Assignment 2

Due at 11:59pm on October 1.

Sagnik Chakravarty & Namit Shrivastava

Github link

Please find our work at the following link Github Link

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)
library(ggplot2)
```

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.

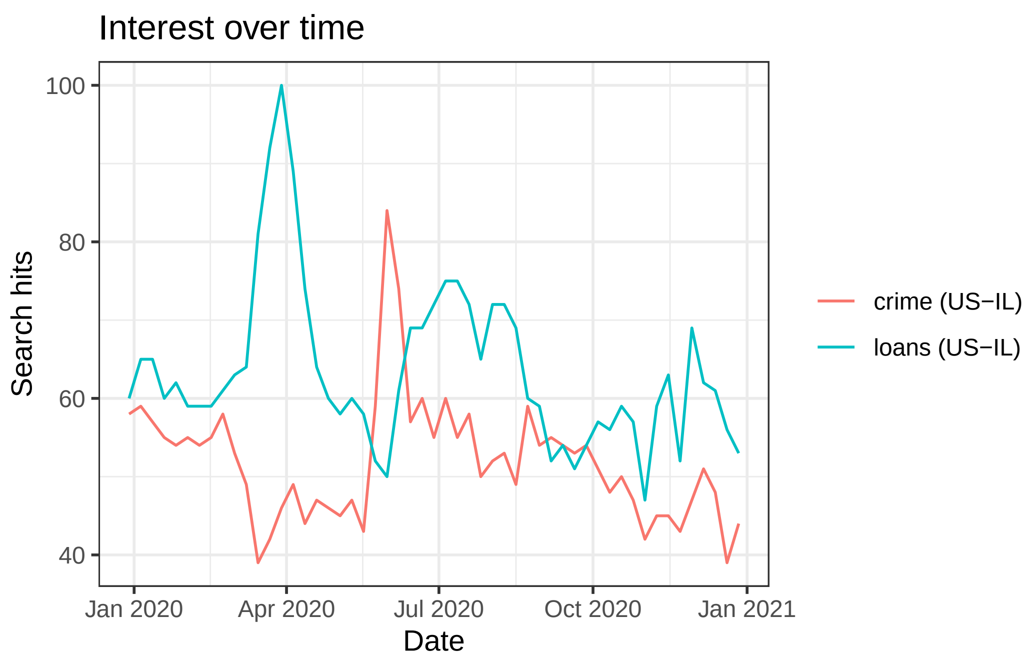
Whenever possible, use piping and `dplyr`. Avoid hard-coding any numbers within the report as much as possible.

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",
```

```r
                     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.
- 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?

```r
# Question 1
head(res$interest_over_time, n = 5)
```

```
        date hits keyword   geo                  time gprop category
1 2019-12-29   58   crime US-IL 2020-01-01 2020-12-31   web        0
2 2020-01-05   59   crime US-IL 2020-01-01 2020-12-31   web        0
```

```
3 2020-01-12   57   crime US-IL 2020-01-01 2020-12-31   web        0
4 2020-01-19   55   crime US-IL 2020-01-01 2020-12-31   web        0
5 2020-01-26   54   crime US-IL 2020-01-01 2020-12-31   web        0
```

```r
sum_res_all <- res$interest_over_time %>% group_by(keyword) %>%
  summarize(mean = mean(hits),
            median = median(hits),
            variance = var(hits))
sum_res_all
```

```
# A tibble: 2 x 4
  keyword  mean median variance
  <chr>   <dbl>  <int>    <dbl>
1 crime    51.9     52     62.2
2 loans    63.5     61    109.
```

From the line graph its clear that on average loans has a higher search volume at Illinois between Jan 2020 to Dec 2020, the summary statistics also proves this point as we can see that the mean search volume for loans is greater than that of crime, the median is also higher one thing of note is $\mu_{loans} > median_{loans}$ we can say that loan is right skewed while $\mu_{crime} \approx median_{crime}$ hence crime is symmetrically distributed. The variance on the other hand for loan is much greater than that of crime that is the data is more scattered and they differ highly from the central tendencies

```r
# Question 2
freq_res_cities <- res$interest_by_city %>%select(c(location, keyword, hits)) %>% filter(h

