Assignment 3

Due at 11:59pm on October 14. Nicole Blackburn and Jianing Zou

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. Include the GitHub link for the repository containing these files.

```r
library(xml2)
library(rvest)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.2
v ggplot2   4.0.0     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter()         masks stats::filter()
x readr::guess_encoding() masks rvest::guess_encoding()
x dplyr::lag()            masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becon
```

```r
library(robotstxt)
```

Click here for Github Link

Web Scraping

In this assignment, your task is to scrape some information from Wikipedia. We start with the following page about Grand Boulevard, a Chicago Community Area.

https://en.wikipedia.org/wiki/Grand_Boulevard,_Chicago

The ultimate goal is to gather the table "Historical population" and convert it to a `data.frame`.

As a first step, read in the html page as an R object. Extract the tables from this object (using the `rvest` package) and save the result as a new object. Follow the instructions if there is an error. Use `str()` on this new object – it should be a list. Try to find the position of the "Historical population" in this list since we need it in the next step.

```
paths_allowed("https://en.wikipedia.org/wiki/Grand_Boulevard,_Chicago")
```

```

 en.wikipedia.org

[1] TRUE
```

```
gb_html <- read_html("https://en.wikipedia.org/wiki/Grand_Boulevard,_Chicago")
```

```
tables <- html_table(gb_html, trim = TRUE)
str(tables)
```

```
List of 7
 $ : tibble [27 x 2] (S3: tbl_df/tbl/data.frame)
  ..$ Grand Boulevard: chr [1:27] "Community area" "Community Area 38 -
Grand Boulevard" "The Harold Washington Cultural Center" "Location within the city of Chicago
  ..$ Grand Boulevard: chr [1:27] "Community area" "Community Area 38 -
Grand Boulevard" "The Harold Washington Cultural Center" "Location within the city of Chicago
 $ : tibble [11 x 4] (S3: tbl_df/tbl/data.frame)
  ..$ Census
  ..$ Pop.
  ..$ .mw-parser-output .sr-only{border:0;clip:rect(0,0,0,0);clip-path:polygon(0px 0px,0px 0p
  ..$ %±
" "18.7%" "10.9%" "-30.1%" ...
 $ : tibble [6 x 17] (S3: tbl_df/tbl/data.frame)
  ..$ Places adjacent to Grand Boulevard, Chicago: chr [1:6] "Armour Square, Chicago\nDouglas
  ..$ Places adjacent to Grand Boulevard, Chicago: chr [1:6] "Armour Square, Chicago\nDouglas
  ..$                                            : chr [1:6] "Armour Square, Chicago" "Oaklan
  ..$                                            : chr [1:6] "Douglas, Chicago" NA NA NA ...
  ..$                                            : chr [1:6] "Oakland, Chicago" NA NA NA ...
  ..$                                            : logi [1:6] NA NA NA NA NA NA
  ..$                                           : logi [1:6] NA NA NA NA NA NA
  ..$                                           : logi [1:6] NA NA NA NA NA NA
  ..$                                           : chr [1:6] "Fuller Park, Chicago" NA NA NA
  ..$                                           : chr [1:6] "Grand Boulevard, Chicago" NA N
  ..$                                           : chr [1:6] "Kenwood, Chicago" NA NA NA ...
  ..$                                           : logi [1:6] NA NA NA NA NA NA
  ..$                                           : logi [1:6] NA NA NA NA NA NA
  ..$                                           : logi [1:6] NA NA NA NA NA NA
  ..$                                           : chr [1:6] "New City, Chicago" NA NA NA ..
  ..$                                           : chr [1:6] "Washington Park, Chicago" NA N
  ..$                                           : chr [1:6] "Hyde Park, Chicago" NA NA NA .
 $ : tibble [5 x 3] (S3: tbl_df/tbl/data.frame)
  ..$ X1: chr [1:5] "Armour Square, Chicago" "" "Fuller Park, Chicago" "" ...
  ..$ X2: chr [1:5] "Douglas, Chicago" "" "Grand Boulevard, Chicago" "" ...
  ..$ X3: chr [1:5] "Oakland, Chicago" "" "Kenwood, Chicago" "" ...
 $ : tibble [9 x 2] (S3: tbl_df/tbl/data.frame)
  ..$ .mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-o
  ..$ .mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-o
 $ : tibble [2 x 2] (S3: tbl_df/tbl/data.frame)
  ..$ vteNeighborhoods in Chicago: chr [1:2] "Recognized by the city" "Other districts and a
  ..$ vteNeighborhoods in Chicago: chr [1:2] "Albany Park\nAndersonville\nArcher Heights\nAs
Green\nCana"| __truncated__
 $ : tibble [2 x 2] (S3: tbl_df/tbl/data.frame)
  ..$ vte Chicago: chr [1:2] "Architecture\nBeaches\nClimate\ntornadoes\nColleges and univer
  ..$ vte Chicago: chr [1:2] "Architecture\nBeaches\nClimate\ntornadoes\nColleges and univer
```

