Yelp Big Data and Text Analysis with 7 million observations

Varun

2/14/2020

```
library(jsonlite)
library(ggplot2)
library(dplyr)
library(broom)
library(tidyr)
library(readxl)
library(ggthemes)
library(DT)
library(scales)
library(ggpubr)
library(lubridate)
library(ggrepel)
library(gridExtra)
library(GGally)
library(corrplot)
library(dplyr)
library('wordcloud')
library(tidytext)
library(tidyverse)
library(tm)
library(textdata)
library(DT)
library(igraph)
library(ggraph)
library(leaflet)
```

```
yelp_business <- read.csv("yelp_academic_dataset_business.json.csv")
```

```
str(yelp_business)
```

```
## 'data.frame':    192609 obs. of  10 variables:
##  $ business_id : Factor w/ 192609 levels "__1uG7MLxWGFIv2fCGPiQQ",..: 11258 138035 75948 179781 80935 243
45 22302 73708 181325 18984 ...
##  $ name        : Factor w/ 145046 levels " L & A Insurance Services",..: 8990 43073 86340 46032 102406 1
30877 41775 121879 139386 92242 ...
##  $ full_address: Factor w/ 151977 levels "","", #107","", 108 3 Avenue SW",..: 62546 65797 2882 28551 88822
149740 41910 94178 111498 48605 ...
##  $ hours       : Factor w/ 51567 levels "","{'Friday': '0:0-0:0', 'Saturday': '0:0-0:0', 'Sunday': '0:0-
0:0'}",..: 1 37940 24939 34158 31860 41266 1 10187 47172 50338 ...
##  $ open        : int  0 1 1 1 1 1 1 1 0 1 ...
##  $ categories  : Factor w/ 93386 levels "","3D Printing, Graphic Design, Local Services, Professional Ser
vices, Printing Services",..: 38896 85437 87285 49686 67370 79649 11667 41015 59564 12268 ...
##  $ city        : Factor w/ 1204 levels "","110 Las Vegas",..: 802 607 166 353 166 607 129 464 347 308 ...
##  $ state       : Factor w/ 36 levels "AB","AK","AL",..: 5 23 16 5 16 23 1 20 5 22 ...
##  $ review_count: int  5 128 170 3 4 3 7 3 8 8 ...
##  $ stars       : num  3 2.5 4 5 4 2.5 3.5 3.5 5 4.5 ...
```

```
yelp_user <- read.csv("yelp_academic_dataset_user.json.csv")
```

```
str(yelp_user)
```

```
## 'data.frame':    1637138 obs. of  11 variables:
##  $ user_id      : Factor w/ 1637138 levels "___DPmKJsBF2X6ZKgAeGqg",..: 875812 176631 370525 474609 94945
3 69864 1296063 838673 1280737 1007019 ...
##  $ name         : Factor w/ 124916 levels " Bernard"," Bill",..: 90969 49513 25127 5655 79786 71143 5722
1 5329 109382 113546 ...
##  $ review_count : int  95 33 16 17 361 214 1122 6 859 124 ...
##  $ average_stars: num  4.03 3.63 3.71 4.85 4.08 4.2 4.39 4.33 4.21 4.53 ...
##  $ cool_votes   : int  25 16 10 14 665 3048 15319 1 1244 185 ...
##  $ funny_votes  : int  17 22 8 4 279 2424 19356 0 693 70 ...
##  $ useful_votes : int  84 48 28 30 1114 3475 13311 1 1630 202 ...
##  $ friends      : Factor w/ 933769 levels "___nmHY7ad0QeXLbNpX6iQ, rMtqiykhsz2YrBcqWV2M6A, -5sJVQlWUw3cya
xjUiti4A, JO-YITQ1rV3waB-h_Ze96A, diWSCN9RrnUxVfR"| __truncated__,..: 237693 476841 95932 700279 532799 1898
66 686177 264856 257254 555683 ...
##  $ elite        : Factor w/ 756 levels "","2006","2006,2007",..: 744 1 1 1 745 745 9 1 8 1 ...
##  $ yelping_since: Factor w/ 1631010 levels "2004-10-12 08:40:43",..: 733083 604138 730586 851760 740410 3
04232 1517 861685 483 53718 ...
##  $ fans         : int  5 4 0 5 39 186 696 0 57 15 ...
```

```
yelp_review <- read.csv("yelp_academic_dataset_review.json.csv")
```

```
str(yelp_review)
```

```
## 'data.frame':    6685900 obs. of  9 variables:
##  $ review_id  : Factor w/ 6685900 levels "___-Bw8LtQgezPiN9xJWaQ",..: 4608584 2611062 501692 6359894 3182
04 2360978 2540153 1077163 4748468 4961828 ...
##  $ user_id    : Factor w/ 1637138 levels "___DPmKJsBF2X6ZKgAeGqg",..: 684209 1582140 977878 470157 126714
1 1436822 795542 467311 1246916 1000576 ...
##  $ business_id: Factor w/ 192606 levels "__1uG7MLxWGFIv2fCGPiQQ",..: 159304 120343 173359 87401 42427 650
61 15916 191750 42537 126093 ...
##  $ stars      : num  1 5 5 5 1 4 3 1 2 3 ...
##  $ date       : Factor w/ 6552820 levels "2004-10-12 10:13:32",..: 1192071 4261842 4084825 5438211 550712
9 1059956 3522157 321771 2254573 730748 ...
##  $ text       : Factor w/ 6668738 levels " \nBrianna was simply amazing on our wedding (Sept 24, 2016). W
e had the trial a few days prior even though I ha"| __truncated__,..: 5805790 1836967 2316420 6347141 578218
7 2936576 5810801 5489712 2799265 3295682 ...
##  $ useful     : int  6 0 3 0 7 0 5 3 1 1 ...
##  $ funny      : int  1 0 0 0 0 0 4 1 0 0 ...
##  $ cool       : int  0 0 0 0 0 0 5 1 0 1 ...
```

