HW4

Shivangi Mundhra SUID:842548148

2022-10-03

```
#install.packages('tidyverse')  # data manipulation
#install.packages('cluster')    # clustering algorithms
#install.packages('factoextra') # clustering algorithms & visualization
#install.packages('gridExtra')
```

```
library(tidyverse, warn.conflicts = FALSE)  # data manipulation
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'pillar'
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'tibble'
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'hms'
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

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

```
library(cluster)    # clustering algorithms
```

```
## Warning: package 'cluster' was built under R version 4.1.2
```

```
library(factoextra) # clustering algorithms & visualization
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(gridExtra)  # subfigure layout package
```

```
##
## Attaching package: 'gridExtra'
```

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

read the data file

```
Essay = read.csv('/Users/shivangi/Desktop/Applied Machine learning/Week 4/HW4-data-fedPapers.csv')
str(Essay)
```

```
## 'data.frame':    85 obs. of  72 variables:
##  $ author  : chr  "dispt" "dispt" "dispt" "dispt" ...
##  $ filename: chr  "dispt_fed_49.txt" "dispt_fed_50.txt" "dispt_fed_51.txt" "dispt_fed_52.txt" ...
##  $ a       : num  0.28 0.177 0.339 0.27 0.303 0.245 0.349 0.414 0.248 0.442 ...
##  $ all     : num  0.052 0.063 0.09 0.024 0.054 0.059 0.036 0.083 0.04 0.062 ...
##  $ also    : num  0.009 0.013 0.008 0.016 0.027 0.007 0.007 0.009 0.007 0.006 ...
##  $ an      : num  0.096 0.038 0.03 0.024 0.034 0.067 0.029 0.018 0.04 0.075 ...
##  $ and     : num  0.358 0.393 0.301 0.262 0.404 0.282 0.335 0.478 0.356 0.423 ...
##  $ any     : num  0.026 0.063 0.008 0.056 0.04 0.052 0.058 0.046 0.034 0.037 ...
##  $ are     : num  0.131 0.051 0.068 0.064 0.128 0.111 0.087 0.11 0.154 0.093 ...
##  $ as      : num  0.122 0.139 0.203 0.111 0.148 0.252 0.073 0.074 0.161 0.1 ...
##  $ at      : num  0.017 0.114 0.023 0.056 0.013 0.015 0.116 0.037 0.047 0.031 ...
##  $ be      : num  0.411 0.393 0.474 0.365 0.344 0.297 0.378 0.331 0.289 0.379 ...
##  $ been    : num  0.026 0.165 0.015 0.127 0.047 0.03 0.044 0.046 0.027 0.025 ...
##  $ but     : num  0.009 0 0.038 0.032 0.061 0.037 0.007 0.055 0.027 0.037 ...
##  $ by      : num  0.14 0.139 0.173 0.167 0.209 0.186 0.102 0.092 0.168 0.174 ...
##  $ can     : num  0.035 0 0.023 0.056 0.088 0 0.058 0.037 0.047 0.056 ...
##  $ do      : num  0.026 0.013 0 0 0 0 0.015 0.028 0 0 ...
##  $ down    : num  0 0 0.008 0 0 0.007 0 0 0 0 ...
##  $ even    : num  0.009 0.025 0.015 0.024 0.02 0.007 0.007 0.018 0 0.006 ...
##  $ every   : num  0.044 0 0.023 0.04 0.027 0.007 0.087 0.064 0.081 0.05 ...
##  $ for.    : num  0.096 0.076 0.098 0.103 0.141 0.067 0.116 0.055 0.127 0.1 ...
##  $ from    : num  0.044 0.101 0.053 0.079 0.074 0.096 0.08 0.083 0.074 0.124 ...
##  $ had     : num  0.035 0.101 0.008 0.016 0 0.022 0.015 0.009 0.007 0 ...
##  $ has     : num  0.017 0.013 0.015 0.024 0.054 0.015 0.036 0.037 0.02 0.019 ...
##  $ have    : num  0.044 0.152 0.023 0.143 0.047 0.119 0.044 0.074 0.074 0.044 ...
##  $ her     : num  0 0 0 0 0 0 0.007 0 0.034 0.025 ...
##  $ his     : num  0.017 0 0 0.024 0.02 0.067 0 0.018 0.02 0.05 ...
##  $ if.     : num  0 0.025 0.023 0.04 0.034 0.03 0.029 0 0 0.025 ...
##  $ in.     : num  0.262 0.291 0.308 0.238 0.263 0.401 0.189 0.267 0.248 0.274 ...
##  $ into    : num  0.009 0.025 0.038 0.008 0.013 0.037 0 0.037 0.013 0.037 ...
##  $ is      : num  0.157 0.038 0.15 0.151 0.189 0.26 0.167 0.083 0.208 0.23 ...
##  $ it      : num  0.175 0.127 0.173 0.222 0.108 0.156 0.102 0.165 0.134 0.131 ...
##  $ its     : num  0.07 0.038 0.03 0.048 0.013 0.015 0 0.046 0.02 0.019 ...
##  $ may     : num  0.035 0.038 0.12 0.056 0.047 0.074 0.08 0.092 0.027 0.106 ...
##  $ more    : num  0.026 0 0.038 0.056 0.067 0.045 0.08 0.064 0.06 0.081 ...
##  $ must    : num  0.026 0.013 0.083 0.071 0.013 0.015 0.044 0.018 0.027 0.068 ...
##  $ my      : num  0 0 0 0 0 0 0.007 0 0 0 ...
##  $ no      : num  0.035 0 0.03 0.032 0.047 0.059 0.022 0.018 0.02 0.044 ...
##  $ not     : num  0.114 0.127 0.068 0.087 0.128 0.134 0.102 0.101 0.094 0.106 ...
##  $ now     : num  0 0 0 0 0 0 0.007 0 0.007 0.012 ...
```

```
##  $ of      : num  0.9 0.747 0.858 0.802 0.869 ...
##  $ on      : num  0.14 0.139 0.15 0.143 0.054 0.141 0.051 0.083 0.127 0.118 ...
##  $ one     : num  0.026 0.025 0.03 0.032 0.047 0.052 0.073 0.046 0.06 0.031 ...
##  $ only    : num  0.035 0 0.023 0.048 0.027 0.022 0.007 0.046 0.02 0.012 ...
##  $ or      : num  0.096 0.114 0.06 0.064 0.081 0.074 0.153 0.037 0.154 0.081 ...
##  $ our     : num  0.017 0 0 0.016 0.027 0.03 0.051 0 0.007 0.025 ...
##  $ shall   : num  0.017 0 0.008 0.016 0 0.015 0.007 0 0.02 0 ...
##  $ should  : num  0.017 0.013 0.068 0.032 0 0.03 0.007 0 0 0.012 ...
##  $ so      : num  0.035 0.013 0.038 0.04 0.027 0.007 0.051 0.018 0.04 0.05 ...
##  $ some    : num  0.009 0.063 0.03 0.024 0.067 0.045 0.007 0.028 0.027 0.025 ...
##  $ such    : num  0.026 0 0.045 0.008 0.027 0.015 0.015 0 0.013 0.031 ...
##  $ than    : num  0.009 0 0.023 0 0.047 0.03 0.109 0.055 0.067 0.044 ...
##  $ that    : num  0.184 0.152 0.188 0.238 0.162 0.208 0.233 0.165 0.208 0.218 ...
##  $ the     : num  1.43 1.25 1.49 1.33 1.19 ...
##  $ their   : num  0.114 0.165 0.053 0.071 0.027 0.089 0.109 0.083 0.154 0.081 ...
##  $ then    : num  0 0 0.015 0.008 0.007 0.007 0.015 0.009 0.007 0.012 ...
##  $ there   : num  0.009 0 0.015 0 0.007 0.007 0.036 0.028 0.02 0 ...
##  $ things  : num  0.009 0 0 0 0 0 0 0 0 0.012 ...
##  $ this    : num  0.044 0.051 0.075 0.103 0.094 0.126 0.08 0.11 0.067 0.093 ...
##  $ to      : num  0.507 0.355 0.361 0.532 0.485 0.445 0.56 0.34 0.49 0.498 ...
##  $ up      : num  0 0 0 0 0 0 0.007 0 0 0 ...
##  $ upon    : num  0 0.013 0 0 0 0 0 0 0 0 ...
##  $ was     : num  0.009 0.051 0.008 0.087 0.027 0.007 0.015 0.018 0.027 0 ...
##  $ were    : num  0.017 0 0.015 0.079 0.02 0.03 0.029 0.009 0.007 0 ...
##  $ what    : num  0 0 0.008 0.008 0.02 0.015 0.015 0.009 0.02 0.025 ...
##  $ when    : num  0.009 0 0 0.024 0.007 0.037 0.007 0 0.02 0.012 ...
##  $ which   : num  0.175 0.114 0.105 0.167 0.155 0.186 0.211 0.175 0.201 0.199 ...
##  $ who     : num  0.044 0.038 0.008 0 0.027 0.045 0.022 0.018 0.04 0.031 ...
##  $ will    : num  0.009 0.089 0.173 0.079 0.168 0.111 0.145 0.267 0.154 0.106 ...
##  $ with    : num  0.087 0.063 0.045 0.079 0.074 0.089 0.073 0.129 0.027 0.081 ...
##  $ would   : num  0.192 0.139 0.068 0.064 0.04 0.037 0.073 0.037 0.04 0.031 ...
##  $ your    : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
typeof(Essay)
```

```
## [1] "list"
```

To remove any missing value that might be present in the data

```
Essay = na.omit(Essay)
```

Remove the label information

```
Essay.unlabeled = Essay[, which(!names(Essay) %in% c('author', 'filename'))]
head(Essay.unlabeled)
```

