Fundamentals of Computing and Data Display

Assignment 2

Juan Gelvez-Ferreira

Setup

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gtrendsR)
```

```
## Warning: package 'gtrendsR' was built under R version 4.2.1
```

```
library(censusapi)
```

```
## Warning: package 'censusapi' was built under R version 4.2.1

##
## Attaching package: 'censusapi'

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

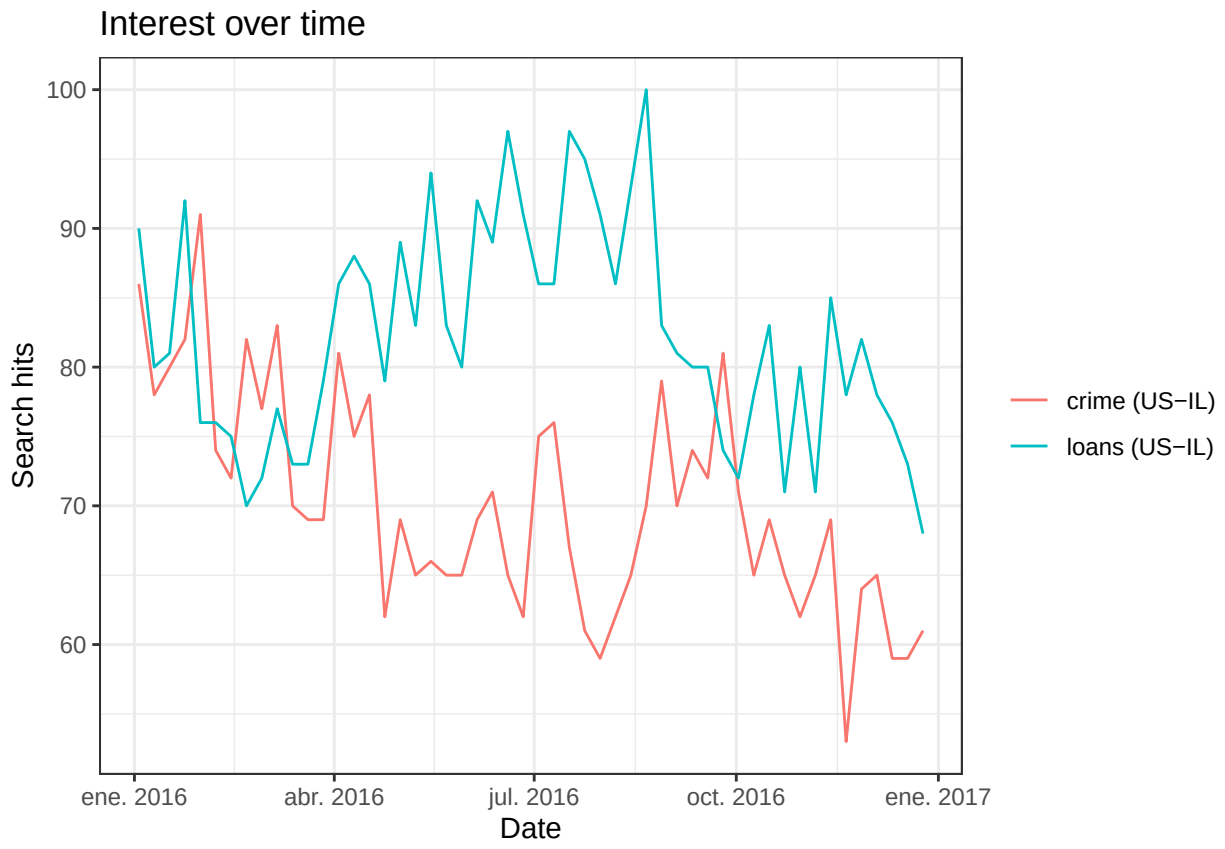
```
library(dplyr)
library(tidyr)
```

Google Trends

In this notebook, your task is to combine and explore web data using APIs and `dplyr`. Try to utilize piping in this notebook when writing your code.

