Assignment 2 Results

Isabel O'Malley, JPSM727, Oct 3rd 2023

Link to github repo: https://github.com/isabelshaheen/JPSM727-assignment2.git

Load packages

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.3     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

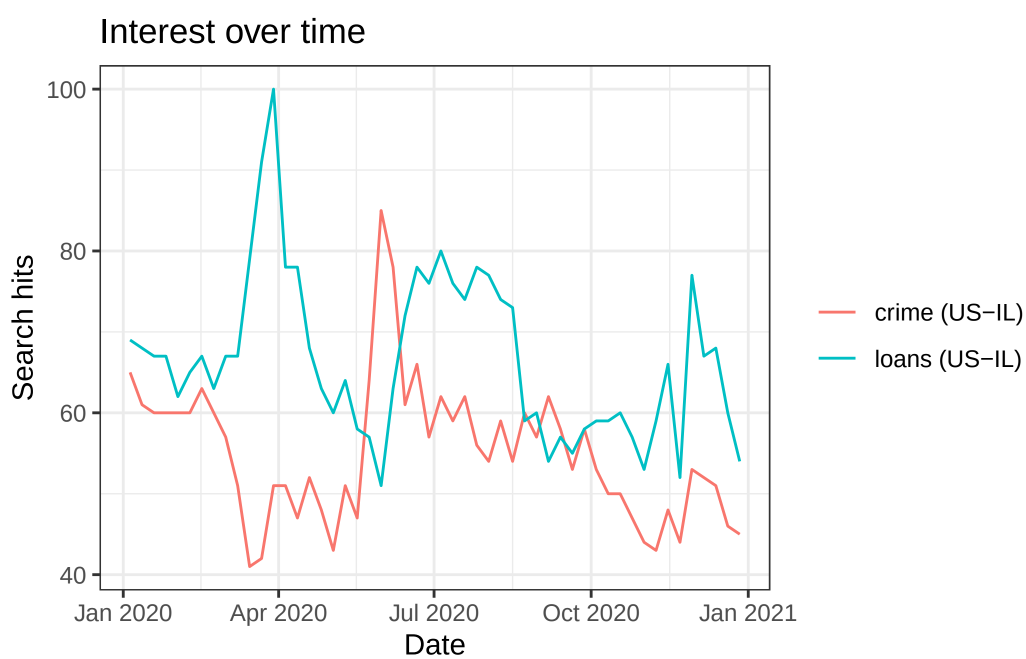
```r
library(gtrendsR)
library(censusapi)
```

```
Attaching package: 'censusapi'

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

    getFunction
```

Pulling from APIs - Crime and Loans

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.

First, we transform the `data.frame` into a `tibble`.

```
res_time <- as_tibble(res$interest_over_time)
glimpse(res_time)
```

```
Rows: 104
Columns: 7
```

```
$ date     <dttm> 2020-01-05, 2020-01-12, 2020-01-19, 2020-01-26, 2020-02-02, ~
$ hits     <int> 65, 61, 60, 60, 60, 60, 63, 60, 57, 51, 41, 42, 51, 51, 47, 5~
$ keyword  <chr> "crime", "crime", "crime", "crime", "crime", "crime", "crime"~
$ geo      <chr> "US-IL", "US-IL", "US-IL", "US-IL", "US-IL", "US-IL", "US-IL"~
$ time     <chr> "2020-01-01 2020-12-31", "2020-01-01 2020-12-31", "2020-01-01~
$ gprop    <chr> "web", "web", "web", "web", "web", "web", "web", "web", "web"~
$ category <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Then, we use the group_by function and we find mean, SD, median, and variance of hits for the two keywords. We output the data frame as a table using the kable option.

```r
summary_hits <- res_time %>%
  group_by(keyword) %>%
  summarize(mean_hits = mean(hits),
            sd_hits = sd(hits),
            median_hits = median(hits))

knitr:: kable(summary_hits,
               caption = "Mean, median, and variance of the search hits for the keywords")
```

Table 1: Mean, median, and variance of the search hits for the keywords

keyword	mean_hits	sd_hits	median_hits
crime	55.01923	8.580622	54.0
loans	66.61538	10.041393	66.5

- **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.

Note that the original results object `res` contains some additional information, such as the search interest by city/ region.