```
##       a   all  also    an   and   any   are    as    at    be  been   but    by
## 1 0.280 0.052 0.009 0.096 0.358 0.026 0.131 0.122 0.017 0.411 0.026 0.009 0.140
## 2 0.177 0.063 0.013 0.038 0.393 0.063 0.051 0.139 0.114 0.393 0.165 0.000 0.139
## 3 0.339 0.090 0.008 0.030 0.301 0.008 0.068 0.203 0.023 0.474 0.015 0.038 0.173
## 4 0.270 0.024 0.016 0.024 0.262 0.056 0.064 0.111 0.056 0.365 0.127 0.032 0.167
## 5 0.303 0.054 0.027 0.034 0.404 0.040 0.128 0.148 0.013 0.344 0.047 0.061 0.209
```

```
## 6 0.245 0.059 0.007 0.067 0.282 0.052 0.111 0.252 0.015 0.297 0.030 0.037 0.186
##     can    do  down  even every  for.  from   had   has  have her   his   if.
## 1 0.035 0.026 0.000 0.009 0.044 0.096 0.044 0.035 0.017 0.044   0 0.017 0.000
## 2 0.000 0.013 0.000 0.025 0.000 0.076 0.101 0.101 0.013 0.152   0 0.000 0.025
## 3 0.023 0.000 0.008 0.015 0.023 0.098 0.053 0.008 0.015 0.023   0 0.000 0.023
## 4 0.056 0.000 0.000 0.024 0.040 0.103 0.079 0.016 0.024 0.143   0 0.024 0.040
## 5 0.088 0.000 0.000 0.020 0.027 0.141 0.074 0.000 0.054 0.047   0 0.020 0.034
## 6 0.000 0.000 0.007 0.007 0.007 0.067 0.096 0.022 0.015 0.119   0 0.067 0.030
##     in.  into    is    it   its   may  more  must my    no   not now    of
## 1 0.262 0.009 0.157 0.175 0.070 0.035 0.026 0.026  0 0.035 0.114   0 0.900
## 2 0.291 0.025 0.038 0.127 0.038 0.038 0.000 0.013  0 0.000 0.127   0 0.747
## 3 0.308 0.038 0.150 0.173 0.030 0.120 0.038 0.083  0 0.030 0.068   0 0.858
## 4 0.238 0.008 0.151 0.222 0.048 0.056 0.056 0.071  0 0.032 0.087   0 0.802
## 5 0.263 0.013 0.189 0.108 0.013 0.047 0.067 0.013  0 0.047 0.128   0 0.869
## 6 0.401 0.037 0.260 0.156 0.015 0.074 0.045 0.015  0 0.059 0.134   0 0.876
##      on   one  only    or   our shall should    so  some  such  than  that
## 1 0.140 0.026 0.035 0.096 0.017 0.017  0.017 0.035 0.009 0.026 0.009 0.184
## 2 0.139 0.025 0.000 0.114 0.000 0.000  0.013 0.013 0.063 0.000 0.000 0.152
## 3 0.150 0.030 0.023 0.060 0.000 0.008  0.068 0.038 0.030 0.045 0.023 0.188
## 4 0.143 0.032 0.048 0.064 0.016 0.016  0.032 0.040 0.024 0.008 0.000 0.238
## 5 0.054 0.047 0.027 0.081 0.027 0.000  0.000 0.027 0.067 0.027 0.047 0.162
## 6 0.141 0.052 0.022 0.074 0.030 0.015  0.030 0.007 0.045 0.015 0.030 0.208
##     the their  then there things  this    to up  upon   was  were  what  when
## 1 1.425 0.114 0.000 0.009  0.009 0.044 0.507  0 0.000 0.009 0.017 0.000 0.009
## 2 1.254 0.165 0.000 0.000  0.000 0.051 0.355  0 0.013 0.051 0.000 0.000 0.000
## 3 1.490 0.053 0.015 0.015  0.000 0.075 0.361  0 0.000 0.008 0.015 0.008 0.000
## 4 1.326 0.071 0.008 0.000  0.000 0.103 0.532  0 0.000 0.087 0.079 0.008 0.024
## 5 1.193 0.027 0.007 0.007  0.000 0.094 0.485  0 0.000 0.027 0.020 0.020 0.007
## 6 1.469 0.089 0.007 0.007  0.000 0.126 0.445  0 0.000 0.007 0.030 0.015 0.037
##   which   who  will  with would your
## 1 0.175 0.044 0.009 0.087 0.192    0
## 2 0.114 0.038 0.089 0.063 0.139    0
## 3 0.105 0.008 0.173 0.045 0.068    0
## 4 0.167 0.000 0.079 0.079 0.064    0
## 5 0.155 0.027 0.168 0.074 0.040    0
## 6 0.186 0.045 0.111 0.089 0.037    0
```

Since we don't want the clustering algorithm to depend on an arbitrary variable unit, we start by scaling/standardizing the data using the R function scale:

```
Essay.unlabeled = scale(Essay.unlabeled)
head(Essay.unlabeled)
```

```
##            a         all        also          an         and         any
## 1 -0.1723368 -0.03534435  0.15950160  0.93187600 -0.23253417 -0.68518464
## 2 -1.5135000  0.43010589  0.63520811 -1.02557977  0.07316143  0.93870812
## 3  0.5959023  1.57257465  0.04057497 -1.29557367 -0.73038129 -1.47518652
## 4 -0.3025468 -1.22012677  0.99198800 -1.49806910 -1.07101353  0.63148517
## 5  0.1271463  0.04928297  2.30018092 -1.16057672  0.16923719 -0.07073873
## 6 -0.6280719  0.26085126 -0.07835166 -0.04685188 -0.89633033  0.45592919
##          are          as         at          be       been        but        by
## 1  1.5654231 -0.05418106 -1.0404489  1.26866739 -1.0162916 -1.3159303 0.2605755
## 2 -0.7567578  0.36449075  2.6603712  1.06077823  3.1791960 -1.8238448 0.2402926
```

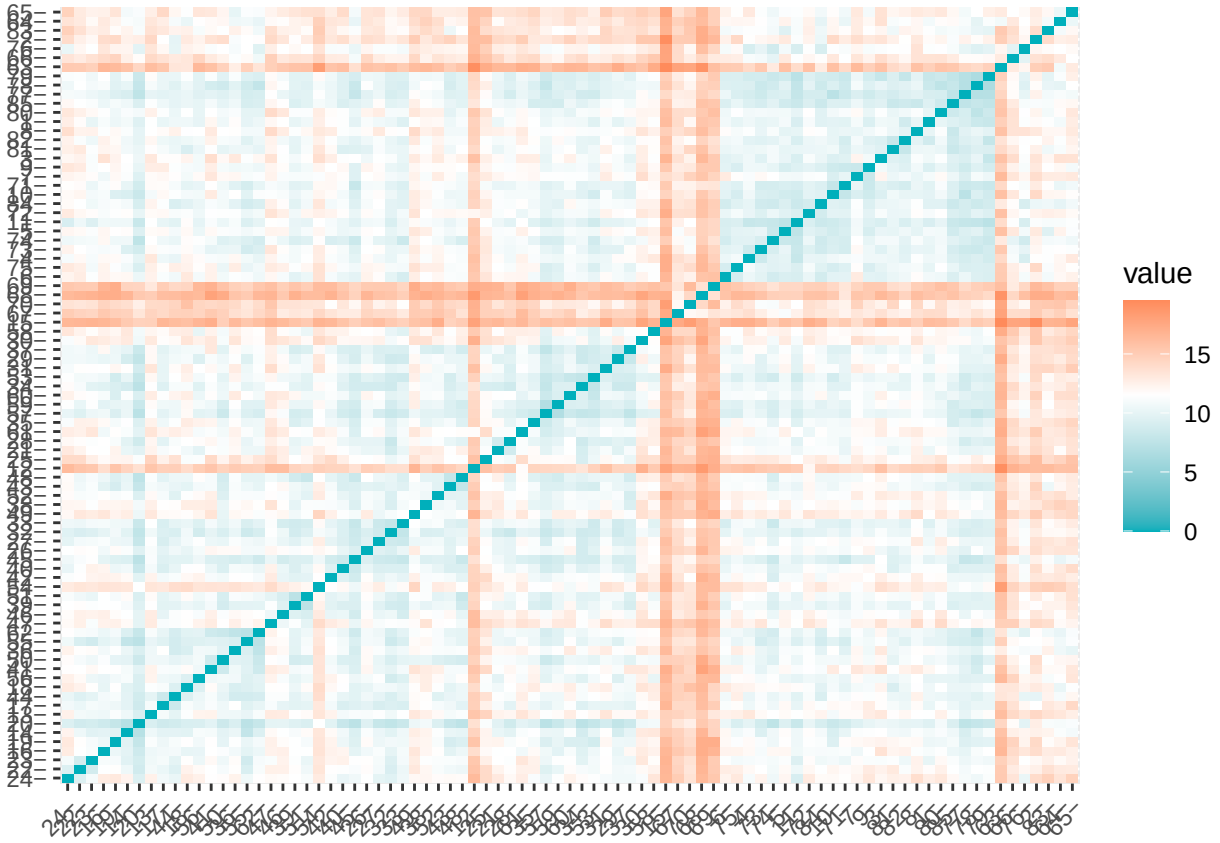
```
## 3 -0.2632943  1.94066694 -0.8115322  1.99627946 -1.3483086  0.3206833 0.9299109
## 4 -0.3794034 -0.32508634  0.4475097  0.73739509  2.0322282 -0.0179264 0.8082135
## 5  1.4783413  0.58614052 -1.1930600  0.49485773 -0.3824410  1.6186872 1.6600949
## 6  0.9848779  3.14742684 -1.1167544 -0.04796397 -0.8955582  0.2642484 1.1935884
##           can         do       down       even       every        for.
## 1 -0.02062228  3.0053860 -0.4239542 -0.2431406  1.09472692  0.06360606
## 2 -1.27268916  1.0262730 -0.4239542  1.3777967 -1.30239174 -0.50550078
## 3 -0.44990235 -0.9528399  1.7936523  0.3647109 -0.04935244  0.12051674
## 4  0.73061785 -0.9528399 -0.4239542  1.2764881  0.87680704  0.26279345
## 5  1.87536471 -0.9528399 -0.4239542  0.8712538  0.16856744  1.34409646
## 6 -1.27268916 -0.9528399  1.5164515 -0.4457577 -0.92103196 -0.76159886
##          from         had        has       have        her        his
## 1 -1.31170510  0.61463684 -1.1001862 -1.3396177 -0.3769551 -0.2733913
## 2  0.77813746  3.54670543 -1.2606596  1.5116901 -0.3769551 -0.6732398
## 3 -0.98172996 -0.58484577 -1.1804229 -1.8940386 -0.3769551 -0.6732398
## 4 -0.02846844 -0.22944351 -0.8193579  1.2740811 -0.3769551 -0.1087477
## 5 -0.21178797 -0.94024802  0.3841920 -1.2604147 -0.3769551 -0.2028298
## 6  0.59481793  0.03710818 -1.1804229  0.6404571 -0.3769551  0.9026339
##          if.        in.       into          is         it         its
## 1 -1.5128966 -0.7758578 -0.7853726  0.01246845  0.4072769  0.72662889
## 2 -0.1289512 -0.3693776  0.0471346 -2.02169914 -0.6633995 -0.34810226
## 3 -0.2396668 -0.1310961  0.7235467 -0.10718847  0.3626654 -0.61678505
## 4  0.7014161 -1.1122552 -0.8374043 -0.09009462  1.4556476 -0.01224877
## 5  0.3692692 -0.7618412 -0.5772458  0.55947150 -1.0872090 -1.18773597
## 6  0.1478379  1.1724439  0.6715150  1.77313452 -0.0165325 -1.12056527
##          may        more       must         my          no        not
## 1 -0.8830840 -0.70385836 -0.2855115 -0.4056847  0.13701191  0.7490012
## 2 -0.7842747 -1.63698791 -0.8122064 -0.4056847 -1.68267750  1.2015142
## 3  1.9165132 -0.27318318  2.0238430 -0.4056847 -0.12294372 -0.8521988
## 4 -0.1914188  0.37282959  1.5376631 -0.4056847 -0.01896147 -0.1908336
## 5 -0.4878468  0.76761517 -0.8122064 -0.4056847  0.76090542  1.2363229
## 6  0.4014371 -0.02195599 -0.7311764 -0.4056847  1.38479893  1.4451751
##          now          of         on        one         only         or
## 1 -0.8475975 -0.07882731  1.8402042 -0.6574777  0.826718310 -0.0164496
## 2 -0.8475975 -1.36832319  1.8141910 -0.7019372 -1.561133119  0.3830407
## 3 -0.8475975 -0.43280657  2.1003361 -0.4796396  0.008026391 -0.8154303
## 4 -0.8475975 -0.90477892  1.9182438 -0.3907206  1.713634554 -0.7266546
## 5 -0.8475975 -0.34009772 -0.3969308  0.2761720  0.280923697 -0.3493582
## 6 -0.8475975 -0.28110118  1.8662174  0.4984695 -0.060197935 -0.5047156
##          our       shall     should         so       some       such       than
## 1 -0.1827435 -0.09108447 -0.4841143  0.3222109 -0.7284739 -0.1937633 -1.3055544
## 2 -0.7005166 -0.97442043 -0.6865729 -1.0471854  2.8824281 -1.7565984 -1.6416073
## 3 -0.7005166 -0.55873292  2.0972330  0.5089468  0.6757658  0.9483085 -0.7828055
## 4 -0.2132007 -0.14304541  0.2751055  0.6334373  0.2745544 -1.2757260 -1.6416073
## 5  0.1218290 -0.97442043 -1.3445634 -0.1757514  3.1499023 -0.1336542  0.1133355
## 6  0.2132007 -0.19500635  0.1738762 -1.4206571  1.6787941 -0.8549627 -0.5214311
##          that        the       their       then      there     things
## 1 -0.47881966  0.7661753  0.68392790 -1.0008309 -0.7880202  1.4470317
## 2 -1.02535321 -0.1422477  1.90906321 -1.0008309 -1.1961687 -0.6067331
## 3 -0.41050296  1.1114823 -0.78143002  1.4673691 -0.5159211 -0.6067331
## 4  0.44345572  0.2402462 -0.34902932  0.3155424 -1.1961687 -0.6067331
## 5 -0.85456148 -0.4663051 -1.40600880  0.1509958 -0.8787198 -0.6067331
## 6 -0.06891949  0.9999216  0.08337138  0.1509958 -0.8787198 -0.6067331
##         this          to         up       upon         was        were
```

```
## 1 -1.4818540 -0.27620743 -0.5120406 -1.0657505 -0.57950415 -0.15447029
## 2 -1.2406879 -1.73456457 -0.5120406 -0.5916546  0.86621899 -0.96910375
## 3 -0.4138329 -1.67699784 -0.5120406 -1.0657505 -0.61392613 -0.25030952
## 4  0.5508314 -0.03634605 -0.5120406 -1.0657505  2.10541024  2.81654586
## 5  0.2407607 -0.48728543 -0.5120406 -1.0657505  0.04009148 -0.01071144
## 6  1.3432342 -0.87106363 -0.5120406 -1.0657505 -0.64834810  0.46848471
##         what       when      which        who       will       with      would
## 1 -1.1025238 -0.2068178  0.4006641  0.4609835 -1.3597325  0.2849510  1.2098029
## 2 -1.1025238 -0.8858546 -1.0164186  0.2198537 -0.1463229 -0.6496151  0.4995915
## 3 -0.4165986 -0.8858546 -1.2254963 -0.9857954  1.1277571 -1.3505396 -0.4518237
## 4 -0.4165986  0.9249103  0.2148172 -1.3073018 -0.2979991 -0.0265710 -0.5054246
## 5  0.6122890 -0.3577148 -0.0639532 -0.2222177  1.0519190 -0.2212723 -0.8270297
## 6  0.1835858  1.9057413  0.6562036  0.5011718  0.1873647  0.3628315 -0.8672304
##         your
## 1 -0.2087646
## 2 -0.2087646
## 3 -0.2087646
## 4 -0.2087646
## 5 -0.2087646
## 6 -0.2087646
```

```
typeof(Essay.unlabeled)
```

```
## [1] "double"
```

```
distance = factoextra::get_dist(Essay.unlabeled) # default method is euclidean

