Code Appendix

Haley Jiang, Handi Yang

12/10/2022

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(cmu.textstat)
```

```
## Loading required package: quanteda.extras
## Loading required package: vnc
## Loading required package: mda.biber
## Loading required package: ngramr.plus
```

```r
library(readtext)
library(quanteda)
```

```
## Package version: 3.2.3
## Unicode version: 14.0
## ICU version: 70.1
## Parallel computing: 8 of 8 threads used.
## See https://quanteda.io for tutorials and examples.
```

```r
library(quanteda.textstats)
library(ggraph)
library(corpus)
library(syuzhet)
library(udpipe)
# library(pseudobibeR)
library(readr)
library(nFactors)
```

```
## Loading required package: lattice
##
## Attaching package: 'nFactors'
```

```
##
## The following object is masked from 'package:lattice':
##
##     parallel
```

```
library(future.apply)
```

```
## Loading required package: future
```

```
library(wordcloud)
```

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

```
library(tidytext)
library(tm)
```

```
## Loading required package: NLP
##
## Attaching package: 'NLP'
##
## The following objects are masked from 'package:quanteda':
##
##     meta, meta<-
##
## The following object is masked from 'package:ggplot2':
##
##     annotate
##
##
## Attaching package: 'tm'
##
## The following object is masked from 'package:quanteda':
##
##     stopwords
```

Data preparation

We will first load data, prepare corpus, and extract tokens.

```
# load data
# setwd("/Users/yanghandi/Desktop")
script_df <- readtext("./468_final_project/data") %>%
   mutate(text = preprocess_text(text,
                                 contractions = TRUE,
                                 hypens = TRUE,
                                 punctuation = TRUE,
                                 # lower_case = TRUE,
                                 accent_replace = TRUE,
                                 remove_numbers = TRUE))
```

```
script <- corpus(script_df)

corpus <- script_df %>%
  mutate(genre = str_extract(doc_id, "^[a-z]+")) %>%
  dplyr::select(doc_id, text)

# get raw tokens
script_tokens <- tokens(script)
```

We will then assigned meta data to our corpus.

```
# assign meta data
movie_genre <- str_extract(script_df$doc_id, "^[a-z]+")
docvars(script, field = "genre") <- movie_genre
release_year <- str_extract(script_df$doc_id,"(\\d+)(?!.*\\d)")
# release_year <- c(2005, 2008, 2012, 2010, 2020, 2014, 1998, 2002, 2000, 2006, 2017)
docvars(script, field = "year") <- release_year
knitr::kable(script %>% summary() %>% base::subset(select=-c(Types, Sentences)), caption = "Summary of 
```

Table 1: Summary of Norlan film corpus.

Text	Tokens	genre	year
action__batman__begins-2005.pdf	27313	action	2005
action__dark1__knight-2008.pdf	30700	action	2008
action__dark2__knight__rises-2012.txt	29713	action	2012
action__incepetion-2010.txt	28502	action	2010
action__tenet-2020.pdf	30938	action	2020
scientific__interstallar-2014.pdf	24702	scientific	2014
scifi__interstallar-2014.pdf	24702	scifi	2014
thriller__following-1998.pdf	16852	thriller	1998
thriller__insomnia-2002.pdf	23741	thriller	2002
thriller__memento-2000.pdf	31433	thriller	2000
thriller__prestige-2006.pdf	27674	thriller	2006
war__dunkirk-2017.pdf	14775	war	2017

To delete the character names before the lines and specific names in the scripts, we will annotate the tokens and delete proper nouns.

```
corpus_split <- split(corpus, seq(1, nrow(corpus), by = 10))

# remove proper noun from dataset
library(future.apply)
ncores <- 4L
plan(multisession, workers = ncores)
annotate_splits <- function(corpus_text) {
  ud_model <- udpipe_load_model("english-ewt-ud-2.5-191206.udpipe")
  x <- data.table::as.data.table(udpipe_annotate(ud_model, x = corpus_text$text,
                                                  doc_id = corpus_text$doc_id))
  return(x)
}
annotation <- future_lapply(corpus_split, annotate_splits, future.seed = T)
annotation <- data.table::rbindlist(annotation)
anno_edit <- annotation %>%
  dplyr::select(doc_id, sentence_id, token_id, token, lemma, upos, xpos, head_token_id, dep_rel) %>%
  rename(pos = upos, tag = xpos)
anno_edit <- structure(anno_edit, class = c("spacyr_parsed", "data.frame"))
tkns <- as.tokens(anno_edit, include_pos = "pos", concatenator = "_")
doc_categories <- names(tkns) %>%
  data.frame(genre = .) %>%
  mutate(genre = str_extract(genre, "^[a-z]+"))
docvars(tkns) <- doc_categories
script_dfm <- dfm(tkns)
script_dfm <- tkns %>%
  tokens_select("^.*[a-zA-Z0-9]+.*_propn", selection = "remove", valuetype = "regex", case_insensitive
  dfm()
```

Frequency Analysis

