Zero-Revelation Linguistic Regulation: Detecting Risk Through Corporate Emails and News

Sanjiv Das, Seoyoung Kim, Bhushan Kothari
October 2016

Main Ideas

- Financials are often delayed indicators of corporate quality.
- Internal discussion may be used as an early warning system for upcoming corporate malaise.
- Emails have the potential to predict such events.
- Software that analyzes emails and produces summary statistics may be helpful because
  - It can analyze vast quantities of textual data not amenable to human processing.
  - It does not require revelation of email content explicitly to regulators.
- Corporate senior management may also use these analyses to better predict and manage impending crisis for their firms.
- The approach requires *zero revelation* of emails.

Tools and Techniques

1. Extract-Transform-Load (ETL) for Emails, News.
2. De-duplication of content.
3. Frequency of content over time.
4. Sentiment analysis.
5. Correlating sentiment with financial performance.
6. Wordclouds.
7. Email networks.
8. “WordPlay” - the frequency of salient words over time.
9. Topic analysis.
10. Parts of speech (POS) analysis.

Enron Data

- Two years: 2000 and 2001 (Enron filed for bankruptcy in October 2001).
- Enron corpus created by the computer science department at CMU, see: https://www.cs.cmu.edu/~./enron/
- Dataset first made available by FERC, collected and distributed by Carnegie Mellon's CALO project.
- Data contains emails from 2000 to 2004, but the data from 2002 onwards is very thin.
- Some emails were culled from the data set for legal reasons, and information about exclusion criteria have not been made public.

Enron_Email_Analysis_S1.Rmd

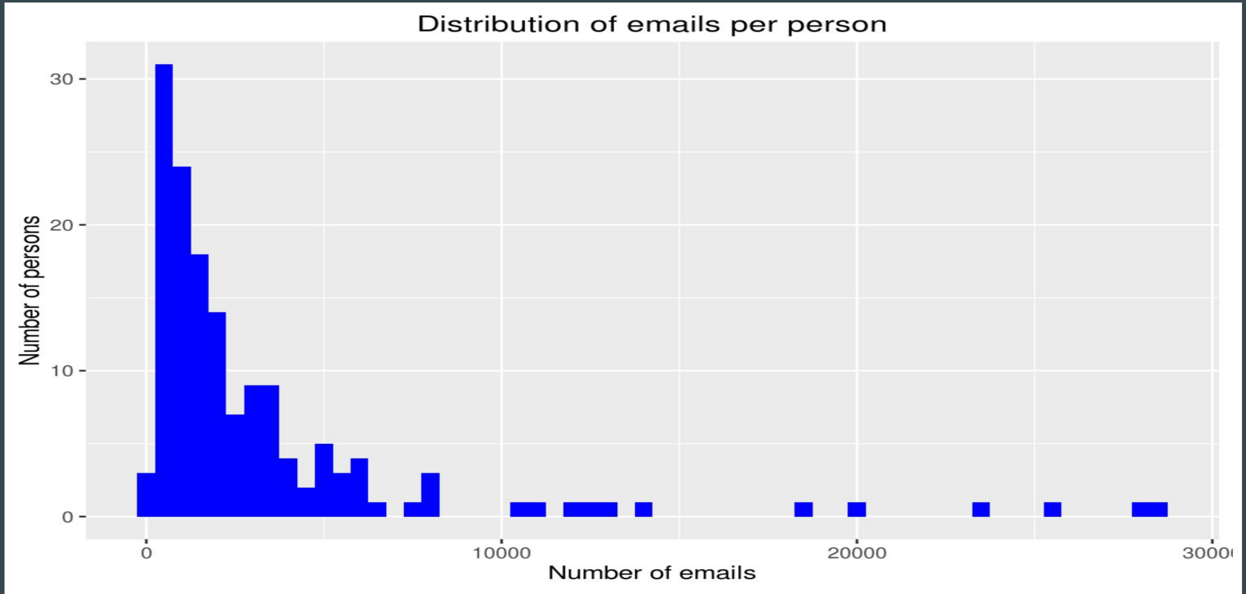
- Reads in the data and provides some basic descriptive statistics.
- Output is stored in **AllMails.RData** (1.3GB).
- Contains a single data frame **EnMails** with user name, mail folder, full message. (517,395 x 3).

Enron_Email_Analysis_S2.Rmd

- Reads in AllMails.RData and then extracts the text (message body) of all emails and stores them in a new data frame.
- Creates columns for the from, to, cc, bcc, etc.
- Creates a file **AllMails2.Rdata** (550MB). (517,394 x 18).
- Has the following columns in the **EnMails** data frame, which is now overwritten from the previous version: {"User", "MailBox", "RawMessage", "Date", "Year", "Month", "Week", "Quarter", “MessageBody", "CharCnt", "from", "toList", "toCount", "ccList", "ccCount", "bccList", "bccCount", "totalCount"}.
- The column RawMessage is converted to MessageBody which is clean text.
- At this stage we still have all emails. No de-duplication has been done so far.

Descriptive Statistics

- Number of managers in the email database = 150
- Total number of email folders = 3349
- Total number of emails = 517,394 (in 2000 & 2001 = 468,895)
- Remove blank emails = 450,345



Statistics (all emails)

- Total not blank = 450,345
- Characters per email = 939
- Mean “to” recipients = 6.38
- Mean “cc” recipients = 1.13
- Mean emails per employee = 3064
- Mean connectedness = 7.84

Enron_Email_Analysis_S3.Rmd

- Reads in AllMails2.RData.
- Does further clean up for users who have more than one email account. (Lots of exception handling here, hope this is not the case with better email data.)
- Creates summary statistics.
- Then only accesses the Sent mails to prevent duplication of emails across email boxes.
- Creates plots of emails over time, mainly for the years 2000 and 2001 (Enron failed in Oct 2001).
- Does mood scoring of the message body. By week and user. Sentiment and Disagreement.
- Calculates weekly returns, uses **ENE_daily.csv**.
- Regressions of sentiment and returns.
- Wordclouds (though not very informative).
- Network analysis, creates list of users for network analysis.
- Saves EnMails DF and Weekly Data for Regressions to **AllMails3.RData**. (113,263 x 19, 57MB).
- Creates **MoodScoredDf.RData** which contains all the mood scoring statistics.

De-duplication of messages

- In order to avoid duplicate messages, we only filtered out messages in the “sent” emails. This ensures that messages are counted only once.
- Count = 116,448
- Keeping only emails with less than 20 recipients, and character count less than 3000 (this filters out mass mailings), and then subset out the 2000, 2001 emails, we get 113,265 (49,633 in 2000; 63,630 in 2001).
- See stats next page.

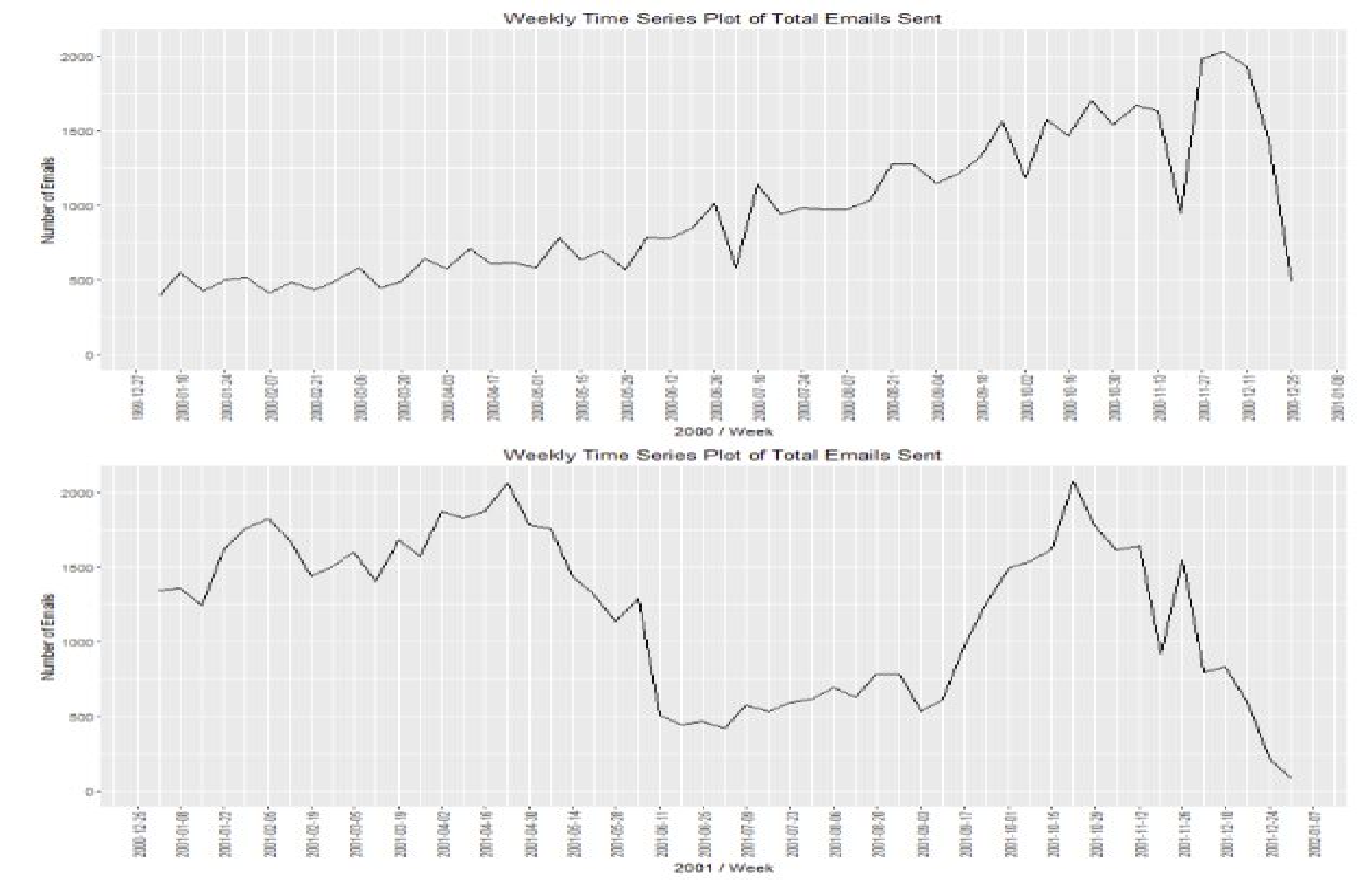
Only “Sent” messages

Table 1. Summary Statistics: Sent Emails

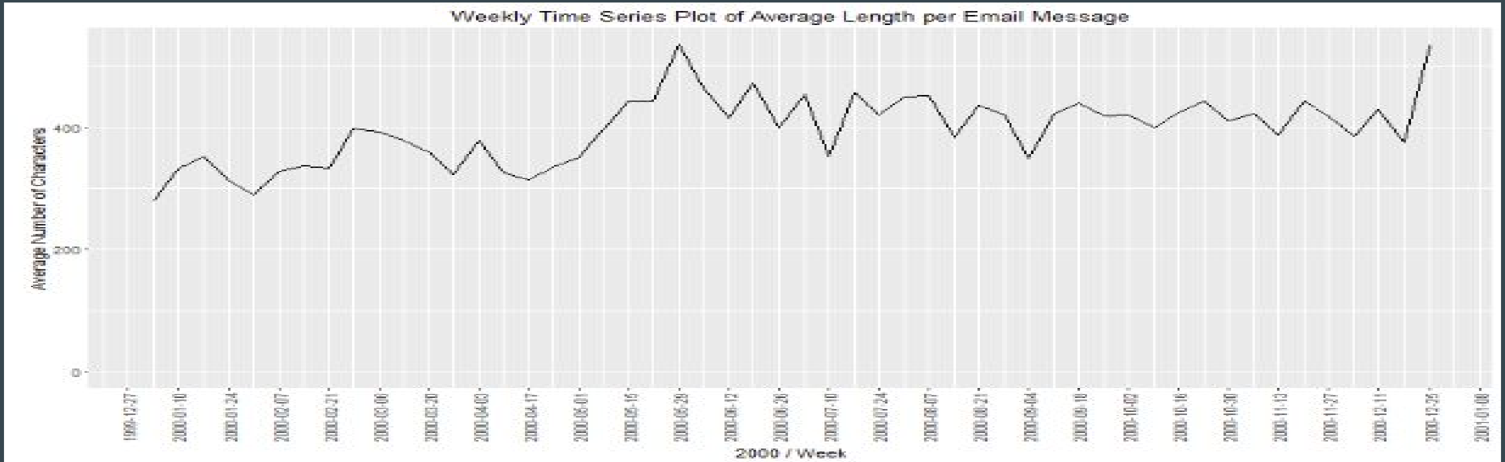
This table presents descriptive statistics of various email characteristics. Our sample encompasses the 113,266 sent emails of 144 employees of Enron Corporation over the period spanning January 2000 through December 2001.

