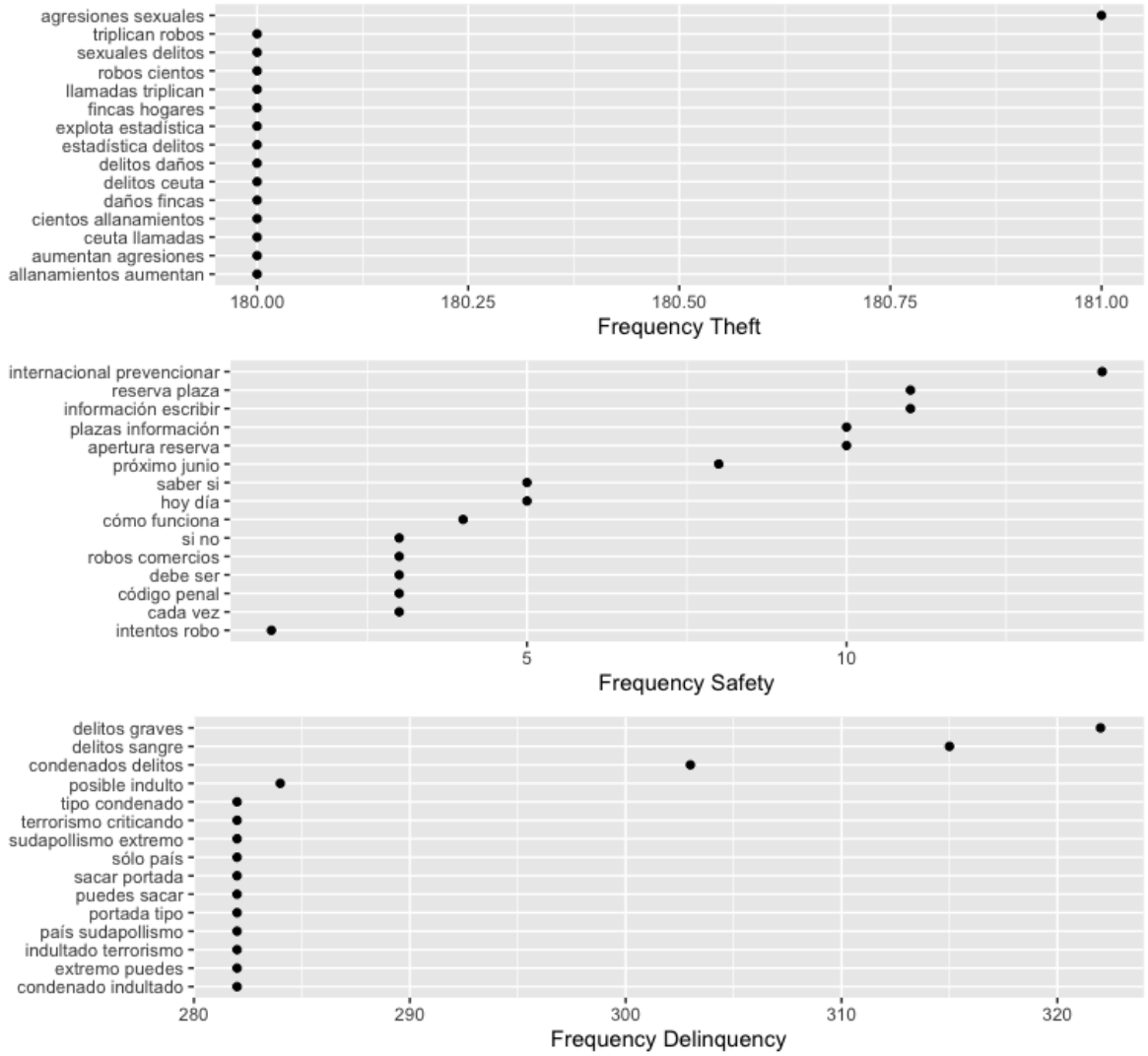

Figura 5.12: Imagen de la frecuencia de palabras compuesta por bigramas y dividida por cada una de las clasificaciones hechas