head(arrange(freq_res_cities, desc(hits)), n = 10)
```

```
           location keyword hits
1     Buffalo Grove   crime  100
2         Long Lake   loans  100
3              Anna   crime   97
4          Oak Lawn   loans   95
5          Rosemont   loans   95
6  East Saint Louis   crime   93
7         Coal City   loans   90
8      Ford Heights   loans   88
9  East Saint Louis   loans   87
10           Dolton   loans   84
```

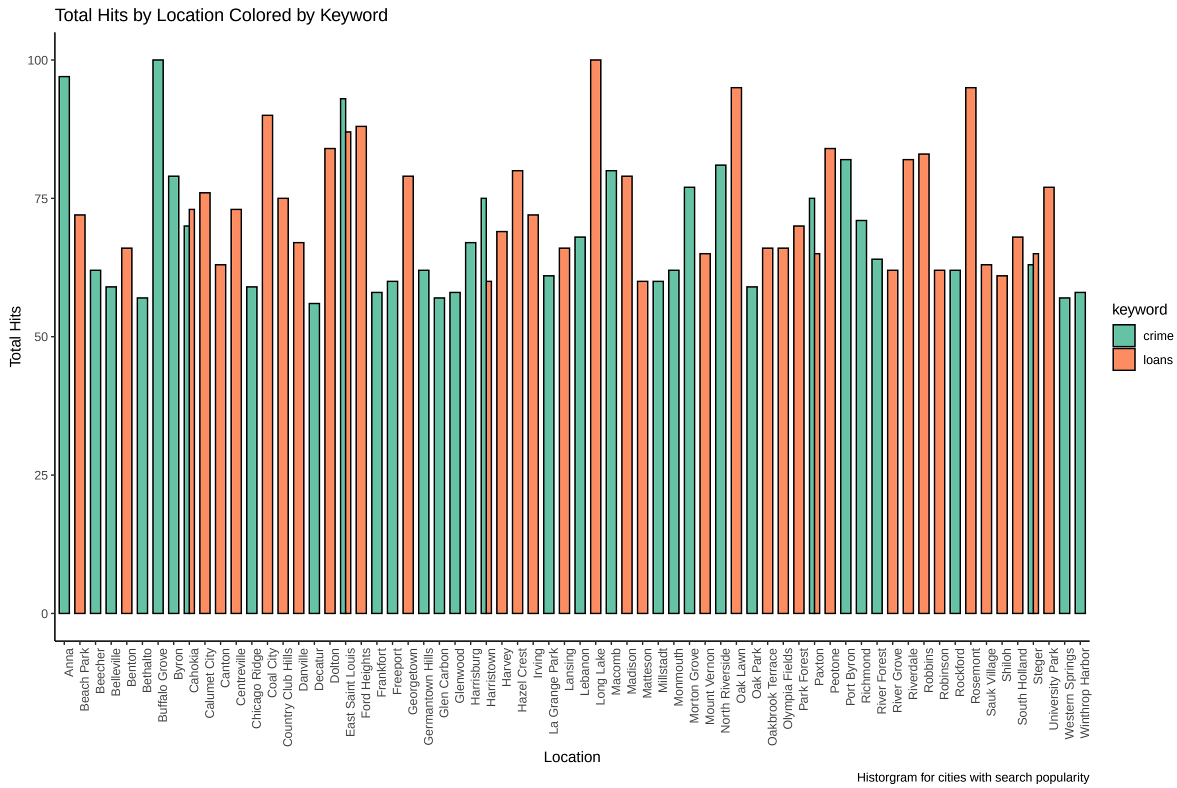
```r
freq_res_city_spread <- spread(freq_res_cities, key = keyword, value = hits)%>%
  mutate(across(where(is.numeric), ~ replace_na(.,0)))
head(freq_res_city_spread, n = 5)
```

```
    location crime loans
1       Anna    97     0
2 Beach Park     0    72
3    Beecher    62     0
4 Belleville    59     0
5     Benton     0    66
```

```r
nrow(freq_res_city_spread)
```

```
[1] 66
```

```r
ggplot(freq_res_cities, aes(x = location, y = hits, fill = keyword))+
  geom_bar(stat = 'identity', position = 'dodge', color = 'black', width = 0.65)+
  labs(title = "Total Hits by Location Colored by Keyword",
       x = "Location",
       y = "Total Hits",
       caption = 'Historgram for cities with search popularity') +
  theme_classic()+
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_brewer(palette = "Set2")
```




```r
il_both <- freq_res_city_spread %>% filter(crime > 0 & loans > 0)
nrow(il_both)
```

```
[1] 5
```

```r
il_both %>%
  arrange(desc(crime))
```

```
           location crime loans
1 East Saint Louis    93    87
2       Harristown    75    60
3           Paxton    75    65
4          Cahokia    70    73
5           Steger    63    65
```

```r
il_both %>%
  arrange(desc(loans))
```

```
          location crime loans
1 East Saint Louis    93    87
2          Cahokia    70    73
3           Paxton    75    65
4           Steger    63    65
5       Harristown    75    60
```

```r
il_both %>%
  mutate(avg_hits = (crime+loans)/2) %>%
  arrange(desc(avg_hits))
```

```
          location crime loans avg_hits
1 East Saint Louis    93    87     90.0
2          Cahokia    70    73     71.5
3           Paxton    75    65     70.0
4       Harristown    75    60     67.5
5           Steger    63    65     64.0
```

As we can see there are 66 cities in Illinois where the keyword loans or crime were searched atleast once, out of those 66 only 6 cities searched for both the keyword crime and loans, we can also see **Anna** has the highest hit for crime at 100 while **Long Lake** has the higher for loans at 100, while **East Saint Louis** city has the highest search volume in crime loans and average number of hits at 75, 87 and 81 respectively where both the keyword has been searched for.

