tutorial_07

R Markdown

Reading in our files and loading required packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(LDAvis)
library(readr)
library(tm)
```

```
## Loading required package: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(lda)

filelist<-list.files(path="../data/", pattern = ".*.txt",full.names = TRUE)
data<-lapply(filelist,FUN=read.delim)


data <- gsub("'", "", data)  # remove apostrophes
data <- gsub("[[:punct:]]", " ", data)  # replace punctuation with space
data <- gsub("[[:cntrl:]]", " ", data)  # replace control characters with space
data <- gsub("^[[:space:]]+", "", data) # remove whitespace at beginning of documents
data <- gsub("[[:space:]]+$", "", data) # remove whitespace at end of documents
data <- tolower(data)  # force to lowercase
stop_words <- stopwords("SMART")
```

```r
pp_rev <- data %>%
str_replace_all("’", "") %>%
str_replace_all("[[:punct:][:cntrl:]]", " ") %>%
str_trim %>%
str_to_lower()

# tokenize on space and output as a list:
doc.list <- str_split(pp_rev, "[[:space:]]+")
# compute the table of terms:
term.table <- table(unlist(doc.list))
term.table <- sort(term.table, decreasing = TRUE)
# remove terms that are stop words or occur fewer than 5 times:
del <- names(term.table) %in% stop_words | term.table < 5
term.table <- term.table[!del]
vocab <- names(term.table)
# now put the documents into the format required by the lda package:
get.terms <- function(x) {
index <- match(x, vocab)
index <- index[!is.na(index)]
rbind(as.integer(index - 1), as.integer(rep(1, length(index))))
}
documents <- lapply(doc.list, get.terms)

# Compute some statistics related to the data set:
D <- length(documents) # number of documents is 73
W <- length(vocab) # number of terms in the vocab
doc.length <- sapply(documents, function(x) sum(x[2, ]))
# number of tokens per document [59, 91, 81, 67, 74, ...]
N <- sum(doc.length) # total number of tokens in the data (5137L)
term.frequency <- as.integer(term.table)

K <- 20
G <- 5000
alpha <- 0.02
eta <- 0.02
# Fit the model:

set.seed(357)
t1 <- Sys.time()
fit <- lda.collapsed.gibbs.sampler(documents = documents, K = K, vocab = vocab,
num.iterations = G, alpha = alpha,
eta = eta, initial = NULL, burnin = 0,
compute.log.likelihood = TRUE)
t2 <- Sys.time()
t2 - t1 # abou
```

```
## Time difference of 21.39093 secs
```

```r
#model visualisation
theta <- t(apply(fit$document_sums + alpha, 2, function(x) x/sum(x)))
phi <- t(apply(t(fit$topics) + eta, 2, function(x) x/sum(x)))
TextReviews <- list(phi = phi,
theta = theta,
```

```
doc.length = doc.length,
vocab = vocab,
term.frequency = term.frequency)
# create the JSON object to feed the visualization:
json <- createJSON(phi = TextReviews$phi,
theta = TextReviews$theta,
doc.length = TextReviews$doc.length,
vocab = TextReviews$vocab,
term.frequency = TextReviews$term.frequency)

#plot(fit$log.likelihoods[1,],type = "l")
#These gives us the document numbers that are closely related to the topic
top.topic.documents(fit$document_sums,10)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]   10   43   57   68   56    7   43   16   23     1    66    59     8
##  [2,]   53   42   28    2   11   40   31   33   29    51     8    21    44
##  [3,]   27    3   49   69   45   41   68   52   40    72    28    34    20
##  [4,]    6   37   26   53   70   13   52   13   58    17    67    39    51
##  [5,]   49   60   62   64   24   12   67    3   28    28    25    32    41
##  [6,]   29   58   50   48   14   68   64   36   54    27     9    14    69
##  [7,]   63   73   69   32   48   35   18   67   30     2    31    17    24
##  [8,]    9    9   71   54   69   54   15   56   19    35    20    22    70
##  [9,]   12   16   33   30   63   26   19   10   65    15    56    18    43
## [10,]   64   54   56   25   68   72   11    4   52    47    17    38    53
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20]
##  [1,]    31     4    71    49    33    25    55
##  [2,]    15    73    47    46     5    60    61
##  [3,]    22     1    23    30    29    27    38
##  [4,]    51    65    50    64    73    33     6
##  [5,]    45     3    30    50    41    47    63
##  [6,]    54    56    13    69    72    61    18
##  [7,]    69    61    16    45    58    35     4
##  [8,]    66    24    37     9    19     5    50
##  [9,]    35    16    34    26    14     4     8
## [10,]    42    51    17    37    48    12    26
```

```
df<-as.data.frame(top.topic.documents(fit$document_sums,10))
#Top words for the topics
top.topic.words(fit$topics,10)
```

