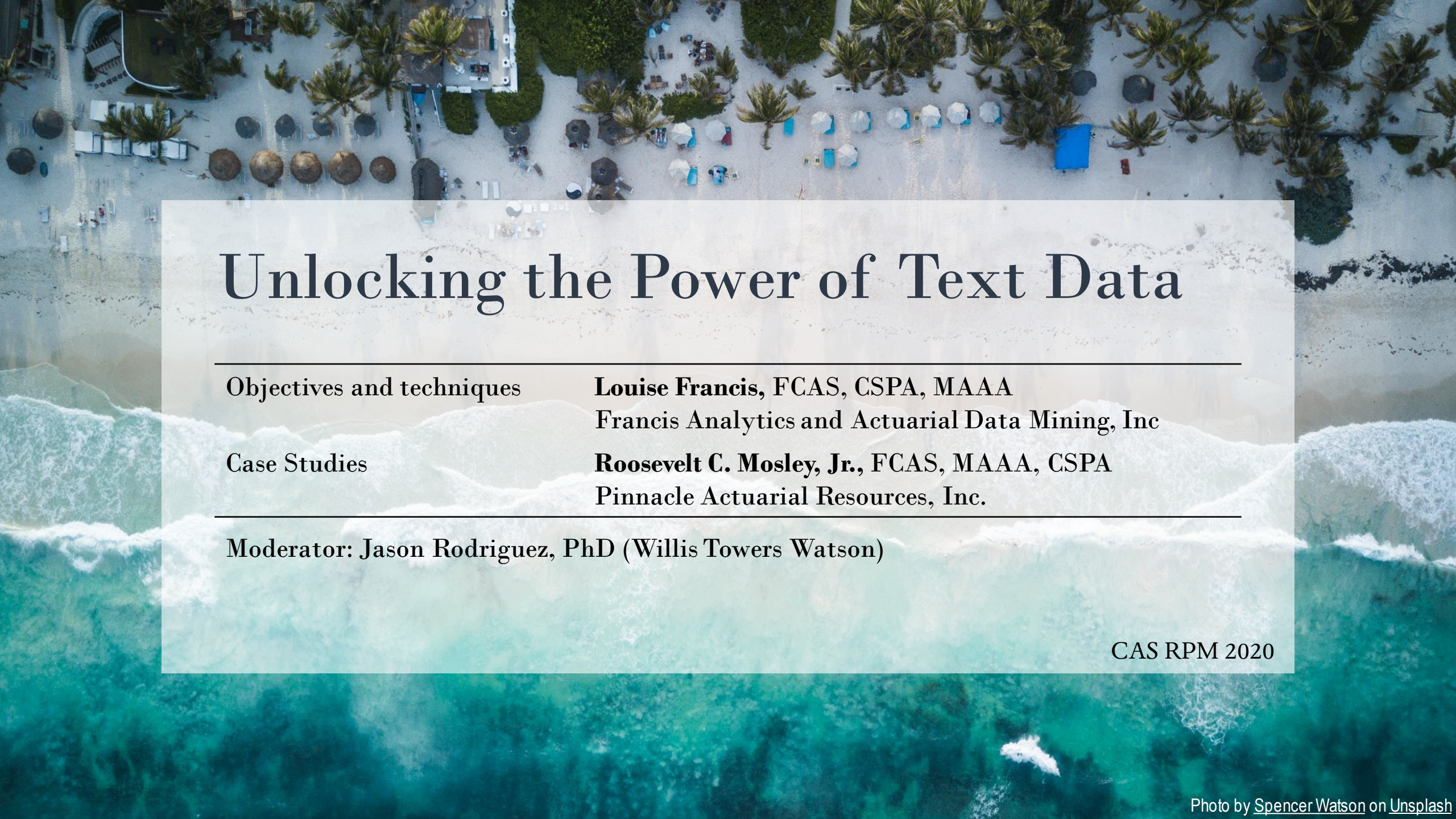


Unlocking the Power of Text Data

Objectives and techniques	**Louise Francis,** FCAS, CSPA, MAAA Francis Analytics and Actuarial Data Mining, Inc
Case Studies	**Roosevelt C. Mosley, Jr.,** FCAS, MAAA, CSPA Pinnacle Actuarial Resources, Inc.

Moderator: Jason Rodriguez, PhD (Willis Towers Watson)

CAS RPM 2020

Poll Question

How familiar are you with text mining?

- Not familiar
- I understand what it is but never tried it.
- I've seen it in action.
- I consider myself an expert.

Text Mining Objectives and Techniques

Louise Francis, FCAS, CSPA, MAAA

Francis Analytics and Actuarial Data Mining, Inc

CAS RPM Seminar 2020

Objectives

- Introduce the concepts underlying text mining
- Introduce common tools used for text mining
- Illustrate with an application using workers compensation data

Open Source Products for Text Mining

- R – a statistical and analytical language with text mining functionality provided by a text mining package tm along with other packages that provide additional capability
- Python – an analytical language used by computer scientists, data scientists and engineers.
- Perl – Historically recognized for its string processing capabilities

Text Data

- Text mining can be applied to many common tasks
  - Internet searches
  - Screening emails for spam
  - Analyzing free form fields in underwriting and claims files
  - Analyzing survey data
  - Analysis of papers, books and articles – such as in the gutenbergr library
- We illustrate with field from a claim file

Claim Data

- WC Claim dataset
- From Crowd Analytix competition: https://www.crowdanalytix.com/contests/predict-costs-of-workers-compensation-claims

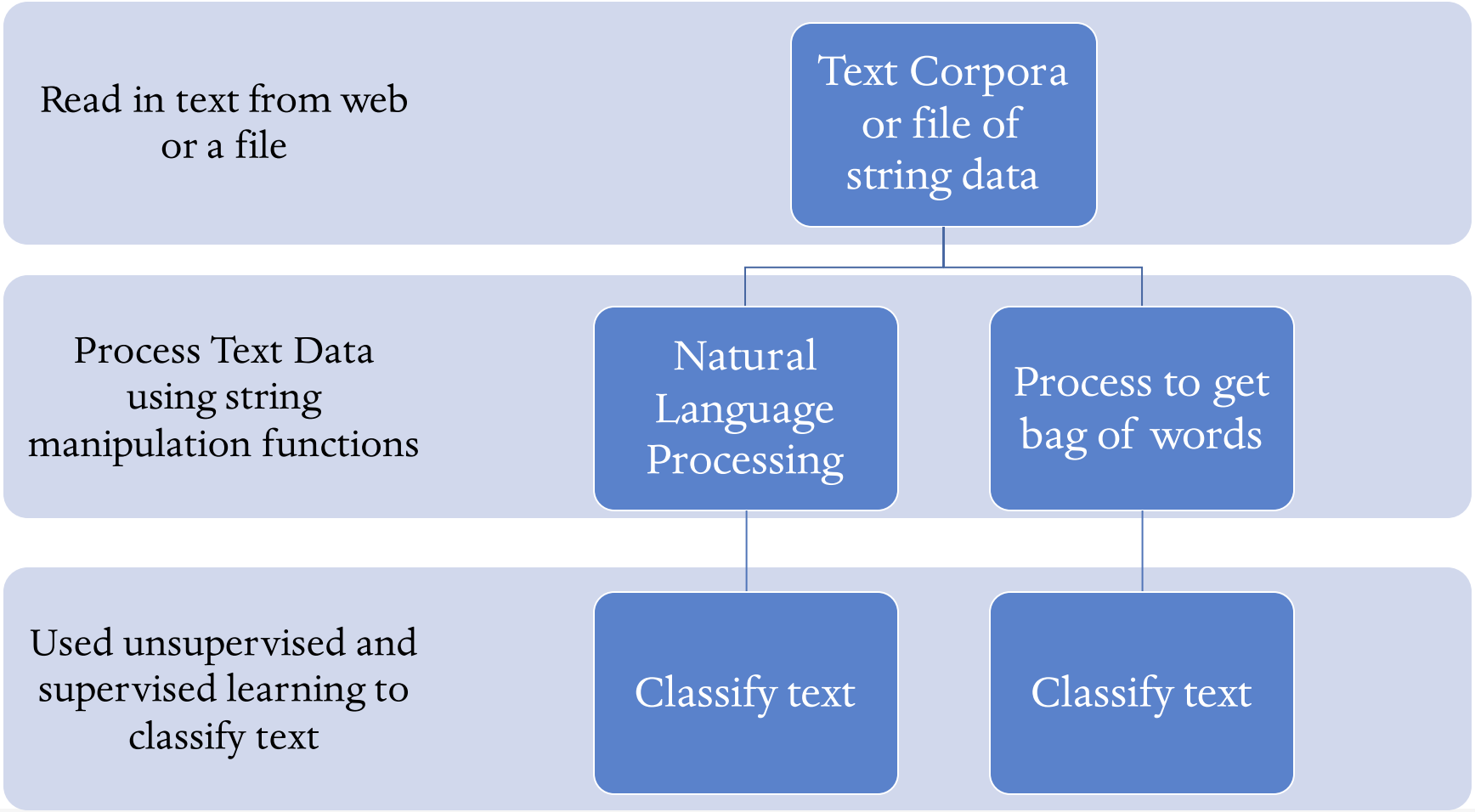
Obs_ID	Dependent	Average Weekly Wage	Body Part	Body Part Code	Cause	Cause Code	Claimant Age	Claimant Gender	Claimant Gender Code	Claimant Hire Date	Claimant Marital Status
Obs_1	98679	500	Pelvis	46	Struck or Injured By	1700	21	Female	F	4/3/2001	
Obs_2	55727	1,037.00	Low Back Area	42	Strain or Injury By	1500		Male	M	5/15/2001	
Obs_3	185833	929	Low Back Area	42	Strain or Injury By	1500	63	Male	M	5/15/2001	Married
Obs_4	98615	1,226.00	Multiple Body Parts	90	Strain or Injury By	1500	49	Male	M		
Obs_5	51396		Other Facial Soft Tissue	18	Miscellaneous Causes	1900	51	Male	M		

Text Variable

- A free form field – how injury occurred

Obs_ID	How Injury Occurred
Obs_2673	EE AD 2 OTHER LABORER WERE MOVING A DEMO WALL THAT STOOD 12
Obs_14692	EMPLOYEE WAS WALKING OVER ROCKY SET WHEN THE ROCKS GAVE WAY.
Obs_14673	EMPLOYEE WAS REMOVING RUBBER MATS FROM THE ROOF OF COURTHOUS
Obs_13477	WHILE RIDING BICYCLE FROM ONE SET TO ANOTHER, HE SKIDDED ON
Obs_578	WALKING DOWN STEPS OF HONEYWAGON, SLIPPED & FELL
Obs_427	EE FAILED TO HEED APPROACHING TRAIN AND TRAIN HIT CAR - ON H
Obs_11523	EE STATED HE WAS WALKING OUTSIDE ON SET - TRIPPED AND FELL O
Obs_5968	THE SPAN SET THAT WAS HOLDING THE I -BEAMS BROKE CAUSING THE
Obs_10433	57-Y/O(ON DOI) SCRIPT SUPERVISOR WHO REPORTED NECK, BACK, RI
Obs_2537	CT 11/01 - 6/19/02 BILATERAL UPPER EXTREM AND SPINE

Text Mining Process



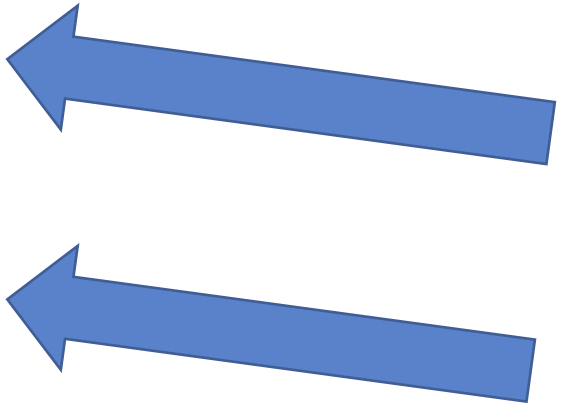
Text Mining Steps

- Data Preprocessing
  - Clean data: remove misspellings, punctuation, numbers, convert to lower case
  - Split individual words from spaces, punctuation
  - Remove stop words
  - Stem words, and replace synonyms
  - Create document term matrix with results
- Data Exploration
- Use analytic techniques to derive meaning
- Use for prediction

String Functions

- Nearly all languages used by actuaries contain string functions
- Some simple string functions can help with data preprocessing in actuarial analyses
- Example – Identification of multiple occurrence claims

Claim_Number
112375-119959-WC-01
112375-119959-WC-02
112375-128321-WC-01
112375-128321-WC-02
112375-128321-WC-03
112375-128321-WC-04
112375-050140-WC-01
112375-050182-WC-01
112375-050218-WC-01



Simple String Functions Let Us:

- Tabulate how many claims there are for each occurrence
- Compute the occurrence number, so data can be aggregated to the occurrence level

Claim_Number	Claimant Number
112375-119959-WC-01	01
112375-119959-WC-02	02
112375-128321-WC-01	01
112375-128321-WC-02	02
112375-128321-WC-03	03
112375-128321-WC-04	04
112375-050140-WC-01	01
112375-050182-WC-01	01

=right(A2,2)

