Nishant Arora

CMSC 320

Worked with William Heberer

Project 1: Data Scraping and Cleaning

Part 1: Data scraping and preparation

Step 1: Scrape Data for SpaceWeatherLive.com

- Use the read_html function to read the html page from the url above
- Use the html_node function to find the page node corresponding to the table
- Use the html_table to parse the table into a data frame
- Finish the pipeline with a call to as_data_frame to make the data frame

```
dl_tab <- url %>%
  read_html() %>%
  html_node(".table-striped") %>%
  html_table() %>%
  as.data.frame()
```

- Rename attributes to some reasonable names

```
names(dl_tab) <- c("rank", "flare_classification", "date", "flare_region", "start_time",
 "maximum_time", "end_time", "movie")
```

The resulting tibble:

```
##   rank flare_classification       date flare_region start_time
## 1    1                X28.0 2003/11/04          486      19:29
## 2    2                X20.0 2001/04/02         9393      21:32
## 3    3                X17.2 2003/10/28          486      09:51
## 4    4                X17.0 2005/09/07          808      17:17
## 5    5                X14.4 2001/04/15         9415      13:19
## 6    6                X10.0 2003/10/29          486      20:37
##   maximum_time end_time             movie
## 1        19:53    20:06 MovieView archive
## 2        21:51    22:03 MovieView archive
## 3        11:10    11:24 MovieView archive
## 4        17:40    18:03 MovieView archive
## 5        13:50    13:55 MovieView archive
## 6        20:49    21:01 MovieView archive
```

Step 2: Tidy the top 50 solar flare data

- Drop the last column of the table

```
dl_tab$movie <- NULL
```

- Combine the date and each of the three time columns into three datetime columns

```
dl_tab <- unite(dl_tab, start_datetime, date, start_time, sep = " ", remove = FALSE)
dl_tab <- unite(dl_tab, max_datetime, date, maximum_time, sep = " ", remove = FALSE)
dl_tab <- unite(dl_tab, end_datetime, date, end_time, sep = " ", remove = FALSE)

# remove extra columns
dl_tab$date <- NULL
dl_tab$start_time <- NULL
dl_tab$maximum_time <- NULL
dl_tab$end_time <- NULL
```

- Convert columns containing datetimes into actual datetime objects

```
# in POSIX form
dl_tab$start_datetime <- as.POSIXct(dl_tab$start_datetime, tz="")
dl_tab$max_datetime <- as.POSIXct(dl_tab$max_datetime, tz="")
dl_tab$end_datetime <- as.POSIXct(dl_tab$end_datetime, tz="")
```

The resulting tibble:

```
##   rank flare_classification      start_datetime        max_datetime
## 1    1                X28.0 2003-11-04 19:29:00 2003-11-04 19:53:00
## 2    2                X20.0 2001-04-02 21:32:00 2001-04-02 21:51:00
## 3    3                X17.2 2003-10-28 09:51:00 2003-10-28 11:10:00
## 4    4                X17.0 2005-09-07 17:17:00 2005-09-07 17:40:00
## 5    5                X14.4 2001-04-15 13:19:00 2001-04-15 13:50:00
## 6    6                X10.0 2003-10-29 20:37:00 2003-10-29 20:49:00
##          end_datetime flare_region
## 1 2003-11-04 20:06:00          486
## 2 2001-04-02 22:03:00         9393
## 3 2003-10-28 11:24:00          486
## 4 2005-09-07 18:03:00          808
## 5 2001-04-15 13:55:00         9415
## 6 2003-10-29 21:01:00          486
```

Step 3: Scrape the NASA data

- Obtain each row of data as a long string.
- Create a data_frame
- Separate each line of text into a data row.
- Choose appropriate names for columns.

```
nasa <- "https://cdaw.gsfc.nasa.gov/CME_list/radio/waves_type2.html"

# scrape nasa table
nasa_tab <- nasa %>%
  read_html() %>%
  html_node("pre") %>%
  html_text() %>%
  str_split("\\n") %>%
  as.data.frame() %>%
  slice(13:n()-3) %>%
  slice(4:n()) %>%
  separate(1, c("start_date","start_time", "end_date", "end_time", "start_frequency", "e
nd_frequency", "flare_location", "flare_region", "flare_classification", "cme_date", "cm
e_time", "cme_angle", "cme_width", "cme_speed"), sep="[:space:]+")
```