```r
res$interest_by_city
```

Make res$interest_by_city into a tibble and shorten name to res_city

Pivot wider to split the hits column into two variables: one for crime and one for loans

Arrange in descending order for loans and make a table with the top 10

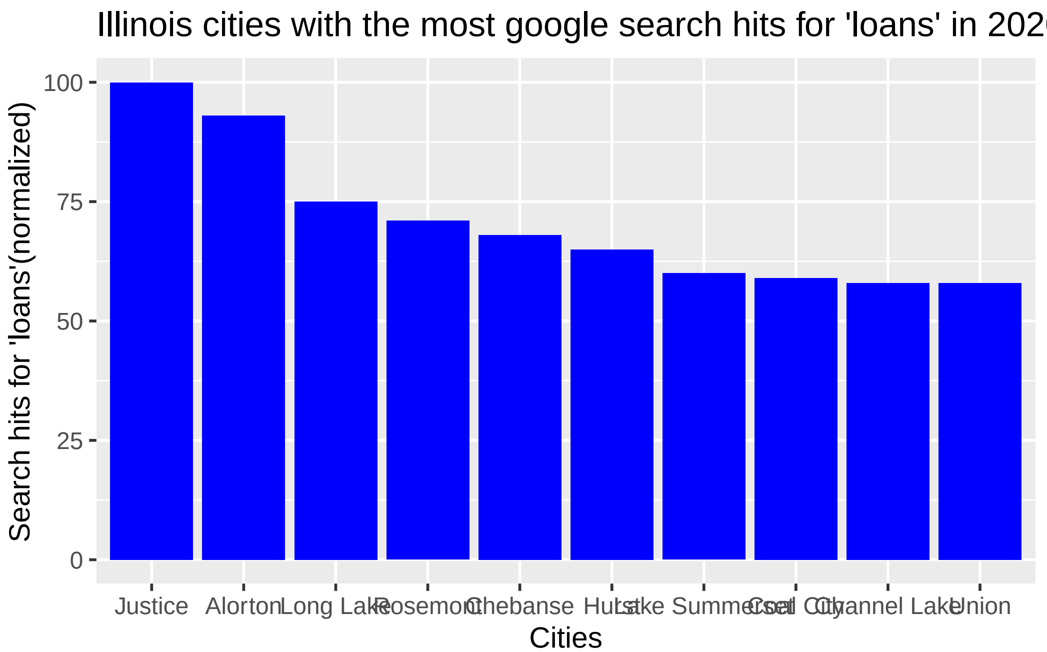
```r
# Arrange the dataframe in descending order of the loans variable
res_city_w <- res_city_w %>%
  arrange(desc(loans))

# Select the top 10 observations
top_10 <- head(res_city_w, 10)

# Use kable to create a table
knitr:: kable(top_10,
               caption = "10 cities with the highest search frequency for loans")
```

Table 2: 10 cities with the highest search frequency for loans

location	geo	gprop	crime	loans
Justice	US-IL	web	NA	100
Alorton	US-IL	web	NA	93
Long Lake	US-IL	web	NA	75
Rosemont	US-IL	web	NA	71
Chebanse	US-IL	web	NA	68
Hurst	US-IL	web	NA	65
Lake Summerset	US-IL	web	NA	60
Coal City	US-IL	web	NA	59
Union	US-IL	web	NA	58
Channel Lake	US-IL	web	NA	58

Plot only the 10 observations with the highest # of hits on loans

```r
# Create a bar plot using ggplot2
ggplot(data = top_10, aes(x = reorder(location, -loans), y = loans)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "Illinois cities with the most google search hits for 'loans' in 2020", x =
```



- Is there a relationship between the search intensities between the two keywords we used?

Convert NAs to 0

```
res_city_w <- res_city_w %>%
  mutate_all(~ifelse(is.na(.), 0, .))
```

Find the correlation between crime and loans hits

```
cor_test_result <- cor.test(res_city_w$crime, res_city_w$loans)

cor_test_result
```

```

	Pearson's product-moment correlation

data:  res_city_w$crime and res_city_w$loans
t = -2.0544, df = 344, p-value = 0.04069
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.213053194 -0.004711302
sample estimates:
       cor
-0.1100914
```

Answer: The p-value is < .001 and the t-value is -4.23 indicating a significant negative correlation between the number of google searches for "crime" and the number of searches for "loans" in Illinois in 2020. In other words, when searches for crime are high, searches for loans tend to be lower, and vice versa. This is consistent with the graph from google trends.

Google Trends - Crime and Loans + Illinois 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, store this key in the `cs_key` object. We will use this object in all following API queries.

```
cs_key <- "410ea52de7d0c298684fa54e92f6118f47a4aec9"
```

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.

