# Project 1: Data Scraping and Cleaning

*Joseph Kahadze*

*March 14, 2019*

## PART 1

### STEP 1

I store the spaceweatherlive url in 'url' and then I use read_html() and html_nodes() to extract the data from the website. Then, I convert the nodes into a table using html_table() and set the column names.

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

weather <- url %>%
  read_html() %>%
  html_node("table") %>%
  html_table() %>%
  magrittr::set_colnames(c("rank", "flare_classification", "date",
                           "flare_region", "start_time", "maximum_time",
                           "end_time", "movie")) %>%
  as_tibble()

weather
```

```
## # 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>
```

### STEP 2

First, I combine the times and dates using unite(). Then I remove the the '_' between them and convert all occurrences of time 24:00 to 23:59 using gsub(). Finally, I use type_convert() to convert the newly created date/time attribute into type dttm.

```
weather <- weather %>%
  select(-movie) %>%
  unite(start_datetime, date, start_time, remove = FALSE) %>%
  unite(max_datetime, date, maximum_time, remove = FALSE) %>%
```

```r
  unite(end_datetime, date, end_time, remove = TRUE) %>%
  select(rank, flare_classification, start_datetime, max_datetime, end_datetime,
         flare_region)


weather$start_datetime <- gsub("_", " ", weather$start_datetime)
weather$start_datetime <- gsub("24:00", "23:59", weather$start_datetime)
weather$max_datetime <- gsub("_", " ", weather$max_datetime)
weather$max_datetime <- gsub("24:00", "23:59", weather$max_datetime)
weather$end_datetime <- gsub("_", " ", weather$end_datetime)
weather$end_datetime <- gsub("24:00", "23:59", weather$end_datetime)

weather <- weather %>%
  type_convert(col_types = cols(start_datetime =
                                col_datetime(format = "%Y/%m/%d %H:%M"))) %>%
  type_convert(col_types = cols(max_datetime =
                                col_datetime(format = "%Y/%m/%d %H:%M"))) %>%
  type_convert(col_types = cols(end_datetime =
                                col_datetime(format = "%Y/%m/%d %H:%M")))

weather
```

```
## # A tibble: 50 x 6
##     rank flare_classific~ start_datetime      max_datetime
##    <int> <chr>            <dttm>              <dttm>
##  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
##  7     7 X9.4             1997-11-06 11:49:00 1997-11-06 11:55:00
##  8     8 X9.3             2017-09-06 11:53:00 2017-09-06 12:02:00
##  9     9 X9.0             2006-12-05 10:18:00 2006-12-05 10:35:00
## 10    10 X8.3             2003-11-02 17:03:00 2003-11-02 17:25:00
## # ... with 40 more rows, and 2 more variables: end_datetime <dttm>,
## #   flare_region <int>
```

**STEP 3**

I do the same thing I did for the spaceweatherlive data set to the NASA data set. I scrape the data from the HTML using read_html() and html_node(). Since the NASA html isn't as clean as the spaceweatherlive html and contains extraneous data, I have to manually select the parts I want to include in my data set. Finally I separate the single column into many column with column names.

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

nasa_node <- nasa_url %>%
  read_html() %>%
  html_node("pre") %>%
  html_text() %>%
  str_split("\n")
```

```r
nasa_node <- nasa_node[[1]][16:(length(nasa_node[[1]])) - 3] %>%
  as_tibble() %>%
  separate(value, 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"), "\\s+",
                     extra = "drop")

nasa_node
```

```
## # 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**

Step 4 included cleaning up the data by converting the missing values to NA and converting the time and date attributes into a single datetime object.

Here I ran into the issue of entities with missing CME date/time and end date/time. Once I converted these values to NA, I could no longer convert the these values into dttm without causing warnings and it was said on Piazza that our code should not produce warnings.

After thinking about the issue, I decided to label these data points as "corrupt" and remove them from the NASA data set. I believe given the context of what we're trying to achieve, this was sufficient solution. We spend the second half of this project creating similarity/match functions to weed out the flares with inaccurate data, so I think it's logical to remove the flares with missing end date considering that's one of the 4 attributes I use in my similarity function.

Furthermore, I posted a question regarding this on Piazza and Professor Bravo confirmed it was okay to just remove the flares with missing end/cme date and time.

Otherwise, everything I do is very similar to what I did in Step 2. I convert the missing data to NA. I separate the all of the instances of 'halo' in cme_angle into a new boolean column using within(). And I combine the start/end/cme dates/times into single columns using unite() and then convert them to type dttm using type_convert(). Since the end/cme dates were missing the year, I had to take the corresponding year for each flare from the start date and concatenate it onto the end/cme date to use type_convert() on it.

