Chapter 4: Readying the Data

1 Introduction

Speeches contains a wealth of information. As humans, we are taught to understand verbal and written communication – pick out the nouns, verbs, and adjectives, then combine the information to decipher meaing. Take the following excerpt from the 2010 State of the Union:

> Now, one place to start is serious financial reform. Look, I am not interested in punishing banks. I'm interested in protecting our economy. A strong, healthy financial market makes it possible for businesses to access credit and create new jobs. It channels the savings of families into investments that raise incomes. But that can only happen if we guard against the same recklessness that nearly brought down our entire economy. We need to make sure consumers and middle-class families have the information they need to make financial decisions. We can't allow financial institutions, including those that take your deposits, to take risks that threaten the whole economy.

To many, text might not be considered data despite the fact that any analytical mind with a command of the English language can identify key terms:

> ~~Now, one place to start is serious~~ financial reform. ~~Look, I am not interested in~~ punishing banks. ~~I'm interested in~~ protecting our economy. ~~A~~ strong, healthy financial market ~~makes it possible for~~ businesses ~~to access~~ credit ~~and~~ create new jobs. ~~It channels the~~ savings of families ~~into~~ investments ~~that~~ raise incomes. ~~But that can only happen if we guard against the same~~ recklessness ~~that nearly brought down our entire~~ economy. ~~We need to make sure~~ consumers ~~and~~ middle-class families ~~have the information they need to make~~financial decisions. ~~We can't allow~~ financial institutions, ~~including those that take your~~ deposits,~~to take risks that threaten the whole~~ economy.

Much like the logic that guides keyword identification, text can be shaped from an unstructured dataset into a well-defined, structured dataset:

Table 1: Most frequent terms found in excerpt.

Terms	Frequency of Term	Number of Characters
financial	4	9
economy	3	7
families	2	8
interested	2	10

Of course, this process could be done manually, but imagine sorting through all 7,304 words in the 2010 address or scaling the process to the roughly *1.9 million words* in addresses State of the Union addresses between 1790 and 2016. All the steps required to convert unstructured text into usable data can be done with a little bit of planning, technical imagination and data manipulation. Every little detail about the data needs to be considered and meticulously converted into a usable form. From a data format perspective, capitalized characters are not the same as lower case. Contractions are not the same as terms that are spelled out. Punctuation affect spacing. Carriage returns and new line markers, while not visible in reading mode, are recorded.

Let's take one line from above and dissect the changes that need to be made:

> "We need to make sure consumers and middle-class families have the information they need to make financial decisions. We can't allow financial institutions, including those that take your deposits, to take risks that threaten the whole economy."

We then turn everything into lower case so all letters of the alphabet are read the same.

> “we need to make sure consumers and middle-class families have the information they need to make financial decisions. we can’t allow financial institutions, including those that take your deposits, to take risks that threaten the whole economy.”

Then, we get rid of punctuation by substituting values with empty quotations ("").

> “we need to make sure consumers and middleclass families have the information they need to make financial decisions we cant allow financial institutions including those that take your deposits to take risks that threaten the whole economy”

Each space between each word can be used as a *delimiter* that can be used as a symbol for a program to break apart words into elements in a list.

Table 2: Terms

we	families	financial	those	that
need	have	decisions	that	threaten
to	the	we	take	the
make	information	cant	your	whole
sure	they	allow	deposits	economy
consumers	need	financial	to	
and	to	institutions	take	
middleclass	make	including	risks	

There are words in there that don’t add much value as they are commonplace and filler. In text processing, these words are known as *stop words*. In each domain, the list of stop words likely differs, thus data scientists may need to build a customized list. For simplicity, we’ve used a stop words list that is used in the mySQL – an open source relational database management system. The result is the list of remaining words.

Table 3: Terms after removing stop words

make	information	financial	risks
consumers	make	institutions	threaten
middleclass	financial	including	economy
families	decisions	deposits	

From that data, we can aggregate the data into a form that is meaningful to answer a research question. For example, the frequency of words may provide a clue as to what the text is about. In this case, each “financial” and “make” appear twice in the text, perhaps indicating that there is an orientation towards action (make) for financial considerations.

Table 4: Term Frequencies

Term	Freq	Term	Freq
financial	2	including	1
make	2	information	1
consumers	1	institutions	1
decisions	1	middleclass	1
deposits	1	risks	1
economy	1	threaten	1
families	1		

This is just the tip of the iceberg. Text processing is just one aspect of readying your data for use. Much of a data scientist's time is spent retrieving, assembling, manipulating, and transforming data so that it is useful. While there are entire texts dedicated to engineering data, this chapter offers a brief review of programming paradigms that are necessary to bring to bear data in the public and social sectors.

2 The Ideal Data Set

Concept of tidy data goes here

3 Retrieval and Assembly

While we briefly covered loading of data in the previous chapter, the retrieval and assembly of a data set can be one of the largest barriers to getting a project off the ground. Here, we lay out a few practices that make the process far simpler.

There are a multitude of data storage formats in use. Fortunately, R is equipped to load virtually all data formats. Below is a recommended set of functions that are easy to use and flexible.

Format	Function	Package	Description
Excel files (.xls, .xlsx)	read_excel	readxl	Load Excel files into a data frame. Note that with ex
Comma Separated Values (.csv)	read_csv	readr	Load any comma separated file into data frame forma
Other delimited values (.tab, .txt)	read_delim	readr	Load delimited file using any delimiter such as tab de
Free form text	readLines	base R	Read a text file, line by line. This is helpful for work
Extensible Markup Language (XML)	xmlToDataFrame	XML	Read XML into a data frame if the structure is simpl
Extensible Markup Language (XML)	xmlToList	XML	If the XML is complicated, read each element as a lis
JavaSript Object Notatin (JSON)	fromJSON	rjson	Read JSON into list. If JSON is a flat, non-hierarchi
Stata (.dta)	read_dta	haven	Load Stata data file
SAS (.sas7bdat)	read_sas	haven	Load SAS data files
SPSS (.sav, .por)	read_spss	haven	Load SPSS data files
R (.Rda)	load	base R	Load saved R data set that can contain multiple obje
R (.RDS)	readRDS	base R	Read individually saved R object.