GY7708 2020-21 Assignment 2

Student ID: 209047191

Date of Submission: 10/05/2021

Contents

1 GY7708 Assignment 2: Geospatial Information Processing Project

This assignment entails developing skills in Wikipedia textual and spatial data analysis. With the data focused in the Wikipedia articles with assigned geo-tags in Greater London for the assigned borough and this document was created in RMarkdown.

This assignment was created using R, Rstudio, RMarkdown and GitHub.

Github Link: https://github.com/space-uni/GY7708_CW2 Shorten by bitly.com: https://bit.ly/2SGNeLV

1.1 References

This assignment would like to acknowledge that this document includes teaching materials from Dr Stefano De Sabbata for the module GY7708 Geospatial Databases and Information Retrieval. And the associated teaching materials can be found here.

This repository / document contains public sector information with Office for National Statistics (https://geoportal.statistics.gov.uk/) licensed under the Open Government Licence v3.0: wikipedia_geotags_in_GreaterLondon.csv and London_Ward.shp. And regarding the data derived from Wikimedia Downloads (https://dumps.wikimedia.org/legal.html) licensed under the GNU Free Documentation License (GFDL, https://www.wikipedia.org/wiki/Wikipedia:Copyrights) and the Creative Commons Attribution-Share-Alike 3.0 License (https://creativecommons.org/licenses/by-sa/3.0/), and text from Wikipedia available under the Creative Commons Attribution-ShareAlike License (https://creativecommons.org/licenses/by-sa/3.0/, additional terms may apply). See also **Statistical GIS Boundary Files for London**.

1.2 Introduction

Main Datasets:

- wikipedia_geotags_in_GreaterLondon.csv: a dataset containing Wikipedia articles focused in Greater London. The articles are also geo-tagged.
- London_Ward.shp: a shapefile containing boundary information within the city of london. Shapefile includes county and town name data. See also **Statistical GIS Boundary Files for London**

```
# load packages
library(stopwords)            # for stopwords data
library(WikipediR)            # for wikipedia data
library(tidyverse)
library(tidytext)             # for tidying text data
library(magrittr)
library(jsonlite)             # for parsing JSOn data
library(httr)                 # for working with HTTP and URLs
library(ggspatial)            # for ggplot visualisation
library(ggplot2)
library(wordcloud2)           # for wordcloud2 visualisation
library(scales)               # for plot visualisaton
library(webshot)              # for screenshots of web pages
library(htmlwidgets)          # for creating HTML widgets
webshot::install_phantomjs()  # for the webshot package
library(sf)
```

1.3 Part 1

```
# reading in the wikipedia dataset
wiki <- readr::read_csv("wikipedia_geotags_in_GreaterLondon.csv")
wiki %>% head(5)
```

```
## # A tibble: 5 x 25
##   lad_name     gt_id gt_page_id gt_globe gt_primary gt_lat gt_lon gt_dim gt_type
##   <chr>        <dbl>      <dbl> <chr>         <dbl>  <dbl>  <dbl>  <dbl> <chr>
## 1 Barking an~ 5.98e8   51135731 earth             1   51.5 0.119     625 NULL
## 2 Barking an~ 5.99e8   51888773 earth             1   51.5 0.0945   1000 NULL
## 3 Barking an~ 6.03e8   51888589 earth             1   51.5 0.0997   1000 NULL
## 4 Barking an~ 6.07e8   51509132 earth             1   51.6 0.143     312 NULL
## 5 Barking an~ 6.09e8   54638106 earth             1   51.6 0.136      61 landma~
## # ... with 16 more variables: gt_name <chr>, gt_country <chr>, gt_region <chr>,
## #   page_id <dbl>, page_namespace <dbl>, page_title <chr>,
## #   page_restrictions <lgl>, page_is_redirect <dbl>, page_is_new <dbl>,
## #   page_random <dbl>, page_touched <dbl>, page_links_updated <dbl>,
## #   page_latest <dbl>, page_len <dbl>, page_content_model <chr>,
## #   page_lang <chr>
```

```
# filtering to Ealing town
Ealing <-
  dplyr::filter(
    wiki, lad_name == "Ealing") %>%
  dplyr::select(
    gt_id, gt_lat, gt_lon, gt_dim, gt_type, page_id, page_title,
    page_random, page_touched, page_links_updated, page_latest, page_len
    )

Ealing %>% head(5)
```

```
## # A tibble: 5 x 12
##       gt_id gt_lat gt_lon gt_dim gt_type  page_id page_title         page_random
##       <dbl>  <dbl>  <dbl>  <dbl> <chr>      <dbl> <chr>                    <dbl>
## 1 445848918   51.5 -0.303   1000 landmark  5.24e7 Ealing_Grove             0.893
## 2 608782873   51.5 -0.291   1000 NULL      5.75e7 Hanger_Hill_Wood         0.828
## 3 610962056   51.5 -0.318   1000 landmark  5.57e7 Northfield_Allotm~       0.384
## 4 621749595   51.5 -0.277   1000 landmark  5.46e7 LH2_Studios              0.113
## 5 623377918   51.5 -0.308    500 NULL      6.18e7 Victoria_Hall_(Ea~       0.894
## # ... with 4 more variables: page_touched <dbl>, page_links_updated <dbl>,
## #   page_latest <dbl>, page_len <dbl>
```

```
page_summary <- function(a_page_title) {
  a_page_summary <-
  httr::GET(
    # Base API URL
    url = "https://en.wikipedia.org/w/api.php",
    # API query definition
    query = list(
    # Use JSON data format
    format = "json",
```

```
    action = "query",
    # Only retrieve the intro
    prop = "extracts",
    exintro = 1,
    explaintext = 1,
    redirects = 1,
    # Set the title
    titles = a_page_title
    )
  ) %>%
    # Get the content
    httr::content(
    as = "text",
    encoding = "UTF-8"
  ) %>%
  # Trasnform JSON content to R list
  jsonlite::fromJSON() %>%
  # Extract the summary from the list
  magrittr::extract2("query") %>%
  magrittr::extract2("pages") %>%
  magrittr::extract2(1) %>%
  magrittr::extract2("extract")

  return(a_page_summary)
}
```

```
Ealing %>% colnames %>% knitr::kable()
```

