Scaling methods

13 May 2019

This document gives some examples of how to apply scaling methods (Wordscores and Wordfish) in R. For theses example, we use the (English) speeches of EP group leaders that are part of the EUSpeech dataset. NB: Use setwd() to set the working directory to the folder that contains EP speeches in the file speeches_ep.csv.

```
Sys.setlocale(locale = "en_US.UTF-8")
```

```
## [1] "en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8"
```

```
#load libraries

library(readtext)
library(quanteda)
```

```
## Warning: package 'quanteda' was built under R version 3.5.2
```

```
library(stringr)
library(ggplot2)

#read in the EP speeches
speeches <- read.csv(file = "speeches_ep.csv",
                     header = TRUE,
                     stringsAsFactors = FALSE,
                     sep = ",",
                     encoding = "UTF-8")

#let's do a bit of manual cleaning to remove some boiler plate terms
speeches$text <- str_replace_all(speeches$text, "ladies and gentlemen", " ")
speeches$text <- str_replace_all(speeches$text, "President", " ")
speeches$text <- str_replace_all(speeches$text, "Mr", " ")
speeches$text <- str_replace_all(speeches$text, "Council", " ")
speeches$text <- str_replace_all(speeches$text, "Commission", " ")

#take a look at how many unique speakers there are in the dataset
unique(speeches$speaker)
```

```
##  [1] "Lothar Bisky"          "Martin Callanan"       "Daniel Cohn-Bendit"
##  [4] "Brian Crowley"         "Joseph Daul"           "Marcel de Graaff"
##  [7] "Nigel Farage"          "Monica Frassoni"       "Rebecca Harms"
## [10] "Syed Kamall"           "Michal Kaminski"       "Philippe Lamberts"
## [13] "Cristiana Muscardini" "Martin Schulz"         "Kathy Sinnott"
## [16] "Francesco Speroni"     "Hannes Swoboda"        "Guy Verhofstadt"
## [19] "Graham Watson"         "Francis Wurtz"         "Jan Zahradil"
## [22] "Gabriele Zimmer"
```

Let's first append the speeches for each speaker to each other using the dplyr library. If you don't have dplyr installed, do so using the install.packages() function.

```
library(dplyr)

#the `%>%` command is the pipe function and helps us with a chain of functions
#think of it as `then`:
#take the speeches dataframe, then
```

```
#group by variable, then
#paste speeches together.

speeches <- speeches %>%
  group_by(speaker) %>%
  summarise(text = paste(text, collapse = " ")) %>%
  ungroup()

#confirm that you have a total of 22 (very long) concatenated speeches, 1 for each EP speaker
dim(speeches)
```

```
## [1] 22  2
```

```
#construct a corpus from the concatenated speeches
corpus <- corpus(speeches)
```

Wordscores and Wordfish take in a dfm object as input, so first we will need to turn the speeches into a dfm:

```
speeches <- corpus(speeches)

#create a dfm
speeches.dfm <- dfm(speeches, stem = FALSE,
                    remove=stopwords("english"),
                    remove_punct=TRUE, ngrams = 1,
                    remove_numbers = TRUE)

#include only thoses features that occur in at least 5 documents
speeches.dfm <- dfm_trim(speeches.dfm, min_docfreq = 5)

#chech the number of documents and features
dim(speeches.dfm)
```

```
## [1]   22 5256
```

```
#change the document names to the speaker names
docnames(speeches.dfm) <- docvars(speeches.dfm, "speaker")
```

