The dataset used is a compressed snippet from the larger dataset found here: https://www.kaggle.com/datasets/kazanova/sentiment140 which contains over 1.6 million tweets extracted from the Twitter API

Inspecting the dataset

The size of the file is 73 MB. Retrieving the file size was done using the commands:

```
ls -1h FIT1043_Dataset.gz
73
```

The ls is used to list the file in directory whilst the -1h actually displays the file size.

The delimiter used to separate the columns in the file are commas. To actually deduce this I used gunzip to decompress the file. Then I used head, which outputs the first 10 lines from the file by default. From this output, I identified that a comma was separating the comments. Please see the below commands and the first two lines from the file:

```
 gunzip FIT1043_Dataset.gz

head FIT1043
```

```
0,1467810672,Mon Apr 06 22:19:49 PDT
2009,NO_QUERY,scotthamilton,is upset that he can't
update his Facebook by texting it... and might cry as
a result  School today also. Blah!

0,1467810917,Mon Apr 06 22:19:53 PDT
2009,NO_QUERY,mattycus,@Kenichan I dived many times
```

```
for the ball. Managed to save 50%  The rest go out of
bounds
```

It is evident that a comma is used to separate the columns.

There are 1471793 lines in the dataset. Retrieving this information was done using the following command:

```
wc -l FIT1043_Dataset.
1471793
```

wc is used to count characters, numbers, lines while the -l specifies to count only lines.

Important Information from the data

There are 1471235 users.

Here is the code used:

```
awk -F, '{print $5}' FIT1043_Dataset | uniq | wc -l
1471235
```

awk command is used to process this action which involves printing the fifth column of the dataset. This is because the 5th column contains all usernames in FIT1043_Dataset.

Reading the man pages of Uniq, it mentions that repeated lines will not be removed if they are not adjacent to each other, thus for all duplicates to be removed we need to sort them. This means the current answer is not correct. Sort is used to sort usernames in order to ensure uniq properly removes all duplicates. Then wc -l is applied to thus count number of usernames filtered by removing duplicates. Thus, here is the new command containing sort

```
awk -F, '{print $5}' FIT1043_Dataset | sort| uniq |
wc -l
626684
```

This returns 626684 users and this is the correct answer

The date range for the Twitter posts are Monday April 6 2009 – Tuesday June 16 2009. Finding these dates was achieved with head and tail commands. Here are the following commands with their explanations:

```
head -n 1 FIT1043_Dataset | awk -F',' '{print $3}'
Mon Apr 06 22:19:49 PDT 2009
```

Head selects first 10 lines of file, but -n 1 ensures only first line printed, then awk used to print the value in 3rd column of first line which is date using -F',' to separate delimiter, and print $3 to access and output this date

```
Last date -> tail -n 1 FIT1043_Dataset | awk -F',' '{print
$3}'
Tue Jun 16 08:40:50 PDT 2009
```

Similar to head command, instead tail used since it prints last 10 lines

Data Aggregation

Here I will be using Linux commands to gather positive, neutral and negative tweets regarding America and Canada so I can compare sentiment of both countries. Measuring this will be done by creating two csv files one for American sentiment and one for Canadian and then using R to create a histogram of the data in these files.

The total number of tweets that mentioned the USA was 5690. Here is the command used to find this:

```
grep -i "USA" FIT1043_Dataset | wc -l
    5690 = total tweets which mention USA
```

To gain specific sentiment counts I replaced the wc -l by rather re-directing all lines that mention USA into a text file called USA.txt. The command is below:

```
grep -I “USA” FIT1043_Dataset > USA.txt
```

Now getting specific sentiments was achieved with awk command where ‘$1 ==0’ tells terminal to find count of USA sentiment that is negative. Here are following commands:

```
awk -F, '$1 == 0' USA.txt | wc -l
    2574
```

There are 2574 negative tweets mentioning USA

```
awk -F, '$1 == 2' USA.txt | wc -l
       0
```

There are 0 neutral tweets mentioning USA

```
awk -F, '$1 == 4' USA.txt | wc -l
    3116
```

There are 3116 positive tweets mentioning USA

Finding the Canada negative neutral and positive tweets was done in same way as USA.

```
awk -F, '$1 == 0' Canada.txt | wc -l
     698
```

There are 698 negative tweets mentioning Canada

```
awk -F, '$1 == 2' Canada.txt | wc -l
       0
```

There are 0 neutral tweets mentioning Canada

```
awk -F, '$1 == 4' Canada.txt | wc -l
     541
```