Our first data source is the Google Trends API. This time we are interested in the search trends for `crime` and `loans` in Illinois in the year 2016.

```
res <- gtrends(c("crime", "loans"), geo = "US-IL", time = "2016-01-01 2016-12-31", low_search_volume = T
plot(res)
```



The resulting list includes a `data.frame` with the search interest by city. Extract this data set as a `tibble` and print the first few observations.

```
names(res)
```

```
## [1] "interest_over_time"  "interest_by_country" "interest_by_region"
## [4] "interest_by_dma"     "interest_by_city"    "related_topics"
## [7] "related_queries"
```

```
is_tibble(res)
```

```
## [1] FALSE
```

```
is_tibble(as_tibble(res$interest_by_city))
```

```
## [1] TRUE
```

```
res1 <- as_tibble(res$interest_by_city)
head(res1)
```

```
## # A tibble: 6 x 5
##   location      hits keyword geo   gprop
##   <chr>        <int> <chr>   <chr> <chr>
## 1 Riverwoods     100 crime   US-IL web
## 2 Braidwood       50 crime   US-IL web
## 3 Sauk Village    46 crime   US-IL web
## 4 Palos Park      34 crime   US-IL web
## 5 Macomb          32 crime   US-IL web
## 6 Park Forest     29 crime   US-IL web
```

Find the mean, median and variance of the search hits for the keywords `crime` and `loans`. This can be done via piping with `dplyr`.

```
res1 %>%
  group_by(keyword) %>%
  summarise(hits_mean = mean(hits, na.rm = TRUE),
            hits_median = median(hits, na.rm = TRUE),
            hits_sd = sd(hits, na.rm = TRUE))
```

```
## # A tibble: 2 x 4
##   keyword hits_mean hits_median hits_sd
##   <chr>       <dbl>       <int>   <dbl>
## 1 crime        23.1          22    13.3
## 2 loans        44.4          40    17.9
```

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 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 `tibble` accordingly and save the result as a new object.

```
res2 <- pivot_wider(res1,
                    names_from = keyword,
                    values_from = hits)
head(res2)
```

```
## # A tibble: 6 x 5
##   location     geo   gprop crime loans
##   <chr>        <chr> <chr> <int> <int>
## 1 Riverwoods   US-IL web     100    NA
## 2 Braidwood    US-IL web      50    45
## 3 Sauk Village US-IL web      46    NA
## 4 Palos Park   US-IL web      34    NA
## 5 Macomb       US-IL web      32    NA
## 6 Park Forest  US-IL web      29    49
```

Which cities (locations) have the highest search frequency for `loans`? Print the first rows of the new `tibble` from the previous chunk, ordered by `loans`.

```
#Atlanta, Willowbrook and Williamsfield are the locations that have the highest search frequency for `l

res2 %>%
  slice_max(loans, n = 6)
```