```
# frequency
freq_df <- textstat_frequency(script_dfm) %>%
  data.frame(stringsAsFactors = F)
knitr::kable(freq_df[1:20,], caption = "The 20 most frequent tokens in the script corpus.")
```

Table 2: The 20 most frequent tokens in the script corpus.

feature	frequency	rank	docfreq	group
the_det	19824	1	12	all
a_det	6795	2	12	all
i_pron	5432	3	12	all
you_pron	5039	4	12	all
to_part	4163	5	12	all
of_adp	4006	6	12	all
and_cconj	4005	7	12	all
it_pron	3584	8	12	all
his_pron	3505	9	12	all
in_adp	3122	10	12	all
he_pron	3093	11	12	all
at_adp	2898	12	12	all
to_adp	2700	13	12	all
is_aux	2406	14	12	all
on_adp	2207	15	12	all
n't_part	1953	16	12	all
up_adp	1740	17	12	all
we_pron	1698	18	12	all
looks_verb	1631	19	12	all
him_pron	1608	20	12	all

```
# war movie
war_dfm <- dfm_subset(script_dfm, genre == "war")
war_freq_df <- textstat_frequency(war_dfm) %>%
```

```r
  data.frame(stringsAsFactors = F)
knitr::kable(war_freq_df[1:20,], caption = "The 20 most frequent tokens in the war movie corpus.")
```

Table 3: The 20 most frequent tokens in the war movie corpus.

feature	frequency	rank	docfreq	group
the_det	1305	1	1	all
of_adp	213	2	1	all
a_det	211	3	1	all
to_part	205	4	1	all
at_adp	204	5	1	all
his_pron	187	6	1	all
he_pron	172	7	1	all
tommy_noun	162	8	1	all
and_cconj	151	9	1	all
to_adp	151	9	1	all
on_adp	141	11	1	all
i_pron	138	12	1	all
in_adp	137	13	1	all
it_pron	120	14	1	all
is_aux	116	15	1	all
peter_noun	116	15	1	all
water_noun	114	17	1	all
up_adp	111	18	1	all
alex_noun	105	19	1	all
dawson_noun	99	20	1	all

```r
# science fiction
sci_fi_dfm <- dfm_subset(script_dfm, genre == "scifi")
scie_freq_df <- textstat_frequency(sci_fi_dfm) %>%
  data.frame(stringsAsFactors = F)
knitr::kable(scie_freq_df[1:20,], caption = "The 20 most frequent tokens in the science fiction movie c
```

Table 4: The 20 most frequent tokens in the science fiction movie corpus.

feature	frequency	rank	docfreq	group
the_det	1338	1	1	all
cooper_noun	649	2	1	all
i_pron	503	3	1	all
a_det	464	4	1	all
to_part	376	5	1	all
you_pron	358	6	1	all
it_pron	316	7	1	all
of_adp	279	8	1	all
and_cconj	267	9	1	all
’s_part	248	10	1	all
murph_noun	246	11	1	all
is_aux	225	12	1	all
at_adp	223	13	1	all

feature	frequency	rank	docfreq	group
we_pron	214	14	1	all
in_adp	213	15	1	all
to_adp	198	16	1	all
nt_part	187	17	1	all
n't_part	183	18	1	all
brand_noun	176	19	1	all
on_adp	174	20	1	all