```r
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.

```r
no_village <- gsub(' village, Illinois', '', acs_il$NAME)
no_city <- gsub(' city, Illinois', '', no_village)

acs_with_location <- acs_il %>%
  mutate(location = no_city)

acs_with_location %>% head(5)
```

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

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.

```r
locations_only_in_acs <- setdiff(acs_with_location$location, res_city_w$location)
locations_only_in_res <- setdiff(res_city_w$location, acs_with_location$location)

count_locations_acs <- length(locations_only_in_acs)
count_locations_res <- length(locations_only_in_res)

cat("Locations unique to acs:", count_locations_acs, "\n")
```

```
Locations unique to acs: 1132
```

```r
cat("Locations unique to res:", count_locations_res, "\n")
```

```
Locations unique to res: 14
```

- Then, create a new data set by joining the Google Trends and the ACS data. Keep only cities that appear in both data sets.

```r
res_join <- left_join(acs_with_location, res_city_w, by = "location")
res_join <- na.omit(res_join)
```

Inspect the result.

```r
str(res_join)
```

```
'data.frame':   334 obs. of  12 variables:
 $ state    : chr  "17" "17" "17" "17" ...
 $ place    : chr  "15300" "15352" "15664" "16691" ...
 $ NAME     : chr  "Cobden village, Illinois" "Coffeen city, Illinois" "Colona city, Illinois
 $ pop      : num  1018 640 5307 16564 8058 ...
 $ age      : num  44.2 33.4 43.1 40.9 49.2 35.4 39.9 48.3 45.7 33.7 ...
 $ hh_income: num  38750 35781 64643 70306 75403 ...
 $ income   : num  19979 26697 33759 30318 48203 ...
 $ location : chr  "Cobden" "Coffeen" "Colona" "Country Club Hills" ...
 $ geo      : chr  "US-IL" "US-IL" "US-IL" "US-IL" ...
 $ gprop    : chr  "web" "web" "web" "web" ...
 $ crime    : num  0 0 0 37 0 0 0 0 0 0 ...
 $ loans    : num  0 0 0 0 31 0 26 32 21 0 ...
 - attr(*, "na.action")= 'omit' Named int [1:1132] 1 4 5 6 7 9 10 11 12 13 ...
  ..- attr(*, "names")= chr [1:1132] "1" "4" "5" "6" ...
```

- 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 a 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?

```r
#Calculate median household income
median_hh_income <- median(res_join$hh_income)

#Group by above or below median household income and calculate mean for each keyword
summary_res_join <- res_join %>%
```

```
  mutate(hh_income_above_median = hh_income > median_hh_income) %>%
  group_by(hh_income_above_median) %>%
  summarise(mean_crime = mean(crime, na.rm = T),
            mean_loans = mean(loans, na.rm = T))

knitr:: kable(summary_res_join,
              caption = "Average search popularity for keywords in cities by average hous
```

Table 3: Average search popularity for keywords in cities by average household income

hh_income_above_median	mean_crime	mean_loans
FALSE	6.45509	14.61677
TRUE	10.49701	12.31138

Answer: For the cities with average household income below the state's median, the search term "loans" is twice as popular as the search term "crime." (21 v. 10) For cities with average household income above the state's median, the search terms "crime" and "loans" are equally as popular (both are 15).

- 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()`.

Calculate correlation between household income and search popularity of the google trends terms

```
cor_income_crime <- cor.test(res_join$hh_income, res_join$crime)

cor_income_loans <- cor.test(res_join$hh_income, res_join$loans)

cor_income_crime
```

```

    Pearson's product-moment correlation

data:  res_join$hh_income and res_join$crime
t = 4.7783, df = 332, p-value = 2.659e-06
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1504464 0.3514138
sample estimates:
```

```
      cor
0.2536655
```

```
cor_income_loans
```

