Final Projects

R Statistical Language

Patricia Hoffman, PhD

Data Sets

Data Sets in R Packages

- List the Data Sets in an R Package

```
install.packages("gcookbook")
library(gcookbook)
data("gcookbook")
try(data(package = "gcookbook"))
```

- Obtain Information about a specific Data Sets in an R Package

```
??gcookbook::countries
```

Package gcookbook Data Sets

aapl	Apple stock data
anthoming	Homing in desert ants
cabbage_exp	Summary of cabbages data set
climate	Global climate temperature anomaly data from 1800 to 2011
corneas	Corneal thickness of eyes
countries	Health and economic data about countries around the world
heightweight	Height and weight of schoolchildren
isabel	Data from simulation of hurricane Isabel
marathon	Marathon and half-marathon times
pg_mean	Means of results from an experiment on plant growth
tophitters2001	Batting averages top hitters in Major League Baseball
uspopage	Age distribution of population in the United States, 1900-2002
uspopchange	Change in population of states in the U.S. between 2000 -2010

And Many More

Package: gcookbook
Data Set: countries

- Description
  - Health and economic data about countries around the world from 1960-2010
- Variables
  - Name: Name of country
  - Code: Short country code
  - Year
  - GDP: Per capita Gross Domestic Product, in adjusted 2011 U.S. Dollars
  - laborrate: Labor rate.
  - healthexp: Health expenditures in U.S. Dollars.
  - infmortality: Infant mortality per 1000 live births.
- Source
  - World Bank: http://data.worldbank.org/

Resources: Datasets

- UCI Repository: http://www.ics.uci.edu/~mlearn/MLRepository.html
- UCI KDD Archive: http://kdd.ics.uci.edu/summary.data.application.html
- Statlib: http://lib.stat.cmu.edu/
- Delve: http://www.cs.utoronto.ca/~delve/
- SVM
  - http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
  - http://www.csie.ntu.edu.tw/~cjlin/papers/guide/data/

Resources: Datasets

- Stanford Large Network Dataset Collection: http://snap.stanford.edu/data/
- Gapminder http://www.gapminder.org/

Specialized Collections

- KDD Nuggets:
  - http://www.kdnuggets.com/datasets/index.html
- CMU Statlab
  - http://lib.stat.cmu.edu/datasets/
- Gene Expression
  - http://www.ncbi.nlm.nih.gov/geo/
- Cornell’s arXiv Bulk Data Access
  - http://arxiv.org/help/bulk_data
- Amazon Web Services Public Data Sets
  - http://aws.amazon.com/publicdatasets/

/

Government Data Sources

- http://www.data.gov/
- US Department of Housing and Urban Development Housing scorecard at http://portal.hud.gov Housing data
- Census bureau at http://www.census.gov for Economic and social demographic indicators
- The Federal Reserve Bank of St Louis at http://research.stlouisfed.org/fred2/ for economic and interest rate related statistics.
- US Department of the Treasury at www.treasury.gov/resource-center/data-chart-center for detailed information on yield curves and treasury bills, bonds price.
- Bureau of Economic Analysis at http://www.bea.gov for comprehensive data sets on GDP and Trade.
- Medicare Data: http://www.medicare.gov/hospitalcompare/search.html?AspxAutoDetectCookieSupport=1
- Bureau of Labor http://www.bls.gov/
- Bureau of Transportation http://www.transtats.bts.gov/DatabaseInfo.asp?DB_ID=120&Link=0

Open Government Sites

- USA Survey Data
  - http://www.asdfree.com
- U.S. http://www.data.gov/
  - List of cities/states with open data
- United Kingdom http://data.gov.uk/
- France http://www.data.gouv.fr/
- Australia http://data.gov.au/
- Germany https://www.govdata.de/
- Many more http://www.data.gov/opendatasites