Extract the "Historical population" table from the list and save it as another object. You can use subsetting via `[[…]]` to extract pieces from a list. Print the result.

```r
hist_pop <- tables[[2]]
print(hist_pop)
```

```
# A tibble: 11 x 4
   Census Pop.    .mw-parser-output .sr-only{border:0;clip:rect(0,0,0,0)~1 `%±`
   <chr>  <chr>   <chr>                                                   <chr>
 1 1930   87,005  ""                                                      -
 2 1940   103,256 ""                                                      18.7%
 3 1950   114,557 ""                                                      10.9%
 4 1960   80,036  ""                                                      -30.~
 5 1970   80,166  ""                                                      0.2%
 6 1980   53,741  ""                                                      -33.~
 7 1990   35,897  ""                                                      -33.~
 8 2000   28,006  ""                                                      -22.~
 9 2010   21,929  ""                                                      -21.~
10 2020   24,589  ""                                                      12.1%
11 [3][1] [3][1]  "[3][1]"                                                [3][~
# i abbreviated name:
#   1: `.mw-parser-output .sr-only{border:0;clip:rect(0,0,0,0);clip-path:polygon(0px 0px,0px
```

You will see that the table needs some additional formatting. Keep only want rows and columns with actual values.

```
hist_pop <- hist_pop |>
  select(Census, `Pop.`, `%±`) |>
  rename('Pop_GrandBoulevard' = `Pop.`,
         'Change_GrandBoulevard' = `%±`) |>
  slice(1:nrow(hist_pop) -1)
hist_pop <- hist_pop
```

Expanding to More Pages

That's it for this page. However, we may want to repeat this process for other community areas. The Wikipedia page https://en.wikipedia.org/wiki/Grand_Boulevard,_Chicago has a section on "Places adjacent to Grand Boulevard, Chicago" at the bottom. Can you find the corresponding table in the list of tables that you created earlier? Extract this table as a new object.

```
adjacent <- tables[[4]]
str(adjacent)
```

```
tibble [5 x 3] (S3: tbl_df/tbl/data.frame)
 $ X1: chr [1:5] "Armour Square, Chicago" "" "Fuller Park, Chicago" "" ...
 $ X2: chr [1:5] "Douglas, Chicago" "" "Grand Boulevard, Chicago" "" ...
 $ X3: chr [1:5] "Oakland, Chicago" "" "Kenwood, Chicago" "" ...
```

Then, grab the community areas east of Grand Boulevard and save them as a character vector. Print the result.

```
east_gb <- adjacent$X3[adjacent$X3 != ""]
print(east_gb)
```

```
[1] "Oakland, Chicago"   "Kenwood, Chicago"   "Hyde Park, Chicago"
```

We want to use this list to create a loop that extracts the population tables from the Wikipedia pages of these places. To make this work and build valid urls, we need to replace empty spaces in the character vector with underscores. The resulting vector should look like this: "Oakland,_Chicago" "Kenwood,_Chicago" "Hyde_Park,_Chicago"

```r
east_gb <- east_gb |>
  str_replace_all(" ", "_")
print(east_gb)
```

```
[1] "Oakland,_Chicago"   "Kenwood,_Chicago"   "Hyde_Park,_Chicago"
```

Build a loop to grab the population tables from each page. Add columns to the original table using `cbind()`.

```r
tables_all <- list()

for (i in east_gb) {
  url <- paste0("https://en.wikipedia.org/wiki/", i)
  webpage <- read_html(url)
  tables <- webpage |>
    html_element(xpath = "//table[caption = 'Historical population']")
  tbl2 <- html_table(tables)
  tables_all[[i]] <- tbl2 |>
    select(Census, `Pop.`, `%±`)
}
```

```r
for (i in names(tables_all)) {
  city <- str_remove(i, ",_.*$")
  tables_all[[i]] <- tables_all[[i]] |>
    slice(1:nrow(tables_all[[i]]) - 1) |>
    rename(!!paste0("Pop_", city) := `Pop.`,
           !!paste0("Change_", city) := `%±`)
}

tables_combined <- reduce(tables_all, full_join, join_by("Census"))
```

```r
hist_pop_all <- full_join(hist_pop, tables_combined, join_by(Census))
hist_pop_all <- hist_pop_all |>
  arrange(Census)

print(hist_pop_all)
```

```
# A tibble: 12 x 9
   Census Pop_GrandBoulevard Change_GrandBoulevard Pop_Oakland Change_Oakland
   <chr>  <chr>              <chr>                 <chr>       <chr>
 1 1910   <NA>               <NA>                  13,763      -
 2 1920   <NA>               <NA>                  16,540      20.2%
 3 1930   87,005             -                     14,962      -9.5%
 4 1940   103,256            18.7%                 14,500      -3.1%
 5 1950   114,557            10.9%                 24,464      68.7%
 6 1960   80,036             -30.1%                24,378      -0.4%
 7 1970   80,166             0.2%                  18,291      -25.0%
 8 1980   53,741             -33.0%                16,748      -8.4%
 9 1990   35,897             -33.2%                8,197       -51.1%
10 2000   28,006             -22.0%                6,110       -25.5%
11 2010   21,929             -21.7%                5,918       -3.1%
12 2020   24,589             12.1%                 6,799       14.9%
# i 4 more variables: Pop_Kenwood <chr>, Change_Kenwood <chr>,
#   Pop_Hyde_Park <chr>, Change_Hyde_Park <chr>
```

