**Topic 4**

**Visualization of frequencies**

The aim:

To learn how to make simple graphs based on corpus frequencies with the help of R.

The types of graphs:

1) bar charts

2) pie charts

3) word clouds

The data: the Universal Declaration of Human Rights, parsed with udpipe (see the previous topic).

We’ll also learn how to correct parsing mistakes.

**1. How to make a bar chart**

Bar charts are useful for visualization of many categories, e.g. for top 20 most frequent nouns. In order to select all lemmas with a particular part of speech, you need to choose the column upos and specify the required part of speech using **double equal sign**!

**freq\_nouns <- table(declaration\_ud$lemma[declaration\_ud$upos == "NOUN"])**

#to display the first 6 nouns with their frequencies, arranged alphabetically:

**head(freq\_nouns)**

access accordance account act activity

1 1 1 4 2

advancement

1

#to sort in descending order, the most frequent nouns first, and select the top 20 nouns:

**sort(freq\_nouns, decreasing = T)[1:20]**

right article freedom rights

34 30 14 11

one law protection country

9 7 7 6

education family religion act

5 5 5 4

declaration discrimination limitation nationality

4 4 4 4

offence person society state

4 4 4 4

There is an obvious parsing error. Let us find all rows with “rights” as a lemma:

**declaration\_ud[declaration\_ud$lemma== "rights",]**

#part of the output omitted

start end term\_id token\_id token lemma upos xpos

15 57 62 12 12 rights rights NOUN NNS

45 33 38 7 7 rights rights NOUN NNS

281 118 123 19 19 rights rights NOUN NNS

334 138 143 25 25 rights rights NOUN NNS

1038 261 266 46 46 rights rights NOUN NNS

1373 128 133 24 24 rights rights NOUN NNS

1514 71 76 13 13 rights rights NOUN NNS

1561 28 33 9 9 rights rights NOUN NNS

1590 196 201 38 38 rights rights NOUN NNS

1619 11 16 5 5 rights rights NOUN NNS

1678 188 193 37 37 rights rights NOUN NNS

We can now correct this mistake by creating a new version of the data declaration\_new and then replacing the lemma of *rights* with *right*.

**declaration\_new <- declaration\_ud**

**declaration\_new$lemma[declaration\_new$token == "rights"] <- "right"**

#let’s check the new version:

**declaration\_new[declaration\_new$token == "rights",]**

#part of the output is omitted

start end term\_id token\_id token lemma upos xpos

15 57 62 12 12 rights right NOUN NNS

45 33 38 7 7 rights right NOUN NNS

281 118 123 19 19 rights right NOUN NNS

334 138 143 25 25 rights right NOUN NNS

673 167 172 38 6 rights right NOUN NNS

1038 261 266 46 46 rights right NOUN NNS

1373 128 133 24 24 rights right NOUN NNS

1514 71 76 13 13 rights right NOUN NNS

1561 28 33 9 9 rights right NOUN NNS

1590 196 201 38 38 rights right NOUN NNS

1619 11 16 5 5 rights right NOUN NNS

1678 188 193 37 37 rights right NOUN NNS

Now everything seems to be correct.

Let’s create a new, corrected vector with noun frequency. Important: instead of declaration\_ud, we need to use declaration\_new, with our corrections:

**freq\_nouns <- table(declaration\_new$lemma[declaration\_new$upos == "NOUN"])**

**sort(freq\_nouns, decreasing = T)[1:20]**

right article freedom one

45 30 14 9

law protection country education

7 7 6 5

family religion act declaration

5 5 4 4

discrimination limitation nationality offence

4 4 4 4

person society state association

4 4 4 3

Are there any other issues? Of course, one is a pronoun, not a noun!

**Exercise**

Find in which examples *one* is annotated as a noun. Can you think of a reason why the parser made this decision?