x
gt_id
gt_lat
gt_lon
gt_dim
gt_type
page_id
page_title
page_random
page_touched
page_links_updated
page_latest
page_len

```
# assigning to a new variable
Ealing_index <- Ealing

# looping thorough page_title column and applying the above page_summary function
# and saving results within the above new variable
for (i in Ealing["page_title"]) {
  Ealing_index <- Ealing_index %>%
  dplyr::mutate(page_summary = lapply(i, page_summary)
  )
```

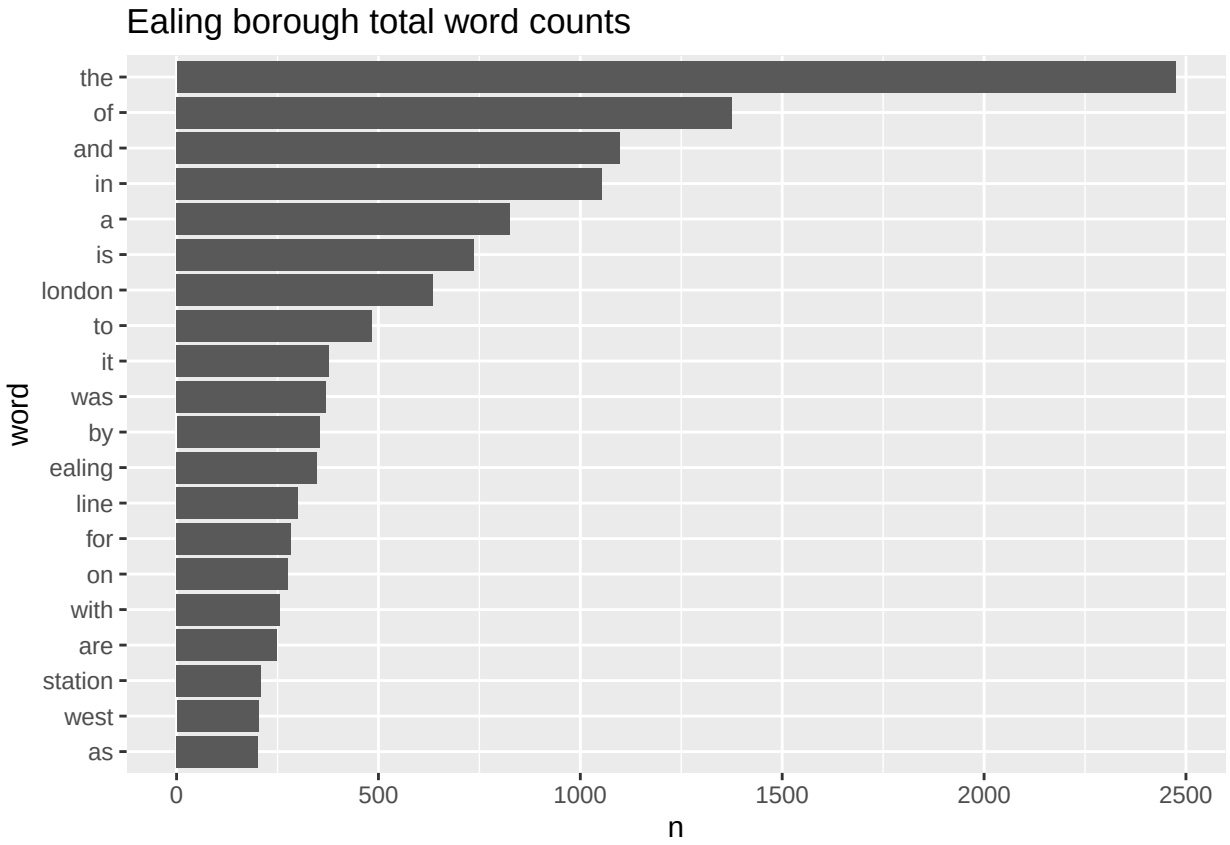
```
}
```

1.4 Part 2

This section will be an spatial frequency analysis of the retrieved page summaries from the Wikipedia articles focused in Ealing borough. As it would be interesting to investigate the frequency of the words across the whole dataset. Of course before any analysis can begin, the textual data of the page summaries needs to be pre-process via tokenisation by single words. Afterwards, to accommodate the unnecessary words such as "and, I, he, etc" named as stopwords, these words are removed via an antijoin function with a stopwords function providing a list of the required stopwords data. Following on producing a wordcloud will be highly beneficial in providing an visualization of the most number of used words in the page summaries of Wikipedia articles from the processed dataset. Finally, investigating the frequency of the words by how it varies spatially in Ealing, for example the distribution of the number of words in different towns in the borough. This last analysis can be done be including a county shapefile of Ealing with town boundary information so the processed data can be analysed further. Combined with the geo-tagged Wikipedia articles, it will possible to determine which points lie in which town polygon area. The resulting plot will be helpful in analysing the frequency of words and how it varies spatially across different towns in Ealing.

```
# tidying the dataset by tokenisation
Ealing_tidy <- Ealing_index %>%
  dplyr::select(
    gt_id, page_id, gt_lat, gt_lon, page_title, page_summary
    ) %>%
  tidytext::unnest_tokens(word, page_summary)

# visualising the number of words counts
Ealing_tidy %>%
  dplyr::count(word, sort = TRUE) %>%
  dplyr::slice_max(n, n = 20) %>%
  dplyr::mutate(word = reorder(word, n)) %>%
  ggplot2::ggplot(aes(n, word)) +
  ggplot2::geom_col() +
  ggplot2::ggtitle("Ealing borough total word counts")
```

