Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D-BSSE
Department of Biosystems Science and Engineering

Statistical Models in Computational Biology

Tutorial: A Primer for programming in R

fabian.schmich@bsse.ethz.ch

Overview

- Introduction
- R Basics
  - R Objects: Vector, Matrix, Data Frame and List
  - Function Definition
  - File I/O
- Conditional and Repetitive Execution
  - `if` and `else`
  - Loops and subsetting
  - *`apply()` functions
- Graphical Procedures
- External Packages and Misc. Functions
- Resources

Introduction: What is R?

The Swiss Army Knife for Data Analysis

Introduction: What is R?

- Interpreted programming language based on S
  - Individual statements compiled to machine code
  - PRO: high flexibility; CON: slow
  - Interface to C, C++, FORTRAN
- Open source
  - Freely available for Linux, OSX, Win (http://r-project.org)
  - Large and active community
- Wide variety of statistical and graphical functions
  - Modeling, statistical tests, classification, clustering, …
  - Easy creation of publication-ready plots
  - > 3500 packages providing additional functionality

ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D-BSSE
Department of Biosystems Science and Engineering

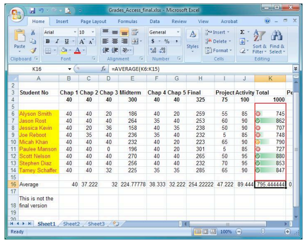
Introduction: Typical Workflow

$$\begin{tikzcd}
 & \text{Data Manipulation: parse, edit, reshape} \arrow[dl] & \\
\text{Data Modeling: model, analyse} \arrow[rr] & & \text{Data Visualisation: plot} \arrow[ul]
\end{tikzcd}$$

Data Manipulation
parse, edit, reshape

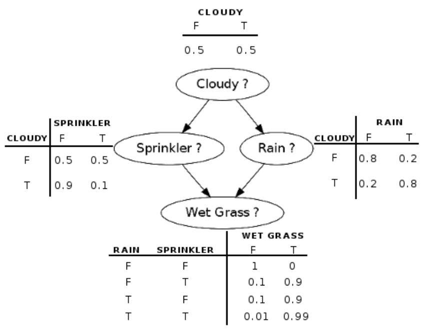
Data Modeling
model, analyse

Data Visualisation
plot



K16 =AVERAGE(K6:K15)

	A	B	C	D	E	F	G	H	I	J	K
2											
3	Student No	Chap 1	Chap 2	Chap 3	Midterm	Chap 4	Chap 5	Final	Project	Activity	Total
4		40	40	40	300	40	40	325	75	100	1000
5											
6	Alyson Smith	40	40	20	186	40	20	259	55	85	745
7	Jason Rost	40	40	40	264	35	40	253	60	90	862
8	Jessica Kevin	40	20	36	158	40	35	238	50	90	707
9	Joe Reboot	40	35	40	236	35	40	232	5	85	748
10	Micah Khan	40	40	40	232	40	20	223	65	90	790
11	Paulee Manson	40	40	0	196	40	20	301	5	85	727
12	Scott Nelson	40	40	40	270	40	40	265	50	95	880
13	Stephen Diaz	40	40	40	256	40	40	232	70	95	853
14	Tamey Schaffer	40	40	32	225	35	35	285	65	90	847
15											
16	Average	40	37.222	32	224.77778	38.333	32.222	254.22222	47.222	89.444	795.444444
17											
18	This is not the final version										
19											

