Analyzing Twitter data regarding anxiety: The role of inter-temporal effects

Author: Ellen Brock

Date: 6 September 2014

In what follows, I will investigate some Twitter related health care data obtained from Saama Technologies and this project is organized by Coursolve and has the following link: https://www.coursolve.org/need/184 . We were given raw data of over 6 million tweets spread over about 896 csv files. Initially, we had to write a one-page proposal what problem we wanted to solve with these data. The data for each of these CSV files are related to one "hashtag" related to one particular disease and are obtained through Topsy. While some people wanted to work on the entire dataset of over 6 million records, I choose to work on only one CSV file/disease since I am new to Twitter and for sure writing Python using the Twitter API to retrieve data.

This code below uses ANOVA analysis which is a statistical technique to study the differences in group means across certain factors (in my case month, day of the week and hour of the day are factors) . In other words, this technique will allow me to *investigate whether the sentiment score differs across months, days of the week or hour of the day*. Here, I am using one-way ANOVA but this can be extended later on two two-way ANOVA, etc.

The raw data of the CSV file consisted of tweets related to anxiety but very few features were given. In total there were only about 7. Amongst these are: the time in UTC of when the tweet was posted (this is different than the local time), the internet link to the tweet, a score related to the influence (Topsy score), the author name and the content/text of the tweet. There is further no information on the location of the author of the tweet, which time zone the author of the tweet is located, the hash tags, etc.

As a first step, others on the project forum started to discuss to extend these data and add a lot of these additional fields. One person who drove this effort and wrote excellent code was George Fisher who posted his code on Github and the code all of us were allowed to use by George and the course organizers. The link to Georges code is here: https://github.com/grfiv/healthcare_twitter_analysis

I could have blindly written the code but my aim was to understand this great piece of work first and learn from it such as accessing the Twitter API, how to work with dictionaries and how to export everything back into a CSV file and how to change it for the purpose of my research question here (and that process is currently ongoing). As a first step, I went over the CSV code of George (see https://github.com/grfiv/healthcare_twitter_analysis/blob/master/code/create_bulkfile.py) and simplified this focusing on processing only one file (anxiety here) in contrast to George's

code who processes many files at the same time. I also commented out a lot of this code (for my own understanding) and I created a lot of print statements. The result of this work can be found here on my Github account: https://github.com/EllenB

This document is structured as follows. First, I do some exploraty analysis and also do some data clean up as there are “extra” years in the original data. In this exploratory analysis below, I also extract extra variables related to the time of the tweet such as year, month and time of the day. Especially, time of the day I explore a little further and determine the next steps for the final research question. I also discuss some of the anomalies I see in the data (mainly between user location and time zone) Subsequently, I discuss the statistical analysis using ANOVA and present the results.

Exploratory data analysis/clean up

As mentioned above, I ran and simplified the code of George Fisher (CSV version) for my own purpose. Since I ran everything on a (small) laptop, with only 4 GB of RAM and an internet that konks off regularly, I had split the initial raw/CSV file of 211 895 observations. I splitted up this dataset into smaller files of 13 500 records each on which I ran the “adapted” code of George. Next, I had to merge the data back. I do so using the following R code after setting the working directory:

```
setwd("E:/Ellen/coursera/DataScience2/project/Finished_Files")
getwd()
```

```
## [1] "E:/Ellen/coursera/DataScience2/project/Finished_Files"
```

```
filenames <- list.files()
filenames
```

```
##  [1] "bigtweet_file_108001_1.csv"
"bigtweet_file_108001_2.csv"
##  [3] "bigtweet_file_11.csv"
"bigtweet_file_12.csv"
##  [5] "bigtweet_file_121501.csv"
"bigtweet_file_135001_1.csv"
##  [7] "bigtweet_file_135001_2.csv"
"bigtweet_file_13501.csv"
##  [9] "bigtweet_file_148501.csv"
"bigtweet_file_162001_1.csv"
## [11] "bigtweet_file_162001_2.csv"
"bigtweet_file_175501.csv"
## [13] "bigtweet_file_189001_1.csv"
"bigtweet_file_189001_2.csv"
## [15] "bigtweet_file_202501.csv"
"bigtweet_file_27001_1.csv"
## [17] "bigtweet_file_27001_2.csv"
"bigtweet_file_40501.csv"
## [19] "bigtweet_file_54001_1.csv"
"bigtweet_file_54001_2.csv"
## [21] "bigtweet_file_67501.csv"
"bigtweet_file_81001.csv"
## [23] "bigtweet_file_94501.csv"
```

```
Anxiety = do.call("rbind", lapply(filenames, read.csv, header
= TRUE, stringsAsFactors = FALSE))
```