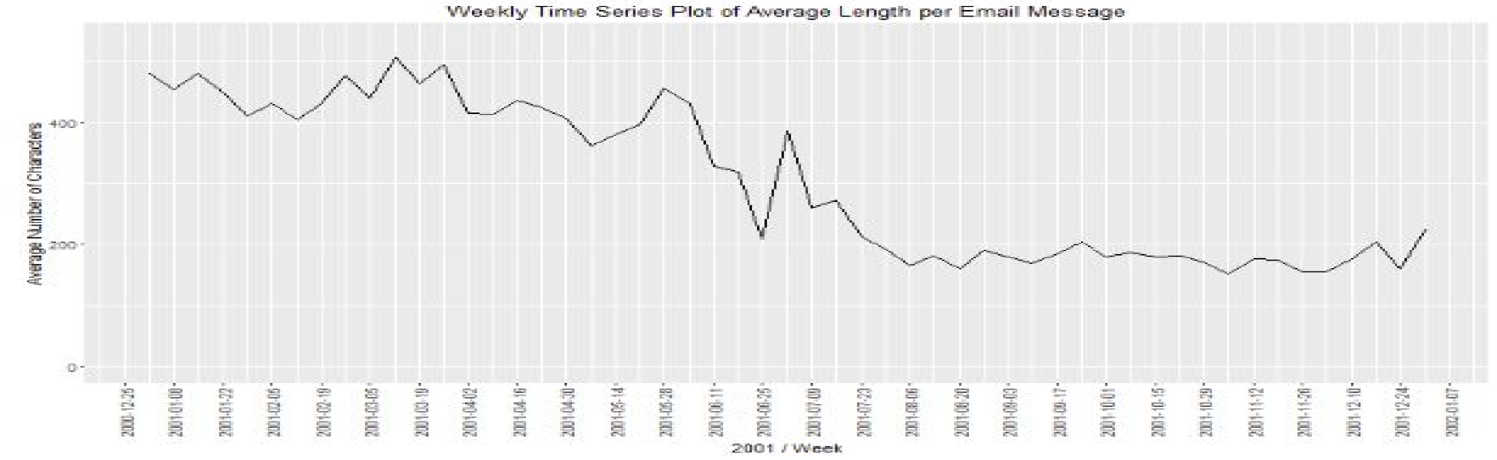
Variable	Mean	Min.	P25	Median	P75	Max.
Panel A. Characteristics by Employee (N=144)						
Number of Emails per Person	787	2	105	349	891	8,793
Average "Connectedness"	1.62	1	1.21	1.44	1.76	4.47
Average Length per Person	279.92	19.15	160.45	227.90	338.07	944.23
Panel A. Email Characteristics (N=113,266)						
Length of Email (# of characters)	362	0	46	163	466	2,998
Direct Recipients per Email ("to")	1.44	0	1	1	1	20
Indirect Recipients per Email ("cc")	0.32	0	0	0	0	19
Total Recipients per Email	1.77	1	1	1	2	20

Frequency of emails

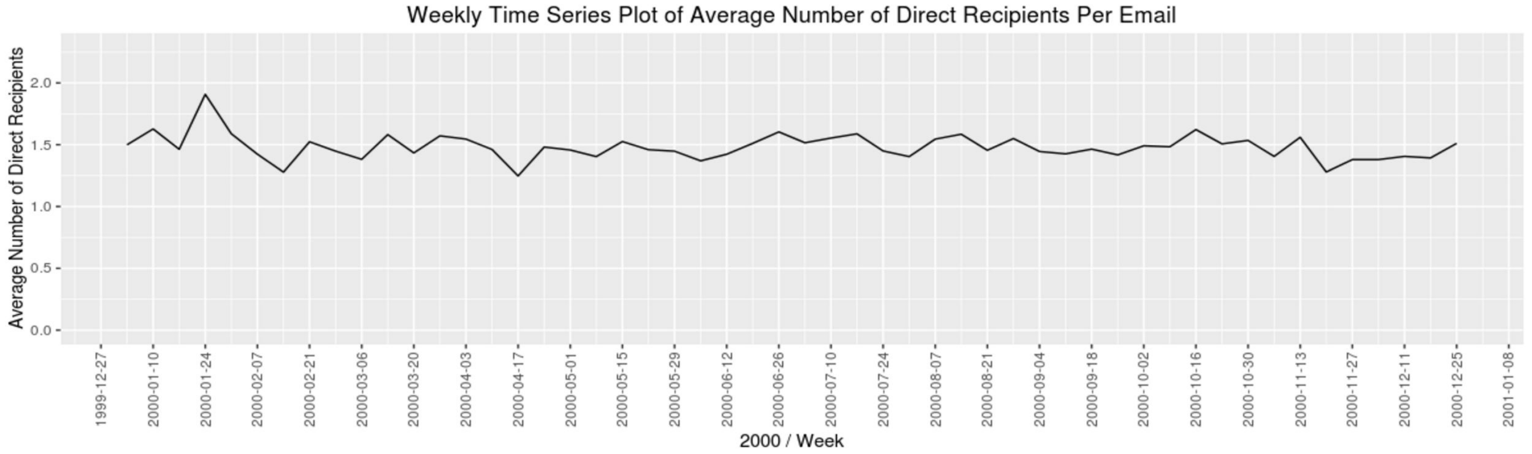


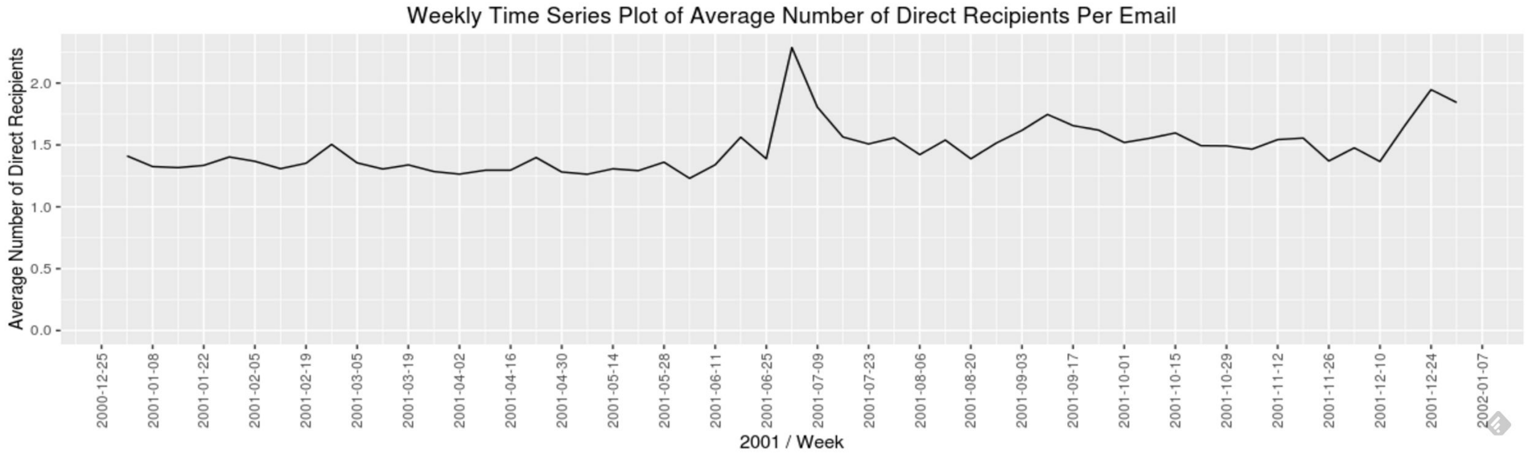
Email Length





Recipients per email





Wordclouds (2000)

1

2

3

4

Wordclouds (2001)

