Assignment 2

Kangrui Liu

9/30/2023

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.

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.

1. Git and GitHub

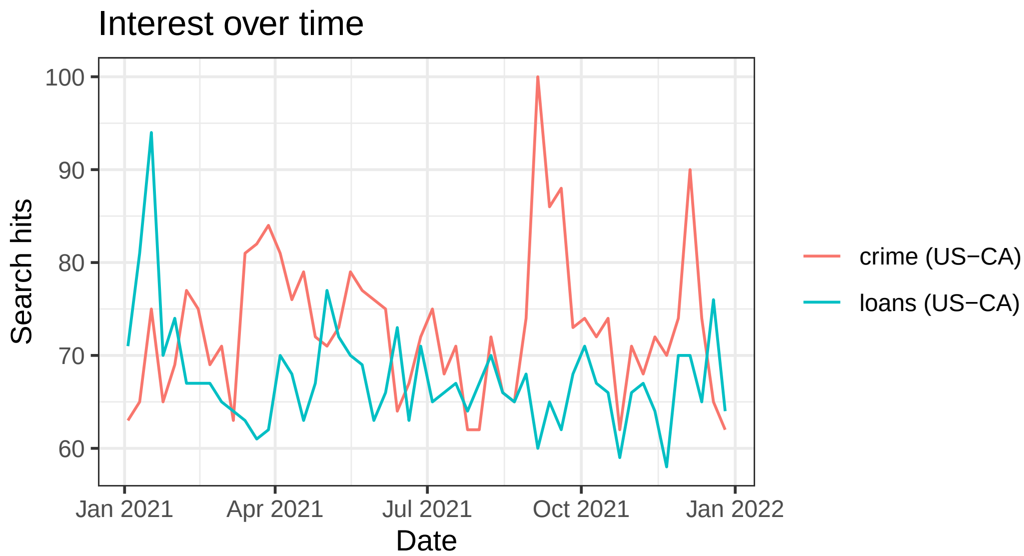
Provide the link to the GitHub repo for Assignment2.

- https://github.com/krliu67/Assignment_SURV727/tree/main/a2

2. 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 Caifornia in the year 2021. We could find this using the following code:

```
res_ca <- gtrends(c("crime", "loans"),
               geo = "US-CA",
               time = "2021-01-01 2021-12-31",
               low_search_volume = TRUE)
plot(res_ca)
```



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

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

```
res_ca_mmv <- res_ca$interest_over_time %>%
  group_by(keyword) %>%
  summarize(mean_hits=mean(hits), median_hits=median(hits), var_hits=var(hits))
res_ca_mmv
```

```
## # A tibble: 2 x 4
##   keyword mean_hits median_hits var_hits
##   <chr>       <dbl>       <dbl>    <dbl>
## 1 crime        72.9          72     60.8
## 2 loans        67.6          67     33.6
```

- According to the table presented above, the mean, median and variance of `covid` are 72.9, 72 and 60.83 separately. And, the mean, median and variance of `shooting` are 67.58,67and 33.58 separately.

b) 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.

```
# handle missing value
res_ca_city <- spread(na.omit(res_ca$interest_by_city), key = keyword, value = hits)
# prevent some data was loaded as other types
res_ca_city$crime <- as.numeric(res_ca_city$crime)
res_ca_city$loans <- as.numeric(res_ca_city$loans)
res_ca_city[is.na(res_ca_city)] <- 0

head(res_ca_city)
```

```
##          location   geo gprop crime loans
## 1          Acton US-CA   web     0     3
## 2          Alamo US-CA   web     0     2
## 3         Alpine US-CA   web     4     3
## 4    Alta Sierra US-CA   web     5     3
## 5       Altadena US-CA   web    15     0
## 6 American Canyon US-CA   web     5     0
```

```r
res_ca_city %>% subset(loans==max(res_ca_city$loans))
```

```
##            location   geo gprop crime loans
## 287 Yosemite Lakes US-CA   web     0   100
```

- Yosemite Lakes has the highest search frequency for `loans` in 2021 in California.