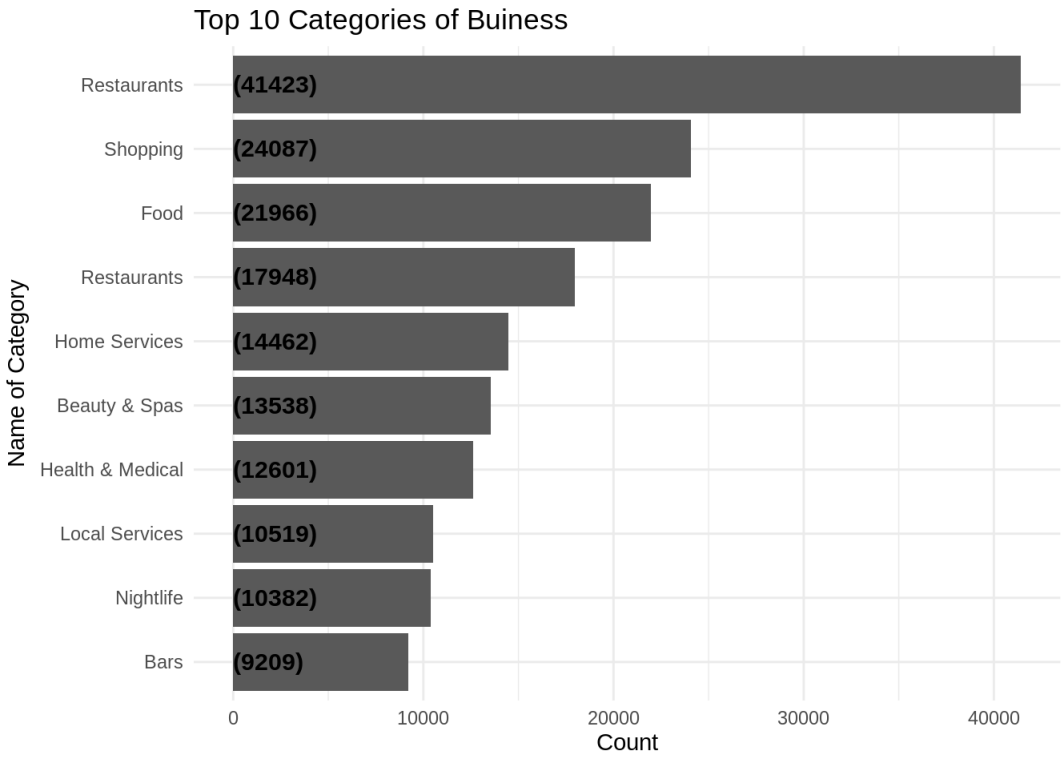
Most Popular Business Categories

The most popular categories of business that are in yelp

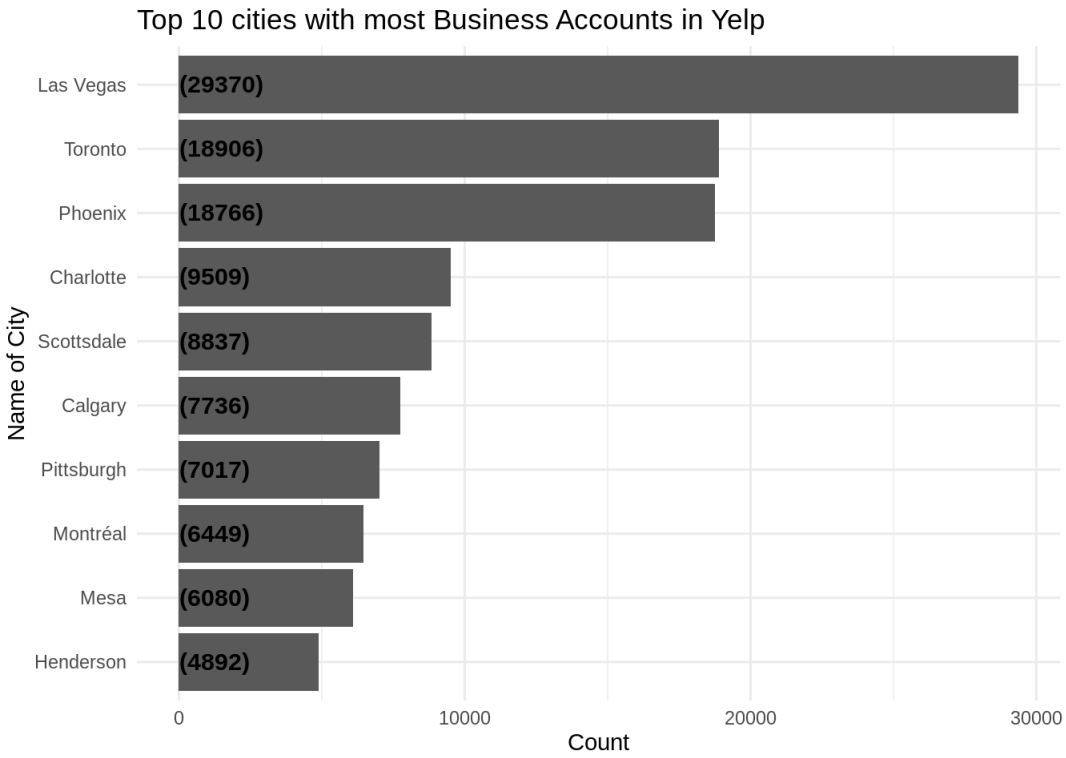
```
library(stringr)
categories <- str_split(yelp_business$categories, ",")
categories <- as.data.frame(unlist(categories))
colnames(categories) <- c("Names")
```

