Data Acces Capstone: Google Trends Project

Antonio Alaia

March 17, 2022

Contents

1 INTRODUCTION

1.1 Questions research

Inspired by a Rovetta A. study published on JMIR, the aim of this project is testing the hypothesis claiming that the volume of researches on Google Italia about right-wing topics, actors, websites and news has raised during the pandemic.

This project tries to answer to few questions:

1. Crossing date and hits variables, will we observe a positive linear trend?
2. Is there a seasonality of the queries?
3. Overall, how effective can be GoogleTrends on following socio-political movements?

- My researches about the topic suggest me that lockdowns and, more generally, COVID-19 Pandemic, has raised internet traffic and extremism has gained popularity among world citizens, therefore it's likely that my plot visualization will show a positive linear trend.

- No prediction could be made for the first question, since that the answer could be affected by important socio-political events occurred during the pandemic or even changes in the lockdowns severity.

- Last question is a secondary quest I added for a more personal curiosity, so the answer will be more subjective and won't be answered with a statistical approach but rather with a researcher point of view that is testing Rstudio's package gtrendsR and GoogleTrends for his socio-political studies!

1.2 Data and tool:*gtrendsR*

To test the hypothesis, I create a dataset that contains as many right-wing related keywords as possible. In order to do it, I start from scraped list of organizations and people linked to right-wing parties and then look for related queries trying to identify the trending keywords among users.

The starting list is obtained by scraping an already existing map made by patriaindipendente.it. The best way to import a dataset with all trending searches on Google is by using Google Trends and, since I'm exclusively using RStudio to work on this project, I employ gtrendsR package (more info…). This package allows us to get a dataset with different variables: geo, time, keywords and onlyInterest (a boolean that allows to call ONLY the interest_over_time or not).

```
library(tidyverse)
library(rvest)
library(stringr)
library(rio)
library(gtrendsR)
```

2 DATA IMPORT AND PREPARATION

2.1 First dataset

Firstly I imported the dataset by scraping a website (see the reference) and running the *gtrendsR* API to obtain the complete dataframe of the Google Trends statistics about a keyword. After I scraped the list, I had to replace all the keywords inside the gtrendsR argument *keyword*, set the *geo* variable as *"IT"* (Italy) and the period of time chosen for the study: *"2020-01-01 2022-02-28"*.

The first problem I encountered was the fact that the package's argument *keyword* allows just 5 words at a time; to solve that issue, I created a function called *divideR* that allows me to divide the list in several groups of 5 so that I could automatically run one group of five at the time.

```
divideR <- function(x,n) split(x, cut(seq_along(x), n, labels = FALSE))

dvd_listaNera <- divideR(scrap_listaNera, 7)

test1_listaNera <- gtrends(keyword = dvd_listaNera$'1',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28"
)
test2_listaNera <- gtrends(keyword = dvd_listaNera$'2',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28"
)
test3_listaNera <- gtrends(keyword = dvd_listaNera$'3',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28"
)
test4_listaNera <- gtrends(keyword = dvd_listaNera$'4',
```

```r
                           geo = "IT",
                           time = "2020-01-01 2022-02-28"
)
test5_listaNera <- gtrends(keyword = dvd_listaNera$'5',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28"
)
test6_listaNera <- gtrends(keyword = dvd_listaNera$'6',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28"
)
test7_listaNera <- gtrends(keyword = dvd_listaNera$'7',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28"
)
test8_listaNera <- gtrends(keyword = c("8chan","8kun","pillola rossa","olocausto italiano","grande sost
                           geo = "IT",
                           time = "2020-01-01 2022-02-28"
)
```

After that, to enlarge my dataset, and to recreate the environment of a right-wing user that surfs the Internet binge-searching the related topics that the Google algorithm provides him/her, I create a first dataframe with the *related queries* that *gtrendsR* package found.

```r
list_1 <- test1_listaNera$related_queries$value
list_2 <- test2_listaNera$related_queries$value
list_3 <- test3_listaNera$related_queries$value
list_4 <- test4_listaNera$related_queries$value
list_5 <- test5_listaNera$related_queries$value
list_6 <- test6_listaNera$related_queries$value
list_7 <- test7_listaNera$related_queries$value

LIST <- c(list_1, list_2, list_3,list_4,list_5,list_6,list_7)

DATASET <- unique(LIST)
DATASET <- DATASET[-c(2,3,6,32,39,41,44,46,48,49,57,58,61,64:110)]
```