```r
# Question 3
corel <- cor(freq_res_city_spread$crime, freq_res_city_spread$loans)
cat("The correlation between crime and loan:\t", corel)
```

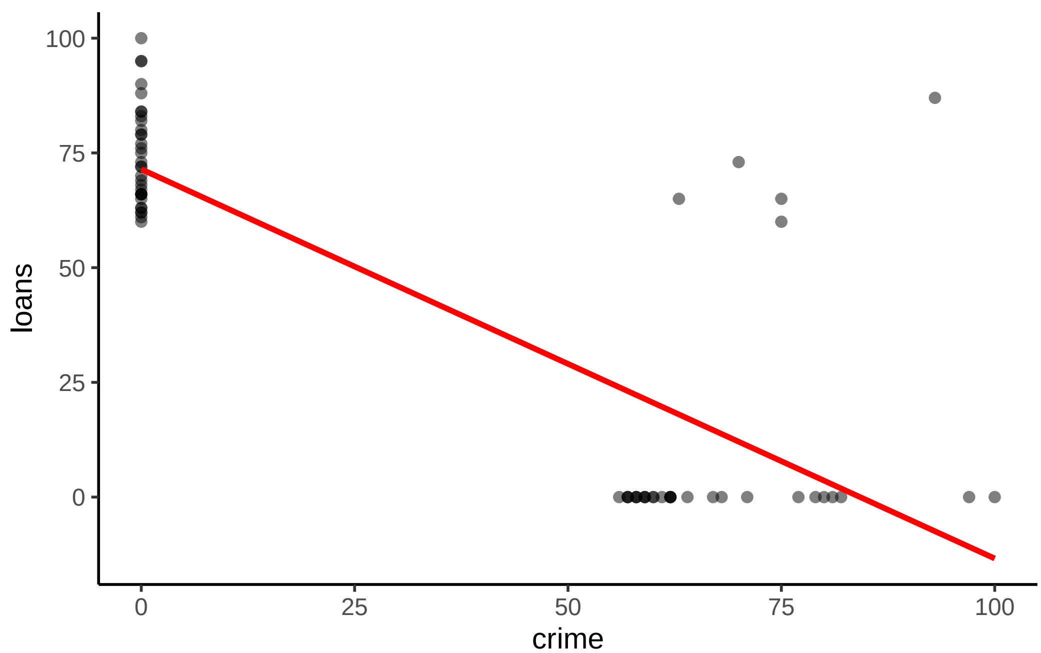
```
The correlation between crime and loan:  -0.7960553
```

```r
cor.test(freq_res_city_spread$crime, freq_res_city_spread$loans)
```

```
	Pearson's product-moment correlation

data:  freq_res_city_spread$crime and freq_res_city_spread$loans
t = -10.522, df = 64, p-value = 1.365e-15
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8703889 -0.6862417
sample estimates:
       cor
-0.7960553
```

```r
ggplot(freq_res_city_spread, aes(crime, loans))+
  geom_point(color = 'black', alpha = 0.5)+
  geom_smooth(method = 'lm', color = 'red', se =FALSE)+
  theme_classic()
```

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



We can see that crime and loans keyword are strongly negatively correlated at -0.79, also the correlation test shows that p_value < 0.05 for which we reject $H_0 : \rho = 0$ hence we reject H_0 at 95% confidence interval and the correlation coefficient lies between -0.89 and -0.67.

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

Once you have an access key, save it as a text file, then read this key in the `cs_key` object. We will use this object in all following API queries. Note that I called my text file `census-key.txt` – yours might be different!

```
cs_key <- read_file("census-key.txt")
```

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. Documentation for the 5-year ACS API can be found here: https://www.census.gov/data/developers/data-sets/acs-5year.html. The information about the variables used here can be found here: https://api.census.gov/data/2022/acs/acs5/variables.html.

