TidyTuesday: Shakespeare Analysis

This is my data analysis of the datasets on shakespeare for the tidytuesday of 9/17. First let's set up the packages I will be using for this analysis.

```
install.packages("tidytuesdayR")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(tidytuesdayR)
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(ggplot2)
install.packages("SnowballC")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(SnowballC)
install.packages("wordcloud")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
install.packages("RColorBrewer")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```r
library(RColorBrewer)
install.packages("tidytext")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```r
library(tidytext)
install.packages("tm")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```r
library(tm)
```

```
## Loading required package: NLP
##
## Attaching package: 'NLP'
##
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

Now let's pull the datasets from the tidytuesday package.

```r
tt_gh <- tt_load_gh("2024-09-17")
```

```
## ---- Compiling #TidyTuesday Information for 2024-09-17 ----
## --- There are 3 files available ---
```

Now I see that it only finds three of the datasets, I will manually add the rest and load all of them.

```r
hamlet <- tt_download_file(tt_gh, "hamlet.csv")
macbeth <- tt_download_file(tt_gh, "macbeth.csv")
romeo_juliet <- tt_download_file(tt_gh, "romeo_juliet.csv")
julius_caesar <- read_csv("julius_caesar.csv")
```

```
## Rows: 2748 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): act, scene, character, dialogue
## dbl (1): line_number
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
othello <- read_csv("othello.csv")
```

```
## Rows: 3742 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): act, scene, character, dialogue
## dbl (1): line_number
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
metadata <- read_csv("metadata.csv")
```

```
## Rows: 42 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Title, Genre, URL, File
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Now then we will begin the analysis. I want to first look at which character had the most lines for each play. My theory is that the main characters should have the most line in their given play.

First I have to set the dataframes that will be used to count the amount of line per person.

```
# Creating dataframes with the all the characters and their line counts
line_count_hamlet <- hamlet %>%
  group_by(character) %>%
  summarize(line_count = n())
line_count_hamlet <- arrange(line_count_hamlet, -line_count)

line_count_macbeth <- macbeth %>%
  group_by(character) %>%
  summarize(line_count = n())
line_count_macbeth <- arrange(line_count_macbeth, -line_count)

line_count_romjul <- romeo_juliet %>%
  group_by(character) %>%
  summarize(line_count = n())
line_count_romjul <- arrange(line_count_romjul, -line_count)

line_count_jc <- julius_caesar %>%
  group_by(character) %>%
  summarize(line_count = n())
line_count_jc <- line_count_jc <- arrange(line_count_jc, -line_count)

line_count_othello <- othello %>%
  group_by(character) %>%
  summarize(line_count = n())
line_count_othello <- arrange(line_count_othello, -line_count)
```

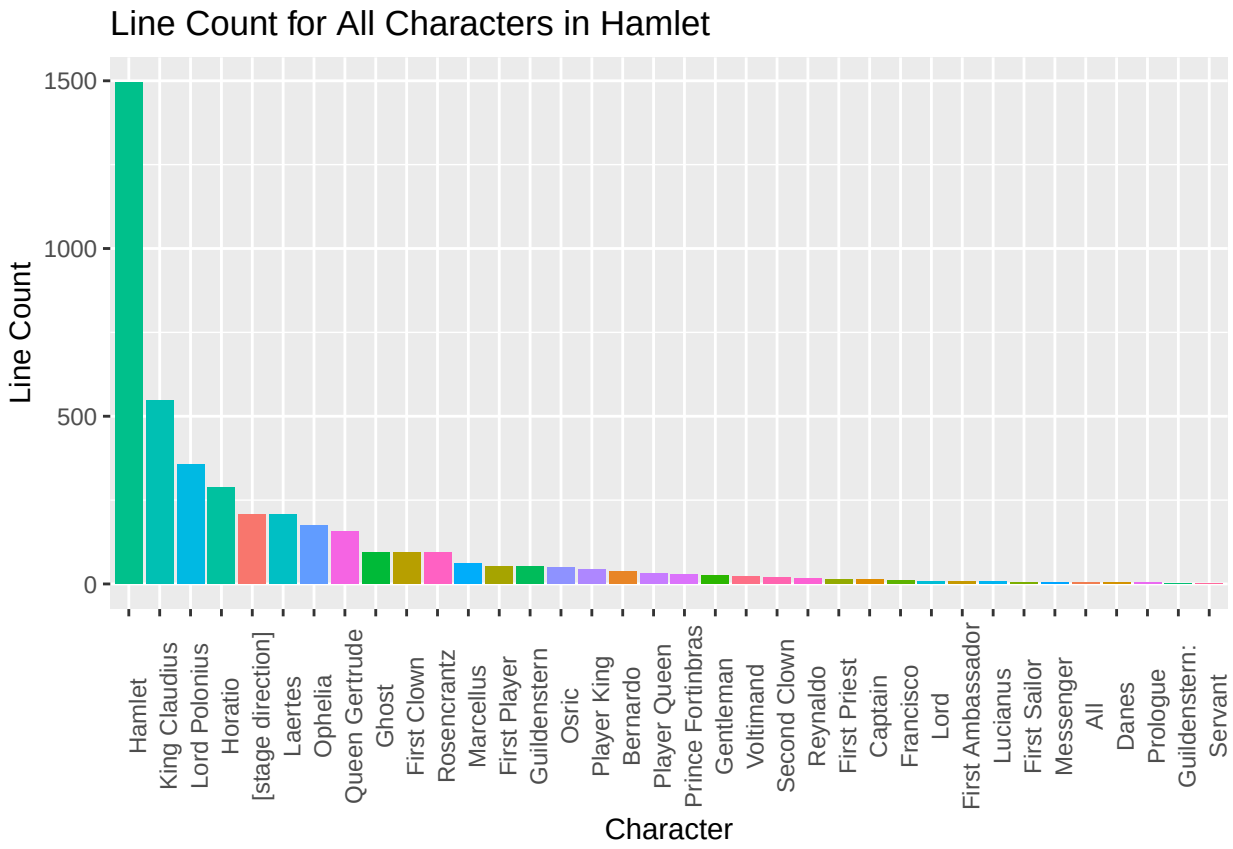
Now let's view the dataframes as well as graph them.

```
knitr::kable(line_count_hamlet[,1:2], format="markdown")
```

character	line_count
Hamlet	1495
King Claudius	546
Lord Polonius	355
Horatio	289
Laertes	206
[stage direction]	206
Ophelia	173
Queen Gertrude	157
Ghost	95
First Clown	94
Rosencrantz	93
Marcellus	62
First Player	52

character	line_count
Guildenstern	52
Osric	48
Player King	44
Bernardo	38
Player Queen	30
Prince Fortinbras	27
Gentleman	24
Voltimand	22
Second Clown	18
Reynaldo	15
First Priest	13
Captain	12
Francisco	10
Lord	7
First Ambassador	6
Lucianus	6
First Sailor	5
Messenger	5
All	4
Danes	3
Prologue	3
Guildenstern:	1
Servant	1