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

```r
cor_ca_city <- cor(res_ca_city$crime, res_ca_city$loans)
cor_ca_city
```

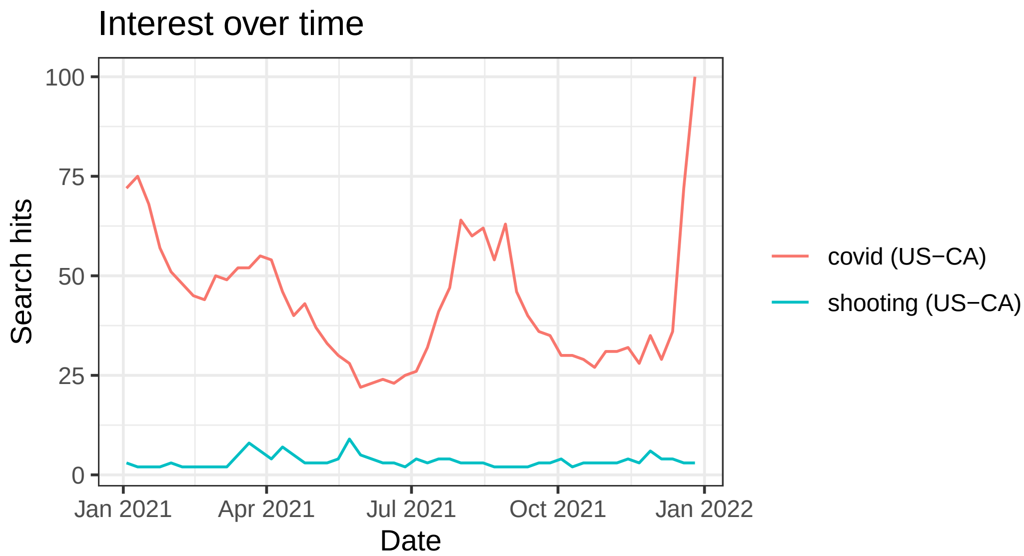
```
## [1] -0.1019966
```

- The correlation index of `loans` and `crime` in 2021 in California is -0.1, which means two keywords are weak negative linear correlated.

d) 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.

2. Answer the following questions for the keywords "covid" and "shooting".

```r
res1_ca <- gtrends(c("covid", "shooting"),
               geo = "US-CA",
               time = "2021-01-01 2021-12-31",
               low_search_volume = TRUE)
plot(res1_ca)
```



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

```
res1_ca_mmv <- res1_ca$interest_over_time %>%
  group_by(keyword) %>%
  summarize(mean_hits=mean(hits), median_hits=median(hits), var_hits=var(hits))
res1_ca_mmv
```

```
## # A tibble: 2 x 4
##   keyword  mean_hits median_hits var_hits
##   <chr>        <dbl>       <dbl>    <dbl>
## 1 covid        43.5         40.5   270.
## 2 shooting      3.44         3       2.37
```

- According to the table presented above, the mean, median and variance of `covid` are 43.5, 40.5 and 269.59 separately. And, the mean, median and variance of `shooting` are 3.44, 3and 2.37 separately.

b) Which cities (locations) have the highest search frequency for `covid` and `shooting`? 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.

```
# handle missing value
res1_ca$interest_by_city <- na.omit(res1_ca$interest_by_city)

# handle 'multiple rows for each city'
temp <- res1_ca$interest_by_city %>% filter(keyword=="covid")
temp <- as.data.frame(table(temp$location)) %>% filter(Freq > 1)
# find the cities which has multiple rows in a keyword
names <- temp[,1]
```

```r
rm(temp)

if (length(names) != 0){
  duplicate_rows <- res1_ca$interest_by_city %>% filter(keyword=="covid" & location==names)
  # keep the rows which keyword is not 'multiple rows for each city'
  temp <- subset(res1_ca$interest_by_city, keyword =="shooting")
  # keep the rows which keyword is  but city don't have multiple rows
  res1_ca$interest_by_city <- subset(res1_ca$interest_by_city, keyword=="covid" & location!=names)
  # delete duplicate rows and add hits up to one row for each city
  duplicate_rows[1,2] = sum(duplicate_rows$hits)
  duplicate_rows <- duplicate_rows[1,]
  res1_ca$interest_by_city <- rbind(res1_ca$interest_by_city, duplicate_rows)
  res1_ca$interest_by_city <- rbind(res1_ca$interest_by_city, temp)
  rm(temp)
  rm(duplicate_rows)
}
```

```r
# group by keyword
res1_ca_city <- spread(res1_ca$interest_by_city, key = keyword, value = hits)
res1_ca_city$covid <- as.numeric(res1_ca_city$covid)
res1_ca_city$shooting <- as.numeric(res1_ca_city$shooting)
res1_ca_city[is.na(res1_ca_city)] <- 0

head(res1_ca_city)
```

```
##   location   geo gprop covid shooting
## 1  Aguanga US-CA   web     0       55
## 2 Ahwahnee US-CA   web     0       39
## 3  Alameda US-CA   web    68        0
## 4    Alamo US-CA   web    83       45
## 5   Albany US-CA   web    73        0
## 6   Alpine US-CA   web     0       45
```

```r
res1_ca_city %>% subset(shooting==max(shooting))
```

```
##         location   geo gprop covid shooting
## 269 San Joaquin US-CA   web    27      100
```

```r
res1_ca_city %>% subset(covid==max(covid))
```

```
##       location   geo gprop covid shooting
## 169 Los Altos US-CA   web   100       37
```

