Eksploracja tekstu i wyszukiwanie informacji w mediach społecznościowych

LABORATORIUM 6

- Słownikowa analiza sentymentu
- Słownik tidytext
- Emocje w sekcjach tekstu
- Emocjonalny wordcloud
- Model Russela

Słownikowa analiza sentymentu

Analiza sentymentu (*sentiment analysis* - SA) jest jednym z najistotniejszych działów text miningu. Jak sama nazwa wskazuje, zajmuje się oceną zawartości emocjonalnej, która można uzyskać analizując daną próbkę tekstu. Ogólnie można pwoiedzieć, że dwoma głównymi podejściami w SA są metody bez nadzory oraz pod nadzorem. Pierwsza gama metod wykorzystuje najczęściej uprzednio stworzone leksykony emocjonalne i opierając się na nich ocenia sentument w tekście. Druga to typowe podjeście typu data mining - na zbiorze uczącym trenujemy określony algorytm, aby później stosować go do innych danych. Na dzisiejszych zajęciach zajmiemy się pierwszą metodą (w najprostszym wydaniu).

Wykorzystamy zbiór czterach książek Juliusza Verne'a:

```
# PRZYKŁAD 6.1

library(gutenbergr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(magrittr)
library(tidytext)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:magrittr':
##
##     extract
```

```
library(ggplot2)

 g <- gutenberg_works()

 id <- c(83, 103, 164, 1268)
 verne <- gutenberg_download(id)
```

```
## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harves
t
```

```
## Using mirror http://aleph.gutenberg.org
```

```
 books <- g[g$gutenberg_id %in% id, c("gutenberg_id","title")]

 verne %<>% left_join(books) %>%
    mutate(gutenberg_id = NULL)
```

```
## Joining, by = "gutenberg_id"
```

I tak jak poprzednio, rozbijamy zbiory na poszczególne tokeny - czyli słowa.

```
# PRZYKŁAD 6.2

verne_books <- verne %>%
  group_by(title) %>%
  mutate(linenumber = row_number()) %>%
  ungroup() %>%
  unnest_tokens(word, text)
```

Słownik tidytext

Teraz, oczywiście, kluczową sprawą jest zdobycie jakiegoś słownika, który powiąże nam poszczególne słowa z zawartymi w nich emocjami. Bibioteka **tidytext** udostępnia zbiór danych **sentiments**, będący de facto zbiorem aż czterech oddzielnych leksykonów: NRC, Bing, Finn Arup Nielsen i Loughran. Każdy z nich w inny sposób opisuje emocje: drugi podaje wartości numeryczne z zakresu $[-5; 5]$, trzy pozostałe korzystają z opisowego określenia sentymentu. Do każdego słownika mozna dostać się za pomocą wrappera **get_sentiments()**. Poniżej wypiszemy emocje, które zwraca slownik NRC.

```
# PRZYKŁAD 6.3

sentiments
```

```
## # A tibble: 27,314 x 4
##    word        sentiment lexicon score
##    <chr>       <chr>     <chr>   <int>
##  1 abacus      trust     nrc        NA
##  2 abandon     fear      nrc        NA
##  3 abandon     negative  nrc        NA
##  4 abandon     sadness   nrc        NA
##  5 abandoned   anger     nrc        NA
##  6 abandoned   fear      nrc        NA
##  7 abandoned   negative  nrc        NA
##  8 abandoned   sadness   nrc        NA
##  9 abandonment anger     nrc        NA
## 10 abandonment fear      nrc        NA
## # ... with 27,304 more rows
```

```
nrc <- get_sentiments("nrc")

table(nrc$sentiment)
```

```
##
##        anger anticipation      disgust         fear          joy
##         1247          839         1058         1476          689
##     negative     positive      sadness     surprise        trust
##         3324         2312         1191          534         1231
```