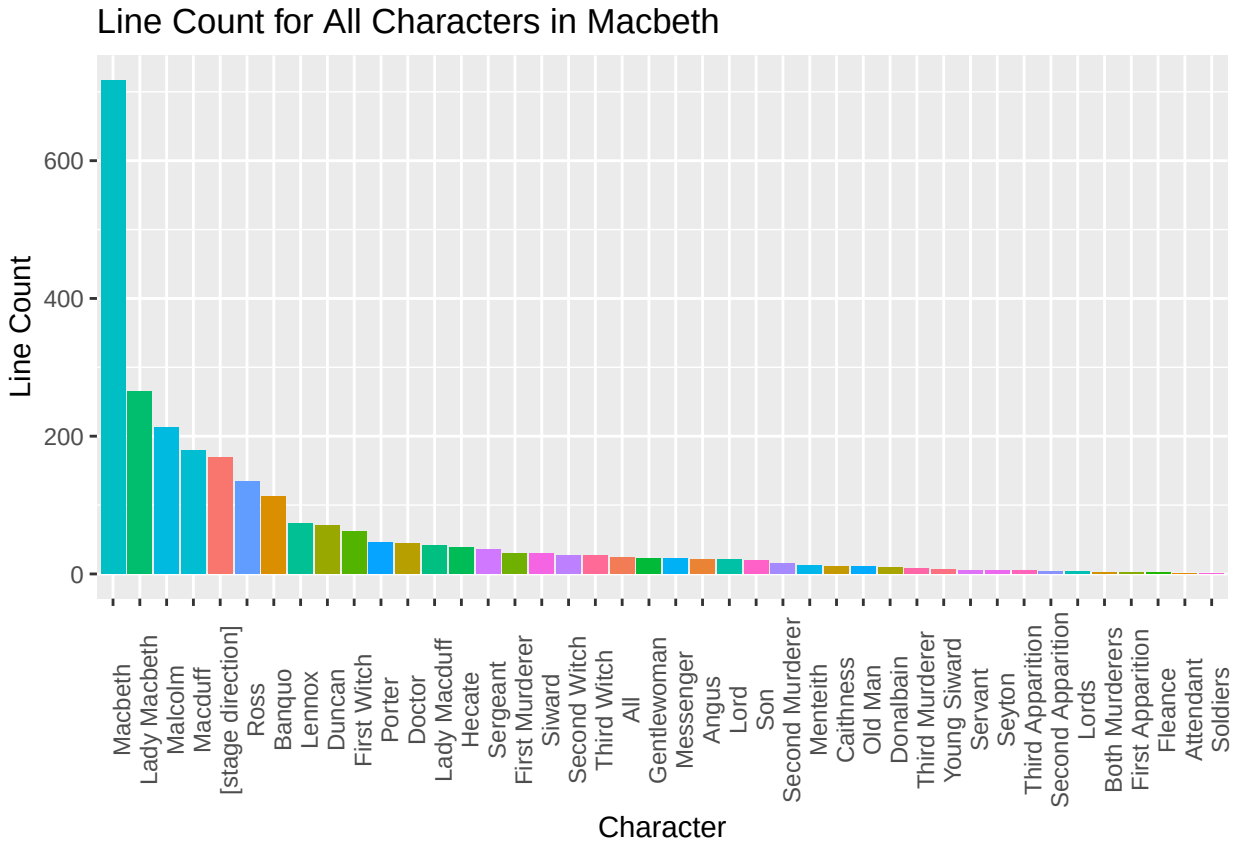
```
ggplot(data = line_count_hamlet, aes(y=line_count,x=reorder(character, -line_count)))+geom_bar(stat='id
```



```
knitr::kable(line_count_macbeth[,1:2], format="markdown")
```

character	line_count
Macbeth	717
Lady Macbeth	265
Malcolm	212
Macduff	180
[stage direction]	169
Ross	135
Banquo	113
Lennox	74
Duncan	70
First Witch	62
Porter	46
Doctor	45
Lady Macduff	41
Hecate	39
Sergeant	35
First Murderer	30
Siward	30
Second Witch	27
Third Witch	27
All	24
Gentlewoman	23
Messenger	23
Angus	21
Lord	21
Son	20
Second Murderer	15
Menteith	12
Caithness	11
Old Man	11
Donalbain	10
Third Murderer	8
Young Siward	7
Servant	5
Seyton	5
Third Apparition	5
Second Apparition	4
Lords	3
Both Murderers	2
First Apparition	2
Fleance	2
Attendant	1
Soldiers	1

```
ggplot(data = line_count_macbeth, aes(y=line_count,x=reorder(character, -line_count)))+geom_bar(stat='i
```