To correct this, we can use the following code, combining two conditions: lemma ‘one’ and part of speech ‘NOUN’. The conditions are combined with the help of &:

**declaration\_new$upos[declaration\_new$lemma == "one"&declaration\_new$upos == "NOUN"] <- "PRON"**

Let’s create our frequency list again:

**freq\_nouns <- table(declaration\_new$lemma[declaration\_new$upos == "NOUN"])**

**sort(freq\_nouns, decreasing = T)[1:20]**

right article freedom law

45 30 14 7

protection country education family

7 6 5 5

religion act declaration discrimination

5 4 4 4

limitation nationality offence person

4 4 4 4

society state association basis

4 4 3 3

Now everything is correct. It makes sense to select 18 top nouns because the nouns with frequency 3 are displayed only partially (alphabetically, the algorithm selects association and basis). For convenience, we’ll create an object called top18, which will contain these frequencies and nouns:

**top18 <- sort(freq\_nouns, decreasing = TRUE)[1:18]**

**top18**

right article freedom law

45 30 14 7

protection country education family

7 6 5 5

religion act declaration discrimination

5 4 4 4

limitation nationality offence person

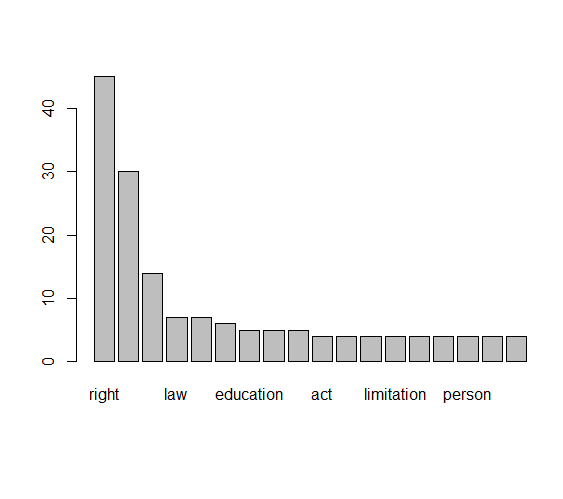
4 4 4 4

society state

4 4

Now we can easily make a first version of our bar plot:

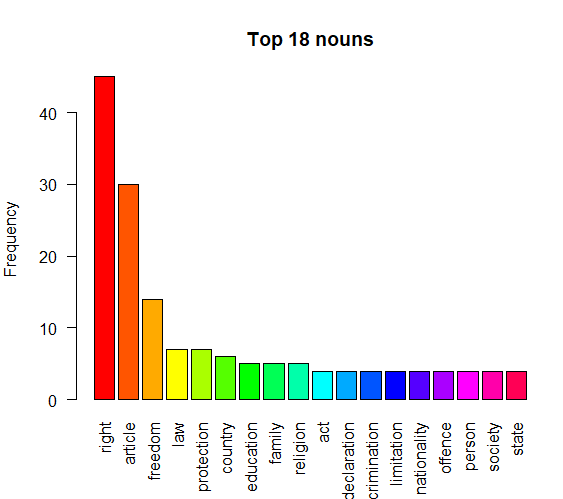
**barplot(top18)**



This doesn’t look very useful, unfortunately. Let us make an enhanced version:

**barplot(top18, las = 2, col = rainbow(18), main = "Top 18 nouns", ylab = "Frequency")**

The argument las specifies the orientation of the text labels (2 means orthogonal to the axis). The argument col specifies the colour. You can also use a specific colour, e.g. col = "blue". Next, main specifies the plot title, and ylab the label for the y-axis (i.e. the vertical axis). If you need to specify the label of the horizontal, x-axis, you can use the argument xlab, e.g. xlab = "Words".



There is an annoying issue: the text labels don’t fit. In order to correct that, we can change the margins, increasing the one at the bottom.