```r
nasa_node <- nasa_node %>%
  na_if("????") %>%
```

```r
  na_if("-----") %>%
  na_if("----") %>%
  na_if("--/--") %>%
  na_if("--:--")

nasa_node$halo <- FALSE

nasa_node <- within(nasa_node, halo[cme_angle == 'Halo'] <- TRUE) %>%
  na_if("Halo")

nasa_node$cme_width_limit <- FALSE

nasa_node <- within(nasa_node,
                    cme_width_limit[grepl(">", cme_width, fixed = TRUE)] <- TRUE)

nasa_node$cme_width <- sub(">","", nasa_node$cme_width)

nasa_node <- drop_na(nasa_node, end_date)
nasa_node <- drop_na(nasa_node, cme_date)

nasa_node <- nasa_node %>%
  unite(start_datetime, start_date, start_time, remove = TRUE) %>%
  unite(end_datetime_pre, end_date, end_time, remove = TRUE) %>%
  unite(cme_datetime_pre, cme_date, cme_time, remove = TRUE)

nasa_node$years <- str_extract_all(nasa_node$start_datetime, "[0-9]{4}",
                                   simplify = TRUE)

nasa_node <- nasa_node %>%
  unite(end_datetime, end_datetime_pre, years, remove = FALSE) %>%
  unite(cme_datetime, cme_datetime_pre, years, remove = FALSE) %>%
  select(1, 2, 4:9, 11:15)

#occurences of 24:00 changed to 23:59

nasa_node$start_datetime <- gsub("_", " ", nasa_node$start_datetime)
nasa_node$start_datetime <- gsub("24:00", "23:59", nasa_node$start_datetime)
nasa_node$cme_datetime <- gsub("_", " ", nasa_node$cme_datetime)
nasa_node$cme_datetime <- gsub("24:00", "23:59", nasa_node$cme_datetime)
nasa_node$end_datetime <- gsub("_", " ", nasa_node$end_datetime)
nasa_node$end_datetime <- gsub("24:00", "23:59", nasa_node$end_datetime)


nasa_node <- nasa_node %>%
  type_convert(col_types = cols(start_datetime =
                                col_datetime(format = "%Y/%m/%d %H:%M"))) %>%
  type_convert(col_types = cols(cme_datetime =
                                col_datetime(format = "%m/%d %H:%M %Y"))) %>%
  type_convert(col_types = cols(end_datetime =
                                col_datetime(format = "%m/%d %H:%M %Y")))

nasa_node
```

```
## # A tibble: 489 x 13
##    start_datetime      end_datetime        start_frequency end_frequency
##    <dttm>              <dttm>                        <dbl>         <dbl>
##  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 479 more rows, and 9 more variables: flare_location <chr>,
## #   flare_region <chr>, flare_classification <chr>, cme_datetime <dttm>,
## #   cme_angle <dbl>, cme_width <chr>, cme_speed <dbl>, halo <lgl>,
## #   cme_width_limit <lgl>
```

# PART 2

### QUESTION 1

I was able to replicate the spaceweatherlive data in the NASA data set by extracting all of the flares with the highest flare classification (X) and then removing the value from the classification. Once I had only the X classification flares, I arranged them by the numerical flare value I extracted from the flare_classification. Finally, I slice the top 50 entities from the data set to match the 50 flares in the spaceweather data set.

While I was able to structurally "match" the NASA data set to the spaceweatherlive data set, from a glance it was obvious that these data sets had some differences. While they both contained the start date/time, end date/time, flare classification, and flare region, the nasa data set also included the start/end frequency, flare location, cme date/time (spaceweatherlive had the max date/time which isn't exactly the same), cme width, cme angle, cme width limit, as well as the columns created above, which the spaceweatherlive data set did not include.

After the analysis in Question 2, we see that only 29 of the top 50 solar flares in the NASA data set had a sufficient match in the spaceweatherlive data set.