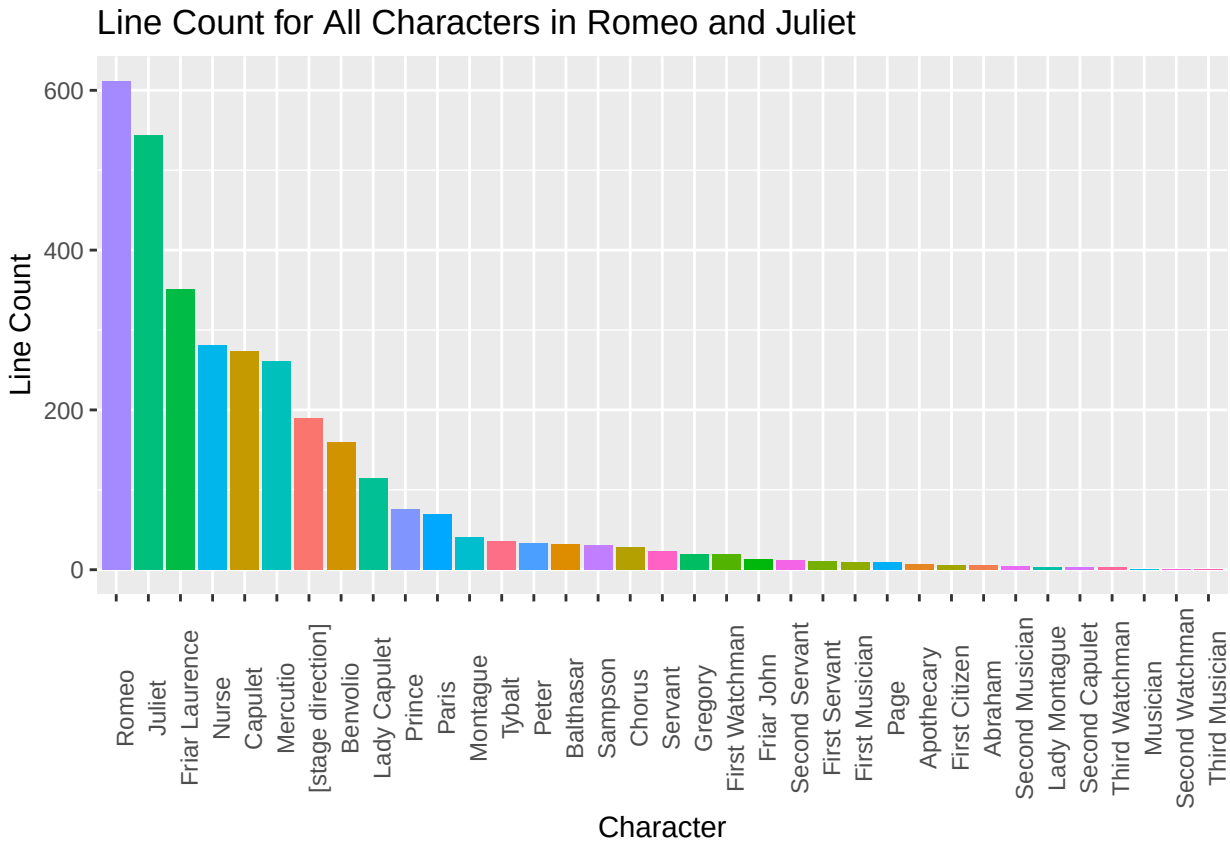



```
knitr::kable(line_count_romjul[,1:2], format="markdown")
```

character	line_count
Romeo	612
Juliet	544
Friar Laurence	351
Nurse	281
Capulet	273
Mercutio	261
[stage direction]	189
Benvolio	160
Lady Capulet	115
Prince	76
Paris	70
Montague	41
Tybalt	36
Peter	33
Balthasar	32
Sampson	31
Chorus	28
Servant	23
Gregory	20
First Watchman	19
Friar John	13
Second Servant	12
First Servant	10
First Musician	9

character	line_count
Page	9
Apothecary	7
First Citizen	6
Abraham	5
Second Musician	4
Lady Montague	3
Second Capulet	3
Third Watchman	3
Musician	1
Second Watchman	1
Third Musician	1

```
ggplot(data = line_count_romjul, aes(y=line_count,x=reorder(character, -line_count)))+geom_bar(stat='id
```

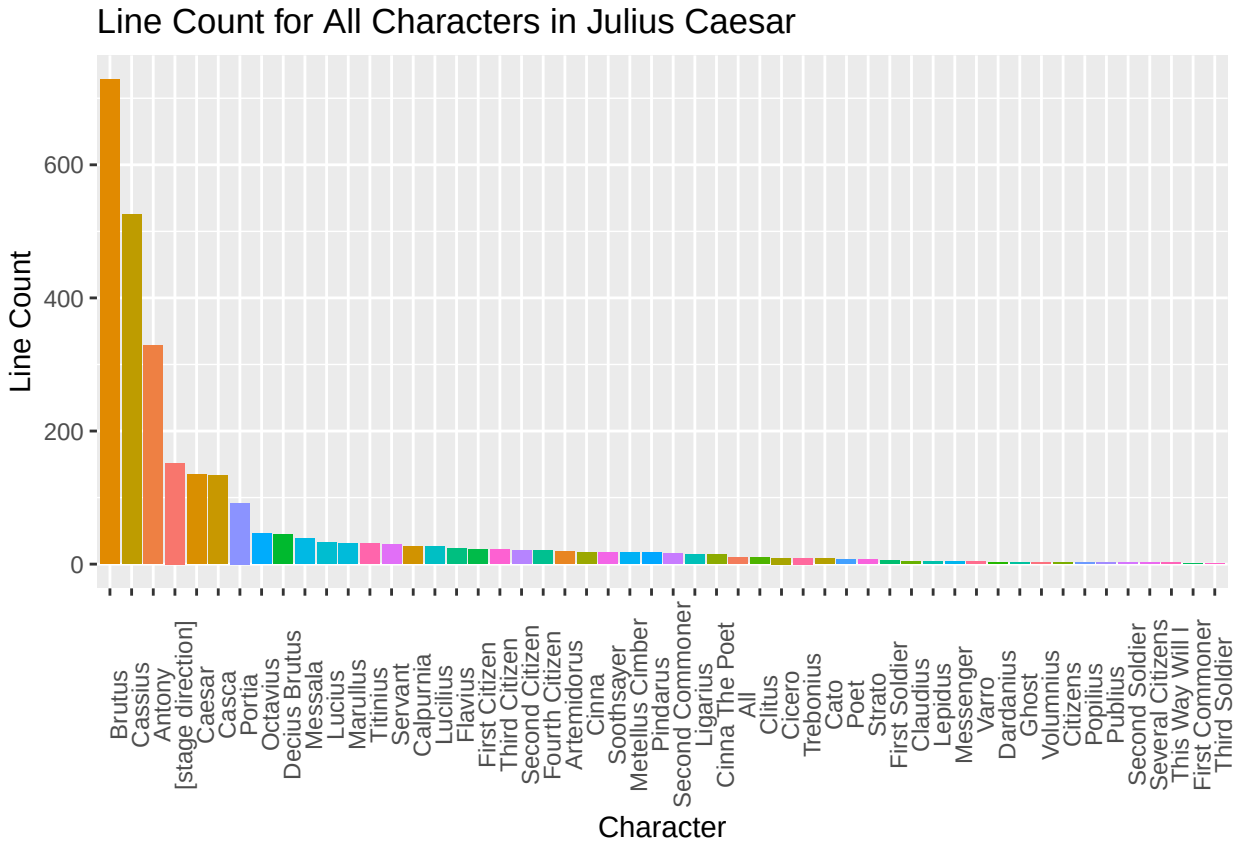



```
knitr::kable(line_count_jc[,1:2], format="markdown")
```

