SPRAWOZDANIE

Zajęcia: Analiza Procesów Uczenia
Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 7
Data 07.06.2023
Temat: "Problemy NLP w uczeniu maszynowym"
Wariant: 2

Szymon Białek
Informatyka II stopień
stacjonarne
1 semestr,
Gr.1

Wszystkie pliki i komendy można obejrzeć pod linkiem:
https://github.com/NynyNoo/Analiza-procesow-uczenia/tree/main/lab7

Polecenie

Dotyczy analizy tekstu, w tym listę częstotliwości słów, budowanie chmury słów, skojarzeń, sentiment analysis, emotion analysis, bigramów, grafów powiązań. Warianty zadania są określone tekstem w języku angielskim umieszczonym na portalu en.wikipedia.org (główna część artykułu bez literatury)
2. https://en.wikipedia.org/wiki/Europe

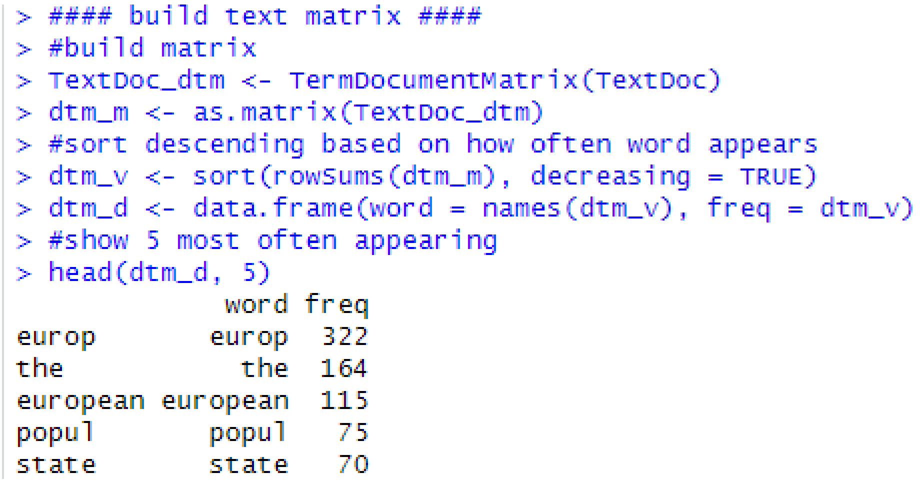
Wykorzystane komendy oraz wyniki działania programu

```
setwd("D:/MGR/APU/lab7")
library("tm")
library("SnowballC")
library("wordcloud")
library("RColorBrewer")
library("syuzhet")
library("ggplot2")

#read text
text <- readLines("Europa.txt", warn=FALSE)

#convert text to object
TextDoc <- Corpus(VectorSource(text))

#clean text
#remove special characters
toSpace <- content_transformer(function(x, pattern) gsub(pattern, " ", x)) # Funkcja zamiany znaku
TextDoc <- tm_map(TextDoc, toSpace, "/")
TextDoc <- tm_map(TextDoc, toSpace, "@")
TextDoc <- tm_map(TextDoc, toSpace, "\\|")
TextDoc <- tm_map(TextDoc, toSpace, ":")
TextDoc <- tm_map(TextDoc, toSpace, ";")
TextDoc <- tm_map(TextDoc, toSpace, ",")
TextDoc <- tm_map(TextDoc, toSpace, "/")
#remove numbers
TextDoc <- tm_map(TextDoc, removeNumbers)
#remove stop characters
TextDoc <- tm_map(TextDoc, removeWords, stopwords("english"))
#remove proprietary characters
TextDoc <- tm_map(TextDoc, removeWords, c("\\[", "\\]"))
#remove punctuation
TextDoc <- tm_map(TextDoc, removePunctuation)
#remove whitespaces
```

