# Project 2

Abhinav,Mukul,Vijay

**Loading libraries**

```
library(readr)
library(stringr)
library(janeaustenr)
library(igraph)
library(dplyr)
library(ggplot2)
library(tm)
library(tidytext)
library(tidyr)
library(ggraph)
library(kableExtra)
```

## Task -1

### 1.1 - Use the solution from homework 1 question 3.1 as the adjacency matrix

```
df_keywords <- read_csv("Keyword_data - Keyword_data.csv")

## drop rows with null keywords
df_keywords1 <- df_keywords[!is.na(df_keywords$`Keyword 1`), ]
temp <- c()

## iterate over the keyword dataframe, pick each keyword and append it into the vector
for (i in 1:dim(df_keywords1)[1]) {
  for (j in 2:dim(df_keywords1)[2]) {
    if (!is.na(df_keywords1[[i, j]])) {
      temp <- c(temp, tolower(df_keywords1[[i, j]]))
    }
  }
}
## Drop duplicates from vector
temp <- unique(temp)

## Create a matrix with dimension equal to length of number of unique keywords
adjacency_matrix <- matrix(0, nrow = length(temp), ncol = length(temp))

## Rename column and row names as the keywords
rownames(adjacency_matrix) <- c(temp)
colnames(adjacency_matrix) <- c(temp)
```

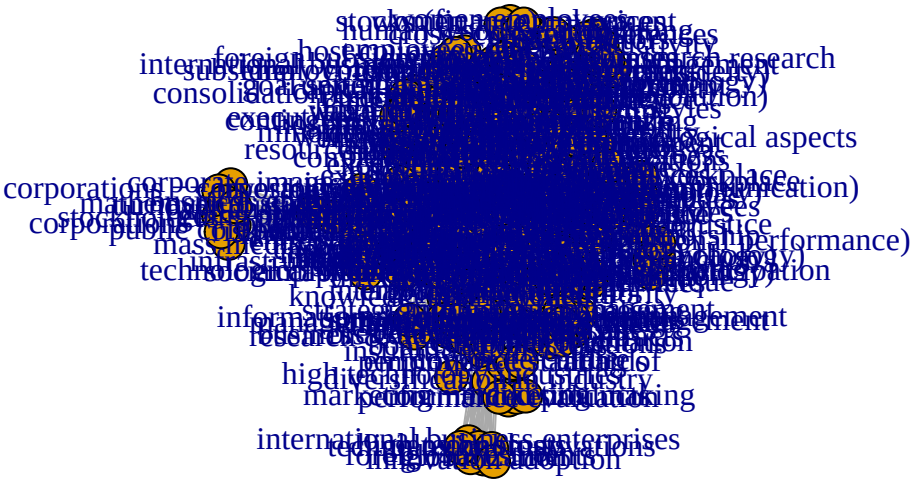
```
## Iterate over keyword dataframe and increment values in the empty matrix
n_rows <- dim(df_keywords1)[1]
for (x in 1:n_rows) {
  for (i in 2:length(df_keywords1)) {
    for (j in 2:length(df_keywords1)) {

      ## Find keyword 1 and keyword 2
      key1 <- tolower(df_keywords1[[x, i]])
      key2 <- tolower(df_keywords1[[x, j]])

      ## Key1 should not be equal to Key2 and either of them should not be null
      if ((!is.na(key1)) && (!is.na(key2)) && (key1 != key2)) {
        adjacency_matrix[key1, key2] <- adjacency_matrix[key1, key2] + 1
      }
    }
  }
}

adjacency_df <- data.frame(adjacency_matrix)

# Sample output of the dataframe
knitr::kable(head(adjacency_df[,1:5],10))
```

| | equity | organizational.sociology | performance | meta.analysis | psychometrics |
|---|---|---|---|---|---|
| equity | 0 | 1 | 1 | 1 | 1 |
| organizational sociology | 1 | 0 | 2 | 1 | 1 |
| performance | 1 | 2 | 0 | 1 | 1 |
| meta-analysis | 1 | 1 | 1 | 0 | 1 |
| psychometrics | 1 | 1 | 1 | 1 | 0 |
| organizational research | 1 | 2 | 1 | 1 | 1 |
| financial performance | 1 | 1 | 1 | 1 | 1 |
| agency theory | 1 | 3 | 1 | 1 | 1 |
| organizational effectiveness | 1 | 4 | 3 | 1 | 1 |
| organizational behavior | 1 | 6 | 2 | 1 | 1 |

Adjacency matrix with sample columns and rows is shown above

## 1.2 - Read the adjacency matrix and convert it into a weighted network

```
g <- graph_from_adjacency_matrix(adjmatrix = adjacency_matrix, mode = 'undirected',weighted = T )
plot(g)
```



```
## reflect weight on the thickness of the edge
E(g)$width <- E(g)$weight # offset=1
plot(g)
```

### 1.3 - Compute node degree and strength

```
#Computing degree for each Node
deg <- degree(g, mode="all")

degree_df = data_frame(Keyword = row.names(adjacency_df), Degree = deg)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
#Computing strength for each node
strength <- strength(g, mode="all")

strength_df = data_frame(Keyword = row.names(adjacency_df), Strength = strength)
```

### 1.4 - Show the top 10 nodes by degree and top 10 nodes by strength

```
#Top 10 nodes by degree
knitr::kable(head(degree_df[order(-degree_df$Degree),], 10))
```

| Keyword | Degree |
|---|---|
| organizational behavior | 166 |
| organizational effectiveness | 104 |
| management science | 102 |
| personnel management | 93 |
| decision making | 90 |
| organizational structure | 74 |
| organizational sociology | 66 |
| strategic planning | 66 |
| industrial management | 64 |
| corporate governance | 62 |

```
#Top 10 nodes by strength
knitr::kable(head(strength_df[order(-strength_df$Strength),], 10))
```

| Keyword | Strength |
|---|---|
| organizational behavior | 265 |
| organizational effectiveness | 144 |
| management science | 136 |
| personnel management | 126 |
| decision making | 112 |
| organizational structure | 107 |
| organizational sociology | 96 |
| corporate governance | 85 |
| industrial management | 84 |
| strategic planning | 80 |