Claim_Number	Occurrence Number
112375-119959-WC-01	112375-119959-WC
112375-119959-WC-02	112375-119959-WC
112375-128321-WC-01	112375-128321-WC
112375-128321-WC-02	112375-128321-WC
112375-128321-WC-03	112375-128321-WC
112375-128321-WC-04	112375-128321-WC
112375-050140-WC-01	112375-050140-WC

=left(A2,15)

Text Processing: Regular Expressions

- A symbolic code convention for string pattern description, used in searching for patterns in text processing
- They can be very helpful in manipulating and using text data
- There can be some variations across languages such as Perl, Python, R

Common Regular Expression Patterns

- There are various shorthand characters to denote types of strings including:
  - 0-9 a digit
  - \d for digit
  - \b for border (blank or punctuation before or after a word
  - a-z lowercase letters and A-Z uppercase letters. Also :alpha: for letters
  - \w for an alphanumeric character
  - \n end of line
  - ^ at beginning denotes beginning of string, $ at the end denotes the end of a string
  - + one or more of the previous pattern
  - * zero or more of the pattern

stringr library

- stringr is an R library for processing strings
- Functions include
  - str_locate – locates the position of a pattern
  - str_detect – used to detect a pattern
  - str_extract – extracts string matching a pattern
  - str_count – counts number of matches to a pattern
  - str_sub – subsets a string using start and end position
  - str_subset – subsets a string using a regular expression pattern
  - str_replace – replaces a string with another string
  - str_c – combines strings

Use regular expression to get claimant number

- pattern="([\\d]{2})$“

At end of string

Escape symbol

2 digits

- Use str_extract to extract last 2 digits
- ClaimantNo=str_extract(ClaimData$Claim_Number,pattern)
- head(ClaimantNo)

Must be character, not factor

[1] "01" "02" "01" "02" "03" "04"

Use Regular Expression to get LOB

- pattern="([:alpha:]{2})“ or pattern="([A-Z]{2})"

2 alphas – (the first 2)

- LOB=str_extract(Claim_Number,pattern)
- head(LOB)

[1] "WC" "WC" "WC" "WC" "WC" "WC"

Subset claim number for Occurrence Number

- OccurrenceNo=str_sub(Claim_Number,1,16)

Start and end position

[1] "112375-119959-WC" "112375-119959-WC"

[3] "112375-128321-WC" "112375-128321-WC"

[5] "112375-128321-WC" "112375-128321-WC"

str_remove to remove the LOB

- All records end with the same LOB so remove it

- pattern="-WC"
- OccurrenceNo=str_remove(OccurrenceNo,pattern)
- head(OccurrenceNo)

[1] "112375-119959" "112375-119959"

[3] "112375-128321" "112375-128321"

[5] "112375-128321" "112375-128321"

Parse words

- In text mining, the words are typically separated out from the rest of the text
- stringr str_split function can be used to split strings

text="EE AD 2 OTHER LABORER WERE MOVING A DEMO WALL THAT STOOD 12"

word_list=str_split(text," ") # also: str_split(text,"\\b+")

word_list

1 or more boundary symbols (space, period, etc.)

```
 [1] "EE"      "AD"      "2"       "OTHER"
 [5] "LABORER" "WERE"    "MOVING"  "A"
 [9] "DEMO"    "WALL"    "THAT"    "STOOD"
[13] "12"
```

Separate words for entire list

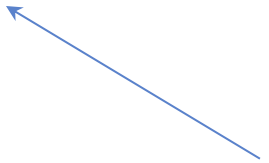
```
text=WCTextData$How.Injury.Occurred

text=text[1:10]

injury_words=str_split(text,"\\b+")

injury_words
```

```
> injury_words=str_split(text,"\\b+")
Error in stri_split_regex(string, pattern, n = n, simplify = simplify,  :
  Syntax error in regexp pattern. (U_REGEX_RULE_SYNTAX)
```



A problem with splitting criteria

Issues with complicated text

- Some terms with complicated punctuation and missing spaces may not split using only boundary condition
- Example:

```
test="57-Y/O(ON DOI) SCRIPT SUPERVISOR WHO REPORTED NECK, BACK,
RI"
split_pattern="(\\W+)"
injury_words=str_split(test,split_pattern)
injury_words
```

```
injury_words
[[1]]
 [1] "57"         "Y"          "O"
 [4] "ON"         "DOI"        "SCRIPT"
 [7] "SUPERVISOR" "WHO"        "REPORTED"
[10] "NECK"       "BACK"       "RI"
```

Removing Punctuation

- Can also remove the complicating punctuation before splitting

```
remove_pattern="(:punct:)+"
text=str_remove(text,remove_pattern)
injury_words=str_split(text,"\\b+")
```

- Split using all non words as boundaries, then output as a matrix

```
split_pattern="(\\W+)"
injury_words=str_split(text,split_pattern)
injury_words_matrix=str_split(text,split_pattern,simplify=TRUE)
injury_words_matrix
write_csv(as.data.frame(injury_words_matrix),"injurymatrix.csv")
```


One or more non-words

The Output Matrix

- str_split matrix has one column for each word in each row. This is closer to what we need for text mining

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17
EE	AD	2	OTHER	LABORER	WERE	MOVING	A	DEMO	WALL	THAT	STOOD	12				
EMPLOYEE	WAS	WALKING	OVER	ROCKY	SET	WHEN	THE	ROCKS	GAVE	WAY						
EMPLOYEE	WAS	REMOVIN	RUBBER	MATS	FROM	THE	ROOF	OF	COURTHOUS							
WHILE	RIDING	BICYCLE	FROM	ONE	SET	TO	ANOTHER	HE	SKIDDED	ON						
WALKING	DOWN	STEPS	OF	HONEYWA	SLIPPED	FELL										
EE	FAILED	TO	HEED	APPROACH	TRAIN	AND	TRAIN	HIT	CAR	ON	H					
EE	STATED	HE	WAS	WALKING	OUTSIDE	ON	SET	TRIPPED	AND	FELL	O					
THE	SPAN	SET	THAT	WAS	HOLDING	THE	I	BEAMS	BROKE	CAUSING	THE					
57	Y	O	ON	DOI	SCRIPT	SUPERVIS	WHO	REPORTED	NECK	BACK	RI					
CT	11	1	6	19	2	BILATERAL	UPPER	EXTREM	AND	SPINE						
WHILE	EE	WAS	WALKING	TO	THE	BASE	CAMP	SHE	SLIPPED	ON	ICE	I				
CT	5	96	3	28	2	CT	3	2	3	28	2	DUE	TO	REP	JOB	DUTIES
WHEEL	SLID	OFF	EDGE	LIFT	GATE	PULLED	EE	DOWNWARD								
EMPLOYEE	WAS	PERFORM	A	STUNT	WHEN	SHE	JAMMED	HER	KNEE	INT						
LIFTED	HEAVY	MATERIAL	ON	SET	IE	JOISTS	PLYWOOD									

Stopwords

- # StopWords.pl

- It is necessary to eliminates stop words and computes the term-document matrix
- Frequently occurring words like the, and, a that often do not contribute to analyzing text word patterns
- a key part is to tabulate the indicator/count of every term - usually a word

Many text mining programs have dictionaries of stopwords that are used. However manual adjustments are often necessary

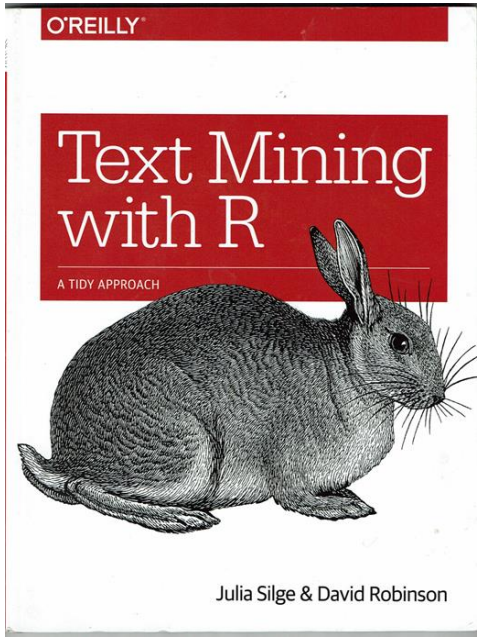
R stopword list

- stopwords("en") gets the list

sw
i
me
my
myself
we
our
ours
ourselves
you
your
yours
yourself
yourselves
he

Popular R text mining libraries

- Two important libraries
  - tm
  - tidytext



Introduction to the **tm** Package
Text Mining in R

Ingo Feinerer

December 12, 2019

Introduction

This vignette gives a short introduction to text mining in R utilizing the text mining framework provided by the **tm** package. We present methods for data import, corpus handling, preprocessing, metadata management, and creation of term-document matrices. Our focus is on the main aspects of getting started with text mining in R—an in-depth description of the text mining infrastructure offered by **tm** was published in the *Journal of Statistical Software* (Feinerer et al., 2008). An introductory article on text mining in R was published in *R News* (Feinerer, 2008).

The tm library

Using tm – getting the data

- Read in or load text data
- Data must be a Corpora so if it is not convert it
- Use tm Vcorpus function (*Vcorpus (VectorSource(text field)))* function to convert character text to a corpus
- txt <-VCorpus(VectorSource(WCText)

Some processing with tm_map function

- Eliminate extra whitespaces
- Convert all text to lower case

```
# Extra whitespace is eliminated by:
txt2 <- tm_map(txt, stripWhitespace)
lapply(txt2[1:10], as.character)
```

Prints 1st 10 lines

```
# Conversion to lower case by:
txt2 <- tm_map(txt2, content_transformer(tolower))
lapply(txt2[1:15], as.character)
```

Note use of content_transformer for string transformation

Remove stopwords

- txt2 <- tm_map(txt2, removeWords, stopwords("english"))
- lapply(txt2[1:15], as.character)

Word stemming

- All words with the same stem (i.e., singular and plural) are replaced with one word, the “stem”
- Example: employee, employees replaced with “employe”
- Dictionary of word stems used

Term Document Matrix

- A Table of indicator variables
- Cycle through every record in the data
- And every word found at least once
- If a word is present, a 1, otherwise a 0

Term Data Matrix

aandb	aaron	aband	abandon	abd	abdom	abdomen	abdomin	abdomina	acciden
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1

Exploratory Analysis – term frequencies

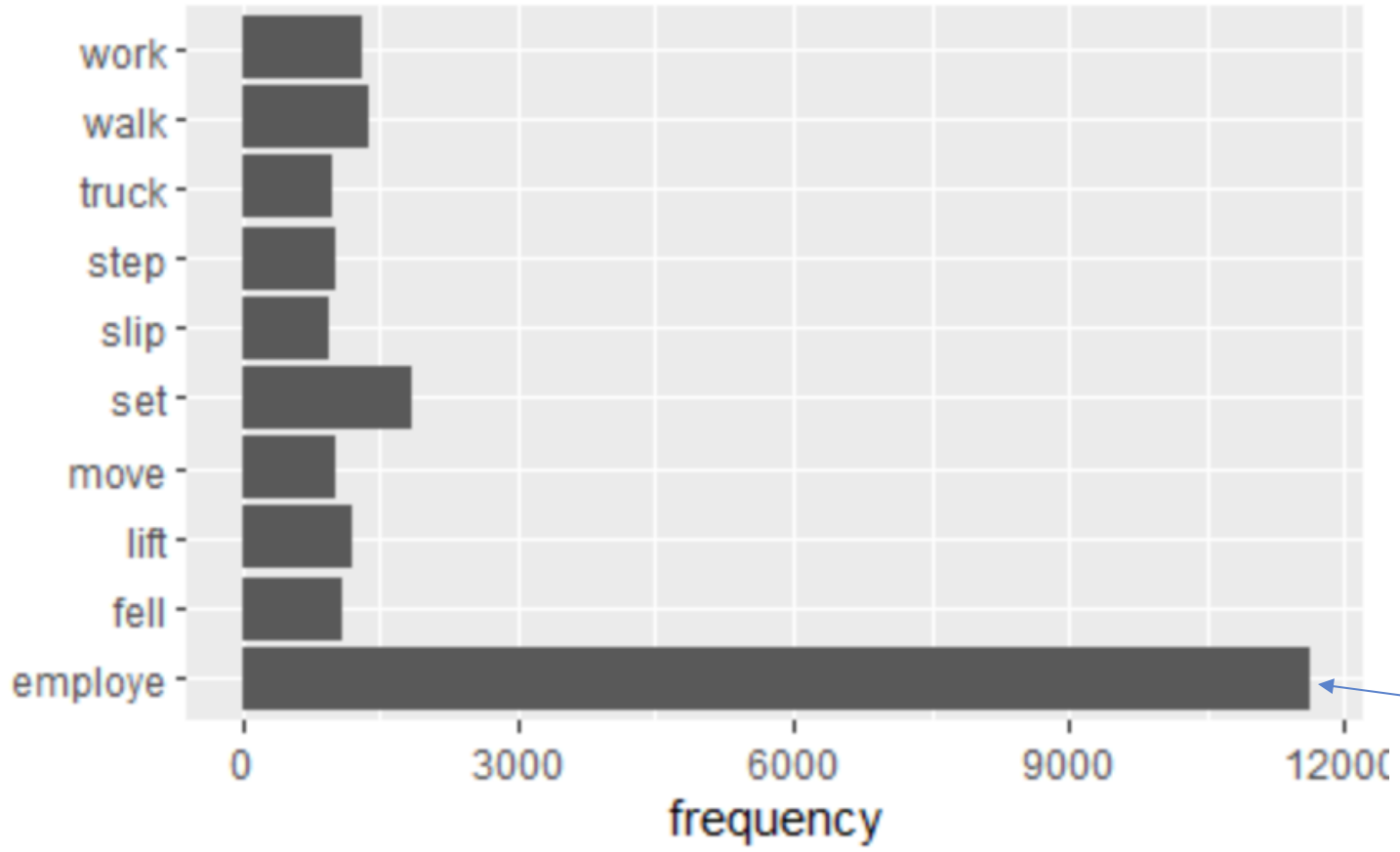
- Count up the number of times each term appears. This is the column sum

```
freq <- colSums(as.matrix(dtm))
```