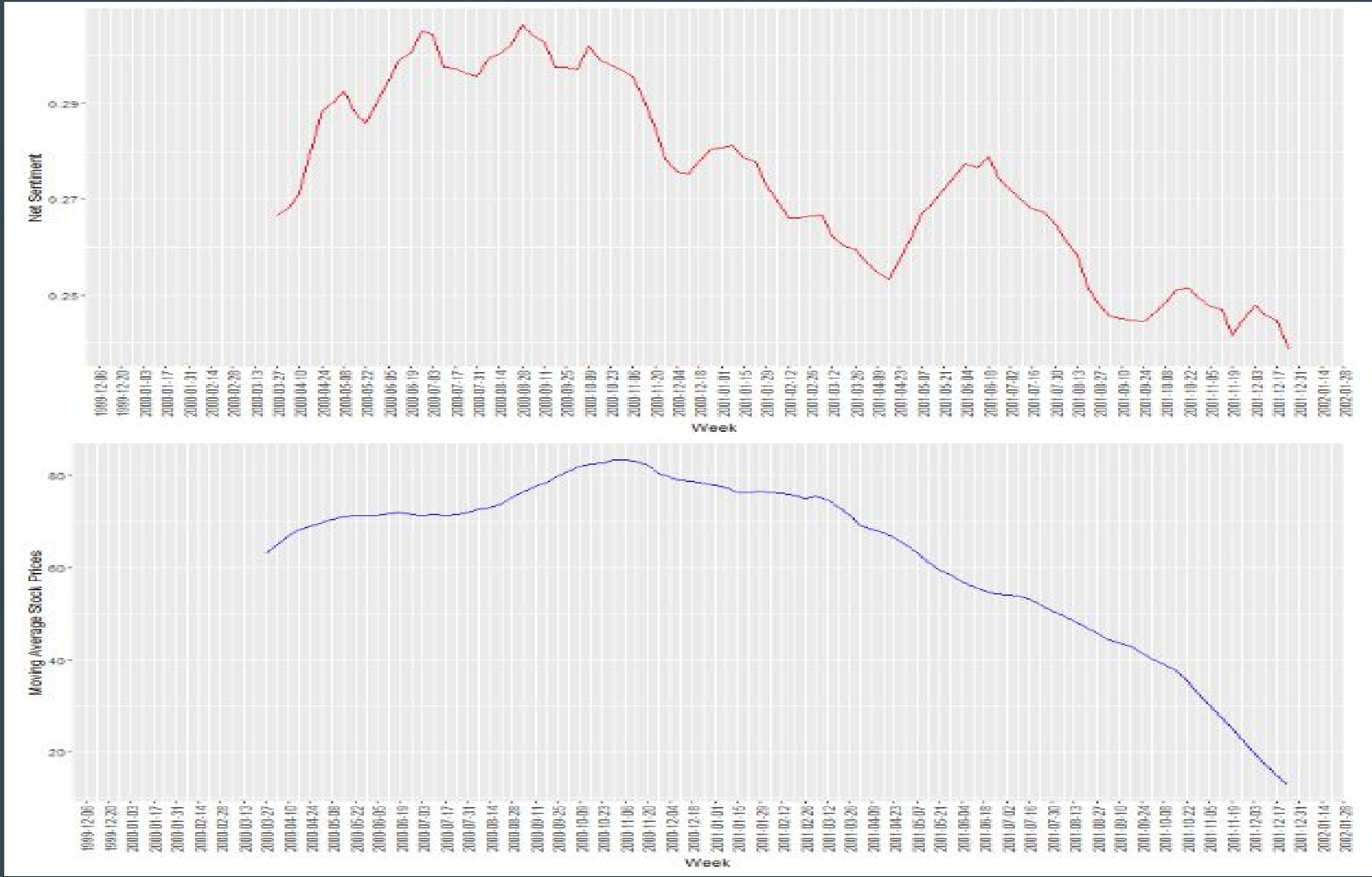
1

2

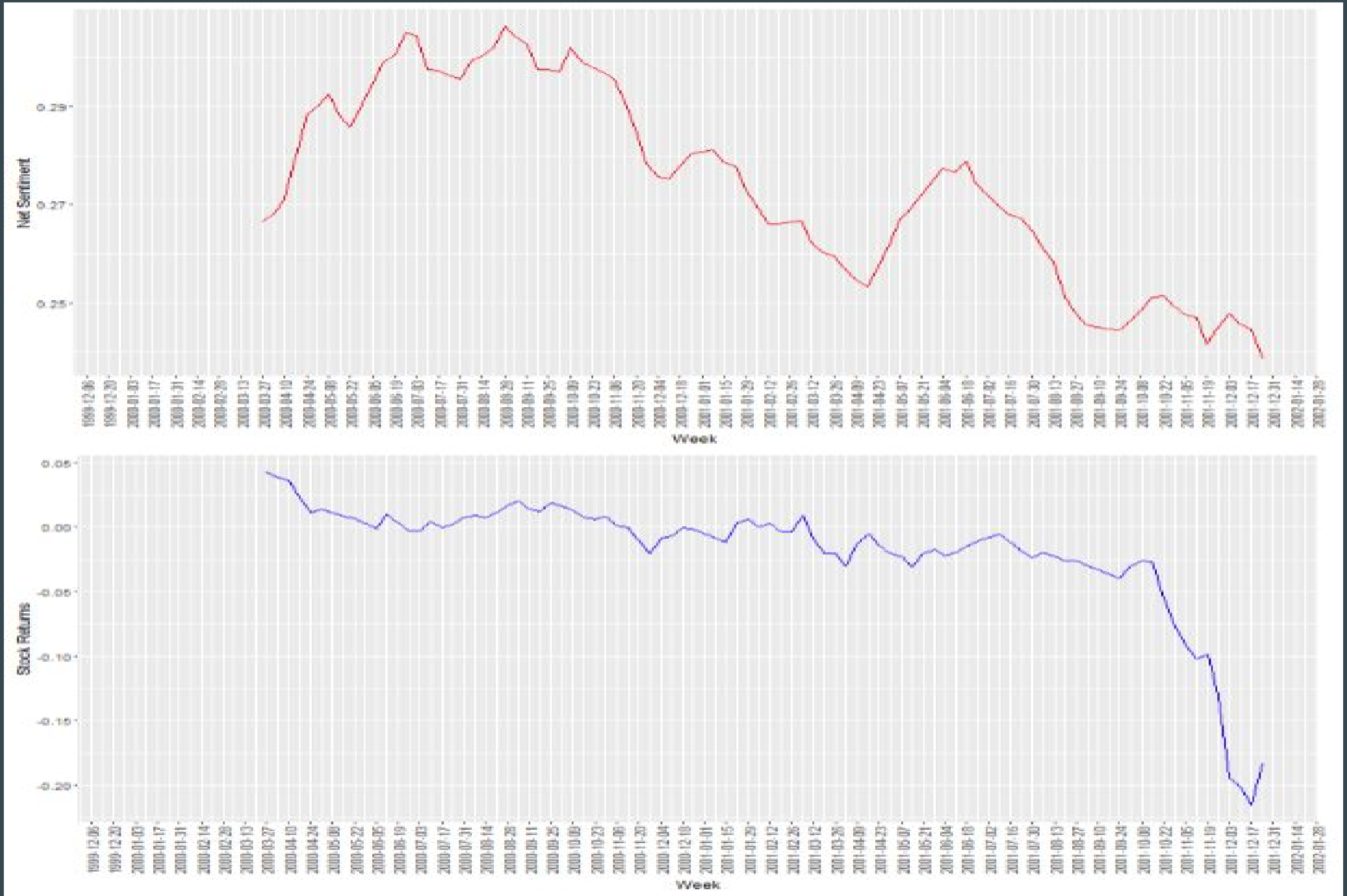
3

4

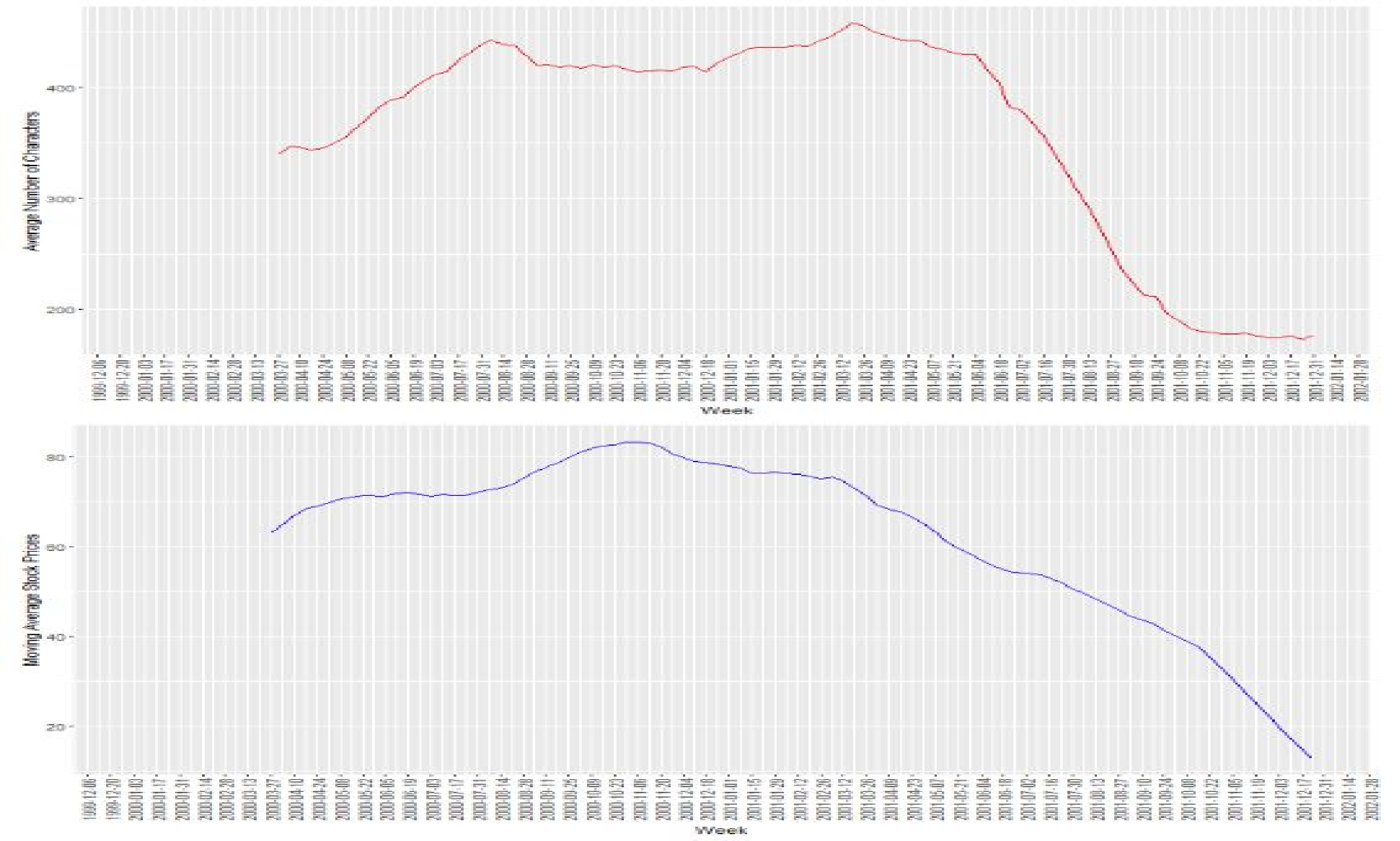
Sentiment and Stock Prices



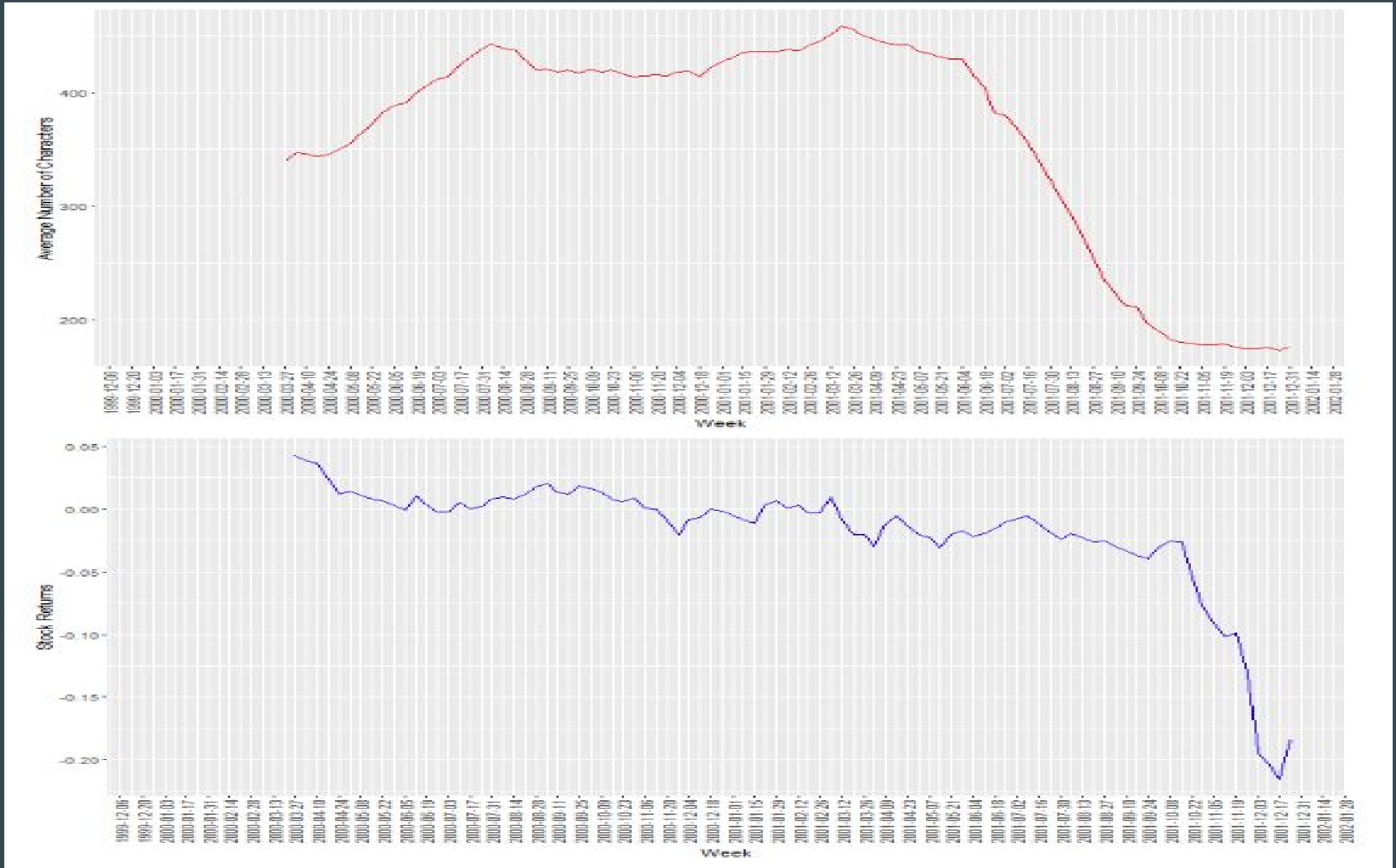
Sentiment and Stock Returns



Stock Prices and Email Length



Stock Returns and Email Length



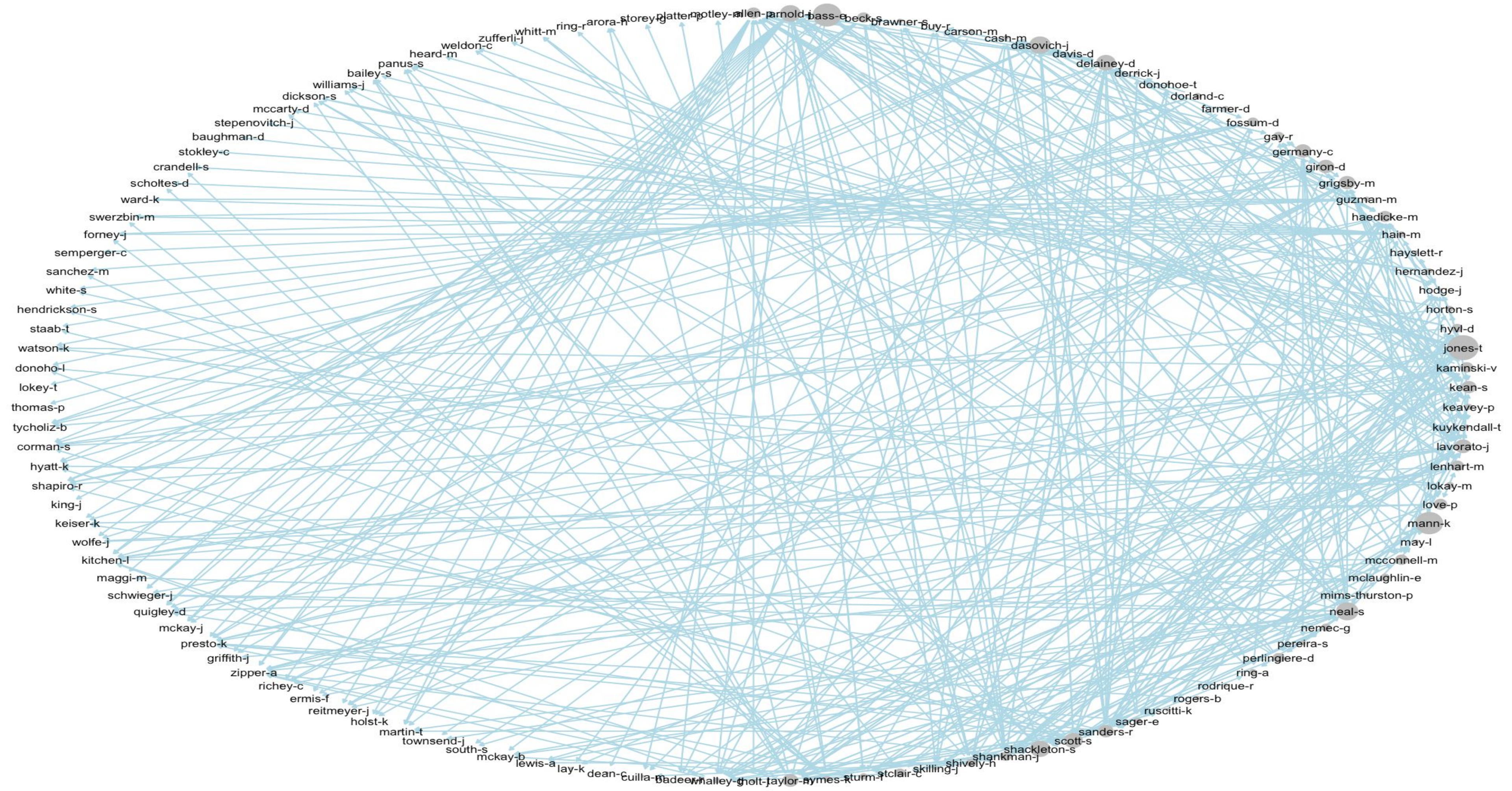
Returns and Email characteristics

Variable	Coefficient Estimate (t-statistic)			
	(1)	(2)	(3)	(4)
MA Net Sentiment $_t$	XXX*** (XXX)	0.575 (0.63)	2.330*** (3.14)	-1.397 (-1.25)
MA Email Length $_t$		0.584*** (2.97)		1.046*** (4.19)
MA Total Emails $_t$			-0.004 (-0.10)	-0.131*** (-2.83)
Intercept		-0.406* (-1.93)	-0.671*** (-3.08)	0.117 (0.43)
Adjusted R-squared	XXX		0.09	0.24
Number of observations	88	88	88	88

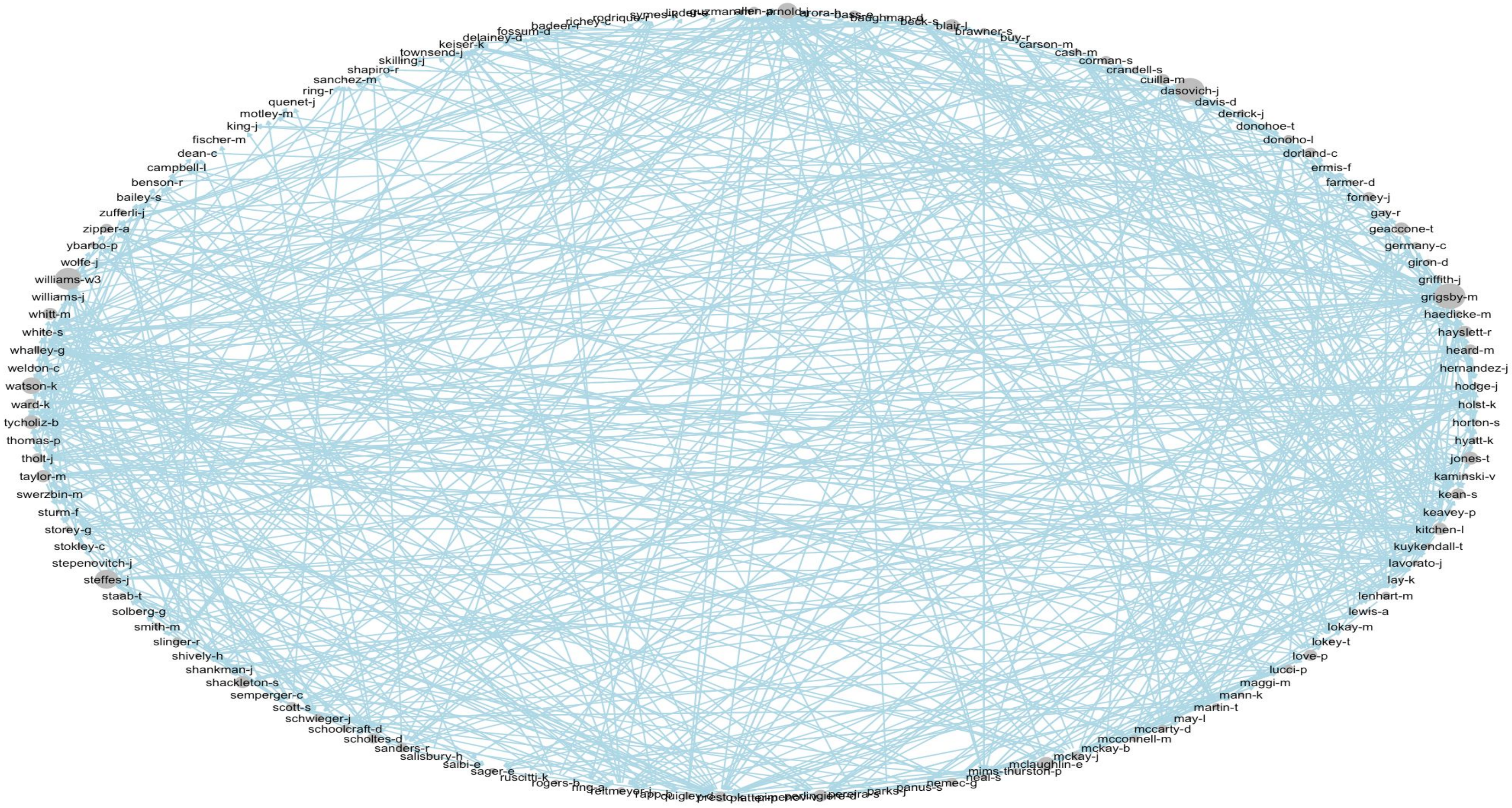
Enron_Email_Analysis_S3_2.Rmd

- Does network analysis.
- Reads in AllMails3.RData.
- Creates adjacency matrix and plots by week.
- Creates weekly stats.
- And plots degree distribution.

