Project 1 - Data Mining

Jakub Hinc

1 listopada 2021

Chosen files

I've chosen ten articles about coffee. There are all from other blogs and there are not related. Below we can see the titles of the files. We can predict that the result will be tightly bonded with brewing coffee.

```
##  [1] "6 Steps To Clean Your Chemex Easily. (Only 6 Steps).txt"
##  [2] "BEST NON-TOXIC COFFEE MAKERS (PLASTIC-FREE, TOXIN-FREE, MOLD FREE).txt"
##  [3] "Can You Use A Coffee Grinder As A Blender 4 Dos & 3 Don’ts.txt"
##  [4] "How Long Does Whole Bean Coffee Last Easy Tips For Better Preservation.txt"
##  [5] "Making a Delicious Cup of Coffee.txt"
##  [6] "Navy Coffee What Is It How To Make It & A Little Bit Of History.txt"
##  [7] "The 5 Best Ways to Make Coffee While Traveling.txt"
##  [8] "The History of Instant Coffee.txt"
##  [9] "What Type Of Coffee Is Used In A Coffee Maker Types & Examples.txt"
## [10] "Why Does Pour-Over Coffee Drip Too Fast Here’s Why And Easy Fixes.txt"
```

We can check how many words are in ours files by using function DocumentTermMatrix. It shows every unique word in all files.

```
docs <- Corpus(DirSource(wd))
DocumentTermMatrix(docs)
```

```
## <<DocumentTermMatrix (documents: 10, terms: 2043)>>
## Non-/sparse entries: 3403/17027
## Sparsity           : 83%
## Maximal term length: 30
## Weighting          : term frequency (tf)
```

As we can see we have 10 documents that contain 2043 unique words, it's the result before performing the Pre-processing step. Now we are going to clean our files and check how many words are left after pre-processing step.

Pre-processing

1. The pre-procesing step is very useful when we want to clean ours datas and get rid of useless elements in our text. First we remove punctuation, marks, numbers and special characters. In this step we use functions written below:

```
docs <- tm_map(docs,removePunctuation)
docs <- tm_map(docs, removeNumbers)
for (j in seq(docs)) {
  docs[[j]] <- gsub("/", " ", docs[[j]])
  docs[[j]] <- gsub("@", " ", docs[[j]])
  docs[[j]] <- gsub("–", " ", docs[[j]])
  docs[[j]] <- gsub("’", " ", docs[[j]])
  docs[[j]] <- gsub("“", " ", docs[[j]])
  docs[[j]] <- gsub("…", " ", docs[[j]])
  docs[[j]] <- gsub("‘", " ", docs[[j]])
  docs[[j]] <- gsub(")", " ", docs[[j]])
  docs[[j]] <- gsub("”", " ", docs[[j]])
}
```

2. After previous step we want to convert all words to lower case in order to normalize the text.

```
docs <- tm_map(docs, tolower)
```

3. The next step helps with removing common words from the text. Firstly we use included stopwords functions which contain a standard list of such words.

```
docs <- tm_map(docs, removeWords, stopwords("English"))
```

5. Then we remove all additional whitespaces:

```
docs <- tm_map(docs, stripWhitespace)
```

6. At the end of pre-processing we want to stem ours files which means reducing related words to their common root.

```
library(SnowballC)
for (j in seq(docs)) {
  docs[[j]]<-stemDocument(docs[[j]], language = "english")
}
```

Result of pre-processing

After all steps of pre-preparing, we have files that are clean and easier to draw any conclusions.

```
DocumentTermMatrix(docs)
```

```
## <<DocumentTermMatrix (documents: 10, terms: 1222)>>
## Non-/sparse entries: 2283/9937
## Sparsity           : 81%
## Maximal term length: 26
## Weighting          : term frequency (tf)
```