Wordscores

Let's see if we can use Wordscores to locate these 22 speakers on a pro-anti EU dimension. We'll first need to determine reference texts to anchor this dimension. On the anti-EU side we'll locate Nigel Farage, for obvious reasons, and on the pro-EU dimension we'll locate Guy Verhofstadt, leader of the liberal ALDE group, and a pro-EU voice:

```
#append an empty reference.score variable to the speeches.dfm data.frame
docvars(speeches.dfm, "reference.score") <- NA

#locate which rows correspond with Guy Verhofstadt (pro.eu) and Nigel Farage (anti.eu)
pro.eu <- which(docvars(speeches.dfm) == "Guy Verhofstadt")
anti.eu <- which(docvars(speeches.dfm) == "Nigel Farage")

#assign reference scores to Guy Verhofstadt (1) and Nigel Farage (1)
docvars(speeches.dfm, "reference.score")[pro.eu] <- 1
docvars(speeches.dfm, "reference.score")[anti.eu] <- -1
```

```
#inspects the reference.score variable:
docvars(speeches.dfm, "reference.score")
```

```
##  [1] NA NA NA NA NA NA NA  1 NA NA NA NA NA NA NA NA NA NA -1 NA NA NA
```

```
#implement wordscores as per Laver, Benoit, Garry (2003)
speeches.wordscores <- textmodel_wordscores(speeches.dfm,
                                            y = docvars(speeches.dfm, "reference.score"),
                                            scale = c("linear"),
                                            smooth = 0)
summary(speeches.wordscores, 10)
```

```
##
## Call:
## textmodel_wordscores.dfm(x = speeches.dfm, y = docvars(speeches.dfm,
##     "reference.score"), scale = c("linear"), smooth = 0)
##
## Reference Document Statistics:
##                       score total min max     mean median
## Brian Crowley            NA 13975   0 190  2.65887      0
## Cristiana Muscardini     NA  7946   0  85  1.51180      0
## Daniel Cohn-Bendit       NA 18299   0 282  3.48154      1
## Francesco Speroni        NA 15127   0 210  2.87804      1
## Francis Wurtz            NA 10551   0 190  2.00742      1
## Gabriele Zimmer          NA  5455   0 132  1.03786      0
## Graham Watson            NA 15366   0 180  2.92352      1
## Guy Verhofstadt           1 42084   0 801  8.00685      1
## Hannes Swoboda           NA 33597   0 445  6.39212      1
## Jan Zahradil             NA  5452   0 125  1.03729      0
## Joseph Daul              NA 39203   0 805  7.45871      2
## Kathy Sinnott            NA 12727   0 154  2.42142      1
## Lothar Bisky             NA  9360   0 135  1.78082      0
## Marcel de Graaff         NA    57   0   2  0.01084      0
## Martin Callanan          NA 15155   0 176  2.88337      1
## Martin Schulz            NA 53648   0 909 10.20700      3
## Michal Kaminski          NA 18948   0 340  3.60502      1
## Monica Frassoni          NA 10356   0 160  1.97032      0
## Nigel Farage             -1 23140   0 327  4.40259      1
## Philippe Lamberts        NA   481   0  10  0.09151      0
## Rebecca Harms            NA 18199   0 291  3.46252      1
## Syed Kamall              NA  6393   0  99  1.21632      0
##
## Wordscores:
## (showing first 10 elements)
##  in-office    meeting       take      place       next      month
##    0.04749    0.58754    0.02120   -0.49543   -0.10714   -0.19614
## particular      focus   economic    affairs
##   -0.38900    0.24516    0.30963    0.85043
```

```
#sort most discriminant words:

#pro-EU
head(sort(speeches.wordscores$wordscores), 10)
```