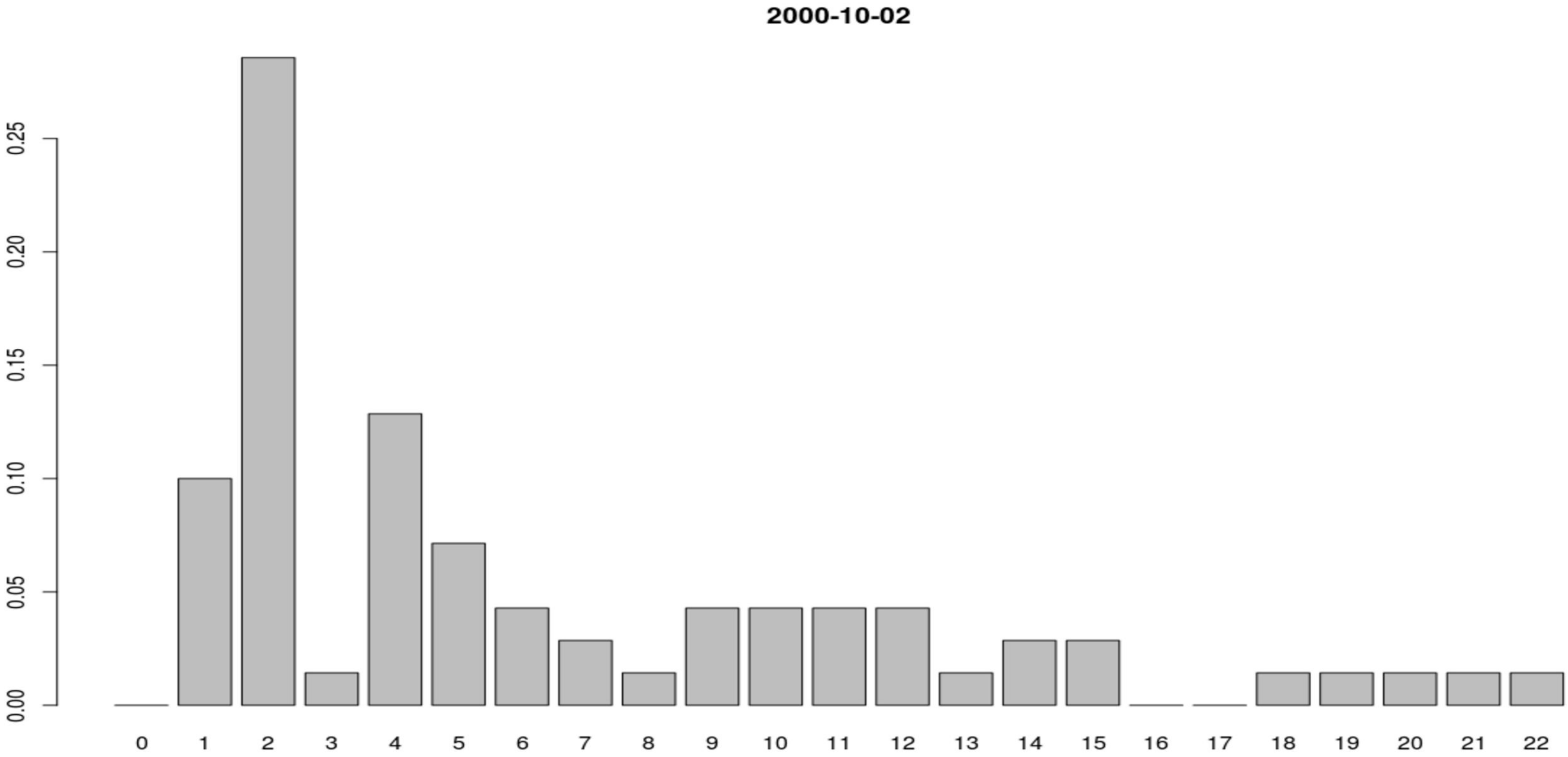
Quarter_beginning: 2000-10-01



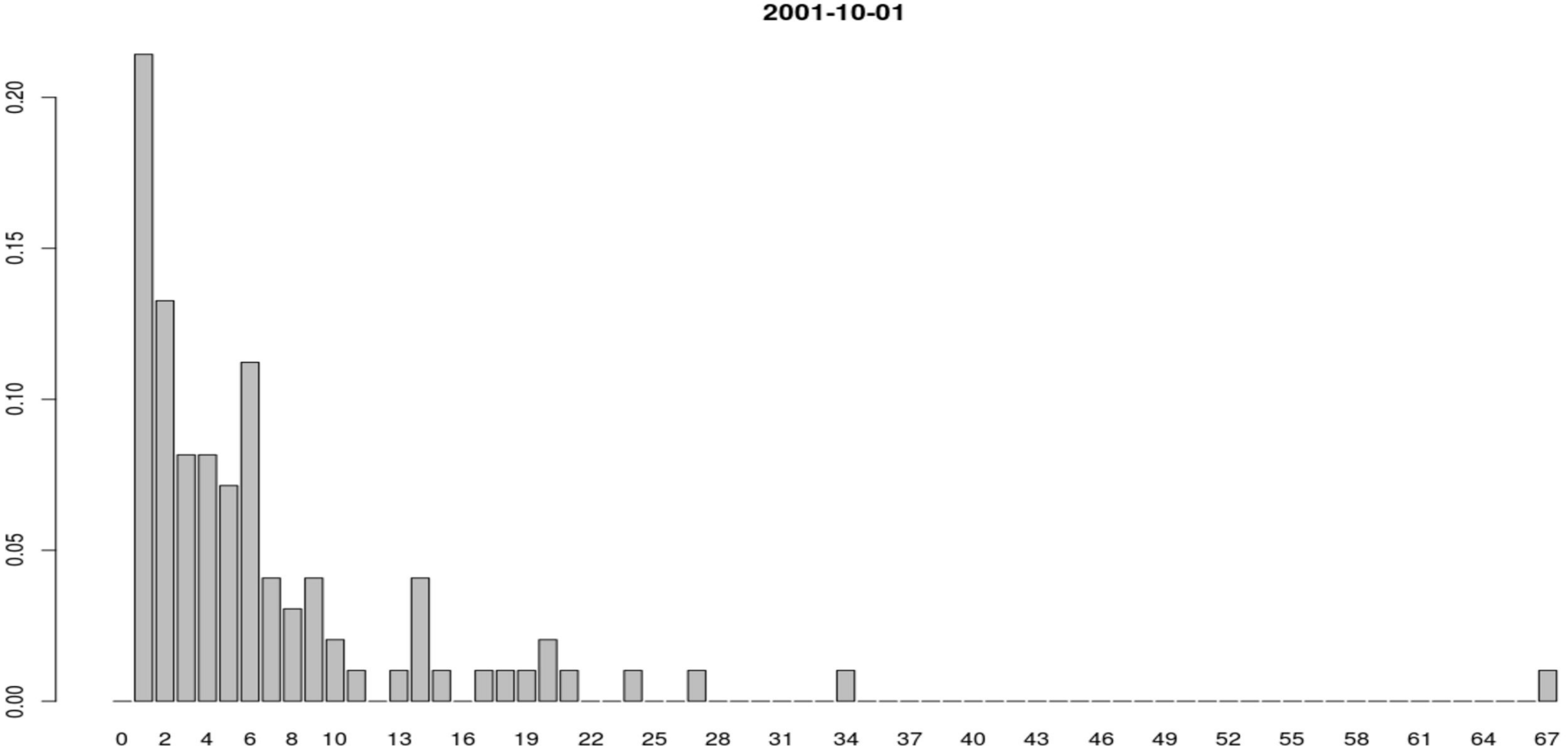
Quarter_beginning: 2001-10-01



Degree Distribution



Degree Distribution



Enron_Email_Analysis_S4.Rmd

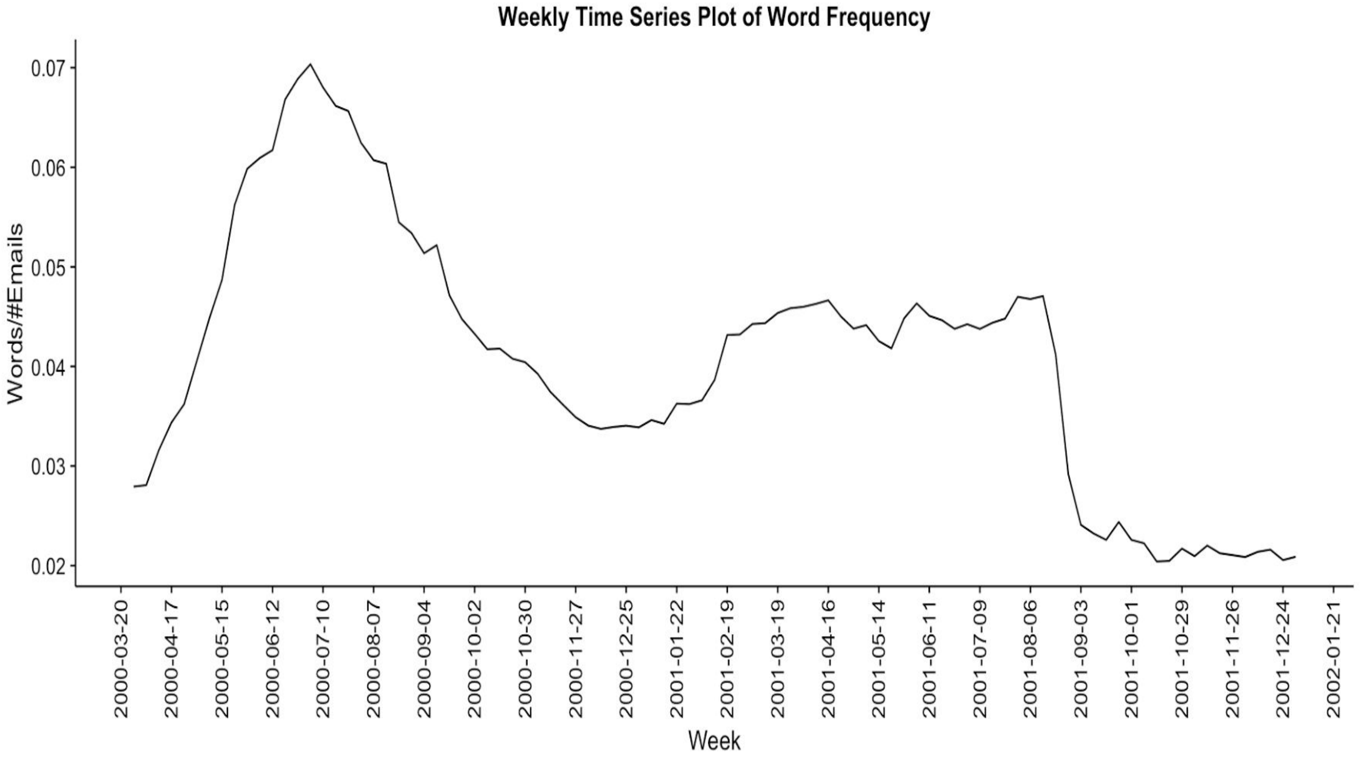
- Reads in AllMails3.RData.
- Creates list of 1000 most important words over time, for further use in Shiny app.
- Shiny app Intersects with the word lists so we only get dictionary words and not nonsense compound words.
- Creates term-document matrix for Shiny app: save(tdm2,freq1000,poswords,negwords,uncwords,weekNames,weeklyMailCount, file="**AllMails4_tdm.RData**"). dim(tdm2) = 76518   105, words x weeks. 1353 negwords, 288 poswords, 206 uncwords, 105 weeks.
- ui.R and server.R : Shiny app to display words over time. And some plots with correspondence of words with sentiment and returns. Uses AllMails4_tdm.Rdata and MoodScoredDf.Rdata. Also WeeklyRet.csv.

Word Play over Time

Select Word

credit

Word over time | Word_vs_Sentiment | Word_vs_Return | Word_Sentiment_Correlation | Word_Return_Correlation

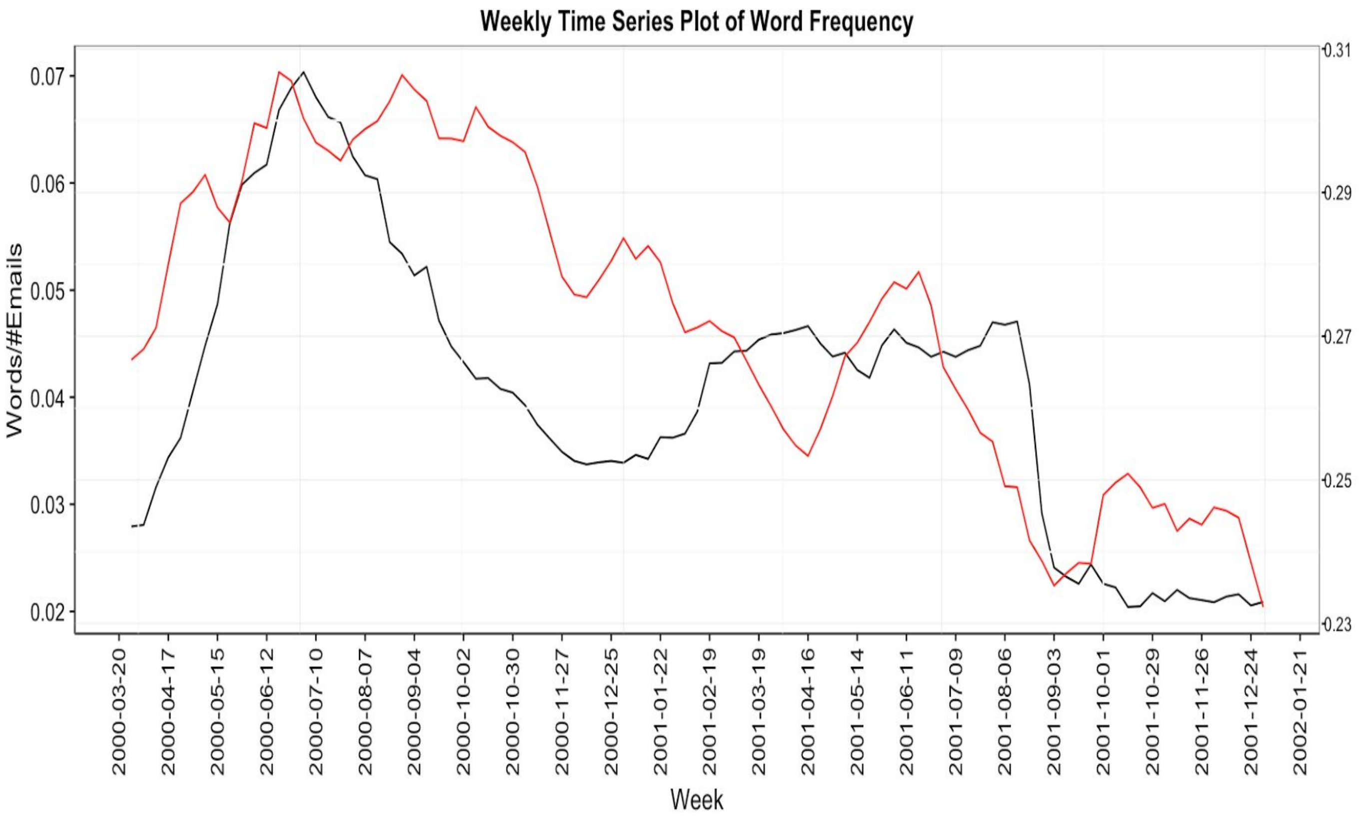


13-week moving average of the selected word

Select Word

credit

Word over time | Word_vs_Sentiment | Word_vs_Return | Word_Sentiment_Correlation | Word_Return_Correlation

