Assignment 2 727

Aulia Dini & Yesdi Calvin

2023-10-03

Github Link :

https://github.com/yesdicalvin/Assignment-2-727.git

```
# open library
library(tidyverse)
library(gtrendsR)
library(censusapi)
library(dplyr)
library(ggplot2)
```

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:

```
set.seed(727)
res <- gtrends(c("crime", "loans"),
               geo = "US-IL",
             time = "2020-01-01 2020-12-31",
             low_search_volume = TRUE)
```

Answer the following questions for the keywords "crime" and "loans".

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

```
# change the tibble into long format
res_time_w <- pivot_wider(res$interest_over_time,
                          names_from = keyword,
                          values_from = hits)
head(res_time_w)
```

```
## # 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    67
## 2 2020-01-12 00:00:00 US-IL 2020-01-01 2020-12-31 web          0    61    70
## 3 2020-01-19 00:00:00 US-IL 2020-01-01 2020-12-31 web          0    59    68
## 4 2020-01-26 00:00:00 US-IL 2020-01-01 2020-12-31 web          0    59    67
```

```
## 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    57    63
```

```r
# change the long format into the wide format
res_time_w <- res_time_w %>%
  pivot_longer(6:7, names_to = "keyword", values_to = "hits")
```

```r
# compute mean, median, var of hits
res_time_w %>%
  group_by(keyword) %>%
  summarize(mean_hits=mean(hits),
            med_hits=median(hits),
            var_sd=var(hits))
```

```
## # A tibble: 2 x 4
##   keyword mean_hits med_hits var_sd
##   <chr>       <dbl>    <dbl>  <dbl>
## 1 crime        54.6       54   79.5
## 2 loans        66.0       66   99.8
```

Analysis :

The summary statistics describe the search popularity of `crime` and `loans` in Illinois from January 1, 2020, to December 31, 2020. The mean of `loans` hits is higher than `crime` hits, with the value of 65.9 and 54.3. The mean from `loans` and `crime` search hits has a small discrepancy with the median, meaning there are no influential observations in the datasets. The variance of `loans` hits is higher than `crime` hits.

2. 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.

```r
# change the data format into wide format
res_city_loans <- res$interest_by_city %>%
  pivot_wider(names_from = keyword,
              values_from= hits)%>%

# sort the data by the value of search hits of loans
arrange(desc(loans))

# print the result
print(res_city_loans)
```

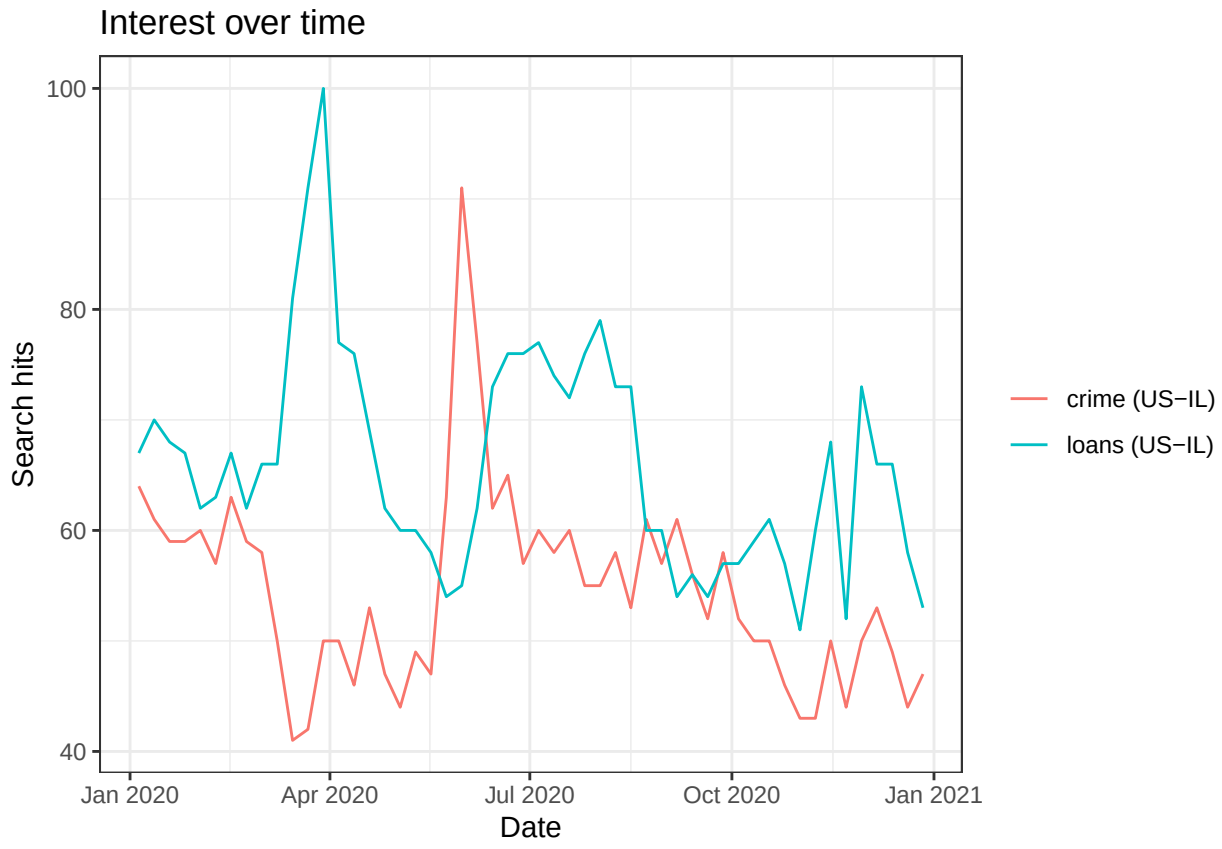
```
## # A tibble: 355 x 5
##    location      geo   gprop crime loans
##    <chr>         <chr> <chr> <int> <int>
##  1 Justice       US-IL web      NA   100
##  2 Roanoke       US-IL web      NA    94
##  3 Hinckley      US-IL web      NA    84
##  4 Roseville     US-IL web      NA    83
##  5 Carrier Mills US-IL web      NA    81
```

```
##  6 Long Lake     US-IL web      NA    80
##  7 Riverton      US-IL web      NA    74
##  8 Rosemont      US-IL web      44    73
##  9 Benld         US-IL web      NA    68
## 10 Dolton        US-IL web      NA    66
## # i 345 more rows
```