```r
nasa_top50 <- nasa_node
nasa_top50 <- nasa_top50[grep("X", nasa_top50$flare_classification), ]

# extracts the numerical value from the flare classification in nasa_node
# for easier analysis
temp_flare_class <- nasa_top50$flare_classification

nasa_top50 <- nasa_top50 %>%
  separate(flare_classification, c("space", "class_num"), "X", extra = "drop")

nasa_top50$flare_class <- temp_flare_class

nasa_top50 <- nasa_top50 %>%
  type_convert(col_types = cols(class_num = col_double())) %>%
  arrange(desc(class_num)) %>%
  select(-space) %>%
  slice(1:50)
```

```
## Warning: package 'bindrcpp' was built under R version 3.5.2
```

```r
nasa_top50
```

```
## # A tibble: 50 x 14
##    start_datetime      end_datetime        start_frequency end_frequency
##    <dttm>              <dttm>                        <dbl>         <dbl>
##  1 2003-11-04 20:00:00 2003-11-04 23:59:00           10000           200
##  2 2001-04-02 22:05:00 2001-04-03 02:30:00           14000           250
##  3 2003-10-28 11:10:00 2003-10-29 23:59:00           14000            40
##  4 2001-04-15 14:05:00 2001-04-16 13:00:00           14000            40
##  5 2003-10-29 20:55:00 2003-10-29 23:59:00           11000           500
##  6 1997-11-06 12:20:00 1997-11-07 08:30:00           14000           100
##  7 2003-11-02 17:30:00 2003-11-03 01:00:00           12000           250
##  8 2005-01-20 07:15:00 2005-01-20 16:30:00           14000            25
##  9 2011-08-09 08:20:00 2011-08-09 08:35:00           16000          4000
## 10 2005-09-09 19:45:00 2005-09-09 22:00:00           10000            50
## # ... with 40 more rows, and 10 more variables: flare_location <chr>,
## #   flare_region <dbl>, class_num <dbl>, cme_datetime <dttm>,
## #   cme_angle <dbl>, cme_width <dbl>, cme_speed <dbl>, halo <lgl>,
## #   cme_width_limit <lgl>, flare_class <chr>
```

**QUESTION 2**

For my similarity function I used 4 attributes I found most important to determine the similarity between two solar flares.

- Start Date
- End Date
- Flare Classification
- Flare Region

I found all of these attributes equally important to determining flare similarity thus they all have a weight of 1. In order for two flares to be considered a "match" they need to at least have a similarity score of 3, or have 3 out of these 4 attributes to be matching.

I did not use data such as the exact time because a lot of flares didn't even have the same date, so matching for time between different days seemed pointless.

```r
# here I'm cleaning up the data in weather to make it easier for analysis
temp_weather_class <- weather$flare_classification

weather <- weather %>%
  separate(flare_classification, c("space", "class_num"), "X")

weather$flare_class <- temp_weather_class

weather <- weather %>%
  type_convert(col_types = cols(class_num = col_double())) %>%
  arrange(desc(class_num)) %>%
  select(-space)
```

```r
# convert flare region to int in nasa_top50
nasa_top50 <- nasa_top50 %>%
  type_convert(col_types = cols(flare_region = col_integer()))

for (num in 1:length(nasa_top50$flare_region)) {
  if (nasa_top50$flare_region[num] > 10000){
    nasa_top50$flare_region[num] <- nasa_top50$flare_region[num] - 10000
  }
}

weather <- weather %>%
  type_convert(col_types = cols(flare_region = col_integer()))



# flare similarity function
flare_similarity <- function(f1, f2){
  sum <- 0

  # compare flare start dates
  f1_sdate <- as.Date(f1$start_datetime, format = "%m/%d/%Y")
  f2_sdate <- as.Date(f2$start_datetime, format = "%m/%d/%Y")
  if (f1_sdate == f2_sdate){
    sum <- sum + 1
  }

  # compare flare start dates
  f1_edate <- as.Date(f1$end_datetime, format = "%m/%d/%Y")
  f2_edate <- as.Date(f2$end_datetime, format = "%m/%d/%Y")
  if (f1_edate == f2_edate){
    sum <- sum + 1
  }

  # compare flare classes
  f1_class <- f1$class_num
  f2_class <- f2$class_num
  if (f1_class == f2_class){
    sum <- sum + 1
  }