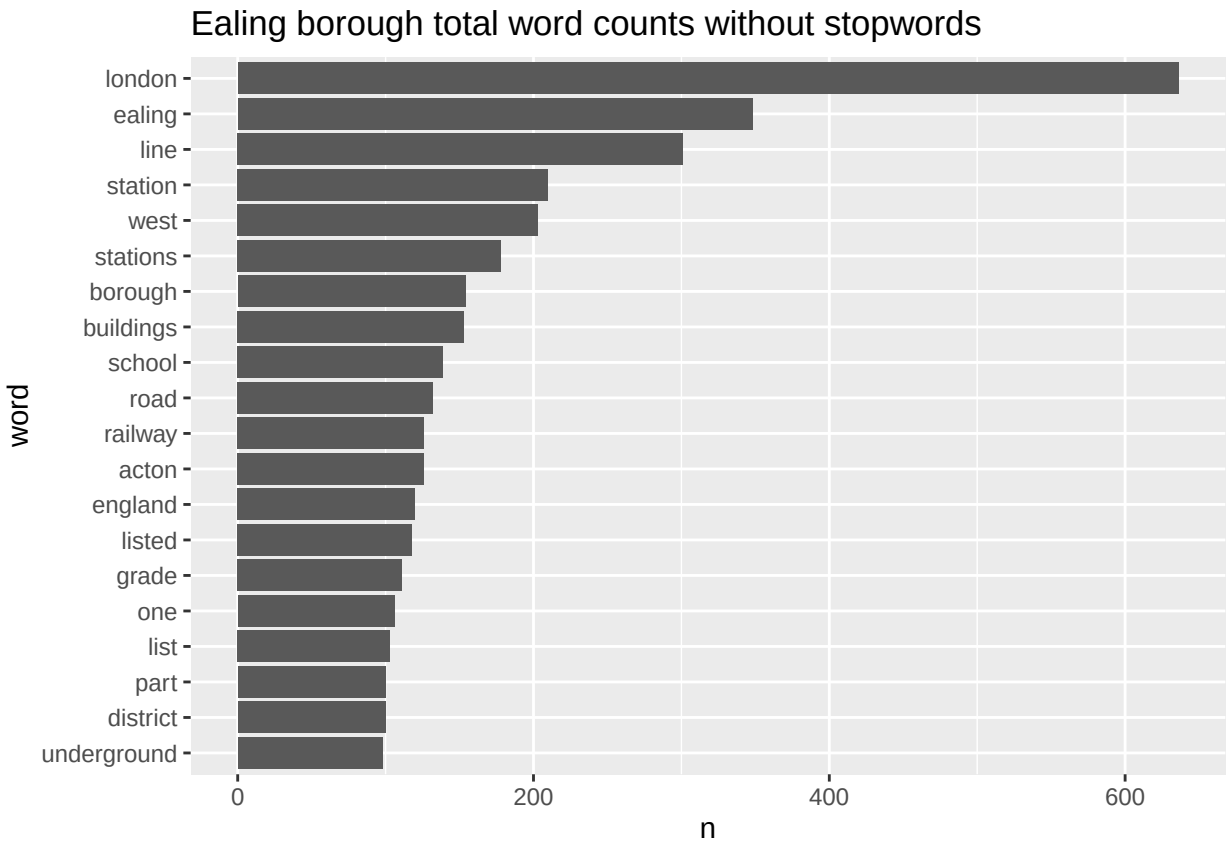



```
# displaying the top 20 words counts without stopwords
Ealing_tidy %>%
  # removing stopwords
  dplyr::anti_join(get_stopwords()) %>%
  dplyr::count(word, sort = TRUE) %>%
  dplyr::slice_max(n, n = 20)
```

```
## # A tibble: 20 x 2
##    word            n
##    <chr>       <int>
##  1 london        636
##  2 ealing        348
##  3 line          301
##  4 station       210
##  5 west          203
##  6 stations      178
##  7 borough       154
##  8 buildings     153
##  9 school        139
## 10 road          132
## 11 acton         126
## 12 railway       126
## 13 england       120
## 14 listed        118
## 15 grade         111
## 16 one           106
```

```
## 17 list          103
## 18 district      100
## 19 part          100
## 20 underground    98
```

```
# visualising the number of words counts without stopwords in a bar chart
Ealing_tidy %>%
  dplyr::anti_join(get_stopwords()) %>%
  dplyr::count(word, sort = TRUE) %>%
  dplyr::slice_max(n, n = 20) %>%
  dplyr::mutate(word = reorder(word, n)) %>%
  ggplot2::ggplot(aes(n, word)) +
  ggplot2::geom_col() +
  ggplot2::ggtitle("Ealing borough total word counts without stopwords")
```




```
# inspired by Yan Holtz at https://www.r-graph-gallery.com/196-the-wordcloud2-library.html
# create wordcloud
wordcloud <- Ealing_tidy %>%
  dplyr::select(word) %>%
  # removing stopwords
  dplyr::anti_join(get_stopwords()) %>%
  dplyr::count(word, sort = TRUE) %>%
  wordcloud2::wordcloud2(shape = "diamond")

# to visualise in a pdf knit by saving as a html format
```

```
# and tranforming it to a png format
saveWidget(wordcloud, "tmp.html", selfcontained = F)
webshot("tmp.html", "wc1.png", delay = 5, vwidth = 750, vheight = 750)
```


```
#![wordcloud](wc1.png)
```

```
# Transforming dataset to a simple feature class
Ealing_tidy_sf <- Ealing_tidy %>%