```r
# action
action_dfm <- dfm_subset(script_dfm, genre == "action")
act_freq_df <- textstat_frequency(action_dfm) %>%
  data.frame(stringsAsFactors = F)
knitr::kable(act_freq_df[1:20,], caption = "The 20 most frequent tokens in the actio n movie corpus.")
```

Table 5: The 20 most frequent tokens in the actio n movie corpus.

feature	frequency	rank	docfreq	group
the_det	9740	1	5	all
a_det	3230	2	5	all
i_pron	2531	3	5	all
you_pron	2492	4	5	all
to_part	1932	5	5	all
of_adp	1813	6	5	all
and_cconj	1654	7	5	all
his_pron	1565	8	5	all
in_adp	1529	9	5	all
it_pron	1484	10	5	all
at_adp	1372	11	5	all
he_pron	1266	12	5	all
to_adp	1248	13	5	all
-_punct	1174	14	1	all
is_aux	1077	15	5	all
on_adp	918	16	5	all
n't_part	852	17	5	all
we_pron	851	18	5	all
him_pron	804	19	5	all
up_adp	775	20	5	all

```r
# thriller
thriller_dfm <- dfm_subset(script_dfm, genre == "thriller")
thr_freq_df <- textstat_frequency(thriller_dfm) %>%
  data.frame(stringsAsFactors = F)
knitr::kable(thr_freq_df[1:20,], caption = "The 20 most frequent tokens in the thriller movie corpus.")
```

Table 6: The 20 most frequent tokens in the thriller movie corpus.

feature	frequency	rank	docfreq	group
the_det	6103	1	4	all

feature	frequency	rank	docfreq	group
a_det	2426	2	4	all
i_pron	1757	3	4	all
you_pron	1739	4	4	all
and_cconj	1666	5	4	all
his_pron	1473	6	4	all
of_adp	1422	7	4	all
he_pron	1361	8	4	all
it_pron	1348	9	4	all
to_part	1274	10	4	all
in_adp	1030	11	4	all
to_adp	905	12	4	all
at_adp	876	13	4	all
will_aux	844	14	4	all
on_adp	800	15	4	all
man_noun	782	16	4	all
is_aux	763	17	4	all
leonard_noun	725	18	1	all
n't_part	691	19	4	all
young_adj	601	20	4	all

Collocations

```
the_collocations <- collocates_by_MI(script_tokens, "the")
mc <- the_collocations %>% filter(col_freq >= 5 & MI_1 >= 5)
knitr::kable(head(the_collocations), digits = 3)
```

token	col_freq	total_freq	MI_1
lobs	4	1	5.972
paintings	4	1	5.972
proceedings	4	1	5.972
waterline	4	1	5.972
marquee	10	3	5.709
adjourns	3	1	5.557

```
knitr::kable(head(mc), digits = 3)
```

token	col_freq	total_freq	MI_1
marquee	10	3	5.709
barricading	6	2	5.557
brutalizes	6	2	5.557
electrics	6	2	5.557
ireland	6	2	5.557
panels	9	3	5.557

```
# action corpus select
script_df <- script_df %>%
  mutate(genre = str_extract(doc_id, "^[a-z]+"))
action_df <- subset(script_df, genre == "action")
action_tokens <- action_df %>%
  mutate(text = preprocess_text(text)) %>%
  corpus() %>%
  tokens(what="fastestword", remove_numbers=TRUE)

# get collocation far action corpus
the_collocations <- collocates_by_MI(action_tokens, "the")
mc <- the_collocations %>% filter(col_freq >= 5 & MI_1 >= 5)
knitr::kable(head(the_collocations), digits = 3)
```

token	col_freq	total_freq	MI_1
hoisting	4	1	5.924
lobs	4	1	5.924
paintings	4	1	5.924
revolver	4	1	5.924
swiped	4	1	5.924
airstream	3	1	5.509

```
knitr::kable(head(mc), digits = 3)
```

token	col_freq	total_freq	MI_1
boost	6	2	5.509
hunters	8	3	5.339
tossing	8	3	5.339
accounts	5	2	5.245
braking	5	2	5.245
crunching	5	2	5.245

```
# war corpus select
war_df <- subset(script_df, genre == "war")
war_tokens <- war_df %>%
  mutate(text = preprocess_text(text)) %>%
  corpus() %>%
  tokens(what="fastestword", remove_numbers=TRUE)

# get collocation far war corpus
the_collocations <- collocates_by_MI(war_tokens, "the")
mc <- the_collocations %>% filter(col_freq >= 5 & MI_1 >= 5)
knitr::kable(head(the_collocations), digits = 3)
```

token	col_freq	total_freq	MI_1
waterline	4	1	5.526
arc	3	1	5.111

token	col_freq	total_freq	MI_1
binoculars	3	1	5.111
bounces	6	2	5.111
burrowing	3	1	5.111
called	3	1	5.111