The resulting tibble:

```
## Warning: Expected 14 pieces. Additional pieces discarded in 511 rows [1, 2,
## 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
## # A tibble: 511 x 14
##    start_date start_time end_date end_time start_frequency end_frequency
##    <chr>      <chr>      <chr>    <chr>    <chr>           <chr>
##  1 1997/04/01 14:00      04/01    14:15    8000            4000
##  2 1997/04/07 14:30      04/07    17:30    11000           1000
##  3 1997/05/12 05:15      05/14    16:00    12000           80
##  4 1997/05/21 20:20      05/21    22:00    5000            500
##  5 1997/09/23 21:53      09/23    22:16    6000            2000
##  6 1997/11/03 05:15      11/03    12:00    14000           250
##  7 1997/11/03 10:30      11/03    11:30    14000           5000
##  8 1997/11/04 06:00      11/05    04:30    14000           100
##  9 1997/11/06 12:20      11/07    08:30    14000           100
## 10 1997/11/27 13:30      11/27    14:00    14000           7000
## # ... with 501 more rows, and 8 more variables: flare_location <chr>,
## #   flare_region <chr>, flare_classification <chr>, cme_date <chr>,
## #   cme_time <chr>, cme_angle <chr>, cme_width <chr>, cme_speed <chr>
```

Step 4: Tidy the NASA the table

- Recode any missing entries as NA

```
nasa_tab[nasa_tab=="????"] <- NA
nasa_tab[nasa_tab=="BACK"] <- NA
nasa_tab[nasa_tab=="Back"] <- NA
nasa_tab[nasa_tab=="Back?"] <- NA
nasa_tab[nasa_tab=="----"] <- NA
nasa_tab[nasa_tab=="---"] <- NA
nasa_tab[nasa_tab=="------"] <- NA
nasa_tab[nasa_tab=="-----"] <- NA
nasa_tab[nasa_tab=="--:--"] <- NA
nasa_tab[nasa_tab=="--/--"] <- NA
nasa_tab[nasa_tab=="LASCO DATA GAP"] <- NA
```

- Create a new (logical) column that indicates if a row corresponds to a halo flare or not, and then replace Halo entries in the cme_angle column as NA.
- Create a new (logical) column that indicates if width is given as a lower bound, and remove any non-numeric part of the width column.

```
tidy_nasa_tab <- nasa_tab %>%
  mutate(halo = cme_angle == "Halo") %>%
  mutate(cme_width_limit = cme_width == str_match(cme_width, ">\\d+")) %>%
  mutate(cme_width_limit = !is.na(cme_width_limit)) %>%
  separate(cme_width, c("trash", "cme_width"), sep=">", fill = "left")
```

- Combine date and time columns for start, end and cme so they can be encoded as datetime objects.

```
# converting dates and times to single datetime columns
tidy_nasa_tab <- unite(tidy_nasa_tab, start_datetime, start_date, start_time, sep = " ",
 remove = FALSE)
tidy_nasa_tab <- unite(tidy_nasa_tab, end_datetime, end_date, end_time, sep = " ", remov
e = FALSE)
tidy_nasa_tab <- unite(tidy_nasa_tab, cme_datetime, cme_date, cme_time, sep = " ", remov
e = FALSE)

# clearing excess columns and getting rid of 'halo'
tidy_nasa_tab$start_date <- NULL
tidy_nasa_tab$start_time <- NULL
tidy_nasa_tab$end_date <- NULL
tidy_nasa_tab$end_time <- NULL
tidy_nasa_tab$cme_date <- NULL
tidy_nasa_tab$cme_time <- NULL
tidy_nasa_tab$trash <- NULL
tidy_nasa_tab[tidy_nasa_tab=="Halo"] <- NA

# grabbing year from start date and adding it on to the others
tidy_nasa_tab <- separate(tidy_nasa_tab, start_datetime, c("temp", "start_datetime"), se
p = "/", extra = "merge")

# uniting year to all datetime columns
tidy_nasa_tab <- unite(tidy_nasa_tab, start_datetime, temp, start_datetime, sep = "/", r
emove = FALSE)
tidy_nasa_tab <- unite(tidy_nasa_tab, end_datetime, temp, end_datetime, sep = "/", remov
e = FALSE)
tidy_nasa_tab <- unite(tidy_nasa_tab, cme_datetime, temp, cme_datetime, sep = "/", remov
e = FALSE)

# changing appropriate columns to datetime
tidy_nasa_tab <- mutate(tidy_nasa_tab, start_datetime = ymd_hm(start_datetime))
tidy_nasa_tab <- mutate(tidy_nasa_tab, end_datetime = ymd_hm(end_datetime))
tidy_nasa_tab <- mutate(tidy_nasa_tab, cme_datetime = ymd_hm(cme_datetime))
```

