Malnad College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

Hassan – 573 202

DATA SCIENCE (18703)

Sentimental Analysis
&
Tableau

Submitted by

Team - 06 **Team - 07**

Team - 08 **Team - 09**

Team - 10

under the guidance of

Dr. Balaji Prabhu B V
Assistant professor

Department of Information Science & Engineering
Malnad College of Engineering
Hassan - 573 202

Tel.: 08172-245093 Fax: 08172-245683 URL: www.mcehassan.ac.in

-: TEAM MEMBERS :-

Team 06 –	4MC18IS013 Harshitha N 4MC18IS025 Nisarga B.G 4MC18IS028 Poojitha U.A 4MC18IS029 Prajna N
Team 07 -	4MC18IS001 Abhishek H.S 4MC18IS045 Shashank S 4MC18IS047 Shreyas S.Y
Team 08 –	4MC18IS005 Ashwin Athreya 4MC18IS014 Hemanth Kumar 4MC18IS023 Nikhil S.Y 4MC18IS057 Yashas R
Team 09 –	4MC18IS002 Adithya H.R 4MC18IS018 Mohan Raj 4MC18IS021 Navjot Singh 4MC18IS027 Nithin B.M
Team 10 –	4MC19IS400 Abdul Azeem 4MC18IS011 Harshavardhan T.P 4MC18IS041 Sanjith B.V

Contents

-: Introduction to Tableau :-

Tableau is a tool which is used for applications related to Business Intelligence and Data Visualization. This can help you extract important insights by analyzing the data and providing objective measurements to support and help in strategic decision making for a business.

The platform supports as easy to learn user interface and additional functionalities to collaborate with other employees in the organization. The user can get data from multiple sources and perform analyses on the aggregated data. Tableau is helping industries to reduce the analysis time and provides functionalities while ensuring flexibility, security and reliability.

-: Key Features of Tableau :-

- **Multiple Integrations:** It houses support for numerous integrations and connectors for increased functionality and compatibility with various data sources.
- **Easy User Interface:** It hosts an easy to use and easy to learn user interface to perform complex data transformations without programming know-how.
- **Real-Time Dashboards:** It has the ability to create and host interactive real-time dashboards highlighting KPIs and data visualizations.
- **Gather Insights:** It helps users convert their queries and questions into visualizations and objective metrics.
- **Multi-Platform Accessibility:** It has a provision to access dashboards and reports on multiple devices such as mobile, web and desktop.
- **Visual Customizations:** There is a huge selection of visual customizations and templates that users can utilize to highlight and analyses critical business data.

-: Tableau Superstore Dataset Part 1: Building a View :-

Step 1: You can start by dragging the attribute of your choice to the columns shelf in the workspace area to represent it on the X axis on the charts.

Step 2: You can proceed to add the attribute of your choice to the rows shelf on the workspace by dragging it from the left pane to represent it on the Y axis on the charts.

Step 3: This will create a Line Chart as depicted above, you can modify the details of the chart by changing the setting in the Marks area on the left side of the workspace.

	Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Region	Product ID	Category	ub-Categor	roduct Nam	Sales	Quantity	Discount	Profit
2	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-BO-10001798	Furniture	Bookcases	Bush Somei	261.96	2	0	41.9136
3	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe	731.94	3	0	219.582
4	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036	West	OFF-LA-10000240	Office Supplies	Labels	Self-Adhesi	14.62	2	0	6.8714
5	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311	South	FUR-TA-10000577	Furniture	Tables	Bretford CR	957.5775	5	0.45	-383.031
6	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311	South	OFF-ST-10000760	Office Supplies	Storage	Eldon Fold	22.368	2	0.2	2.5164
7	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	FUR-FU-10001487	Furniture	Furnishings	Eldon Expre	48.86	7	0	14.1694
8	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	OFF-AR-10002833	Office Supplies	Art	Newell 322	7.28	4	0	1.9656
9	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	TEC-PH-10002275	Technology	Phones	Mitel 5320	907.152	6	0.2	90.7152
10	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	OFF-BI-10003910	Office Supplies	Binders	DXL Angle-\	18.504	3	0.2	5.7825
11	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	OFF-AP-10002892	Office Supplies	Appliances	Belkin F5C2	114.9	5	0	34.47
12	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	FUR-TA-10001539	Furniture	Tables	Chromcraft	1706.184	9	0.2	85.3092
13	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	TEC-PH-10002033	Technology	Phones	Konftel 250	911.424	4	0.2	68.3568
14	AA-10480	Andrew Allen	Consumer	United States	Concord	North Carolina	28027	South	OFF-PA-10002365	Office Supplies	Paper	Xerox 1967	15.552	3	0.2	5.4432
15	IM-15070	Irene Maddox	Consumer	United States	Seattle	Washington	98103	West	OFF-BI-10003656	Office Supplies	Binders	Fellowes PE	407.976	3	0.2	132.5922
16	HP-14815	Harold Pawlan	Home Office	United States	Fort Worth	Texas	76106	Central	OFF-AP-10002311	Office Supplies	Appliances	Holmes Rep	68.81	5	0.8	-123.858
17	HP-14815	Harold Pawlan	Home Office	United States	Fort Worth	Texas	76106	Central	OFF-BI-10000756	Office Supplies	Binders	Storex Dura	2.544	3	0.8	-3.816
18	PK-19075	Pete Kriz	Consumer	United States	Madison	Wisconsin	53711	Central	OFF-ST-10004186	Office Supplies	Storage	Stur-D-Stor	665.88	6	0	13.3176
19	AG-10270	Alejandro Grove	Consumer	United States	West Jordan	Utah	84084	West	OFF-ST-10000107	Office Supplies	Storage	Fellowes Su	55.5	2	0	9.99
20	ZD-21925	Zuschuss Donatelli	Consumer	United States	San Francisco	California	94109	West	OFF-AR-10003056	Office Supplies	Art	Newell 341	8.56	2	0	2.4824
21	ZD-21925	Zuschuss Donatelli	Consumer	United States	San Francisco	California	94109	West	TEC-PH-10001949	Technology	Phones	Cisco SPA 5	213.48	3	0.2	16.011
22	ZD-21925	Zuschuss Donatelli	Consumer	United States	San Francisco	California	94109	West	OFF-BI-10002215	Office Supplies	Binders	Wilson Jon	22.72	4	0.2	7.384
23	KB-16585	Ken Black	Corporate	United States	Fremont	Nebraska	68025	Central	OFF-AR-10000246	Office Supplies	Art	Newell 318	19.46	7	0	5.0596
24	KB-16585	Ken Black	Corporate	United States	Fremont	Nebraska	68025	Central	OFF-AP-10001492	Office Supplies	Appliances	Acco Six-Ou	60.34	7	0	15.6884
25	SF-20065	Sandra Flanagan	Consumer	United States	Philadelphia	Pennsylvania	19140	East	FUR-CH-10002774	Furniture	Chairs	Global Delu	71.372	2	0.3	-1.0196
26	EB-13870	Emily Burns	Consumer	United States	Orem	Utah	84057	West	FUR-TA-10000577	Furniture	Tables	Bretford CR	1044.63	3	0	240.2649
27	EH-13945	Eric Hoffmann	Consumer	United States	Los Angeles	California	90049	West	OFF-BI-10001634	Office Supplies	Binders	Wilson Jon	11.648	2	0.2	4.2224
28	EH-13945	Eric Hoffmann	Consumer	United States	Los Angeles	California	90049	West	TEC-AC-10003027	Technology	Accessories	Imation 8G	90.57	3	0	11.7741