After some cleanings, I had my first list of *queries*. Then, I runned again *gtrendsR* but this time I only asked for *interest_over_time* by adding the argument *interestOnly=T*

```r
DATASET <- divideR(DATASET, 10)


group_1_onlytime <- gtrends(keyword = DATASET$'1',
                            geo = "IT",
                            time = "2020-01-01 2022-02-28",
                            onlyInterest = TRUE

)

group_2_onlytime <- gtrends(keyword = DATASET$'2',
                            geo = "IT",
                            time = "2020-01-01 2022-02-28",
                           onlyInterest = TRUE

)

group_3_onlytime <- gtrends(keyword = DATASET$'3',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28",
                           onlyInterest = TRUE

)

group_4_onlytime <- gtrends(keyword = DATASET$'4',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28",
                           onlyInterest = TRUE

)

group_5_onlytime <- gtrends(keyword = DATASET$'5',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28",
                           onlyInterest = TRUE

)

group_6_onlytime <- gtrends(keyword = DATASET$'6',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28",
                           onlyInterest = TRUE

)

group_7_onlytime <- gtrends(keyword = DATASET$'7',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28",
                           onlyInterest = TRUE

)
group_8_onlytime <- gtrends(keyword = DATASET$'8',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28",
                           onlyInterest = TRUE

)

group_9_onlytime <- gtrends(keyword = DATASET$'9',
                           geo = "IT",
                           time = "2020-01-01 2022-02-28",
                           onlyInterest = TRUE

)

group_10_onlytime <- gtrends(keyword = DATASET$'10',
                              geo = "IT",
                              time = "2020-01-01 2022-02-28",
                              onlyInterest = TRUE

)
```

2.2 Second dataset

Following the method above, I did the same for the second dataset, scraped from another list of keywords found online (see references). This time, my list was quite longer than the first so I had to use a for loop.

```
download.file(url = nuova_lista,
              destfile = here::here("nuovaLista.html"))

scrap_nuovaLista <- read_html(here::here("nuovaLista.html")) %>%
  html_elements(css = "a") %>%
  html_text(trim = TRUE)

new_list <- unique(scrap_nuovaLista)
new_list <- new_list[-c(1)]

DATASET_2 <- vector()
v <- vector()

for (i in 1:446) {
  tryCatch({v <- gtrendsR::gtrends(keyword = new_list[i], geo = "IT", time = "2020-01-01 2022-02-28", o
           error = function(e){})
  cat(i, " ")
  v <- as.data.frame(v$interest_over_time)
  DATASET_2 <- rbind(DATASET_2, v)
}
```

Before merging the two dataset to create the final one, I had to clean the datasets with regex to delete the most general keywords and the other unrelated keywords that fell into the dataframe due to weird outcomes of the Google Algorithm. For the sake of the next regression, I had to cut all the values that fall below the $hits = 5$, and in the next part we'll see what does it mean.

3 TEST AND VISUALIZATION

3.1 Is there a seasonality of the queries?

Having the full dataset for testing my hypothesis, I started with the first: is there a seasonality of the queries? Having the opportunity to cross *hits* that it's a value that express the level of popularity of the term; when it's 100 is the peak of popularity, whilst a value of 50 means that the term is half as popular. Scores of 0 mean that a sufficient amount of data was not available for the selected term.