```
##       [,1]       [,2]     [,3]           [,4]        [,5]       [,6]
##  [1,] "studying" "200"    "people"       "friends"   "2"        "cake"
##  [2,] "online"   "â"      "migrant"      "game"      "months"   "baking"
##  [3,] "thing"    "231s"   "experience"   "month"     "allowed"  "youtube"
##  [4,] "nus"      "235"    "author"       "pumpkin"   "change"   "made"
##  [5,] "sports"   "year"   "makes"        "situation" "meals"    "make"
##  [6,] "camp"     "231t"   "farmworkers"  "unable"    "coding"   "cookies"
##  [7,] "hostel"   "april"  "free"         "hall"      "lives"    "videos"
##  [8,] "main"     "entire" "book"         "feel"      "felt"     "egg"
##  [9,] "regular"  "person" "grow"         "staying"   "part"     "design"
## [10,] "part"     "learnt" "due"          "face"      "busy"     "sleep"
```

```
##       [,7]      [,8]         [,9]      [,10]      [,11]      [,12]
##  [1,] "made"    "started"    "reading" "learn"    "climbing" "time"
##  [2,] "cooking" "period"     "books"   "cooking"  "life"     "breaker"
##  [3,] "rice"    "start"      "world"   "dishes"   "enjoyed"  "circuit"
##  [4,] "baked"   "series"     "run"     "activity" "days"     "home"
##  [5,] "chinese" "home"       "videos"  "fitness"  "gym"      "list"
##  [6,] "food"    "helped"     "java"    "guitar"   "mind"     "family"
##  [7,] "make"    "pandemic"   "months"  "journey"  "house"    "online"
##  [8,] "learnt"  "learning"   "module"  "kitchen"  "great"    "spent"
##  [9,] "cooked"  "activities" "ranging" "mobile"   "bit"      "things"
## [10,] "dishes"  "set"        "youtube" "cook"     "happy"    "friends"
##       [,13]        [,14]        [,15]      [,16]        [,17]     [,18]
##  [1,] "covid"      "started"    "data"     "language"   "felt"    "house"
##  [2,] "19"         "day"        "learning" "long"       "started" "2"
##  [3,] "china"      "running"    "science"  "till"       "good"    "learnt"
##  [4,] "quarantine" "run"        "taught"   "movie"      "cooking" "finals"
##  [5,] "classes"    "week"       "skills"   "stick"      "life"    "started"
##  [6,] "stayed"     "3"          "working"  "ups"        "korean"  "â"
##  [7,] "class"      "back"       "team"     "cycle"      "dramas"  "coffee"
##  [8,] "days"       "exercising" "explore"  "knowing"    "daily"   "stock"
##  [9,] "singapore"  "cardio"     "machine"  "experience" "late"    "similar"
## [10,] "hard"       "exercises"  "neural"   "dramas"     "fit"     "meant"
##       [,19]       [,20]
##  [1,] "day"       "exams"
##  [2,] "morning"   "final"
##  [3,] "practice"  "summer"
##  [4,] "back"      "examinations"
##  [5,] "hall"      "vacation"
##  [6,] "singapore" "life"
##  [7,] "lunch"     "courses"
##  [8,] "dinner"    "internet"
##  [9,] "afternoon" "university"
## [10,] "built"     "students"
```

