FEAA
FACULTATEA DE ECONOMIE
ȘI ADMINISTRAREA AFACERILOR
UNIVERSITATEA "AL. I. CUZA" IAȘI

Analiza text a serialului de dramă fantastică „Game Of Thrones”

Student: Dancă Alexandra-Simona

310440105001SM211018

DM21

IAȘI 2022

FEAA
FACULTATEA DE ECONOMIE
ȘI ADMINISTRAREA AFACERILOR
UNIVERSITATEA "AL. I. CUZA" IAȘI

CUPRINS

1. Introducere

Game of Thrones este una dintre cele mai vizionate emisiuni de televiziune din istorie. Bazat pe seria de cărți intitulată „A Song of Ice and Fire" de George R.R. Martin, este plasată în țara fictivă Westeros. Această lume conține magie, săbii, sânge, dragoni și câteva scene explicite. Adaptarea pentru televiziune își ia numele de la titlul primei cărți din seria lui Martin: *A Game of Thrones*.

Pentru această lucrare, doresc să folosesc metodele de extragere a textului pe care le-am învățat pe parcursul semestrului pentru a analiza caracteristicile unice ale dialogul personajelor din serialul HBO *Game of Thrones*. Mai întâi, examinăm câteva statistici descriptive despre baza de date. Aceasta este împărțită pe episoade și sezoane, astfel se pot face analize atât pe baza citatelor, cât și pe sezoane. De asemenea, o importanță pentru acest proiect o poate aduce rafinarea celor mai importante cuvinte din dialogul personajelor, analiza sentimentelor pe sezoane, clasificarea cuvintelor în pozitive sau negative și evidențierea personajelor negative și pozitive pe baza sentimentelor rezultate. În cele din urmă, efectuăm modelarea subiectelor folosind Latent Dirichlet Allocation (LDA).

2. Prezentarea bazei de date

Primul pas din analiză a fost găsirea online a unei baze de date cu dialogurile pe care personajele le recită în serial. Baza de date utilizată în această lucrare este obținută de pe https://www.kaggle.com/datasets/albenft/game-of-thrones-script-all-seasons?resource=download. Aceasta conține următoarele coloane:

- **Release Date:** datele originale ale episodului;
- **Season:** numărul sezonului;
- **Episode:** numărul episodului;
- **Episode Title:** titlul fiecărui episod;
- **Name:** numele personajului din Game of Thrones;
- **Sentence**: propoziție rostită de personaj.

Pentru analizele următoare, o serie de pachete specifice R sunt necesare. De asemenea, importăm baza de date sub numele *Game_of_Thrones* și vizualizăm structura generală a acesteia.

```
### incarcam pachetele pentru text
library(readr)
library(dplyr)
library(tidyr)
library(ggplot2)
library(broom)
library(textdata)
library(stringr)
library(tidytext)
library(wordcloud)
library(reshape2)
library(scales)
library(igraph)
library(ggraph)
library(topicmodels)
library(widyr)

### importam baza de date
Game_of_Thrones<- read_csv("D:/OneDrive/Desktop/Master/An 2/PSDT/PROIECT/Game_of_Thrones_Script.csv")
head(Game_of_Thrones)
```

Figura 1. Input code

```
> head(Game_of_Thrones)
# A tibble: 6 x 6
  `Release Date` Season   Episode   `Episode Title`  Name         Sentence
  <date>         <chr>    <chr>     <chr>            <chr>        <chr>
1 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce What do you expect? They're savages. One lot~
2 2011-04-17     Season 1 Episode 1 Winter is Coming will        I've never seen wildlings do a thing like th~
3 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce How close did you get?
4 2011-04-17     Season 1 Episode 1 Winter is Coming will        Close as any man would.
5 2011-04-17     Season 1 Episode 1 Winter is Coming gared       We should head back to the wall.
6 2011-04-17     Season 1 Episode 1 Winter is Coming royce       Do the dead frighten you?
```

Figura 2. Baza de date

Avem 23911 de observații și 6 coloane. Coloana *Release Date* este alcătuită din observații de tip data, iar celelalte sunt de tip caracter.

Baza de date include nu numai informații despre linii de dialog, numele personajului și numele episodului, ci și despre numărul episodului și sezonul. Așadar, avea posibilitatea unor rezultate mai ample și mai interesante pentru toate cele 8 sezoane din serialul *Game of Thrones*.

3. Formatarea datelor text

Înainte să trecem la sistematizarea cuvintelor, doresc să vizualizez care au fost top 10 personaje care au avut cele mai multe linii de dialog pe tot parcursul serialului, de la sezonul 1 la sezonul 8.

Mai întâi am numărat frecvența caracterelor, le-am aranjat în ordine descrescătoare și le-am scos pe primele 10 personaje. Apoi am folosit ggplot pentru a reprezenta grafic rezultatul obținut.

```
Game_of_Thrones %>%
  count(Name) %>%
  arrange(desc(n)) %>%
  slice(1:10) %>%
  ggplot(aes(y=reorder(Name, n), x=n)) +
  geom_bar(stat="identity", aes(fill=n), show.legend=FALSE) +
  geom_label(aes(label=n)) +
  scale_fill_gradient(low="dodgerblue", high="dodgerblue4") +
  labs(x="Linii de dialog", y="Personaj",
       title="Linii de dialog per personaj") +
  theme_bw()
```

Figura 3. Input code

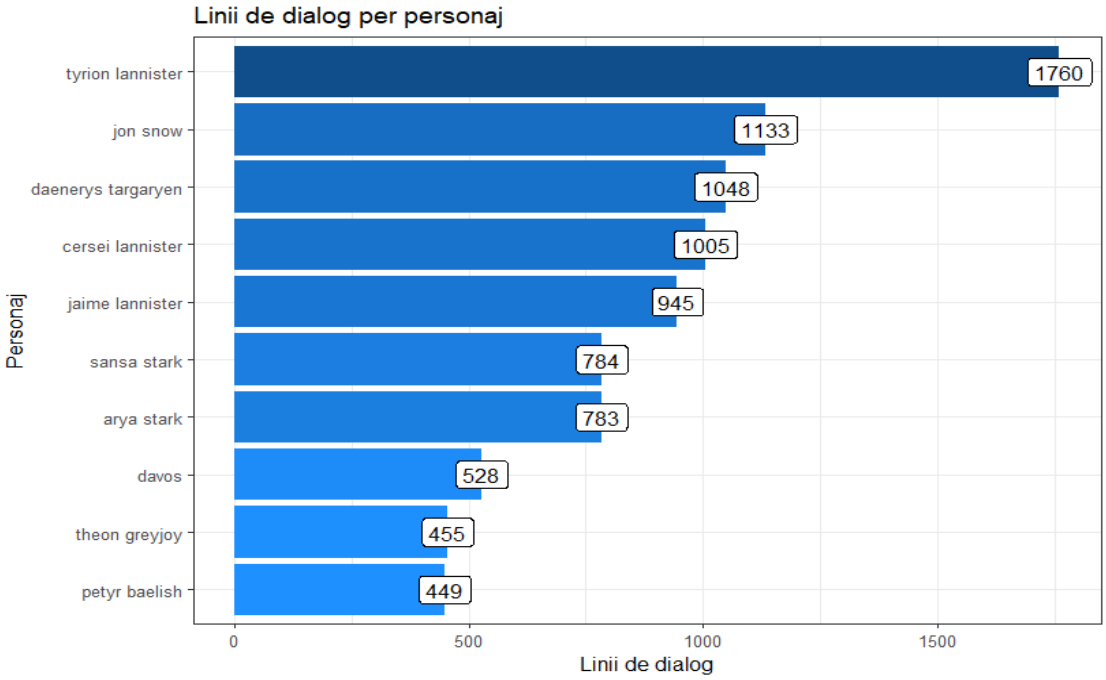

Figura 4. Număr de linii de dialog per personaj

Se poate observa că personajul cu cele mai multe linii de dialog este Tyrion Lannister cu 1760 de linii. Acesta este urmat de Jon Snow cu 1133 și Daenerys Targaryen cu 1048. Pe locul 10 se afla Petyr Baelish cu 449 de linii de dialog.

Mai departe urmează sistematizarea textului care presupune extragerea cuvintelor din liniile de dialog și gruparea fiecăruia pe fiecare rând.

```
### TOKENIZARE, grupat pe sezoane
GOT_tidy <- Game_of_Thrones %>%
  group_by(Season) %>%
  mutate(linenumber = row_number())%>%
  ungroup() %>%
  unnest_tokens(word, Sentence)
GOT_tidy
```

Figura 5. Input code

```
GOT_tidy              287775 obs. of 7 variables
```

Figura 6. DataFrame GOT_tidy înainte de eliminarea cuvintelor comune

De asemenea, se înlătură cuvintele comune care nu sunt de folos analizei.

```
### eliminarea cuvintelor comune (ex: the, of, etc.)
data(stop_words)
GOT_tidy <- GOT_tidy%>%
  anti_join(stop_words)
```

Figura 7. Input code

```
GOT_tidy              88595 obs. of 7 variables
```

Figura 8. DataFrame GOT_tidy dupa eliminarea cuvintelor comune

Înainte de eliminarea cuvintelor comune din noua bază de date creata GOT_tidy, erau 287775 de observații, adică tot atâte cuvinte pentru fiecare rând. După eliminarea acestora, au mai ramas 88595 de cuvinte per rând.

În continuare doresc să afișez o listă a cuvintelor des folosite în serial.

```
GOT_tidy%>%
  count(word, sort = TRUE)
```

Figura 9. Input code

```
# A tibble: 9,216 x 2
   word        n
   <chr>   <int>
 1 lord     1112
 2 king      812
 3 father    668
 4 grace     531
 5 time      523
 6 lady      493
 7 queen     446
 8 north     419
 9 people    418
10 brother   398
```

Figura 10. Top 10 cele mai folosite cuvinte

Cuvântul care a fost cel mai folosit este ”lord”. Acest cuvânt a fost folosit de 1112 ori și este urmat de ”king” cu 812. Frecvența acestor cuvinte precum și a altora cum sunt ”lady” și ”queen” se datorează faptului că personajele serialului provin din familii regale și nobile sau au interacțiuni cu aceștia.

