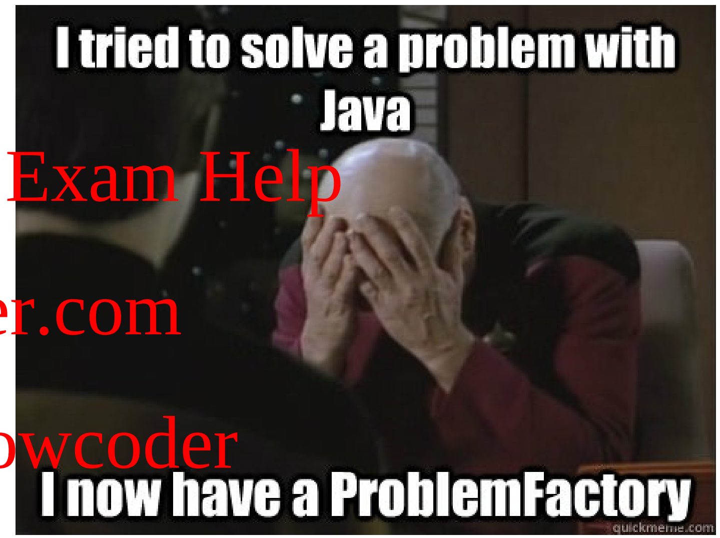
Text Mining

Mo’ Packages Mo’ Problems

```
install.packages(“qdap”)
```

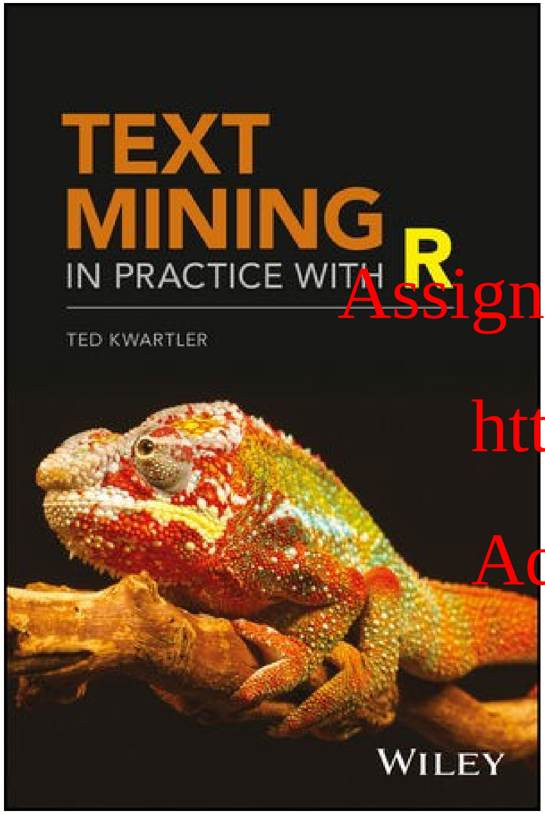
QDAP requires JAVA…so if you don’t have Java it won’t install.




```
install.packages(“tidytext”)
```

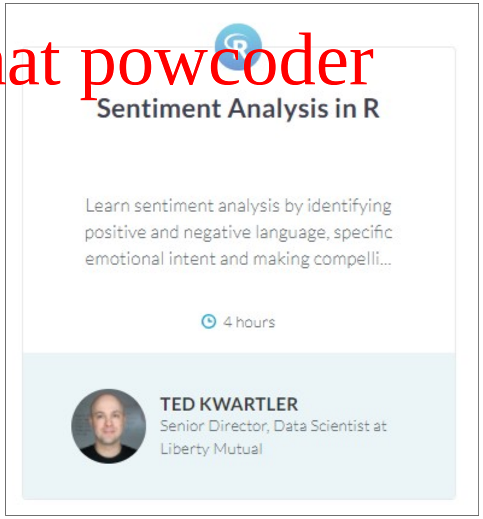
Tidytext is a “tidyverse” package, works in tibbles and with “%>%” so it’s a bit complicated.

Shameless Plug #1



Shameless Plug #2

DataCamp



Agenda

Start	End	Item
		What is Text Mining (TM)?
		Keyword Scanning
		Preparation DTM/TDM
		Associations & Dendrograms (clustering)
		Simple Wordcloud
		Comparison Wordcloud
		Polarity/Sentiment

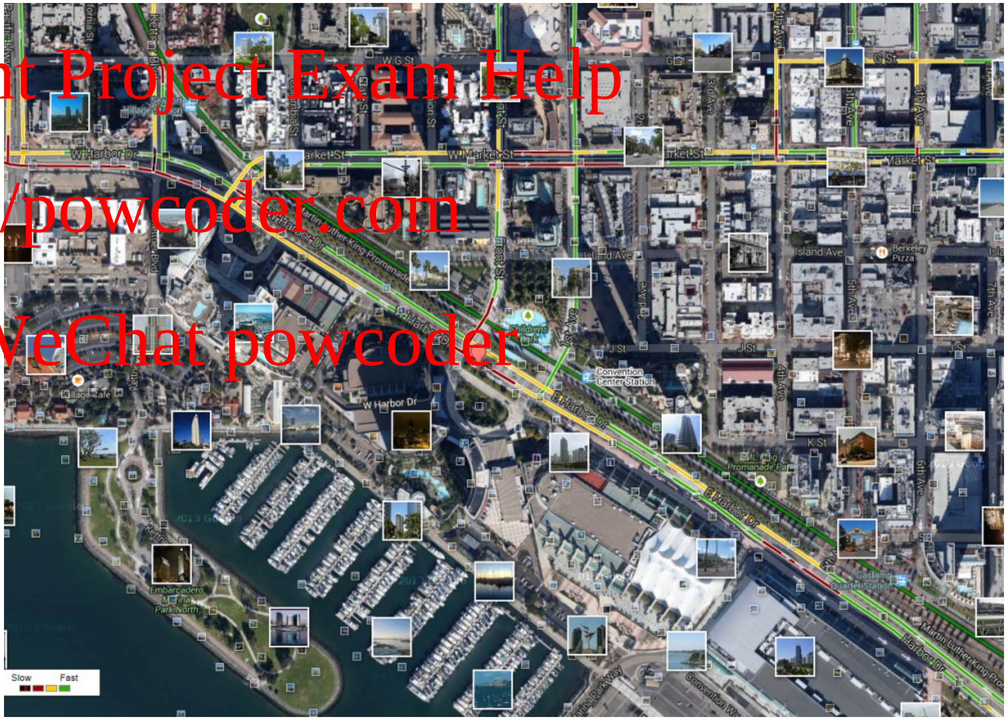
Goals:

- Learn the basics of text mining
- Apply methods to real (& messy) data

What is Text Mining?

- **Extract new insights from text**
- Let's you drink from a fire hose of information
- Language is hard; many unsolved problems
  - Unstructured
  - Expression is individualistic
  - Multi-language/cultural implications

Before Text Mining

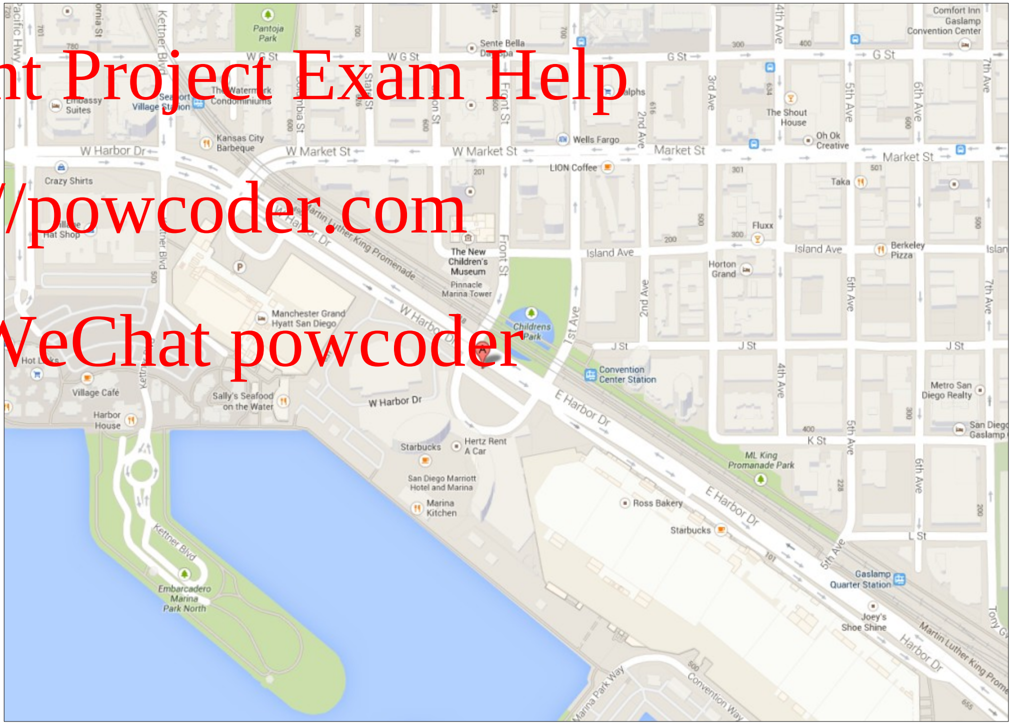


HARVARD UNIVERSITY

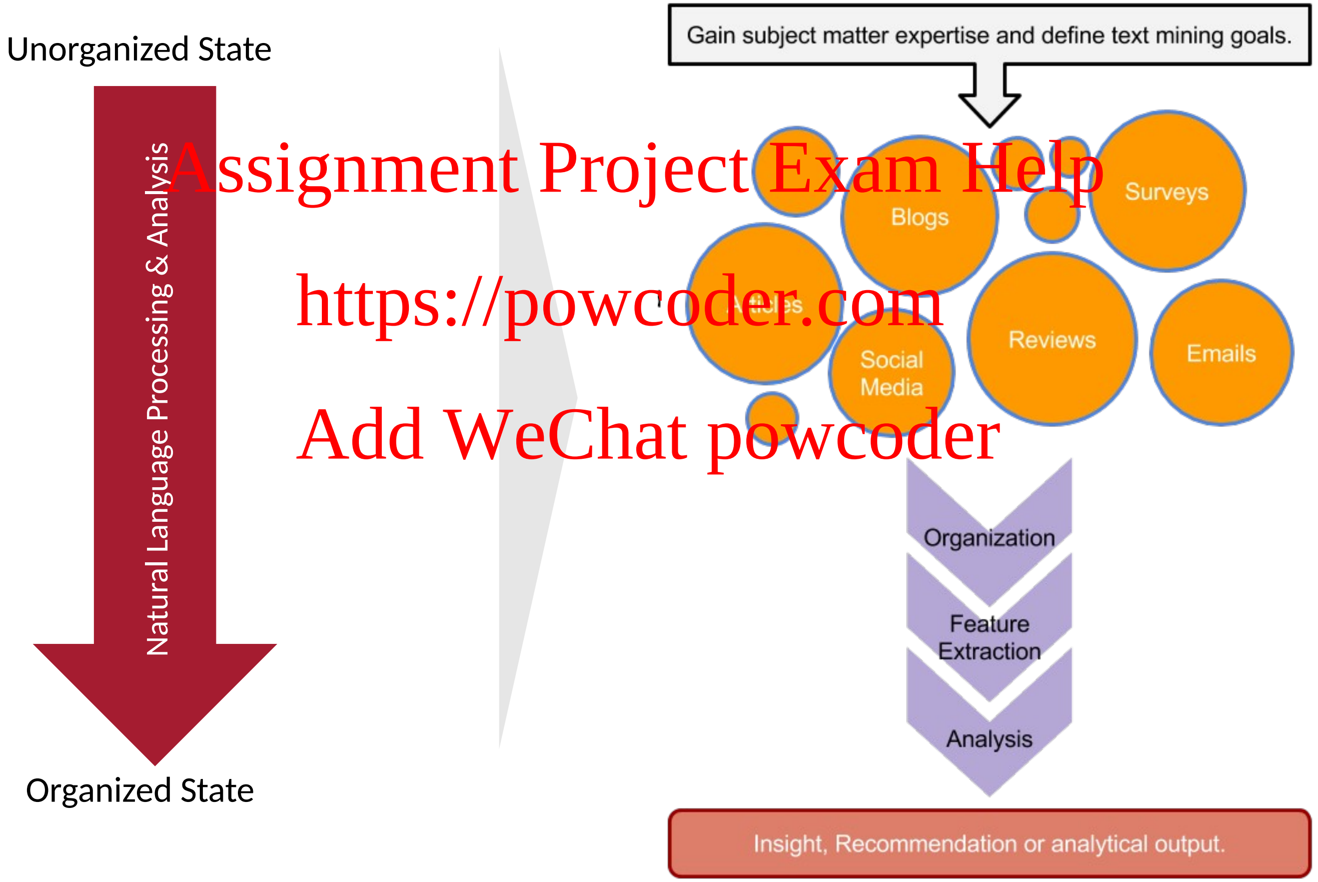
What is Text Mining?

