BOA Run Segment Google Trends Searches

MM

9/22/2021

```
knitr::opts_chunk$set(echo = TRUE)
```

Load required packages:

```
library(tinytex)
library(gtrendsR)
library(tidyverse)
library(lubridate)
library(ggplot2)
library(gridExtra)
library(readr)
library(gtrendsR)
library(purrr)
```

Google Trends Keyword Data:

- Normalized to the highest search volume for each keyword within the time frame analyzed
- Search "interest" for each keyword scaled between 0 - 100
- Valuable for interest in specific terms over time, but not really useful for comparing the absolute popularity of different search terms relative to one another

Pros & cons of using this scraping function in R vs. the Goole Trends interface:

- Pro: Can scrape more than the Google-Trends-interface limit of 5 keywords
- Con: Keywords are normalized to their own maximal popularity across the time frame analyzed, rather than normalized to all keywords in a given search

Write function to scrape data from Google Trends for list of keywords:

```
# The function wrap all the arguments of the gtrendR::trends function and return only the interest_over_time
googleTrendsData <- function (keywords) {

  # Set the geographic region, time span, etc.
  country <- c('US')
  time <- ('2016-01-01 2021-09-21')
  channel <- 'web'

  trends <- gtrends(keywords,
                    gprop = channel,
                    geo = country,
                    time = time )

  results <- trends$interest_over_time
  }
```

Load keywords list (.csv file):

```
kwlist <- readLines("KWlist_BOARunning.csv")
```

```
## Warning in readLines("KWlist_BOARunning.csv"): incomplete final line found on
## 'KWlist_BOARunning.csv'
```

```
kwlist
```

```
##  [1] "BOA running shoes"       "BOA running"
##  [3] "BOA dial"                "BOA dials"
##  [5] "BOA laces"               "BOA lace"
##  [7] "BOA trail"               "BOA trail running"
##  [9] "BOA trail running shoes" "BOA racing"
## [11] "BOA racing shoes"        "BOA performance"
## [13] "BOA performance fit"     "BOA wrap"
## [15] "BOA wrap running"        "BOA wrap running shoes"
```

Run function on keywords list:

```
# googleTrendsData function is executed over the kwlist
output <- map_df(1:n * kwlist,
                 ~( googleTrendsData ) )

# Download the dataframe "output" as a .csv file
write.csv(output, "BOARunning_Trends_US.csv")
head(output)
```

	date <dttm>	hits <dbl>	keyword <chr>	geo <chr>	time <chr>	gprop <chr>	category <chr>
1	2016-01-01	0	BOA running shoes	US	2016-01-01 2021-09-21	web	0
2	2016-02-01	0	BOA running shoes	US	2016-01-01 2021-09-21	web	0
3	2016-03-01	0	BOA running shoes	US	2016-01-01 2021-09-21	web	0
4	2016-04-01	0	BOA running shoes	US	2016-01-01 2021-09-21	web	0
5	2016-05-01	23	BOA running shoes	US	2016-01-01 2021-09-21	web	0
6	2016-06-01	24	BOA running shoes	US	2016-01-01 2021-09-21	web	0

6 rows

If data scraping was already performed, load CSV data for US or WORLD instead:

```
setwd('/Users/melissamasso/Documents/Data Sets')
  # US
  output <- read.csv('BOARunning_Trends_US.csv')
  outputDate <- as.Date(output$date)
  #World
  output <- read.csv('BOARunning_Trends_WORLD.csv')
  outputDate <- as.Date(output$date)
```

Extrapolate total 2021 searches based on Jan - Sept of 2021:

```
# Group data into months#
monthdat <- output %>%
  group_by(keyword,month = lubridate::floor_date(date, "month")) %>%
  summarize(month_sum = sum(hits))
```

```
## `summarise()` has grouped output by 'keyword'. You can override using the `.groups` argument.
```

```
monthdatSmonth <- month(monthdat$month)
head(monthdat)
```

keyword <chr>	month <dbl>	month_sum <int>
BOA dial	1	17
BOA dial	2	18
BOA dial	3	0
BOA dial	4	0
BOA dial	5	34
BOA dial	6	18

6 rows

```
# Group data yearwise
yeardat <- output %>%
  group_by(keyword,year = lubridate::floor_date(date, "year")) %>%
  summarize(year_sum = sum(hits))
```

```
## `summarise()` has grouped output by 'keyword'. You can override using the `.groups` argument.
```

```
yeardatSyear <- year(yeardat$year)
head(yeardat)
```

keyword <chr>	year <dbl>	year_sum <int>
BOA dial	2016	242
BOA dial	2017	143
BOA dial	2018	136
BOA dial	2019	174
BOA dial	2020	201
BOA dial	2021	249