```
serVis(json, out.dir = 'vis', privacy.file_unique_origin=TRUE)
```

```
## Loading required namespace: servr
```

Hierarchical clustering

With reference to what we learnt in topic 6, we first get a 20 x 20 matrix of the correlation values With this matrix, we pivot it longer to form a dataframe of 20 rows and 2 columns.

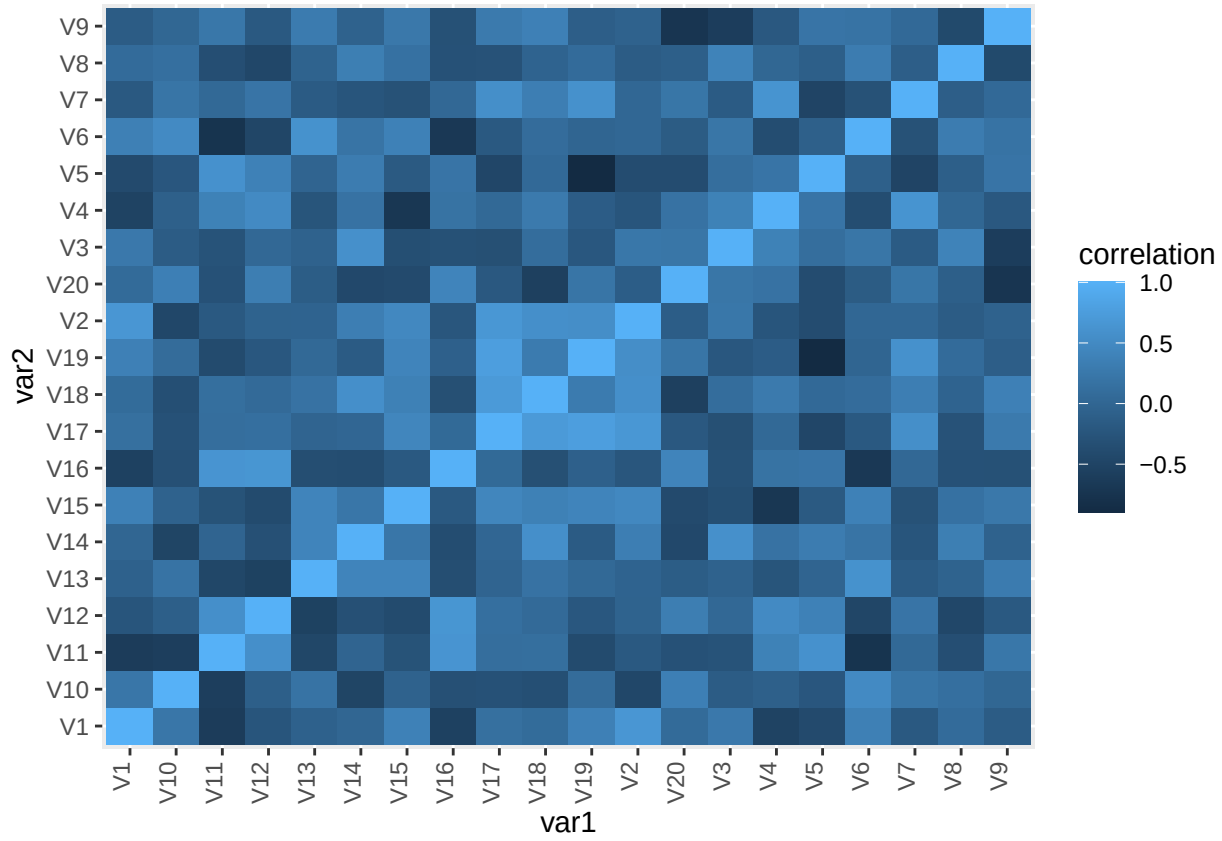
```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

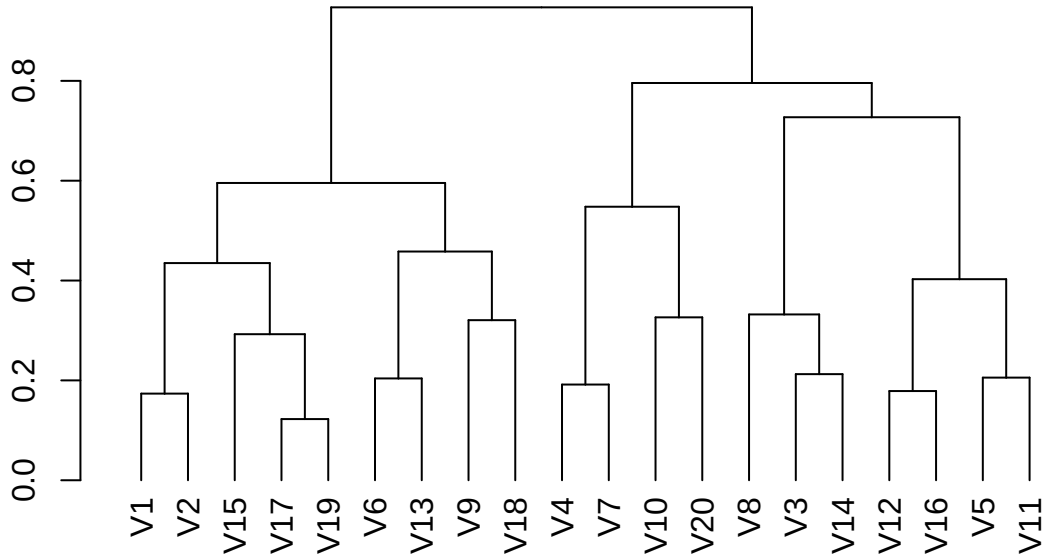
```
df2<-df%>%
  cor(use="pair")
topics_df<-as.data.frame(df2)%>%
  mutate(var1 = row.names(df2))%>%
  pivot_longer(V1:V20,names_to="var2",values_to="correlation")

ggplot(topics_df) +
  geom_tile(aes(x=var1, y=var2, fill=correlation)) +
  theme(axis.text.x=element_text(angle=90,
                                 vjust=0, hjust=1))
```



