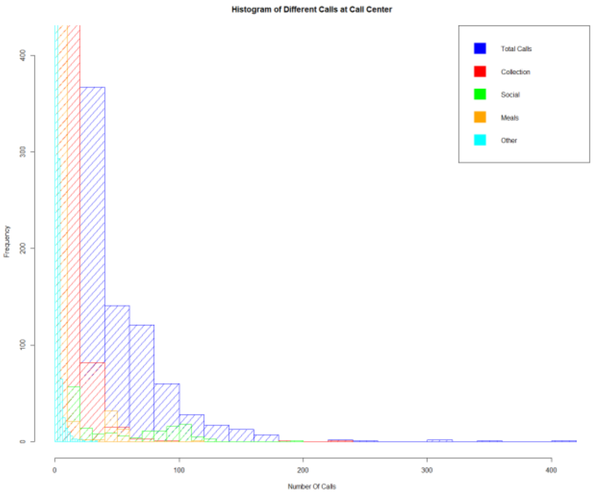
```
# Name: Swapnil Ukey
# Module Code: 2223-MS5108 Applied Customer Analytics
# Student Id: 22220959
```

Question1

For this, I got the data from Link.

Total calls made to local authorities and the number of calls made for different types of services such as collection and delivery, social, meals, other, otherrequest, callback, forum meetings, etc.



Histogram:

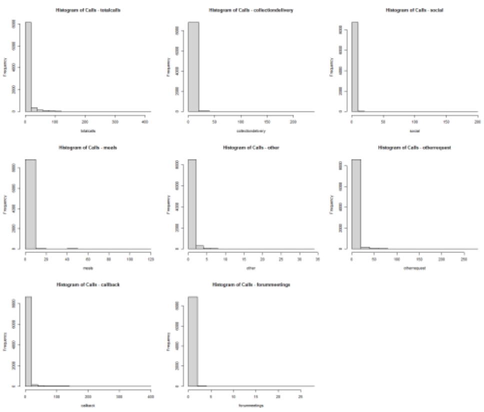
The hist() method of the r programming language was used to generate this histogram. This includes details on various services that are offered and contains numerical statistics. The total number of calls received by Ireland's local authority is indicated by the color blue. Red, green, orange, and blue are the colors for collecting, social gatherings, meals, and other things, respectively.

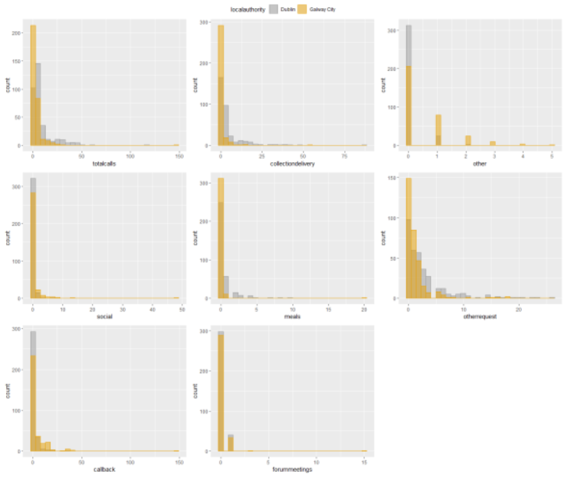
The histogram shows that the mean for each column is close to zero. Moreover, the columns with the greatest total call deviation. Moreover, the same pattern is maintained in other columns.

I also plotted all the histograms using a loop to comprehend each column better.

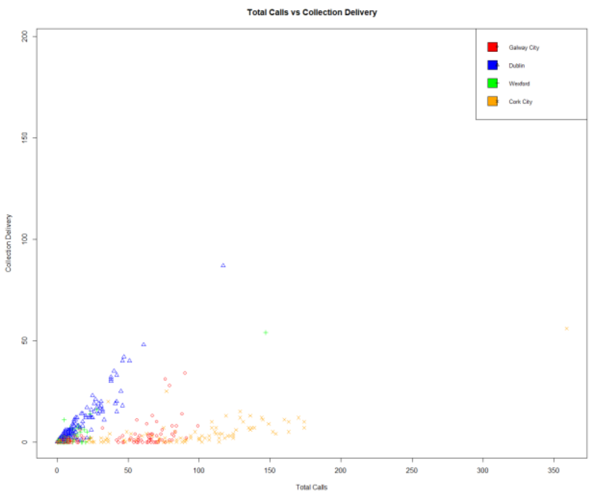
As we can see, the standard deviation for total calls is higher than for other columns, and the mean for all the columns is close to zero.