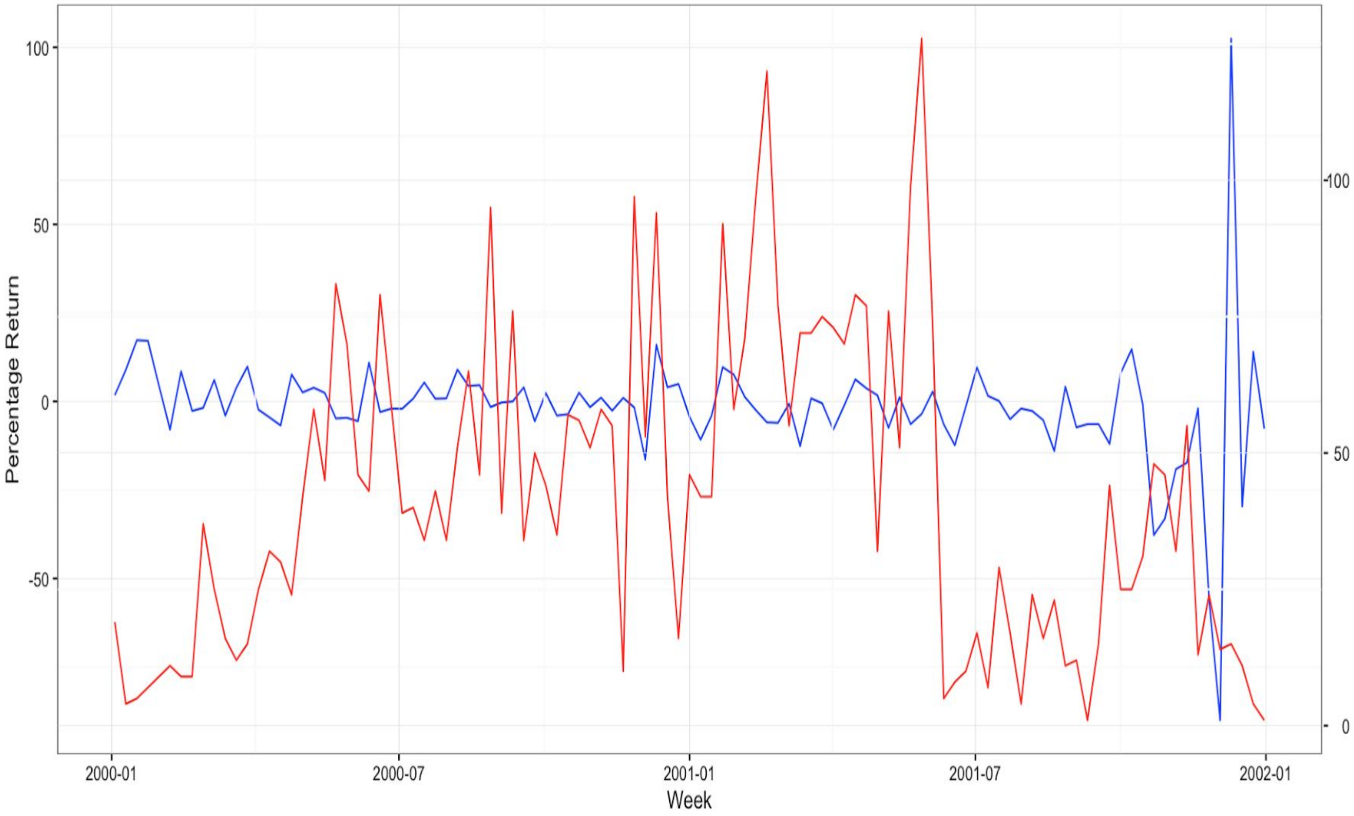


Black: words/#emails , Red: sentiment

Select Word

credit

Word over time | Word_vs_Sentiment | Word_vs_Return | Word_Sentiment_Correlation | Word_Return_Correlation



Blue: Stock Returns (left axis) / Red: Quarterly Mean of Words per Email (right axis)

Select Word

credit

Word over time | Word_vs_Sentiment | Word_vs_Return | Word_Sentiment_Correlation | Word_Return_Correlation

Show 25 entries

Search: cre

wNames	corrDf	absVal
create	0.30519694	0.30519694
credit	0.24560269	0.24560269
incredible	0.21271165	0.21271165
credentials	0.18220654	0.18220654
decrease	-0.16795778	0.16795778
creep	-0.16696068	0.16696068
credibility	0.16631007	0.16631007
discretion	0.15226727	0.15226727
creek	0.14474197	0.14474197
crept	0.11695333	0.11695333
cream	0.11685996	0.11685996
scream	0.11366155	0.11366155
increase	0.10754947	0.10754947
crest	0.09981181	0.09981181
secret	0.09777703	0.09777703
crete	0.08876355	0.08876355
credo	-0.08531154	0.08531154

Word over time | Word_vs_Sentiment | Word_vs_Return | Word_Sentiment_Correlation | Word_Return_Correlation

Select Word

credit

Show 25 entries

Search: cre

wNames	wrCorrDf	absVal
decree	-0.17363923	0.17363923
crew	-0.17037077	0.17037077
discrepancy	0.11519270	0.11519270
concrete	-0.10779474	0.10779474
creativity	0.10439160	0.10439160
creative	0.10320859	0.10320859
incredible	0.10030781	0.10030781
creation	-0.07852615	0.07852615
secretary	0.07210966	0.07210966
cream	-0.05807845	0.05807845
massacre	0.05632156	0.05632156
creep	-0.04709850	0.04709850
recreation	0.04302101	0.04302101
screw	0.04131260	0.04131260
creator	0.04099252	0.04099252
scream	0.03690762	0.03690762
create	0.03670083	0.03670083

Enron_Email_Analysis_S5.Rmd

- Topic analysis.
- Uses AllMails3.Rdata and AllMails4_tdm.Rdata.
- Plots topic share, and also sentiment scores the top 20 words in each topic.

Latent Semantic Analysis

Latent Semantic Analysis (LSA) is an approach for reducing the dimension of the Term-Document Matrix (TDM), or the corresponding Document-Term Matrix (DTM), in general used interchangeably, unless a specific one is invoked. Dimension reduction of the TDM offers two benefits:

- The DTM is usually a sparse matrix, and sparseness means that our algorithms have to work harder on missing data, which is clearly wasteful. Some of this sparseness is attenuated by applying LSA to the TDM.
- The problem of synonymy also exists in the TDM, which usually contains thousands of terms (words). Synonymy arises becauses many words have similar meanings, i.e., redundancy exists in the list of terms. LSA mitigates this redundancy, as we shall see through the ensuing analysis of LSA.
- While not precisely the same thing, think of LSA in the text domain as analogous to PCA in the data domain.

Implementing LSA through SVD

LSA is the application of Singular Value Decomposition (SVD) to the TDM, extracted from a text corpus. Define the TDM to be a matrix $M \in \mathcal{R}^{m \times n}$, where m is the number of terms and n is the number of documents.

The SVD of matrix M is given by

$$M = T \cdot S \cdot D^\top$$

where $T \in \mathcal{R}^{m \times n}$ and $D \in \mathcal{R}^{n \times n}$ are orthonormal to each other, and $S \in \mathcal{R}^{n \times n}$ is the "singluar values" matrix, i.e., a diagonal matrix with singular values on the diagonal. These values denote the relative importance of the terms in the TDM.

SVD tries to connect the correlation matrix of terms ($M \cdot M^\top$) with the correlation matrix of documents ($M^\top \cdot M$) through the singular matrix.

To see this connection, note that matrix T contains the eigenvectors of the correlation matrix of terms. Likewise, the matrix D contains the eigenvectors of the correlation matrix of documents. To see this, let's compute

Dimension reduction of the TDM via LSA

If we wish to reduce the dimension of the latent semantic space to $k < n$ then we use only the first k eigenvectors. The **lsa** function does this automatically.

We call LSA and ask it to automatically reduce the dimension of the TDM using a built-in function **dimcalc_share**.

And LDA, what does it have to do with LSA?

It is similar to LSA, in that it seeks to find the most related words and cluster them into topics. It uses a Bayesian approach to do this, but more on that later. Here, let's just do an example to see how we might use the **topicmodels** package.

Topic Analysis: Shallow Dive into LDA

Latent Dirichlet Allocation (LDA) was created by David Blei, Andrew Ng, and Michael Jordan in 2003, see their paper titled "Latent Dirichlet Allocation" in the *Journal of Machine Learning Research*, pp 993–1022.

The simplest way to think about LDA is as a probability model that connects documents with words and topics. The components are:

- A Vocabulary of V words, i.e., $w_1, w_2, \ldots, w_i, \ldots, w_V$, each word indexed by i.
- A Document is a vector of N words, i.e., $\mathbf{w}$.
- A Corpus D is a collection of M documents, each document indexed by j, i.e. d_j.

Next, we connect the above objects to K topics, indexed by l, i.e., t_l. We will see that LDA is encapsulated in two matrices: Matrix A and Matrix B.

Matrices A and B

Matrix A: Connecting Documents with Topics

- This matrix has documents on the rows, so there are M rows.
- The topics are on the columns, so there are K columns.
- Therefore $A \in \mathcal{R}^{M \times K}$.
- The row sums equal 1, i.e., for each document, we have a probability that it pertains to a given topic, i.e., $A_{jl} = Pr[t_l|d_j]$, and $\sum_{l=1}^{K} A_{jl} = 1$.

Matrix B: Connecting Words with Topics

- This matrix has topics on the rows, so there are K rows.
- The words are on the columns, so there are V columns.
- Therefore $B \in \mathcal{R}^{K \times V}$.
- The row sums equal 1, i.e., for each topic, we have a probability that it pertains to a given word, i.e., $B_{li} = Pr[w_i|t_l]$, and $\sum_{i=1}^{V} B_{li} = 1$.

Distribution of Topics in a Document

- Using Matrix A, we can sample a K-vector of probabilities of topics for a single document. Denote the probability of this vector as $p(\theta|\alpha)$, where $\theta, \alpha \in \mathcal{R}^K$, $\theta, \alpha \geq 0$, and $\sum_l \theta_l = 1$.

- The probability $p(\theta|\alpha)$ is governed by a Dirichlet distribution, with density function

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{l=1}^{K} \alpha_l)}{\prod_{l=1}^{K} \Gamma(\alpha_l)} \prod_{l=1}^{K} \theta_l^{\alpha_l - 1}$$

where $\Gamma(\cdot)$ is the Gamma function. - LDA thus gets its name from the use of the Dirichlet distribution, embodied in Matrix A. Since the topics are latent, it explains the rest of the nomenclature. - Given θ, we sample topics from matrix A with probability $p(t|\theta)$.

Distribution of Words and Topics for a Document

- The number of words in a document is assumed to be distributed Poisson with parameter ξ.
- Matrix B gives the probability of a word appearing in a topic, $p(w|t)$.
- The topics mixture is given by θ.
- The joint distribution over K topics and K words for a topic mixture is given by

$$p(\theta, \mathbf{t}, \mathbf{w}) = p(\theta|\alpha) \prod_{l=1}^{K} p(t_l|\theta) p(w_l|t_l)$$

- The marginal distribution for a document's words comes from integrating out the topic mixture θ, and summing out the topics $\mathbf{t}$, i.e.,

$$p(\mathbf{w}) = \int p(\theta|\alpha) \left(\prod_{l=1}^{K} \sum_{t_l} p(t_l|\theta) p(w_l|t_l) \right) d\theta$$

Likelihood of the Entire Corpus

- This is given by:

$$p(D) = \prod_{j=1}^{M} \int p(\theta_j|\alpha) \left(\prod_{l=1}^{K} \sum_{t_{jl}} p(t_l|\theta_j) p(w_l|t_l) \right) d\theta_j$$

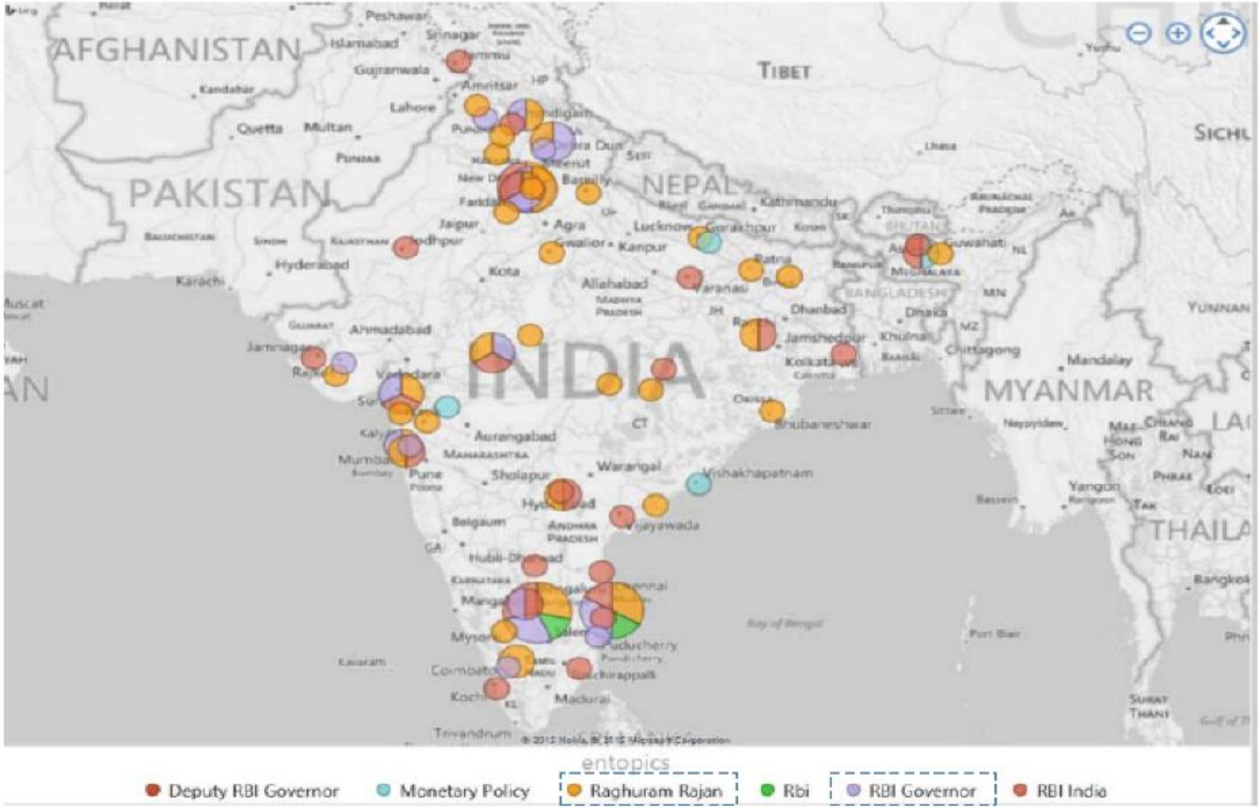
- The goal is to maximize this likelihood by picking the vector α and the probabilities in the matrix B. (Note that were a Dirichlet distribution not used, then we could directly pick values in Matrices A and B.)