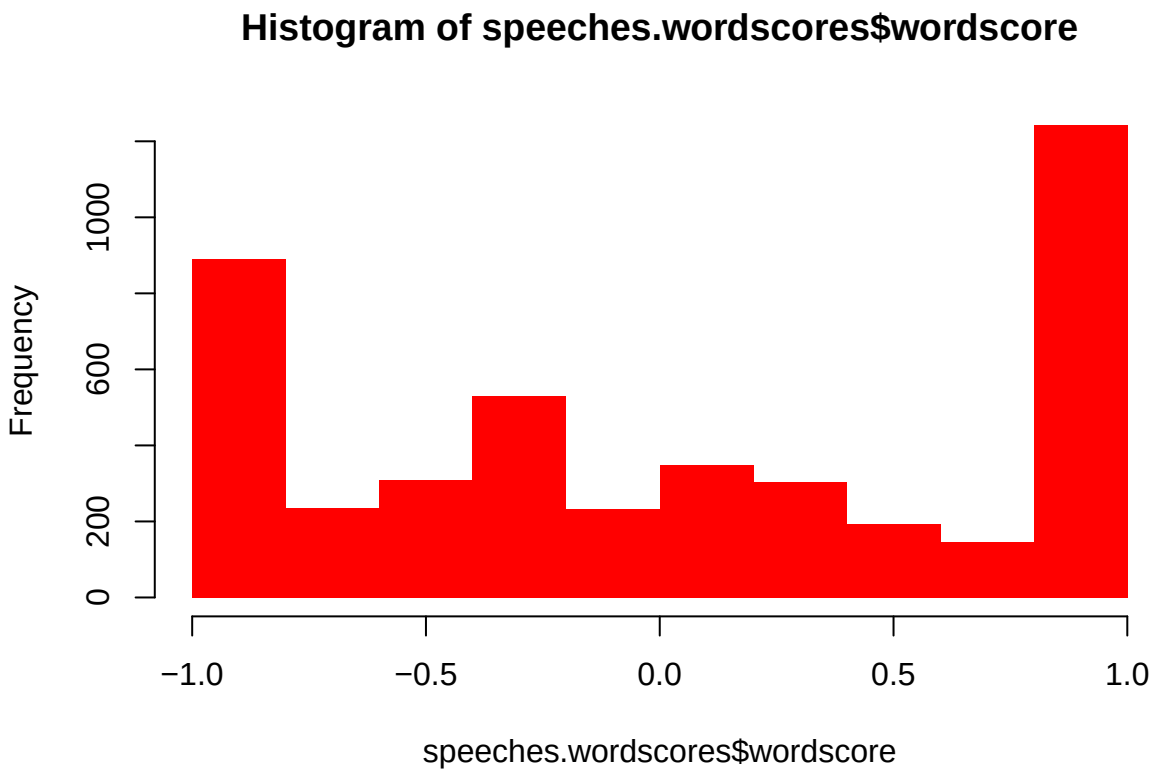
```
##        claim collectively        bites        lying   employment
```

```
##           -1           -1           -1           -1           -1
##       wisdom     strength      worried      suppose     lifelong
##           -1           -1           -1           -1           -1
```

```
#anti-EU
tail(sort(speeches.wordscores$wordscores), 10)
```

```
##   registration     weaknesses      divisions       receives      religions
##              1              1              1              1              1
##  announcements    unification           wife      dedicated anti-europeans
##              1              1              1              1              1
```

```
#histogram of wordscores
hist(speeches.wordscores$wordscore, col = "red", border = 0)
```



How would you interpret this histogram? And why do we see these peaks at -1 and at 1? How would you interpret the pro-EU and anti-EU discriminant words. Do they make sense?

Let's use the Wordscores model to predict the document scores of the speeches of the 20 remaining group leaders

```
speeches.wordscores.predict <- predict(speeches.wordscores,
                                       newdata = speeches.dfm)
```

```
## Warning: 821 features in newdata not used in prediction.
```

```
#which speakers are most like Farage
speeches.wordscores.predict[order(speeches.wordscores.predict, decreasing = FALSE)][1:5]
```