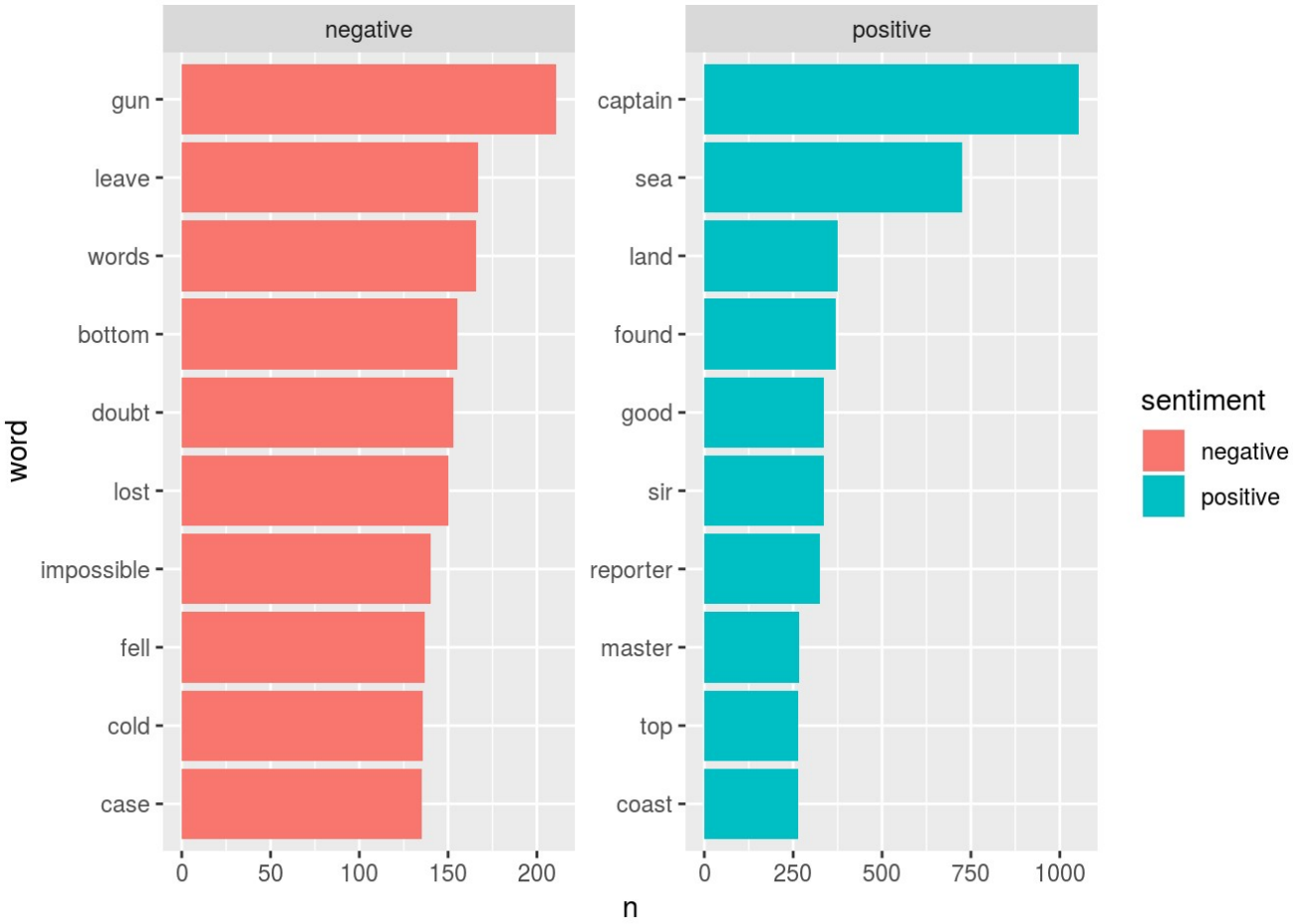
Korzystając z takiego zbioru możemy np. określić jakie słowa w wybranych książkach odpowiadają za negatywny lub pozytywny sentyment

```
# PRZYKŁAD 6.4

pos_neg <- verne_books %>%
  inner_join(nrc) %>%
  filter(sentiment %in% c("positive", "negative")) %>%
  group_by(sentiment) %>%
  count(word, sort = T) %>%
  top_n(10, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n))
```

```
## Joining, by = "word"
```

```
ggplot(pos_neg) + geom_col(aes(word, n, fill = sentiment)) +
  coord_flip() +
  facet_wrap( ~ sentiment, scales = "free")
```



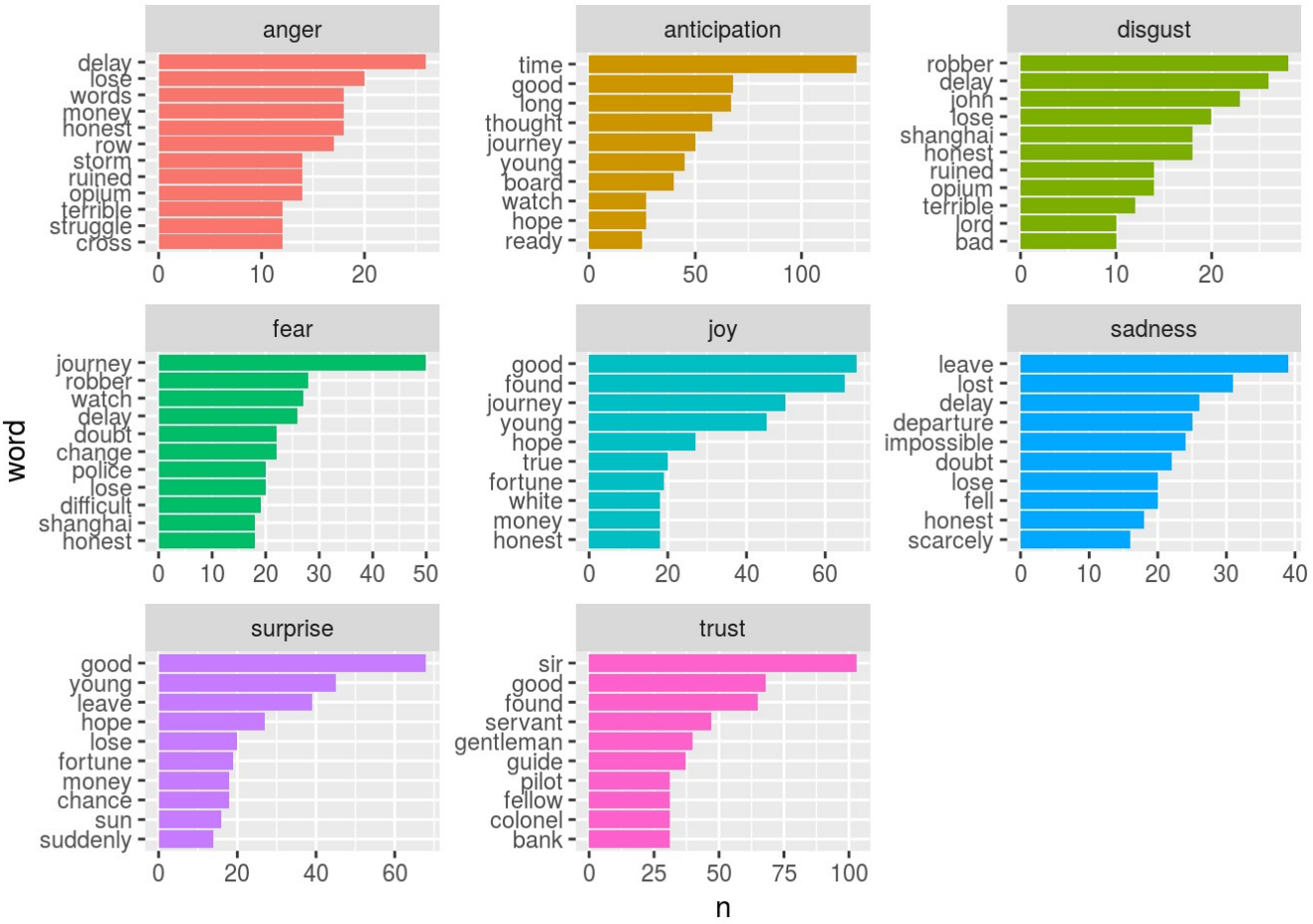
W podobny sposób określimy emocje inne niż tylko negatywny/pozytywny.

```
# PRZYKŁAD 6.5

nrc_class <- verne_books %>%
  filter(title == books[["title"]][2]) %>%
  inner_join(nrc) %>%
  filter(!(sentiment %in% c("positive", "negative"))) %>%
  group_by(sentiment) %>%
  count(word, sort = T) %>%
  top_n(10, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n))
```

```
## Joining, by = "word"
```

```
ggplot(nrc_class) + geom_col(aes(word, n, fill = sentiment), show.legend = FALSE) +
  coord_flip() +
  facet_wrap(~sentiment, nrow = 3, scales = "free")
```