The locations with highest value of hits for `loans` in Illinois are Alorton, Rosemont, and Coal City.

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

```
# create plot for res
plot(res)
```




```
# compute correlation and do correlation test for  median income with coronavirus
corr_test_res <-cor.test(res_city_loans$crime, res_city_loans$loans)
print(corr_test_res)
```

```
##
##  Pearson's product-moment correlation
##
## data:  res_city_loans$crime and res_city_loans$loans
## t = -0.56705, df = 11, p-value = 0.5821
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6583784  0.4216065
## sample estimates:
##        cor
## -0.1685276
```

Analysis :

We can diagnose the relationship of search hits between these two keywords by looking at the plot above. There is a pattern between these two variables, which indicates a relationship. From January to October 2020, generally, they moved to different direction although after that until December 2020 they tend to had same direction. In other words, we can diagnose that there is a negative significant relationship. To make sure, we need to test it using the pearson correlation test. The correlation of hits between `crime` and `loans` is -0.54. There is a significant and negative relationship between the search intensities between `crime` and `loans`. As the search for `crime` is high, the search for `loans` is low.

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.

We try to use `coronavirus` and `hospital` as keywords.

```
set.seed(727)
cov <- gtrends(c("coronavirus", "hospital"),
               geo = "US-IL",
               time = "2020-01-01 2020-12-31",
               low_search_volume = TRUE)
```

1. Compute mean, median, and variance of hits related to `coronavirus` and `hospital`.

```
# change the data format of interest over time into tibble
res_cov <- as_tibble(cov$interest_over_time)
as.numeric(res_cov$hits)
```

```
##   [1]  NA  NA   6  12   7   5   3  13  24  64 100  68  54  52  40  37  19  19
##  [19]  19  14  12   8   9   9  10  12  10  12   9   9   8   6   5   4   5   4
##  [37]   3   4   4   4   4   5   6   4   6   7   4   4   4   3   3   3   4   4
##  [55]   4   5   4   4   4   4   4   4   4   4   4   4   3   3   3   3   3   3
##  [73]   3   4   3   4   4   3   4   4   4   4   4   4   4   4   4   3   3   3
##  [91]   4   4   4   4   4   3   4   4   3   4   4   4   3   3
```

```
# remove the value of hits with less than 1
res_cov_sub <- subset(res_cov, hits >= 1)

# change the format of hits into numeric
res_cov_sub$hits <- as.numeric(res_cov_sub$hits)

# compute mean, median, var of hits
res_cov_sub %>%
  group_by(keyword) %>%
```

```
  summarize(mean_hits=mean(hits),
            med_hits=median(hits),
            var_sd=var(hits))
```

```
## # A tibble: 2 x 4
##   keyword     mean_hits med_hits  var_sd
##   <chr>           <dbl>    <dbl>   <dbl>
## 1 coronavirus     15.1       7.5 402.
## 2 hospital         3.71      4     0.248
```

Analysis:

The mean of `coronavirus` hits is higher than hospital hits, with values of 15.8 and 3.8. A high discrepancy between the mean and median of `coronavirus` hits indicates the extremely high popularity of `coronavirus` searches at certain times. The variance of the `coronavirus` search is higher than the `hospital` search, with values of 426.16 and 0.21.

2. Find cities with highest hits of `coronavirus`

```
# change the data format into wide format
 cov_city <- cov$interest_by_city
 cov_city <-  cov_city[!duplicated(cov_city), ] #remove duplication

  cov_city_hos <- cov_city %>%
  pivot_wider(names_from = keyword,
  values_from = hits) %>%

# sort the data by the value of search hits of coronavirus
arrange(desc(coronavirus))
```

