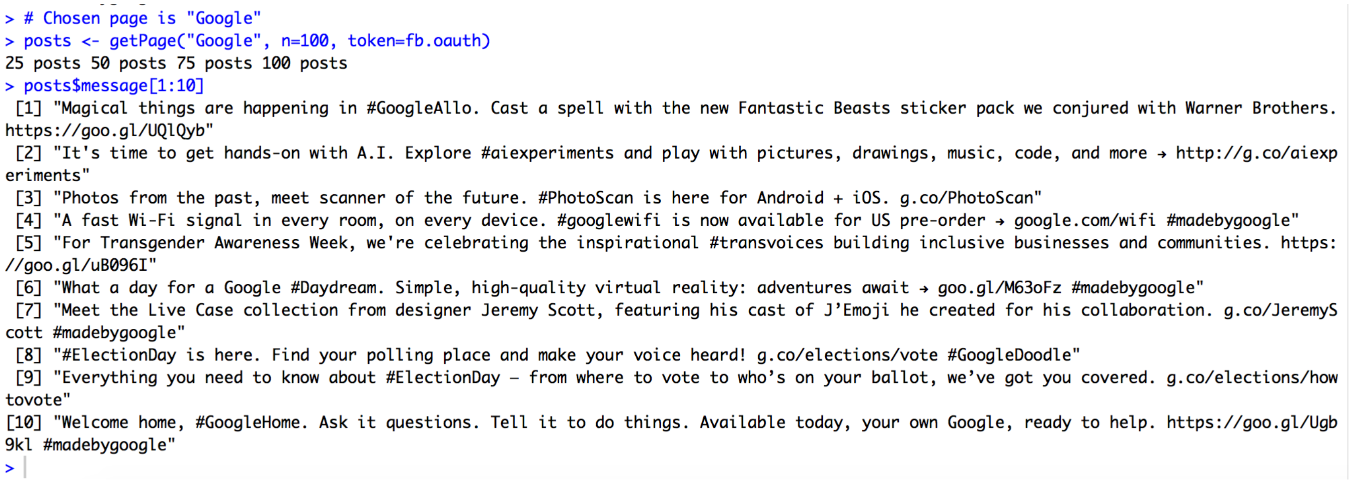
CS688 Web Analytics and Mining
Homework#4

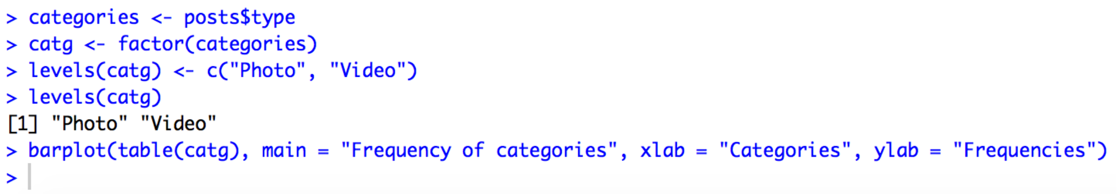
Facebook mining:

a) Retrieve the 100 recent posts from the page of your favorite Facebook user.
Chosen Facebook page is : Google