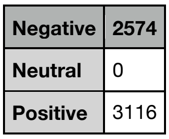
There are 541 positive tweets mentioning Canada

Storing this sentiment data into csv files was done using the echo command which displays output and then using > to direct it to specific csv file. Additionally, I manually added in the output retrieved from previous commands. Here is what all commands looked like:

```
echo "Negative, 2574" > sentiment-USA.csv

echo “Neutral, 0” > sentiment-USA.csv

echo “Positive, 3116” > sentiment-USA.csv

echo “Negative, 698” > sentiment-canada.csv

echo “Neutral, 0” > sentiment-canada.csv
echo “Positive, 541” > sentiment-canada.csv
```

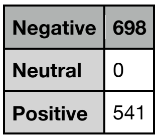
Here is what csv files look like:

USA:



Negative	2574
Neutral	0
Positive	3116

Canada:



Negative	698
Neutral	0
Positive	541

Here is the code used to read both files in R:

# Reading the csv file, checking that no header column and thus setting custom column names to the sentiment and the count of this sentiment using R vector

```
> USA_sentiment <- read.csv('sentiment-USA.csv',
header=FALSE, col.names = c("Sentiment", "Count"))

> Canada_sentiment <- read.csv('sentiment-
canada.csv', header=FALSE, col.names = c("Sentiment",
"Count"))
```

Plotting the side-by-side bar chart was completed by first combining the datasets into a matrix which would then be inputted in the plot so all data is present

```
    sentiment_matrix <- matrix(c(USA_sentiment$Count,
    Canada_sentiment$Count), nrow = 3, ncol = 2,
    byrow = FALSE)
```

Next, I assigned column and row names to this matrix

```
    >rownames(sentiment_matrix) <-
    Canada_sentiment$Sentiment

    > colnames(sentiment_matrix) <- c("USA",
    "Canada")
```

Then I created a vector containing colours that will be assigned to Negative, Neutral, Positive tweets where colours represent sentiments in that order

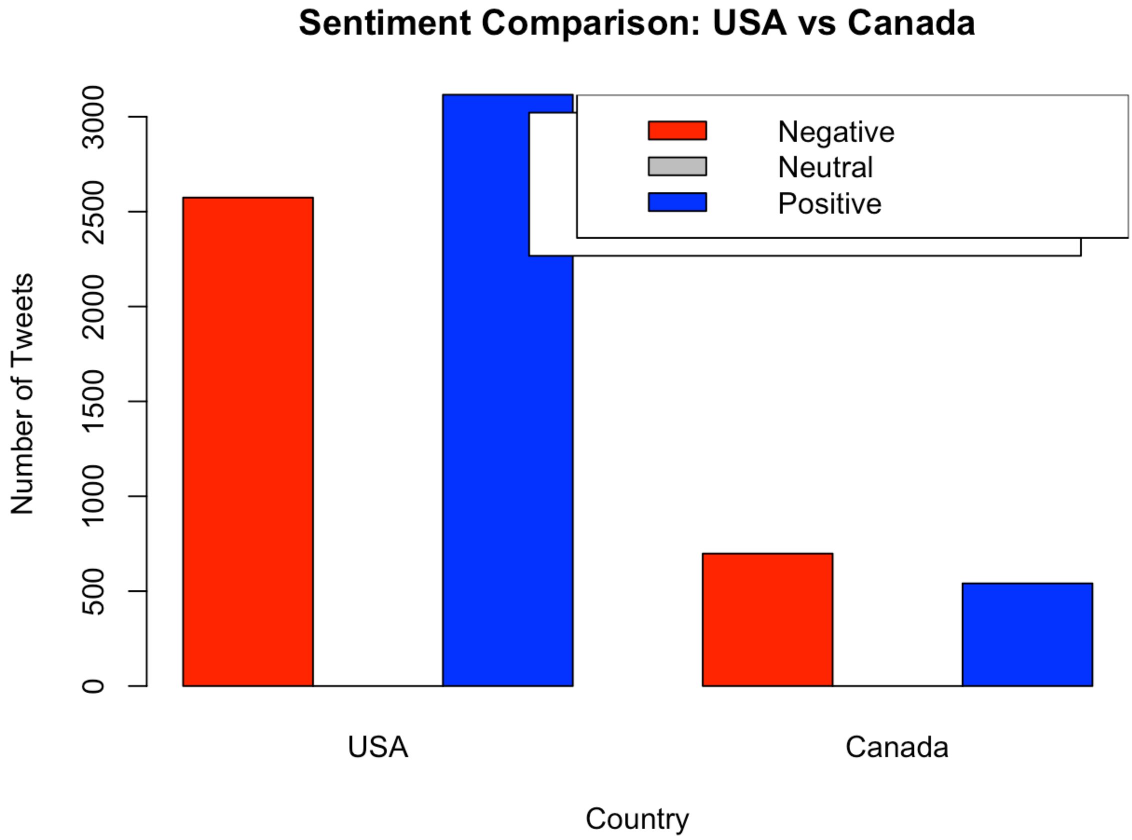
```
    colors <- c("red", "grey", "blue")
```

