Assignment2_727

Akari & Zhuoer

2023-10-03

```
#load libraries
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 become errors
```

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

```
##
## Attaching package: 'censusapi'
##
## The following object is masked from 'package:methods':
##
##     getFunction
```

Github link = https://github.com/ZuorW/SURV727.git

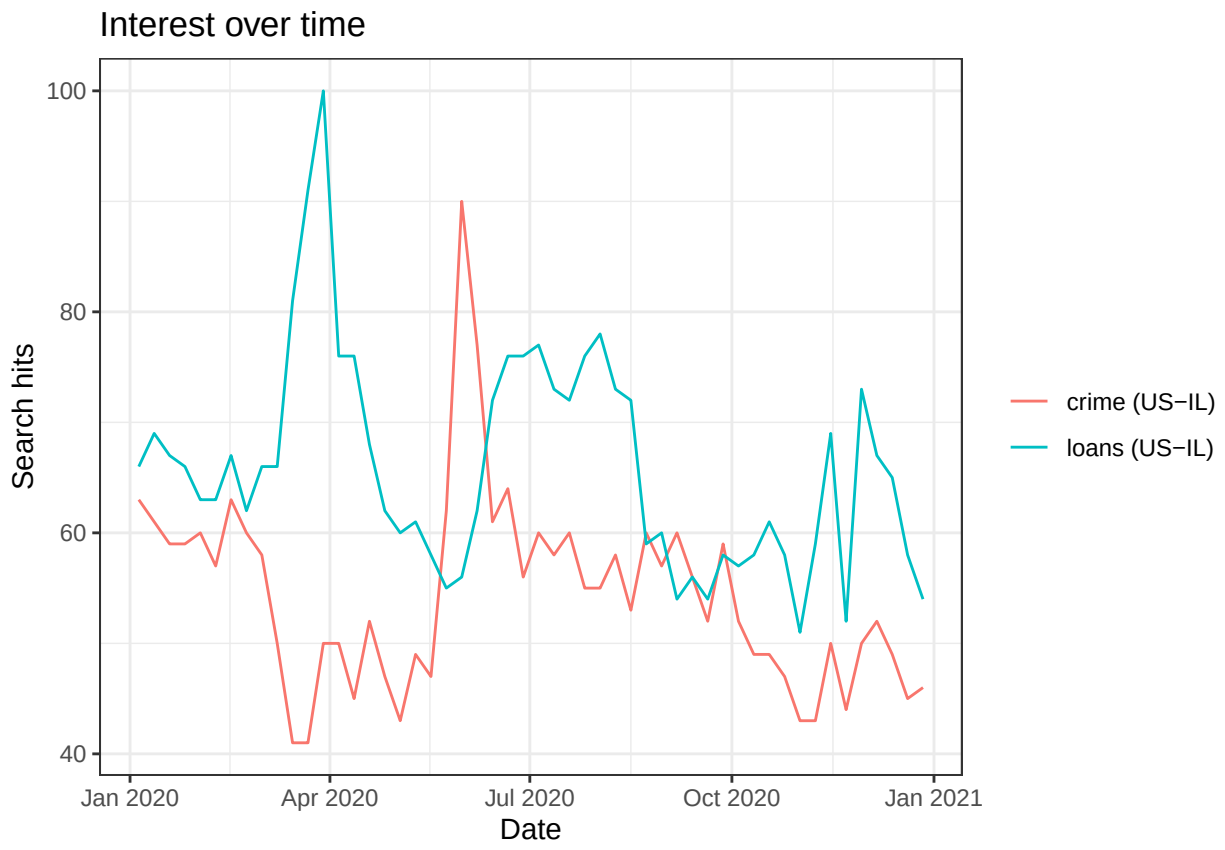
In this assignment, you will pull from APIs to get data from various data sources and use your data wrangling skills to use them all together. You should turn in a report in PDF or HTML format that addresses all of the questions in this assignment, and describes the data that you pulled and analyzed.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

```
# Check what the data looks like
#res$interest_over_time %>% head()

#transform the data.frame into tibble
res_time = as_tibble(res$interest_over_time)


# Also compute the mean, sd, variance of each keyword
res_time %>%
  group_by(keyword) %>%
  summarize(mean_hits = mean(hits),
            median = median(hits),
            var_hits = var(hits))
```

```
## # A tibble: 2 x 4
```

```
##   keyword mean_hits median var_hits
##   <chr>       <dbl>  <dbl>    <dbl>
## 1 crime        54.4   54       78.0
## 2 loans        65.9   65.5     96.3
```