Also, I examined each call service column in accordance with Dublin's and Galway City's respective local authorities. As you can see, Dublin is the silver hue, while Galway City is the brown color.



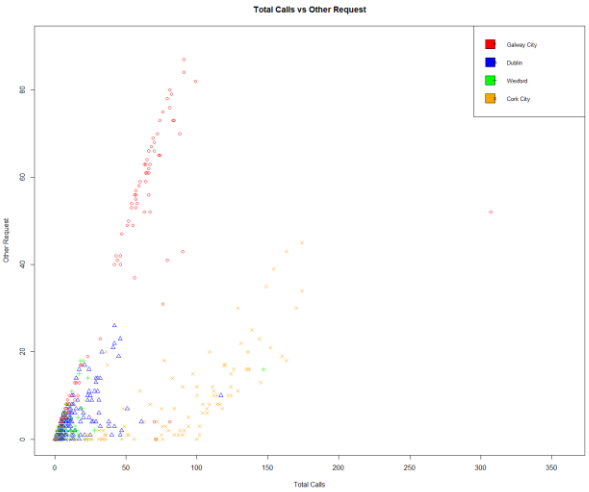


To do this, I used the ggplot() library to help two distinct local authorities understand one another better. As we can see, Galway City perceived more calls overall than Dublin City, yet Dublin City demonstrated a higher deviation than Galway.



Scatterplot:

Galway, Cork, Dublin, and Wexford are the four cities I sought to include in the scatter plot. Different shapes and colors were assigned to each city in order to better comprehend each city. In our initial trial, we attempted to generate using the r language's built-in plot() method; afterwards, we used ggplot for better comprehension.

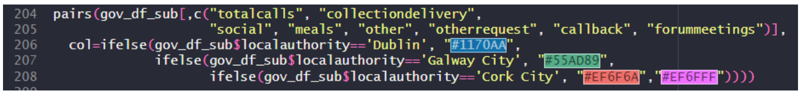


In total I have created 3 scatter plot as follows

1. Total Calls vs Collection Delivery
2. Total Calls vs Social
3. Total Calls vs Other Requests

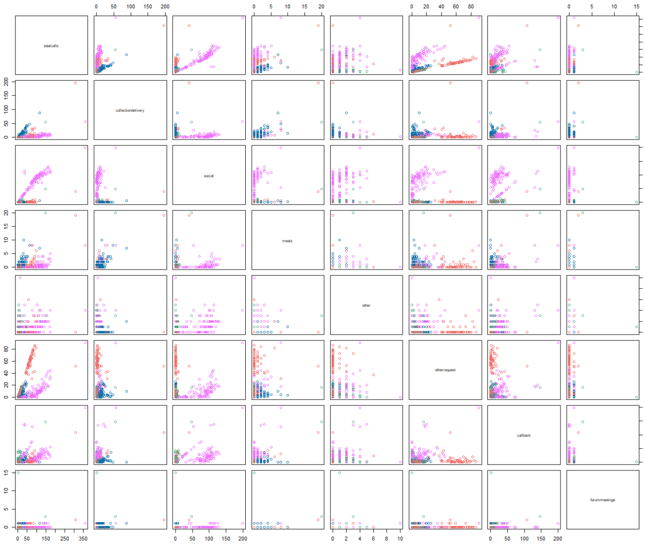
This allows us to examine how different call types are more prevalent in different cities when compared to total calls. For example, when comparing total calls to other requests, we can see that other request call types were more prevalent in Galway than other cities. We can also observe a trend in Dublin when comparing total calls to collection delivery.

Along with that, I also made a pairs() plot across all the numeric columns and assigned various colors to each of the cities I was looking at.In addition, I made a pairs() plot among all the number columns and, along with that, I gave each of the cities I was looking at a distinct color.



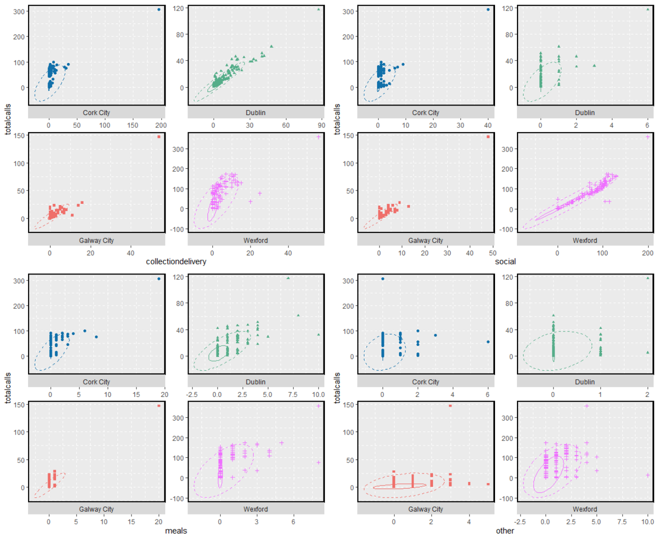
```
pairs(gov_df_sub[,c("totalcalls", "collectiondelivery",
                    "social", "meals", "other", "otherrequest", "callback", "forummeetings")],
  col=ifelse(gov_df_sub$localauthority=='Dublin', "#1170AA",
             ifelse(gov_df_sub$localauthority=='Galway City', "#55AD89",
                    ifelse(gov_df_sub$localauthority=='Cork City', "#EF6F6A","#EF6FFF"))))
```

