

# Homework 5

*Yongshi JIE*  
*Xiamen University*  
*Master Finance*

# 1. Do a word cloud for Shakespeare's dramas. Romeo and Julia, Julius Caesar, Hamlet.

```
rm(list = ls())
#install.packages("RCurl")
#install.packages("XML")
library(RCurl)
library(XML)
url1 = "http://shakespeare.mit.edu/romeo_juliet/full.html"
url2 = "http://shakespeare.mit.edu/julius_caesar/full.html"
url3 = "http://shakespeare.mit.edu/hamlet/full.html"
html1 = readLines(url1, encoding = "UTF-8")
html2 = readLines(url2, encoding = "UTF-8")
html3 = readLines(url3, encoding = "UTF-8")
html1 = htmlParse(html1, encoding = "UTF-8")
html2 = htmlParse(html2, encoding = "UTF-8")
html3 = htmlParse(html3, encoding = "UTF-8")
```

## #HW5.1 Wordcloud

```
#install.packages("bitops")
#install.packages("stringr")
library(bitops)
library(stringr)
abs1 = lapply(url1, FUN = function(x) htmlParse(x, encoding =
"Latin-1"))
abs2 = lapply(url2, FUN = function(x) htmlParse(x, encoding =
"Latin-1"))
abs3 = lapply(url3, FUN = function(x) htmlParse(x, encoding =
"Latin-1"))
clean_txt = function(x) {
  cleantxt = xpathApply(x, "//body//text()
    [not(ancestor :: script)][ not(ancestor :: style)]
    [not(ancestor :: noscript)] ",xmlValue)
  cleantxt = paste(cleantxt, collapse="\n")
  cleantxt = str_replace_all(cleantxt, "\n", " ")
  cleantxt = str_replace_all(cleantxt, "\r", "")
  cleantxt = str_replace_all(cleantxt, "\t", "")
  cleantxt = str_replace_all(cleantxt, "<br>", "")
  return(cleantxt)
}
cleantxt1 = lapply(abs1,clean_txt)
cleantxt2 = lapply(abs2,clean_txt)
cleantxt3 = lapply(abs3,clean_txt)
```

```
vec_abs1 = unlist(cleantxt1)
vec_abs2 = unlist(cleantxt2)
vec_abs3 = unlist(cleantxt3)
###Text Mining
#install.packages("tm")
#install.packages("SnowballC")
library(tm)
library(SnowballC)
```