- Sort in descending order by frequency
- Print out or output a table

term	frequency
employe	11631
set	1867
walk	1371
work	1308
lift	1196
fell	1106
move	1044
step	1029
truck	981
slip	949
cut	942
back	878
felt	805
hit	689

Graph of most Common Terms



Frequent terms may not be informative

Examine words for mis/alternative spellings

- Even after "stemming" there appear to be many versions of some terms

term	frequency
abrad	2
abras	12
abrasio	2
abrupt	5
abscess	1
acc	9
accept	1
access	3
accessori	2
acci	9
accid	62
acciden	8
accident	243
accidenta	9
accidental	13
accidenti	1
accidentlli	1

Sparse terms

- Words that occur only a few times are not useful in interpreting text
- We remove the sparse terms with *removeSparseTerms* in tm
- Select a threshold (say 25)
- Select a sparsity percentage
- All terms that occur fewer times are eliminated

Explore Associations

- Pick a commonly occurring word
- Find which words are associated with it (i.e., occur in the same accident description)
- Use tm function findAssocs to find the associations

findAssocs(dtm, c("employee", "set","walk"), corlimit = 0.15)

```
$employee
    clare hallworth       aid      pond
     0.41      0.29      0.24      0.24
   greens
     0.20
```

Wordclouds

- Size depends on number of times word occurs. Create with wordcloud library

More wordclouds

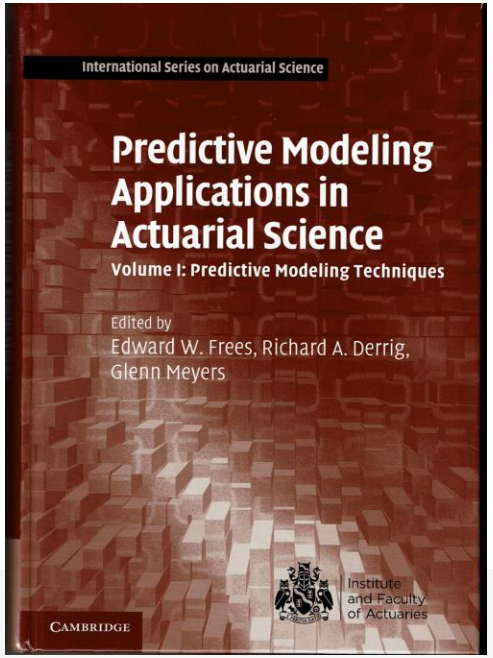
- *wordcloud2* library has easy color features
- Use: *wordcloud2(dataframe,color,size)*

Dimension Reduction: Column-wise and Row-wise

CLAIM NUMBER	DATE OF LOSS	STATUS	INCURRED LOSS
1998001	09/15/97	C	407.81
1998002	09/25/97	C	0.00
1998003	09/26/97	C	0.00
1998004	09/29/97	C	8,247.16
1998005	09/29/97	C	0.00
1998006	10/02/97	C	0.00
1998007	10/10/97	C	0.00
1998008	10/24/97	C	0.00
1998009	10/29/97	C	21,211.66
1998010	10/29/97	C	0.00
1998011	11/03/97	C	0.00
1998012	11/03/97	C	0.00
1998013	11/04/97	C	451.66
1998014	11/04/97	C	0.00
1998015	11/04/97	C	0.00
1998016	11/06/97	C	15,903.66
1998017	11/11/97	C	465.10

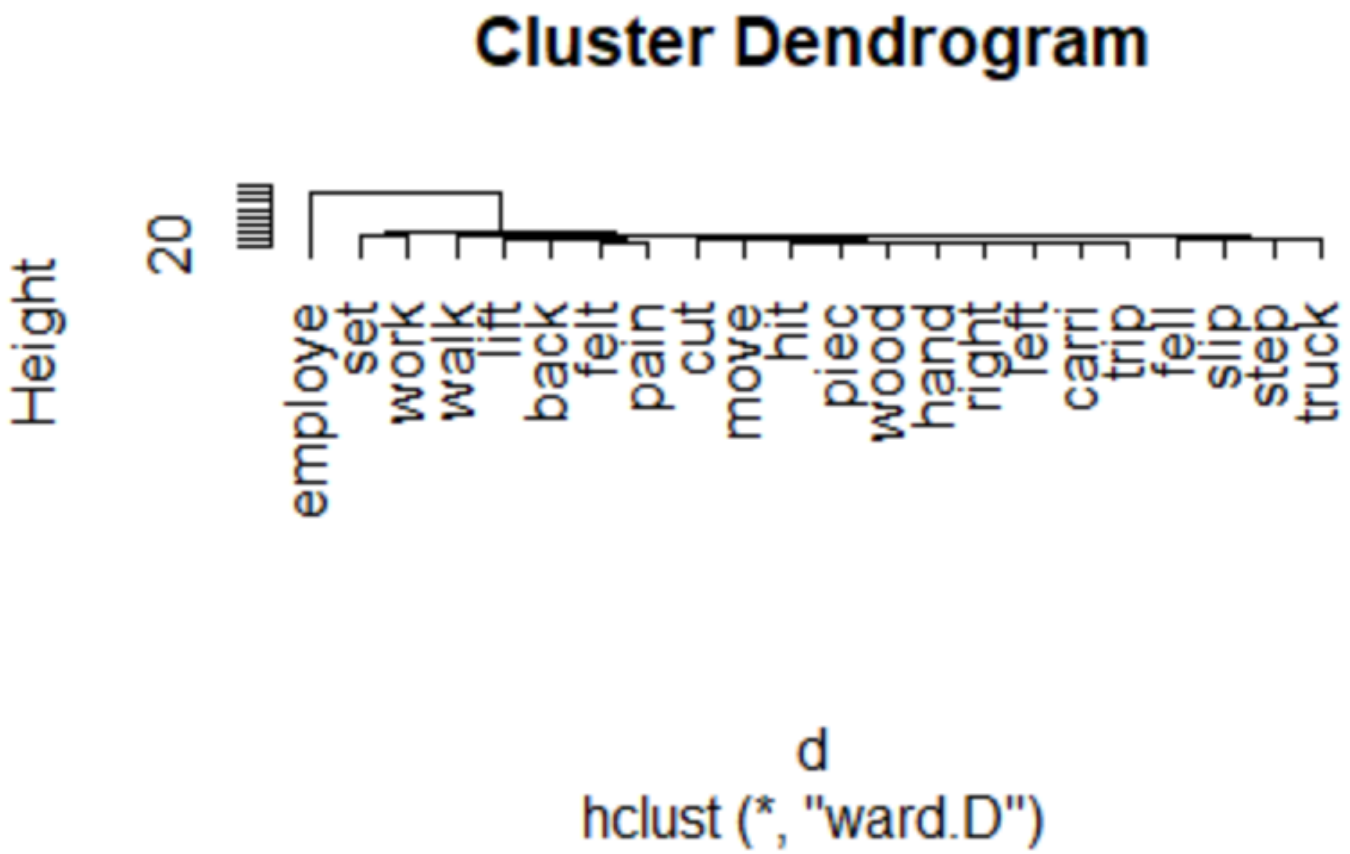
Dimension reduction – Unsupervised Learning

- Next step: Create a new feature (variable) in the data that is a topic or concept and can be used to predict outcomes of interest
- Cluster records with correlated or similar terms : these one topic code
- Use unsupervised learning or dimension reduction to do this
- Use clustering and Principal Components
- See chapter 12 Predictive Modeling Applications



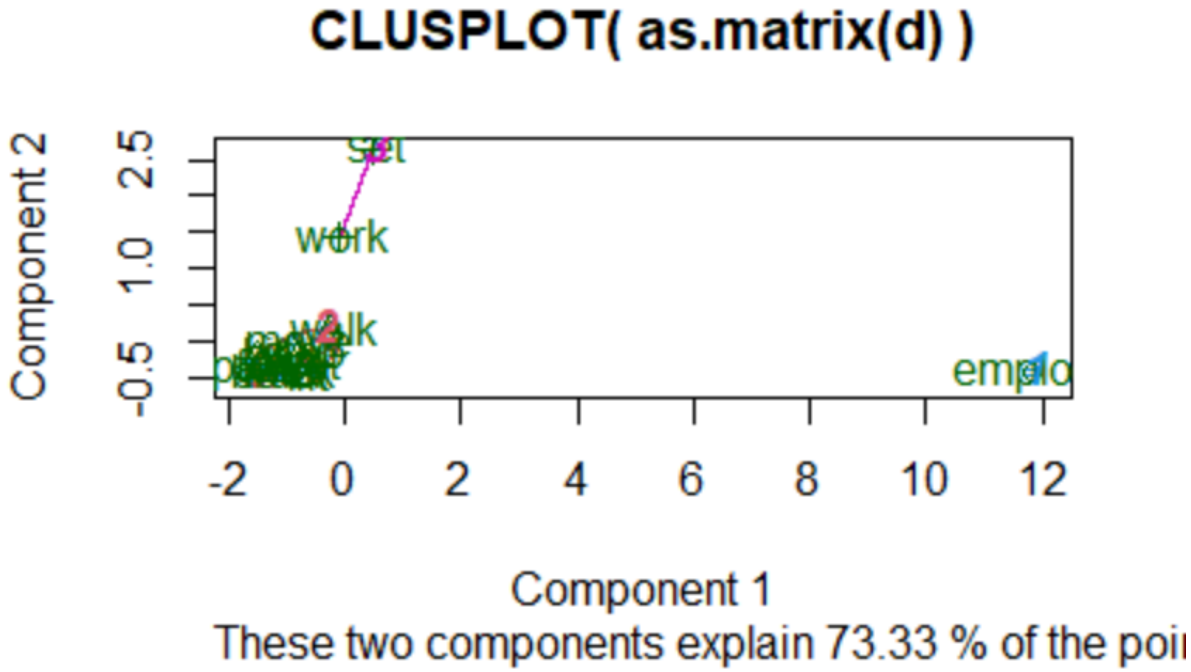
Clustering

- Use library *cluster*
- Hierarchical clustering



kmeans clustering

- Used *pam* function
- Partition against medioids



Principal Components

- Reduces columns to smaller number of variables which are weighted sums of variables
- Uses correlations between variables for computing weights
- R *prcomp* and *princomp* functions can be used

Interpretation of Principal Components

- Use loadings of terms on each component to understand them.
- For some components, only a couple of variables have high load

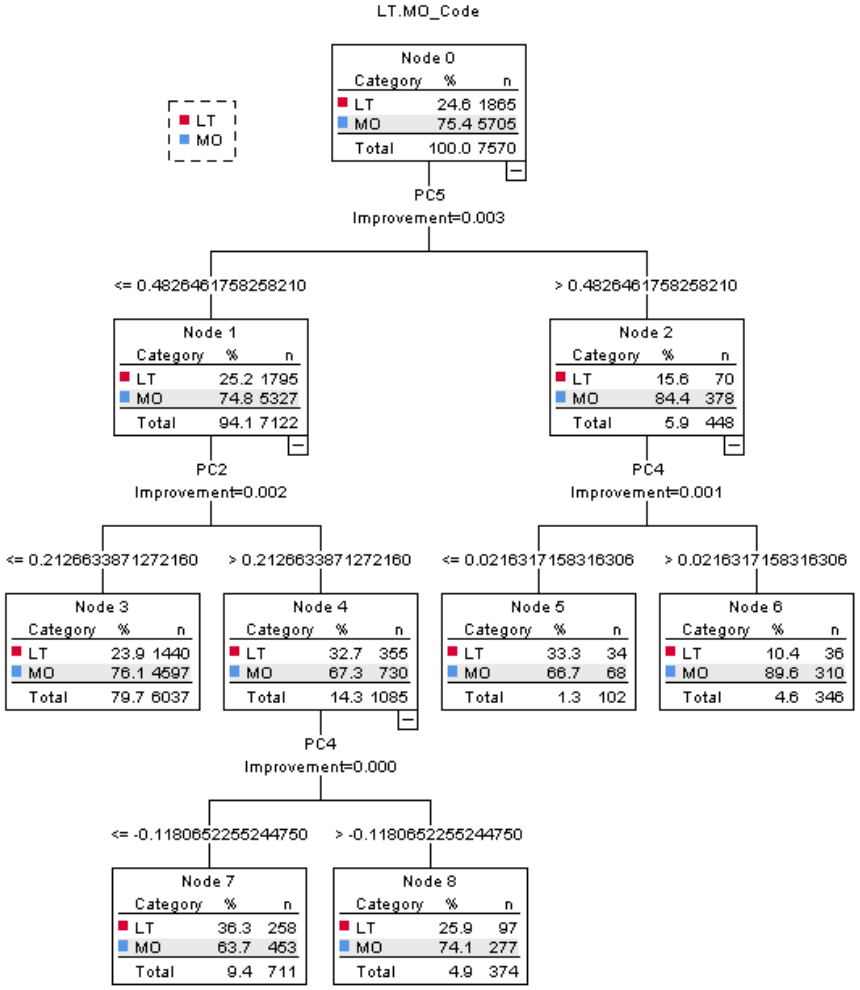
	PC1	PC2	PC3
back	-0.037493905	0.07928627	-0.088664055
cut	-0.029294063	0.07223889	-0.007073785
employe	0.984005841	0.15382928	-0.023449034
fell	-0.060695169	0.04421962	0.165036621
felt	0.018047664	0.04076044	-0.203406887
hit	-0.020063579	0.01732019	0.024555957
lift	-0.009904502	0.08619897	-0.345057679
move	0.010737545	-0.02104473	-0.110330958
pain	-0.011223116	0.05499100	-0.162006012
set	0.117174292	-0.84954031	0.075911527
slip	-0.011177351	0.05111800	0.164044410
step	0.023953223	0.02643917	0.323665141
truck	-0.003494094	0.09555338	0.008637332
walk	0.059448019	-0.08303083	0.716554457
work	0.082111321	-0.45712413	-0.337466341

