Assignment 2

Chloe Chen

2023-09-21

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)
```

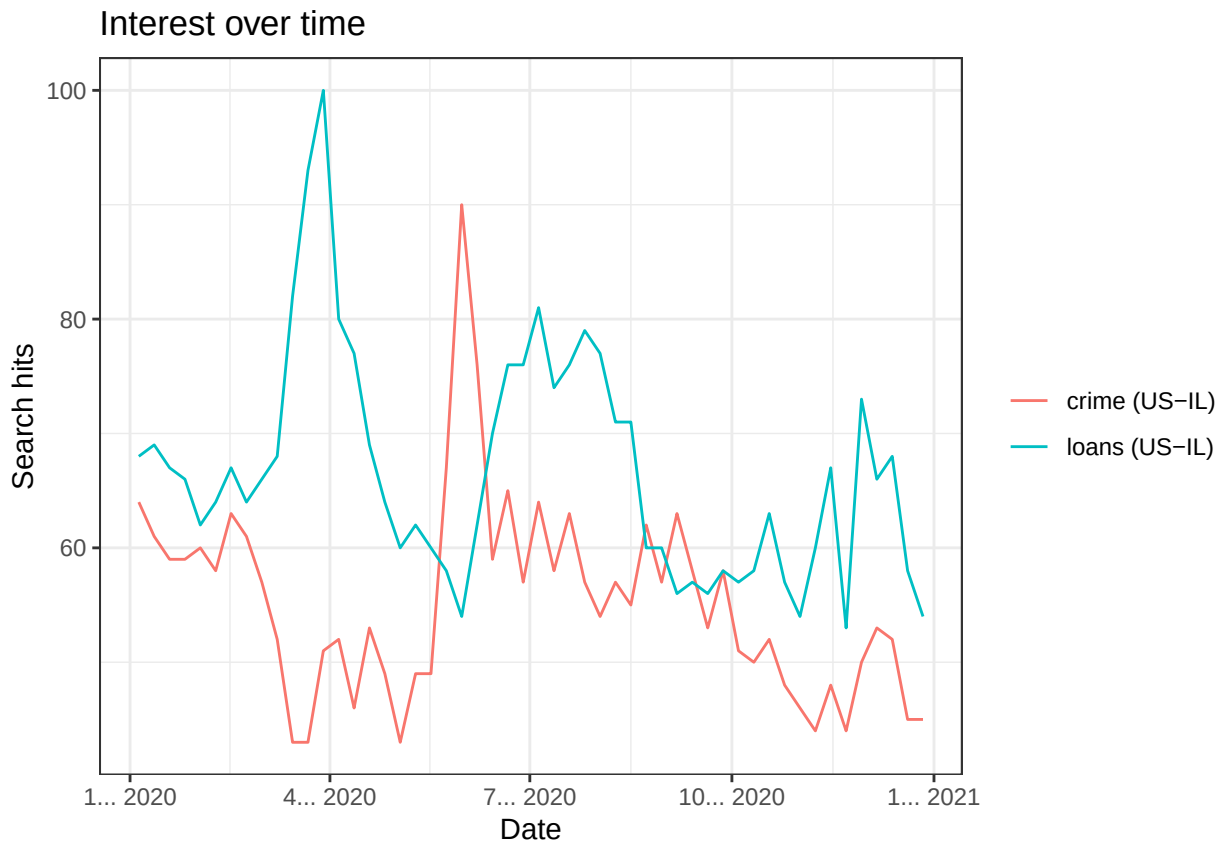
In this assignment, you will pull from APIs to get data from various data sources and use your data wrangling skills to use them all together. You should turn in a report in PDF or HTML format that addresses all of the questions in this assignment, and describes the data that you pulled and analyzed. 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.