Subsequently, I write this into a CSV file:

```
write.csv(Anxiety, file =
"E:/Ellen/coursera/DataScience2/project/Anxiety_all.csv",
    row.names = FALSE)
```

First, I look at some summary statistics:

```
nrow(Anxiety)
```

```
## [1] 208388
```

```
# summary(Anxiety) # I commented this here in order to
suppress the output.
# str(Anxiety) # Commented this to supress the output
names(Anxiety)
```

```
##  [1] "user_lang"
"user_description"
##  [3] "hashtags"
"user_time_zone_placename"
##  [5] "user_friends_count"
"user_utc_offset"
##  [7] "id"
"sentiment"
##  [9] "user_screen_name"
"user_location"
## [11] "user_favourites_count"
"user_verified"
## [13] "content"
"score"
## [15] "tweet_favorite_count"
"user_name"
## [17] "tweet_retweet_count"
"user_followers_count"
## [19] "user_time_zone_coordinates"
"user_mentions"
## [21] "trackback_author_url"
"user_location_coordinates"
## [23] "tweet_geo"
"firstpost_date"
## [25] "user_listed_count"
"url"
## [27] "created_at"
"tweet_retweeted"
## [29] "user_geo_enabled"
"user_time_zone"
## [31] "tweet_favorited"
"user_location_placename"
## [33] "trackback_author_nick"
"tweet_place"
## [35] "trackback_permalink"
"tweet_coordinates"
## [37] "tweet_in_reply_to_screen_name"
"user_statuses_count"
## [39] "tweet_timestamp"
```

From the above, we can observe that the dataset contains 208388 observations and has 39 features. One of the first things I did was try to play/construct the data for the dates and put this into a format R can work with such that I can extract which month, day of the week and hour one tweet is related to. Upon inspection (see the code below on how to do that),I found that there were data from other years and I decided to drop them. Moreover, the data for June are very little so I decided to drop these for the time being. First I needed to convert the character format in which the dates are into a date formar R can work with. As an example, the format looks something like this: format: Sat Mar 22 15:41:14 +0000 2014. First, I strip the “+0000” as follows:

```
Anxiety$created_at_new = sub("+0000 ", "", Anxiety$created_at)
## One can still see the '+' sign and this needs to be removed
too:
Anxiety$created_at_new = gsub("[+]", "",
Anxiety$created_at_new)
```

Now the date looks something like this e.g.: Sat Mar 22 15:41:14 2014. R needs to understand that this is a date. See e.g.: http://en.wikibooks.org/wiki/R_Programming/Times_and_Dates for more information.

```
Anxiety$UTC_Tweet = as.POSIXct(Anxiety$created_at_new, format
= "%a %b %d %H:%M:%S %Y")
```

Subsequently, I try to understand the UTC offset variable. The idea of this work is to look at the sentiment of the tweet according to the time of the day. All the raw dara are expressed in Coordinated Universal Time (UTC). In order to obtain the local time of when the sender of the tweet has sent the tweet, I need the investigate the “local time” of when the tweet was sent sent all the tweets were expressed in one common time, the UTC time. In order to convert into the local time, I need to investigate the status of the “user_utc_offset” variable.

I try to run the summary command, I see that this is expressed as a character still. Running the summary statistics, we see that the variable is character and we need to convert it into a numeric variable first:

```
class(Anxiety$user_utc_offset)
```

```
## [1] "character"
```

```
Anxiety$user_utc_offset = as.numeric(Anxiety$user_utc_offset)
```

```
## Warning: NAs introduced by coercion
```

```
summary(Anxiety$user_utc_offset)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  -39600  -21600  -21600  -15300  -14400   46800   55049
```

There are 55049 missing observations in this dataset.I make a subset of these observations where this variable is missing and then see how many data are there for the user location in case I want to impute this variable later on.

```
Anxiety_NA_UTC = subset(Anxiety, is.na
(Anxiety$user_utc_offset))
v = which(Anxiety_NA_UTC$user_location == "")
length(v)
```

```
## [1] 14228
```

```
head(v)
```

```
## [1]  8  9 10 15 19 21
```