Components 4 and 5

- Loadings on Last 2 more evenly distributed

	PC4	PC5
back	-0.20185558	-0.0246538865
cut	0.36111567	0.4907097819
employe	0.02396862	0.0396113778
fell	0.09046113	-0.0082484808
felt	-0.31315119	-0.0335771088
hit	0.04209558	-0.0003631697
lift	-0.57554928	-0.0651295914
move	-0.05328359	0.2517837664
pain	-0.24492210	-0.0166290598
set	-0.25837083	0.3640824183
slip	0.15487994	0.1064319755
step	-0.10416055	-0.2810495821
truck	-0.09017977	-0.1739369709
walk	-0.24672391	-0.2264335834
work	0.39443079	-0.6192891981

Components can be used in a predictive model



Additional Resources

- Francis, "Taming Text: An introduction to text mining", www.casact.org
- Francis L, Flynn M, "Text Mining Handbook", www.casact.org
- Mosley R, "Social Media Analytics: Data Mining Applied to Insurance Twitter Posts", www.casact.org
- Vignette for *tm* library on cran.r-project.org web site
- *stringr* help file and vignettes on cran.r-project.org web site

Unlocking the Power of Text Data – Case Studies

Homeowner Claims and Social Media

Commitment Beyond Numbers

Roosevelt C. Mosley, Jr., FCAS, MAAA, CSPA

July 29, 2020

PINNACLE
ACTUARIAL RESOURCES, INC.

Unlocking the Power of Text Data

- Case studies
  - Homeowner claims
  - Social media
- Data
  - Background
  - Processing
- Analysis
  - Themes
  - Sentiments
  - Challenges

Homeowner Claims

Commitment Beyond Numbers

PINNACLE
ACTUARIAL RESOURCES, INC.

Poll Question 2

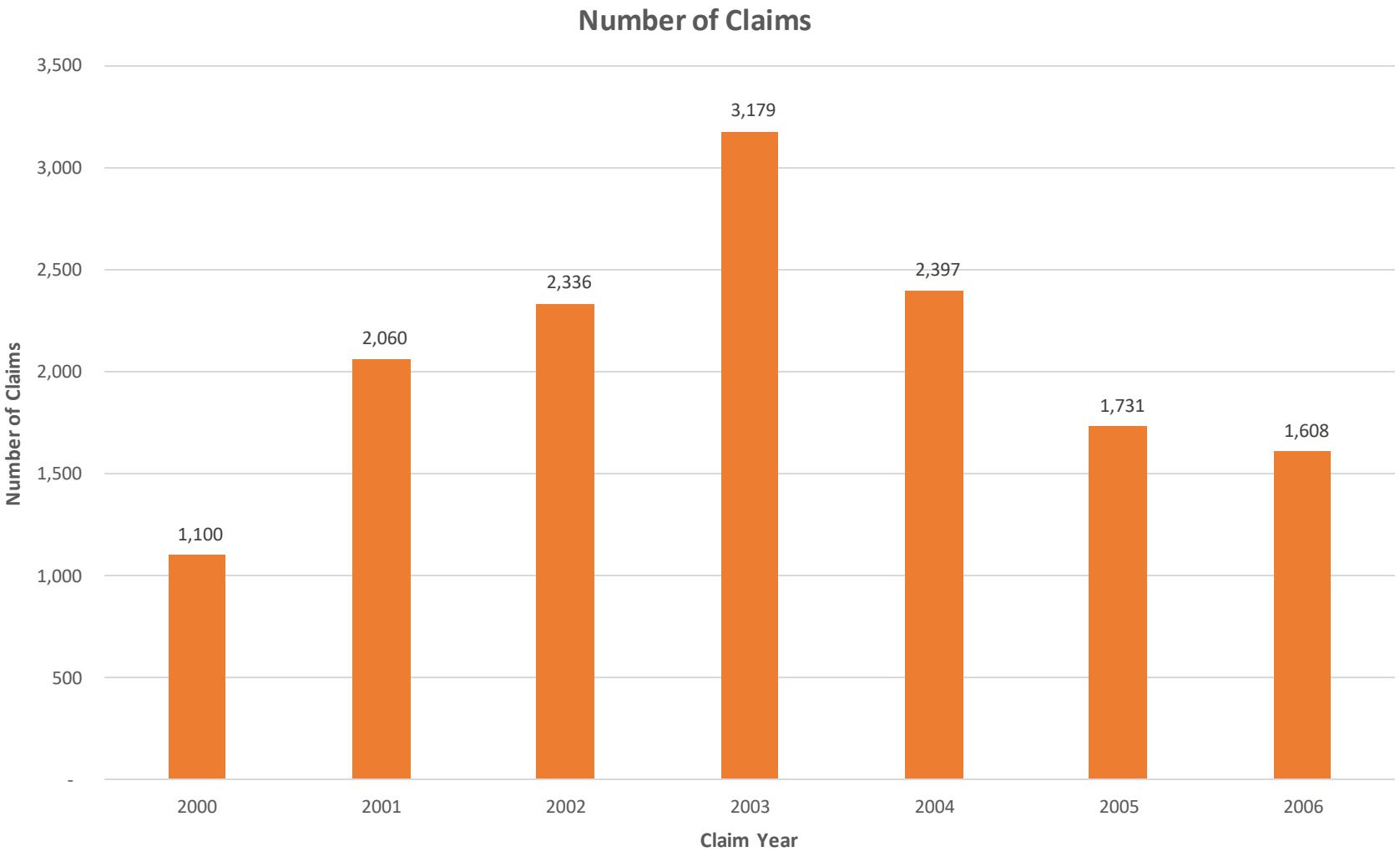
- In what function do you think text analytics can have the largest impact?
  - Underwriting
  - Ratemaking
  - Claims
  - Marketing
  - Customer service
  - Other

Data Description – Homeowner Claims

- Homeowners data
  - 85,000 claim transactions
  - 14,000 unique claims

- Key fields
  - Claimant
  - Report date
  - Accident date
  - Transaction date
  - State
  - Payment amounts
  - Reserve changes
  - Salvage indicator
  - Subrogation indicator
  - Claim status
  - **Claim description**

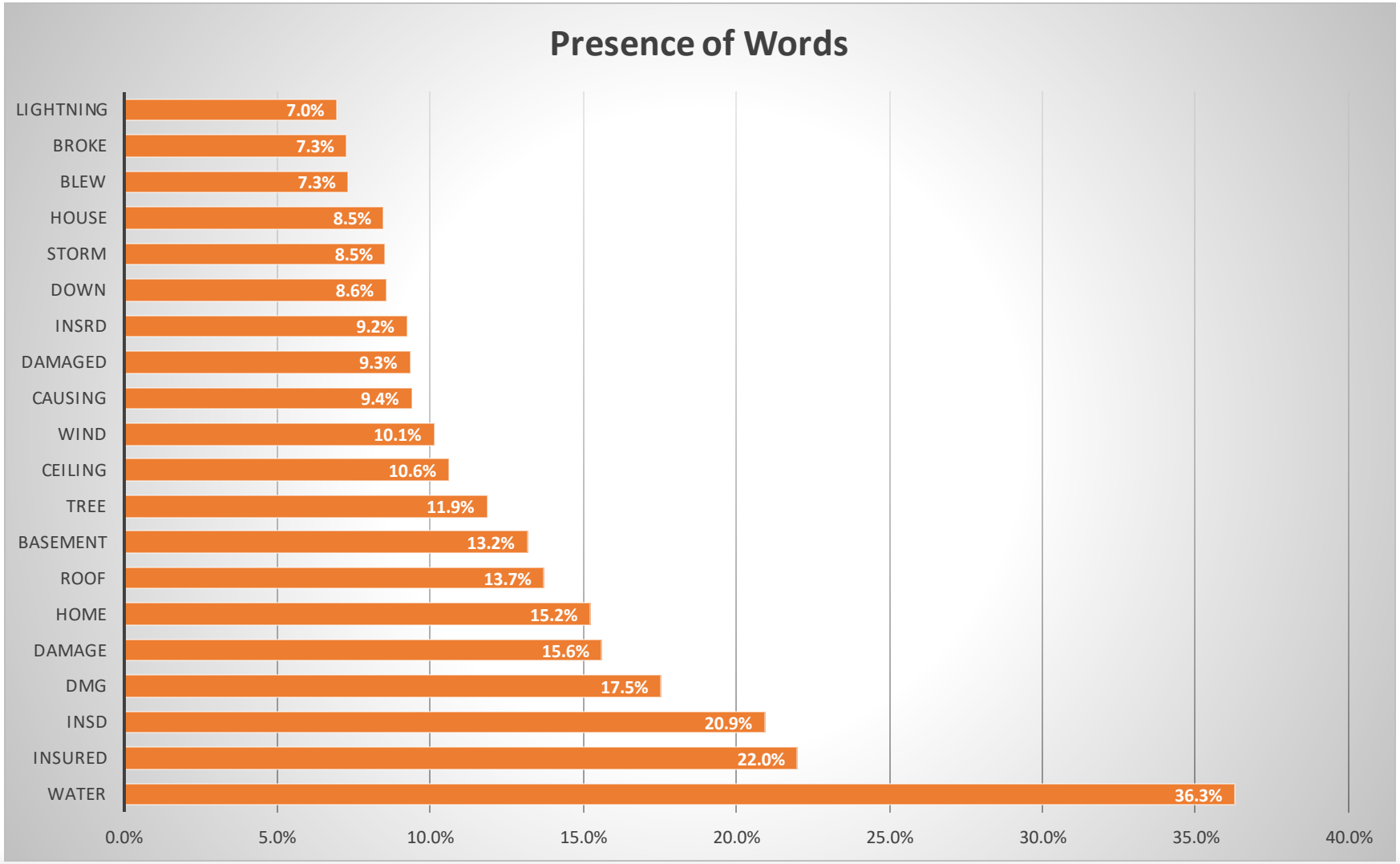
Homeowner Claims – Number of Claims



Word Indicators

- Parse text fields to identify words that are present
- Add indicators to table to indicate which keywords are present in each claim description
- Adjust indicators for misspellings, abbreviations and tenses