pairs()



Now, as we can see, pairs() methods offer a matrix of scatter plots, where each dot represents a single point made up entirely of numerical data. With this, we can comprehend the relevance of the types of calls that were made and the cities where they were made.

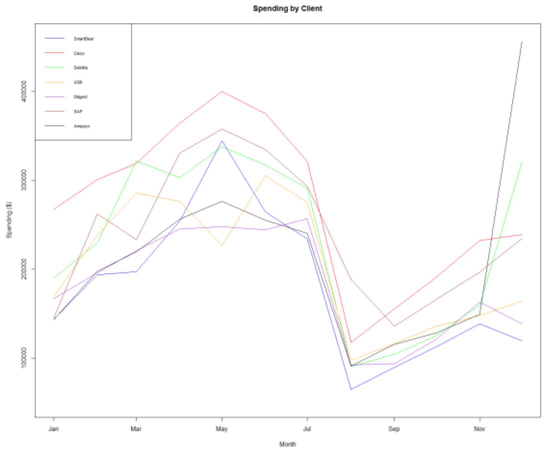
The matrix shows that Wexford, which is shown in pink, has the most dispersed data. This matrix can give the right understanding of the cities and the calls that were made out of totalcalls. Along with that, we can also provide other information, such as the variety of call types that were made and their particulars,for instance Social vs Other Request



GGPLOT():

This scatter plot was created using ggplot(). Although we are examining the same data as before, we are able to plot related to various cities and in all separate sections thanks to ggplot. Also, I have included separate scales for each axis such that each plot is in accordance with their scale.

In order to show the sections of the data where the data points are denser and to provide an outside layer of eclipse that aids in showing any outliers in the data, I also used stat ellipse(). Each plot was created separately, saved in a variable, and the matrix was created using ggpubr::ggarrange.

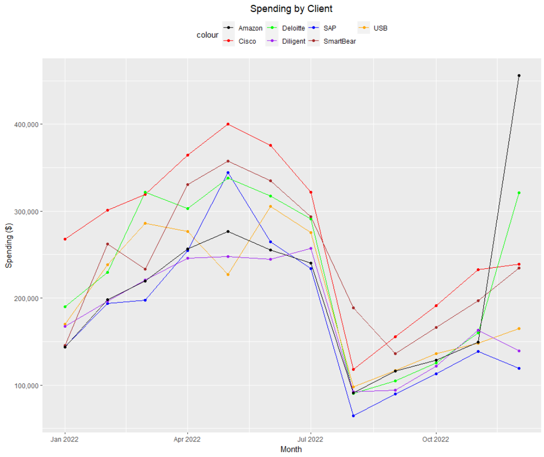


Question2:

We are given information about various businesses and their expenditure over time in the client dataset. The spending over the course of a year is currently being examined.

I utilized the R language's core library, which is lines() for plotting the timeline. for this, Initially, I used the as. Date() function to transform the month column into the time datatype.

While the numeric datatype was first provided in string format, I later transformed it into an integer. Make a plot() next, and I've added all the lines to that plot after that.



The information provided shows how much money different companies spend each month. The information comprises 7 firms and 12 months (January to December) (SmartBear, Cisco, Deloitte, USB, Diligent, SAP, and Amazon). Each company's expenditures are listed in thousands of dollars.

We may create a line graph showing the annualized monthly spending for each company to compare the spending patterns between them. To comprehend each company's general expenditure trends, we can also compute some summary data for them.

