

Business Analytics Assignment 7

PART 1:

1. Find out the list of successful data-driven companies and select 30 organizations.

- I found the list of top data-driven companies from

<http://www.forbes.com/sites/louiscolumbus/2016/05/05/the-best-big-data-and-business-analytics-companies-to-work-for-in-2016/> - 43c758974434

2. Create a corpus of their mission statements.

The screenshot shows an RStudio interface. The script pane contains the following R code:

```
1 library(NLP)
2 library(tm)
3 part1<-read.csv("part1.csv",header=TRUE, stringsAsFactors=FALSE)
4 part1
5 dim(part1)
6 names(part1)
7 str(part1)
8 corpus_mission<-corpus(part1$mission, docnames=part1$company)
9 names(corpus_mission)
10 summary(corpus_mission)
11 head(corpus_mission)
12 #clean corpus: removes punctuation, digits, converts to lower case
13 newcorpus_mission<- toLower(corpus_mission, keepAcronyms = FALSE)
14 cleancorpus_mission <- tokenize(corpus_mission,
15                                   removeNumbers=TRUE,
16                                   removePunct = TRUE,
17                                   removeSeparators=TRUE,
18                                   removeTwitter=FALSE,
19                                   verbose=TRUE)
20 head(cleancorpus_mission)
21
```

The console pane shows the execution of the script and its output:

```
...tokenizing texts...total elapsed: 0.0020000000768341 seconds.
...replacing Twitter characters (#, @)...total elapsed: 0.0009999998928979 seconds.
...replacing names...total elapsed: 0 seconds.
Finished tokenizing and cleaning 30 texts.
> head(cleancorpus_mission)
$shortwords
[1] "Our"          "mission"       "is"            "simple"        "To"           "revolutionize"
[7] "and"          "commoditize"  "the"          "storage"       "and"          "processing"
[13] "of"           "big"          "data"          "via"          "open"         "source"
[19] "In"           "order"        "to"           "accomplish"   "this"         "mission"
[25] "we"           "are"          "focused"      "on"           "accelerating" "the"
[31] "development"  "and"          "adoption"     "of"           "Apache"       "Hadoop"
[37] "By"           "making"       "it"           "easier"       "to"           "consume"
[43] "for"          "enterprises" "and"          "technology"   "vendors"      "we"
[49] "believe"      "we"          "can"          "overcome"     "any"          "technology"
[55] "and"          "knowledge"   "gaps"         "that"         "could"       "keep"
[61] "Apache"        "Hadoop"       "from"         "reaching"     "its"          "potential"
```

3. Create a corpus of their core values.

The screenshot shows the RStudio interface with the following code in the script pane:

```

1 library(NLP)
2 library(tm)
3 part1<-read.csv("part1.csv",header=TRUE, stringsAsFactors=FALSE)
4 part1
5 dim(part1)
6 names(part1)
7 str(part1)
8 corpus_values<-corpus(part1$core.values, docnames=part1$core.values)
9 names(corpus_values)
10 summary(corpus_values)
11 head(corpus_values)
12 #clean corpus: removes punctuation, digits, converts to lower case
13 newcorpus_values<- tolower(corpus_values, keepAcronyms = FALSE)
14 cleancorpus_values <- tokenize(corpus_values,
15                               removeNumbers=TRUE,
16                               removePunct = TRUE,
17                               removeSeparators=TRUE,
18                               removeTwitter=FALSE,
19                               verbose=TRUE)
20 head(cleancorpus_values)
21
21:1 (Top Level) <

```

Console output:

```

Starting tokenization...
...preserving Twitter characters (#, @)...total elapsed: 0.0010000001839362 seconds.
...tokenizing texts...total elapsed: 0.002999999969732 seconds.
...replacing Twitter characters (#, @)...total elapsed: 0.0009999998928979 seconds.
...replacing names...total elapsed: 0 seconds.
Finished tokenizing and cleaning 30 texts.
> head(cleancorpus_values)
$...
[1] "We"          "believe"      "strongly"     "in"           "our"
[6] "core"        "set"          "of"           "shared"       "values"
[11] "These"       "values"       "shape"        "who"          "we"
[16] "are"         "and"          "how"          "we"          "make"
[21] "decisions"   "on"           "a"            "daily"        "basis"
[26] "At"          "Hortonworks" "we"           "are"          "Passionate"

```

The right pane shows the 'Environment' and 'History' tabs, and the 'System Library' pane listing various R packages.

4. Analyze the corpus and provide insight on how to structure a firm for data analysis readiness.

- Performed frequency analysis to find out the most frequent text usage

MISSION:

The screenshot shows the RStudio interface with the following code in the script pane:

```

19
20 head(cleancorpus_mission)
21
22 #create document feature matrix from clean corpus + stem
23 help(dfm)
24 dfm.simple<-dfm(cleancorpus_mission,
25                   tolower = TRUE,
26                   ignoredFeatures =stopwords("english"),
27                   verbose=TRUE,
28                   stem=FALSE)
29 head(dfm.simple)
30 dfm.simple
31 #to display most frequent terms in dfm
32 topfeatures<-topfeatures(dfm.simple, n=50)
33 topfeatures
34
34:1 (Top Level) <

```

Console output:

```

microsoft    2    0    0    0    0    0    0
> dfm.simple
Document-feature matrix of: 30 documents, 1,012 features.
> topfeatures<-topfeatures(dfm.simple, n=50)
> topfeatures
   data      s business customers software world analytics
    70      27      23      23     20      19      19
platform technology make products people mission can
    17      16      16      15     15      14      14
create    help company services enterprise cloud intelligence
    13      13      12      12     11      11      11
around    new customer use big power provide
    10      10      10      9      8      8      8
need    businesses vision way machine solutions believe
     8      8      8      8      8      8      7
build   information insights search technologies re organizations
     7      7      7      7      7      7      7
us      every mapr will mongodb google success
     7      7      7      6      6      6      6
better
     6

```

The right pane shows the 'Environment' and 'History' tabs, and the 'System Library' pane listing various R packages.