```
head(res$interest_over_time)
```

```
##         date hits keyword   geo                 time gprop category
## 1 2020-01-05   64   crime US-IL 2020-01-01 2020-12-31   web        0
## 2 2020-01-12   61   crime US-IL 2020-01-01 2020-12-31   web        0
## 3 2020-01-19   59   crime US-IL 2020-01-01 2020-12-31   web        0
## 4 2020-01-26   59   crime US-IL 2020-01-01 2020-12-31   web        0
## 5 2020-02-02   60   crime US-IL 2020-01-01 2020-12-31   web        0
## 6 2020-02-09   58   crime US-IL 2020-01-01 2020-12-31   web        0
```

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

```
## # A tibble: 6 x 7
##   date                hits keyword geo   time                gprop category
##   <dttm>             <int> <chr>   <chr> <chr>               <chr>    <int>
## 1 2020-01-05 00:00:00    64 crime   US-IL 2020-01-01 2020-12-31 web          0
## 2 2020-01-12 00:00:00    61 crime   US-IL 2020-01-01 2020-12-31 web          0
## 3 2020-01-19 00:00:00    59 crime   US-IL 2020-01-01 2020-12-31 web          0
```

```
## 4 2020-01-26 00:00:00    59 crime   US-IL 2020-01-01 2020-12-31 web          0
## 5 2020-02-02 00:00:00    60 crime   US-IL 2020-01-01 2020-12-31 web          0
## 6 2020-02-09 00:00:00    58 crime   US-IL 2020-01-01 2020-12-31 web          0
```

```
res_time%>%
  group_by(keyword) %>%
  summarise(mean = mean(hits, na.rm = T),
            median = median(hits, na.rm = T),
            variance = var(hits, na.rm = T))
```

```
## # A tibble: 2 x 4
##   keyword  mean median variance
##   <chr>   <dbl>  <dbl>    <dbl>
## 1 crime    55.2   54.5     76.2
## 2 loans    66.7   66       98.2
```

- For the keyword "crime",the mean is 55.25, the median is 54.50, and the variance is 76.15.
- For the keyword "loans",the mean is 66.69, the median is 66.00, and the variance is 98.22.

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.

```
res_city <- as_tibble(res$interest_by_city)
head(res_city)
```

```
## # A tibble: 6 x 5
##   location         hits keyword geo   gprop
##   <chr>           <int> <chr>   <chr> <chr>
## 1 Hebron            100 crime   US-IL web
## 2 Mokena             91 crime   US-IL web
## 3 Anna               83 crime   US-IL web
## 4 Cahokia            76 crime   US-IL web
## 5 North Riverside    75 crime   US-IL web
## 6 Macomb             71 crime   US-IL web
```

```
library(dplyr)
city_loan<-res_city %>%
  filter(keyword == "loans") %>%
  group_by(location)
city_loan
```

```
## # A tibble: 200 x 5
## # Groups:   location [200]
##    location         hits keyword geo   gprop
##    <chr>           <int> <chr>   <chr> <chr>
##  1 Alorton           100 loans   US-IL web
##  2 Oakwood            79 loans   US-IL web
##  3 Roseville          77 loans   US-IL web
```

```
##  4 Rosemont           76 loans   US-IL web
##  5 Witt               73 loans   US-IL web
##  6 Washington Park    70 loans   US-IL web
##  7 Kingston           69 loans   US-IL web
##  8 Crainville         67 loans   US-IL web
##  9 Coal City          67 loans   US-IL web
## 10 Robbins            66 loans   US-IL web
## # i 190 more rows
```

```
max_loan_city <- city_loan[which.max(city_loan$hits), ]
max_loan_city
```

```
## # A tibble: 1 x 5
## # Groups:   location [1]
##   location  hits keyword geo   gprop
##   <chr>    <int> <chr>   <chr> <chr>
## 1 Alorton    100 loans   US-IL web
```

- *White City* has the highest search frequency for *loans*

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

```
res_time_filtered <- res_time %>%
  filter(keyword %in% c("crime", "loans"))
# Pivot the data into its wider version ("crime" and "loans" as columns)
wider_res <- pivot_wider(res_time_filtered, names_from = keyword, values_from = hits)
head(wider_res)
```

```
## # A tibble: 6 x 7
##   date                geo   time                   gprop category crime loans
##   <dttm>              <chr> <chr>                  <chr>    <int> <int> <int>
## 1 2020-01-05 00:00:00 US-IL 2020-01-01 2020-12-31 web          0    64    68
## 2 2020-01-12 00:00:00 US-IL 2020-01-01 2020-12-31 web          0    61    69
## 3 2020-01-19 00:00:00 US-IL 2020-01-01 2020-12-31 web          0    59    67
## 4 2020-01-26 00:00:00 US-IL 2020-01-01 2020-12-31 web          0    59    66
## 5 2020-02-02 00:00:00 US-IL 2020-01-01 2020-12-31 web          0    60    62
## 6 2020-02-09 00:00:00 US-IL 2020-01-01 2020-12-31 web          0    58    64
```