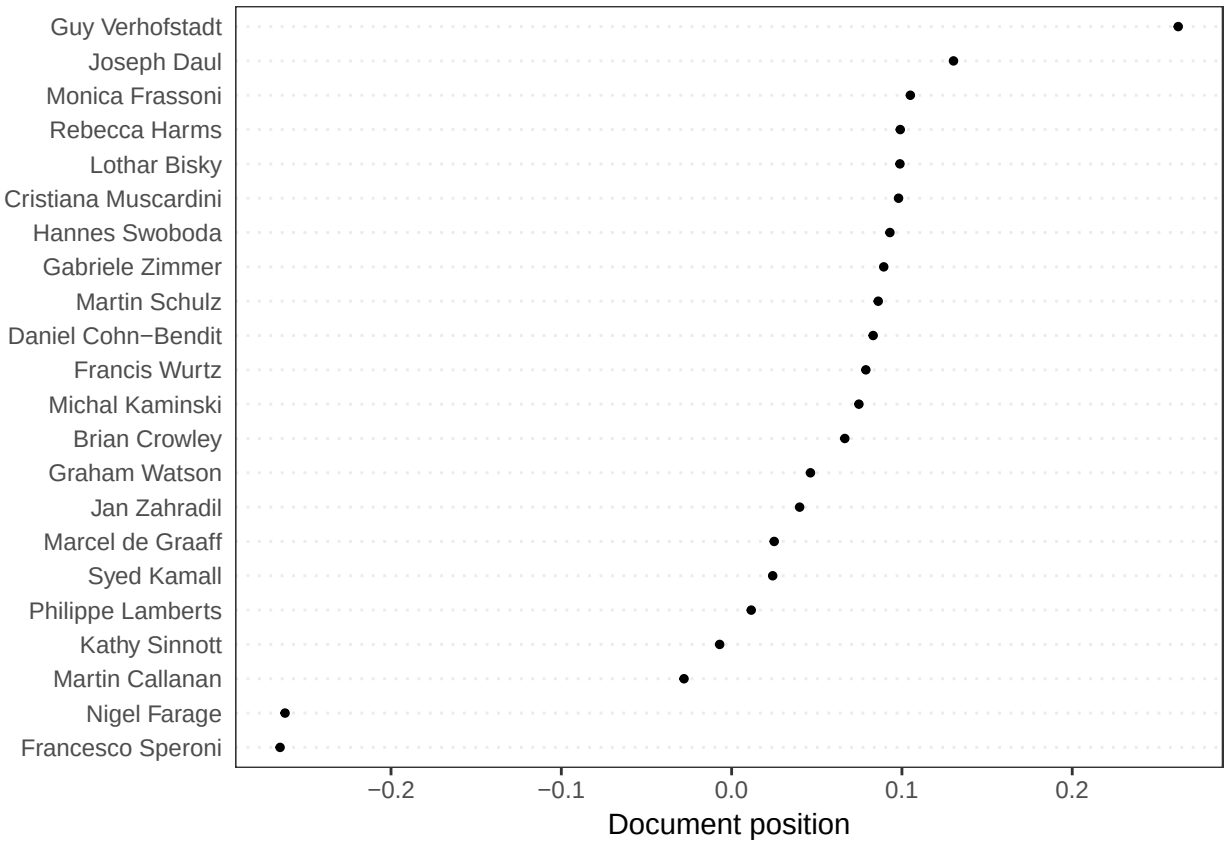
```
## Francesco Speroni      Nigel Farage   Martin Callanan     Kathy Sinnott
##      -0.265144131      -0.262232517      -0.027960357      -0.007029817
## Philippe Lamberts
##       0.011512981
```

```
#which speakers are most like Verhofstadt
speeches.wordscores.predict[order(speeches.wordscores.predict, decreasing = TRUE)][1:5]
```

```
## Guy Verhofstadt     Joseph Daul Monica Frassoni   Rebecca Harms
##      0.26223252      0.13028775      0.10495276      0.09901755
##    Lothar Bisky
##      0.09883396
```

Visualize the document scores in a plot:

```
#standard plot in quanteda
textplot_scale1d(speeches.wordscores.predict)
```



How would you interpret this outcome?