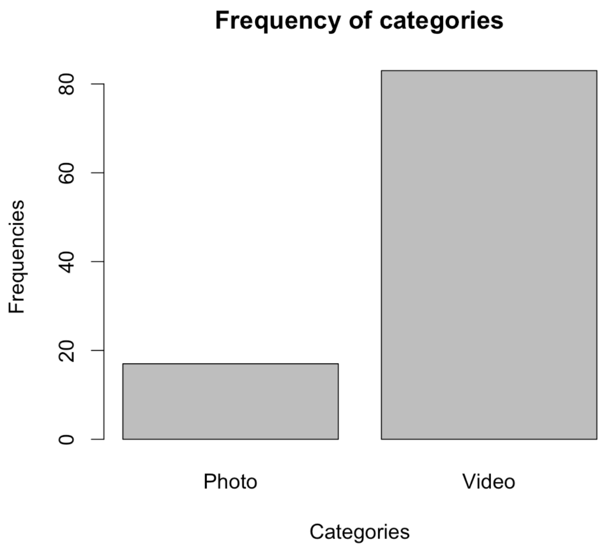


```
> # Chosen page is "Google"
> posts <- getPage("Google", n=100, token=fb.oauth)
25 posts 50 posts 75 posts 100 posts
> posts$message[1:10]
 [1] "Magical things are happening in #GoogleAllo. Cast a spell with the new Fantastic Beasts sticker pack we conjured with Warner Brothers.
https://goo.gl/UQlQyb"
 [2] "It's time to get hands-on with A.I. Explore #aiexperiments and play with pictures, drawings, music, code, and more → http://g.co/aiexp
eriments"
 [3] "Photos from the past, meet scanner of the future. #PhotoScan is here for Android + iOS. g.co/PhotoScan"
 [4] "A fast Wi-Fi signal in every room, on every device. #googlewifi is now available for US pre-order → google.com/wifi #madebygoogle"
 [5] "For Transgender Awareness Week, we're celebrating the inspirational #transvoices building inclusive businesses and communities. https:
//goo.gl/uB096I"
 [6] "What a day for a Google #Daydream. Simple, high-quality virtual reality: adventures await → goo.gl/M63oFz #madebygoogle"
 [7] "Meet the Live Case collection from designer Jeremy Scott, featuring his cast of J'Emoji he created for his collaboration. g.co/JeremyS
cott #madebygoogle"
 [8] "#ElectionDay is here. Find your polling place and make your voice heard! g.co/elections/vote #GoogleDoodle"
 [9] "Everything you need to know about #ElectionDay – from where to vote to who's on your ballot, we've got you covered. g.co/elections/how
tovote"
[10] "Welcome home, #GoogleHome. Ask it questions. Tell it to do things. Available today, your own Google, ready to help. https://goo.gl/Ugb
9kl #madebygoogle"
> 
```

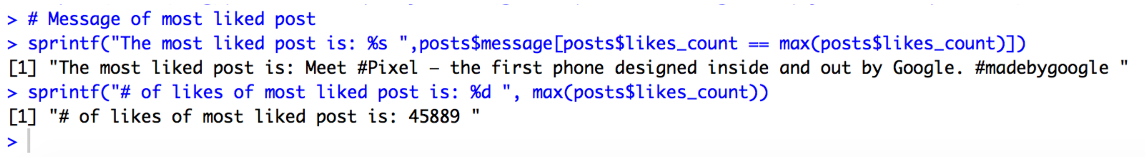
b) Frequencies of the categories:



```
> categories <- posts$type
> catg <- factor(categories)
> levels(catg) <- c("Photo", "Video")
> levels(catg)
[1] "Photo" "Video"
> barplot(table(catg), main = "Frequency of categories", xlab = "Categories", ylab = "Frequencies")
> 
```

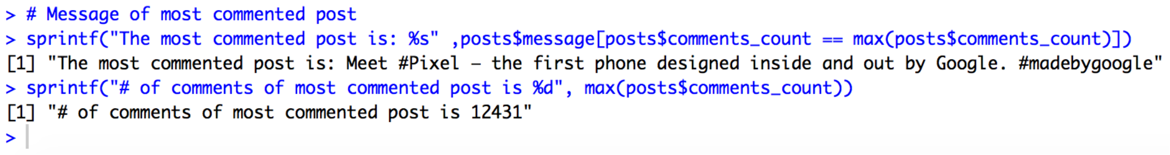


c) Most Liked Post:



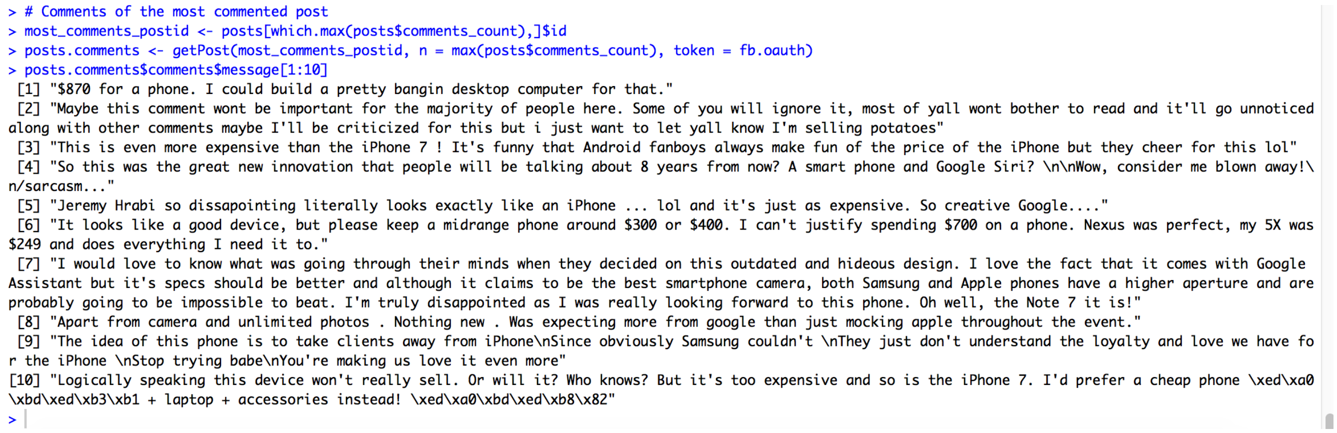
```
> # Message of most liked post
> sprintf("The most liked post is: %s ",posts$message[posts$likes_count == max(posts$likes_count)])
[1] "The most liked post is: Meet #Pixel – the first phone designed inside and out by Google. #madebygoogle "
> sprintf("# of likes of most liked post is: %d ", max(posts$likes_count))
[1] "# of likes of most liked post is: 45889 "
> 
```