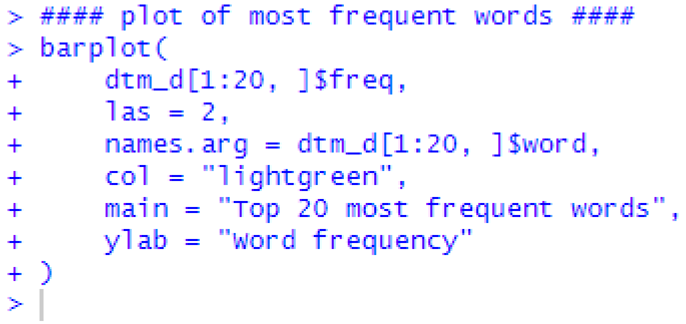
```
TextDoc <- tm_map(TextDoc, stripWhitespace)
#change to basic form
TextDoc <- tm_map(TextDoc, stemDocument)
#to lower
TextDoc <- tm_map(TextDoc, content_transformer(tolower))

#build text matrix
TextDoc_dtm <- TermDocumentMatrix(TextDoc)
dtm_m <- as.matrix(TextDoc_dtm)
#sort descending based on how often word appears
dtm_v <- sort(rowSums(dtm_m), decreasing = TRUE)
dtm_d <- data.frame(word = names(dtm_v), freq = dtm_v)
#show 5 most often appearing
head(dtm_d, 5)

#plot of most frequent words
barplot(
  dtm_d[1:20, ]$freq,
  las = 2,
  names.arg = dtm_d[1:20, ]$word,
  col = "lightgreen",
  main = "Top 20 most frequent words",
  ylab = "Word frequency"
)

#generate word cloud
set.seed(1234)
wordcloud(
  words = dtm_d$word,
  freq = dtm_d$freq,
  scale = c(5, 0.5),
  min.freq = 1,
  max.words = 100,
  random.order = FALSE,
  rot.per = 0.40,
  colors = brewer.pal(8, "Dark2")
)

#Kojarzenia slow
findAssocs(
  TextDoc_dtm,
  terms = c("learn", "machine", "algorithm", "train"),
  corlimit = 0.5
)
#find asoociation for words that appear at least 20 times
findAssocs(
  TextDoc_dtm,
  terms = findFreqTerms(TextDoc_dtm, lowfreq = 20),
  corlimit = 0.5
)

#sentiment analysis
```

```
syuzhet_vector <- get_sentiment(text, method = "syuzhet")
bing_vector <- get_sentiment(text, method = "bing")
nrc_vector <- get_sentiment(text, method = "nrc")
#compare analysis
rbind(
  sign(head(syuzhet_vector)),
  sign(head(bing_vector)),
  sign(head(nrc_vector))
)

#emotion classification
d <- get_nrc_sentiment(as.vector(dtm_d$word))
head(d,10)
#transpose
td <- data.frame(t(d))
#sum frequency of emotions for first 56 words
td_new <- data.frame(rowSums(td[1:56]))
#clear result
names(td_new)[1] <- "count"
td_new <- cbind("sentiment" = rownames(td_new), td_new)
rownames(td_new) <- NULL
td_new2 <- td_new[1:8,]
#plot - words tied to emotions
quickplot(
  sentiment,
  data = td_new2,
  weight = count,
  geom = "bar",
  fill = sentiment,
  ylab = "count"
) + ggtitle("Survey sentiments")
#plot - percent of each emotion
barplot(
  sort(colSums(prop.table(d[, 1:8]))),
  horiz = TRUE,
  cex.names = 0.7,
  las = 1,
  main = "Emotions in Text",
  xlab = "Percentage"
)
```