  dplyr::anti_join(get_stopwords()) %>%
  sf::st_as_sf(coords = c("gt_lon","gt_lat")) %>%
  sf::st_set_crs("OGC:CRS84") %>%
  # projecting to a British National Grid CRS
```

```
  sf::st_transform(27700)

st_crs(Ealing_tidy_sf)
```

```
## Coordinate Reference System:
##   User input: EPSG:27700
##   wkt:
## PROJCRS["OSGB 1936 / British National Grid",
##     BASEGEOGCRS["OSGB 1936",
##         DATUM["OSGB 1936",
##             ELLIPSOID["Airy 1830",6377563.396,299.3249646,
##                 LENGTHUNIT["metre",1]]],
##         PRIMEM["Greenwich",0,
##             ANGLEUNIT["degree",0.0174532925199433]],
##         ID["EPSG",4277]],
##     CONVERSION["British National Grid",
##         METHOD["Transverse Mercator",
##             ID["EPSG",9807]],
##         PARAMETER["Latitude of natural origin",49,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8801]],
##         PARAMETER["Longitude of natural origin",-2,
##             ANGLEUNIT["degree",0.0174532925199433],
##             ID["EPSG",8802]],
##         PARAMETER["Scale factor at natural origin",0.9996012717,
##             SCALEUNIT["unity",1],
##             ID["EPSG",8805]],
##         PARAMETER["False easting",400000,
##             LENGTHUNIT["metre",1],
##             ID["EPSG",8806]],
##         PARAMETER["False northing",-100000,
##             LENGTHUNIT["metre",1],
##             ID["EPSG",8807]]],
##     CS[Cartesian,2],
##         AXIS["(E)",east,
##             ORDER[1],
##             LENGTHUNIT["metre",1]],
##         AXIS["(N)",north,
##             ORDER[2],
##             LENGTHUNIT["metre",1]],
##     USAGE[
##         SCOPE["Engineering survey, topographic mapping."],
##         AREA["United Kingdom (UK) - offshore to boundary of UKCS within 49Â°45'N to 61Â°N and 9Â°W t
##         BBOX[49.75,-9,61.01,2.01]],
##     ID["EPSG",27700]]
```