character	line_count
Brutus	728
Cassius	525
Antony	329
[stage direction]	152
Caesar	135
Casca	133
Portia	92
Octavius	46

character	line_count
Decius Brutus	44
Messala	39
Lucius	33
Marullus	31
Titinius	31
Servant	30
Calpurnia	27
Lucilius	26
Flavius	24
First Citizen	22
Third Citizen	22
Second Citizen	21
Fourth Citizen	20
Artemidorus	19
Cinna	18
Soothsayer	18
Metellus Cimber	17
Pindarus	17
Second Commoner	16
Ligarius	15
Cinna The Poet	14
All	10
Clitus	10
Cicero	9
Trebonius	9
Cato	8
Poet	7
Strato	7
First Soldier	5
Claudius	4
Lepidus	4
Messenger	4
Varro	4
Dardanius	3
Ghost	3
Volumnius	3
Citizens	2
Popilius	2
Publius	2
Second Soldier	2
Several Citizens	2
This Way Will I	2
First Commoner	1
Third Soldier	1

```
ggplot(data = line_count_jc, aes(y=line_count,x=reorder(character, -line_count)))+geom_bar(stat='identi
```

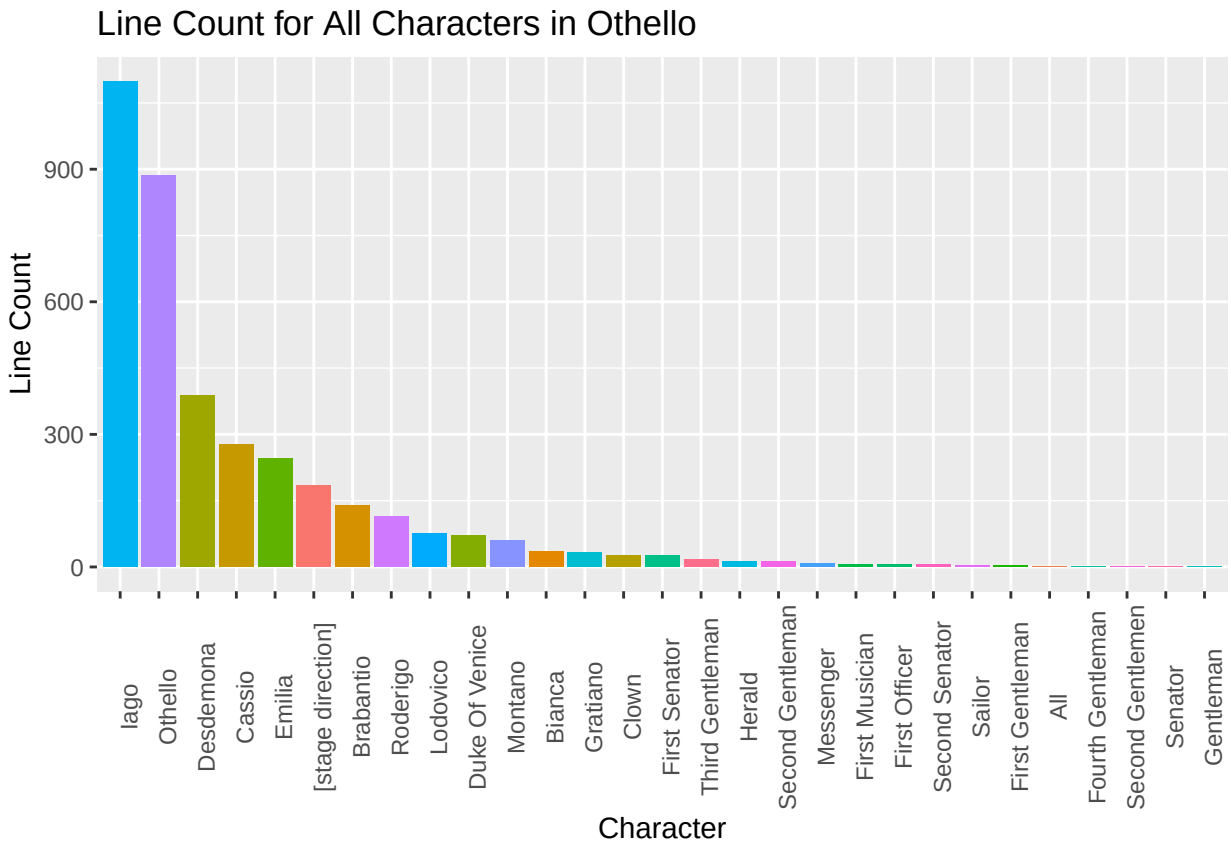



```
knitr::kable(line_count_othello[,1:2], format="markdown")
```

character	line_count
Iago	1099
Othello	887
Desdemona	388
Cassio	277
Emilia	245
[stage direction]	185
Brabantio	139
Roderigo	114
Lodovico	76
Duke Of Venice	71
Montano	61
Bianca	34
Gratiano	32
Clown	27
First Senator	26
Third Gentleman	17
Herald	12
Second Gentleman	12
Messenger	9
First Musician	5
First Officer	5
Second Senator	5
Sailor	4
First Gentleman	3

character	line_count
All	2
Fourth Gentleman	2
Second Gentlemen	2
Senator	2
Gentleman	1

```
ggplot(data = line_count_othello, aes(y=line_count,x=reorder(character, -line_count)))+geom_bar(stat='i
```



As we can see in Othello, Othello does not have the most even while being the protagonist in the play. It is the antagonist that has the most lines, Iago.