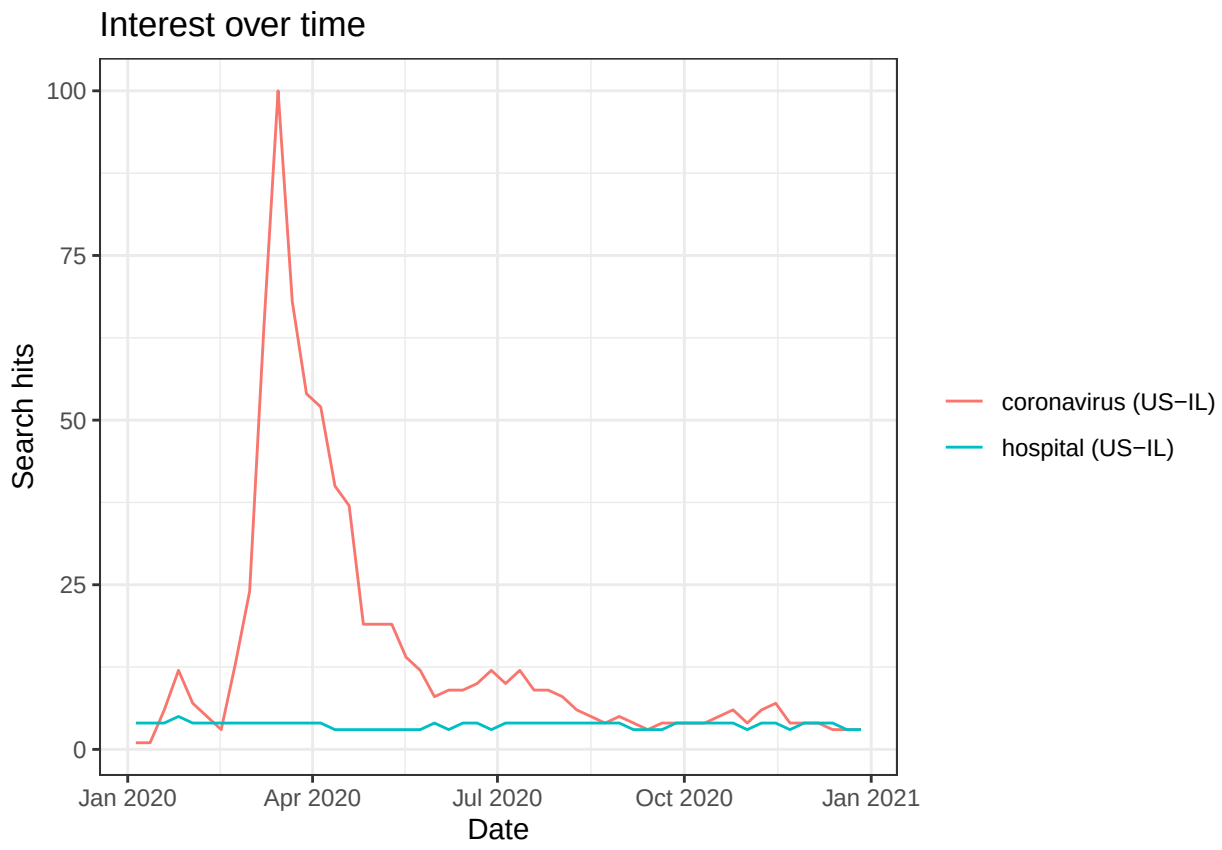
```
# print the result
print(cov_city_hos)
```

```
## # A tibble: 348 x 5
##    location        geo   gprop coronavirus hospital
##    <chr>           <chr> <chr>       <int>    <int>
##  1 Belknap         US-IL web           100       NA
##  2 Barrington      US-IL web            86       NA
##  3 Wheeling        US-IL web            77       NA
##  4 Skokie          US-IL web            75       NA
##  5 Ellis Grove     US-IL web            75       NA
##  6 Hoffman Estates US-IL web            75       NA
##  7 Woodhull        US-IL web            74       NA
##  8 Naperville      US-IL web            74       NA
##  9 Willowbrook     US-IL web            73       NA
## 10 Ashton          US-IL web            73       NA
## # i 338 more rows
```

The cities with highest hits of `coronavirus` in Illinois are Ullin, Clarendon Hills, and Belknap.

3. Check relationship between hits coronavirus and hospital

```
# create plot for cov
plot(cov)
```




```
# compute correlation and do correlation test for  median income with coronavirus
corr_test_cov <-cor.test(cov_city_hos$hospital, cov_city_hos$coronavirus)
print(corr_test_cov)
```

```
##
##  Pearson's product-moment correlation
##
## data:  cov_city_hos$hospital and cov_city_hos$coronavirus
## t = -0.89457, df = 43, p-value = 0.376
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4123427  0.1649077
## sample estimates:
##        cor
## -0.1351692
```

Analysis:

The graph shows the trend of the popularity of `coronavirus` search and `hospital` search in Illinois from January 1, 2020, to December 31, 2020. The plot of `coronavirus` search and `hospital` search popularity

are different over time. There is a very high interest in `coronavirus` searches from February 2020 to May 2020. The popularity of `hospital` search is relatively stable over time. In other words, it means that they do not have a similar pattern. The correlation between `coronavirus` search and `hospital` search is -0.33, indicating a weak negative relationship. If the `coronavirus` search is high, the search of `hospital` tends to be low.

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.

```
# get API key from getCensus
 cs_key <- "7b7f40a6d561e9e43edb14fec7e8b645055a65ed"
```

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.

```
# generate the data with getCensus function
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.

```
# convert the missing value to NA
acs_il[acs_il == -666666666] <- NA

# print the data
head(acs_il)
```

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

```
# rename the column name
acs_il <-
  acs_il %>%
  rename(pop = B01001_001E,
         age = B06002_001E,
         hh_income = B19013_001E,
         income = B19301_001E)
```

Then, we save the ACS data into .csv so that we don't need to generate the data repeatedly. Here, we turn of the code to create .csv file, so the data will not be replaced with new file.

```
#save the acs_il
#write.csv(acs_il, file = "C:/Users/ASUS/Documents/SURVMETH727_2/project2727part2/acs_il.csv")

#read the .csv file from directory
acs_il <- read.csv("C:/Users/ASUS/Documents/SURVMETH727_2/project2727part2/acs_il.csv")
```

```
# show the dataset format
head(acs_il)
```

```
##   X state place                         NAME  pop  age hh_income income
## 1 1    17 15261 Coatsburg village, Illinois  180 35.6     55714  27821
## 2 2    17 15300    Cobden village, Illinois 1018 44.2     38750  19979
## 3 3    17 15352      Coffeen city, Illinois  640 33.4     35781  26697
## 4 4    17 15378   Colchester city, Illinois 1347 42.2     43942  24095
## 5 5    17 15469    Coleta village, Illinois  230 27.7     56875  23749
## 6 6    17 15495    Colfax village, Illinois 1088 32.5     58889  24861
```

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.