Homeowner Claims - Presence of Words

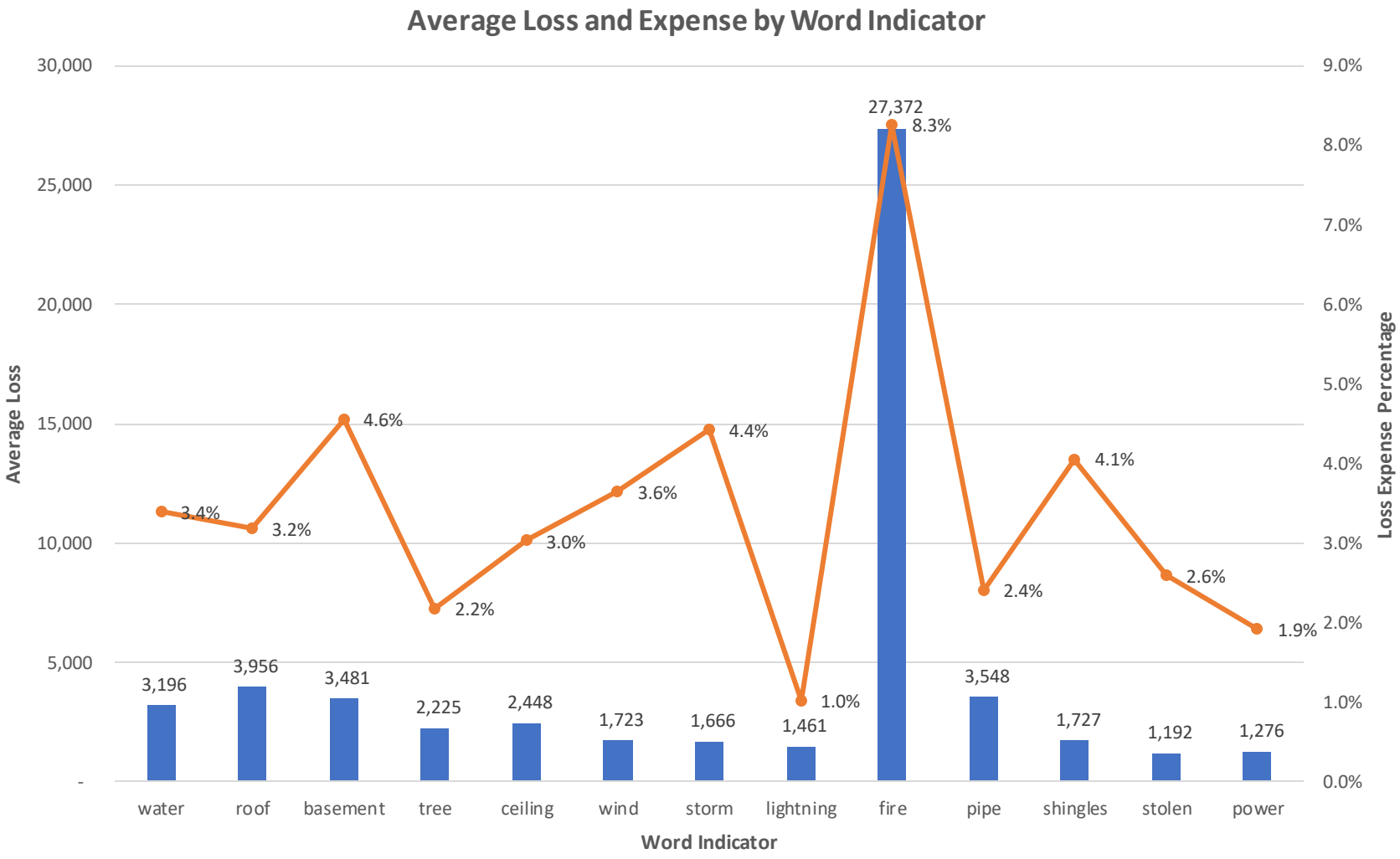


Word Indicators

- Add indicators to original table to indicate which keywords are present in each claim description
- 94 keywords

water_ind	insured_ind	insd_ind	dmg_ind	damage_ind	home_ind	roof_ind	basement_ind	tree_ind	ceiling_ind	wind_ind
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	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1
1	0	1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	1	0	0	0
0	1	0	0	1	0	1	0	0	0	0
0	0	0	0	1	0	1	0	1	0	0
0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	1	0	1	1	1
0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	1	0	0	0	0

Average Loss and Expense by Word Indicator

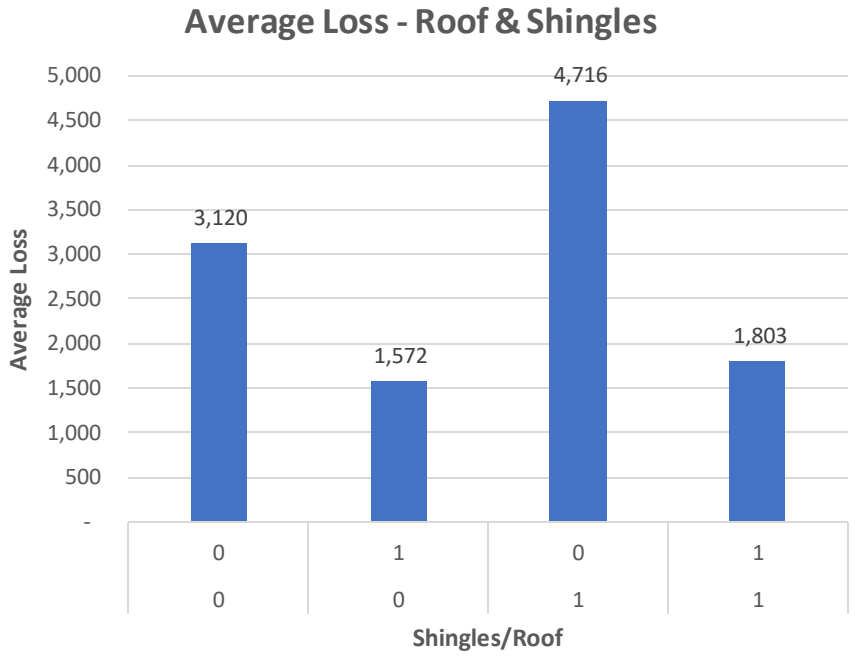


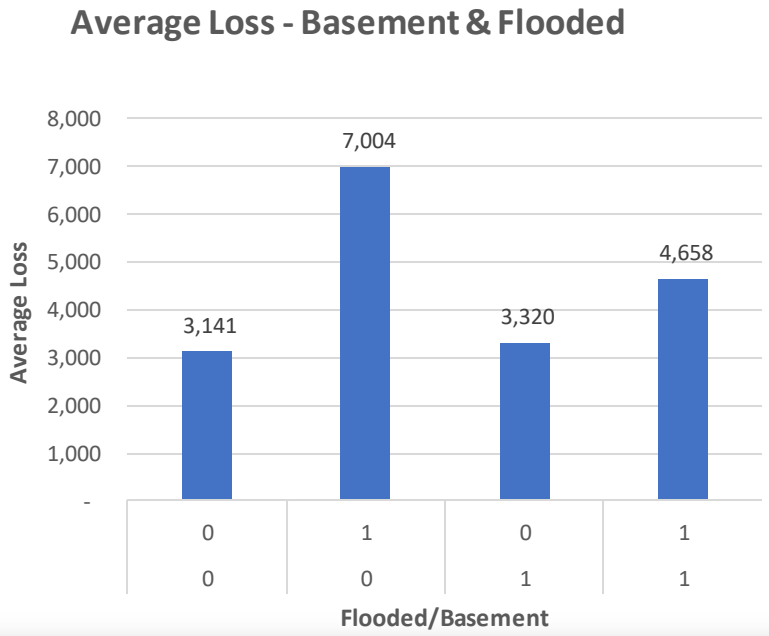
Correlation Results

Cramer's V	Variable 1	Variable 2
0.475	wind_ind	blew_ind
0.391	tree_ind	fell_ind
0.375	roof_ind	shingles_ind
0.349	water_ind	basement_ind
0.309	dmg_ind	causing_ind
0.304	water_ind	ceiling_ind
0.293	ceiling_ind	bathroom_ind
0.277	wind_ind	shingles_ind
0.275	basement_ind	flooded_ind
0.244	basement_ind	sump_ind
0.233	tree_ind	down_ind
0.233	roof_ind	wind_ind
0.229	ceiling_ind	room_ind
0.229	tree_ind	large_ind
0.227	water_ind	damage_ind
0.226	ceiling_ind	upstairs_ind
0.224	water_ind	pipe_ind
0.223	tree_ind	neighbors_ind
0.220	water_ind	floor_ind
0.214	water_ind	causing_ind

- Calculate pairwise correlation for every combination of keywords
- Themes begin to emerge
- Only compares word pairs, so still do not see the full picture

Correlation Results





Clustering/Segmentation

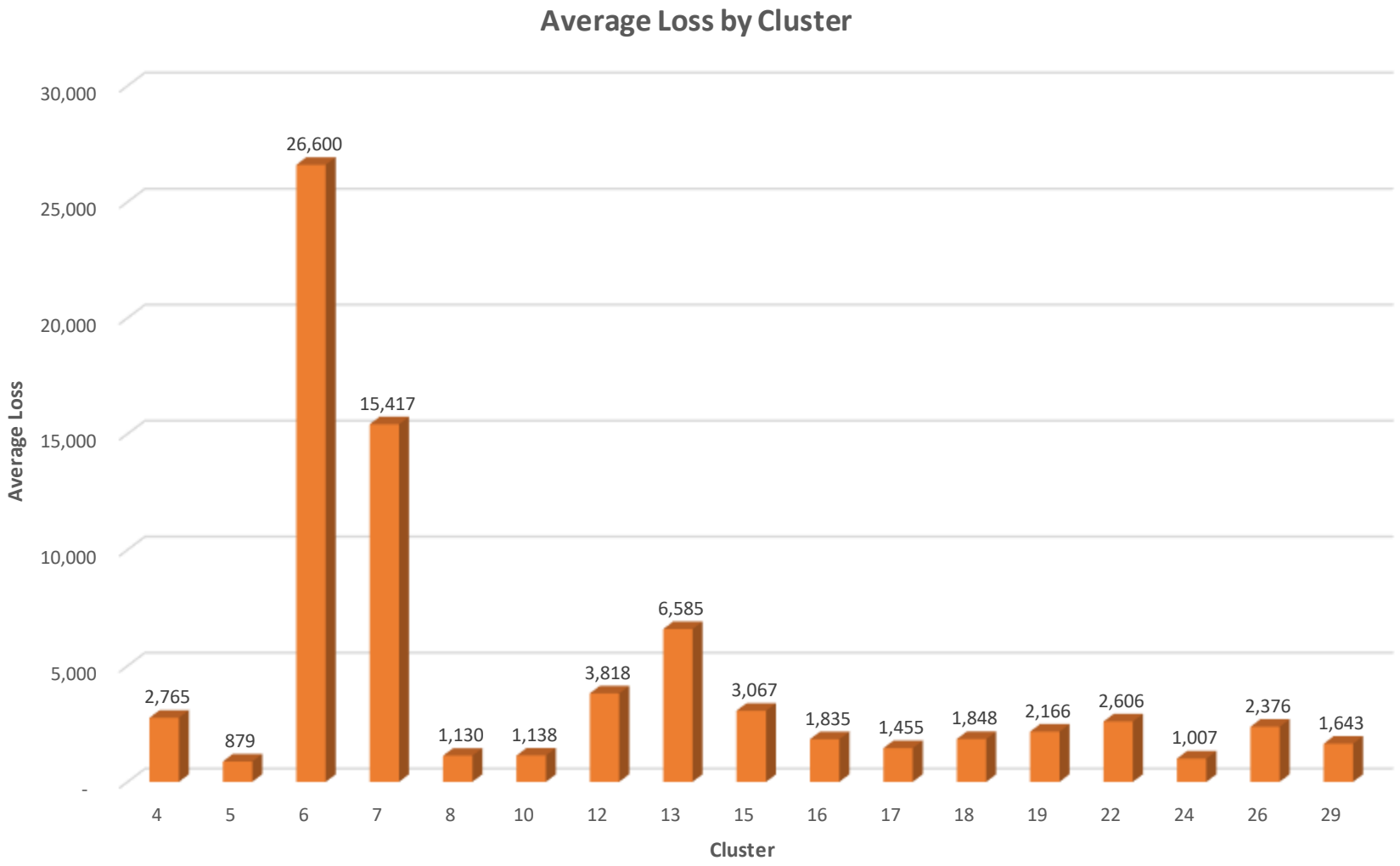
- Unsupervised learning technique
- Groups data into set of discrete clusters or contiguous groups of cases
- Performs disjoint cluster analysis on the basis of Euclidean distances computed from one or more quantitative input variables and cluster seeds
- Data points are grouped based on the distances from the seed values
- Objects in each cluster tend to be similar, objects in different clusters tend to be dissimilar

Cluster Lift

$$\text{Cluster Lift (word)} = \frac{\text{Percentage of claim descriptions in a cluster that include word}}{\text{Percentage of all claim descriptions that include word}}$$

Cluster	water_ind	basement_ind	fire_ind	pipe_ind	kitchen_ind	bathroom_ind	shingles_ind	struck_ind	stolen_ind
1	1.226	0.000	0.000	6.653	0.000	3.684	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	23.899
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	23.899
4	0.000	0.059	0.000	0.000	0.000	0.000	0.142	1.178	0.314
5	0.000	0.000	0.316	0.000	0.000	0.000	0.000	1.088	8.464
6	0.287	0.455	21.343	0.068	1.963	0.225	0.037	0.038	0.000
7	0.920	0.407	7.232	0.907	1.950	0.502	0.000	0.000	0.000
8	0.017	0.042	0.429	0.000	0.101	0.000	0.102	1.267	7.553
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	3.730	0.000
10	0.130	0.090	0.344	0.101	0.108	0.112	0.000	0.452	9.535
11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	23.899
12	2.711	6.636	0.213	0.562	0.000	0.104	0.000	0.105	0.112
13	1.034	1.113	3.247	0.736	0.889	1.019	0.696	0.103	0.991
14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
15	2.678	5.584	0.062	2.272	0.562	0.459	0.195	0.000	0.000
16	0.952	0.789	0.000	1.022	0.264	0.063	0.061	0.085	7.317
17	0.160	0.063	0.138	0.020	0.151	0.089	0.087	12.668	0.000
18	0.652	0.700	0.000	0.307	0.356	0.115	0.211	0.022	0.024
19	0.267	0.021	0.013	0.035	0.174	0.077	5.468	0.104	0.000
20	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
21	0.334	0.000	0.000	1.815	0.000	0.000	0.000	0.000	2.173
22	2.740	0.885	0.012	4.494	5.020	6.511	0.409	0.012	0.000
23	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
24	0.027	0.000	1.010	0.148	0.000	0.000	0.000	1.160	8.320
25	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
26	0.344	0.055	0.000	0.081	0.218	0.135	4.999	0.364	0.000
27	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	23.899
28	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
29	0.660	0.550	0.816	0.819	0.770	0.793	0.553	1.721	1.593
30	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	23.899