Open Government Sites

- United Nations http://data.un.org/
- U.S. http://www.data.gov/
  - List of cities/states with open data
- United Kingdom http://data.gov.uk/
- France http://www.data.gouv.fr/
- Ghana http://data.gov.gh/
- Australia http://data.gov.au/
- Germany https://www.govdata.de/

Open Government Sites

- Hong Kong http://www.gov.hk/en/theme/psi/datasets/
- Japan http://www.data.go.jp/
- England http://data.london.gov.uk/dataset
- Many more http://www.data.gov/opendatasites

Yahoo! Data Sets

http://webscope.sandbox.yahoo.com/

- Advertising and Market Data
- Competition Data
- Computing Systems Data
- Graph and Social Data
- Image Data
- Language Data
- Ratings and Classification Data

More Data Sites

- Info Chimps Market Place
  - http://www.infochimps.com/marketplace
- Kaggle
  - http://www.kaggle.com/
- Data Scientist
  - ((http://blog.mortardata.com/post/67652898761/6-dataset-lists-curated-by-data-scientists)
- Hilary Mason
  - http://bitly.com/bundles/hmason/1
- Peter Skomoroch
  - https://delicious.com/pskomoroch/dataset
- Jeff Hammerbacher
  - http://www.quora.com/Jeff-Hammerbacher/Introduction-to-Data-Science-Data-Sets
- Gregory Piatetsky-Shapiro
  - http://www.kdnuggets.com/gps.html

Even More Data Sites

- http://factfinder2.census.gov/faces/nav/jsf/pages/index.xhtml
- https://archive.ics.uci.edu/ml/machine-learning-databases/00215/
- Google’s Public Data Sets
- http://www.google.com/publicdata/directory
- List of Data Sites & Contests
  - http://www.rdatamining.com/resources/data

Free Financial Data Sources

- Yahoo finances at http://finance.yahoo.com for stocks and fundamental analysis data.
- Online Data Robert Shiller at http://www.econ.yale.edu/~shiller/data.htm for predictive models for housing and stock market confidence.
- RBS Group databank at http://www.databank.rbs.com for currency exchange rates and commodity prices.
- Simian Savants Charts at http://www.sschart.com/cmedata.shtml for currency and stock market indices.
- Kumo at http://pages.swcp.com/stocks/ for all S&P 500 stocks.
- StockCharts at http://www.stockcharts.com
- Reuters at http://www.reuters.com for stocks and market indices.
- Nasdaq at http://www.nasdaq.com for corporate data financial metrics

API’s with R interfaces

- twitter and twitteR package
- figshare and rfigshare
- PLoS and rplos
- rOpenSci
- Facebook and RFacebook
- Google maps and RGoogleMaps
- Tutorials

  http://thinktostart.com/category/r-tutorials/

Financial Data using REST API

- Quandl at http://www.quandl.com
- Xignite Global Real Time at http://www.xignite.com/product/global-real-time-stock-quote-data/
- Golb at http://www.goldb.org/ystockquote.html

Journal Paper

Resources: Journals

- Journal of Machine Learning Research
  - www.jmlr.org
- Machine Learning http://jmlr.csail.mit.edu/papers/
- Annals of Statistics
- Journal of the American Statistical Association
  - http://www.jstatsoft.org/

Neural Networks

- Neural Computation also Neural Networks
- IEEE Transactions
  - Neural Networks
  - Pattern Analysis and Machine Intelligence

Extra Topics To Explore

- New Packages
- Extra Functions

3000 + Packages

- List of Tasks: http://cran.r-project.org/web/views
- Keyword Search for Tasks or Functions: http://rseek.org
- Search, Tag, and Review Packages: http://crantastic.org/
- Package Downloads Over Time – App
  - https://dgrtwo.shinyapps.io/cranview/

Packages

- List of All Packages
  - http://cran.r-project.org/web/packages/available_packages_by_name.html
- Popular Packages
  - http://www.r-statistics.com/2013/06/top-100-r-packages-for-2013-jan-may/
- Task Views
  - http://cran.r-project.org/web/views/ReproducibleResearch.html
- R Studio Quick List
  - https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages

Package: caret

- **caret package** (short for classication and regression training)
  - contains functions to streamline the model training process
  - Evaluate effect of model tuning parameters on performance
  - Choose “optimal” model across these parameters
  - Estimate model performance from training set
- **Help Pages**
  - http://caret.r-forge.r-project.org/
- **Vignettes in package caret**
  - A Short Introduction to the caret Package - PDF source R code
- **Manual**
  - *cran.r-project.org/web/packages/**caret**/**caret**.**pdf***
- **Introduction**
  - *cran.r-project.org/web/packages/**caret**/vignettes/**caret**.**pdf***
- **Tutorial**
  - *www.edii.uclm.es/.../user_**caret**_2up.**pd**...*
- A Short Introduction to the caret Package - PDF source R code

Caret Functions

- createDataPartition()
  - A series of test/training partitions
- createResample()
  - creates one or more bootstrap samples.
- createFolds()
  - splits the data into k groups
- createTimeSlices()
  - creates cross-validation sample information to be used with time series data.

plyr: Tools for splitting, applying and combining data

plyr is a set of tools that solves a common set of problems: you need to break a big problem down into manageable pieces, operate on each pieces and then put all the pieces back together. For example, you might want to fit a model to each spatial location or time point in your study, summarise data by panels or collapse high-dimensional arrays to simpler summary statistics.

```
ddply(Orange, .(Tree), summarize, COVARIANCE = cov(age, circumference),
CORRELATION = cor(age, circumference))
```

http://cran.r-project.org/web/packages/plyr/

zoo: Time Series

- An S3 class with methods for totally ordered indexed observations. It is particularly aimed at irregular time series of numeric vectors/matrices and factors. zoo's key design goals are independence of a particular index/date/time class and consistency with ts and base R by providing methods to extend standard generics.
- http://cran.r-project.org/web/packages/zoo/

tseries and timeSeries: Time series analysis and computational finance

- Time Series manipulations for
  - Exonometircs
  - Environmetrics
  - Finance
  - Time Series
  - Web Technologies
- http://cran.r-project.org/web/packages/tseries/
- http://cran.r-project.org/web/packages/timeSeries/

Package: “stringr”

Description stringr is a set of simple wrappers that make R's string functions more consistent, simpler and easier to use. It does this by ensuring that: function and argument names (and positions) are consistent, all functions deal with NA's and zero length character appropriately, and the output data structures from each function matches the input data structures of other functions

http://cran.r-project.org/web/packages/stringr/stringr.pdf

Text Mining Packages

- tm
  - Vignettes:Extensions - PDF source R code
  - Introduction to the tm Package - PDF source R code
- Rstem
- openNLP
- Isa

API's

- twitteR
  - Download twitter data using R
    - http://www.r-bloggers.com/getting-started-with-twitter-in-r/
- Rfacebook
  - Download facebook data using R
    - http://thinktostart.com/analyzing-facebook-with-r/
- Rlinkedin
  - https://github.com/mpiccirilli/Rlinkedin
- httr: Talk to web API's from R
  - http://cran.r-project.org/web/packages/httr/vignettes/quickstart.html
- jsonlite: converts a data frame to JSON formatted data

R Parallel Computing

- snow: http://mran.revolutionanalytics.com/packages/info/?snow
- parallel: http://www.inside-r.org/r-doc/parallel
- pnmath: http://homepage.stat.uiowa.edu/~luke/R/experimental/

sparkR

- Interactive R programs at Scale
  - https://www.youtube.com/watch?v=CUX1SG9zTkU&index=1&list=PL-x35fyliRwiuc6qy9z2erka2VX8LY53x
  - http://blog.revolutionanalytics.com/2015/01/a-first-look-at-spark.html