Emocje w sekcjach tekstu

Poprzednio wyznaczaliśmy wartości emocji w całych tekstach, ale oczywiście tekst nie zawsze jest zwarta i jednolitą jednostką. Stąd sens ma rozłożenie książki na poszczególne fragmenty, i badania jak zmienia się sentyment w trakcie jak toczy się fabuła. Tym razem wykorzystamy słownik **bing** oraz funkcję **spread()**, która, jak sama nazwa wskazuje, rozrzuca wartości po kolumnach.

```
# PRZYKŁAD 6.6

verne_senti_bing <- verne_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(title, index = linenumber %/% 80, sentiment)
```

```
## Joining, by = "word"
```

```
verne_senti_bing
```

```
## # A tibble: 1,393 x 4
##    title                          index sentiment     n
##    <chr>                          <dbl> <chr>     <int>
##  1 Around the World in Eighty Days     0 negative      7
##  2 Around the World in Eighty Days     0 positive     15
##  3 Around the World in Eighty Days     1 negative      8
##  4 Around the World in Eighty Days     1 positive     19
##  5 Around the World in Eighty Days     2 negative     12
##  6 Around the World in Eighty Days     2 positive     25
##  7 Around the World in Eighty Days     3 negative      9
##  8 Around the World in Eighty Days     3 positive     31
##  9 Around the World in Eighty Days     4 negative     16
## 10 Around the World in Eighty Days     4 positive     36
## # ... with 1,383 more rows
```

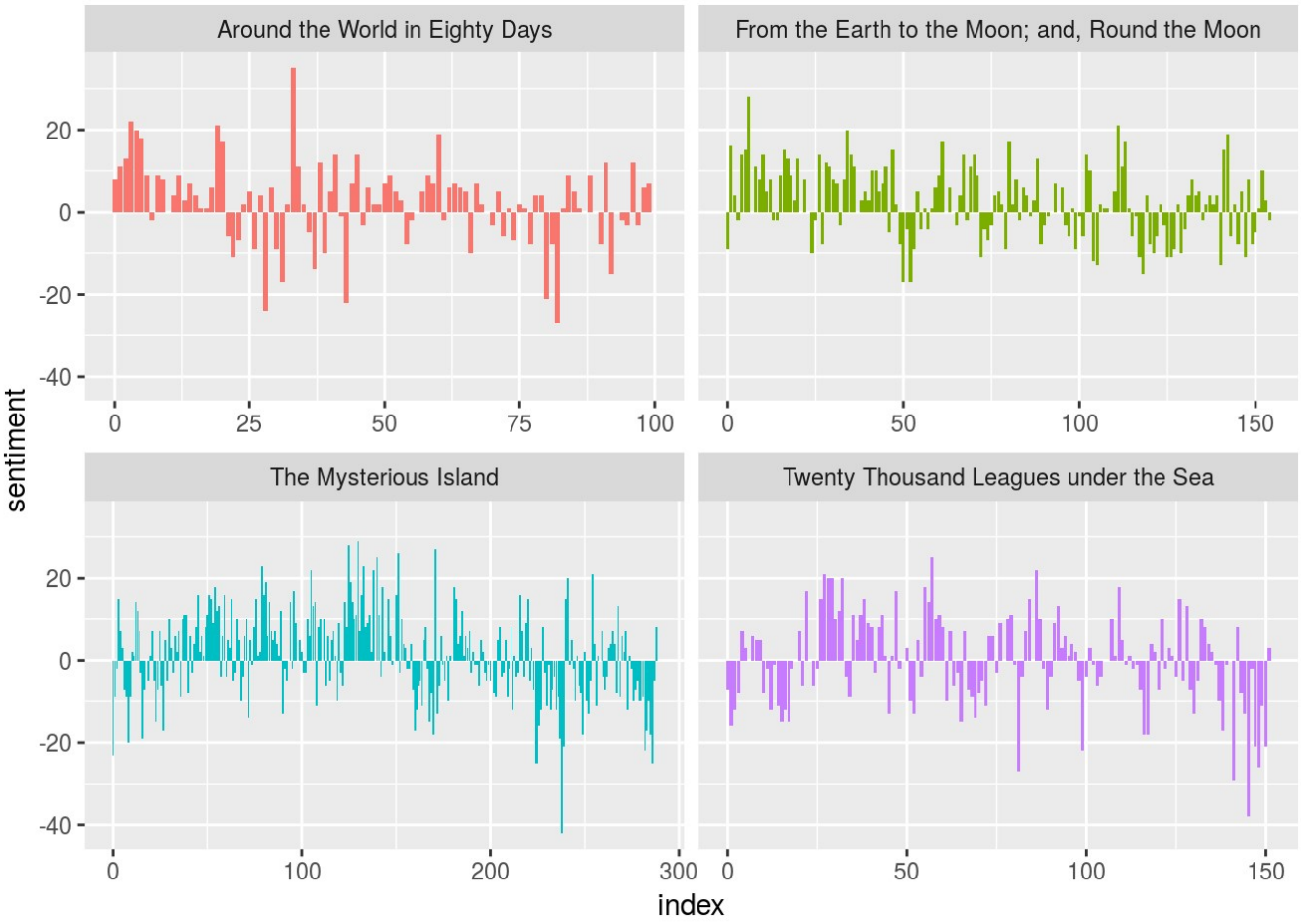
```
verne_senti_bing %<>%
  spread(sentiment, n, fill = 0)

verne_senti_bing
```

```
## # A tibble: 697 x 4
##    title                          index negative positive
##    <chr>                          <dbl>    <dbl>    <dbl>
##  1 Around the World in Eighty Days     0        7       15
##  2 Around the World in Eighty Days     1        8       19
##  3 Around the World in Eighty Days     2       12       25
##  4 Around the World in Eighty Days     3        9       31
##  5 Around the World in Eighty Days     4       16       36
##  6 Around the World in Eighty Days     5        6       24
##  7 Around the World in Eighty Days     6       13       22
##  8 Around the World in Eighty Days     7       15       13
##  9 Around the World in Eighty Days     8        9       18
## 10 Around the World in Eighty Days     9       18       26
## # ... with 687 more rows
```

```
verne_senti_bing %<>%
  mutate(sentiment = positive - negative)

ggplot(verne_senti_bing, aes(index, sentiment, fill = title)) +
  geom_col(show.legend = FALSE) +
  facet_wrap( ~ title, ncol = 2, scales = "free_x")
```