Subsequently, I make a frequency table of the different user_locations that are given in this dataset with the missing offset variables. Later on, these location data will be linked to the MapQuest API in order to obtain the coordinates and the UTC offset variables. This will be done using the code of George Fisher (2004). I write this dataset into a CSV file that can later be used for linking it to the MapQuest API:

```
Location_Table = table(Anxiety_NA_UTC$user_location)
# Location_Table # To print this to the screen uncomment this
line
write.csv(Location_Table, file =
"E:/Ellen/coursera/DataScience2/project/Location_Table.csv",
    row.names = FALSE)
```

In my understanding during a Google Hangout, the code of George selects randomly one location in case there are multiple locations for a city place. In his paper, Fisher (2014) gives the example of Pasadena which consists of 47 locations ranging from Pasadena in Texas and Pasadena in California as one of these locations. My suggestion is to use UTC offset in case it is available while using this algorithm.

In later steps, I also want to understand the UTC offset variable and the user location variable a little better. When looking at the data here (which just used the algorithm without running the MapQuest algorithm and I explain just the raw twitter data), I find a few anomalies. To give an example (and there are several examples), there are several observations that have as user location Boulder Colorada but the UTC offset or time zone gives the Arizone time zone (UTC -25 200) instead of Mountain Standard time zone (UTC -21 600) in quite a number of cases.

Subsequently, I construct separate variables for year, month, day of the week and hour using the variable “created_at” which refers to the UTC time.

```
DD = strptime(Anxiety$created_at_new, format = "%a %b %d %H:%
M:%S %Y")

Anxiety$month = DD$mon + 1
table(Anxiety$month)
```

```
##
##     1     2     3     4     5     6     7     8     9    10
11    12
## 30999 39440 53134 48344 36159   260     3     8     6    15
5    15
```

From the above, I see that there are other months in the dataset although the dataset was supposed to be data from Jan-July 2014. Further, I observe data from the earlier years.

```
Anxiety$year = DD$year + 1900
table(Anxiety$year)
```

```
##
##   2011   2013   2014
##      1     66 208321
```

There are data from the year 2011 and 2013 which I removed from the dataset:

```
v = which(Anxiety$year == 2011 | Anxiety$year == 2013)
# v
length(v)  # 67
```

```
## [1] 67
```

```
nrow(Anxiety)
```

```
## [1] 208388
```

```
Anxiety = Anxiety[-v, ]
nrow(Anxiety)
```

```
## [1] 208321
```

```
table(Anxiety$month)
```

```
##
##     1     2     3     4     5     6
## 30999 39436 53134 48341 36157   254
```

Subsequently, I also compute the day of the week:

```
DD = strptime(Anxiety$created_at_new, format = "%a %b %d %H:%
M:%S %Y")
Anxiety$DayOfWeek = weekdays(DD)
table(Anxiety$DayOfWeek)
```

```
##
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday
Wednesday
##     29586     31822     26762     27007     31460     30797
30887
```

After all this clean-up I am left with 208321 observations. Looking at the number of observations during the month of June, I observe only 254 observations. Since this is a low number in contrast to the other months, I have decided to drop the data of this months for the time being as well which leaves me with 208 067 observations:

```
v = which(Anxiety$month == 6)
# v
length(v)
```

```
## [1] 254
```

```
Anxiety2 = Anxiety[-v, ]
nrow(Anxiety2)
```

```
## [1] 208067
```

Methodology

In what follows, I will not work with the variable “hour” as for the question at hand, I would need the local time of when the tweet which was sent based on the location/time zone of the sender of the tweet (see above). Next, I try to answer whether the sentiment score differs across months and day of the week for anxiety related tweets. The methodology I will follow for this is Analysis Of Variance (ANOVA).

First, I look at the type of the “month” and “day of the week” variables:

```
class(Anxiety2$month)
```

```
## [1] "numeric"
```

```
class(Anxiety2$DayOfWeek)
```

```
## [1] "character"
```

```
table(Anxiety2$DayOfWeek)
```