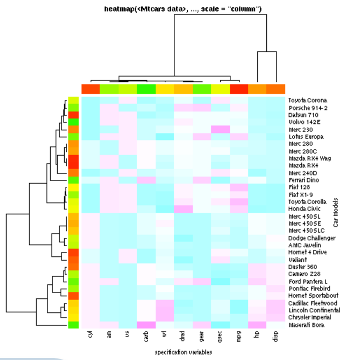


CLOUDY F	CLOUDY T
0.5	0.5

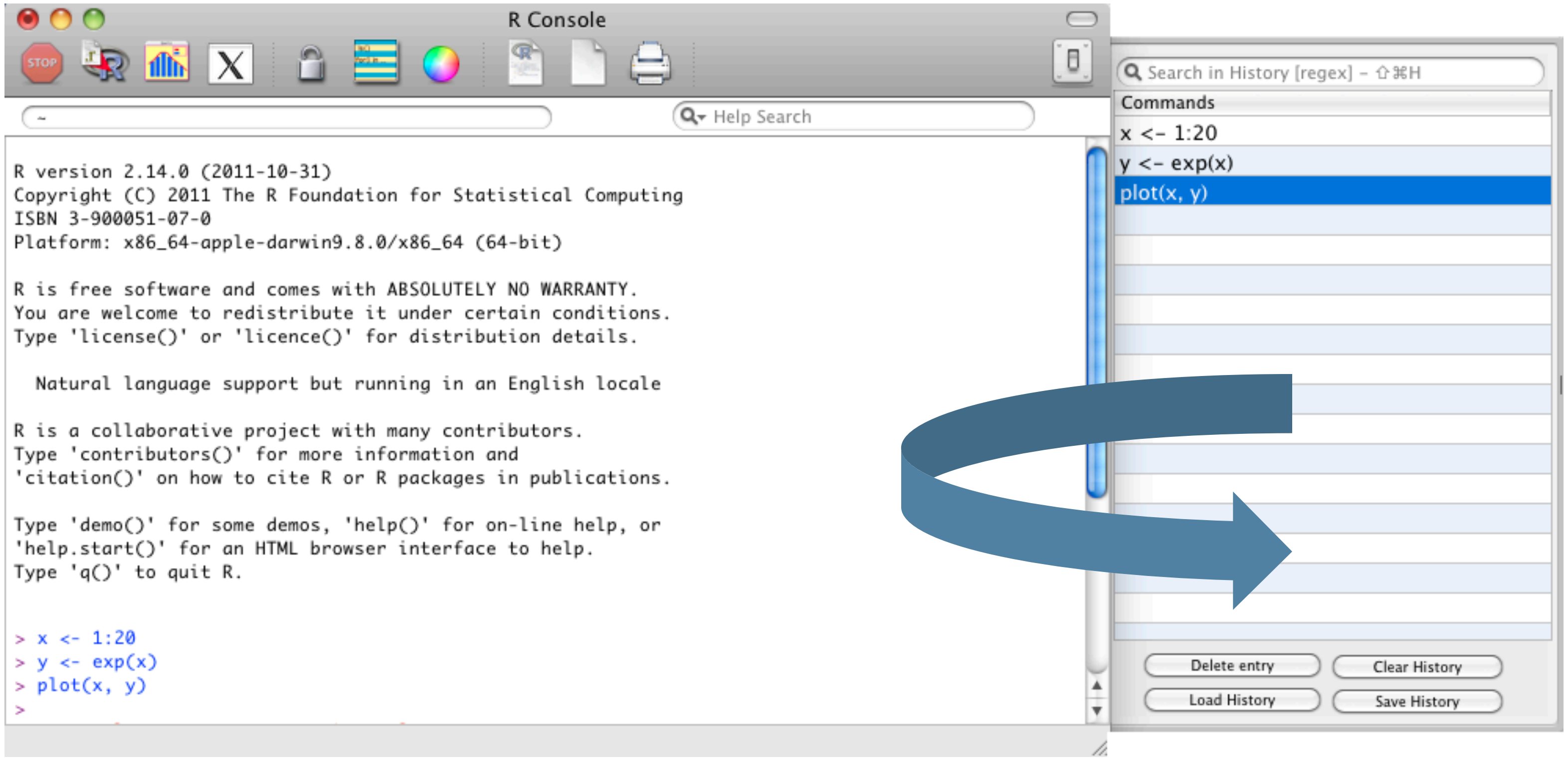
CLOUDY	SPRINKLER F	SPRINKLER T
F	0.5	0.5
T	0.9	0.1