```
# Calculate the correlation
correlation <- cor(wider_res$crime, wider_res$loans, use = "complete.obs")
correlation
```

```
## [1] -0.09250297
```

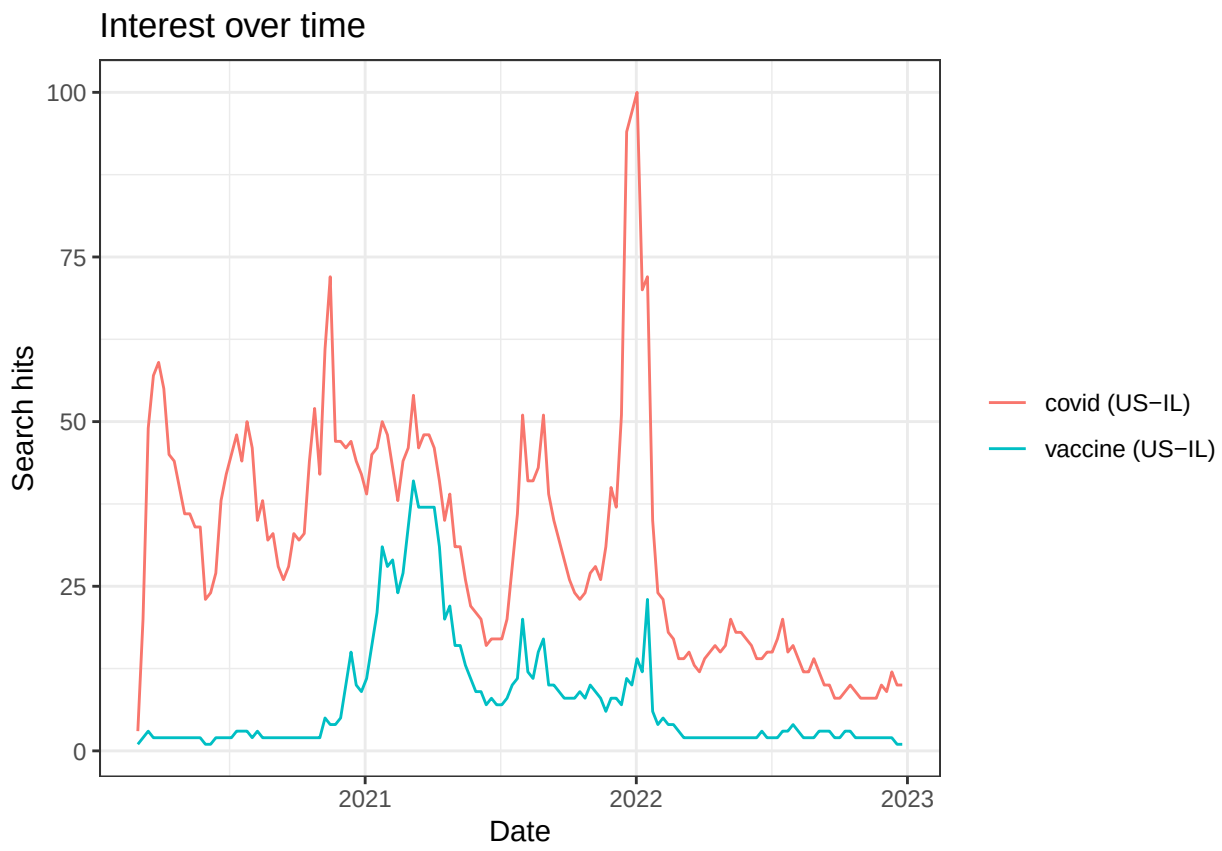
```
# Hypothesis test
cor_test <- cor.test(wider_res$crime, wider_res$loans, method = "pearson")
print(cor_test)
```

```
##
##  Pearson's product-moment correlation
##
## data:  wider_res$crime and wider_res$loans
## t = -0.65691, df = 50, p-value = 0.5143
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3564061  0.1850693
## sample estimates:
##         cor
## -0.09250297
```

- The correlation coefficient of -0.05918096 suggests a very weak negative linear relationship between the search intensities for the keywords "crime" and "loans" in Illinois in 2020. This correlation is close to zero, also by doing the hypothesis test, the p-value is greater than 0.05, this means we fail to reject the null hypothesis and conclude that there is no significant correlation, which means that there is essentially no meaningful linear relationship between these two variables.

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.