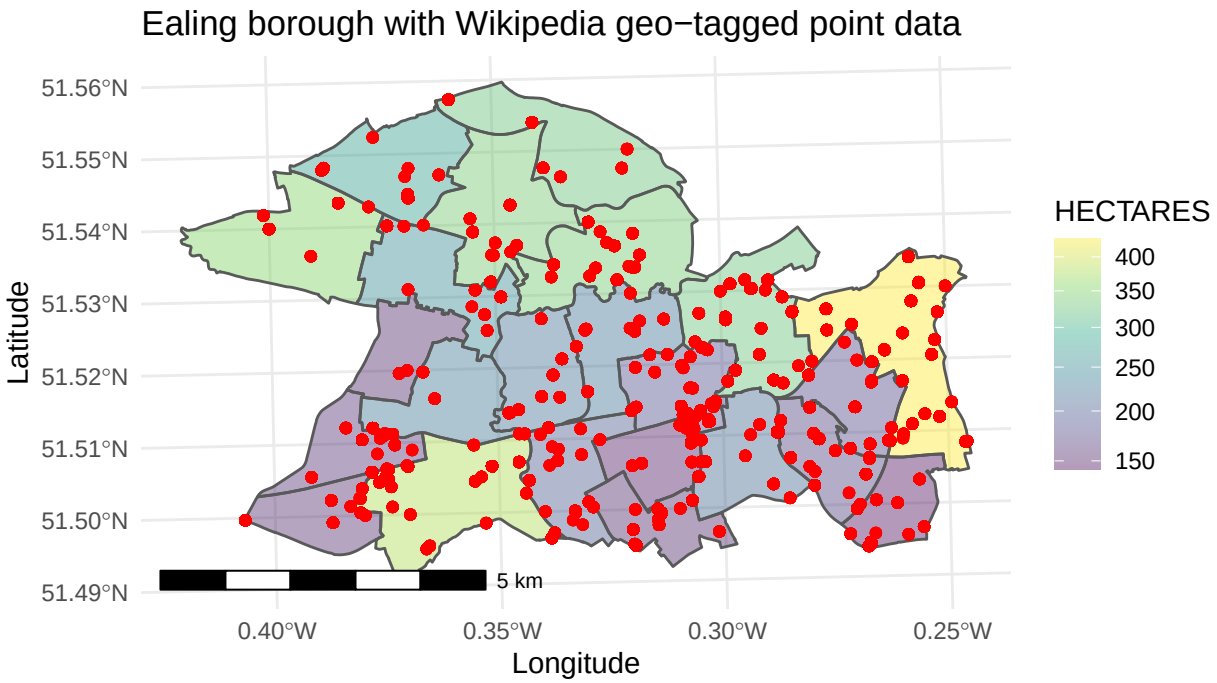
```
Ealing_sf <- st_read("shapefiles/London_Ward.shp") %>%
  # filtering to Ealing town
  dplyr::filter(BOROUGH == "Ealing") %>%
  # removing unnecessary columns
  dplyr::select(-NONLD_AREA, -BOROUGH) %>%
  # renaming town column name
  dplyr::rename(TOWN_NAME = NAME) %>%
```

```
  #projecting to British National Grid
  sf::st_transform(27700) %>%
  sf::st_cast("POLYGON") %>%
  sf::st_as_sf()
```

```
# checking the columns of the pre-processed shapefile
Ealing_sf %>% colnames
```

```
## [1] "TOWN_NAME" "GSS_CODE"  "HECTARES"  "LB_GSS_CD" "POLY_ID"   "geometry"
```