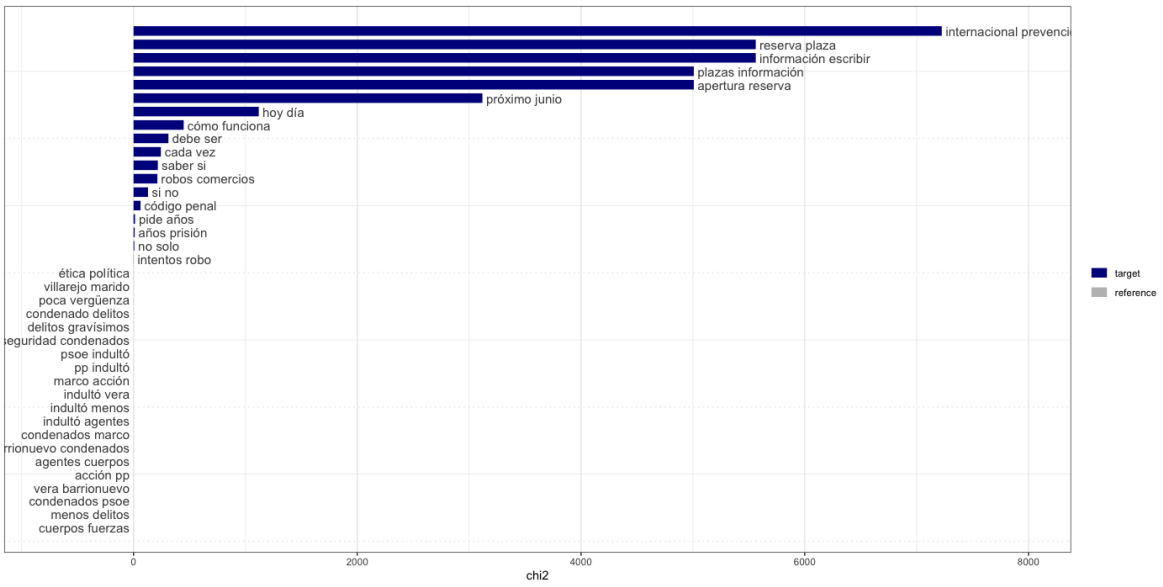

Figura 5.13: Imagen de la comparativa de bigramas de la clase Seguridad

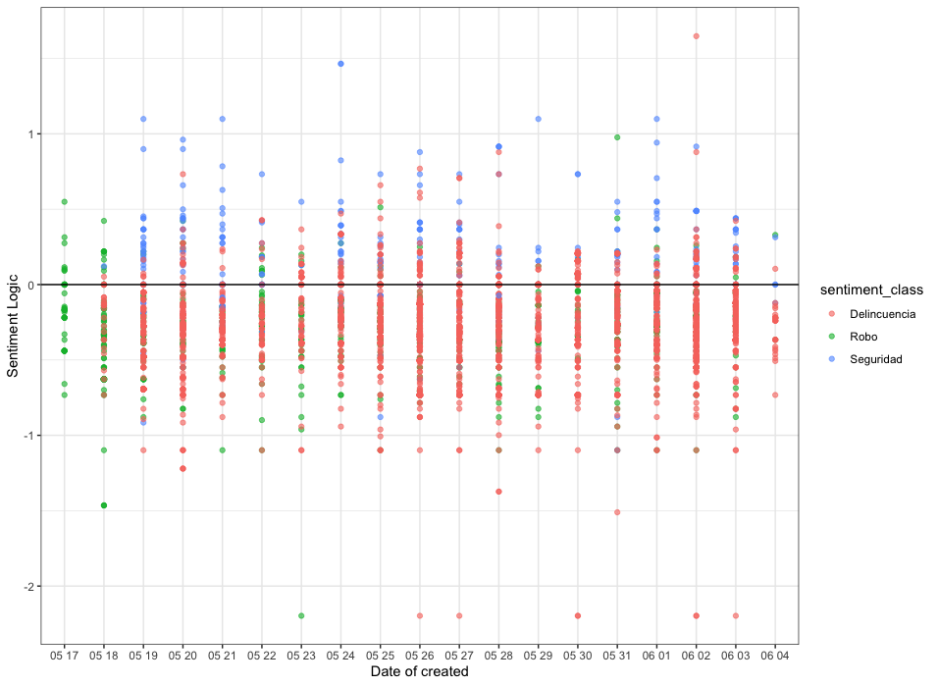

Figura 5.14: Imagen del cálculo del sentimiento de bigramas en función del método Lógico

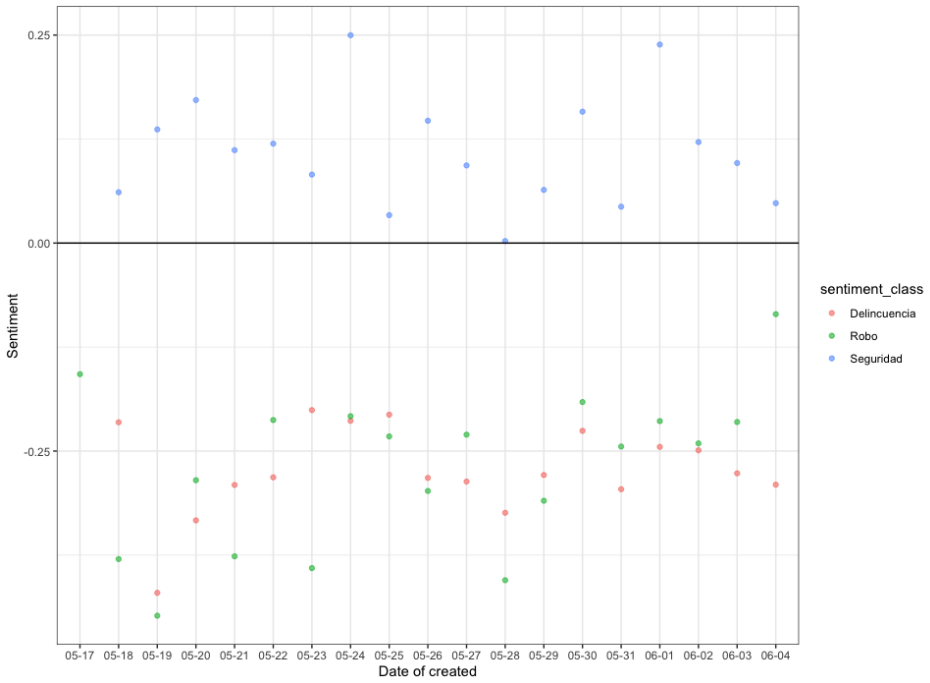

Figura 5.15: Imagen del cálculo de la media del sentimiento de bigramas en función del método Lógico

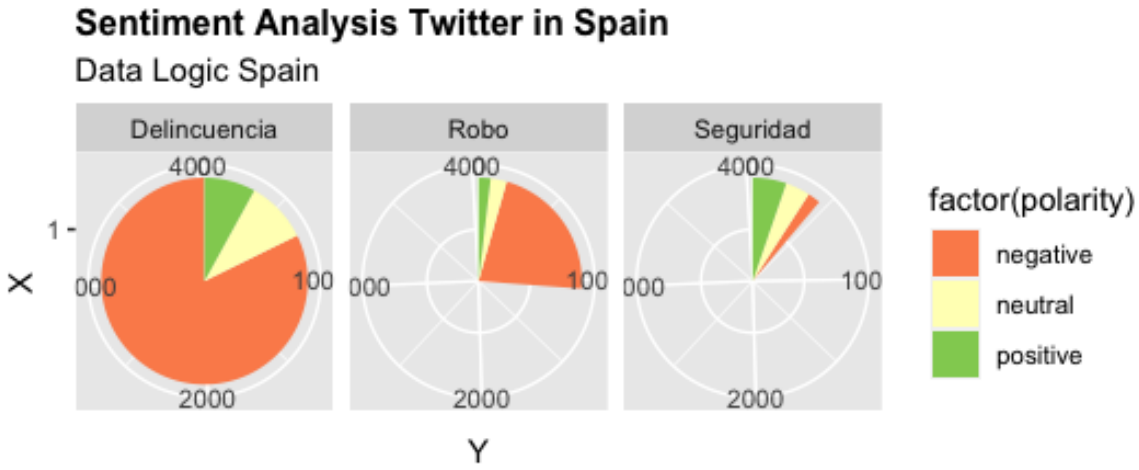

Figura 5.16: Gráfico circular de bigramas. Agrupa los tweets en función de su polaridad y su clasificación

