# Project_1

*Shashwat Kapoor*

*3/15/2019*

## Part 1

### Step 1:

```
url_sf <- "https://www.spaceweatherlive.com/en/solar-activity/top-50-solar-flares"

solar_flares <- url_sf %>%
  read_html() %>%
  html_node("table.table.table-striped.table-responsive-md") %>%
  html_table() %>%
  set_colnames(c("rank", "flare_classification", "date", "flare_region",
                 "start_time",  "maximum_time", "end_time", "movie")) %>%
  as_tibble()

solar_flares
```

```
## # A tibble: 50 x 8
##     rank flare_classific~ date  flare_region start_time maximum_time
##    <int> <chr>            <chr>        <int> <chr>      <chr>
##  1     1 X28.0            2003~          486 19:29      19:53
##  2     2 X20.0            2001~         9393 21:32      21:51
##  3     3 X17.2            2003~          486 09:51      11:10
##  4     4 X17.0            2005~          808 17:17      17:40
##  5     5 X14.4            2001~         9415 13:19      13:50
##  6     6 X10.0            2003~          486 20:37      20:49
##  7     7 X9.4             1997~         8100 11:49      11:55
##  8     8 X9.3             2017~         2673 11:53      12:02
##  9     9 X9.0             2006~          930 10:18      10:35
## 10    10 X8.3             2003~          486 17:03      17:25
## # ... with 40 more rows, and 2 more variables: end_time <chr>, movie <chr>
```

I get the html from the website and look for the class id “table table-striped table-responsive-md”. Then, I extract the table from it using html_table(), add column names and convert it to a tibble.

### Step 2:

```
solar_flares <- solar_flares %>%
  mutate(date_cp1 = date) %>%
  mutate(date_cp2 = date) %>%
  select(-c("movie")) %>%
  unite("start_datetime", date, start_time, sep = " ") %>%
  unite("max_datetime", date_cp1, maximum_time, sep = " ") %>%
  unite("end_datetime", date_cp2, end_time, sep = " ") %>%
  type_convert(col_types = cols(start_datetime = col_datetime(format = "%Y/%m/%d %H:%M"),
                                max_datetime = col_datetime(format = "%Y/%m/%d %H:%M"),
                                end_datetime = col_datetime(format = "%Y/%m/%d %H:%M")))
```

```
solar_flares
```

```
## # A tibble: 50 x 6
##     rank flare_classific~ start_datetime      flare_region
##    <int> <chr>            <dttm>                     <int>
##  1     1 X28.0            2003-11-04 19:29:00          486
##  2     2 X20.0            2001-04-02 21:32:00         9393
##  3     3 X17.2            2003-10-28 09:51:00          486
##  4     4 X17.0            2005-09-07 17:17:00          808
##  5     5 X14.4            2001-04-15 13:19:00         9415
##  6     6 X10.0            2003-10-29 20:37:00          486
##  7     7 X9.4             1997-11-06 11:49:00         8100
##  8     8 X9.3             2017-09-06 11:53:00         2673
##  9     9 X9.0             2006-12-05 10:18:00          930
## 10    10 X8.3             2003-11-02 17:03:00          486
## # ... with 40 more rows, and 2 more variables: max_datetime <dttm>,
## #   end_datetime <dttm>
```

I create 2 copies of the date column, remove the movie column, combine the date-start_time, date-max_time and date-end_time, and format the 3 resulting columns as datetime type.

### Step 3:

```
url_nasa <- "http://www.hcbravo.org/IntroDataSci/misc/waves_type2.html"

nasa_data <- url_nasa %>%
  read_html() %>%
  html_nodes("pre") %>%
  html_text() %>%
  str_split("\n") %>%
  purrr::as_vector() %>%
  str_subset("[0-9]{4}/[0-9]{2}/[0-9]{2}") %>%
  as_tibble() %>%
  separate(value, extra = "drop", c("start_date", "start_time", "end_date",
                                    "end_time", "start_frequency", "end_frequency",
                                    "flare_location", "flare_region", "flare_classification",
                                    "cme_date", "cme_time", "cme_angle", "cme_width",
                                    "cme_speed"), sep="[ ]{1,}")
```

```
## Warning: Calling `as_tibble()` on a vector is discouraged, because the behavior is likely to change i
## This warning is displayed once per session.
```

```
nasa_data
```

```
## # A tibble: 482 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 472 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>
```

I get the html from the website and look for the id "pre" to get the html text underneath it. I split the resulting string and convert it to a vector so that I can use subset on it. After using str_subset on the vector, I convert it to a tibble and separate it into 14 columns.