Package: H2O

- Big Data Analytics Package
  - Includes many model building tools
- The Open Source In-Memory, Prediction Engine for Big Data Science
- More
- Package Dependences include:
  - RCurls, bitops, rjson, statmod, and tools

More Packages

- XML
- RMySql : SQL
- xml2 : work with html and xml
    - install.packages("xml2")
- RapidXML : work with excel
    - install.packages("readxl")
    - http://rapidxml.sourceforge.net/
- readr : easy access to many types of tabular data

readr Package

- readr : easy access to many types of tabular data
  - read_lines() works the same way as readLines(), but is a lot faster.
  - read_file() reads a complete file into a string.
  - type_convert() attempts to coerce all character columns to their appropriate type. This is useful if you need to do some manual munging (e.g. with regular expressions) to turn strings into numbers. It uses the same rules as the read_* functions.
  - write_csv() writes a data frame out to a csv file. It's quite a bit faster than write.csv() and it never writes row.names. It also escapes " embedded in strings in a way that read_csv() can read.

haven

- Haven makes it easy to read data from SAS, SPSS and Stata. Haven has the same goal as the foreign package, but it:
  - Can read binary SAS7BDAT files.
  - Can read Stata13 files.
  - Always returns a data frame.
- Haven is a binding to the excellent ReadStat C library

igraph: Network analysis and visualization

- Routines for simple graphs and network analysis. igraph can handle large graphs very well and provides functions for generating random and regular graphs, graph visualization, centrality indices and much more.
- http://cran.r-project.org/web/packages/igraph/

Iterators and Object Orientation

- iterators: Support for iterators, which allow a programmer to traverse through all the elements of a vector, list, or other collection of data.
  - http://cran.r-project.org/web/packages/iterators/
- proto: Prototype object-based programming
- sp: classes and methods for spatial data

Package Iterators

- Vignettes in package **iterators**
- iterators Manual - PDF source R code
- Writing Custom Iterators - PDF source R code

Dates and Times

- chron
  - Chronological objects which can handle dates and times
- timeDate
  - Rmetrics – Chronological and Calendar Objects

Visualization

- scales: Scale functions for graphics
- labeling: Axis Labeling
- maps: Draw Geographical Maps
- maptools: Tools for reading and handling spatial objects
- rgdal: Bindings for the Geospatial Data Abstraction Library
- vcd: Visualizing Categorical Data

Package: ggmap

- Install from github
  - install_github("dkahle/ggmap")

```
downtown <- subset(crime, -95.39681 <= lon &
                    lon <= -95.34188 & 29.73631 <= lat &
                    lat <= 29.78400 )

qmplot(lon, lat, data = downtown,
                    maptype = "toner-background", color = I("blue"))
```

Package: rworldmap

- mapping of country level and gridded user datasets
  - joins modern world maps with visualization options.
  - Country borders
- Ukraine Map using rworldmap code:
  - #install.packages('rworldmap', dep = TRUE)
  - library(rworldmap)
  - mapUkr <- get_map(location = 'Ukraine', zoom = 5)
  - ggmap(mapUkr)

More Visualization Packages

- lattice
- iplot
- ggplot2
  - http://wiki.stdout.org/rcookbook/Graphs/
- GGobi
- rggobi
- hexbin
- rCharts

Package rgl

- Package rgl: real-time 3D engine written in C++
  - http://rgl.neoscientists.org/about.shtml
  - http://cran.r-project.org/src/contrib/Descriptions/rgl.html

Interactive Graphics

- GGobi Home Page
  - http://www.ggobi.org/
- GGobi Manual
  - http://www.ggobi.org/docs/manual.pdf
- rggobi Introduction
  - www.ggobi.org/rggobi/introduction.pdf
- rggobi Manual
  - http://cran.r-project.org/web/packages/rggobi/rggobi.pdf

Model Building Packages

- rattle
- Rweka
- Rcmdr
- leaps
- forcast
  - http://robjhyndman.com/hyndsight/revolutionr2013/

Rcmdr: R Commander

- A platform-independent basic-statistics GUI (graphical user interface) for R, based on the tcltk package.
- http://cran.r-project.org/web/packages/Rcmdr/

Extra Topics To Explore

- Timing Functions
  - More Efficient Code
- apply and aggregate Functions
- Caching

Timing Code

- proc.time() returns the current time.
- system.time() times the evaluation of expression
- R has a profiler; records which functions are being run, many times per second. Rprof(filename) turns on the profiler,
- Rprof(NULL) turns it off.
- summaryRprof(filename) reports how much time was spent in each function.