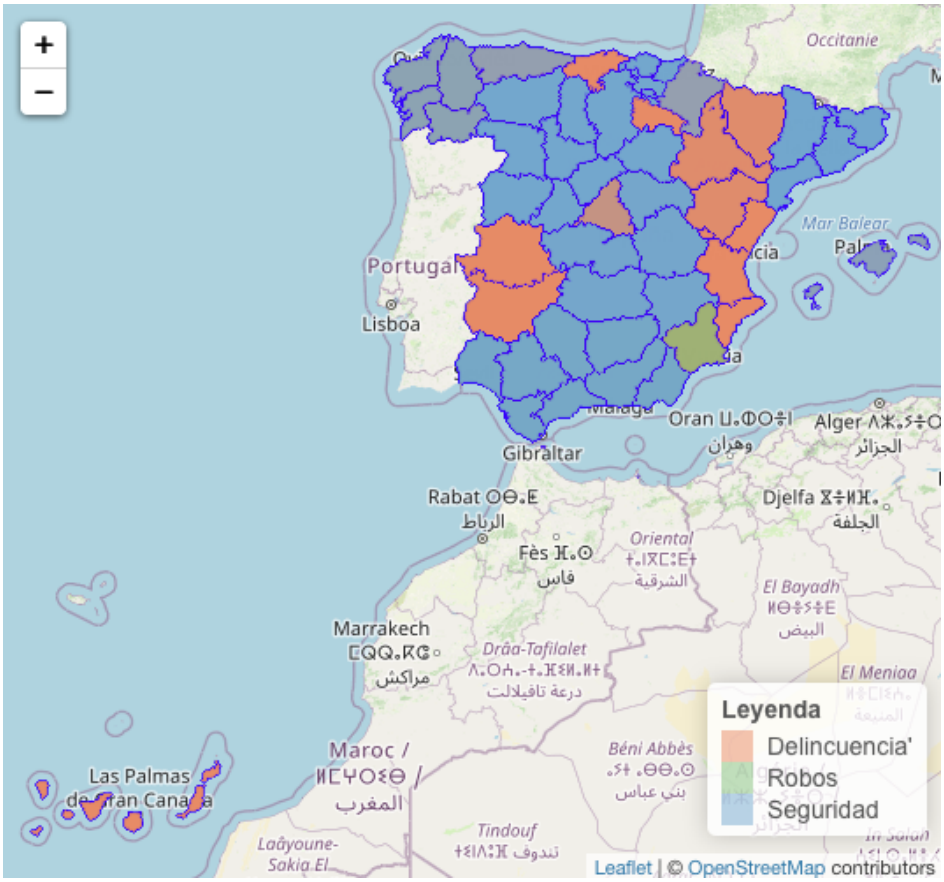

Figura 5.17: Mapa de España. Compuesto por una separación de bigramas y definido por el número de tweets que pertenece a clase Robos, Delincuencia y Seguridad

5.5. Comparativa

Para la comparativa hemos escogido dividir los mapas por cada una de las clasificaciones. Es decir, que hay un mapa de solo Robos, solo Seguridad y solo Delincuencia. Haremos una comparativa entre unigramas y bigramas por cada una de las clasificaciones. Y a continuación un único mapa por clasificación que visualiza la totalidad de los casos en cada provincia.

5.5.1. Sentimiento

En el caso de los Robos, Seguridad y Delincuencia, podemos observar tanto bigramas como unigramas están en el mismo rango de sentimiento. Si bien hay una inclinación para bigramas de normalizar el espectro de unigramas, acotándolo en menor o mayor medida.

Robos

Podemos ver que el espectro se mantiene en el sentimiento negativo. Vislumbrándose en la leyenda de las figuras 5.18 y 5.19. Mientras que en la figura 5.18 de unigramas encontramos con mayor negatividad Navarra, Aragón, Valencia, Extremadura y Asturias. En la figura 5.19 encontramos la mayor negatividad en La Rioja, Navarra, Valencia y Extremadura. Los menos negativos, por otro lado, no han sufrido cambios. El mayor cambio atisbado entre los dos es en Canarias. Pasando del rosado pálido en unigramas a un verde en bigramas. Hecho que nos hace pensar que delante de las palabras observadas había un «no» delante.