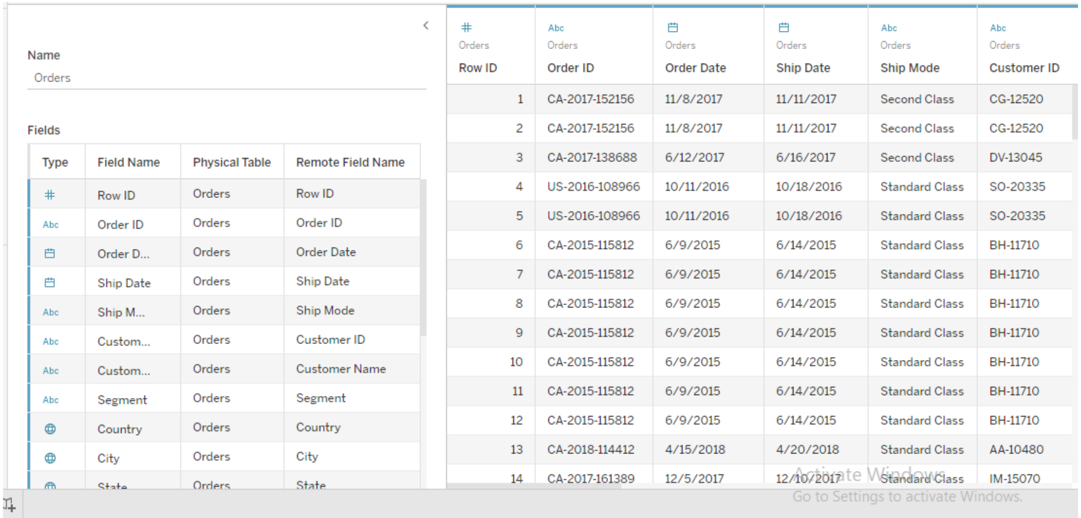
Orders

Data set

-: Tableau Superstore Dataset Part 2: Refining the View :-

Step 1: You can start by adding additional dimensions and metrics to the Columns Shelf of the workplace. For example you can include both YEAR and Category attributes in the Columns Shelf to categories the visualized data further.

Step 2: You can also add attributes from the left pane to the View by just Double clicking on the attribute. Tableau automatically makes the assumption of showcasing that data in either Columns or Rows Shelf. This procedure will help you get the right amount of detail in your Data Visualization.



Refined Data set

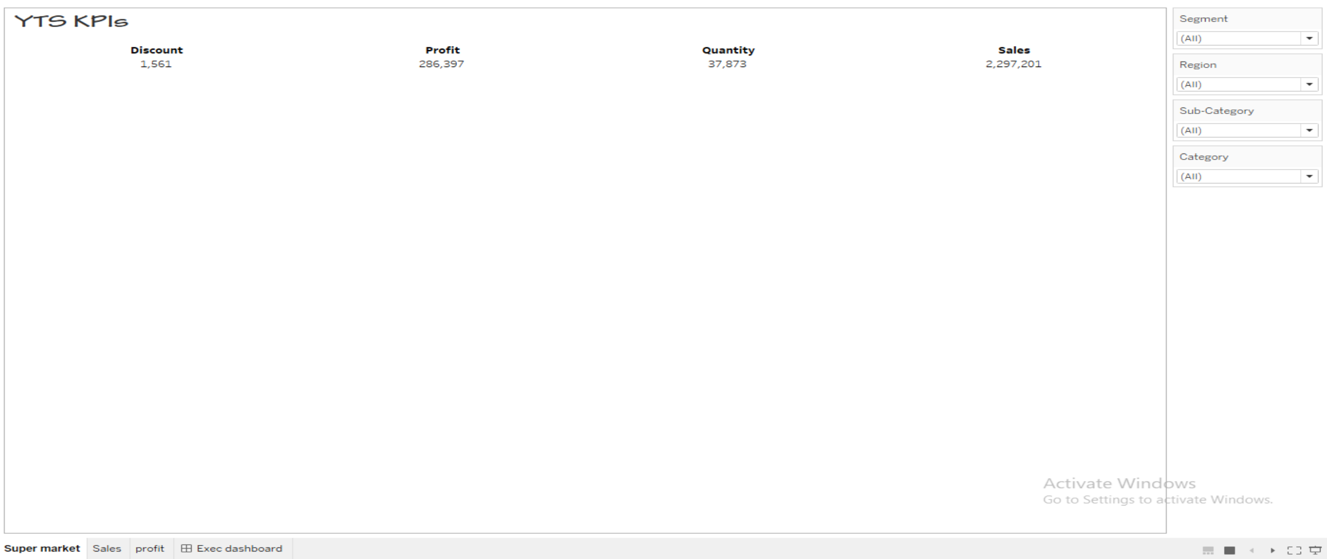
-: Understanding filters using Tableau Superstore Dataset :-

Filter and Colors can be a useful way to include and exclude crucial information in Big Datasets. For example, showcasing Sales of different products with different colors, including data from a certain location to check for customer trends etc.

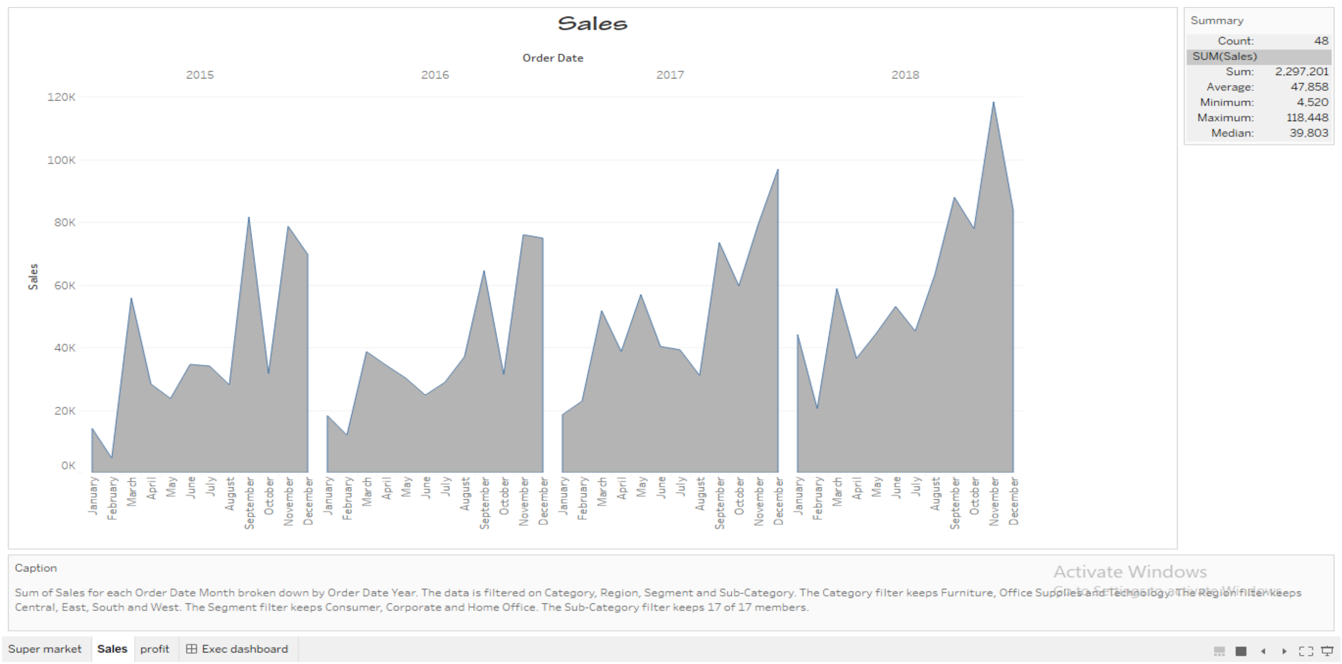
Step 1: You can start by right-clicking on an attribute on the left pane and selecting Show Filter.