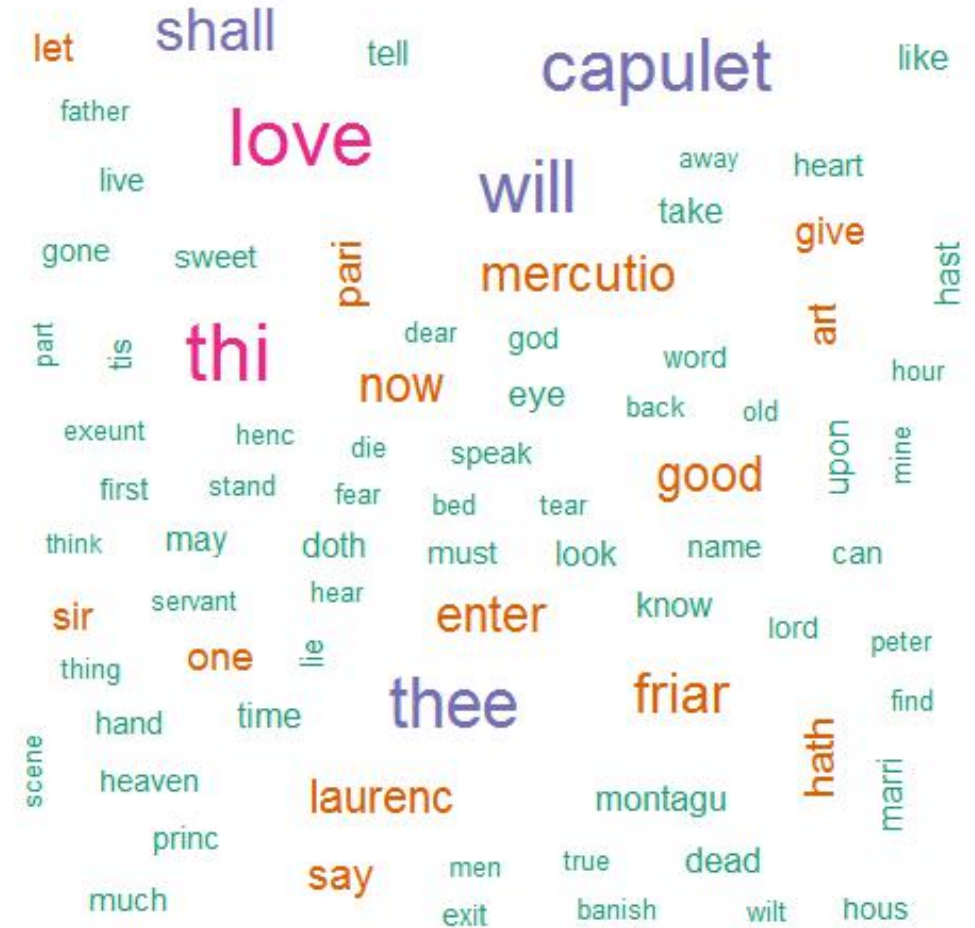
```
abs1 = Corpus(VectorSource(vec_abs1))
abs2 = Corpus(VectorSource(vec_abs2))
abs3 = Corpus(VectorSource(vec_abs3))
```

```
abs_dtm1 = DocumentTermMatrix(abs1, control = list(
  stemming = TRUE, stopwords = TRUE, minWordLength = 3,
  removeNumbers = TRUE, removePunctuation = TRUE))
abs_dtm2 = DocumentTermMatrix(abs2, control = list(
  stemming = TRUE, stopwords = TRUE, minWordLength = 3,
  removeNumbers = TRUE, removePunctuation = TRUE))
abs_dtm3 = DocumentTermMatrix(abs3, control = list(
  stemming = TRUE, stopwords = TRUE, minWordLength = 3,
  removeNumbers = TRUE, removePunctuation = TRUE))
```

**1. Do a word cloud for Shakespeare's dramas. Romeo and Julia, Julius Caesar, Hamlet.**

```