### 1.5- Show the top 10 node pairs by weight

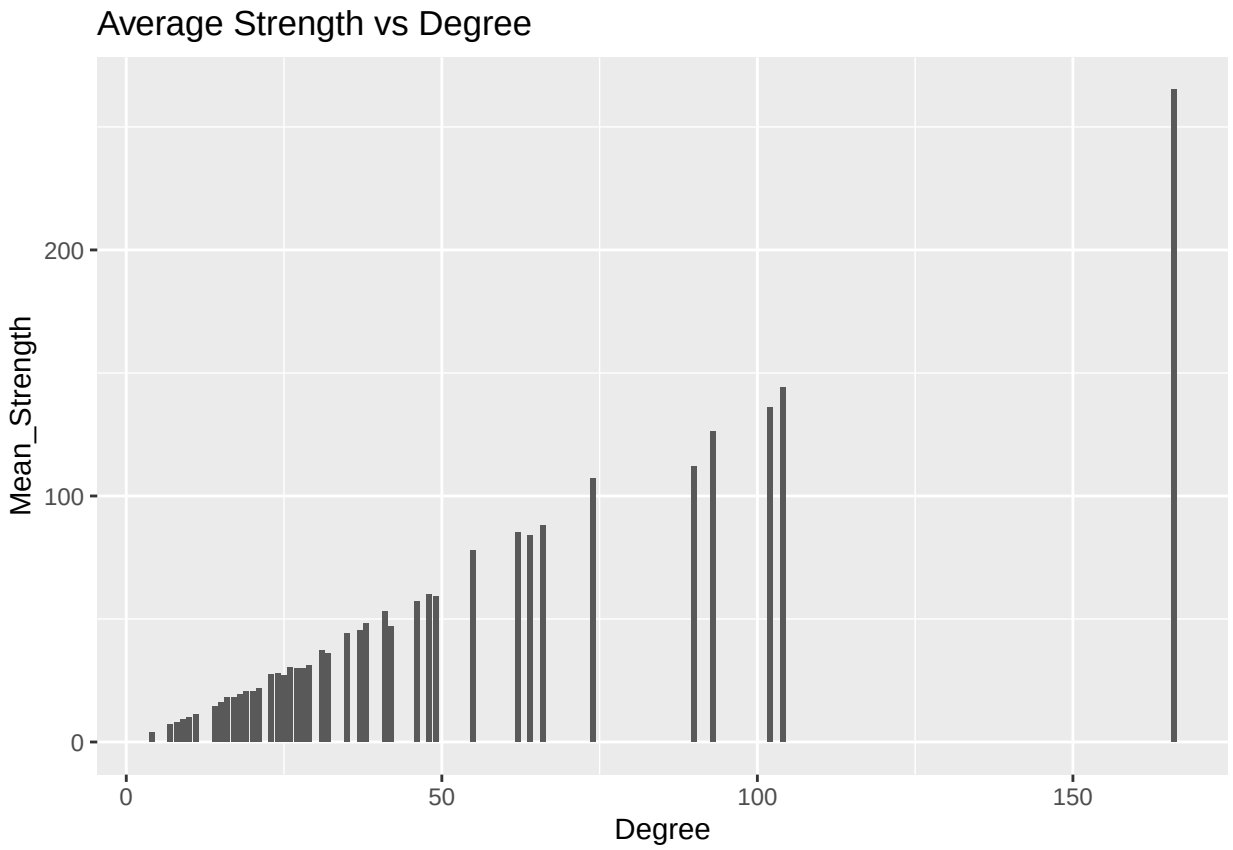
```
node_pairs=data_frame()
for (i in 1:length(rownames(adjacency_df)))
{
  for (j in 1:length(colnames(adjacency_df)))
  {
    if(j>i)
    {
      k1 = rownames(adjacency_matrix)[i]
      k2 = colnames(adjacency_matrix)[j]
      # print(k1)
      # print(k2)
      a= data_frame(keyword1=k1, keyword2 = k2,weight=adjacency_df[i,j])
      node_pairs=rbind(node_pairs,a)
    }
  }
}
```

```
knitr::kable(head(node_pairs[order(-node_pairs$weight),],10))
```

| keyword1 | keyword2 | weight |
|---|---|---|
| organizational effectiveness | organizational behavior | 11 |
| organizational behavior | organizational structure | 9 |
| organizational behavior | personnel management | 8 |
| organizational behavior | management science | 7 |
| organizational sociology | organizational behavior | 6 |
| organizational effectiveness | organizational structure | 6 |
| organizational behavior | corporate governance | 6 |
| organizational behavior | decision making | 6 |
| organizational sociology | teams in the workplace | 5 |
| organizational behavior | industrial relations | 5 |

### 1.6 Plot average strength on y-axis and degree on x-axis

```
## Merge 2 dataframes
final <- merge(degree_df, strength_df, by = 'Keyword')

## Use groupby and find mean strength
final1 <- final %>%
  group_by(Degree) %>%
  summarise(Mean_Strength = mean(Strength))

# Barplot
ggplot(final1, aes(x=Degree, y=Mean_Strength)) +
  geom_bar(stat = "identity") + ggtitle("Average Strength vs Degree")
```




