ASSIGNMENT 3 Web Scrappinng

Lugu R Nicholas & Doris Odei

2025-10-13

Repo:https://github.com/Dodei123/Fall-25-SURV727.git

Setup

Part 1 — "Historical population" table from Grand Boulevard

```r
base_page <- "https://en.wikipedia.org/wiki/Grand_Boulevard,_Chicago"

gb_html <- read_html(base_page)
table_nodes <- html_elements(gb_html, "table")
tables_list <- html_table(table_nodes, fill = TRUE, header = TRUE, convert = FALSE)

str(tables_list, max.level = 1)
```

```
## List of 7
##  $ : tibble [27 x 2] (S3: tbl_df/tbl/data.frame)
##  $ : tibble [11 x 4] (S3: tbl_df/tbl/data.frame)
##  $ : tibble [6 x 17] (S3: tbl_df/tbl/data.frame)
##  $ : tibble [4 x 3] (S3: tbl_df/tbl/data.frame)
##  $ : tibble [9 x 2] (S3: tbl_df/tbl/data.frame)
##  $ : tibble [2 x 2] (S3: tbl_df/tbl/data.frame)
##  $ : tibble [2 x 2] (S3: tbl_df/tbl/data.frame)
```

```r
length(tables_list)
```

```
## [1] 7
```

```r
table_captions <- map_chr(table_nodes, ~{
  cap <- html_element(.x, "caption")
  if (is.na(cap)) "" else html_text2(cap)
})

hist_idx <- which(str_detect(str_to_lower(table_captions), "historical population"))

if (length(hist_idx) == 0) {
  hist_idx <- tables_list %>%
    imap_lgl(~{
      nm <- names(.x) %>% str_to_lower()
      any(str_detect(nm, "census|^year$|^date$")) &&
        any(str_detect(nm, "pop|population"))
    }) %>% which()
}

gb_hist_raw <- tables_list[[hist_idx[1]]]
head(gb_hist_raw)
```

```
## # 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 0px,0px
```

```r
gb_hist <- gb_hist_raw %>%
  remove_empty(c("rows", "cols")) %>%
  clean_names()

year_col <- names(gb_hist)[str_detect(names(gb_hist), "^year$|census|^date$")][1]
pop_col  <- names(gb_hist)[str_detect(names(gb_hist), "pop")][1]

gb_hist <- gb_hist %>%
  select(Year = all_of(year_col), Grand_Boulevard = all_of(pop_col)) %>%
  mutate(
    Year = parse_number(Year),
    Grand_Boulevard = parse_number(Grand_Boulevard)
  ) %>%
  filter(!is.na(Year), !is.na(Grand_Boulevard), Year > 1000, Year < 2100) %>%
  distinct(Year, .keep_all = TRUE) %>%
  arrange(Year)

gb_hist
```

```
## # A tibble: 10 x 2
##      Year Grand_Boulevard
##     <dbl>           <dbl>
##  1   1930           87005
##  2   1940          103256
##  3   1950          114557
##  4   1960           80036
##  5   1970           80166
##  6   1980           53741
##  7   1990           35897
##  8   2000           28006
##  9  2010           21929
## 10  2020           24589
```

```
#table_captions

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

# Find the navbox that says "Places adjacent to Grand Boulevard, Chicago"
#adj_box <- html_elements(
 # gb_html,
  #xpath = "//table[contains(@class,'navbox')][.//text()[contains(., 'Places adjacent to Grand Boulevar

#length(adj_box)
#cat(substr(as.character(adj_box[[1]]), 1, 2000))  # visual inspection
```