Frequency analysis using stop word list

```

28 head(dfm.simple)
29 dfm.simple
30 #to display most frequent terms in dfm
31 topfeatures<-topfeatures(dfm.simple, n=50)
32 topfeatures
33
34
35 #to create a custom dictionary list of stop words
36 swlist = c("will", "must", "can", "just", "s", "ever", "said", "will", "us", "take", "one")
37 dfm.stem<- dfm(cleancorpus.mission, tolower = TRUE,
38 ignoredFeatures = c(swlist, stopwords("english")),
39 verbose=TRUE,
40 stem=FALSE)
41 topfeatures.stem<-topfeatures(dfm.stem, n=50)
42 topfeatures.stem
43
42:17 (Top Level) <

```

Console ~/Desktop/Business Analytics/

```

... removed 63 features, from 101 supplied (gives 38 feature types)
... created a 30 x 1005 sparse dfm
... complete.
Elapsed time: 0.015 seconds.
> topfeatures.stem<-topfeatures(dfm.stem, n=50)
> topfeatures.stem
      data business customers software world analytics platform
technology      make products    people mission create help
       70        23        23       20     19      19      17
       16        16        15       15     14      13      13
company services enterprise cloud intelligence around new
       12        12        11       11     11      10      10
customer use big power provide need businesses
       10         9        8       8      8      8      8
vision way machine solutions believe build information
       8          8        8       8      7      7      7
insights search technologies re_organizations every mapr
       7          7        7       7      7      7      7
mongodb google success better time everyone enable
       6          6        6       6      6      6      6
oracle
       6

```

Frequency analysis using bigrams

```

42 topfeatures.stem
43
44 #dfm with bigrams
45 cleancorpus.mission <- tokenize(newcorpus.mission,
46 removeNumbers=TRUE,
47 removePunct = TRUE,
48 removeSeparators=TRUE,
49 removeTwitter=FALSE,
50 ngrams=2, verbose=TRUE)
51 dfm.bigram<- dfm(cleancorpus.mission, tolower = TRUE,
52 ignoredFeatures = c(swlist, stopwords("english")),
53 verbose=TRUE,
54 stem=FALSE)
55 topfeatures.bigram<-topfeatures(dfm.bigram, n=50)
56 topfeatures.bigram
57:1 (Top Level) <

```

Console ~/Desktop/Business Analytics/

```

> topfeatures.bigram<-topfeatures(dfm.bigram, n=50)
> topfeatures.bigram
      big_data enterprise_data digital_transformation business_intelligence
      8                  4                      4                  4
open_source apache_hadoop data_platform data_management
      3                  3                      3                  3
means_making customer_success machine_data machine_intelligence
      3                  3                      3                  3
five_years business_challenges data_hub cloudera_enterprise
      2                  2                      2                  2
products_work help_businesses cloud_computing create_technology
      2                  2                      2                  2
communities_around oracle_cloud way_people solve_problems
      2                  2                      2                  2
global_leader machine_learning splunk_platform predictive_analytics
      2                  2                      2                  2
giving_customers online_shopping shopping_experience won_t
      2                  2                      2                  2
t_stop stop_pushing pushing_ahead management_software
      2                  2                      2                  2
converged_data core_values value_statements advanced_analytics
      2                  2                      2                  2
data_analytics analytics_solutions data_integration services_sas
      2                  ?                      ?                  ?

```

CORE VALUES:

The screenshot shows an RStudio interface with the following details:

- Code Area:**

```

19 head(cleancorpus_values)
20 verbose=TRUE)
21
22 #create document feature matrix from clean corpus + stem
23 help(dfm)
24 dfm.simple<-dfm(cleancorpus_values,
25   tolower = TRUE,
26   ignoredFeatures =stopwords("english"),
27   verbose=TRUE,
28   stem=FALSE)
29 head(dfm.simple)
30 dfm.simple
31 #to display most frequent terms in dfm
32 topfeatures<-topfeatures(dfm.simple, n=50)
33 topfeatures
34
34:1 (Top Level) <

```
- Console Output:**

```

> cleancorpus
Document-feature matrix of: 30 documents, 1,295 features.
> topfeatures<-topfeatures(dfm.simple, n=50)
> topfeatures
      work    customers     people    success     open employees
41       39           32        32       31       30        29
every    company     values     take business customer
27       25           23        23       22       22        21
make    leaders    believe    can innovation ideas
20       20           19        19       19       19        18
will    core      us great strive always
18       17           17        17       15       15        15
respect trust      s integrity time team
15       15           14        14       14       14        13
organization best standards others environment value
13       13           13        13       13       13        13
oracle   community excellence responsibility data committed
13       12           12        12       12       12        11
new     personal everything one technology act
11       11           11        11       11       11        11
quality  splunk
11       11           11

```
- Environment View:** Shows the current session's environment with various objects and their descriptions.
- System Library:** Shows the installed R packages and their versions.

Frequency Analysis using stop word list

The screenshot shows an RStudio interface with the following details:

- Code Area:**

```

28 head(dfm.simple)
29 stem=FALSE)
30 dfm.simple
31 #to display most frequent terms in dfm
32 topfeatures<-topfeatures(dfm.simple, n=50)
33 topfeatures
34
35 #to create a custom dictionary list of stop words
36 swlist = c("will", "must", "can", "just", "s", "ever", "said", "will", "us", "take", "one")
37 dfm.swlist<- dfm(cleancorpus_values, tolower = TRUE,
38   ignoredFeatures = c(swlist, stopwords("english")),
39   verbose=TRUE,
40   stem=FALSE)
41 topfeatures.swlist<-topfeatures(dfm.swlist, n=50)
42 topfeatures.swlist
43
43:1 (Top Level) <

```
- Console Output:**

```

... completed.
Elapsed time: 0.04 seconds.
> topfeatures.swlist<-topfeatures(dfm.swlist, n=50)
> topfeatures.swlist
      work    customers     people    success     open employees
41       39           32        32       31       30        29
every    company     values     business customer make
27       25           23        23       22       21        20
leaders  believe innovation ideas core great
20       19           19        19       18       17        15
strive   always respect trust integrity time
15       15           15        15       15       14        14
team    organization best standards others environment
13       13           13        13       13       13        13
value   oracle     community excellence responsibility data
13       13           12        12       12       12        12
committed new     personal everything technology act
11       11           11        11       11       11        11
quality  splunk    decisions partners right improve
11       11           10        10       10       10        10
fun     passion
10       10

```
- Environment View:** Shows the current session's environment with various objects and their descriptions.
- System Library:** Shows the installed R packages and their versions.