**Step 4:**

```
nasa_data <- nasa_data %>%
  mutate(start_frequency = ifelse(start_frequency == "????", NA_character_, start_frequency),
         end_frequency = ifelse(end_frequency == "????", NA_character_, end_frequency),
         flare_location = ifelse(flare_location == "------", NA_character_, flare_location),
         flare_region = ifelse(flare_region == "-----", NA_character_, flare_region),
         flare_classification = ifelse(flare_classification == "----", NA_character_,
                                       flare_classification),
         cme_date = ifelse(cme_date == "--/--", NA_character_, cme_date),
         cme_time = ifelse(cme_time == "--:--", NA_character_, cme_time),
         cme_angle = ifelse(cme_angle == "----", NA_character_, cme_angle),
         cme_width = ifelse(cme_width == "---", NA_character_, cme_width),
         cme_width = ifelse(cme_width == "----", NA_character_, cme_width),
         cme_speed = ifelse(cme_speed == "----", NA_character_, cme_speed)) %>%
  mutate(halo = ifelse(cme_angle == "Halo", TRUE, FALSE),
         cme_angle = ifelse(cme_angle == "Halo", NA_character_, cme_angle)) %>%
  mutate(cme_width = ifelse(cme_width == "360h", 360, cme_width),
         width_limit = ifelse(grepl(">", cme_width), TRUE, FALSE)) %>%
  mutate(end_time = ifelse(end_time == "24:00", "23:59", end_time)) %>%
  mutate(end_date = paste(substring(start_date, 1,5), end_date, sep = "")) %>%
  mutate(cme_date = paste(substring(start_date, 1,5), cme_date, sep = "")) %>%
  unite("start_datetime", start_date, start_time, sep = " ") %>%
  unite("end_datetime", end_date, end_time, sep = " ") %>%
  unite("cme_datetime", cme_date, cme_time, sep = " ") %>%
  type_convert(col_types = cols(start_datetime = col_datetime(format = "%Y/%m/%d %H:%M"),
                                max_datetime = col_datetime(format = "%Y/%m/%d %H:%M"),
                                end_datetime = col_datetime(format = "%Y/%m/%d %H:%M"))) %>%
  mutate(start_frequency = as.integer(start_frequency)) %>%
  mutate(end_frequency = as.integer(end_frequency)) %>%
  mutate(cme_datetime = ifelse(grepl("NA", cme_datetime), NA_character_, cme_datetime))

nasa_data
```

```
## # A tibble: 482 x 13
##    start_datetime      end_datetime        start_frequency end_frequency
##    <dttm>              <dttm>                        <int>         <int>
##  1 1997-04-01 14:00:00 1997-04-01 14:15:00            8000          4000
##  2 1997-04-07 14:30:00 1997-04-07 17:30:00           11000          1000
##  3 1997-05-12 05:15:00 1997-05-14 16:00:00           12000            80
##  4 1997-05-21 20:20:00 1997-05-21 22:00:00            5000           500
##  5 1997-09-23 21:53:00 1997-09-23 22:16:00            6000          2000
##  6 1997-11-03 05:15:00 1997-11-03 12:00:00           14000           250
##  7 1997-11-03 10:30:00 1997-11-03 11:30:00           14000          5000
##  8 1997-11-04 06:00:00 1997-11-05 04:30:00           14000           100
##  9 1997-11-06 12:20:00 1997-11-07 08:30:00           14000           100
## 10 1997-11-27 13:30:00 1997-11-27 14:00:00           14000          7000
## # ... with 472 more rows, and 9 more variables: flare_location <chr>,
## #   flare_region <chr>, flare_classification <chr>, cme_datetime <chr>,
## #   cme_angle <dbl>, cme_width <chr>, cme_speed <dbl>, halo <lgl>,
## #   width_limit <lgl>
```

I use mutate to replace the missing entries with NA, create a new column "halo" whether a flare has a halo, to change "360h" to "360" in cme_width (according to a piazza post) and to change the end_time of 24:00 to 23:59 (according to piazza post). I also combine the date-start_time, date-max_time and date-end_time, and format the 3 resulting columns as datetime type. I then convert start_frequency and end_frequency to integer columns.