CLOUDY	RAIN F	RAIN T
F	0.8	0.2
T	0.2	0.8

RAIN	SPRINKLER	WET GRASS F	WET GRASS T
F	F	1	0
F	T	0.1	0.9
T	F	0.1	0.9
T	T	0.01	0.99



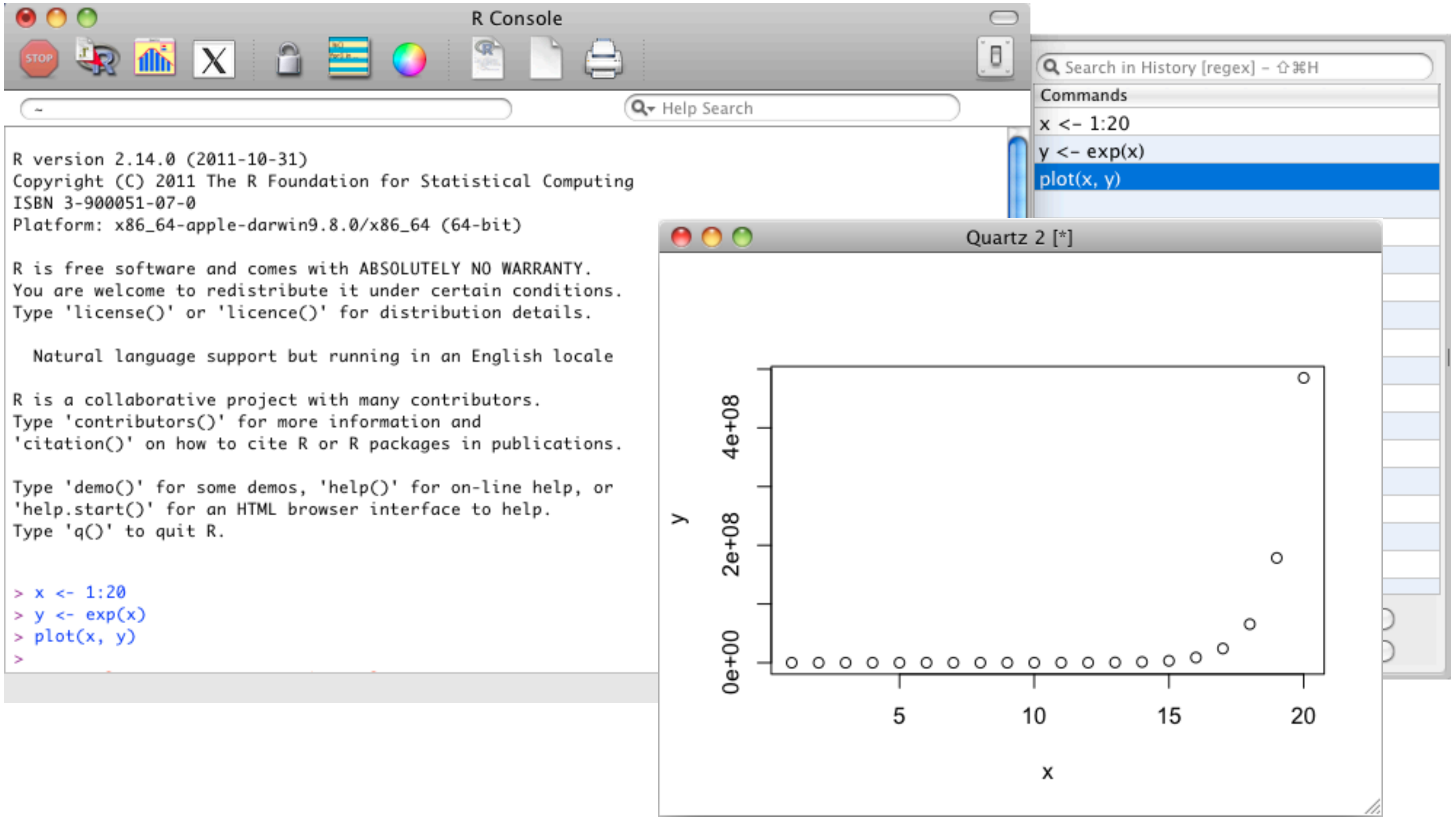
Introduction: Your First R Session



```
R version 2.14.0 (2011-10-31)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

> x <- 1:20
> y <- exp(x)
> plot(x, y)
>
```

```
x <- 1:20
y <- exp(x)
plot(x, y)
```

Introduction: Your First R Session



```
R version 2.14.0 (2011-10-31)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

> x <- 1:20
> y <- exp(x)
> plot(x, y)
>
```

R Basics: Getting Started

- Hello World

```
> print(“Hello World!”)
```

- Help on operators, functions etc. available internally

```
> help(“*”)
> help(exp)
```

- Assignment operator

```
> e <- m*c^2
```

- Display defined objects

```
> ls()
```

- Remove object `e`

```
> rm(e)
```

- `# This is a comment`

R Basics: Vector

- Ordered collection of items of same type/mode (logical, integer, double, complex, character, factor, raw)
- Create using combine function

```
> vec <- c(0, 1, 1, 2, 3, 5, 8)
```

- Create using sequence and repetition

```
> vec <- seq(1, 5, by = 1) # short: 1:5
> vec <- rep(1, length = 17)
```

- Operators `+`, `-`, `*`, `/`, and `^` are applied *element wise*

```
> c(2, 5, 10) * c(1, 5, -1.7)
[1] 2 25 -17
```

- Recycling

```
> c(1, 2, 3, 4) * c(1, 2)
[1] 1 4 3 8
```

R Basics: Vector

- Indexing (R is *one-indexed*!)

```
> vec <- c(0, 1, 2, 3, 4)
> vec[c(2,4)]
[1] 1 3
```

- Basic functions

Number of elements	`length(vec)`
Sum of elements	`sum(vec)`
Arithmetic mean	`mean(vec)`
Empirical variance	`var(vec)`
Smallest and largest element	`range(vec)`

R Basics: Character Vector

- Character vectors

```
> names <- c(“Einstein”, “Curie”)
```

- Length of strings

```
> nchar(names)
[1] 8 5
```

- String manipulations

```
> substring(names[1], 4, 8)
[1] "stein“
> paste(c("A.", "M."), names)
[1] "A. Einstein" "M. Curie”
> strsplit(“String with spaces”, “ “)
[1] “String” “with” “spaces”
```