Step 2: This will showcase the options of various filters on the right side of the workspace and showcase above. You can use the checkboxes to implement your desired filters on the dataset.

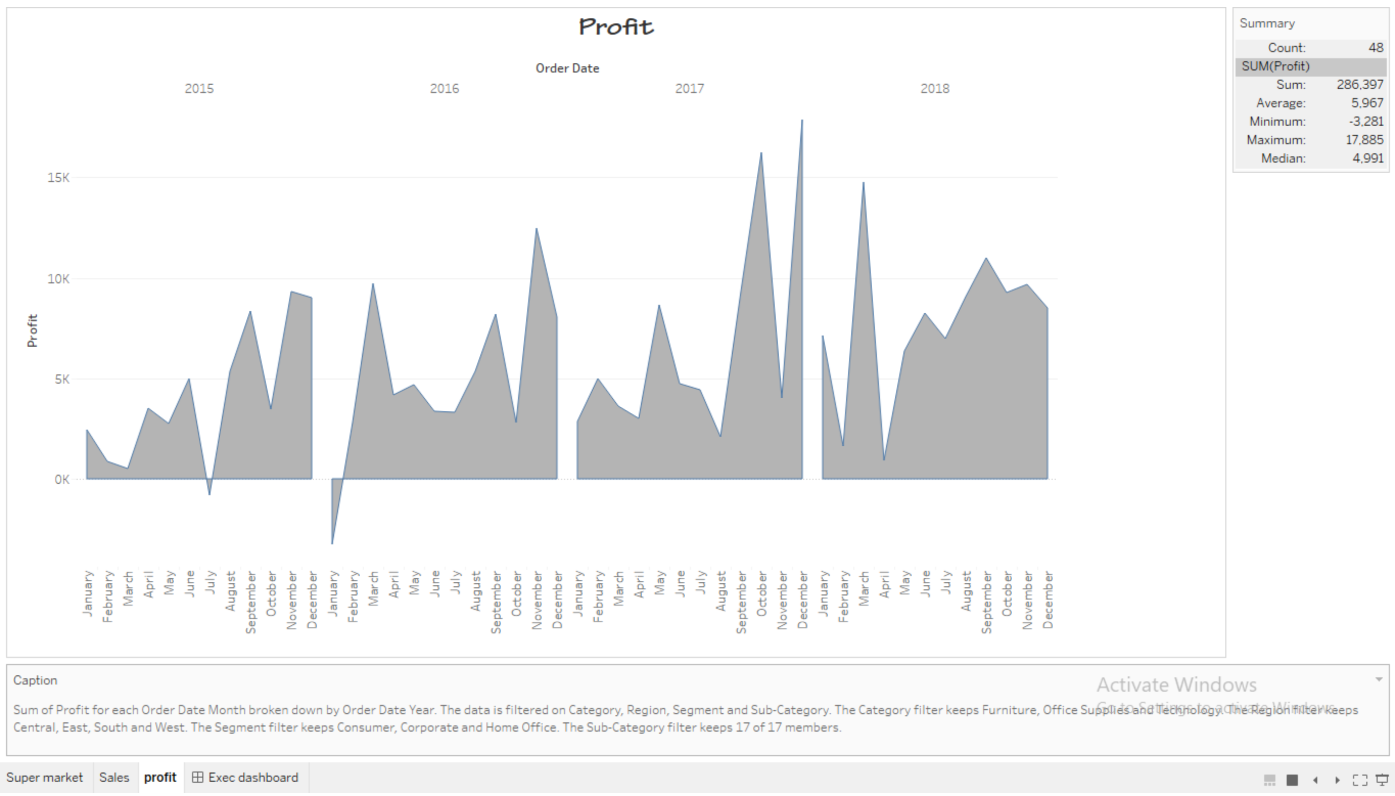
Step 3: Colors can be added into the visualization by dragging the desired attribute to the Color option in the Marks Card. The desired colors can be selected by clicking on the colors option and the effects will be implemented like the one shown below.