Frequency analysis using bigrams

The screenshot shows the RStudio interface with the following components:

- Code Editor:** Displays R code for generating bigram frequency analysis. The code includes tokenization, stemming, and filtering for English stopwords.
- Console:** Shows the output of the code, which is a frequency matrix of bigrams. The matrix includes columns for core_values, open_source, oracle_employees, customer_success, and other business-related terms like work_hard, great_people, etc.
- Environment:** Shows the current environment variables and objects.
- Files:** Shows the system library with various R packages listed.

```

43
44 #dfm with bigrams
45 cleancorpus_values <- tokenize(newcorpus_values,
46   removeNumbers=TRUE,
47   removePunct = TRUE,
48   removeSeparators=TRUE,
49   removeTwitter=FALSE,
50   ngrams=2, verbose=TRUE)
51 dfm.bigram_values<- dfm(cleancorpus_values,
52   ignoredFeatures = c(swlis, stopwords("english")),
53   verbose=TRUE,
54   stem=FALSE)
55 topfeatures.bigram_values<-topfeatures(dfm.bigram_values, n=50)
56 topfeatures.bigram_values
57
57:1 (Top Level) <-
  
```

core_values	open_source	oracle_employees	customer_success
11	10	10	7
right_thing	every_day	don_t	data_integration
6	6	5	5
integration_systems	work_hard	great_people	customers_employees
5	4	4	4
source_data	make_decisions	great_work	share_information
4	3	3	3
big_challenges	customers_success	new_ideas	products_services
3	3	3	3
customers_successful	shared_values	delivering_great	hire_smart
3	2	2	2
t tolerate	community_partners	team_members	always_looking
2	2	2	2
good_ideas	good_listeners	think_big	open_standards
2	2	2	2
work_backwards	business_practices	every_dollar	business_integrity
2	2	2	2
oracle's_business	mutual_respect	consistently_treat	work_together
2	2	2	2
customer_satisfaction	continuous_improvement	never_settle	giving_back
2	2	2	2

INSIGHT:

Mission Statement:

Most of the data driven companies aim to provide excellent big data solutions, business intelligence, digital transformations and analytical solutions.

Core Values:

From the analysis we come to know that most companies are looking out for individuals who:

Work hard, make great decisions, share information, take big challenges, good listeners and help them to successfully deliver the product to the customers.

5. Are there any other data-driven approaches you would recommend the CEO to implement?
- According to me, the CEO should also take into consideration the tools and technologies that the top data driven companies use to provide analytical solutions and business insights.

PART 2:

1. Search for “Donald Trump Speech Transcript” and select three speeches of your choice.

- I selected the following three speech transcripts of Donald Trump for my analysis:
- Transcript1:**

<http://thehill.com/blogs/pundits-blog/campaign/294817-transcript-of-donald-trumps-speech-on-national-security-in>

- Transcript 2:**

<http://www.vox.com/2016/7/21/12253426/donald-trump-acceptance-speech-transcript-republican-nomination-transcript>

- Transcript 3:**

<http://heavy.com/news/2016/08/read-donald-trump-full-transcript-speech-foreign-policy-address-remarks-prepared-august-15/>

2. Create a corpus for the speeches.

```

library(NLP)
library(tm)

datatext<- read.csv("textdata.csv", header=TRUE, stringsAsFactors=FALSE)
datatext
dim(datatext)
names(datatext)
str(datatext)
newcorpus<- corpus(datatext$content,
                     docnames=datatext$id)
names(newcorpus)
summary(newcorpus)

```

```

url<-
precorpus<- read.csv("https://raw.githubusercontent.com/jcbon...
header=TRUE, stringsAsFactors=FALSE)
str(precorpus)
require(quanteda)
newcorpus<- corpus(precorpus$Full.text,
docnames=precorpus$Document_ID,
docvar=data.frame(Year=precorpus$Publication.year, Subject= pr...
names(newcorpus) #to explore the output of the corpus functi...
newcorpus<- corpus(datatext$content,
docnames=datatext$id)
newcorpus<- corpus(datatext$content,
docnames=datatext$id)
names(newcorpus)
summary(newcorpus)

```

Name	Description	Version
acepack	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
assertthat	Easy pre and post assertions.	0.1
BH	Boost C++ Header Files	1.60.0-2
BiocInstaller	Install/Update Bioconductor, CRAN, and github Packages	1.24.0
bitops	Bitwise Operations	1.0-6
boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-18
ca	Simple, Multiple and Joint Correspondence Analysis	0.64
chron	Chronological Objects which can Handle Dates and Times	2.3-47
class	Functions for Classification	7.3-14
cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.0.5
codetools	Code Analysis Tools for R	0.2-15
colorspace	Color Space Manipulation	1.2-7

Console ~/Desktop/Business Analytics/

```

> newcorpus<- corpus(datatext$content,
+                      docnames=datatext$id)
> newcorpus<- corpus(datatext$content,
+                      docnames=datatext$id)
> names(newcorpus)
[1] "documents" "metadata" "settings" "tokens"
> summary(newcorpus)
Corpus consisting of 3 documents.

Text Types Tokens Sentences
T1  958   5514     276
T2 1445   5836     354
T3 1405   4586     213

Source: /Users/Home/Desktop/Business Analytics/* on x86_64 by Home
Created: Mon Nov 7 00:38:19 2016
Notes:

> |