Wordfish

Estimate a Wordfish model and inspect its output:

```
speeches.wordfish <- textmodel_wordfish(speeches.dfm)
summary(speeches.wordfish)
```

```
##
## Call:
## textmodel_wordfish.dfm(x = speeches.dfm)
##
## Estimated Document Positions:
##                          theta       se
```

```
## Brian Crowley         0.61505 0.016398
## Cristiana Muscardini  1.08069 0.017022
## Daniel Cohn-Bendit   -0.26211 0.017163
## Francesco Speroni    -2.59448 0.019159
## Francis Wurtz         0.63663 0.018727
## Gabriele Zimmer       0.74145 0.024987
## Graham Watson         0.44876 0.016463
## Guy Verhofstadt      -0.45547 0.011555
## Hannes Swoboda        0.22798 0.011730
## Jan Zahradil         -0.07229 0.030657
## Joseph Daul           0.74680 0.009302
## Kathy Sinnott         1.00083 0.014232
## Lothar Bisky          0.49966 0.020790
## Marcel de Graaff     -0.40650 0.298255
## Martin Callanan      -0.78979 0.019762
## Martin Schulz         0.30504 0.009131
## Michal Kaminski       1.29464 0.009132
## Monica Frassoni       0.51741 0.019662
## Nigel Farage         -2.35874 0.015793
## Philippe Lamberts    -0.82523 0.110502
## Rebecca Harms         0.20678 0.016005
## Syed Kamall          -0.55710 0.029908
##
## Estimated Feature Scores:
##      in-office meeting   take    place    next   month particular  focus
## beta    0.7637  0.1355 -0.069 -0.05518 -0.2014 -0.3243      0.406 0.4658
## psi     2.2081  2.2719  4.022  3.12380  3.1185  1.1368      2.801 1.7250
##      economic affairs european   union proposals   must created  return
## beta    0.015  0.2857  0.05151 -0.1069    0.2145 0.2879  0.1909 0.07925
## psi     3.998  2.1943  5.55259  4.7402    2.4699 4.6706  1.4412 1.77115
##      economy previous  state implemented immediately restart    today
## beta  0.2297  0.07284 -0.147      0.4581      0.1584   0.184 -0.05661
## psi   2.8379  1.29870  3.545      1.1055      1.4039  -1.166  4.04644
##      speak future   europe  people   claim     care    much
## beta 0.264 0.2179 0.008845 -0.2433 -0.1681 -0.03262 -0.1538
## psi  2.344 3.4114 5.038114  4.6319  0.7497  1.31618  3.3770
```

```
#generate a dataframe with word level parameters beta and psi
wordfish.word.data <- data.frame(beta = speeches.wordfish$beta,
                           psi = speeches.wordfish$psi,
                           features = speeches.wordfish$features)

dim(wordfish.word.data)
```

```
## [1] 5256    3
```