R Basics: Logical Vector

- Logical vectors
  - `> logi <- c(TRUE, TRUE, FALSE)`
- Often result from comparison `<`, `<=`, `>`, `>=`, `==`, and `!=`
- Can be used for indexing
  - `> vec <- c(5, 19, 2, 7)`
  - `> ind <- vec > 3 #c(TRUE, TRUE, FALSE, TRUE)`
  - `> vec[ind]`
  - `[1] 5 19 7`
- Operators on logical vectors `x` and `y` (lazy: `&&`, `||`)
  - NOT `! x`
  - AND `x & y`
  - OR `x | y`
  - XOR `xor(x,y)`

R Basics: Matrix

- 2D extension of vector (n-dimensional: use `array()`)
- Create matrix (default: by column)

```
> m.test <- matrix(seq(1, 10), nrow=2, ncol=5)
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
```

- Indexing

```
> m.test[1, 1:3]
[1] 1 3 5
```

- All standard matrix operators available, e.g.
  - Transposition `t()`
  - Matrix mult. `%*% #without “%”: element wise`
  - Inversion `solve()`

R Basics: Data Frame

- Extension of matrix: columns of *different* type
- Frequently used data structure (resembles spreadsheet)
- Create data frame

```
> fname <- c(“Rebecca”, “Peter”, “Jessica”)
> age <- c(27, 58, 28)
> people <- data.frame(first.names = fname, age = age)
     first.names    age
1    Rebecca        27
2    Peter          58
3    Jessica        28
```

- Indexing by column/row number or `name`
  - `> people$age # or: people[,”age”]; or: people[,2]`
  - `[1] 27 58 28`
- Concatenate using `cbind()` and `rbind()`

R Basics: List

- Ordered collection of (named) objects of *different* types

```
> lst <- list(7, name=“John”, child.ages=c(3, 5))
```

- Indexing by `[[position]]` and `$name`

```
> lst[[3]] # or: lst$child.ages
[1] 3 5
```

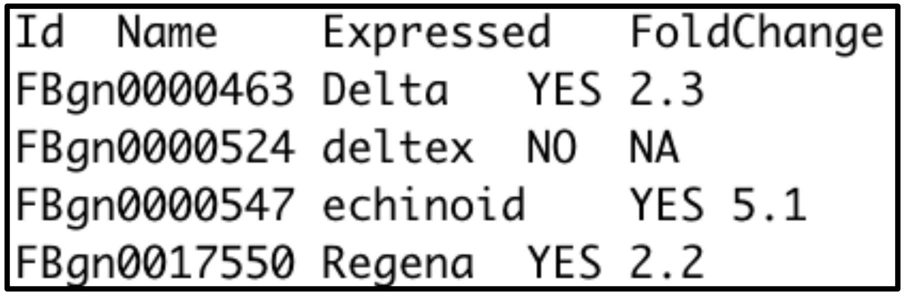
- Concatenation using combine
  - `> lst2 = c(lst, list(wife=“Mary”))`

R Basics: File I/O

- Std. method to read (write) data frames from (to) *text files*
  - `read.table(file, header=FALSE, sep=“”, skip=0, …)`
  - `write.table(x, file=“”, append=FALSE, sep=“ “, …)`
- Similar methods:
  - `read.delim()`
  - `read.csv()`
  - `read.xls()`
  - `scan()`
- Check `help()` for a complete list of parameters and their defaults (eg. `colClasses`, `stringsAsFactors, …`)
- Read and write data as R object: `load()` and `save()`

R Basics: File I/O (Text File Example)

- Tab separated file `mydata.txt`



```
Id	Name	Expressed	FoldChange
FBgn0000463	Delta	YES	2.3
FBgn0000524	deltex	NO	NA
FBgn0000547	echinoid	YES	5.1
FBgn0017550	Regena	YES	2.2
```

- Read mydata.txt

```
> md <- read.table(“mydata.txt”, header=T, sep=“\t”)
> md$FoldChange
[1] 2.3  NA 5.1 2.2
```

- Write `Id` and `FoldChange` columns to `md_procd.txt`

```
> write.table(md[, c(1,4)], file=“md_procd.txt”,
sep=“\t”, row.names=F)
```

R Basics: Defining Functions

- Syntax

```
add <- function(a, b) {
    s = a + b
    return(s)
}
```

- `return()` missing: result of last computation is returned
- Named arguments and defaults

```
foo <- function(x1, x2, x3=2) {
    x1 / x2 * x3
}
> foo(x2=7, x1=21) # x1, x2: reversed, x3: default
[1] 6
```

Conditional and Repetitive Execution

- if/else statement

```
if (expr) {
    …
} else {
    …
}
```

- For Loop

```
for (i in expr) {
    …
}
```

- While Loop

```
while (expr) {
    …
}
```

- WARNING:
  - Loops are not very efficient in R
  - … and not very elegant
  - Use *subsetting* or *`apply()` functions

Subsetting

- Assume we have data frame `df` with gene Ids and an expression value. Task: Remove rows with neg. expr value
- Using a for loop (complicated…)

```
> result = c()
> for (i in 1:nrow(df)) {
+   if (df$expr[i] > 0) {
+         result = rbind(result, df[i, ])
+   }
+ }
```