The keyword `crime` had a mean search hit of 54.4 with a median of 54 and a variance of 78.0 The keyword `loans` had a mean search hit of 65.9 with a median of 65.5 and a variance of 96.3

- 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.

```
#transform the data.frame into tibble
rest_city <- tibble(res$interest_by_city)

# Reshape the data & Sort loans column in descending order
city_ranking <- rest_city %>%
  pivot_wider(names_from = keyword, values_from = hits) %>%
  arrange(., desc(loans))

#display first few rows of the ranking to find the highest searched
head(city_ranking)
```

```
## # A tibble: 6 x 5
##   location  geo   gprop crime loans
##   <chr>     <chr> <chr> <int> <int>
## 1 Alorton   US-IL web      NA   100
## 2 Rosemont  US-IL web      38    53
## 3 Coal City US-IL web      25    51
## 4 Cobden    US-IL web      NA    49
## 5 Dolton    US-IL web      NA    46
## 6 Irving    US-IL web      NA    46
```

```
wide <-
  rest_city %>%
  pivot_wider(names_from = keyword,
              values_from = hits)
```

The cities Alorton, Rosemont,and Coal City have the highest search frequency for `loans`.

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

```
# Run Pearson correlation test
cor.test(wide$loans,wide$crime)
```

```
##
##  Pearson's product-moment correlation
##
```

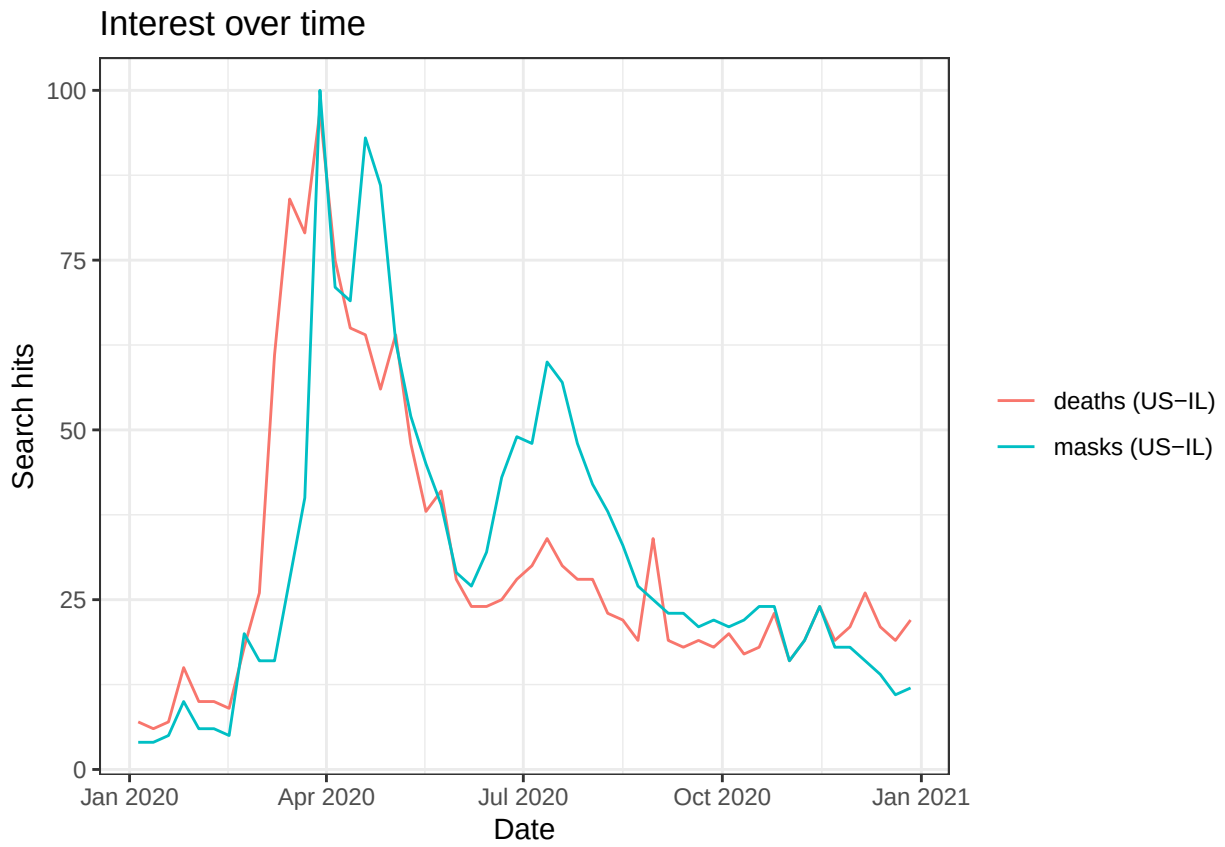
```
## data:  wide$loans and wide$crime
## t = -2.5934, df = 16, p-value = 0.0196
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8061395 -0.1034120
## sample estimates:
##       cor
## -0.544017
```