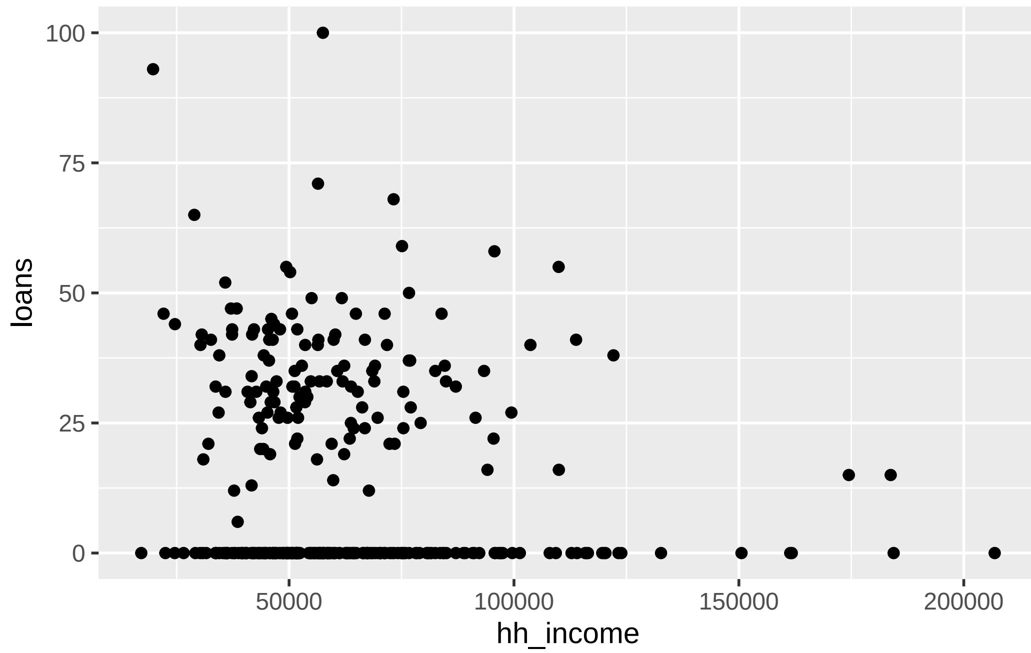
```

	Pearson's product-moment correlation

data:  res_join$hh_income and res_join$loans
t = -2.0841, df = 332, p-value = 0.03792
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.218291288 -0.006402381
sample estimates:
       cor
-0.1136388
```

Answer: There is a significant positive relationship between a city's household income and searches for crime. There is a significant negative relationship between a city's household income and searches for loans.

Plot household income and searches for loans

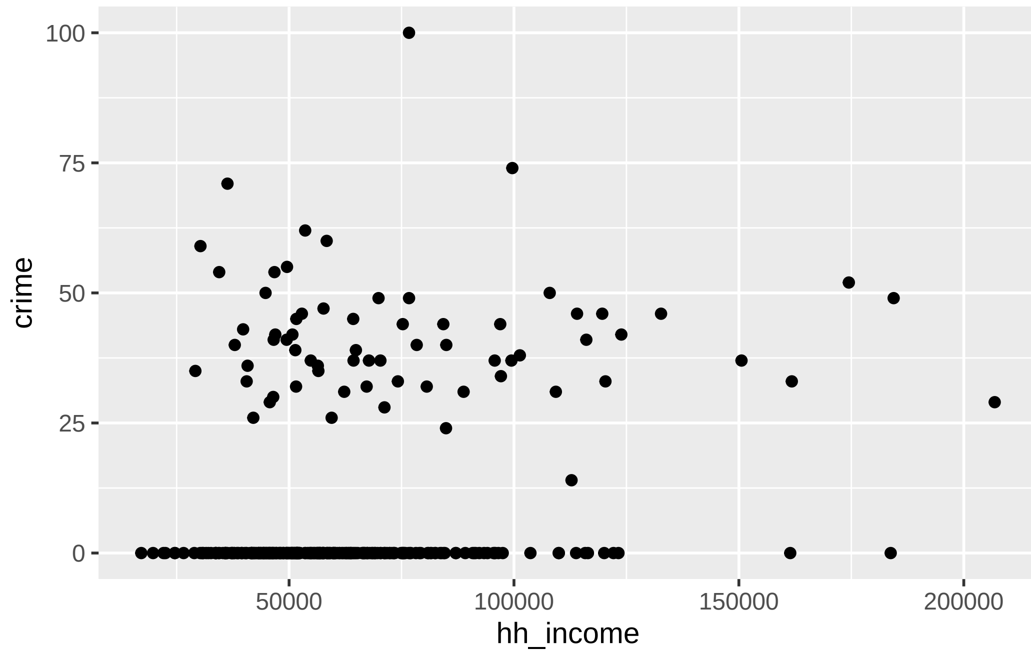
```
res_join %>%
  qplot(x = hh_income, y = loans, data = .,
        geom = "point")
```

```
Warning: `qplot()` was deprecated in ggplot2 3.4.0.
```



