Big Data Analytics Mini Project Report

Title: - Google Trends Analysis

Group members: -
1. Md. Farhan(1019128)
2. Abhisheik Jadhav(1019130)
3. Aditya Jadhav(1019131)
4. Joel Thomas(1019133)

Abstract: -
For most of us, Google Web Search and other major Google products are our go-to weapon whenever we need to find anything on the internet or in the real world. Whether it's getting the latest Covid pandemic news, the latest scores from your favorite sport, or learning how to make that delicious dish, Google is our #1 source of information right away. now. One way to use some of the information that Google has about us to our benefit is to use Google Trends. It facilitates discovering trends and analyzing the behavior of our customers and users in general. Google Trends is one of the best tools for knowledge discovery and shows in real time (or near) the relevance of a topic, at least in terms of web searches and interest. public interest. In this project, we will see how to extract data from Google Trends in R environment and how we can play with this data, creating interesting visualizations.

Technology/Tool used: -
Google Colab, R programming,
gtrendsR, dplyr, patchwork

Agnel Charities'

Fr. C. Rodrigues Institute of Technology, Vashi, Navi-Mumbai
Department of Computer Engineering
Second Half of 2022

Important Code Snippets: -

```
[1] system("apt-get -y update")
    system("apt-get install -y libudunits2-dev libgdal-dev libgeos-dev libproj-dev")
```

```
install.packages("gtrendsR")
install.packages("dplyr")
install.packages("patchwork")
install.packages("ggforce")
install.packages("ggrepel")
install.packages("spData")
install.packages("sf")
install.packages("tmap")
```

```
Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)

also installing the dependencies 'Rcpp', 'BH', 'anytime'


Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)

Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)
```

```
library(gtrendsR)
library(dplyr)
library(ggplot2)
library(patchwork)
library(ggforce)
library(ggrepel)
library(spData)
library(tmap)
library(sf)
```

```
Attaching package: 'dplyr'


The following objects are masked from 'package:stats':

    filter, lag


The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union


To access larger datasets in this package, install the spDataLarge
package with: `install.packages('spDataLarge',
repos='https://nowosad.github.io/drat/', type='source')`

Linking to GEOS 3.6.2, GDAL 2.2.3, PROJ 4.9.3; sf_use_s2() is TRUE
```
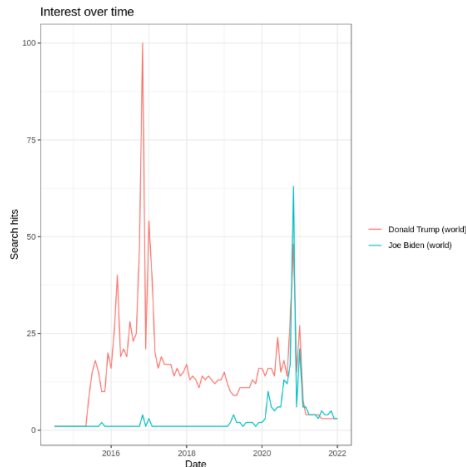
```
[4] Sys.setenv(TZ = "UTC")
```

```
[5] keyword <- c("Donald Trump", "Joe Biden")
    time <- "2014-06-30 2022-01-01"
    geo = ""
```

```r
search <- gtrends(keyword = keyword, time = time, geo = geo)
plot(search)
```



```r
time_trend<-search$interest_over_time %>%
  dplyr::mutate(hits=ifelse(hits=="<1",0.5,as.numeric(hits)),
                date=as.Date(date))

head(time_trend)
```

```
Warning message in ifelse(hits == "<1", 0.5, as.numeric(hits)):
"NAs introduced by coercion"
```