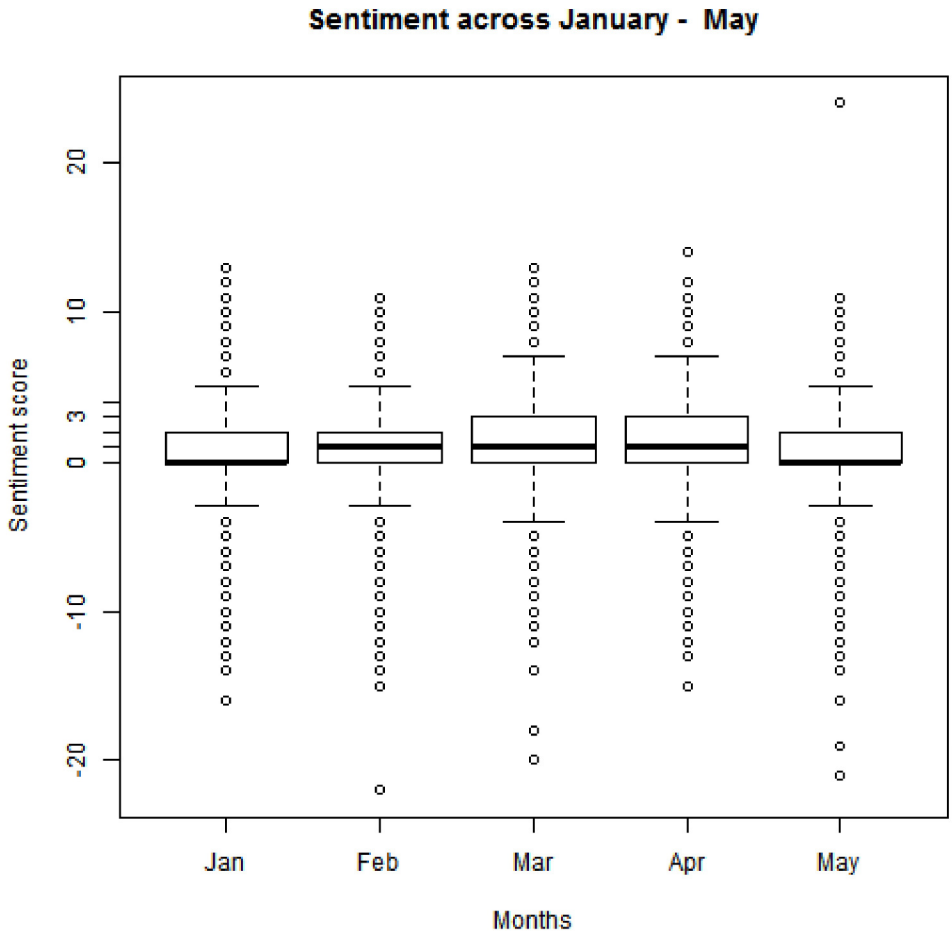
```
##
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday
Wednesday
##     29586     31822     26762     26753     31460     30797
30887
```

First, I convert these variables into factor variables:

```
Anxiety2$month = factor(Anxiety2$month, labels = c("Jan",
"Feb", "Mar", "Apr",
    "May"))
Anxiety2$DayOfWeek = factor(Anxiety2$DayOfWeek)
```

Before moving on to the ANOVA analysis, I create some box plots. First, the code for the boxplot for sentiment across months:

```
boxplot(sentiment ~ month, data = Anxiety2, main = "Sentiment
across January -  May",
    xlab = "Months", ylab = "Sentiment score")
axis(2, at = seq(0, 4, 1))
```



The box plots reveal very little difference across the months but somehow it is clear that March and April have more positive sentiment than the other months. Subsequently, I also compute the average tweet sentiment per month:

```
tapply(Anxiety2$sentiment, Anxiety2$month, mean)
```

```
##     Jan     Feb     Mar     Apr     May
## 0.5731 0.8336 1.0639 1.0520 0.4911
```

The results show that the sentiment score is highest in March then followed by April in comparison to the other months.

Next, I perform an ANOVA analysis (I assume normality for the time being):

```
anov1 = aov(Anxiety2$sentiment ~ Anxiety2$month)
summary(anov1)
```