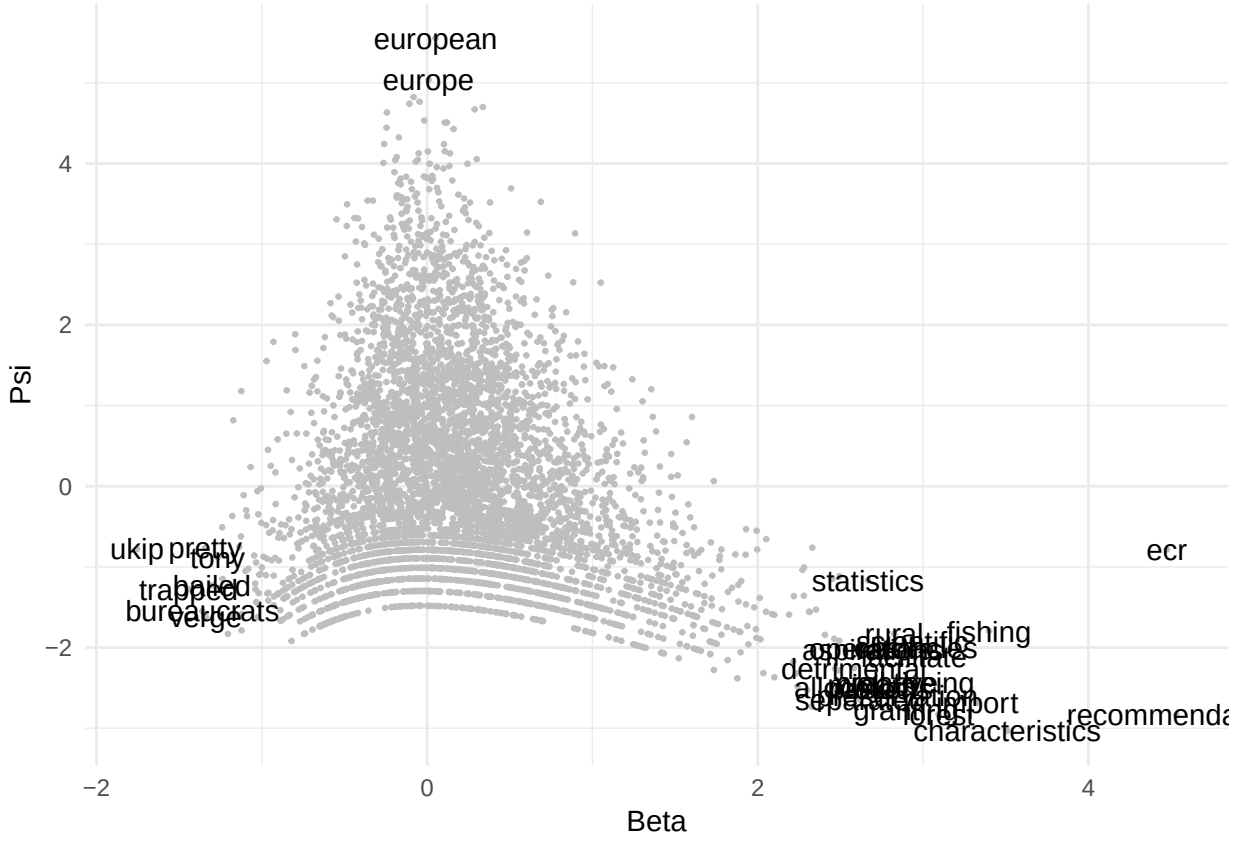
```
head(wordfish.word.data)
```

```
##          beta      psi  features
## 1  0.76368331 2.208140 in-office
## 2  0.13553638 2.271861   meeting
## 3 -0.06899931 4.021915      take
## 4 -0.05518008 3.123804     place
## 5 -0.20137518 3.118454      next
## 6 -0.32429683 1.136820     month
```

```
word.plot <- ggplot(data = wordfish.word.data, aes(x = beta, y = psi))
word.plot <- word.plot +
  geom_point(size = .5, color = "grey") +
  labs(x = "Beta", y = "Psi") + guides(size = "none", color = guide_legend("")) +
  theme_minimal()

word.plot <- word.plot +
  geom_text(data=subset(wordfish.word.data, beta > 2.5 | beta < -1.25 | psi > 5),
            aes(x = beta, y = psi, label = features))

print(word.plot)
```



Question: How would you interpret the word plot?