Top 10 cities with most Busniess Accounts in Yelp

```
categories %>%
  group_by(Names) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count)) %>%
  ungroup() %>%
  mutate( Names = reorder(Names,Count)) %>%
  head(10) %>%


  ggplot( aes (Names, Count))+
  geom_bar(stat = "identity")+
  geom_text(aes(x = Names, y = 1, label = paste0("(",Count,")",sep="")),
            hjust=0, vjust=.5, size = 4, colour = 'black',
            fontface = 'bold') +
  labs(x = "Name of Category", y = "Count", title = "Top 10 Categories of Buiness")+
  coord_flip()+
  scale_colour_brewer(palette="Set1")+
  theme_minimal()
```




```
yelp_business %>%
group_by(city) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count)) %>%
  ungroup() %>%
  mutate( city = reorder(city,Count)) %>%
  head(10) %>%


  ggplot( aes (city, Count))+
  geom_bar(stat = "identity")+
  geom_text(aes(x = city, y = 1, label = paste0("(",Count,")",sep="")),
            hjust=0, vjust=.5, size = 4, colour = 'black',
            fontface = 'bold') +
  labs(x = "Name of City", y = "Count", title = "Top 10 cities with most Business Accounts in Yelp")+
  coord_flip()+
  scale_colour_brewer(palette="Set1")+
  theme_minimal()
```



Business with most Five star reviews from Users

```{r}{r,message = FALSE, warning=FALSE} five_star <- yelp_review %>% filter(stars == 5) %>% group_by(business_id) %>% summarise(Count = n()) %>% arrange(desc(Count)) %>% ungroup() %>% mutate(BusinessID = reorder(business_id,Count)) %>% head(10)

five_star <- inner_join(five_star,yelp_business, by = “business_id”) five_star %>% mutate(name = reorder(name,Count)) %>% ggplot(aes (name, Count))+ geom_bar(stat = “identity”)+ geom_text(aes(x = name, y = 1, label = paste0(“(”,Count,“)”,sep=""“)), hjust=0, vjust=.5, size = 4, colour = ‘black’, fontface = ‘bold’) + labs(x =”Name of Business“, y =”Count“, title =”Business with most Five star reviews from Users“)+ coord_flip()+ scale_colour_brewer(palette=”Set1")+ theme_minimal()

````

Word Cloud of Mon Ami Gabi

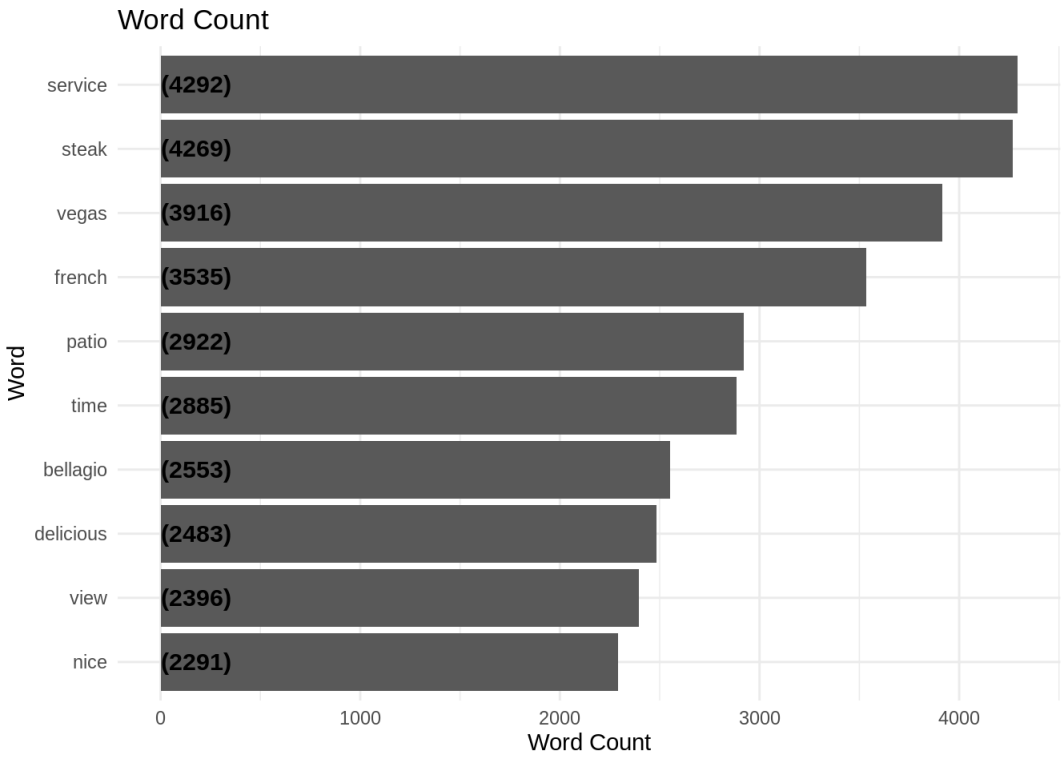
By tokenising single word from review dataset for Mon Ami Gabi, we now look at what common words were used in reviews by users.

```
CreateWordCloud = function(train) {
  train %>%
    unnest_tokens(word,text) %>%
    filter(!word %in% stop_words$word)%>%
    count(word,sort = TRUE) %>%
  ungroup()  %>%
  head(30) %>%

  with(wordcloud(word, n, max.words = 30,colors=brewer.pal(8, "Dark2")))
}
yelp_review["text"] <- lapply(yelp_review["text"], as.character)

CreateWordCloud(yelp_review %>%
                  filter(business_id == "4JNXUYY8wbaaDmk3BPzlWw"))
```

