Quick data cooking using R

Daisuke Takahashi

Lunch Seminar, Jan 27, 2016

What is R?

R is a language and environment for statistical computing and graphics. *It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.* *www.r-project.org*

What can R do?

Almost everything you need!

- *an **effective data handling** and storage facility,*
- *a suite of operators for **calculations on arrays**, in particular **matrices**,*
- *a large, coherent, integrated collection of intermediate tools for **data analysis**,*
- ***graphical facilities** for data analysis and display either on-screen or on hardcopy, and*
- *a well-developed, **simple and effective programming language** which includes conditionals, loops, user-defined recursive functions and input and output facilities.*

www.r-project.org

Why R?

- Free
  - Free of charge
    You can install R as many as you want
  - Free software
    You can investigate R's source code to check every single piece of R

Why R?

- Domain specific (↔ general purpose)
  - Pros:
    You can focus on what you want to do
    Everything for basic analysis are already there
  - Cons:
    Hard to solve a problem outside of its domain
    Text mining, machine learning, GUI apps, …
    it is better to use something different

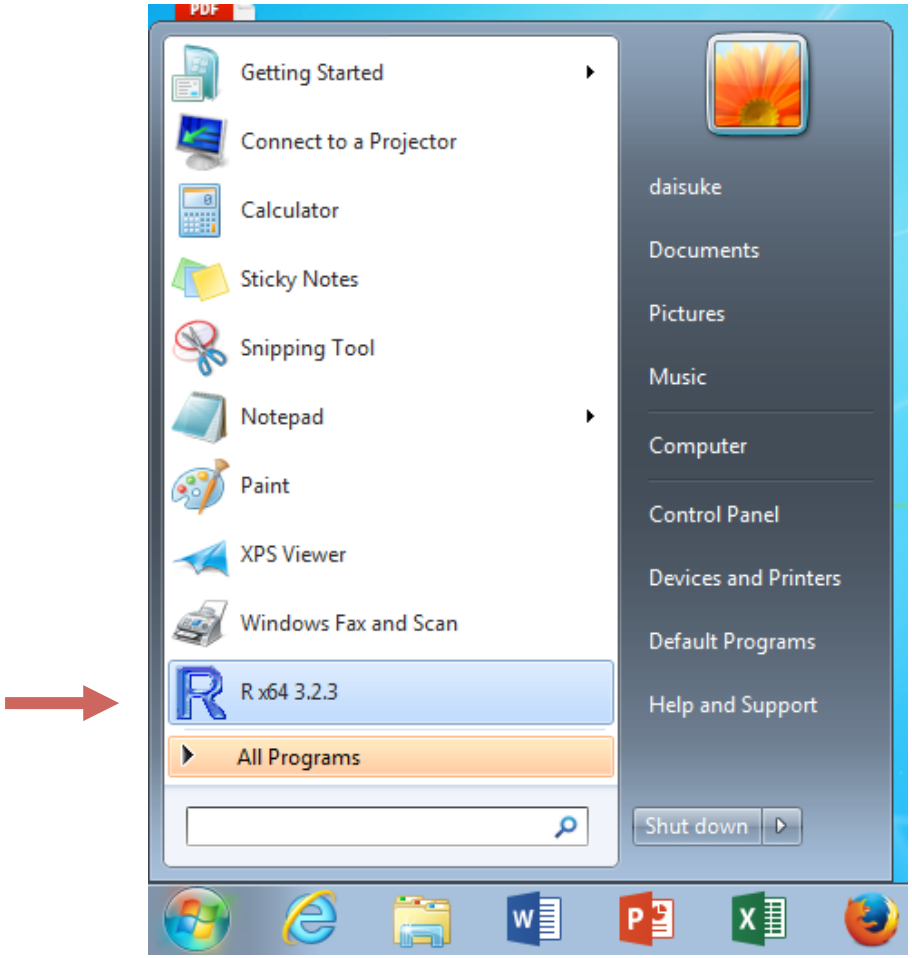
Why R?

- Extensible
  - R is a programing language
    You can write your customized routine
  - Packages
    Many statistical/numerical packages are available

Why R?