# Task -2

## Reading data

```
tweet_2021_df <- read_csv('2021.csv')
```

## Preprocessing data

Adding extra column of year and serial number, removing unneeded columns, creating simpler form of dataframe "tibble" for each tweet year

```
# Extracting year
tweet_2021_df$year <- format(tweet_2021_df$date, format = "%Y")

# # Combining all tweets into a single dataframe
combined_tweets <- tibble(s_no = c(1:length(tweet_2021_df$tweet)), text=tweet_2021_df$tweet,year=tweet_



combined_tweets<- subset(combined_tweets, year %in% c("2017","2018","2019","2020","2021"))


combined_tweets$year <- as.numeric(combined_tweets$year)
```

### 2.1- Compute word frequencies for each year. Exclude the stop words

```
# Unnesting words in column text to create individual words
combined_tweets_words <- combined_tweets %>%
  unnest_tokens(word, text)


# Group words and year, count words, arrange descending
combined_tweets_words <- combined_tweets_words %>%
  group_by(word,year) %>%
  summarise(count=n()) %>%
  arrange(desc(year),desc(count))
```

```
## `summarise()` has grouped output by 'word'. You can override using the `.groups` argument.
```

```
# List of protocols / web jargon
web_words <- c("t.co", "amp", "https","http")

# List of stop words
stop_words <- sort(c(stopwords(), web_words, "will", "like", "likely", "yes", "yeah", "just", "can", "n
# Removing stop words
combined_tweets_words<- filter(combined_tweets_words, !(word %in%  stop_words))

# Group by year again and count total words / year
combined_tweets_words <- combined_tweets_words %>%
    group_by(year) %>%
   mutate(total= sum(count))
```

**A few rows of this dataframe are shown below:**

```
knitr::kable(head(combined_tweets_words,10))
```

| word | year | count | total |
|---|---|---|---|
| spacex | 2021 | 57 | 4326 |
| erdayastronaut | 2021 | 42 | 4326 |
| tesla | 2021 | 37 | 4326 |
| year | 2021 | 27 | 4326 |
| ppathole | 2021 | 23 | 4326 |
| wholemarsblog | 2021 | 22 | 4326 |
| high | 2021 | 20 | 4326 |
| nasaspaceflight | 2021 | 19 | 4326 |
| great | 2021 | 18 | 4326 |
| next | 2021 | 18 | 4326 |

### 2.2 - Show top 10 words (for each year) by the highest value of word frequency

```
top_10_words_frequency <- combined_tweets_words %>%
  group_by(year) %>%
  mutate(frequency= round(count/total,5) ) %>%
  slice(1:10)
```

**Top 10 words for each year, sorted by highest value of frequency**

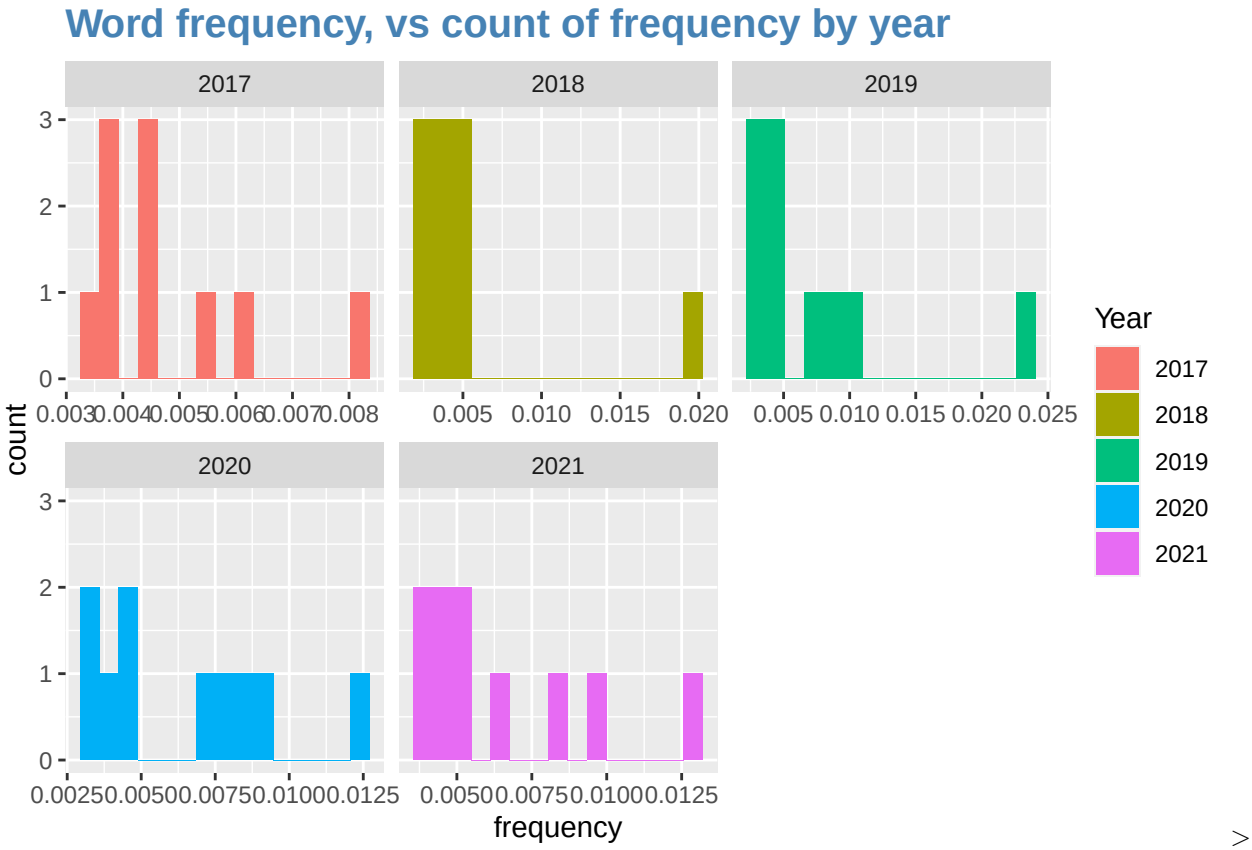
```
knitr::kable(head(top_10_words_frequency,50))
```

