Assignment 3

Due at 11:59pm on October 14.

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.

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.

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.

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

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

```
 en.wikipedia.org
```

```
[1] TRUE
```

```r
Chi_html <- read_html("https://en.wikipedia.org/wiki/Grand_Boulevard,_Chicago")
tables<- html_table(Chi_html, fill = TRUE)
```

```r
hist_pop<-tables[[2]]

head(hist_pop)
```

```
# A tibble: 6 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.~
# i abbreviated name:
#   1: `.mw-parser-output .sr-only{border:0;clip:rect(0,0,0,0);clip-path:polygon(0px 0px,0px
```

```r
hist_pop_clean <- hist_pop %>%
  slice(-11) %>%
  select(-3)
```

```r
hist_pop_clean <- hist_pop_clean %>%
  mutate(across(where(is.character), ~ str_remove_all(., ","))) %>%
  mutate(across(where(is.character), ~ str_trim(.))) %>%
  mutate(across(where(~ all(str_detect(., "^\\d+$"))), as.numeric))

print(hist_pop_clean)
```

```
# A tibble: 10 x 3
   Census   Pop. `%±`
    <dbl>  <dbl> <chr>
 1   1930  87005 -
 2   1940 103256 18.7%
 3   1950 114557 10.9%
 4   1960  80036 -30.1%
 5   1970  80166 0.2%
 6   1980  53741 -33.0%
 7   1990  35897 -33.2%
 8    2000  28006 -22.0%
 9    2010  21929 -21.7%
10    2020  24589 12.1%
```

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.

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

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"

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

```r
##Finding the Corresponding tables that leads to the other pages and tables
adjacent_table <- tables[[4]]

adjacent_table
```

```
# A tibble: 5 x 3
  X1                         X2                           X3
  <chr>                      <chr>                        <chr>
1 "Armour Square, Chicago" "Douglas, Chicago"           "Oakland, Chicago"
2 ""                         ""                           ""
3 "Fuller Park, Chicago"   "Grand Boulevard, Chicago" "Kenwood, Chicago"
4 ""                         ""                           ""
5 "New City, Chicago"      "Washington Park, Chicago" "Hyde Park, Chicago"
```

```r
adjacent_table <- adjacent_table[-c(2, 4), ]

adjacent_table
```

```
# A tibble: 3 x 3
  X1                    X2                        X3
  <chr>                 <chr>                     <chr>
1 Armour Square, Chicago Douglas, Chicago         Oakland, Chicago
2 Fuller Park, Chicago   Grand Boulevard, Chicago Kenwood, Chicago
3 New City, Chicago      Washington Park, Chicago Hyde Park, Chicago
```

```
east <- adjacent_table[, 3, drop = FALSE]

# the 3rd column renamed east
east <- as.character(unlist(adjacent_table[, 3, drop = FALSE]))


print(east)
```

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

```
east_links <- gsub(" ", "_", east)

print(east_links)
```