În graficul care urmează, doresc să evidențiez cuvintele care apar mai mai mult de 500 ori.

```
GOT_tidy%>%
  count(word, sort = TRUE)%>%
  filter(n>500)%>%
  mutate(word = reorder(word, n))%>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
```

Figura 11. Input code

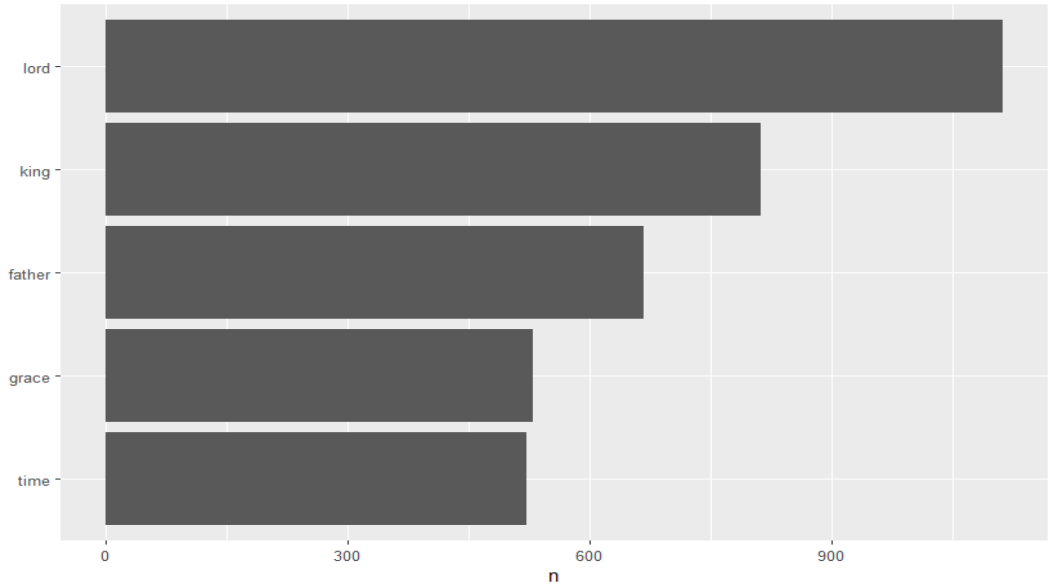

Figura 12. Cuvintele care apar de mai mult de 500 ori în serial

Cuvintele care apar de mai mult de 500 ori sunt: lord, king, father, grace și time.

```
GOT_tidy %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 50,rot.per=0.25, colors = brewer.pal(8,"Dark2"), size = 1))
```

Figura 13. Input code

Figura 14. Wordcloud

Am creat norul de cuvinte folosind funcția wordcloud() și am inclus 50 de cuvinte care se aflau în top-ul celor mai folosite cuvinte. Din Figura 14 se poate observa că mărimea cuvintelor se micșorează cu cât frecvența acestora este mai slabă .

Mai departe calculam frecvențele cuvintelor care apar atât în episodul 1 din sezonul 1 și din sezonul 2 și sa le comparăm cu același episod din sezonul 3. Rezultatele sunt prezentate grafic în Figura 16.

```
GOT_frequency_episode1<- GOT_tidy%>%
  filter(Episode == 'Episode 1') %>%
  mutate(word = str_extract(word, "[a-z']+"))%>%
  count(Season, word)%>%
  group_by(Season)%>%
  mutate(proportion = n / sum(n))%>%
  select(-n)%>%
  spread(Season, proportion)%>%
  gather(Season, proportion, 'Season 1','Season 2')

### grafic pentru ce este mai sus
ggplot(GOT_frequency_episode1, aes(x = proportion, y = `Season 3`,
                                   color = abs(`Season 3`- proportion)))+
  geom_abline(color = "gray40", lty = 2)+
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3)+
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5)+
  scale_x_log10(labels = percent_format())+
  scale_y_log10(labels = percent_format())+
  scale_color_gradient(limits = c(0, 0.001),
                       low = "darkslategray4", high = "gray75")+
  facet_wrap(~Season, ncol = 2)+
  theme(legend.position = "none")+
  labs(y="Season 3", x = NULL)
```

Figura 15. Input code

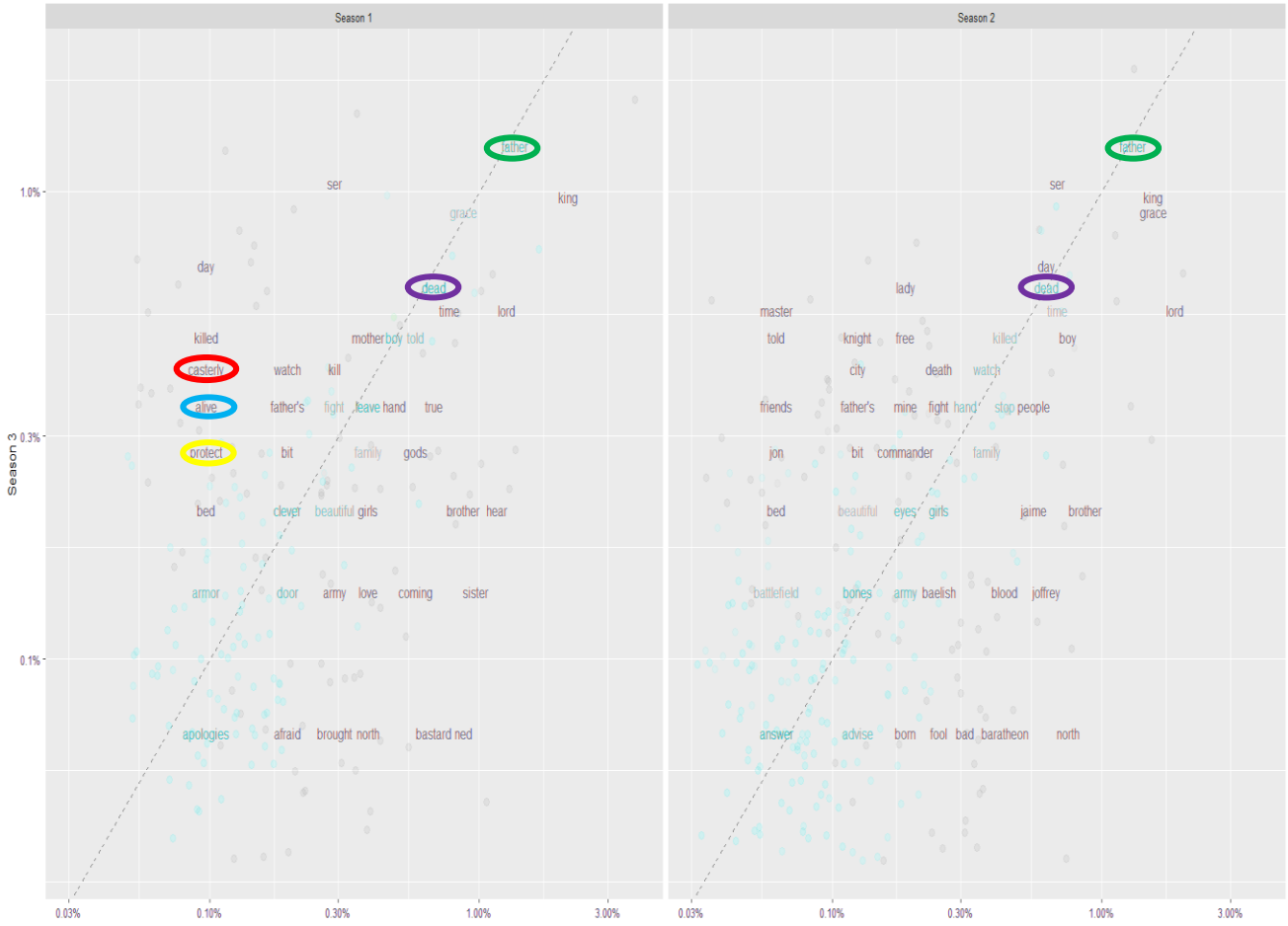

Figura 16. Frecvența cuvintelor

Proporțiile cuvintelor care se regăsesc atât în episodul 1 din sezonul 1 cât și în sezonul 2 și care apar cel mai des in ambele lucrări sunt cuvintele: ”father” și ”dead”. Găsim mai des în sezonul 1 decât ăn sezonul 2 cuvinte precum : ”alive”, ”casterly” și ”protected”.

Calculam corelația între aceste frecvențe ale cuvintelor.

```
cor.test(data = GOT_frequency_episode1[GOT_frequency_episode1$Season == "Season 1",],
         ~proportion+`Season 3`)
cor.test(data = GOT_frequency_episode1[GOT_frequency_episode1$Season == "Season 2",],
         ~proportion+`Season 3`)
```

Figura 17. Input code

```
> cor.test(data = GOT_frequency_episode1[GOT_frequency_episode1$Season == "Season 1",],
+          ~proportion+`Season 3`)

        Pearson's product-moment correlation

data:  proportion and Season 3
t = 7.9951, df = 188, p-value = 1.281e-13
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3892847 0.6028487
sample estimates:
      cor
0.5037237

> cor.test(data = GOT_frequency_episode1[GOT_frequency_episode1$Season == "Season 2",],
+          ~proportion+`Season 3`)

        Pearson's product-moment correlation

data:  proportion and Season 3
t = 13.165, df = 272, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5456863 0.6912452
sample estimates:
     cor
0.623846
```

Figura 18. Corelația între frecvența cuvintelor

Coeficientul de corelatie este mai puternic pentru episodul 1 din sezonul 2 cu sezonul 3 decât episodul 1 din sezonul 1 cu sezonul 3. Acest lucru determină o legătura mai puternică intre cele 2 sezoane.