Słowniki dostępne w pakiecie **tidytext** nie są, rzecz jasna, jedynymi dostępnymi materiałami związanymi z sentymentem. W 2013 roku została opublikowana interesująca praca przez Amy Warriner i współpracowników (https://link.springer.com/article/10.3758%2Fs13428-012-0314-x), w której zawarto wartości walencji, pobudzenia i dominancji prawie 14000 angielskich słów. Jak zostało wspomniane na Wykładzie 5 (http://www.if.pw.edu.pl/~julas/TEXT/pliki/TEXT5.pdf), walencja i pobudzenie do dwie często wykorzystywane składowe emocji, pierwsza mówi o nacechowaniu emocji (negatywna, pozytywna), druga o intensywności. We wspomnianej pracy dla obu zmiennych pojawiają się wartości w skali $[1; 9]$. Zbiór ma kilkadziesiąt kolumn, ale nas interesuje jedynie średnia walencja i pobudzenie poszczególnych słów:

```
# PRZYKŁAD 6.7

emo <- as_tibble(read.csv("http://www.fizyka.pw.edu.pl/~julas/TEXT/lab/Ratings_Warrine
r_et_al.csv", stringsAsFactors = F))
emo
```

```
## # A tibble: 13,915 x 65
##        X Word  V.Mean.Sum V.SD.Sum V.Rat.Sum A.Mean.Sum A.SD.Sum A.Rat.Sum
##    <int> <chr>      <dbl>    <dbl>     <int>      <dbl>    <dbl>     <int>
##  1     1 aard.       6.26     2.21        19       2.41     1.4          22
##  2     2 abal.       5.3      1.59        20       2.65     1.9          20
##  3     3 aban.       2.84     1.54        19       3.73     2.43         22
##  4     4 aban.       2.63     1.74        19       4.95     2.64         21
##  5     5 abbey       5.85     1.69        20       2.2      1.7          20
##  6     6 abdo.       5.43     1.75        21       3.68     2.23         22
##  7     7 abdo.       4.48     1.59        23       3.5      1.82         22
##  8     8 abdu.       2.42     1.61        19       5.9      2.57         20
##  9     9 abdu.       2.05     1.31        19       5.33     2.2          21
## 10    10 abide       5.52     1.75        21       3.26     2.22         23
## # ... with 13,905 more rows, and 57 more variables: D.Mean.Sum <dbl>,
## #   D.SD.Sum <dbl>, D.Rat.Sum <int>, V.Mean.M <dbl>, V.SD.M <dbl>,
## #   V.Rat.M <int>, V.Mean.F <dbl>, V.SD.F <dbl>, V.Rat.F <int>,
## #   A.Mean.M <dbl>, A.SD.M <dbl>, A.Rat.M <int>, A.Mean.F <dbl>,
## #   A.SD.F <dbl>, A.Rat.F <int>, D.Mean.M <dbl>, D.SD.M <dbl>,
## #   D.Rat.M <int>, D.Mean.F <dbl>, D.SD.F <dbl>, D.Rat.F <int>,
## #   V.Mean.Y <dbl>, V.SD.Y <dbl>, V.Rat.Y <int>, V.Mean.O <dbl>,
## #   V.SD.O <dbl>, V.Rat.O <int>, A.Mean.Y <dbl>, A.SD.Y <dbl>,
## #   A.Rat.Y <int>, A.Mean.O <dbl>, A.SD.O <dbl>, A.Rat.O <int>,
## #   D.Mean.Y <dbl>, D.SD.Y <dbl>, D.Rat.Y <int>, D.Mean.O <dbl>,
## #   D.SD.O <dbl>, D.Rat.O <int>, V.Mean.L <dbl>, V.SD.L <dbl>,
## #   V.Rat.L <int>, V.Mean.H <dbl>, V.SD.H <dbl>, V.Rat.H <int>,
## #   A.Mean.L <dbl>, A.SD.L <dbl>, A.Rat.L <int>, A.Mean.H <dbl>,
## #   A.SD.H <dbl>, A.Rat.H <int>, D.Mean.L <dbl>, D.SD.L <dbl>,
## #   D.Rat.L <int>, D.Mean.H <dbl>, D.SD.H <dbl>, D.Rat.H <int>
```

```
emo <- emo %>%
  select(word = Word, valence = V.Mean.Sum, arousal = A.Mean.Sum)
emo
```

```
## # A tibble: 13,915 x 3
##    word        valence arousal
##    <chr>         <dbl>   <dbl>
##  1 aardvark       6.26    2.41
##  2 abalone        5.3     2.65
##  3 abandon        2.84    3.73
##  4 abandonment    2.63    4.95
##  5 abbey          5.85    2.2
##  6 abdomen        5.43    3.68
##  7 abdominal      4.48    3.5
##  8 abduct         2.42    5.9
##  9 abduction      2.05    5.33
## 10 abide          5.52    3.26
## # ... with 13,905 more rows
```

