Reproducible Research

Knitr allows you integrate multiple languages: HTML, Markdown, LaTeX

Steps in a data analysis

Define the question

Define the ideal data set

Determine what data you can access

Obtain the data

Clean the data

Exploratory data analysis

Statistical prediction/modeling

Interpret results

Challenge results

Synthesize/write up results

Create reproducible code

Steps in a data analysis

"Ask yourselves, what problem have you solved, ever, that was worth solving, where you knew all of the given information in advance? Where you didn’t have a surplus of information and have to filter it out, or you had insufficient information and have to go find some?"

Dan Myer, Mathematics Educator

Defining a question – THIS IS THE MOST POWERFUL DIMENSION REDUCTION TOOL YOU CAN EMPLOY (allows you to figure out what variables you should be working with)

Statistical methods development

Danger zone!!!

Proper data analysis

An example

Start with a general question

Can I automatically detect emails that are SPAM that are not?

Make it concrete

Can I use quantitative characteristics of the emails to classify them as SPAM/HAM?

Define the ideal data set

The data set may depend on your goal

Descriptive - a whole population

Exploratory - a random sample with many variables measured

Inferential - the right population, randomly sampled

Predictive - a training and test data set from the same population

Causal - data from a randomized study

Mechanistic - data about all components of the system

Our example

<http://www.google.com/about/datacenters/inside/>

Determine what data you can access

Sometimes you can find data free on the web

Other times you may need to buy the data

Be sure to respect the terms of use

If the data don't exist, you may need to generate it yourself

Back to our example

A possible solution

http://archive.ics.uci.edu/ml/datasets/Spambase

Obtain the data

Try to obtain the raw data

Be sure to reference the source

Polite emails go a long way

If you will load the data from an internet source, record the url and time accessed

Our data set

http://search.r-project.org/library/kernlab/html/spam.html

Clean the data

Raw data often needs to be processed

If it is pre-processed, make sure you understand how

Understand the source of the data (census, sample, convenience sample, etc.)

May need reformating, subsampling - **record these steps**

Determine if the data are good enough - if not, quit or change data

Our cleaned data set

# If it isn't installed, install the kernlab package with install.packages()

library(kernlab)

data(spam)

str(spam[, 1:5])

'data.frame': 4601 obs. of 5 variables:

$ make : num 0 0.21 0.06 0 0 0 0 0 0.15 0.06 ...

$ address: num 0.64 0.28 0 0 0 0 0 0 0 0.12 ...

$ all : num 0.64 0.5 0.71 0 0 0 0 0 0.46 0.77 ...

$ num3d : num 0 0 0 0 0 0 0 0 0 0 ...

$ our : num 0.32 0.14 1.23 0.63 0.63 1.85 1.92 1.88 0.61 0.19 ...

<http://search.r-project.org/library/kernlab/html/spam.html>

An example

Start with a general question

Can I automatically detect emails that are SPAM or not?

Make it concrete

Can I use quantitative characteristics of the emails to classify them as SPAM/HAM?

Our data set

http://search.r-project.org/library/kernlab/html/spam.html

Subsampling our data set

We need to generate a test and training set (prediction)

# If it isn't installed, install the kernlab package

library(kernlab)

data(spam)

# Perform the subsampling to split data into two random groups, one for building model and other for testing

set.seed(3435)

trainIndicator = rbinom(4601, size = 1, prob = 0.5)

table(trainIndicator)

## trainIndicator

## 0 1

## 2314 2287

trainSpam = spam[trainIndicator == 1, ]

testSpam = spam[trainIndicator == 0, ]

Exploratory data analysis

Look at summaries of the data

Check for missing data

Create exploratory plots

Perform exploratory analyses (e.g. clustering)

Names

names(trainSpam)