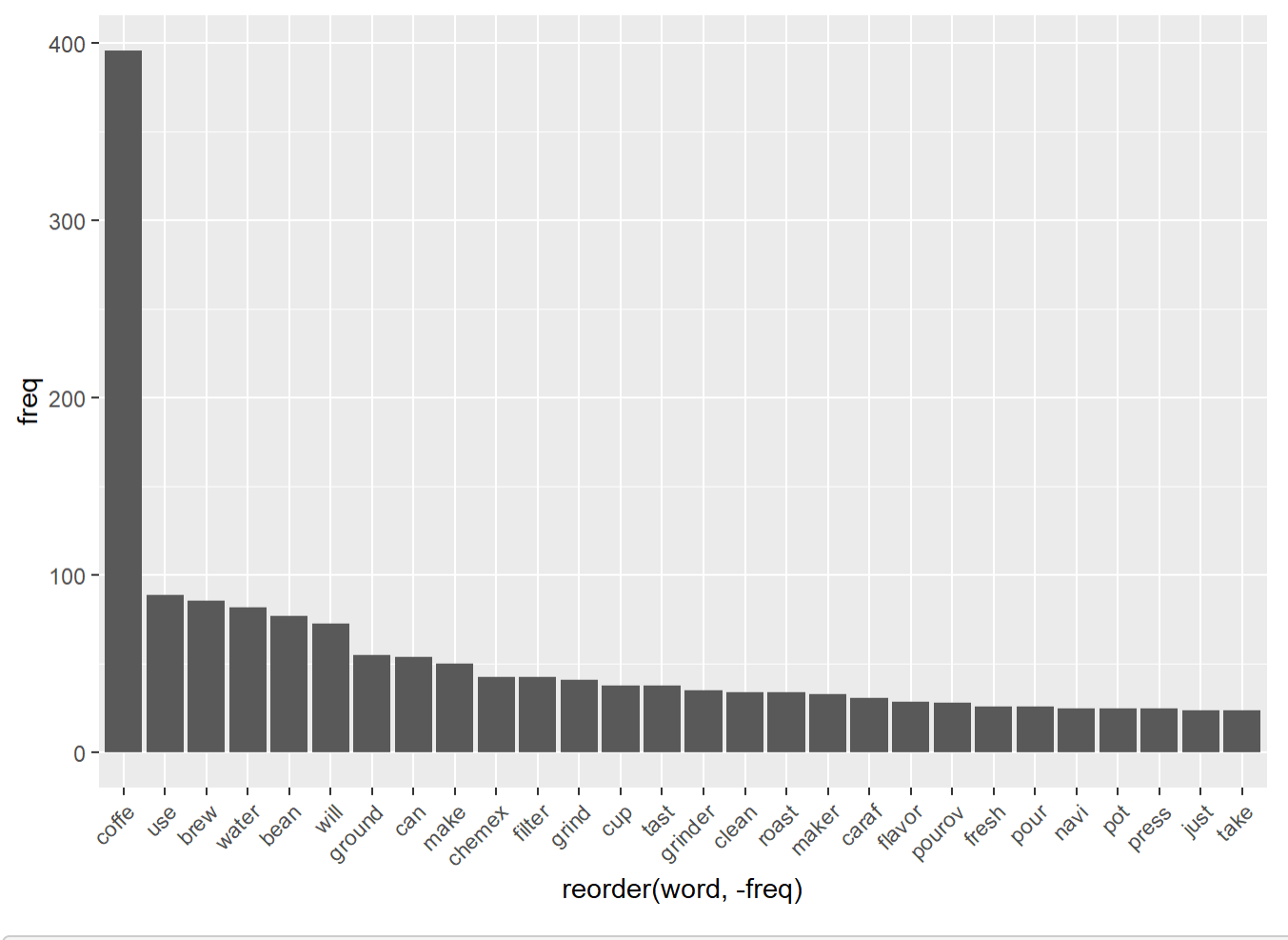
We can see that number of words decreased by almost about 800. It shows that many of the words from the text are useless and do not affect the analysis of the final result.