Scraping and Analyzing Text Data

Suppose we wanted to take the actual text from the Wikipedia pages instead of just the information in the table. Our goal in this section is to extract the text from the body of the pages, then do some basic text cleaning and analysis.

First, scrape just the text without any of the information in the margins or headers. For example, for "Grand Boulevard", the text should start with, "**Grand Boulevard** on the South Side of Chicago, Illinois, is one of the …". Make sure all of the text is in one block by using something like the code below (I called my object `description`).

```
text_only <- html_nodes(gb_html, xpath = '//*[(@id = "mw-content-text")]//p')

description <- text_only %>% paste(collapse = ' ')
```

Using a similar loop as in the last section, grab the descriptions of the various communities areas. Make a tibble with two columns: the name of the location and the text describing the location.

```
communities <- c(east_gb)

all_text <- tibble(page = character(), paragraph = character())

for (i in communities) {
```

```r
  url <- paste0("https://en.wikipedia.org/wiki/", i)
  webpage <- read_html(url)

  paragraphs <- webpage |>
    html_nodes(xpath = '//*[(@id = "mw-content-text")]//p') |>
    html_text2() |>
    str_squish()

  all_text <- all_text |>
    add_row(page = i, paragraph = paragraphs |>  paste(collapse = ' '))
}

all_text <- all_text |>
  mutate(page = page |>
           str_remove("^/wiki/") |>
           str_remove("\\(neighborhood\\)") |>
           str_replace_all("_", " ") |>
           str_squish())

print(all_text)
```

```
# A tibble: 3 x 2
  page                 paragraph
  <chr>                <chr>
1 Oakland, Chicago     " Oakland, located on the South Side of Chicago, Illinois,~
2 Kenwood, Chicago     " Kenwood, one of Chicago's 77 community areas, is on the ~
3 Hyde Park, Chicago " Hyde Park is a neighborhood on the South Side of Chicago~
```

Let's clean the data using `tidytext`. If you have trouble with this section, see the example shown in https://www.tidytextmining.com/tidytext.html

```r
library(tidytext)
```

Create tokens using `unnest_tokens`. Make sure the data is in one-token-per-row format. Remove any stop words within the data. What are the most common words used overall?

Answer

The most common words used overall are: park: 84, hyde: 75, chicago: 51, kenwood: 40, street: 34, south: 27, oakland: 25.

```r
unnest_tidy <- all_text |>
  unnest_tokens(word, paragraph)

data(stop_words)

unnest_tidy <- unnest_tidy |>
  anti_join(stop_words)
```

```
Joining with `by = join_by(word)`
```

```r
head(unnest_tidy)
```

```
# A tibble: 6 x 2
  page             word
  <chr>            <chr>
1 Oakland, Chicago oakland
2 Oakland, Chicago located
3 Oakland, Chicago south
4 Oakland, Chicago chicago
5 Oakland, Chicago illinois
6 Oakland, Chicago usa
```

```r
unnest_tidy |>
  count(word, sort = TRUE)
```

```
# A tibble: 1,080 x 2
   word             n
   <chr>        <int>
 1 park            84
 2 hyde            75
 3 chicago         51
 4 kenwood         40
 5 street          34
 6 south           27
 7 oakland         25
 8 community       23
 9 lake            23
10 neighborhood    23
# i 1,070 more rows
```

Plot the most common words within each location. What are some of the similarities between the locations? What are some of the differences?

Answer

After plotting the most common words for each page, some similarities across all locations are the name of the location, 'Chicago', 'street' and words that have to do with groups of people like 'community' and 'neighborhood'.