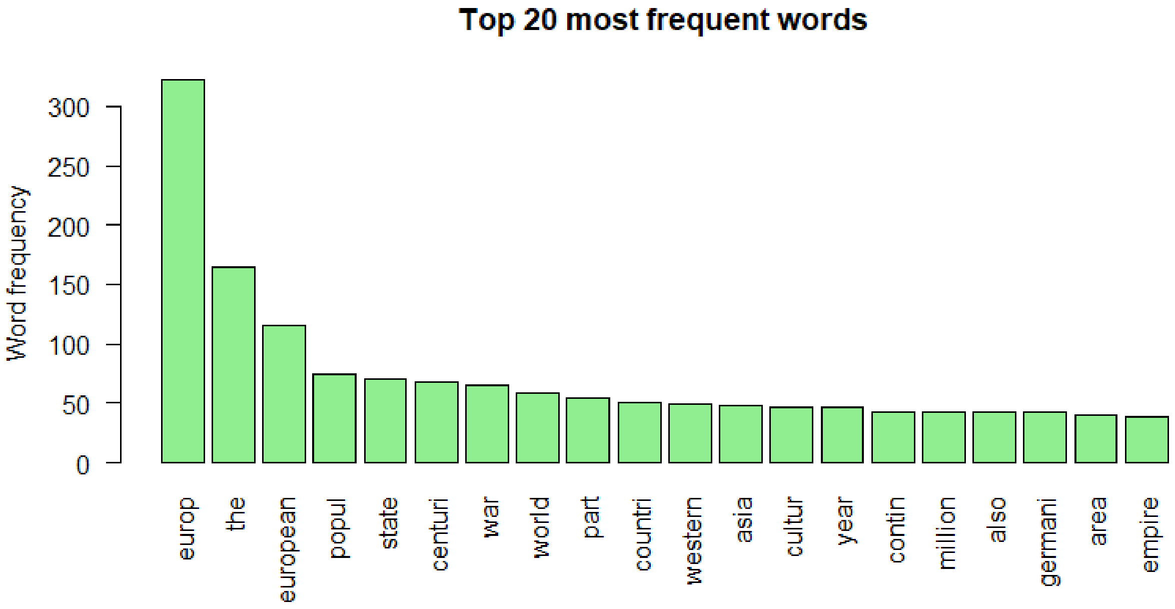
Wizualizacja Danych

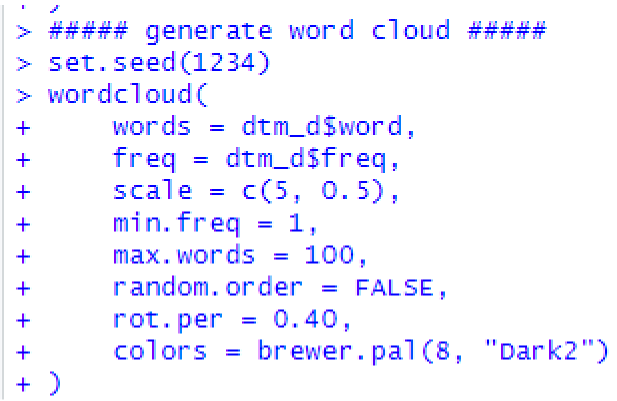


```
> #### build text matrix ####
> #build matrix
> TextDoc_dtm <- TermDocumentMatrix(TextDoc)
> dtm_m <- as.matrix(TextDoc_dtm)
> #sort descending based on how often word appears
> dtm_v <- sort(rowSums(dtm_m), decreasing = TRUE)
> dtm_d <- data.frame(word = names(dtm_v), freq = dtm_v)
> #show 5 most often appearing
> head(dtm_d, 5)
                   word freq
europ             europ  322
the                 the  164
european       european  115
popul             popul   75
state             state   70
```