- Using *subsetting*

```
> result = df[which(df$expr > 0), ]
> result = df[-which(df$expr <= 0),] #minus indexing
```

Family of `*apply()` functions

- Syntax (other: `lapply()`, `sapply()`, …)

```
apply(X, MARGIN, FUN, …)
```

- Given a matrix

```
> m <- matrix(1:10, nrow=2, ncol=5)
```

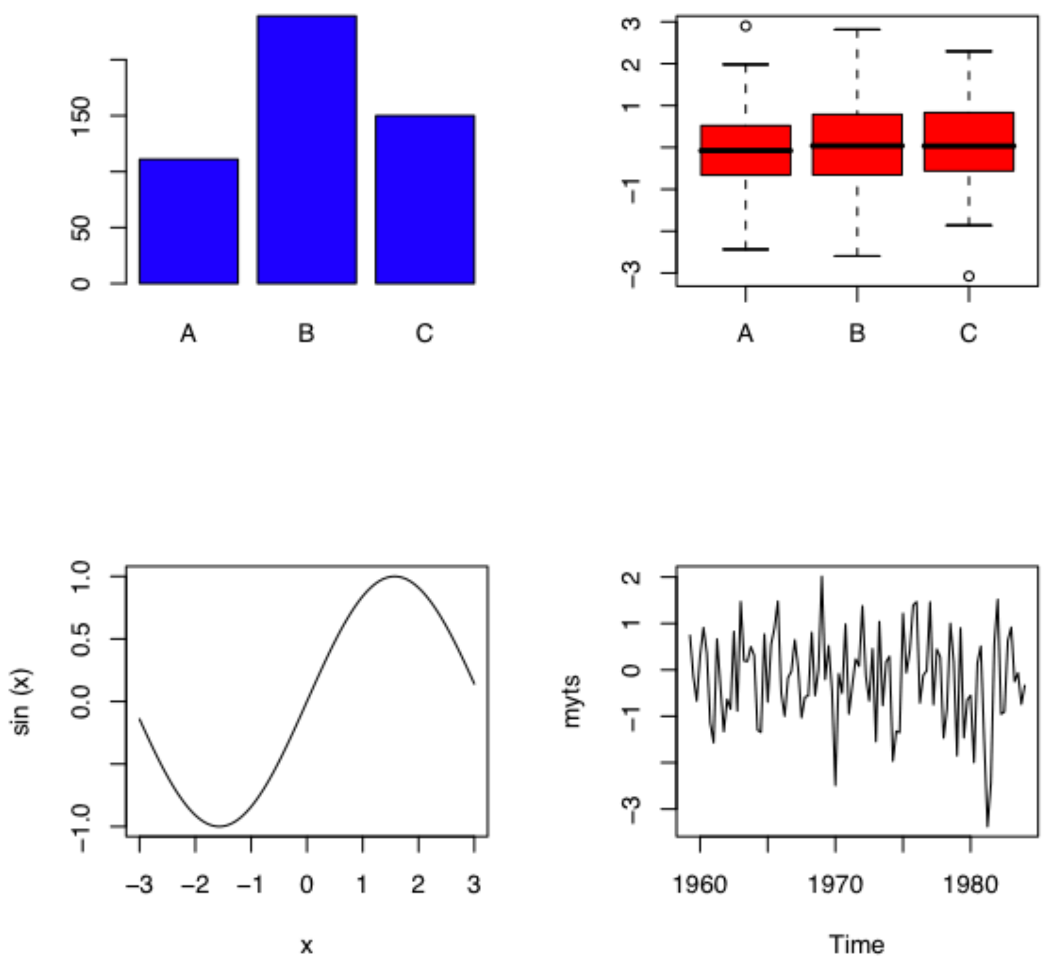
- Compute the sum of each **column**

```
> apply(m, 2, sum)
```

- Applying a custom function

```
> check <- function(row, dist, C) {
+   avg.to.dist <- mean(row[1:dist])
+   return(if(avg.to.dist> C) TRUE else FALSE)
+ }
> apply(m, 1, check, dist=3, C=3)
[1] FALSE TRUE
```