Super Store main window



Sales data visualization



Profit data visualization

-: Setting up Sales Dashboard using Tableau Superstore Dataset :-

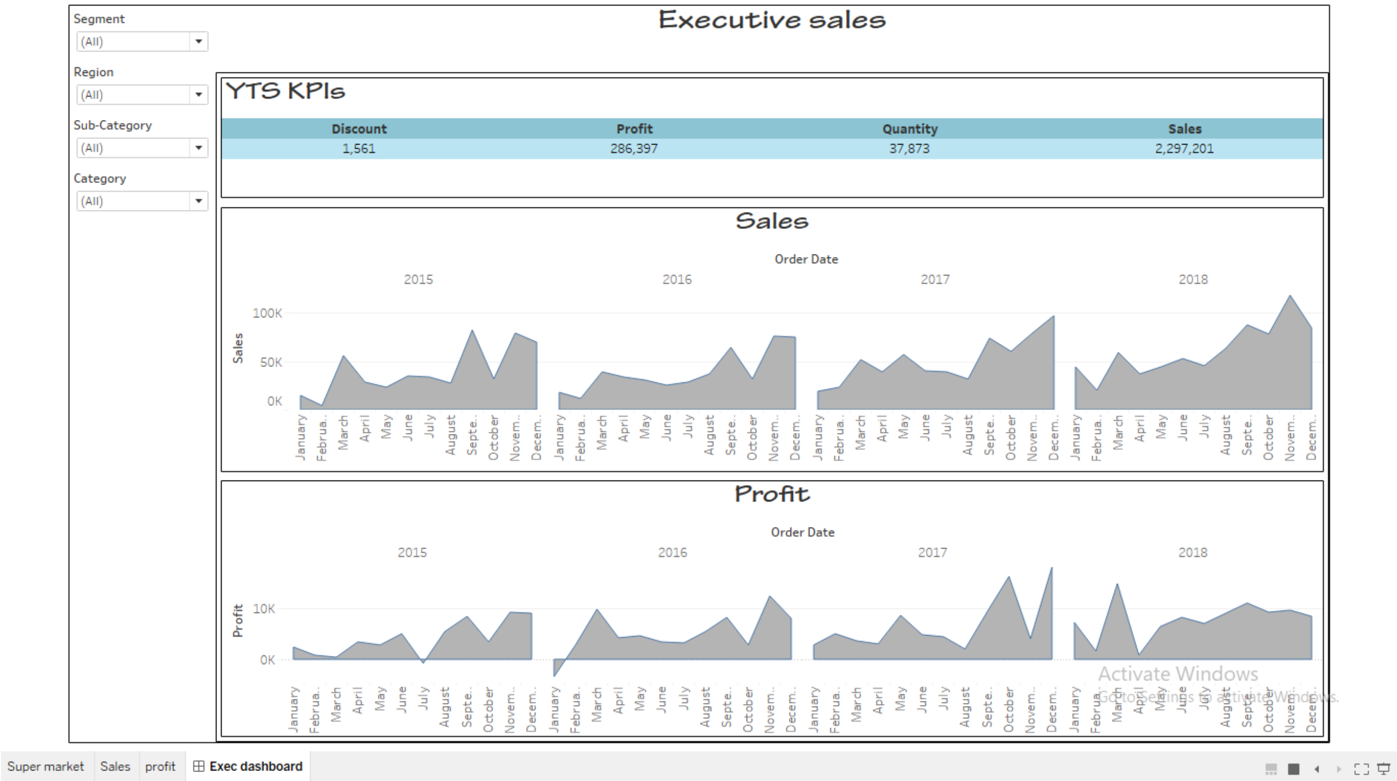
Dashboards are required to showcase the visualized data in an organized manner. They help highlight important information pertaining to a certain theme or objective. Dashboards allow interactive data visualizations to be shared between employees and management. Users can leverage the Tableau Superstore Dataset to create such Dashboards and learn to organize the visual elements in a helpful manner.

Step 1: You are firstly required to click on the New Dashboard button.

Step 2: You can select various Worksheets, Views and visualisations that you may have created in the other Worksheets. For this example, you can choose and drag "Sales in South" and "Profit Map" to the empty dashboard. The Dashboard will start displaying the data as follows:

Step 3: Unnecessary information can be removed from the Dashboard by right clicking on the Column area of the desired View and removing the check-box on "Show Header" option.

Step 4: You can implement filters on the data to highlight the required information. For this example, you can click on the Drop Down arrow on the "Year of the Order Date" filer and proceed by selecting "Single Value" (Slider).



Super Store Dashboard

-: Conclusion :-

- In this article, you were introduced to Tableau and its key features.
- You learned about Tableau Superstore Dataset which is a sample Dataset provided by Tableau.
- Steps to connect Tableau to the sample Dataset.
- Various methods to interact with the elements of the Dataset and visualize them in the workspace.
- You also learned about procedures to create a Geographical Map View to get a more granular view of the data and set up a Sales Dashboard using this sample Dataset to the interactive visualization with other members of the company.

-: References :-

Source: https://hevodata.com/learn/tableau-superstore-data/#intro

Tableau Software: https://public.tableau.com/s/

Malnad College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

Hassan – 573 202

Data Science Activity

"Sentimental Analysis On Twitter data"

Contents

1. Introduction

1.1 Sentiment Analysis

Sentiment analysis gives us insight into the things that automate mining of attitudes, opinions, views and emotions from text, speech, tweets and database sources. However, to fully explore the possibilities of this text analysis technique, we need data visualization tools to help organize the results. Visually representing the content of a text document is one of the most important tasks in the field of text mining.

However, there are some gaps between visualizing unstructured (text) data and structured data. Many text visualizations do not represent the text directly, they represent an output of a language model. In this post, we will use tweets extracted using Twitter API, store tweets as text data, classify opinions in text into categories like positive, or negative or neutral, create a function to calculate the score of each type of opinion in the text and try to explore and visualize as much as we can, using R libraries.

Tweets can be imported into R using Twitter API, then the text data has to be cleaned before analysis, for example removing emoticons, removing URLs, etc.

2.Loading all the required R libraries

```
library(Twitter)

library(ROAuth)

library(hms)

library(lubridate) library(tidytext)

library™

library(wordcloud)

library(igraph)

library(glue)

library(networkD3)

library(rtweet)

library(plyr)

library(stringr)

library(ggplot2)
```

```r
library(ggeasy)

library(plotly)

library(dplyr)

library(hms)

library(lubridate)

library(magrittr)

library(tidyverse)

library(janeaustenr)

library(widyr)
```

3.How to Perform Sentiment Analysis on Tweets

3.1 Twitter authorization to extract tweets:

As a first step, we need to get authorized credentials from Twitter to use the API for extracting the tweets. Steps involve creating a Twitter developer account, creating an app and then we have necessary credentials. Reference for obtaining access tokens: https://cran.r-project.org/web/packages/rtweet/vignettes/auth.html

```r
#Note: Replace below with your credentials following above reference

api_key <- "xxxxxxxxxxxxxxxxxxxxxxxxx"

api_secret <- "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

access_token <- "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

access_token_secret <- "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

#Note: This will ask us permission for direct authentication, type '1' for yes:

setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)

## [1] "Using direct authentication"
```

```r
# extracting 4000 tweets related to global warming topic

tweets <- searchTwitter("#globalwarming", n=4000, lang="en")

n.tweet <- length(tweets)

# convert tweets to a data frame
```

```r
tweets.df <- twListToDF(tweets)

tweets.txt <- sapply(tweets, function(t)t$getText())

# Ignore graphical Parameters to avoid input errors

tweets.txt <- str_replace_all(tweets.txt,"[^[:graph:]]", " ")

## pre-processing text:

clean.text = function(x){

  # convert to lower case

x = tolower(x)  # remove rt

x = gsub("rt", "", x)  # remove at

x = gsub("@\\w+", "", x)  # remove punctuation

 x = gsub("[[:punct:]]", "", x)  # remove numbers

x = gsub("[[:digit:]]", "", x)  # remove links http

 x = gsub("http\\w+", "", x)  # remove tabs

x = gsub("[ |\t]{2,}", "", x)  # remove blank spaces at the beginning

x = gsub("^ ", "", x)  # remove blank spaces at the end

x = gsub(" $", "", x)  # some other cleaning text

 x = gsub('https://','',x)  x = gsub('http://','',x)

x = gsub('[^[:graph:]]', ' ',x)

 x = gsub('[[:punct:]]', '', x)

x = gsub('[[:cntrl:]]', '', x)

x = gsub('\\d+', '', x)

 x = str_replace_all(x,"[^[:graph:]]", " ")

return(x)}

cleanText <- clean.text(tweets.txt)

# remove empty results (if any)

idx <- which(cleanText == " ")

cleanText <- cleanText[cleanText != " "]
```