## [1] "make" "address" "all"

## [4] "num3d" "our" "over"

## [7] "remove" "internet" "order"

## [10] "mail" "receive" "will"

## [13] "people" "report" "addresses"

## [16] "free" "business" "email"

## [19] "you" "credit" "your"

## [22] "font" "num000" "money"

## [25] "hp" "hpl" "george"

## [28] "num650" "lab" "labs"

## [31] "telnet" "num857" "data"

## [34] "num415" "num85" "technology"

## [37] "num1999" "parts" "pm"

## [40] "direct" "cs" "meeting"

## [43] "original" "project" "re"

## [46] "edu" "table" "conference"

## [49] "charSemicolon" "charRoundbracket" "charSquarebracket"

## [52] "charExclamation" "charDollar" "charHash"

## [55] "capitalAve" "capitalLong" "capitalTotal"

## [58] "type"

Head

head(trainSpam)

## make address all num3d our over remove internet order mail receive

## 1 0.00 0.64 0.64 0 0.32 0.00 0.00 0 0.00 0.00 0.00

## 7 0.00 0.00 0.00 0 1.92 0.00 0.00 0 0.00 0.64 0.96

## 9 0.15 0.00 0.46 0 0.61 0.00 0.30 0 0.92 0.76 0.76

## 12 0.00 0.00 0.25 0 0.38 0.25 0.25 0 0.00 0.00 0.12

## 14 0.00 0.00 0.00 0 0.90 0.00 0.90 0 0.00 0.90 0.90

## 16 0.00 0.42 0.42 0 1.27 0.00 0.42 0 0.00 1.27 0.00

## will people report addresses free business email you credit your font

## 1 0.64 0.00 0 0 0.32 0 1.29 1.93 0.00 0.96 0

## 7 1.28 0.00 0 0 0.96 0 0.32 3.85 0.00 0.64 0

## 9 0.92 0.00 0 0 0.00 0 0.15 1.23 3.53 2.00 0

## 12 0.12 0.12 0 0 0.00 0 0.00 1.16 0.00 0.77 0

## 14 0.00 0.90 0 0 0.00 0 0.00 2.72 0.00 0.90 0

## 16 0.00 0.00 0 0 1.27 0 0.00 1.70 0.42 1.27 0

## num000 money hp hpl george num650 lab labs telnet num857 data num415

## 1 0 0.00 0 0 0 0 0 0 0 0 0.00 0

## 7 0 0.00 0 0 0 0 0 0 0 0 0.00 0

## 9 0 0.15 0 0 0 0 0 0 0 0 0.15 0

## 12 0 0.00 0 0 0 0 0 0 0 0 0.00 0

## 14 0 0.00 0 0 0 0 0 0 0 0 0.00 0

## 16 0 0.42 0 0 0 0 0 0 0 0 0.00 0

## num85 technology num1999 parts pm direct cs meeting original project re

## 1 0 0 0.00 0 0 0.00 0 0 0.0 0 0

## 7 0 0 0.00 0 0 0.00 0 0 0.0 0 0

## 9 0 0 0.00 0 0 0.00 0 0 0.3 0 0

## 12 0 0 0.00 0 0 0.00 0 0 0.0 0 0

## 14 0 0 0.00 0 0 0.00 0 0 0.0 0 0

## 16 0 0 1.27 0 0 0.42 0 0 0.0 0 0

## edu table conference charSemicolon charRoundbracket charSquarebracket

## 1 0 0 0 0.000 0.000 0

## 7 0 0 0 0.000 0.054 0

## 9 0 0 0 0.000 0.271 0

## 12 0 0 0 0.022 0.044 0

## 14 0 0 0 0.000 0.000 0

## 16 0 0 0 0.000 0.063 0

## charExclamation charDollar charHash capitalAve capitalLong capitalTotal

## 1 0.778 0.000 0.000 3.756 61 278

## 7 0.164 0.054 0.000 1.671 4 112

## 9 0.181 0.203 0.022 9.744 445 1257

## 12 0.663 0.000 0.000 1.243 11 184

## 14 0.000 0.000 0.000 2.083 7 25

## 16 0.572 0.063 0.000 5.659 55 249

## type

## 1 spam

## 7 spam

## 9 spam

## 12 spam

## 14 spam

## 16 spam

Summaries

table(trainSpam$type)