```
if (!require(gtrendsR)) install.packages("censusapi")
library(censusapi)
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)
```

```
  state place                          NAME B01001_001E B06002_001E B19013_001E
1    17 15261 Coatsburg village, Illinois         180        35.6       55714
2    17 15300    Cobden village, Illinois        1018        44.2       38750
3    17 15352      Coffeen city, Illinois         640        33.4       35781
```

```
4    17 15378   Colchester city, Illinois        1347         42.2        43942
5    17 15469    Coleta village, Illinois         230         27.7        56875
6    17 15495    Colfax village, Illinois        1088         32.5        58889
  B19301_001E
1       27821
2       19979
3       26697
4       24095
5       23749
6       24861
```

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 %>%
  dplyr :: rename(pop = B01001_001E,
         age = B06002_001E,
         hh_income = B19013_001E,
         income = B19301_001E)
head(acs_il, n = 5)
```

```
  state place                        NAME  pop  age hh_income income
1    17 15261 Coatsburg village, Illinois  180 35.6     55714  27821
2    17 15300    Cobden village, Illinois 1018 44.2     38750  19979
3    17 15352      Coffeen city, Illinois  640 33.4     35781  26697
4    17 15378   Colchester city, Illinois 1347 42.2     43942  24095
5    17 15469    Coleta village, Illinois  230 27.7     56875  23749
```

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()`.

q1 <- qplot(hh_income, crime, data = joined_il,

color = I("red"), shape = I(16), size = I(3),

xlab = "Median Household Income",

ylab = "Search Popularity",

main = "Scatterplot of Median Household Income vs. Search Popularity")

Overlay the scatterplot for loans

q2 <- qplot(hh_income, loans, data = joined_il,

color = I("blue"), shape = I(17), size = I(3),

add = TRUE)

Print the combined plot

print(q1 + q2)

```
names(acs_il)
```

```
[1] "state"     "place"     "NAME"      "pop"       "age"       "hh_income"
[7] "income"
```

```
acs_il_copy <- acs_il
acs_il_copy$NAME <- unlist(lapply(strsplit(acs_il_copy$NAME, ','), function(x) x[1]))
names(acs_il_copy)[names(acs_il_copy) == "NAME"] <-'location'

# Trim white spaces from the location column and
# remove the name village and city from the acs_il dataframe
acs_il_copy$location <- str_trim(str_replace(acs_il_copy$location,
                                             "\\s*(village|city)$", ""),
```

```
                                         side = "both")
acs_il_copy$location <- tolower(acs_il_copy$location)
freq_res_city_spread$location <- tolower(freq_res_city_spread$location)
```

Before starting we will be doing some pre processing we changed the name of the column to NAME to location to facilitate joining, we also removed the illinois after the city name, and removed any leading or trailing white space along with “village” or “city” from the names, then we made the city name in both dataset in lower case.

```
# Question 1
cat("No of cities not appearing in both the dataset:\t",
    nrow(anti_join(acs_il_copy, freq_res_city_spread,
                   by = 'location')))
```

```
No of cities not appearing in both the dataset:  1402
```

```
joined_il <- inner_join(freq_res_city_spread, acs_il_copy,
                        by = 'location')
head(joined_il, n = 5)
```

```
     location crime loans state place   pop  age hh_income income
1        anna    97     0    17 01543  4149 42.4     36303  22455
2 beach park     0    72    17 04303 13433 35.4     71250  28292
3     beecher    62     0    17 04585  4443 42.0     86576  34290
4 belleville    59     0    17 04845 41256 38.1     52843  27896
5      benton     0    66    17 05300  6977 40.5     44795  27932
```

There are 1403 cities which appears in the Census data but not in the google trend data where atleast the keyword crime or loans where searched atleast once. the dataset joined_il contains the dataset after joining both the dataset using location as the primary key.

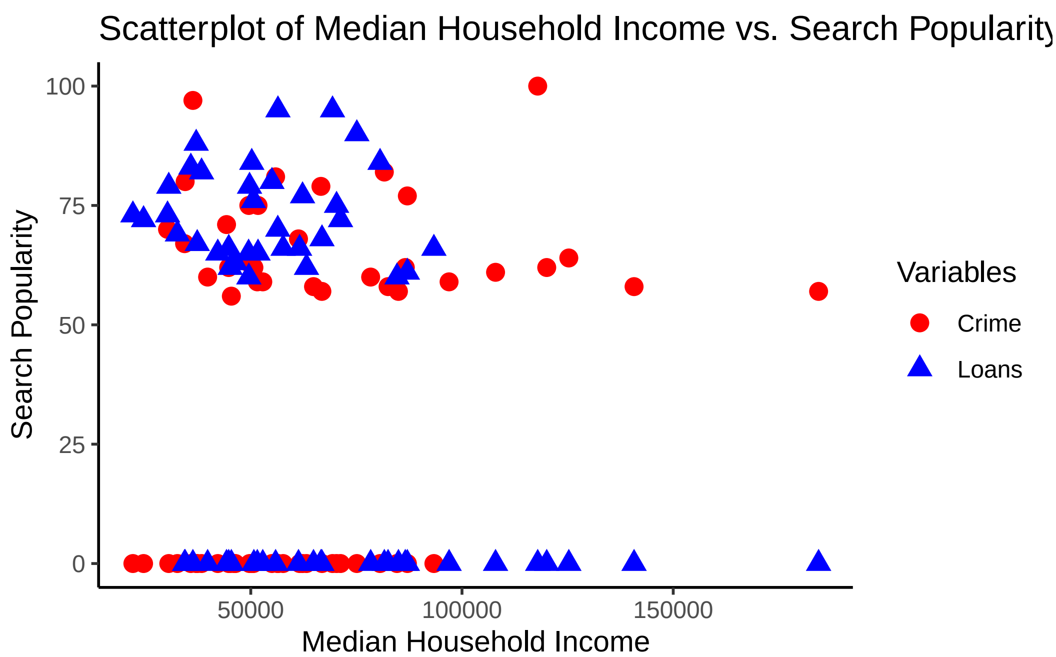
```
# Question 2

results <- joined_il %>%
  mutate(income_group = if_else(hh_income > median(acs_il$hh_income, na.rm =TRUE), 'high i
  group_by(income_group) %>%
  summarise(mean_crime = mean(crime),
            mean_loans = mean(loans)) %>%
  ungroup()
```

```r
print(results)
```

```
# A tibble: 2 x 3
  income_group mean_crime mean_loans
  <chr>             <dbl>      <dbl>
1 high income        37.3       31.4
2 low income         30.5       49.4
```