| word | year | count | total | frequency |
|---|---|---|---|---|
| tesla | 2017 | 79 | 9617 | 0.00821 |
| model | 2017 | 59 | 9617 | 0.00613 |
| spacex | 2017 | 52 | 9617 | 0.00541 |
| first | 2017 | 43 | 9617 | 0.00447 |
| next | 2017 | 43 | 9617 | 0.00447 |
| 3 | 2017 | 42 | 9617 | 0.00437 |
| boring | 2017 | 36 | 9617 | 0.00374 |
| right | 2017 | 36 | 9617 | 0.00374 |
| car | 2017 | 35 | 9617 | 0.00364 |
| rocket | 2017 | 33 | 9617 | 0.00343 |
| tesla | 2018 | 452 | 22693 | 0.01992 |
| car | 2018 | 120 | 22693 | 0.00529 |
| 3 | 2018 | 112 | 22693 | 0.00494 |
| model | 2018 | 98 | 22693 | 0.00432 |
| spacex | 2018 | 89 | 22693 | 0.00392 |
| don't | 2018 | 74 | 22693 | 0.00326 |
| people | 2018 | 73 | 22693 | 0.00322 |
| even | 2018 | 65 | 22693 | 0.00286 |
| next | 2018 | 64 | 22693 | 0.00282 |
| great | 2018 | 62 | 22693 | 0.00273 |
| tesla | 2019 | 580 | 24798 | 0.02339 |
| erdayastronaut | 2019 | 267 | 24798 | 0.01077 |
| spacex | 2019 | 215 | 24798 | 0.00867 |
| flcnhvy | 2019 | 185 | 24798 | 0.00746 |
| great | 2019 | 97 | 24798 | 0.00391 |
| starship | 2019 | 94 | 24798 | 0.00379 |
| teslaownerssv | 2019 | 93 | 24798 | 0.00375 |
| teslarati | 2019 | 88 | 24798 | 0.00355 |
| high | 2019 | 75 | 24798 | 0.00302 |
| 3 | 2019 | 73 | 24798 | 0.00294 |
| tesla | 2020 | 349 | 28261 | 0.01235 |
| erdayastronaut | 2020 | 250 | 28261 | 0.00885 |
| flcnhvy | 2020 | 244 | 28261 | 0.00863 |
| ppathole | 2020 | 225 | 28261 | 0.00796 |
| spacex | 2020 | 208 | 28261 | 0.00736 |
| great | 2020 | 130 | 28261 | 0.00460 |
| thirdrowtesla | 2020 | 123 | 28261 | 0.00435 |
| teslaownerssv | 2020 | 116 | 28261 | 0.00410 |
| teslarati | 2020 | 95 | 28261 | 0.00336 |
| high | 2020 | 91 | 28261 | 0.00322 |
| spacex | 2021 | 57 | 4326 | 0.01318 |
| erdayastronaut | 2021 | 42 | 4326 | 0.00971 |
| tesla | 2021 | 37 | 4326 | 0.00855 |
| year | 2021 | 27 | 4326 | 0.00624 |
| ppathole | 2021 | 23 | 4326 | 0.00532 |
| wholemarsblog | 2021 | 22 | 4326 | 0.00509 |
| high | 2021 | 20 | 4326 | 0.00462 |
| nasaspaceflight | 2021 | 19 | 4326 | 0.00439 |
| great | 2021 | 18 | 4326 | 0.00416 |
| next | 2021 | 18 | 4326 | 0.00416 |

### 2.3 - Plot histogram of word frequencies for each year

```
plot_theme = theme(plot.title = element_text(colour="steelblue",family = "Helvetica", face = "bold", si

ggplot(top_10_words_frequency, aes(frequency, fill=factor(year))) +
  # Number of bins
  geom_histogram( bins = 15) +
  # Wrap plots in 3 columns, change legend to year
  facet_wrap(~year, ncol = 3, scales = "free_x","free_y") + scale_fill_discrete(name="Year") +
  # Add title and theme
  ggtitle("Word frequency, vs count of frequency by year") + plot_theme
```

```
## Warning: Coercing ‘nrow‘ to be an integer.

## Warning in sanitise_dim(nrow): NAs introduced by coercion

## Warning: ‘nrow‘ is missing or less than 1 and will be treated as NULL.
```