3.3 Frequency of Tweets

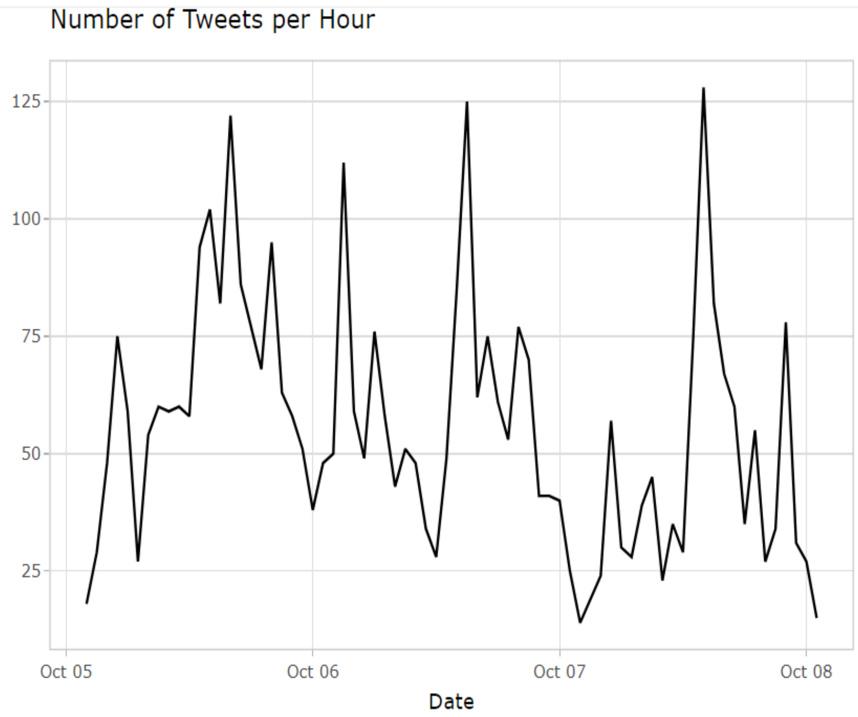
```
tweets.df %<>%

mutate(

created = created %>%

# Remove zeros.

 str_remove_all(pattern = '\\+0000') %>%

  # Parse date.

 parse_date_time(orders = '%y-%m-%d %H%M%S')

)

tweets.df %<>%

 mutate(Created_At_Round = created%>% round(units = 'hours') %>% as.POSIXct())

tweets.df %>% pull(created) %>% min()
```

```
## [1] "2021-10-05 01:34:17 UTC"
```

```
tweets.df %>% pull(created) %>% max()
```

```
## [1] "2021-10-08 01:25:52 UTC"
```

```
plt <- tweets.df %>%

dplyr::count(Created_At_Round) %>%

ggplot(mapping = aes(x = Created_At_Round, y = n)) +

theme_light() +

geom_line() +

xlab(label = 'Date') +

ylab(label = NULL) +

ggtitle(label = 'Number of Tweets per Hour')

plt %>% ggplotly()
```



3.4 Estimating Sentiment Score

There are many resources describing methods to estimate sentiment. For the purpose of this tutorial, we will use a very simple algorithm which assigns sentiment score of the text by simply counting the number of occurrences of "positive" and "negative" words in a tweet.

Hu & Liu have published an "Opinion Lexicon" that categorizes approximately 6,800 words as positive or negative, which can be downloaded from this link:http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

3.5 Loading sentiment word lists

```
positive = scan('resources/twitter_sentiment_analysis/positive-words.txt', what = 'character', comment.char = ';')

negative = scan('resources/twitter_sentiment_analysis/negative-words.txt', what = 'character', comment.char = ';')

# add your list of words below as you wish if missing in above read lists

pos.words = c(positive,'upgrade','Congrats','prizes','prize','thanks','thnx', 'Grt','gr8','plz','trending','recovering','brainstorm','leader')

neg.words = c(negative,'wtf','wait','waiting','epicfail','Fight','fighting', 'arrest','no','not')
```

3.6 Sentiment scoring function:

```
score.sentiment = function(sentences, pos.words, neg.words, .progress='none')

{  require(plyr)

 require(stringr)

# we are giving vector of sentences as input.

# plyr will handle a list or a vector as an "l" for us

# we want a simple array of scores back, so we use "l" + "a" + "ply" = laply:

 scores = laply(sentences, function(sentence, pos.words, neg.words) {

# clean up sentences with R's regex-driven global substitute, gsub() function:    sentence = 
gsub('https://','',sentence)

sentence = gsub('http://','',sentence)

sentence = gsub('[^[:graph:]]', ' ',sentence)

sentence = gsub('[[:punct:]]', '', sentence)

sentence = gsub('[[:cntrl:]]', '', sentence)

sentence = gsub('\\d+', '', sentence)

sentence = str_replace_all(sentence,"[^[:graph:]]", " ")

 # and convert to lower case:

 sentence = tolower(sentence)

# split into words. str_split is in the stringr package

word.list = str_split(sentence, '\\s+')

# sometimes a list() is one level of hierarchy too much

words = unlist(word.list)

# compare our words to the dictionaries of positive & negative terms

pos.matches = match(words, pos.words)

neg.matches = match(words, neg.words)

# match() returns the position of the matched term or NA

# we just want a TRUE/FALSE:

 pos.matches = !is.na(pos.matches)

 neg.matches = !is.na(neg.matches)

# TRUE/FALSE will be treated as 1/0 by sum():
```

```r
score = sum(pos.matches) - sum(neg.matches)

return(score)  },

pos.words, neg.words, .progress=.progress )

scores.df = data.frame(score=scores, text=sentences)

 return(scores.df)

}
```

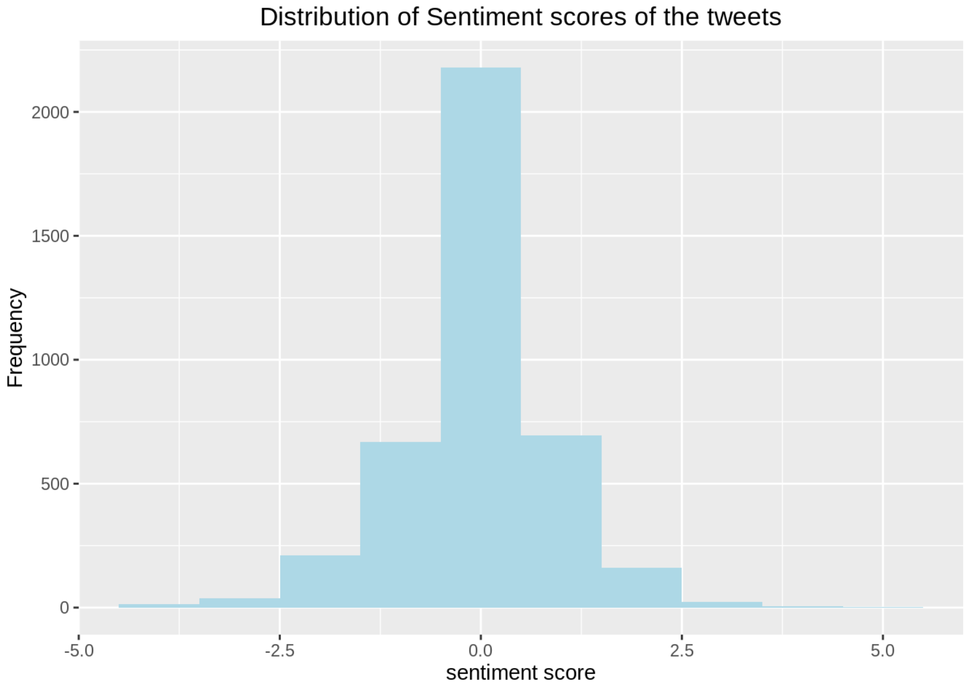
3.7 Calculating the sentiment score

```r
analysis <- score.sentiment(cleanText, pos.words, neg.words)

# sentiment score frequency table

table(analysis$score)
```

```
##

##   -4   -3   -2   -1    0    1    2    3    4    5

##   15   39  211  669 2178  694  162   24    6    2
```