- Convert columns to appropriate types

```
tidy_nasa_tab <- type_convert(tidy_nasa_tab)
```

The resulting tibble:

```
## Warning: Expected 14 pieces. Additional pieces discarded in 511 rows [1, 2,
## 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
## Warning: 22 failed to parse.
```

```
## Parsed with column specification:
## cols(
##   temp = col_integer(),
##   start_frequency = col_integer(),
##   end_frequency = col_integer(),
##   flare_location = col_character(),
##   flare_region = col_character(),
##   flare_classification = col_character(),
##   cme_angle = col_integer(),
##   cme_width = col_character(),
##   cme_speed = col_integer()
## )
```

```
## # A tibble: 511 x 14
##    start_datetime      end_datetime        cme_datetime         temp
##    <dttm>              <dttm>              <dttm>              <int>
##  1 1997-04-01 14:00:00 1997-04-01 14:15:00 1997-04-01 15:18:00  1997
##  2 1997-04-07 14:30:00 1997-04-07 17:30:00 1997-04-07 14:27:00  1997
##  3 1997-05-12 05:15:00 1997-05-14 16:00:00 1997-05-12 05:30:00  1997
##  4 1997-05-21 20:20:00 1997-05-21 22:00:00 1997-05-21 21:00:00  1997
##  5 1997-09-23 21:53:00 1997-09-23 22:16:00 1997-09-23 22:02:00  1997
##  6 1997-11-03 05:15:00 1997-11-03 12:00:00 1997-11-03 05:28:00  1997
##  7 1997-11-03 10:30:00 1997-11-03 11:30:00 1997-11-03 11:11:00  1997
##  8 1997-11-04 06:00:00 1997-11-05 04:30:00 1997-11-04 06:10:00  1997
##  9 1997-11-06 12:20:00 1997-11-07 08:30:00 1997-11-06 12:10:00  1997
## 10 1997-11-27 13:30:00 1997-11-27 14:00:00 1997-11-27 13:56:00  1997
## # ... with 501 more rows, and 10 more variables: start_frequency <int>,
## #   end_frequency <int>, flare_location <chr>, flare_region <chr>,
## #   flare_classification <chr>, cme_angle <int>, cme_width <chr>,
## #   cme_speed <int>, halo <lgl>, cme_width_limit <lgl>
```

Part 2: Analysis

Question 1: Replication

- Can you replicate the top 50 solar flare table in SpaceWeatherLive.com exactly using the data obtained from NASA? That is, if you get the top 50 solar flares from the NASA table based on their classification (e.g., X28 is the highest), do you get data for the same 50 solar flare events in the SpaceWeatherLive page? If not, why not?
- Include code used to get the top 50 solar flares from the NASA table * Write a sentence or two discussing how well you can replicate the SpaceWeatherLive data from the NASA data.