While `loans` had a higher mean search frequency over time, there does not seem to be a large difference between the search frequency of `crime`. However, patterns can be seen in the plot. The two keywords seems to have an inverse relationship where search frequencies for `loans` are high when `crime` is low in the first peak/dip around April 2020. However, the pattern fades after around July 2020. We tested the relationship between the variables for interest by city to properly with a Pearson correlation test. The test suggests that there is a negative correlation between loans and crime (r = -0.544, p = 0.0196).

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.

Key Words: masks & deaths

```
# Create another dataset with new keywords
res_2 <- gtrends(c("masks", "deaths"),
                 geo = "US-IL",
                 time = "2020-01-01 2020-12-31",
                 low_search_volume = TRUE)
plot(res_2)
```



In the plot, the frequencies have a similar shape in the first half of 2020, but the pattern becomes unclear in the second half. The initial spike in search hits of both keywords understandably corresponds to near the beginning of the pandemic when everyone was required to wear masks.

```
#check data
#res_2 %>% head()

#transform data into tibble
res_time2 <- tibble(res_2$interest_over_time)

# Compute the mean, standard deviation, and variance of search hits per keyword
res_time2 %>%
  group_by(keyword) %>%
  summarize(mean_hits = mean(hits),
            median_hits = median(hits),
            var_hits = var(hits))
```

```
## # A tibble: 2 x 4
##   keyword mean_hits median_hits var_hits
##   <chr>       <dbl>       <dbl>    <dbl>
## 1 deaths       30.7        23.5     450.
## 2 masks        32          24       528.
```

The search frequency for `masks` over time had a mean of 32 with a median of 24 and a variance of 528. The search frequency for `deaths` over time had a mean of 30.7 with a median of 23.5 and a variance of 450. Both keywords have a similar mean and have high variances.

```r
# Transform data into tibble
rest_city2 <- res_2$interest_by_city

# Check data
rest_city2 %>%
  arrange(desc(location)) %>%
  glimpse()
```

```
## Rows: 400
## Columns: 5
## $ location <chr> "Yorkville", "Wyanet", "Worden", "Woodlawn", "Woodhull", "Won~
## $ hits     <int> 41, NA, NA, NA, NA, 41, NA, NA, 88, 51, 33, 61, 49, 51, 36, 3~
## $ keyword  <chr> "deaths", "deaths", "masks", "deaths", "masks", "deaths", "de~
## $ geo      <chr> "US-IL", "US-IL", "US-IL", "US-IL", "US-IL", "US-IL", "US-IL"~
## $ gprop    <chr> "web", "web", "web", "web", "web", "web", "web", "web", "web"~
```

```r
#highest search frequency for "masks"
city_ranking2_masks <- rest_city2 %>%
  pivot_wider(names_from = keyword, values_from = hits) %>%
  arrange(., desc(masks))
head(city_ranking2_masks)
```

```
## # A tibble: 6 x 5
##   location   geo   gprop masks deaths
##   <chr>      <chr> <chr> <int>  <int>
## 1 Geneva     US-IL web     100     NA
## 2 Winnetka   US-IL web      88     51
## 3 Lanark     US-IL web      83     NA
## 4 Hudson     US-IL web      83     NA
## 5 Lake Bluff US-IL web      76     NA
## 6 Northfield US-IL web      74     49
```

```r
#highest search frequency for "deaths"
city_ranking2_deaths <- rest_city2 %>%
  pivot_wider(names_from = keyword, values_from = hits) %>%
  arrange(., desc(deaths))
head(city_ranking2_deaths)
```

```
## # A tibble: 6 x 5
##   location   geo   gprop masks deaths
##   <chr>      <chr> <chr> <int>  <int>
## 1 Hebron     US-IL web      44    100
## 2 Camanche   US-IL web      NA     96
## 3 Galena     US-IL web      NA     90
## 4 Carthage   US-IL web      NA     85
## 5 Glasford   US-IL web      NA     70
## 6 New Athens US-IL web      41     69
```

We found that Hebron has the highest search frequency for the keyword “deaths” followed by Camanche and Galena. For the keyword “masks”, Geneva has the highest search frequency followed by Winnetka and Lanark.

```r
# Reshape data from long to wide format using keywords
wide_2 <-
  rest_city2 %>%
  pivot_wider(names_from = keyword,
              values_from = hits)

# Run Pearson correlation test
cor.test(wide_2$masks, wide_2$deaths)
```

```
##
##  Pearson's product-moment correlation
##
## data:  wide_2$masks and wide_2$deaths
## t = 1.6067, df = 34, p-value = 0.1174
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.0689042  0.5464868
## sample estimates:
##       cor
## 0.2656451
```

We conducted a Pearson correlation test to see if the search frequencies of the two keywords have a relationship. The test revealed that there is no significant correlation between masks and deaths($r = 0.266$, $p = 0.547$).