```r
knitr::kable(head(mc), digits = 3)
```

token	col_freq	total_freq	MI_1
bounces	6	2	5.111
duck	6	2	5.111

```r
# thriller corpus select
thriller_df <- subset(script_df, genre == "thriller")
thriller_tokens <- thriller_df %>%
  mutate(text = preprocess_text(text)) %>%
  corpus() %>%
  tokens(what="fastestword", remove_numbers=TRUE)

# get collocation far thriller corpus
the_collocations <- collocates_by_MI(thriller_tokens, "the")
mc <- the_collocations %>% filter(col_freq >= 5 & MI_1 >= 5)
knitr::kable(head(the_collocations), digits = 3)
```

token	col_freq	total_freq	MI_1
ocean	4	1	6.049
proceedings	4	1	6.049
marquee	10	3	5.786
adjourns	3	1	5.634
axis	3	1	5.634
baton	3	1	5.634

```r
knitr::kable(head(mc), digits = 3)
```

token	col_freq	total_freq	MI_1
marquee	10	3	5.786
gallery	8	3	5.464
gathering	8	3	5.464
withdraws	8	3	5.464
bedspread	5	2	5.371
empties	5	2	5.371

```r
# science fiction movie
sci_fi_df <- subset(script_df, genre == "scifi")
sci_fi_tokens <- sci_fi_df %>%
  mutate(text = preprocess_text(text)) %>%
  corpus() %>%
  tokens(what="fastestword", remove_numbers=TRUE)

# get collocation far science corpus
the_collocations <- collocates_by_MI(sci_fi_tokens, "the")
mc <- the_collocations %>% filter(col_freq >= 5 & MI_1 >= 5)
knitr::kable(head(the_collocations), digits = 3)
```

token	col_freq	total_freq	MI_1
arrangement	3	1	5.835
backside	3	1	5.835
barricading	3	1	5.835
brutalizes	3	1	5.835
catwalk	6	2	5.835
construction	3	1	5.835

```
knitr::kable(head(mc), digits = 3)
```

token	col_freq	total_freq	MI_1
catwalk	6	2	5.835
alongside	5	2	5.572
batter	5	2	5.572
center	5	2	5.572
rocks	5	2	5.572
top	9	4	5.420

```
# science fiction
sci_fi_kw <- textstat_keyness(script_dfm, docvars(script_dfm, "genre") == "scifi", measure = "lr")
kableExtra::kbl(head(sci_fi_kw), caption = "Tokens with the highest keyness values in the science ficti
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
```

Table 17: Tokens with the highest keyness values in the science fiction text-type when compared to the rest of the sample corpus.

feature	G2	p	n_target	n_reference
cooper_noun	1585.10	0	649	649
murph_noun	597.84	0	246	246
brand_noun	426.14	0	176	177
brand_adv	415.23	0	173	177
mann_noun	378.70	0	156	156
cooper_adj	315.48	0	130	130

```
# thriller
thriller_dfm <- dfm_subset(script_dfm, genre == "thriller")
thriller_kw <- textstat_keyness(script_dfm, docvars(script_dfm, "genre") == "thriller", measure = "lr")
```

```
kableExtra::kbl(head(thriller_kw), caption = "Tokens with the highest keyness values in the thriller te
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
```

Table 18: Tokens with the highest keyness values in the thriller text-type when compared to the rest of the sample corpus.

feature	G2	p	n_target	n_reference
leonard_noun	1656.21	0	725	0
will_aux	1007.98	0	844	222
young_adj	821.02	0	601	118
borden_noun	739.28	0	324	0
cutter_noun	712.82	0	318	1
angier_noun	657.07	0	288	0

```
# war
war_dfm <- dfm_subset(script_dfm, genre == "war")
war_kw <- textstat_keyness(script_dfm, docvars(script_dfm, "genre") == "war", measure = "lr")
kableExtra::kbl(head(war_kw), caption = "Tokens with the highest keyness values in the war text-type whe
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
```

Table 19: Tokens with the highest keyness values in the war text-type when compared to the rest of the sample corpus.

feature	G2	p	n_target	n_reference
tommy_noun	987.90	0	162	0
peter_noun	665.88	0	116	5
alex_noun	639.93	0	105	0
dawson_noun	594.03	0	99	0
farrier_noun	527.06	0	88	0
collins_noun	490.53	0	82	0