- The computation is undertaken using MCMC with Gibbs sampling as shown in the example earlier.

Real-World Example 1

Conversations across India and around RBI

topycs



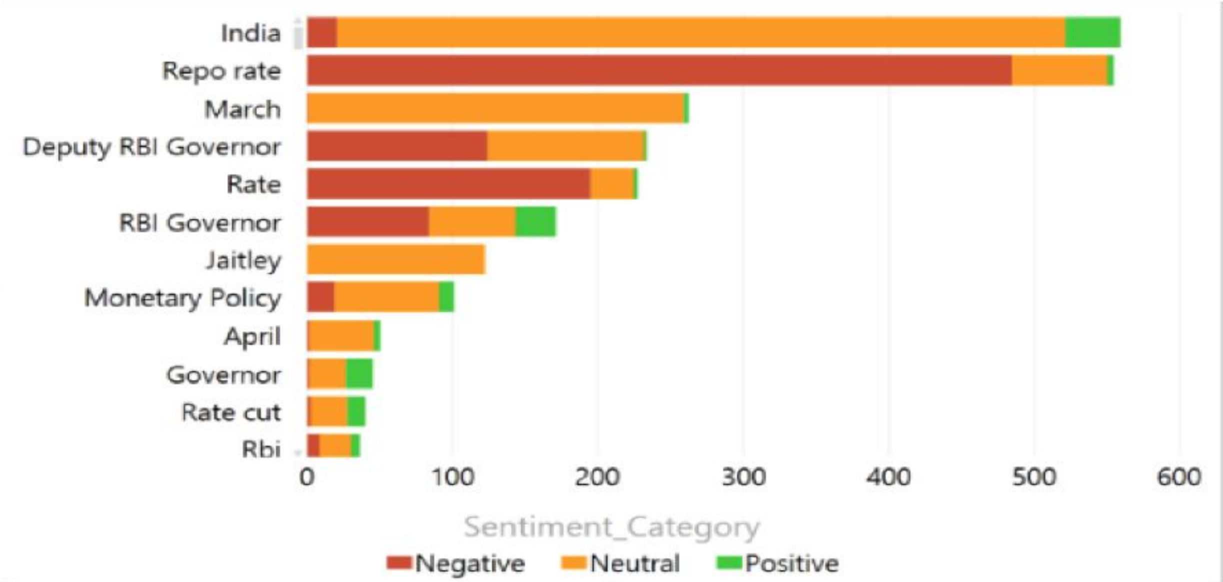
- Conversations across India on RBI, its people and the monetary policy
- Governor features in many conversations across both rural and urban areas
- Some conversations specifically around monetary policy

- Bubbles show split of conversations around Deputy RBI Governor, Monetary Policy, Raghuram Rajan, RBI and RBI Governor.
- Based on count of unique conversations
- Date Range: 1st – 14th April, 2015

Real-World Example 2

Top Topics along with RBI

topycs



text

"@NDTVProfit: RBI unlikely to change repo rate at policy review smlion

"Digging India's RBI Out of Morass of Debt" by on

"Financial stability is like Pornography. You can't define it but when you see it you know it" - D Subbarao (RBI Governor)

"I was disappointed by the fiscal relaxation." Ex-RBI Governor on India's budget and growth:

"Rajan is perfect, he explains complex economic," PM Modi on RBI governor.

- Repo rate evokes negative sentiment as people don't expect it to be changed
- Repo rate, rate cut and monetary policy are discussed frequently with RBI

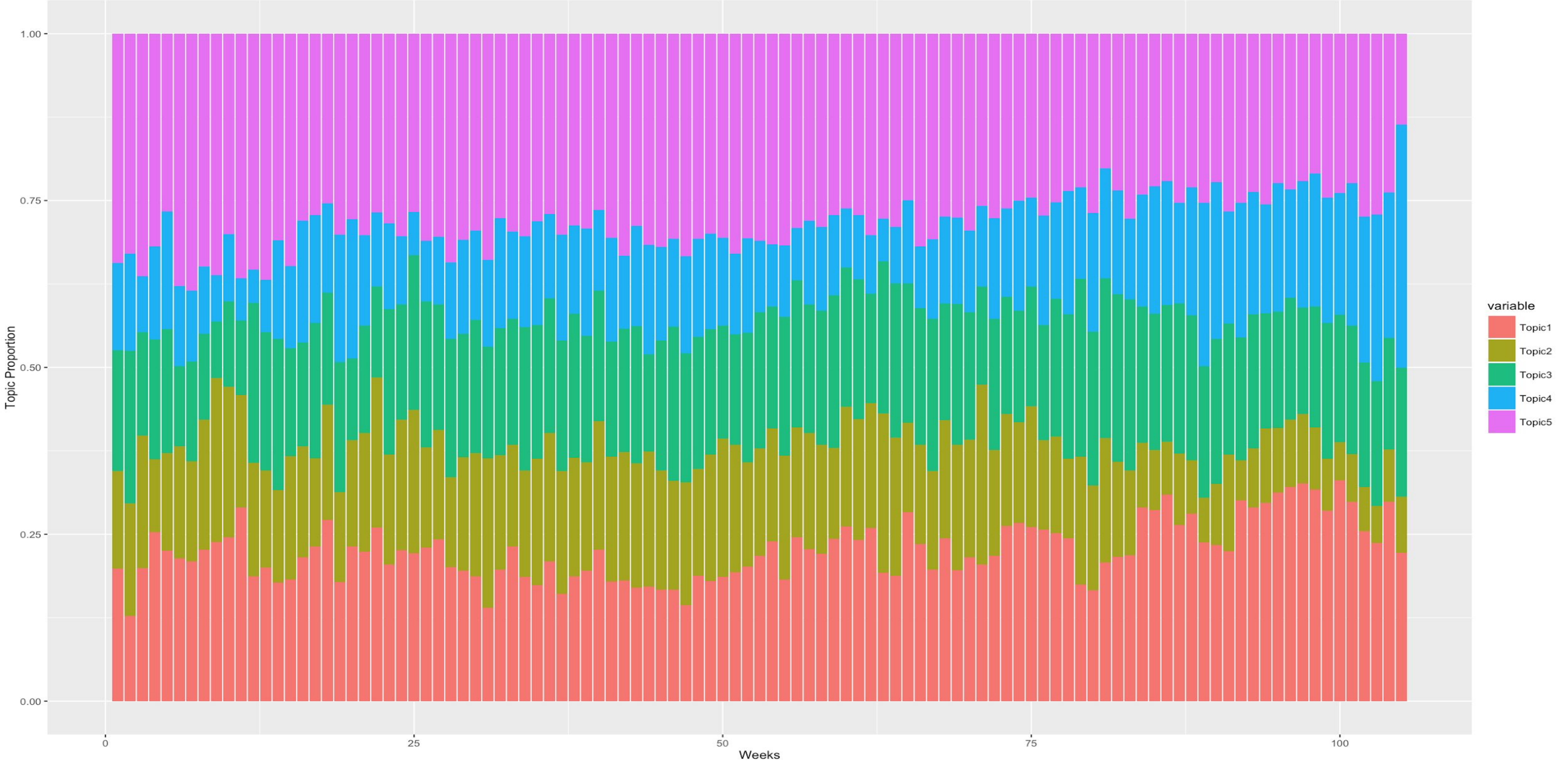
- Vertical Axis – Topics of Discussion
- Horizontal Axis – Count of Unique Conversations
- Date – 25th March - 14th of April
- Colors represent sentiment for conversation, Negative – Red, Neutral – Orange, Positive - Green

Topic Analysis Using LDA

```
> print(ldaOut.terms)
      Topic 1      Topic 2       Topic 3       Topic 4       Topic 5
 [1,] "please"     "subject"     "subject"     "please"      "subject"
 [2,] "the"        "please"      "deal"        "time"        "deal"
 [3,] "time"       "the"         "time"        "the"         "the"
 [4,] "corp"       "company"     "call"        "legal"       "please"
 [5,] "make"       "call"        "good"        "power"       "this"
 [6,] "call"       "time"        "america"     "today"       "from"
 [7,] "agreement"  "power"       "phone"       "letter"      "ill"
 [8,] "phone"      "week"        "the"         "good"        "week"
 [9,] "forward"    "mark"        "texas"       "call"        "call"
[10,] "day"        "information" "energy"      "make"        "gas"
[11,] "texas"      "legal"       "year"        "back"        "today"
[12,] "north"      "this"        "back"        "work"        "good"
[13,] "let"        "credit"      "work"        "information" "mark"
[14,] "gas"        "you"         "you"         "mark"        "group"
[15,] "send"       "corp"        "make"        "agreement"   "office"
[16,] "ill"        "day"         "agreement"   "america"     "agreement"
[17,] "work"       "conference"  "north"       "send"        "talk"
[18,] "mark"       "gas"         "corp"        "list"        "review"
[19,] "copy"       "message"     "draft"       "credit"      "send"
[20,] "back"       "street"      "check"       "company"     "give"
```

Topic Share by Week

```
> get_sentiment(text)
[1] 1.85 2.05 1.75 3.55 2.40
```



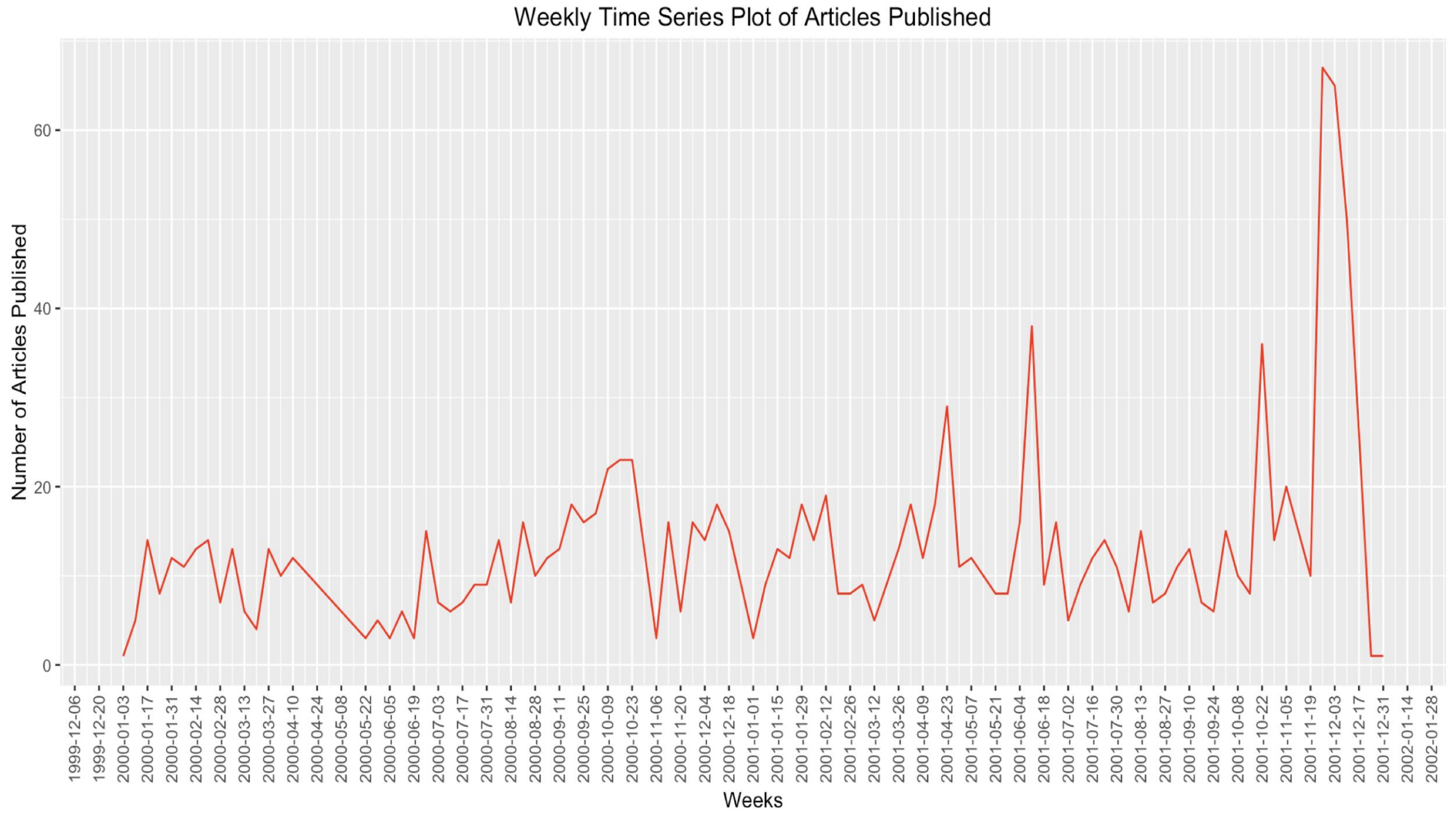
Analysis of News Using Factiva

Enron_Factiva_Analysis.Rmd & Enron_Factiva_Extract.Rmd

- Reads in **EnNews.RData** (created by **Enron_Factiva_Extract.Rmd** - this gives a data frame with {Date, Author, Header, Body}).
- Then the Analysis program proceeds to work on this data frame and adds more columns.
- Creates **MoodScoredDF_News.RData**. Creates **TDMWklyArt.RData**.
- The file is an entire program to do all aspects of text analytics: creating the text Body, sentiment extraction, wordclouds, topic modeling, POS tagging.
- POS tagging is a new analysis that is being tried out. See “The Secret Life of Pronouns: What Our Words Say About Us” by James Pennebaker.