- Good documentation
  - Online help
    Easily accessible from R environment
  - More documents on www
    Also many tutorials and forums

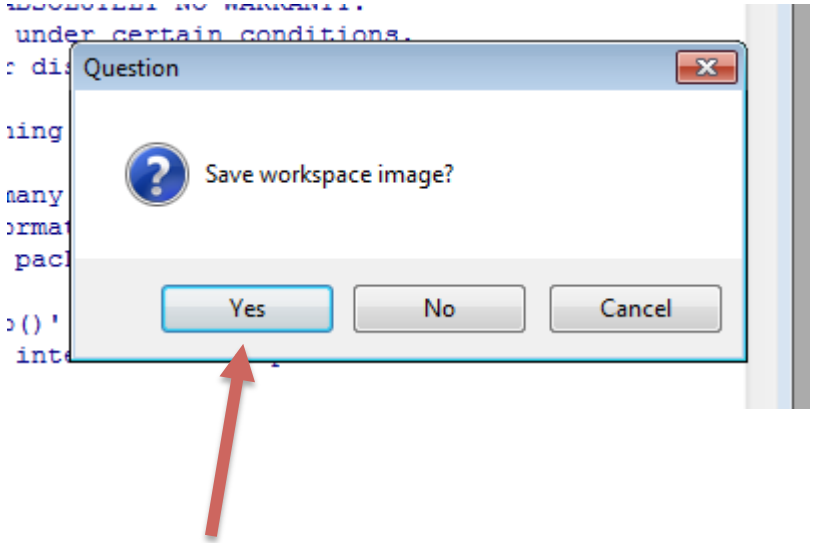
Launch R



Or type R in a terminal window

Quit R

Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. **Type 'q()' to quit R.**

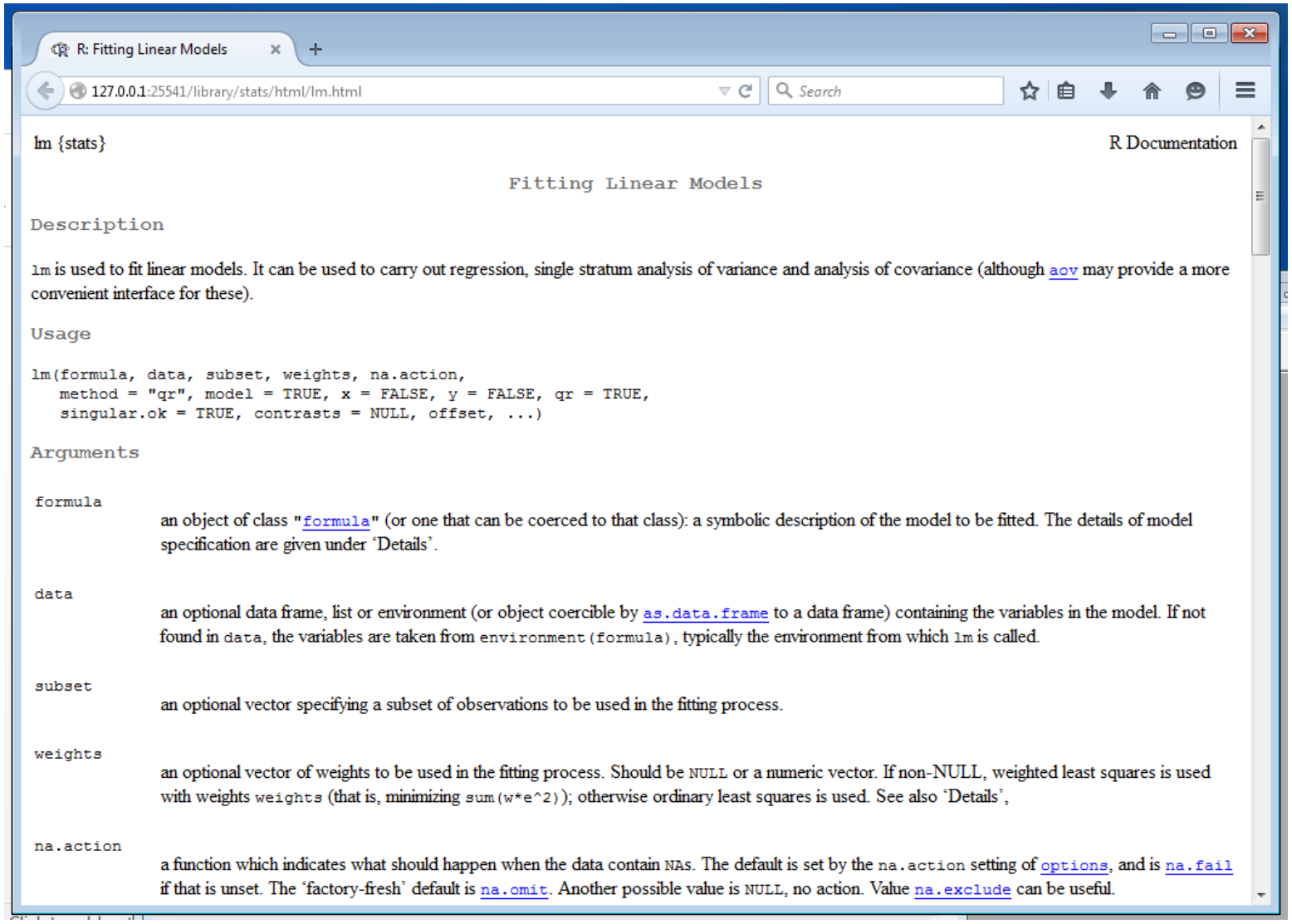


You can save a current environment for your next analysis!