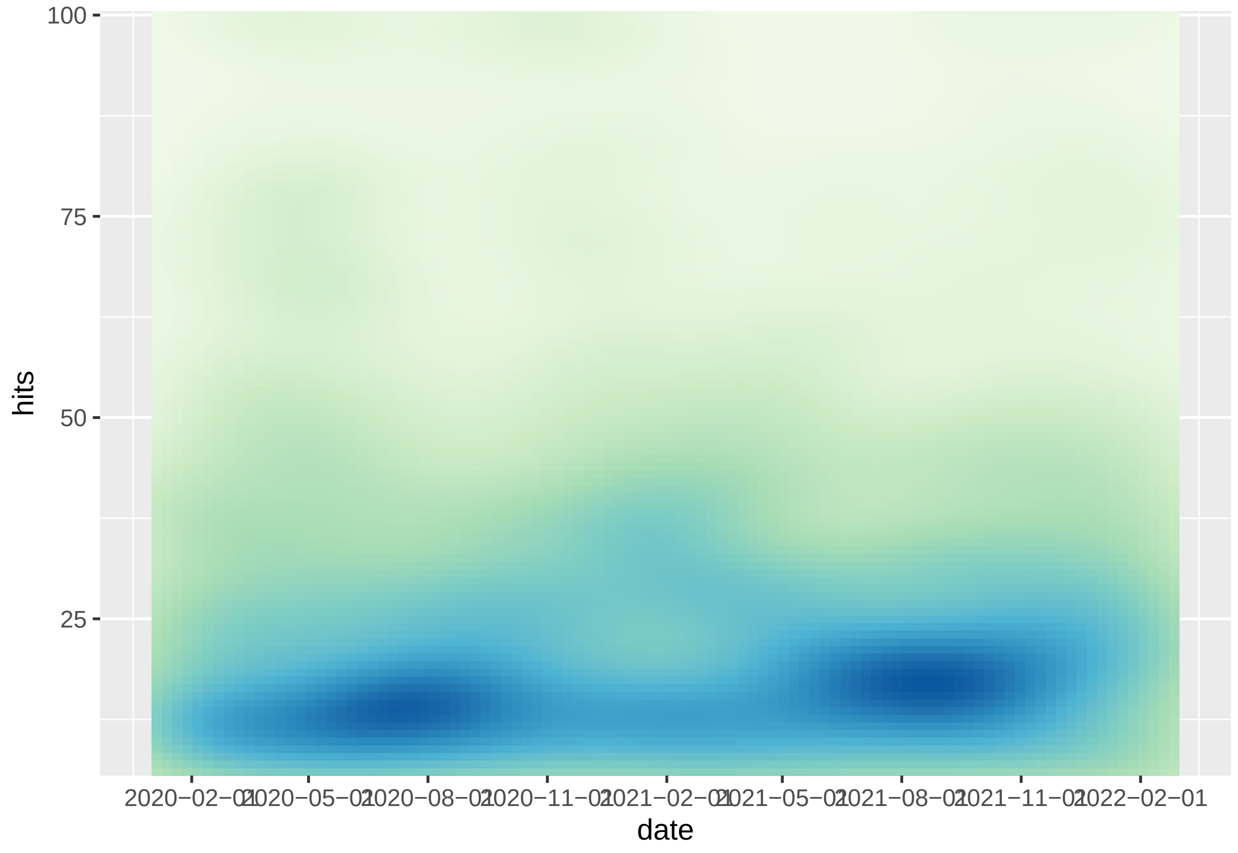
To answer the first question, I did a density plot with *ggplot2*

```
COMPLETE_DATASET_regex <- import("Complete_dataset(cleanedREGEXS).csv")
```

```r
density_plot <- ggplot(COMPLETE_DATASET_regex, aes(x=date, y=hits) ) +
  stat_density_2d(aes(fill = ..density..), geom = "raster", contour = FALSE) +
  scale_fill_distiller(palette=4, direction=1) +
  scale_x_datetime(date_breaks = "3 months") +
  scale_y_continuous(expand = c(0, 0)) +
  theme(
    legend.position='none'
  )

density_plot
```

```
## Warning: Raster pixels are placed at uneven horizontal intervals and will be
## shifted. Consider using geom_tile() instead.
```



As we can see, the plot shows that the first peak has been reached during the summer of 2020, between June and August 2020, while the seconds occurred between August and October 2021. Therefore, even if there isn't a precise pattern that we can notice and can helps us making a prediction model, we can observe that the trending occurred just once a year; furthermore we can spot a little "cloud" during January-February 2021 that indicate a cluster of very trending queries.