Wordcloud and Zipf's distribution

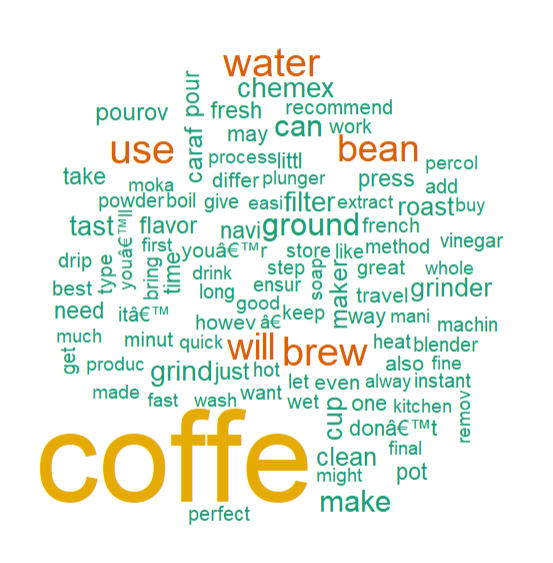
Now we are going to check to most repeated words and if there are any which are common and useless. To present our results we will use two other methods, first Zipf's distribution and second wordcloud.

```
dtmr <-DocumentTermMatrix(docs)
freqr <- colSums(as.matrix(dtmr))
freq <- sort(freqr, decreasing=TRUE)
mk<-min(head(freq, 30))
wf=data.frame(word=names(freq),freq=freq)
```

```
p <- ggplot(subset(wf, freq>mk), aes(x = reorder(word, -freq), y = freq))
p <- p + geom_bar(stat="identity")
p <- p + theme(axis.text.x=element_text(angle=45, hjust=1))
p
```




```
set.seed(142)
dark2 <- brewer.pal(6, "Dark2")
wordcloud(names(freq), freq, max.words=100, rot.per=0.2, colors=dark2)
```



As we can see from plots there are some words which are not giving us any information but there are frequently appear in ours files. For example: will, use, just. To remove them from text we will use file Stopwords which contains every useless word which we want to get rid of.

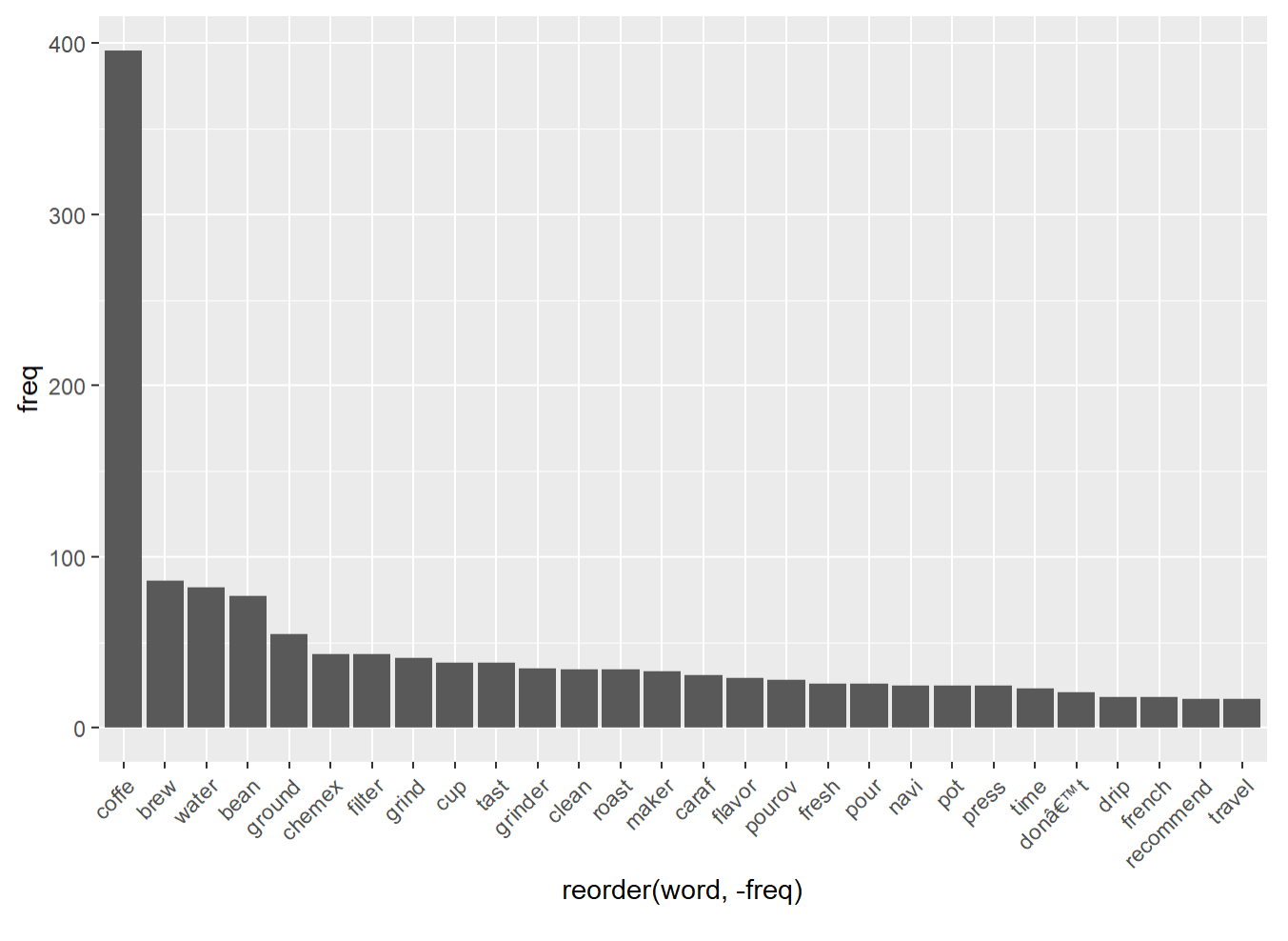
```
StWW<-as.character(StW$V1)
StWW
```

```
##  [1] "will"  "need"  "can"   "use"   "just"  "make"  "way"   "littl" "one"
## [10] "may"   "even"  "howev" "take"  "also"  "great" "best"
```

```
docs <- tm_map(docs, removeWords, StWW)
```