The words steak, service, vegas,french,patio,bellagio,delicious, nice are the words, which have been used very frequently in the reviews.Note that if we choose a word which is not food related, it is Service and we will see in the subsequent sections of sentiment analysis and topic modelling.

Top Ten most common Words of the business “Mon Ami Gabi”

```
yelp_review %>%
  filter(business_id == "4JNXUYY8wbaaDmk3BPzlWw") %>%
  unnest_tokens(word,text) %>%
  filter(!word %in% stop_words$word) %>%
  filter(!word %in% c('food','restaurant')) %>%
  count(word,sort = TRUE) %>%
  ungroup() %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  head(10) %>%

  ggplot(aes(x = word,y = n)) +
  geom_bar(stat='identity') +
  geom_text(aes(x = word, y = 1, label = paste0("(",n,")",sep="")),
            hjust=0, vjust=.5, size = 4, colour = 'black',
            fontface = 'bold') +
  labs(x = 'Word', y = 'Word Count',
       title = 'Word Count') +
  coord_flip() +
  theme_minimal()
```



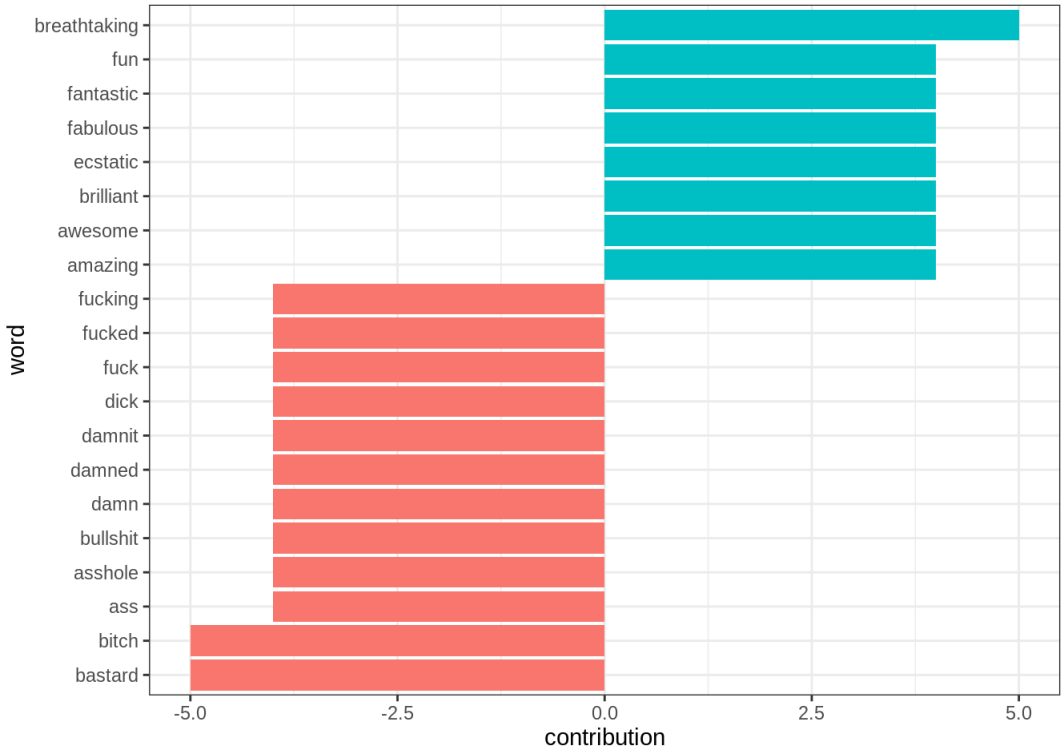
Sentiment Analysis - Postive and Not So Postive Words of “Mon Ami Gabi”

```
contributions <- yelp_review %>%
  filter(business_id == "4JNXUYY8wbaaDmk3BPzlWw") %>%
  unnest_tokens(word, text) %>%
  count(word,sort = TRUE) %>%
  ungroup() %>%

  inner_join(get_sentiments("afinn"), by = "word") %>%
  group_by(word) %>%
  summarize(occurences = n()
            )
contributions
```

```
## # A tibble: 1,177 x 2
##    word         occurences
##    <chr>             <int>
##  1 ability               1
##  2 absorbed              1
##  3 abused                1
##  4 accept                1
##  5 accepted              1
##  6 accepting             1
##  7 accepts               1
##  8 accident              1
##  9 accidental            1
## 10 accidentally          1
## # … with 1,167 more rows
```