# visualization of the distance matrix
factoextra::fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```



k-Means Clustering

```
# the kmeans() function is available in pre-loaded {stats} package
model.r = kmeans(Essay.unlabeled, centers = 7, nstart = 25)
model.r
```

```
## K-means clustering with 7 clusters of sizes 15, 7, 4, 15, 1, 24, 19
##
## Cluster means:
##             a        all        also         an         and        any
## 1  0.57333261 -0.5543919  0.01678964  0.6708819 -0.65235612  0.4325217
## 2 -0.79362463 -0.4524361  0.14251208 -0.7845138  1.24728209 -0.9735980
## 3 -1.98551138 -0.6065787  1.52715783 -1.3208856  2.84844078  0.1377340
## 4  0.40319151  0.4554941 -0.36377557  0.1713932 -0.10326860 -0.5535177
## 5  1.27299447 -0.3315400  0.87306137 -1.2955737 -1.00114025  0.1487062
## 6 -0.04049915  0.1180427  0.39239958 -0.4391868  0.04586718 -0.2938411
## 7 -0.07639257  0.2408079 -0.64168833  0.5251089 -0.46789679  0.7885616
##           are         as         at         be        been        but
## 1 -0.70837900  0.0295533  0.4653143  0.3916831 -0.06853399 -0.4393074
## 2 -0.09327752 -0.2899038  0.5456169 -1.6483805  0.26434540 -0.5177788
## 3  0.47690082  0.9617138 -0.8019940  0.1195023 -1.31812527  1.0261202
## 4 -0.10074167 -0.3841929  0.4882060 -0.4983905  0.23329965 -0.2248546
## 5 -0.49551242 -0.8422692 -1.1167544  0.8644385 -0.89555817 -0.1307963
## 6  0.26040583  0.2721367 -0.3982103  0.1993856  0.02880736  0.1960561
```

```
## 7  0.26989062 -0.1151023 -0.2231761  0.3690301  0.06077789  0.2583078
##           by        can          do        down       even      every
## 1 -0.5439792 -0.3378126  0.01134333  0.53700864  0.3376953 -0.3144883
## 2  0.5851017 -1.0478282 -0.51787001 -0.42395419 -0.3444492 -0.6408493
## 3  0.4329801  0.2029611  0.60761453 -0.42395419 -0.1924863 -1.1253318
## 4 -0.2343270 -0.2161832  0.03164193 -0.10979326  0.4795273 -0.4561362
## 5 -1.0578124  0.8737112  1.33075194 -0.42395419 -1.1549178  1.1492069
## 6  0.6873613  0.0375094 -0.10283626 -0.08900320 -0.1080625  0.6952071
## 7 -0.5048367  0.6873133  0.08879849  0.04291034 -0.2804648  0.1427611
##          for.        from        had          has         have         her
## 1  0.46577490  0.08396753  0.1763074 -0.110600733 -0.014287599 -0.35522174
## 2 -0.06241046  0.38007221  2.0743247 -0.062840815  0.493365873  0.12202513
## 3 -0.05732915  0.84229929 -0.4182510 -0.859476253 -0.613590249  0.48461743
## 4 -0.53395613 -0.09201921 -0.0902443  0.651647564  0.323645173  0.08565501
## 5 -1.81444652 -0.17512406 -0.9402480 -0.578647933 -0.917201724  6.60876813
## 6  0.09561832 -0.25761785 -0.1702098 -0.145704273 -0.195574450 -0.00244272
## 7  0.06360606  0.02363289 -0.4796280 -0.008545329 -0.001503958 -0.27890987
##          his         if.        in.        into         is         it
## 1  1.0280766  0.48736590  0.5192723 -0.37952535  0.1560568  0.5143446
## 2  0.3213415 -0.49273111 -0.4514745  0.59718401 -1.0668858 -0.4594612
## 3 -0.4086342  1.76705416 -0.7022708  0.97069729 -1.2695699  0.6582167
## 4 -0.0507305 -0.11418907 -0.1301616 -0.20955512 -0.2427996 -0.7496485
## 5 -0.6732398  0.97820523 -1.0842221  4.20967065 -0.3635961 -1.4887126
## 6 -0.1979296 -0.36422189 -0.5101272  0.02545472  0.3173087 -0.1373554
## 7 -0.5184997 -0.07650691  0.7084221 -0.21302390  0.3471480  0.4683243
##          its         may        more       must          my         no
## 1 -0.2563023 -0.02454088  0.04264529 -0.3395315 -0.09861408 -0.2650528
## 2 -0.3624960 -1.23597444 -0.38085197 -0.7022372  0.03891560 -0.7988284
## 3 -0.5915960  0.01443391  1.86224792 -0.6197602 -0.40568466 -0.8248239
## 4 -0.0727024 -0.58446033 -0.42152685 -0.1315546 -0.05711806 -0.3863654
## 5 -0.6167850  1.91651322  1.59307593  0.4842733 -0.40568466  0.6569232
## 6 -0.1619835  0.27518071  0.24721600  0.2074209 -0.22413956  0.3536416
## 7  0.7549113  0.48464490 -0.34874023  0.4736115  0.49849187  0.5009498
##           not        now          of         on        one        only
## 1  0.09691833 -0.4075517  0.50271004 -0.7801919 -0.1654591 -0.15116370
## 2 -0.66820994  0.7774954 -0.64350851  0.3017094  0.1110367 -0.44030490
## 3  0.79251208 -0.6018273 -2.45765222  0.1688563  1.7433358  1.38956900
## 4 -0.46698253  0.3508250  0.87972605 -0.4489572 -0.3877567 -0.28306407
## 5 -0.60853789 -0.8475975 -0.18839232 -0.6310496  4.1886087 -1.56113312
## 6  0.25297731 -0.2741336 -0.20103444  1.0305434  0.1224162  0.22407009
## 7  0.08397193  0.2759235 -0.07306073 -0.4448498 -0.3462611  0.01161715
##           or        our       shall      should          so        some
## 1  0.3238570 -0.3979747 -0.06337197  0.51468154  0.46330021 -0.14003062
## 2 -0.4666689 -0.3263276 -0.53646395 -0.46242227 -0.14907485  0.80950286
## 3  1.8755810  1.5457052  0.03881788  0.83186670  0.92910244 -0.12665691
## 4 -0.0401231  0.6456936  0.10983116 -0.05557687  0.07737943 -0.08207787
## 5 -1.1483389 -0.7005166  1.36382183 -1.34456335  0.94466376 -1.33029091
## 6 -0.2032483 -0.1345195 -0.20150147 -0.33437925 -0.02013818  0.35814012
## 7 -0.1297553 -0.1939646  0.33554219  0.12592549 -0.59181199 -0.47859668
##         such        than        that        the       their       then
## 1  0.0867456  0.26767087 -0.03817698  0.3620776 -0.17126459 -0.2877954
## 2 -0.2366983 -0.45208682 -0.57153517 -0.4549213  1.39086872  0.3390491
## 3  1.8349169  0.87878924  0.12322121 -2.3814309  1.09831190  0.2744058
## 4 -0.3740904 -0.29988509 -0.41505741 -0.3200366 -0.41789314 -0.3316745
```

```
## 5 -0.8549627  3.95927391  0.98998928 -1.2153557 -1.69427594 -1.0008309
## 6 -0.2388450 -0.06402576 -0.22619021  0.3883298 -0.02973344  0.1235713
## 7  0.2744545 -0.12052588  0.77605015  0.2092105 -0.15179392  0.2029579
##        there      things        this          to          up        upon
## 1  0.2399095 -0.40896320  0.17874661  0.7228951  0.09571931  0.75040160
## 2 -0.8204129 -0.60673314 -0.56640730 -1.2219466  0.05510862 -0.72190069
## 3 -0.4025465 -0.20738998 -1.53353242 -0.3673547 -0.51204063 -1.06575048
## 4  0.1310699  1.00585259  0.36708582  0.2777124  0.56624313  0.46594405
## 5  3.5655645 -0.60673314 -1.96418610  1.0574218  1.69353979  0.02831704
## 6 -0.7105475 -0.15984913 -0.07792298 -0.7751191 -0.29148259 -1.01560572
## 7  0.8039979  0.02981385  0.30241221  0.6610193 -0.15605221  0.81143905
##           was        were       what        when      which        who
## 1  0.06762907  0.23208127  0.4007955  0.74383380 -0.5223756  0.7905275
## 2  2.01689658  1.28311817 -0.4043500  0.05186294 -0.6480435  0.6561838
## 3 -0.42460524  0.02522827  0.5051132  0.60425401 -1.3590738  0.5815484
## 4 -0.12283922 -0.07140962 -0.3365740 -0.33256531  0.2411455 -0.6107046
## 5 -0.88930196 -0.96910375 -1.1025238  2.50932955  0.1683554  0.5011718
## 6 -0.07177995 -0.06661766 -0.1022163 -0.23511095  0.2709584 -0.2942217
## 7 -0.47261484 -0.46973092  0.1790732 -0.30609214  0.3957734 -0.1608776
##          will        with      would        your
## 1 -0.07756304  0.01496527  0.5594458 -0.20876463
## 2 -1.01521083  0.22932210 -0.7293996 -0.20876463
## 3  0.82819663  0.58673800  0.5866929  0.61658391
## 4 -0.55281512  0.40696383  0.5147784 -0.13998559
## 5  0.55138758  2.73818703  1.4510068 -0.20876463
## 6  0.32387329 -0.10931904 -0.5165914 -0.06260916
## 7  0.25921133 -0.54714071 -0.1266922  0.31250813
##
## Clustering vector:
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  6  2  6  6  6  6  6  6  6  6  6  7  4  4  5  4  1  4  4  4  7  4  4  1  7  7
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
##  1  7  7  7  7  7  1  7  7  2  7  1  1  1  4  1  4  4  1  1  1  1  1  4  1  1
## 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
##  7  7  4  4  7  7  7  7  7  4  2  2  2  2  3  3  3  3  6  6  6  6  6  2  6  6
## 79 80 81 82 83 84 85
##  6  6  6  6  6  6  6
##
## Within cluster sum of squares by cluster:
## [1]  756.1569  427.1200  278.8521  718.4455    0.0000 1108.9977 1105.4565
##  (between_SS / total_SS =  25.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Print just the centroids

```
model.r$centers
```

```
##             a        all        also         an         and        any
## 1  0.57333261 -0.5543919  0.01678964  0.6708819 -0.65235612  0.4325217
```