Average Loss by Cluster



Association Analysis

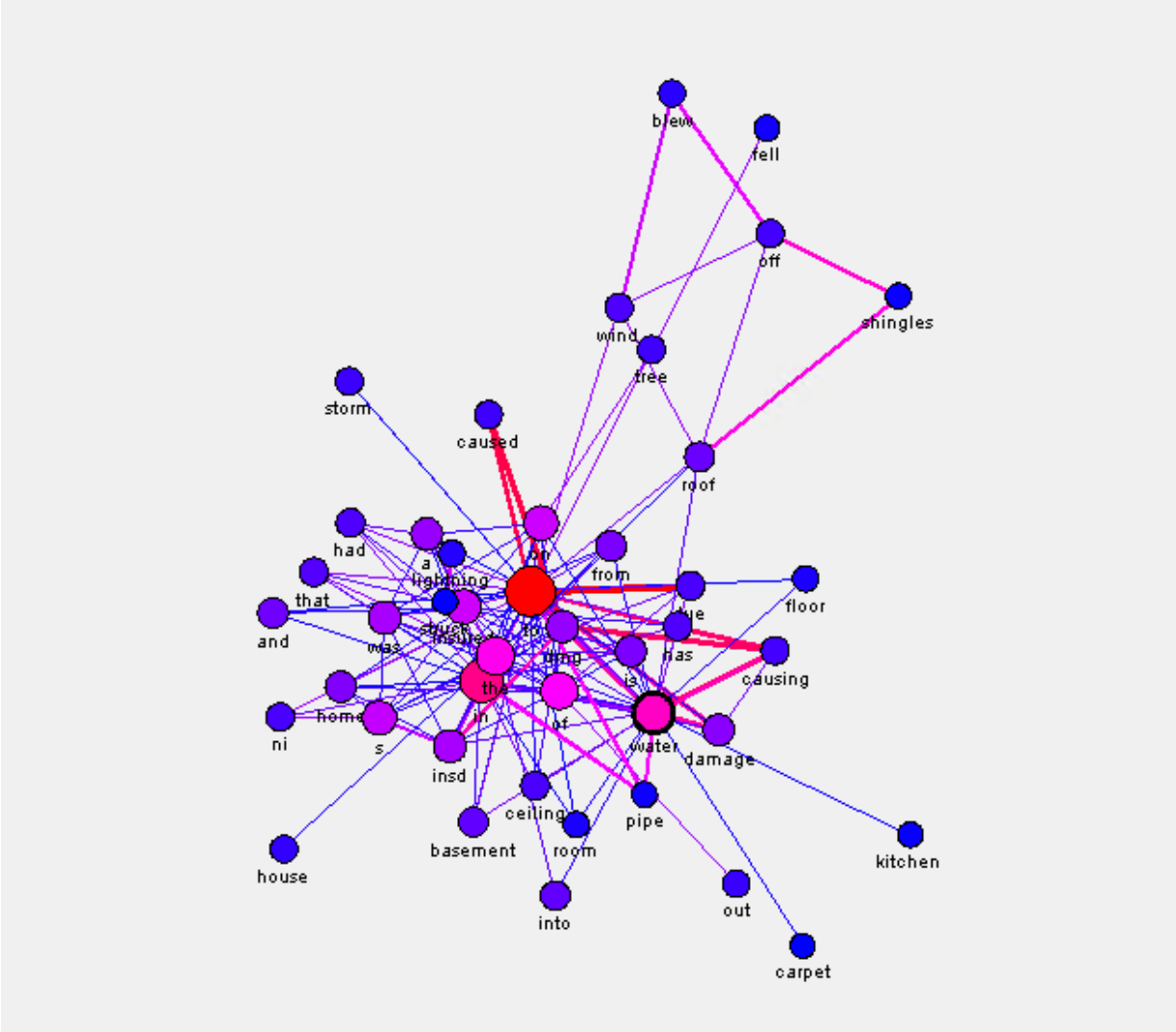
- Background in market basket analysis
- Identification of items that occur together in the same record
- Produces event occurrence as well as confidence interval around the occurrence likelihood
- Can lead to sequence analysis as well, which considers timing and ordering of events

$$\text{Support} = \frac{\text{Transactions that contain items A \& B}}{\text{All transactions}}$$

$$\text{Confidence} = \frac{\text{Transactions that contain items A \& B}}{\text{Transactions that contain item A}}$$

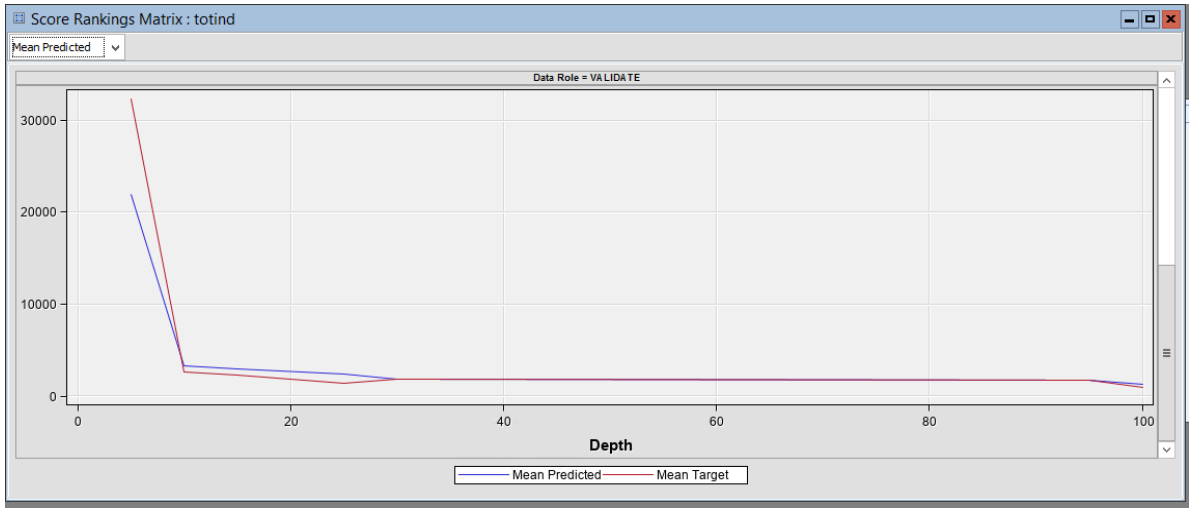
$$\text{Expected Confidence} = \frac{\text{Transactions that contain item B}}{\text{All transactions}}$$

Association Analysis – Link Graph

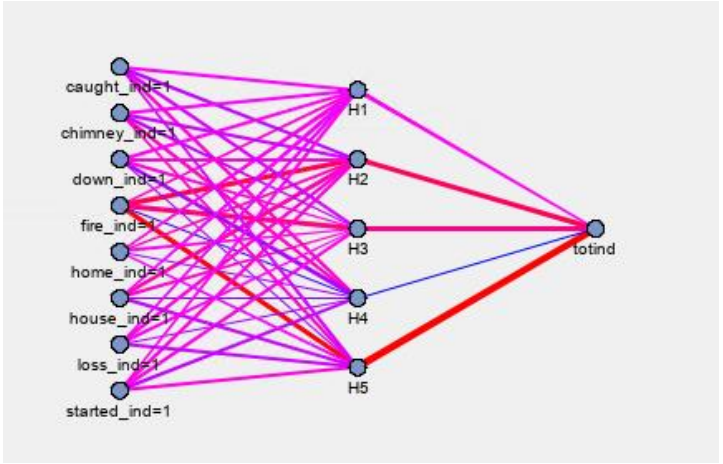


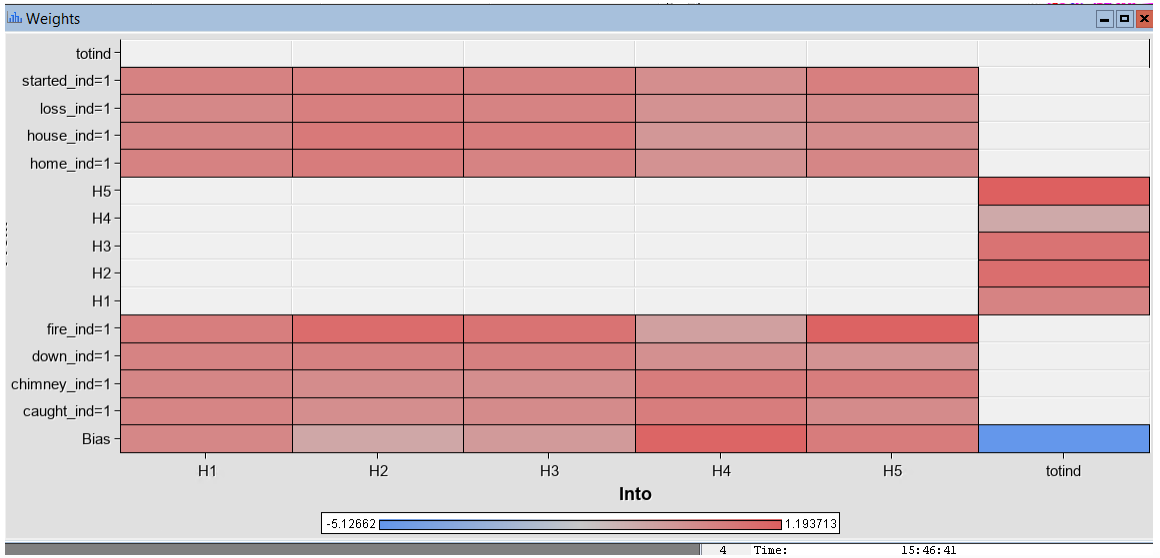
Map	Rule
RULE15	water ==> damage
RULE17	water ==> basement
RULE23	water ==> ceiling
RULE49	wind ==> blew
RULE62	water ==> in ==> basement
RULE125	lightning ==> struck
RULE127	shingles ==> off
RULE132	causing ==> water ==> dmg
RULE133	dmg ==> ceiling

Analytics Model Enhancement



Improve model prediction by adding unstructured elements





Social Media

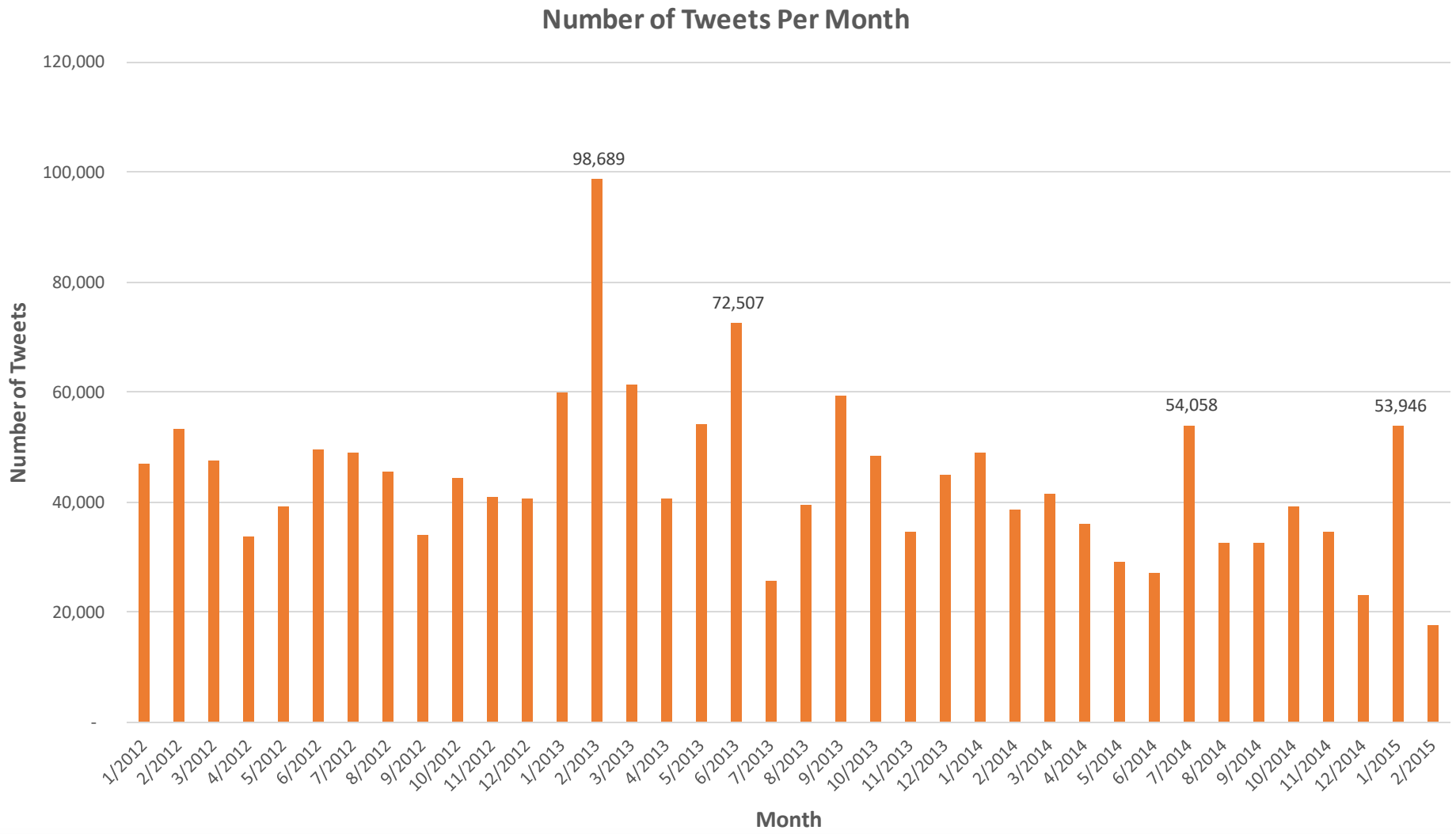
Commitment Beyond Numbers

PINNACLE
ACTUARIAL RESOURCES, INC.