knitr::knit_exit()

```
positiveWordsBarGraph <- function(SC) {
  contributions <- SC %>%
    unnest_tokens(word,text) %>%
    count(word,sort = TRUE) %>%
    ungroup() %>%

    inner_join(get_sentiments("afinn"), by = "word") %>%
    group_by(word) %>%
    summarize(occurences = n(),
              contribution = sum(value))

  contributions %>%
    top_n(20, abs(contribution)) %>%
    mutate(word = reorder(word, contribution)) %>%
    head(20) %>%
    ggplot(aes(word, contribution, fill = contribution > 0)) +
    geom_col(show.legend = FALSE) +
    coord_flip() + theme_bw()
}

positiveWordsBarGraph(yelp_review %>%
                        filter(business_id == "4JNXUYY8wbaaDmk3BPzlWw"))
```



Calculate Sentiment for the reviews

We calculate the sentiment scores for all the reviews using the AFINN sentiment lexicon. We display the Top Six sentiments here.

```
calculate_sentiment <- function(review_text)
{
  sentiment_lines  =  review_text %>%

                  unnest_tokens(word, text) %>%
                  inner_join(get_sentiments("afinn"), by = "word") %>%
                  group_by(review_id) %>%
                  summarize(sentiment = mean(value),words = n()) %>%
                  ungroup() %>%
                  filter(words >= 5)

  return(sentiment_lines)

}

sentiment_lines = calculate_sentiment(yelp_review %>%
                        filter(business_id == "4JNXUYY8wbaaDmk3BPzlWw"))
head(sentiment_lines)
```

```
## # A tibble: 6 x 3
##   review_id              sentiment words
##   <fct>                      <dbl> <int>
## 1 _1N4hvnYpeOM7WKGIVHulQ      2.86     7
## 2 _1XPTEzPVMoRkhVwOjVJ9g      1.30    23
## 3 _3EPByOAmNyX1enMV5EJFA      2.4     15
## 4 _3pnt6kJnptmMY2NoIspuw      1.7     10
## 5 _43hPBEjsPp4oYxcC9ieCQ      0       10
## 6 _47c2Tfu5LgwQdNo5U5mEA      1.92    12
```

Top ten negetive reviews

```
display_neg_sentiments <- function(sentiment_lines,review_text)
{
  neg_sentiment_lines = sentiment_lines %>%
  arrange(desc(sentiment))  %>%
  top_n(-10, sentiment) %>%
  inner_join(review_text, by = "review_id") %>%
  select(date,sentiment,text)

datatable(neg_sentiment_lines, style="bootstrap", class="table-condensed", options = list(dom = 'tp',scrollX
= TRUE))
}

display_neg_sentiments(sentiment_lines,yelp_review %>%
                        filter(business_id == "4JNXUYY8wbaaDmk3BPzlWw"))
```

```
## Warning in instance$preRenderHook(instance): It seems your data is too big
## for client-side DataTables. You may consider server-side processing: https://
## rstudio.github.io/DT/server.html
```

Top ten positive review

```
display_neg_sentiments <- function(sentiment_lines,review_text)
{
  neg_sentiment_lines = sentiment_lines %>%
  arrange(desc(sentiment))  %>%
  top_n(10, sentiment) %>%
  inner_join(review_text, by = "review_id") %>%
  select(date,sentiment,text)

datatable(neg_sentiment_lines, style="bootstrap", class="table-condensed", options = list(dom = 'tp',scrollX
= TRUE))
}

display_neg_sentiments(sentiment_lines,yelp_review %>%
                        filter(business_id == "4JNXUYY8wbaaDmk3BPzlWw"))
```

```
## Warning in instance$preRenderHook(instance): It seems your data is too big
## for client-side DataTables. You may consider server-side processing: https://
## rstudio.github.io/DT/server.html
```