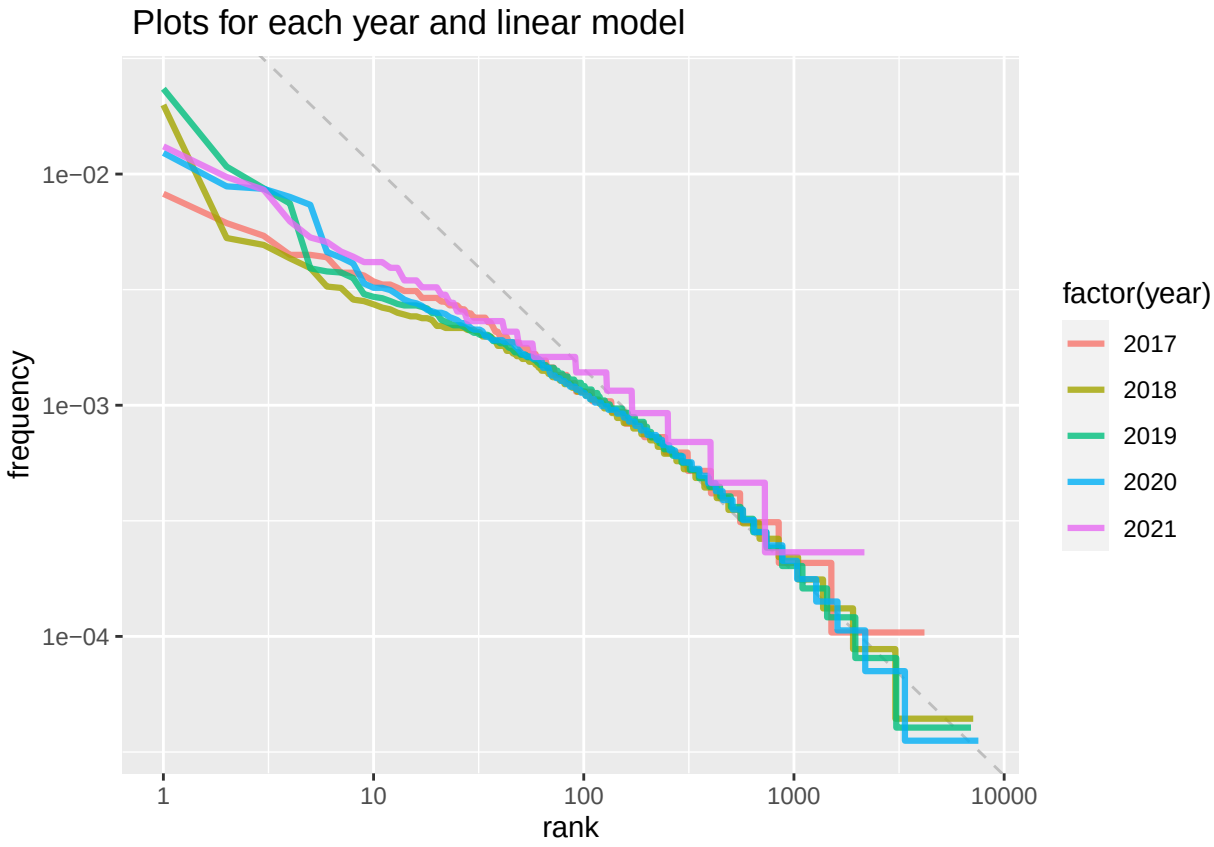



>

As seen in above plots, two type of words dominate the histograms -> The one which occur the least which are towards the left of the histogram, but are high in count, which might be phrases or mention of other user twitter handlenames, and the ones which are mentioned the most, which are generally words related to businesses operated by Elon such as Tesla, spacex, and his business related terminology such as “astronaut”,“car”, etc. This pattern is repeated for each year.

### 2.4 - Use Zipf's law and plot log-log plots of word frequencies and rank for each year

```
# New dataframe to avoid data consistency issues
tweet_zipfs <- combined_tweets_words %>%
  group_by(year) %>%
  mutate(frequency= count/total, rank = row_number() )

# Fitting linear model in log frequency and rank
linear_model <- lm(log10(frequency) ~ log10(rank), data = tweet_zipfs)



  ggplot(tweet_zipfs, aes(rank, frequency, color = factor(year))) +
    scale_fill_discrete(name="Year")  +
  geom_abline(intercept = linear_model$coefficients[1], slope = linear_model$coefficients[2],
              color = "grey", linetype = 2) +
  geom_line(size = 1.1, alpha = 0.8) +
  scale_x_log10() +
  scale_y_log10()  + ggtitle(" Plots for each year and linear model ")
```




Above plot reflects that each year is plotted almost the same and follows zipfs low ( rank is inversely proportional to frequency of words)