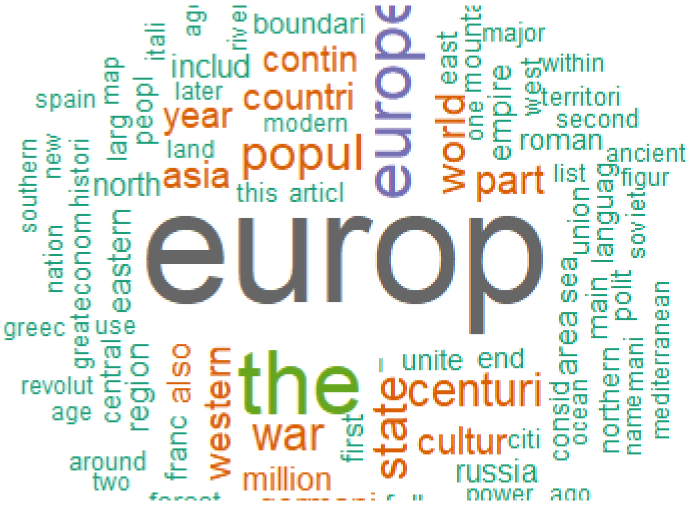


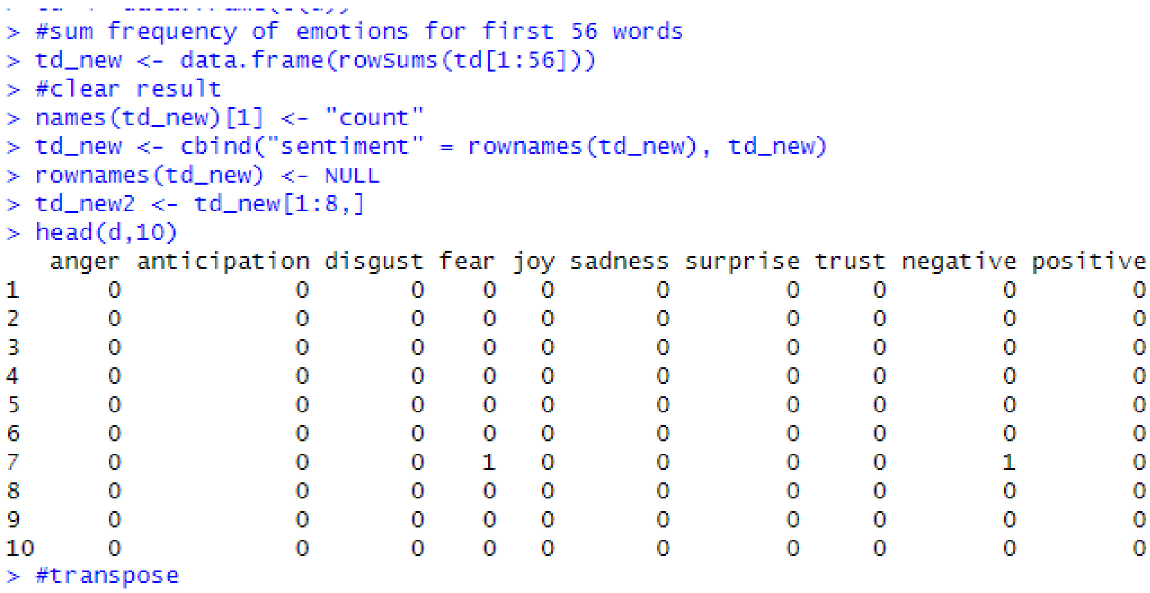
```
> #### plot of most frequent words ####
> barplot(
+     dtm_d[1:20, ]$freq,
+     las = 2,
+     names.arg = dtm_d[1:20, ]$word,
+     col = "lightgreen",
+     main = "Top 20 most frequent words",
+     ylab = "Word frequency"
+ )
>
```