```
## 2 -0.79362463 -0.4524361  0.14251208 -0.7845138  1.24728209 -0.9735980
## 3 -1.98551138 -0.6065787  1.52715783 -1.3208856  2.84844078  0.1377340
## 4  0.40319151  0.4554941 -0.36377557  0.1713932 -0.10326860 -0.5535177
## 5  1.27299447 -0.3315400  0.87306137 -1.2955737 -1.00114025  0.1487062
## 6 -0.04049915  0.1180427  0.39239958 -0.4391868  0.04586718 -0.2938411
## 7 -0.07639257  0.2408079 -0.64168833  0.5251089 -0.46789679  0.7885616
##           are         as         at         be        been        but
## 1 -0.70837900  0.0295533  0.4653143  0.3916831 -0.06853399 -0.4393074
## 2 -0.09327752 -0.2899038  0.5456169 -1.6483805  0.26434540 -0.5177788
## 3  0.47690082  0.9617138 -0.8019940  0.1195023 -1.31812527  1.0261202
## 4 -0.10074167 -0.3841929  0.4882060 -0.4983905  0.23329965 -0.2248546
## 5 -0.49551242 -0.8422692 -1.1167544  0.8644385 -0.89555817 -0.1307963
## 6  0.26040583  0.2721367 -0.3982103  0.1993856  0.02880736  0.1960561
## 7  0.26989062 -0.1151023 -0.2231761  0.3690301  0.06077789  0.2583078
##           by        can          do        down       even      every
## 1 -0.5439792 -0.3378126  0.01134333  0.53700864  0.3376953 -0.3144883
## 2  0.5851017 -1.0478282 -0.51787001 -0.42395419 -0.3444492 -0.6408493
## 3  0.4329801  0.2029611  0.60761453 -0.42395419 -0.1924863 -1.1253318
## 4 -0.2343270 -0.2161832  0.03164193 -0.10979326  0.4795273 -0.4561362
## 5 -1.0578124  0.8737112  1.33075194 -0.42395419 -1.1549178  1.1492069
## 6  0.6873613  0.0375094 -0.10283626 -0.08900320 -0.1080625  0.6952071
## 7 -0.5048367  0.6873133  0.08879849  0.04291034 -0.2804648  0.1427611
##          for.        from        had          has         have         her
## 1  0.46577490  0.08396753  0.1763074 -0.110600733 -0.014287599 -0.35522174
## 2 -0.06241046  0.38007221  2.0743247 -0.062840815  0.493365873  0.12202513
## 3 -0.05732915  0.84229929 -0.4182510 -0.859476253 -0.613590249  0.48461743
## 4 -0.53395613 -0.09201921 -0.0902443  0.651647564  0.323645173  0.08565501
## 5 -1.81444652 -0.17512406 -0.9402480 -0.578647933 -0.917201724  6.60876813
## 6  0.09561832 -0.25761785 -0.1702098 -0.145704273 -0.195574450 -0.00244272
## 7  0.06360606  0.02363289 -0.4796280 -0.008545329 -0.001503958 -0.27890987
##          his         if.        in.        into         is         it
## 1  1.0280766  0.48736590  0.5192723 -0.37952535  0.1560568  0.5143446
## 2  0.3213415 -0.49273111 -0.4514745  0.59718401 -1.0668858 -0.4594612
## 3 -0.4086342  1.76705416 -0.7022708  0.97069729 -1.2695699  0.6582167
## 4 -0.0507305 -0.11418907 -0.1301616 -0.20955512 -0.2427996 -0.7496485
## 5 -0.6732398  0.97820523 -1.0842221  4.20967065 -0.3635961 -1.4887126
## 6 -0.1979296 -0.36422189 -0.5101272  0.02545472  0.3173087 -0.1373554
## 7 -0.5184997 -0.07650691  0.7084221 -0.21302390  0.3471480  0.4683243
##          its         may        more       must          my         no
## 1 -0.2563023 -0.02454088  0.04264529 -0.3395315 -0.09861408 -0.2650528
## 2 -0.3624960 -1.23597444 -0.38085197 -0.7022372  0.03891560 -0.7988284
## 3 -0.5915960  0.01443391  1.86224792 -0.6197602 -0.40568466 -0.8248239
## 4 -0.0727024 -0.58446033 -0.42152685 -0.1315546 -0.05711806 -0.3863654
## 5 -0.6167850  1.91651322  1.59307593  0.4842733 -0.40568466  0.6569232
## 6 -0.1619835  0.27518071  0.24721600  0.2074209 -0.22413956  0.3536416
## 7  0.7549113  0.48464490 -0.34874023  0.4736115  0.49849187  0.5009498
##           not        now          of         on        one        only
## 1  0.09691833 -0.4075517  0.50271004 -0.7801919 -0.1654591 -0.15116370
## 2 -0.66820994  0.7774954 -0.64350851  0.3017094  0.1110367 -0.44030490
## 3  0.79251208 -0.6018273 -2.45765222  0.1688563  1.7433358  1.38956900
## 4 -0.46698253  0.3508250  0.87972605 -0.4489572 -0.3877567 -0.28306407
## 5 -0.60853789 -0.8475975 -0.18839232 -0.6310496  4.1886087 -1.56113312
## 6  0.25297731 -0.2741336 -0.20103444  1.0305434  0.1224162  0.22407009
## 7  0.08397193  0.2759235 -0.07306073 -0.4448498 -0.3462611  0.01161715
```

```
##           or        our       shall      should          so        some
## 1  0.3238570 -0.3979747 -0.06337197  0.51468154  0.46330021 -0.14003062
## 2 -0.4666689 -0.3263276 -0.53646395 -0.46242227 -0.14907485  0.80950286
## 3  1.8755810  1.5457052  0.03881788  0.83186670  0.92910244 -0.12665691
## 4 -0.0401231  0.6456936  0.10983116 -0.05557687  0.07737943 -0.08207787
## 5 -1.1483389 -0.7005166  1.36382183 -1.34456335  0.94466376 -1.33029091
## 6 -0.2032483 -0.1345195 -0.20150147 -0.33437925 -0.02013818  0.35814012
## 7 -0.1297553 -0.1939646  0.33554219  0.12592549 -0.59181199 -0.47859668
##         such        than        that         the       their        then
## 1  0.0867456  0.26767087 -0.03817698  0.3620776 -0.17126459 -0.2877954
## 2 -0.2366983 -0.45208682 -0.57153517 -0.4549213  1.39086872  0.3390491
## 3  1.8349169  0.87878924  0.12322121 -2.3814309  1.09831190  0.2744058
## 4 -0.3740904 -0.29988509 -0.41505741 -0.3200366 -0.41789314 -0.3316745
## 5 -0.8549627  3.95927391  0.98998928 -1.2153557 -1.69427594 -1.0008309
## 6 -0.2388450 -0.06402576 -0.22619021  0.3883298 -0.02973344  0.1235713
## 7  0.2744545 -0.12052588  0.77605015  0.2092105 -0.15179392  0.2029579
##        there      things        this          to          up        upon
## 1  0.2399095 -0.40896320  0.17874661  0.7228951  0.09571931  0.75040160
## 2 -0.8204129 -0.60673314 -0.56640730 -1.2219466  0.05510862 -0.72190069
## 3 -0.4025465 -0.20738998 -1.53353242 -0.3673547 -0.51204063 -1.06575048
## 4  0.1310699  1.00585259  0.36708582  0.2777124  0.56624313  0.46594405
## 5  3.5655645 -0.60673314 -1.96418610  1.0574218  1.69353979  0.02831704
## 6 -0.7105475 -0.15984913 -0.07792298 -0.7751191 -0.29148259 -1.01560572
## 7  0.8039979  0.02981385  0.30241221  0.6610193 -0.15605221  0.81143905
##           was        were       what        when      which        who
## 1  0.06762907  0.23208127  0.4007955  0.74383380 -0.5223756  0.7905275
## 2  2.01689658  1.28311817 -0.4043500  0.05186294 -0.6480435  0.6561838
## 3 -0.42460524  0.02522827  0.5051132  0.60425401 -1.3590738  0.5815484
## 4 -0.12283922 -0.07140962 -0.3365740 -0.33256531  0.2411455 -0.6107046
## 5 -0.88930196 -0.96910375 -1.1025238  2.50932955  0.1683554  0.5011718
## 6 -0.07177995 -0.06661766 -0.1022163 -0.23511095  0.2709584 -0.2942217
## 7 -0.47261484 -0.46973092  0.1790732 -0.30609214  0.3957734 -0.1608776
##          will        with      would        your
## 1 -0.07756304  0.01496527  0.5594458 -0.20876463
## 2 -1.01521083  0.22932210 -0.7293996 -0.20876463
## 3  0.82819663  0.58673800  0.5866929  0.61658391
## 4 -0.55281512  0.40696383  0.5147784 -0.13998559
## 5  0.55138758  2.73818703  1.4510068 -0.20876463
## 6  0.32387329 -0.10931904 -0.5165914 -0.06260916
## 7  0.25921133 -0.54714071 -0.1266922  0.31250813
```

Get cluster assignment

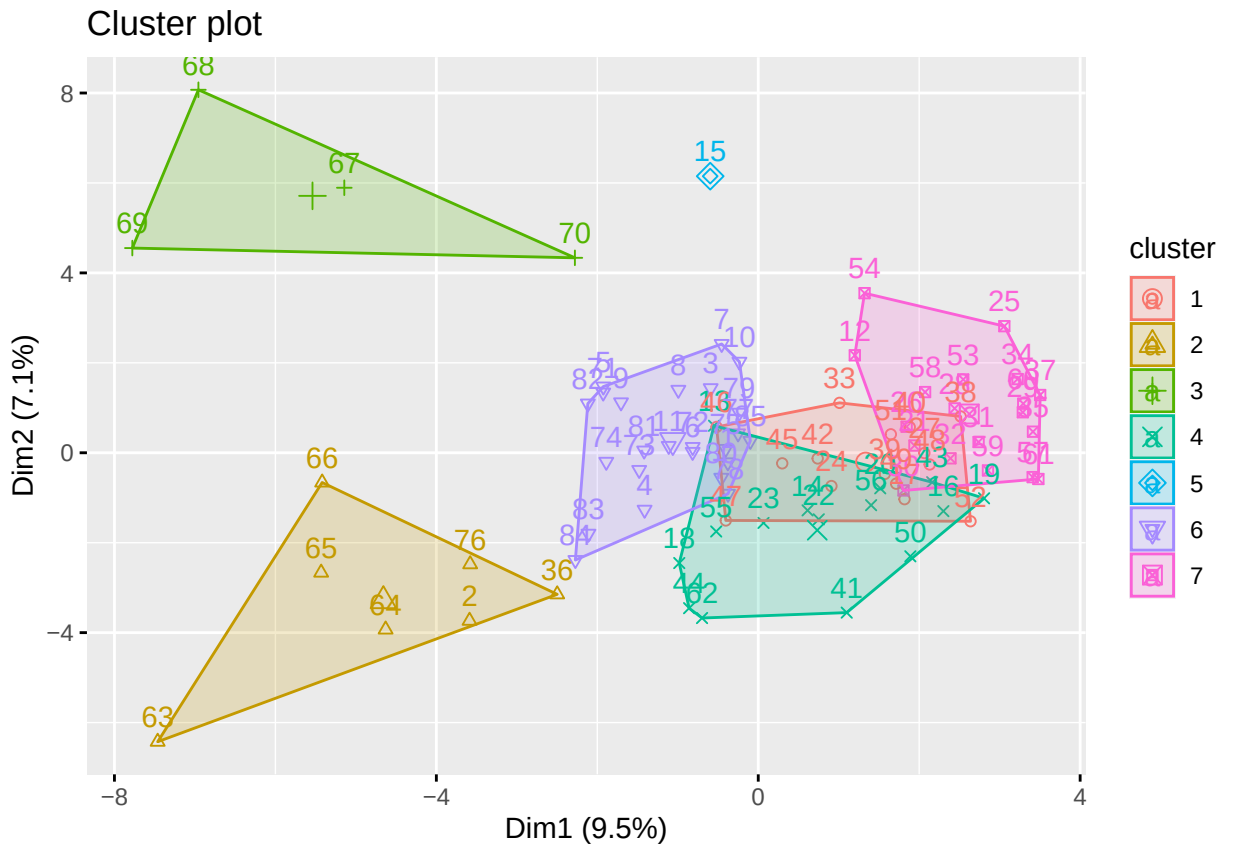
```
cluster.assignment = data.frame(Essay, model.r$cluster) |> relocate(model.r.cluster, .after = author)
glimpse(cluster.assignment)
```