```
# clean name of location in the acs_il data
acs_il <- acs_il %>%
  # remove all words after certain character
  mutate(location = gsub("(.*),\\s*(.*?)\\s*$", "\\1", NAME))

  # remove the last word
  acs_il$location <- sub("\\s+\\w+$", "", acs_il$location)

# check the result
head(acs_il)
```

```
##   X state place                         NAME  pop  age hh_income income
## 1 1    17 15261 Coatsburg village, Illinois  180 35.6     55714  27821
## 2 2    17 15300    Cobden village, Illinois 1018 44.2     38750  19979
## 3 3    17 15352      Coffeen city, Illinois  640 33.4     35781  26697
## 4 4    17 15378   Colchester city, Illinois 1347 42.2     43942  24095
## 5 5    17 15469    Coleta village, Illinois  230 27.7     56875  23749
## 6 6    17 15495    Colfax village, Illinois 1088 32.5     58889  24861
##     location
## 1  Coatsburg
## 2     Cobden
## 3    Coffeen
## 4 Colchester
## 5     Coleta
## 6     Colfax
```

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

1. 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 cities in the res2 but don't appear in acs
res2_not_in_acs <- res_city_loans %>%
  anti_join(acs_il, by = "location")

# check cities in the acs but don't appear in res2
acs_il_not_in_res2 <- acs_il %>%
  anti_join(res_city_loans, by = "location")

# compute total cities in the res2 but don't appear in acs
count_res2_not_in_acs <- nrow(res2_not_in_acs)

# compute total cities in the acs but don't appear in res2
count_acs_il_not_in_res2 <- nrow(acs_il_not_in_res2)

# print the result
cat("Cities in 'res2' but not in 'acs_il':", count_res2_not_in_acs, "\n")
```

```
## Cities in 'res2' but not in 'acs_il': 10
```

```
cat("Cities in 'acs_il' but not in 'res2':", count_acs_il_not_in_res2, "\n")
```

```
## Cities in 'acs_il' but not in 'res2': 1118
```

```
cat("Total:",count_res2_not_in_acs+count_acs_il_not_in_res2,"\n")
```

```
## Total: 1128
```

After cleaning the name and matching the location, we found that 1126 cities do not appear in both data sets.

Below, we create a new data set by joining the Google Trends and the ACS data by keeping only cities that appear in both data sets.

```
# join the data of res2 and acs with matched cities
join_res2_acsil <- res_city_loans %>%
  inner_join(acs_il, by = "location")

# check the result
head(join_res2_acsil)
```

```
## # A tibble: 6 x 13
##   location geo   gprop crime loans     X state place NAME    pop   age hh_income
##   <chr>    <chr> <chr> <int> <int> <int> <int> <int> <chr> <int> <dbl>     <int>
## 1 Justice  US-IL web      NA   100   687    17 38830 Just~ 12677  32       57523
```

```
## 2 Roanoke  US-IL web      NA    94  1424    17 64590 Roan~  1880  41.3     64167
## 3 Hinckley US-IL web      NA    84   610    17 35268 Hinc~  2190  37.1     70278
## 4 Rosevil~ US-IL web      NA    83  1448    17 65845 Rose~   852  49.8     42368
## 5 Carrier~ US-IL web      NA    81  1359    17 11397 Carr~  1819  40       42711
## 6 Long La~ US-IL web      NA    80   798    17 44550 Long~  3065  36.4     95764
## # i 1 more variable: income <int>
```

2. 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?

```r
# create a grouping variable based on above/below average median "hh_income"
join_res2_acsil <- join_res2_acsil %>%
  mutate(income_group = ifelse(hh_income > mean(hh_income, na.rm = TRUE),
                               "Above Average", "Below Average"))

# remove the observation with NA value of hh_income
join_res2_acsil <- join_res2_acsil[complete.cases(join_res2_acsil[, "hh_income"]), ]

# group by the "income_group" variable and calculate the mean for "crime" and "loans"
mean_bygroup <- join_res2_acsil %>%
  group_by(income_group) %>%
  summarise(
    Mean_Crime_Pop = mean(crime, na.rm = TRUE),
    Mean_Loans_Pop = mean(loans, na.rm = TRUE)
  )

# print the result
print(mean_bygroup)
```

```
## # A tibble: 2 x 3
##   income_group  Mean_Crime_Pop Mean_Loans_Pop
##   <chr>                  <dbl>          <dbl>
## 1 Above Average           45.2           41.1
## 2 Below Average           51.7           43.9
```

Analysis:

Based on the information above, we can see that the average search popularity of `crime` and `loans` is higher for cities with a median household income below the average. In other words, people living in cities with median household incomes below the average searched these two keywords more frequently.