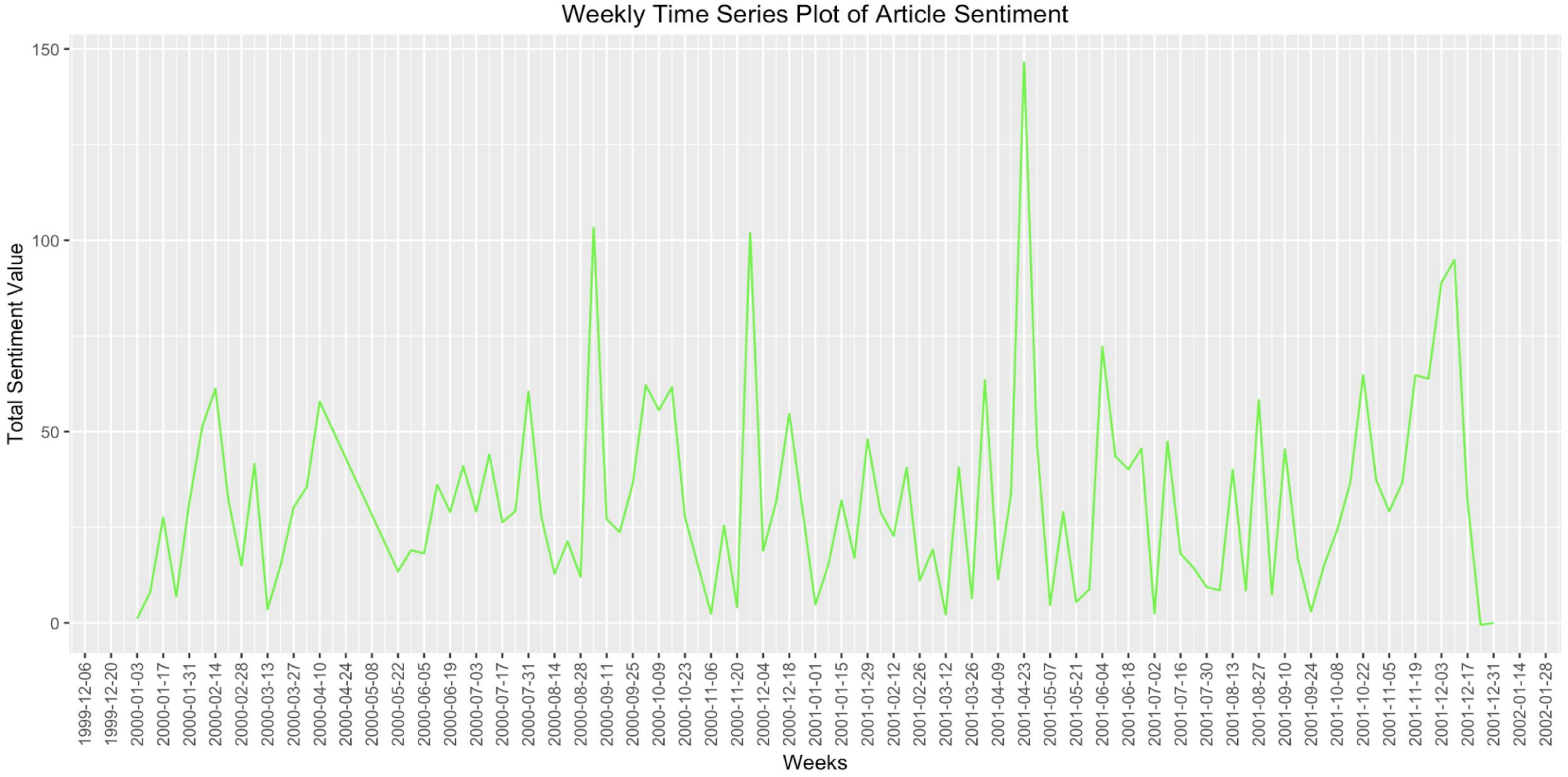
Factiva Data

- 1302 articles extracted from Factiva for the years 2000 and 2001.
- All pulled using the “Enron” keyword.
- Many are long articles with many pages; some have extensive financials.

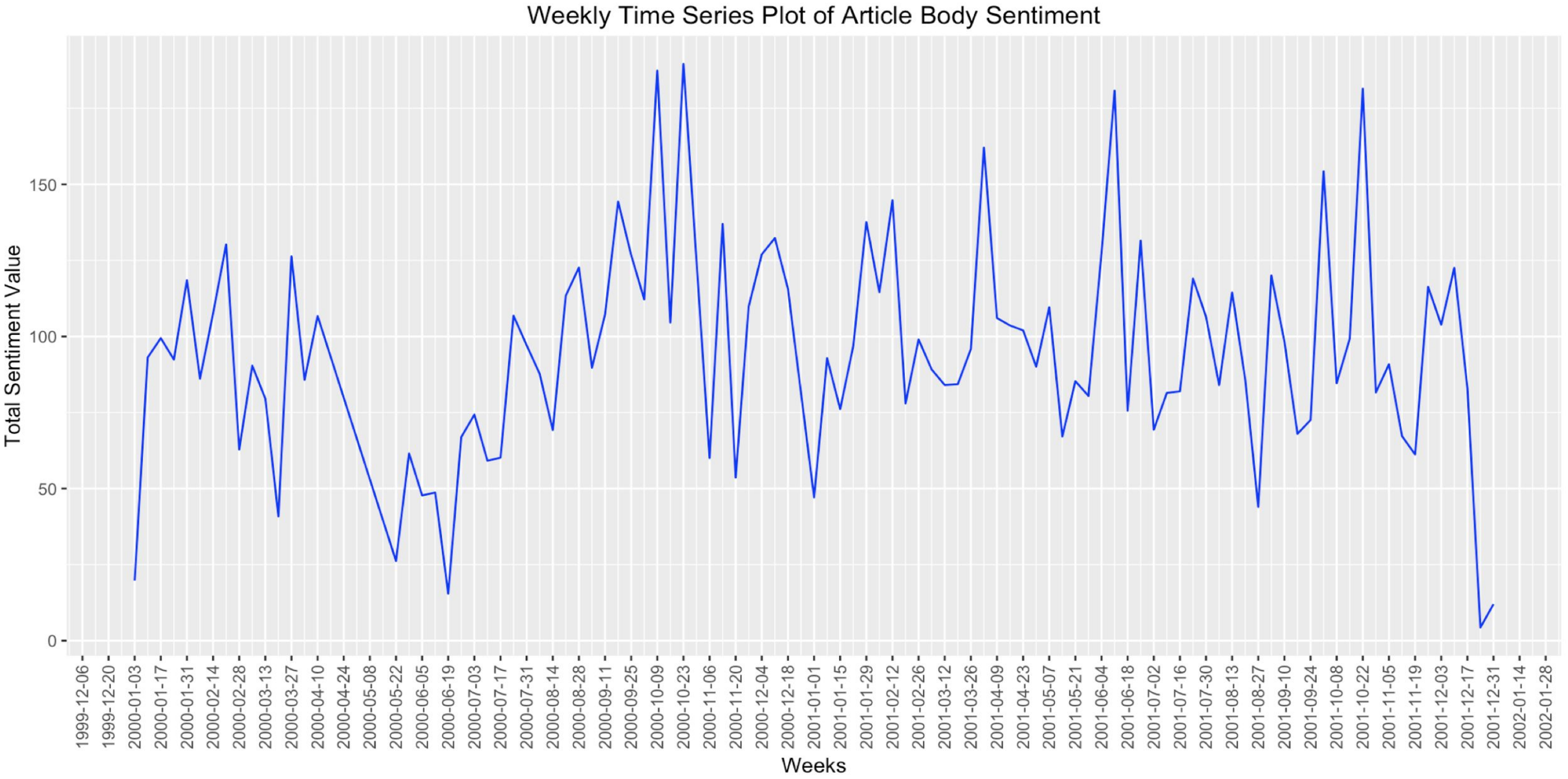


Weekly Time Series Plot of Articles Published
Number of Articles Published
0
20
40
60
Weeks
1999-12-06
1999-12-20
2000-01-03
2000-01-17
2000-01-31
2000-02-14
2000-02-28
2000-03-13
2000-03-27
2000-04-10
2000-04-24
2000-05-08
2000-05-22
2000-06-05
2000-06-19
2000-07-03
2000-07-17
2000-07-31
2000-08-14
2000-08-28
2000-09-11
2000-09-25
2000-10-09
2000-10-23
2000-11-06
2000-11-20
2000-12-04
2000-12-18
2001-01-01
2001-01-15
2001-01-29
2001-02-12
2001-02-26
2001-03-12
2001-03-26
2001-04-09
2001-04-23
2001-05-07
2001-05-21
2001-06-04
2001-06-18
2001-07-02
2001-07-16
2001-07-30
2001-08-13
2001-08-27
2001-09-10
2001-09-24
2001-10-08
2001-10-22
2001-11-05
2001-11-19
2001-12-03
2001-12-17
2001-12-31
2002-01-14
2002-01-28

Sentiment from Headers



Sentiment from Message Body

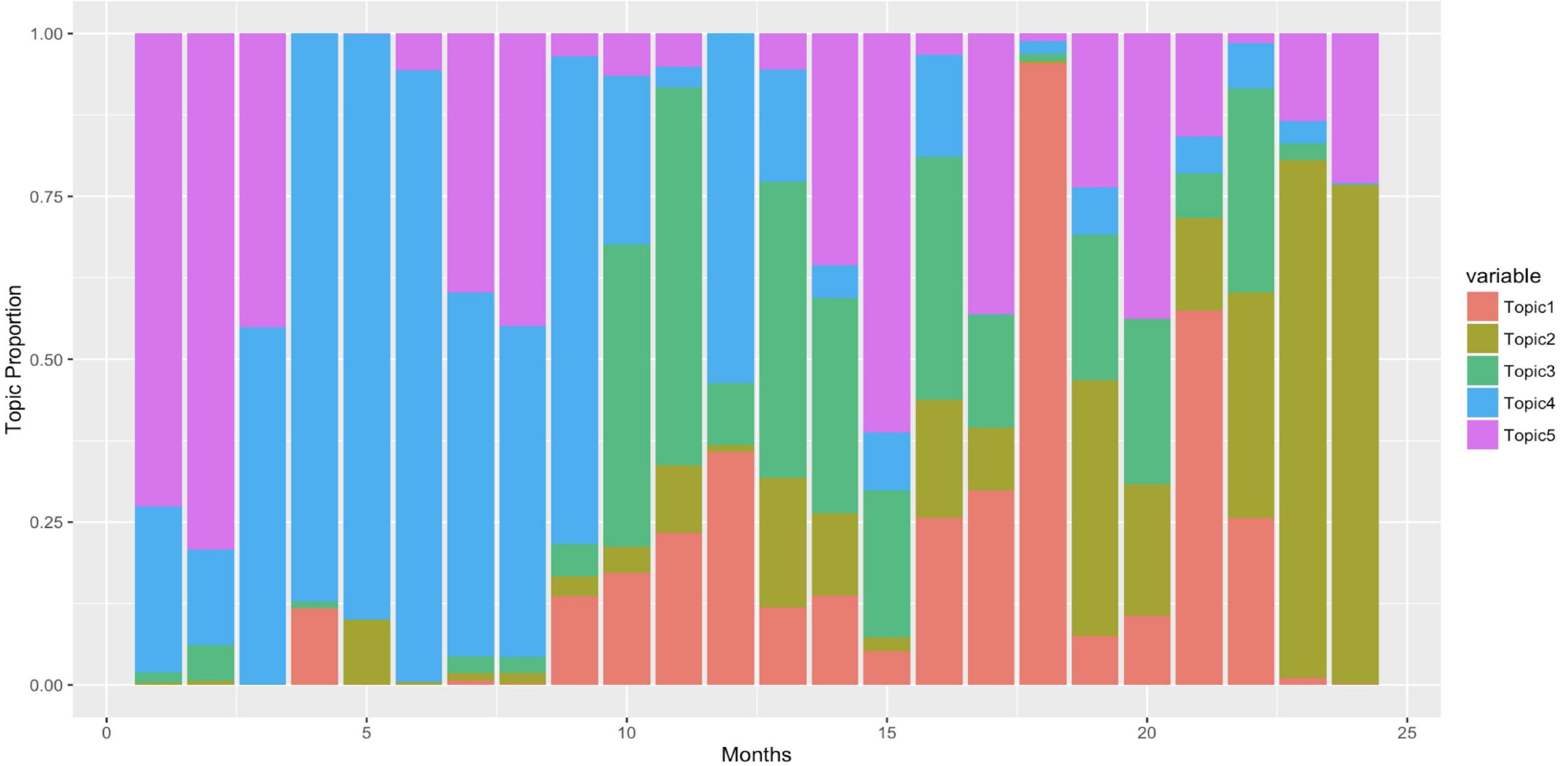


Topic Analysis of News

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
[1,]	"tibco"	"enron"	"tibco"	"the"	"the"
[2,]	"software"	"the"	"the"	"services"	"gas"
[3,]	"business"	"company"	"services"	"enron"	"company"
[4,]	"the"	"energy"	"quarter"	"company"	"million"
[5,]	"customers"	"gas"	"energy"	"energy"	"net"
[6,]	"information"	"million"	"net"	"internet"	"energy"
[7,]	"statements"	"statements"	"software"	"inc"	"income"
[8,]	"technology"	"nyse"	"information"	"business"	"quarter"
[9,]	"forwardlooking"	"financial"	"inc"	"technology"	"inc"
[10,]	"tibcos"	"natural"	"income"	"companies"	"oil"
[11,]	"integration"	"securities"	"customers"	"information"	"statements"
[12,]	"inc"	"business"	"statements"	"content"	"development"
[13,]	"companies"	"services"	"technology"	"network"	"production"
[14,]	"energy"	"forwardlooking"	"results"	"customers"	"total"
[15,]	"results"	"make"	"business"	"luminant"	"president"
[16,]	"including"	"results"	"management"	"service"	"year"
[17,]	"products"	"year"	"million"	"gas"	"companies"
[18,]	"systems"	"your"	"total"	"management"	"business"
[19,]	"services"	"click"	"forwardlooking"	"communications"	"cash"
[20,]	"realtime"	"count"	"share"	"industry"	"corporation"