Part 2 — "Places adjacent to Grand Boulevard, Chicago"

```
extract_cell_titles <- function(td) {
  if (!length(td)) return(list(titles = character(0), pretty = character(0)))
  links  <- rvest::html_elements(td, css = "a[href^='/wiki/']")
  if (!length(links)) return(list(titles = character(0), pretty = character(0)))

  hrefs  <- rvest::html_attr(links, "href")
  hrefs  <- hrefs[!is.na(hrefs)]
  slugs  <- sub("^/wiki/", "", hrefs)
  slugs  <- sub("[#?].*$", "", slugs)


  is_chi <- grepl(",_Chicago$", slugs)
  ltxt   <- trimws(rvest::html_text2(links))
  ltxt   <- gsub(",\\s*Chicago.*$", "", ltxt)
  ltxt   <- gsub("\\s*\\(.*?\\)$", "", ltxt)
  ltxt[ltxt == ""] <- NA_character_

  coerced <- if (any(!is_chi) && length(ltxt)) paste0(gsub("\\s+", "_", ltxt[!is_chi]), ",_Chicago") els

  titles <- unique(c(slugs[is_chi], coerced))
  titles <- titles[grepl("^[A-Za-z][A-Za-z_\\-]*,_Chicago$", titles)]
  pretty <- gsub("_", " ", sub(",_Chicago$", ", Chicago", titles))
  list(titles = titles, pretty = pretty)
}

# 1) Finding the adjacent-places navbox
adj_box <- rvest::html_elements(
  gb_html,
  xpath = "//table[contains(@class,'navbox')][.//text()[contains(., 'Places adjacent to Grand Boulevard
)
if (!length(adj_box)) {
  adj_box <- rvest::html_elements(
    gb_html,
    xpath = "//table[contains(@class,'navbox')][.//text()[contains(., 'Places adjacent to')]][.//text()
```

```r
  )
}
stopifnot(length(adj_box) >= 1)

# 2) Inner grid table (3×3)
inner_tbl <- rvest::html_element(adj_box[[1]], xpath = ".//table[@role='presentation']")
if (!length(inner_tbl)) inner_tbl <- rvest::html_element(adj_box[[1]], xpath = ".//table")
stopifnot(length(inner_tbl) >= 1)

# 3) Locating the center cell (Grand Boulevard, Chicago) and its row/col index
center_td <- rvest::html_elements(
  inner_tbl,
  xpath = ".//td[
    .//b[contains(normalize-space(.), 'Grand Boulevard, Chicago')]
    or .//a[contains(@href,'/wiki/Grand_Boulevard,_Chicago')]
    or contains(normalize-space(.), 'Grand Boulevard, Chicago')
  ]"
)
stopifnot(length(center_td) >= 1)
center_td <- center_td[[1]]

center_tr <- rvest::html_element(center_td, xpath = "./ancestor::tr[1]")
row_tds   <- rvest::html_elements(center_tr, xpath = ".//td")

col_idx <- {
  hits <- which(vapply(row_tds, function(x) identical(x, center_td), logical(1)))
  if (length(hits)) hits[[1]] else 2L
}

# 4) All rows and find nearest NON-EMPTY row above/below the center row
all_rows <- rvest::html_elements(inner_tbl, xpath = ".//tr")

row_idx <- {
  hits <- which(vapply(all_rows, function(x) identical(x, center_tr), logical(1)))
  if (length(hits)) hits[[1]] else 3L
}

is_nonempty_row <- function(tr) {
  if (!length(tr)) return(FALSE)
  tds <- rvest::html_elements(tr, xpath = ".//td")
  if (!length(tds)) return(FALSE)
  any(trimws(rvest::html_text2(tds)) != "")
}

# nearest non-empty above
row_above_idx <- NA_integer_
for (i in seq(row_idx - 1, 1, by = -1)) {
  if (is_nonempty_row(all_rows[[i]])) { row_above_idx <- i; break }
}

# nearest non-empty below
row_below_idx <- NA_integer_
for (i in seq(row_idx + 1, length(all_rows), by = 1)) {
  if (is_nonempty_row(all_rows[[i]])) { row_below_idx <- i; break }
}

# 5) Cells by direction aligned to the center column
get_td_at <- function(tr, col) {
  if (!length(tr)) return(NULL)
  tds <- rvest::html_elements(tr, xpath = ".//td")
  if (!length(tds)) return(NULL)
  if (col < 1) col <- 1
  if (col > length(tds)) col <- length(tds)
  tds[[col]]
}

east_td <- get_td_at(center_tr, col_idx + 1L)                              # same row, one to the righ
ne_td   <- if (!is.na(row_above_idx)) get_td_at(all_rows[[row_above_idx]], col_idx + 1L) else NULL
se_td   <- if (!is.na(row_below_idx)) get_td_at(all_rows[[row_below_idx]], col_idx + 1L) else NULL

# east-side neighbors (E + NE + SE)
east_e  <- extract_cell_titles(east_td)
east_ne <- extract_cell_titles(ne_td)
east_se <- extract_cell_titles(se_td)

east_titles <- unique(c(east_e$titles, east_ne$titles, east_se$titles))
east_pretty <- unique(c(east_e$pretty, east_ne$pretty, east_se$pretty))

east_titles <- east_titles[order(east_titles)]
east_pretty <- east_pretty[order(east_pretty)]

east_titles
```