4. Analiza sentimentelor

Analiza sentimentelor ne permite să evaluăm opinia sau emoția din text. Pachetul tidytext conține trei lexiconi de sentiment în setul de date sentimente: NRC, AFINN și Bing.

```
sentiments
get_sentiments("afinn")
get_sentiments("bing")
get_sentiments("nrc")
```

Figura 19. Input code

Lexicul NRC clasifică cuvintele în categorii pozitive, negative, furie, anticipare, dezgust, frică, bucurie, tristețe, surpriză și încredere.

```
> get_sentiments("nrc")
# A tibble: 13,872 x 2
   word        sentiment
   <chr>       <chr>
 1 abacus      trust
 2 abandon     fear
 3 abandon     negative
 4 abandon     sadness
 5 abandoned   anger
 6 abandoned   fear
 7 abandoned   negative
 8 abandoned   sadness
 9 abandonment anger
10 abandonment fear
```

Figura 20. Lexicul NRC

Lexicul Bing clasifică cuvintele în categorii pozitive și negative.

```
> get_sentiments("bing")
# A tibble: 6,786 x 2
   word        sentiment
   <chr>       <chr>
 1 2-faces     negative
 2 abnormal    negative
 3 abolish     negative
 4 abominable  negative
 5 abominably  negative
 6 abominate   negative
 7 abomination negative
 8 abort       negative
 9 aborted     negative
10 aborts      negative
# ... with 6,776 more rows
```

Figura 21. Lexicul Bing

Lexicul AFINN atribuie cuvintelor cu un scor care variază între -5 și 5, cu scoruri negative indicând sentimente negative și scoruri pozitive indicând sentimente pozitive.

```
# A tibble: 2,477 x 2
   word       value
   <chr>      <dbl>
 1 abandon       -2
 2 abandoned     -2
 3 abandons      -2
 4 abducted      -2
 5 abduction     -2
 6 abductions    -2
 7 abhor         -3
 8 abhorred      -3
 9 abhorrent     -3
10 abhors        -3
# ... with 2,467 more rows
```

Figura 22. Lexicul AFINN

În continuare am salvat o lista cu cei mai frecvenți termeni din fiecare emoție a lexicului NRC apoi am aranjat cuvintele în ordine descrescătoare și am afișat top 10 cuvinte pentru fiecare emoție al serialului Game of Thrones.

```
nrc <- get_sentiments("nrc")%>%
  mutate(lexicon = "nrc",
         words_in_lexicon = n_distinct(word))
### Apoi am aranjat cuvintele în ordine descrescătoare top 10 cuvinte.
GOT_tidy %>%
  inner_join(nrc, "word") %>%
  count(sentiment, word, sort=T) %>%
  group_by(sentiment) %>%
  arrange(desc(n)) %>%
  slice(1:10) %>%
  ggplot(aes(x=reorder(word, n), y=n)) +
  geom_col(aes(fill=sentiment), show.legend=FALSE) +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  facet_wrap(~sentiment, scales="free_y") +
  labs(y="Frecventa", x="Cuvinte",
       title="Cele mai frecvente cuvinte pentru fiecare sentiment NRC") +
  coord_flip() +
  theme_bw()
```

Figura 23. Input code

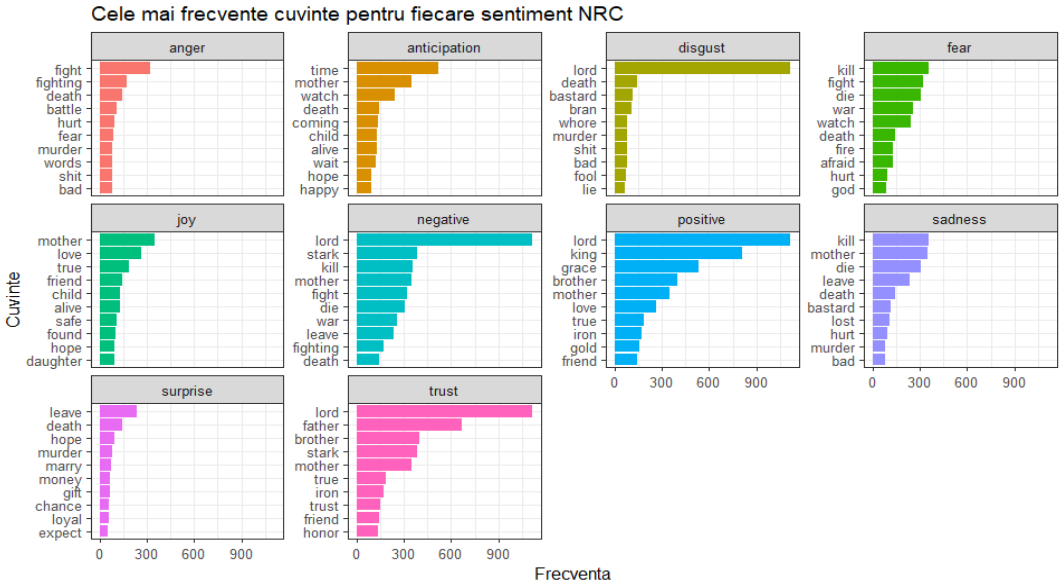

Figura 24. Frecvența cuvintelor per sentiment

Cuvintele din serial care au o frecvență mare și care fac parte dintr-o categorie de sentimente NRC sunt:

- Sentiment de **furie**: fight (luptă), death (moarte), battle (luptă);
- Sentiment de **anticipare**: time (timp), mother (mama), watch (ceas);
- Sentiment de **dezgust**: lord, death (moarte), bastard;
- Sentiment de **frică**: kill (ucide), fight (luptă), die (moarte);
- Sentiment de **bucurie**: mother (mama), love (iubire), true (adevăr);
- Sentiment **negativ**: lord, stark, kill (ucide);
- Sentiment **pozitiv**: lord, king, grace (grație);
- Sentiment de **tristețe**: kill (ucide), mother (mama), die (moarte);
- Sentiment de **surpriză**: leave (părăsește), dead (moarte), hope (speranță);
- Sentiment de **încredere**: lord, father (tata), brother (frate);

Mai departe salvăm într-un DataFrame cuvintele pentru sentimentul frică și afișăm cuvinele care exprimă acest sentiment pentru episoadele din sezonul 1.

```
nrc_fear <- get_sentiments("nrc") %>%
filter(sentiment == "fear")

GOT_tidy %>%
  filter(Season == "Season 1") %>%
  inner_join(nrc_fear) %>%
  count(word, sort = TRUE)
```

Figura 25. Input code

```
# A tibble: 221 x 2
   word       n
   <chr>  <int>
 1 kill      50
 2 watch     43
 3 die       34
 4 war       32
 5 hurt      26
 6 death     25
 7 fight     25
 8 dragon    24
 9 fear      18
10 mad       16
# ... with 211 more rows
```

Figura 26. Top 10 cuvinte care exprimă frică din sezonul 1

Cuvinte precum *kill*, *watch*, *die* , *war* , *hurt*, *death*, *fight*, *dragon*, *fear* și *mad* fac parte din cele mai întâlnite cuvinte care exprimă emoții de frică din sezonul 1.

Mai jos, am folosit lexiconul de sentmente BING pentru un alt tip de analiză pe sezoane. Folosind numărul de cuvinte negative și pozitive dintr-un sezon, am creat un raport care determină cât de puternice pozitiv/negativ sunt toate sezoanele în ordine crescătoare.

```
ratio_seasons <- GOT_tidy %>%
  inner_join(get_sentiments("bing")) %>%
  group_by(Season, sentiment) %>%
  summarize(score = n()) %>%
  spread(sentiment, score) %>%
  ungroup() %>%
  mutate(ratio = positive / (positive + negative),
         Season = reorder(Season, ratio))

ratio_seasons %>%
  ggplot(aes(x = Season, y = ratio)) +
  geom_point(color = "blue", size = 4) +
  coord_flip() +
  labs(title = "Top sezoane pozitive",
       x = "",
       caption = "ratio = positive to positive and negative words jointly") +
  theme_minimal() +
  theme(plot.title = element_text(size = 16, face = "bold"),
        panel.grid = element_line(linetype = "dashed", color = "darkgrey", size = .5))
```

Figura 27. Input code

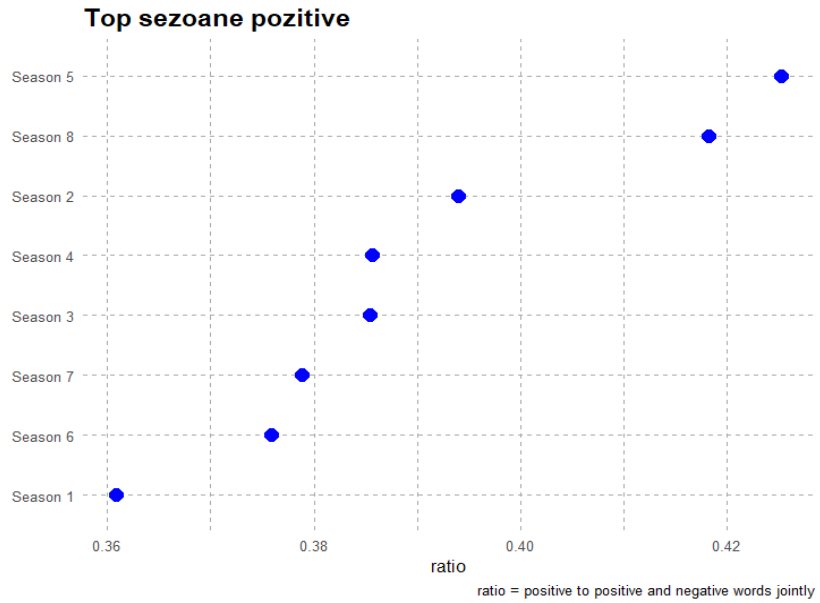

Figura 28. Top sezoane pozitive

Din figura 28 se poate observa faptul că sezonul 5 este sezonul cu sentimentele cele mai pozitive dintre toate. Acesta este urmat de sezonul 8, sezonul 2, sezonul 4, sezonul 3, sezonul 7, sezonul 6 și ultimul este sezonul 1.