d) Most Commented Post:



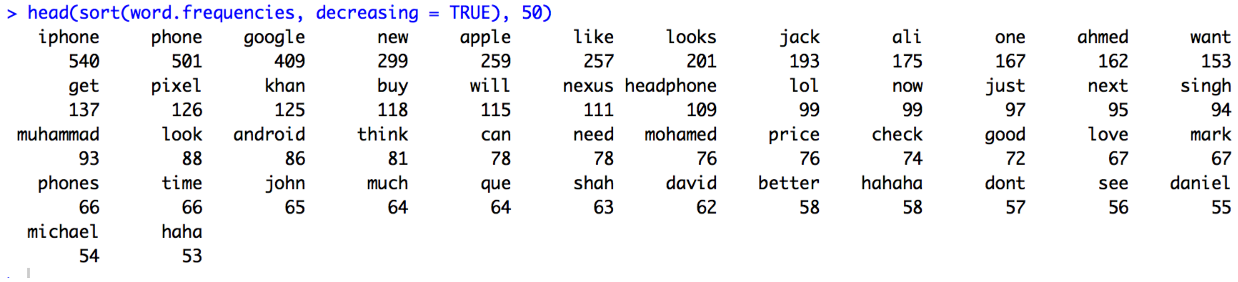
```
> # Message of most commented post
> sprintf("The most commented post is: %s" ,posts$message[posts$comments_count == max(posts$comments_count)])
[1] "The most commented post is: Meet #Pixel – the first phone designed inside and out by Google. #madebygoogle"
> sprintf("# of comments of most commented post is %d", max(posts$comments_count))
[1] "# of comments of most commented post is 12431"
> 
```

e) Comments associated with most commented post:



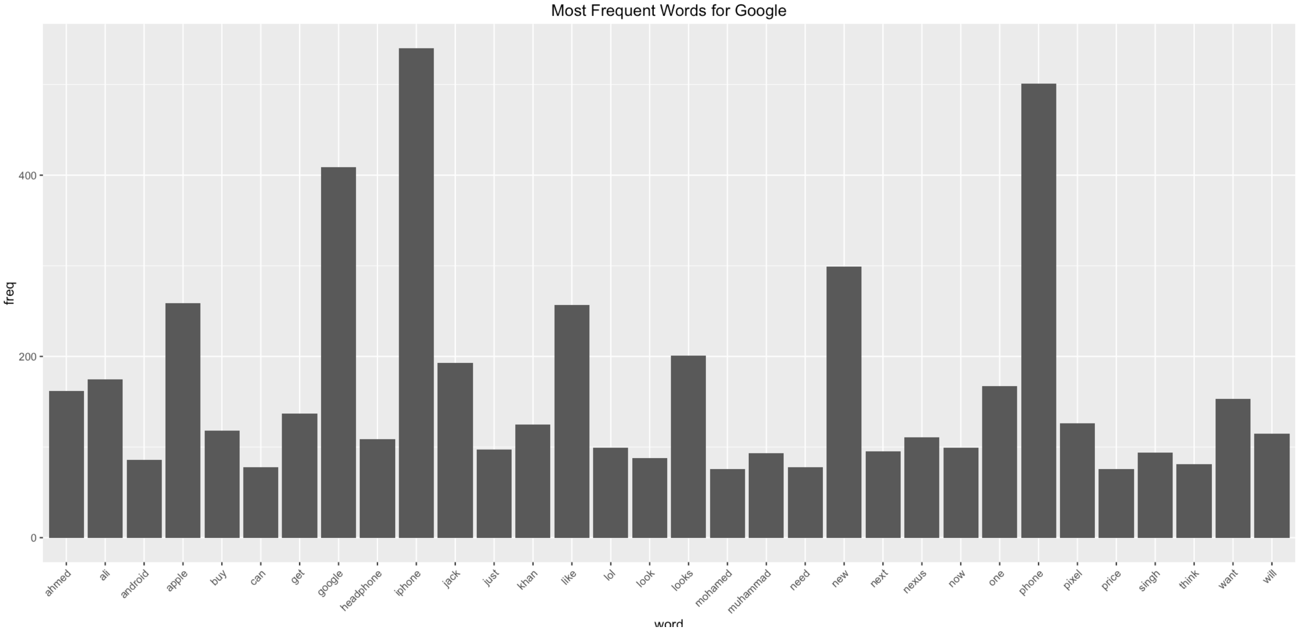
```
> # Comments of the most commented post
> most_comments_postid <- posts[which.max(posts$comments_count),]$id
> posts.comments <- getPost(most_comments_postid, n = max(posts$comments_count), token = fb.oauth)
> posts.comments$comments$message[1:10]
 [1] "$870 for a phone. I could build a pretty bangin desktop computer for that."
 [2] "Maybe this comment wont be important for the majority of people here. Some of you will ignore it, most of yall wont bother to read and it'll go unnoticed along with other comments maybe I'll be criticized for this but i just want to let yall know I'm selling potatoes"
 [3] "This is even more expensive than the iPhone 7 ! It's funny that Android fanboys always make fun of the price of the iPhone but they cheer for this lol"
 [4] "So this was the great new innovation that people will be talking about 8 years from now? A smart phone and Google Siri? \n\nWow, consider me blown away!\n/sarcasm..."
 [5] "Jeremy Hrabi so dissapointing literally looks exactly like an iPhone ... lol and it's just as expensive. So creative Google...."
 [6] "It looks like a good device, but please keep a midrange phone around $300 or $400. I can't justify spending $700 on a phone. Nexus was perfect, my 5X was $249 and does everything I need it to."
 [7] "I would love to know what was going through their minds when they decided on this outdated and hideous design. I love the fact that it comes with Google Assistant but it's specs should be better and although it claims to be the best smartphone camera, both Samsung and Apple phones have a higher aperture and are probably going to be impossible to beat. I'm truly disappointed as I was really looking forward to this phone. Oh well, the Note 7 it is!"
 [8] "Apart from camera and unlimited photos . Nothing new . Was expecting more from google than just mocking apple throughout the event."
 [9] "The idea of this phone is to take clients away from iPhone\nSince obviously Samsung couldn't \nThey just don't understand the loyalty and love we have for the iPhone \nStop trying babe\nYou're making us love it even more"
[10] "Logically speaking this device won't really sell. Or will it? Who knows? But it's too expensive and so is the iPhone 7. I'd prefer a cheap phone \xed\xa0\xbd\xed\xb3\xb1 + laptop + accessories instead! \xed\xa0\xbd\xed\xb8\x82"
> 
```

f) Frequent terms in the comments:

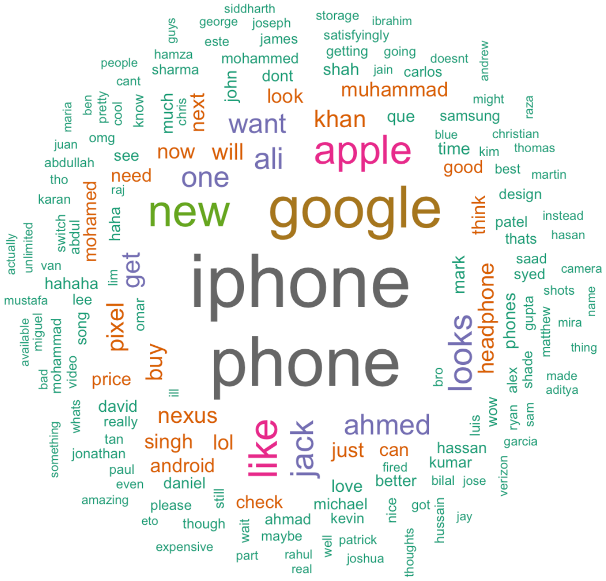


```
> head(sort(word.frequencies, decreasing = TRUE), 50)
   iphone     phone    google       new     apple      like     looks      jack       ali       one     ahmed      want 
      540       501       409       299       259       257       201       193       175       167       162       153 
      get     pixel      khan       buy      will     nexus headphone       lol       now      just      next     singh 
      137       126       125       118       115       111       109        99        99        97        95        94 
 muhammad      look   android     think       can      need   mohamed     price     check      good      love      mark 
       93        88        86        81        78        78        76        76        74        72        67        67 
   phones      time      john      much       que      shah     david    better    hahaha      dont       see    daniel 
       66        66        65        64        64        63        62        58        58        57        56        55 
  michael      haha 
       54        53 
```