We can see that the mean crime at location with income higher than that of the median household income at all the states comes out as 34.4 and that of loan is 34.89 which are surprisingly very close, while the in the lower income area search for loan is more than search for crime at 46.63 and 32.25 respectively

```r
# Question 3
ggplot(joined_il) +
  geom_point(aes(x = hh_income, y = crime, color = "Crime"),
             shape = 16, size = 3) +
  geom_point(aes(x = hh_income, y = loans, color = "Loans"),
             shape = 17, size = 3) +
  labs(x = "Median Household Income",
       y = "Search Popularity",
       title = "Scatterplot of Median Household Income vs. Search Popularity") +
  scale_color_manual(name = "Variables",
                     values = c("Crime" = "red", "Loans" = "blue")) +
  theme_classic()
```




```
cor.test(joined_il$hh_income, joined_il$loans)
```

```
	Pearson's product-moment correlation

data:  joined_il$hh_income and joined_il$loans
t = -3.2943, df = 62, p-value = 0.001634
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5770224 -0.1548364
sample estimates:
       cor
-0.3859561
```

```
cor.test(joined_il$hh_income, joined_il$crime)
```

```
	Pearson's product-moment correlation

data:  joined_il$hh_income and joined_il$crime
```

```
t = 2.1708, df = 62, p-value = 0.03378
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.02136576 0.48021552
sample estimates:
      cor
0.2657792
```

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

Solution

over here our goal is simple scrape data for covid for the city of illinois for each city, then compare the house hold income in which cities have wore mask or not.

```r
library(tidyverse)
library(jsonlite)
```

```
Attaching package: 'jsonlite'

The following object is masked from 'package:purrr':

    flatten
```

```r
library(httr)

url_country <- GET("https://covidmap.umd.edu/api/country")
response <- content(url_country, as = 'text', type = "UTF-8")
```

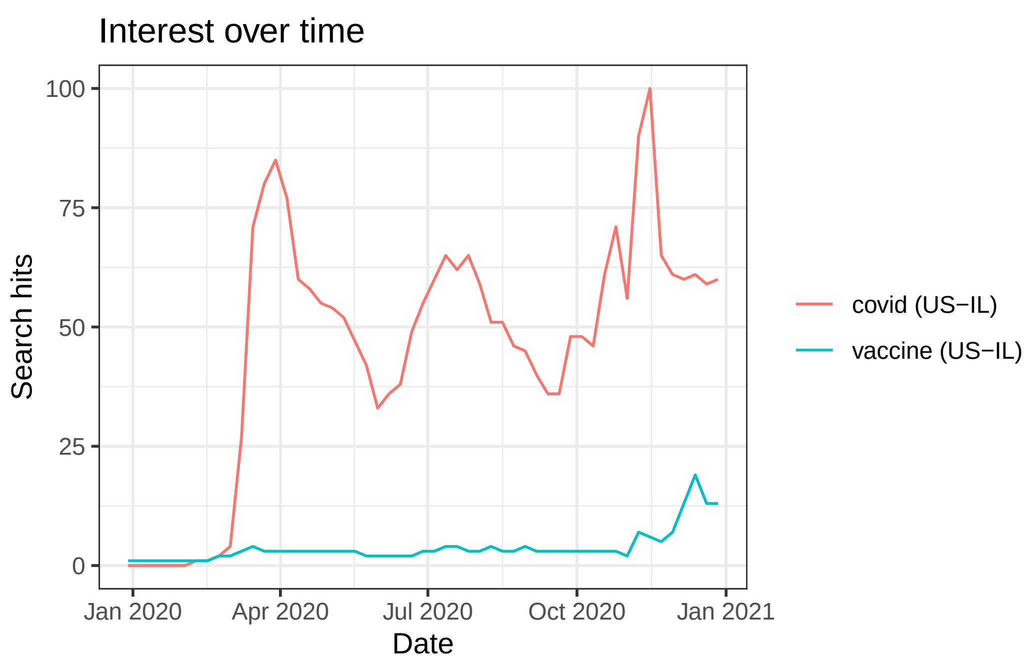
```
No encoding supplied: defaulting to UTF-8.
```

```r
country_list <- fromJSON(response, flatten = TRUE)$data
country_list %>%
  mutate(country = tolower(country)) %>%
  filter(country == 'united states') %>%
  pull(country)
```

```
character(0)
```