The breakdown of each company's spending is as follows:

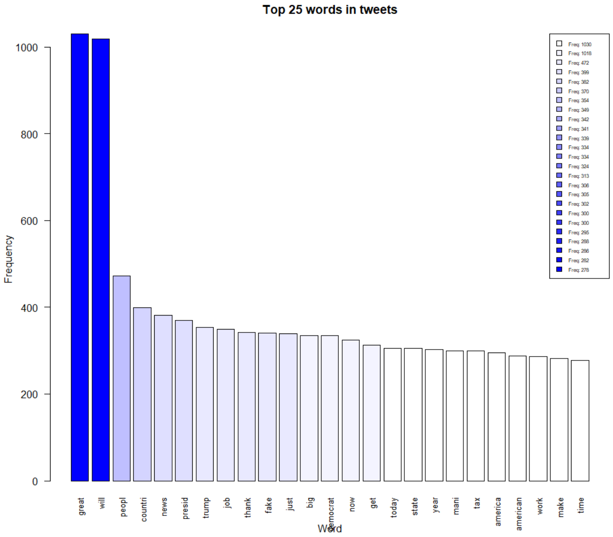
- SmartBear: From a low of $136,099 in September to a high of $357,823 in May, spending fluctuates. $240,503 is the average monthly expenditure.
- Cisco: From a low of $117,937 in August to a high of $400,092 in May, expenditure fluctuates greatly. $256,404 is the average monthly expenditure.
- Deloitte: From a low of $90,638 in August to a high of $321,112 in December, spending fluctuates. $207,695 is the average monthly expenditure.
- USB: From a low of $97,969 in August to a high of $170,001 in January, expenditure fluctuates. $145,165 is the average monthly expenditure.
- SAP: From a low of $64,937 in August to a high of $344,401 in May, spending fluctuates. $176,725 is the average monthly expenditure.
- Amazon: A low of $91,166 was spent on Amazon in August, while a high of $455,990 was spent in December. $222,148 is the average monthly expenditure.

According to the summary, Cisco and Amazon have the largest monthly spending averages, respectively. While USB, Diligent, and SAP have the lowest average monthly expense, Deloitte and SmartBear have similar average monthly expenditures.

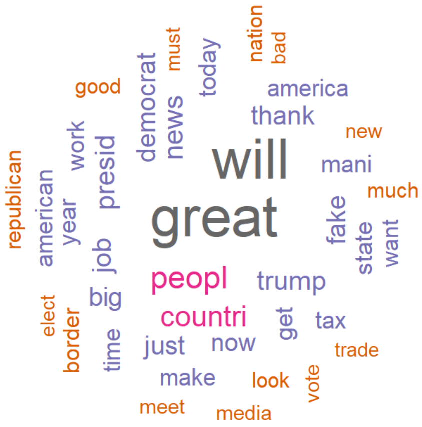
The majority of businesses appear to spend the most in May and the least in August based on annual expenditure trends. The expenditure patterns of some businesses, however, do not always follow this pattern and change significantly over the course of the year. Overall, the data indicate that, among the listed organizations, Amazon and Cisco spend the most money, while USB, Diligent, and SAP spend less.

__Question3__:

The data is provided as a table with five columns: source, text, date, index, and id. A user's tweet is represented by each row. The position of the tweet in the table is indicated by the index, which is an integer. The id is a special integer identification that Twitter assigns to every tweet. The timestamp of the tweet's creation, together with the timezone, is displayed in the date column (in this case, UTC). The tweet's text is located in the text column. Lastly, the source column displays the tool or program that was used to post the tweet (in this case, Twitter for iPhone or Twitter Web Client).



__Question 3.1__: First, I use the Corpus() function from the "tm" package, which turns a character vector into a corpus object, to generate a corpus from the text of the tweets. Subsequently, using the tm map() function from the "tm" package, I pre-process the corpus. Whitespace is removed, all text is made lowercase, digits are removed, punctuation is removed, stopwords are removed, and the words are stemmed.



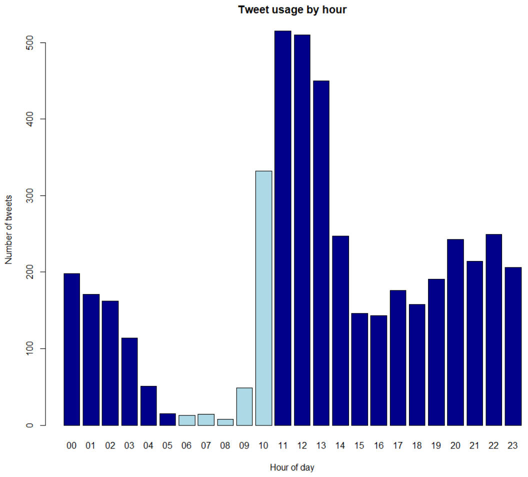
The code then uses DocumentTermMatrix() to construct a document term matrix (corpus). Using colSums (as.matrix(dtm)), this matrix is used to determine the frequency of each word in the corpus. Sort(freq, decreasing = TRUE) is used to sort the frequencies into descending order, and barplot is used to plot the top 25 words (). The word is on the x-axis of the bar graph we made, which you can see above, and the frequency is on the y-axis.