- Los Altos has the highest search frequency for `covid` in 2021 in California. And, San Joaquin has the highest search frequency for `shooting` in 2021 in California.

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

```
cor1_ca_city <- cor(res1_ca_city$covid, res1_ca_city$shooting)
cor1_ca_city
```

```
## [1] -0.6374916
```

- The correlation index of `covid` and `shooting` in 2021 in California is -0.64‘, which means two keywords are negative linear correlated.

3. 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.

```
library(dplyr)
library(magrittr)
cs_key <- read.table("D:/suds/727/acs-key.txt")[1,1]
```

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 = 2021,
                    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 00113      Abingdon city, Illinois        3586        38.6       44042
## 2    17 00178          Adair CDP, Illinois         210        51.3  -666666666
## 3    17 00191          Adams CDP, Illinois          47        55.3  -666666666
## 4    17 00230 Addieville village, Illinois         359        32.6       88333
```

```
## 5    17 00243    Addison village, Illinois       35999         37.9        75960
## 6    17 00295    Adeline village, Illinois          95         40.5        53438
##   B19301_001E
## 1       22466
## 2       29101
## 3       34834
## 4       34871
## 5       32779
## 6       22506
```

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)
acs_il %<>%
  separate(NAME, c("location","state"), sep = ",") %T>%
  str(.)
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.

```
# Clean Data
acs_ca <- getCensus(name = "acs/acs5",
                    vintage = 2021,
                    vars = c("NAME",
                             "B01001_001E",
                             "B06002_001E",
                             "B19013_001E",
                             "B19301_001E"),
                    region = "place:*",
                    regionin = "state:06",
                    key = cs_key)
acs_ca[acs_ca == -666666666] <- NA
acs_ca <-
  acs_ca %>%
  rename(pop = B01001_001E,
```

```r
         age = B06002_001E,
         hh_income = B19013_001E,
         income = B19301_001E)
# split NAME into location & state
acs_ca %<>%
  separate(NAME, c("location","state"), sep = ",") %T>%
  str(.)
```

```
## 'data.frame':    1611 obs. of  7 variables:
##  $ place    : chr  "00135" "00156" "00212" "00296" ...
##  $ location : chr  "Acalanes Ridge CDP" "Acampo CDP" "Acton CDP" "Adelanto city" ...
##  $ state    : chr  " California" " California" " California" " California" ...
##  $ pop      : num  1074 263 6809 37229 171 ...
##  $ age      : num  46 28 49 28.1 67.2 44.8 51.1 53.7 58.1 27.7 ...
##  $ hh_income: num  161806 24446 109632 58040 37600 ...
##  $ income   : num  65050 19328 49046 15823 22980 ...
```

```r
head(acs_ca)
```

```
##   place           location       state   pop  age hh_income income
## 1 00135 Acalanes Ridge CDP  California  1074 46.0    161806  65050
## 2 00156         Acampo CDP  California   263 28.0     24446  19328
## 3 00212          Acton CDP  California  6809 49.0    109632  49046
## 4 00296      Adelanto city  California 37229 28.1     58040  15823
## 5 00310           Adin CDP  California   171 67.2     37600  22980
## 6 00394  Agoura Hills city  California 20362 44.8    141099  70983
```

- I change the state to California, and transformed `NAME` into `location` and `state` by cutting comma.

1) Answer the following questions with the "crime" and "loans" Google trends data and the ACS data.

a) 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.