Well thats it as we can see there is no united states in the covid dataset, so lets use the google trends data for covid in usa and for the second variable lets search for say vaccine as its related to covid i feel like.

```r
library(censusapi)
library(ggplot2)
library(tidyr)
library(gtrendsR)
covid <- gtrends(c('covid', 'vaccine'),
                 geo = 'US-IL',
                 time = "2020-01-01 2020-12-31",
                 low_search_volume = TRUE)

plot(covid)
```



From the line chart we can see that covid has been searched way more than vaccine, which is as expected since 2020 is the hay days of covid, but in closer inspection we see that the trend of covid and vaccine are very similar when ever there is more interest in covid we have a proportional interest in vaccine, but the search for vaccine really started to have some steam as we were closer to the end of the year.

```r
sum_covid_all <- covid$interest_over_time %>%
                  group_by(keyword) %>%
                  mutate(hits = as.numeric(as.character(hits))) %>%
                  filter(!is.na(hits)) %>%
                  summarise(mean_hits = mean(hits, na.rm = TRUE),
                            variance = var(hits, na.rm = TRUE),
                            median_hits = median(hits, na.rm = TRUE))
sum_covid_all
```

```
# A tibble: 2 x 4
  keyword mean_hits variance median_hits
  <chr>       <dbl>    <dbl>       <dbl>
1 covid       47.6     613.           52
2 vaccine      3.75     11.8           3
```

We can see that covid has be searched way more and the data is spread way more than the search for vaccine from the mean median and variance, over here mean<median for covid suggesting a left skewdness in the dataset, but for vaccine mean is very close to median which suggest a symmetric distribution

```r
covid_data_freq <- covid$interest_by_city %>%
                    select(location, hits, keyword) %>%
                    mutate(hits = as.numeric(as.character(hits))) %>%
                    filter(hits>0)
covid_data_freq_spread <- spread(covid_data_freq, key = keyword, value = hits) %>%
                          mutate(across(where(is.numeric), ~ replace_na(.,0)))
head(covid_data_freq, n = 5)
```

```
  location hits keyword
1 Bartelso  100   covid
2 Oak Lawn   99   covid
3   Albany   95   covid
4   Geneva   95   covid
5 Winnetka   92   covid
```

```r
nrow(covid_data_freq)
```

```
[1] 271
```

```r
covid_data_freq %>%
  filter(keyword == 'covid') %>%
  arrange(desc(hits)) %>%
  head(n = 2)
```

```
  location hits keyword
1 Bartelso  100   covid
2 Oak Lawn   99   covid
```

```r
covid_data_freq %>%
  filter(keyword == 'vaccine') %>%
  arrange(desc(hits)) %>%
  head(n = 2)
```

```
        location hits keyword
1          Hurst  100 vaccine
2 Evergreen Park   54 vaccine
```

```r
covid_both <- covid_data_freq_spread %>% filter(vaccine > 0 & covid > 0)
head(covid_both, n = 5)
```

```
           location covid vaccine
1 Arlington Heights    74      34
2        Barrington    88      33
3  Barrington Hills    79      35
4         Brimfield    73      36
5       Bull Valley    88      38
```

```r
nrow(covid_both)
```

```
[1] 60
```

```r
covid_both %>%
  arrange(desc(covid)) %>%
  head()
```

```
        location covid vaccine
1       Oak Lawn    99      40
2       Wilmette    92      38
3       Winnetka    92      45
4 Evergreen Park    91      54
5     Northbrook    91      39
6 Willow Springs    91      42
```

```
covid_both %>%
  arrange(desc(vaccine)) %>%
  head()
```