3.2 Crossing date and hits: a positive linear trend?

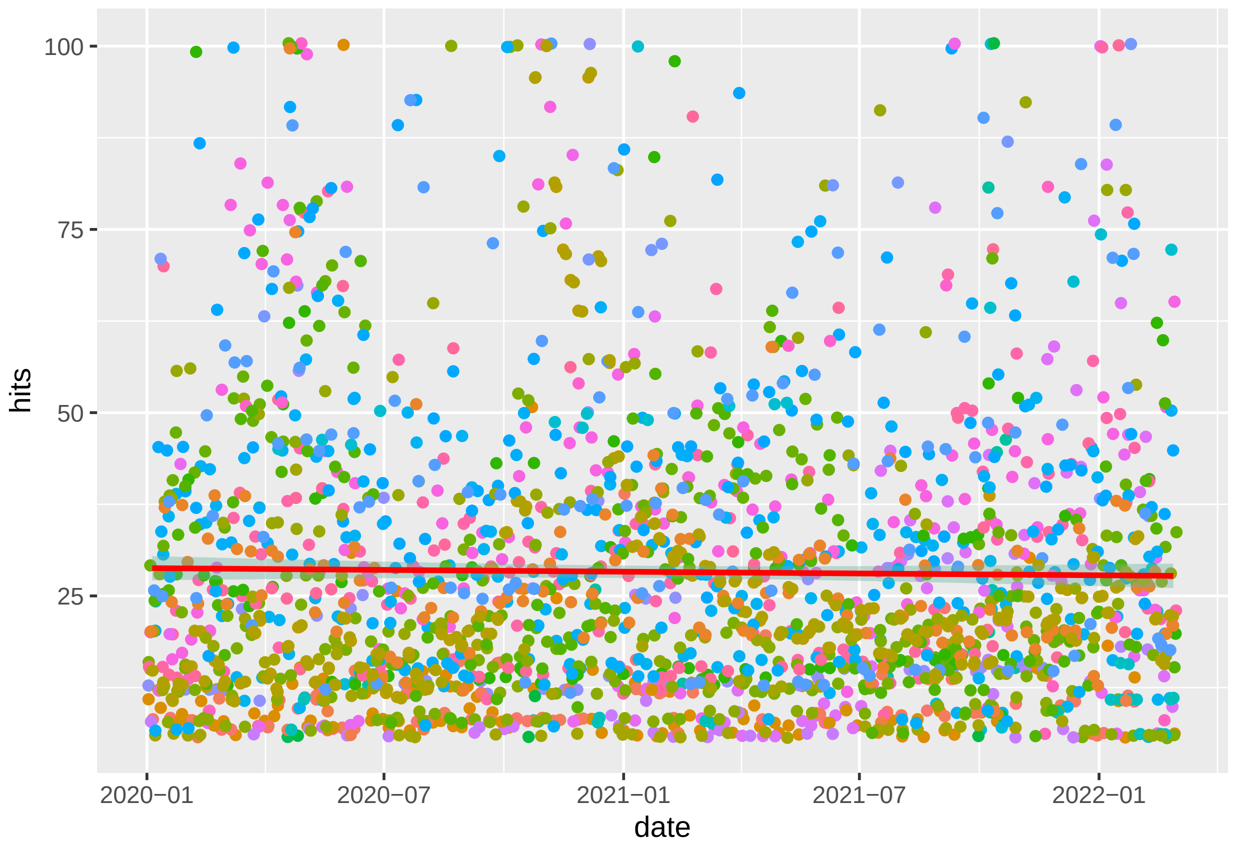
For the second question, let's have a look at the linear trend plot:

```
COMPLETE_DATASET_regex <- import("Complete_dataset(cleanedREGEXS).csv")

linear_plot<- ggplot(COMPLETE_DATASET_regex, aes(x=date, y=hits) ) +
  geom_point(aes(date, hits,color = keyword), position = 'jitter') +
  geom_smooth(method=lm , color="red", fill="#69b3a2", se=TRUE) +
  theme(
    legend.position ='none')

linear_plot
```

```
## 'geom_smooth()' using formula 'y ~ x'
```



It's clear that, even if I removed all the *hits* below a certain value, the majority of the queries is condensed in at the bottom of the plot and, even if the linear trend shows a slight negative slope, the results it's not statistically significant.

4 Conclusion and personal considerations