##WordCloud
#instal.packages("ggplot2")
#install.packages("wordcloud")
library(ggplot2)
library(wordcloud)
freq1 = colSums(as.matrix(abs_dtm1))
freq2 = colSums(as.matrix(abs_dtm2))
freq3 = colSums(as.matrix(abs_dtm3))
wf1  = data.frame(word=names(freq1), freq=freq1)
wf2  = data.frame(word=names(freq2), freq=freq2)
wf3  = data.frame(word=names(freq3), freq=freq3)

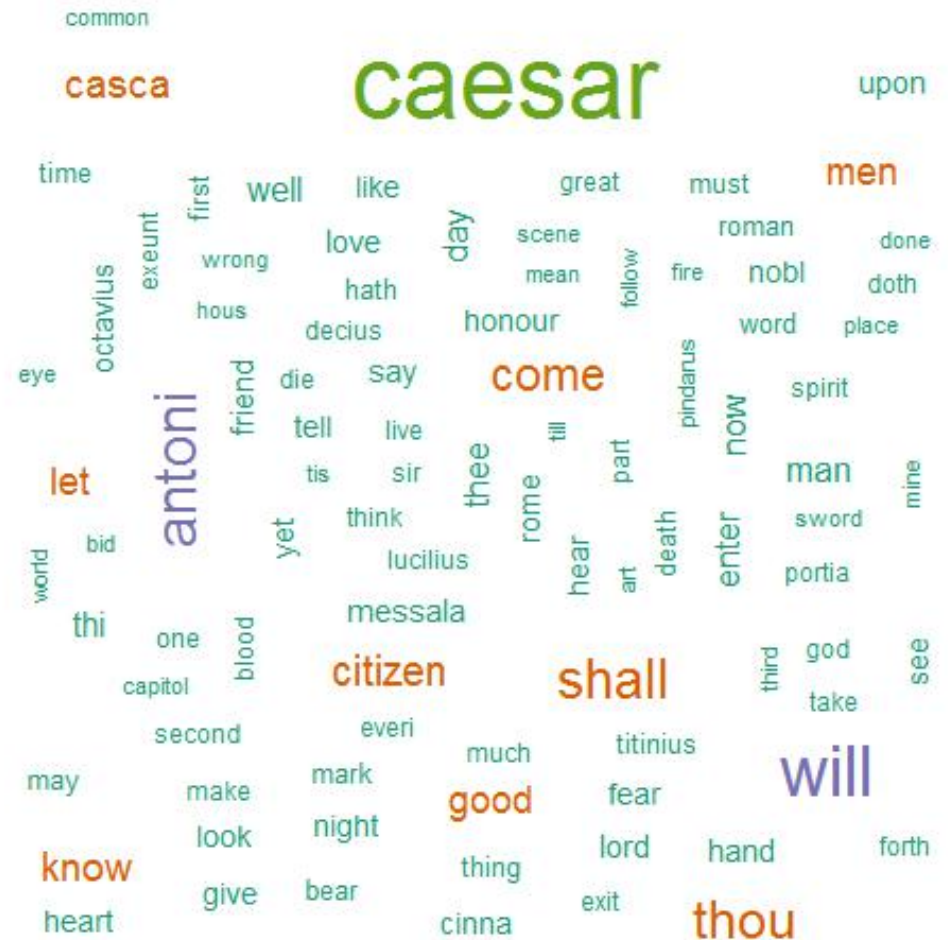
#Romeo and Juliet
plot1 = ggplot(subset(wf1, freq>15), aes(word, freq1))
plot1 = plot1 + geom_bar(stat="identity")
plot1 = plot1 + theme(axis.text.x=element_text(angle=45,
hjust=1))
freq1 = colSums(as.matrix(abs_dtm1))
dark2_1 = brewer.pal(6, "Dark2")
wordcloud(names(freq1), freq1, max.words=100, rot.per=0.2,
colors=dark2_1)
```