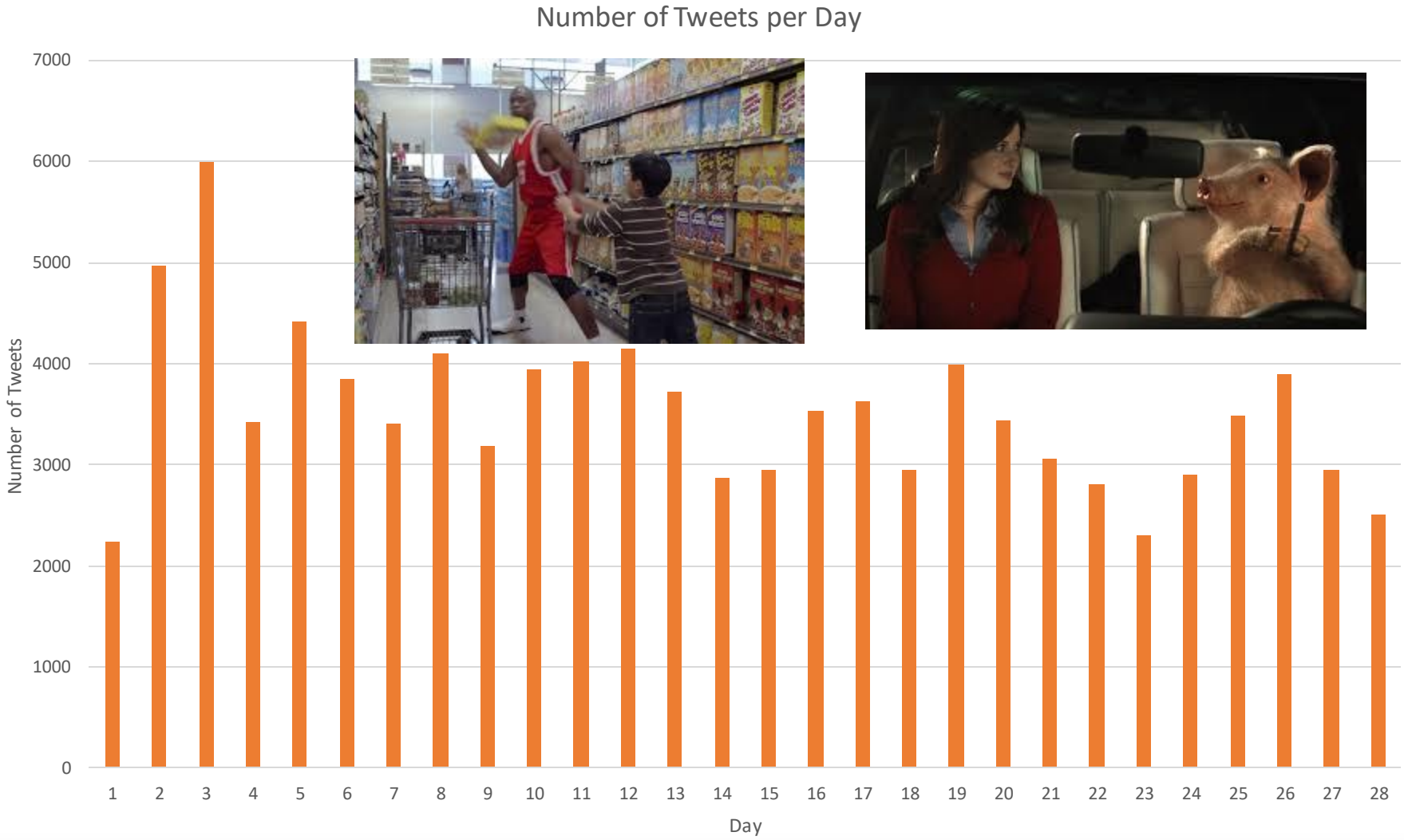
Social Media Data

- Twitter data - over 6 million insurance tweets total (January 2012 to present)
  - 1.6 million GEICO tweets
- Data
  - Content
  - Recipient
  - Sender
  - Language
  - Place of origination
  - Link to a picture of user
  - Latitude and longitude of the user
  - Date and time
  - Device/platform
- Advantages of social media data
  - Unfiltered
  - Broad view of non-customer reactions
  - Facilitates more timely analysis of trends
- Sources
  - Third party data aggregators (Hootsuite, GNIP)
  - API
  - Company developers
  - Screen scraping

Tweets per Month – All Companies



Tweets per Day – February 2013



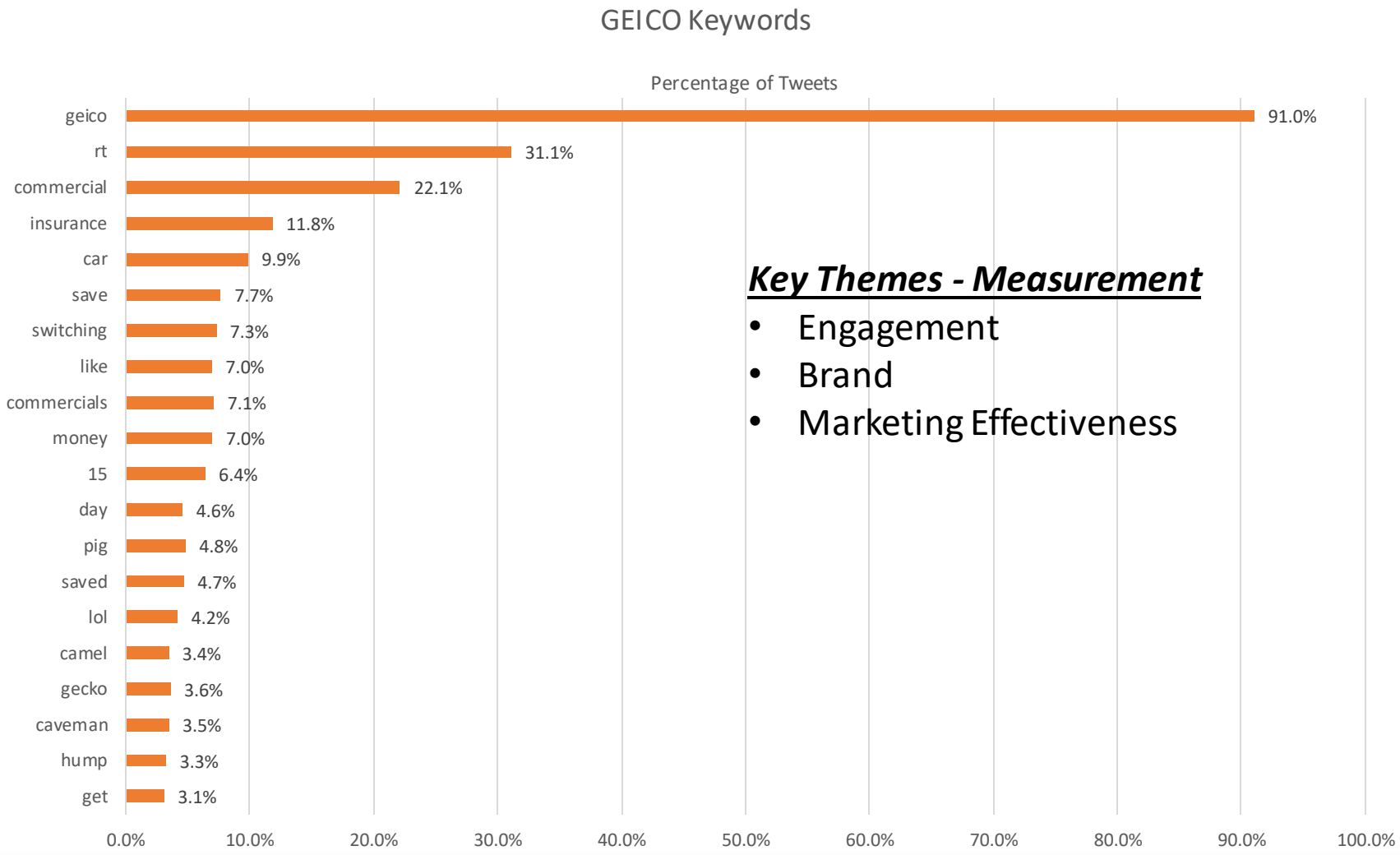
Data Processing Steps

- Remove punctuation and symbols (retain @ and #)
- Parse the tweet (35 words worked for Twitter – will need many more for other sources)
- Change table structures from tweets in rows to tweets in columns – keep indicator of order
- Correct spelling errors
- Add word indicators