```r
library(stringr)
# clean data, if location contains CDP or city, delete
for (x in 1:dim(acs_ca)[1]) {
  temp <- acs_ca$location[x]
  if (str_detect(acs_ca$location[x],"CDP") == TRUE){
      temp <- gsub("CDP",'',temp)
  }
  if (str_detect(acs_ca$location[x],"city") == TRUE){
      temp <- gsub("city",'',temp)
  }
  temp <- trimws(temp)
  acs_ca$location[x] <- temp
}
rm(temp)
```

```r
# find common cities in res1_ca_city and acs_ca
common_cities <- intersect(res_ca_city$location, acs_ca$location)
temp1 <- res_ca_city[res_ca_city$location %in% common_cities,]
temp2 <- acs_ca[acs_ca$location %in% common_cities,]
temp2_dup_names <- as.data.frame(table(temp2$location)) %>% filter(Freq > 1)
temp2_dup <- acs_ca[acs_ca$location %in% temp2_dup_names$Var1,]
temp2 <- temp2[!(temp2$location %in% temp2_dup$location),]
temp2_dup_names <- unique(temp2_dup$location)
# clean data and pre-process data
for (x in 1:length(temp2_dup_names)) {
  temp_rows <- temp2_dup[temp2_dup$location %in% temp2_dup_names[x],]
  temp_df <- data.frame(
    place=temp_rows$place[1],
    location=temp2_dup_names[x],
    state=temp_rows$state[1],
    pop=sum(temp_rows$pop),
    age=(temp_rows$pop[1]*temp_rows$age[1]/sum(temp_rows$pop))+(temp_rows$pop[2]*temp_rows$age[2]/sum(t
    hh_income=(temp_rows$pop[1]*temp_rows$hh_income[1]/sum(temp_rows$pop))+(temp_rows$pop[2]*temp_rows$
    income=(temp_rows$pop[1]*temp_rows$income[1]/sum(temp_rows$pop))+(temp_rows$pop[2]*temp_rows$income
  )
  temp2 <- rbind(temp2,temp_df)
}
rm(temp_df)
rm(temp_rows)
rm(temp2_dup)

merged_df <- cbind(temp1,temp2,by = "location")
merged_df <- merged_df[, !colnames(merged_df) %in% "location.1"]

rm(temp1)
rm(temp2)

head(merged_df)
```

```
##          location   geo gprop crime loans place      location.1      state
## 1           Acton US-CA   web     0     3 00212           Acton California
## 2           Alamo US-CA   web     0     2 00618           Alamo California
## 3          Alpine US-CA   web     4     3 01192          Alpine California
## 5        Altadena US-CA   web    15     0 01290        Altadena California
## 6 American Canyon US-CA   web     5     0 01640 American Canyon California
## 8         Antioch US-CA   web     5     3 02252         Antioch California
##      pop  age hh_income income       by
## 1   6809 49.0    109632  49046 location
## 2  13852 52.5    250001 120635 location
## 3  15648 41.8    103503  47948 location
## 5  43384 43.9    109743  54378 location
## 6  21735 36.8    117846  38544 location
## 8 114750 37.0     82244  33379 location
```

- Due there have "CDP" and "city" in `acs_ca$location`, the common cities we intend to find will be difficult, so I delete these two dirty words in`acs_ca$location`. Then we can find common cities in `acs_ca` and `res_ca_city` so that combining those cities to a new data. Considering that the `age`, `hh_income` and `income` are Relative numbers, so I do computations of summing two rows up by proportion each `pop` of row has.

b) 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?

```
merged_df[is.na(merged_df)] <- 0

above_hh <- merged_df %>%
  filter(hh_income > mean(hh_income))%>%
  summarize(mean_crime_hits=mean(crime),mean_loans_hits=mean(loans))
below_hh <- merged_df %>%
  filter(hh_income <= mean(hh_income))%>%
  summarize(mean_crime_hits=mean(crime), mean_loans_hits=mean(loans))