Topic Contribution per Month

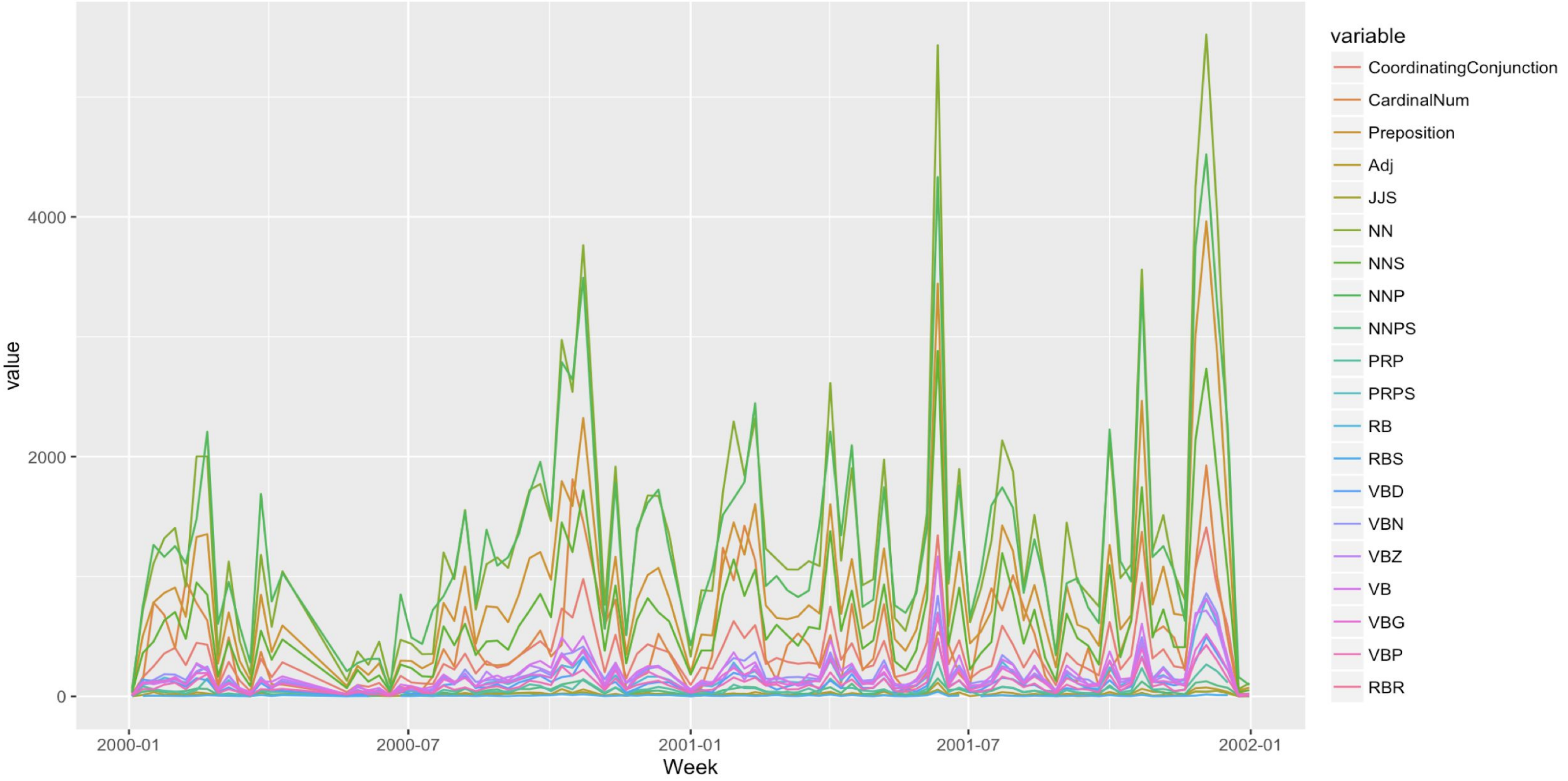
```
> get_sentiment(text)
[1] 1.35 0.85 1.65 1.50 0.25
```



Wordcloud 2000

Wordcloud 2001

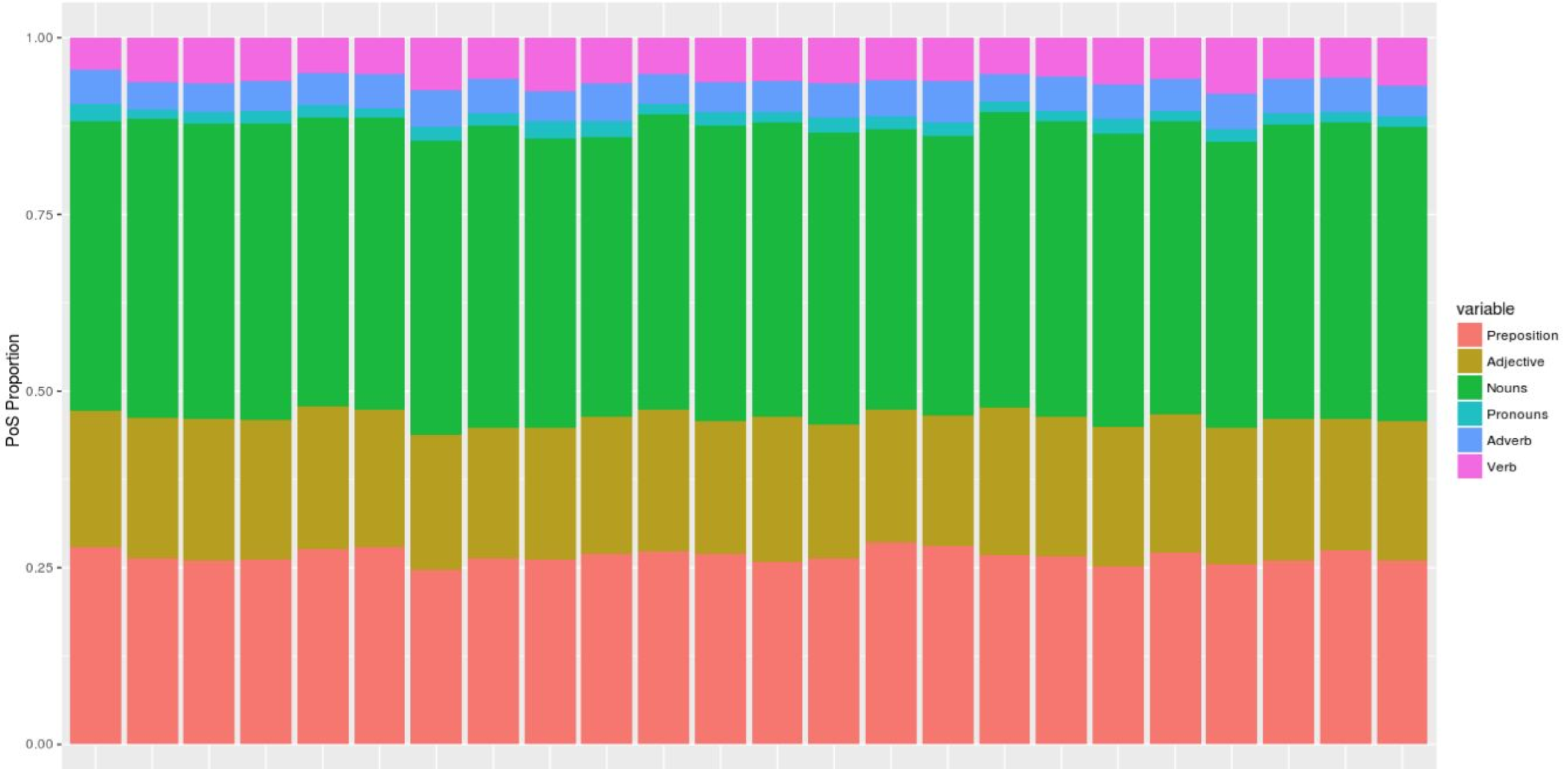
Parts of Speech (POS) Tagging



POS Tags

Part–of–Speech Tag	Part–of–Speech category
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non–3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh–determiner
WP	Wh–pronoun
WP$	Possessive wh–pronoun
WRB	Wh–adverb

Summarized POS Tagging

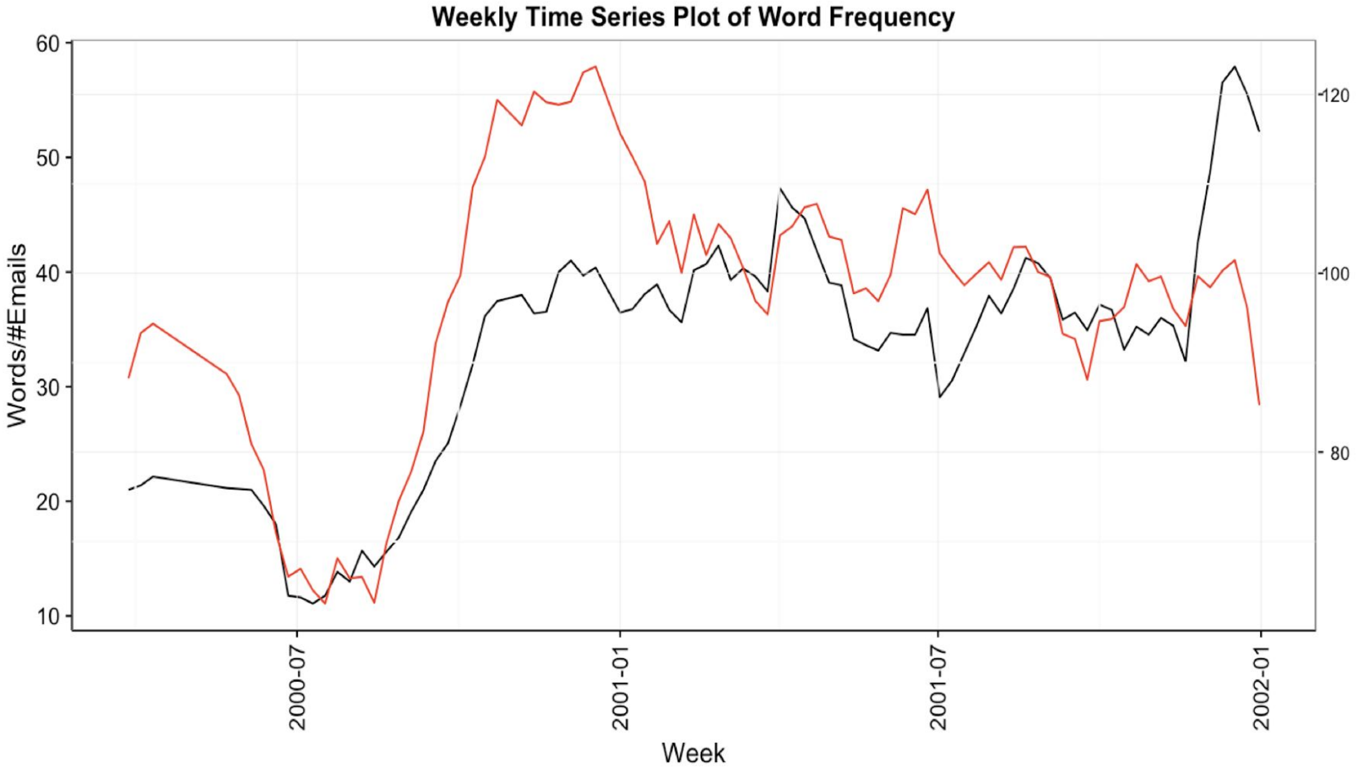


Word Play over Time for News Articles

Select a Word

energy

Word_vs_Sentiment Plot | Word_vs_Return Plot | Word_Sentiment_Correlation | Word_Return_Correlation

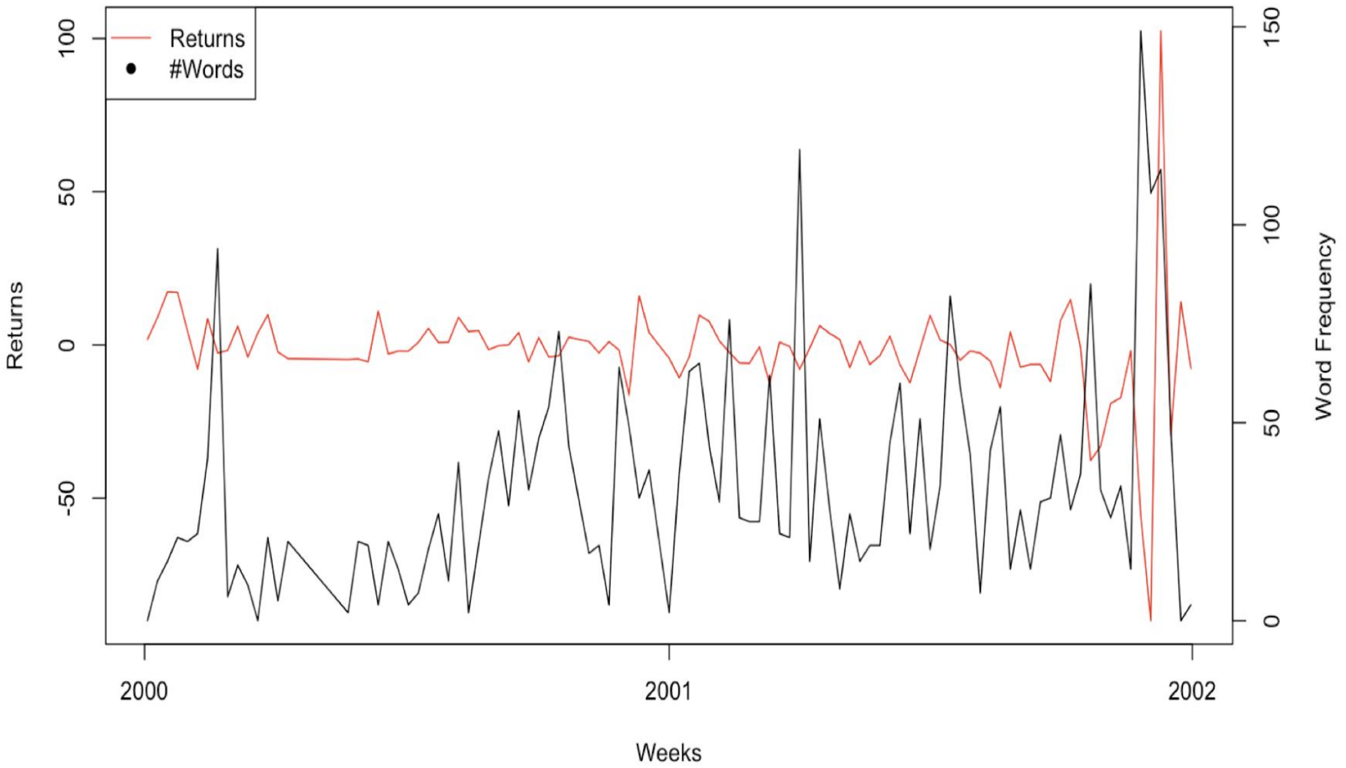


Black: words/#emails , Red: sentiment

Select a Word

energy

Word_vs_Sentiment Plot | Word_vs_Return Plot | Word_Sentiment_Correlation | Word_Return_Correlation



Blue: Stock Returns (left axis) / Red: Quarterly Mean of Words per Email (right axis)

Word_vs_Sentiment Plot | Word_vs_Return Plot | Word_Sentiment_Correlation | Word_Return_Correlation

Show 25 entries

Search: ener

wNames	corrDf	absVal
energy	0.56259664	0.56259664
general	0.44162487	0.44162487
generation	0.28745520	0.28745520
generate	0.25699226	0.25699226
energetic	0.10499805	0.10499805
energize	0.07957884	0.07957884
generosity	-0.02828462	0.02828462
generous	0.02575829	0.02575829
wNames	corrDf	absVal

Showing 1 to 8 of 8 entries (filtered from 4,205 total entries)

Previous 1 Next

Word_vs_Sentiment Plot | Word_vs_Return Plot | Word_Sentiment_Correlation | Word_Return_Correlation

Show 25 entries

Search: ener

wNames	wrCorrDf	absVal
energize	0.610292823	0.610292823
general	-0.280540083	0.280540083
energy	-0.205861335	0.205861335
generate	0.200333498	0.200333498
generation	-0.182518961	0.182518961
generosity	0.053955689	0.053955689
generous	0.047869085	0.047869085
energetic	0.003309663	0.003309663
wNames	wrCorrDf	absVal

Showing 1 to 8 of 8 entries (filtered from 4,205 total entries)

Previous 1 Next