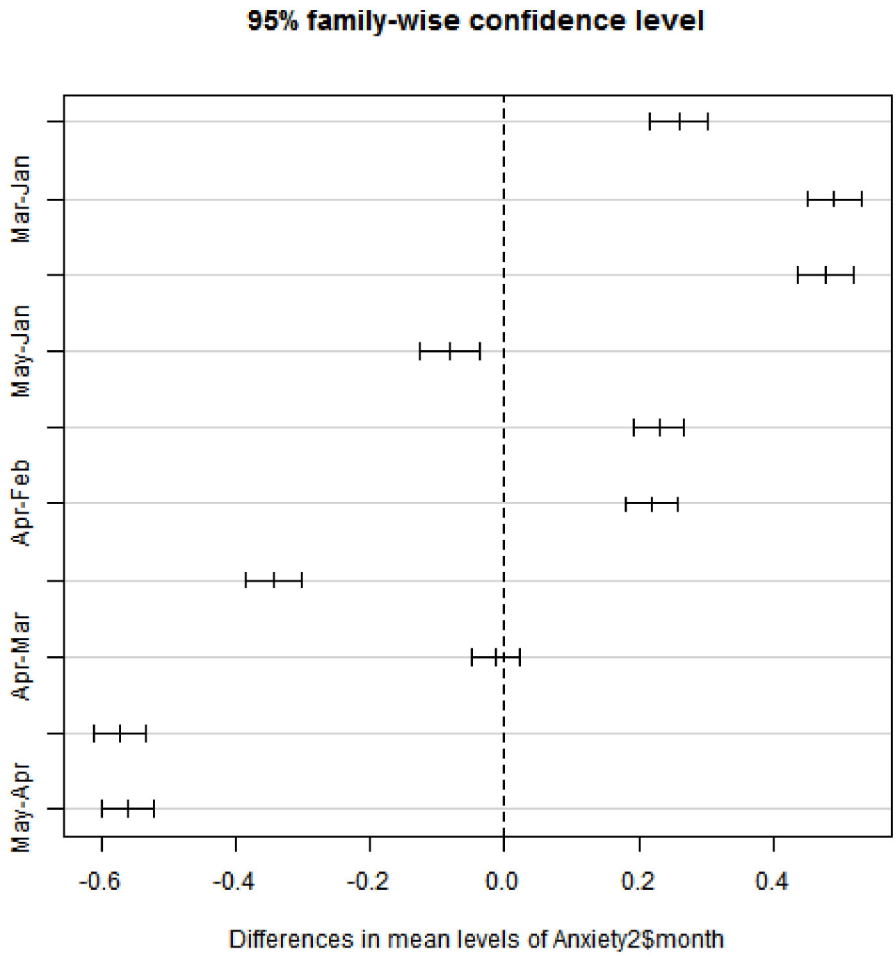
```
##                      Df Sum Sq Mean Sq F value Pr(>F)
## Anxiety2$month        4  11442    2861     659 <2e-16 ***
## Residuals        208062 903600       4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
```

The null hypothesis is that there is no difference in the mean of the sentiment score across months. The results reveal (see Figure 3) that there is a difference between sentiment score across months. This test does not tell where the difference lies in terms of which month is different from another month. In order to answer this question, one needs to conduct post hoc tests:

```
TukeyHSD(anov1)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Anxiety2$sentiment ~ Anxiety2$month)
##
## $`Anxiety2$month`
##              diff     lwr      upr  p adj
## Feb-Jan  0.26046  0.2173  0.30361 0.0000
## Mar-Jan  0.49074  0.4501  0.53137 0.0000
## Apr-Jan  0.47887  0.4375  0.52023 0.0000
## May-Jan -0.08206 -0.1261 -0.03806 0.0000
## Mar-Feb  0.23028  0.1925  0.26806 0.0000
## Apr-Feb  0.21841  0.1798  0.25698 0.0000
## May-Feb -0.34253 -0.3839 -0.30114 0.0000
## Apr-Mar -0.01187 -0.0476  0.02386 0.8946
## May-Mar -0.57280 -0.6116 -0.53405 0.0000
## May-Apr -0.56093 -0.6005 -0.52141 0.0000
```