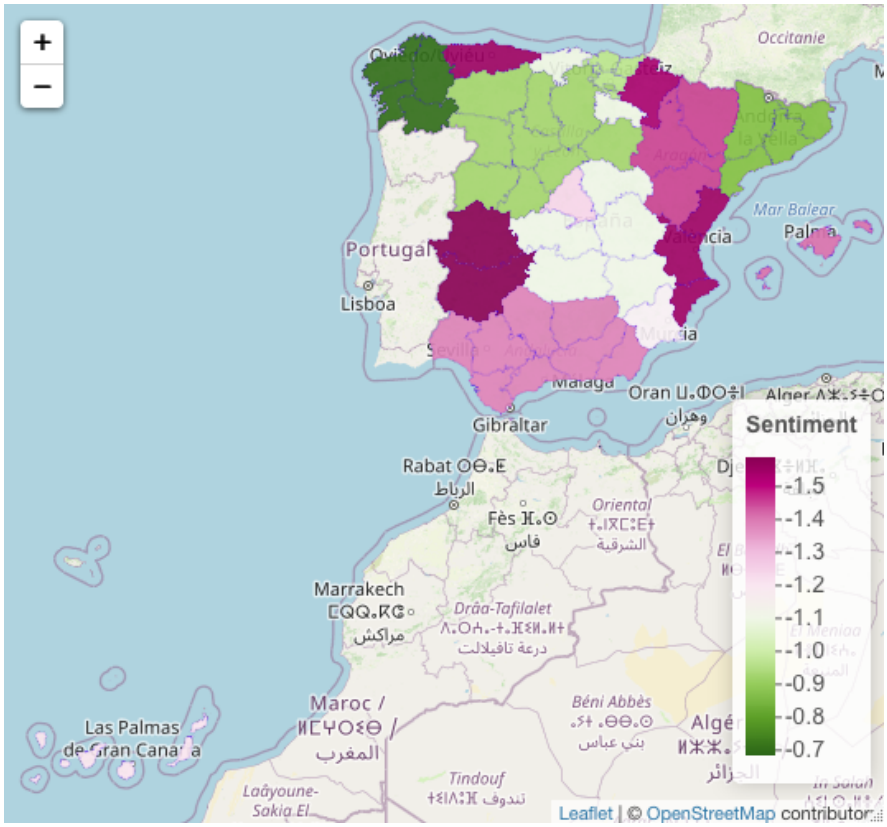

Figura 5.18: Mapa de España. Robos unigram media del sentimiento.

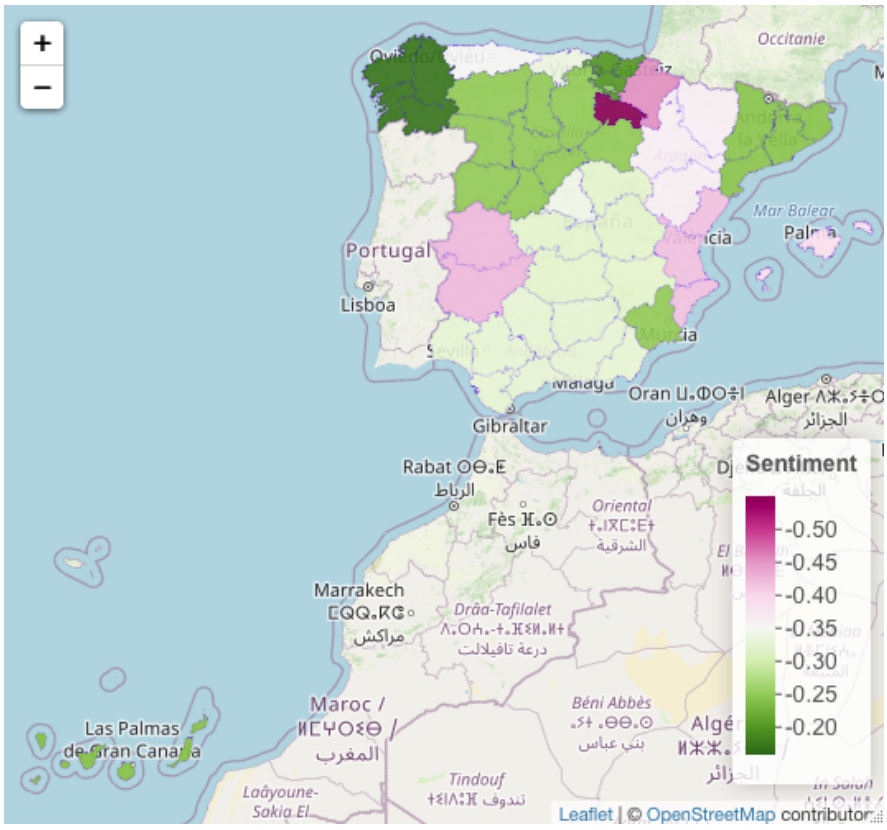

Figura 5.19: Mapa de España. Robos bigram media del sentimiento.

Seguridad

Podemos ver que el espectro se mantiene en el sentimiento positivo. Vislumbrándose en la leyenda de las figuras 5.20 y 5.21. Mientras que en la figura 5.20 de unigramas encontramos con mayor positividad Navarra, Castilla y León y Castilla la Mancha. En la figura 5.19 encontramos la mayor positividad en Castilla La Mancha, Galicia y Cataluña. La mayor negatividad sobre la seguridad permanece inmutable en Cantabria. Encontramos relevante el cambio que sufren Navarra especialmente y Aragón y Valencia, de una mayor positividad a una negatividad considerable que tiene sentido si se la relaciona con el índice de casos negativos encontrados anteriormente en Robos.

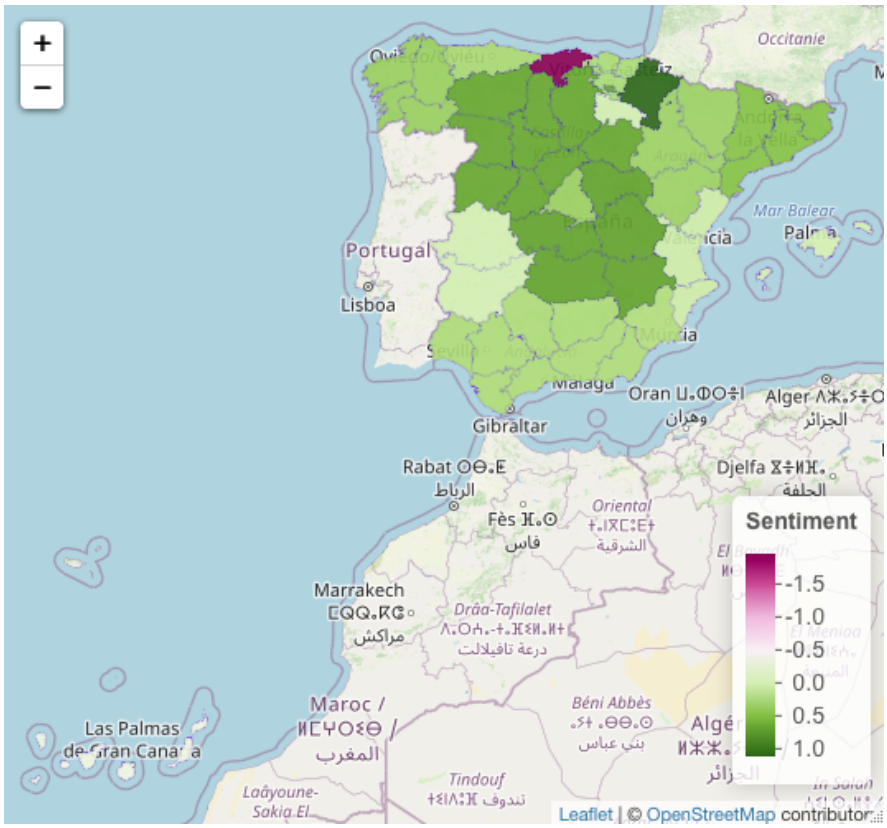

Figura 5.20: Mapa de España. Seguridad unigram media del sentimiento.