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

```
east_pretty
```

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

Part 3 — Loop to collect population tables and combine via `cbind()`

```
get_hist_population <- function(page_title, col_name = NULL){
  url <- paste0("https://en.wikipedia.org/wiki/", page_title)
  tryCatch({
    pg <- read_html(url)
    tnodes <- html_elements(pg, "table")
    tlist  <- html_table(tnodes, fill = TRUE, header = TRUE, convert = FALSE)

    caps <- tnodes %>% map_chr(~{
      cap <- html_element(.x, "caption")
      if (is.na(cap)) "" else html_text2(cap)
    })
    idx <- which(str_detect(str_to_lower(caps), "historical population"))
    if (length(idx) == 0) {
      idx <- tlist %>% imap_lgl(~{
        nm <- names(.x) %>% str_to_lower()
        any(str_detect(nm, "census|^year$|^date$")) &&
          any(str_detect(nm, "pop|population"))
      }) %>% which()
    }

    tab <- tlist[[idx[1]]] %>%
      remove_empty(c("rows","cols")) %>%
      clean_names()

    yr  <- names(tab)[str_detect(names(tab), "^year$|census|^date$")][1]
    pop <- names(tab)[str_detect(names(tab), "pop")][1]

    nm <- if (is.null(col_name)) {
      page_title %>%
        sub("_,_Chicago$|,_Chicago$", "", .) %>%
        gsub("_"," ", ., fixed = TRUE)
    } else col_name

    out <- tab %>%
      select(Year = all_of(yr), !!nm := all_of(pop)) %>%
      mutate(Year = parse_number(Year), across(-Year, parse_number)) %>%
      filter(!is.na(Year), Year > 1000, Year < 2100) %>%
      distinct(Year, .keep_all = TRUE) %>%
      arrange(Year)
    out
  }, error = function(e){
    warning("Failed on: ", page_title, " - ", conditionMessage(e))
    NULL
  })
}
```

```
cb_result <- gb_hist %>% arrange(Year)
neighbor_colnames <- east_titles %>%
  str_replace(",_Chicago$", "") %>%
  str_replace_all("_", " ")

neighbors <- map2(east_titles, neighbor_colnames, ~ get_hist_population(.x, col_name = .y))

for (tab in neighbors) {
  if (!is.null(tab)) {
    aligned <- left_join(cb_result %>% select(Year), tab, by = "Year") %>% arrange(Year)
    cb_result <- cbind(cb_result, aligned %>% select(-Year))
  }
}
cb_result
```

```
##    Year Grand_Boulevard Hyde Park Kenwood Oakland
## 1  1930           87005     48017   26942   14962
## 2  1940          103256     50550   29611   14500
## 3  1950          114557     55206   35705   24464
## 4  1960           80036     45577   41533   24378
## 5  1970           80166     33531   26890   18291
## 6  1980           53741     31198   21974   16748
## 7  1990           35897     28630   18178    8197
## 8  2000           28006     29920   18363    6110
## 9  2010           21929     25681   17841    5918
## 10 2020           24589     29456   19116    6799
```

Part 4 — Scraping and Analyzing Text Data

```r
get_description <- function(page_title){
  url <- paste0("https://en.wikipedia.org/wiki/", page_title)
  tryCatch({
    pg <- read_html(url)
    ps <- html_elements(pg, css = "#mw-content-text .mw-parser-output > p, #mw-content-text .mw-parser-
    txt <- html_text2(ps)
    txt <- txt[nchar(txt) > 0]
    paste(txt, collapse = " ")
  }, error = function(e){
    warning("Failed to get description for ", page_title, " - ", conditionMessage(e))
    NA_character_
  })
}

text_pages <- c("Grand_Boulevard,_Chicago", east_titles) %>% unique()

descriptions <- tibble(
  page_title = text_pages,
  location = page_title %>% str_replace(",_Chicago$", "") %>% str_replace_all("_"," "),
  text = map_chr(page_title, get_description)
)

descriptions %>% select(location, text)
```