```

```

6 dim(datatext)
7 names(datatext)
8 str(datatext)
9 newcorpus<- corpus(datatext$content,
10                         docnames=datatext$id)
11 names(newcorpus)
12 summary(newcorpus)
13 head(newcorpus)
14
15 #clean corpus: removes punctuation, digits, converts to lower case
16 newcorpus<- toLower(newcorpus, keepAcronyms = FALSE)
17 cleancorpus <- tokenize(newcorpus,
18                         removeNumbers=TRUE,
19                         removePunct = TRUE,
20                         removeSeparators=TRUE,
21                         removeTwitter=FALSE,
22                         verbose=TRUE)
23 head(cleancorpus)
24

```

(Top Level) R Script

Console ~ /Desktop/Business Analytics/

```

> head(cleancorpus)
$T1
[1] "today"          "i"           "am"           "here"
[5] "to"             "talk"         "about"        "three"
[9] "crucial"        "words"        "that"         "should"
[13] "be"             "at"           "the"          "center"
[17] "of"             "our"          "foreign"      "policy"
[21] "peace"          "through"      "strength"    "we"
[25] "want"           "to"           "achieve"     "a"
[29] "stable"          "peaceful"    "world"        "with"
[33] "less"            "conflict"    "and"          "more"
[37] "common"          "ground"      ";"            "am"
[41] "proposing"      "a"           "new"          "foreign"
[45] "policy"          "focused"     "on"           "advancing"
[49] "america"         "s"           "core"         "national"
[53] "interests"       "promoting"   "regional"     "stability"
[57] "and"             "producing"   "on"           "easing"
[61] "of"              "tensions"    "in"           "the"
[65] "world"           "this"        "will"         "require"

```

Environment History

newcorpus<- corpus(datatext\$content, docnames=datatext\$id)
newcorpus<- corpus(datatext\$content, docnames=datatext\$id)
names(newcorpus)
summary(newcorpus)
head(newcorpus)
newcorpus<- toLower(newcorpus, keepAcronym...
cleancorpus <- tokenize(newcorpus, removeNumbers=TRUE,
removePunct = TRUE,
removeSeparators=TRUE,
removeTwitter=FALSE,
verbose=TRUE)
head(cleancorpus)

Files Plots Packages Help Viewer

Install Update

Name Description V...

System Library

- acepack ACE and AVAS for Selecting Multiple Regression Transformations 1.4.1
- assert... Easy pre and post assertions. 0.1
- BH Boost C++ Header Files 1.60
- BiocIn... Install/Update Biocconductor, CRAN, and github Packages 1.24
- bitops Bitwise Operations 1.0- 6
- boot Bootstrap Functions (Originally by Angelo Canty for S) 1.3- 18
- ca Simple, Multiple and Joint Correspondence Analysis 0.64
- chron Chronological Objects 2.3-

3. Complete a frequency analysis of the word usage

• Frequency analysis without stemming:

```

19
20
21
22
23 head(cleancorpus)
24
25 #create document feature matrix from clean corpus + stem
26 help(dfm)
27 dfm.simple<-dfm(cleancorpus,
28                   tolower = TRUE,
29                   ignoredFeatures =stopwords("english"),
30                   verbose=TRUE,
31                   stem=FALSE)
32 head(dfm.simple)
33 dfm.simple
34 #to display most frequent terms in dfm
35 topfeatures<-topfeatures(dfm.simple, n=50)
36 topfeatures
37

```

(Top Level) R Script

Console ~ /Desktop/Business Analytics/

```

> dfm.simple
Document-feature matrix of: 3 documents, 2,257 features.
> topfeatures<-topfeatures(dfm.simple, n=50)
> topfeatures
      will      country      new      america      clinton      one      hillary      isis
246      59       55       48       48       48      43       41
      now      people      defense      s      also      world      military      terrorism
39      39       39       39       38       38      36       36       36
      president      american      can      going      make      middle      state      every
36      32       30       30       30       28      27       27       27
      states      just      americans      must      east      radical      obama      united
27      26       26       26       26       25      25       25       25
      immigration      iraq      billion      islamic      t      us      work      office
23      23       23       23       22       22      22       20       20
      way      war      time      year      another      today      policy      administration
20      20       20       20       20      19      18      18      18
      plan      ever      ever      ever      ever      ever      ever      ever
18      18       18       18       18

```

Environment History

help(dfm)
dfm.simple<-dfm(cleancorpus, toLower = TRUE, ignoredFeatures =stopwords("english"), verbose=TRUE, stem=FALSE)
dfm.simple<-dfm(cleancorpus, toLower = TRUE, ignoredFeatures =stopwords("english"), verbose=TRUE, stem=FALSE)
head(dfm.simple)
dfm.simple
topfeatures<-topfeatures(dfm.simple, n=50)
topfeatures

Files Plots Packages Help Viewer

R: create a document-feature matrix Find in Topic

dfm {quanteda} R Documentation

create a document-feature matrix

Description

Create a sparse matrix document-feature matrix from a corpus or a vector of texts. The sparse matrix construction uses the **Matrix** package, and is both much faster and much more memory efficient than the corresponding dense (regular matrix) representation. For details on the structure of the dfm class, see [dfm-class](#).

Usage

dfm(x, ...)