Finally, barplot() was used to plot the chart and legend was added for easy identification

```
    barplot(sentiment_matrix, beside=TRUE,
    +         col=colors,
    +         legend=rownames(sentiment_matrix),
    +          main="Sentiment Comparison: USA vs
    Canada",
    +          xlab="Country", ylab="Number of
    Tweets")


    legend("topright",
    legend=rownames(sentiment_matrix), fill=colors)
```

Below is the graph:

NOTE: Copy to clipboard was used



<u>**Analysis**</u>

Overall, it is clear that there are more positive tweets about the USA compared to Canada. In fact, positive sentiment in USA outweighs negative, whereas in Canada there is more negative sentiment. Thus, we can contend that US citizens are more patriotic than their northern

neighbours. Interestingly, both datasets contain no neutral tweets indicating a high degree of polarity in opinions regarding these countries. While USA is more positive, it's overall sentiment distribution is imbalanced relative to Canada. Indeed, the plots affirm this considering there is a 500 tweet difference with USA's negative and positive counts which is greater than what is approximately 100-200 tweet count difference with Canada. Additionally, the total volume of tweets is significantly larger for the USA relative to Canada which suggests that sentiment about the US is a more popular topic to tweet about on Twitter. This could be attributed to the USA having a much larger population which corresponds to more online engagement.

Tweets about Australia

Here, I have challenged myself to gather the amount of tweets that mention Australia from the API to detect patterns that detail the frequency of how much Australia is mentioned each day throughout the time period.

To extract timestamps and save them into aus_time.txt following command was used:

```
awk -F',' 'BEGIN{IGNORECASE=1} /Australia/ {print
$3}' FIT1043_Dataset > aus_time.txt
```

The awk 'F',' means that awk will treat the file as a csv and split each line into fields based on commas meaning $1 is first field(column) etc. BEGIN{IGNORECASE=1} ensures that awk always performs case insensitive matching meaning /Australia/ pattern will tell awk to match lines containing the string 'Australia' in any case. The {print $3} ensures that awk prints the third field which contains the timestamps whilst the > will direct these timestamps to the text file aus_time.txt.

1. Reading the textfile was done with the following code:

```
timestamps <- readLines("aus_time.txt")
```

This function reads the text data line by line and returns each line of the file as a character vector

When trying to do strptime() initially, errors popped up because R did not support parsing the timezone, denoted by “%Z”. Thus, I removed this feature from the timestamps with the following code:

```
timestamps <- gsub(" PDT", "", timestamps)
```

This removes the timezone, in this case always “PDT” to an empty string.

Next, I created a format filter that is used when parsing the date_time.

```
filter_string = "%a %b %d %H:%M:%S %Y"
```

NOTE: %a = shorthand for day, %b = shorthand for month, %d = actual numerical date of month, %H:%M:%S = hours, minutes and seconds and %Y = year

This means I intended to parse the data in the following format:

Day month numerical_date hours:minutes:seconds Year

To parse the format in the R file, following function was called:

```
date_time <- strptime(timestamps, format =
filter_string)
```

This actually converts text values into recognisable timestamps.

Here is an example of output:

```
"2009-04-06 23:49:29 AEST"
```

What is noticeable is the absence of shorthand weekday. This is attributed to the fact that the timezone is removed which affects what day is recognised when tweet was sent out. For example, a tweet from someone in Canada on a Monday could have been a Tuesday for someone in Australia. This inconsistency is thus removed.