Graphical Procedures



Lam, 2010

Graphical Procedures

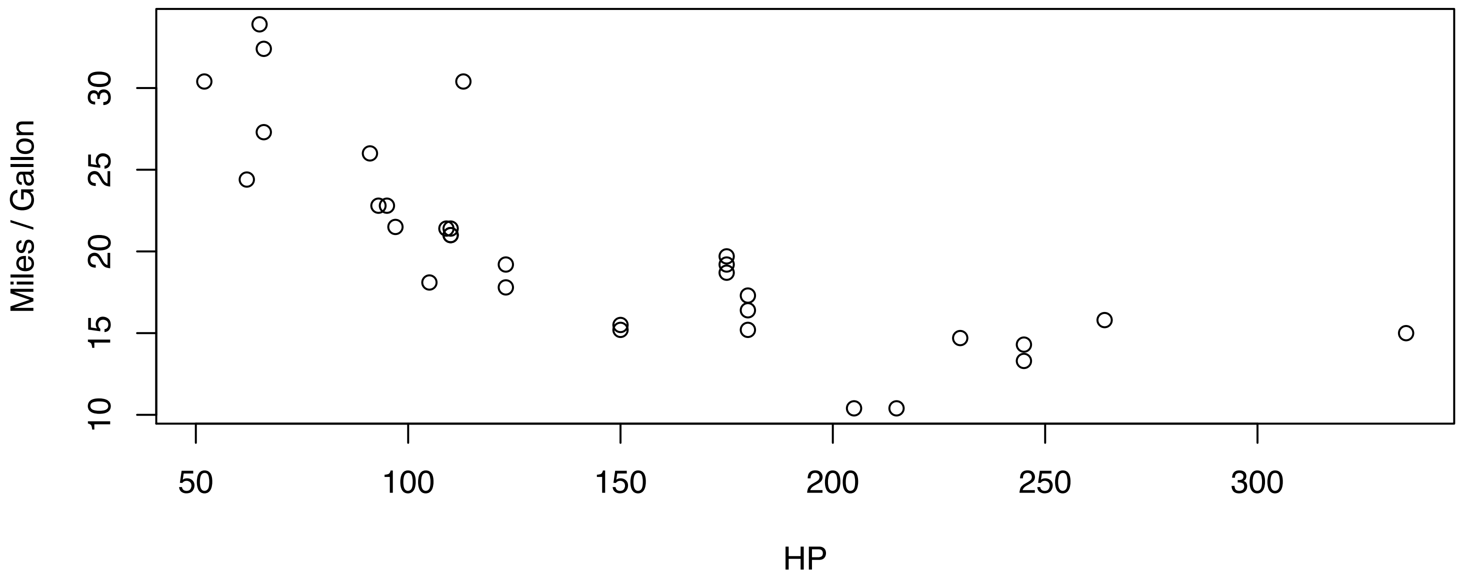
- Generic function for plotting: `plot(x, y, …)`
- Frequently used arguments

Description	Argument
Style of plot (points, lines, …)	`type=“”`
Plot title	`main=“”`
Axis labels	`xlab=“”, ylab=“”`
Axis limits	`xlim=c(a,b), ylim=c(c,d)`
Colour	`col=“green”`
Legend	`legend()`

- Selection of *high*-level plotting functions
  `boxplot(), barplot(), hist(), pie(), …`
- Selection of *low*-level plotting functions
  `lines(), points(), text(), abline(), …`

Graphical Procedures (Example)

```
# Attach built in mtcars data set, ?mtcars for info
> data(mtcars)

# Plot horse power against miles per gallon
> plot(mtcars$hp, mtcars$mpg, xlab=”HP", ylab="Miles / Gallon”)
```