## Part 2

### Question 1:

```r
nasa_data <- nasa_data %>%
  separate(flare_classification, c("flare_class", "flare_degree"), sep = 1,
           extra = "drop", remove = FALSE) %>%
  type_convert(col_types = cols(flare_degree = col_double(),
                                flare_region = col_integer()))

top50_unselected <- nasa_data %>%
  arrange(desc(flare_class), desc(flare_degree)) %>%
  slice(1:50) %>%
  tibble::rowid_to_column() %>%
  mutate(rank = rowid) %>%
  mutate(flare_classification = gsub("\\.$", ".0", flare_classification)) %>%
  separate(start_datetime, c("date", "start_time"), sep = " ", remove = FALSE) %>%
  separate(cme_datetime, c("date1", "maximum_time"), sep = " ", remove = FALSE) %>%
  separate(end_datetime, c("date2", "end_time"), sep = " ", remove = FALSE)

top50_tbl <- top50_unselected %>%
  select(c("rank", "flare_classification", "date", "flare_region",
           "start_time", "maximum_time", "end_time"))

top50_tbl
```

```
## # A tibble: 50 x 7
##     rank flare_classific~ date  flare_region start_time maximum_time
##    <int> <chr>            <chr>        <int> <chr>      <chr>
##  1     1 X28.0            2003~        10486 20:00:00   19:54
##  2     2 X20.0            2001~         9393 22:05:00   22:06
##  3     3 X17.0            2003~        10486 11:10:00   11:30
##  4     4 X14.0            2001~         9415 14:05:00   14:06
##  5     5 X10.0            2003~        10486 20:55:00   20:54
##  6     6 X9.4             1997~         8100 12:20:00   12:10
##  7     7 X9.0             2006~        10930 10:50:00   <NA>
##  8     8 X8.3             2003~        10486 17:30:00   17:30
##  9     9 X7.1             2005~        10720 07:15:00   06:54
## 10    10 X6.9             2011~        11263 08:20:00   08:12
## # ... with 40 more rows, and 1 more variable: end_time <chr>
```

No, I cannot replicate the top 50 solar flare table in SpaceWeatherLive.com exactly as they have more flare datapoints than in the NASA dataset. My code replicates it as closely as possible and even orders it in the same manner as the SpaceWeatherLive.com data table. The only limitation is the data itself that was provided to me. Also, they SpaceWeatherLive.com use maximum_time but since the NASA dataset didn't have maximum_time, I used the cme_time to approximate the maximum_time.

### Question 2:

### Section 1

```r
char_similarity <- function(v1, v2) {
  if (is.na(v1) || is.na(v2)) {
    return(0)
  }
  else {
    ifelse(v1 == v2, 1, 0)
  }
}

num_similarity <- function(v1, v2) {
  if (is.na(v1) || is.na(v2)) {
    return(0)
  }
  else {
    exp(-1*((v1 - v2)^2))
  }
}

date_similarity <- function(v1, v2) {
  if (is.na(v1) || is.na(v2)) {
    return(0)
  }
  else {
    exp(-1*(((as.numeric(v1) - as.numeric(v2))/3600)^2))
  }
}

solar_flares_unselected <- solar_flares %>%
  separate(flare_classification, c("flare_class", "flare_degree"), sep = 1,
           extra = "drop") %>%
  type_convert(col_types = cols(flare_degree = col_double()))

flare_similarity <- function(df1, df2) {
  score <- num_similarity(df1$flare_degree, df2$flare_degree) +
    date_similarity(df1$start_datetime, df2$start_datetime) +
    date_similarity(df1$end_datetime, df2$end_datetime) +
    num_similarity(df1$flare_region, df2$flare_region) +
    char_similarity(df1$flare_class, df2$flare_class)

  score
}
```

```
flare_similarity(solar_flares_unselected, top50_unselected)
```

```
##  [1] 2.765716 3.738968 2.137435 1.000123 1.000000 1.697676 1.852144
##  [8] 1.367879 1.027052 1.140858 1.055576 1.444858 1.236928 1.444858
## [15] 1.527292 1.444858 1.527292 1.527292 1.140858 1.105399 1.077305
## [22] 1.055576 1.105399 1.105399 1.105399 1.527292 1.527292 1.298197
## [29] 1.298197 1.367879 1.367879 1.367879 1.367879 1.444858 1.298197
## [36] 1.367879 1.367879 1.298197 1.298197 1.367879 1.367879 1.444858
## [43] 1.527292 1.527292 1.527292 1.612626 1.527292 1.444858 1.527292
## [50] 1.527292
```

I define my similarity function using 2 aux functions: char_similarity, num_similarity and date_similarity. They all calculate their respective similarities. I use these functions to calculate the similarities of flare_degree, start_datetime, end_datetime, flare_region and flare_class, and then add them all up to get the final similarity score.

### Section 2

```
flare_match <- function(df1, df2) {
  matches <- c(0)

  for (i in seq(1, nrow(df1))) {
    max <- 0
    maxid <- 0

    for (j in seq(1, nrow(df2))) {
      ele <- flare_similarity(df1[i,], df2[j,])

      if (ele > max) {
        maxid <- j
        max <- ele
      }
    }

    matches[i] <- ifelse(max > 2, maxid, NA_character_)
  }

  as.integer(matches)
}
```

For my flare_match function, I disregard any below or equal to 2 as I would like the flare_class to match up perfectly (amounting to a score of 1) and the flare_degree to match somewhat perfectly (amounting to a floor value of 1.8). I also want the datetimes to match up with a few hours of each other, so the bare minimum valid score should be above 2.