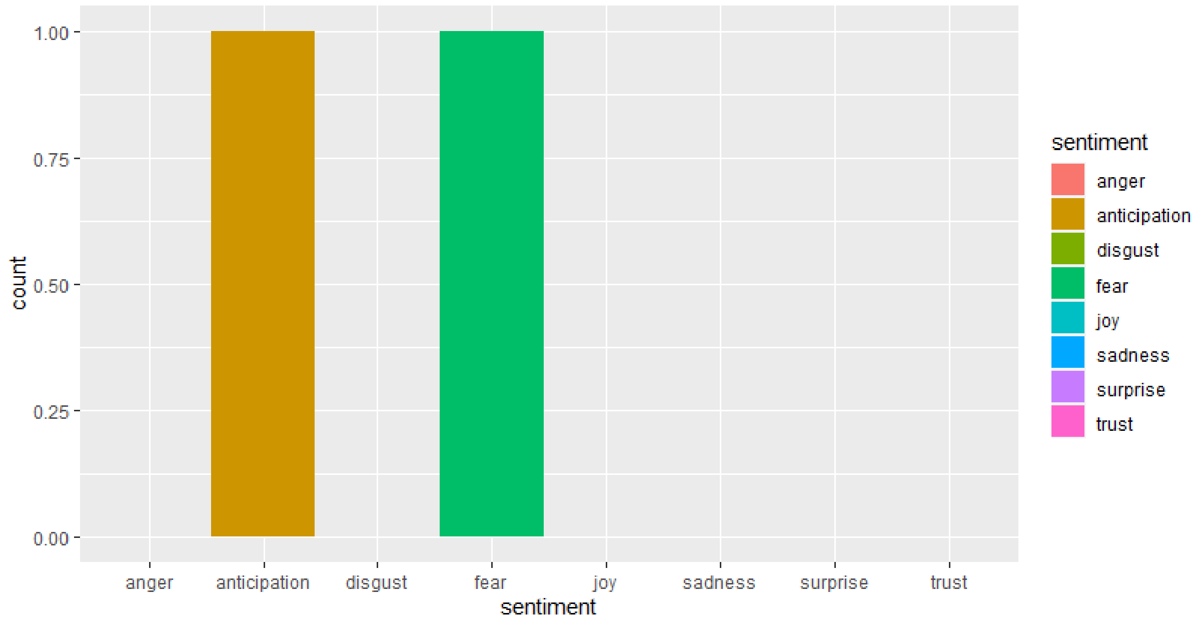


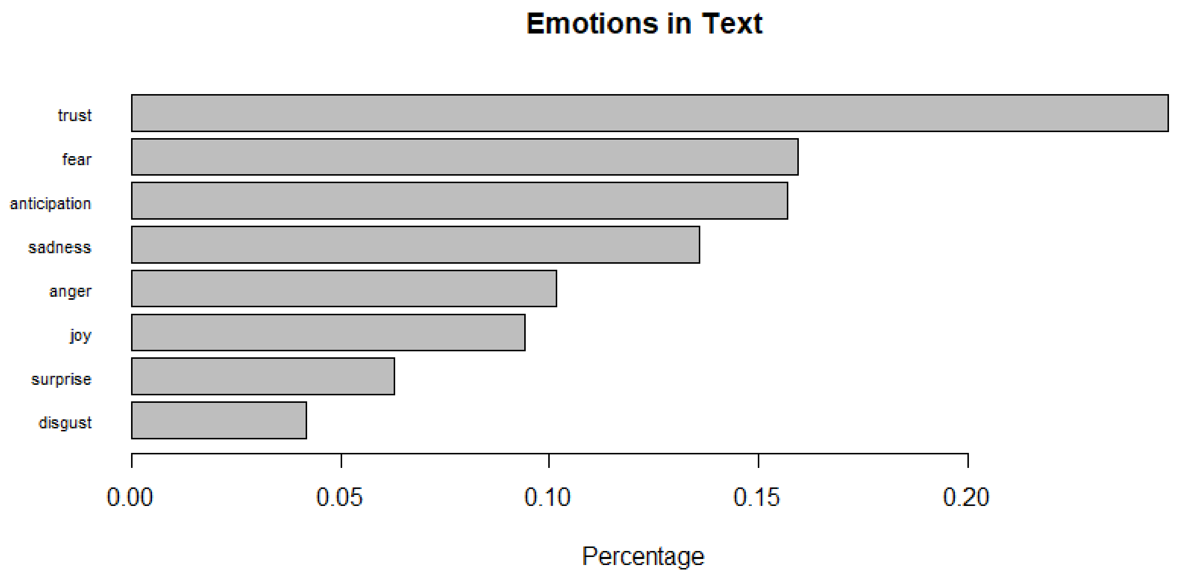
```
> ##### generate word cloud #####
> set.seed(1234)
> wordcloud(
+     words = dtm_d$word,
+     freq = dtm_d$freq,
+     scale = c(5, 0.5),
+     min.freq = 1,
+     max.words = 100,
+     random.order = FALSE,
+     rot.per = 0.40,
+     colors = brewer.pal(8, "Dark2")
+ )
```