**1. Do a word cloud for Shakespeare's dramas. Romeo and Julia, Julius Caesar, Hamlet.**

# #Julius Caesar

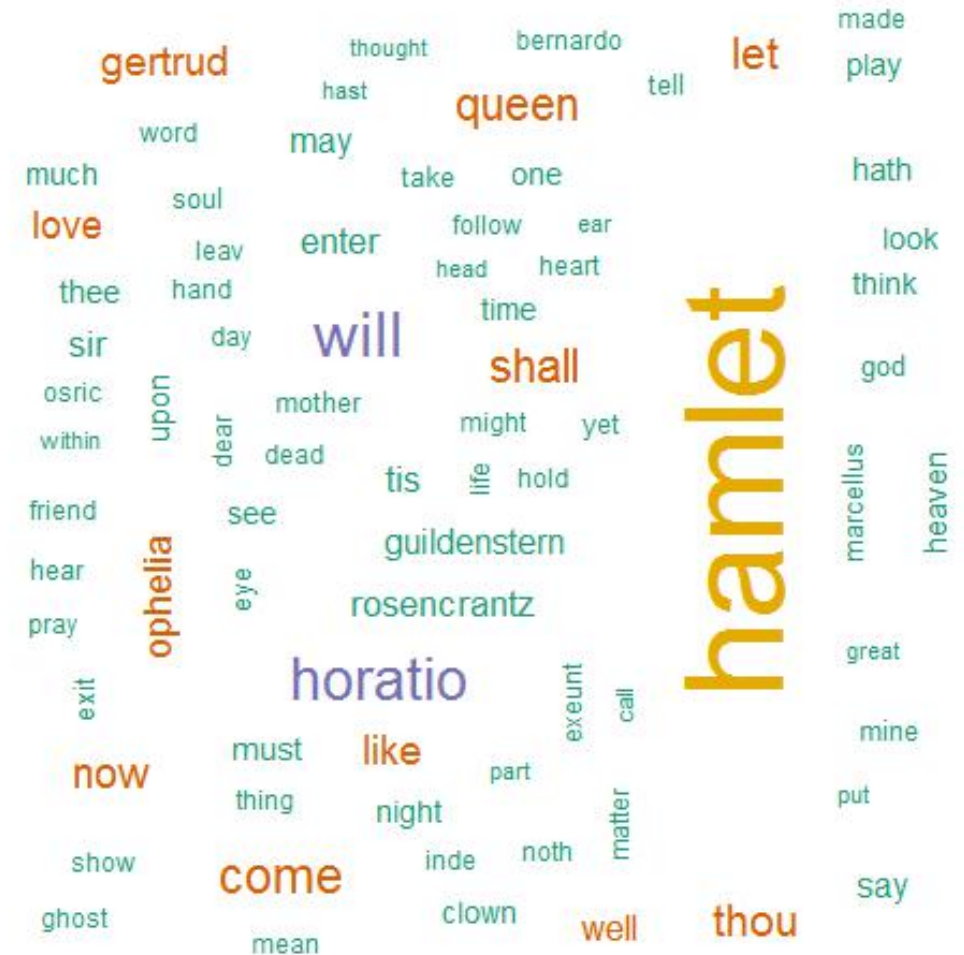
```
plot2 = ggplot(subset(wf2, freq>15), aes(word, freq2))
plot2 = plot2 + geom_bar(stat="identity")
plot2 = plot2 + theme(axis.text.x=element_text(angle=45,
hjust=1))
freq2 = colSums(as.matrix(abs_dtm2))
dark2_2 = brewer.pal(6, "Dark2")
wordcloud(names(freq2), freq2, max.words=100, rot.per=0.2,
colors=dark2_2)
```