After this process, we have data without useless words, by using Zipf's distribution we can see that there left only words that could give us some pieces of information about researched texts.

```
p <- ggplot(subset(wf, freq>mk), aes(x = reorder(word, -freq), y = freq))
p <- p + geom_bar(stat="identity")
p <- p + theme(axis.text.x=element_text(angle=45, hjust=1))
p
```

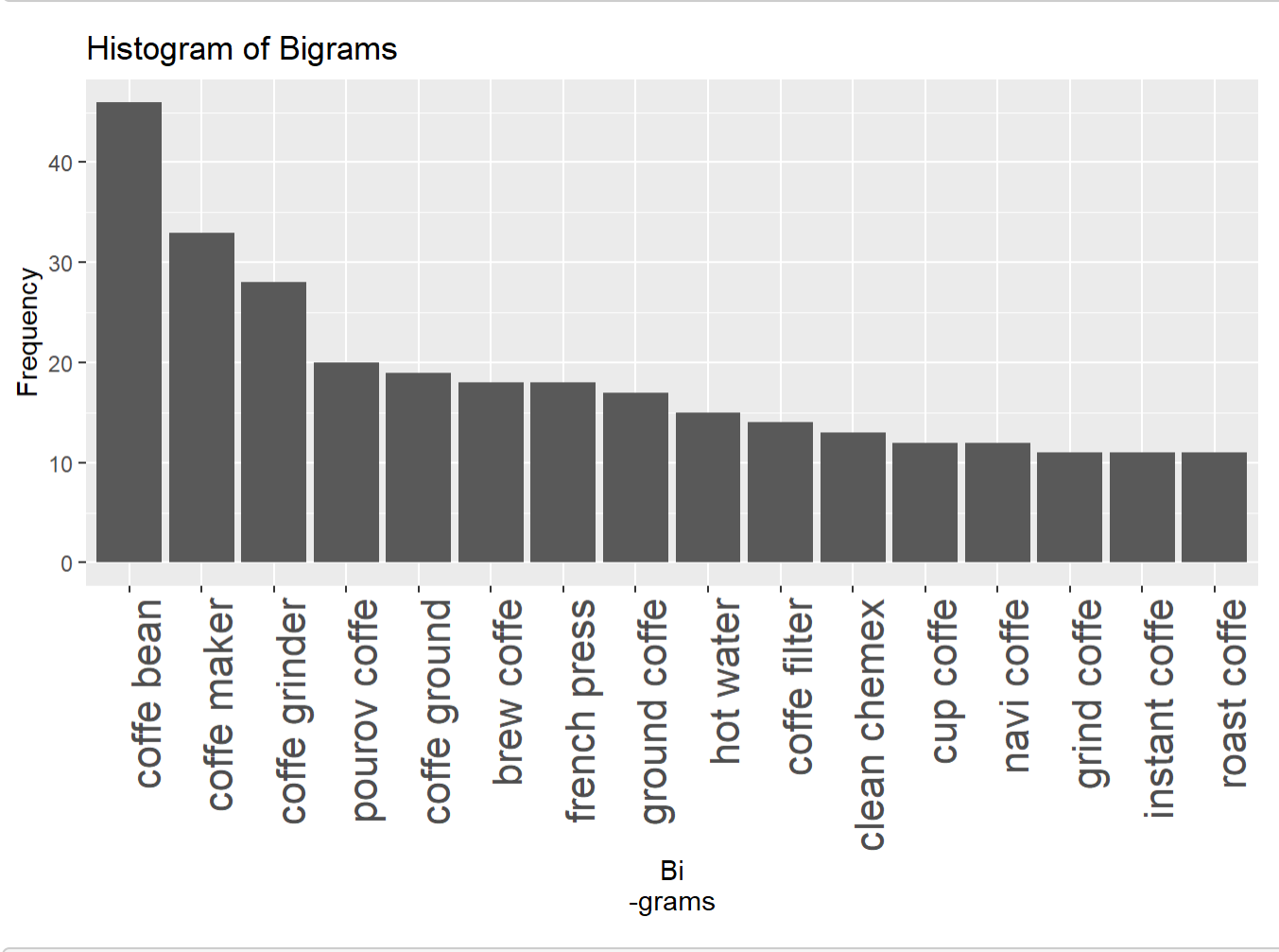


Associations

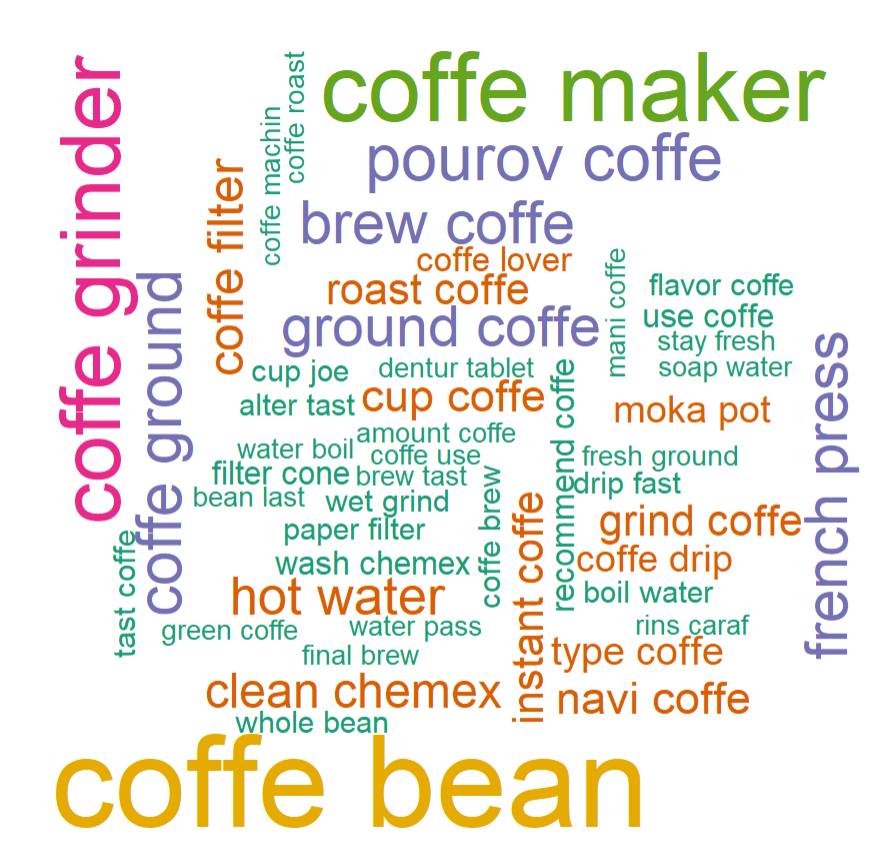
Now we can start looking for associations with the most significant words. To present the most noticeable correlations we can use once again wordcloud and Zipf's distribution. For this step we use functions written below:

Bigrams

```
wf=data.frame(word=names(freq_n),freq=freq_n)
p <- ggplot(subset(wf, freq>=mk), aes(x = reorder(word, -freq), y = freq))
p <- p + geom_bar(stat="identity")+ ggtitle("Histogram of Bigrams") +labs(x="Bi
-grams",y="Frequency")
p <- p + theme(axis.text.x=element_text(angle=90, hjust=1, size=16))
p
```

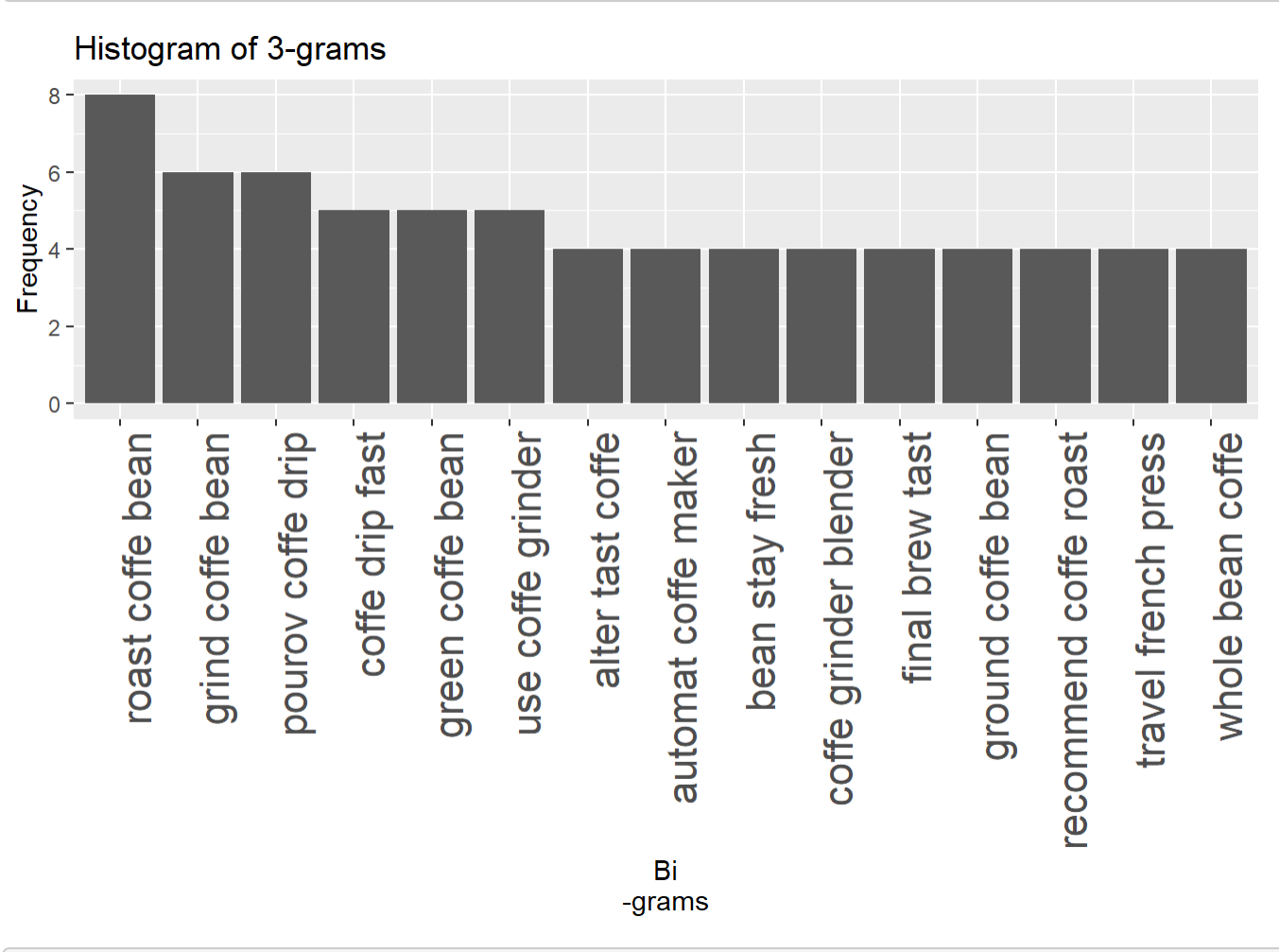



```
set.seed(142)
dark2 <- brewer.pal(6, "Dark2")
wordcloud(names(freq_n), freq_n, max.words=50, rot.per=0.2, colors=dark2)
```