Next I want to check which acts have the most lines and which character in those acts have the most lines. Firstly I will analysis what acts has the most lines

```
line_act_hamlet <- hamlet %>%
  group_by(act) %>%
  summarize(line_count = n())
line_act_hamlet <- arrange(line_act_hamlet, -line_count)

line_act_macbeth <- macbeth %>%
  group_by(act) %>%
  summarize(line_count = n())
line_act_macbeth <- arrange(line_act_macbeth, -line_count)

line_act_romjul <- romeo_juliet %>%
  group_by(act) %>%
```

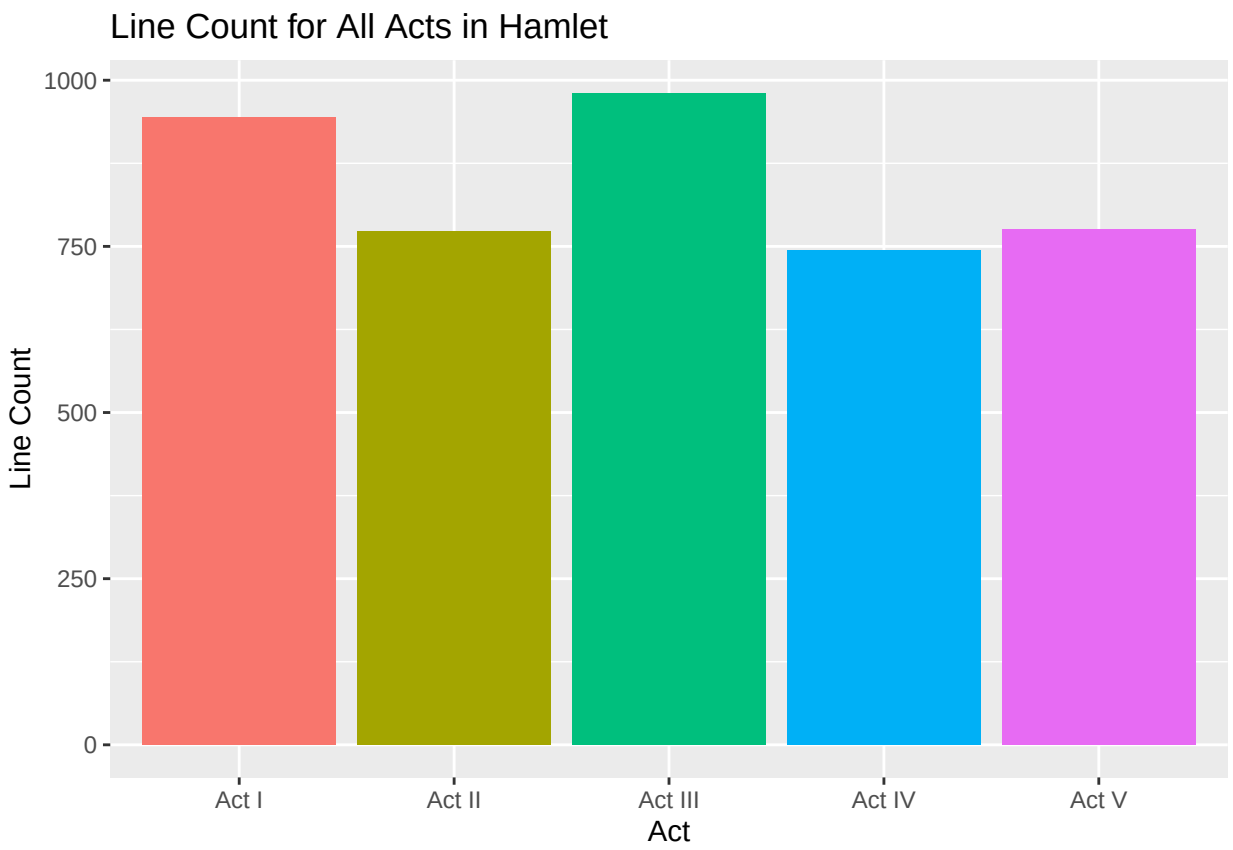
```
  summarize(line_count = n())
line_act_romjul <- arrange(line_act_romjul, -line_count)

line_act_jc <- julius_caesar %>%
  group_by(act) %>%
  summarize(line_count = n())
line_act_jc <- arrange(line_act_jc, -line_count)

line_act_othello <- othello %>%
  group_by(act) %>%
  summarize(line_count = n())
line_act_othello <- arrange(line_act_othello, -line_count)
```

Now to view the dataframe and graph them accordingly.

```
line_act_hamlet[,1:2]
```

```
## # A tibble: 5 x 2
##   act     line_count
##   <chr>        <int>
## 1 Act III        981
## 2 Act I          944
## 3 Act V          775
## 4 Act II         773
## 5 Act IV         744
```

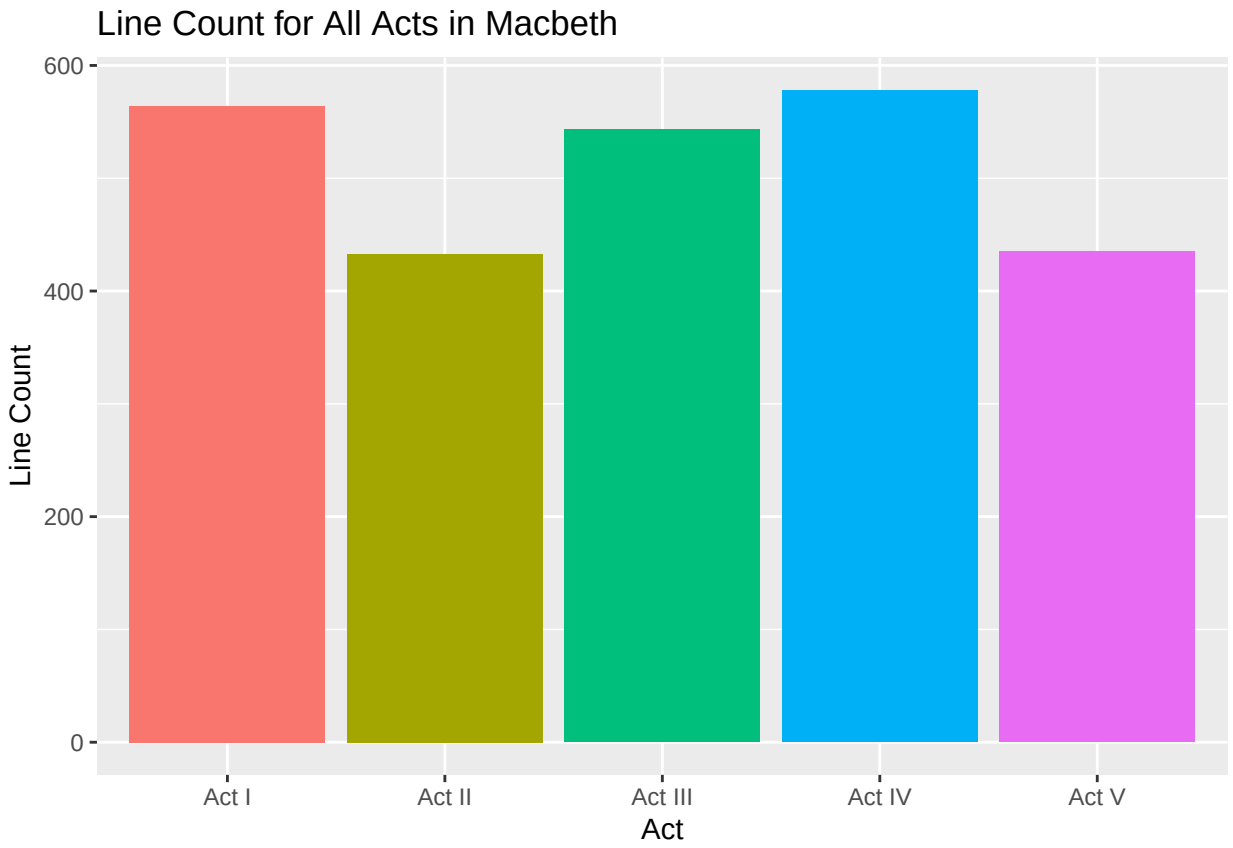
```
ggplot(data = line_act_hamlet, aes(x=act, y=line_count))+geom_bar(stat='identity', aes(fill=act))+labs(
```



```
line_act_macbeth[,1:2]
```

```
## # A tibble: 5 x 2
##   act     line_count
##   <chr>        <int>
## 1 Act IV         578
## 2 Act I          564
## 3 Act III        543
## 4 Act V          435
## 5 Act II         433
```