Mając już wczytany zbiór, możemy pokusić się o stworzenie podobnego wykresu, co poprzednio, tyle, że

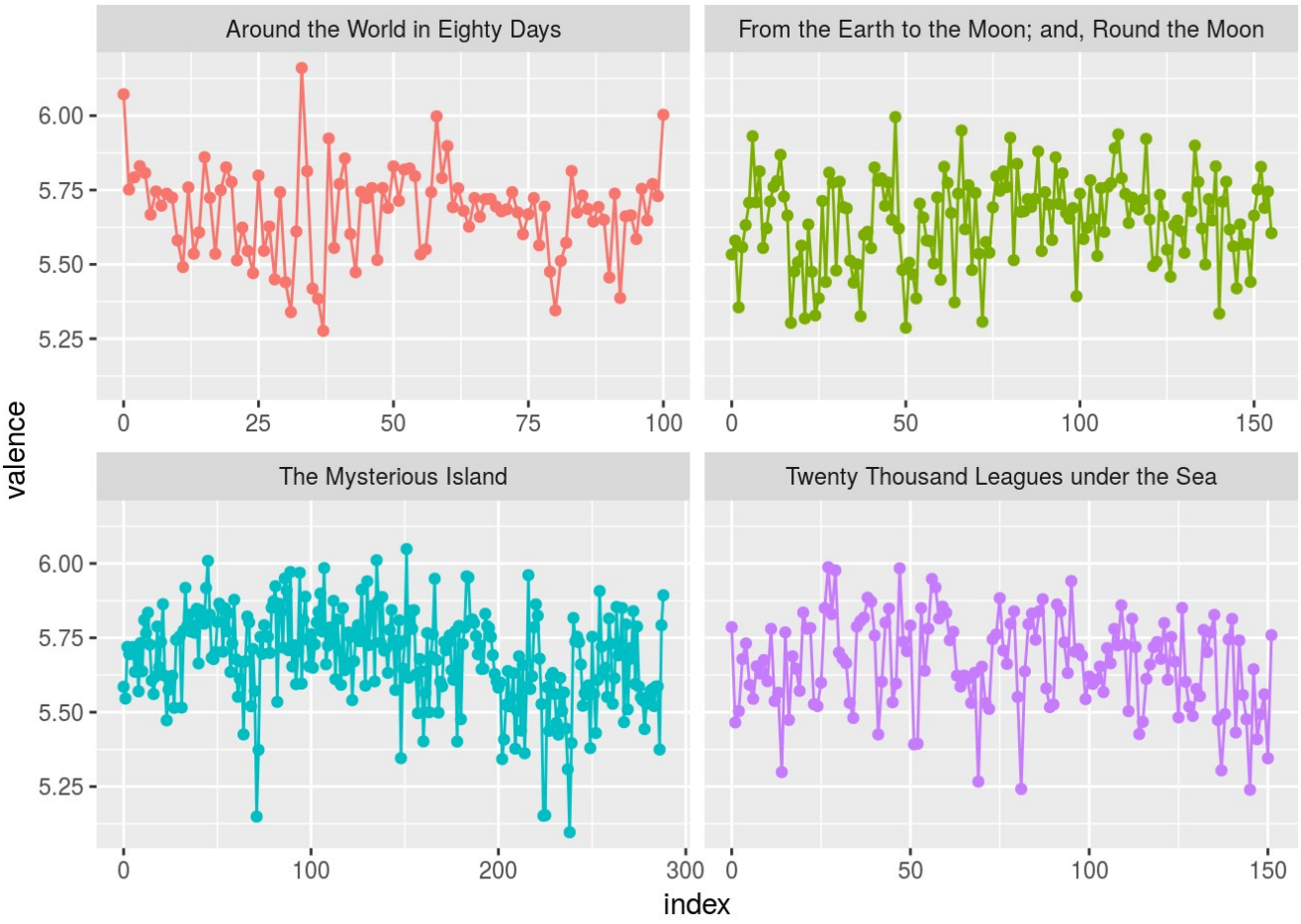
korzystając z innych danych.

```
# PRZYKŁAD 6.8

verne_senti_warriner <- verne_books %>%
  inner_join(emo) %>%
  group_by(title, index = linenumber %/% 80) %>%
  summarise(valence = mean(valence))
```

```
## Joining, by = "word"
```

```
ggplot(verne_senti_warriner, aes(index, valence, color = title)) +
  geom_point(show.legend = FALSE) +
  geom_line(show.legend = FALSE) +
  facet_wrap( ~ title, ncol = 2, scales = "free_x")
```



Emocjonalny wordcloud

Chmury słów (*wordclouds*) są bardzo często wykorzystywanym narzędziem do wizualizacji istotności (np. liczby) słów w danym dokumencie. Tym wygodniejsze jest ubranie tej metody w możliwość wyświetlania w różny sposób słów pozytywnych i negatywnych. Wkorzystamy tu funkcję **acast()** z pakietu **reshape2**, która po prostu w zgrabny sposób transformuje zliczenia słów jako pozytywnych i negatywnych, a następnie "nagniemy" funkcję **comparison.cloud()**, która w swoim zamyśle ma porównywać słowa z różnych dokumentów.

```
# PRZYKŁAD 6.9

library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(wordcloud)
```

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

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

```
## Joining, by = "word"
```

Model Russela

Zgodnie z Wykładem 5 (http://www.if.pw.edu.pl/~julas/TEXT/pliki/TEXT5.pdf), emocje opisane za pomocą walencji i pobudzenia tworzą tzw. model kołowy Russela. Korzystając z dostępnych danych można sprawdzić, na ile ten model faktycznie jest adekwatny do rzeczywistości. Na początek sprawdźmy jak wyglądają we współrzędnych walencja-pobudzenie słowa ze słownika NRC.