Plot household income and searches for crime

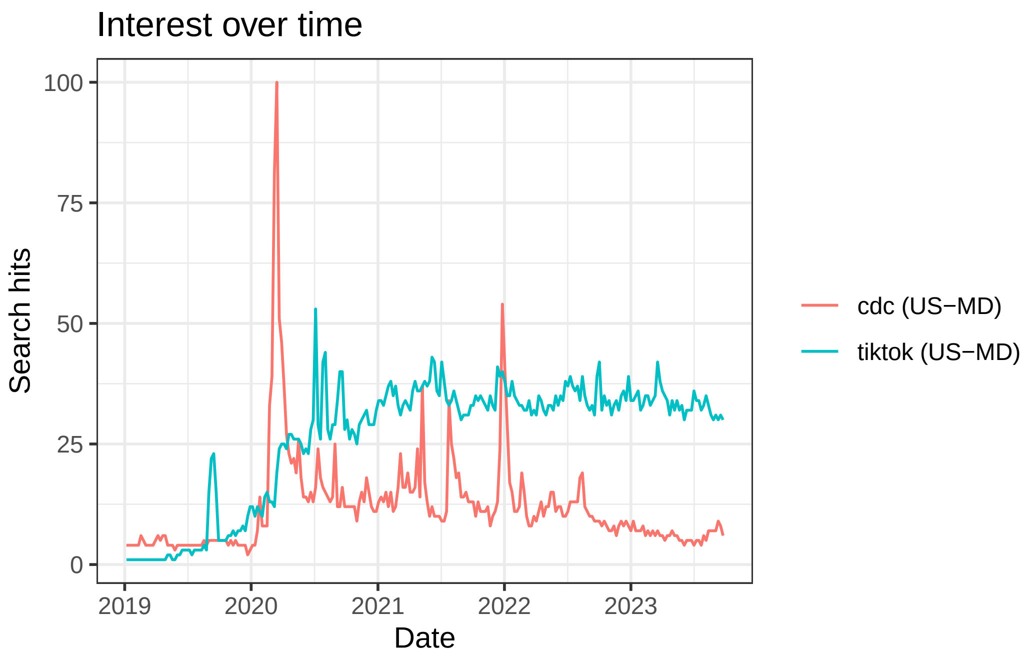
```
res_join %>%
  qplot(x = hh_income, y = crime, data = .,
        geom = "point")
```



Pulling from APIs - Covid Keywords

Our data source is the Google Trends API. Suppose we are interested in the search trends for `CDC` and `Tiktok` in Maryland in the years 2019-2023. We could find this using the following code:

```
res <- gtrends(c("cdc", "tiktok"),
               geo = "US-MD",
               time = "2019-01-01 2023-9-30",
               low_search_volume = TRUE)
plot(res)
```



Answer the following questions for the keywords.

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

First, we transform the `data.frame` into a `tibble`.

```
res_time <- as_tibble(res$interest_over_time)
glimpse(res_time)
```

```
Rows: 494
Columns: 7
$ date     <dttm> 2019-01-06, 2019-01-13, 2019-01-20, 2019-01-27, 2019-02-03, ~
$ hits     <int> 4, 4, 4, 4, 4, 4, 6, 5, 4, 4, 4, 4, 5, 6, 5, 6, 6, 4, 4, 4, 3~
$ keyword  <chr> "cdc", "cdc", "cdc", "cdc", "cdc", "cdc", "cdc", "cdc", "cdc"~
$ geo      <chr> "US-MD", "US-MD", "US-MD", "US-MD", "US-MD", "US-MD", "US-MD"~
$ time     <chr> "2019-01-01 2023-9-30", "2019-01-01 2023-9-30", "2019-01-01 2~
$ gprop    <chr> "web", "web", "web", "web", "web", "web", "web", "web", "web"~
$ category <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Then, we use the group_by function and we find mean, SD, median, and max hits for the two keywords.