  # compare flare region
  f1_region <- f1$flare_region
  f2_region <- f2$flare_region
  if (f1_region == f2_region){
    sum <- sum + 1
  }

  sum
}

# flare match function
flare_match <- function(flare, table){
    sim_table <- data.frame(matrix(nrow=50, ncol=1))
```

```r
    max_index <- 0
    max_value <- 0
    colnames(sim_table) <- "values"

    for (x in 1:nrow(table)){
      sim_table[x,1] <- flare_similarity(flare, slice(table, x))
      if (sim_table[x,1] > max_value & sim_table[x, 1] >= 3){
        max_index <- x
        max_value <- sim_table[x,1]
      }
    }

    if (max_value == 0){
      max_index <- NA
    }

    max_index
}

# add new column with matching indicies

nasa_top50$match_index <- NA

for (x in 1:nrow(nasa_top50)){
  match <- flare_match(slice(nasa_top50, x), weather)
  nasa_top50$match_index[x] <- match
}

# drop all of the flares without a match
nasa_top50 <- nasa_top50[!is.na(nasa_top50$match_index), ]

nasa_top50
```

```
## # A tibble: 29 x 15
##    start_datetime      end_datetime        start_frequency end_frequency
##    <dttm>              <dttm>                        <dbl>         <dbl>
##  1 2003-11-04 20:00:00 2003-11-04 23:59:00           10000           200
##  2 2001-04-02 22:05:00 2001-04-03 02:30:00           14000           250
##  3 2003-10-29 20:55:00 2003-10-29 23:59:00           11000           500
##  4 1997-11-06 12:20:00 1997-11-07 08:30:00           14000           100
##  5 2003-11-02 17:30:00 2003-11-03 01:00:00           12000           250
##  6 2005-01-20 07:15:00 2005-01-20 16:30:00           14000            25
##  7 2011-08-09 08:20:00 2011-08-09 08:35:00           16000          4000
##  8 2005-09-09 19:45:00 2005-09-09 22:00:00           10000            50
##  9 2000-07-14 10:30:00 2000-07-15 14:30:00           14000            80
## 10 2001-04-06 19:35:00 2001-04-07 01:50:00           14000           230
## # ... with 19 more rows, and 11 more variables: flare_location <chr>,
## #   flare_region <dbl>, class_num <dbl>, cme_datetime <dttm>,
## #   cme_angle <dbl>, cme_width <dbl>, cme_speed <dbl>, halo <lgl>,
## #   cme_width_limit <lgl>, flare_class <chr>, match_index <int>
```

**QUESTION 3**

My intent with the plot was to show the percentage of total solar flares that were "top 50" each year. My hypothesis is that the percentage will stay relatively stable over the years with little variance.

I think my plot contextualizes the "top 50" NASA solar flare data well by showing what percentage of the total flares were in the top 50. The actual number of solar flares considered in the "top 50" are 29 because the rest didn't have a sufficient match using the flare_match and flare_similarity function to match the NASA flares with the ones in the SpaceWeatherLive data set.

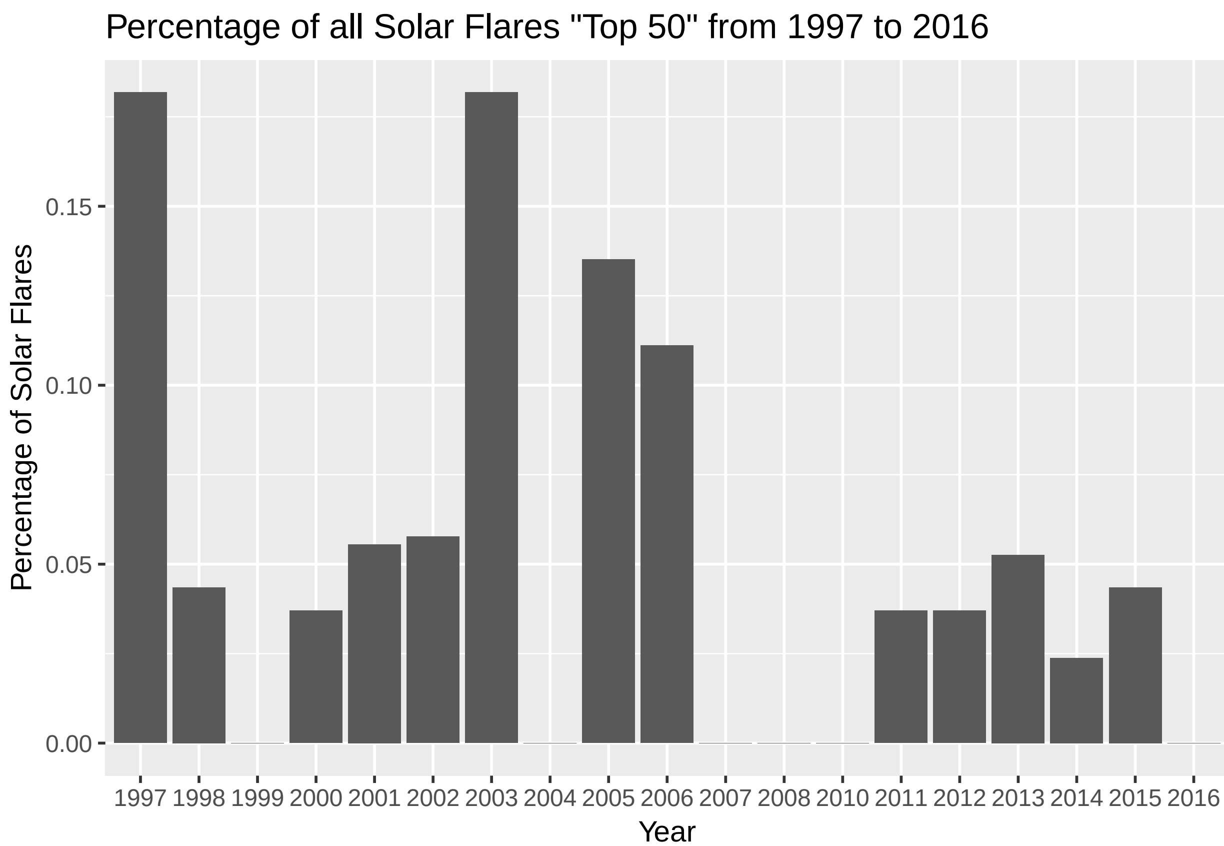
I created this bar plot by first grouping the total NASA flare data by number of occurrences per year. Then doing the same for the "top 50" data and using a full join to combine the data into one table. Then I created a new column, "perc" which equaled the # of top 50 flares / # of total flares per year. Finally, I plotted the data using a geom_bar with the x axis representing years and the y axis representing percentage.

My analysis of the plot is that there appears to be a high variance between the percentages of top 50 solar flares by year. For example in 2003 it appears that approximately 18% of all the solar flares were in the top 50. The next year, 2004, not a single one of the solar flares was in the top. This analysis suggests that the number of solar flares within a given year does not correlate directly with the number of intense solar flares per year, as the percentage of solar flares which can be considered "top", varies drastically year to year, disproving my initial hypothesis.

I think it would be interesting to see this type of breakdown on different time scales, to see if the variance hold day to day or decade to decade (if there's enough data).