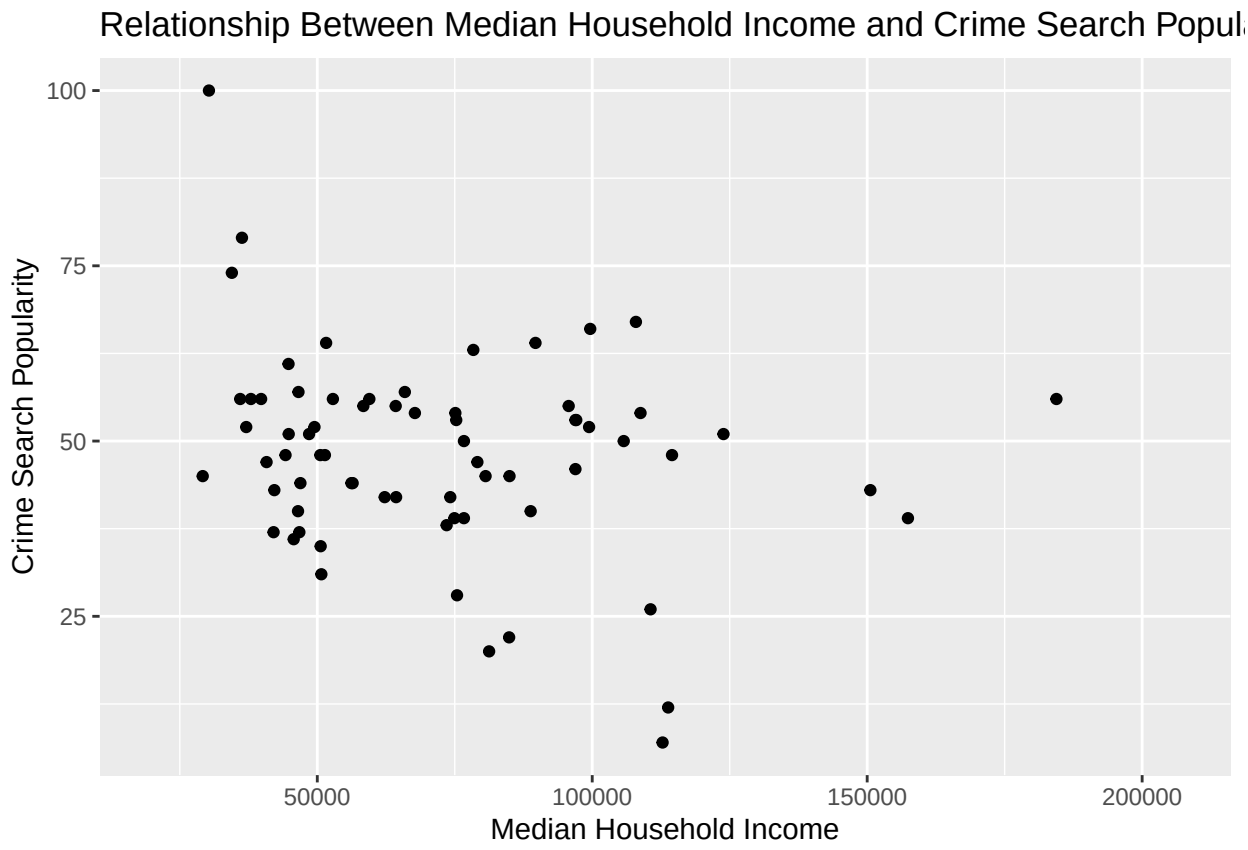
3. Is there a relationship between the median household income and the search popularity of the Google trends terms? Describe the relationship and use a scatterplot with qplot().

```r
# Create a scatterplot to visualize the relationship
sc_plot1 <- qplot(x = hh_income, y = crime, data = join_res2_acsil,
          main = "Relationship Between Median Household Income and Crime Search Popularity",
          xlab = "Median Household Income", ylab = "Crime Search Popularity")
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

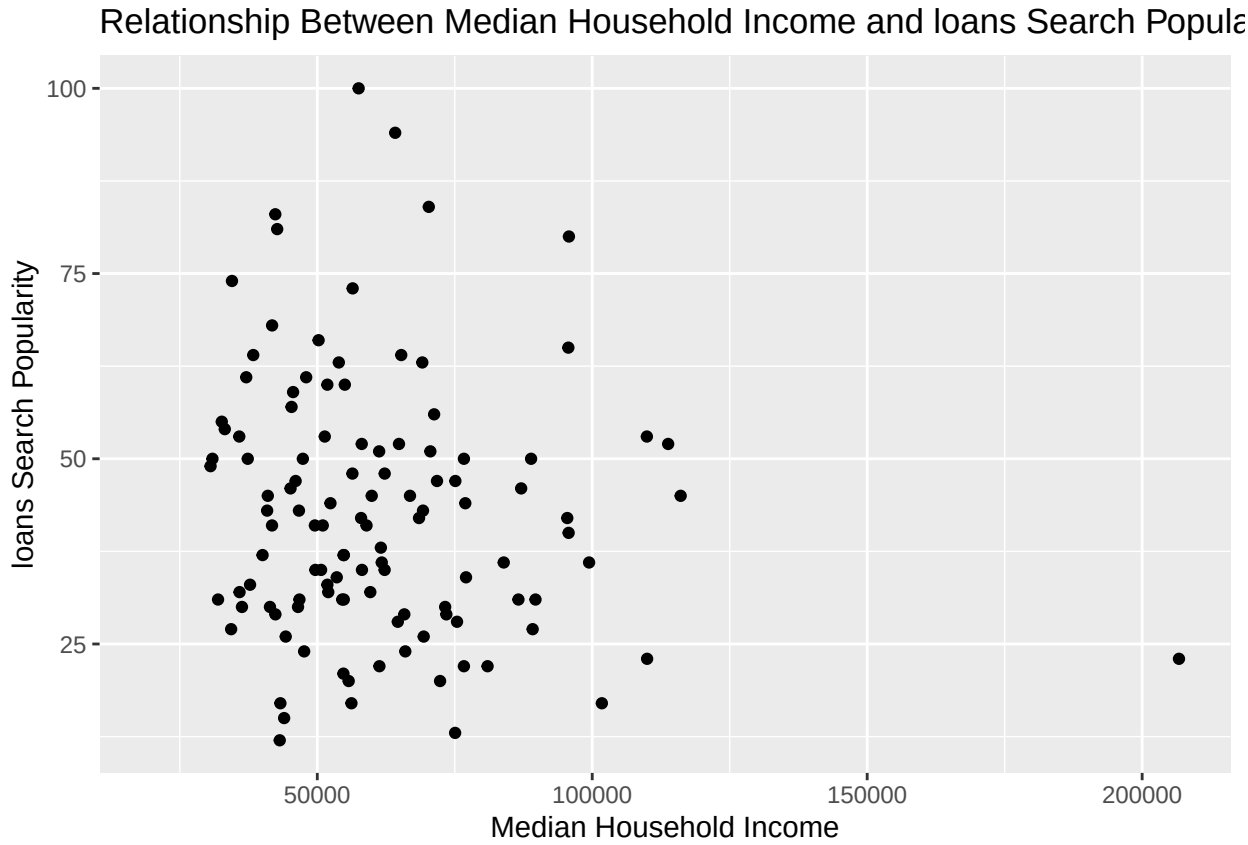
```
# Create a scatterplot to visualize the relationship
sc_plot2 <- qplot(x = hh_income, y = loans, data = join_res2_acsil,
          main = "Relationship Between Median Household Income and loans Search Popularity",
          xlab = "Median Household Income", ylab = "loans Search Popularity")