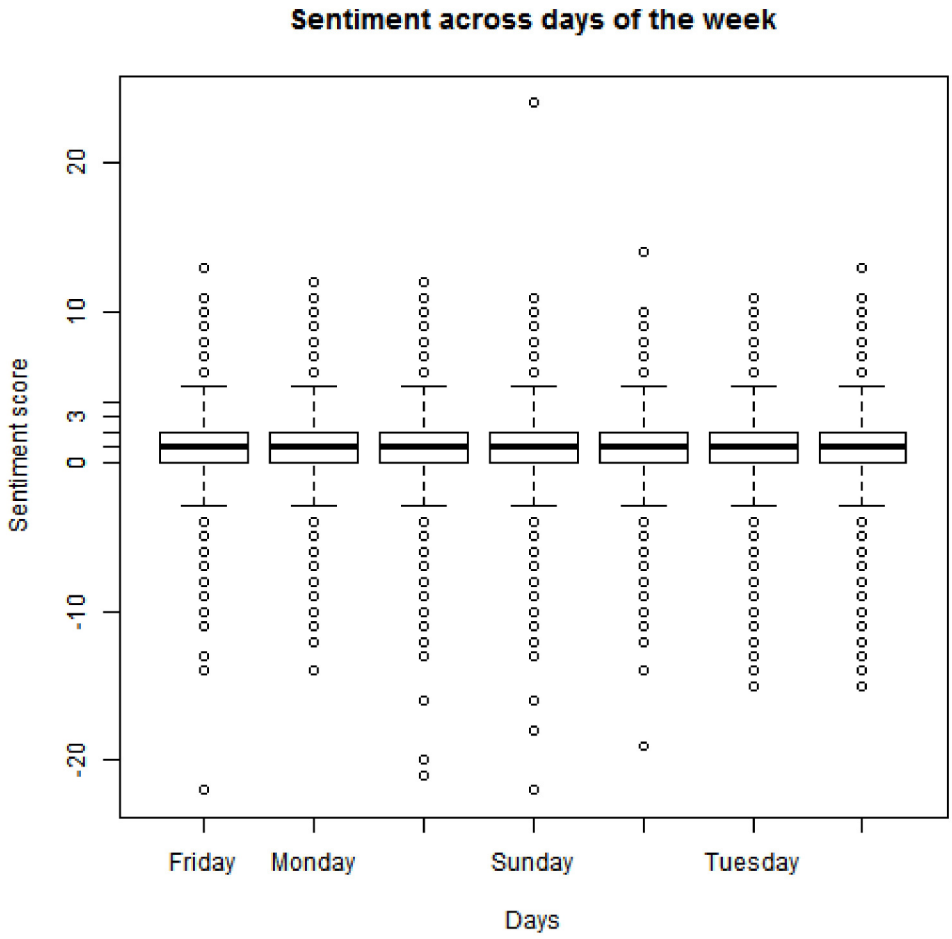
```
plot(TukeyHSD(anov1))
```



The results reveal that there are significant differences between all the months except between the months April and March.

Next, I proceed with the ANOVA code for the days of the week and first create a box plot:

```
boxplot(sentiment ~ DayOfWeek, data = Anxiety2, main =
"Sentiment across days of the week",
    xlab = "Days", ylab = "Sentiment score")
axis(2, at = seq(0, 4, 1))
```



From the figure, not a single difference across the days is observed (the bar in the rectangle is the median, not the mean, which seems to be equal). Then, I ran the code to compute the mean across days of the week:

```
tapply(Anxiety2$sentiment, Anxiety2$DayOfWeek, mean)
```

```
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday
Wednesday
##    0.8391    0.8114    0.9610    0.9740    0.7908    0.7772
0.7945
```

It can be seen that during Saturdays and Sundays the sentiment score is higher compared to the other days of the week.

Next, I perform the ANOVA analysis:

```
anov2 = aov(Anxiety2$sentiment ~ Anxiety2$DayOfWeek)
summary(anov2)
```