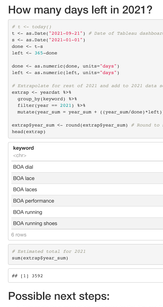
6 rows

First, is the data for the remaining months (Oct, Nov, Dec) typically similar to the first 9 months of the year?

```
# Visualize data monthwise
bykeyword <- ggplot(monthdat,aes(x=month,y=month_sum,color=keyword)) + geom_point(aes(fill=keyword)) +
  scale_x_discrete(name = "Month", limits=c("1","2","3","4","5","6","7","8","9","10","11","12")) +
  geom_smooth(method='loess',aes(fill=keyword),alpha = 0.1) + theme(legend.position='none') +
  scale_y_continuous("Total Hits per Month",limits = c(0, 100))

total <- ggplot(monthdat,aes(x=month,y=month_sum)) + geom_point(aes(fill=keyword)) +
  scale_x_discrete(name = "Month", limits=c("1","2","3","4","5","6","7","8","9","10","11","12")) +
  geom_smooth(method='loess',alpha = 0.7) + theme(legend.position='none') +
  scale_y_continuous("Total Hits per Month",limits = c(0, 100))

grid.arrange(bykeyword,total,ncol=2)
```

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


```
# Split into two groups (before Sept & Sept and after)
avgs <- monthdat %>%
  group_by(month > 9)
colnames(avgs) <- c("keyword", "month","month_sum","future")

ggplot(avgs,aes(x=month>9,y=month_sum)) + geom_boxplot(outlier.alpha = 0.1,lwd=1,width=0.25) + geom_point(aes(col
or=keyword),position = position_jitterdodge(dodge.width=0.01,jitter.width = 0.2)) + theme(legend.position='none')
+ theme_classic() + scale_x_discrete(name="Past vs. Future Months",labels = c("PAST" = "Jan - Aug","FUTU" = "Sept -
Dec")) + scale_y_continuous("Total Hits per Month")
```




```
# Test for a major difference with a linear regression
summary(lm(data = avgs,formula = month_sum ~ future))
```

```
##
## Call:
## lm(formula = month_sum ~ future, data = avgs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.560 -26.177 -3.177  16.973  73.823
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   26.177      1.295  20.218  <2e-16 ***
## futureTRUE    -3.382      1.293  -2.617   0.0114 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.37 on 550 degrees of freedom
## Multiple R-squared:  0.0062,    Adjusted R-squared:  0.00449
## F-statistic: 3.487 on 1 and 550 DF,  p-value: 0.06239
```

```
# No significant difference between past & future months
```

How many days left in 2021?

```
# t <- today()
t <- as.Date('2021-09-21') # Date of Tableau dashboard creation
e <- as.Date('2021-01-01')
done <- t-e
left <- 365-done

done <- as.numeric(done, units="days")
left <- as.numeric(left, units="days")

# Extrapolate for rest of 2021 and add to 2021 data so far for each keyword
extrap <- yeardat %>%
  group_by(keyword) %>%
  filter(year == 2021) %>%
  mutate(year_sum = year_sum + ((year_sum/done)*left))

extrap$year_sum <- round(extrap$year_sum) # Round to nearest integer
head(extrap)
```

keyword <chr>	year <dbl>	year_sum <dbl>
BOA dial	2021	346
BOA lace	2021	609
BOA laces	2021	597
BOA performance	2021	169
BOA running	2021	693
BOA running shoes	2021	412

6 rows

```
# Estimated total for 2021
sum(extrap$year_sum)
```

```
## [1] 3592
```

Possible next steps:

- Extrapolate the number of hits anticipated for Oct, Nov and Dec for each keyword, using the trendlines created with previous years' data and the data for Jan - Sept of 2021
- Investigate whether certain keywords have a cyclic trend across the months of a year (i.e., do searches for "BOA + trail" increase in summer or winter, or stay relatively constant across the year?)

Now let's examine searches for a specific shoe with and without BOA terms, the La Sportiva Cyklon:

```
# Load keywords list (.csv file)
setwd('Users/melissamasso/Documents/Data Sets')
kwlist <- readLines("KWlist_BOACyklon.csv")

# The function wrap all the arguments of the gtrendR::trends function and return only the interest_over_time
googleTrendsData <- function (keywords) {

  # Set the geographic region, time span, etc.
  country <- c('')
  time <- ('2016-01-01 2021-09-21')
  channel <- 'web'

  trends <- gtrends(keywords,
                    gprop = channel,
                    geo = country,
                    time = time )

  results <- trends$interest_over_time
  }

# googleTrendsData function is executed over the kwlist
output <- map_df(1:n * kwlist,
                 ~( googleTrendsData ) )

# Download the dataframe "output" as a .csv file
write.csv(output, "BOARunning_Cyklon.csv")
```

If data scraping was already performed, load CSV data for Cyklon instead:

```
  # Cyklon
  output <- read.csv('BOARunning_Trends_Cyklon.csv')
  outputDate <- as.Date(output$date)
```

Connect Tableau workbook to CSV data sources