**1. Do a word cloud for Shakespeare's dramas. Romeo and Julia, Julius Caesar, Hamlet.**

# #Hamlet

```
plot3 = ggplot(subset(wf3, freq>15), aes(word, freq3))
plot3 = plot3 + geom_bar(stat="identity")
plot3 = plot3 + theme(axis.text.x=element_text(angle=45,
hjust=1))
freq3 = colSums(as.matrix(abs_dtm3))
dark2_3 = brewer.pal(6, "Dark2")
wordcloud(names(freq3), freq3, max.words=100, rot.per=0.2,
colors=dark2_3)
```

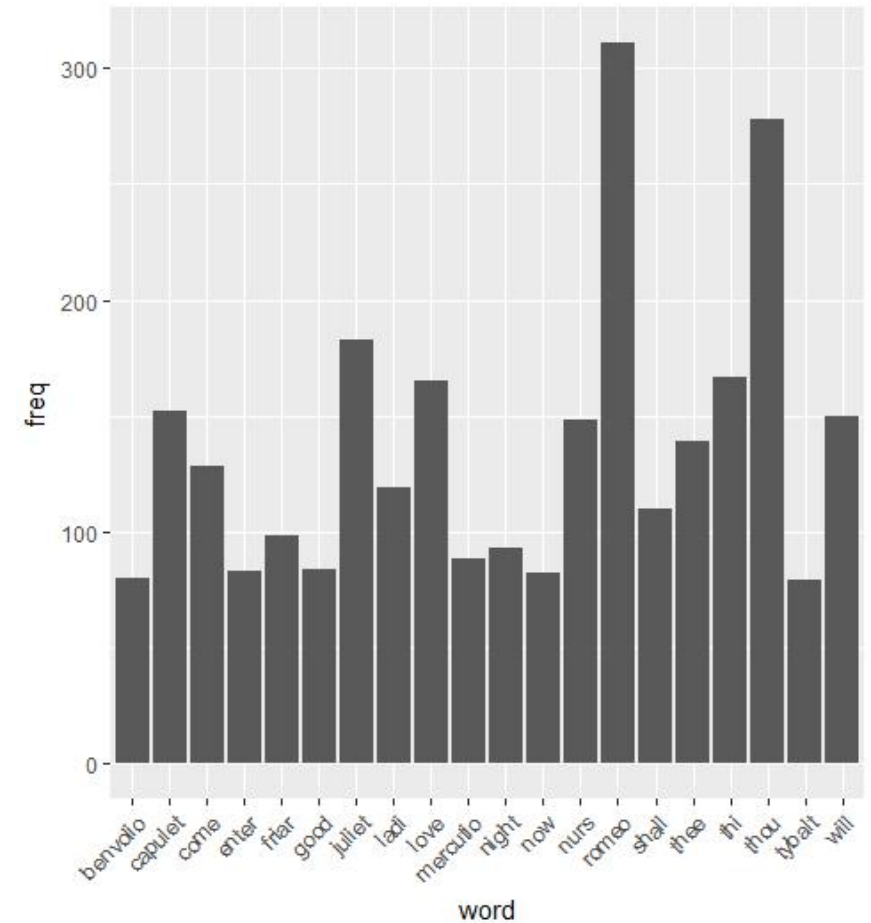




## 2. Calculate the histogram of words

### #Romeo and Juliet

```
wf1 <- wf1[order(-wf1$freq),]  
wf1 <- wf1[c(1:20),]  
p1 = ggplot(subset(wf1, freq > 15), aes(word, freq))  
p1 = p1 + geom_bar(stat = "identity")  
p1 = p1 + theme(axis.text.x = element_text(angle = 45, hjust = 1))  
p1
```



## 2. Calculate the histogram of words

#Julius Caesar

```
wf2 <- wf2[order(-wf2$freq),]
```

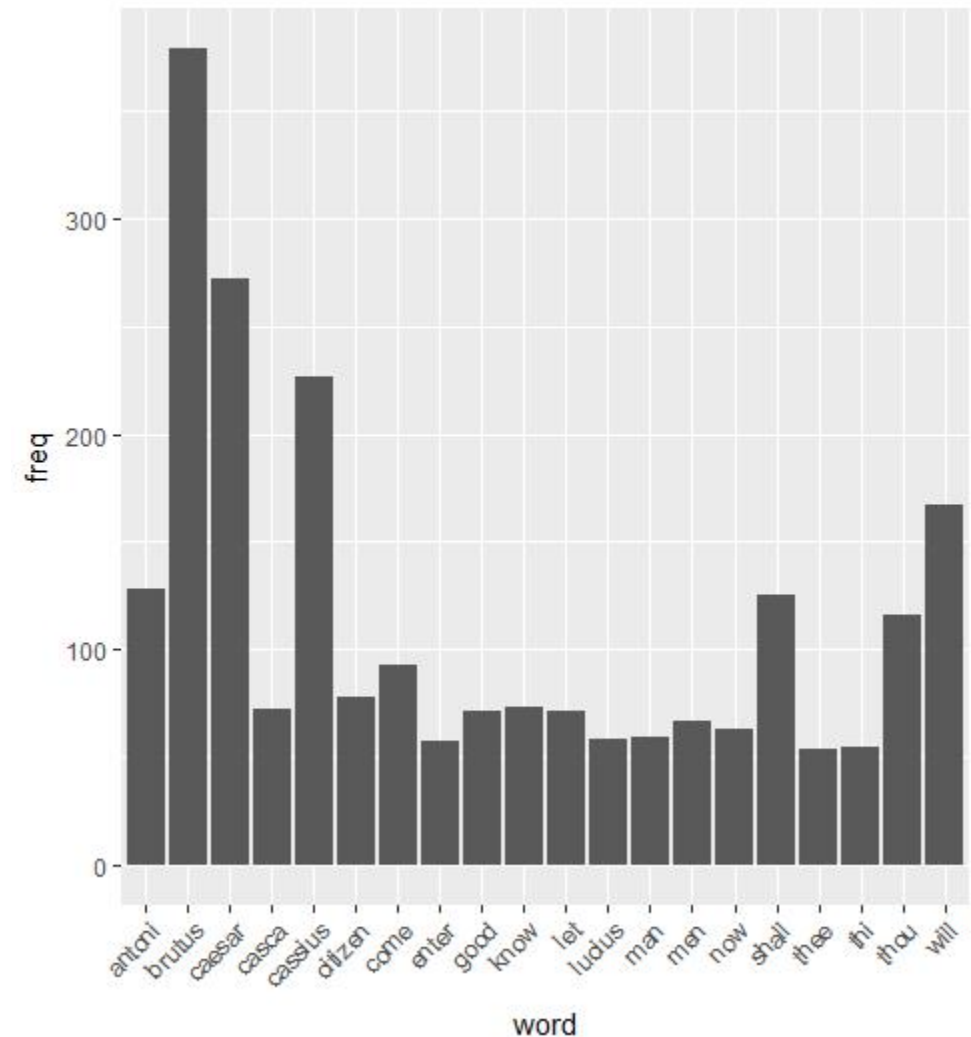
```
wf2 <- wf2[c(1:20),]
```

```
p2 = ggplot(subset(wf2, freq > 15), aes(word, freq))
```

```
p2 = p2 + geom_bar(stat = "identity")
```

```
p2 = p2 + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
p2
```