- Frequency analysis with stemming

```

28     tolower = TRUE,
29     ignoredFeatures = stopwords("english"),
30     verbose=TRUE,
31     stem=FALSE)
32 head(dfm.simple)
33 dfm.simple
34 #to display most frequent terms in dfm
35 topfeatures<-topfeatures(dfm.simple, n=50)
36 topfeatures
37
38 #to create a custom dictionary list of stop words
39 swlist = c("will", "must", "can", "just", "s", "ever")
40 dfm.stem<- dfm(cleancorpus, tolower = TRUE,
41   ignoredFeatures = c(swlist, stopwords("english")),
42   verbose=TRUE,
43   stem=TRUE)
44 topfeatures.stem<-topfeatures(dfm.stem, n=50)
45 topfeatures.stem
46
46:1 (Top Level) <-
  
```

Console -/Desktop/Business Analytics/

```

... indexing features: 2,382 feature types
... removed 131 features, from 180 supplied (glob) feature types
... stemming features (English), trimmed 498 feature variants
... created a 3 x 1753 sparse dfm
... complete.

Elapsed time: 0.058 seconds.
> topfeatures.stem<-topfeatures(dfm.stem, n=50)
> topfeatures.stem
  country american    new    america    state    clinton    one    hillary    defens    isi    presid    now
  74      60      55      54      54      51      48      43      43      41      40      39
  terror    peopl    nation    also    year    militari    go    make    immigr    polici    islam
  39      39      38      38      37      36      36      35      32      32      29      29
  radic    offic    middl    everi    support    work    war    east    obama    unit    law    billion
  28      28      27      27      26      26      25      25      25      25      25      24
  want    take    ask    iraq    job    plan    t    us    protect    way    time    put
  23      23      23      23      23      22      22      22      21      20      20      20
  need    administr
  20      19
  
```

R Script

Environment History To Console To Source

```

tolower = TRUE,
ignoredFeatures = stopwords("english"),
verbose=TRUE,
stem=FALSE)
head(dfm.simple)
dfm.simple
topfeatures<-topfeatures(dfm.simple, n=50)
topfeatures
swlist = c("will", "must", "can", "just", "s", "ever")
dfm.stem<- dfm(cleancorpus, tolower = TRUE,
ignoredFeatures = c(swlist, stopwords("english")),
verbose=TRUE,
stem=TRUE)
topfeatures.stem<-topfeatures(dfm.stem, n=50)
topfeatures.stem
  
```

Files Plots Packages Help Viewer

R: create a document-feature matrix Find in Topic

dfm {quanteda} R Documentation

create a document-feature matrix

Description

Create a sparse matrix document-feature matrix from a corpus or a vector of texts. The sparse matrix construction uses the **Matrix** package, and is both much faster and much more memory efficient than the corresponding dense (regular matrix) representation. For details on the structure of the dfm class, see [dfm-class](#).

Usage

dfm(x, ...)

- Frequency analysis using Bigrams

```

44 topfeatures.stem<-topfeatures(dfm.stem, n=50)
45 topfeatures.stem
46
47 #exploration in context
48 kwic(cleancorpus, "clinton", 2)
49
50 #dfm with bigrams
51 cleancorpus <- tokenize(newcorpus,
52   removeNumbers=TRUE,
53   removePunct = TRUE,
54   removeSeparators=TRUE,
55   removeTwitter=FALSE,
56   ngrams=2, verbose=TRUE)
57 dfm.bigram<- dfm(cleancorpus, tolower = TRUE, |
58   ignoredFeatures = c(swlist, stopwords("english")),
59   verbose=TRUE,
60   stem=FALSE)
61 topfeatures.bigram<-topfeatures(dfm.bigram, n=50)
62 topfeatures.bigram
57:47 (Top Level) <-
  
```

Console -/Desktop/Business Analytics/

```

> topfeatures.bigram<-topfeatures(dfm.bigram, n=50)
> topfeatures.bigram
  hillary_clinton    middle_east    united_states    president_obama    islamic_terrorism
  41                  25              25                19                  18
  make_america    radical_islamic    foreign_policy    missile_defense    didn_t
  14                  13              12                12                  7
  american_people    radical_islam    cold_war        state_sponsor    air_force
  7                   7               6                 6                  6
  last_year    cyber_capabilities    inner_cities    whole_world    spread_across
  6                   6               6                 5                  5
  billion_dollars    one_american    america_safe    years_ago    police_officers
  5                   5               5                 5                  5
  failed_policies    defeating_radical    taking_office    weeks_ago    national_security
  4                   4               4                 4                  4
  north_korea    t_even_classified_information    civil_war    refugee_crisis
  4                   4               4                 4                  4
  now_threatens    nuclear_weapons    barack_obama    defense_spending    marine_corps
  4                   4               4                 4                  4
  current_missions    spent_billion    defense_budget    ask_congress    defense_sequester
  
```

R Script

Environment History To Console To Source

```

topfeatures.stem<-topfeatures(dfm.stem, n=50)
topfeatures.stem
kwic(cleancorpus, "clinton", 2)
cleancorpus <- tokenize(newcorpus,
removeNumbers=TRUE,
removePunct = TRUE,
removeSeparators=TRUE,
removeTwitter=FALSE,
ngrams=2, verbose=TRUE)
dfm.bigram<- dfm(cleancorpus, tolower = TRUE,
ignoredFeatures = c(swlist, stopwords("english")),
verbose=TRUE,
stem=FALSE)
topfeatures.bigram<-topfeatures(dfm.bigram, n=50)
topfeatures.bigram
  
```

Files Plots Packages Help Viewer

R: create a document-feature matrix Find in Topic

dfm {quanteda} R Documentation

create a document-feature matrix

Description

Create a sparse matrix document-feature matrix from a corpus or a vector of texts. The sparse matrix construction uses the **Matrix** package, and is both much faster and much more memory efficient than the corresponding dense (regular matrix) representation. For details on the structure of the dfm class, see [dfm-class](#).

Usage

dfm(x, ...)

4. Complete a sentiment analysis.

- Created a dictionary with positive and negative words to perform sentiment analysis.