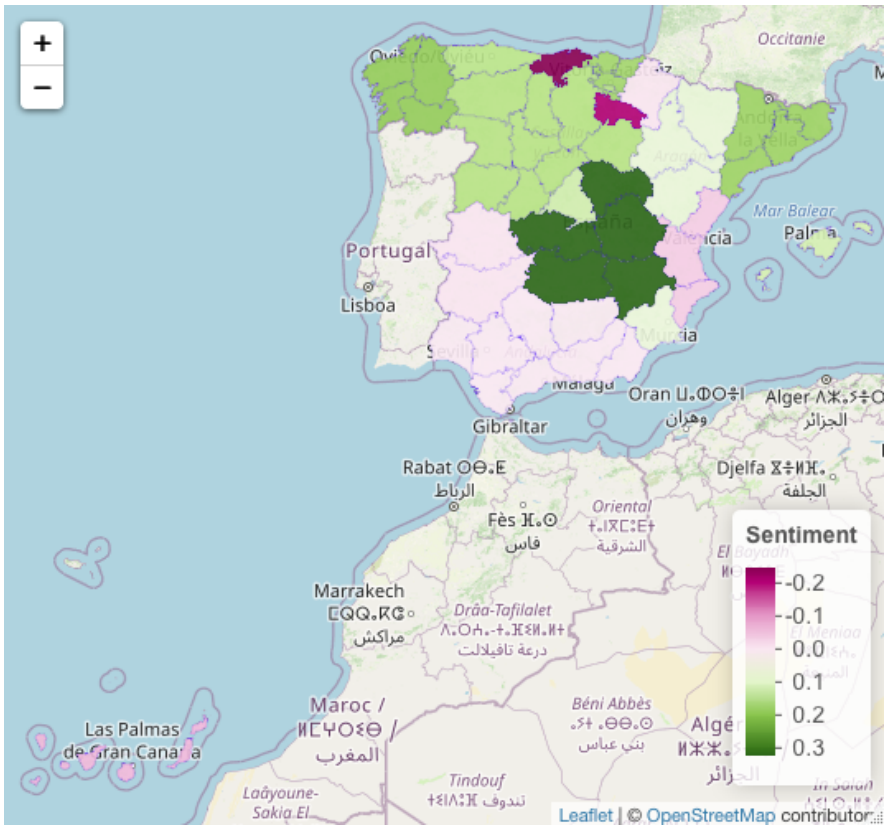

Figura 5.21: Mapa de España. Seguridad bigram media del sentimiento.

Delincuencia

Podemos ver que el espectro se mantiene en el sentimiento negativo tal y como se predecía por los datos encontrados en Robos. Vislumbrándose en la leyenda de las figuras 5.22 y 5.23. Mientras que en la figura 5.22 de unigramas encontramos con mayor negatividad Valencia, Asturias, País Vasco y Canarias. En la figura 5.23 encontramos la mayor negatividad en Extremadura, País Vasco, Castilla La Mancha y Valencia. No encontramos ningún cambio significativo en el comportamiento, sino que más bien el índice de casos se mantiene regular.

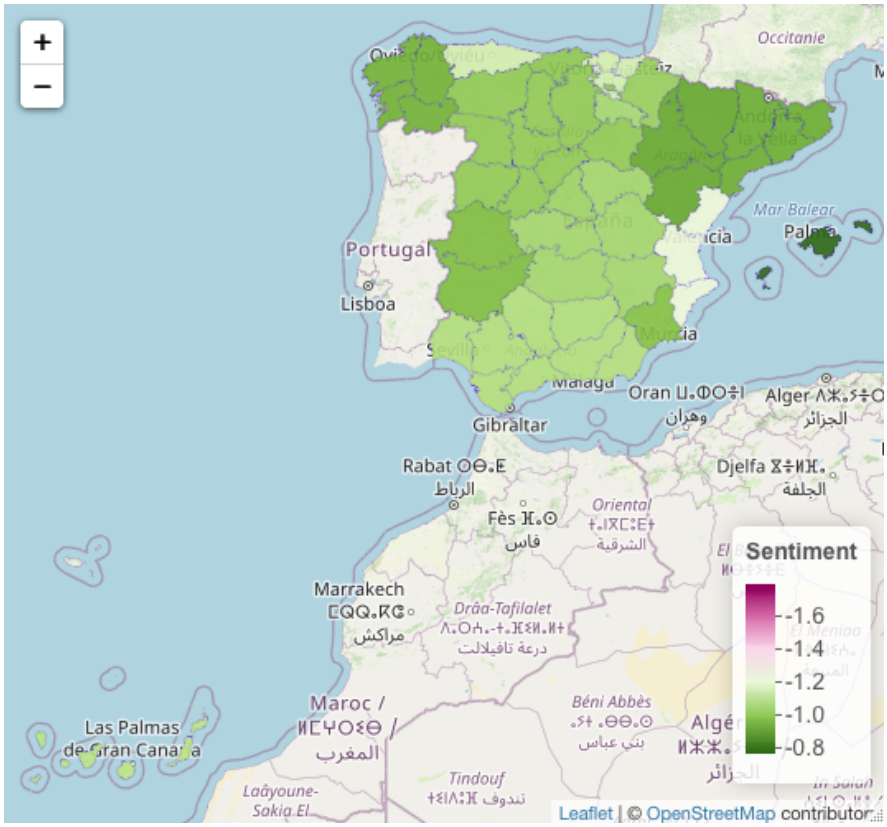

Figura 5.22: Mapa de España. Delincuencia unigram media del sentimiento.