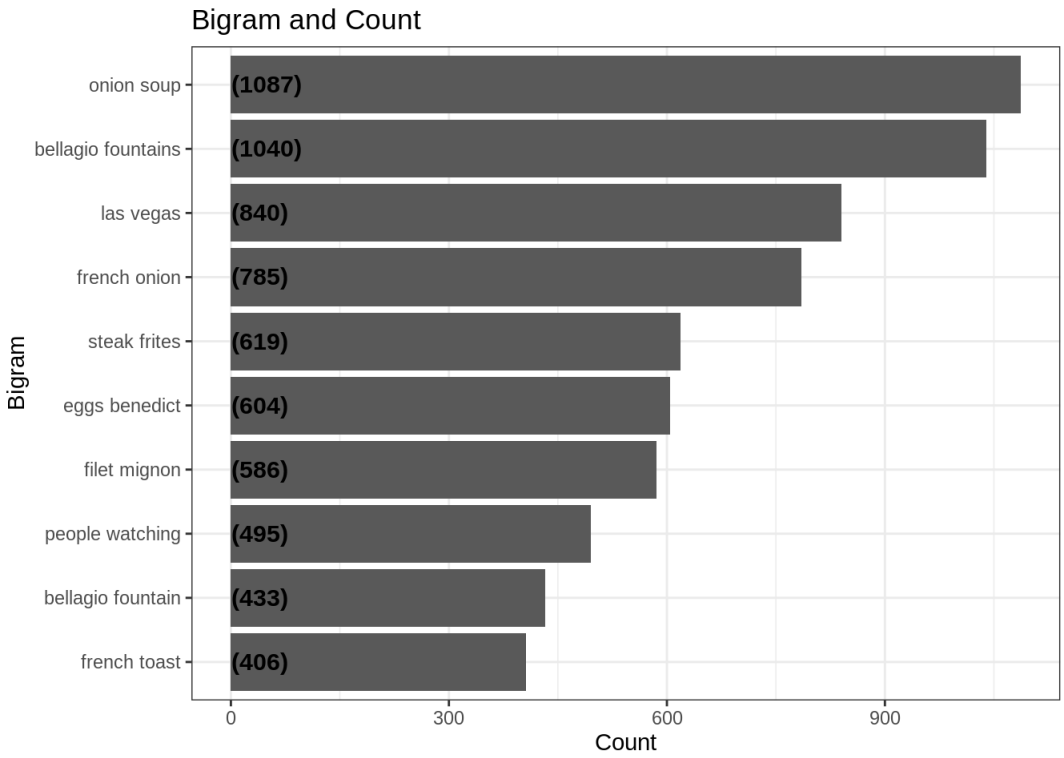
Most Common Bigrams of “Mon Ami Gabi”

A Bigram is a collection of Two words. We examine the most common Bigrams and plot them in a bar plot.

```
count_bigram <- function(dataset) {
  dataset %>%
    unnest_tokens(bigram,text,token = "ngrams", n= 2)%>%
    separate(bigram, c("word1", "word2"), sep = " ") %>%
    filter(!word1 %in% stop_words$word,
           !word2 %in% stop_words$word) %>%
    count(word1, word2,sort = TRUE)

  }
set.seed(2016)
  a <- grid::arrow(type = "closed", length = unit(.15, "inches"))


yelp_review %>%
  filter(business_id == "4JNXUYY8wbaaDmk3BPzlWw") %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2) %>%
  select(bigram,review_id) %>%
  head(10)
```

```
##                 bigram              review_id
## 1             again my _0IY4DAS5x2JcoZdHE-c7g
## 2          my favorite _0IY4DAS5x2JcoZdHE-c7g
## 3       favorite vegas _0IY4DAS5x2JcoZdHE-c7g
## 4     vegas resturaunt _0IY4DAS5x2JcoZdHE-c7g
## 5  resturaunt directly _0IY4DAS5x2JcoZdHE-c7g
## 6      directly across _0IY4DAS5x2JcoZdHE-c7g
## 7          across from _0IY4DAS5x2JcoZdHE-c7g
## 8             from the _0IY4DAS5x2JcoZdHE-c7g
## 9         the bellagio _0IY4DAS5x2JcoZdHE-c7g
## 10  bellagio fountains _0IY4DAS5x2JcoZdHE-c7g
```