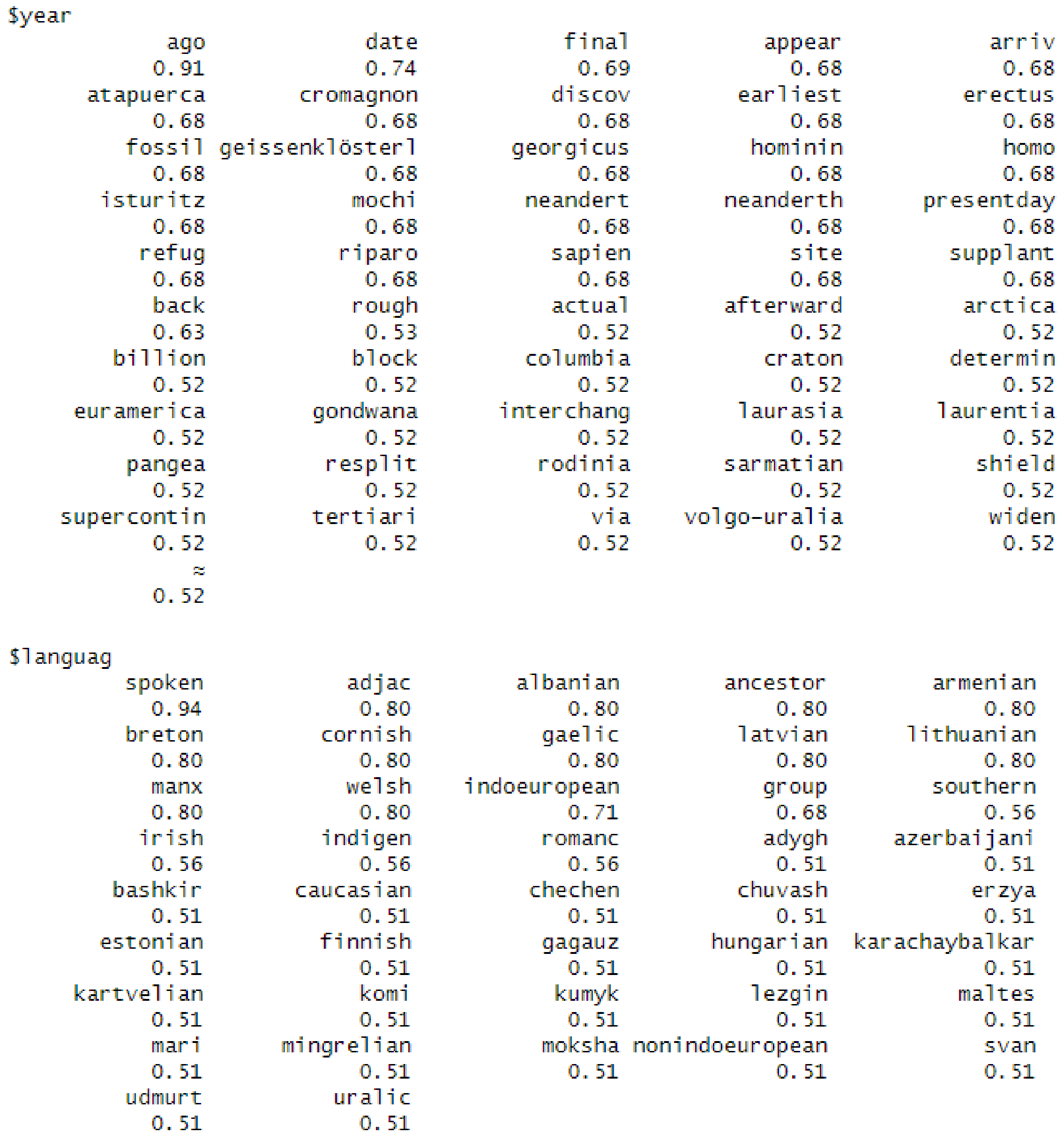


```
> #sum frequency of emotions for first 56 words
> td_new <- data.frame(rowSums(td[1:56]))
> #clear result
> names(td_new)[1] <- "count"
> td_new <- cbind("sentiment" = rownames(td_new), td_new)
> rownames(td_new) <- NULL
> td_new2 <- td_new[1:8,]
> head(d,10)
   anger anticipation disgust fear joy sadness surprise trust negative positive
1      0            0       0    0   0       0        0     0        0        0
2      0            0       0    0   0       0        0     0        0        0
3      0            0       0    0   0       0        0     0        0        0
4      0            0       0    0   0       0        0     0        0        0
5      0            0       0    0   0       0        0     0        0        0
6      0            0       0    0   0       0        0     0        0        0
7      0            0       0    1   0       0        0     0        1        0
8      0            0       0    0   0       0        0     0        0        0
9      0            0       0    0   0       0        0     0        0        0
10     0            0       0    0   0       0        0     0        0        0
> #transpose
```