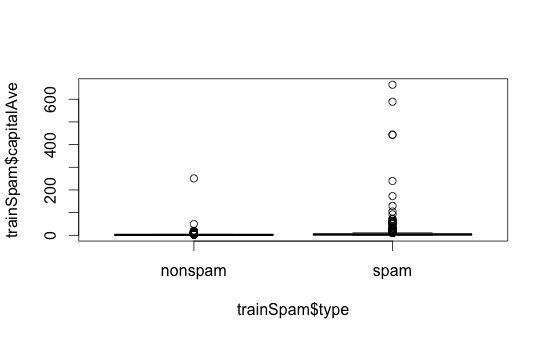
##

## nonspam spam

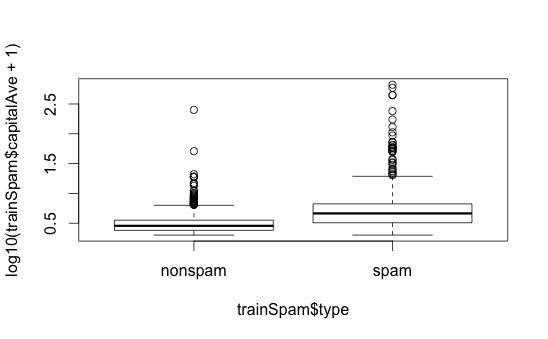
## 1381 906

Plots

plot(trainSpam$capitalAve ~ trainSpam$type)

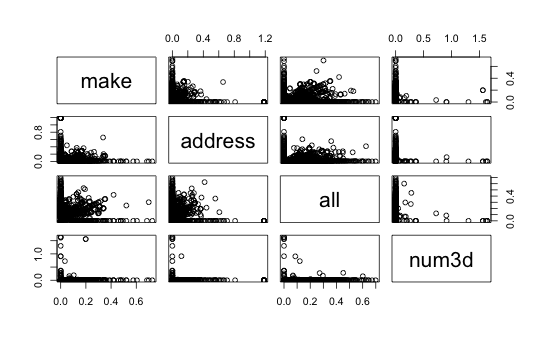


plot(log10(trainSpam$capitalAve + 1) ~ trainSpam$type)



Relationships between predictors (pairwise plots, log transformation of each of the variables)

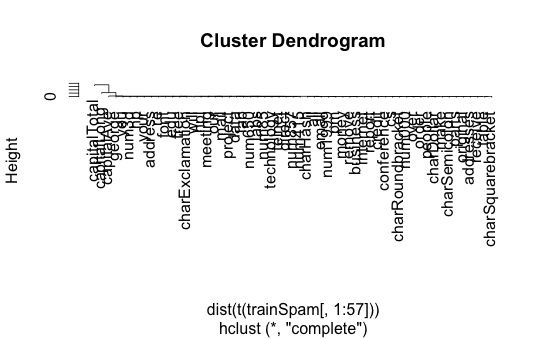
plot(log10(trainSpam[, 1:4] + 1))



Clustering – what words cluster together

hCluster = hclust(dist(t(trainSpam[, 1:57])))

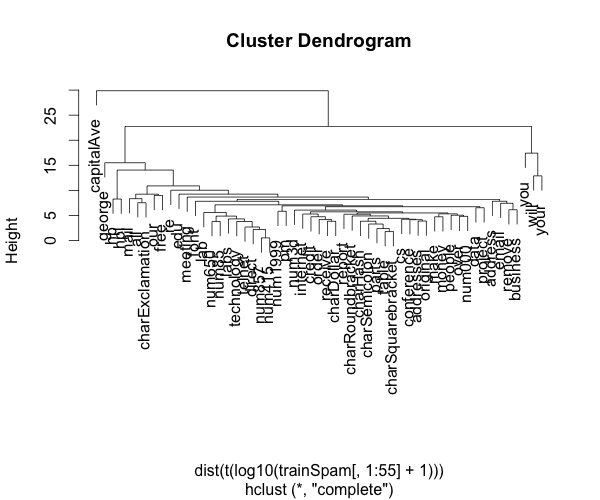
plot(hCluster)



New clustering – redo after a transformation of the predictors, here take log10 and add 1 to avoid taking the log of zero

hClusterUpdated = hclust(dist(t(log10(trainSpam[, 1:55] + 1))))

plot(hClusterUpdated)