Available Models in R

- linear models (lm)
- generalized linear models (glm)
- generalized additive models (gam)
- linear mixed effects models (lme)
- quantile regression (qr)
- vector general additive models (vgam)
- lasso, ridge, and elastic net models (glmnet)
- non-linear models (nlm)
- Boosted Ensemble (gbm)
- Support Vector Machine (svm)
- linear mixed effects models (nlmer)
- linear discriminant analysis (lda)
- quadratic discriminate analysis (qda)
- trees (tree) (rpart)
- random forests (randomForrest)
- support vector machines (svm)
- neural networks (nnet)
- k-nearest neighbors (knn)
- Naïve Bayes (NaiveBayes)

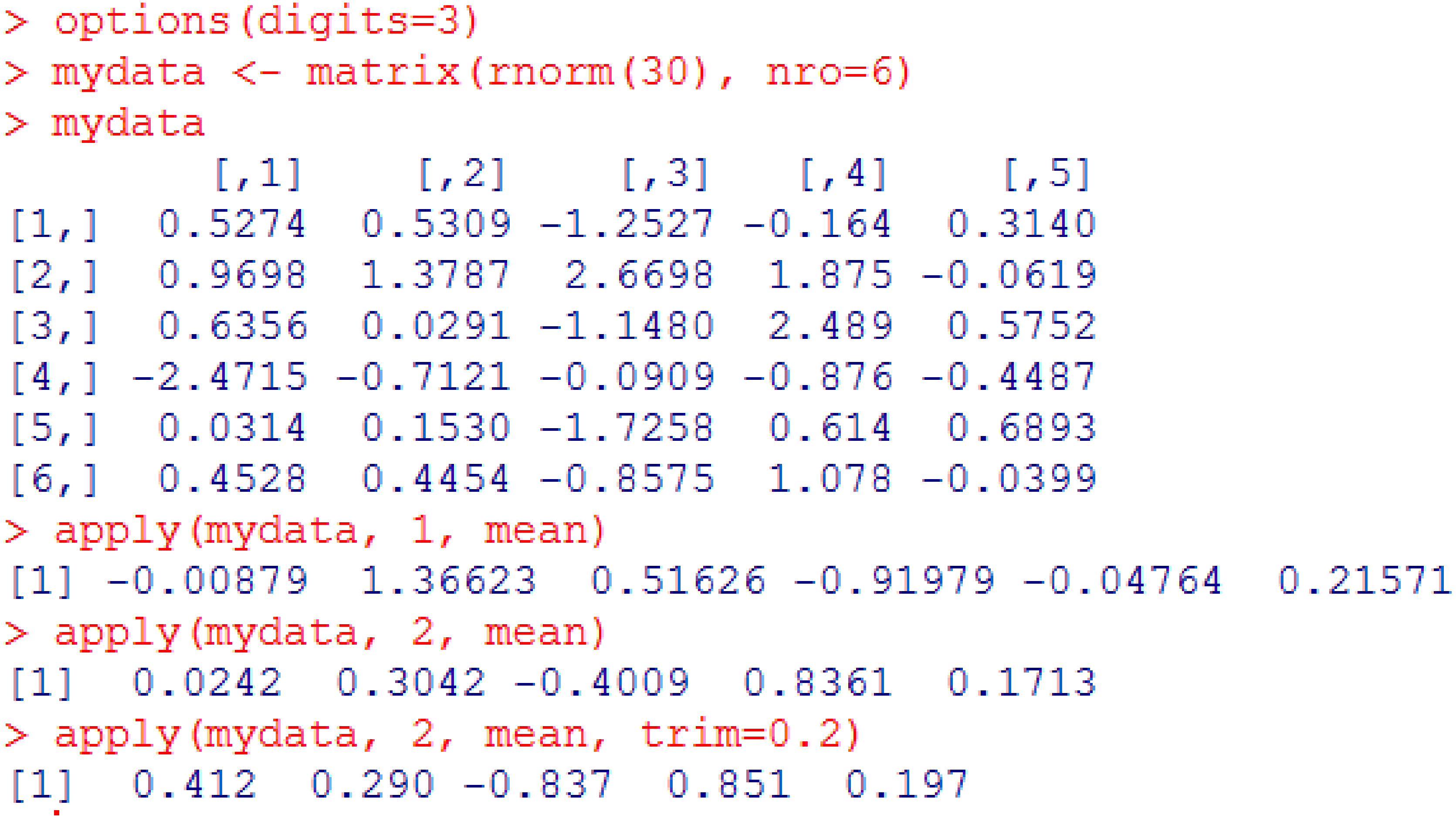
Apply Functions

- base::apply Apply Functions Over Array Margins
- base::by Apply a Function to a Data Frame Split by Factors
- base::eapply Apply a Function Over Values in an Environment
- base::lapply Apply a Function over a List or Vector
- base::mapply Apply a Function to Multiple List or Vector Arguments
- base::rapply Recursively Apply a Function to a List
- base::tapply Apply a Function Over a Ragged Array

Applying functions to matrices and data frames (2)

- Description: “Returns a vector or array or list of values obtained by applying a function to margins of an array or matrix.”
- apply(*x*, *MARGIN*, *FUN*, ...)