Apoi am creat un grafic similar pentru cele mai negative episoade. Raportul aici a fost calculat prin scăderea raportului „pozitiv” din unitate.

```
ratio_episodes <- GOT_tidy %>%
  inner_join(get_sentiments("bing")) %>%
  group_by(`Episode Title`, sentiment) %>%
  summarize(score = n()) %>%
  spread(sentiment, score) %>%
  ungroup() %>%
  mutate(ratio = positive / (positive + negative),
         `Episode Title` = reorder(`Episode Title`, ratio))

ratio_episodes %>%
  mutate(ratio = 1 - ratio,
         `Episode Title` = reorder(`Episode Title`, ratio)) %>%
  top_n(20) %>%
  ggplot(aes(x = `Episode Title`, y = ratio)) +
  geom_point(color = "red", size = 4) +
  coord_flip() +
  labs(title = "Top 20 episoade negative",
       x = "",
       caption = "ratio = negative to positive and negative words jointly") +
  theme_minimal() +
  theme(plot.title = element_text(size = 16, face = "bold"),
        panel.grid = element_line(linetype = "dashed", color = "darkgrey", size = .5))
```

Figura 29. Input code

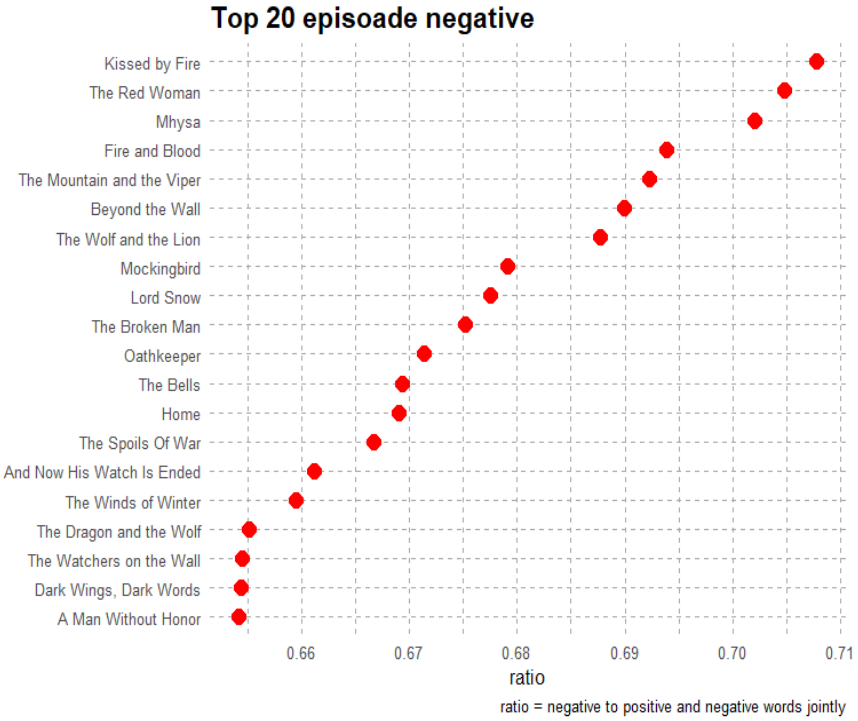

Figura 30. Top 20 episoade negative

”Kissed by Fire” este episodul cu cele mai multe sentimente negative dintre toate episoadele serialului *Game of Thrones*.

În urmatoarele figuri, analizam care sunt cele mai frecvente cuvinte care influențeaza un anumit sentiment.

```
bing_word_counts <- GOT_tidy %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
bing_word_counts
```

Figura 31. Input code

```
# A tibble: 1,638 x 3
   word    sentiment     n
   <chr>   <chr>     <int>
 1 grace   positive    531
 2 stark   negative    389
 3 dead    negative    381
 4 kill    negative    355
 5 die     negative    305
 6 love    positive    268
 7 killed  negative    250
 8 gold    positive    161
 9 trust   positive    151
10 free    positive    149
# ... with 1,628 more rows
```

Figura 32. Output

Cuvântul ”grace” influențează sentimental pozitiv de 531 ori. Următorul cuvânt care influențează sentimental pozitiv este ”love” care apare de 268 ori.

Cuvintele ”stark” și ”dead” influențează sentimental negativ de 389 ori, respectiv 381.

```
bing_word_counts %>%
  group_by(sentiment) %>%
  top_n(5) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to sentiment", x = NULL) +
  coord_flip()
```

Figura 33. Input code

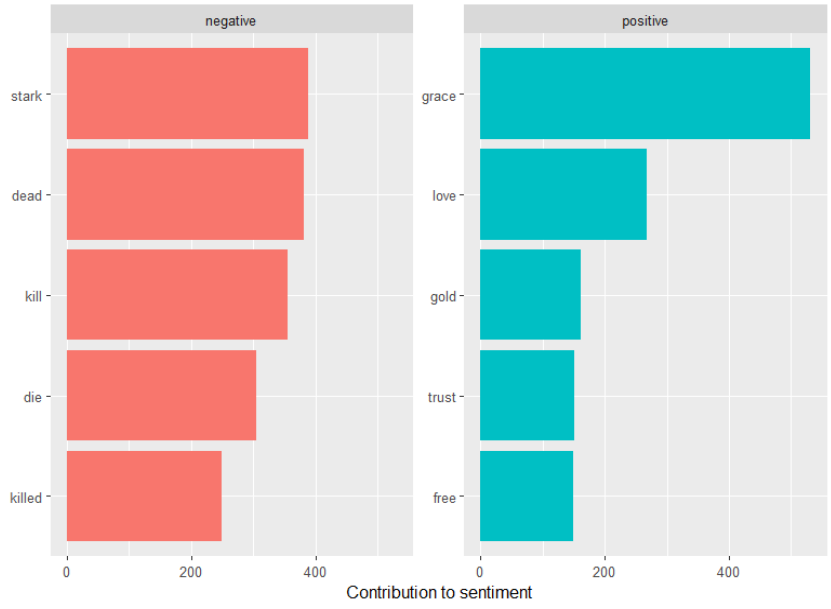

Figura 34. Output

Graficul afișează primele 5 cuvinte din sentimentul negativ și alte 5 cuvinte pentru sentimentul pozitiv. Se poate observa că doar cuvântul pozitiv ”grace” depășește pragul de 500. Restul cuvintelor au fost prezente de mai puțin de 400 ori.

```
GOT_tidy %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("red", "green"),
                   max.words = 100)
```

Figura 35. Input code

Figura 36. Wordcloud pe sentimente negative și pozitive

Wordcloud-ul este împărțit în două categorii de sentimente: roșu reprezintă sentimentul negativ, iar culoarea verde reprezinta sentimentul pozitiv.

Mai departe doresc să analizez sentimentele pe care 10 personaje le au în serial. Personajele au fost alese pe baza rolurilor pe care aceștia îl au în serial astfel s-au ales 10 personaje importante și care joacă un rol principal în serial.

```
GOT_tidy %>%
  filter(Name %in% c("tyrion lannister","jon snow","daenerys targaryen", "cersei lannister",
                     "jaime lannister","sansa stark","arya stark","davos",
                     "theon greyjoy","petyr baelish")) %>%
  inner_join(nrc, "word") %>%
  count(Name, sentiment, sort=TRUE) %>%
  ggplot(aes(x=sentiment, y=n)) +
  geom_col(aes(fill=sentiment), show.legend=FALSE) +
  facet_wrap(~Name, scales="free_x") +
  labs(x="Sentiment (NRC)", y="Frecventa",
       title="Analiza sentimentelor pentru 10 personaje din serial") +
  coord_flip() +
  theme_bw()
```

Figura 37. Input code

Cu ajutorul ggplot()-ului se pot analiza mai ușor sentimentele pe care un personaj din serial le poate avea.

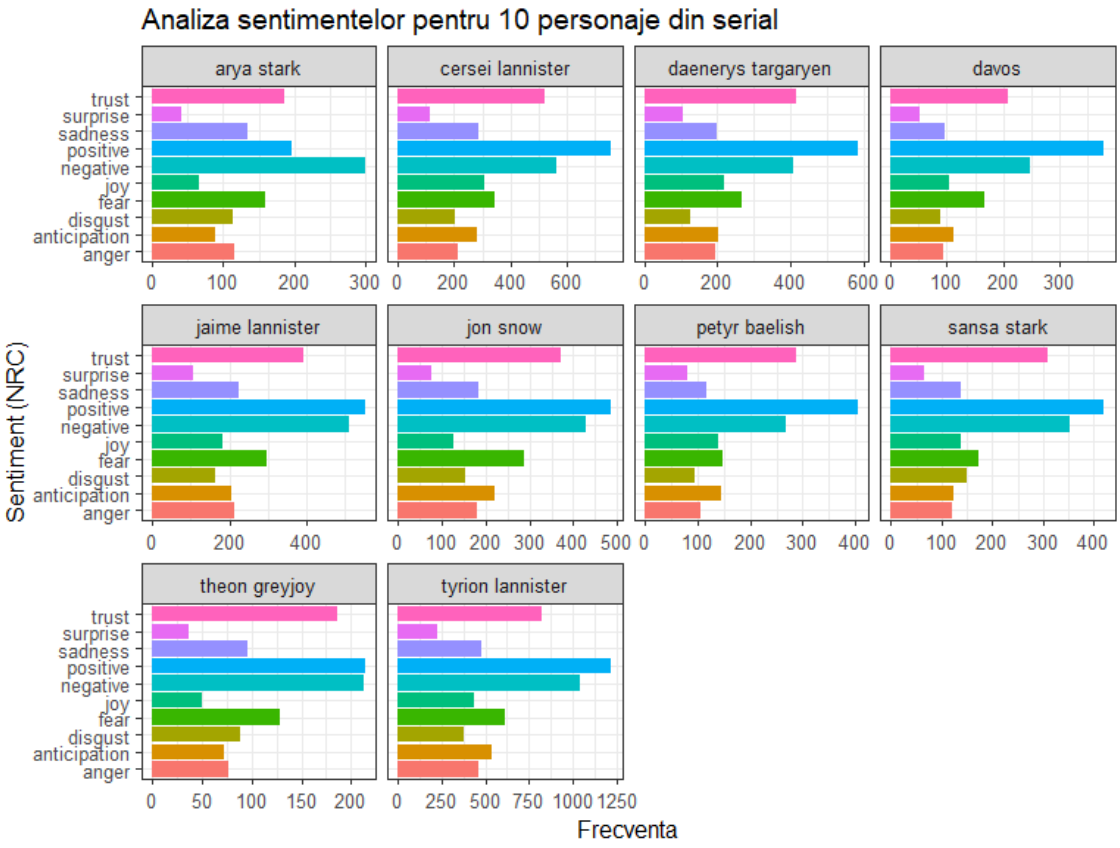

Figura 38. Analiza sentimentelor pentru 10 personaje din serial

Arya Stark este predominată de sentimentul de negativitate. De asemena, prezintă puține cuvinte care derivă din sentimentul de surprindere.

Cersei Lannister, Daenerys Targaryen, Davos, Jamie Lannister, Jon Snow, Petyr Baelish, Sansa Stark, Theon Greyjoy și Tyrion Lannister sunt predominați de sentimentul pozitiv. De asemena, aceștia prezintă puține cuvinte care derivă din sentimentul de surprindere.

Din aceste rezultate putem spune că cele mai importante personaje care joacă roluri principale în serialul *Game of Thrones* folosesc cuvinte care provoacă emoții pozitive, negative și de încredere.