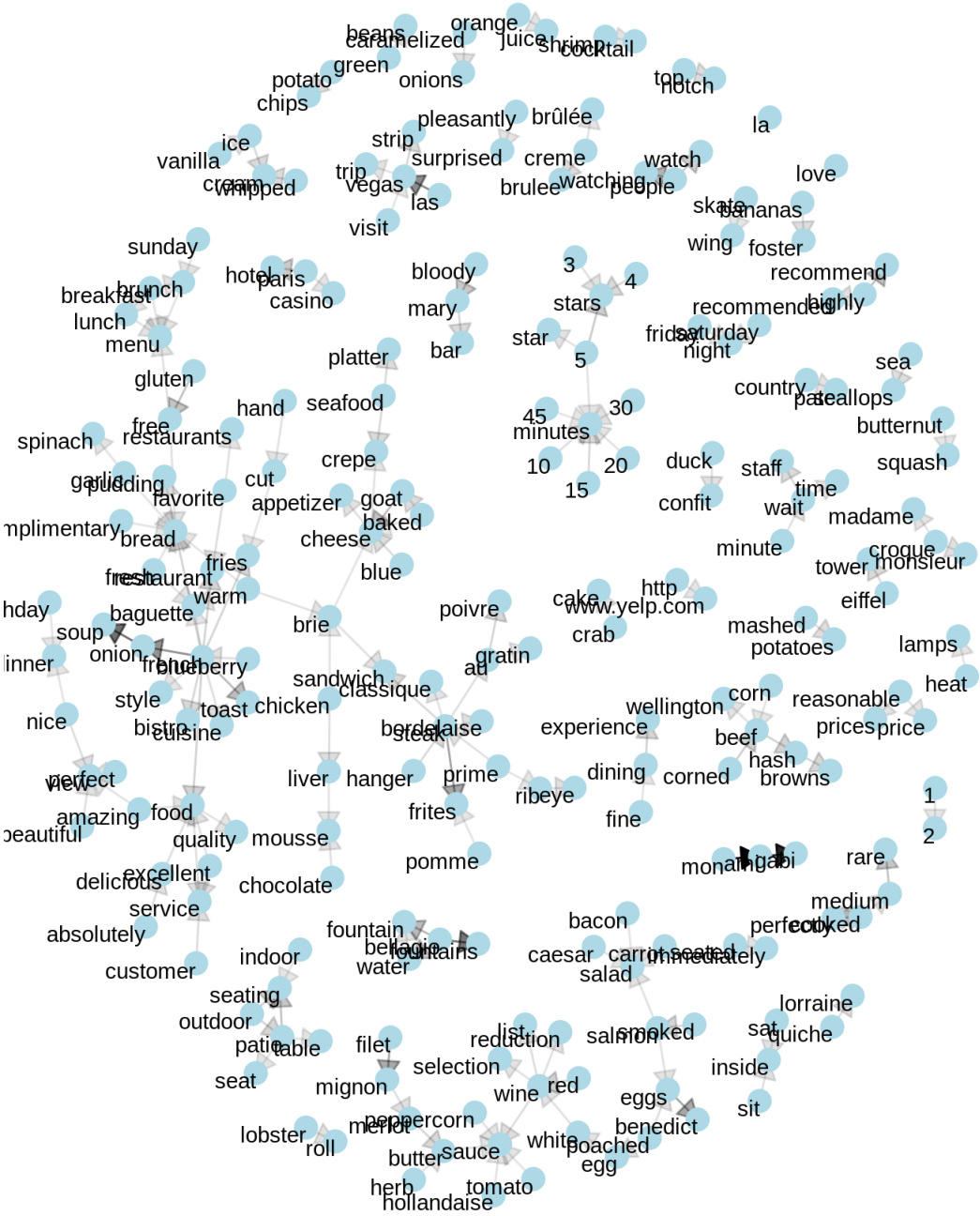
```
yelp_review %>%
  filter(business_id == "4JNXUYY8wbaaDmk3BPzlWw") %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word,
         !word2 %in% stop_words$word) %>%
  filter(!word1 %in% c("mon","ami")) %>%
   filter(!word2 %in% c("gabi")) %>%
  unite(bigramWord, word1, word2, sep = " ") %>%
  group_by(bigramWord) %>%
  tally() %>%
  ungroup() %>%
  arrange(desc(n)) %>%
  mutate(bigramWord = reorder(bigramWord,n)) %>%
  head(10) %>%

  ggplot(aes(x = bigramWord,y = n)) +
  geom_bar(stat='identity') +
  geom_text(aes(x = bigramWord, y = 1, label = paste0("(",n,")",sep="")),
            hjust=0, vjust=.5, size = 4, colour = 'black',
            fontface = 'bold') +
  labs(x = 'Bigram',
       y = 'Count',
       title = 'Bigram and Count') +
  coord_flip() +
  theme_bw()
```



Relationship among words

We explore the different relationship among the various words in Mon Ami Gabi reviews here through a network graph

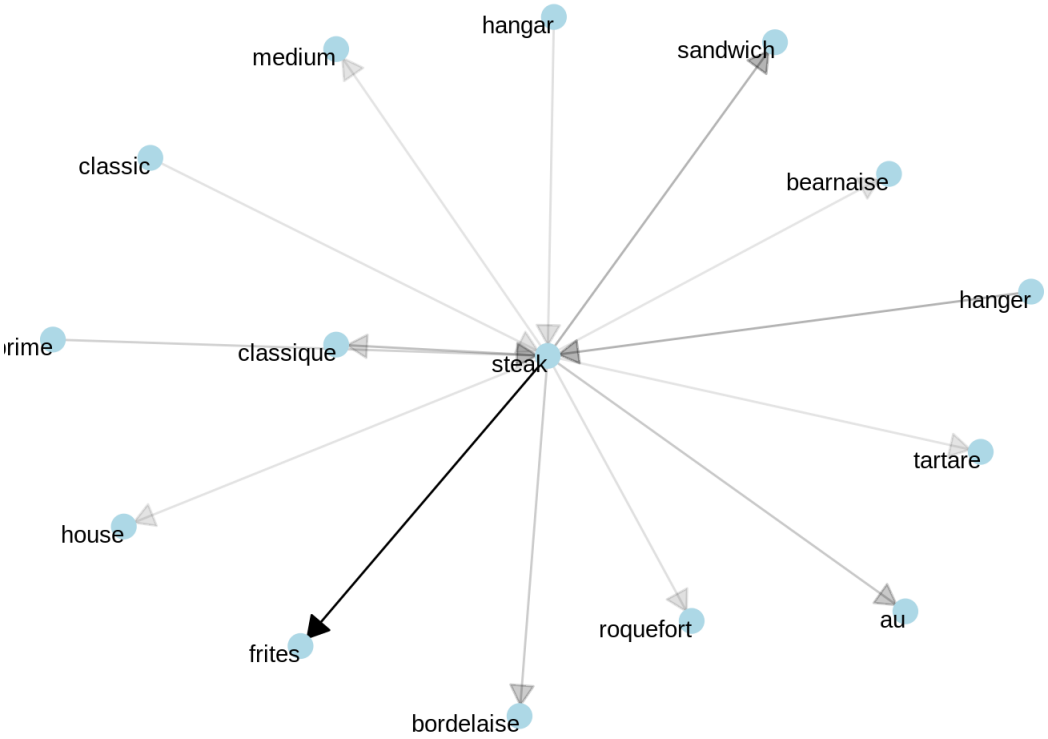
```
bigramsMonAmiGabi <- yelp_review %>%
  filter(business_id == "4JNXUYY8wbaaDmk3BPzlWw") %>%
  count_bigram()

bigramsMonAmiGabi %>%
  filter(n > 50) %>%
    graph_from_data_frame() %>%
    ggraph(layout = "fr")+
    geom_edge_link(aes(edge_alpha = n), show.legend = FALSE, arrow = a)+
    geom_node_point(color = "lightblue", size = 5)+
    geom_node_text(aes(label = name), vjust = 1, hjust = 1)+
    theme_void()
```