  - x is an matrix or data frame
  - *MARGIN* = 1 (rows) or 2 (column)
  - *FUN* is a function
  - … are optional parameters passed to *FUN*

Applying functions to matrices and data frames (3)



```
> options(digits=3)
> mydata <- matrix(rnorm(30), nro=6)
> mydata
        [,1]    [,2]    [,3]   [,4]    [,5]
[1,]  0.5274  0.5309 -1.2527 -0.164  0.3140
[2,]  0.9698  1.3787  2.6698  1.875 -0.0619
[3,]  0.6356  0.0291 -1.1480  2.489  0.5752
[4,] -2.4715 -0.7121 -0.0909 -0.876 -0.4487
[5,]  0.0314  0.1530 -1.7258  0.614  0.6893
[6,]  0.4528  0.4454 -0.8575  1.078 -0.0399
> apply(mydata, 1, mean)
[1] -0.00879  1.36623  0.51626 -0.91979 -0.04764  0.21571
> apply(mydata, 2, mean)
[1]  0.0242  0.3042 -0.4009  0.8361  0.1713
> apply(mydata, 2, mean, trim=0.2)
[1]  0.412  0.290 -0.837  0.851  0.197
```

apply Example
(MyFirstRLesson.r)

```
A
#      [,1] [,2]
# [1,]    2    5
# [2,]    1    3
```

```
apply(A,1,sum) # sum of rows
#[1] 7 4
apply(A,2,mean) # mean of columns
#[1] 1.5 4.0
apply(A,1,function(x) min(x))
# [1] 2 1
```

Aggregating data

- aggregate(*x*, *by*, *FUN*)
  - *x* is the data object to be collapsed
  - *by* is a list of variables that will be cross to form new observations
  - *FUN* is a scalar function used to calculate summary statistics that will make up the new observation values

