I created CSV file featuring movies:

```
Movies_prep <- read_csv("R/movies/Movies_prep.csv")
Movies_prep [1:5,]
```

X1	title	year	duration	genre	feature	type	IMDB	unit duration
1	12 angry men	1957	96	Drama	IMDB Top 100	movie	8.9	96
2	A beautiful mind	2001	135	Drama	Academy Awards	movie	8.2	135
3	American Beauty	1999	122	Drama	IMDB Top 100	movie	8.3	122
4	Anger management	2003	105	Comedy	Other	movie	6.2	105
5	Argo	2012	120	Thriller	Academy Awards	movie	7.7	120

Resulting table will be plugged with three more parameters needed to be calculated.

1. Number of words (for entire movie).
2. Words per minute (Number of words / Movie length).
3. Vocabulary (Number of unique words per 1000 words).

```
library (tidyverse)
library (tidytext)
library (textstem)
library (gt)
```

Bind clean text (described in Movies text analysis. Part 1) to the titles

```
movies_ <- select (Movies_prep, title = title)
movies_bind <- cbind (movies_, text)
```

Let`s use *tidytext* package to Unnest text to words

```
 movies_words  <- unnest_tokens (movies_bind, word, text)
```

Simply calculate nwords and words per minute for each movie

```
title_ <- movies_bind$title
movies_nword <- function (i){movies_nwords1<- movies_words %>% filter (title==i)
%>% nrow ()
movies_nwords1}
movies_nwords <- sapply (title_, movies_nword)

Movies_prep1 <-  Movies_prep %>% mutate (words = movies_nwords, wpminute = round
(movies_nwords/duration))
```

Let`s look at the *Words per minute* parameter first.

```
wpm_summary <- Movies_prep1$wpminute %>% summary ()
```

wpm_summary

Min. 1st Qu. Median Mean 3rd Qu. Max.

37.00 66.00 80.00 85.89 104.00 182.00

Let`s check ten most wordy movies in our dataset

```
Movies_prep1_GT <- Movies_prep1 %>% select (title, wpminute, genre, type)
Movies_preptop_GT <- Movies_prep1_GT %>% arrange (desc(wpminute)) %>% slice
(1:10) %>% gt ()
```

I added tiny command gt () from *gt* package and my slice became nice looking table:

title	wpminute	genre	type
Shark Tunk	182	Reality Show	TV
Horrible Bosses	169	Comedy	movie
12 angry men	154	Drama	movie
South Park	151	Animation	TV
The Social Network	140	Drama	movie
Suits	134	Drama	TV
How I Met Your Mother	132	Comedy	TV
Murder Mystery	129	Action & Adventure	movie
Fahrenheit 9-11	127	Documentary	movie
Four christmasses	126	Romance	movie

The same for ten least wordy movies

```
Movies_prepbot_GT <- Movies_prep1_GT %>% arrange (wpminute) %>% slice (1:10) %>%
gt ()
```

title	wpminute	genre	type
The Lord of the Rings Return of the King	37	Fantasy & Sci-Fi	movie
Titanic	42	Drama	movie
The Lord of the Rings The Two Towers	47	Fantasy & Sci-Fi	movie
The Umbrella Academy	47	Action & Adventure	TV
Hulk	49	Super Hero	movie
Terminator 2 Judgment Day	50	Action & Adventure	movie
Star Wars Episode VII	52	Fantasy & Sci-Fi	movie
The Lord of the Rings The Fellowship of the Ring	53	Fantasy & Sci-Fi	movie
Batman	57	Super Hero	movie
The Godfather II	57	Crime	movie

Only Reality Show I added as control value is obvious outlier. While the most silent are well known epic movies.

We will explore gt package more deep a later.

Now, let`s check the range for Total Number of words

```
words_summary <- Movies_prep1$words %>% summary ()
```

Min. 1st Qu. Median Mean 3rd Qu. Max.
5981 8619 10311 10815 12467 18710

I wish all of them were exact 10,000 words length. Or any other but equal length for all movies. Real life is not so round. Unlike music vocabulary, we cannot take “99.7 songs” to have the same length for every peer. Why we need that? We cannot properly compare Number of unique words within different pieces of text unless they all are the same length. Any one sentence has up to 100% words uniqueness, 100 sentences – up to 50%. Any entire movie – much less due to marginal saturation e.g. usage the words we already used. Before we solve this problem we should lemmatize clean text.

```
movies_lem<- lemmatize_words (movies_words$word)
movies <- movies_words %>% mutate (word = movies_lem)
```

And, vocabulary unique words per 1000 for each movie (I use minimal length along the all movies in dataset with sampling all movies text the same size = length of the shortest (N of words) movie.

```
nwords_min <- min(movies_nwords)
vocab1 <- function (z) {vocab2<- movies_words %>% filter (title==z)
vocab3<- as.data.frame (replicate (5, sample (vocab2$word, nwords_min, replace =
FALSE)), stringsAsFactors = FALSE)
vocab4 <- round (mean (sapply (sapply (vocab3, unique), length))/nwords_min
*1000)
vocab4}
vocab <- sapply (title_, vocab1)

Summary (vocab)
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
152.0   176.0   188.0   188.7   200.0   251.0

Movies_final <-  Movies_prep1 %>% mutate (vocabulary = vocab)
```

I plan to compare and visualize how wide vocabulary is within iconic movies, different genres, Movies vs TV Shows, IMDB TOP 100 vs Box Office All Time Best etc. in my next post (Movies text analysis. Part 3.)