The screenshot shows the RStudio interface with the following details:

- Environment:** Shows the code being run and its output.
- Console:** Displays the R code and its execution results. The results show the creation of a document-feature matrix from a tokenizedTexts object, indexing documents, indexing features, applying a dictionary, and creating a sparse dfm. The output includes the following text:

```

positive negative
31      14
> View(dfm.sentiment)
> mydict <- dictionary(list(negative = c("detriment*", "bad*", "awful*", "terrib*", "horrib*", "brutal", "disappointing", "disgusting", "senseless", "dreadful", "flawed"),
+                           positive = c("good", "great*", "best", "super*", "excellent", "awesome", "pleasant", "powerful", "insightful", "clever", "fascinating", "success")))
> dfm.sentiment <- dfm(cleancorpus, dictionary = mydict)
Creating a dfm from a tokenizedTexts object ...
  ... indexing documents: 3 documents
  ... indexing features: 8,120 feature types
  ... applying a dictionary consisting of 2 keys
  ... created a 3 x 2 sparse dfm
  ... complete.
Elapsed time: 0.04 seconds.
> topfeatures(dfm.sentiment)
positive negative
31      14

```

The screenshot shows the RStudio interface with the following details:

- Environment:** Shows the code being run and its output.
- Console:** Displays the R code and its execution results. The results show the creation of a document-feature matrix from a tokenizedTexts object, indexing documents, indexing features, applying a dictionary, and creating a sparse dfm. The output includes the following text:

```

positive negative
31      14
> View(dfm.sentiment)
> mydict <- dictionary(list(negative = c("detriment*", "bad*", "awful*", "terrib*", "horrib*", "brutal", "disappointing", "disgusting", "senseless", "dreadful", "flawed"),
+                           positive = c("good", "great*", "best", "super*", "excellent", "awesome", "pleasant", "powerful", "insightful", "clever", "fascinating", "success")))
> dfm.sentiment <- dfm(cleancorpus, dictionary = mydict)
Creating a dfm from a tokenizedTexts object ...
  ... indexing documents: 3 documents
  ... indexing features: 8,120 feature types
  ... applying a dictionary consisting of 2 keys
  ... created a 3 x 2 sparse dfm
  ... complete.
Elapsed time: 0.04 seconds.
> topfeatures(dfm.sentiment)
positive negative
31      14
> View(dfm.sentiment)

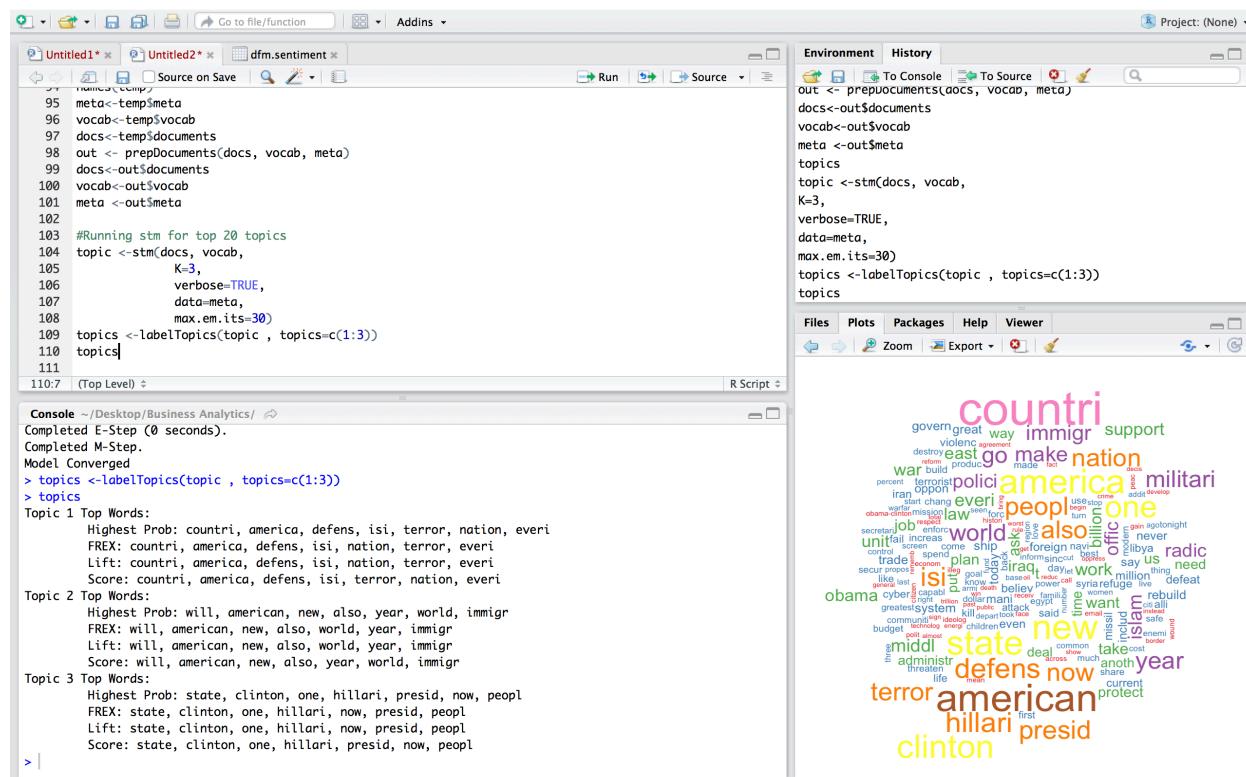
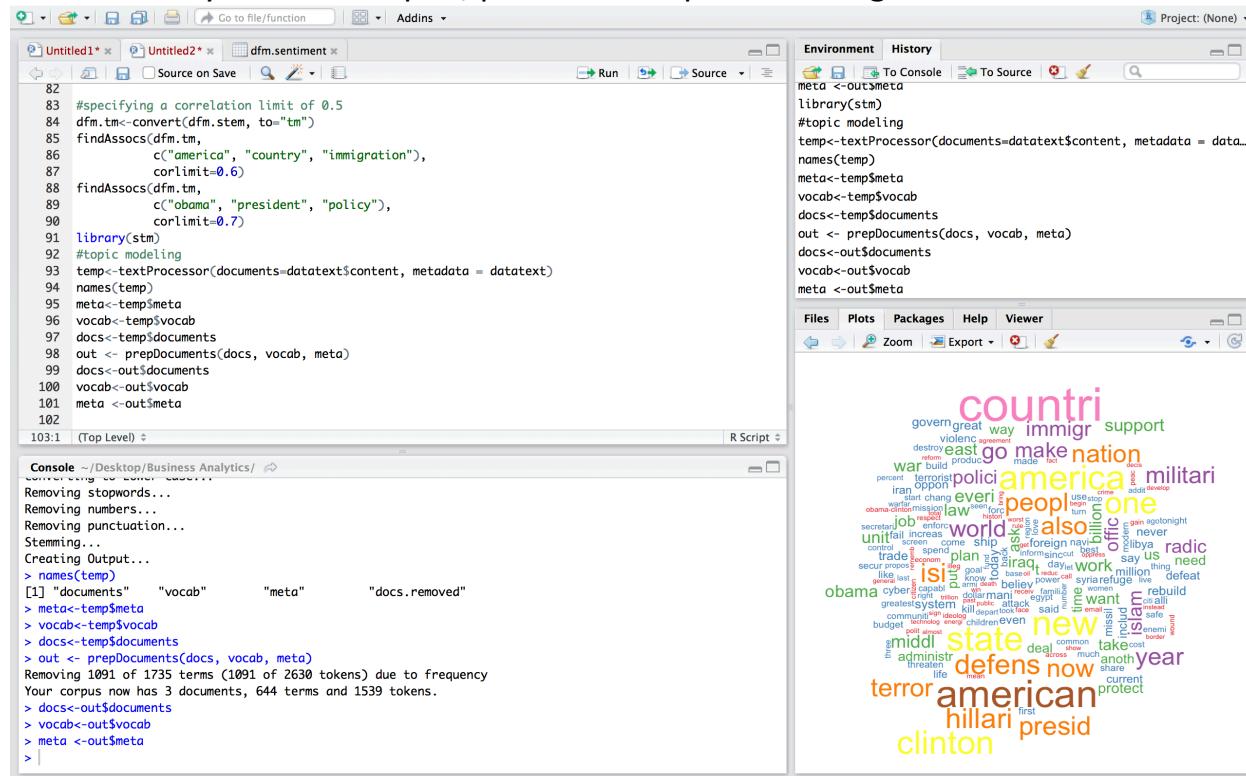
```