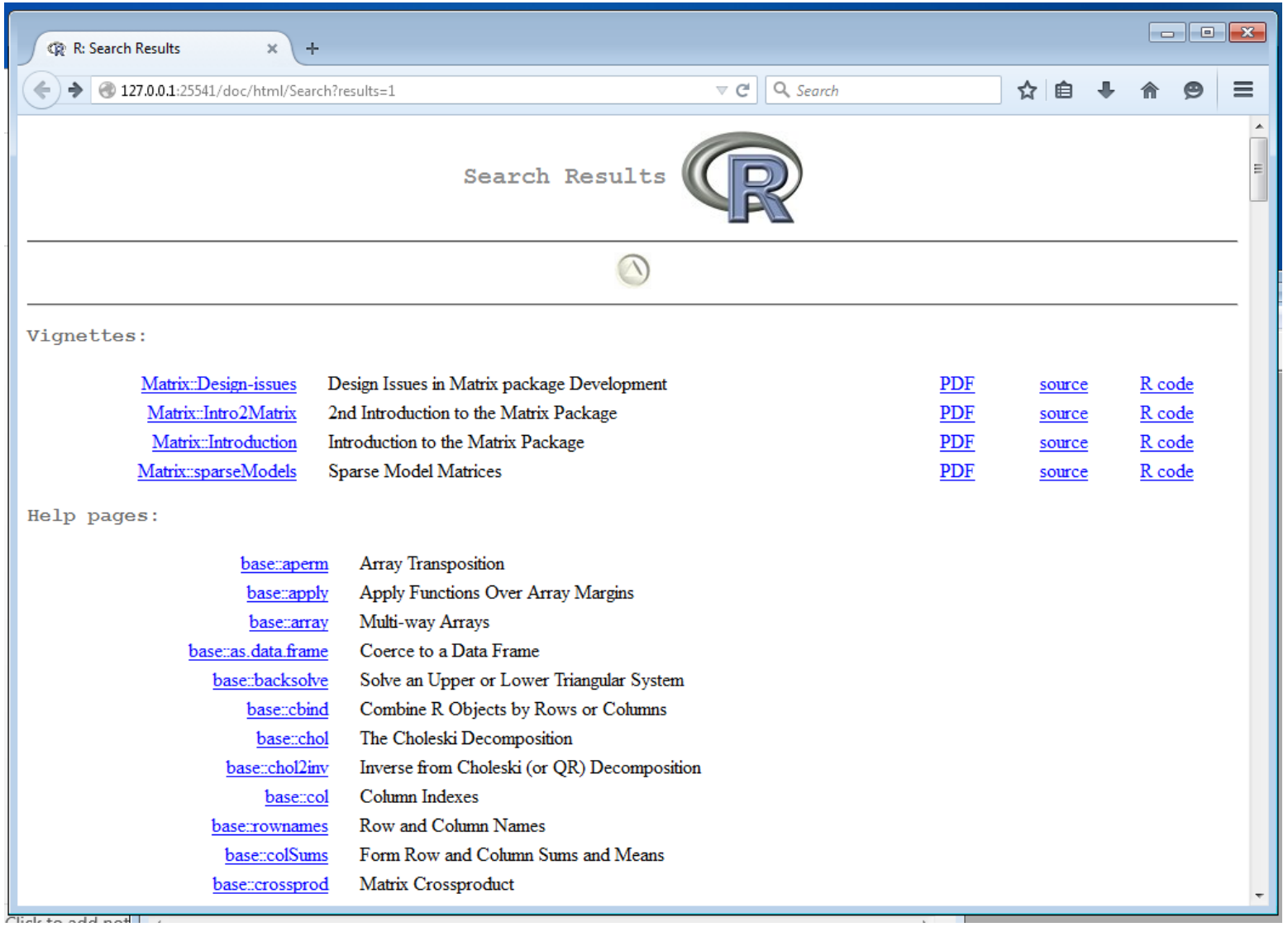
Online help

- `> ?function-name`
  equivalent to `help(function-name)`
- `> ??keyword`
  equivalent to `help.search(keyword)`

?lm (= help(lm))


R: Fitting Linear Models
127.0.0.1:25541/library/stats/html/lm.html
Search
lm {stats}
R Documentation
Fitting Linear Models
Description
lm is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although aov may provide a more convenient interface for these).
Usage
lm(formula, data, subset, weights, na.action,
method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,
singular.ok = TRUE, contrasts = NULL, offset, ...)
Arguments
formula
an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data
an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
subset
an optional vector specifying a subset of observations to be used in the fitting process.
weights
an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. If non-NULL, weighted least squares is used with weights weights (that is, minimizing sum(w*e^2)); otherwise ordinary least squares is used. See also 'Details',