### 2.5 - Create bigram network graphs for each year

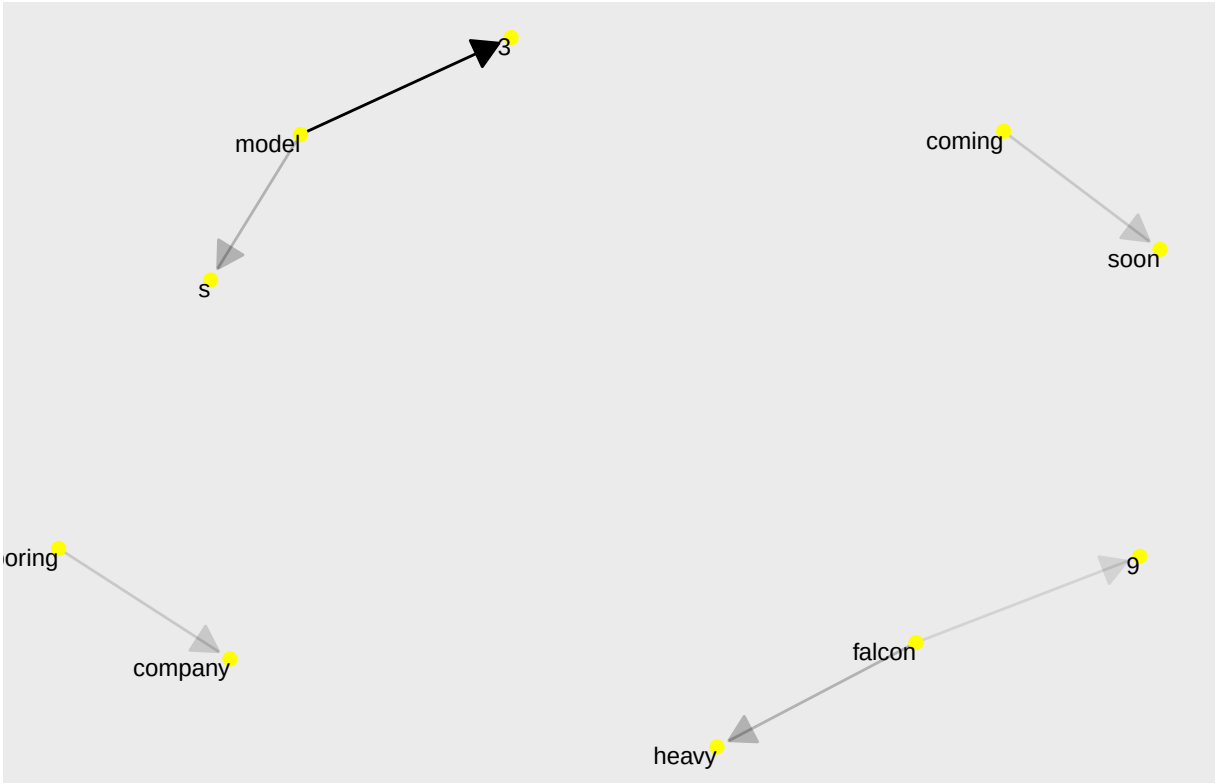
```
# Unnest words to form bigrams
tweet_bigrams <- combined_tweets %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2)

# Separate and filter stop words
tweet_bigrams_separated <- tweet_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

tweet_bigrams_filtered <- tweet_bigrams_separated %>%
  filter(!word1 %in% stop_words) %>%
  filter(!word2 %in% stop_words)

# Group by year, count words and sort
tweet_bigram_counts <- tweet_bigrams_filtered %>%
  group_by(year) %>%
  count(word1, word2, sort = TRUE)

# Drop unneeded columns andna
tweet_bigram_counts <- tweet_bigram_counts[, c(2,3,4,1)] %>% drop_na()

# Subset to create yearly bigram graphs

tweet_bigram_2017 <- tweet_bigram_counts %>%
  subset(year==2017)

tweet_bigram_2018 <- tweet_bigram_counts %>%
  subset(year==2018)

tweet_bigram_2019 <- tweet_bigram_counts %>%
  subset(year==2019)

tweet_bigram_2020 <- tweet_bigram_counts %>%
  subset(year==2020)

tweet_bigram_2021 <- tweet_bigram_counts %>%
  subset(year==2021)

# Filtering each bigram per year based on frequently occuring bigrams
# Taking only relevant columns
tweet_bigram_graph_2017 <- tweet_bigram_2017[,-4] %>%
  filter(n > 10) %>%
  graph_from_data_frame()

tweet_bigram_graph_2018 <- tweet_bigram_2018[,-4] %>%
  filter(n > 10) %>%
  graph_from_data_frame()

tweet_bigram_graph_2019 <- tweet_bigram_2019[,-4] %>%
  filter(n > 10) %>%
  graph_from_data_frame()
```

```
tweet_bigram_graph_2020 <- tweet_bigram_2020[,-4] %>%
  filter(n > 10) %>%
  graph_from_data_frame()

tweet_bigram_graph_2021 <- tweet_bigram_2021[,-4] %>%
  filter(n > 5) %>%
  graph_from_data_frame()

# Defining arrow for graphs
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))
```

**Bigram graph for year 2017**

```
ggraph(tweet_bigram_graph_2017, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "yellow", size = 2) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1, size=3) +
  ggtitle("Bigram word graph for Elon musks tweets in 2017")
```

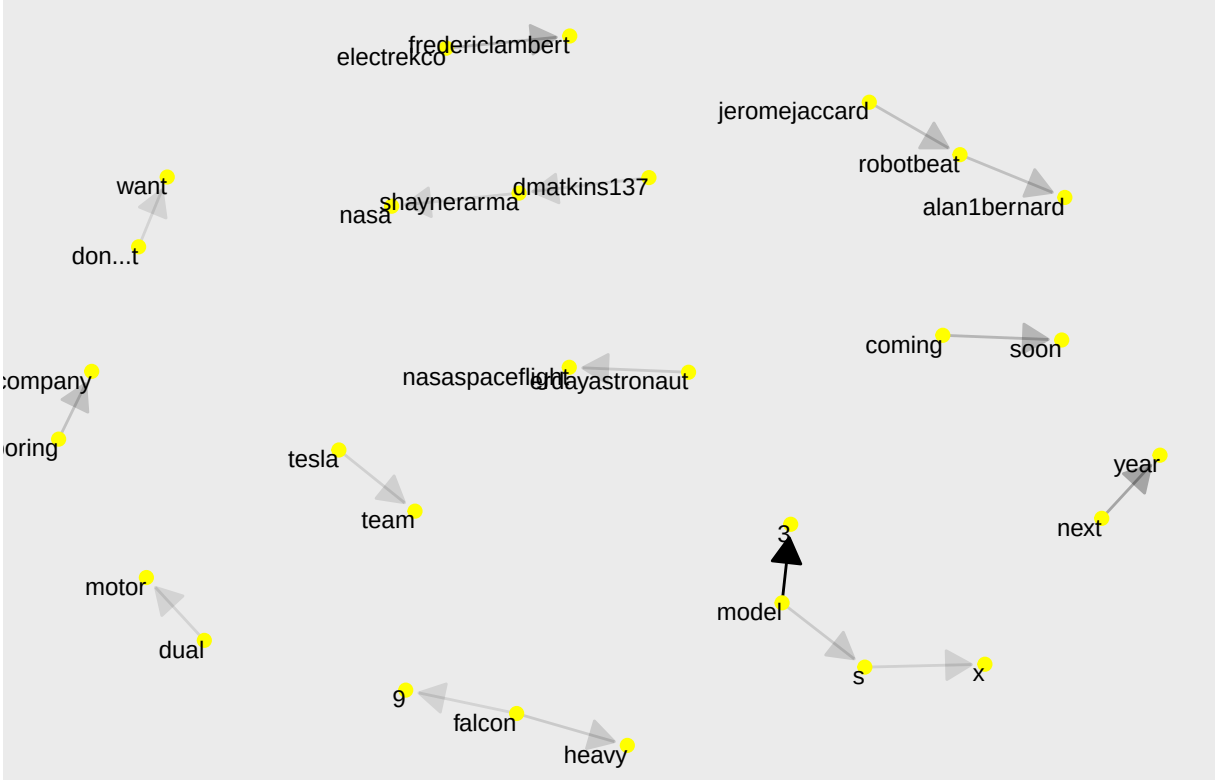
Bigram word graph for Elon musks tweets in 2017