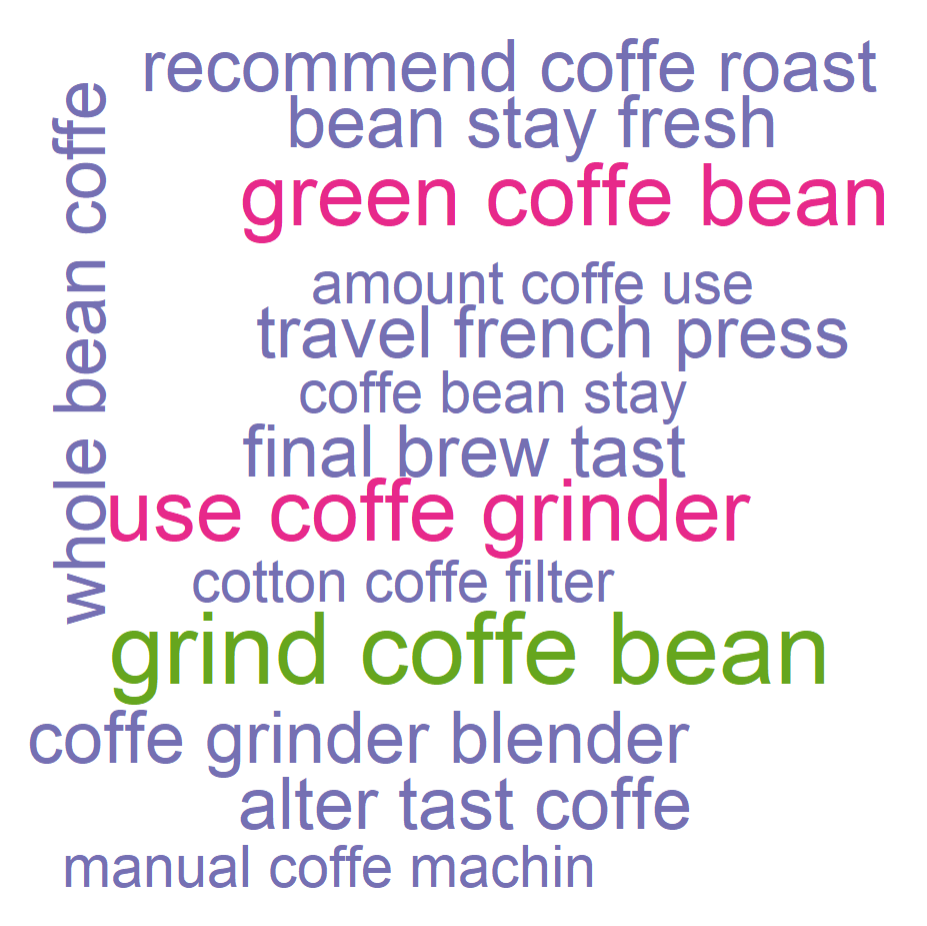


3-grams

```
wf=data.frame(word=names(freq_n2),freq=freq_n2)
p <- ggplot(subset(wf, freq>=mk), aes(x = reorder(word, -freq), y = freq))
p <- p + geom_bar(stat="identity")+ ggtitle("Histogram of 3-grams") +labs(x="Bi
-grams",y="Frequency")
p <- p + theme(axis.text.x=element_text(angle=90, hjust=1, size=16))
p
```




```
set.seed(142)
dark2 <- brewer.pal(6, "Dark2")
wordcloud(names(freq_n2), freq_n2, max.words=20, rot.per=0.2, colors=dark2)
```