3.8 Histogram of sentiment scores

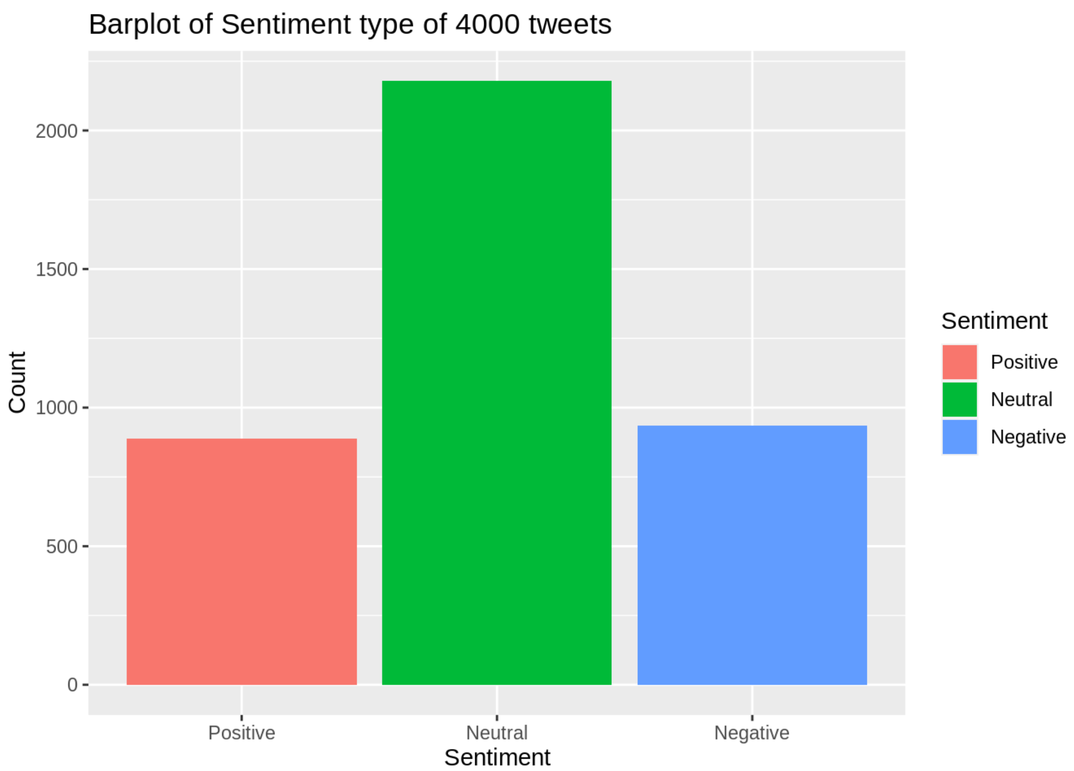
```r
analysis %>%

ggplot(aes(x=score)) +

geom_histogram(binwidth = 1, fill = "lightblue")+

ylab("Frequency") +

xlab("sentiment score") +

ggtitle("Distribution of Sentiment scores of the tweets") +

ggeasy::easy_center_title()
```



Analysis: From the Histogram of Sentiment scores, we can see that around half of the tweets have sentiment score as zero i.e. Neutral and overall as expected, the distribution depicts negative sentiment in the tweets related to global warming, since it is a major issue of concern.

3.9 Bar plot of sentiment type

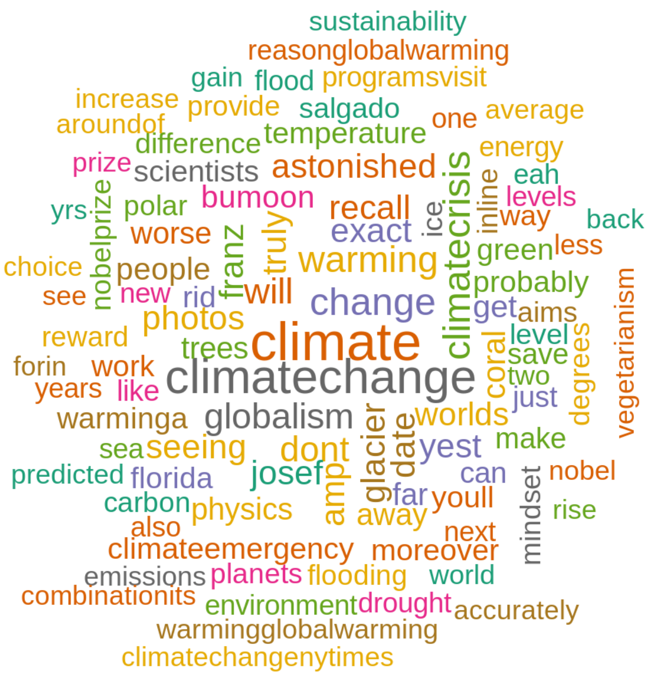
```
neutral <- length(which(analysis$score == 0))

positive <- length(which(analysis$score > 0))

negative <- length(which(analysis$score < 0))

Sentiment <- c("Positive","Neutral","Negative")

Count <- c(positive,neutral,negative)

output <- data.frame(Sentiment,Count)

output$Sentiment<-factor(output$Sentiment,levels=Sentiment)

ggplot(output, aes(x=Sentiment,y=Count))+

geom_bar(stat = "identity", aes(fill = Sentiment))+

ggtitle("Barplot of Sentiment type of 4000 tweets")
```



Analysis: It is also clear from this barplot of sentiment type that around half of the tweets have sentiment score as zero i.e. Neutral and there are more negative sentiment tweets than that of positive sentiment. This barplot helps us to identify overall opinion of the people about global warming.