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

Tokenization and Stopword Removal

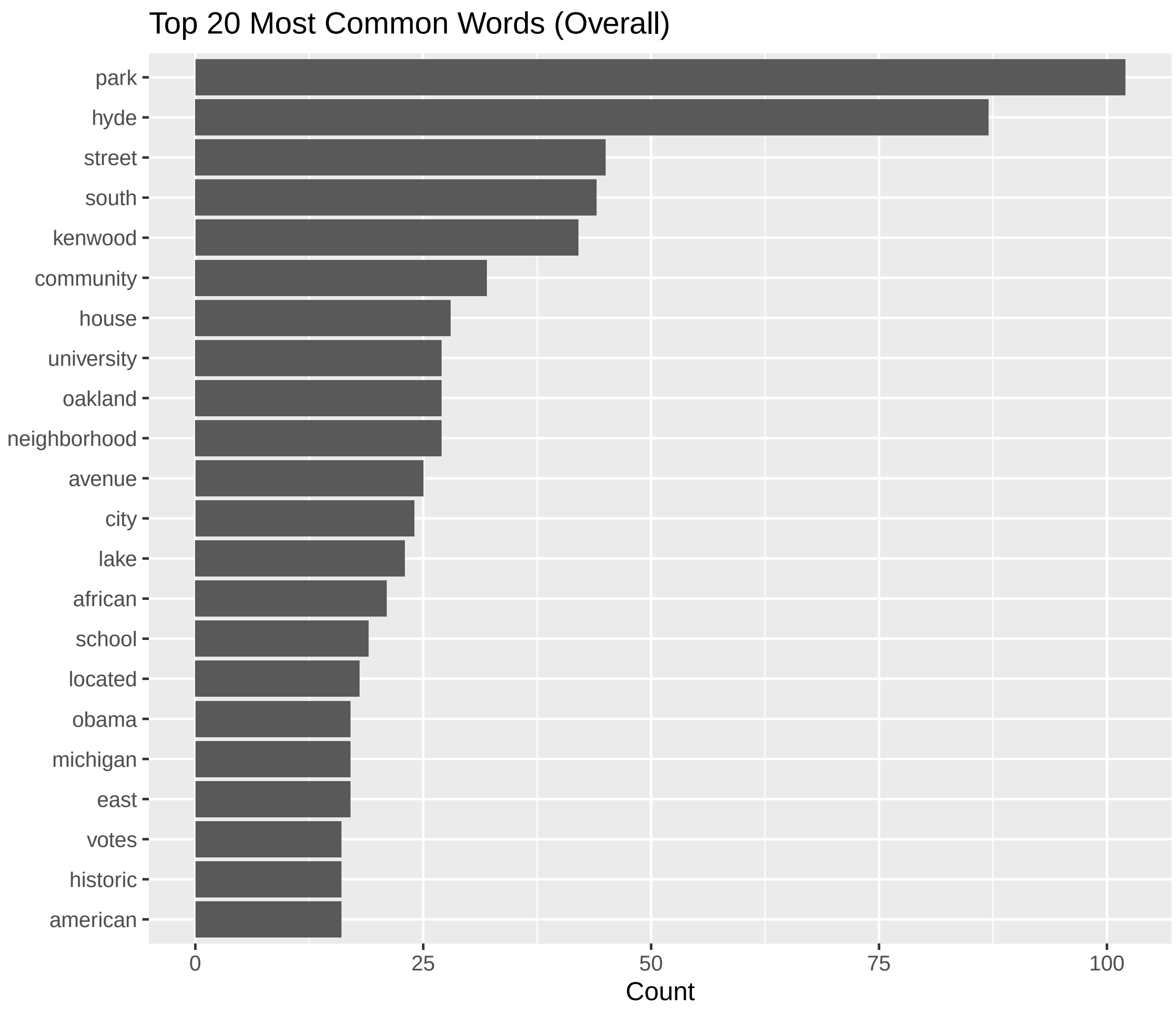
```r
data("stop_words")

tokens <- descriptions %>%
  select(location, text) %>%
  unnest_tokens(token, text) %>%
  anti_join(stop_words, by = c("token" = "word")) %>%
  filter(!str_detect(token, "^[0-9]+$")) %>%
  filter(!token %in% c("chicago","illinois","grand","boulevard"))

top_overall <- tokens %>%
  count(token, sort = TRUE) %>%
  slice_max(n, n = 20)
head(top_overall)
```

```
## # A tibble: 6 x 2
##   token         n
##   <chr>     <int>
## 1 park        102
## 2 hyde         87
## 3 street       45
## 4 south        44
## 5 kenwood      42
## 6 community    32
```

```r
top_overall %>%
  mutate(token = fct_reorder(token, n)) %>%
  ggplot(aes(x = token, y = n)) +
  geom_col() +
  coord_flip() +
  labs(title = "Top 20 Most Common Words (Overall)", x = NULL, y = "Count")
```