It isn't possible replicate the table from SpaceWeatherLive exactly from the NASA data because the NASA data and SpaceWeatherLive data have some small yet significant differences. Specifically, the NASA data is missing a few events that the SWL table has, so naturally, the tables are going to be a little different. They do, however, share a significant amount of events as well, meaning that the SWL data can be fairly well replicated, but not completely.

```
# getting top 50
top_fifty <- tidy_nasa_tab %>%
  separate(flare_classification, c("class", "number"), sep=1) %>%
  filter(class=="X") %>%
  type_convert() %>%
  arrange(desc(number)) %>%
  slice(1:50) %>%
  unite(flare_classification, class, number, sep="", remove = FALSE)
```

```
## Parsed with column specification:
## cols(
##   flare_location = col_character(),
##   flare_region = col_integer(),
##   class = col_character(),
##   number = col_double(),
##   cme_width = col_integer()
## )
```

```
# cleanups
tidy_nasa_tab$temp <- NULL
top_fifty$class <- NULL
top_fifty$number <- NULL

top_fifty
```

```
## # A tibble: 50 x 14
##    start_datetime      end_datetime        cme_datetime         temp
##    <dttm>              <dttm>              <dttm>              <int>
##  1 2003-11-04 20:00:00 2003-11-05 00:00:00 2003-11-04 19:54:00  2003
##  2 2001-04-02 22:05:00 2001-04-03 02:30:00 2001-04-02 22:06:00  2001
##  3 2003-10-28 11:10:00 2003-10-30 00:00:00 2003-10-28 11:30:00  2003
##  4 2001-04-15 14:05:00 2001-04-16 13:00:00 2001-04-15 14:06:00  2001
##  5 2003-10-29 20:55:00 2003-10-30 00:00:00 2003-10-29 20:54:00  2003
##  6 1997-11-06 12:20:00 1997-11-07 08:30:00 1997-11-06 12:10:00  1997
##  7 2006-12-05 10:50:00 2006-12-05 20:00:00 NA                   2006
##  8 2003-11-02 17:30:00 2003-11-03 01:00:00 2003-11-02 17:30:00  2003
##  9 2005-01-20 07:15:00 2005-01-20 16:30:00 2005-01-20 06:54:00  2005
## 10 2011-08-09 08:20:00 2011-08-09 08:35:00 2011-08-09 08:12:00  2011
## # ... with 40 more rows, and 10 more variables: start_frequency <int>,
## #   end_frequency <int>, flare_location <chr>, flare_region <int>,
## #   flare_classification <chr>, cme_angle <int>, cme_width <int>,
## #   cme_speed <int>, halo <lgl>, cme_width_limit <lgl>
```

Question 2: Entity Resolution

- Write a function flare_similarity which computes a similarity s(e1,e2) between flares e1∈E1 and e2∈E2.

```
flare_similarity <- function(flare1, flare2) {

  date_f1 <- select(flare1, start_datetime)[[1,1]]
  region_f1 <- select(flare1, flare_region)[[1,1]]
  class_f1 <- select(flare1, flare_classification)[[1,1]]
  endDate_f1 <- select(flare1, end_datetime)[[1,1]]

  date_f2 <- select(flare2, start_datetime)[[1,1]]
  region_f2 <- select(flare2, flare_region)[[1,1]]
  class_f2 <- select(flare2, flare_classification)[[1,1]]
  endDate_f2 <- select(flare2, end_datetime)[[1,1]]

  sim <- as.numeric(min(c(date_f1, date_f2))) / as.numeric(max(c(date_f1, date_f2))) * 1
00
  sim <- as.numeric(min(c(endDate_f1, endDate_f2))) / as.numeric(max(c(endDate_f1, endDa
te_f2))) * 100

  if (!is.na(region_f1) && !is.na(region_f2) && region_f1 == region_f2) {
    sim <- sim + 1
  }

  if (!is.na(class_f1) && !is.na(class_f2) && class_f1 == class_f2) {
    sim <- sim + 1
  }

  sim <- sim / 4

  print(sim)
}
```

- Write a second function flare_match that computes for each flare e1∈E1 which flare e2∈E2 is the most similar.
- Add the result of flare_match to the top 50 table as the index of the best matching row in the NASA table, or NA.

```
flare_match <- function(df, flare) {
  vec <- c()

  for(i in 1:511) {
    vec[i] <- flare_similarity(flare, slice(df, i))
  }

  if(as.numeric(max(vec)) < 25) {
    NA
  } else {
    row <- which(vec == as.numeric(max(vec)))
    mutate(dl_tab, index=NA)
    dl_tab[as.numeric(select(flare, rank)), "index"] = row
  }
}
```