For this part I want to explore further *gt* package and display movies ranking by its vocabulary:

```
Movies_final_GT <- Movies_final %>% select (title, vocabulary, genre, type,
feature) %>%
  top_n (20, vocabulary) %>% mutate (N = seq (20))
Movies_final_GT <- Movies_final_GT [,c(6,1,2,3,4,5)] %>% gt() %>%
  tab_header(
  title = "Number of Unique Words Used in Movie / TV Show (per each 1000
words)")
```

Using gt package we put top 15 movies by number of unique words in table. This time with header.

Number of Unique Words Used in Movie / TV Show (per each 1000 words)

N	title	vocabulary	genre	type	feature
1	The Simpsons	251	Animation	TV	IMDB Top 100
2	The Shindler'S List	250	Drama	movie	IMDB Top 100
3	Orange Is New Black	228	Comedy	TV	Netflix Production
4	Argo	223	Thriller	movie	Academy Awards
5	Fahrenheit 9-11	221	Documentary	movie	Other
6	The Darkest Hour	218	History	movie	Academy Awards
7	Black Mirror	218	Drama	TV	Netflix Production
8	Fight club	217	Drama	movie	IMDB Top 100
9	V For Vendetta	216	Super Hero	movie	Other
10	House Of Cards	215	Drama	TV	Netflix Production
11	The Big Bang Theory	213	Comedy	TV	Academy Awards
12	Iron Man3	212	Super Hero	movie	Box Office
13	The Silence of the Lamb	211	Thriller	movie	IMDB Top 100
14	Laundromat	211	Comedy	movie	Netflix Production
15	A beautiful mind	210	Drama	movie	Academy Awards

And we can easily save the table as graphical output.

```
gtsave(Movies_final_GT, filename = 'Movies_final_GT.png')
gtsave(Movies_final_GT, filename = 'Movies_final_GT.pdf')
```

Let`s format the table a bit.

```
Movies_final_GTT <- Movies_final_GT %>%
    tab_style(
    style = list(
      cell_text(weight = "bold")),
    locations = cells_column_labels(
      columns = vars(N, title, vocabulary, genre, type, feature))) %>%
  tab_style(
  style = list(
      cell_text(weight = "bold")),
locations = cells_body(
  columns = vars(title, vocabulary))) %>%
  tab_style(
    style = list(
      cell_text(style = "italic")),
locations = cells_body(
        columns = vars(title, type)))
```

Number of Unique Words Used in Movie / TV Show (per each 1000 words)

N	**title**	**vocabulary**	**genre**	**type**	**feature**
1	***The Simpsons***	**251**	Animation	*TV*	IMDB Top 100
2	***The Shindler'S List***	**250**	Drama	*movie*	IMDB Top 100
3	***Orange Is New Black***	**228**	Comedy	*TV*	Netflix Production
4	***Argo***	**223**	Thriller	*movie*	Academy Awards
5	***Fahrenheit 9-11***	**221**	Documentary	*movie*	Other
6	***The Darkest Hour***	**218**	History	*movie*	Academy Awards
7	***Black Mirror***	**218**	Drama	*TV*	Netflix Production
8	***Fight club***	**217**	Drama	*movie*	IMDB Top 100
9	***V For Vendetta***	**216**	Super Hero	*movie*	Other
10	***House Of Cards***	**215**	Drama	*TV*	Netflix Production
11	***The Big Bang Theory***	**213**	Comedy	*TV*	Academy Awards
12	***Iron Man3***	**212**	Super Hero	*movie*	Box Office
13	***The Silence of the Lamb***	**211**	Thriller	*movie*	IMDB Top 100
14	***Laundromat***	**211**	Comedy	*movie*	Netflix Production
15	***A beautiful mind***	**210**	Drama	*movie*	Academy Awards

We can change cells and text colors and even make it conditional (for instance, only ‘type == TV’).

```
Movies_final_GTT <- Movies_final_GTT %>%
  tab_style(
    style = list(
      cell_text(color = "blue")),
locations = cells_body(
      columns = vars(vocabulary))) %>%
  tab_style(
    style = list(
      cell_fill(color = "#F9E3D6")),
locations = cells_body(
      columns = vars(type),
      rows = type == "TV"))
```

Number of Unique Words Used in Movie / TV Show (per each 1000 words)

N	title	vocabulary	genre	type	feature
1	***The Simpsons***	**251**	Animation	*TV*	IMDB Top 100
2	***The Shindler`S List***	**250**	Drama	*movie*	IMDB Top 100
3	***Orange Is New Black***	**228**	Comedy	*TV*	Netflix Production
4	***Argo***	**223**	Thriller	*movie*	Academy Awards
5	***Fahrenheit 9-11***	**221**	Documentary	*movie*	Other
6	***The Darkest Hour***	**218**	History	*movie*	Academy Awards
7	***Black Mirror***	**218**	Drama	*TV*	Netflix Production
8	***Fight club***	**217**	Drama	*movie*	IMDB Top 100
9	***V For Vendetta***	**216**	Super Hero	*movie*	Other
10	***House Of Cards***	**215**	Drama	*TV*	Netflix Production
11	***The Big Bang Theory***	**213**	Comedy	*TV*	Academy Awards
12	***Iron Man3***	**212**	Super Hero	*movie*	Box Office
13	***The Silence of the Lamb***	**211**	Thriller	*movie*	IMDB Top 100
14	***Laundromat***	**211**	Comedy	*movie*	Netflix Production
15	***A beautiful mind***	**210**	Drama	*movie*	Academy Awards

That`s it for tables and for Part 2 of my Movies text analysis. In Part 3 I plan to use *ggplot2* to visualize and compare how wide vocabulary is within iconic movies, different genres, Movies vs TV Shows, IMDB TOP 100 vs Box Office All Time Best etc. See ya.