```
## Rows: 85
## Columns: 73
## $ author          <chr> "dispt", "dispt", "dispt", "dispt", "dispt", "dispt", ~
## $ model.r.cluster <int> 6, 2, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 4, 4, 5, 4, 1, 4, ~
## $ filename        <chr> "dispt_fed_49.txt", "dispt_fed_50.txt", "dispt_fed_51.~
## $ a               <dbl> 0.280, 0.177, 0.339, 0.270, 0.303, 0.245, 0.349, 0.414~
## $ all             <dbl> 0.052, 0.063, 0.090, 0.024, 0.054, 0.059, 0.036, 0.083~
```

```
## $ also           <dbl> 0.009, 0.013, 0.008, 0.016, 0.027, 0.007, 0.007, 0.009~
## $ an             <dbl> 0.096, 0.038, 0.030, 0.024, 0.034, 0.067, 0.029, 0.018~
## $ and            <dbl> 0.358, 0.393, 0.301, 0.262, 0.404, 0.282, 0.335, 0.478~
## $ any            <dbl> 0.026, 0.063, 0.008, 0.056, 0.040, 0.052, 0.058, 0.046~
## $ are            <dbl> 0.131, 0.051, 0.068, 0.064, 0.128, 0.111, 0.087, 0.110~
## $ as             <dbl> 0.122, 0.139, 0.203, 0.111, 0.148, 0.252, 0.073, 0.074~
## $ at             <dbl> 0.017, 0.114, 0.023, 0.056, 0.013, 0.015, 0.116, 0.037~
## $ be             <dbl> 0.411, 0.393, 0.474, 0.365, 0.344, 0.297, 0.378, 0.331~
## $ been           <dbl> 0.026, 0.165, 0.015, 0.127, 0.047, 0.030, 0.044, 0.046~
## $ but            <dbl> 0.009, 0.000, 0.038, 0.032, 0.061, 0.037, 0.007, 0.055~
## $ by             <dbl> 0.140, 0.139, 0.173, 0.167, 0.209, 0.186, 0.102, 0.092~
## $ can            <dbl> 0.035, 0.000, 0.023, 0.056, 0.088, 0.000, 0.058, 0.037~
## $ do             <dbl> 0.026, 0.013, 0.000, 0.000, 0.000, 0.000, 0.015, 0.028~
## $ down           <dbl> 0.000, 0.000, 0.008, 0.000, 0.000, 0.007, 0.000, 0.000~
## $ even           <dbl> 0.009, 0.025, 0.015, 0.024, 0.020, 0.007, 0.007, 0.018~
## $ every          <dbl> 0.044, 0.000, 0.023, 0.040, 0.027, 0.007, 0.087, 0.064~
## $ for.           <dbl> 0.096, 0.076, 0.098, 0.103, 0.141, 0.067, 0.116, 0.055~
## $ from           <dbl> 0.044, 0.101, 0.053, 0.079, 0.074, 0.096, 0.080, 0.083~
## $ had            <dbl> 0.035, 0.101, 0.008, 0.016, 0.000, 0.022, 0.015, 0.009~
## $ has            <dbl> 0.017, 0.013, 0.015, 0.024, 0.054, 0.015, 0.036, 0.037~
## $ have           <dbl> 0.044, 0.152, 0.023, 0.143, 0.047, 0.119, 0.044, 0.074~
## $ her            <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.007, 0.000~
## $ his            <dbl> 0.017, 0.000, 0.000, 0.024, 0.020, 0.067, 0.000, 0.018~
## $ if.            <dbl> 0.000, 0.025, 0.023, 0.040, 0.034, 0.030, 0.029, 0.000~
## $ in.            <dbl> 0.262, 0.291, 0.308, 0.238, 0.263, 0.401, 0.189, 0.267~
## $ into           <dbl> 0.009, 0.025, 0.038, 0.008, 0.013, 0.037, 0.000, 0.037~
## $ is             <dbl> 0.157, 0.038, 0.150, 0.151, 0.189, 0.260, 0.167, 0.083~
## $ it             <dbl> 0.175, 0.127, 0.173, 0.222, 0.108, 0.156, 0.102, 0.165~
## $ its            <dbl> 0.070, 0.038, 0.030, 0.048, 0.013, 0.015, 0.000, 0.046~
## $ may            <dbl> 0.035, 0.038, 0.120, 0.056, 0.047, 0.074, 0.080, 0.092~
## $ more           <dbl> 0.026, 0.000, 0.038, 0.056, 0.067, 0.045, 0.080, 0.064~
## $ must           <dbl> 0.026, 0.013, 0.083, 0.071, 0.013, 0.015, 0.044, 0.018~
## $ my             <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.007, 0.000~
## $ no             <dbl> 0.035, 0.000, 0.030, 0.032, 0.047, 0.059, 0.022, 0.018~
## $ not            <dbl> 0.114, 0.127, 0.068, 0.087, 0.128, 0.134, 0.102, 0.101~
## $ now            <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.007, 0.000~
## $ of             <dbl> 0.900, 0.747, 0.858, 0.802, 0.869, 0.876, 0.902, 1.029~
## $ on             <dbl> 0.140, 0.139, 0.150, 0.143, 0.054, 0.141, 0.051, 0.083~
## $ one            <dbl> 0.026, 0.025, 0.030, 0.032, 0.047, 0.052, 0.073, 0.046~
## $ only           <dbl> 0.035, 0.000, 0.023, 0.048, 0.027, 0.022, 0.007, 0.046~
## $ or             <dbl> 0.096, 0.114, 0.060, 0.064, 0.081, 0.074, 0.153, 0.037~
## $ our            <dbl> 0.017, 0.000, 0.000, 0.016, 0.027, 0.030, 0.051, 0.000~
## $ shall          <dbl> 0.017, 0.000, 0.008, 0.016, 0.000, 0.015, 0.007, 0.000~
## $ should         <dbl> 0.017, 0.013, 0.068, 0.032, 0.000, 0.030, 0.007, 0.000~
## $ so             <dbl> 0.035, 0.013, 0.038, 0.040, 0.027, 0.007, 0.051, 0.018~
## $ some           <dbl> 0.009, 0.063, 0.030, 0.024, 0.067, 0.045, 0.007, 0.028~
## $ such           <dbl> 0.026, 0.000, 0.045, 0.008, 0.027, 0.015, 0.015, 0.000~
## $ than           <dbl> 0.009, 0.000, 0.023, 0.000, 0.047, 0.030, 0.109, 0.055~
## $ that           <dbl> 0.184, 0.152, 0.188, 0.238, 0.162, 0.208, 0.233, 0.165~
## $ the            <dbl> 1.425, 1.254, 1.490, 1.326, 1.193, 1.469, 1.259, 1.176~
## $ their          <dbl> 0.114, 0.165, 0.053, 0.071, 0.027, 0.089, 0.109, 0.083~
## $ then           <dbl> 0.000, 0.000, 0.015, 0.008, 0.007, 0.007, 0.015, 0.009~
## $ there          <dbl> 0.009, 0.000, 0.015, 0.000, 0.007, 0.007, 0.036, 0.028~
## $ things         <dbl> 0.009, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000~
```

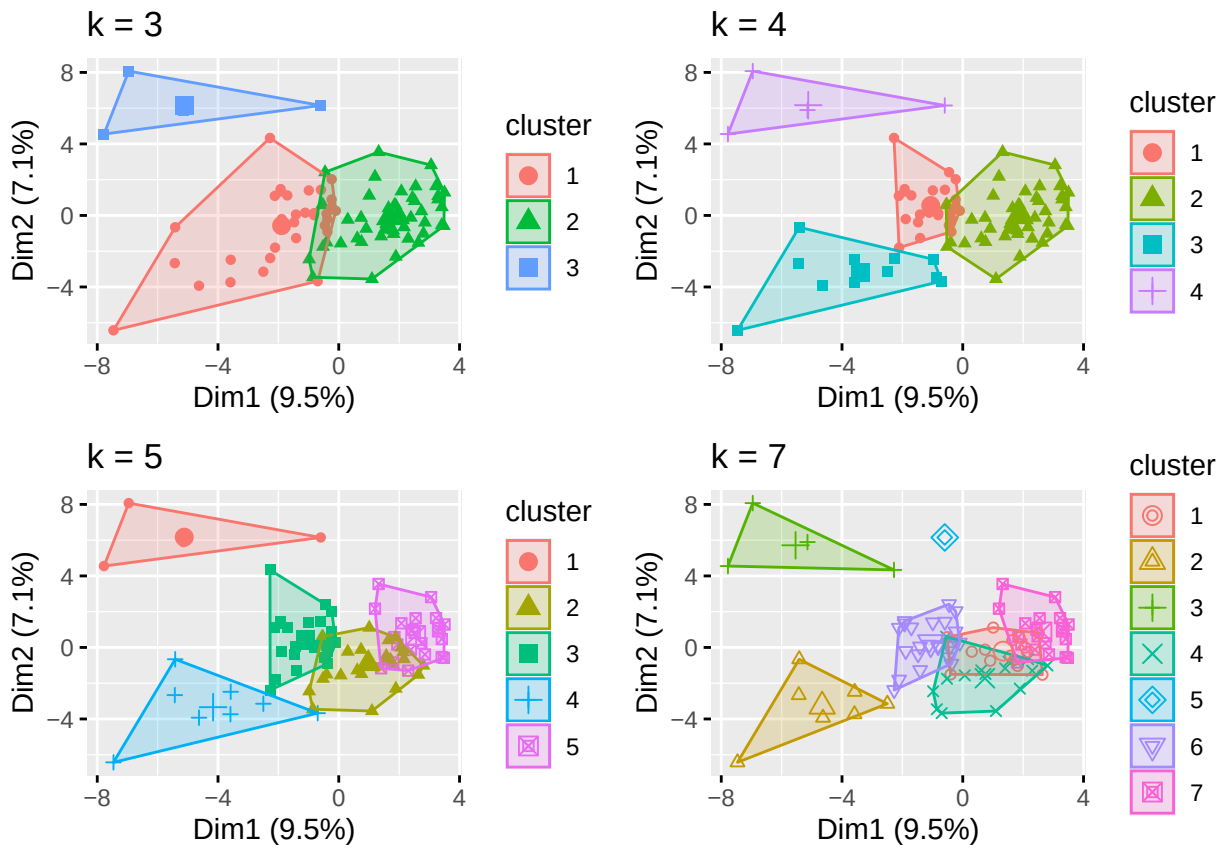
```
## $ this            <dbl> 0.044, 0.051, 0.075, 0.103, 0.094, 0.126, 0.080, 0.110~
## $ to              <dbl> 0.507, 0.355, 0.361, 0.532, 0.485, 0.445, 0.560, 0.340~
## $ up              <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.007, 0.000~
## $ upon            <dbl> 0.000, 0.013, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000~
## $ was             <dbl> 0.009, 0.051, 0.008, 0.087, 0.027, 0.007, 0.015, 0.018~
## $ were            <dbl> 0.017, 0.000, 0.015, 0.079, 0.020, 0.030, 0.029, 0.009~
## $ what            <dbl> 0.000, 0.000, 0.008, 0.008, 0.020, 0.015, 0.015, 0.009~
## $ when            <dbl> 0.009, 0.000, 0.000, 0.024, 0.007, 0.037, 0.007, 0.000~
## $ which           <dbl> 0.175, 0.114, 0.105, 0.167, 0.155, 0.186, 0.211, 0.175~
## $ who             <dbl> 0.044, 0.038, 0.008, 0.000, 0.027, 0.045, 0.022, 0.018~
## $ will            <dbl> 0.009, 0.089, 0.173, 0.079, 0.168, 0.111, 0.145, 0.267~
## $ with            <dbl> 0.087, 0.063, 0.045, 0.079, 0.074, 0.089, 0.073, 0.129~
## $ would           <dbl> 0.192, 0.139, 0.068, 0.064, 0.040, 0.037, 0.073, 0.037~
## $ your            <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000~
```