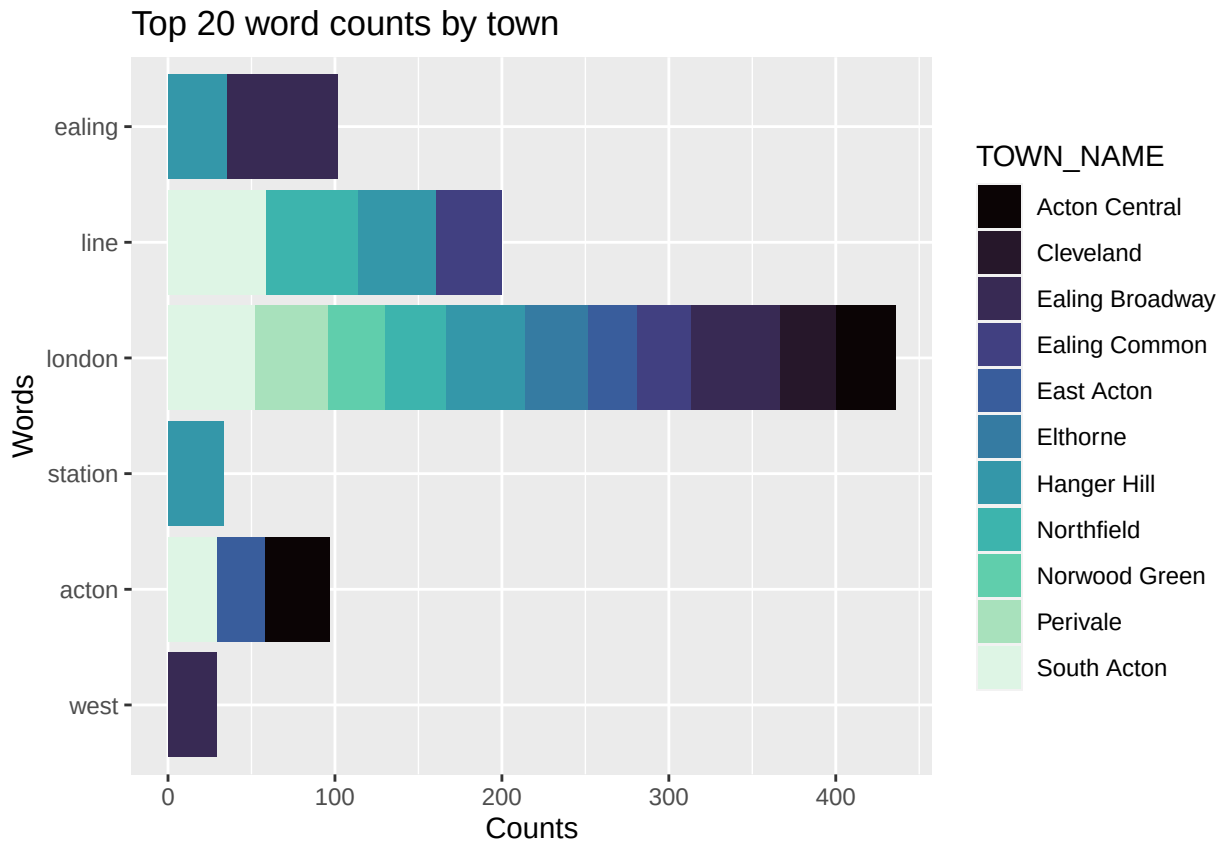
```
# style adapted from https://r-spatial.org/r/2018/10/25/ggplot2-sf-2.html
ggplot2::ggplot(cex=0.4) +
  ggspatial::geom_sf(data = Ealing_sf, aes(fill = HECTARES)) +
  ggplot2::scale_fill_viridis_c(trans = "sqrt", alpha = .4) +
  ggspatial::geom_sf(data = Ealing_tidy_sf, col="red") +
  ggplot2::theme_minimal() +
  ggspatial::annotation_scale(location = "bl", width_hint = 0.4) +
  ggspatial::annotation_north_arrow(
    location = "bl", which_north = "true",pad_x = unit(4.5, "in"),
    pad_y = unit(2.5, "in"),
    style = north_arrow_fancy_orienteering) +
  ggplot2::xlab("Longitude") +
  ggplot2::ylab("Latitude") +
  ggplot2::ggtitle("Ealing borough with Wikipedia geo-tagged point data")
```



```r
Ealing_sf_within <- Ealing_tidy_sf %>%
  sf::st_join(Ealing_sf, join = st_within)
```

```r
# assigning a new variable for top 20 word counts by town
Ealing_sf_within_count <- Ealing_sf_within %>%
  dplyr::count(word, TOWN_NAME, sort = TRUE) %>%
  dplyr::slice_max(n, n = 20)
```

```r
# inspired by Yan Holtz found at
# https://www.r-graph-gallery.com/48-grouped-barplot-with-ggplot2.html
library(viridis)
library(hrbrthemes)
Ealing_sf_within_count %>%
  dplyr::mutate(word = reorder(word, n)) %>%
  ggplot2::ggplot(aes(n, word,fill=TOWN_NAME)) +
  ggplot2::geom_bar(position="stack", stat="identity") +
  #ggplot2::geom_col()
  viridis::scale_fill_viridis(discrete = T, option = "mako") +
    ggplot2::ggtitle("Top 20 word counts by town") +
    ggplot2::xlab("Counts") +
    ggplot2::ylab("Words")
```