```r
summary_hits <- res_time %>%
  group_by(keyword) %>%
  summarize(mean_hits = mean(hits),
            sd_hits = sd(hits),
            median_hits = median(hits),
            max_hits = max(hits))

knitr:: kable(summary_hits,
               caption = "Mean, median, and variance of the search hits for the keywords")
```

Table 4: Mean, median, and variance of the search hits for the keywords

keyword	mean_hits	sd_hits	median_hits	max_hits
cdc	11.91093	10.76203	10	100
tiktok	26.31174	12.84761	32	53

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

Pivot wider res_time to split the hits column into two variables

```r
#pivot wider
res_time_w <- pivot_wider(res_time,
                          names_from = keyword,
                          values_from = hits)
```

Make res$interest_by_city into a tibble and shorten name to res_city

```r
res_city <- as_tibble(res$interest_by_city)
glimpse(res_city)
```

```
Rows: 400
Columns: 5
$ location <chr> "Indian Head", "Chevy Chase", "Sabillasville", "Brooklandvill~
$ hits     <int> 100, 61, 57, 56, 55, 55, 55, 52, 51, 48, 47, 47, 47, 46, 46, ~
$ keyword  <chr> "cdc", "cdc", "cdc", "cdc", "cdc", "cdc", "cdc", "cdc", "cdc"~
$ geo      <chr> "US-MD", "US-MD", "US-MD", "US-MD", "US-MD", "US-MD", "US-MD"~
$ gprop    <chr> "web", "web", "web", "web", "web", "web", "web", "web", "web"~
```

Pivot wider with res_city

```
#identify duplicates
duplicates <- res_city %>%
  dplyr::group_by(location, geo, gprop, keyword) %>%
  dplyr::summarise(n = dplyr::n(), .groups = "drop") %>%
  dplyr::filter(n > 1L)

#remove duplicates
unique_res_city <- res_city %>%
  anti_join(duplicates, by = c("location", "geo", "gprop", "keyword"))

#pivot wider
res_city_w <- pivot_wider(unique_res_city,
                          names_from = keyword,
                          values_from = hits)
```

Let's find the cities with the highest numbers of hits for our keywords using `dplyrs` `arrange()` function.

```
res_city_w %>%
  select(location, cdc) %>%
  arrange(desc(cdc))

cdc_top10 <- head(res_city_w, 10)

knitr:: kable(cdc_top10,
              caption = "Cities with highest numbers of hits for CDC")
```

- Is there a relationship between the search intensities between the two keywords we used?

Convert NAs to 0

```
res_city_w <- res_city_w %>%
  mutate_all(~ifelse(is.na(.), 0, .))
```

Find the correlation between the two keywords

```
cor_test_result <- cor.test(res_city_w$cdc, res_city_w$tiktok)

cor_test_result
```

```
	Pearson's product-moment correlation

data:  res_city_w$cdc and res_city_w$tiktok
t = -1.0648, df = 277, p-value = 0.2879
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.17993012  0.05398892
sample estimates:
        cor 
-0.06384753 
```

Answer: The p-value is .4224 indicating there is no significant correlation between the number of google searches for "CDC" and the number of searches for "tiktok" in Maryland from 2019-2023. This is consistent with the google trends graph, where CDC hits spike around covid-19 disease incidence rises, and tiktok spokes are more aligned with the

Google Trends (covid keywords) + Maryland 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, store this key in the `cs_key` object. We will use this object in all following API queries.

```
cs_key <- "410ea52de7d0c298684fa54e92f6118f47a4aec9"
```

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 Maryland.**

Convert values that represent missings to NAs.

Now, it might be useful to rename the socio-demographic variables (`B01001_001E` etc.) in our data set and assign more meaningful names.

```
acs_md <-
  acs_md %>%
  rename(pop = B01001_001E,
         age = B06002_001E,
```

```r
           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, without the suffix (village, city, town, or CDP)

```r
no_village <- gsub(' village, Maryland', '', acs_md$NAME)
no_city <- gsub(' city, Maryland', '', no_village)
no_town <- gsub(' town, Maryland', '', no_city)
no_CDP <- gsub(' CDP, Maryland', '', no_town)

acs_with_location <- acs_md %>%
  mutate(location = no_CDP)

head(acs_with_location)
```

```
  state place                          NAME   pop  age hh_income income
1    24 66275     Rising Sun town, Maryland  2790 33.3     72021  30426
2    24 66400             Riva CDP, Maryland  4321 48.4    126792  61411
3    24 66635 Riverdale Park town, Maryland  7216 35.9     84695  33307
4    24 66762        Riverside CDP, Maryland  5888 33.4     79620  37306
5    24 66850   Riviera Beach CDP, Maryland 12780 39.0     94773  38869
6    24 67000        Robinwood CDP, Maryland  7482 37.9     67109  39706
        location
1     Rising Sun
2           Riva
3 Riverdale Park
4      Riverside
5  Riviera Beach
6      Robinwood
```

The following questions are answered with the Maryland “cdc” and “tiktok” Google trends data and the ACS data.