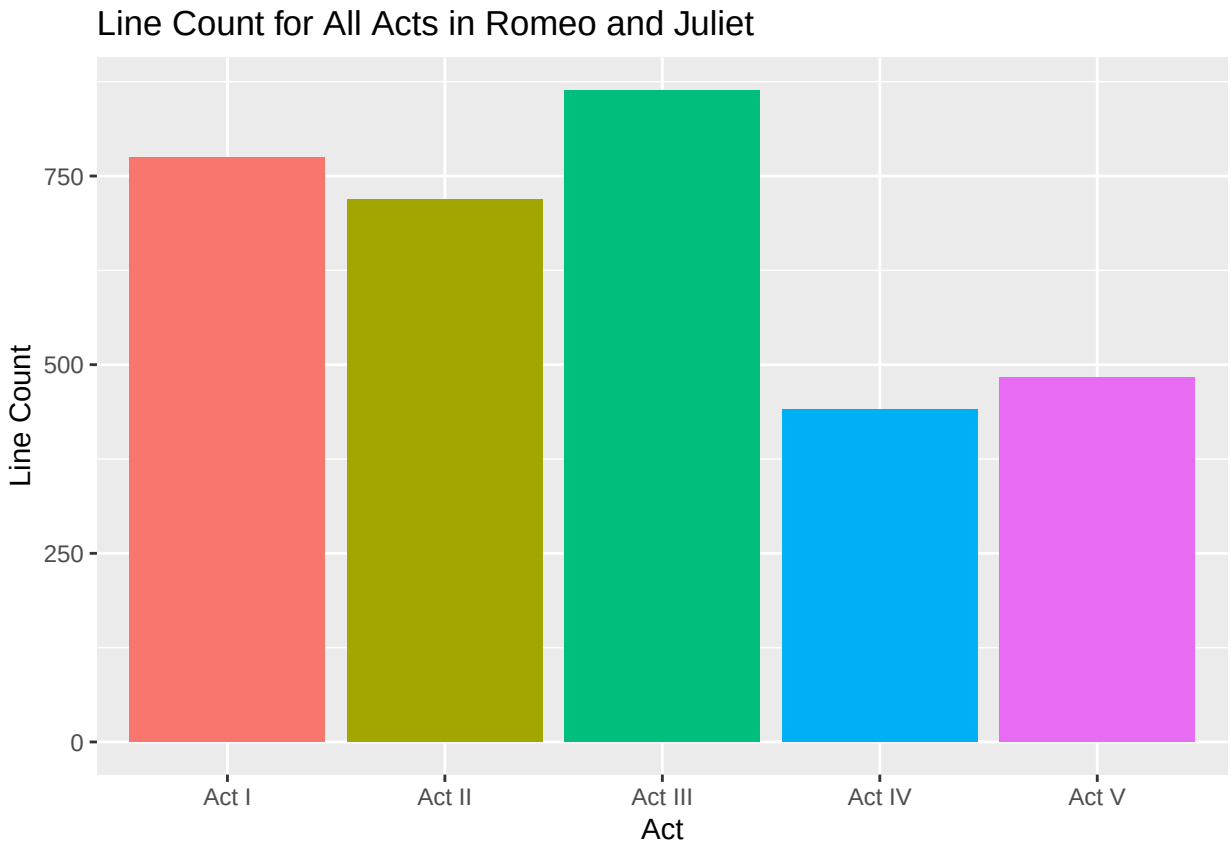
```
ggplot(data = line_act_macbeth, aes(x=act, y = line_count))+geom_bar(stat='identity', aes(fill=act))+la
```




```
line_act_romjul[,1:2]
```

```
## # A tibble: 5 x 2
##   act     line_count
##   <chr>        <int>
## 1 Act III        864
## 2 Act I          775
## 3 Act II         719
## 4 Act V          483
## 5 Act IV         441
```