```
# action
action_kw <- textstat_keyness(script_dfm, docvars(script_dfm, "genre") == "action", measure = "lr")
kableExtra::kbl(head(action_kw), caption = "Tokens with the highest keyness values in the action text-t
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
```

Table 20: Tokens with the highest keyness values in the action text-type when compared to the rest of the sample corpus.

feature	G2	p	n_target	n_reference
protagonist_noun	1123.97	0	744	0
wayne_noun	996.88	0	660	0
-_punct	996.10	0	1174	161
wayne_pron	780.62	0	517	0
gordon_noun	724.69	0	480	0
batman_noun	603.80	0	400	0

```r
thri_action <- keyness_table(thriller_dfm, action_dfm)
kableExtra::kbl(head(thri_action), caption = "Tokens with the highest keyness values in the thriller mo
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
```

Table 21: Tokens with the highest keyness values in the thriller movie when compared to action genre.

Token	LL	LR	PV	AF_Tar	AF_Ref	Per_10.5_Tar	Per_10.5_Ref	DP_Tar	DP_Ref
leonard_noun	1311.53	11.06	0	725	0	712.17	0.00	0.68	NaN
will_aux	771.20	2.82	0	844	176	829.06	117.55	0.71	0.09
young_adj	594.17	3.03	0	601	108	590.36	72.13	0.79	0.53
borden_noun	586.12	9.90	0	324	0	318.26	0.00	0.73	NaN
cutter_noun	562.78	8.87	0	318	1	312.37	0.67	0.73	0.81
angier_noun	520.99	9.73	0	288	0	282.90	0.00	0.73	NaN

```r
action_thri <- keyness_table(action_dfm, thriller_dfm)
kableExtra::kbl(head(action_thri), caption = "Tokens with the highest keyness values in the action movi
  kableExtra::kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::kable_classic()
```

Table 22: Tokens with the highest keyness values in the action movie when compared to the thriller movie.

Token	LL	LR	PV	AF_Tar	AF_Ref	Per_10.5_Tar	Per_10.5_Ref	DP_Tar	DP_Ref
protagonist_noun	771.92	9.98	0	744	0	496.92	0.00	0.79	NaN
wayne_noun	684.76	9.81	0	660	0	440.82	0.00	0.42	NaN
wayne_pron	536.40	9.46	0	517	0	345.31	0.00	0.44	NaN
-_punct	526.43	2.31	0	1174	161	784.13	158.15	0.79	0.68
gordon_noun	498.01	9.35	0	480	0	320.60	0.00	0.41	NaN
batman_noun	415.01	9.09	0	400	0	267.16	0.00	0.41	NaN

Principal Component Analysis

```r
ds_counts <- script_tokens %>%
  tokens_lookup(dictionary = quanteda.extras::ds_dict, levels = 1, valuetype = "fixed") %>%
  dfm() %>%
  convert(to = "data.frame")

tot_counts <- quanteda::ntoken(script_tokens) %>%
  data.frame(tot_counts = .) %>%
  tibble::rownames_to_column("doc_id") %>%
  dplyr::as_tibble()

ds_counts <- dplyr::full_join(ds_counts, tot_counts, by = "doc_id")