- Data View:** Shows a preview of the document-feature matrix. The matrix has 3 rows (T1, T2, T3) and 2 columns (negative, positive). The values are:

	negative	positive
T1	0	6
T2	12	20
T3	2	5

5. What are the common topics in the corpus?

- To identify common topics, performed topic modelling.



The screenshot shows an RStudio interface. The top-left pane displays R code for topic modeling, specifically using the STM package. The code includes functions like `labelTopics`, `findThoughts`, and various `plot.STM` functions for different types of visualizations. The top-right pane shows the R environment and history. The bottom-left pane is the console, which contains a large block of text from Donald Trump's speech. The bottom-right pane is a text box containing three topics identified by the STM model.

```

109 topics <-labelTopics(topic , topics=c(1:3))
110 topics
111
112 #explore the topics in context. Provides an example of the text
113 help("findThoughts")
114 findThoughts(topic, texts = datatext$content, topics = 3, n = 2)
115
116 help("plot.STM")
117 plot.STM(topic, type="summary")
118 plot(STM(topic, type="labels", topics=c(1,2,3))
119 plot(STM(topic, type="perspectives", topics = c(1,2,3))
120
121 # to aid in assignment of labels & interpretation of topics
122 help(topicCorr)
119:56 (Top Level) 

```

Console ~/Desktop/Business Analytics/

that would not stand, I mean they said Trump does not have a chance of being here tonight, not a chance, the same people. We love defeating those people, don't we? Love it.
 No longer can we rely on those same people. In the media and politics who, will say anything to keep a rigid system in place. Instead, we must choose to believe in America.
 History is watching us now. It's we don't have much time. We don't have much time. It's waiting to see if we will rise to the occasion, and if we will show the whole world that America is still free and independent and strong.
 I am asking for your support tonight so that I can be year champion in the White House. And I will be a champion.Your champion.
 My opponent asks her supporters to recite a three-word loyalty pledge. It reads: "I'm with her."
 I choose to recite a different pledge. My pledge reads: "I'm with you the American people."
 I am your voice. So to every parent who dreams for their child, and every child who dreams for their future, I say these words to you tonight: I'm with you, and I will fight for you, and I will win for you.
 To all Americans tonight, in all our cities and towns, I make this promise:
 We will make America strong again.
 We will make America proud again.
 We will make America safe again.
 And we will make America great again!
 God bless you and goodnight! I love you!

```

> plot(STM(topic, type="summary")
> plot(STM(prevfit, type="labels", topics=c(15,4,5))
> plot(STM(prevfit, type="labels", topics=c(1,2,3))
> plot(STM(topic, type="labels", topics=c(1,2,3))
> 

```

Environment History

```

stem=FALSE)
topfeatures.bigram_values<-topfeatures(dfm.bigram_values, n=50)
topfeatures.bigram_values
findThoughts(prevfit, texts = precorpus$Full.text, topics = 10,...)
plot.STM(prevfit, type="summary")
plot.STM(prevfit, type="labels", topics=c(15,4,5))
plot.STM(prevfit, type="perspectives", topics = c(19,10))
findThoughts(topic, texts = datatext$content, topics = 3, n = 2)
plot(STM(topic, type="summary")
plot(STM(prevfit, type="labels", topics=c(15,4,5))
plot(STM(prevfit, type="labels", topics=c(1,2,3))
plot(STM(topic, type="labels", topics=c(1,2,3))