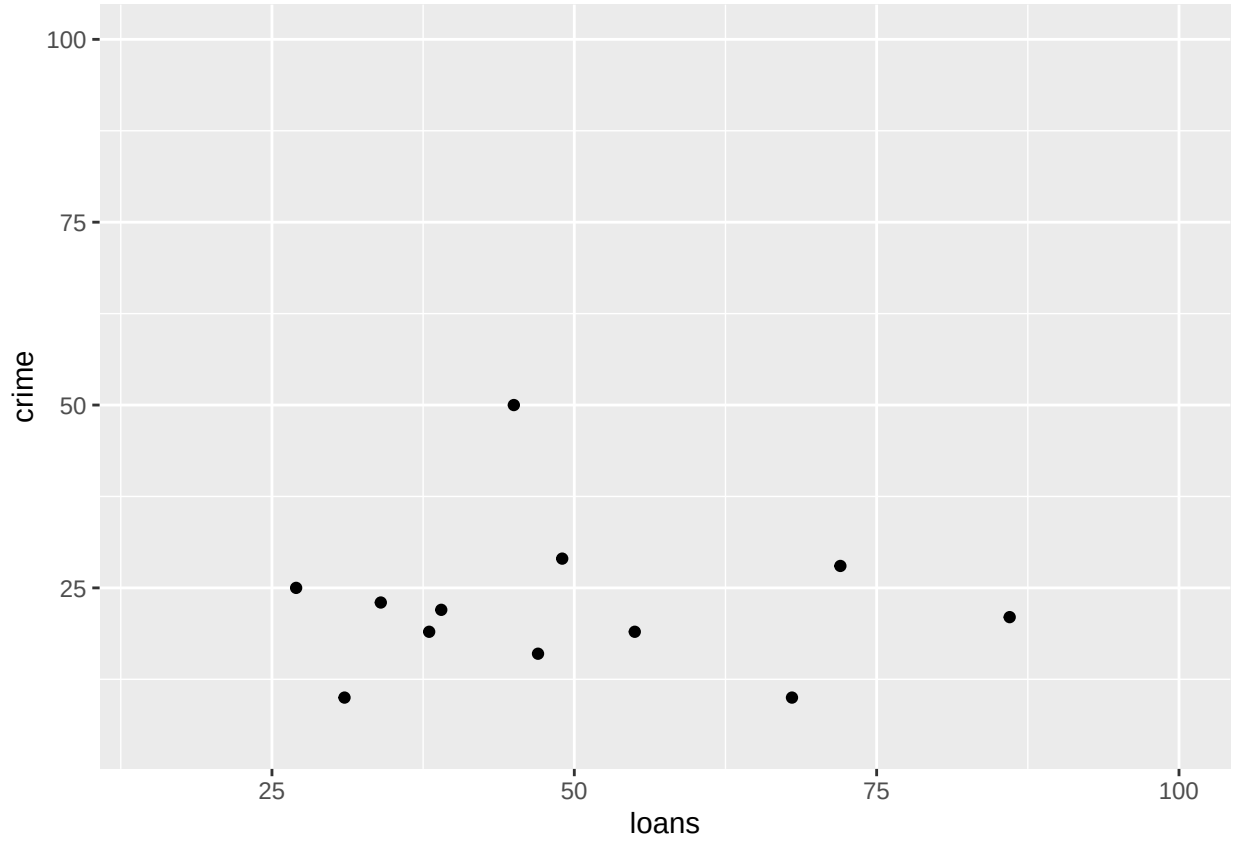
```
## # A tibble: 6 x 5
##   location  geo   gprop crime loans
##   <chr>     <chr> <chr> <int> <int>
## 1 Riverton  US-IL web      NA   100
## 2 Zion      US-IL web      21    86
## 3 Madison   US-IL web      NA    80
## 4 Robbins   US-IL web      NA    79
## 5 Hines     US-IL web      NA    79
## 6 Hoopeston US-IL web      NA    76
```

Is there a relationship between the search intensities between the two keywords we used? Create a scatterplot of `crime` and `loans` with `qplot()`.

```
#There isn´t a clear a relationship between the search intensities between the crime and loans.

qplot(loans, crime, data=res2)
```

```
## Warning: Removed 337 rows containing missing values (geom_point).
```



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 <- "412c45f7f8f25f31dfa9121b48f369df61b2711c"
```

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 = 2016,
                    vars = c("NAME", "B01001_001E", "B06002_001E", "B19013_001E", "B19301_001E"),
                    region = "place:*",
                    regionin = "state:17",
                    key = "412c45f7f8f25f31dfa9121b48f369df61b2711c")
head(acs_il)
```

```
##   state place                          NAME B01001_001E B06002_001E B19013_001E
## 1    17 11202    Carlinville city, Illinois        5297        36.7       40250
## 2    17 21410  Eagarville village, Illinois         165        39.2       48750
## 3    17 57043     Owaneco village, Illinois         201        44.6       42500
## 4    17 34137     Henning village, Illinois         243        31.9       55500
## 5    17 00880    Allerton village, Illinois         288        42.6       58125
## 6    17 57693 Parkersburg village, Illinois         146        41.1       48000
##   B19301_001E
## 1       22441
## 2       31400
## 3       22708
## 4       18009
## 5       24356
## 6       24795
```

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)
```

Print the first rows of the variable `NAME`.

```
acs_il %>%
  slice_max(NAME, n = 6)
```

```
##   state place                          NAME   pop  age hh_income income
## 1    17 84220          Zion city, Illinois 24195 32.7     46735  19814
## 2    17 84155       Zeigler city, Illinois  1771 34.7     36667  17075
## 3    17 84038     Yorkville city, Illinois 18436 32.3     85045  30534
## 4    17 83817 Yates City village, Illinois   722 38.7     51071  29135
## 5    17 83765       Yale village, Illinois   129 26.8     47500  12970
## 6    17 83739      Xenia village, Illinois   475 35.1     44750  17916
```

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.