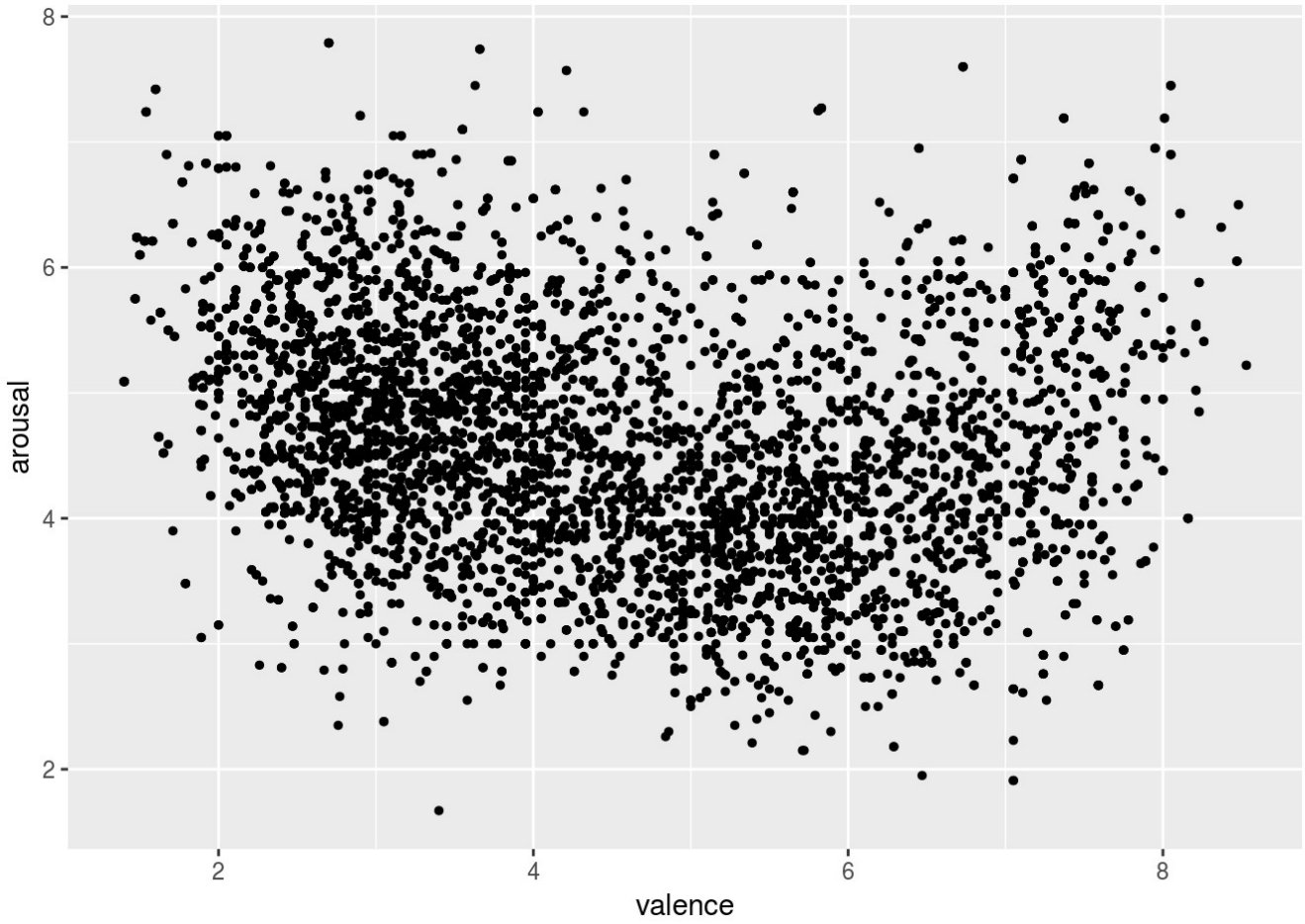
```
# PRZYKŁAD 6.10

sent <- nrc %>%
  filter(!(sentiment %in% c("negative", "positive")))

sent_comb <- inner_join(sent, emo)
```

```
## Joining, by = "word"
```