na.action
a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset. The 'factory-fresh' default is na.omit. Another possible value is NULL, no action. Value na.exclude can be useful.

??matrix (= help.search(matrix))



Basic flow of an analysis

1. Read data from a file
2. Extract data
3. Process data
4. Plot results
5. Save results

Read data

- Read an RData
  - `load(`*`filename`*`)`

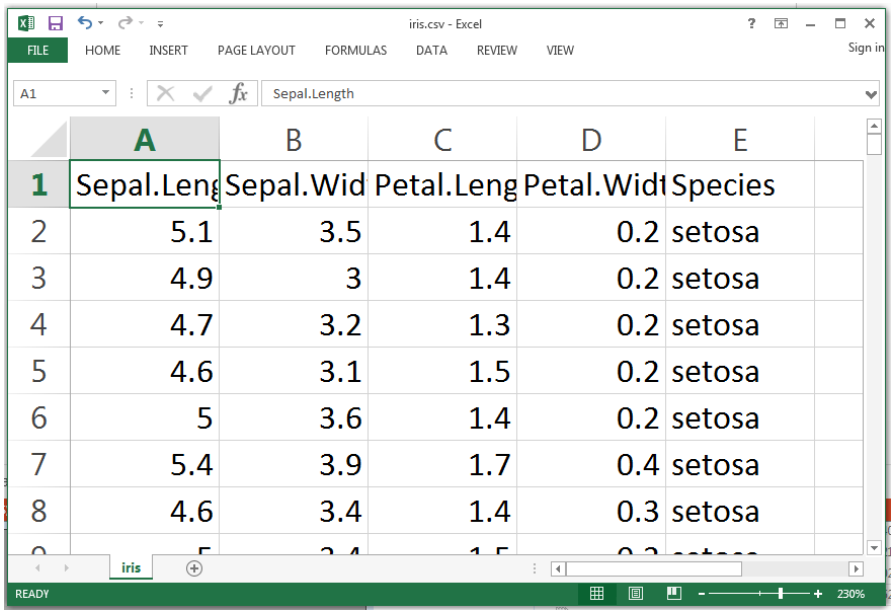
  Import data from other R session
- Read a text file containing a table
  - `read.table(`*`filename`*`)`
  - `read.csv(`*`filename`*`)`

  Import data from external software

Read data from excel

data.csv

Export to csv




```
> tbl <- read.csv("data.csv")
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.4         0.2     setosa
...
```

Extract data elements

`tbl[`*`row`*`,`*`column`*`]`

row and *column* can be,

- an integer — 1 (first element), 2, …
- an integer vector — c(1,3,4), c(3,5), …
- a range — 1:3, 2:4, …
- a name — “Sepal.Length”, “Species”, …
- a logical vector — tbl[,1] > 7, …
- or not specified