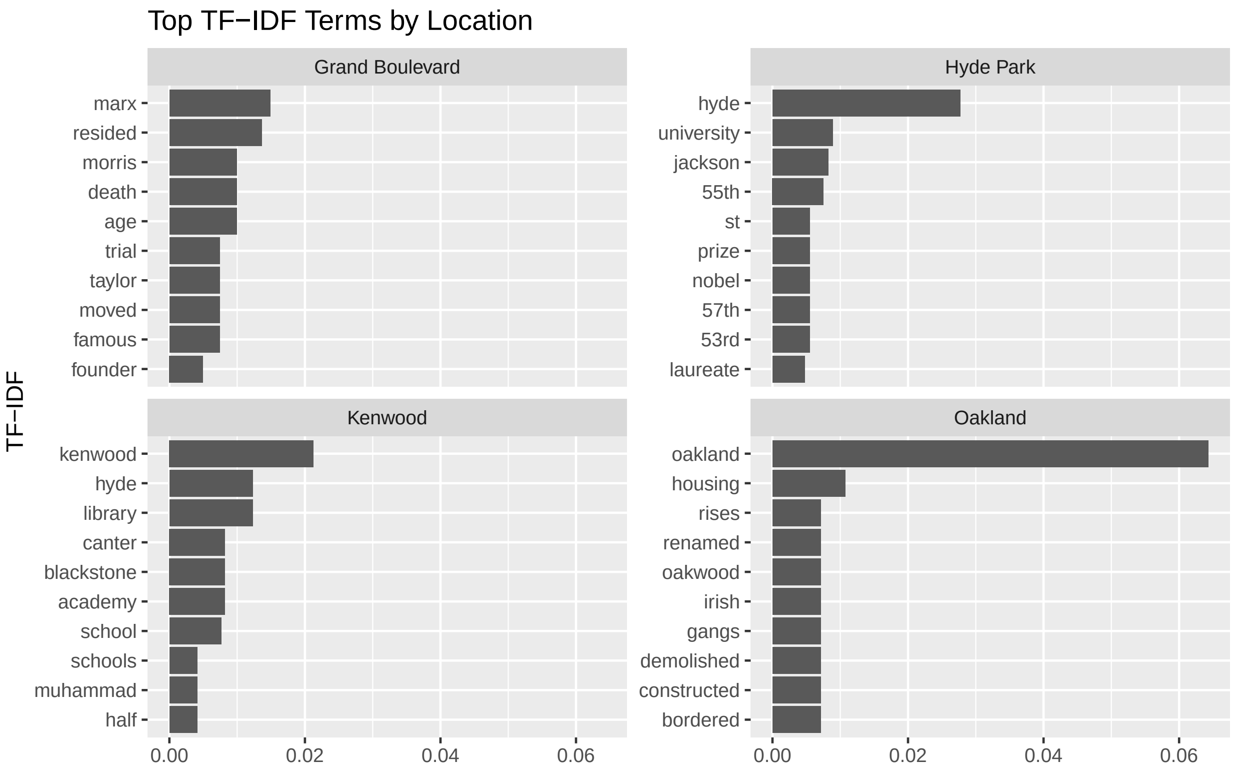


Distinctive Terms per Location (TF–IDF)

```r
counts <- tokens %>%
  count(location, token, sort = TRUE)

tfidf_top <- counts %>%
  bind_tf_idf(token, location, n) %>%
  filter(n >= 2) %>%
  group_by(location) %>%
  slice_max(tf_idf, n = 10, with_ties = FALSE) %>%
  ungroup() %>%
  mutate(token = fct_reorder(token, tf_idf))

ggplot(tfidf_top, aes(x = token, y = tf_idf)) +
  geom_col() +
  coord_flip() +
  facet_wrap(~ location, scales = "free_y") +
  labs(title = "Top TF-IDF Terms by Location", x = "TF-IDF", y = NULL)
```