above_hh;below_hh
```

```
##   mean_crime_hits mean_loans_hits
## 1        3.872549        2.401961
```

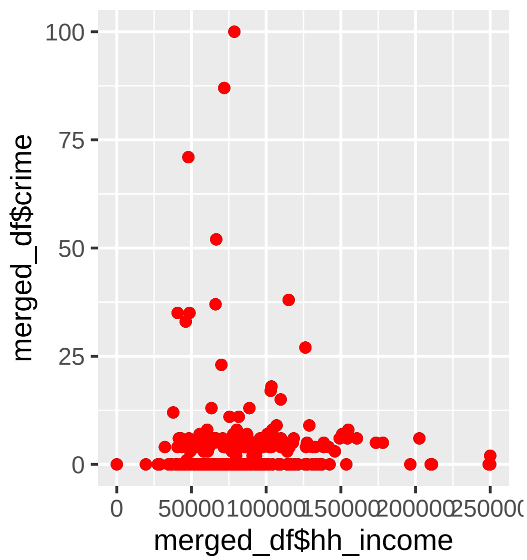
```
##   mean_crime_hits mean_loans_hits
## 1        5.030303        1.854545
```

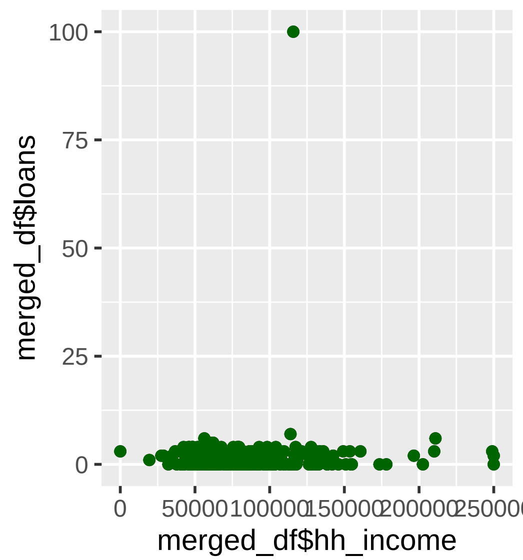
- There are 2 conclusions I draw from above tables. One is, In both subsets, the search frequency of `crime` is more than `loans`. Another is, Cities which have an below average median household income search both keywords more frequent than which have an above average median.

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

```
library(ggplot2)
p1 <- qplot(x=merged_df$hh_income,y=merged_df$crime)+
  geom_point(color="red")
p2 <- qplot(x=merged_df$hh_income,y=merged_df$loans)+
  geom_point(color="darkgreen")

library(gridExtra)
library(grid)
grid.arrange(p1, p2, ncol = 2)
```






```
cor_hh_cr <- cor(merged_df$hh_income,merged_df$crime)
cor_hh_lo <-cor(merged_df$hh_income,merged_df$loans)
cor_hh_cr;cor_hh_lo
```

```
## [1] -0.03500577
```

```
## [1] 0.03275659
```

- According to plots, I found the distribution of points are chaos, and I guess that there is no clear relationship between the median household income and the search popularity of the Google trends terms. Observed from data, the correlation index of househould income and crime hits is -0.04. and the correlation index of househould income and loans hits is 0.03. In my view, both correlation index were close to 0.00, which had weak relationships. Plus, plus or minus sign means the correlation of two variables is positive or negative.

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

a) 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.

```
# find common cities in res1_ca_city and acs_ca
common_cities1 <- intersect(res1_ca_city$location, acs_ca$location)
temp1 <- res1_ca_city[res1_ca_city$location %in% common_cities1,]
temp2 <- acs_ca[acs_ca$location %in% common_cities1,]
temp2_dup_names <- as.data.frame(table(temp2$location)) %>% filter(Freq > 1)
temp2_dup <- acs_ca[acs_ca$location %in% temp2_dup_names$Var1,]
temp2 <- temp2[!(temp2$location %in% temp2_dup$location),]
temp2_dup_names <- unique(temp2_dup$location)
# clean data and pre-process data
for (x in 1:length(temp2_dup_names)) {
```

```r
  temp_rows <- temp2_dup[temp2_dup$location %in% temp2_dup_names[x],]
  temp_df <- data.frame(
    place=temp_rows$place[1],
    location=temp2_dup_names[x],
    state=temp_rows$state[1],
    pop=sum(temp_rows$pop),
    age=(temp_rows$pop[1]*temp_rows$age[1]/sum(temp_rows$pop))+(temp_rows$pop[2]*temp_rows$age[2]/sum(te
    hh_income=(temp_rows$pop[1]*temp_rows$hh_income[1]/sum(temp_rows$pop))+(temp_rows$pop[2]*temp_rows$h
    income=(temp_rows$pop[1]*temp_rows$income[1]/sum(temp_rows$pop))+(temp_rows$pop[2]*temp_rows$income
  )
  temp2 <- rbind(temp2,temp_df)
}
rm(temp_df)
rm(temp_rows)
rm(temp2_dup)