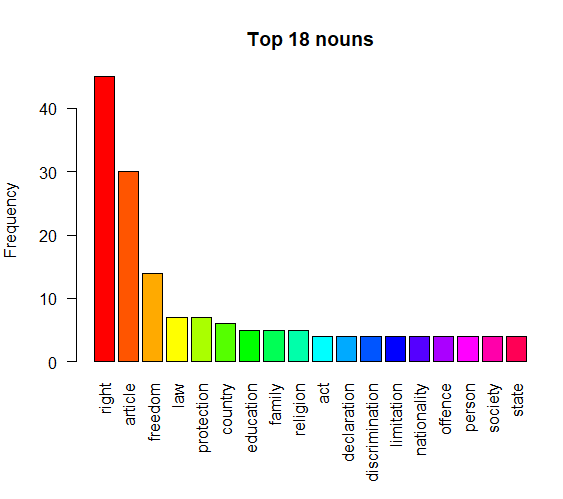
The default values are:

5.1 (bottom), 4.1 (left), 4.1 (top), 2.1 (right)

To increase the bottom value, we can type in the following:

**par(mar = c(7, 4, 4, 1))**

**barplot(top18, las = 2, col = rainbow(18), main = "Top 18 nouns", ylab = "Frequency")**



Important: how to save the plot? In RStudio, go to the Plots panel, click on *Export > Save as Image*, then choose the name, format and size and click on “Save”. The file will be in your working directory.

**Exercise**

Make a bar plot with top most frequent verbs. How many verbs would you like to plot?

**2. How to make a wordcloud**

If you want to visualize more words, it is convenient to use wordclouds. This is how you can create a wordcloud for top 40 nouns.

**library(wordcloud)** #you need to install it first!

**top40 <- sort(freq\_nouns, decreasing = TRUE)[1:40]**

**wordcloud(words = names(top40), freq = top40, colors = rainbow(40), random.color = TRUE)**



**Exercise**

Make a word cloud with another part of speech. Choose wisely the number of words.

Can you make sense of the frequencies?

**3. How to make a pie chart**

When there are few items, it can be useful to represent their frequencies with the help of a pie chart. Let us focus on the parts of speech.

**pos\_freq <- table(declaration\_new$upos)**

**pos\_freq**

ADJ ADP ADV AUX CCONJ DET NOUN NUM PART PRON

133 212 29 82 113 164 389 31 33 88

PROPN PUNCT SCONJ VERB X

8 230 9 131 32

Let us focus on the main notional categories: adjectives, adverbs, nouns and verbs, and create a vector with their frequencies manually:

**pos\_freq\_main <- c(133, 29, 389, 131)**

[1] 133 29 389 131

**pos\_freq\_main**

Next, we’ll add the text labels:

**names(pos\_freq\_main) <- c("Adjective", "Adverb", "Noun", "Verb")**

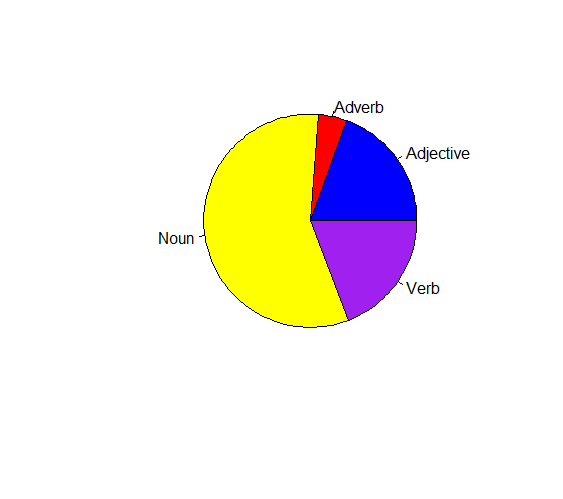
**pos\_freq\_main**

Adjective Adverb Noun Verb

133 29 389 131

Now we can create a pie chart. We will use manually defined colours:

**pie(pos\_freq\_main, col = c("blue", "red", "yellow", "purple"))**



**Exercise**

Make a pie chart with arguments (subjects, objects and obliques). Tip: check the column called *dep\_rel*. How can you interpret the results?

**Practice**

Create your own small corpus. Parse it with udpipe. Make three graphs representing frequency lists. Interpret the results.