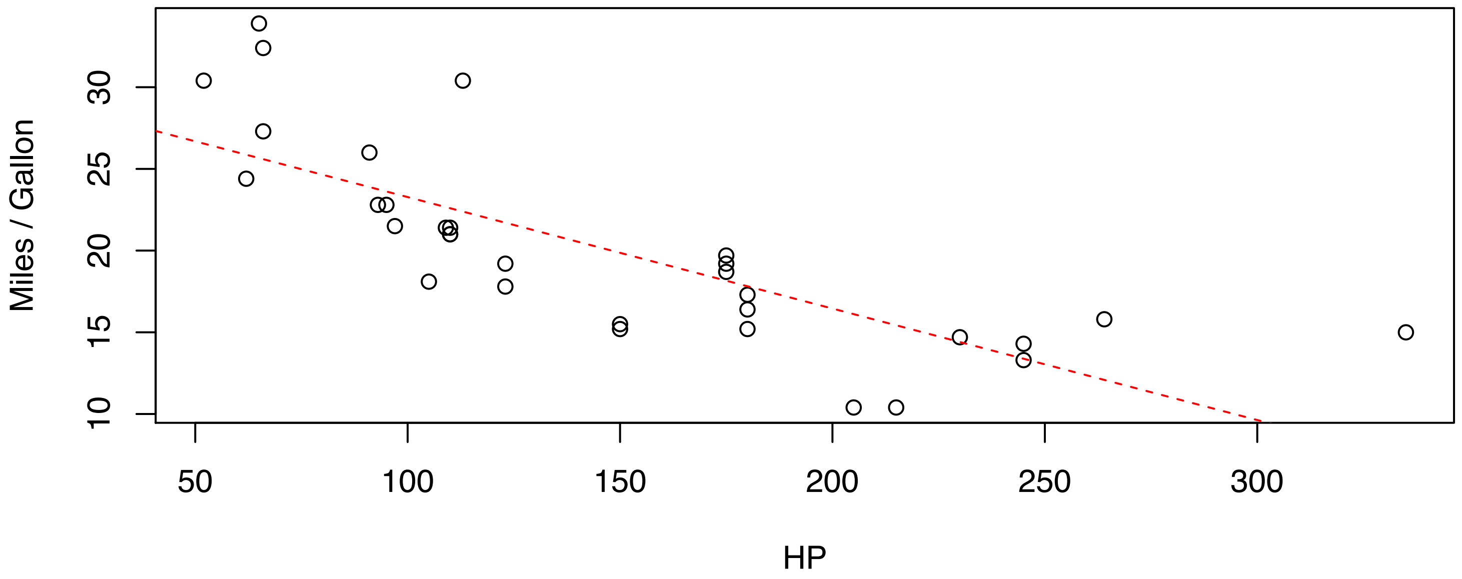


Graphical Procedures (Example)

```
# Fit a simple linear regression
> lsr = lm(mtcars$mpg ~ mtcars$hp) # mpg = a * hp + b

# Show coefficients
> lsr
Call:
lm(formula = mtcars$mpg ~ mtcars$hp)

Coefficients:
(Intercept)    mtcars$hp
   30.09886     -0.06823
```

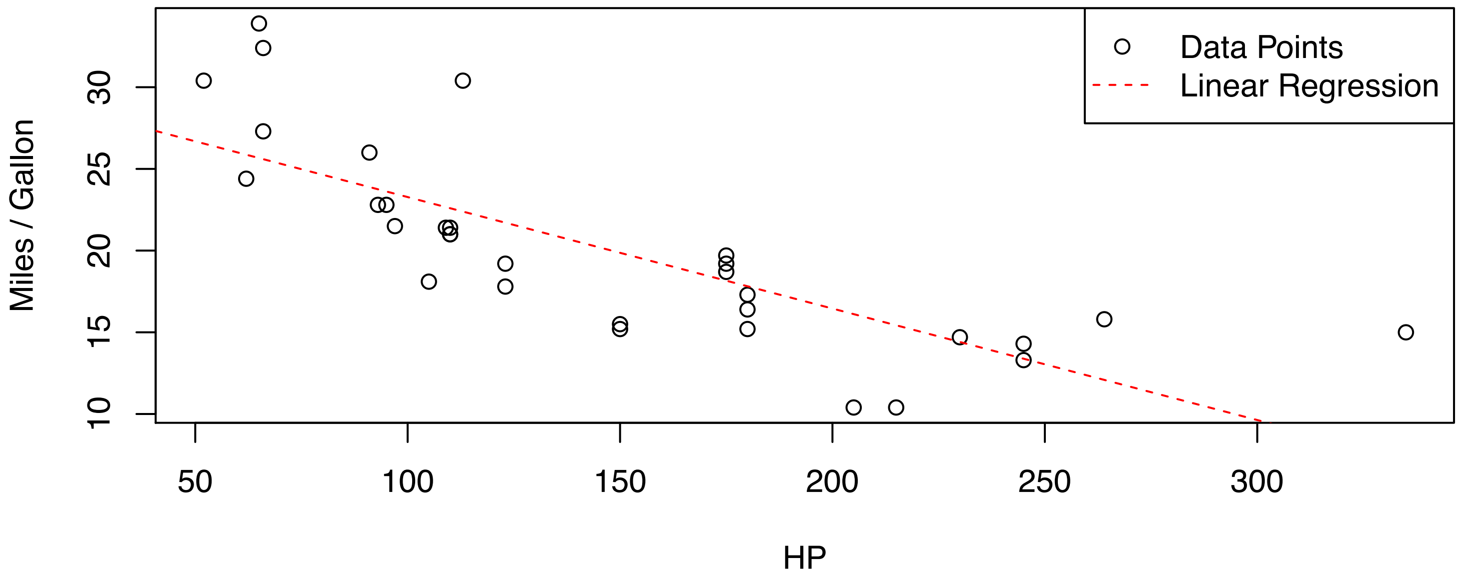
Graphical Procedures (Example)

```
# Add the model to the plot using low-level plotting function
> abline(lsr$coefficients, lty=2, col="red")
```



Graphical Procedures (Example)

```
# Add a legend
> legend("topright", c("Data Points", "Linear Regression"), pch=c(1,
NA), lty=c(0, 2), col=c("black", "red"))
```



Using External Packages

- Main sources for packages
  - http://cran.r-project.org/web/packages/
  - http://www.bioconductor.org/packages/release/bioc/
- Installing downloaded packages from command line

  `R CMD INSTALL [–l lib_path]` package_name.tar.gz
- Downloading and installing packages from bioconductor

```
> source(”http://bioconductor.org/biocLite.R”)
> biocLite(”package_name")
```

- Loading packages in R

```
> library(package_name)
> help(package=package_name)
```