merged_df1 <- cbind(temp1,temp2,by = "location")
merged_df1 <- merged_df1[, !colnames(merged_df1) %in% "location.1"]

rm(temp1)
rm(temp2)
```

b) 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
merged_df1[is.na(merged_df1)] <- 0

above_hh1 <- merged_df1 %>%
  filter(hh_income > mean(hh_income))%>%
  summarize(mean_covid_hits=mean(covid),mean_shooting_hits=mean(shooting))
below_hh1 <- merged_df1 %>%
  filter(hh_income <= mean(hh_income))%>%
  summarize(mean_covid_hits=mean(covid), mean_shooting_hits=mean(shooting))

above_hh1;below_hh1
```

```
##   mean_covid_hits mean_shooting_hits
## 1        37.01515           24.07576
```

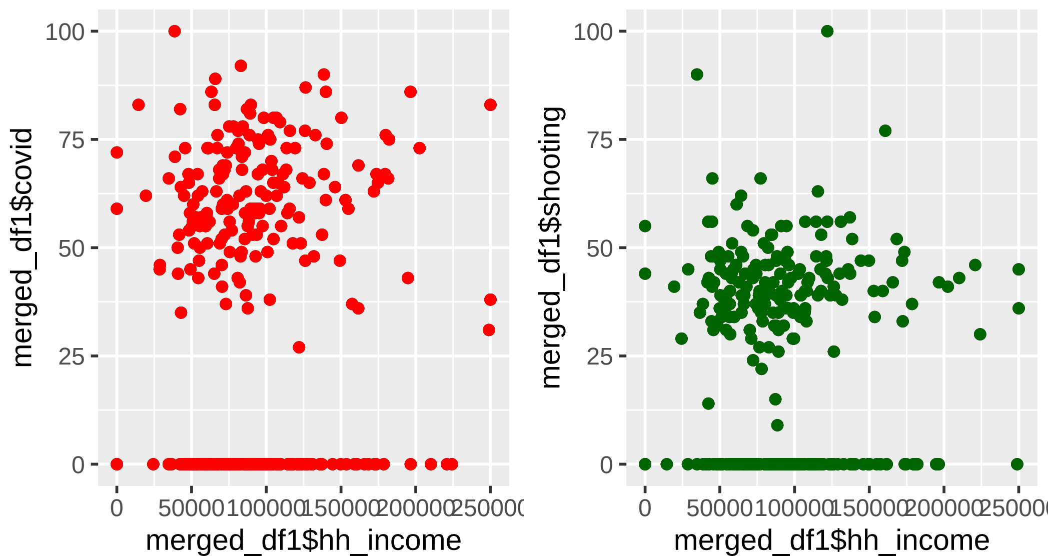
```
##   mean_covid_hits mean_shooting_hits
## 1        36.14368           25.02874
```

- Also, there are 2 conclusions I draw from above tables. A is, In both subsets, the search frequency of `covid` is more than `shooting`. B is, Cities which have below average median household income search `shooting` keyword more frequent than which have an above average median, whereas families which had more wealth paid more attentions to `covid` rather than `shooting`.

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

```
library(ggplot2)
p3 <- qplot(x=merged_df1$hh_income,y=merged_df1$covid)+
  geom_point(color="red")
p4 <- qplot(x=merged_df1$hh_income,y=merged_df1$shooting)+
  geom_point(color="darkgreen")

library(gridExtra)
library(grid)
grid.arrange(p3, p4, ncol = 2)
```




```
cor1_hh_co <- cor(merged_df1$hh_income,merged_df1$covid)
cor1_hh_sh <- cor(merged_df1$hh_income,merged_df1$shooting)
cor1_hh_co;cor1_hh_sh
```

```
## [1] 0.02253245
```

```
## [1] -0.01591029
```

- According to plots, I found the distribution of points are chaos, and I guess that there is no clear relationship between the median household income and the search popularity of the Google trends terms. According to number, the correlation index of househould income and covid hits is 0.02. and the correlation index of househould income and shooting hits is -0.02. In my view, both correlation index were close to 0.00, which had weak relationships. Plus, plus or minus sign means the correlation of two variables is positive or negative.