# Print the scatterplot
print(sc_plot1)
```

```
## Warning: Removed 278 rows containing missing values (`geom_point()`).
```




```
print(sc_plot2)
```

```
## Warning: Removed 232 rows containing missing values (`geom_point()`).
```



Analysis:

Based on the scatterplot above, it can be seen that for the keyword `loans`, the data tends to gather in the lower area of median household income (hh_income) and less in the higher hh_income range. Meanwhile, `crime` spreads across a wider range than hh_income and shows no discernible pattern. So it can be diagnosed that the hh_income variable may have a significant negative relationship with the frequency of searches for `loans`. In contrast, hh_income and `crime` are not significantly related even though they appear to have a negative direction.

```
# compute correlation and do correlation test for  median income with crime
correlation_test2 <- cor.test(join_res2_acsil$hh_income, join_res2_acsil$crime)
print(correlation_test2)
```

```
##
##  Pearson's product-moment correlation
##
## data:  join_res2_acsil$hh_income and join_res2_acsil$crime
## t = -1.9661, df = 67, p-value = 0.05343
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.445605651  0.003306069
## sample estimates:
##        cor
## -0.2335574
```

```r
# compute correlation and do correlation test for median income with loans
correlation_test3 <- cor.test(join_res2_acsil$hh_income, join_res2_acsil$loans)
print(correlation_test3)
```

```
##
##  Pearson's product-moment correlation
##
## data:  join_res2_acsil$hh_income and join_res2_acsil$loans
## t = -1.3201, df = 113, p-value = 0.1895
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.29958479  0.06125819
## sample estimates:
##        cor
## -0.1232346
```

Analysis:

The result shows that the correlation coefficient between the median income and `crime` is -0.01, and it is statistically not significant. This value indicates a weak and negative relationship between the two variables. If the median income is high, the search for `crime` tends to be high. On the other hand, the correlation between median income and `loan` is -0.27. This value shows a weak and negative relationship. This relationship is statistically insignificant.

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

1. Check how many cities don't appear in both data sets

```r
# check cities in the cov_city_hos but don't appear in acs
cov2_not_in_acs <- cov_city_hos %>%
  anti_join(acs_il, by = "location")

# check cities in the acs but don't appear in cov_city_hos
acs_il_not_in_cov2 <- acs_il %>%
  anti_join(cov_city_hos, by = "location")

# compute total cities in the cov_city_hos but don't appear in acs
count_cov2_not_in_acs <- nrow(cov2_not_in_acs)

# compute total cities in the acs but don't appear in cov_city_hos
count_acs_il_not_in_cov2 <- nrow(acs_il_not_in_cov2)

# print the result
cat("Cities in 'cov2' but not in 'acs_il':", count_cov2_not_in_acs, "\n")
```

```
## Cities in 'cov2' but not in 'acs_il': 9
```

```r
cat("Cities in 'acs_il' but not in 'cov2':", count_acs_il_not_in_cov2, "\n")
```

```
## Cities in 'acs_il' but not in 'cov2': 1123
```

```r
cat("Total:",count_cov2_not_in_acs+count_acs_il_not_in_cov2,"\n")
```

```
## Total: 1132
```

Here, we found 1133 cities that do not appear in both data sets.

Below, we create a new data set by joining the Google Trends and the ACS data by keeping only cities that appear in both data sets.

```r
# join the data of cov_city_hos and acs with matched cities
join_cov2_acsil <- cov_city_hos %>%
  inner_join(acs_il, by = "location")

# check the result
head(join_cov2_acsil)
```

```
## # A tibble: 6 x 13
##   location  geo   gprop coronavirus hospital     X state place NAME    pop   age
##   <chr>     <chr> <chr>       <int>    <int> <int> <int> <int> <chr> <int> <dbl>
## 1 Belknap   US-IL web           100       NA  1067    17  4715 Belk~   142  50.1
## 2 Barringt~ US-IL web            86       NA  1045    17  3844 Barr~ 10442  40.8
## 3 Wheeling  US-IL web            77       NA   530    17 81087 Whee~ 38684  38.4
## 4 Skokie    US-IL web            75       NA   174    17 70122 Skok~ 63300  42.9
## 5 Ellis Gr~ US-IL web            75       NA   246    17 23503 Elli~   316  36.9
## 6 Hoffman ~ US-IL web            75       NA   615    17 35411 Hoff~ 50464  38.2
## # i 2 more variables: hh_income <int>, income <int>
```

2. 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.