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.

```r
cs_key <- "7b1cc9af0a42634e3ba57f9a8f5d0098cdedc5e4"
```

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.

```r
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 %>%
  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.

```
# Check headers
#acs_il %>% head()

# Create new location variable
no_village <- gsub(' village, Illinois', '', acs_il$NAME) #remove "village, IL" from NAME and store
no_cityvill <- gsub(' city, Illinois', '', no_village) #take above and remove remaining "city, IL"
acs_with_loc <-
  acs_il %>%
  mutate(location = no_cityvill) #add new variable with only city names

acs_with_loc %>%
  head()
```

```
##   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
## 6    17 15495    Colfax village, Illinois 1088 32.5     58889  24861     Colfax
```

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 cities cannot be matched from ACS data
acs_with_loc %>%
  anti_join(wide, by = "location") %>%
  count() #show number of rows (cities)
```

```
##      n
## 1 1132
```

```
# Merge ACS to gtrends data by city only keeping cases that match
merged <-
  wide %>%
  inner_join(acs_with_loc, by = "location")

nrow(merged)
```

```
## [1] 334
```

```
#cites not in both data sets
n = nrow(acs_with_loc) - nrow(merged) -(nrow(wide)-nrow(merged))
n
```

```
## [1] 1113
```

There is 1113 cities that don't appear in both 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?

```
# If household income is greater than its median, name group as above average, if not, name group as ab
# Then compute mean by group
merged %>%
  group_by(
    hhinc_med =
      ifelse(hh_income > mean(hh_income, na.rm = TRUE),
                        "above", "below")) %>%
                        summarize(mean_crime = mean(crime, na.rm = TRUE),
                        mean_loans = mean(loans, na.rm = TRUE)) #code doesn't work if I don't use na.rm
```

```
## # A tibble: 2 x 3
##   hhinc_med mean_crime mean_loans
##   <chr>          <dbl>      <dbl>
## 1 above           45.1       25.2
## 2 below           43.8       29.0
```

For cities that have an above average median household income, the search popularity of `crime` was 45.1 and 25.2 for `loans`. For cities that have a below average median household income, the search popularity of `crime` was 43.8 and 29.0 for `loans`. Those in cities with below average household income had a higher search rate for both keywords.
???? We conclude that crime rates may be higher in below average cities which may lead to more search hits, and that people in these cities may search for `loans` more because more people in these cities may take out loans to support their lives due to a lower financial status.

- 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()`.