Most Common Words by Location

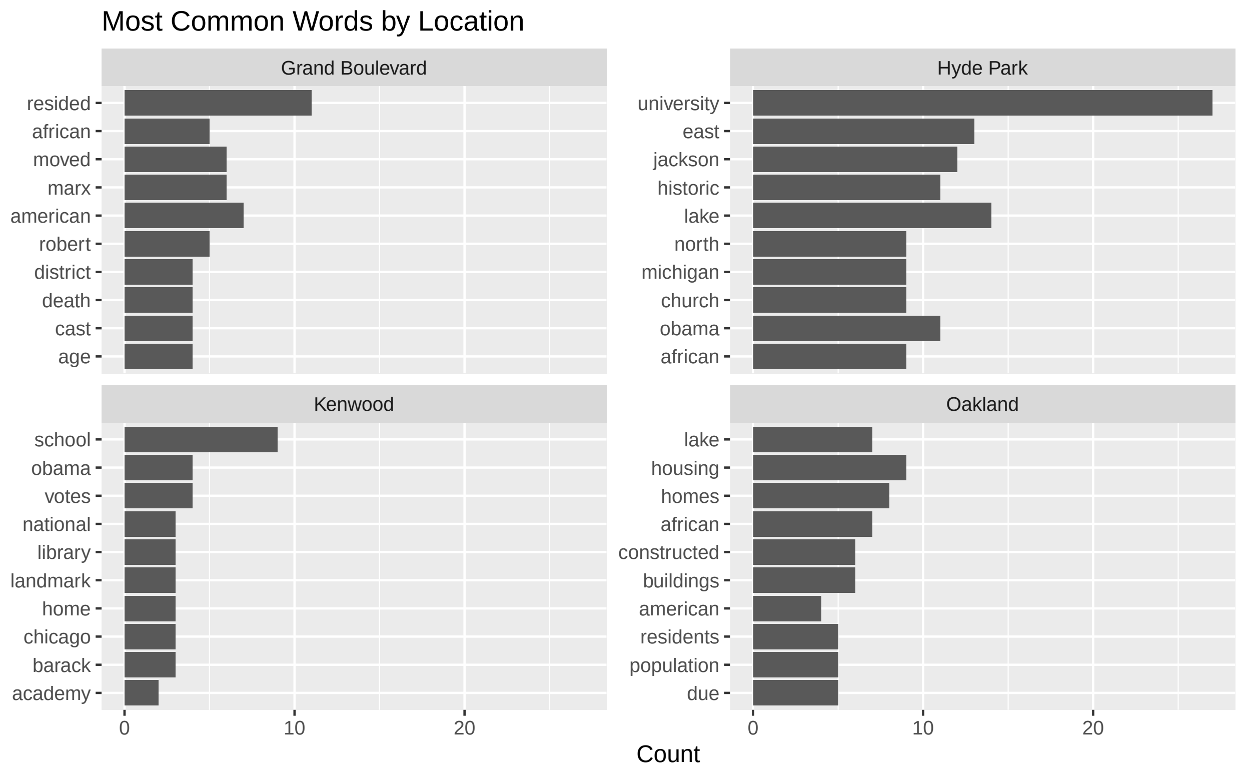
```
tokens_clean <- tokens %>%
  mutate(token = str_replace(token, "'s$", "")) %>%
  filter(str_detect(token, "^[a-z]+$"))

loc_words <- tolower(descriptions$location) |>
  str_split("\\s+") |> unlist() |> unique()

custom_stop <- tibble(word = c(loc_words, "south","street","avenue","park","house","city","community","

top_by_loc_clean <- tokens_clean %>%
  anti_join(stop_words, by = c("token" = "word")) %>%
  anti_join(custom_stop, by = c("token" = "word")) %>%
  count(location, token, sort = TRUE) %>%
  group_by(location) %>%
  slice_max(n, n = 10, with_ties = FALSE) %>%
  ungroup() %>%
  mutate(token = fct_reorder(token, n))

ggplot(top_by_loc_clean, aes(x = token, y = n)) +
  geom_col() +
  coord_flip() +
  facet_wrap(~ location, scales = "free_y") +
  labs(title = "Most Common Words by Location", x = NULL, y = "Count")
```



Discussion

Similarities: All four areas share history & demographics vocabulary: *african, american, residents, historic*, which fits South Side community histories.

Differences: The chart shows the most common words appearing in Wikipedia text for each Chicago neighborhood.

Grand Boulevard emphasizes historical and demographic terms such as resided, African, district, and American, reflecting its Bronzeville heritage and focus on community identity.

Hyde Park is dominated by words like university, historic, church, and Obama, highlighting its academic, cultural, and architectural significance—anchored by the University of Chicago.

Kenwood features school, Obama, library, and landmark, pointing to its residential and historical prominence, as well as ties to notable figures.

Oakland includes lake, housing, homes, and population, suggesting themes of urban development, residential life, and community revitalization near the lakeshore.

Overall, each neighborhood's vocabulary aligns with its distinct social, cultural, and historical identity within Chicago's South Side.