5. Analiza frecvențelor

5.1. Frecvența relativă a cuvântului

În acest subcapitol analizăm numărul aparițiilor ale unui cuvânt raportat la numărul total de cuvinte din serialul *Game Of Thrones*.

Pentru o analiză mai bună și distinctă se vor face 2 analize: una fara cuvinte comune și cealalta cu cuvinte comune.

A. Împreuna cu cuvinte comune

Pentru început se v-a face sistematizarea textului fără a elimina cuvintele comune. În alt dataFrame se adună toate cuvintele pentru fiecare sezon. În cele din urmă analizăm frecvența cuvintelor dintre numărul de apariții raportat la totalul cuvintelor pe fiecare sezon.

```
### TOKENIZARE fara eliminarea cuvintelor cheie (stop_words)
GOT_words <- Game_of_Thrones%>%
  unnest_tokens(word, Sentence) %>%
  count(Season, word, sort = TRUE) %>%
  ungroup()

### df cu totalul cuvintelor pentru fiecare sezon
total_words <- GOT_words %>%
  group_by(Season) %>%
  summarize(total = sum(n))

### analizam frecventa cuvintelor
GOT_words <- left_join(GOT_words, total_words)
GOT_words
```

Figura 39. Input code

```
# A tibble: 28,472 x 4
   Season   word      n total
   <chr>    <chr> <int> <int>
 1 Season 2 the    1908 46833
 2 Season 2 you    1789 46833
 3 Season 5 the    1691 37413
 4 Season 3 you    1659 42125
 5 Season 6 the    1643 36392
 6 Season 4 you    1638 41543
 7 Season 3 the    1629 42125
 8 Season 1 the    1608 37669
 9 Season 4 the    1594 41543
10 Season 7 the    1586 32274
# ... with 28,462 more rows
```

Figura 40. Frecvența cuvintelor

"the" apare de 1908 in sezonul 2. Este cuvantul cu cea mai mare frecventa. Acesta este urmat de "you" care apare de 1789 in acelasi sezon.

Putem reprezenta grafic distribuția raportului dintre frecvență și totalul de cuvinte pe fiecare sezon. Aceasta este frecvența relativă a cuvântului.

```
### putem reprezenta grafic distributia
ggplot(GOT_words, aes(n/total, fill = Season)) +
  geom_histogram(show.legend = FALSE) +
  xlim(NA, 0.0009) +
  facet_wrap(~Season, ncol = 2, scales = "free_y")
```

Figura 41. Input code

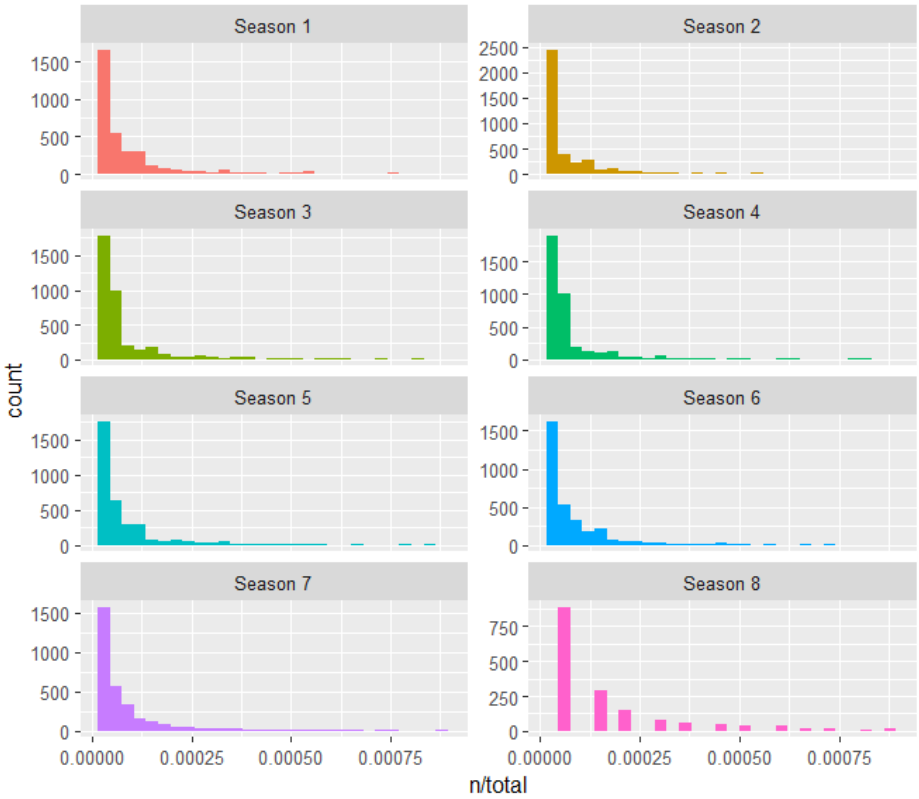

Figura 42. Frecvența relativă a cuvântului

Avem o distributie putern asimetrica la dreapta. Avem cateva cuvinte care apar de multe ori.

B. Făra cuvinte comune

Pentru început se v-a face sistematizarea textului și se elimina cuvintele comune. În alt dataFrame se adună toate cuvintele pentru fiecare sezon. În cele din urmă analizăm frecvența cuvintelor dintre numărul de apariții raportat la totalul cuvintelor pe fiecare sezon.

```
GOT_words_2 <- Game_of_Thrones%>%
  unnest_tokens(word, Sentence) %>%
  count(Season, word, sort = TRUE) %>%
  ungroup()

GOT_words_2<-GOT_words_2%>%
  anti_join(stop_words)

### df cu totalul cuvintelor pentru fiecare sezon
total_words <- GOT_words_2 %>%
  group_by(Season) %>%
  summarize(total = sum(n))

### analizam frecventa cuvintelor
GOT_words_2 <- left_join(GOT_words_2, total_words)
GOT_words_2
```

Figura 43. Input code

```
# A tibble: 24,505 x 4
   Season   word       n total
   <chr>    <chr>  <int> <int>
 1 Season 1 lord     239 11864
 2 Season 2 lord     205 14336
 3 Season 3 lord     177 12713
 4 Season 2 king     173 14336
 5 Season 1 king     159 11864
 6 Season 4 lord     154 12986
 7 Season 6 lord     121 10969
 8 Season 5 lord     118 11483
 9 Season 2 grace    115 14336
10 Season 2 father   113 14336
# ... with 24,495 more rows
```

Figura 44. Frecvența cuvintelor

"lord" apare de 239 în sezonul 1 din totalul de 11864 de cuvinte din acest sezon. Este cuvantul cu cea mai mare frecvență. Acesta este urmat de "lord" care apare de 205 în sezonul 2.

Putem reprezenta grafic distribuția raportului dintre frecvență și totalul de cuvinte pe fiecare sezon. Aceasta este frecvența relativă a cuvântului.

```
ggplot(GOT_words_2, aes(n/total, fill = Season)) +
  geom_histogram(show.legend = FALSE) +
  xlim(NA, 0.0009) +
  facet_wrap(~Season, ncol = 2, scales = "free_y")
```

Figura 45. Input code

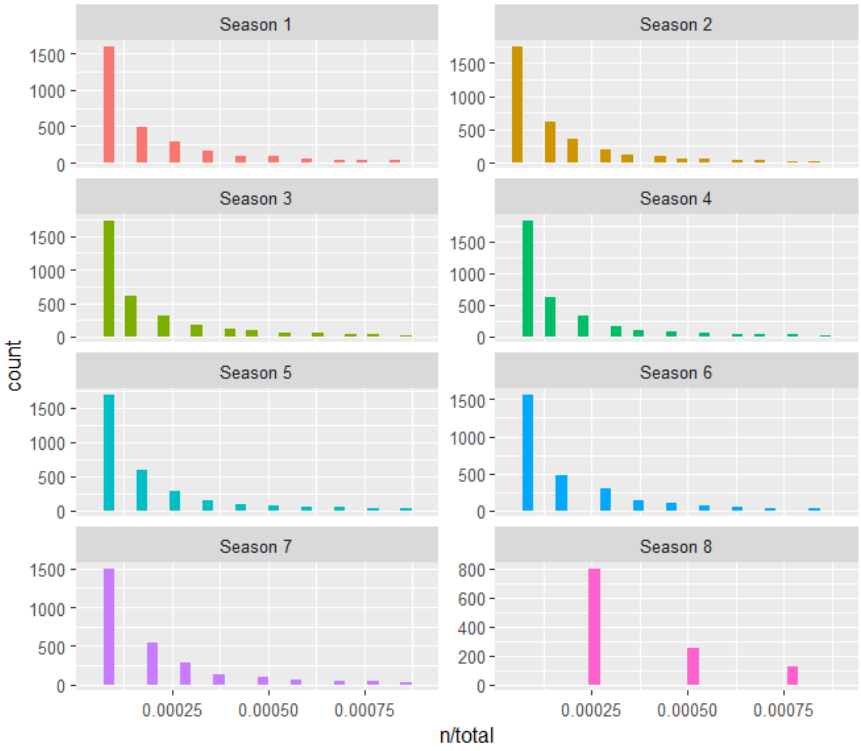

Figura 46. Frecvența relativă a cuvântului

5.2. Legea lui Zipf

Legea lui Zipf spune că frecvența cu care apare un cuvânt este invers proporțională cu rangul său.

```
freq_by_rank <- GOT_words_2 %>%
  group_by(Season) %>%
  mutate(rank = row_number(), `term frequency` = n/total)
freq_by_rank
```

Figura 47. Input code

```
# A tibble: 24,505 x 6
# Groups:   Season [8]
   Season   word       n total  rank `term frequency`
   <chr>    <chr>  <int> <int> <int>            <dbl>
 1 Season 1 lord     239 11864     1           0.0201
 2 Season 2 lord     205 14336     1           0.0143
 3 Season 3 lord     177 12713     1           0.0139
 4 Season 2 king     173 14336     2           0.0121
 5 Season 1 king     159 11864     2           0.0134
 6 Season 4 lord     154 12986     1           0.0119
 7 Season 6 lord     121 10969     1           0.0110
 8 Season 5 lord     118 11483     1           0.0103
 9 Season 2 grace    115 14336     3           0.00802
10 Season 2 father   113 14336     4           0.00788
# ... with 24,495 more rows
```

Figura 48. Rangul cuvintelor

Rangul de 0,0201 este raportul dintre 239/11864 (n/total) pentru cuvantul "lord" din sezonul 1.