3.10 Wordcloud

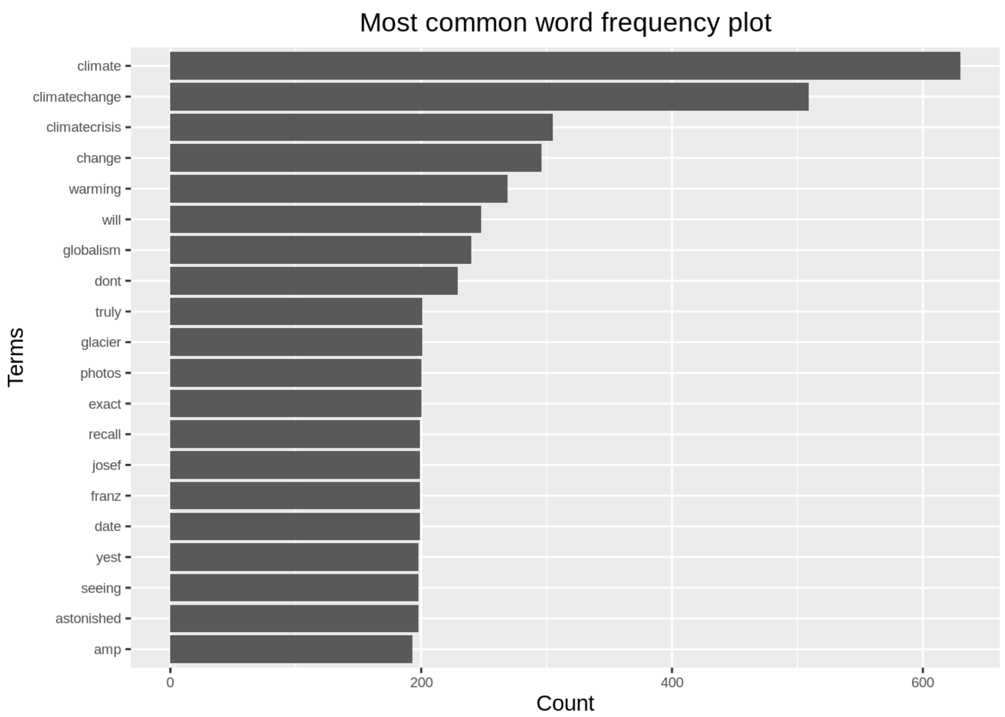
```
text_corpus <- Corpus(VectorSource(cleanText))

text_corpus <- tm_map(text_corpus, content_transformer(tolower))

text_corpus <- tm_map(text_corpus, function(x)removeWords(x,stopwords("english")))

text_corpus <- tm_map(text_corpus, removeWords, c("global","globalwarming"))

tdm <- TermDocumentMatrix(text_corpus)tdm <- as.matrix(tdm)

tdm <- sort(rowSums(tdm), decreasing = TRUE)

tdm <- data.frame(word = names(tdm), freq = tdm)

set.seed(123)

wordcloud(text_corpus, min.freq = 1, max.words = 100, scale = c(2.2,1),

colors=brewer.pal(8, "Dark2"), random.color = T, random.order = F)
```



Analysis: Wordcloud helps us to visually understand the important terms frequently used in the tweets related to global warming, here for example, “climate change”, “environmental”, “temperature”, “emissions”, etc.

3.11 Word Frequency plot

```
ggplot(tdm[1:20,], aes(x=reorder(word, freq), y=freq)) +

geom_bar(stat="identity") +

xlab("Terms") +

ylab("Count") +

coord_flip() +

theme(axis.text=element_text(size=7)) +

ggtitle('Most common word frequency plot') +

ggeasy::easy_center_title()
```



Analysis: we can infer that the most frequently used terms in the tweets related to global warming are, “climate”, “climatechange”, “since”, “biggest”, “hoax”, etc.

3.12 Network Analysis

We are using a weighted network (graph) to describe how to encode and visualize text data. In this section we are counting pairwise relative occurrence of words.

Bigram analysis and Network definition

Bigram counts pairwise occurrences of words which appear together in the text.

```
#bigram

bi.gram.words <- tweets.df %>%

unnest_tokens(

input = text,

output = bigram,

token = 'ngrams',

n = 2  ) %>%

filter(! is.na(bigram))

bi.gram.words %>%

select(bigram) %>%

head(10)
```

```
##   bigram
## 1       rt antalyadf
## 2  antalyadf adfdata
## 3   adfdata focusing
## 4        focusing on
## 5   on globalwarming
## 6  globalwarming the
## 7             the 10
## 8         10 warmest
## 9      warmest years
## 10          years on
```

```
extra.stop.words <- c('https')
stopwords.df <- tibble(
 word = c(stopwords(kind = 'es'),
stopwords(kind = 'en'),
extra.stop.words)
)
```

Next, we filter for stop words and remove white spaces.

```
bi.gram.words %<>%
separate(col = bigram, into = c('word1', 'word2'), sep = ' ') %>%
filter(! word1 %in% stopwords.df$word) %>%
filter(! word2 %in% stopwords.df$word) %>%
filter(! is.na(word1)) %>%
filter(! is.na(word2))
```

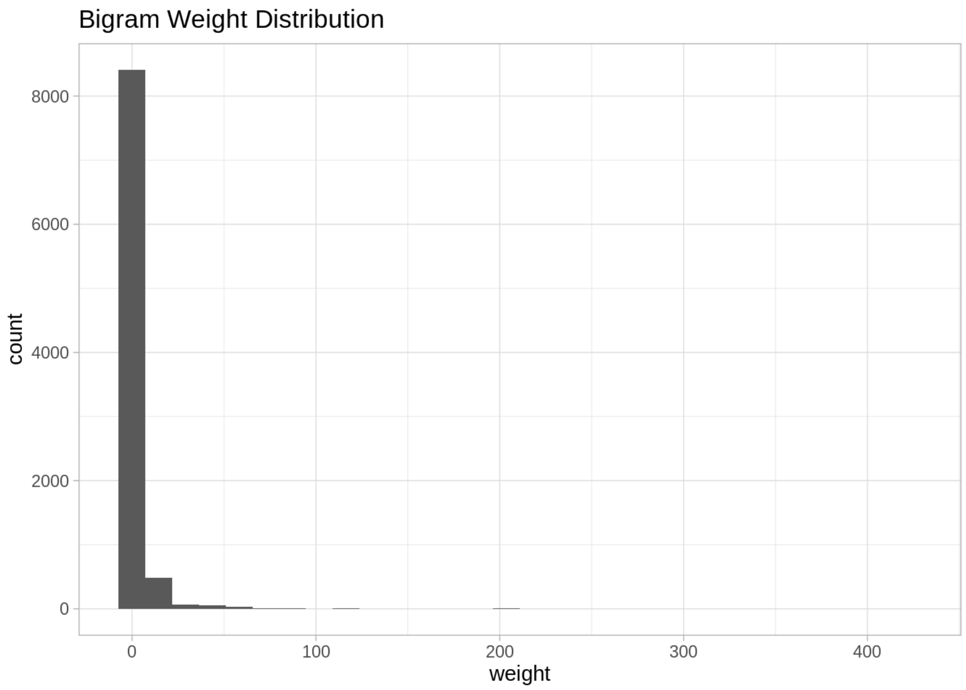
Finally, we group and count by diagram.

```
bi.gram.count <- bi.gram.words %>%
dplyr::count(word1, word2, sort = TRUE) %>%
dplyr::rename(weight = n)
bi.gram.count %>% head()
```

```
##            word1         word2 weight
## 1         global       warming    423
## 2        climate        change    371
## 3             rt  johnrmoffitt    334
## 4 globalwarming climatechange    268
## 5          franz         josef    199
## 6     astonished          yest    198
```