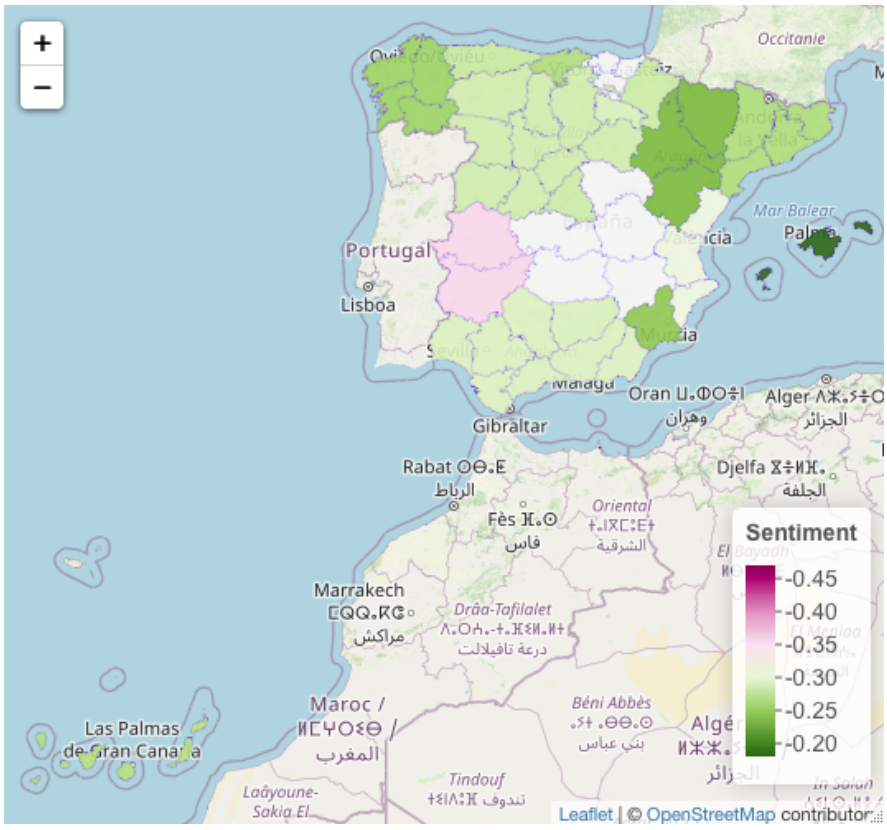

Figura 5.23: Mapa de España. Delincuencia bigram media del sentimiento.

5.5.2. Total

Podemos observar de manera superficial en las figuras 5.24, 5.25 y 5.26 que el mayor número de casos en sentimiento está alojado en Madrid para las tres clasificaciones. Siguiéndose después con una mayoría de casos en Andalucía y en Cataluña. Este dato nos hace indicativo que la mayor intensidad de opiniones reside en esas tres Comunidades Autónomas. Hecho que en contraste con el sentimiento encontrado en las tres clasificaciones, pasa desapercibido. Mientras que las Comunidades que han sido tildadas de mayor negatividad son además las que menor cantidad de opiniones se encuentran en la búsqueda.

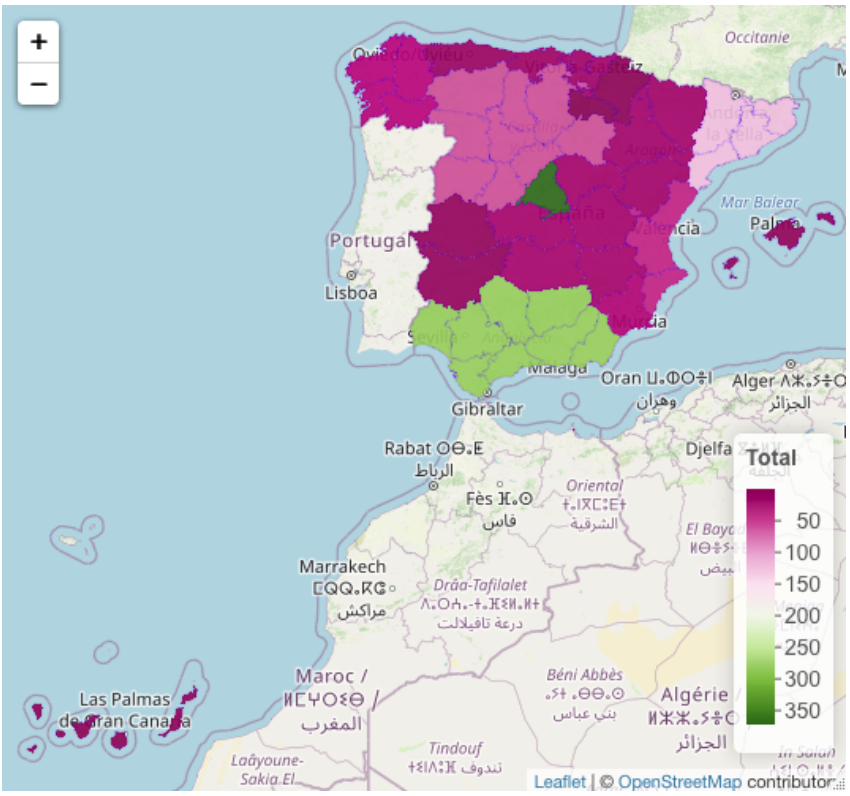

Figura 5.24: Mapa de España. Cómputo total de sentimiento de Robos por Provincia.