- First, check how many cities don’t appear in both data sets, i.e. cannot be matched.

```r
locations_only_in_acs <- setdiff(acs_with_location$location, res_city_w$location)
locations_only_in_res <- setdiff(res_city_w$location, acs_with_location$location)
```

```r
count_locations_acs <- length(locations_only_in_acs)
count_locations_res <- length(locations_only_in_res)

cat("Locations unique to acs:", count_locations_acs, "\n")
```

```
Locations unique to acs: 308
```

```r
cat("Locations unique to res:", count_locations_res, "\n")
```

```
Locations unique to res: 54
```

- Then, create a new data set by joining the Google Trends and the ACS data. Keep only cities that appear in both data sets.

```r
res_join <- left_join(acs_with_location, res_city_w, by = "location")
res_join <- na.omit(res_join)
```

Inspect the result.

```r
head(res_join)
```

```
  state place                               NAME   pop  age hh_income income
1    24 66275     Rising Sun town, Maryland  2790 33.3     72021  30426
2    24 66400            Riva CDP, Maryland  4321 48.4    126792  61411
3    24 66635 Riverdale Park town, Maryland  7216 35.9     84695  33307
6    24 67000       Robinwood CDP, Maryland  7482 37.9     67109  39706
7    24 67400      Rock Hall town, Maryland  1563 49.1     47639  29202
9    24 67675      Rockville city, Maryland 68155 39.5    111797  54611
        location   geo gprop cdc tiktok
1     Rising Sun US-MD   web   0     76
2           Riva US-MD   web  35      0
3 Riverdale Park US-MD   web  30      0
6      Robinwood US-MD   web   0      0
7      Rock Hall US-MD   web  26      0
9      Rockville US-MD   web  43     75
```

- 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 a 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?

```
#Calculate median household income
median_hh_income <- median(res_join$hh_income)

#Group by above or below median household income and calculate mean for each keyword
res_join %>%
  mutate(hh_income_above_median = hh_income > median_hh_income) %>%
  group_by(hh_income_above_median) %>%
  summarise(mean_cdc = mean(cdc, na.rm = T),
            mean_tiktok = mean(tiktok, na.rm = T))
```

```
# A tibble: 2 x 3
  hh_income_above_median mean_cdc mean_tiktok
  <lgl>                     <dbl>       <dbl>
1 FALSE                      14.9        29.7
2 TRUE                       29.7        22.7
```

Answer: The number of searches for “tiktok” is essentially the same across cities where average household income is below the state median and cities where average household income is above the state median. However, the average number of searches for “cdc” in the richer cities (with household income above the median) is nearly double that of the average number of searches for “cdc” in the poorer cities (with household income below the median). Together, these results suggest that Maryland residents’ interest in tiktok was similar across income groups, but interest in the CDC was higher among people in richer neighborhoods than in poorer neighborhoods.

- 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()`.

Calculate correlation between household income and search popularity of the google trends terms

```
cor_income_cdc <- cor.test(res_join$hh_income, res_join$cdc)
cor_income_tiktok <- cor.test(res_join$hh_income, res_join$tiktok)

cor_income_cdc
```

```

    Pearson's product-moment correlation

data:  res_join$hh_income and res_join$cdc
t = 8.2748, df = 221, p-value = 1.213e-14
```

```
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3792086 0.5806326
sample estimates:
      cor 
0.4863557 
```