Summarize associations

As we can see from previous examples there are some words and grams which can clearly well describe main topic. I've chosen some words, bigrams and 3-grams :

1. coffe
2. brew
3. bean
4. ground
5. coffe maker
6. coffe filter
7. french press
8. coffe grinder
9. roast coffee
10. green coffe bean
11. grind coffe bean
12. use coffee grinder
13. roast coffee bean

In my opinion, all of the written frazes are tightly collarated with brewing coffee. It's confirms the assumptions that I presented at the beginning of the report. Bigrams mostly present some stuff needed for brewing coffee like coffee maker, french press, or coffee filter. On the other hand, 3-grams concerned mostly on stages of making coffee: grind coffee bean, use a coffee grinder, or roast coffee bean. We can see that all phrases are easy to understand and there are together well presents the main topic.

Matrix sparsity changes

1. Removing rarely used words

```
dtmr1 <- DocumentTermMatrix(docs, control=list(bounds = list(global = c(2,Inf))))
dtmr1
```

```
## <<DocumentTermMatrix (documents: 10, terms: 440)>>
## Non-/sparse entries: 1390/3010
## Sparsity           : 68%
## Maximal term length: 11
## Weighting          : term frequency (tf)
```

2. Word length limitation

```
dtmr2 <- DocumentTermMatrix(docs, control=list(wordLengths=c(3, 20)))
dtmr2
```

```
## <<DocumentTermMatrix (documents: 10, terms: 1204)>>
## Non-/sparse entries: 2154/9886
## Sparsity           : 82%
## Maximal term length: 15
## Weighting          : term frequency (tf)
```

3. Automated sparsity reduction

```
dtmr3 <- removeSparseTerms(dtm, 0.70)
dtmr3
```

```
## <<DocumentTermMatrix (documents: 10, terms: 134)>>
## Non-/sparse entries: 678/662
## Sparsity           : 49%
## Maximal term length: 9
## Weighting          : term frequency (tf)
```

We can see that all procedures gives us more efficient matrix, especially automated sparsity reduction which help to reduce of sparsity to 47% and to 223 terms. Which helps us not to store so many useless data and irrelevant words. I think it is best opction to clean our datas from all 3 posibilities.

Normalized Matrix

```
doc_length <- as.data.frame(rowSums(as.matrix(dtm)))
max_length<-max(doc_length)
min_length<-min(doc_length)
aver_length<-mean(rowSums(as.matrix(dtm)))
```

```
max_length
```

```
## [1] 719
```

```
min_length
```

```
## [1] 171
```

```
aver_length
```

```
## [1] 462
```

We can see that there is big difference between max length and min. This points that there are difference in document length which can influence on ours analysis. To prevent that problem we can normalize our documents by function wrriten below:

```
nn<-rowSums(as.matrix(dtm))
dtm_Norm<-dtm/nn
```

As a result, we have normalized the Document-Term Matrix which is proven by values of max_length 1 and max_length 1 which are equal to 1.

```
max_length1
```

```
## [1] 1
```

```
min_length1
```

```
## [1] 1
```

```
aver_length1
```

```
## [1] 1
```

Business insights

As we can see from the report coffee blogs are tightly bonded with well brewing coffee process. Most frequent words such as flavor, fresh, taste (taste) show that for coffee fans taste of the prepared drinks is significant. Also, the way that they are going to prepare their coffee is very important. They like to experiment with various equipment, which we can see in Zipf's distribution: Chemex, french press, coffee maker, moka pot. They also like to use rarely seen stuff which I have never heard about, such as cotton coffee filter, wet grind, green coffee bean or rins carafe. I noticed that there is a bigram "detures cleaning" and I get interested so I googled it. I found out thanks to the report that detures tablets are helpful with mug cleaning after coffee what I would not expect in my life. It also showed me that coffee enthusiasts like it when their equipment is clean. This is also indicated by expressions such as: "clean Chemex", "wash Chemex", "soap water", "clean". It shows that coffee fans are focused on the way how to prepare coffee more than types and coffee producers. It indicates that if we want to get to know more about coffee we should first focus on the way we brew and after this what kind of coffee we buy. It shows that taste doesn't only depend on this how much we spent on our beans but how much time and effort we push to prepare ours cup of coffee.