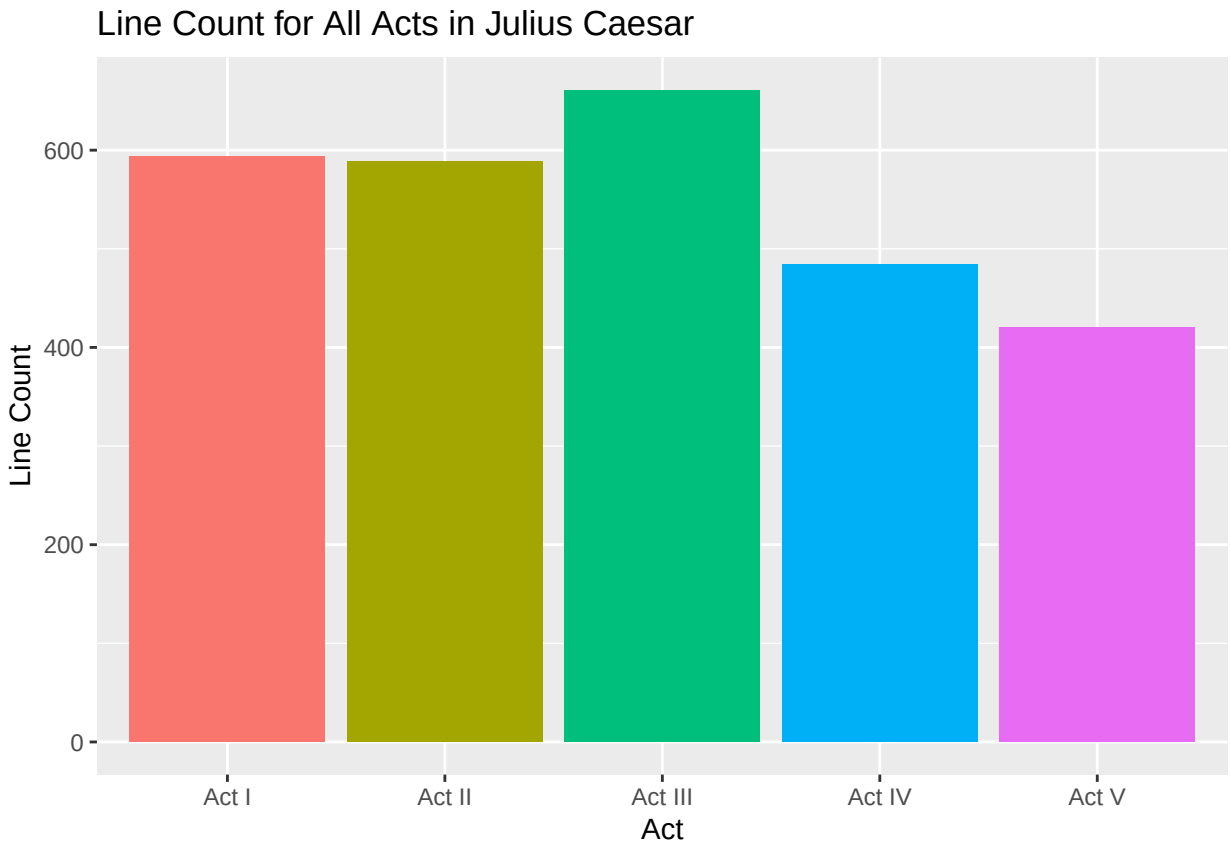
```
ggplot(data = line_act_romjul, aes(x=act, y=line_count))+geom_bar(stat='identity', aes(fill=act))+labs(
```




```
line_act_jc[,1:2]
```

```
## # A tibble: 5 x 2
##   act     line_count
##   <chr>        <int>
## 1 Act III        661
## 2 Act I          594
## 3 Act II         589
## 4 Act IV         484
## 5 Act V          420
```

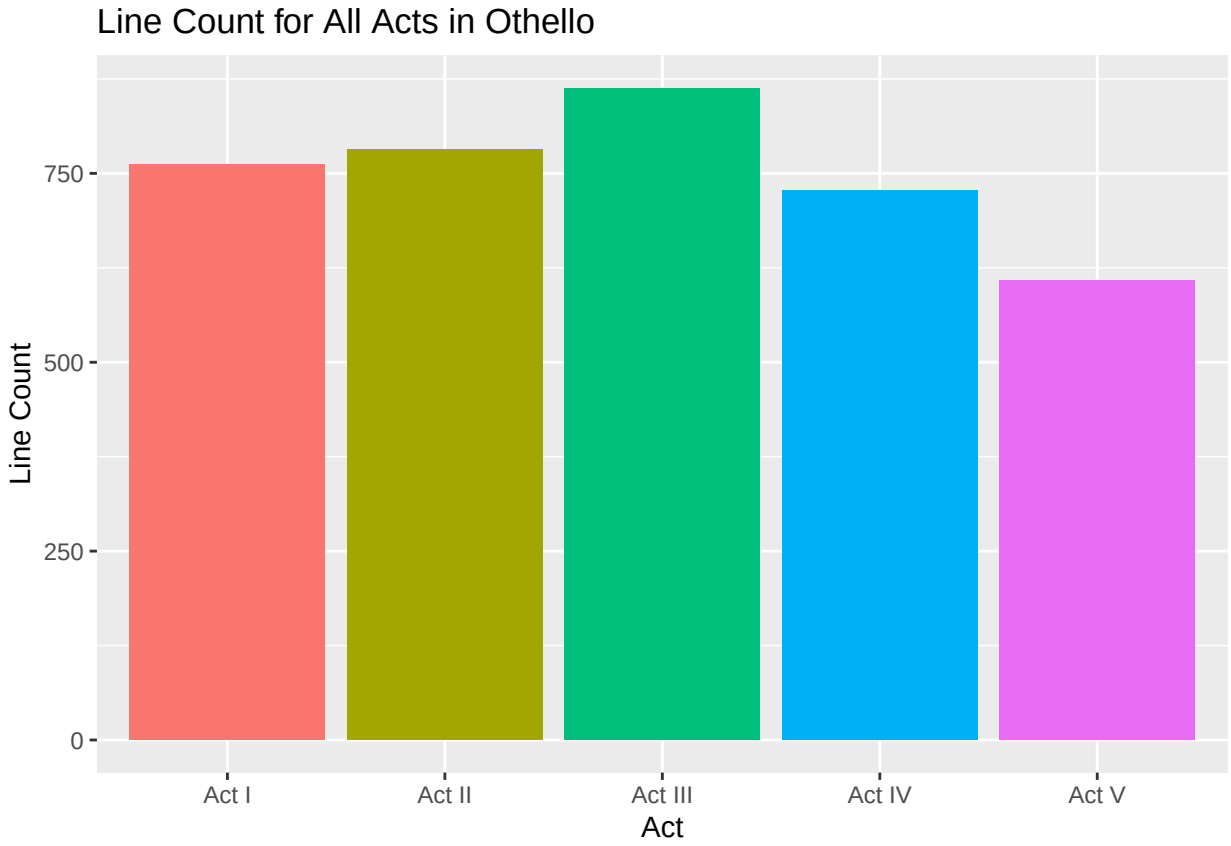
```
ggplot(data = line_act_jc, aes(x=act, y=line_count))+geom_bar(stat='identity', aes(fill=act))+labs(titl
```




```
line_act_othello[,1:2]
```

```
## # A tibble: 5 x 2
##   act     line_count
##   <chr>        <int>
## 1 Act III        863
## 2 Act II         782
## 3 Act I          762
## 4 Act IV         727
## 5 Act V          608
```

```
ggplot(data = line_act_othello, aes(x=act, y=line_count))+geom_bar(stat='identity', aes(fill=act))+labs
```