```
res2 <- gtrends(c("covid", "vaccine"),
                geo = "US-IL",
                time = "2020-2-28 2022-12-31",
                low_search_volume = TRUE)
plot(res2)
```




```
head(res2$interest_over_time)
```

```
##         date hits keyword   geo                 time gprop category
## 1 2020-03-01    3   covid US-IL 2020-2-28 2022-12-31   web        0
## 2 2020-03-08   20   covid US-IL 2020-2-28 2022-12-31   web        0
## 3 2020-03-15   49   covid US-IL 2020-2-28 2022-12-31   web        0
## 4 2020-03-22   57   covid US-IL 2020-2-28 2022-12-31   web        0
## 5 2020-03-29   59   covid US-IL 2020-2-28 2022-12-31   web        0
## 6 2020-04-05   55   covid US-IL 2020-2-28 2022-12-31   web        0
```

```
res2_time <- as_tibble(res2$interest_over_time)
head(res2_time)
```

```
## # A tibble: 6 x 7
##   date                hits keyword geo   time                gprop category
##   <dttm>             <int> <chr>   <chr> <chr>               <chr>    <int>
## 1 2020-03-01 00:00:00     3 covid   US-IL 2020-2-28 2022-12-31 web          0
## 2 2020-03-08 00:00:00    20 covid   US-IL 2020-2-28 2022-12-31 web          0
## 3 2020-03-15 00:00:00    49 covid   US-IL 2020-2-28 2022-12-31 web          0
## 4 2020-03-22 00:00:00    57 covid   US-IL 2020-2-28 2022-12-31 web          0
## 5 2020-03-29 00:00:00    59 covid   US-IL 2020-2-28 2022-12-31 web          0
## 6 2020-04-05 00:00:00    55 covid   US-IL 2020-2-28 2022-12-31 web          0
```

```
res2_time%>%
  group_by(keyword) %>%
  summarise(mean = mean(hits, na.rm = T),
            median = median(hits, na.rm = T),
            variance = var(hits, na.rm = T))
```

```
## # A tibble: 2 x 4
##   keyword  mean median variance
##   <chr>   <dbl>  <dbl>    <dbl>
## 1 covid   31.3      30    329.
## 2 vaccine  7.72      3     81.2
```

- For the keyword "covid",the mean is 31.34, the median is 30, and the variance is 329.04.
- For the keyword "vaccine",the mean is 7.72, the median is 3, and the variance is 81.18.

```
# The location with the most hits for the keywords.
res2_city <- as_tibble(res2$interest_by_city)
head(res2_city)
```

```
## # A tibble: 6 x 5
##   location         hits keyword geo   gprop
##   <chr>           <int> <chr>   <chr> <chr>
## 1 Downers Grove     100 covid   US-IL web
## 2 Highland Park      99 covid   US-IL web
## 3 Northfield         99 covid   US-IL web
## 4 Rolling Meadows    98 covid   US-IL web
## 5 Barrington         98 covid   US-IL web
## 6 Northbrook         98 covid   US-IL web
```

```
library(dplyr)
city_covid<-res2_city %>%
  filter(keyword == "covid") %>%
  group_by(location)
city_covid
```

```
## # A tibble: 200 x 5
## # Groups:   location [200]
##    location         hits keyword geo   gprop
##    <chr>           <int> <chr>   <chr> <chr>
##  1 Downers Grove     100 covid   US-IL web
##  2 Highland Park      99 covid   US-IL web
##  3 Northfield         99 covid   US-IL web
##  4 Rolling Meadows    98 covid   US-IL web
##  5 Barrington         98 covid   US-IL web
##  6 Northbrook         98 covid   US-IL web
##  7 Deer Park          97 covid   US-IL web
##  8 Hudson             95 covid   US-IL web
##  9 Western Springs    94 covid   US-IL web
## 10 Lake Forest        94 covid   US-IL web
## # i 190 more rows
```

```r
max_covid_city <- city_covid[which.max(city_covid$hits), ]
max_covid_city
```

```
## # A tibble: 1 x 5
## # Groups:   location [1]
##   location       hits keyword geo   gprop
##   <chr>         <int> <chr>   <chr> <chr>
## 1 Downers Grove   100 covid   US-IL web
```