- **Extract new insights from text**
- Let's you drink from a fire hose of information
- Language is hard; many unsolved problems
  - Unstructured
  - Expression is individualistic
  - Multi-language/cultural implications

After Text Mining



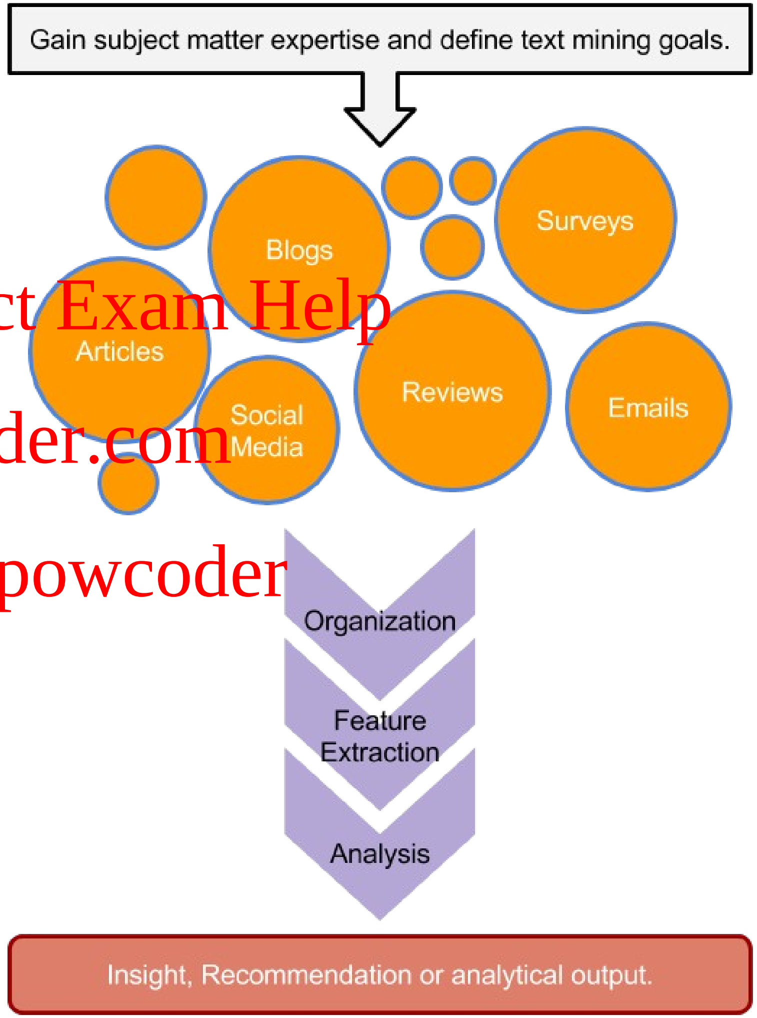
TM Project Workflow



HARVARD UNIVERSITY

Text Mining Workflow

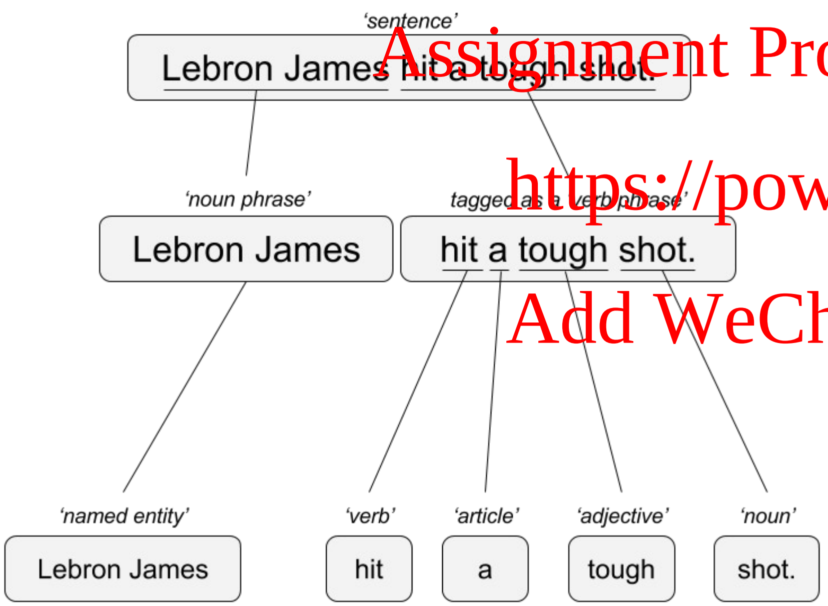
1. Problem Definition
2. Identify Text Sources
3. Text Organization
4. Feature Extraction
5. Analytics
6. Reach Insight or Recommendation

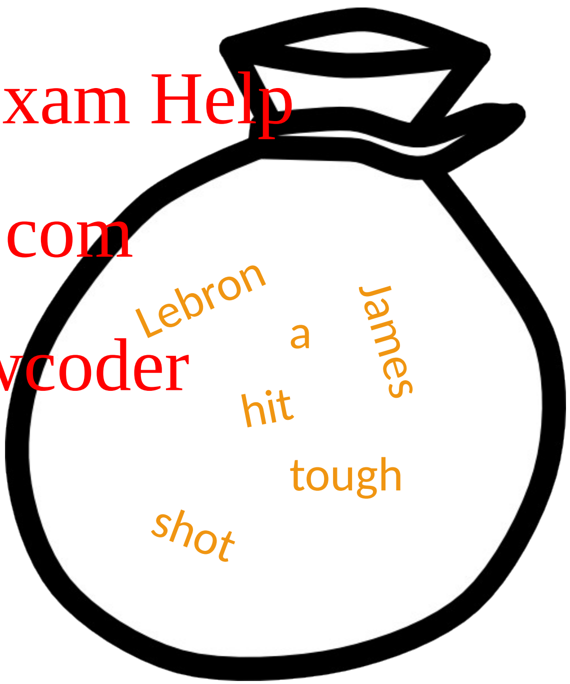


Two Popular Approaches

"Lebron James hit a tough shot."

Syntactic Parsing	Bag of Words





There are other approaches usually based on DNN, that I refer to "abstractive"

TM Challenges - Disambiguation

I made her duck.

- I cooked waterfowl for her to eat.
- I cooked waterfowl belonging to her.
- I created the (clay?) duck and gave it to her.
- Duck!!

TM Challenges - Misc

Other Challenges

- Compound words (tokenization) change meaning
- Sarcasm
- Cultural differences

Examples

- "Bad" vs "not bad"
- "I like it…NOT!"
- "It's *wicked* good" (in Boston)

HARVARD UNIVERSITY

Why text mining is an art & science?

Challenges because human expression is diverse, often ambiguous, affected by age, demographics, socio-economics, medium/channel & regional attributes of the author.

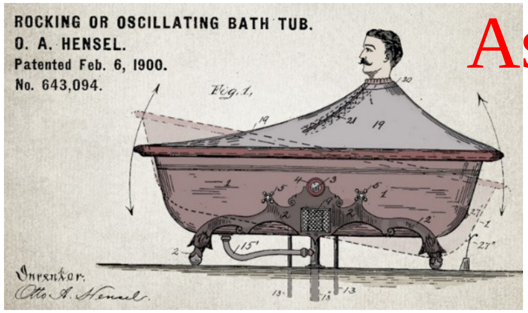
Common Sources

- Books
- Electronic Docs (PDFs)
- Blogs
- Websites
- Social Media
- Customer Records
- Customer Service Notes
- Notes
- Emails
- Legal Documents
- …

Source & context are important, directly impacting data integrity.

HARVARD UNIVERSITY

Channel affects language



Legal documents are verbose & technical.

"Be the change you want to see in the world."

"kappa" indicates sarcasm, irony, or a joke among online gamers.

Expression is context specific making analysis challenging.

HARVARD UNIVERSITY

Messenger affects language

Boomers 1946-1964
* Make a difference
* Consensus/team
* Experiment/try new things
* "Imagine if…"
* Save time
* Features and benefits

Gen X 1965-1980
•Best, finest, world-class
•"You will benefit by…"
•"This is in your best interest."

Gen Y 1981-2000
•Global citizen
•Balance
•Diversity
•Community/connections

Beyond generation, other factors like socio-economic, gender & demographic makeup impact expression.

https://www.anneloehr.com/

Gen Y/Z expression is evolving rapidly due to technology.

- “Thank you, next.”
- “Woke”
- “Cray”
- “Gucci”
- “Squad goal”
- “Bible…”
- “Adulting”
- “Turnt”
- “Hundo P”
- “Trill”
- “TFW”

Beyond generation, other factors like socio-economic, gender & demographic makeup impact expression.

Gen Y/Z expression is evolving rapidly due to technology.

- “Thank you, next.” – moving on with positive connotation
- “Woke” - The more woke one is, the more sympathetic & knowledgeable one is about a topic or type of person
- “Cray” - crazy
- “Gucci” – fine, good
- “Squad goal” – friend group behavior
- “Vibe…” – what follows is TRUTH
- “Adulting” – activities associated with growing up
- “Turnt” – “turned up” i.e. really excited
- “Hundo P” – agree “100%”
- “Trill” – True & Real
- “TFW” – “That Feeling When” to describe an emotion

Beyond generation, other factors like socio-economic, gender & demographic makeup impact expression.

HARVARD UNIVERSITY

Audience affects language




And we all context switch based on who were are speaking too.

