

# STA312H5S Tutorial 2

Matthew C. Scicluna

University of Toronto Mississauga

January 26, 2015

# What We Are Going to Cover

A bit more information about this

Today we are going to go over assignment 1 and talk about the tm package

# Firstly we are going to talk about installing packages in R

To install packages into R type this into the R console

```
>install.packages('tm')
```

To use the objects in the package, at the beginning of each R session type

```
>library('tm')
```

Isn't this so much easier than with Python?

# The TM Package

A short introduction to tm

`http://cran.r-project.org/web/packages/tm/vignettes/tm.pdf`

And a longer one, if you're into that kind of thing

`http://cran.r-project.org/web/packages/tm/tm.pdf`

Always read the source documentation of a package before you use it!

# Some Terminology Before We Begin

## Corpus

*a large and structured set of texts (nowadays usually electronically stored and processed). They are used to do statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory. via wikipedia*

## Text Mining

*The process of deriving high-quality information from text, typically through the devising of patterns and trends through means such as statistical pattern learning. via wikipedia*

# Data Entry into R

We need to create a corpus object so we can manipulate it using this package.

1. Firstly lets read in the data. We use the command:

```
DataframeSource(dataset)
DirSource(dataset)
VectorSource(dataset)
#dataset is some dataset read into R
```

2. An example bit of code that would work is:

```
dataset=read.csv("Death Row Data.csv") #stores the
data as a dataframe
data=DataframeSource(dataset) #Tells R that dataset is
a dataframe and each entry is a seperate document
```

# Building the Corpus object

Now we need to actually build the corpus object.

1. We can use the corpus constructor for this:

```
mycorpus<-Corpus(data,  
readerControl=list(language="eng", reader=readPlain))
```

Notice readerControl is a list that specifies the language of the text using and the reader you are using

2. Other readers available are:

```
readPlain()
```

```
readDOC()
```

```
readPDF() #need pdfinfo and pdftotext to be installed  
and accessible on your system
```

Depending on the type of files you are reading from. In our case our source is interpreted as text anyways, so readPlain will suffice.

Whats so great about a corpus anyways?

**ALOT**



# Whats so great about a corpus anyways?

We can now clean up the data.

So we use some functions to perform transformations on the documents in the corpus

```
mycorpus <- tm_map(mycorpus, removePunctuation)
mycorpus <- tm_map(mycorpus, removeNumbers)
mycorpus <- tm_map(mycorpus, stemDocument)
stopwordseng=stopwords(kind = "en")
mycorpus <- tm_map(mycorpus, removeWords, stopwordseng)
mycorpus <- tm_map(mycorpus, content_transformer(tolower))
```

## Whats so great about a corpus anyways?

```
#For the corpus containing prisoners last statements  
  from assignment 1...  
> print(mycorpus)  
<<VCorpus (documents: 495, metadata (corpus/indexed): 0/0)  
>Inspect(mycorpus)  
<<PlainTextDocument (metadata: 7)>>
```

To the victims family, I want you to know that I hope  
you let go of all of the hate because of all my actions.  
I came in as a lion and I come as peaceful as a lamb.  
Im at peace.

# Whats so great about a corpus anyways?

We can see before and after data transformation...

```
>Inspect(mycorpus) #before tm_map...
```

```
<<PlainTextDocument (metadata: 7)>>
```

```
To the victims family, I want you to know that I hope  
  you let go of all of the hate because of all my actions.  
I came in as a  lion and I come as peaceful as a lamb.  
Im at peace.
```

```
>Inspect(mycorpus) #after tm_map...
```

```
to  victim famili i want    know  i hope  let go    hate  
becaus    action i came    lion  i come  peac    lamb  
im  peac  
#less reader friendly but easier to analyze!
```

Finally, lets build a term document matrix

```
>TDM=TermDocumentMatrix(mycorpus)
```

```
>TDM
```

```
<<TermDocumentMatrix (terms: 2339, documents: 495)>>
```

```
Non-/sparse entries: 13683/1144122
```

```
Sparsity           : 99%
```

```
Maximal term length: 14
```

```
Weighting          : term frequency (tf)
```

```
#Very sparse matrix, i.e. a lot of terms appear  
infrequently... Lets fix this!
```

```
>TDM2=removeSparseTerms(TDM, 0.9)
```

```
>TDM2
```

```
<<TermDocumentMatrix (terms: 53, documents: 495)>>
```

```
Non-/sparse entries: 4738/21497
```

```
Sparsity           : 82%
```

```
Maximal term length: 9
```

```
Weighting          : term frequency (tf)
```

Lets analyze this term document matrix

	Docs																		
Terms	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
and	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
ask	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
can	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
caus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
come	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
death	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
declin	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0
done	0	2	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
dont	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0
famili	2	1	0	0	5	0	0	2	0	0	1	1	0	1	1	0	1	0	0
forgiv	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Lets analyze this term document matrix

#Can also build a document term matrix (transpose of term document matrix)

```
>DTM=DocumentTermMatrix(mycorpus,list(dictionary = c("god",  
"jesus", "death")))
```

```
> inspect(DTM)
```

Terms

Docs	death	god	jesus
------	-------	-----	-------

1	0	0	0
---	---	---	---

2	0	2	0
---	---	---	---

3	0	0	0
---	---	---	---

4	0	0	0
---	---	---	---

5	0	5	0
---	---	---	---

6	0	0	0
---	---	---	---

7	0	0	0
---	---	---	---

8	0	1	1
---	---	---	---

9	0	0	0
---	---	---	---

10	0	0	0
----	---	---	---

11	0	0	0
----	---	---	---

12	0	1	0
----	---	---	---

Lets analyze the term document matrix for word freuqencies and associations!

```
> head(findFreqTerms(TDM, 30))  
[1] "all"      "alway"    "and"      "ani"      "apolog"   "ask"
```

```
> head(findAssocs(TDM, "jesus", 0.3))
```

	jesus
christ	0.70
ask	0.57
johnson	0.56
margi	0.56
mchenri	0.56
resurrect	0.56

```
#What did you expect?
```

# Assignment 2

Assignment 2 will involve analyzing some words in this way. I will post the assignment soon!