```
fviz_cluster(model.r, data = Essay.unlabeled)
```




```
k3 = kmeans(Essay.unlabeled, centers = 3, nstart = 25)
k4 = kmeans(Essay.unlabeled, centers = 4, nstart = 25)
k5 = kmeans(Essay.unlabeled, centers = 5, nstart = 25)

# plots to compare
p1 = fviz_cluster(k3, geom = "point",  data = Essay.unlabeled) + ggtitle("k = 3")
p2 = fviz_cluster(k4, geom = "point",  data = Essay.unlabeled) + ggtitle("k = 4")
p3 = fviz_cluster(k5, geom = "point",  data = Essay.unlabeled) + ggtitle("k = 5")
```

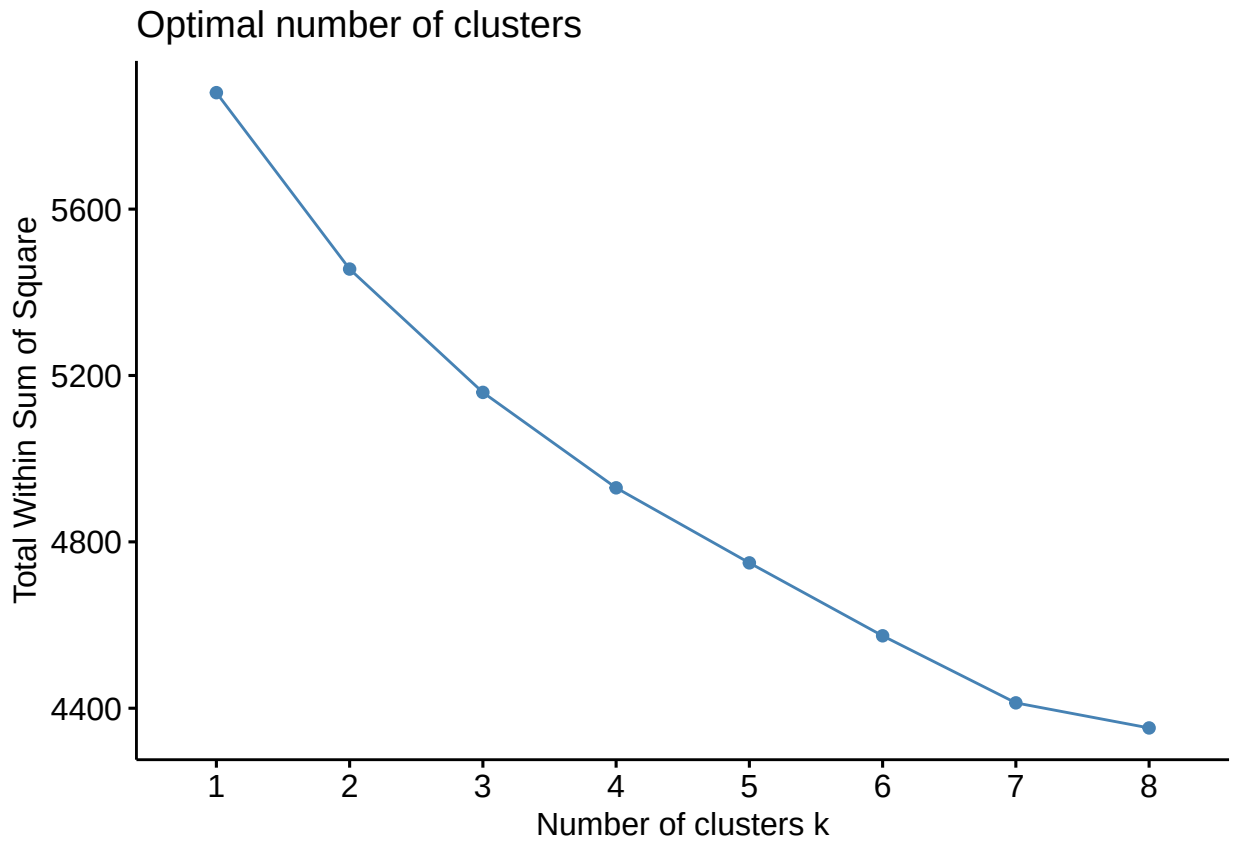
```
p4 = fviz_cluster(model.r, geom = "point", data = Essay.unlabeled) + ggtitle("k = 7")

library(gridExtra)
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



Elbow method

Fortunately, this process to compute the "Elbow method" has been wrapped up in a single function (fviz_nbclust):

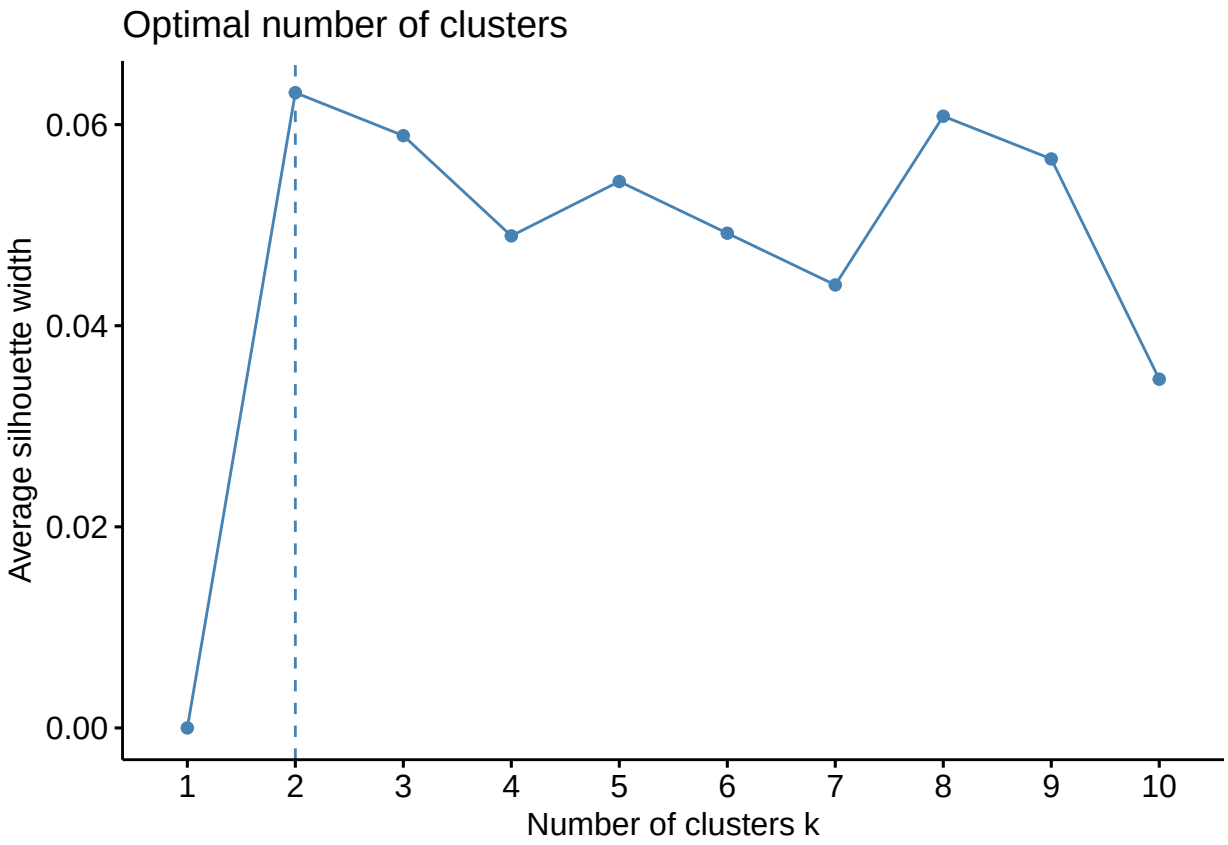
```
set.seed(123)
fviz_nbclust(Essay.unlabeled, kmeans, method = "wss", k.max = 8)
```