Legea lui Zipf este adesea vizualizată prin reprezentarea rangului pe axa x și a frecvenței termenului pe axa y, pe scale logaritmice. Trasând astfel, o relație invers proporțională va avea o pantă constantă, negativă.

```
freq_by_rank %>%
  ggplot(aes(rank, `term frequency`, color = Season)) +
  geom_line(size = 1.1, alpha = 0.8, show.legend = TRUE) +
  scale_x_log10() +
  scale_y_log10()
```

Figura 49. Input code

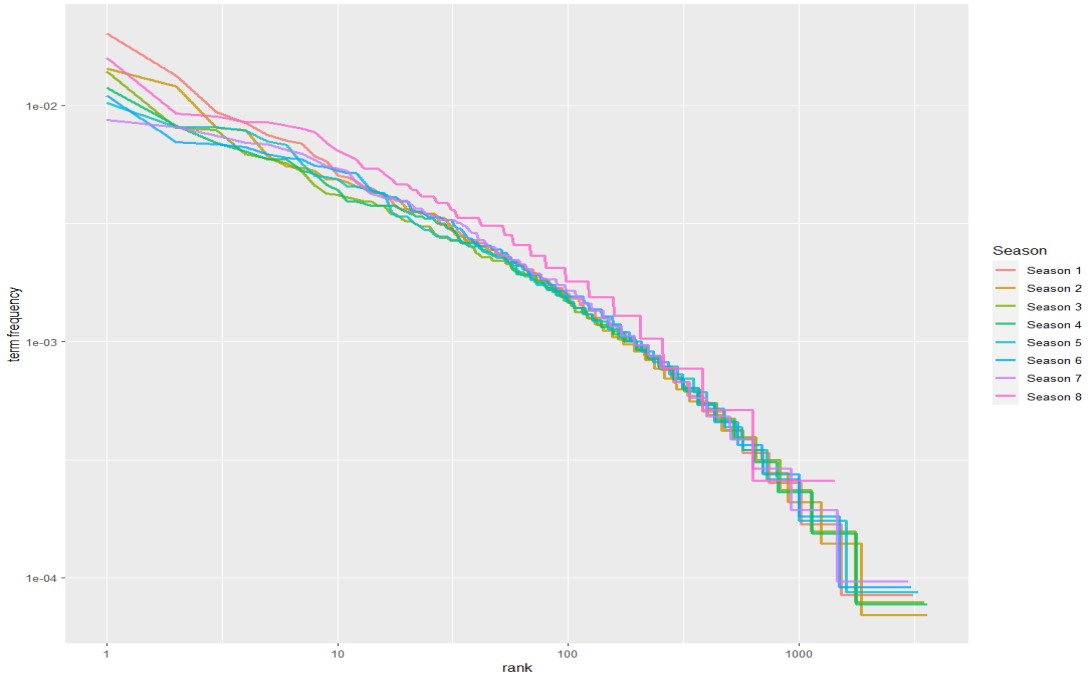

Figura 50. Output

Graficul ilustrează faptul că distribuția este similară în cele 8 sezoane.

6. Relații între cuvinte

În acest capitol analiza text se bazează pe relațiile dintre cuvinte.

6.1. Tokenizare prin Bi-gram

La fel cum am avut tokenizarea secvențială a cuvintelor pe fiecare rând, la fel ne putem folosi de funcția pentru a tokeniza în secvențe consecutive de cuvinte, numite N-grame . Văzând cât de des este urmat cuvântul X de cuvântul Y, putem construi un model al relațiilor dintre ele.

```
GOT_bigrams <- Game_of_Thrones%>%
  unnest_tokens(bigram, Sentence, token = "ngrams", n = 2)
GOT_bigrams
```

Figura 51. Input code

```
# A tibble: 265,114 x 6
   `Release Date` Season   Episode   `Episode Title`  Name         bigram
   <date>         <chr>    <chr>     <chr>            <chr>        <chr>
 1 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce what do
 2 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce do you
 3 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce you expect
 4 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce expect they're
 5 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce they're savages
 6 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce savages one
 7 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce one lot
 8 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce lot steals
 9 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce steals a
10 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce a goat
# ... with 265,104 more rows
```

Figura 52. Output

6.1.1. Numărarea și filtrarea Bi-gram

După cum se observă, multe dintre cele mai comune bigrame sunt perechi de cuvinte comune. Așadar împărțim o coloană în două delimitate de un separator. Acest lucru ne permite să-l separăm în două coloane, „word1” și „word2”, moment în care putem elimina cazurile în care oricare dintre ele este un cuvânt comun. Funcția *unite()* ne permite să recombinăm coloanele într-una singură.

```
### separam cele 2 cuvinte
bigrams_separat <- GOT_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

### Scoatem cuvintele cheie
bigrams_filtru <- bigrams_separat %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

### Numaram noile bigram
bigram_nr <- bigrams_filtru %>%
  count(word1, word2, sort = TRUE)

### Unim bigramele pentru a forma cuvinte
bigrams_unit <- bigrams_filtru %>%
  unite(bigram, word1, word2, sep = " ")
bigrams_unit
```

Figura 53. Input code

```
# A tibble: 19,840 x 6
   `Release Date` Season   Episode   `Episode Title`  Name         bigram
   <date>         <chr>    <chr>     <chr>            <chr>        <chr>
 1 2011-04-17     Season 1 Episode 1 Winter is Coming waymar royce lot steals
 2 2011-04-17     Season 1 Episode 1 Winter is Coming royce        dead frighten
 3 2011-04-17     Season 1 Episode 1 Winter is Coming royce        south run
 4 2011-04-17     Season 1 Episode 1 Winter is Coming royce        moved camp
 5 2011-04-17     Season 1 Episode 1 Winter is Coming gared        NA NA
 6 2011-04-17     Season 1 Episode 1 Winter is Coming jon snow     father's watching
 7 2011-04-17     Season 1 Episode 1 Winter is Coming eddard stark practicing bran
 8 2011-04-17     Season 1 Episode 1 Winter is Coming robb stark   bow arm
 9 2011-04-17     Season 1 Episode 1 Winter is Coming jonrobb      quick bran
10 2011-04-17     Season 1 Episode 1 Winter is Coming jonrobb      bran faster
# ... with 19,830 more rows
```

Figura 54. Output

6.1.2. Analiza exploratorie a Bi-gram

În acest subcapitol, ne putem uita la tf-idf al bigramelor din serialului *Game Of Thrones*. Aceste valori tf-idf pot fi vizualizate în fiecare sezon.

```
bigram_tf_idf <- bigrams_unit %>%
  count(Season, bigram) %>%
  bind_tf_idf(bigram, Season, n) %>%
  arrange(desc(tf_idf))
bigram_tf_idf
```

Figura 55. Input code

```
# A tibble: 14,716 x 6
   Season   bigram           n      tf   idf  tf_idf
   <chr>    <chr>        <int>   <dbl> <dbl>   <dbl>
 1 Season 6 door hold       18 0.00741 2.08  0.0154
 2 Season 8 mama mama        4 0.00510 2.08  0.0106
 3 Season 4 prince oberyn   15 0.00487 2.08  0.0101
 4 Season 8 night king       8 0.0102  0.981 0.0100
 5 Season 6 dosh khaleen    11 0.00453 2.08  0.00941
 6 Season 8 sansa told       3 0.00383 2.08  0.00796
 7 Season 8 ser brienne      3 0.00383 2.08  0.00796
 8 Season 1 ser hugh        10 0.00376 2.08  0.00783
 9 Season 8 iron fleet       6 0.00765 0.981 0.00751
10 Season 5 oysters clams    9 0.00352 2.08  0.00731
# ... with 14,706 more rows
```

Figura 56. Output

Se observă faptul ca cele mai multe bigrame sunt nume sau adresări ale personajelor din serial.

6.1.3. Utilizarea Bi-gram în analiza sentimentelor

Efectuând o analiză a sentimentelor pe datele bigramelor, putem examina cât de des cuvintele asociate sentimentelor sunt precedate de "not", "no", "never", "without" sau alte cuvinte de negare.

```
negation_words <- c("not", "no", "never", "without")

### ne ofera scorul sentimentelor numerice, care indica directia sentimentului
AFINN <- get_sentiments("afinn")

not_words <- bigrams_separat %>%
  filter(word1 %in% negation_words) %>%
  inner_join(AFINN, by = c(word2 = "word")) %>%
  count(word2, value, sort = TRUE)
not_words

### grafic
not_words %>%
  mutate(contribution = n * value) %>%
  arrange(desc(abs(contribution))) %>%
  head(20) %>% mutate(word2 = reorder(word2, contribution)) %>%
  ggplot(aes(word2, n * value, fill = n * value > 0)) +
  geom_col(show.legend = FALSE) +
  xlab("Words preceded by negation") +
  ylab("Sentiment value * number of occurrences") +
  coord_flip()
```

Figura 57. Input code

```
# A tibble: 208 x 3
   word2  value     n
   <chr>  <dbl> <int>
 1 no        -1   111
 2 matter     1    30
 3 afraid    -2    16
 4 die       -3    16
 5 like       2    16
 6 doubt     -1    14
 7 good       3    14
 8 please     1    14
 9 forget    -1    13
10 better     2    12
# ... with 198 more rows
```

Figura 58. Output not_words

Cel mai frecvent cuvânt asociat sentimentelor care urmează de o negație a fost ”no”, care ar avea în mod normal un scor negativ de -1. Următorul cuvânt clasat pe poziția doi, se află cuvântul ”matter”, care ar avea în mod normal un scor pozitiv de 1. În acest caz trebuie să analizăm care cuvinte au contribuit cel mai mult în direcția opusă sensului cuvântului.