```r
city_vaccine<-res2_city %>%
  filter(keyword == "vaccine") %>%
  group_by(location)
city_vaccine
```

```
## # A tibble: 200 x 5
## # Groups:   location [199]
##    location         hits keyword geo   gprop
##    <chr>           <int> <chr>   <chr> <chr>
##  1 Wheeling          100 vaccine US-IL web
##  2 Willowbrook        98 vaccine US-IL web
##  3 Forsyth            94 vaccine US-IL web
##  4 Rolling Meadows    91 vaccine US-IL web
##  5 Lincolnwood        88 vaccine US-IL web
##  6 Barrington         85 vaccine US-IL web
##  7 Northbrook         84 vaccine US-IL web
##  8 Winfield           83 vaccine US-IL web
##  9 Northfield         83 vaccine US-IL web
## 10 Naperville         82 vaccine US-IL web
## # i 190 more rows
```

```r
max_vaccine_city <- city_vaccine[which.max(city_vaccine$hits), ]
max_vaccine_city
```

```
## # A tibble: 1 x 5
## # Groups:   location [1]
##   location  hits keyword geo   gprop
##   <chr>    <int> <chr>   <chr> <chr>
## 1 Wheeling   100 vaccine US-IL web
```

```r
#The correlation between two keywords
res2_time_filtered <- res2_time %>%
  filter(keyword %in% c("covid", "vaccine"))
pivot_res2 <- pivot_wider(res2_time_filtered, names_from = keyword, values_from = hits)
head(pivot_res2)
```

```
## # A tibble: 6 x 7
##   date                geo   time                gprop category covid vaccine
##   <dttm>              <chr> <chr>               <chr>    <int> <int>   <int>
## 1 2020-03-01 00:00:00 US-IL 2020-2-28 2022-12-31 web          0     3       1
## 2 2020-03-08 00:00:00 US-IL 2020-2-28 2022-12-31 web          0    20       2
## 3 2020-03-15 00:00:00 US-IL 2020-2-28 2022-12-31 web          0    49       3
```

```
## 4 2020-03-22 00:00:00 US-IL 2020-2-28 2022-12-31 web          0    57       2
## 5 2020-03-29 00:00:00 US-IL 2020-2-28 2022-12-31 web          0    59       2
## 6 2020-04-05 00:00:00 US-IL 2020-2-28 2022-12-31 web          0    55       2
```

```
correlation2 <- cor(pivot_res2$covid, pivot_res2$vaccine, use = "complete.obs")
correlation2
```

```
## [1] 0.4333686
```

```
cor_test_result <- cor.test(pivot_res2$covid, pivot_res2$vaccine, method = "pearson")
print(cor_test_result)
```

```
##
##  Pearson's product-moment correlation
##
## data:  pivot_res2$covid and pivot_res2$vaccine
## t = 5.8104, df = 146, p-value = 3.761e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2924750 0.5558469
## sample estimates:
##       cor
## 0.4333686
```

- Among all the locations in IL, *Wheeling* had the most hits for both "covid" and "vaccine".
- The correlation coefficient of 0.4366658 suggests a significantly positive linear relationship between the search intensities for the keywords "covid" and "vaccine" in IL during the examined period.

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, store this key in the cs_key object. We will use this object in all following API queries.

```
cs_key <- "3ce4bbc5d8a79c7bc1800141e8002f293c05d73d"
```

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 = "3ce4bbc5d8a79c7bc1800141e8002f293c05d73d")
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 %>%
  rename(pop = B01001_001E,
         age = B06002_001E,
         hh_income = B19013_001E,
         income = B19301_001E)
head(acs_il)
```