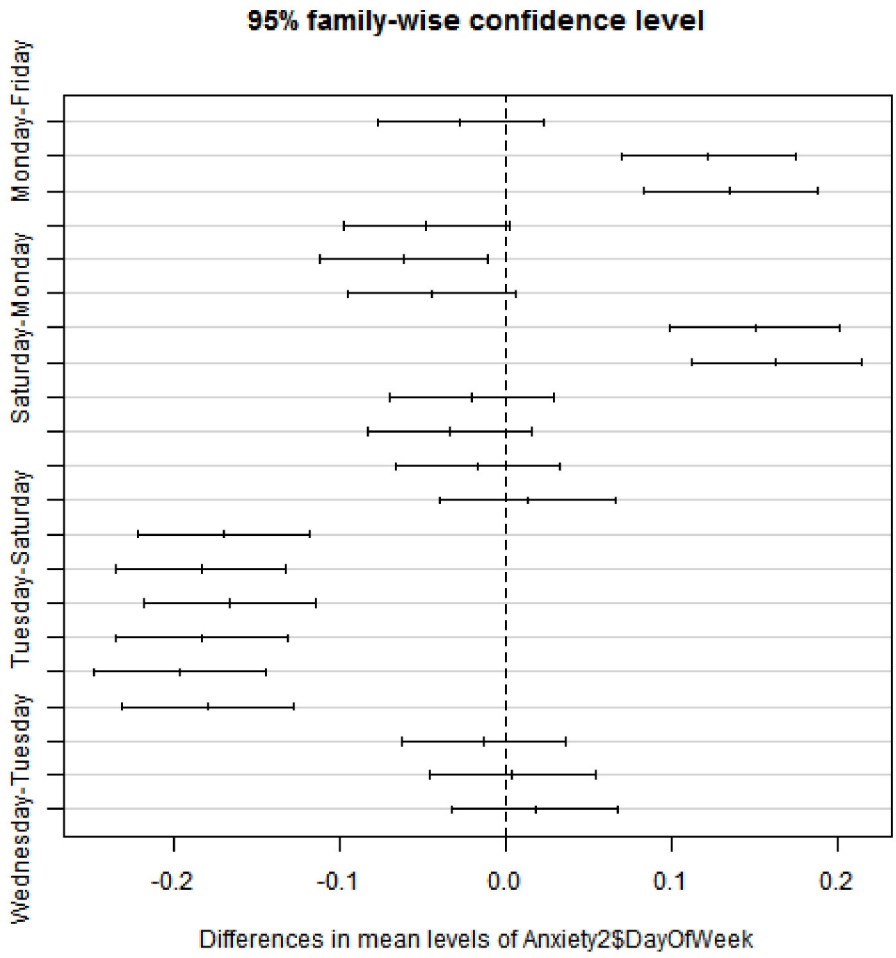
```
##                          Df Sum Sq Mean Sq F value Pr(>F)
## Anxiety2$DayOfWeek        6   1155   192.5    43.8 <2e-16 ***
## Residuals            208060 913887     4.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
```

This statistic only says that there is a difference between the different days of the week. This does not say between which days there is a difference. In order to see between which days of the week there is a difference, I run the following code for the post hoc tests:

```
TukeyHSD(anov2)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Anxiety2$sentiment ~ Anxiety2$DayOfWeek)
##
## $`Anxiety2$DayOfWeek`
##                           diff      lwr       upr  p adj
## Monday-Friday       -0.027727 -0.07763  0.022177 0.6574
## Saturday-Friday      0.121843  0.06972  0.173970 0.0000
## Sunday-Friday        0.134837  0.08271  0.186969 0.0000
## Thursday-Friday     -0.048397 -0.09844  0.001645 0.0657
## Tuesday-Friday      -0.061961 -0.11226 -0.011659 0.0052
## Wednesday-Friday    -0.044638 -0.09490  0.005628 0.1204
## Saturday-Monday      0.149570  0.09832  0.200820 0.0000
## Sunday-Monday        0.162564  0.11131  0.213819 0.0000
## Thursday-Monday     -0.020670 -0.06980  0.028458 0.8784
## Tuesday-Monday      -0.034234 -0.08363  0.015159 0.3871
## Wednesday-Monday    -0.016911 -0.06627  0.032445 0.9519
## Sunday-Saturday      0.012995 -0.04043  0.066417 0.9916
## Thursday-Saturday   -0.170239 -0.22162 -0.118855 0.0000
## Tuesday-Saturday    -0.183803 -0.23544 -0.132165 0.0000
## Wednesday-Saturday  -0.166480 -0.21808 -0.114877 0.0000
## Thursday-Sunday     -0.183234 -0.23462 -0.131845 0.0000
## Tuesday-Sunday      -0.196798 -0.24844 -0.145155 0.0000
## Wednesday-Sunday    -0.179475 -0.23108 -0.127868 0.0000
## Tuesday-Thursday    -0.013564 -0.06310  0.035968 0.9843
## Wednesday-Thursday   0.003759 -0.04574  0.053255 1.0000
## Wednesday-Tuesday    0.017323 -0.03244  0.067082 0.9481
```

```
plot(TukeyHSD(anov2))
```



These test reveals that a) there is not really a difference in sentiment between the weekdays (Mon-Fri), b) there is a difference between any weekday and a weekend and c) there is not really a difference in sentiment between Saturday and Sunday.

Conclusion/summary

In this note I have studied how the sentiment score differs across months and days of the week for all tweets related to anxiety. I find that there are significant differences between all the months except between the months April and March. Also, people seem to be happier in weekends than in weekdays and there is no difference in happiness between any weekday but there is a difference between any weekday and any day of the weekend.