HARVARD UNIVERSITY

Agenda

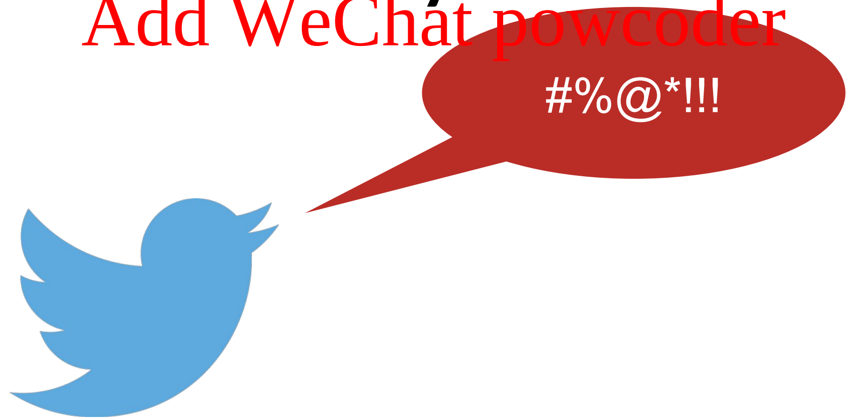
Start	End	Item
		What is Text Mining (TM)?
		Keyword Scanning
		Preparation DTM/TDM
		Associations & Dendrograms (clustering)
		Simple Wordcloud
		Comparison Wordcloud
		Polarity/Sentiment

<u>Goals:</u>

- Learn the basics of text mining
- Apply methods to real (& messy) data

Warning: Twitter Profanity

- Twitter demographics skew young and as a result have profanity that appear in the examples. "Keyboard Courage" is rampant.
- **It's the easiest place to get a lot of messy text fast, if it is offensive feel free to talk to me and I will work to get you other texts for use on your own. No offense is intended.**



1_Keyword_Scanning.R

Basic R Unix Commands

grepl returns a vector of T/F if the pattern is present at least once

```
grepl("pattern", searchable object, ignore.case=TRUE)
```

```
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[18] FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```


grep returns the position of the pattern in the document


```
grep("pattern", searchable object, ignore.case=TRUE)
```

```
[1]   4 214 276 366 479 534 549 620
```


"library(stringi)" Functions

stri_count counts the number of patterns in a document

```
stri_count(searchable object, fixed="pattern")
```

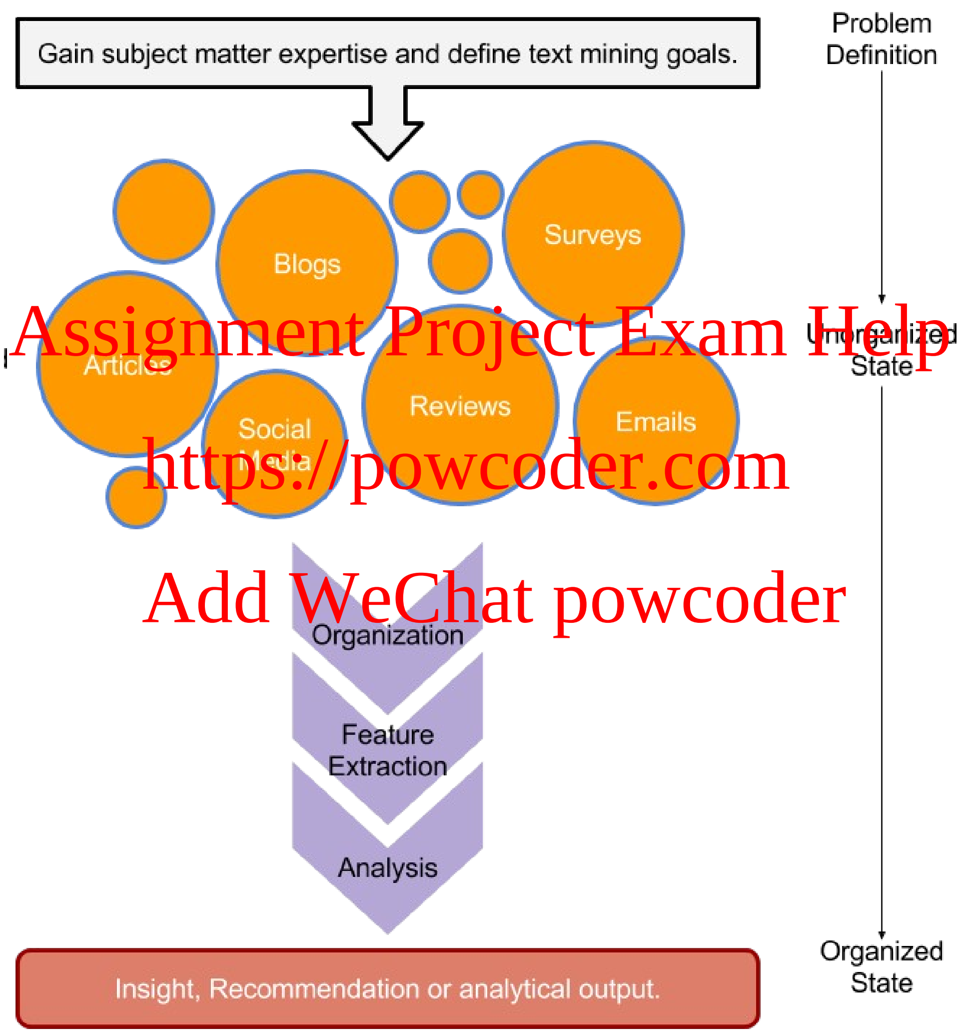
```
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[54] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
```


HARVARD UNIVERSITY

Let's Practice!

Open 1_Keyword_Scanning_revised.R

Remember This?



HARVARD UNIVERSITY

R for Cleaning Steps

Tomorrow I'm going to have a nice glass of Chardonnay and wind down with a good book in the corner of the county :-)

1. Remove Punctuation
2. Remove extra white space
3. Remove Numbers
4. Make Lower Case
5. Remove "stop" words

tomorrow going nice glass chardonnay wind down good book corner county

Library TM Functions

VCorpus creates a corpus held in memory.

```
VCorpus(source)
```

tm_map applies the transformations for the cleaning

```
tm_map(corpus, function)
```


getTransformations() will list all standard tm corpus transformations. We can apply string functions from other packages with content_transformer(FUNCTION)


```
tm_map(corpus, removePunctuation) - removes punctuation from the documents
tm_map(corpus, stripWhitespace) - extra spaces, tabs are removed
tm_map(corpus, removeNumbers) - removes numbers
tm_map(corpus, content_transformer(tolower)) – makes all case lower
tm_map(corpus, removeWords) - removes specific “stopwords”
```


New Text Mining Concepts

Corpus- A collection of documents that analysis will be based on.
Stopwords – are common words that provide very little insight, often articles like “a”, “the”. Customizing them is sometimes key in order to extract valuable insights.

Library qdap Functions

Multiple Global Substitutions

```
mgsub("search pattern", "replacement pattern", text object)
```

Family of Replace Functions

```
replace_abbreviation()-  Replace Abbreviations
replace_contraction()- Replace Contractions
replace_number()-  Replace Numbers With Text Representation
replace_ordinal()- Replace Mixed Ordinal Numbers With Text Representation
replace_symbol()- Replace Symbols with Word Equivalents
```


To use on a corpus you need to apply content_transformer

```
tm_map(corpus, content_transformer(replace_abbreviation))
```

New Text Mining Concepts

Lemmatization in linguistics, is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item.

Poor Man’s Lemmatization

`library(lexicon)`

```
> hash_lemmas
            token     lemma
    1:  furtherst   further
    2:    skilled     skill
    3:     'cause   because
    4:         'd     would
    5:        'em      them
   ---
41529:       zoos       zoo
41530:  zoospores  zoospore
41531:  zucchinis  zucchini
41532:      zulus      zulu
41533:    zygotes    zygote
```

Qdap’s mgsub function can help you lemmatize words.

1. Bring in a lemmatization lexicon.
2. Supply a vector of patterns to search for
3. Supply a vector of patters to replace or change into.
4. Specify the vector the lexicon and substitutions should be applied to.


```
# Poor Man's Lemmatization
library(lexicon)
library(qdap)
data(hash_lemmas)
text$text <- mgsub(hash_lemmas$token,hash_lemmas$lemma,text$text)
```

Warning,: Not done in class because it takes a long time.

HARVARD UNIVERSITY

Custom Functions in 2_Cleaning_and_Frequency_Count.R

"tryTolower" is poached to account for errors when making lowercase.

```
tryTolower <- function(x){
  # return NA when there is an error
  y = NA
  # tryCatch error
  try_error = tryCatch(tolower(x), error =
function(e) e)
  # if not an error
  if (!inherits(try_error, 'error'))
    y = tolower(x)
  return(y)}
```


"clean.corpus" makes applying all transformations easier.

```
cleanCorpus<-function(corpus){
  corpus <- tm_map(corpus,
content_transformer(qdapRegex::rm_url))
  corpus <- tm_map(corpus, removePunctuation)
  corpus <- tm_map(corpus, stripWhitespace)
  corpus <- tm_map(corpus, removeNumbers)
  corpus <- tm_map(corpus, content_transformer(tryTolower))
  corpus <- tm_map(corpus, removeWords, customStopwords)
  return(corpus)
}
```

Base: tolower (basic)
Stringr: str_to_lower (wrapper)
Custom: tryTolower (handles errors)

HARVARD UNIVERSITY

Meta Example

'$doc_id'

'$text'

META: $favorited, $created …

doc_id	text	favorited	replyToSN	created	truncated	replyToSID	id	replyToUID	statusSource	screenName	retweetCount	retweeted	longitude	latitude
1	@ayyytylerb that is so true drink lots of coffee	FALSE	ayyytylerb	8/9/2013 2:43	FALSE	3.65664E+17	3.65665E+17	1637123977	<a href="http://twitter.c	thejennagibson	0	FALSE	NA	NA
2	RT @bryzy_brib: Senior March tmw morning at 7:25 A.M. i	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65665E+17	NA	<a href="http://twitter.c	carolynmosia	1	FALSE	NA	NA
3	If you believe in #gunsense tomorrow would be a very go	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65665E+17	NA	web	janeCkay	0	FALSE	NA	NA
4	My cute coffee mug. http://t.co/2udvMU6XIG	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65665E+17	NA	<a href="http://twitter.c	AlexandriaOOTD	0	FALSE	NA	NA
5	RT @slaredo21: I wish we had Starbucks here... Cause coff	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65665E+17	NA	<a href="http://twitter.c	Rooosssaaaa	2	FALSE	NA	NA
6	Does anyone ever get a cup of coffee before a cocktail??	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65665E+17	NA	<a href="http://twitter.c	E_Z_MAC	0	FALSE	NA	NA
7	"I like my coffee like I like my women...black, bitter, and	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65665E+17	NA	<a href="http://twitter.c	Charlie_31191	0	FALSE	NA	NA
8	@dreamwwediva ya didn't have coffee did ya?	FALSE	dreamwwe	8/9/2013 2:43	FALSE	3.65564E+17	3.65665E+17	1316942208	<a href="http://twitter.c	JessicaSalvato5	0	FALSE	NA	NA
9	RT @iDougherty42: I just want some coffee.	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65665E+17	NA	<a href="http://twitter.c	kaytiekirk	1	FALSE	NA	NA
10	RT @Dorkv76: I can't care before coffee.	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65664E+17	NA	<a href="http://tapbots.c	lissteria	2	FALSE	NA	NA
11	No lie I wouldn't mind coming home smelling like coffee	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65664E+17	NA	<a href="http://twitter.c	DOPECROOK	0	FALSE	NA	NA
12	RT @JonasWorldFeed: Play Ping Pong with Joe. Take a tou	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65664E+17	NA	<a href="http://www.ech	TiffCaruso	6	FALSE	NA	NA
13	Have I ever told any of you that Tate Donovan bought my	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65664E+17	NA	web	CurlysCrazyMofo	0	FALSE	NA	NA
14	RT @JonasWorldFeed: Play Ping Pong with Joe. Take a tou	FALSE	NA	8/9/2013 2:43	FALSE	NA	3.65664E+17	NA	web	JoeJonasVA	6	FALSE	NA	NA
15	@HeatherWhaley I was about 2 joke it takes 2 hands to ho	FALSE	HeatherWh	8/9/2013 2:42	FALSE	3.65647E+17	3.65664E+17	26035764	<a href="http://twitter.c	AnnaDuleep	0	FALSE	NA	NA
16	RT @MoveTheSticks: Charlie Whitehurst looks like he sho	FALSE	NA	8/9/2013 2:42	FALSE	NA	3.65664E+17	NA	<a href="http://twitter.c	mpr4437	42	FALSE	NA	NA
17	Coffee always makes everything better.	FALSE	NA	8/9/2013 2:42	FALSE	NA	3.65664E+17	NA	web	sharkshukri	0	FALSE	NA	NA
18	RT @AdelaideReview: Food For Thought: @Annabelleats	FALSE	NA	8/9/2013 2:42	FALSE	NA	3.65664E+17	NA	<a href="http://twitter.c	thepaulbaker	1	FALSE	NA	NA
19	RT @LittleMelss: lmfao!!!"@bryanlaca: nahhh Melanie u i	FALSE	NA	8/9/2013 2:42	FALSE	NA	3.65664E+17	NA	web	bryanlaca	1	FALSE	NA	NA
20	I wonder if Christian Colon will get a cup of coffee once th	FALSE	NA	8/9/2013 2:42	FALSE	NA	3.65664E+17	NA	<a href="http://www.my	Shauncore	0	FALSE	NA	NA
21	Shouldn't have drank coffee I'm jittery as fuck.	FALSE	NA	8/9/2013 2:42	FALSE	NA	3.65664E+17	NA	<a href="http://twitter.c	DylanBaur	0	FALSE	NA	NA
22	#good_morning <U+2615><ed><U+00A0><U+00BD><ed><U	FALSE	NA	8/9/2013 2:42	FALSE	NA	3.65664E+17	NA	<a href="http://instagran	LadyMonyAna1	0	FALSE	NA	NA
23	@kungfupussy You might need to do a bulk shipment to M	FALSE	kungfupus	8/9/2013 2:42	FALSE	3.65664E+17	3.65664E+17	19478601	<a href="http://janetter.i	Gridlock_Coffee	0	FALSE	NA	NA
24	Gold Coast JCC Friday News features profile on new coffe	FALSE	NA	8/9/2013 2:42	FALSE	NA	3.65664E+17	NA	web	_GCJCC	0	FALSE	NA	NA
25	Sometimes I start dancing on my coffee table because I ca	FALSE	NA	8/9/2013 2:42	FALSE	NA	3.65664E+17	NA	<a href="http://twitter.c	Rileydreams	0	FALSE	NA	NA

- ID is for organization
- Text is the information we want to examine
- Meta adds context to our observations.

Nuances & Inputs for Setting Up a TM Project

"custom.stopwords" combines vectors of words to remove from the corpus