ds_counts <- ds_counts %>%
  dplyr::mutate_if(is.numeric, list(~./tot_counts), na.rm = TRUE) %>%
  dplyr::mutate_if(is.numeric, list(~.*100), na.rm = TRUE) %>%
  dplyr::select(-tot_counts)
```

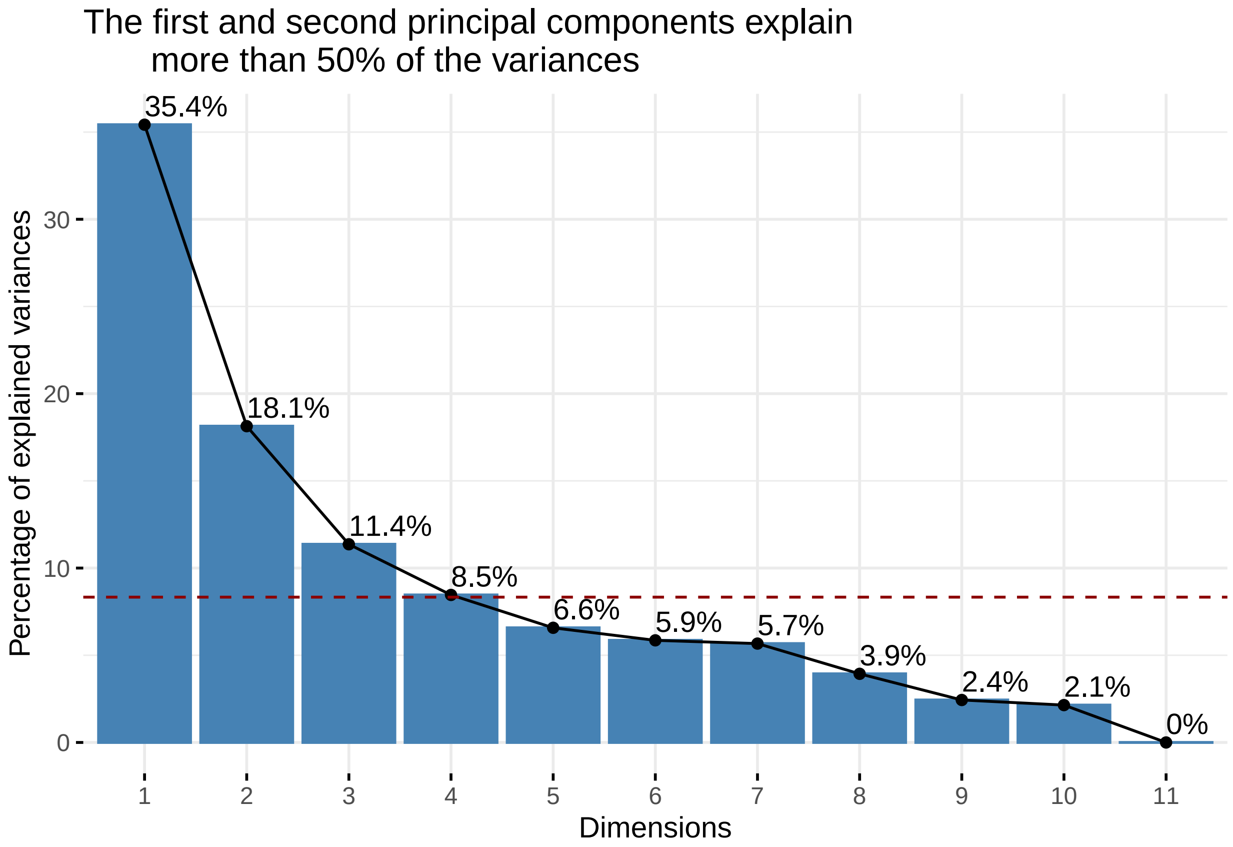
```
pca <- prcomp(ds_counts[-1], center = TRUE, scale. = TRUE)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6    PC7
## Standard deviation     3.6203 2.5905 2.0505 1.76915 1.55973 1.47203 1.4484
## Proportion of Variance 0.3542 0.1814 0.1136 0.08459 0.06575 0.05856 0.0567
## Cumulative Proportion  0.3542 0.5356 0.6492 0.73383 0.79958 0.85815 0.9149
##                            PC8     PC9    