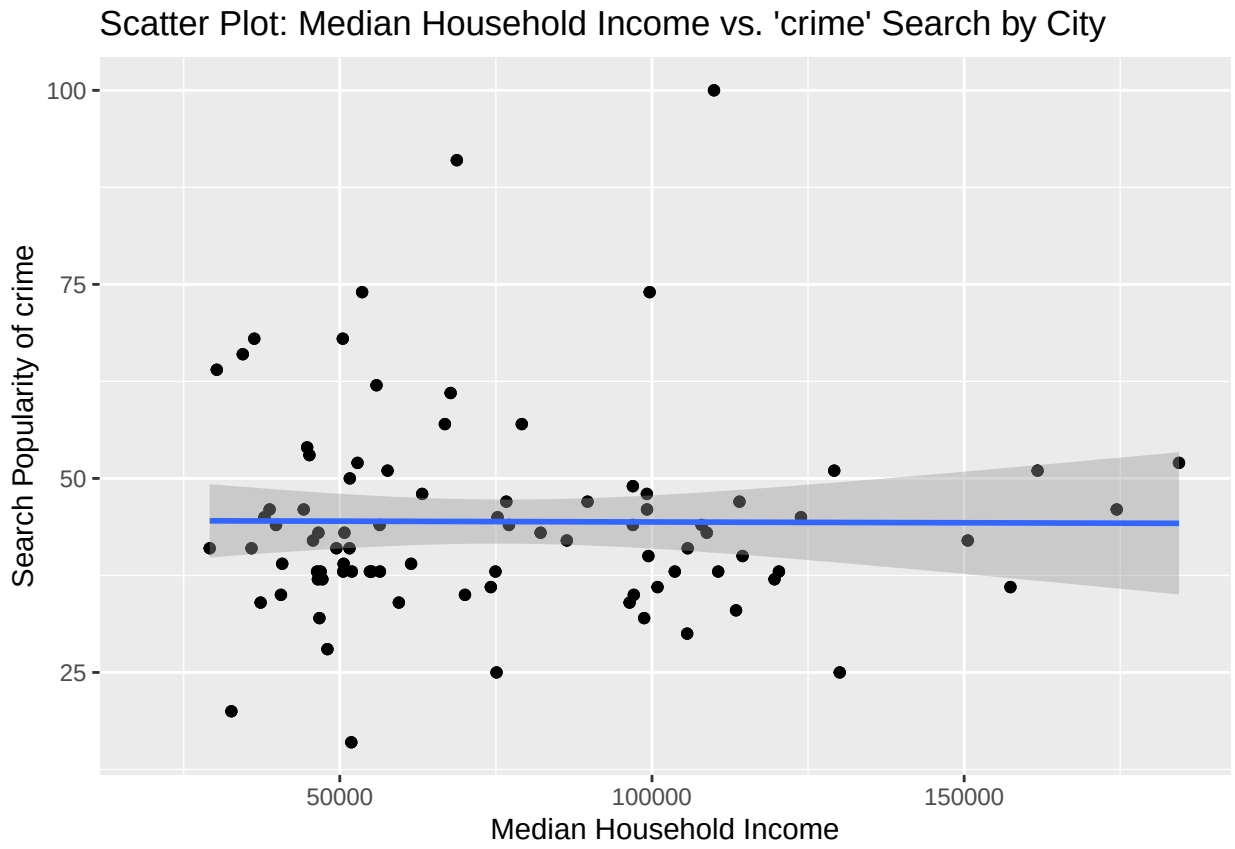
```
# Plot for crime
qplot(hh_income, crime, data = merged)+
  geom_point() +
  geom_smooth(method = lm) +
  labs(title = "Scatter Plot: Median Household Income vs. 'crime' Search by City",
       x = "Median Household Income",y = "Search Popularity of crime")
```

```
## 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.
```

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

```
## Warning: Removed 247 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 247 rows containing missing values (`geom_point()`).
## Removed 247 rows containing missing values (`geom_point()`).
```




```
# Correlation test
cor.test(merged$hh_income, merged$crime)
```

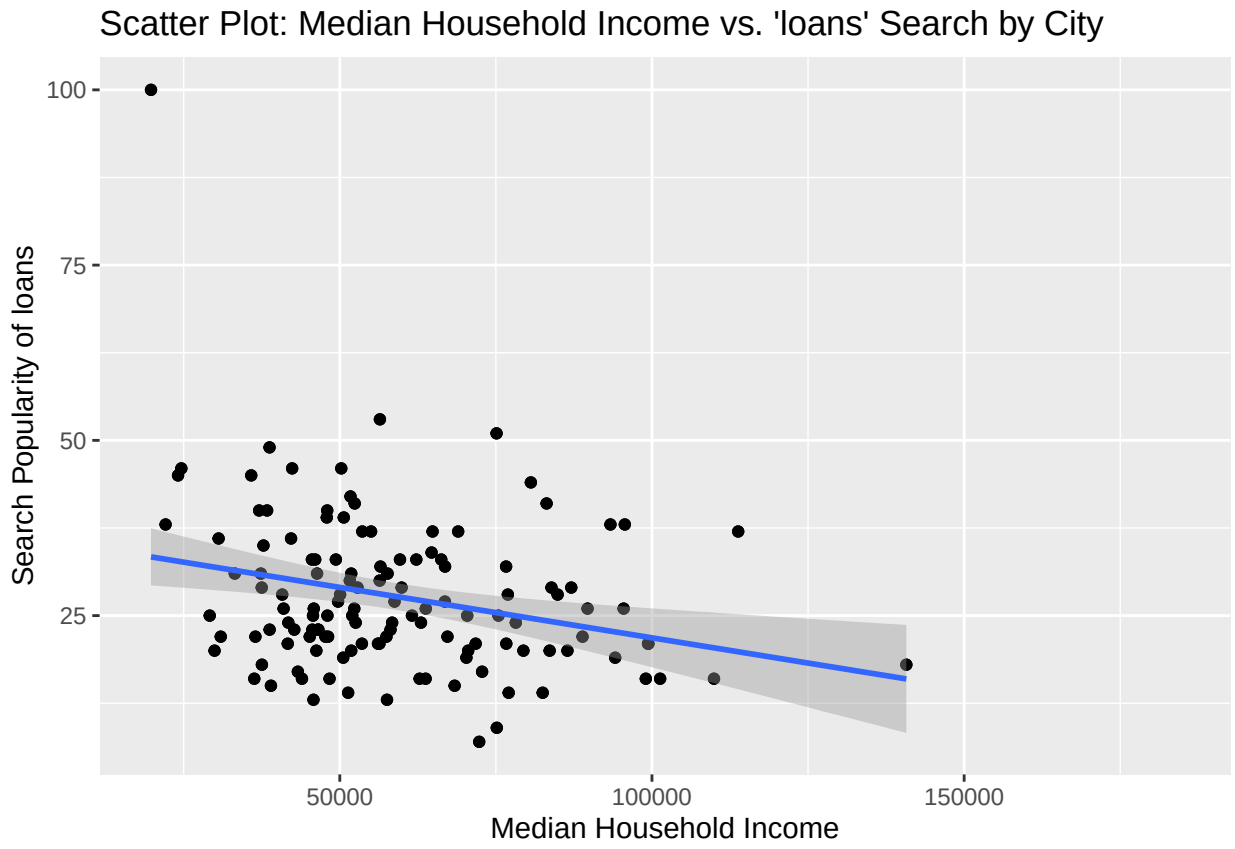
```
##
##  Pearson's product-moment correlation
##
## data:  merged$hh_income and merged$crime
## t = -0.05156, df = 85, p-value = 0.959
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2159862  0.2052977
## sample estimates:
##          cor
## -0.005592347
```

```
# Plot for loans
qplot(hh_income, loans, data = merged) +
  geom_point() +
  geom_smooth(method = lm) +
  labs(title = "Scatter Plot: Median Household Income vs. 'loans' Search by City",
       x = "Median Household Income",y = "Search Popularity of loans")
```