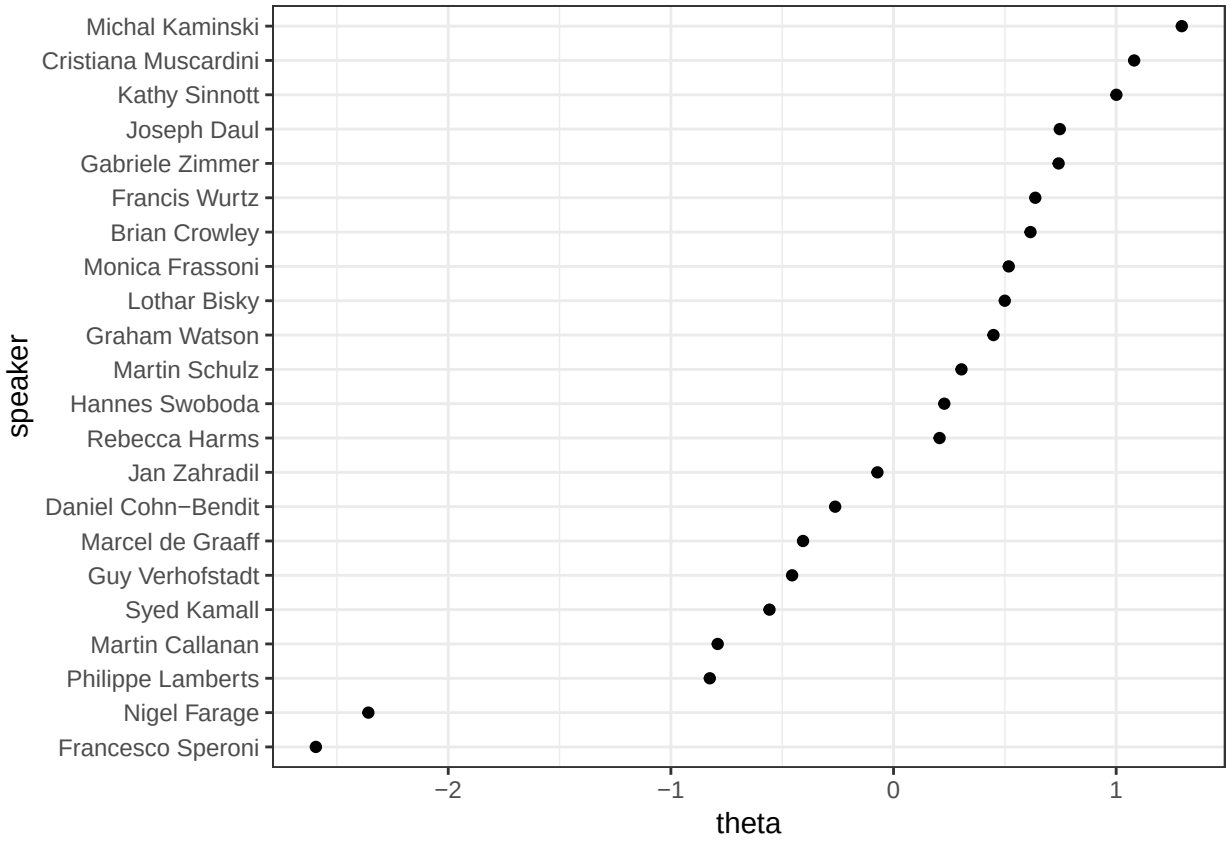
Plot the document positions generated by Wordfish:

```
#generate a dataframe with document level alpha beta and omega
wordfish.document.data <- data.frame(alpha = speeches.wordfish$alpha,
                                     theta = speeches.wordfish$theta,
                                     speaker = speeches.wordfish$docs)

#order the speaker factor by theta
wordfish.document.data $speaker <- reorder(wordfish.document.data$speaker,
                                           wordfish.document.data$theta)


#plot wordfish results using ggplot2
wordfish.plot <- ggplot(wordfish.document.data,
```

```
                          aes(x= theta, y = speaker))
wordfish.plot <- wordfish.plot + geom_point() + theme_bw()
print(wordfish.plot)
```