?"[" for detail

Extract data elements

```
> tbl
  Treatment  Height  Width
1      TRUE     102     23
2      TRUE      32      3
3      TRUE      NA     NA
4      TRUE     120     30
5     FALSE      43     20
6     FALSE     100     -2
7     FALSE      20      2
```

by a range

```
> tbl[1:3,]
  Treatment  Height  Width
1      TRUE     102     23
2      TRUE      32      3
3      TRUE      NA     NA
```

by an integer index

```
> tbl[,2]
[1] 102  32  NA  120  43
100  20
```

by a column name

```
> tbl[,"Height"]
[1] 102  32  NA  120  43
100  20
```

Clean erroneous entries

A table with erroneous values

```
> tbl
  Treatment  Height  Width
1      TRUE     102     23
2      TRUE      32      3
3      TRUE      NA     NA
4      TRUE     120     30
5     FALSE      43     20
6     FALSE     100     -2
7     FALSE      20      2
```

← NA (not available) values

← Negative width

Incorrect data must be removed before analysis!

Clean erroneous entries

Remove all entries with NA values

```
> tbl
  Treatment Height Width
1      TRUE    102    23
2      TRUE     32     3
3      TRUE     NA    NA
4      TRUE    120    30
5     FALSE     43    20
6     FALSE    100    -2
7     FALSE     20     2
```

```
> tbl2 <- na.omit(tbl)
  Treatment Height Width
1      TRUE    102    23
2      TRUE     32     3
4      TRUE    120    30
5     FALSE     43    20
6     FALSE    100    -2
7     FALSE     20     2
```

Clean erroneous entries

Remove row with negative “width”

```
> tbl2
  Treatment  Height  Width
1      TRUE     102     23
2      TRUE      32      3
4      TRUE     120     30
5     FALSE      43     20
6     FALSE     100     -2
7     FALSE      20      2
```

```
> tbl3 <- tbl2[tbl2[,"Width"] > 0,]
  Treatment  Height  Width
1      TRUE     102     23
2      TRUE      32      3
4      TRUE     120     30
5     FALSE      43     20
7     FALSE      20      2
```

Functions for data processing

`mean(x, na.rm), sd(x, na.rm)`	Basic statistics
`t.test(x, y), wilcox.test(x, y), kruskal.test(x, y)`	Statistical tests
`approx(x, y, xout), approxfun(x, y), spline(x, y, xout), splinefun(x, y)`	Interpolation
`lm(formula, data), glm(formula, data)`	Model fitting
`aggregate(x, by, FUN, …), by(data, INDICES, FUN, …), tapply(X, INDEX, FUN, …)`	Summarize by groups