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

## Warning: Removed 205 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 205 rows containing missing values (`geom_point()`).
## Removed 205 rows containing missing values (`geom_point()`).
```




```
# Correlation test
cor.test(merged$hh_income, merged$loans)
```

```
##
##  Pearson's product-moment correlation
##
## data:  merged$hh_income and merged$loans
## t = -3.1164, df = 127, p-value = 0.002264
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.42003323 -0.09819994
## sample estimates:
##        cor
## -0.2665302
```

In the plot for the median household income and the search popularity of `crime`, much of the data is gathered in the lower half of the median household income but there is no clear pattern. A Pearson correlation test supports this by showing that there is no correlation between the two variables ($r = -0.108$, $p = .352$). On the other hand, the plot for `loans` shows a clear pattern in which higher search hits are centered around the lower end of median household income, suggesting a relationship between the two variables. We tested this relationship using a Pearson correlation test. There was a significant correlation ($r = -0.344$, $p < .001$).

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

```
# Check how many cities cannot be matched from ACS data to covid gtrends data
acs_with_loc %>%
  anti_join(wide_2, by = "location") %>%
  count() #show number of rows (cities)
```

```
##      n
## 1 1147
```

```
# Merge ACS to gtrends data by city only keeping cases that match
merged_2 <-
  wide_2 %>%
  inner_join(acs_with_loc, by = "location")

merged_2 %>%
  head()
```

```
## # A tibble: 6 x 12
##   location   geo   gprop masks deaths state place NAME       pop   age hh_income
##   <chr>      <chr> <chr> <int>  <int> <chr> <chr> <chr>    <dbl> <dbl>     <dbl>
## 1 Geneva     US-IL web     100     NA 17    28872 Geneva ~ 21843  40.4    116083
## 2 Winnetka   US-IL web      88     51 17    82530 Winnetk~ 12361  42.1    250001
## 3 Lanark     US-IL web      83     NA 17    41859 Lanark ~  1453  43.9     47917
## 4 Hudson     US-IL web      83     NA 17    36438 Hudson ~  2128  35       96538
## 5 Lake Bluff US-IL web      76     NA 17    40910 Lake Bl~  5540  45      174444
## 6 Northfield US-IL web      74     49 17    53663 Northfi~  5678  52.3    143661
## # i 1 more variable: income <dbl>
```

```
nrow(merged_2)
```

```
## [1] 319
```

```
#cites not in both data sets
n = nrow(acs_with_loc) - nrow(merged) -(nrow(wide_2)-nrow(merged))
n
```

```
## [1] 1121
```

```
# If household income is greater than its median, name group as above average, if not, name group as ab
# Then compute mean by group
merged_2 %>%
  group_by(
    hhinc_med =
      ifelse(hh_income > median(hh_income, na.rm = TRUE),
                        "above", "below")) %>%
                        summarize(mean_masks = mean(masks, na.rm = TRUE),
                        mean_deaths = mean(deaths, na.rm = TRUE)) #code doesn't work if I don't use na.r
```