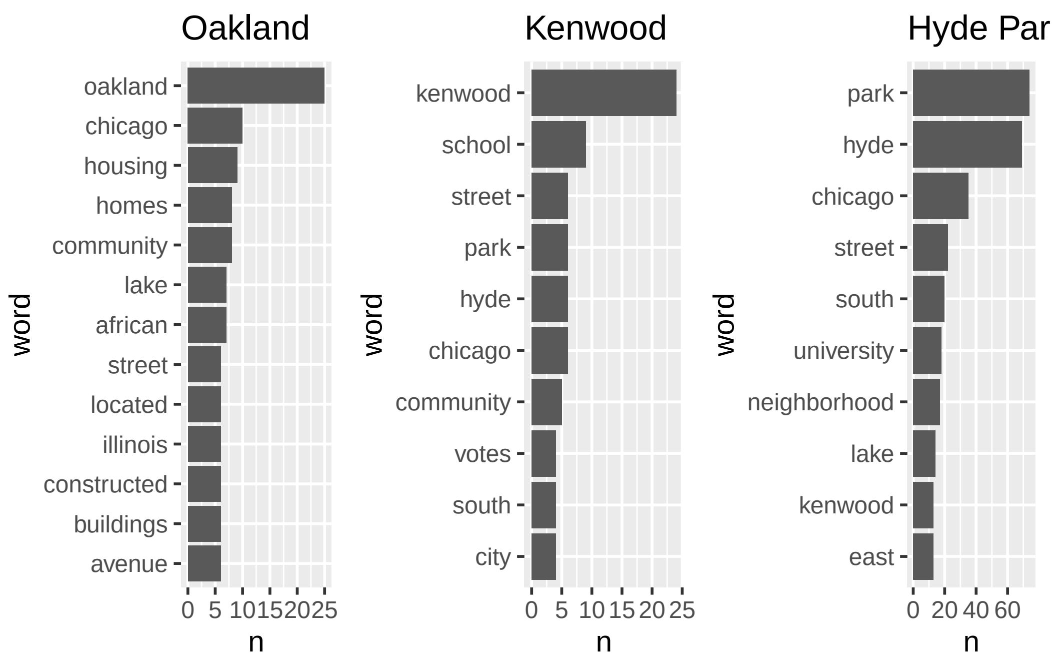
As for differences, Oakland has 'housing', 'homes', 'constructed', 'buildings', and 'avenue' maybe from construction in the community and housing discussions. It also lists 'african' as one of the most common words, maybe reflecting the demographics of the community. Kenwood has 'school' as the second most common word, reflecting discussion about a school in Kenwood. It also lists 'votes', maybe notable elections or voting behavior happens in Kenwood. Hyde Park has 'university', maybe there is a university in Hyde Park.

Using the quanteda library, we see that text1(Oakland, Chicago) – text3(Hyde Park, Chicago) are the most similar because Jaccard score is 0.156,which is the highest among all pairs. That might because they share similar words more than sharing with other groups.

```r
library(ggplot2)
library(gridExtra)
```

```

Attaching package: 'gridExtra'


The following object is masked from 'package:dplyr':

    combine
```

```r
p1 <- unnest_tidy |>
  count(page, word, sort = TRUE) |>
  filter(page == "Oakland, Chicago") |>
  slice_max(n, n = 10) |>
  mutate(word = reorder(word, n)) |>
  ggplot(aes(n, word)) +
  geom_col() +
  labs(title = "Oakland")

p2 <- unnest_tidy |>
  count(page, word, sort = TRUE) |>
  filter(page == "Kenwood, Chicago") |>
  slice_max(n, n = 10) |>
  mutate(word = reorder(word, n)) |>
  ggplot(aes(n, word)) +
  geom_col() +
  labs(title = "Kenwood")

p3 <- unnest_tidy |>
  count(page, word, sort = TRUE) |>
  filter(page == "Hyde Park, Chicago") |>
  slice_max(n, n = 10) |>
  mutate(word = reorder(word, n)) |>
  ggplot(aes(n, word)) +
  geom_col() +
  labs(title = "Hyde Park")

grid.arrange(p1, p2, p3, ncol = 3)
```



```r
library(quanteda)
```

```
Package version: 4.3.1
Unicode version: 15.1
ICU version: 74.1
```

```
Parallel computing: 12 of 12 threads used.
```

```
See https://quanteda.io for tutorials and examples.
```

```r
library(quanteda.textstats)

text_df <- unnest_tidy |>
  group_by(page) |>
  summarise(text = paste(word, collapse = " ")) |>
  ungroup()

T_DFM <- text_df |>
  corpus(text_field = "text") |>
  tokens() |>
  dfm()

cat("\nInitial sparsity: \n")
```

```
Initial sparsity:
```

```r
dim(T_DFM)
```

```
[1]    3 1080
```

```r
t1 <- textstat_simil(T_DFM,
               margin = "documents",
               method = "jaccard")
t1
```

```
textstat_simil object; method = "jaccard"
      text1 text2 text3
text1 1.000 0.126 0.156
text2 0.126 1.000 0.143
text3 0.156 0.143 1.000
```