**Bigram graph for year 2018**

```
ggraph(tweet_bigram_graph_2018, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "yellow", size = 2) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1, size=3) +
  ggtitle("Bigram word graph for Elon musks tweets in 2018")
```

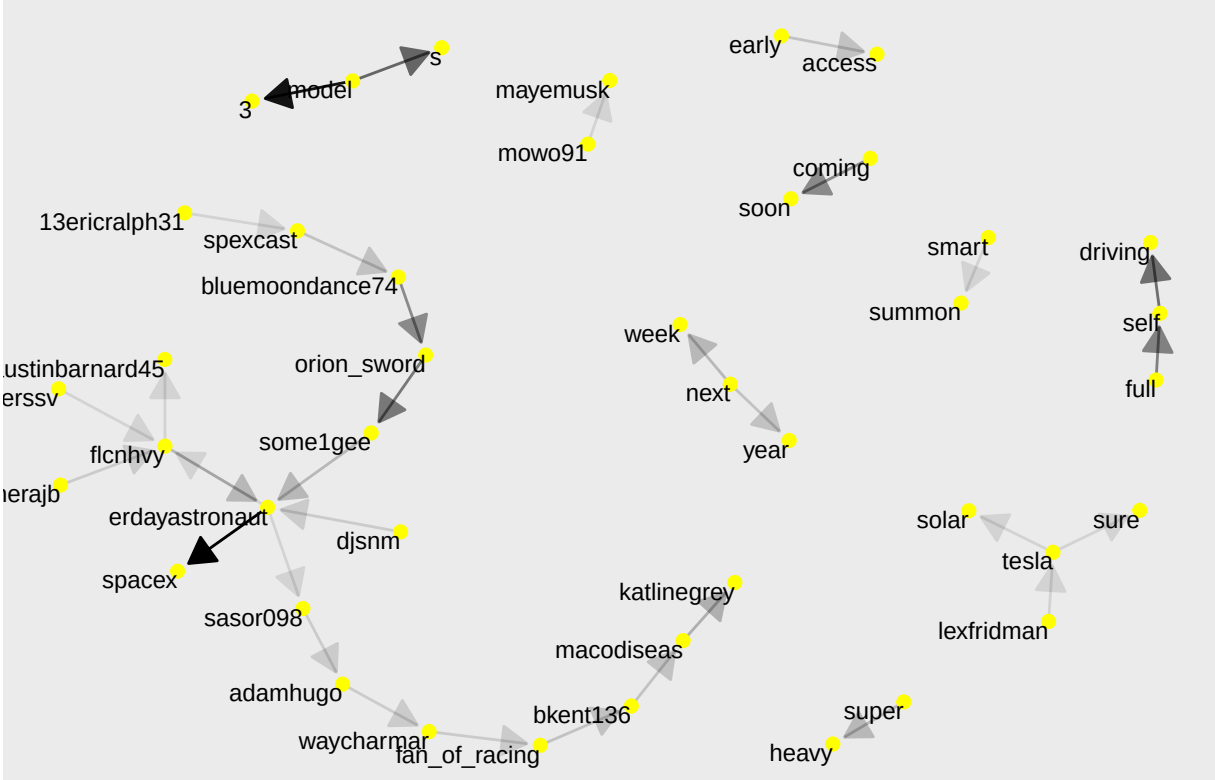
Bigram word graph for Elon musks tweets in 2018




**Bigram graph for year 2019**

```
ggraph(tweet_bigram_graph_2019, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "yellow", size = 2) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1, size=3) +
  ggtitle("Bigram word graph for Elon musks tweets in 2019")
```