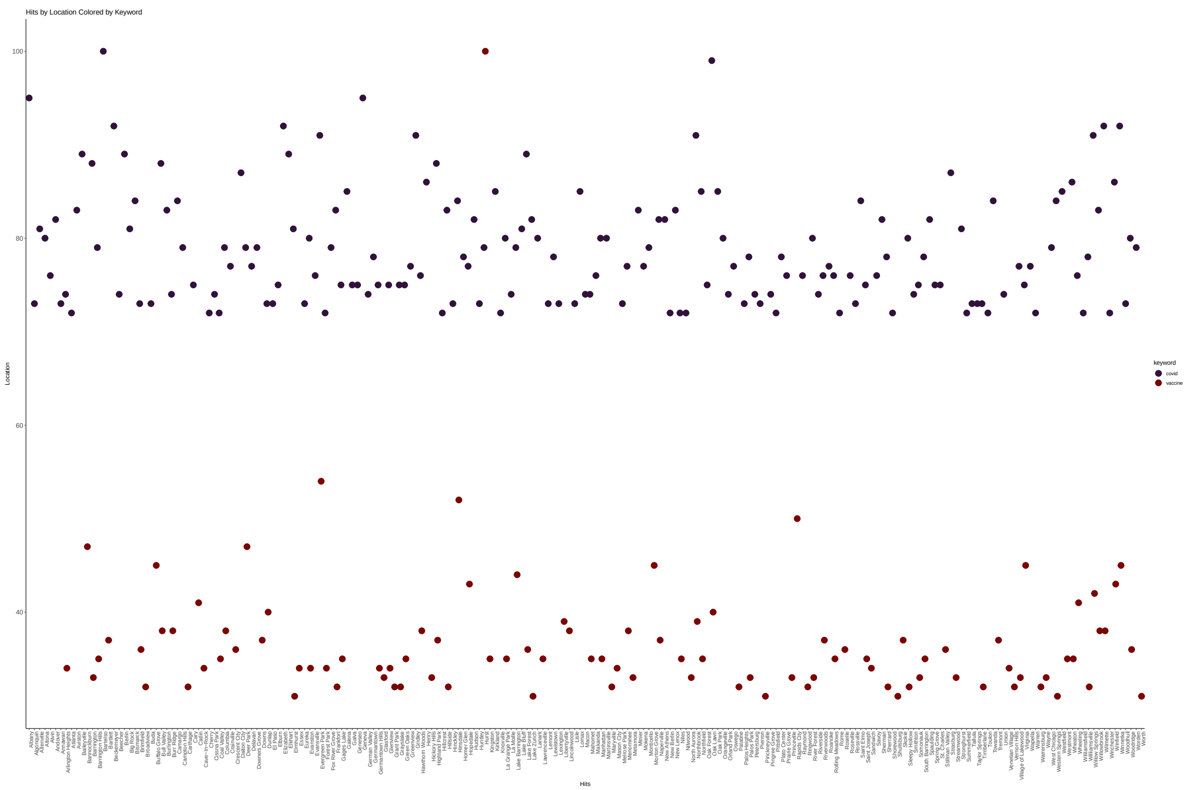
```
        location covid vaccine
1          Hurst    79     100
2 Evergreen Park    91      54
3       Hinsdale    84      52
4      Deer Park    79      47
5       Virginia    75      45
6       Winnetka    92      45
```

```
covid_both %>%
  mutate(avg_hits = (covid+vaccine)/2) %>%
  arrange(desc(avg_hits)) %>%
  head()
```

```
        location covid vaccine avg_hits
1          Hurst    79     100     89.5
2 Evergreen Park    91      54     72.5
3       Oak Lawn    99      40     69.5
4       Winnetka    92      45     68.5
5       Hinsdale    84      52     68.0
6 Willow Springs    91      42     66.5
```

We can see that there are 266 places which have searched for atleast Covid or vaccine on which 58 have searched for both the keyword **Bartelso** has the highest search for Covid while **Hurst** has the highest search for vaccine and also the place where Covid was searched atleast once, while **Oak Lawn** being the place with highest search for Covid where vaccine was also searched atleast once, **Hurst** still comes on top when it comes to the highest place with more average Covid and vaccine search

```r
ggplot(covid_data_freq, aes(y = hits, x = location, color = keyword)) +
  geom_point(position = position_dodge(width = 0.5), size = 5) +
  labs(
    title = "Hits by Location Colored by Keyword",
    x = "Hits",
    y = "Location",
    caption = 'Dot plot for cities with search popularity'
  ) +
  theme_classic() +
  theme(
    axis.text.x = element_text(angle = 90, hjust = 1, size = 10),
    axis.text.y = element_text(size = 12)
  ) +
  scale_color_viridis_d(option = 'H')
```




```r
covid_both$location <- tolower(covid_both$location)
joined_covid <- inner_join(acs_il_copy, covid_both,
```