For the results of Part 1, it can be noted for the first initial spatial data bar chart plot, that the top three most included words were "the" by a large margin, then "of" and "and". This result is not helpful, therefore the related stopwords are removed in the following analysis. Thus, the results show that "London", "Ealing" and "line" has the top three most counts in the data in within the surrounding point data. It is also noted

that London has the most number of words in the dataset. This is evident and make sense in the dataset as London is present throughout which Ealing is a part of. The third most common word is “line” which is most interesting, which this can correlate to transportation modes such as station information. A word cloud including the entire dataset was also produced which reinforces the results obtained from the last plot and excellently displays as “London” as the most included word in the page summaries of the Wikipedia articles. Next the last plot of the top 20 word counts by towns in Ealing and shows again “London” prevalent in most of the towns with Ealing Broadway occupying the largest number of counts. Interestingly apart from towns relating to location and transportation information of words (e.g. “west” and “station”), is that “Acton” has several towns occupying a significant amount of total counts. This is significant as this is the first instance that a town name is referenced, therefore display the importance of this town and that more Wikipedia articles include reference to this town.

1.5 Part 3

To further the spatial analysis of the Wikipedia articles would be best to represent the sentiment of the words in the articles. By including an sentiment analysis of the textual data of the page summaries of the articles, it will be interesting to investigate whether the emotional intent have either an positive or negative connotation. In this particular analysis the most common positive and negative words are correlated against the town names of Ealing, therefore displaying which towns have an either emotional intent or neither of the options and have an balanced intent.