```
#Create custom stop words
customStopwords <- c(stopwords('english'), 'lol',
'smh')
```

Add channel specific stop words.
E.g. Twitter abbreviations

Retaining Meta Data Information


```
# Data
text<-read.csv('coffee.csv', header=TRUE)

# As of tm version 0.7-3 tabular was deprecated
names(text)[1]<-'doc_id' #first 2 columns must be 'doc_id' &
'text'


txtCorpus <- VCorpus(DataframeSource(text))
txtCorpus<-cleanCorpus(txtCorpus)
```


HARVARD UNIVERSITY

How do you retain meta information?

What is Meta?

- Meta information is data associated with the data you are analyzing. These can add context and allow you to partition data in insightful ways.
  - Timestamp (pre 9/11 Vs post 9/11)
  - Language (American Vs King's English)
  - Author (Trump Vs Clinton)
  - Channel (Twitter Vs Legal Documents)

What is content?

- Content is simply the text (strings) you are analyzing. These data points represent the information of interest you are looking to gain insights from.

```
> txtCorpus[[4]]
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 16
```


Retaining/Extracting Meta

```
> t(meta(txtCorpus[4]))
             4
favorited    "FALSE"
replyToSN    NA
created      "2013-08-09 02:43:10"
truncated    "FALSE"
replyToSID   NA
id           "3.656645e+17"
replyToUID   NA
statusSource "<a href=\"http://twitter.com/download/android\" rel=\"no
screenName   "AlexandriaOOTD"
retweetCount "0"
retweeted    "FALSE"
longitude    NA
latitude     NA
```

SINGLE BRACKET

Examining Content

```
> content(txtCorpus[[4]])
[1] " cute coffee mug"
```

DOUBLE BRACKET

During an analysis it may be helpful to examine both meta & content information.

HARVARD UNIVERSITY

For Bag of Words, how is data organized?

Term Document Matrix

	Tweet1	Tweet 2	Tweet3	Tweet4	…	Tweet_n
Term1	0	0	0	0	0	0
Term2	1	1	0	0	0	0
Term3	1	0	0	2	0	0
…	0	0	3	0	1	1
Term_n	0	0	0	1	1	0

Document Term Matrix

	Term1	Term2	Term3	…	Term_n
Tweet1	0	1	1	0	0
Tweet2	0	1	0	0	0
Tweet3	0	0	0	3	0
…	0	0	0	1	1
Tweet_n	0	0	0	1	0

Code to Create the DTM/TDM and change to a matrix

```
txtDtm<-
DocumentTermMatrix(txtCorpus)
txtTdm<-
TermDocumentMatrix(txtCorpus)
txtDtmM<-as.matrix(txtDtm)
```

Why are DTM & TDM Sparse? What do they represent?

??

The matrices are sparse (many 0's) so additional steps may be needed to extract information.

Open 2_Cleaning_and_Frequency_Count_revised.R

Agenda

Start	End	Item
		What is Text Mining (TM)?
		Keyword Scanning
		Preparation DTM/TDM
		Associations & Dendrograms (Clustering)
		Simple Wordcloud
		Comparison Wordcloud
		Polarity/Sentiment

Goals:
- Learn the basics of text mining
- Apply methods to real (& messy) data

Once cleaned, let's get word frequencies.

```
beerFreq<-rowSums(beerTDMm)
beerFreq<-data.frame(word=names(beerFreq),frequency=beerFreq)
```

Term Document Matrix

	Tweet1	Tweet 2	Tweet3	Tweet4	…	Tweet_n
Term1	0	0	0	0	0	0
Term2	1	1	0	0	0	0
Term3	1	0	0	2	0	0
…	0	0	3	0	1	1
Term_n	0	0	0	1	1	0

Word Frequency Matrix

word	freq
Term1	0
Term2	2
Term3	3
…	5
Term_n	2

HARVARD UNIVERSITY

What about a DTM?

```
beerFreq<-???????(beerDTMm)
beerFreq<-data.frame(word=names(beerFreq),frequency=beerFreq)
```

Document Term Matrix

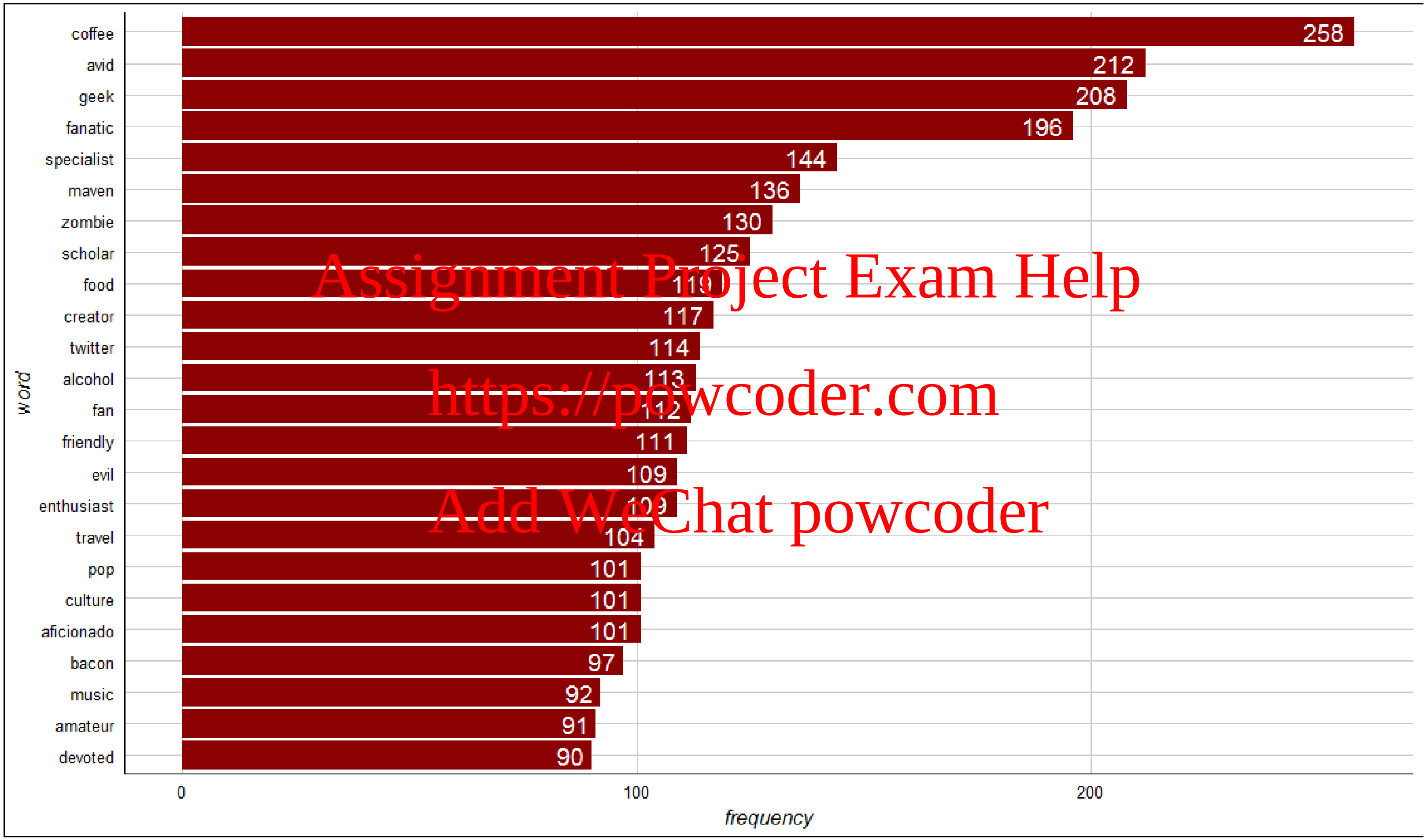
	Term1	Term2	Term3	…	Term_n
Tweet1	0	1	1	0	0
Tweet2	0	1	0	0	0
Tweet3	0	0	0	3	0
…	0	0	0	1	1
Tweet_n	0	0	0	1	0

Word Frequency Matrix

word	freq
Term1	0
Term2	2
Term3	3
…	5
Term_n	2

Can anyone think of how you could get a DTM to be a WFM?

Open 3_Dendrogram.R to visualize the WFM



HARVARD UNIVERSITY

Open 3_Dendrogram.R to visualize the WFM

Zombies!! Word Association

```
# Inspect word associations
associations<-findAssocs(beerTDM, 'zombie', 0.30)