Statistical prediction/modeling

Should be informed by the results of your exploratory analysis

Exact methods depend on the question of interest

Transformations/processing should be accounted for when necessary

Measures of uncertainty should be reported

Statistical prediction/modeling

## Go thru each of the variables in the data set and try and fit a generalizable linear model, in this case a logistic regression, to see if we can predict whether an email is spam or not by using just one variable

trainSpam$numType = as.numeric(trainSpam$type) - 1

costFunction = function(x, y) sum(x != (y > 0.5))

cvError = rep(NA, 55)

library(boot)

for (i in 1:55) {

lmFormula = reformulate(names(trainSpam)[i], response = "numType")

glmFit = glm(lmFormula, family = "binomial", data = trainSpam)

cvError[i] = cv.glm(trainSpam, glmFit, costFunction, 2)$delta[2]

}

## Which predictor has minimum cross-validated error?

names(trainSpam)[which.min(cvError)]

## [1] "charDollar"

Get a measure of uncertainty

## Use the best model from the group

predictionModel = glm(numType ~ charDollar, family = "binomial", data = trainSpam)

## Get predictions on the test set

predictionTest = predict(predictionModel, testSpam)

predictedSpam = rep("nonspam", dim(testSpam)[1])

## Classify as `spam' for those with prob > 0.5

predictedSpam[predictionModel$fitted > 0.5] = "spam"

Get a measure of uncertainty

## Classification table

table(predictedSpam, testSpam$type)

##

## predictedSpam nonspam spam

## nonspam 1346 458

## spam 61 449

## Error rate

(61 + 458)/(1346 + 458 + 61 + 449)

## [1] 0.2243

Interpret results

Use the appropriate language

describes

correlates with/associated with

leads to/causes

predicts

Give an explanation

Interpret coefficients

Interpret measures of uncertainty

Our example

The fraction of characters that are dollar signs can be used to predict if an email is Spam

Anything with more than 6.6% dollar signs is classified as Spam

More dollar signs always means more Spam under our prediction

Our test set error rate was 22.4%

Challenge results

Challenge all steps:

Question

Data source

Processing

Analysis

Conclusions

Challenge measures of uncertainty

Challenge choices of terms to include in models

Think of potential alternative analyses

Synthesize/write-up results

Lead with the question

Summarize the analyses into the story

Don't include every analysis, include it

If it is needed for the story

If it is needed to address a challenge

Order analyses according to the story, rather than chronologically

Include "pretty" figures that contribute to the story

In our example

Lead with the question

Can I use quantitative characteristics of the emails to classify them as SPAM/HAM?

Describe the approach

Collected data from UCI -> created training/test sets

Explored relationships

Choose logistic model on training set by cross validation

Applied to test, 78% test set accuracy

Interpret results

Number of dollar signs seems reasonable, e.g. "Make money with Viagra \$ \$ \$ \$!"

Challenge results

78% isn't that great

I could use more variables

Why logistic regression?

Create reproducible code

Organizing Data Analysis Files

Data

* Raw data
* Processed data

Figures

* Exploratory figures
* Final figures

R code

* Raw / unused scripts
* Final scripts
* R Markdown files

Text

* README files
* Text of analysis / report

Raw Data

Should be stored in your analysis folder

If accessed from the web, include url, description, and date accessed in README

Processed data

Processed data should be named so it is easy to see which script generated the data.

The processing script - processed data mapping should occur in the README

Processed data should be [tidy](http://vita.had.co.nz/papers/tidy-data.pdf)

Exploratory figures

Figures made during the course of your analysis, not necessarily part of your final report.

They do not need to be "pretty"

Final Figures

Usually a small subset of the original figures

Axes/colors set to make the figure clear

Possibly multiple panels

Raw scripts

May be less commented (but comments help you!)