Multi-Dimensional Scaling

```
dist<-as.dist((1 - df2)/2)
hc <-hclust(dist)
plot(as.dendrogram(hc))
abline(h=4)
```

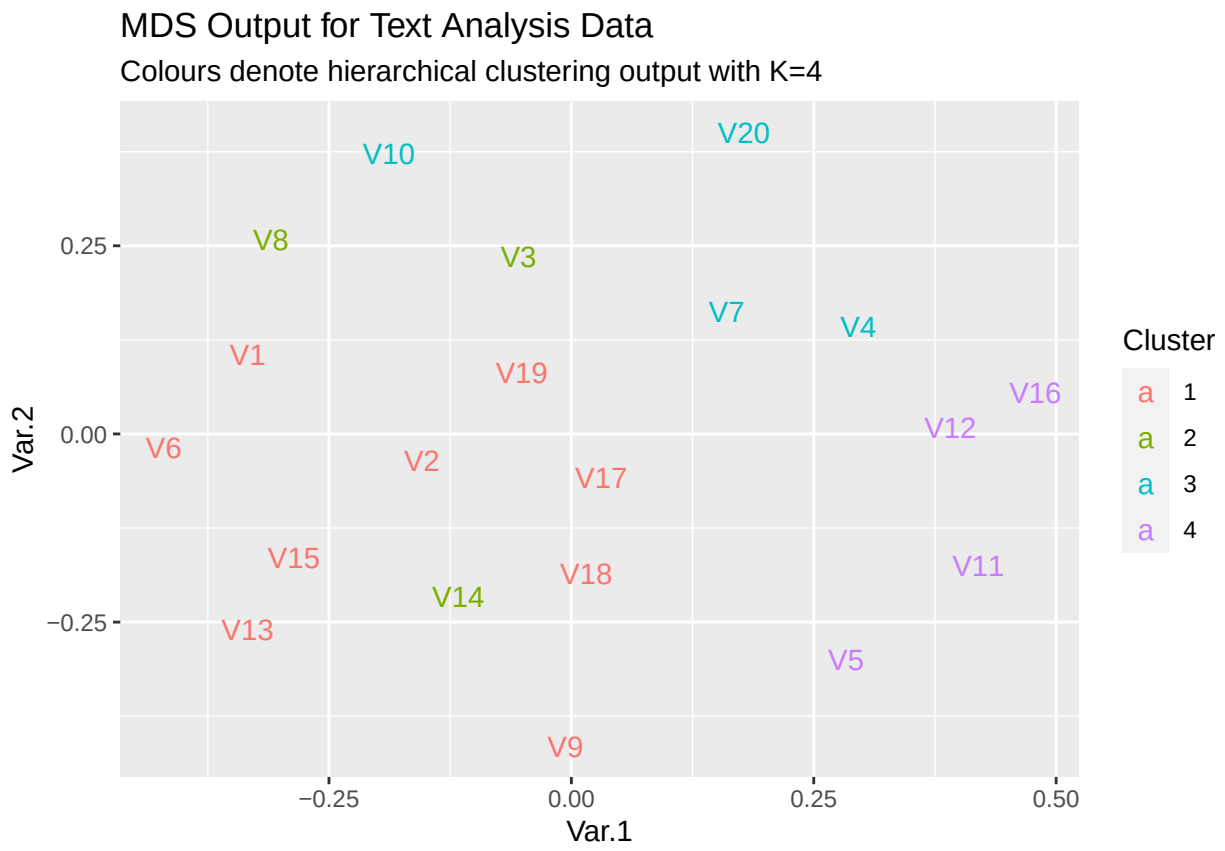



```
mds2 <- MASS::sammon(dist, k = 2)
```

```
## Initial stress        : 0.11406
## stress after  10 iters: 0.07827, magic = 0.500
## stress after  20 iters: 0.07756, magic = 0.500
## stress after  30 iters: 0.07749, magic = 0.500
```

```
grps <- as.factor(cutree(hc, k=4))
mds_df <- data.frame(mds2$points) %>%
  mutate(label = row.names(mds2$points), Cluster=grps) %>%
  rename('Var.1' = 'X1', 'Var.2'='X2')
```

```
ggplot(mds_df) +
  geom_text(aes(x=Var.1, y=Var.2, label=label, col=Cluster),
            show.legend = TRUE) +
  labs(title="MDS Output for Text Analysis Data",
  subtitle = "Colours denote hierarchical clustering output with K=4")
```


MDS Output for Text Analysis Data
Colours denote hierarchical clustering output with K=4
V20
V10
V8
V3
V7
V4
V1
V19
V16
V12
V6
V2
V17
V15
V18
V11
V14
V13
V5
V9
0.25
0.00
−0.25
Var.2
−0.25
0.00
0.25
0.50
Var.1
Cluster
a 1
a 2
a 3
a 4