# Make a dot plot
zombieDF<-data.frame(terms=names(associations[[1]]),
                     value=unlist(associations))
zombieDF$terms<-factor(zombieDF$terms, levels=zombieDF$terms)

ggplot(zombieDF, aes(y=terms)) +
  geom_point(aes(x=value), data=zombieDF, col='#c00c00') +
  theme_gdocs() +
  geom_text(aes(x=value, label=value), colour="red",hjust=-.25,
size=3)
```

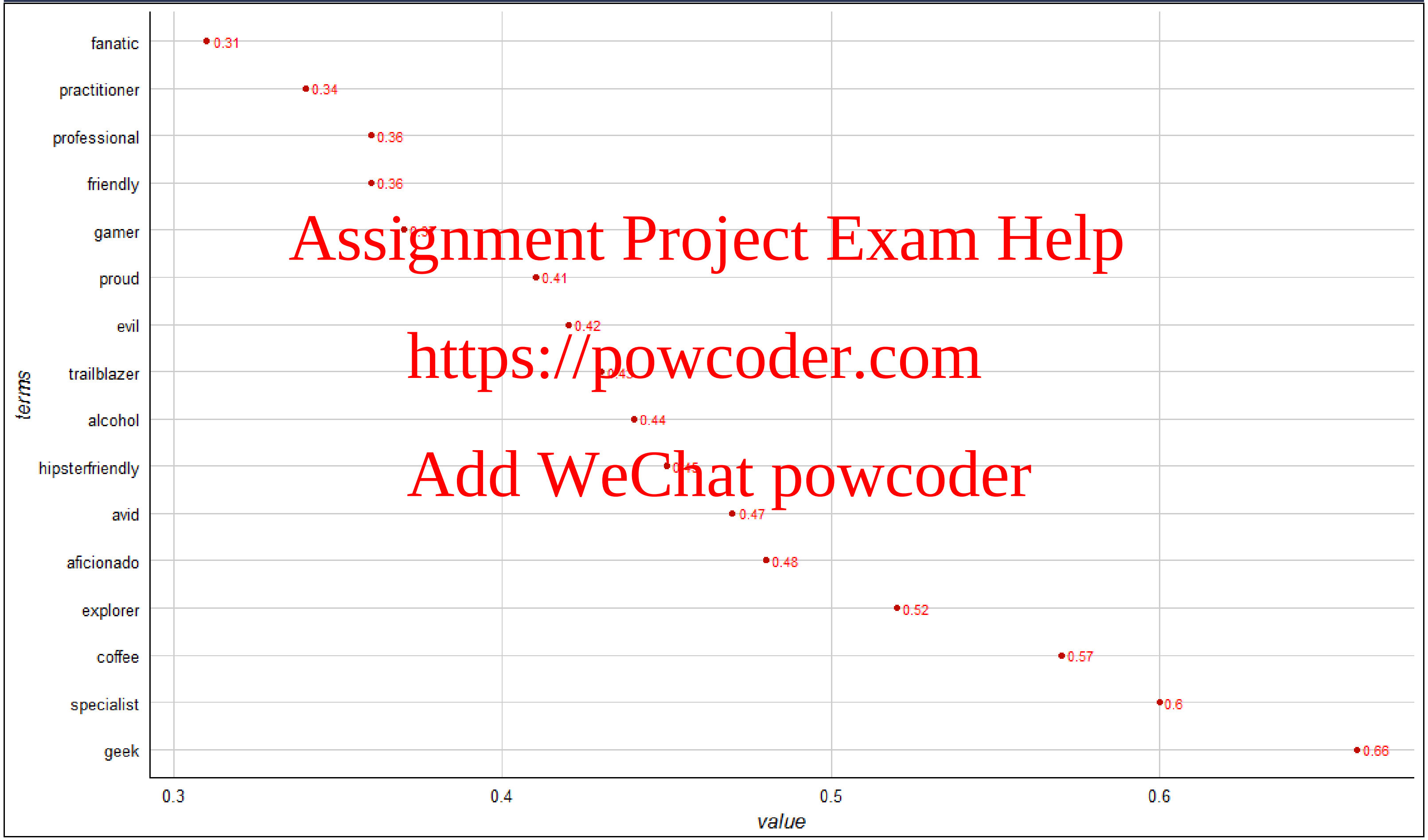

- Adjust 0.30 to get the terms that are associated .30 or more with the ‘zombie’ term.
- Treating the terms as factors lets ggplot2 sort them for a cleaner look.

Word Association is similar to correlation. When word A appears, how often does word B? Unlike correlation, terms can only be positively associated. This is because there are so many terms that everything would be “negatively correlated (associated).

Back to 3_Dendrogram_revised.R

Zombies!! Word Association

Geek has the highest word association with “zombie”



Alcohol geek. Avid tv buff. Friendly beer aficionado. Coffee guru. Zombie junkie.

Dendrograms

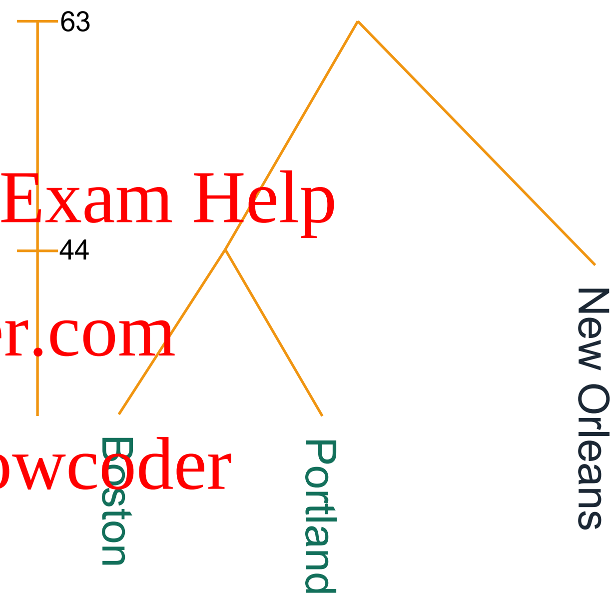
Real Rainfall Data

City	Annual Rainfall
Portland	43.5
Boston	43.8
New Orleans	62.7

Keep in Mind a Dendrogram

- Reduces information much like average is a reduction of many observations' values
- Word clusters emerge often showing related terms
- Term frequency is used to construct the word cluster. Put another way, term A & term B have similar freq. distances & are considered a cluster

Rainfall Data as a Dendrogram



Boston & Portland are a cluster at height 44, losing precision to create the cluster.

Visualizes hierarchical data. For text, the frequency distances are calculated to create the hc object.

Back to 3_Dendrogram_revised.R

Agenda

Start	End	Item
		What is Text Mining (TM)?
		Keyword Scanning
		Preparation DTM/TDM
		Associations & Dendrograms (Clustering)
		Simple Wordcloud
		Comparison Wordcloud
		Polarity/Sentiment

Goals:

- Learn the basics of text mining
- Apply methods to real (& messy) data

Tokenization

```
#bigram token maker
bigramTokens <-function(x)
  unlist(lapply(NLP::ngrams(words(x), 2), paste, collapse = " "), use.names = FALSE)
```

```
wineTDM<-TermDocumentMatrix(txtCorpus, control=list(tokenize=bigramTokens))
```


<u>Text Mining is so fun. So do Text Mining!</u>

Unigram

```
        Docs
Terms      1
  fun.     1
  mining   2
  text     2
```

Bigram

```
               Docs
Terms            1
  do text        1
  fun so         1
  is so          1
  mining is      1
  so do          1
  so fun         1
  text mining    2
```


**with common stopwords*

New Text Mining Concept

Tokenization- So far we have created single word n-grams. We can create multi word “tokens” like bigrams, or trigrams with this line function. It is applied when making the term document matrix.

What is a word cloud?

To make a word cloud we follow the previous steps and create a data frame with the word and the frequency.

```
# Get Row Sums
wineTDMv <- sort(rowSums(wineTDMm),decreasing=TRUE)
wineDF <- data.frame(word =
names(wineTDMv),freq=wineTDMv)
```


Term Document Matrix

	Tweet1	Tweet 2	Tweet3	Tweet4	…	Tweet_n
Term1	0	0	0	0	0	0
Term2	1	1	0	0	0	0
Term3	1	0	0	2	0	0
…	0	0	3	0	1	1
Term_n	0	0	0	1	1	0

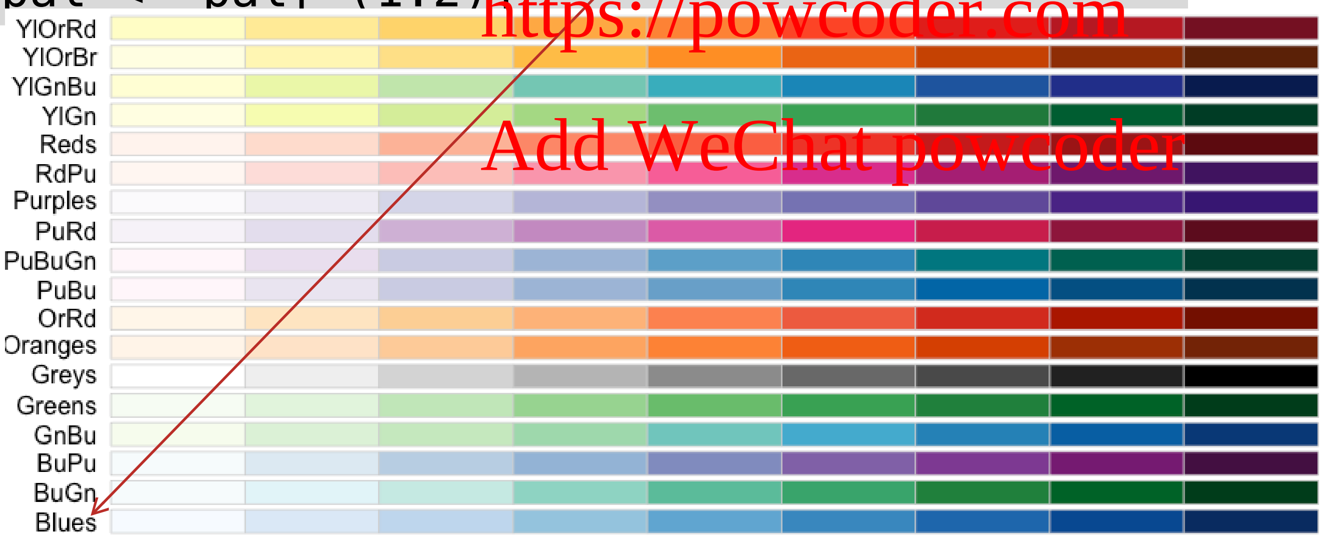
word	freq
Term1	0
Term2	2
Term3	3
…	5
Term_n	2

A word cloud is a visualization of term (token) frequencies.

HARVARD UNIVERSITY

Selecting a color for your word cloud.