```r
# create a grouping variable based on above/below average median "hh_income"
join_cov2_acsil <- join_cov2_acsil %>%
  mutate(income_group = ifelse(hh_income > mean(hh_income, na.rm = TRUE),
                               "Above Average", "Below Average"))

# remove the observation with NA value of hh_income
join_cov2_acsil <- join_cov2_acsil[complete.cases(join_cov2_acsil[, "hh_income"]), ]

# group by the "income_group" variable and calculate the mean for "coronavirus" and "hospital"
mean_bygroup <- join_cov2_acsil %>%
  group_by(income_group) %>%
  summarise(
    Mean_Coronavirus_Pop = mean(coronavirus, na.rm = TRUE),
    Mean_Hospital_Pop = mean(hospital, na.rm = TRUE)
  )

# print the result
print(mean_bygroup)
```

```
## # A tibble: 2 x 3
##   income_group  Mean_Coronavirus_Pop Mean_Hospital_Pop
##   <chr>                        <dbl>             <dbl>
## 1 Above Average                 58.3              28.3
## 2 Below Average                 51.2              29.0
```

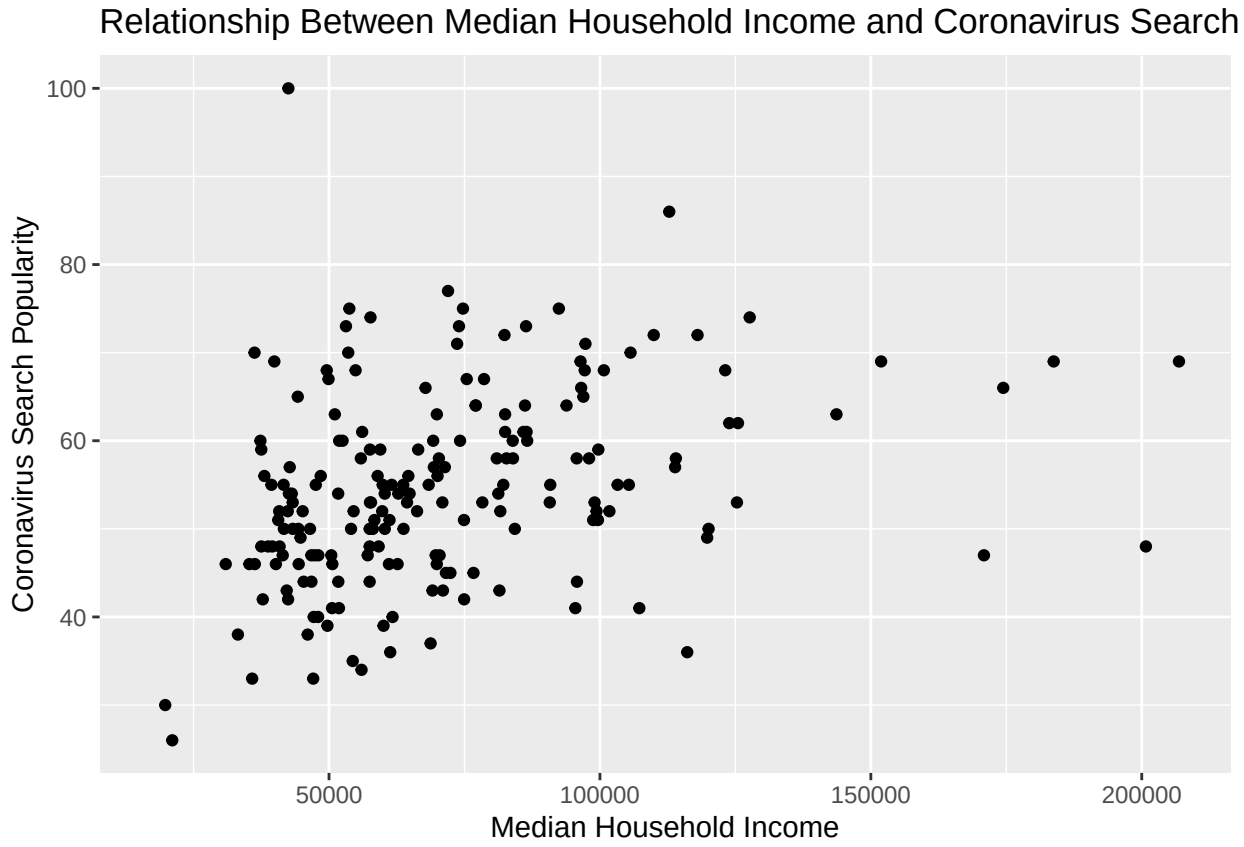
Analysis :

Based on the table above, we can see that the average search for `coronavirus` and `hospital` is higher for cities with median household income above the average. People living in cities with a median household income above the average search the two keywords more frequently.

3. Check the relationship between the median household income and the search popularity of the Google trends terms. Describe the relationship and use a scatterplot with `qplot()`.

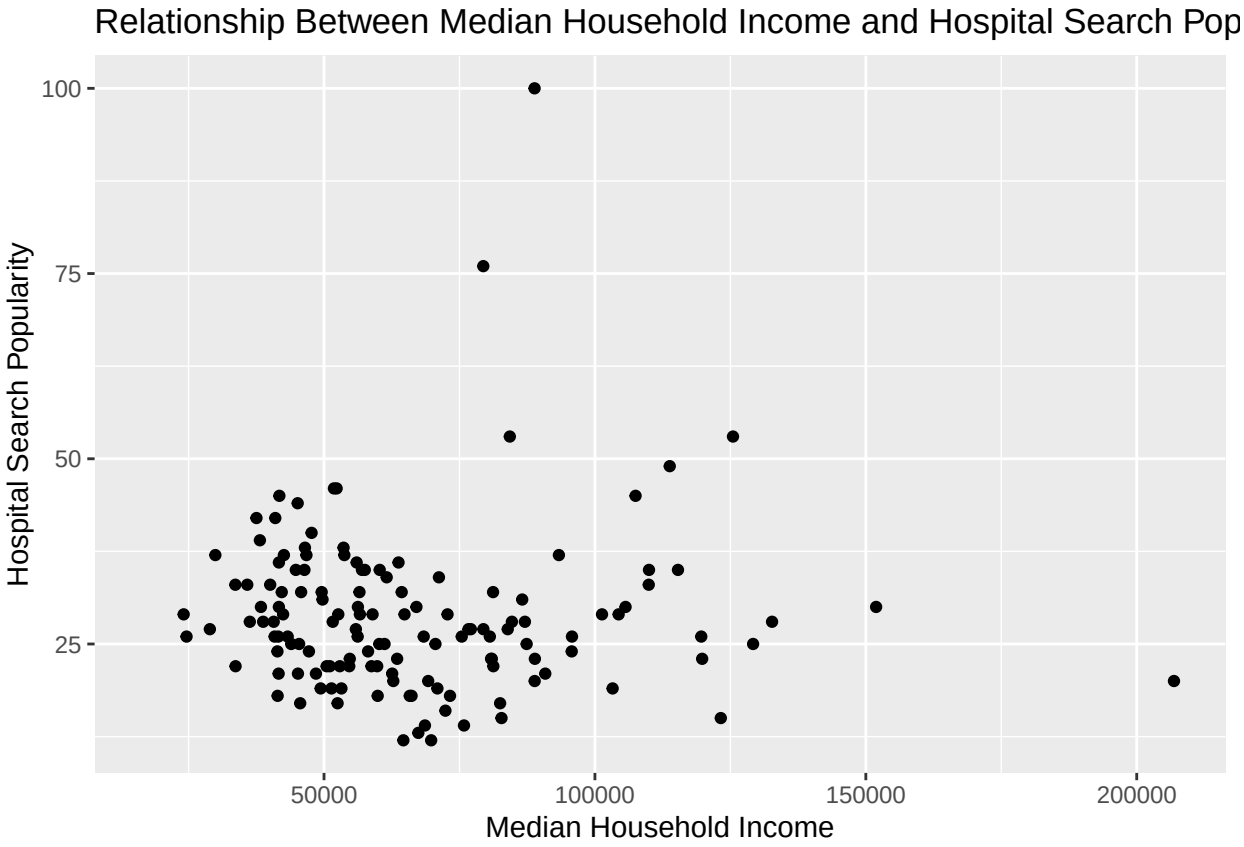
```
# Create a scatterplot to visualize the relationship
sc_plot1 <- qplot(x = hh_income, y = coronavirus, data = join_cov2_acsil,
          main = "Relationship Between Median Household Income and Coronavirus Search Popularity",
          xlab = "Median Household Income", ylab = "Coronavirus Search Popularity")