Process data

Vector arithmetic

```
> ratio <- iris[,"Sepal.Length"]/iris[,"Sepal.Width"]
```

Calculate average value

```
> mean(ratio)
[1] 1.953681
```

Calculate average value for each species

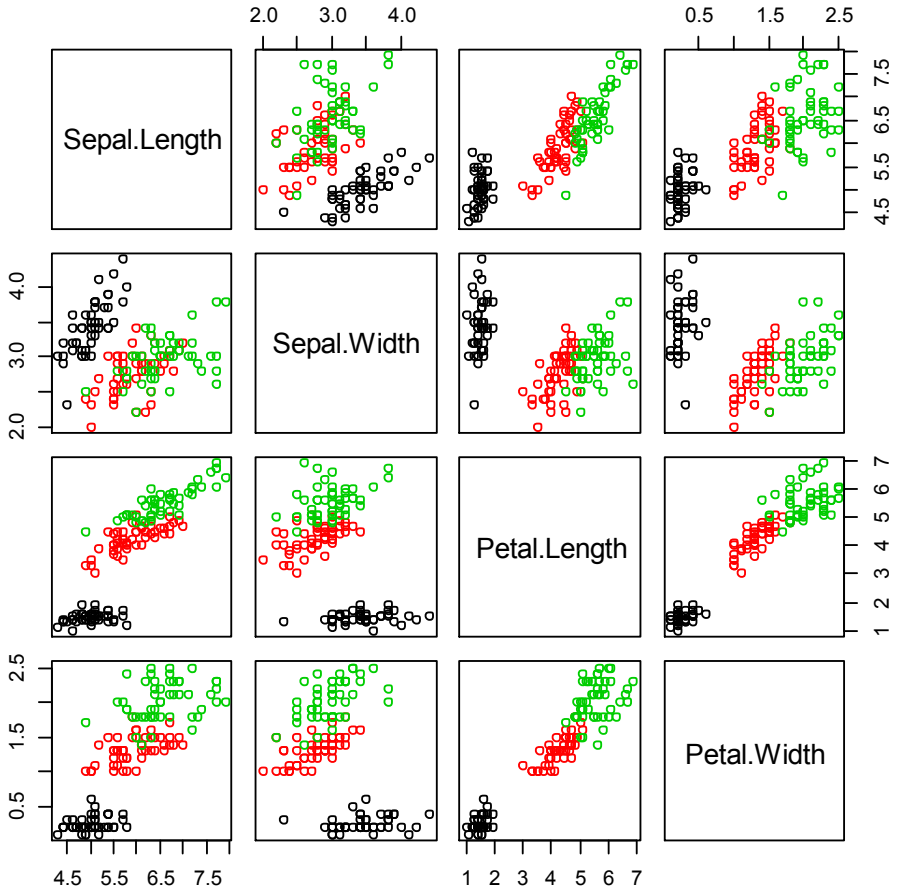
```
> by(ratio, iris[,"Species"], mean)
iris[,"Species"]: setosa
[1] 1.470188
------------------------------------------------------------
iris[,"Species"]: versicolor
[1] 2.160402
------------------------------------------------------------
iris[,"Species"]: virginica
[1] 2.230453
```

Plotting

- Plot by high-level plotting commands `plot()`, `hist()`, `boxplot()`, `image()`,…
  - Lots of options to customize the plot
- Save a plot
  1. From the GUI menu
  2. Redirect to a “graphic device” `pdf()`, `png()`, `postscript()`, …
     - don’s forget to close the device by `dev.off()`

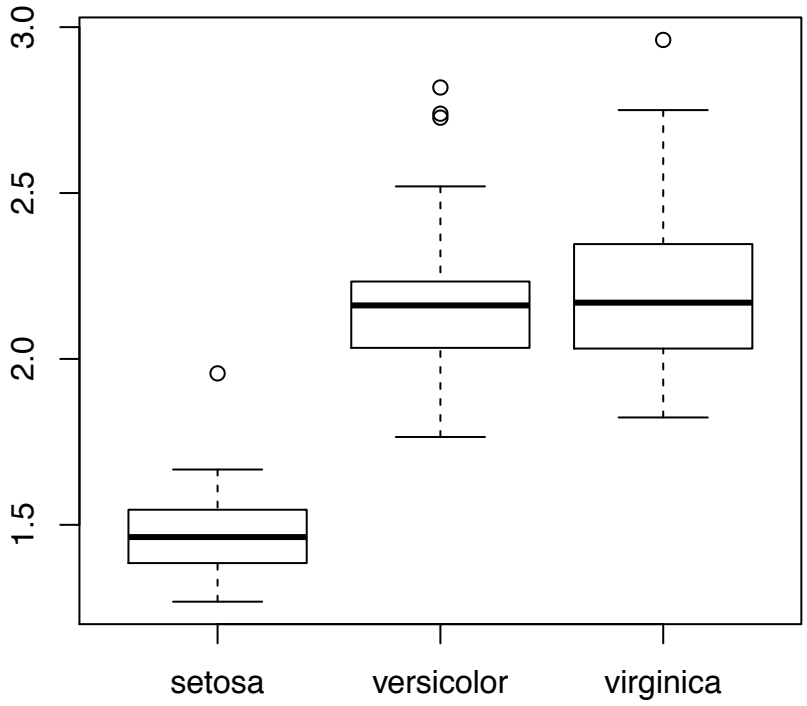
Plotting

```
> plot(iris[,1:4], col=iris["Species"])
```