```
## # A tibble: 2 x 3
##   hhinc_med mean_masks mean_deaths
##   <chr>          <dbl>       <dbl>
## 1 above           48.1        43.7
## 2 below           45.1        38.6
```

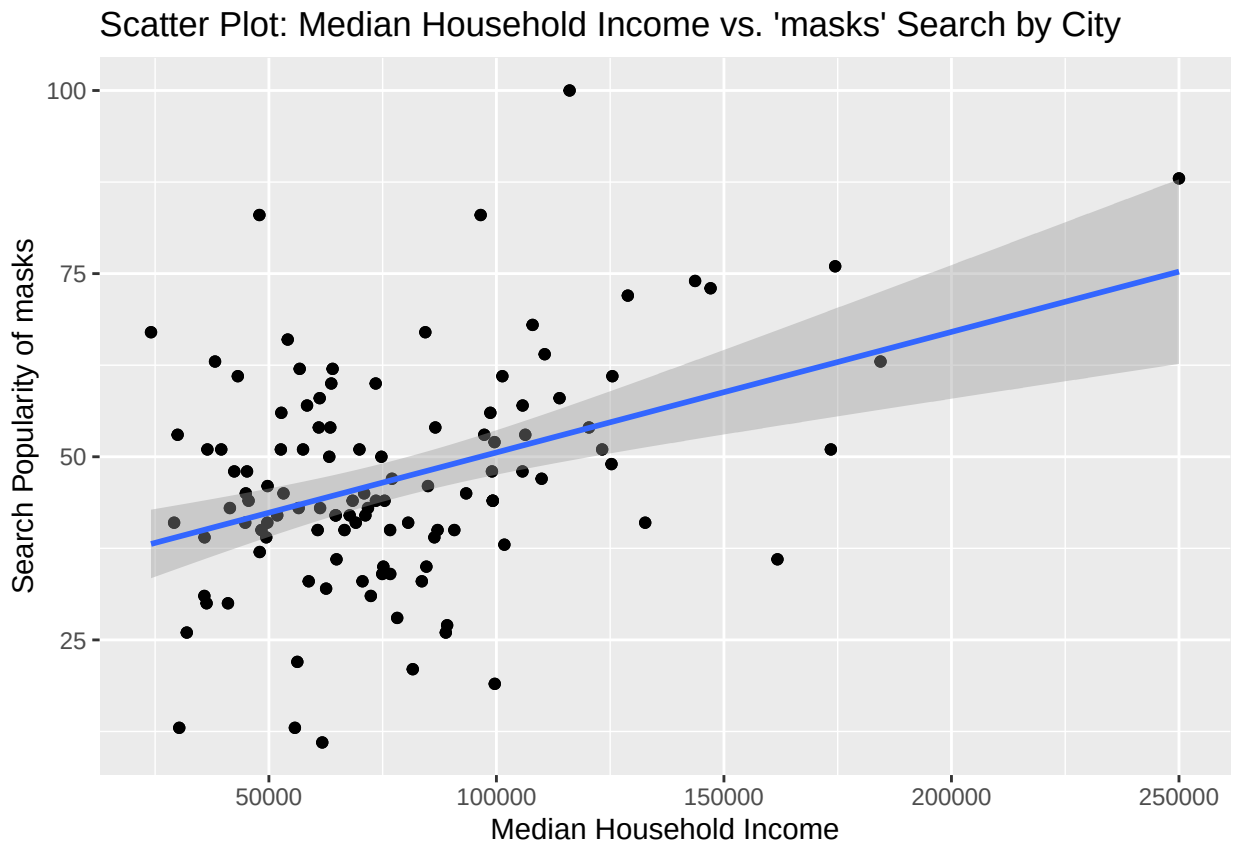
It is weird that poeple living in above average median household income have higher search frequency for both deaths and masks.

```
# Plot for masks
qplot(hh_income, masks, data = merged_2) +
  geom_point() +
  geom_smooth(method = lm) +
  labs(title = "Scatter Plot: Median Household Income vs. 'masks' Search by City",
       x = "Median Household Income", y = "Search Popularity of masks")
```

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