# Create a scatterplot to visualize the relationship
sc_plot2 <- qplot(x = hh_income, y = hospital, data = join_cov2_acsil,
          main = "Relationship Between Median Household Income and Hospital Search Popularity",
          xlab = "Median Household Income", ylab = "Hospital Search Popularity")

# Print the scatterplot
print(sc_plot1)
```

```
## Warning: Removed 147 rows containing missing values (‘geom_point()‘).
```




```
print(sc_plot2)
```

```
## Warning: Removed 204 rows containing missing values (‘geom_point()‘).
```




```
# compute correlation and do correlation test for  median income with coronavirus
correlation_test2<-cor.test(join_cov2_acsil$hh_income, join_cov2_acsil$coronavirus)
print(correlation_test2)
```

```
##
##  Pearson's product-moment correlation
##
## data:  join_cov2_acsil$hh_income and join_cov2_acsil$coronavirus
## t = 4.8454, df = 193, p-value = 2.59e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1979691 0.4490538
## sample estimates:
##       cor
## 0.3293207
```

```
# compute correlation and do correlation test for median income with hospital
correlation_test3<-cor.test(join_cov2_acsil$hh_income, join_cov2_acsil$hospital)
print(correlation_test3)
```

```
##
##  Pearson's product-moment correlation
##
## data:  join_cov2_acsil$hh_income and join_cov2_acsil$hospital
## t = 0.073082, df = 136, p-value = 0.9418
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1610068  0.1731901
## sample estimates:
##         cor
## 0.006266614
```

Analysis:

The scatter plot shows a slightly increasing pattern between median household income and the `coronavirus` popularity search. The distribution of median household income was concentrated from 0 to 75,000. The correlation value confirms this finding with a value of 0.27. This correlation score means that the search hits for `coronavirus` have a weak and positive relationship with the median household income. If the median household income is high, the value of search hits of `coronavirus` is also high.

The scatterplot between search hits of `hospital` and median household income shows a linear pattern. There is not any specific trend between the two variables. The correlation coefficient is insignificant, with a value of 0.00033, which means there is not any relationship between search hits of `hospital` and the median household income.