PC10      PC11      PC12
## Standard deviation     1.20682 0.94931 0.89055 3.479e-15 2.968e-16
## Proportion of Variance 0.03936 0.02436 0.02143 0.000e+00 0.000e+00
## Cumulative Proportion  0.95421 0.97857 1.00000 1.000e+00 1.000e+00
```

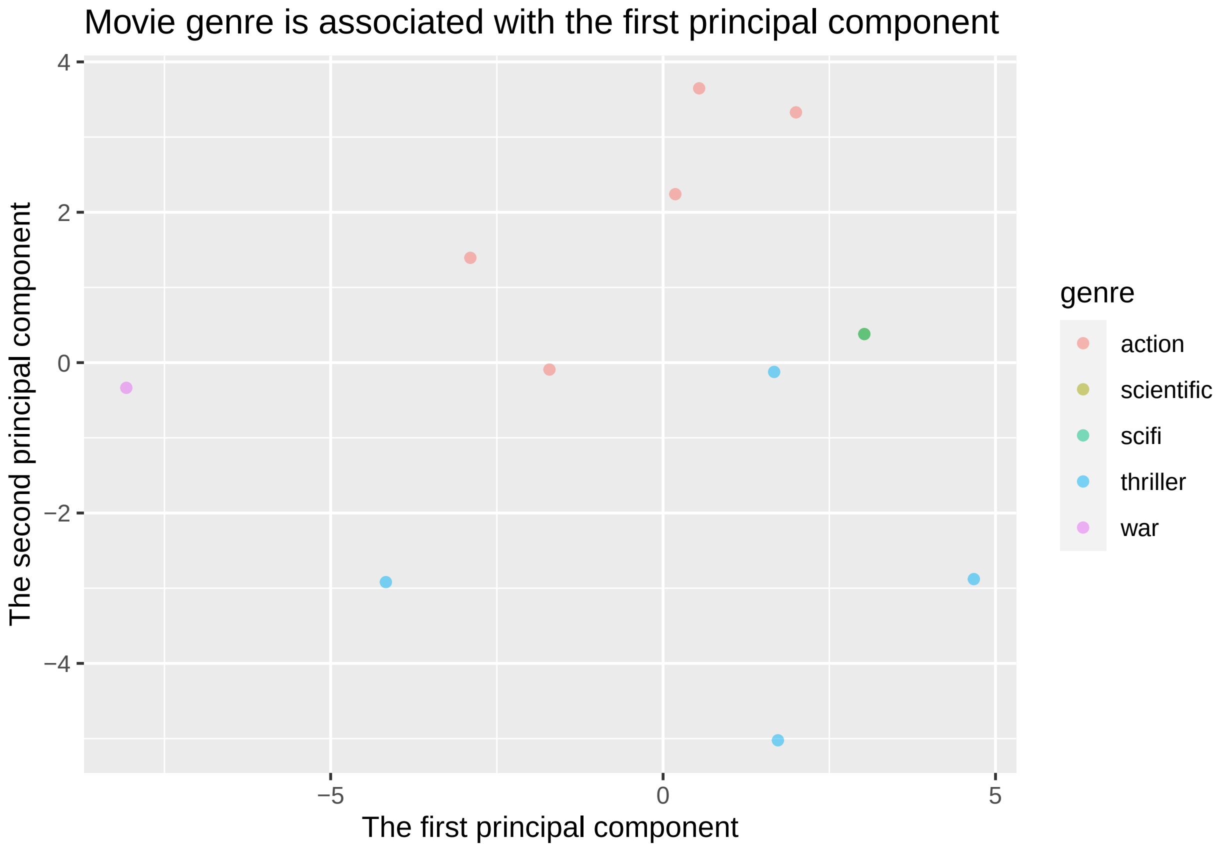
```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_eig(pca, addlabels = TRUE, ncp = 11) +
  geom_hline(yintercept = 100 * (1 / ncol(pca$x)),
             linetype = "dashed", color = "darkred") +
  labs(title = "The first and second principal components explain
       more than 50% of the variances")
```

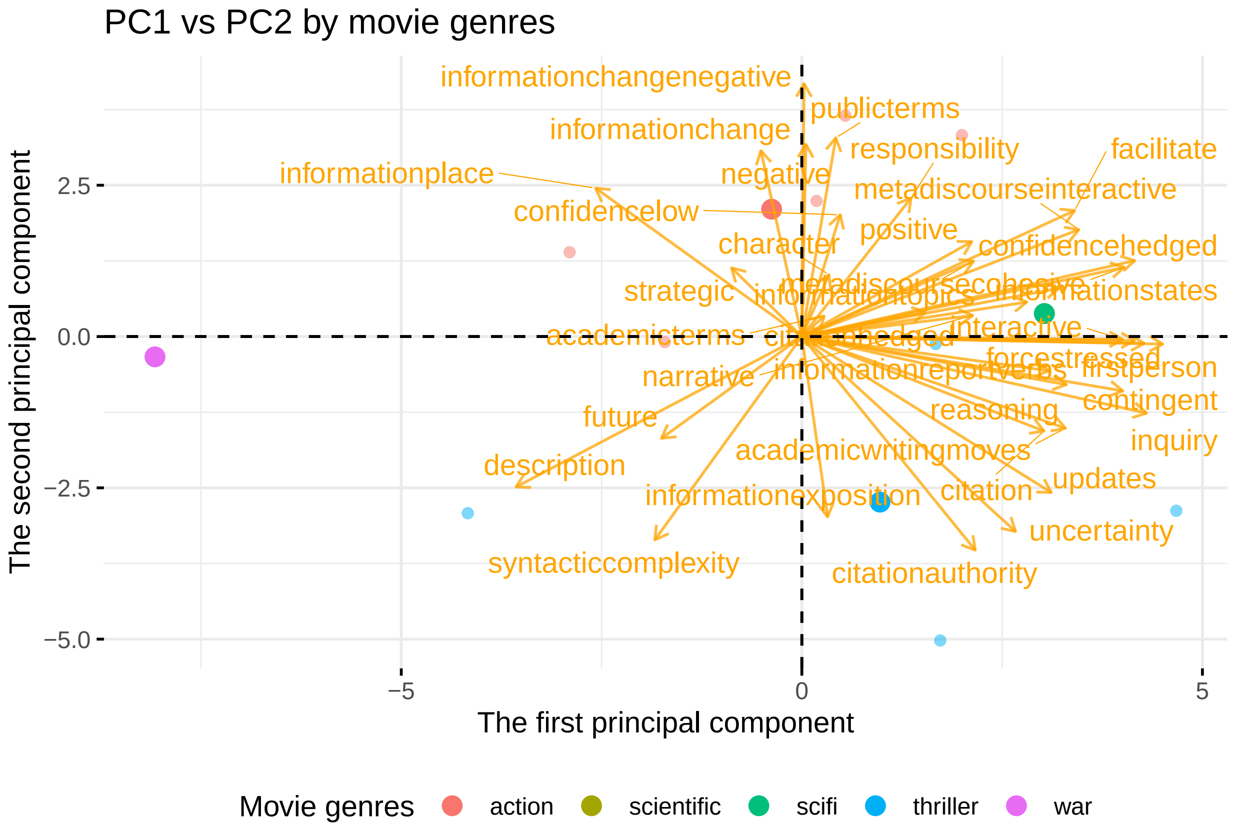