```
# Review all Pallettes
display.brewer.all()

# Choose a color & drop light ones
pal <- brewer.pal(8, "Blues")
pal <- pal[-(1:2)]
```

HARVARD UNIVERSITY

Let’s Practice!

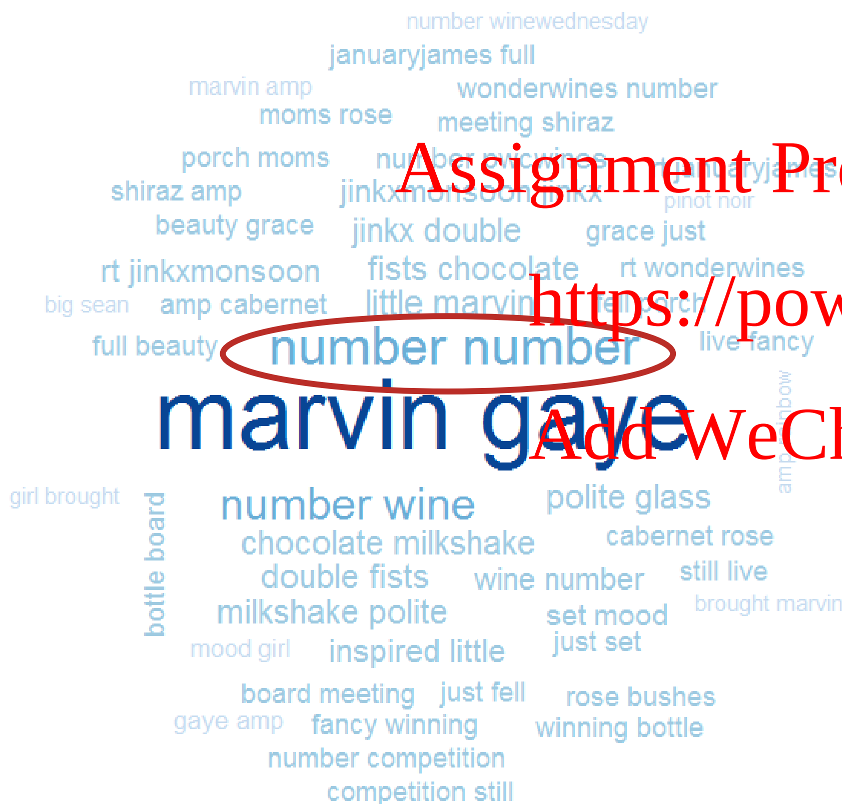
4_Simple_Wordcloud_revised.R

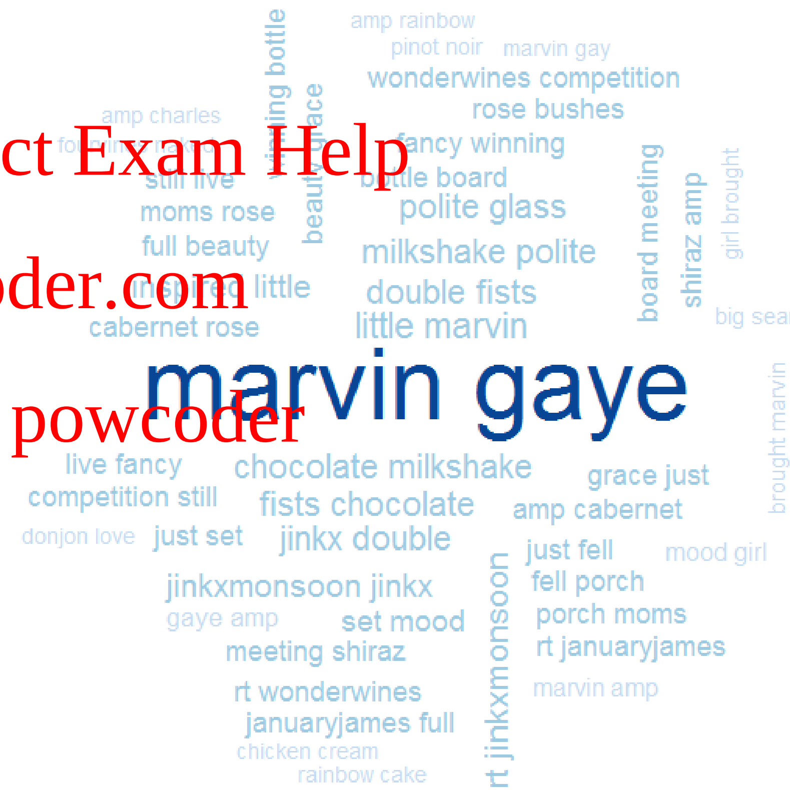
4_Simple_Wordcloud_revised.R

- Bigram Tokenization has captured “marvin gaye”
- A word cloud is a frequency visualization. The larger the term (or bigram here) the more frequent the term.
- You may get warnings if certain tokens are to large to be plotted in the graphics device.
- In cleanCopurs() the function … changes numeric with the generic string “number” so be careful with your preprocessing steps!

4_Simple_Wordcloud.R

In cleanCorpus() the function replace_symbol() changes numeric with the generic string "number" so be careful with your preprocessing steps!





HARVARD UNIVERSITY

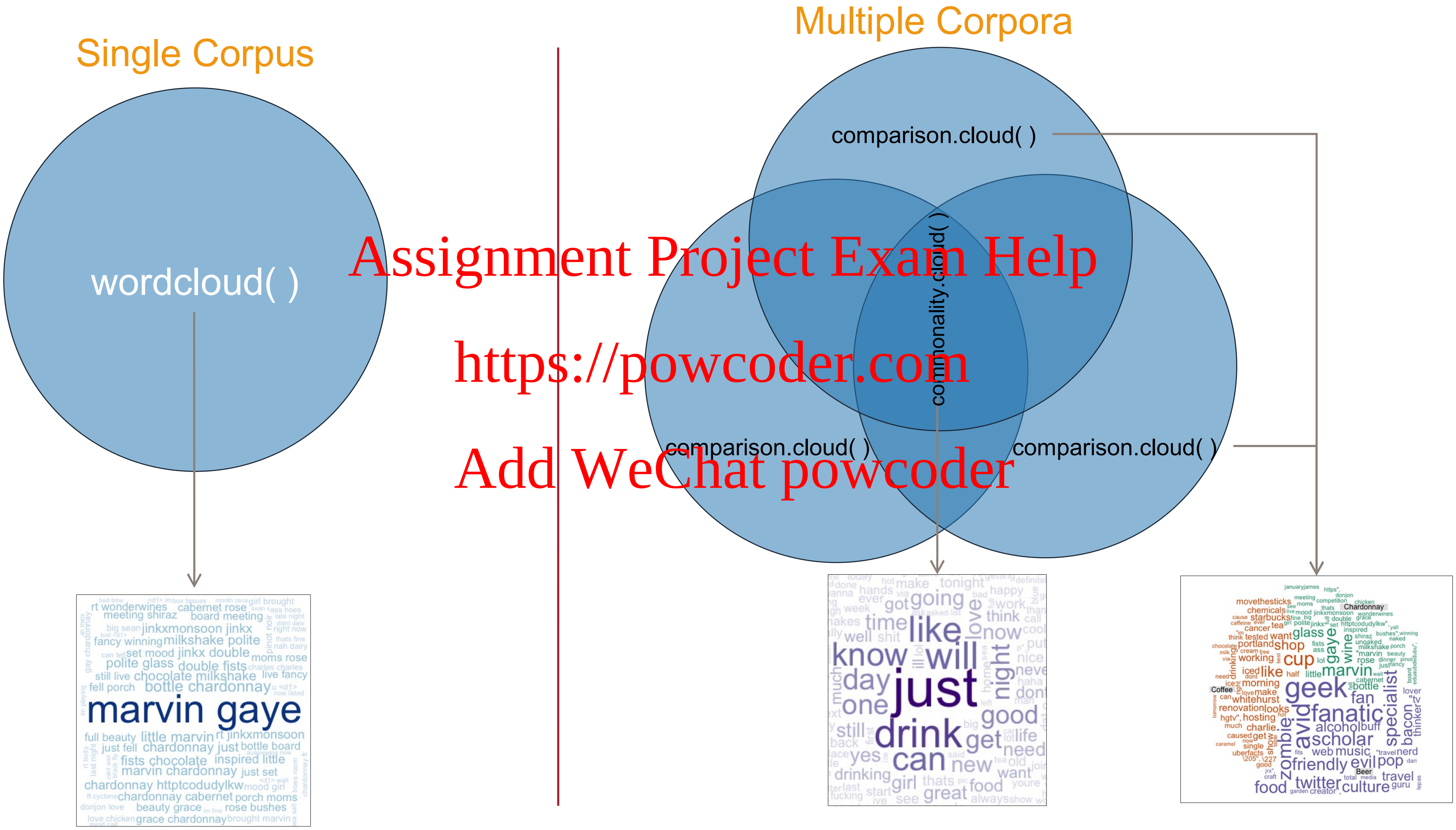
Agenda

Start	End	Item
		What is Text Mining (TM)?
		Keyword Scanning
		Preparation DTM/TDM
		Associations & Dendrograms (Clustering)
		Simple Wordcloud
		Comparison Wordcloud
		Polarity/Sentiment

Goals:

- Learn the basics of text mining
- Apply methods to real (& messy) data

Types of Wordclouds



Dealing with many text files

The challenge is working with multiple corpora efficiently. Many ways to do it…

```
# Read in multiple files as individuals
txtFiles<-c('chardonnay.csv','coffee.csv','beer.csv') #use
list.files() for a lot
for (i in 1:length(txtFiles)){
  assign(txtFiles[i], read.csv(txtFiles[i]))
  cat(paste('read completed:',txtFiles[i],'\n'))
}

# Read them into a single list with individual elements
all<-pblapply(txtFiles,read.csv)
```


Two example ways to import csv files:

1. Each file is read in and an object created for each.
2. A list called “all” is created. Each list element represents a single document. Using data.table::rbindlist() one can create a single document from all files in the folder.

Lets make some improved word clouds

Open 5_Other_Wordclouds_revised.R

Agenda

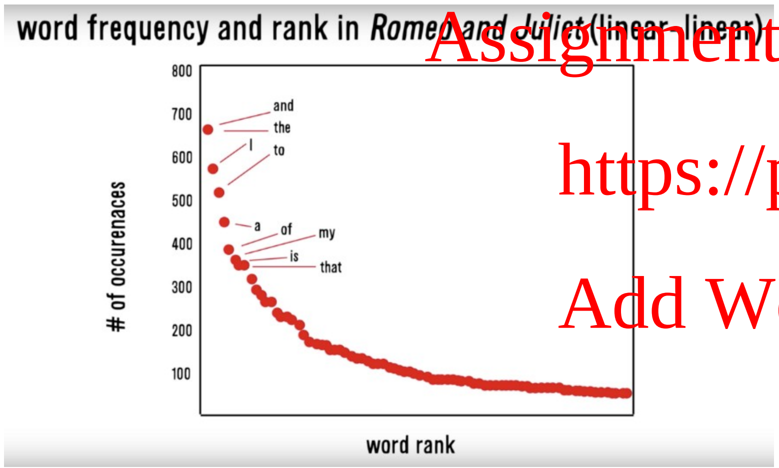
Start	End	Item
		What is Text Mining (TM)?
		Keyword Scanning
		Preparation DTM/TDM
		Associations & Dendrograms (clustering)
		Simple Wordcloud
		Comparison Wordcloud
		Polarity/Sentiment

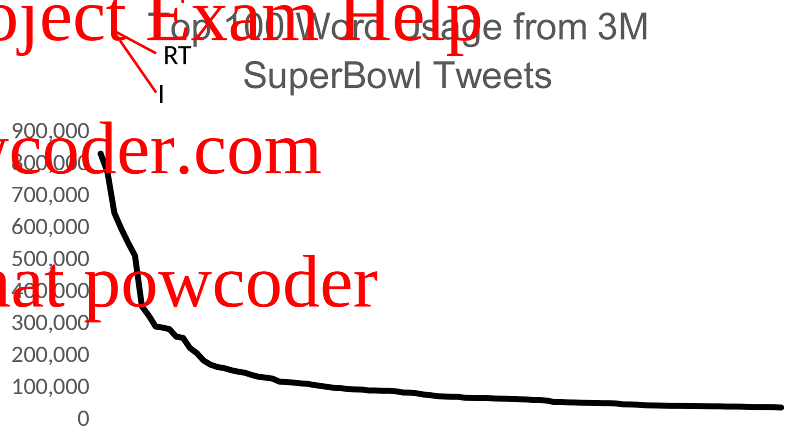
Goals:

- Learn the basics of text mining
- Apply methods to real (& messy) data

Zipf's Law: Our words are less diverse than we think

Two very different contexts, channel, & messengers yet very similar pattern.





Many words in natural language but also a steep decline in actual usage. Follows a predictable pattern.

HARVARD UNIVERSITY

Simple Sentiment Polarity

Scoring

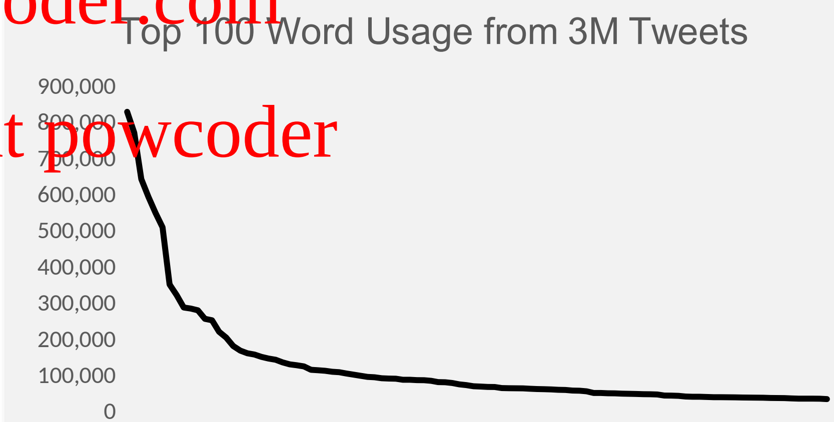
Surprise is a sentiment.
Hit by a bus! – Negative Polarity
Won the lottery!- Positive Polarity

- I loathe BestBuy Service -1
- I love BestBuy Service. They are the best. +2
- I like shopping at BestBuy but hate traffic. 0

R's QDAP polarity function scans for positive words, and negative words as defined by MQPA Academic Lexicon research. It adds positive words and subtracts negative ones along with valence shifters. The final score represents the polarity of the social interaction.

Zipf's Law

Many words in natural language but there is steep decline in everyday usage. Follows a predictable pattern.



Simple Sentiment Polarity

Scoring