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.

```
res_city_filtered <- res_city %>%
  filter(keyword %in% c("crime", "loans"))
res_city_wider <- pivot_wider(res_city_filtered, names_from = keyword, values_from = hits)
head(res_city_wider)
```

```
## # A tibble: 6 x 5
##   location        geo   gprop crime loans
##   <chr>           <chr> <chr> <int> <int>
## 1 Hebron          US-IL web     100    NA
## 2 Mokena          US-IL web      91    NA
## 3 Anna            US-IL web      83    NA
## 4 Cahokia         US-IL web      76    46
## 5 North Riverside US-IL web      75    NA
## 6 Macomb          US-IL web      71    NA
```

```
library(tidyverse)
library(magrittr)
acs_il2<-acs_il %<>%
  separate(NAME, c("location", "GEO"), sep = ",")
head(acs_il2)
```

```
##   state place          location       GEO 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
```

```
library(stringr)
# Remove the last word from the location column
acs_il3<-acs_il2 %>%
  mutate(location = str_remove(location, "\\s[[:alnum:]]+$"))
# Print the head of the modified ACS data frame
head(acs_il3)
```

```
##   state place   location       GEO B01001_001E B06002_001E B19013_001E
## 1    17 15261  Coatsburg  Illinois         180        35.6       55714
## 2    17 15300     Cobden  Illinois        1018        44.2       38750
## 3    17 15352    Coffeen  Illinois         640        33.4       35781
## 4    17 15378 Colchester  Illinois        1347        42.2       43942
## 5    17 15469     Coleta  Illinois         230        27.7       56875
## 6    17 15495     Colfax  Illinois        1088        32.5       58889
##   B19301_001E
## 1       27821
## 2       19979
## 3       26697
## 4       24095
## 5       23749
## 6       24861
```

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.

```
# Check how many unique cities are in each data set
unique_g <- unique(res_city_wider$location)
unique_acs <- unique(acs_il3$location)

# Find cities that are in one data set but not the other
gtrends_only <- setdiff(unique_g, unique_acs)
acs_only <- setdiff(unique_acs, unique_g)

# Print the number of cities in each category
cat("Cities in Google Trends data but not in ACS data:", length(gtrends_only), "\n")
```

```
## Cities in Google Trends data but not in ACS data: 6
```

```
cat("Cities in ACS data but not in Google Trends data:", length(acs_only), "\n")
```

```
## Cities in ACS data but not in Google Trends data: 1111
```

```
# Create a new data set by joining the two data frames for matching cities
merged_data <- inner_join(res_city_wider, acs_il3, by = "location" )

# Print the head of the merged data set
head(merged_data)
```

```
## # A tibble: 6 x 12
##   location     geo   gprop crime loans state place GEO   B01001_001E B06002_001E
##   <chr>        <chr> <chr> <int> <int> <chr> <chr> <chr>       <dbl>       <dbl>
## 1 Hebron       US-IL web     100    NA 17    33851 " Il~        1730        32.7
## 2 Mokena       US-IL web      91    NA 17    49854 " Il~       20720        42
## 3 Anna         US-IL web      83    NA 17    01543 " Il~        4149        42.4
## 4 Cahokia      US-IL web      76    46 17    10370 " Il~       14035        32.4
## 5 North River~ US-IL web      75    NA 17    54144 " Il~        6460        39.3
## 6 Macomb       US-IL web      71    NA 17    45889 " Il~       17658        27.2
## # i 2 more variables: B19013_001E <dbl>, B19301_001E <dbl>
```

```
merged_data <-
  merged_data %>%
  rename(pop = B01001_001E,
         age = B06002_001E,
         hh_income = B19013_001E,
         income = B19301_001E)
merged_data[merged_data == -666666666] <- NA
```

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?

```
library(dplyr)
summary(merged_data$hh_income)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   19764   45363   56118   61823   71989  184412       2
```

```
# Step 1: Calculate the average median household income
average_hhincome <- mean(merged_data$hh_income, na.rm = TRUE)
average_hhincome
```

```
## [1] 61823.22
```

```
# Step 2: Create a grouping variable for income categories
merged_data$income_group <- ifelse(merged_data$hh_income > average_hhincome, "Above Average", "Below Av
# Step 3: Group the data by income category
group_mean <- merged_data %>%
  group_by(income_group)%>%
  summarise(mean_crime_popularity = mean(crime, na.rm = TRUE),
            mean_loan_popularity = mean(loans, na.rm = TRUE))
print(group_mean)
```

```
## # A tibble: 3 x 3
##   income_group  mean_crime_popularity mean_loan_popularity
##   <chr>                         <dbl>                <dbl>
## 1 Above Average                  45.0                 34.3
## 2 Below Average                  48.5                 40.7
## 3 <NA>                           NaN                  69
```

- **Conclusion**: the hits of both "crime" and "loans" are higher in cities that have an below average median household income, this result might indicate people living in less wealthy communities may care more about security issues physically and financially.