```
pc_matrix <- pca$x

quant <- ds_counts %>%
  mutate(pc1 =  pc_matrix[,1],
         pc2 =  pc_matrix[,2],
         doc_id = ds_counts$doc_id,
         genre = movie_genre,
         release_year = release_year)

quant %>%
  ggplot(aes(x = pc1, y = pc2, color = genre)) +
  geom_point(alpha = 0.5) +
  labs(x = "The first principal component",
        y = "The second principal component",
title = "Movie genre is associated with the first principal component")
```



```
fviz_pca_biplot(pca,
                pointshape = 19,
                # geom.ind = "points",
                label =  "var",
                # Plot PC1 and PC2
                axes = c(1, 2),
                # Change the alpha for the observations -
                # which is represented by ind
                alpha.ind = 0.5,
                # Modify the alpha for the variables (var):
                alpha.var = 0.75,
                repel = TRUE,
                # Set the color of the points to decades variable:
                col.ind = (quant$genre),
                # Modify the color of the variables
                col.var = "orange") +
  labs(title = "PC1 vs PC2 by movie genres",
       x = "The first principal component",
       y = "The second principal component",
       color = "Movie genres") +
  theme(legend.position = "bottom")
```

```
## Warning: ggrepel: 2 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
rotation_df <- as.data.frame(pca$rotation)
action_feature_df <- rotation_df[rotation_df$PC1 > 0 & rotation_df$PC2 > 0,] %>%
  mutate(rat_pc1_pc2 =
           rotation_df[rotation_df$PC1 > 0 & rotation_df$PC2 > 0,]$PC1/
           rotation_df[rotation_df$PC1 > 0 & rotation_df$PC2 > 0,]$PC2) %>%
  subset(select = c(PC1, PC2, rat_pc1_pc2))
```

```
action_feature_neg_df <- rotation_df[rotation_df$PC1 < 0 & rotation_df$PC2 < 0,] %>%
  mutate(rat_pc1_pc2 =
           rotation_df[rotation_df$PC1 < 0 & rotation_df$PC2 < 0,]$PC1/
           rotation_df[rotation_df$PC1 < 0 & rotation_df$PC2 < 0,]$PC2) %>%
  subset(select = c(PC1, PC2, rat_pc1_pc2))
```