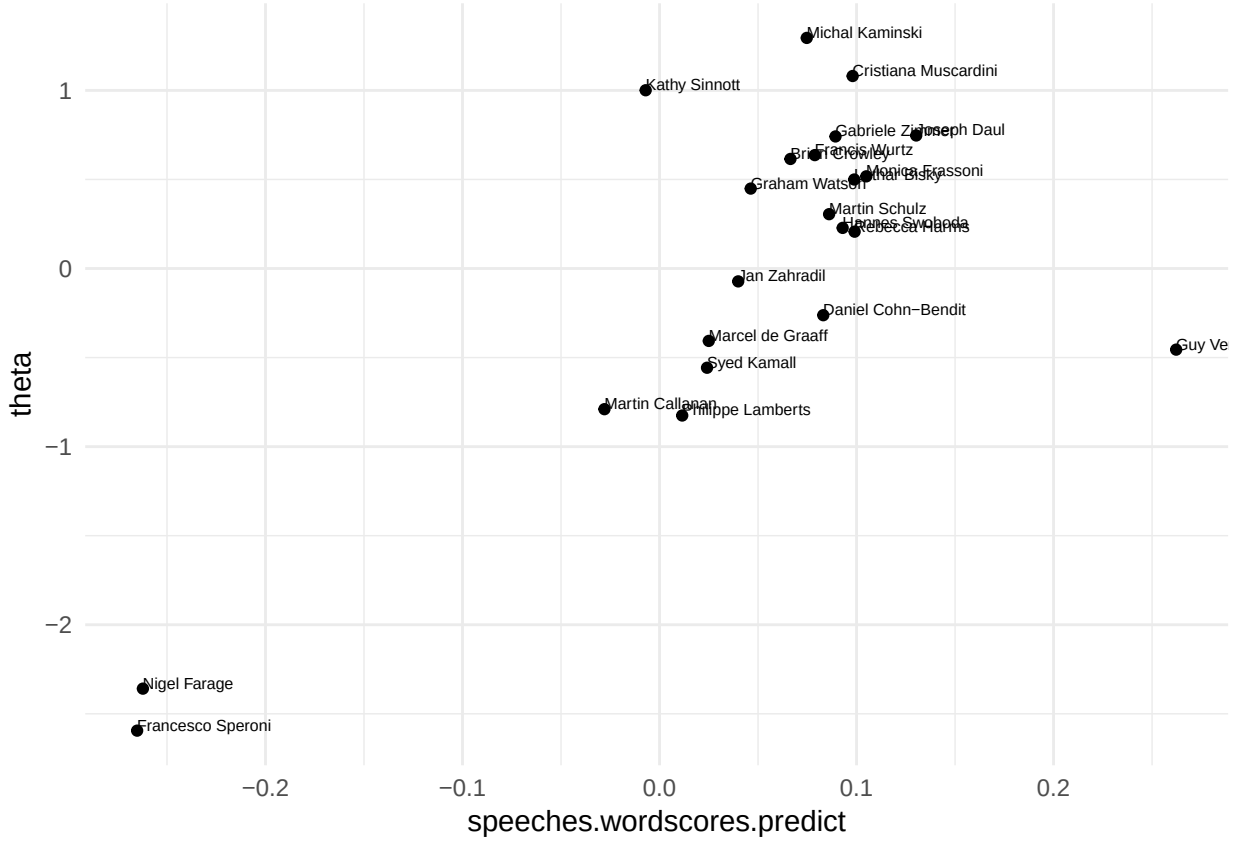


Both Wordscores and Wordfish are scaling models and estimated on the same text data they should lead to similar results. Let's see if this indeed the case.

```
scaling.data <- rbind(data.frame(speeches.wordscores.predict, wordfish.document.data))

scaling.plot <- ggplot(scaling.data, aes(x = speeches.wordscores.predict,
                                         y = theta,
                                         label = speaker))

scaling.plot <- scaling.plot + geom_point() + theme_minimal() + geom_text(aes(label=speaker),
                                                                          hjust=0,
                                                                          vjust=0,
                                                                          size = 2)
print(scaling.plot)
```




```
correlation <- cor.test(x=scaling.data$speeches.wordscores.predict,
                        y=scaling.data$theta,
                        method = 'pearson')
print(correlation)
```

```
##
##  Pearson's product-moment correlation
##
## data:  scaling.data$speeches.wordscores.predict and scaling.data$theta
## t = 4.9746, df = 20, p-value = 7.286e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4691577 0.8871289
## sample estimates:
##      cor
## 0.743668
```

Question: How would you interpret this correlation?

Question

Note that Wordscores are calculated from their relative occurrences in the reference texts. We currently have two reference texts (Verhofstadt and Farage). Change the code to add Francesco Speroni, an Italian anti-EU group leader, as a third reference text and run the code. How does it alter the results, if at all?