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

```
head(res2_city)
```

```
## # A tibble: 6 x 5
##   location         hits keyword geo   gprop
##   <chr>           <int> <chr>   <chr> <chr>
## 1 Downers Grove     100 covid   US-IL web
## 2 Highland Park      99 covid   US-IL web
## 3 Northfield         99 covid   US-IL web
## 4 Rolling Meadows    98 covid   US-IL web
## 5 Barrington         98 covid   US-IL web
## 6 Northbrook         98 covid   US-IL web
```

```r
#res2_city_filtered <- res2_city %>%
#  filter(keyword %in% c("covid", "vaccine"))
res2_city_wider <- pivot_wider(res2_city, names_from = keyword, values_from = hits, values_fn = list(hi
head(res2_city_wider)
```

```
## # A tibble: 6 x 5
##   location        geo   gprop covid vaccine
##   <chr>           <chr> <chr> <int>   <int>
## 1 Downers Grove   US-IL web     100      75
## 2 Highland Park   US-IL web      99      71
## 3 Northfield      US-IL web      99      83
## 4 Rolling Meadows US-IL web      98      91
## 5 Barrington      US-IL web      98      85
## 6 Northbrook      US-IL web      98      84
```

```r
# Check how many unique cities are in each data set
unique_g2 <- unique(res2_city_wider$location)

# Find cities that are in one data set but not the other
gtrends2_only <- setdiff(unique_g2, unique_acs)
acs_only <- setdiff(unique_acs, unique_g2)

# Print the number of cities in each category
cat("Cities in Google Trends data but not in ACS data:", length(gtrends2_only), "\n")
```

```
## Cities in Google Trends data but not in ACS data: 9
```

```r
cat("Cities in ACS data but not in Google Trends data:", length(acs_only), "\n")
```

```
## Cities in ACS data but not in Google Trends data: 1122
```

```r
# Create a new data set by joining the two data frames for matching cities
merged_data2 <- inner_join(res2_city_wider, acs_il3, by = "location" )

# Print the head of the merged data set
head(merged_data2)
```

```
## # A tibble: 6 x 12
##   location   geo   gprop covid vaccine state place GEO   B01001_001E B06002_001E
##   <chr>      <chr> <chr> <int>   <int> <chr> <chr> <chr>       <dbl>       <dbl>
## 1 Downers G~ US-IL web     100      75 17    20591 " Il~       49263        43.1
## 2 Highland ~ US-IL web      99      71 17    34722 " Il~       29596        47.2
## 3 Northfield US-IL web      99      83 17    53663 " Il~        5678        52.3
## 4 Rolling M~ US-IL web      98      91 17    65338 " Il~       23288        38
## 5 Barrington US-IL web      98      85 17    03844 " Il~       10442        40.8
## 6 Northbrook US-IL web      98      84 17    53481 " Il~       33216        49.7
## # i 2 more variables: B19013_001E <dbl>, B19301_001E <dbl>
```

```r
merged_data2 <-
  merged_data2 %>%
```

```
  rename(pop = B01001_001E,
         age = B06002_001E,
         hh_income = B19013_001E,
         income = B19301_001E)
merged_data2[merged_data2 == -666666666] <- NA
```

```
library(dplyr)
summary(merged_data2$hh_income)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   19188   49375   61563   69950   83906  231477       1
```

```
# Step 1: Calculate the average median household income
average_hhincome2 <- mean(merged_data2$hh_income, na.rm = TRUE)
average_hhincome2
```

```
## [1] 69949.72
```

```
# Step 2: Create a grouping variable for income categories
merged_data2$income_group <- ifelse(merged_data2$hh_income > average_hhincome2, "Above Average", "Below
# Step 3: Group the data by income category
group_mean2 <- merged_data2 %>%
  group_by(income_group)%>%
  summarise(mean_covid_popularity2 = mean(covid, na.rm = TRUE),
            mean_vaccine_popularity2 = mean(vaccine, na.rm = TRUE))
print(group_mean2)
```

```
## # A tibble: 3 x 3
##   income_group  mean_covid_popularity2 mean_vaccine_popularity2
##   <chr>                          <dbl>                    <dbl>
## 1 Above Average                   78.8                     64.8
## 2 Below Average                   63.7                     45.8
## 3 <NA>                           NaN                       66
```

- **Conclusion**: the hits of both "covid" and "vaccine" are higher in cities that have an above average median household income, this result might indicate people living in wealthier communities may care more about covid and vaccine issues.