Relationship of words with dinner

The following network diagram shows the words associated with the word french

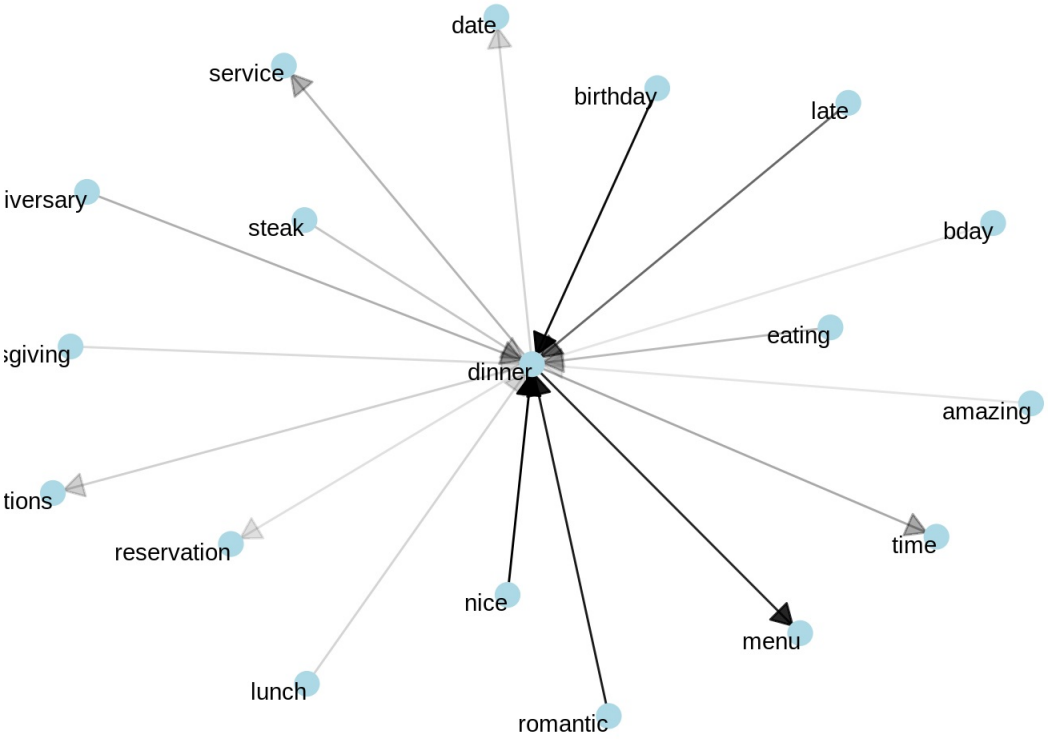
```
bigramsMonAmiGabi %>%
  filter(word1 == "steak" | word2 == "steak") %>%
  filter(n > 30) %>%graph_from_data_frame() %>%
    ggraph(layout = "fr") +
    geom_edge_link(aes(edge_alpha = n), show.legend = FALSE, arrow = a,end_cap = circle(.07, 'inches')) +
    geom_node_point(color = "lightblue", size = 5) +
    geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
    theme_void()
```



Relationship of words with dinner

The following network diagram shows the words associated with the word french

```
bigramsMonAmiGabi %>%
  filter(word1 == "dinner" | word2 == "dinner") %>%
  filter(n > 10) %>%graph_from_data_frame() %>%
    ggraph(layout = "fr") +
    geom_edge_link(aes(edge_alpha = n), show.legend = FALSE, arrow = a,end_cap = circle(.07, 'inches')) +
    geom_node_point(color = "lightblue", size = 5) +
    geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
    theme_void()
```



LasvegasCoords = yelp_business %>% filter(city == “Las Vegas”)

center_lon = median(LasvegasCoords*longitude*, *na*. *rm* = *TRUE*)*center*_*at* = *median*(*LasvegasCoords*latitude,na.rm = TRUE)

head(center_lon)

```
leaflet(LasvegasCoords) %>% addProviderTiles("Esri.NatGeoWorldMap") %>% addCircles(lng = ~longitude, lat = ~latitude,radius =
~sqrt(review_count)) %>%
=center_lat,zoom = 13)
```