

# Assignment 2

Due at 11:59pm on October 3.

Leng Seong Che; Bozhou (Peter) Tan

You may work in pairs or individually for this assignment. Make sure you join a group in Canvas if you are working in pairs. Turn in this assignment as an HTML or PDF file to ELMS. Make sure to include the R Markdown or Quarto file that was used to generate it.

```
library(tidyverse)
library(gtrendsR)
library(censusapi)
```

In this assignment, you will pull from APIs to get data from various data sources and use your data wrangling skills to use them all together. You should turn in a report in PDF or HTML format that addresses all of the questions in this assignment, and describes the data that you pulled and analyzed. You do not need to include full introduction and conclusion sections like a full report, but you should make sure to answer the questions in paragraph form, and include all relevant tables and graphics.

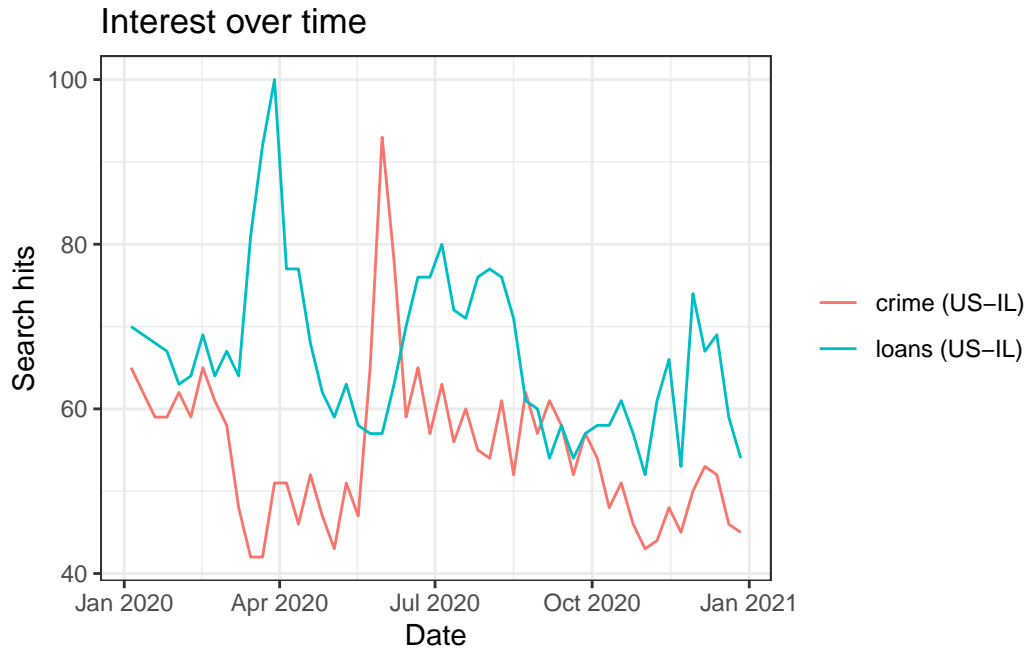
Whenever possible, use piping and `dplyr`. Avoid hard-coding any numbers within the report as much as possible.

## Pulling from APIs

Our first data source is the Google Trends API. Suppose we are interested in the search trends for `crime` and `loans` in Illinois in the year 2020. We could find this using the following code:

```
res = gtrends(c("crime", "loans"),
              geo = "US-IL",
              time = "2020-01-01 2020-12-31",
              low_search_volume = TRUE)

plot(res)
```



Answer the following questions for the keywords “crime” and “loans”.

- Find the mean, median and variance of the search hits for the keywords.

```
# transfer the data into tibble
rest = as.tibble(res$interest_over_time)
```

Warning: `as.tibble()` was deprecated in tibble 2.0.0.  
 i Please use `as\_tibble()` instead.  
 i The signature and semantics have changed, see `?as\_tibble`.

```
# find the mean, median and variance of the search hits
library(dplyr)
library(knitr)

descriptive = rest %>%
  group_by(keyword) %>%
  summarise(n = n(),
            mean = mean(hits),
            median = median(hits),
            variance = var(hits))
```

```
kable(descriptive)
```

keyword	n	mean	median	variance
crime	52	55.00000	54	86.43137
loans	52	66.48077	65	95.39178

- Which cities (locations) have the highest search frequency for **loans**? Note that there might be multiple rows for each city if there were hits for both “crime” and “loans” in that city. It might be easier to answer this question if we had the search hits info for both search terms in two separate variables. That is, each row would represent a unique city.
- Is there a relationship between the search intensities between the two keywords we used?

Repeat the above for keywords related to covid. Make sure you use multiple keywords like we did above. Try several different combinations and think carefully about words that might make sense within this context.

## Google Trends + ACS

Now lets add another data set. The `censusapi` package provides a nice R interface for communicating with this API. However, before running queries we need an access key. This (easy) process can be completed here:

[https://api.census.gov/data/key\\_signup.html](https://api.census.gov/data/key_signup.html)

Once you have an access key, store this key in the `cs_key` object. We will use this object in all following API queries.

```
cs_key <- "your key here"
```

In the following, we request basic socio-demographic information (population, median age, median household income, income per capita) for cities and villages in the state of Illinois.

```
acs_il <- getCensus(name = "acs/acs5",  
  vintage = 2020,  
  vars = c("NAME",  
    "B01001_001E",  
    "B06002_001E",  
    "B19013_001E",  
    "B19301_001E"),  
  region = "place:*",
```

```

    regionin = "state:17",
    key = cs_key)
head(acs_il)

```

Convert values that represent missings to NAs.

```

acs_il[acs_il == -666666666] <- NA

```

Now, it might be useful to rename the socio-demographic variables (B01001\_001E etc.) in our data set and assign more meaningful names.

```

acs_il <-
  acs_il %>%
  rename(pop = B01001_001E,
         age = B06002_001E,
         hh_income = B19013_001E,
         income = B19301_001E)

```

It seems like we could try to use this location information listed above to merge this data set with the Google Trends data. However, we first have to clean `NAME` so that it has the same structure as `location` in the search interest by city data. Add a new variable `location` to the ACS data that only includes city names.

Answer the following questions with the “crime” and “loans” Google trends data and the ACS data.

- First, check how many cities don’t appear in both data sets, i.e. cannot be matched. Then, create a new data set by joining the Google Trends and the ACS data. Keep only cities that appear in both data sets.
- Compute the mean of the search popularity for both keywords for cities that have an above average median household income and for those that have an below average median household income. When building your pipe, start with creating the grouping variable and then proceed with the remaining tasks. What conclusions might you draw from this?
- Is there a relationship between the median household income and the search popularity of the Google trends terms? Describe the relationship and use a scatterplot with `qplot()`.

Repeat the above steps using the covid data and the ACS data.