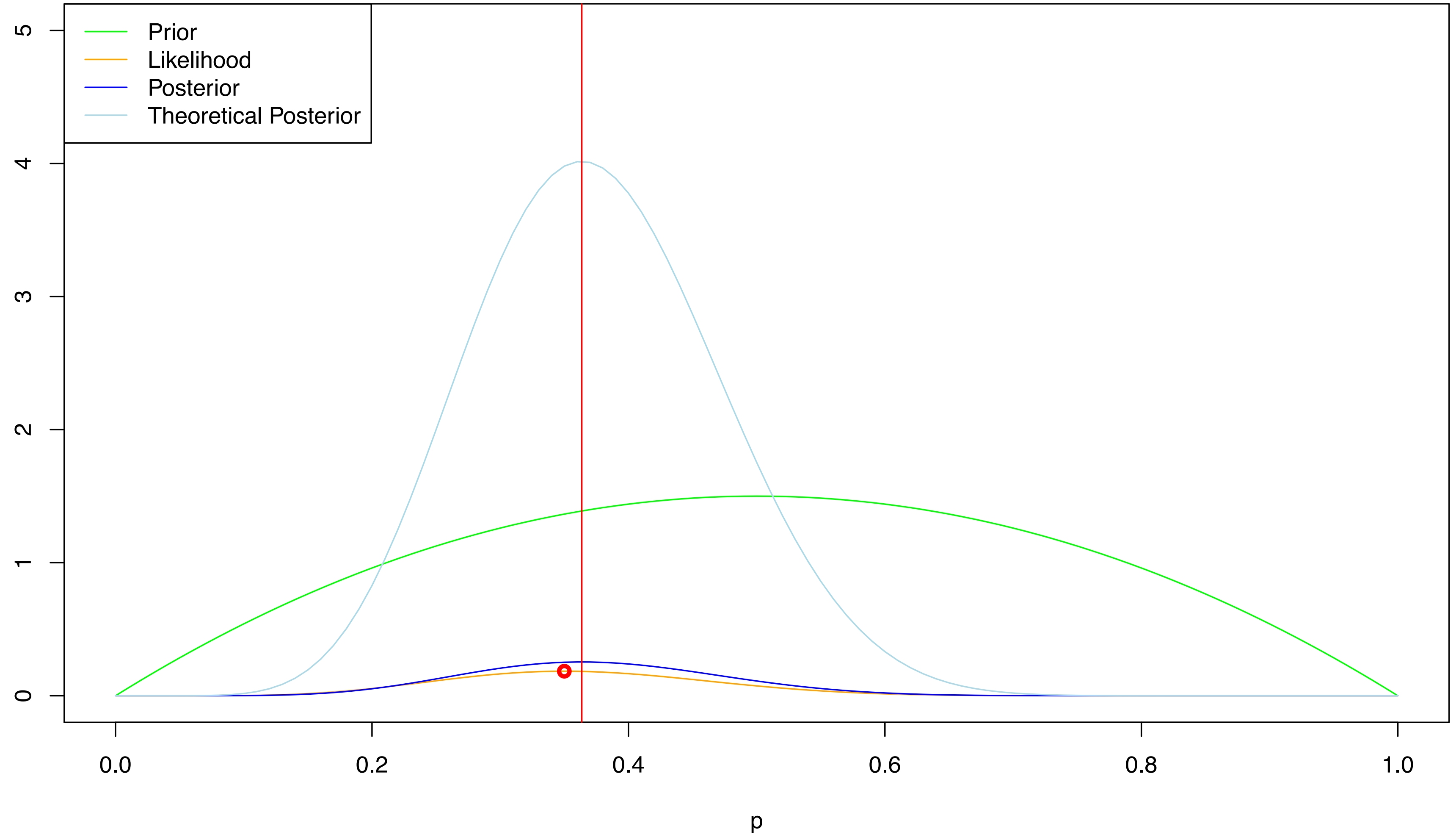
Misc. Useful Functions

Purpose	Function Name
Generic object summary statistics	`summary(obj)`
Show first and last part of object	`head(obj), tail(obj)`
Cast objects	`as.<type>(obj)`
Show object structure	`str(obj)`
Show list of attached packages	`search()`
Show list of attached objects	`ls()`
Clear workspace	`rm(list=ls())`
Identify masked objects	`conflicts()`
Include code from external script	`source(file)`
Run system command	`system(“cmd”)`

Resources

- R Manuals and Tutorials
  http://cran.r-project.org/manuals.html
  http://www.statmethods.net/
- R/Matlab Reference
  http://www.math.umaine.edu/~hiebeler/comp/matlabR.html
- R-Help Mailing List
  https://stat.ethz.ch/mailman/listinfo/r-help
- StatET plugin for Eclipse IDE
  http://www.walware.de/goto/statet

Bayes Example Output (fair coin, `n=20`)


Prior
Likelihood
Posterior
Theoretical Posterior
p