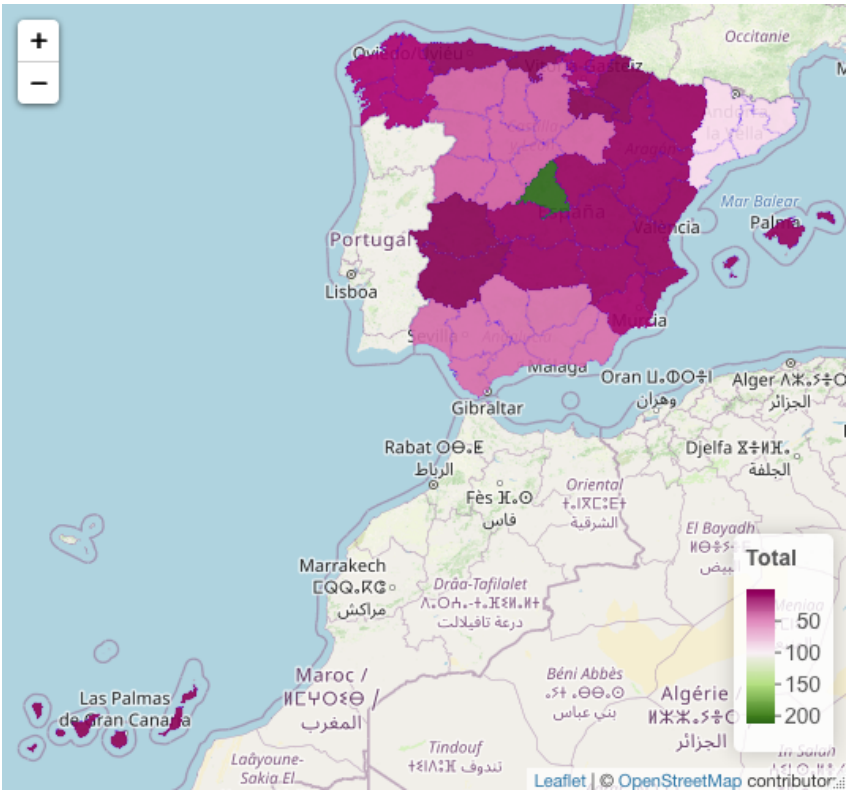

Figura 5.25: Mapa de España. Cómputo total de sentimiento de Seguridad por Provincia.

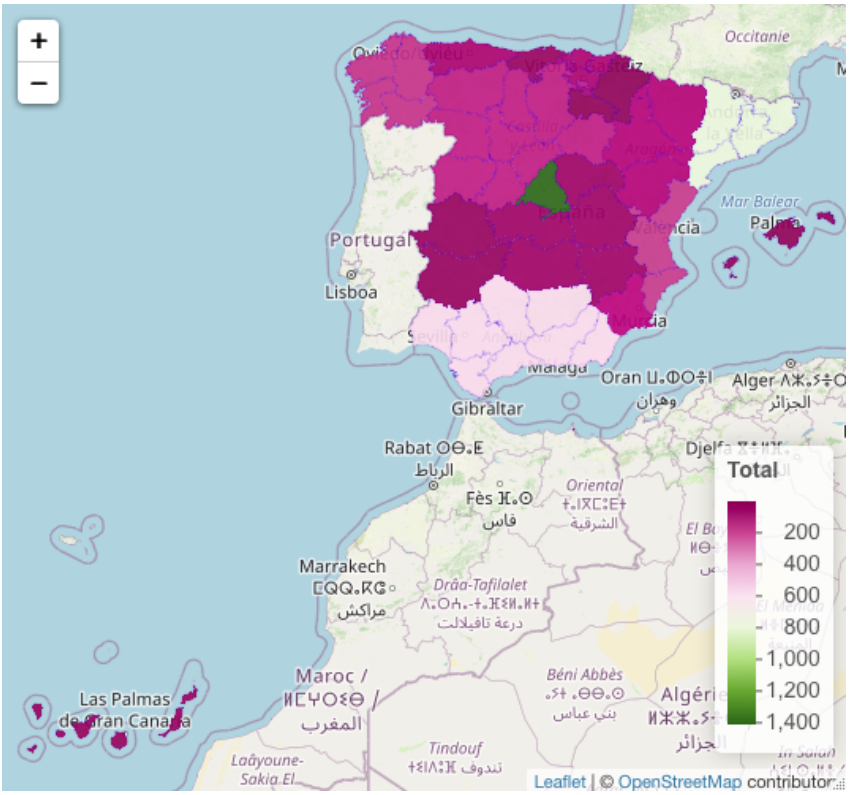

Figura 5.26: Mapa de España. Cómputo total de sentimiento de Delincuencia por Provincia.

5.6. Conclusiones

Para hacer las distintas deliberaciones tomamos en cuenta la figura 5.27 que muestra los delitos penales referentes al primer trimestre del año 2021 y la figura 5.21 anterior que muestra el sentimiento de la comunidad referente a la seguridad, enumerado como la figura 5.28.

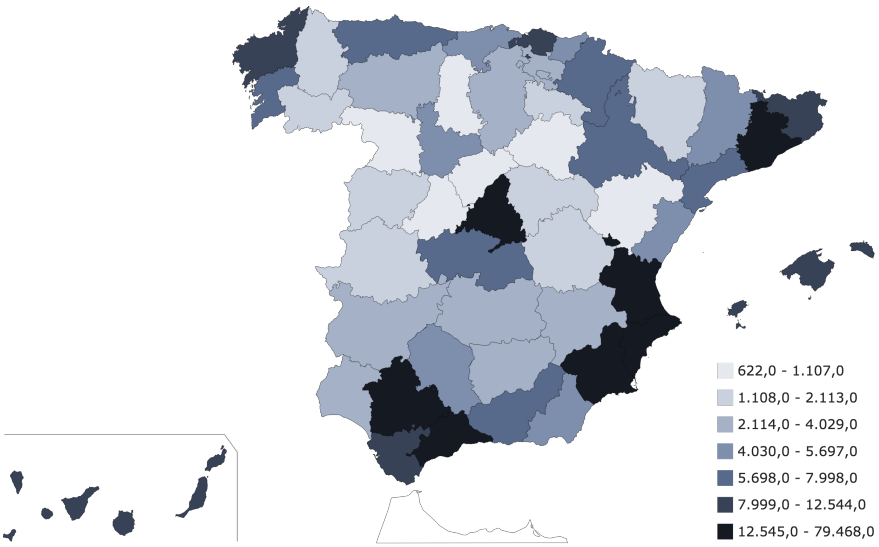

Figura 5.27: Mapa de los delitos penales referentes al primer trimestre del año 2021. Fuente: [2]