Number of tweets for each day was calculated with following piece of code:

```
tweet_count <- table(as.Date(date_time))
```

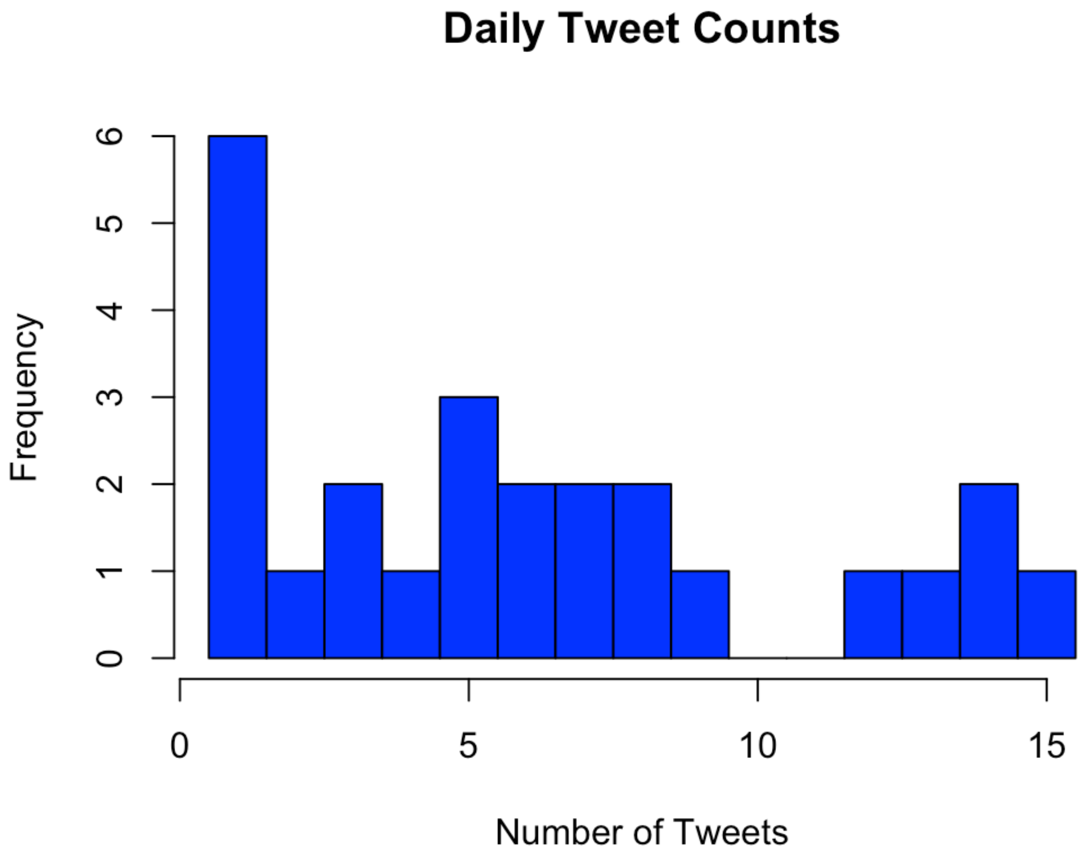
Indeed, table() creates a frequency table of what's inputted and as.Date(date_time) converts date_time into date objects. This means the code is counting the number of times there was a tweet mentioning Australia on that specific date. It excludes other information from the timestamp.

Creating the histogram was done with the following code:

```
hist(as.numeric(tweet_count),
+      main="Daily Tweet Counts",
+      xlab="Number of Tweets",
+      ylab="Frequency",
+      breaks=seq(0.5, max(tweet_count)+0.5, by=1),
+      col="blue")
```

as.numeric converts table into numeric vector so hist() can work, and breaks start at 0.5 to ensure each frequency gets its own bar. Other pieces of code are done for purely labelling and visualisation purposes.

Here is the histogram:



This histogram visualises the frequency of tweets mentioning Australia, and the amount of days containing these frequencies from the date range of April 6 – June 16. For example, there are 6 days in this date range where on each day there was 1 tweet that mentioned Australia. This distribution is clearly positively skewed. Indeed, the tail of tweets on the right side of the histogram is longer than the left side, with most tweets clustering towards left side of the x-axis indicating it was more common for a few tweets to mention Australia as compared to a higher number. Indeed, the mode frequency was 1 tweet a day with 6 days fulfilling this. The number of tweets ranged from 0 – 15 tweets a day. Nevertheless, there is a discrepancy with the fact there are no days with exactly 10 or 11 tweets, which gives presence to clear outliers with the small cluster of days with 12-15 tweets a day. Overall, relative to the USA and Canada there are few tweets that mention Australia in this time range which could be attributed to its smaller population which might mean less online presence on a social platform like Twitter.