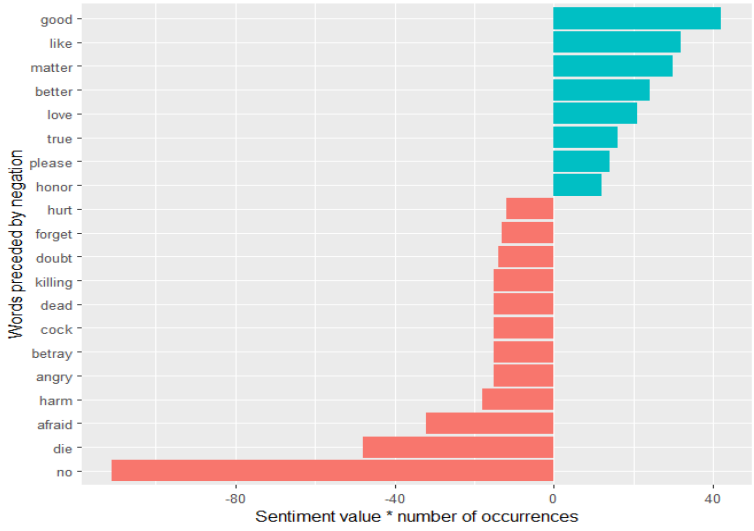

Figura 59. Cuvintele precedate de „nu” care au avut cea mai mare contribuție la valorile sentimentului, fie într-o direcție pozitivă, fie negativă

Bigramele „not good” și „not like” au fost în mare parte cele mai mari cauze de identificare greșită, făcând textul să pară mult mai pozitiv decât este. Dar putem vedea expresii precum „nu răni” și „nu uita” uneori sugerează că textul este mai negativ decât este defapt.

6.1.4. Rețele cu Bi-gram

În acest subcapitol dorim să vizualizăm toate relațiile dintre cuvinte simultan cu ajutorul unei rețele care leagă cuvintele între ele.

```
bigram_graph <- bigram_nr %>%
  filter(n > 20) %>%
  graph_from_data_frame()
bigram_graph
### grafic
set.seed(2017)
ggraph(bigram_graph, layout = "fr") +
  geom_edge_link() +
  geom_node_point() +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1)
```

Figura 60. Input code

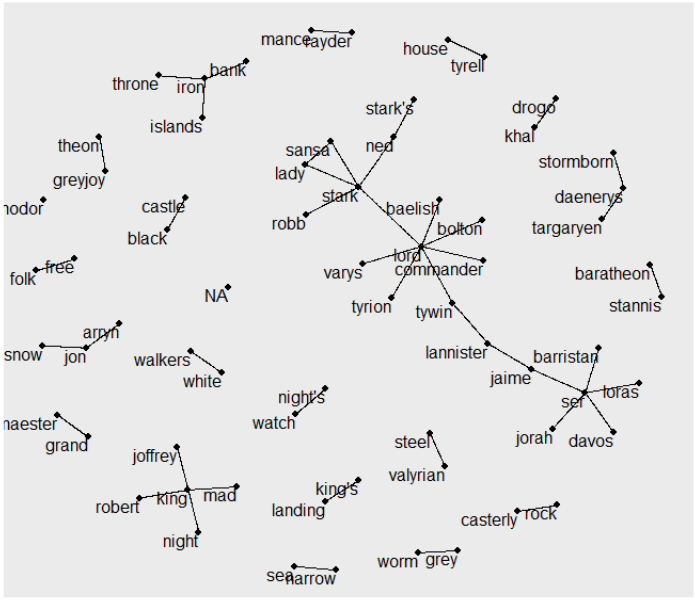

Figura 61. Bigramele obișnuite în serialul Game Of Thrones, arătând cele care au apărut de mai mult de 20 de ori și în care niciun cuvânt nu a fost un cuvânt stop

Observăm formule de adresare precum "ser", "lord", "king" urmate de nume proprii. De asemenea vedem si perechi care formeaza nume de personaje cum sunt ”Daenerys” cu ”Targaryen” sau ”Stormborn”, chiar dacă este vorba de același personaj principal.

6.2. Numărarea și corelarea perechilor de cuvinte

În acest subcapitol ne interesează și cuvintele care tind să apară împreună în anumite sezoane, chiar dacă nu apar unul lângă celălalt. Pe parcursul acestui subcapitol, analizele sunt facute doar pe sezonul 1 al serialului *Game Of Thrones*.

6.2.1. Numărarea și corelarea

Luând în considerare doar sezonul 1, ne interesează ce cuvinte tind să apară în dialogul recitat de același personaj.

```
GOT_s1_words <- GOT_tidy %>%
  filter(Season == "Season 1")
### permite să numărăm seturi frecvente de cuvinte care apar recitate de acelasi personaj
word_pairs <- GOT_s1_words %>%
  pairwise_count(word, Name , sort = TRUE)
word_pairs
```

Figura 62. Input code

```
# A tibble: 1,758,420 x 3
   item1  item2      n
   <chr>  <chr>  <dbl>
 1 king   lord      36
 2 lord   king      36
 3 stark  lord      30
 4 boy    lord      30
 5 lord   stark     30
 6 lord   boy       30
 7 father king      29
 8 king   father    29
 9 father lord      28
10 lord   father    28
# ... with 1,758,410 more rows
```

Figura 63. Output code

Cea mai comună pereche de cuvinte dintr-o linie de dialog este „king” și „lord” care sunt forme de adresare des întâlnite în familiile regale și nobile din serial.

În continuare, putem găsi cu ușurință cuvintele care apar cel mai des cu ”lord”.

```
word_pairs %>%
  filter(item1 == "lord")
```

Figura 64. Input code

```
# A tibble: 2,870 x 3
   item1 item2       n
   <chr> <chr>   <dbl>
 1 lord  king       36
 2 lord  stark      30
 3 lord  boy        30
 4 lord  father     28
 5 lord  lady       25
 6 lord  hand       25
 7 lord  brother    23
 8 lord  king's     23
 9 lord  honor      22
10 lord  day        22
# ... with 2,860 more rows
```

Figura 65. Cuvinte des întâlnite împreuna cu ”lord”

Cele mai utilizate forme de adresare din serial sunt: ”lord king”, ”lord stark”, ”lord boy”, ș.a.m.d.

6.2.2. Corelație în perechi

Examinăm corelația (asocierea) dintre cuvinte, ceea ce indică cât de des ele apar împreună, în raport cu cât de des apar separat.

```
word_cors <- GOT_s1_words %>%
  group_by(word) %>%
  filter(n() >= 20) %>%
  pairwise_cor(word, Name, sort = TRUE)
word_cors
```

Figura 66. Input code

```
# A tibble: 8,742 x 3
   item1    item2    correlation
   <chr>    <chr>          <dbl>
 1 king's   landing        0.808
 2 landing  king's         0.808
 3 drogo    khal           0.785
 4 khal     drogo          0.785
 5 arryn    jon            0.691
 6 jon      arryn          0.691
 7 khal     dothraki       0.667
 8 dothraki khal           0.667
 9 brother  kill           0.635
10 kill     brother        0.635
# ... with 8,732 more rows
```

Figura 67. Ordinea descrescătoare a corelațiilor dintre cuvinte

Cuvintele ”king's landing” sunt corelate cel mai puternic pozitiv.

În continuare putem să alegem și alte cuvinte interesante și să găsim celelalte cuvinte care sunt cele mai asociate cu ele.

```
word_cors %>%
  filter(item1 %in% c("king", "blood", "kill", "throne")) %>%
  group_by(item1) %>%
  top_n(6) %>%
  ungroup() %>%
  mutate(item2 = reorder(item2, correlation)) %>%
  ggplot(aes(item2, correlation)) +
  geom_bar(stat = 'identity') +
  facet_wrap(~item1, scales = 'free') +
  coord_flip()
```

Figura 68. Input code

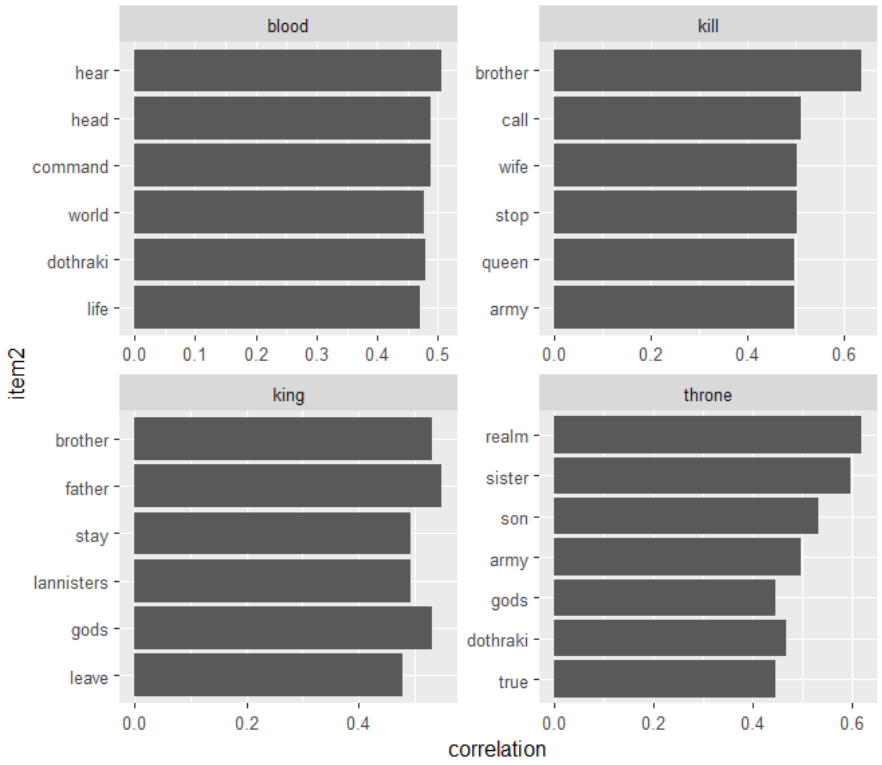

Figura 69. Cuvinte din sezonul 1 care au fost cele mai corelate cu „blood", „kill", „king" și „throne"

Cuvântul care este cel mai corelat cu "blood" este "hear", cu "kill" este "brother", cu "king" este "father", iar cu "throne" este "realm".

7. LDA

O altă metodă de analiză a datelor este Latent Dirichlet Allocation (LDA), folosită pentru modelarea subiectelor. Fiecare document (episod) este un amestec de subiecte. Ne imaginăm că fiecare episod conține cuvinte din mai multe subiecte în proporții speciale. Fiecare subiect este un amestec de cuvinte. LDA este o metodă matematică pentru estimarea celor două simultan: găsirea amestecului de cuvinte asociate fiecărui subiect.