A data.frame: 6 × 7

	date	hits	keyword	geo	time	gprop	category
	<date>	<dbl>	<chr>	<chr>	<chr>	<chr>	<int>
1	2014-07-01	1	Donald Trump	world	2014-06-30 2022-01-01	web	0
2	2014-08-01	1	Donald Trump	world	2014-06-30 2022-01-01	web	0
3	2014-09-01	1	Donald Trump	world	2014-06-30 2022-01-01	web	0
4	2014-10-01	1	Donald Trump	world	2014-06-30 2022-01-01	web	0
5	2014-11-01	1	Donald Trump	world	2014-06-30 2022-01-01	web	0
6	2014-12-01	1	Donald Trump	world	2014-06-30 2022-01-01	web	0

```r
plot_nosmooth<-ggplot(time_trend, aes(x=date, y=hits, colour = keyword)) +
  geom_line() +
  labs(title="No Smoothing") +
  theme(legend.position = "none")

plot_smooth_20<-ggplot(time_trend, aes(x=date, y=hits, colour = keyword)) +
  geom_smooth(formula = y ~ x, method="loess",span=0.2, se=FALSE) +
  labs(title="smoothing factor = 0.2")+
  theme(legend.position = "none")

plot_smooth_40<-ggplot(time_trend, aes(x=date, y=hits, colour = keyword)) +
  geom_smooth(formula = y ~ x, method="loess",span=0.4, se=FALSE) +
  labs(title="smoothing factor = 0.4")+
  theme(legend.position = "none")

plot_smooth_60<-ggplot(time_trend, aes(x=date, y=hits, colour = keyword)) +
  geom_smooth(formula = y ~ x, method="loess",span=0.6, se=FALSE) +
  labs(title="smoothing factor = 0.6")+
  theme(legend.position = "none")


(plot_nosmooth + plot_smooth_20) / (plot_smooth_40 + plot_smooth_60)
```
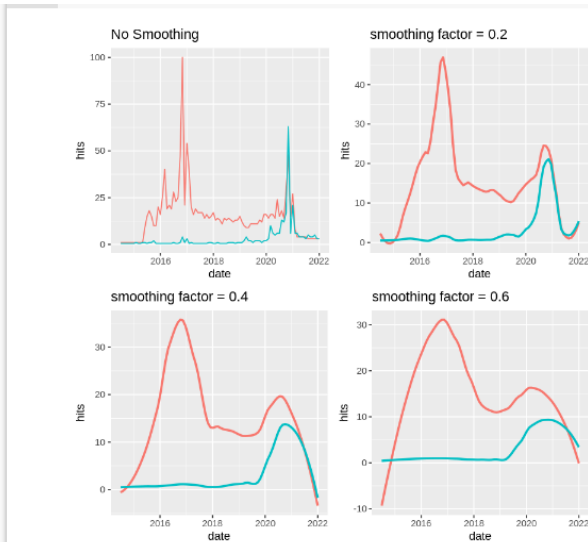
```
[ ]  time_trend<-iot2020 %>%
        dplyr::mutate(hits=ifelse(hits=="<1",0.5,as.numeric(hits)),
                      date=as.Date(date))

     head(time_trend)
```

```
Warning message in ifelse(hits == "<1", 0.5, as.numeric(hits)):
"NAs introduced by coercion"
```

A data.frame: 6 × 7

|	date	hits	keyword	geo	time	gprop	category
	<date>	<dbl>	<chr>	<chr>	<chr>	<chr>	<int>
1	2019-12-01	0.0	Covid	world	2019-11-30 2022-08-01	web	0
2	2019-12-08	0.5	Covid	world	2019-11-30 2022-08-01	web	0
3	2019-12-15	0.0	Covid	world	2019-11-30 2022-08-01	web	0
4	2019-12-22	0.0	Covid	world	2019-11-30 2022-08-01	web	0
5	2019-12-29	0.0	Covid	world	2019-11-30 2022-08-01	web	0
6	2020-01-05	0.5	Covid	world	2019-11-30 2022-08-01	web	0

Screens shots: -

```
iot2020 %>%
  ggplot(aes(x = date,
             y = hits,group=keyword,
             color = keyword))  +
  theme_bw()+
  labs(title = "Google Web searches for 'Covid'",
       caption = "Obs: 3/22 was the day with the most searches",
       x= NULL, y = "Interest")+
  ggforce::facet_zoom(xlim = c(as.POSIXct(as.Date("2021-12-15")),as.POSIXct(as.Date("2022-02-12")))) +
  geom_smooth(span=0.1,se=FALSE) + geom_vline(xintercept = as.POSIXct(as.Date("2022-01-01")),color = "red", lwd = 0.5,linetype="dashed")+
  theme(legend.position = "none") +
  geom_point(color="black")+
  geom_label_repel(data = subset(iot2020, hits == 100),
                   aes(label = as.character(date)),
                   size = 5,
                   box.padding = unit(0.35, "lines"),
                   point.padding = unit(0.3, "lines"))
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```
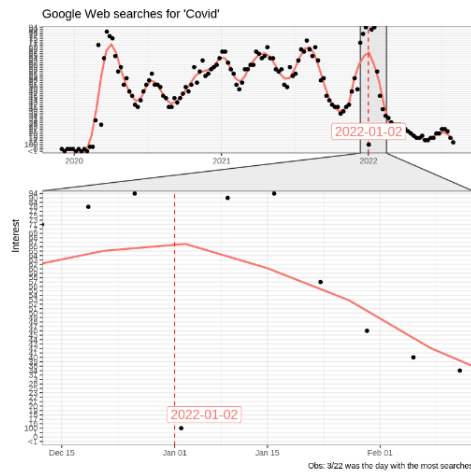


```
trends$related_queries %>%
  filter(related_queries=="top") %>%
  mutate(value=factor(value,levels=rev(as.character(value))),
         subject=as.numeric(subject)) %>%
  top_n(10,value) %>%
  ggplot(aes(x=value,y=subject,fill="red")) +
  geom_bar(stat='identity',show.legend = F) +
  coord_flip() + labs(title="Queries most related with 'covid'")
```