The top 25 words in the corpus of tweets are displayed in the resulting bar plot. These include words like "great", "will", "amp", "people", "country", "news", "president", "Trump", "job", "thank you", "fake", "just", "big", "Democrat", "now", "get", "today", "state", "year", "many", "tax", "America", "American", "work", "make", "time", "want", and "border".

__Question 3.2__: The most frequently used words in the data frame can likewise be obtained via their frequency. This can be a useful method for determining which terms are used most frequently. And so, after creating a word cloud, we can say that the word "great" is the most frequently used, and the word "will" is used more and its the second most common word. Also, we can observe that a word appears more frequently the larger the font is, and that color also indicates a range.

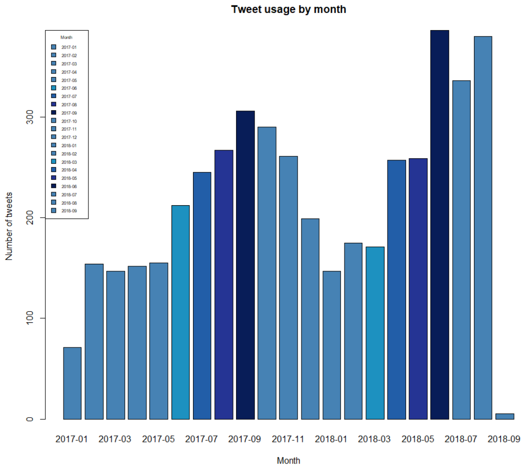
__Question 3.3__: The program creates a bar plot showing the number of tweets sent at each hour of the day. Secondly, it uses the format() function to extract the hour part of the date and time information from the tweets data and puts it in the tweet_times variable.

The hourly_tweet_counts variable then contains the frequency table of the tweet times that was created using the table() function.



The amount of tweets sent during each hour of the day is shown in this data. At 0:00, 198 tweets were sent, followed by 171 at 1:00, 162 at 2:00, and so on until 206 tweets were sent at 23:00. 515 tweets were sent during the busiest hour of 11:00, while just 15 were sent during the quietest hour of 05:00. The data made available indicates that there were 932 tweets between 6 and 10 in total.

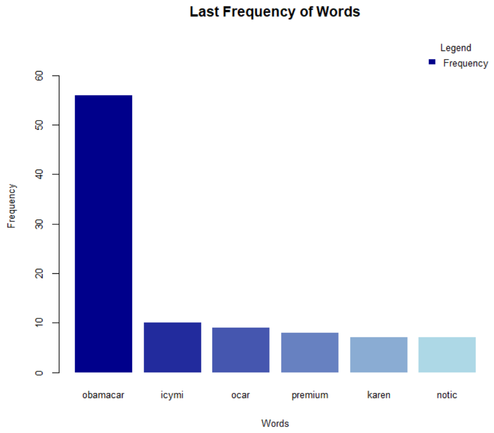
<u>Question 3.4</u>: From the beginning of the dataset until around June or July 2017, there is typically an upward trend in the number of tweets posted. Thereafter, there is typically a downward tendency.



Although there are some dips in the volume of tweets published, overall, the number of tweets published each month appears to be quite consistent following the first rise and subsequent decline.

Compared to earlier months, September 2018 had a significant decrease in the number of tweets published. But, it's crucial to keep in mind that this dataset only contains data from the first nine days of September, so this low figure probably isn't indicative of the full month.

<u>Question 3.5</u>: 'will', 'great', 'amp', 'people', 'news', 'big', 'fake', 'just', 'now', 'president', 'trump', 'country', 'many', 'thank', and 'democrats' are the top 15 terms used in 'Twitter for iPhone. This source appears to be often used to communicate political news and updates etc.



With 'Twitter for Media Studio,' on the other hand, the top 15 terms used are 'great', 'amp', 'will', 'honor', 'today', 'america', 'american', 'thank', 'people', 'together', 'welcome', 'president', 'country', 'border', and 'national' this source is utilized to spread information on national events, political rituals and etc. Several words used in both sources, like "will," "great," "amp," "people," "country," "president," and "thank," are similar.

<u>Question3.6</u>:

According to the analysis, it appears that icymi was mentioned 10 times, obamacare was stated 56 times, and so on. Nevertheless, these phrases were not used in the previous six months.