```r
library(qdap)

text1<-'i love St Peters University'
text2<-'this lecture is good'
text3<-'this lecture is very good'
text4<-'data science is hard I like it a little'
text5<-'data science is hard'

polarity(text1)
polarity(text2)
polarity(text3)
polarity(text4)
polarity(text5)
```

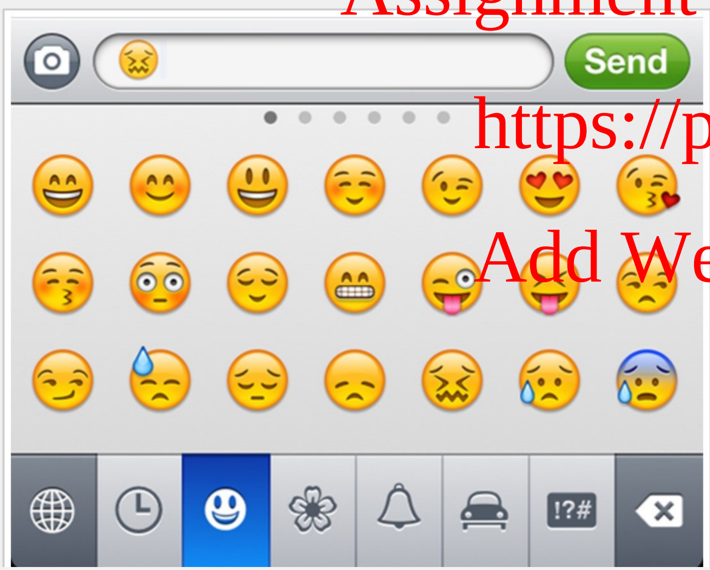
- Text 1: "love" was identified as positive. The text has 5 words and so 1/sqrt(5) = .447
- Text 2: "good" was identified positively. So 1/sqrt(4)=.5
- Text 3: "good" was found along with the amplifier "very". So (.8+1)/sqrt(5)=.805
- Text 4: hard and like cancel each other out so the polarity is zero. 1-1/sqrt(9)=0
- Text 5: "hard" is -1/sqrt(4)=-.50

First it looks for the polarized word. Then identifies valence shifters (default 4 words before and two words after) Amplifiers are assigned +.8 and de-amplifiers weight is constrained to -1. Lastly the sum is divided by the square root of the total number of words in the passage.

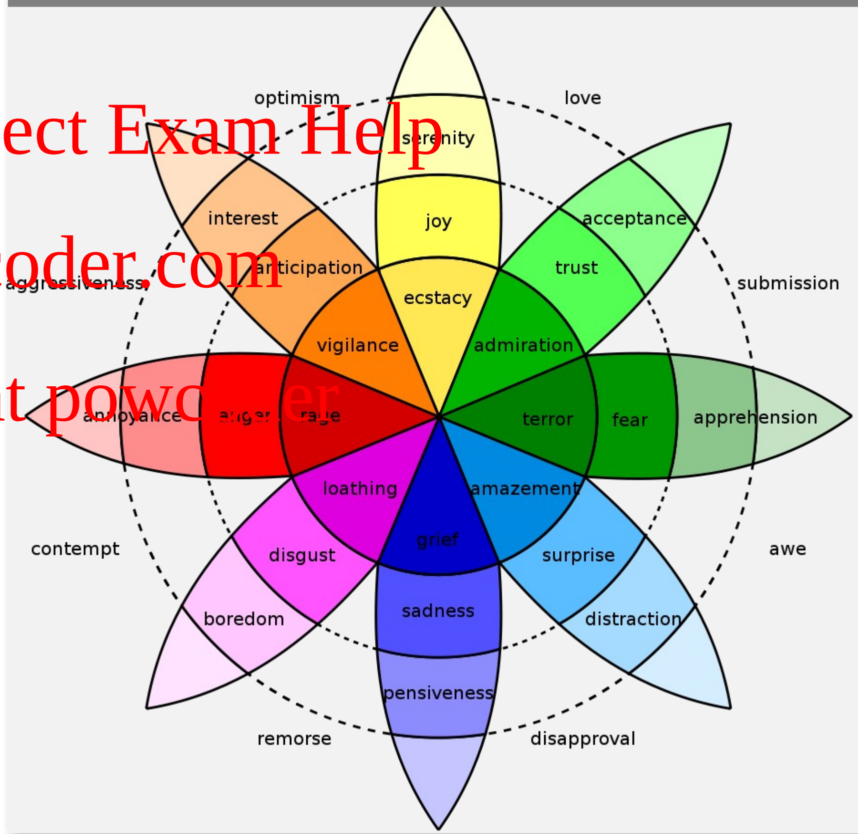
HARVARD UNIVERSITY

In reality sentiment is more complex.

Many Many Emoji



Plutchik's Wheel



HARVARD UNIVERSITY

Kanjoya's Experience Corpus

Sentiment the Tidy Way uses joins with existing lexicons

```
##Tidy Sentiment Analysis
data(sentiments)
sentiments

#Stopwords
data(stop_words)
stop_words

#Add stopwords
custom.stopwords<-data.frame(word=c('amp','beer'),
lexicon='custom')

stop_words<-rbind(stop_words,custom.stopwords)
```

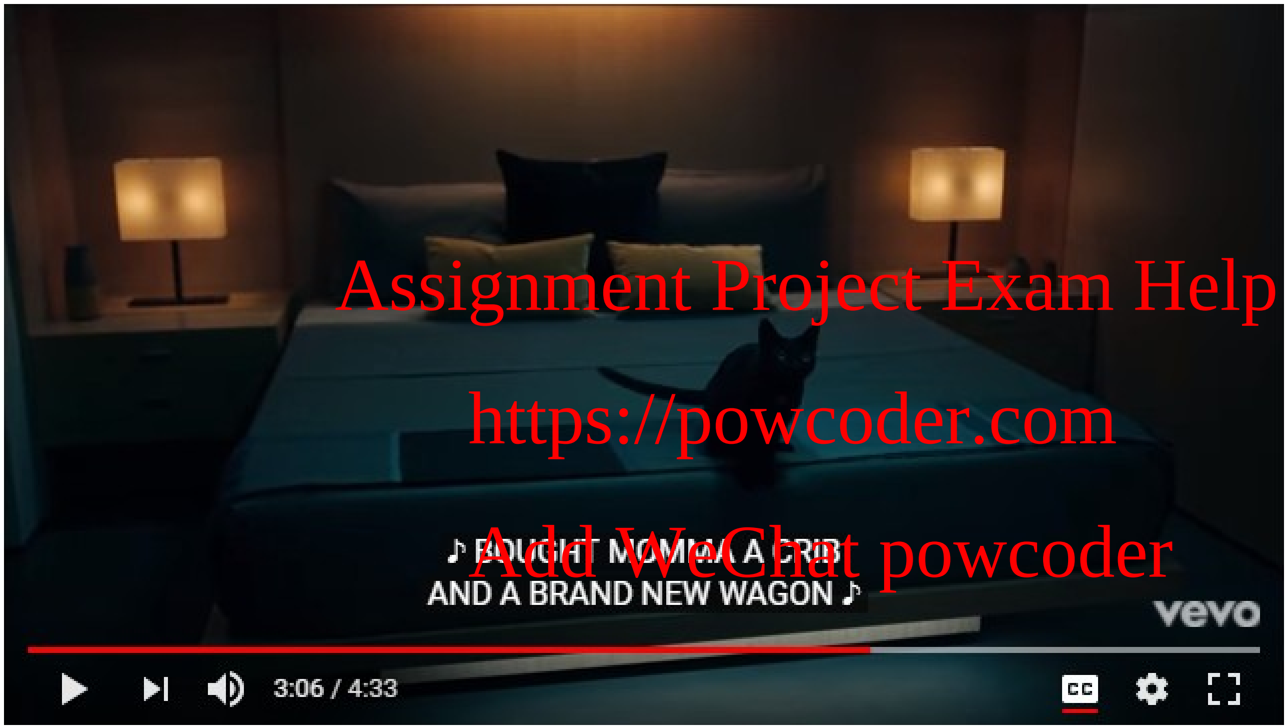


```
> sentiments
# A tibble: 23,165 x 4
          word sentiment lexicon score
         <chr>     <chr>   <chr> <int>
1       abacus     trust     nrc    NA
2      abandon      fear     nrc    NA
3      abandon  negative     nrc    NA
4      abandon   sadness     nrc    NA
5    abandoned     anger     nrc    NA
6    abandoned      fear     nrc    NA
7    abandoned  negative     nrc    NA
8    abandoned   sadness     nrc    NA
9  abandonment     anger     nrc    NA
10 abandonment      fear     nrc    NA
# ... with 23,155 more rows
```

```
> stop_words
# A tibble: 1,151 x 2
          word lexicon
*        <chr>   <chr>
1            a   SMART
2          a's   SMART
3         able   SMART
4        about   SMART
5        above   SMART
6    according   SMART
7  accordingly   SMART
8       across   SMART
9     actually   SMART
10       after   SMART
# ... with 1,141 more rows
```


HARVARD UNIVERSITY

In this exercise we will examine song lyrics



The Weeknd - Starboy (official) ft. Daft Punk

Tidy data uses %>% to forward objects

Tidytext is part of the tidy universe including ggplot and dplyr. Code is structured so it is more easily read using the %>%. The data format is a tibble and is in “tidy” format (long form).