We can see in that Act III tends to have the most lines except for Macbeth where it was Act IV that had the most lines. Now to see which character spoke the most in those scenes.

```
act_hamlet <- hamlet %>%
  filter(act == "Act III")
fav_char_hamlet <- act_hamlet %>%
  group_by(character) %>%
  summarize(line_count = n())
fav_char_hamlet <- arrange(fav_char_hamlet, -line_count)

act_macbeth <- macbeth %>%
  filter(act == "Act IV")
fav_char_macbeth <- act_macbeth %>%
  group_by(character) %>%
  summarize(line_count = n())
fav_char_macbeth <- arrange(fav_char_macbeth, -line_count)

act_romjul <- romeo_juliet %>%
  filter(act == "Act III")
fav_char_romjul <- act_romjul %>%
  group_by(character) %>%
  summarize(line_count = n())
fav_char_romjul <- arrange(fav_char_romjul, -line_count)

act_jc <- julius_caesar %>%
  filter(act == "Act III")
fav_char_jc <- act_jc %>%
  group_by(character) %>%
```

```
  summarize(line_count = n())
fav_char_jc <- arrange(fav_char_jc, -line_count)

act_othello <- othello %>%
  filter(act == "Act III")
fav_char_othello <- act_othello %>%
  group_by(character) %>%
  summarize(line_count = n())
fav_char_othello <- arrange(fav_char_othello, -line_count)
```

Now to find which is the character will the most lines during the acts with the most lines.

```
fav_char_hamlet[1,]
```

```
## # A tibble: 1 x 2
##   character line_count
##   <chr>          <int>
## 1 Hamlet           502
```

```
fav_char_macbeth[1,]
```

```
## # A tibble: 1 x 2
##   character line_count
##   <chr>          <int>
## 1 Malcolm          143
```

```
fav_char_romjul[1,]
```

```
## # A tibble: 1 x 2
##   character line_count
##   <chr>          <int>
## 1 Juliet           221
```

```
fav_char_jc[1,]
```

```
## # A tibble: 1 x 2
##   character line_count
##   <chr>          <int>
## 1 Antony           246
```

```
fav_char_othello[1,]
```

```
## # A tibble: 1 x 2
##   character line_count
##   <chr>          <int>
## 1 Othello          259
```

Interesting, we can see a majority of the time one of the main protagonists are mainly talking during the most vocal acts. However, two plays, Macbeth and Julius Caesar, it is Malcolm and Anthony talking the most during their plays most active act.

Looking back at the graphs with the line count for each character we can see that both of these character ranked third for their respectively plays.

Considering this, I want to look into how much these two outliers talked throughout each act to see if they talk a majority of the time in one act.

First, let's start with Malcolm in Macbeth.

```
malcolm_macbeth <- macbeth %>%
  filter(character == "Malcolm")

mal_act_macbeth <- malcolm_macbeth %>%
  group_by(act) %>%
  summarize(line_count = n())

malcolm_macbeth <- arrange(mal_act_macbeth, -line_count)

malcolm_macbeth
```

```
## # A tibble: 4 x 2
##   act    line_count
##   <chr>       <int>
## 1 Act IV        143
## 2 Act V          39
## 3 Act I          16
## 4 Act II         14
```

Next, Antony in Julius Caesar.

```
antony_jc <- julius_caesar %>%
  filter(character == "Antony")

ant_act_jc <- antony_jc %>%
  group_by(act) %>%
  summarize(line_count = n())
antony_jc <- arrange(ant_act_jc, -line_count)

antony_jc
```

```
## # A tibble: 5 x 2
##   act     line_count
##   <chr>        <int>
## 1 Act III        246
## 2 Act IV          38
## 3 Act V           38
## 4 Act I            6
## 5 Act II           1
```

Now we can see that they did talk a majority of the talk in those acts specifically.

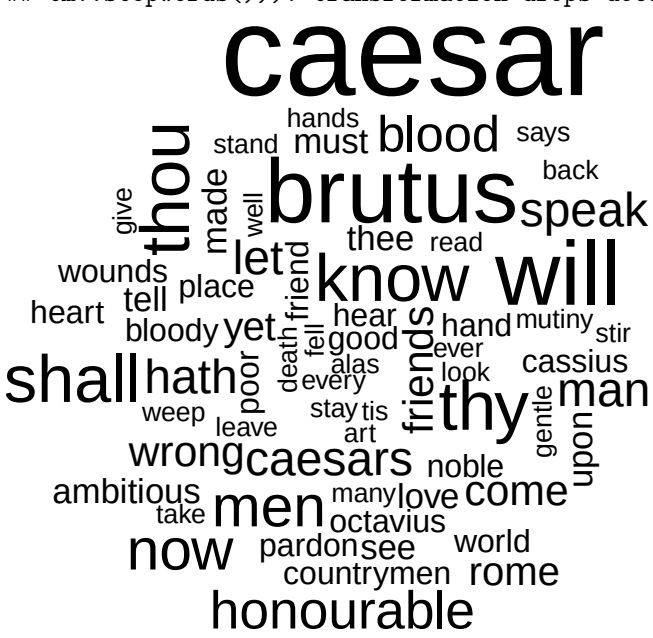
I want to see what they were saying using a word cloud to see what words they used the most.

Here is the word cloud from Julius Caesar using Antony's dialouge to fill it.

```
lines_ant_jc <- julius_caesar %>%
  filter(character == "Antony", act == "Act III")

lines_ant <- lines_ant_jc %>%
  unnest_tokens(word, dialogue)
words_ant <- lines_ant$word

wordcloud(words_ant)
```

```
## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
```

```
## Warning in tm_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,
## tm::stopwords())): transformation drops documents
```



Now for Malcolm in Macbeth.

```
lines_mal_macbeth <- macbeth %>%
  filter(character == "Malcolm", act == "Act IV")

lines_mal <- lines_mal_macbeth %>%
  unnest_tokens(word, dialogue)
words_mal <- lines_mal$word

wordcloud(words_mal)
```

```
## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
```

```
## Warning in tm_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,
## tm::stopwords())): transformation drops documents
```

Now we can see what is their most frequent words used was looks like the name of the main protagonist is frequently used. There were a couple of interesting things that observed and now I can understand who the main character was in each play by looking at the character with the most lines and cross checking with the internet.

I learned that Julius Caesar's main protagonist was not in fact himself but Brutus. I learned that the King in Macbeth had the most lines during the most vocal act as well as Antony in Julius Caesar. Both of them not being the main character but support characters. I also learned that Shakespeare usually put the bulk of the dialogue towards to middle to the end of the play.

Side note: The only play I knew prior to this analysis was Romeo and Juliet. :)