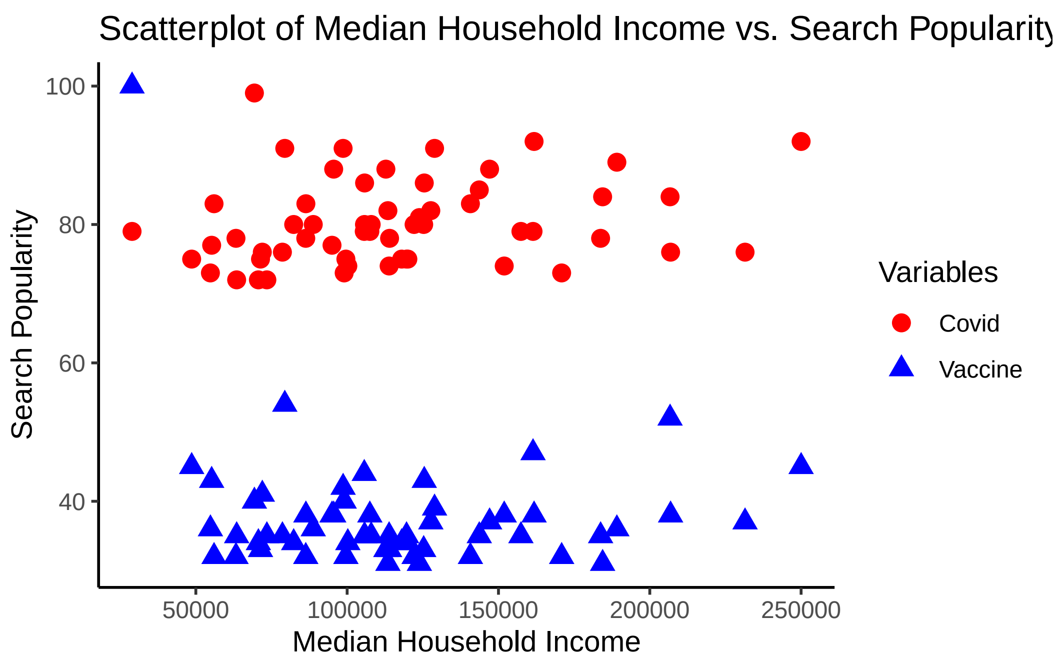
```r
                                by = 'location')
nrow(joined_covid)
```

```
[1] 57
```

```r
results_covid <- joined_covid %>%
  mutate(income_group = if_else(hh_income > median(acs_il$hh_income, na.rm =TRUE), 'high i
  group_by(income_group) %>%
  summarise(mean_covid = mean(covid),
            mean_vaccine = mean(vaccine)) %>%
  ungroup()

print(results_covid)
```

```
# A tibble: 2 x 3
  income_group mean_covid mean_vaccine
  <chr>             <dbl>        <dbl>
1 high income        80.6         36.7
2 low income         77.4         51.2
```

```r
ggplot(joined_covid) +
  geom_point(aes(x = hh_income, y = covid, color = "Covid"),
             shape = 16, size = 3) +
  geom_point(aes(x = hh_income, y = vaccine, color = "Vaccine"),
             shape = 17, size = 3) +
  labs(x = "Median Household Income",
       y = "Search Popularity",
       title = "Scatterplot of Median Household Income vs. Search Popularity") +
  scale_color_manual(name = "Variables",
                     values = c("Covid" = "red", "Vaccine" = "blue")) +
  theme_classic()
```




```r
cor.test(covid_data_freq_spread$covid, covid_data_freq_spread$vaccine)
```

```
	Pearson's product-moment correlation

data:  covid_data_freq_spread$covid and covid_data_freq_spread$vaccine
t = -6.6877, df = 209, p-value = 2.034e-10
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5251391 -0.3019023
sample estimates:
       cor
-0.4198504
```

```r
cor.test(joined_covid$hh_income, joined_covid$covid)
```

```
	Pearson's product-moment correlation

data:  joined_covid$hh_income and joined_covid$covid
```

```
t = 1.7415, df = 55, p-value = 0.08719
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.03398937  0.46167047
sample estimates:
      cor
0.2286029
```

```r
cor.test(joined_covid$hh_income, joined_covid$vaccine)
```

```

	Pearson's product-moment correlation

data:  joined_covid$hh_income and joined_covid$vaccine
t = -1.194, df = 55, p-value = 0.2376
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.4028394  0.1060012
sample estimates:
       cor
-0.1589569
```

From the correlation test we can see that the p value for test where we test covid and vaccine with house hold income both comes out as greater than 0.05 hence we failed to reject H0 at 95% confidence intereval therefore the correlation between both vaccine and household income and covid and household income is 0, while we can see that the correlation between vaccine and covid are statistically significant hence both have a negetive correlation of -0.42. We can observe that covid has been searched way more than vaccine irrespective of income than vaccine but still in high income covid was searched the most, vaccination search was least in the high income and highest in the lower income region