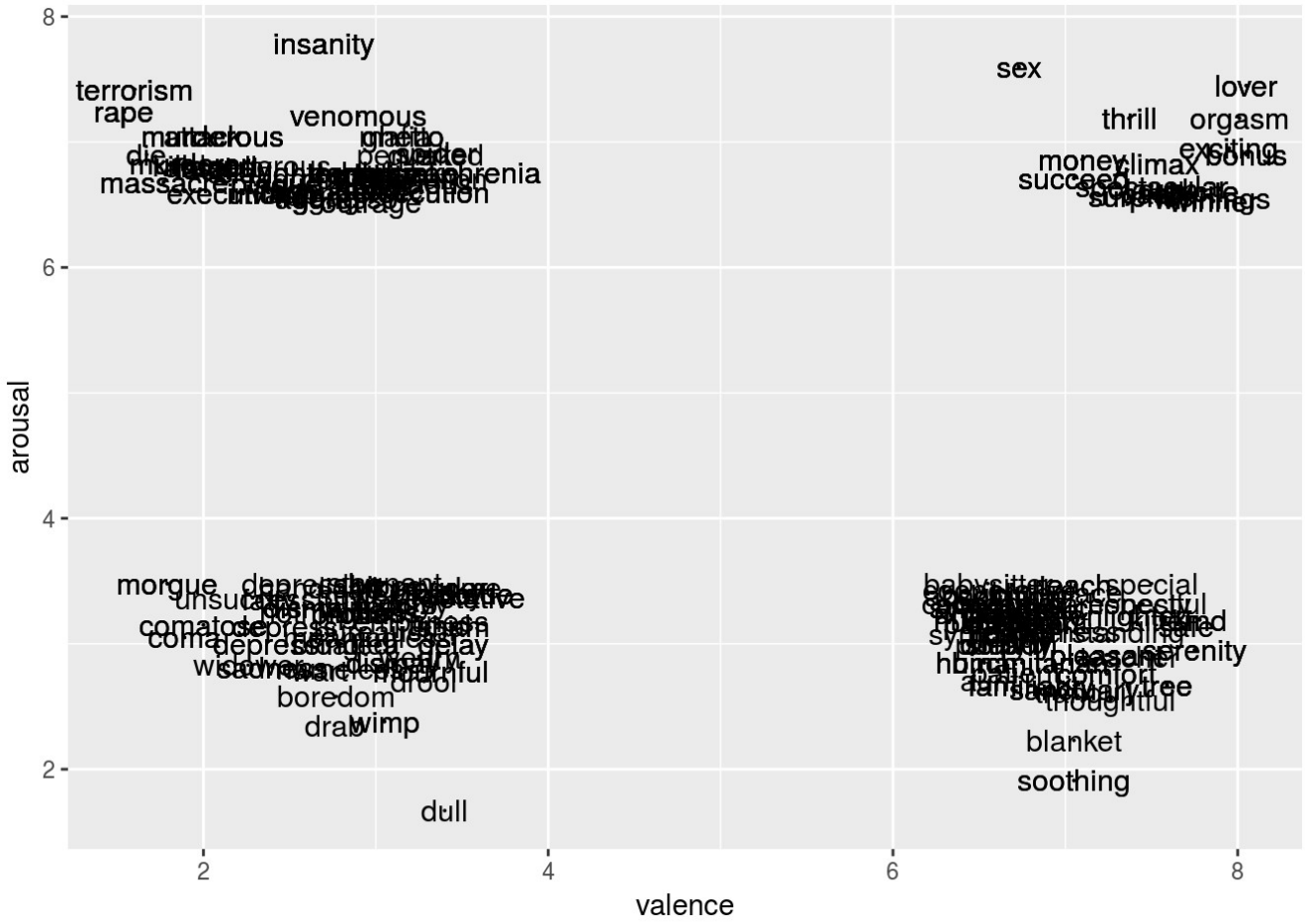
```
ggplot(sent_comb, aes(x = valence, y = arousal)) +
  geom_point(size = 1)
```



Jak widać, ciężko jest uzyskac duzy rozrzut pobudzenia dla obiektywnych wartości walencji. Ciekawe może okazać się wypisanie najbardziej skrajnych słów.

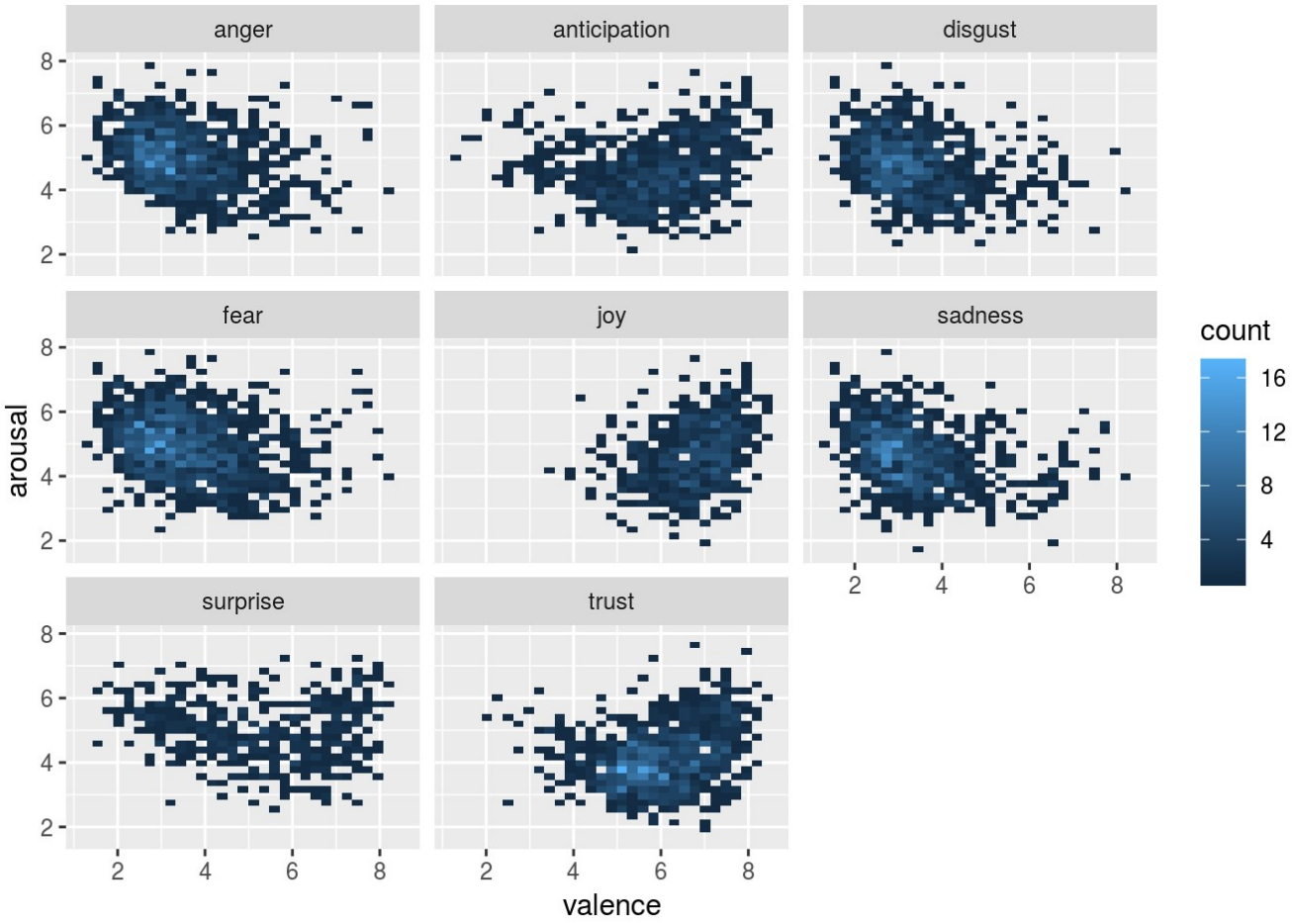
```
# PRZYKŁAD 6.11

ggplot(sent_comb %>% filter(abs(arousal - 5) > 1.5 & abs(valence - 5) > 1.5), aes(x =
valence, y = arousal)) +
  geom_point(size = 0) +
  geom_text(aes(label = word), size = 4)
```