Apply Functions

- base::apply Apply Functions Over Array Margins
- base::by Apply a Function to a Data Frame Split by Factors
- base::eapply Apply a Function Over Values in an Environment
- base::lapply Apply a Function over a List or Vector
- base::mapply Apply a Function to Multiple List or Vector Arguments
- base::rapply Recursively Apply a Function to a List
- base::tapply Apply a Function Over a Ragged Array

apply Function

- apply(A, MARGIN, FUN, ...)
  - A an array, including a matrix
  - MARGIN for matrix
    - 1 indicates rows
    - 2 indicates columns
    - c(1,2) indicates rows and columns
  - FUN indicates a function
  - … are optional arguments to FUN

Aggregate example

```
options(digits=3)
attach(mtcars)
aggdata <-aggregate(mtcars, by=list(cyl,gear), FUN=mean,
   na.rm=TRUE)
detach(mtcars)
aggdata
```

```
  Group.1 Group.2  mpg cyl disp  hp drat   wt qsec  vs   am gear carb
1       4       3 21.5   4  120  97 3.70 2.46 20.0 1.0 0.00    3 1.00
2       6       3 19.8   6  242 108 2.92 3.34 19.8 1.0 0.00    3 1.00
3       8       3 15.1   8  358 194 3.12 4.10 17.1 0.0 0.00    3 3.08
4       4       4 26.9   4  103  76 4.11 2.38 19.6 1.0 0.75    4 1.50
5       6       4 19.8   6  164 116 3.91 3.09 17.7 0.5 0.50    4 4.00
6       4       5 28.2   4  108 102 4.10 1.83 16.8 0.5 1.00    5 2.00
7       6       5 19.7   6  145 175 3.62 2.77 15.5 0.0 1.00    5 6.00
8       8       5 15.4   8  326 300 3.88 3.37 14.6 0.0 1.00    5 6.00
```

Melt data

```
md <- melt(mydata, id=c("id", "time"))
```

mydata

ID	Time	X1	X2
1	1	5	6
1	2	3	5
2	1	6	1
2	2	2	4

→

ID	Time	Variable	Value
1	1	X1	5
1	2	X1	3
2	1	X1	6
2	2	X1	2
1	1	X2	6
1	2	X2	5
2	1	X2	1
2	2	X2	4

Cast data without Aggregation

md <− melt(mydata, id=c("id", "time"))

ID	Time	Variable	Value
1	1	X1	5
1	2	X1	3
2	1	X1	6
2	2	X1	2
1	1	X2	6
1	2	X2	5
2	1	X2	1
2	2	X2	4

cast(md, id + time ~ variable)

ID	Time	X1	X2
1	1	5	6
1	2	3	5
2	1	6	1
2	2	2	4

cast(md, id + variable ~ time)

ID	Variable	Time1	Time 2
1	X1	5	3
1	X2	6	5
2	X1	6	2
2	X2	1	4

cast(md, id ~ variable + time)

ID	X1 Time1	X1 Time2	X2 Time1	X2 Time2
1	5	3	6	5
2	6	2	1	4

Cast with Aggregation

```
md <– melt(mydata, id=c("id", "time"))
```

ID	Time	Variable	Value
1	1	X1	5
1	2	X1	3
2	1	X1	6
2	2	X1	2
1	1	X2	6
1	2	X2	5
2	1	X2	1
2	2	X2	4

```
cast(md, id ~ variable, mean)
```

ID	X1	X2
1	4	5.5
2	4	2.5

```
cast(md, time ~ variable, mean)
```

Time	X1	X2
1	5.5	3.5
2	2.5	4.5

```
cast(md, id ~ time, mean)
```

ID	Time1	Time2
1	5.5	4
2	3.5	3

grep

- SCCCoal<-SCC$SCC[grepl("Coal",SCC$EI.Sector)]
- coalEISectors <- grep("[Cc]oal",SCC$EI.Sector)

More Examples

- Sampling
- Working with Graphs

More to Explore

- help(plotmath)
- help(image)
- help(grep)

Caching

- For Calculations Which take a long time Caching is Recommended
- http://userprimary.net/papers/weaver-paper-falcon.pdf

- leaps