```
## Warning: Removed 205 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 205 rows containing missing values (`geom_point()`).
## Removed 205 rows containing missing values (`geom_point()`).
```

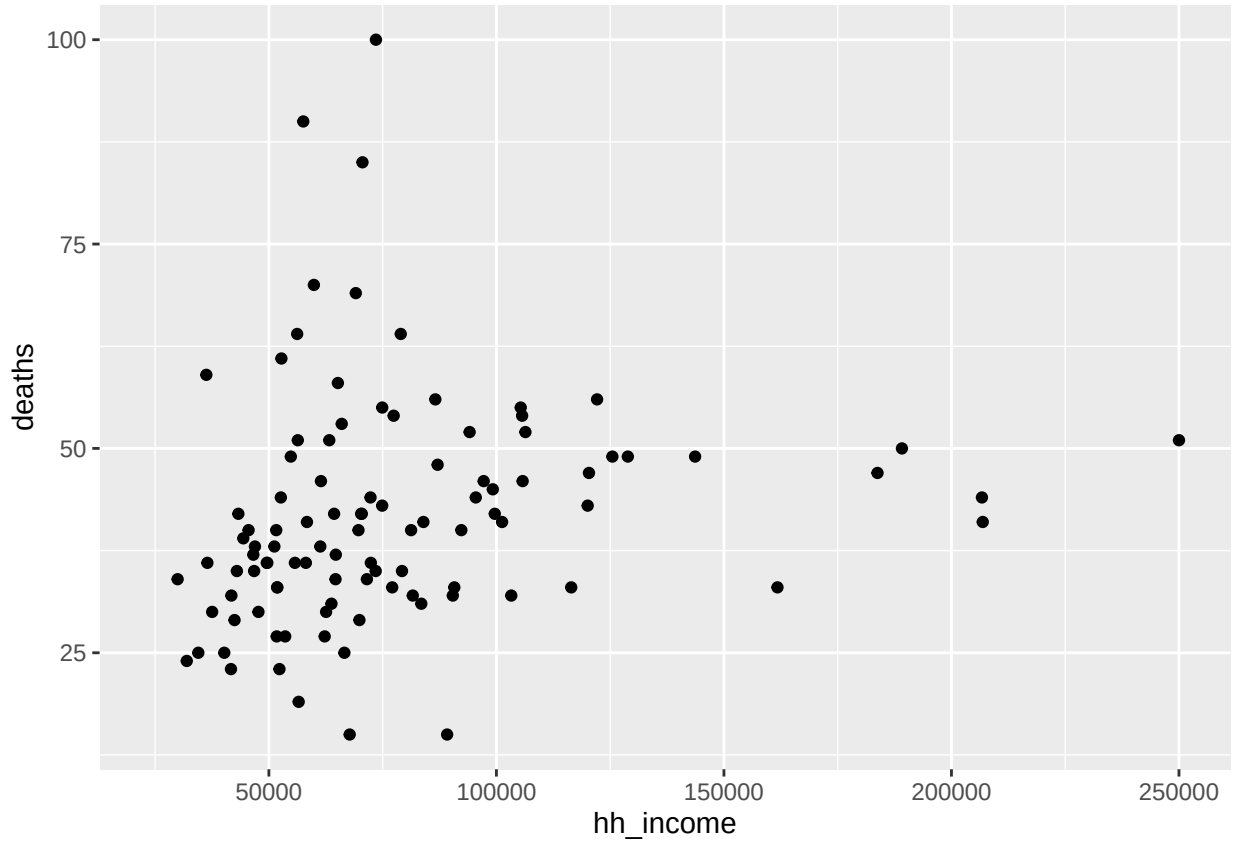


```r
# Correlation test
cor.test(merged_2$hh_income, merged_2$masks)
```

```
##
##  Pearson's product-moment correlation
##
## data:  merged_2$hh_income and merged_2$masks
## t = 4.5418, df = 112, p-value = 1.414e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2269226 0.5391849
## sample estimates:
##       cor
## 0.3943782
```

```r
# Plot for deaths
qplot(hh_income, deaths, data = merged_2)
```

```
## Warning: Removed 217 rows containing missing values (`geom_point()`).
```




```r
# Correlation test
cor.test(merged_2$hh_income, merged_2$deaths)
```

```
##
```

```
##  Pearson's product-moment correlation
##
## data:  merged_2$hh_income and merged_2$deaths
## t = 1.8733, df = 100, p-value = 0.06395
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.01073418  0.36551184
## sample estimates:
##       cor
## 0.1841251
```

The Pearson correlation test shows that `masks` have a relationship with median household income ($r = 0.394$, $p < .001$). The data for `mask` search hits in the plot has less outliers with most of the data points gathering around the lower side of income. On the other hand, `deaths` did not have a relationship with median household income ($r = 0.184$, $p = 0.366$). This coincides with the data points in the plot for `deaths` being more spread out.Notably, people with lower household income who may be more at risk of being infected or spreading COVID-19 due to their socioeconomic status, may have searched for `masks` frequently to buy or make them.