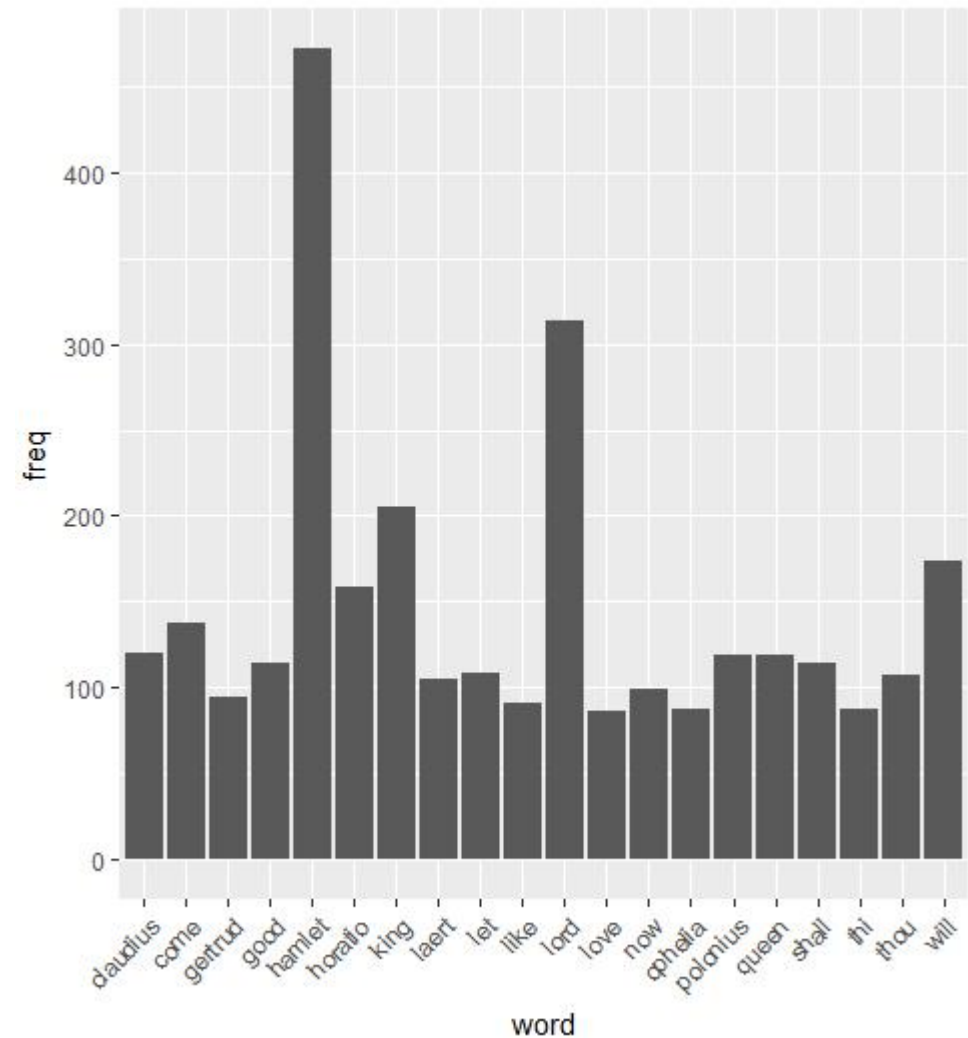




## 2. Calculate the histogram of words

#Hamlet

```
wf3 <- wf3[order(-wf3$freq),]  
wf3 <- wf3[c(1:20),]  
p3 = ggplot(subset(wf3, freq > 15), aes(word, freq))  
p3 = p3 + geom_bar(stat = "identity")  
p3 = p3 + theme(axis.text.x = element_text(angle = 45,  
hjust = 1))  
p3
```



Thanks for your attention!