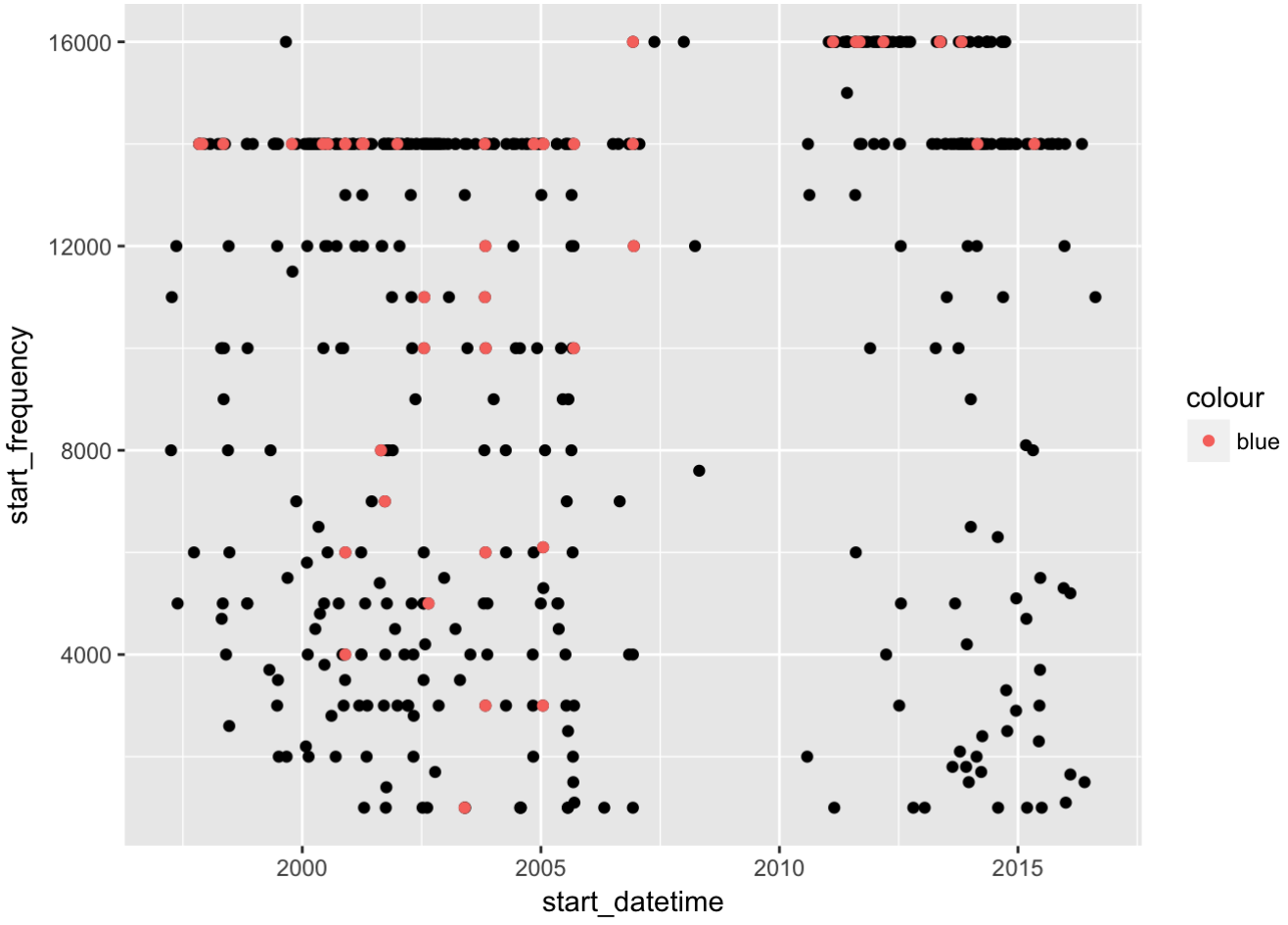
The similarity function used the main attributes from SpaceWeatherLive (start_datetime, end_datetime, classification, and region). Because the classification and region are categorical data types, I added them as 1, then added a percentage of similarity between the dates on top of all of that. I then divided everything by four to get an average similarity. Next, I applied the function to every row of the NASA table in order to find which flare matched best. If no row has a similarity rating above 25, NA is returned.

Question 3: Analysis

- Prepare one plot that shows the top 50 solar flares in context with all data available in the NASA dataset.

```
plot <- tidy_nasa_tab %>%
  ggplot(aes(x = start_datetime, y = start_frequency)) +
  geom_point() +
  geom_point(data = top_fifty, aes(colour="blue"))
plot
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```



This plot is the start frequency of the flares over time. The top 50 flares are represented with blue points, while the rest are in black. The intent of this plot is to observe the relationship between frequency and time of the flares, which we can can show with covariance. In this plot, it seems like there isn't much covariance between the two variables, as the points are scattered all over the place with no visible trend.