```
mtcars %>% group_by(cyl) %>% mutate(rank = min_rank(desc(mpg)))
```


This reads as “Using the mtcars object *then* group by the cyl vector *then* mutate a new variable called rank.

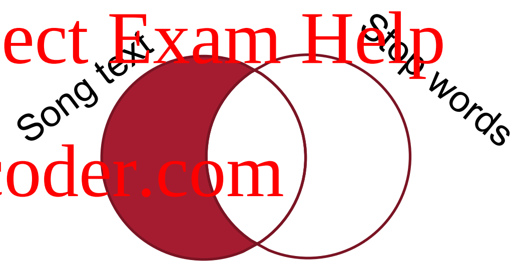
Tidy can seem complicated but not impossible.

The pipe operator

```
%>%
```

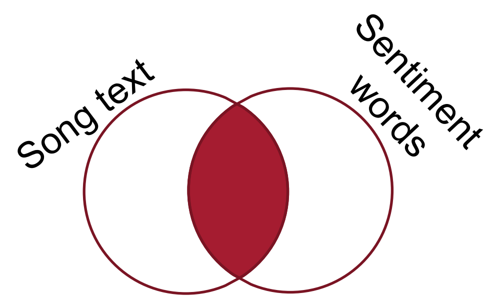
Forwards an object so the code is easy to understand & concise.


```
all.tidy <- all.tidy %>%
  anti_join(stop_words)
```

anti_join()


```
all.sentiment <- all.tidy %>
%
  inner_join(nrc.lexicon) %>
%
  count(tweet,sentiment) %>%
  spread(tweet, n, fill = 0)
```



inner_join()

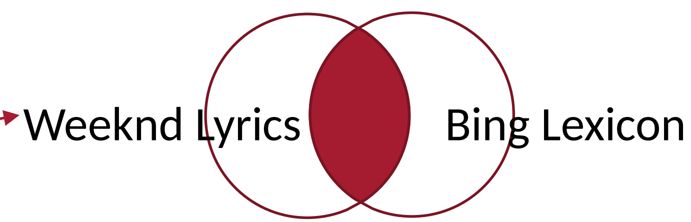
HARVARD UNIVERSITY

Starting with a DTM, its straightforward

```
# DTM
txtDTM<-DocumentTermMatrix(txtCorpus)
txtDTM
dim(txtDTM)

# Tidy
tidyCorp<-tidy(txtDTM)
tidyCorp
dim(tidyCorp)

# Get bing lexicon
# "afinn", "bing", "nrc", "loughran"
bing<-get_sentiments(lexicon =
c("bing"))
head(bing)

# Perform Inner Join
bingSent<-inner_join(tidyCorp,bing,
by=c('term'='word'))
```

DTM is from the “tm” library

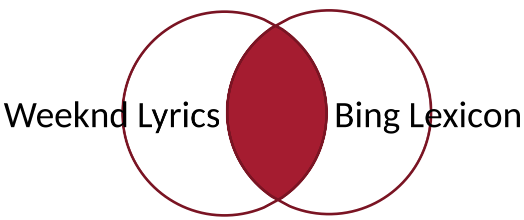
Easy way to make it into a tibble.



HARVARD UNIVERSITY

Similar polarity scores in both methods

```
> table(bingSent$sentiment)

negative positive 
      11        4 
> 
> # Compare with Polarity
> polarity(text)
  all total.sentences total.words ave.polarity sd.polarity stan.mean.polarity
1 all               1         383       -0.409          NA                 NA
```


```
bing<-get_sentiments(lexicon = c("bing"))

bingSent<-inner_join(tidyCorp,bing, by=c('term'='word'))

> table(bingSent$sentiment)

negative positive 
      11        4 
> 4/11
[1] 0.3636364
```

The polarity function from qdap and the inner_join show similar negative results.

TidyText has other sentiment lexicons

Each can be used in an inner join to get different ways of assessing sentiment.

AFINN- Dutch researcher

Words scored -5 to 5

```
> head(afinn)
# A tibble: 6 x 2
        word score
       <chr> <int>
1    abandon    -2
2  abandoned    -2
3   abandons    -2
4   abducted    -2
5  abduction    -2
6 abductions    -2
```

Bing- U of I-Chi Researcher

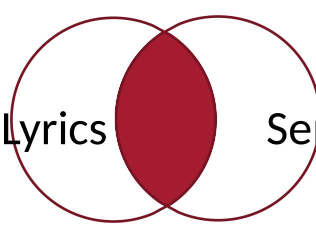
Words scored Pos/Neg

```
> head(bing)
# A tibble: 6 x 2
        word sentiment
       <chr>     <chr>
1    2-faced  negative
2    2-faces  negative
3         a+  positive
4   abnormal  negative
5    abolish  negative
6 abominable  negative
```

NRC – mTurk

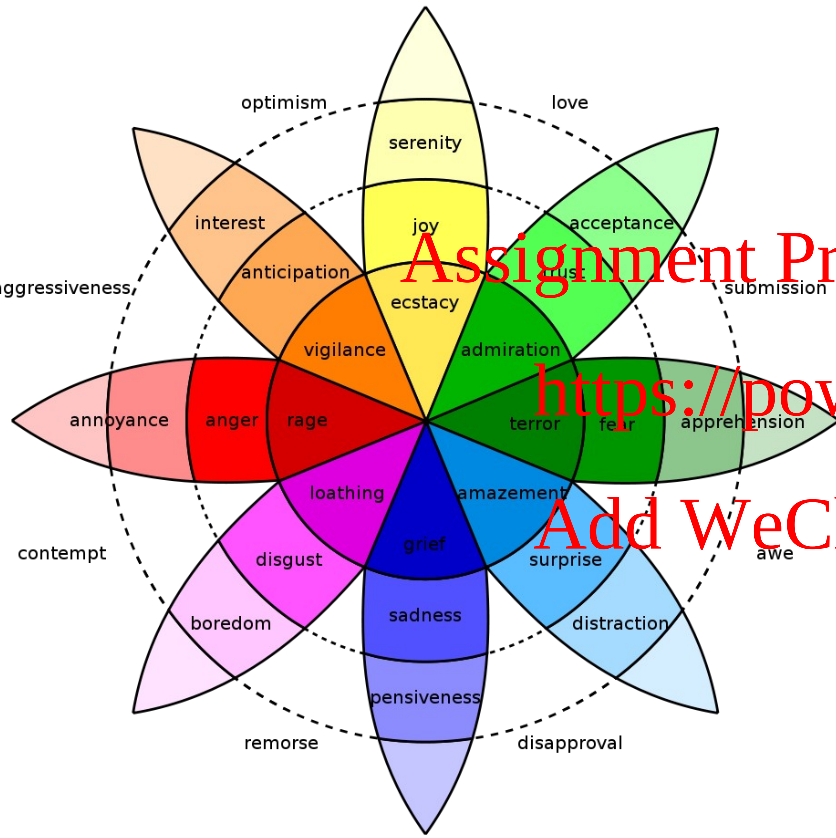
Words classified into 8 primary & pos/neg

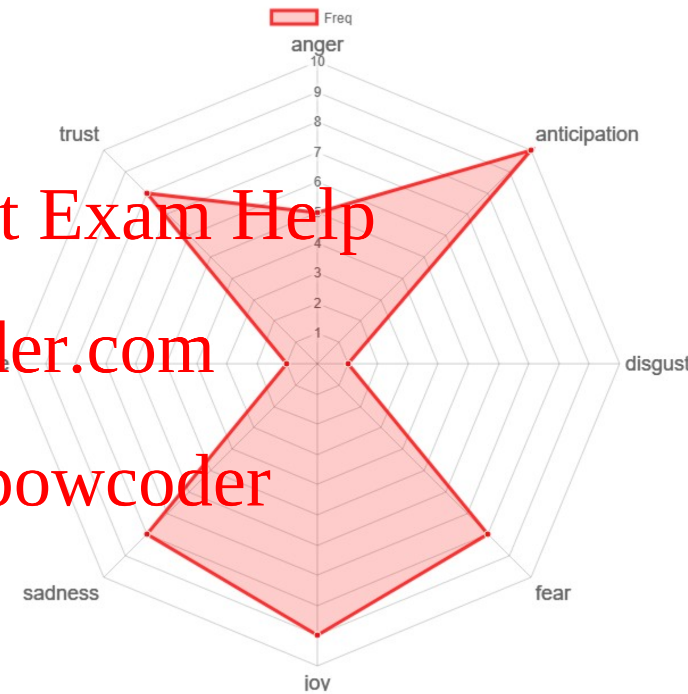
```
> head(nrc)
# A tibble: 6 x 2
       word sentiment
      <chr>     <chr>
1    abacus     trust
2   abandon      fear
3   abandon  negative
4   abandon   sadness
5 abandoned     anger
6 abandoned      fear
```



HARVARD UNIVERSITY

Remember Plutchik's Wheel of emotion? Let's mimic it!

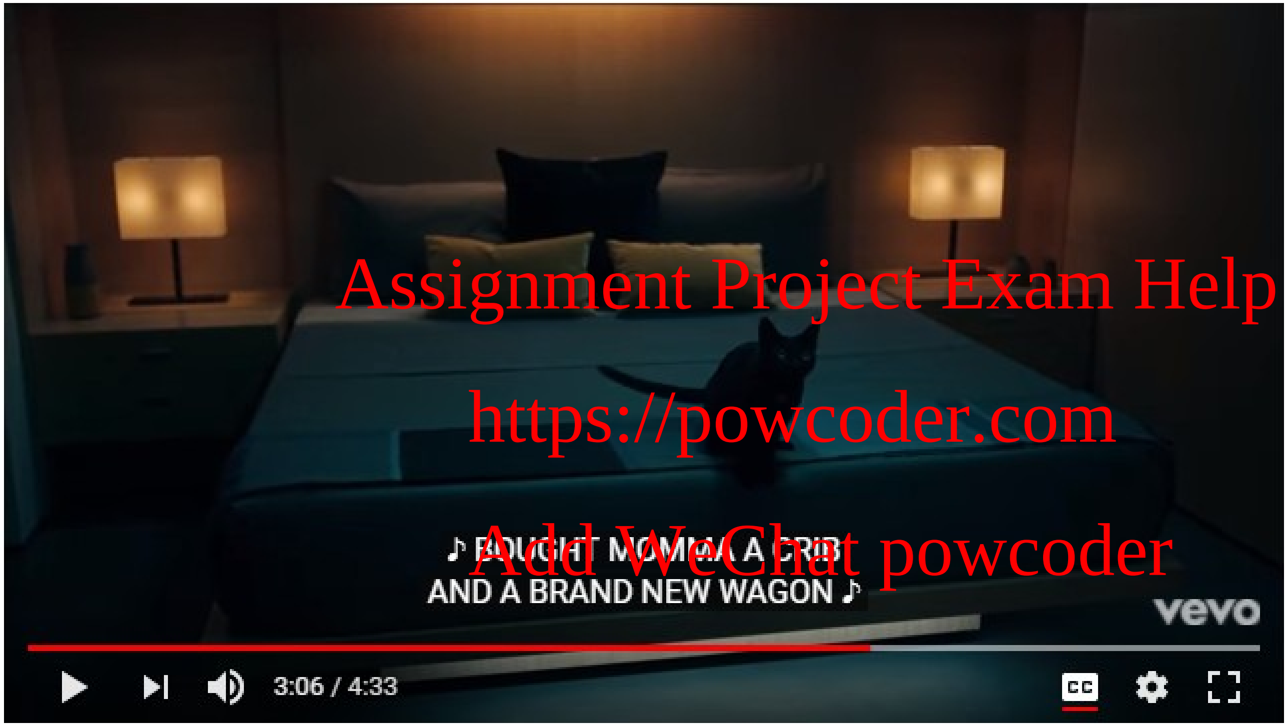




The script drops positive & negative to focus on the explicit emotions.

HARVARD UNIVERSITY

Let’s practice sentiment analysis



The Weeknd - Starboy (official) ft. Daft Punk

Open 6_TidyText_Sentiment_revised.R