Tweet ID	User	Tweet	Word1	Word2	…	Word35
1	@mosley	Text of tweet	W1	W2	…	W35

Tweet ID	Word Order	Word
1	1	Word1
1	2	Word2
…	…	…
1	35	Word35

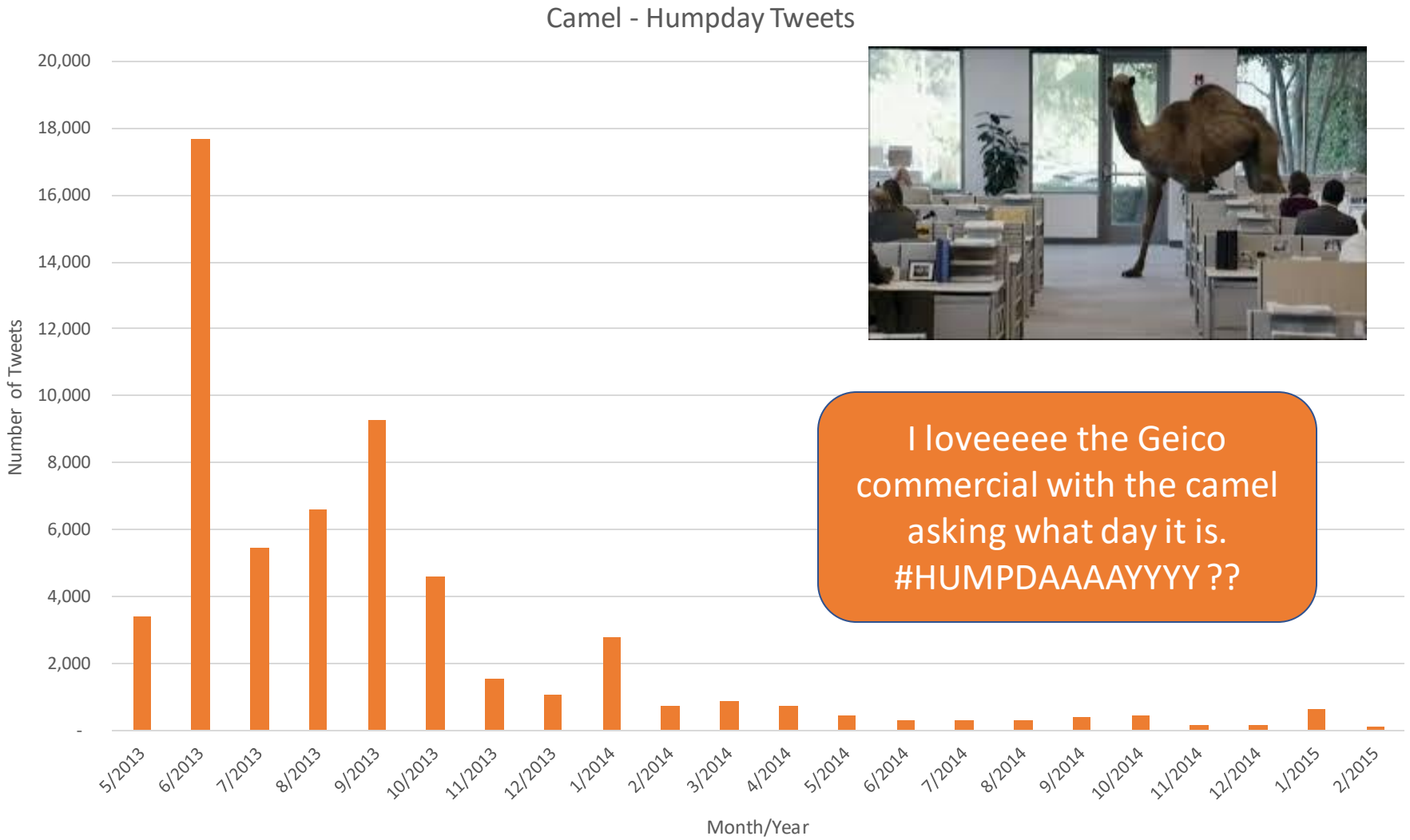
Top 10 Keywords



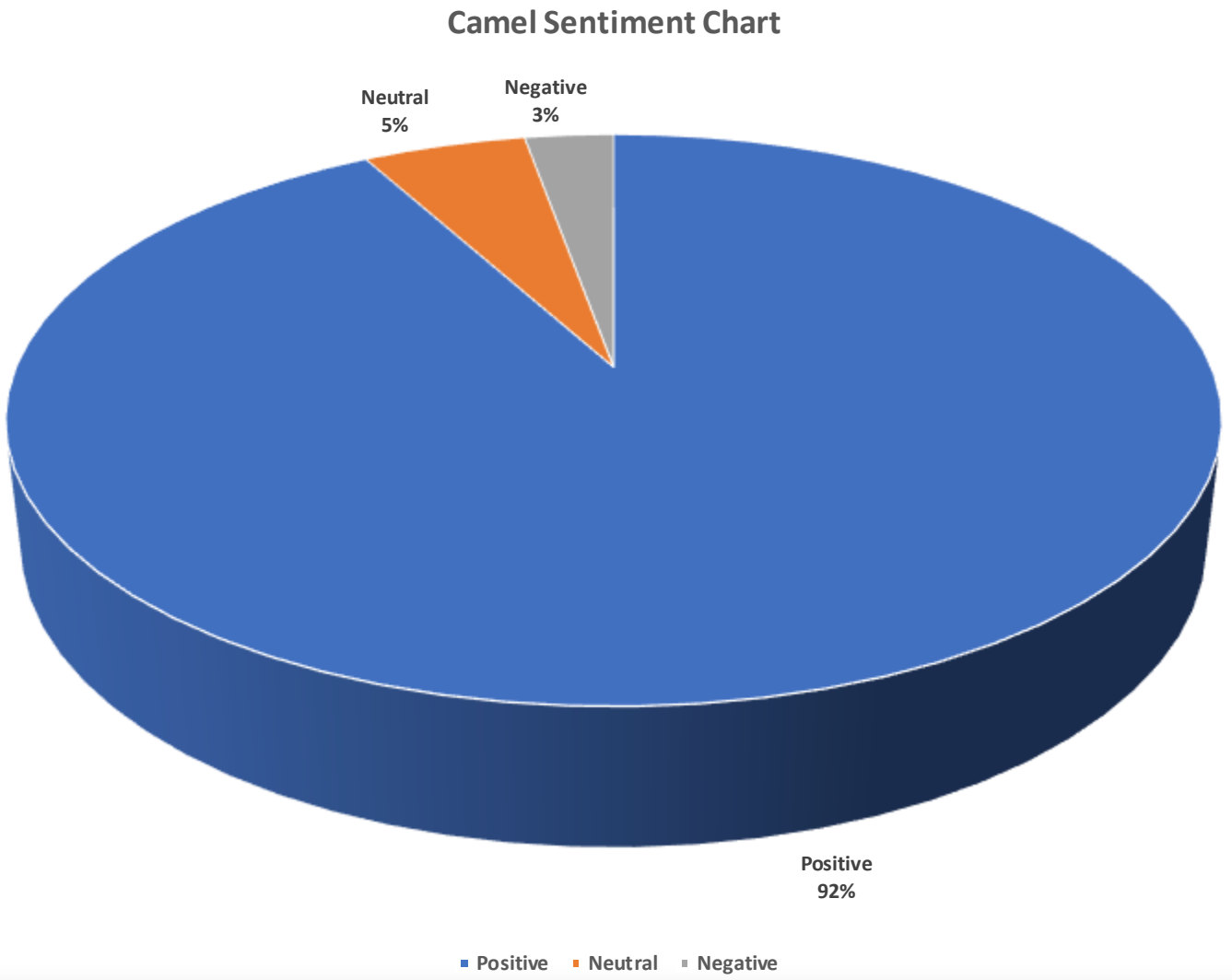
Key Themes - Measurement

- Engagement
- Brand
- Marketing Effectiveness

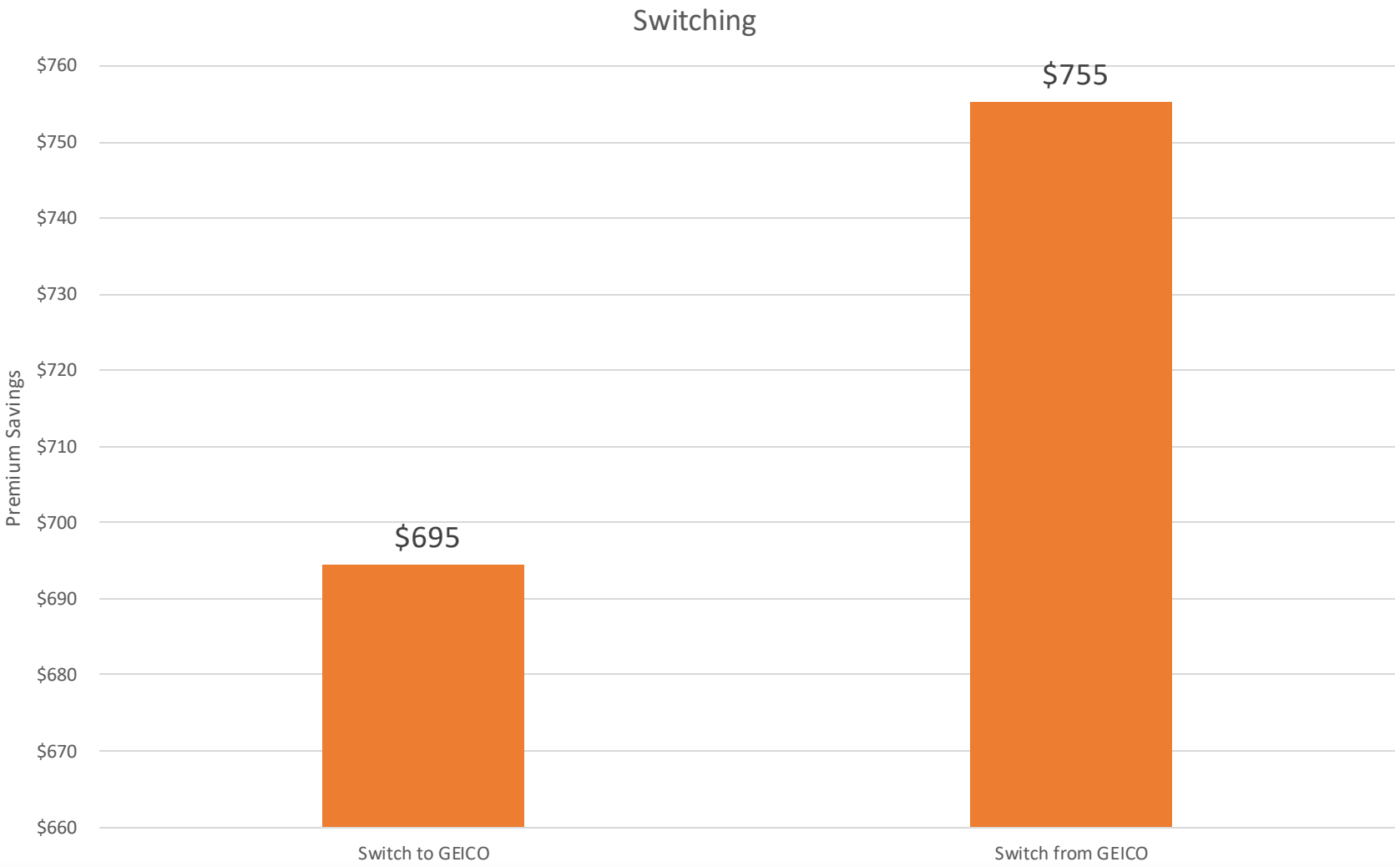
Advertising “Focus Groups”



Camel – Sentiment Chart



Savings


Switching
$760
$750
$740
$730
$720
$710
$700
$690
$680
$670
$660
Premium Savings
$695
$755
Switch to GEICO
Switch from GEICO

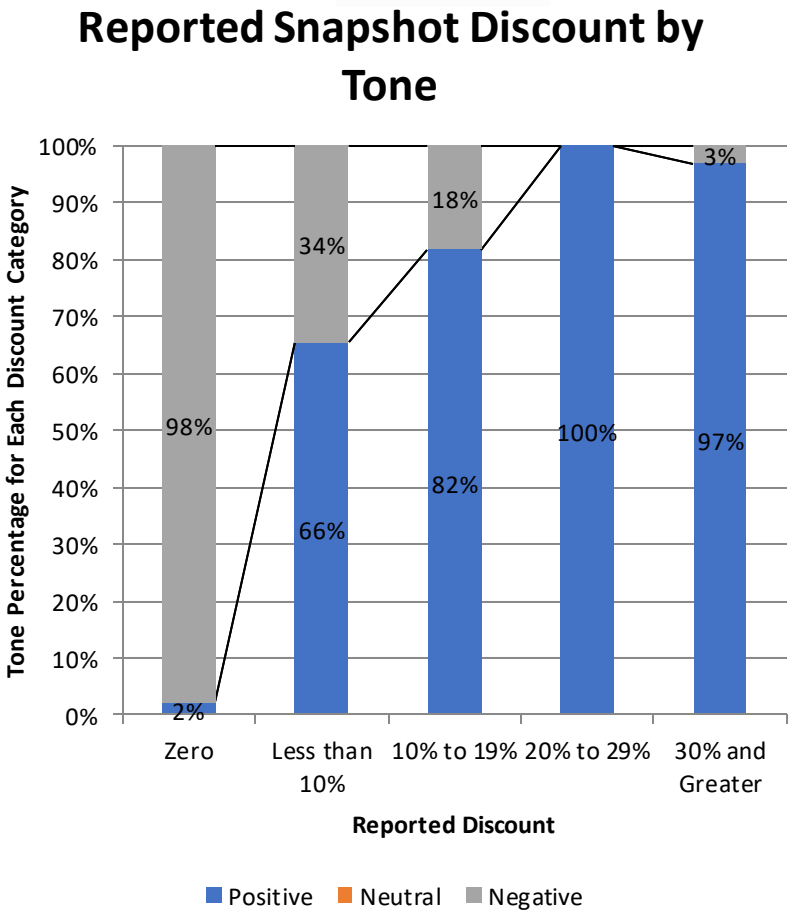
Analytics Model Training

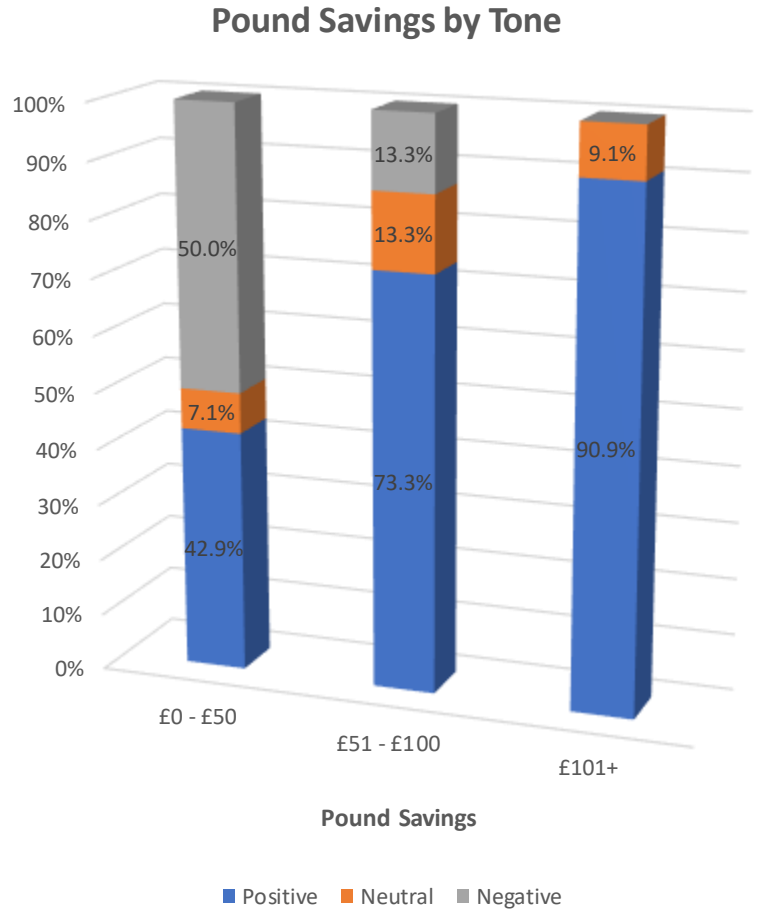
- Random sample of tweets are scored for a target variable
- Model can be developed for the random sample to predict the target
- Model scoring can be applied to new tweets to determine predicted classification and improve/enhance prediction algorithm

Measuring Customer Sentiment

Tweet

> Thought I earned a 30% discount through the @Progressive snapshot, only to learn that policy rates went up the amount of my discount #notcool





Thank You for Your Attention

Roosevelt C. Mosley, FCAS, MAAA, CSPA

309.807.2330

rmosley@pinnacleactuaries.com

PINNACLE
ACTUARIAL RESOURCES, INC.