count
1.00
0.75
0.50
0.25
0.00
anger
anticipation
disgust
fear
joy
sadness
surprise
trust
sentiment
sentiment
anger
anticipation
disgust
fear
joy
sadness
surprise
trust


Emotions in Text
trust
fear
anticipation
sadness
anger
joy
surprise
disgust
0.00
0.05
0.10
0.15
0.20
Percentage

Przykładowe skojarzenia:



```
$year
         ago           date          final         appear          arriv 
        0.91           0.74           0.69           0.68           0.68 
   atapuerca      cromagnon         discov       earliest        erectus 
        0.68           0.68           0.68           0.68           0.68 
      fossil geissenklösterl      georgicus        hominin           homo 
        0.68           0.68           0.68           0.68           0.68 
     isturitz          mochi        neandert      neanderth     presentday 
        0.68           0.68           0.68           0.68           0.68 
       refug         riparo         sapien           site       supplant 
        0.68           0.68           0.68           0.68           0.68 
        back          rough         actual      afterward        arctica 
        0.63           0.53           0.52           0.52           0.52 
     billion          block       columbia         craton       determin 
        0.52           0.52           0.52           0.52           0.52 
   euramerica       gondwana     interchang       laurasia      laurentia 
        0.52           0.52           0.52           0.52           0.52 
      pangea        resplit        rodinia      sarmatian         shield 
        0.52           0.52           0.52           0.52           0.52 
  supercontin        tertiari            via   volgo-uralia          widen 
        0.52           0.52           0.52           0.52           0.52 
           ≈ 
        0.52 

$languag
      spoken          adjac       albanian       ancestor       armenian 
        0.94           0.80           0.80           0.80           0.80 
      breton        cornish         gaelic        latvian     lithuanian 
        0.80           0.80           0.80           0.80           0.80 
        manx          welsh    indoeuropean          group       southern 
        0.80           0.80           0.71           0.68           0.56 
       irish         indigen         romanc          adygh    azerbaijani 
        0.56           0.56           0.56           0.51           0.51 
     bashkir      caucasian        chechen        chuvash          erzya 
        0.51           0.51           0.51           0.51           0.51 
    estonian        finnish         gagauz      hungarian   karachaybalkar 
        0.51           0.51           0.51           0.51           0.51 
   kartvelian           komi          kumyk         lezgin         maltes 
        0.51           0.51           0.51           0.51           0.51 
        mari     mingrelian         moksha nonindoeuropean          svan 
        0.51           0.51           0.51           0.51           0.51 
      udmurt         uralic 
        0.51           0.51 
```