```r
# add a new column containing just the year to the nasa tables
nasa_top50$year <- format(as.Date(nasa_top50$start_datetime,
                                  format="%Y/%m/%d %h/%m/%s"),"%Y")
nasa_node$year <- format(as.Date(nasa_node$start_datetime,
                                 format="%Y/%m/%d %h/%m/%s"),"%Y")

# create table with number of top 50 solar flares per year
top_p <- nasa_top50 %>%
  group_by(year) %>%
  summarize(class_num = n())

# create table with total number of solar flares per year
nasa_p <- nasa_node %>%
  group_by(year) %>%
  summarize(class_num = n())

# combine two tables
total_p <- nasa_p %>%
  full_join(top_p, by="year")

# add column with percentage of solar flares that are top 50 by year
total_p[is.na(total_p)] <- 0
colnames(total_p)[2] <- "num_all"
colnames(total_p)[3] <- "num_top"
total_p <- transform(total_p, perc = num_top / num_all)

# graph the data
total_p %>%
  ggplot(mapping=aes(y=perc, x=year)) +
  geom_bar(stat="identity") +
```

```r
  ggtitle("Percentage of all Solar Flares \"Top 50\" from 1997 to 2016") +
  labs(y= "Percentage of Solar Flares", x = "Year")
```




Finally, I clean up the 3 main data sets by removing the extraneous columns I added for analysis purposes.

```r
weather <- weather %>%
  select(-class_num)

nasa_node <- nasa_node %>%
  select(-year)

nasa_top50 <- nasa_top50 %>%
  select(1:6, 8:15)
```