```
acs_il2 <- acs_il %>% separate(NAME, c('location', 'NAME'))
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 1368 rows [1, 2, 3,
## 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
print(" Data frame after splitting: ")
```

```
## [1] " Data frame after splitting: "
```

```
acs_il2 %>%
  slice_max(location, n = 6)
```

```
##   state place  location    NAME   pop  age hh_income income
## 1    17 84220      Zion    city 24195 32.7     46735  19814
## 2    17 84155   Zeigler    city  1771 34.7     36667  17075
## 3    17 84038 Yorkville    city 18436 32.3     85045  30534
## 4    17 83817     Yates    City   722 38.7     51071  29135
## 5    17 83765      Yale village   129 26.8     47500  12970
## 6    17 83739     Xenia village   475 35.1     44750  17916
```

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

That's a lot, unfortunately. However, we can still try using the data. Create a new data set by joining the Google Trends and the ACS data. Keep only cities that appear in both data sets.

```
new_data<- inner_join(res2, acs_il2, by ='location')
head(new_data)
```

```
## # A tibble: 6 x 12
##   location   geo   gprop crime loans state place NAME      pop   age hh_income
##   <chr>      <chr> <chr> <int> <int> <chr> <chr> <chr>   <dbl> <dbl>     <dbl>
## 1 Riverwoods US-IL web     100    NA 17    64538 village  3759  48.3    187857
## 2 Braidwood  US-IL web      50    45 17    07770 city     6102  44.2     61074
## 3 Macomb     US-IL web      32    NA 17    45889 city    18771  24.2     35459
## 4 Dunlap     US-IL web      29    NA 17    21176 village  1351  34.4    101667
## 5 Riverdale  US-IL web      28    72 17    64278 village 13047  35       31438
## 6 Freeport   US-IL web      25    NA 17    27884 city    24784  42.7     35552
## # ... with 1 more variable: income <dbl>
```

Now we can utilize information from both data sources. As an example, print the `crime` and `loans` search popularity for the first ten cities in Illinois with the highest population (in 2016).

```
new_data %>%
  slice_max(pop, n = 10)
```

```
## # A tibble: 10 x 12
##    location geo   gprop crime loans state place NAME       pop   age hh_income
##    <chr>    <chr> <chr> <int> <int> <chr> <chr> <chr>    <dbl> <dbl>     <dbl>
##  1 Rockford US-IL web      23    NA 17    65000 city    149597  36       40143
##  2 Evanston US-IL web      22    NA 17    24582 city     75472  35.3     71317
##  3 Quincy   US-IL web      10    31 17    62367 city     40689  39.9     42078
##  4 Lansing  US-IL web      23    NA 17    42028 village  28369  40.9     50107
##  5 Alton    US-IL web      NA    40 17    01114 city     27175  37.5     37108
##  6 Harvey   US-IL web      23    NA 17    33383 city     25625  35.4     21909
##  7 Freeport US-IL web      25    NA 17    27884 city     24784  42.7     35552
##  8 Zion     US-IL web      21    86 17    84220 city     24195  32.7     46735
##  9 Maywood  US-IL web      NA    42 17    47774 village  24029  33.6     44126
## 10 Dolton   US-IL web      NA    32 17    20292 village  23113  36.2     44511
## # ... with 1 more variable: income <dbl>
```

Next, 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.

```
new_data$hh_income_level <- "Below average median household income"

new_data$hh_income_level[which(new_data$hh_income > 54379.19)] <-"Above average median household income

mean_crime_hhincome <- new_data %>%
    group_by(hh_income_level) %>%
    summarise(mean_crime = mean(crime, na.rm = TRUE))

mean_loans_hhincome <- new_data %>%
    group_by(hh_income_level) %>%
    summarise(mean_loans = mean(loans, na.rm = TRUE))

mean_crime_hhincome
```

```
## # A tibble: 2 x 2
##   hh_income_level                       mean_crime
##   <chr>                                      <dbl>
## 1 Above average median household income       25.3
## 2 Below average median household income       20.3
```

```
mean_loans_hhincome
```

```
## # A tibble: 2 x 2
##   hh_income_level                       mean_loans
##   <chr>                                      <dbl>
## 1 Above average median household income       38.9
## 2 Below average median household income       46.1
```

Is there a relationship between the median household income and the search popularity of `loans`? Plot a scatterplot with `qplot()`.

```
# There is a positive but aparently non-significant correlation between loans and household income.

qplot(loans, hh_income, data=new_data, , geom = c("point", "smooth"))
```

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

## Warning: Removed 201 rows containing non-finite values (stat_smooth).

## Warning: Removed 201 rows containing missing values (geom_point).
```