HW9

Aidan Pizzo

November 2, 2021

#data.R

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(ggplot2)

load_dataset <- function()
{
  d <- data.table::fread("../../datasets/twitter support dataset/twcs.csv",
                         sep=",",
                         header=T,
                         stringsAsFactors = F)
  return(d)
}

get_all_companies <- function()
{
  d <- data.table::fread("../../datasets/twitter support dataset/twcs.csv",
                         sep=",",
                         header=T,
```

```
                          stringsAsFactors = F)
  company_ndx <- d$inbound == FALSE
  # grepping on alphabet: grepl("[[:alpha:]]", d$author_id)
  company_vec <- d$author_id[company_ndx] %>% unique
  return(company_vec)
}
```

#company.R

```
#class Company

construct.Company <- function(d, company_name)
{
  tweet_ndx <- d$author_id == company_name | grepl(company_name, d$text)
  company_tweets <- dplyr:: filter(d, tweet_ndx)
  return(company_tweets)
}

get_number_of_conversations.Company <- function(comp)
{
  authors <- comp$author_id %>% unique
  numeric_string_ndx <- grepl("[[:digit:]]", authors)
  authors <- authors[numeric_string_ndx]
  num_convos <- authors %>% length
  return(num_convos)
}

get_conversation.Company <- function(comp, i)
{
  authors <- comp$author_id %>% unique
  numeric_string_ndx <- grepl("[[:digit:]]", authors)
  authors <- authors[numeric_string_ndx]
  conv_with <- authors[i]
  conv_ndx <- comp$author_id == conv_with | grepl(conv_with, comp$text)
  out_df <- dplyr::filter(comp, conv_ndx)
  #conversation <- sort_convo(out_df)
  return(out_df)
}
```

#analysis.R

```
sort_convo <- function(convo_df)
{
  sorted_list <- list()
  ndx <- is.na(convo_df$in_response_to_tweet_id)
  ndx_row <- dplyr::filter(convo_df, ndx)
  sorted_list[[1]] <- ndx_row
  # while (ndx_row$response_tweet_id != "")
  for (i in 1:(nrow(convo_df)-1)) #maybe try while ndx_row$response_tweet_id is not empty
  {
    to_find <- strsplit(ndx_row$response_tweet_id, ",")
    find_this <- to_find[[1]][1]
    ndx <- grepl(find_this, convo_df$tweet_id)
    ndx_row <- dplyr::filter(convo_df, ndx)
    sorted_list[[i+1]] <- ndx_row
```

```
  }
  out_df <- do.call(rbind, sorted_list)
  out_df <- dplyr::select(out_df, author = author_id, text)
  return(out_df)
}

VirAtl_longest_conv <- function(d)
{
  VirAtl <- construct.Company(d, "VirginAtlantic")
  VirAtl_convos <- get_conv_lengths(VirAtl)
  max_convo <- which.max(VirAtl_convos$conv.length)
  out_convo <- get_conversation.Company(VirAtl, max_convo)
  return(out_convo)
}

get_conv_lengths <- function(comp)
{
  customers <- comp$author_id %>% unique
  numeric_string_ndx <- grepl("[[:digit:]]", customers)
  customers <- customers[numeric_string_ndx]
  count_list <- list()
  for (i in 1:length(customers))
  {
    user_id <- customers[i]
    convos_ndx <- comp$author_id == user_id | grepl(user_id, comp$text)
    convos <- comp$author_id[convos_ndx]
    out_df <- data.frame(conv = i, conv.length = convos %>% length,
                         stringsAsFactors = F)
    count_list[[i]] <- out_df
  }
  out_df <- do.call(rbind, count_list)
  return(out_df)
}

max_min_convo <- function(d)
{
  companies <- get_all_companies()
  companies <- setdiff(companies, "AmericanAir")
  avg_length <- Map(function(thisCompany){
    comp <- construct.Company(d, thisCompany)
    conv_df <- get_conv_lengths(comp)
    avg_num <- mean(conv_df$conv.length)
    out_df <- data.frame(company = thisCompany, avg.conv.length = avg_num,
                         stringsAsFactors = F)
    print(head(out_df))
    return(out_df)
  }, companies)
  mmconvo <- do.call(rbind, avg_length)
  return(mmconvo)
}

max_df <- function(companies_avg)
{
```

```
  max_ndx <- which.max(companies_avg$avg.conv.length)
  max <- companies_avg$company[max_ndx]
  return(max)
}

min_df <- function(companies_avg)
{
  min_ndx <- which.min(companies_avg$avg.conv.length)
  min <- companies_avg$company[min_ndx]
  return(min)
}
```

#presentation.R

```
d <- load_dataset()
VirAtl_longest_conv(d)
```

```
##      tweet_id      author_id inbound                       created_at
##   1:      191          78417    TRUE Tue Oct 03 20:50:15 +0000 2017
##   2:      222          78432    TRUE Tue Oct 03 13:39:47 +0000 2017
##   3:      288 VirginAtlantic   FALSE Tue Oct 03 10:43:02 +0000 2017
##   4:     9985          80743    TRUE Sat Sep 23 17:32:24 +0000 2017
##   5:     9981 VirginAtlantic   FALSE Tue Oct 03 07:46:38 +0000 2017
##  ---
## 105:  1758139         190890    TRUE Tue Nov 07 02:08:36 +0000 2017
## 106:  1758134 VirginAtlantic   FALSE Tue Nov 07 04:31:49 +0000 2017
## 107:  1758135 VirginAtlantic   FALSE Tue Nov 07 04:31:40 +0000 2017
## 108:  1758140 VirginAtlantic   FALSE Tue Nov 07 02:35:15 +0000 2017
## 109:  1830221          78418    TRUE Thu Nov 09 13:35:46 +0000 2017
##
##   1:                                                  @78418 @VirginAtlantic What's going on with fligh
##   2:
##   3:                           @78452 2/2 With regards to your holiday (hotel) you'll need to speak to
##   4:                                                   @78418 @VirginAtlantic - Hello, Can you advise h
##   5:                            @80743 1/2 OK, I can't speak for @78418 because they're a separate com
##  ---
## 105:                                        @78418 @VirginAtlantic @78389 i guess after 4 months a cu
## 106:                                  @190890 @78418 @78389 required.  Please email details to our team
## 107:         @190890 @78418 @78389 OK Tony, you'll need to submit an official formal complaint as we
## 108: @190890 @78418 @78389 You should certainly have had a response by now Tony as our Customer Relat
## 109:                                       @VirginAtlantic @526875 Hi Jamie,it looks like this was be
##      response_tweet_id in_response_to_tweet_id
##   1:           190,192                      NA
##   2:               221                      NA
##   3:                                       285
##   4:              9984                      NA
##   5:                                      9980
##  ---
## 105:   1758138,1758140                      NA
## 106:                                   1758133
## 107:                                   1758133
## 108:                                   1758139
## 109:                                   1830220
```

```
# avg_convo_lengths <- max_min_convo(d)
# max_df(avg_convo_lengths)
# min_df(avg_convo_lengths)
```