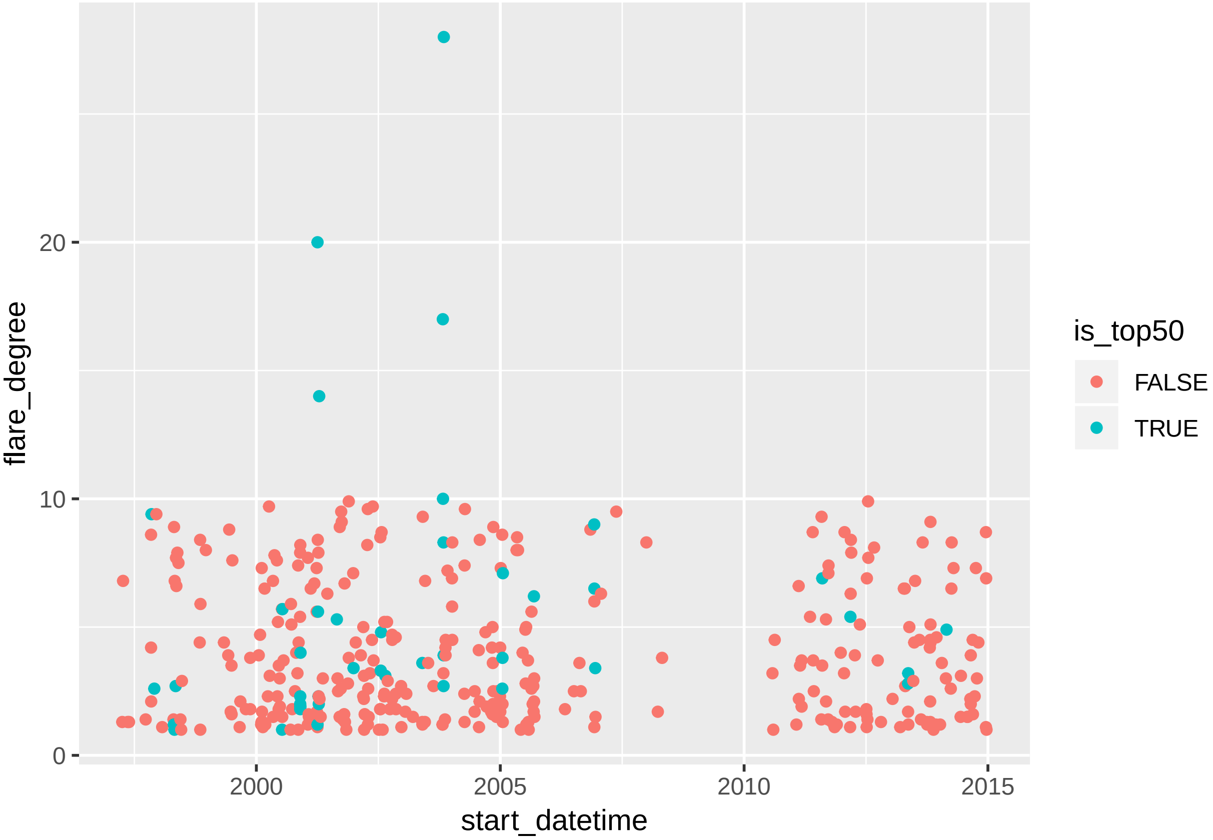
### Section 3

```
top50_tbl <- top50_tbl %>%
  mutate(best_match_index = flare_match(solar_flares_unselected, top50_unselected))
```

**Question 3:**

```
nasa_data <- nasa_data %>%
  mutate(is_top50 = ifelse(is.na(flare_match(nasa_data, solar_flares_unselected)), FALSE, TRUE))

ggplot(nasa_data, aes(x=start_datetime, y=flare_degree, colour = is_top50)) +
  geom_point()
```



Intention: Is there covariance between intensity (flare_degree) & coronal mass ejection (CME) speed in solar flares?

It is clear that most solar flares tend to stay around an intensity of around 10 and the rest are in SpaceWeatherLive's top 50. The other place where top 50 solar flares are distinct from non-top-50 flares is on the higher end of coronal mass ejection speed.

A positive correlation between flare intensity and CME speed is observed. While high coronal mass ejection speed does not necessarily imply high flare intensity, it does imply top 50 ranking.