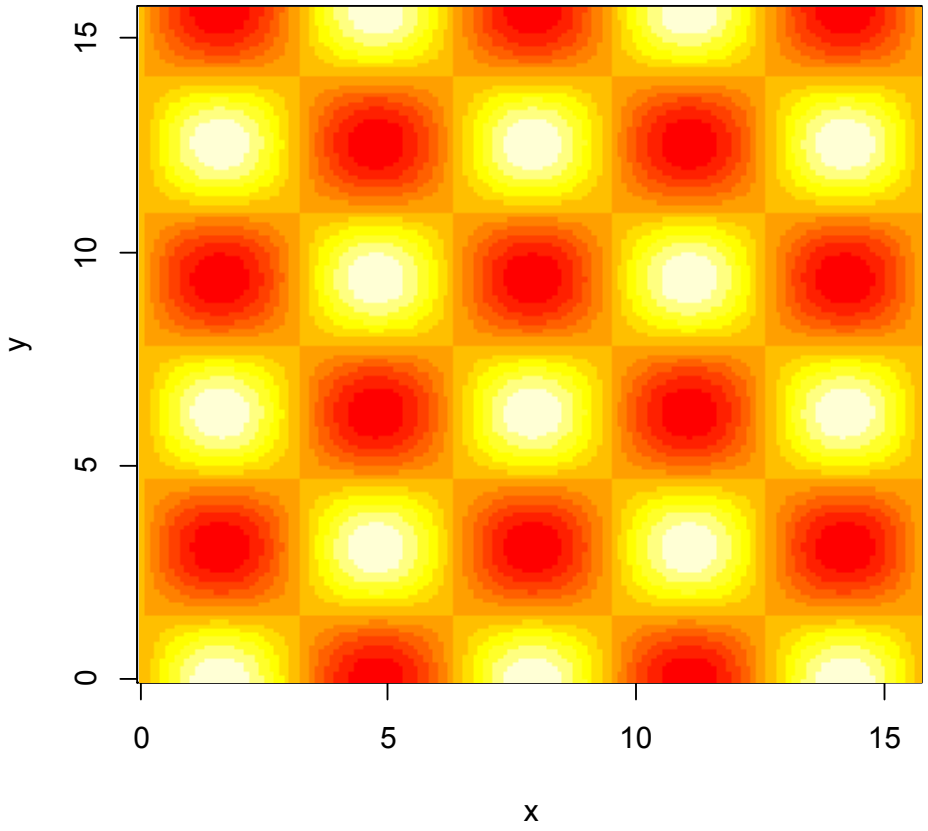
Plotting

```
> boxplot(Sepal.Length/Sepal.Width ~ Species,
          data=iris)
```



Plotting

```
> x <- y <- seq(0, 5*pi, 0.01)
> m <- outer(sin(x), cos(y), "*")
> image(x, y, m)
```



Save results

- Save as an RData
  - `save(…, file=`*`filename`*`)`

  Binary format not compatible with other software

  Compatible with other R
- Save as an external format
  - write.table(x, file=filename)
  - write.csv(…)

  To export results to other software

Save results

```
> ratio
[1] 1.457143 1.633333 1.468750 1.483871 1.388889 1.384615 ...
```

Save "ratio" as a space-separated text

```
> write.table(ratio, file="ratio.txt")
```

Save "ratio" as a csv file

```
> write.csv(ratio, file="ratio.csv")
```

Save as an R-specific binary file

```
> save(ratio, file="ratio.RData")
```

Extend R

CRAN is a network of ftp and web servers *around the world that store identical, up-to-date, versions* ***of code and documentation for R.*** *Please use the CRAN mirror nearest to you to minimize network load.*

cran.r-project.org

Package management

```
> install.packages(package-name)

messages from download and installation process

> library(package-name)
> now you can use the package
```

Summary

1. Read data from a file — `Load(), read.table(), read.csv()`
2. Extract data
3. Process data — `na.omit(), mean(), by(),…`
4. Plot results — `plot(), boxplot(), image(),…`
5. Save results — `save(), write.table(), write.csv()`

`install.packages(), library()`

? and ?? to lookup manuals

Useful links

- Comprehensive R Archive Network (CRAN)
  - https://cran.r-project.org/
  - https://cran.r-project.org/doc/manuals/R-intro.html (official introduction)
- Quick-R (many tutorials)
  - http://www.statmethods.net/
- R-bloggers (also many tutorials)
  - http://www.r-bloggers.com/
- and Google
  - https://www.google.com/search?q=R+tutorial