Dla nas istotne jest to, że możemy bezpośrednio powiązać poszczególne emocje (strach, radość etc) i wartościami walencji v i pobudzenia a. W tym celu histogramujemy ile słów o konkretnej emocji znajduje się w określonym punkcie v, a.

```
# PRZYKŁAD 6.12
ggplot(sent_comb) +
  geom_bin2d(aes(x = valence, y = arousal)) +
  facet_wrap(~sentiment)
```

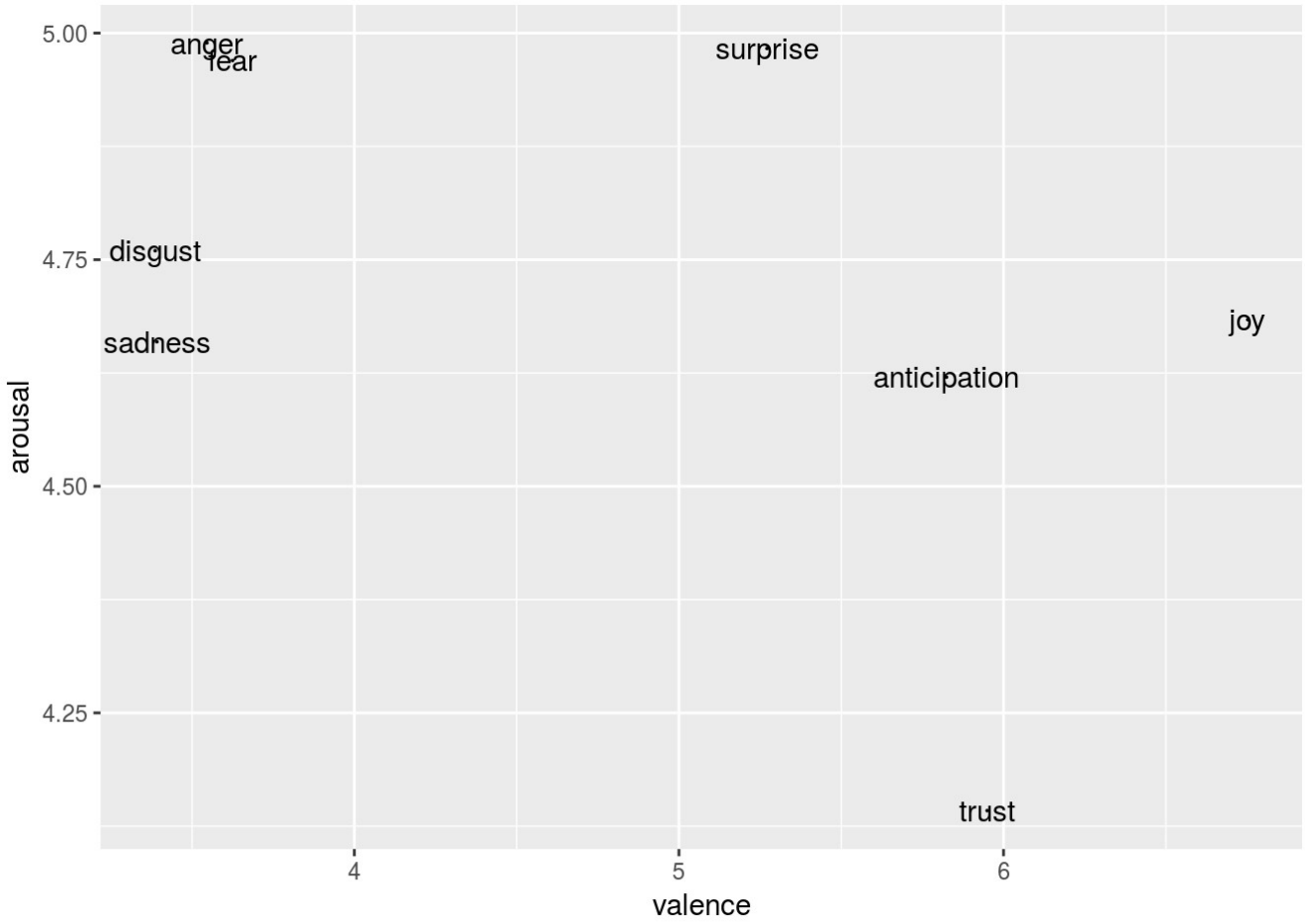


Wreszcie, po dokonaniu uśrednienia możemy porównać otrzymane pozycje emocji z modelem Russela.

```
# PRZYKŁAD 6.13

sent_comb_sum <- sent_comb %>%
  group_by(sentiment) %>%
  summarise(valence = mean(valence), arousal = mean(arousal))

ggplot(sent_comb_sum, aes(x = valence, y = arousal)) +
  geom_point(size = 0) + geom_text(aes(label = sentiment))
```


5.00
4.75
4.50
4.25
arousal
anger
fear
surprise
disgust
sadness
joy
anticipation
trust
4
5
6
valence