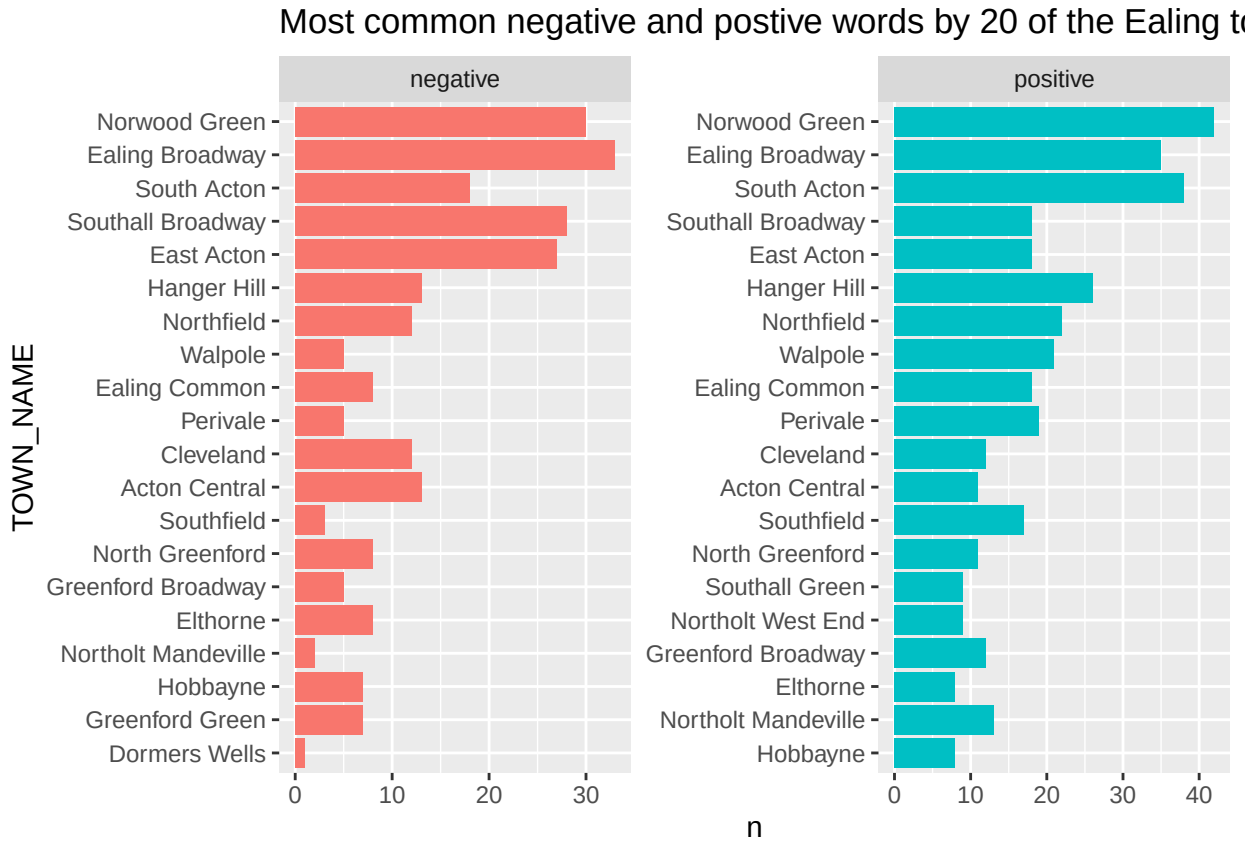
```
# adapted from Julia Silge at
# https://juliasilge.shinyapps.io/learntidytext/#section-shakespeare-gets-sentimental
Ealing_sentiment <- Ealing_sf_within %>%
  dplyr::inner_join(get_sentiments("bing")) %>%
  dplyr::count(TOWN_NAME, sentiment)

Ealing_sentiment %>% head(10)
```

```
## Simple feature collection with 10 features and 3 fields
## Geometry type: GEOMETRY
## Dimension:     XY
## Bounding box:  xmin: 512661.2 ymin: 179627.8 xmax: 520643 ymax: 182017.2
## Projected CRS: OSGB 1936 / British National Grid
## # A tibble: 10 x 4
##    TOWN_NAME    sentiment     n                                        geometry
##    <chr>        <chr>     <int>                                  <GEOMETRY [m]>
##  1 Acton Centr~ negative     13 MULTIPOINT ((520027.1 180176.4), (520098.7 1808~
##  2 Acton Centr~ positive     11 MULTIPOINT ((519382.8 181309.5), (520027.1 1801~
##  3 Cleveland    negative     12 MULTIPOINT ((515798.2 181749.5), (515973.8 1810~
##  4 Cleveland    positive     12 MULTIPOINT ((515798.2 181749.5), (515948.8 1820~
##  5 Dormers Wel~ negative      1                    POINT (513433.1 181351.1)
##  6 Dormers Wel~ positive      6 MULTIPOINT ((512661.2 180488.5), (513433.1 1813~
##  7 Ealing Broa~ negative     33 MULTIPOINT ((516708 181425.4), (517009.6 181352~
##  8 Ealing Broa~ positive     35 MULTIPOINT ((516722 180813.7), (516932.6 181619~
##  9 Ealing Comm~ negative      8 MULTIPOINT ((517688.2 179739.5), (518900.2 1804~
## 10 Ealing Comm~ positive     18 MULTIPOINT ((517688.2 179739.5), (517726.6 1803~
```

```
common_words <- Ealing_sentiment %>%
  # group by sentiment
  dplyr::group_by(sentiment) %>%
  dplyr::slice_max(n, n = 20) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(TOWN_NAME = reorder(TOWN_NAME, n))

ggplot(common_words, aes(n, TOWN_NAME, fill = sentiment)) +
  ggplot2::geom_col(show.legend = FALSE) +
  ggplot2::facet_wrap(~ sentiment, scales = "free") +
  ggplot2::ggtitle("Most common negative and postive words by 20 of the Ealing towns")
```



From the results of the sentiment analysis from the first plot is that overall the average of the total word counts are leaning towards a greater positive intent. With having "Norwood Green", "Ealing Broadway" and "South Acton" having a greater positive ratio compared to the negative. It is also interesting to note that apart from the aforementioned towns, the rest of the data seems to have fairly balanced responses with multiple towns leaning towards either side only slightly. The results from this analysis is indicative that certain towns have an positive intent in their respective Wikipedia articles with the rest of the data being fairly equal. Thus this is beneficial as towards the reader on these Wikipedia articles, the information will be less biased, therefore providing a informative and fair read.