Am efectuat LDA pe toate datele disponibile, pentru un model cu 8 subiecte. Numărul 8 reprezintă și numărul de sezoane pe care le are serialul analizat.

```
tidy_lda <- Game_of_Thrones %>%
  ungroup() %>%
  unnest_tokens(word, Sentence) %>%
  distinct() %>%
  anti_join(stop_words) %>%
  filter(nchar(word) > 2) %>%
  select(Name,Episode,Season,word)


topics <- LDA(cast_dtm(data = tidy_lda %>%
                         count(Name, word) %>%
                         ungroup(),
                       term = word,
                       document = Name,
                       value = n),
              k = 8, control = list(seed = 1234)) %>%
  tidy(matrix = "beta") %>%
  group_by(topic) %>%
  arrange(desc(beta)) %>%
  top_n(12, beta) %>%
  ungroup()

topics %>%
  arrange(topic, -beta) %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = 'free') +
  coord_flip() +
  ggtitle("Topic modeling using LDA")
```

Figura 70. Input code

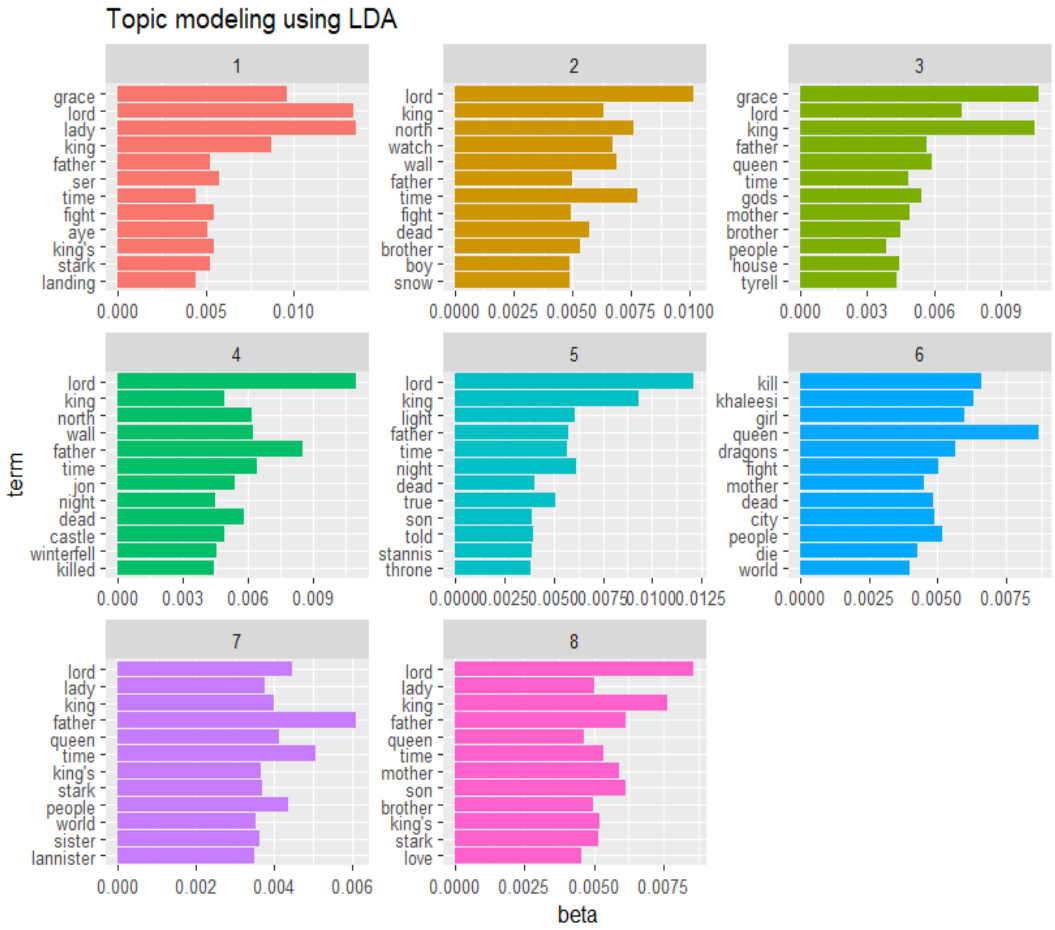

Figura 71. Lda pe cele 8 subiecte

Când analizăm întregul set de date, putem observa că subiectele tind să se formeze în jurul cuvintelor cu anumite sentimente. În subiectele 1, 2, 3, 5, 7 și 8 putem vedea predominant cuvinte cu sentiment pozitiv, în timp ce subiectul 6 este mai conotat cu sentimente negative și emoții de frică și tristețe. Subiectul 4 este, de asemenea, compus în principal din cuvinte cu sentimente negative, dar arată mai mult frică și tristețe.

```
tidy_lda_S8 <- Game_of_Thrones %>%
  filter(Season=="Season 8") %>%
  ungroup() %>%
  unnest_tokens(word, Sentence) %>%
  distinct() %>%
  anti_join(stop_words) %>%
  filter(nchar(word) > 2) %>%
  select(Name,Episode,Season,word)


topics_S8 <- LDA(cast_dtm(data = tidy_lda_S8 %>%
                            count(Name, word) %>%
                            ungroup(),
                          term = word,
                          document = Name,
                          value = n),
                 k = 7, control = list(seed = 1234)) %>%
  tidy(matrix = "beta") %>%
  group_by(topic) %>%
  arrange(desc(beta)) %>%
  top_n(7, beta) %>%
  ungroup()

topics_S8 %>%
  arrange(topic, -beta) %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = 'free') +
  coord_flip() +
  ggtitle("Topic modeling using LDA in the S8")
```

Figura 72. Input code

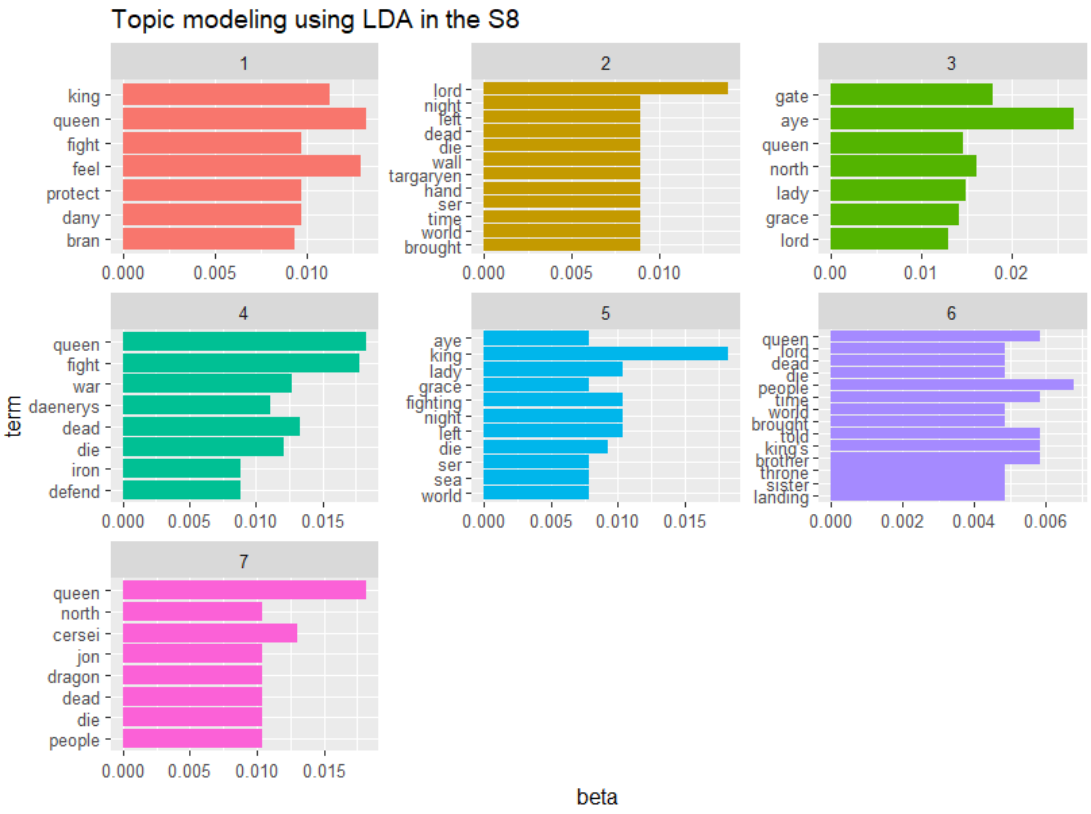

Figura 73. Lda pe sezonul 8

Când evaluăm doar sezonul 8, putem observa că subiectele sunt distribuite mai uniform, cu o majoritate de cuvinte care apar în fiecare subiect, ceea ce indică faptul că motivele de bază ale subiectului 8 tind să oscileze în jurul unor subiecte similare din sezoanele anterioare.

8. Concluzii

Pe baza rezultatelor acestui studiu, se poate spune că analiza sentimentelor, procesarea textului, relațiile dintre cuvinte și modelarea subiectelor pentru serialul Game Of Thrones au ajutat la îndeplinirea obiectivelor propuse la începutul lucrării .

Conform rezultatelor care au dorit să evidențieze personajele care au avut cele mai multe linii de dialog din serial, s-au remarcat personaje precum Tyrion Lannister cu 1760 de linii de dialog, Jon Snow cu 1133 și Daenerys Targaryen cu 1048. Pașii următori au fost de procesare a textului, care au inclus normalizarea, tokenizarea și eliminarea cuvintelor comune din text care au fost efectuate cu ajutorul aplicației RStudio.

Cuvântul care a fost cel mai folosit în serial este ”lord”. Acest cuvânt a fost folosit de 1112 ori și este urmat de ”king” cu 812.

Procesul de etichetare a sentimentului a obținut date clasificate pozitiv cel mai mult pentru sezonul 8. Tot cu ajutorul sentimentelor, au rezultat că cele mai importante personaje care joacă roluri principale în serialul *Game of Thrones* folosesc cuvinte care provoacă emoții pozitive, negative și de încredere.

În analiza frecvenței relative a cuvintelor, "lord" apare de 239 în sezonul 1 din totalul de 11864 de cuvinte din acest sezon. Acesta este cuvantul cu cea mai mare frecvență și este urmat de "lord" care apare de 205 în sezonul 2. De asemenea, cea mai comună pereche de cuvinte dintr-o linie de dialog este „king” și „lord” care sunt forme de adresare des întâlnite în familiile regale și nobile din serial.