Let us plot the distribution of the weightvalues:

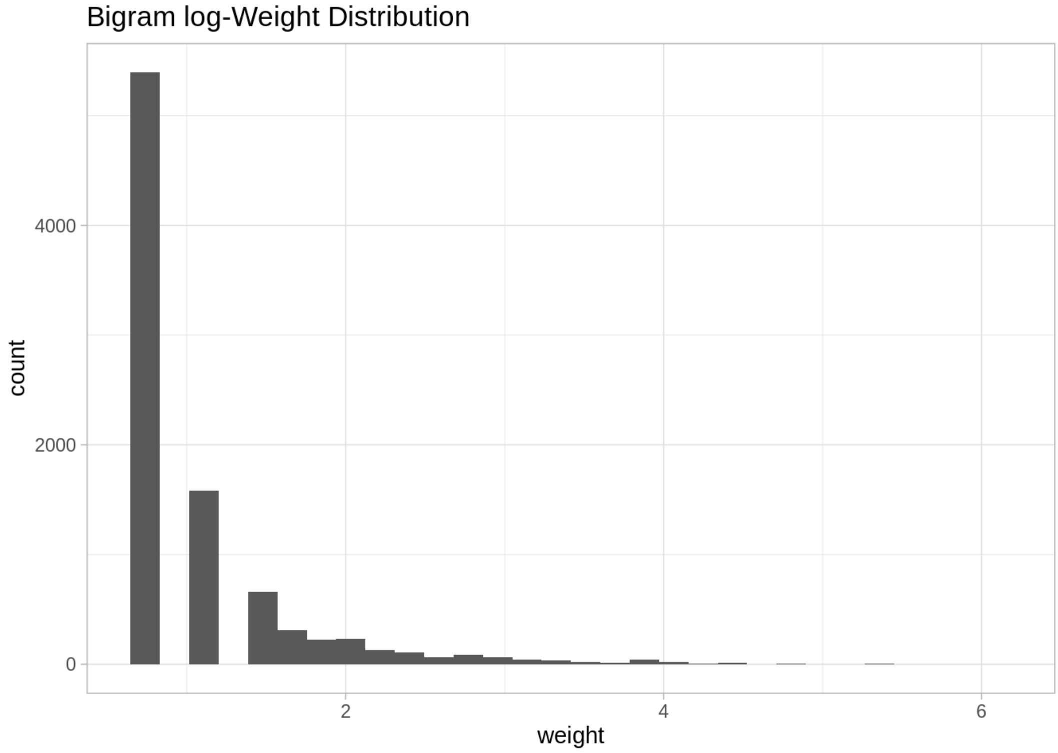
```
bi.gram.count %>%
 ggplot(mapping = aes(x = weight)) +
theme_light() +  geom_histogram() +
 labs(title = "Bigram Weight Distribution")
```



Note that it is very skewed, for visualization purposes it might be a good idea to perform a transformation, eg log transform:

```
bi.gram.count %>%
mutate(weight = log(weight + 1)) %>%
ggplot(mapping = aes(x = weight)) +
theme_light() +
```

```
geom_histogram() +

labs(title = "Bigram log-Weight Distribution")
```



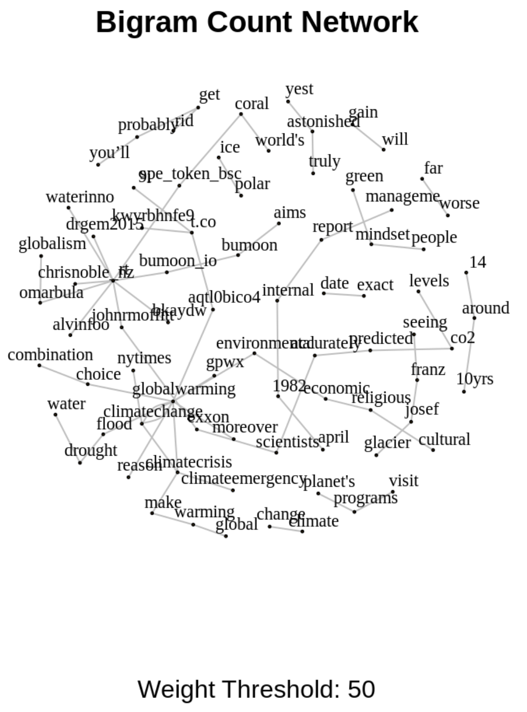
In order to define weighted network from a bigram count we used the following structure.

- Each word is going to represent a node.
- Two words are going to be connected if they appear as a bigram.
- The weight of an edge is the number of times the bigram appears in the corpus.

3.13 Network visualization

```
threshold <- 50

# For visualization purposes we scale by a global factor.

ScaleWeight <- function(x, lambda) {

x / lambda}

network <-  bi.gram.count %>%

filter(weight > threshold) %>%

mutate(weight = ScaleWeight(x = weight, lambda = 2E3)) %>%   graph_from_data_frame(directed = FALSE)

plot(

network,
```

```
vertex.size = 1,

vertex.label.color = 'black',

 vertex.label.cex = 0.7,

vertex.label.dist = 1,

edge.color = 'gray',

 main = 'Bigram Count Network',

sub = glue('Weight Threshold: {threshold}'),

alpha = 50)
```



We can even improvise the representation by setting the sizes of the nodes and the edges by the degree and weight respectively.

```
threshold <- 50network <-  bi.gram.count %>%

filter(weight > threshold) %>%

graph_from_data_frame(directed = FALSE)

# Store the degree.

V(network)$degree <- strength(graph = network)
```

```r
# Compute the weight shares.

E(network)$width <- E(network)$weight/max(E(network)$weight)

# Create networkD3 object.

network.D3 <- igraph_to_networkD3(g = network)

# Define node size.

network.D3$nodes %<>% mutate(Degree = (1E-2)*V(network)$degree)

# Define color group

network.D3$nodes %<>% mutate(Group = 1)

# Define edges width.

 network.D3$links$Width <- 10*E(network)$width

forceNetwork(

Links = network.D3$links,

Nodes = network.D3$nodes,

Source = 'source',

Target = 'target',

NodeID = 'name',

Group = 'Group',

opacity = 0.9,

Value = 'Width',

Nodesize = 'Degree',

 # We input a JavaScript function.

linkWidth = JS("function(d) { return Math.sqrt(d.value); }"),

fontSize = 12,

zoom = TRUE,

opacityNoHover = 1)
```

4. References:

Bing Liu, Minqing Hu and Junsheng Cheng. “Opinion Observer: Analyzing and Comparing Opinions on the Web.” Proceedings of the 14th International World Wide Web conference (WWW-2005), May 10-14, 2005, Chiba, Japan.