```

Files Plots Packages Help Viewer

Topic 1:
 ica, defens, isi, terror, nation, everi, must, law, , take, plan, time, way, need, deal, great, today

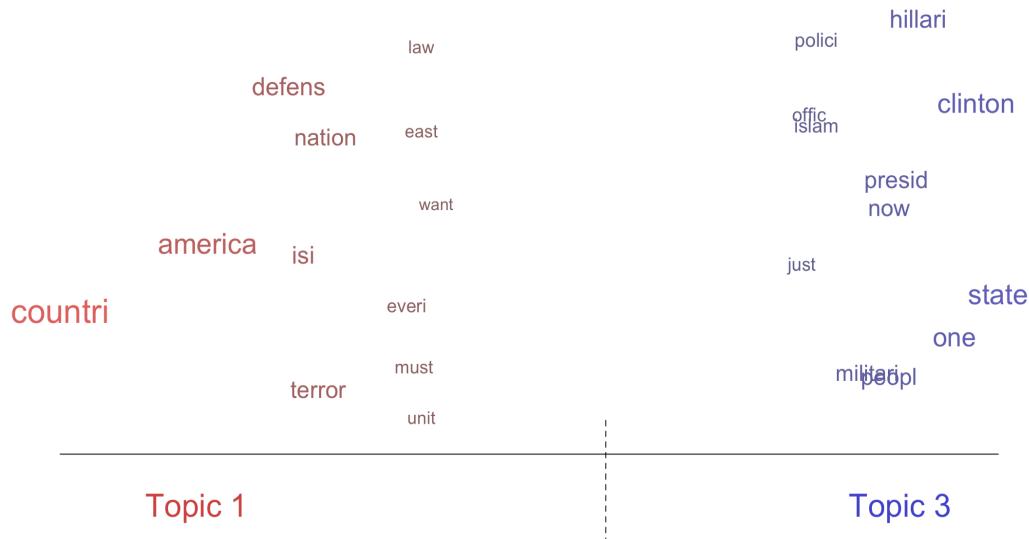
Topic 2:
 ew, also, year, world, immigr, make, radic, war, q, ask, put, anoth, govern, ever, trade, defeat

Topic 3:
 on, one, hillari, presid, now, peopl, militari, polici, dl, work, support, obama, job, protect, adminis

Topic 1:
 countri, america, defens, isi, terror, nation, everi, must, law, unit, east, want, take, plan, time, way, need, deal, great, today

Topic 2:
 will, american, new, also, year, world, immigr, make, radic, war, billion, can, iraq, ask, put, anoth, govern, ever, trade, defeat

Topic 3:
 state, clinton, one, hillari, presid, now, peopl, militari, polici, islam, offic, just, middl, work, support, obama, job, protect, administr, rebuild



```

112 #explore the topics in context. Provides an example of the text
113 help("findThoughts")
114 findThoughts(topic, texts = datatext$content, topics = 3, n = 2)
115
116 help("plot.STM")
117 plot.STM(topic, type="summary")
118 plot.STM(topic, type="labels", topics=c(1,2,3))
119 plot.STM(topic, type="perspectives", topics = c(1,3))
120
121 # to aid on assigment of labels & intepretation of topics
122 help(topicCorr)
123 mod.out.corr <- topicCorr(topic) #Estimates a graph of topic correlations
124 plot.topicCorr(mod.out.corr)
125
125:1 (Top Level) <

```

Console -/Desktop/Business Analytics/ ↵

```

I choose to recite a different pledge. My pledge reads: "I'm with you the American people."
I am your voice. So to every parent who dreams for their child, and every child who dreams for their future,
I say these words to you tonight: I'm with you, and I will fight for you, and I will win for you.
To all Americans tonight, in all our cities and towns, I make this promise:
We will make America strong again.
We will make America proud again.
We will make America safe again.
And we will make America great again!
God bless you and goodnight! I love you!
> plot.STM(topic, type="summary")
> plot.STM(prefit, type="labels", topics=c(15,4,5))
> plot.STM(prefit, type="labels", topics=c(1,2,3))
> plot.STM(topic, type="labels", topics=c(1,2,3))
> plot.STM(topic, type="perspectives", topics = c(1,2,3))
Error in plot.STM(topic, type = "perspectives", topics = c(1, 2, 3)) :
  Too many topics specified.
> plot.STM(topic, type="perspectives", topics = c(1,2,3))
Error in plot.STM(topic, type = "perspectives", topics = c(1, 2, 3)) :
  Too many topics specified.
> plot.STM(topic, type="perspectives", topics = c(1,3))
> mod.out.corr <- topicCorr(topic) #Estimates a graph of topic correlations
> plot.topicCorr(mod.out.corr)
>

```

Environment History

Files Plots Packages Help Viewer

Topic 2

Topic 1

Topic 3

6. Write a memo style report summarizing on Trump's linguistic effectiveness.

Trump's linguistic effectiveness

- Result from frequency Analysis: The most frequent words found in his transcripts are 'will', 'country', 'new', 'america', 'clinton', 'hillary', 'isis', 'people', 'defense', 'radic' etc.
- By performing sentiment analysis and topic modeling, we can understand that trump has always been talking of developing a new America.
- He has always expressed his views on terrorism and means to fight against it.
- From the three transcripts, we understand that he loves talking about his opponent and the drawbacks of having a democratic government.
- He also mentions his views of bringing back jobs to the locals and improving American economy.
- His usage of words, both negative and positive, are very strong and effective.
- Frequent topics of discussion mainly include:

Topic 1:
countri, america, defens, isi, terror, nation, everi, must, law, unit, east,
want, take, plan, time, way, need, deal, great, today

Topic 2:
will, american, new, also, year, world, immigr, make, radic, war, billion, can,
iraq, ask, put, anoth, govern, ever, trade, defeat

Topic 3:
state, clinton, one, hillari, presid, now, peopl, militari, polici, islam,
offic, just, middl, work, support, obama, job, protect, administr, rebuild