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

```
urls <- paste0("https://en.wikipedia.org/wiki/", east_links)
print(urls)
```

```
[1] "https://en.wikipedia.org/wiki/Oakland,_Chicago"
[2] "https://en.wikipedia.org/wiki/Kenwood,_Chicago"
[3] "https://en.wikipedia.org/wiki/Hyde_Park,_Chicago"
```

```
Chi_html <- read_html("https://en.wikipedia.org/wiki/Oakland,_Chicago")
tables <- html_table(Chi_html, fill = TRUE)
length(tables)
```

```
[1] 7
```

```
View(tables[[1]])
View(tables[[2]])
View(tables[[3]])

Table2<-tables[[2]]
```

```r
# Start with your Grand Boulevard population table
combined_pop <- hist_pop_clean

for (url in urls) {
  message("Scraping: ", url)

  Chi_html <- read_html(url)
  tables <- html_table(Chi_html, fill = TRUE)

  # Select correct table
  if (grepl("Hyde_Park", url)) {
    Table <- tables[[4]]
    Table <- Table[-c(1, 2), ]
  } else {
    Table <- tables[[2]]
  }

  # Clean table BEFORE cbind()
  Table <- Table %>%
    select(1:2) %>%
    rename(Census = 1, Pop = 2) %>%
    filter(!is.na(Census)) %>%
    mutate(
      Pop = str_remove_all(Pop, ","),
      Pop = as.numeric(Pop)
    )

  # Align Census years to base table
  Table <- Table[Table$Census %in% hist_pop_clean$Census, ]
  Table <- Table[match(hist_pop_clean$Census, Table$Census), ]

  colnames(Table)[2] <- gsub("https://en.wikipedia.org/wiki/|,_Chicago", "", url)

  combined_pop <- cbind(combined_pop, Table[, 2, drop = FALSE])
}
```

```
Scraping: https://en.wikipedia.org/wiki/Oakland,_Chicago

Scraping: https://en.wikipedia.org/wiki/Kenwood,_Chicago

Scraping: https://en.wikipedia.org/wiki/Hyde_Park,_Chicago
```

```
print(combined_pop)
```

```
   Census   Pop.     %± Oakland Kenwood Hyde_Park
1    1930  87005      -   14962   26942        NA
2    1940 103256  18.7%   14500   29611        NA
3    1950 114557  10.9%   24464   35705     55206
4    1960  80036 -30.1%   24378   41533     45577
5    1970  80166   0.2%   18291   26890     33531
6    1980  53741 -33.0%   16748   21974     31198
7    1990  35897 -33.2%    8197   18178     28630
8    2000  28006 -22.0%    6110   18363     29920
9    2010  21929 -21.7%    5918   17841     25681
10   2020  24589  12.1%    6799   19116     29456
```

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`).

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

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

description <- Chi_html %>%
  html_nodes("p") %>%
  html_text() %>%
  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.

```r
urls_all <- c(
  "https://en.wikipedia.org/wiki/Grand_Boulevard,_Chicago",
  "https://en.wikipedia.org/wiki/Oakland,_Chicago",
  "https://en.wikipedia.org/wiki/Kenwood,_Chicago",
  "https://en.wikipedia.org/wiki/Hyde_Park,_Chicago"
)


descriptions <- tibble(
  location = character(),
  text = character()
)


for (url in urls_all) {
  message("Scraping text from: ", url)

  Chi_html <- read_html(url)

#by paragraph
  description <- Chi_html %>%
    html_nodes("p") %>%
    html_text() %>%
    paste(collapse = " ")

#clean location name
  location_name <- gsub("https://en.wikipedia.org/wiki/|,_Chicago", "", url)

  #tibble
  descriptions <- add_row(descriptions,
                          location = location_name,
                          text = description)
}
```

```
Scraping text from: https://en.wikipedia.org/wiki/Grand_Boulevard,_Chicago

Scraping text from: https://en.wikipedia.org/wiki/Oakland,_Chicago

Scraping text from: https://en.wikipedia.org/wiki/Kenwood,_Chicago

Scraping text from: https://en.wikipedia.org/wiki/Hyde_Park,_Chicago
```

```
View(descriptions)
print(descriptions)
```

```
# A tibble: 4 x 2
  location        text
  <chr>           <chr>
1 Grand_Boulevard "\n\n Grand Boulevard on the South Side of Chicago, Illinois,~
2 Oakland         "\n\n Oakland, located on the South Side of Chicago, Illinois~
3 Kenwood         "\n\n Kenwood, one of Chicago's 77 community areas, is on the~
4 Hyde_Park       "\n\n Hyde Park is a neighborhood on the South Side of Chicag~
```

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

```
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?

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

```
tokens <- descriptions %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words, by = "word")