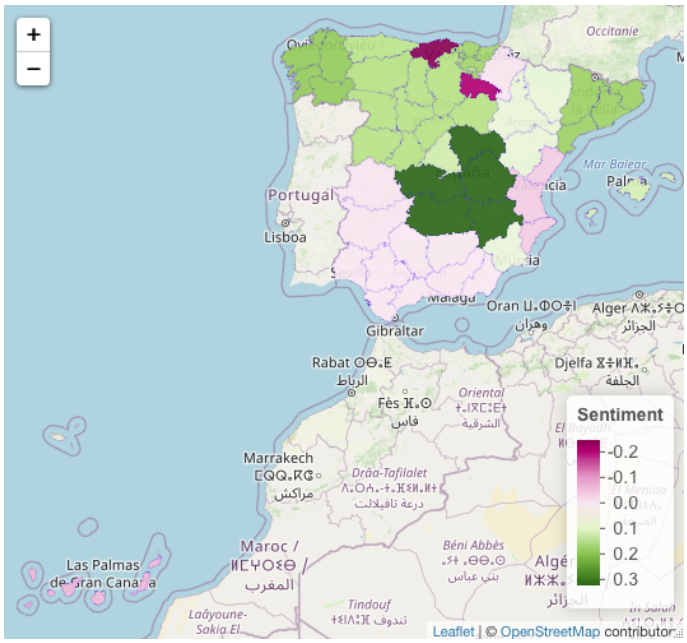

Figura 5.28: Mapa de España. Seguridad bigram media del sentimiento.

Concluimos que la actuación en Madrid es muy positiva en relación al número de delitos encontrados, pues estos alcanzan la totalidad de **79000** casos. Y la actuación en Cantabria y La Rioja es muy deficiente, pues el sentimiento negativo para un número de

pequeño de **5000** y **1000**, respectivamente. Por otro lado, la sensación de inseguridad es alarmante de las Palmas de Gran Canaria, Valencia e Andalucía. Por lo que se requeriría hacer una modificación en los efectivos del cuerpo policial en Castilla La Mancha, reagrupándolos hacia las zonas afectadas.

6. Conclusiones

Los datos encontrados para el análisis de sentimiento de la población en cada una de las localidades definidas en los mapas referenciados en las conclusiones del capítulo de pruebas 5.6 ofrecen distintos puntos de vista. Desde poblaciones que se sienten muy seguras con un índice de casos grande hasta poblaciones que se sienten inseguras con un índice de casos pequeño. Sin embargo, tal como se ha demostrado en ese estudio, la actuación de la seguridad en el país deja mucho que desear en Las Palmas de Gran Canaria, Valencia, Andalucía. Además de una necesidad de fortalecer La Rioja y Cantabria por la negatividad encontrada en la comunidad.

Se concluye, por ende, que hay una posibilidad de mejorar la propuesta sobre como se agrupan los efectivos policiales a lo largo de toda el área de España. Que si bien este es un primer paso y nos faltan datos en referencia a la visualización de los datos, poniendo de manifiesto el grueso de la población en cada una de las comunidades y contabilizando el número de efectivos agrupado en cada zona.

7. Trabajo a futuros

Tal como se ha indicado en el capítulo de las conclusiones 6 el trabajo a futuro está en la optimización de ese sentimiento en relación al número de efectivos colocado en una provincia. También se debe de hacer referencia al cómputo de población que protegen para determinar lo buenos o lo malos que son en esa zona en concreto del mapa.

En otro contexto, teniendo en cuenta que los datos en el portal Estadístico de Criminalidad se muestran en la página por trimestre, puede ser interesante hacer un cálculo en esa ventana de tiempo para observar los datos recogidos sobre la comunidad, en la búsqueda de predecir cuales serán las zonas más afectadas para el trimestre posterior y poder calcular hacia adónde deberían de reagruparse los efectivos del cuerpo policial.

Analizar además el sentimiento de la comunidad con un diccionario construido para determinar la totalidad de las palabras, analizándolas como un sentimiento positivo, negativo o neutral.

Puede llegar a ser interesante expandir el número de palabras que se utilizan para calcular el sentimiento. En lugar de ser solo una palabra o dos, que sean tres o cuatro.

Implementar un método sobre la distancia en cuanto a palabras con la variable *location* de los datos en Twitter, para no aislar los casos en los que ha habido un error en la escritura.

8. Bibliografía

[1] Kenneth Benoit, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. quanteda: An r package for the quantitative analysis of textual data. *Journal of Open Source Software*, 3(30):774, 2018. doi: 10.21105/joss.00774. URL https://quanteda.io.

[2] Ministerio del Interior. Portal estadístico de criminalidad. Technical report, Gobierno de España, 2019. URL https://estadisticasdecriminalidad.ses.mir.es/publico/portalestadistico/portal.html.

[3] Christian Graul. *leafletR: Interactive Web-Maps Based on the Leaflet JavaScript Library*, 2016. URL http://cran.r-project.org/package=leafletR. R package version 0.4-0.

[4] TWITTER INC. Developer portal twitter, 2021. URL https://developer.twitter.com/en/portal/projects-and-apps.

[5] Michael W. Kearney. rtweet: Collecting and analyzing twitter data. *Journal of Open Source Software*, 4(42):1829, 2019. doi: 10.21105/joss.01829. URL https://joss.theoj.org/papers/10.21105/joss.01829. R package version 0.7.0.

[6] Louveaux M. Analysing twitter data with r, 2020. URL https://marionlouveaux.fr/blog/twitter-analysis-part1/.

[7] Rachael Tatman. Sentiment lexicons for 81 languages, 2017. URL https://www.kaggle.com/rtatman/sentiment-lexicons-for-81-languages.

[8] Pau Urquizu. Longitud y latitud de los municipios de españa, 2011. URL https://www.businessintelligence.info/varios/longitud-latitud-pueblos-espana.html.

[9] Kevin Ushey. renv: Project environments for r, 2021. URL https://rstudio.github.io/renv/index.html. version 0.13.2.

[10] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

[11] Ed Freyfogle y Marc Tobias. *OpenCage Geocoding API*, 2013. URL https://opencagedata.com.