May be multiple versions

May include analyses that are later discarded

Final scripts

Clearly commented

Small comments liberally - what, when, why, how

Bigger commented blocks for whole sections

Include processing details

Only analyses that appear in the final write-up

R markdown files

[R markdown](http://www.rstudio.com/ide/docs/authoring/using_markdown) files can be used to generate reproducible reports

Text and R code are integrated

Very easy to create in [Rstudio](http://www.rstudio.com/)

Readme files

Not necessary if you use R markdown

Should contain step-by-step instructions for analysis

Here is an example <https://github.com/jtleek/swfdr/blob/master/README>

Text of the document

It should include a title, introduction (motivation), methods (statistics you used), results (including measures of uncertainty), and conclusions (including potential problems)

It should tell a story

It should not include every analysis you performed

References should be included for statistical methods

Further resources

Information about a non-reproducible study that led to cancer patients being mistreated: [The Duke Saga Starter Set](http://simplystatistics.org/2012/02/27/the-duke-saga-starter-set/)

[Reproducible research and Biostatistics](http://biostatistics.oxfordjournals.org/content/10/3/405.full)

[Managing a statistical analysis project guidelines and best practices](http://www.r-statistics.com/2010/09/managing-a-statistical-analysis-project-guidelines-and-best-practices/)

[Project template](http://projecttemplate.net/) - a pre-organized set of files for data analysis

**WEEK 2**

Coding Standards

1. Always save code in text files and formats, so can always be read by anyone
2. Indent your code, makes structure of your code obvious
3. Limit the width of your code to about 80 columns (80 characters), and 4 or 8 spaces for indent
4. Limit the length of functions to ONE ACTIVITY (eg read the data), otherwise split it up into different, logical functions

Markdown

What is Markdown?

"Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML)."

- [**John Gruber, creator of Markdown**](http://daringfireball.net/projects/markdown/)

Markdown Syntax

Italics

*\*This text will appear italicized!\**

This text will appear italicized!

Markdown Syntax

Bold

**\*\*This text will appear bold!\*\***

**This text will appear bold!**

Markdown Syntax

Headings (one pound sing is primary heading)

**## This is a secondary heading**

## This is a secondary heading

**### This is a tertiary heading**

### This is a tertiary heading

Markdown Syntax

Unordered Lists

- first item in list

- second item in list

- third item in list

* first item in list
* second item in list
* third item in list

Markdown Syntax

Ordered Lists

1. first item in list

2. second item in list

3. third item in list

1. first item in list
2. second item in list
3. third item in list

Markdown Syntax

Links

[Johns Hopkins Bloomberg School of Public Health](<http://www.jhsph.edu/>)

[Download R](<http://www.r-project.org/>)

[RStudio](<http://www.rstudio.com/>)

[Johns Hopkins Bloomberg School of Public Health](http://www.jhsph.edu/)  
[Download R](http://www.r-project.org/)  
[RStudio](http://www.rstudio.com/)

Markdown Syntax

Advanced Linkin

I spend so much time reading [R bloggers][1] and [Simply Statistics][2]!

[1]: http://www.r-bloggers.com/ "R bloggers"

[2]: http://simplystatistics.org/ "Simply Statistics"

I spend so much time reading [R bloggers](http://www.r-bloggers.com/) and [Simply Statistics](http://simplystatistics.org/)!

Markdown Syntax

Newlines

Newlines require a double space after the end of a line.

First line

Second line

First line Second line

First line[TWO SPACES]

Second line

First line  
Second line

Markdown Resources

[The Offical Markdown Documentation](http://daringfireball.net/projects/markdown/basics)

[Github's Markdown Guide](https://help.github.com/articles/github-flavored-markdown)

What is R markdown?

Basically, the input has just R code, and the output is the code in grey boxes, and then the results, either computation or function result, or a plot

Integrates R code with markdown (actually executes the code in the mkd file)

Allows one to create docs containing „live“ R code

R code is evaluated as part of the processing oft he markdown

Results from R code are inserted into markdown document

A core tool in literate statistical programming

R mkd can be converted to standard mkd using teh knitr package in R

Mkd can be converted to HTML using the mkd package in R

Any basic text editor can be used to create a mkd doc

The R mkd [only one you ever edit)🡪 mkd 🡪 HTML work flow can easily be managed using R Studio

These slides (☺) were written in R mkd and converted to slides using the slidify package

Knitr

The point is to bring data, text and code all together (integrates analysis with textual representations so it’s all linked together)

if you want JUST THE RESULTS, no code showing, just set echo=FALSE when you {r simulationg, echo=FALSE}

What if one code chunk takes a long time to run? The cache=TRUE option can be set on a chunk by chunk basis to store results of computation

After first run, results are loaded from the cache