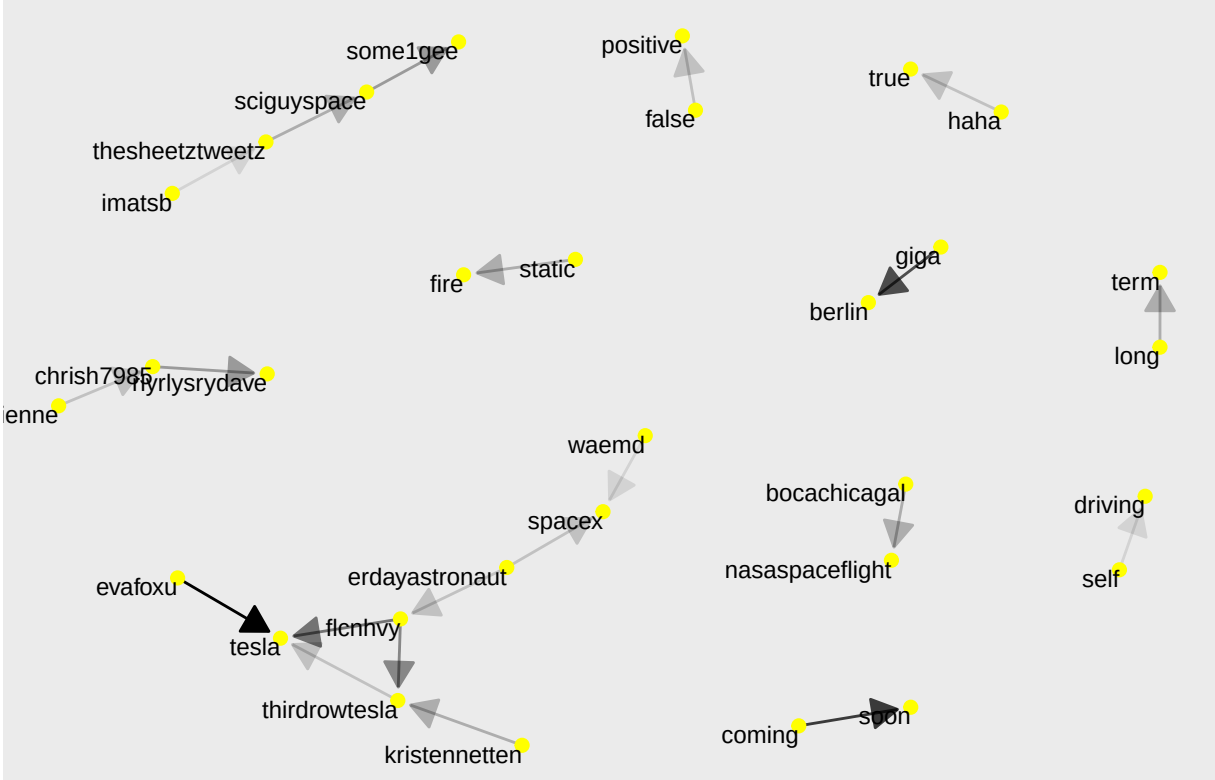
Bigram word graph for Elon musks tweets in 2019




**Bigram graph for year 2020**

```
ggraph(tweet_bigram_graph_2020, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "yellow", size = 2) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1, size=3) +
  ggtitle("Bigram word graph for Elon musks tweets in 2020")
```

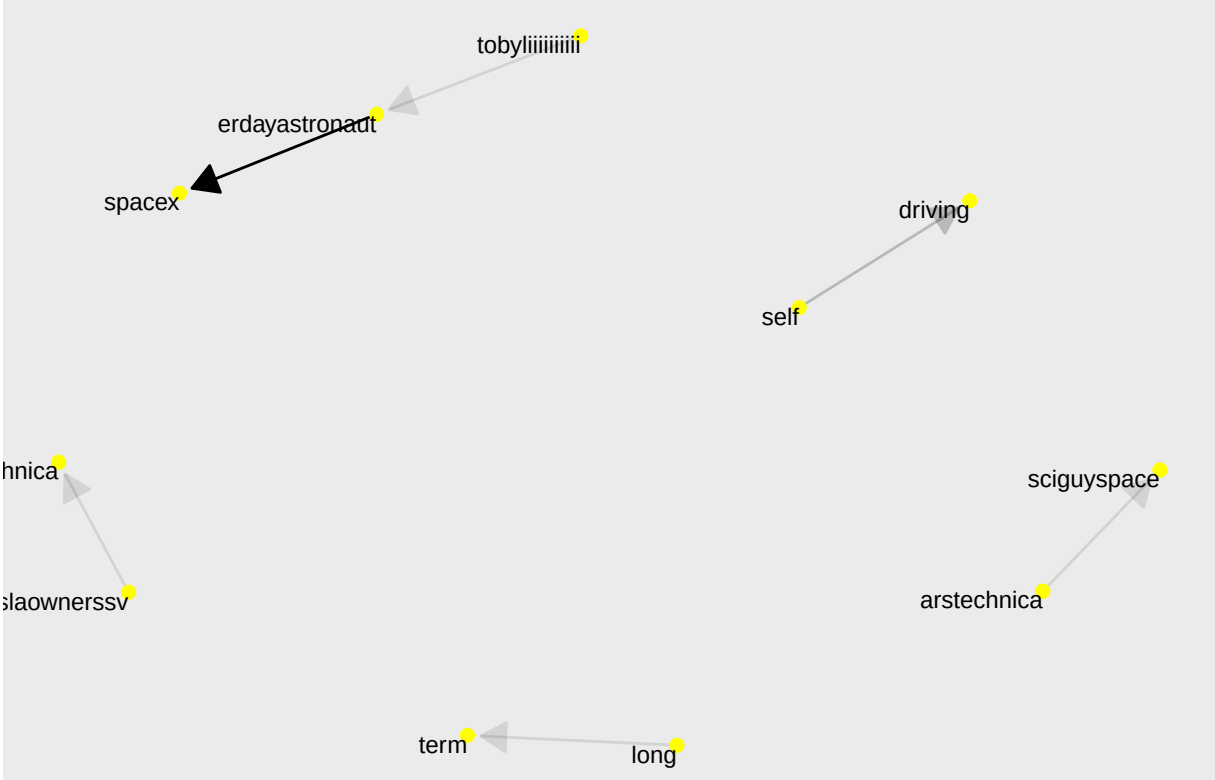
Bigram word graph for Elon musks tweets in 2020




**Bigram graph for year 2021**

```
ggraph(tweet_bigram_graph_2021, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "yellow", size = 2) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1, size=3) +
  ggtitle("Bigram word graph for Elon musks tweets in 2021")
```

Bigram word graph for Elon musks tweets in 2021




As we can see from the above plots, some words that **commonly occur together across different years are: "boring"-"company","climate"-"change","electric"-cars"**.

**Kindly note for plotting, value of n (frequency of bigrams for each year) was taken by observation of the data and set as a threhhold**