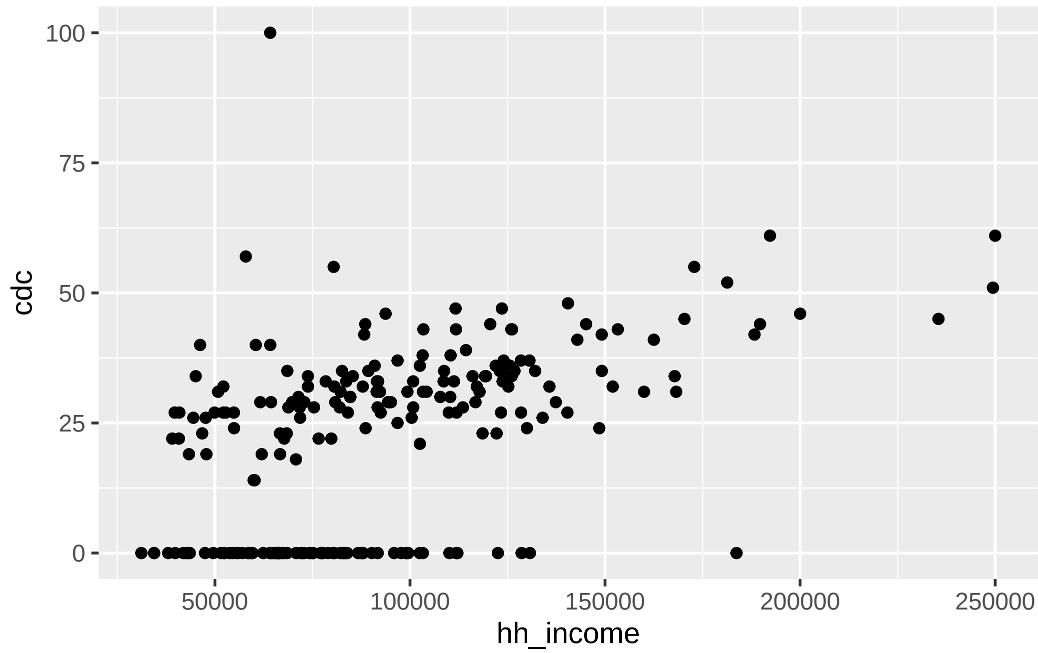
```
cor_income_tiktok
```

```

    Pearson's product-moment correlation

data:  res_join$hh_income and res_join$tiktok
t = -1.0328, df = 221, p-value = 0.3028
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.19887436  0.06263909
sample estimates:
        cor 
-0.06930824 
```

Answer: There is a significant positive relationship ($p < .001$, $t = 6.39$, CI: .28, .51) between a city's average household income and searches for cdc. This seems consistent with the graph, where at higher income levels it appears that the number of hits for CDC are slightly higher. There is no significant relationship ($p = .67$, $t = -.42$, CI: -.16, .10) between a city's average household income and searches for tiktok. This also seems consistent with the graph for tiktok, where no clear income trend emerges from the data points.

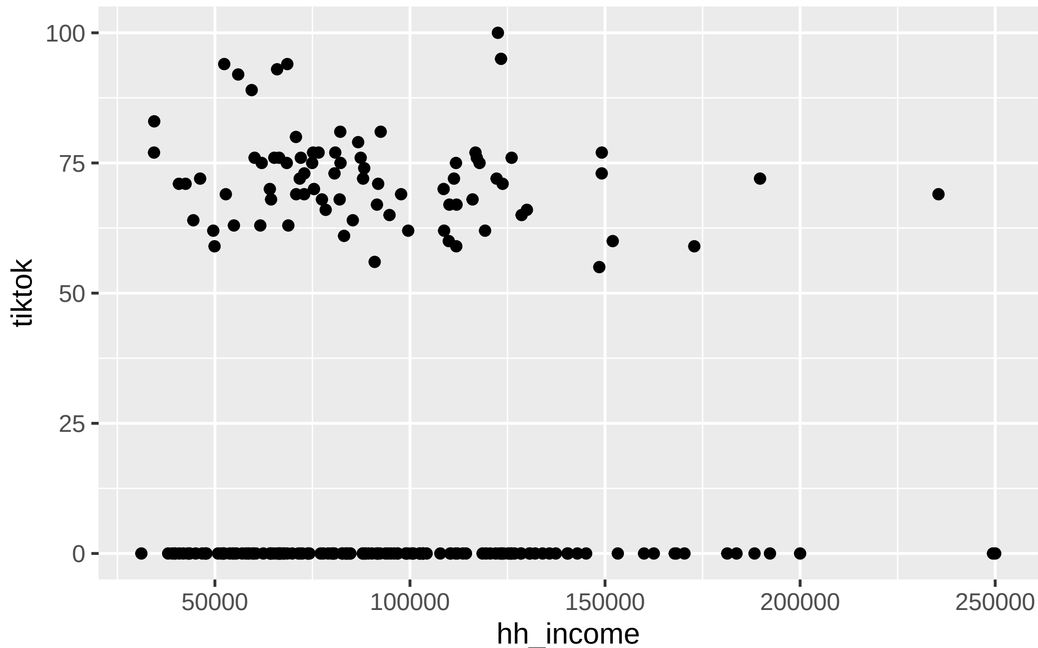
Household income and searches for cdc

```
res_join %>%
  qplot(x = hh_income, y = cdc, data = .,
        geom = "point")
```



Household income and searches for tiktok

```
res_join %>%
  qplot(x = hh_income, y = tiktok, data = .,
        geom = "point")
```



100
75
50
25
0
tiktok
50000
100000
150000
200000
250000
hh_income

~ The End ~