head(tokens)
```

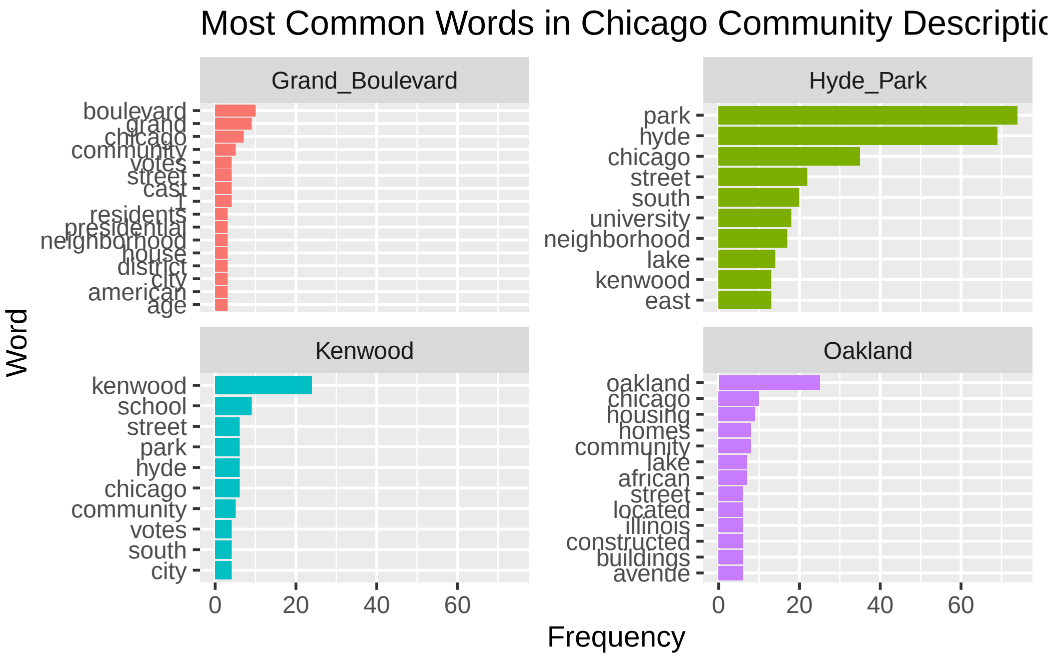
```
# A tibble: 6 x 2
  location        word
  <chr>           <chr>
1 Grand_Boulevard grand
2 Grand_Boulevard boulevard
3 Grand_Boulevard south
4 Grand_Boulevard chicago
5 Grand_Boulevard illinois
6 Grand_Boulevard city's
```

```
tokens %>%
  count(word, sort = TRUE) %>%
  head(10)
```

```
# A tibble: 10 x 2
   word             n
   <chr>        <int>
 1 park            85
 2 hyde            75
 3 chicago         58
 4 kenwood         40
 5 street          38
 6 south           29
 7 community       28
 8 neighborhood    26
 9 oakland         25
10 lake            23
```

The most common words used are in descending order starting with the highest amount of words on the pages are Park, Hyde, Chicago, Kenwood, Street, South, Community, Neighborhood, Oakland, and Lake.

```
tokens %>%
  count(location, word, sort = TRUE) %>%
  group_by(location) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder_within(word, n, location)) %>%
  ggplot(aes(n, word, fill = location)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ location, scales = "free_y") +
  scale_y_reordered() +
  labs(
    title = "Most Common Words in Chicago Community Descriptions",
    x = "Frequency",
    y = "Word"
  )
```



The plot above shows the four Chicago suburb Wikipedia pages Grand boulevard, Hyde Park, Kenwood, and Oakland and what words show up the most in each. What these frequency plots show is the distribution of words that show up in the text. What I found after examining the words is that the name of the suburb show up the most for each of the Wikipedia pages. Another word that show up a lot are words like neighborhood and community.Grand Boulevard and Hyde Park have neighborhood show up a few times while Grand Boulevard, Kenwood and, Oakland have the word community show up between the three of them. All four locations have the word street show up around that average amount between the most words used in each Wikipedia page. Another stand out is that Hyde Park and Kenwood both have the word Kenwood in their top words plot Hyde Park having the word Kenwood showing up a few times is interesting because it was the only page that have another locations name enough times to make the most words used list. Some differences that stick out to me is that Oakland is the only page that has the word African show up between the locations which can be assumed that this locations have a historic population or connection to African Americans or African communities. Another interesting one is that Oakland has the word lake as well which can suggest that there might be lakes in the area or certain location with the name of lake that show up in the Wikipedia page. Finally what stood out to me is that Grand boulevard is the only location that the words votes, cast, and presidential in the page suggesting some sort of voting or political information that is important or historic to this part of Chicago.