The barplot of the most frequent terms is shown below



Wordcloud for the frequent terms is



R-code:

```
rm(list = ls())
library(Rfacebook)
library(httr)
library(rjson)
library(httpuv)
library(tm)
library(SnowballC)
library(ggplot2)
library(wordcloud)
library(RColorBrewer)
cat("\014")

# Chosen page is "Google"
posts <- getPage("Google", n=100, token=fb.oauth)
posts$message[1:10]
categories <- posts$type
catg <- factor(categories)
levels(catg) <- c("Photo", "Video")
levels(catg)
barplot(table(catg), main = "Frequency of categories", xlab = "Categories", ylab = "Frequencies")

# Message of most liked post
sprintf("The most liked post is: %s ",posts$message[posts$likes_count == max(posts$likes_count)])
sprintf("# of likes of most liked post is: %d ", max(posts$likes_count))

# Message of most commented post
sprintf("The most commented post is: %s" ,posts$message[posts$comments_count == max(posts$comments_count)])
sprintf("# of comments of most commented post is %d", max(posts$comments_count))

# Comments of the most commented post
most_comments_postid <- posts[which.max(posts$comments_count),]$id
posts.comments <- getPost(most_comments_postid, n = max(posts$comments_count), token = fb.oauth)
posts.comments$comments$message[1:10]

# Creating a corpus
posts.corpus <- Corpus(VectorSource(posts.comments$comments$message))
posts.corpus[[1]]$content
posts.tmp <- posts.corpus

# Function to remove URL's
removeUrl <- content_transformer(function(x) gsub("(f|ht)tp[[:alnum:][:punct:]]*", " ", x))
# Function to remove Non Ascii characters
removeNonASCII <- content_transformer(function(x) iconv(x, "latin1", "ASCII", sub=“"))

post.processed <- tm_map(posts.tmp, removeUrl) # Remove URL's
post.processed <- tm_map(post.processed, removeNumbers) # Remove Numbers
post.processed <- tm_map(post.processed, removePunctuation) #Remove Punctuations
post.processed <- tm_map(post.processed, removeNonASCII) # Remove the NonASCII
post.processed <- tm_map(post.processed, removeWords, stopwords("english"), lazy = TRUE) # Remove the stopwords
post.processed <- tm_map(post.processed, stripWhitespace, lazy = TRUE) # Remove the whitespaces
post.processed <- tm_map(post.processed, content_transformer(tolower), lazy = TRUE) # Convert the characters to lowercase
```

```
# ----- Creating a Term document Matrix -----
posts.tdm <- TermDocumentMatrix(post.processed)
inspect(posts.tdm[1:10,1:10])
dim(posts.tdm)

# -----Finding the frequencies -----
word.frequencies <- rowSums(as.matrix(posts.tdm))
head(sort(word.frequencies, decreasing = TRUE), 50)
posts.word.frequencies <- data.frame(word=names(word.frequencies), freq=word.frequencies)

# ----- Barplot of the most frequent terms -----
ggplot(subset(posts.word.frequencies, freq>75), aes(word, freq)) +
  geom_bar(stat = "identity") +
  ggtitle("Most Frequent Words for Google") +
  theme(axis.text.x=element_text(angle=45,hjust = 1))

# ----- Wordcloud of the most frequent terms -----
wordcloud(words = names(word.frequencies), freq = word.frequencies, min.freq = 25, random.order = FALSE, rot.per = 0.35,
colors=brewer.pal(8, "Dark2"))
```