Average Silhouette Method

Similar to the elbow method, this process to compute the "average silhoutte method" has been wrapped up in a single function

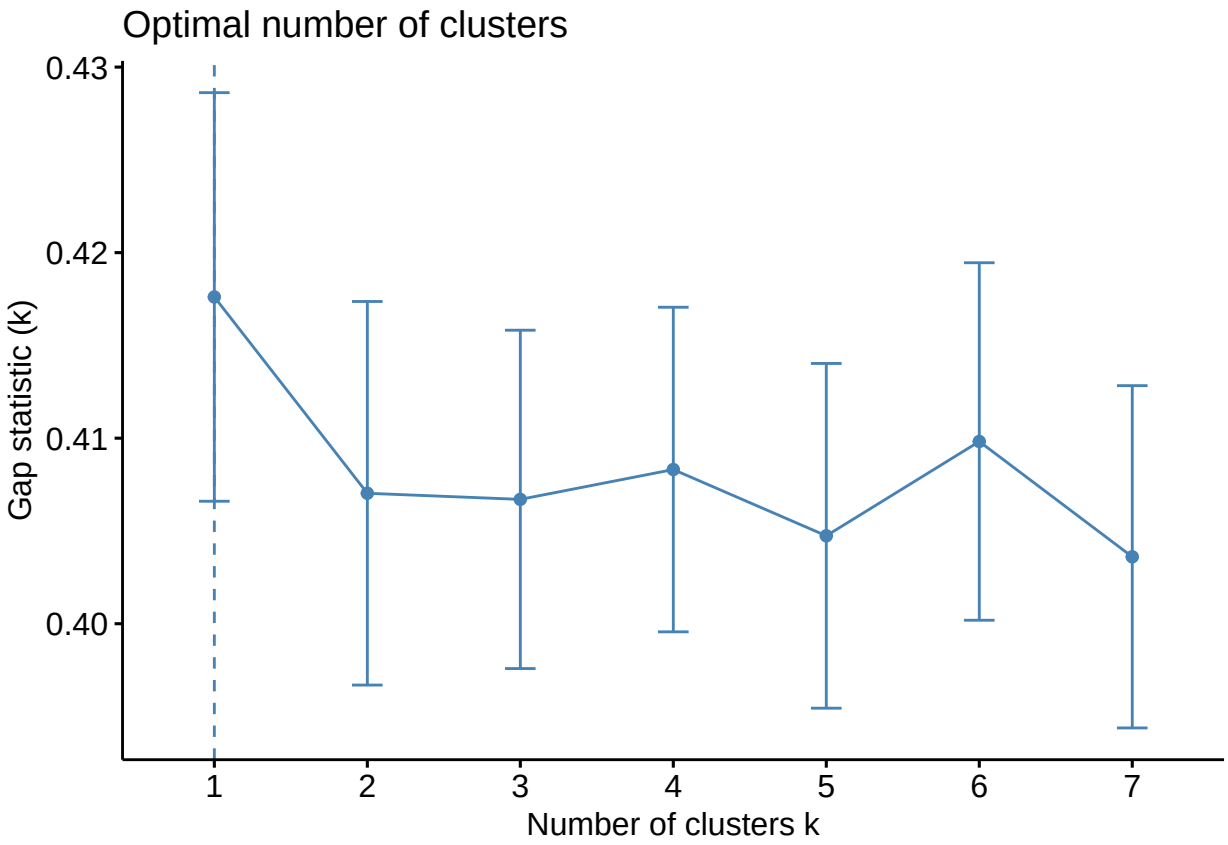
```
set.seed(123)
fviz_nbclust(Essay.unlabeled, kmeans, method = "silhouette")
```



according to silhouette method, the number of clusters k should be 2 or 8. We can take 2 as we want to decide between two authors.

Gap Statistic Method

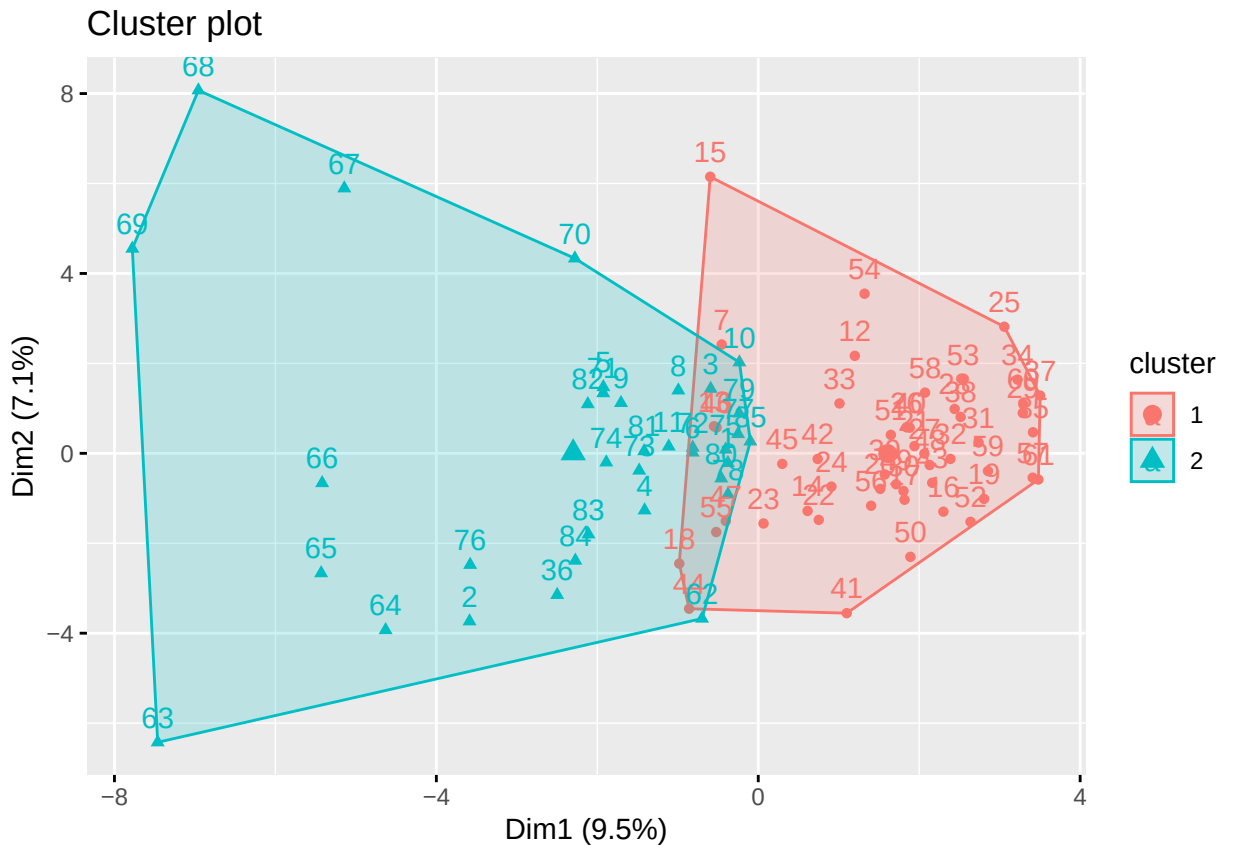
```
set.seed(123)
gap.stat = clusGap(Essay.unlabeled, FUN = kmeans, K.max = 7, B = 50)

set.seed(123)
fviz_gap_stat(gap.stat)
```



2, 5 and 7 clusters offer least gap statistic. We can use 2.

Extracting Results

```
set.seed(123)

final.res = kmeans(Essay.unlabeled, 2, nstart = 25)

fviz_cluster(final.res, data = Essay.unlabeled)
```




```
Essay.unlabeled = as.data.frame(Essay.unlabeled)
Essay.unlabeled$cluster = final.res$cluster

# Essay.unlabeled |> group_by(cluster) |> summarise_all(mean)
```

Let's inspect the clusters

```
Essay$cluster = final.res$cluster
Essay |>
  count(cluster) |>
  left_join(
    Essay |>
      select(cluster, author) |>
      group_by(cluster) |>
      slice(1:10) |> # just preview 10
      mutate(authors = str_c(author, collapse = ', ')) |> # collapse all animal names into one vector
      select(-author) |>
      ungroup() |>
      distinct()
  )
```

```
## Joining, by = "cluster"

##   cluster  n
## 1       1 50
```

```
## 2        2 35
##                                                                                  authors
## 1 dispt, Hamilton, Hamilton, Hamilton, Hamilton, Hamilton, Hamilton, Hamilton, Hamilton, Hamilton
## 2                               dispt, dispt, dispt, dispt, dispt, dispt, dispt, dispt, dispt, dispt
```

Conclusion

We observe that the documents in the left cluster have features that are similar to those of Hamilton, while the documents in the right cluster have traits that are more similar to those of other. We can suspect that the other author is Madison. Since the cluster with Madison includes most disputed documents, this suggests that Madison was the author for most of the disputed documents but Hamilton was the author for one of the disputed documents.