

Tutorial session for BMI 217
13 January 2009
Erik Corona, TA (ecoronap@stanford.edu)
Tiffany Chen, TA (tiff.chen@stanford.edu)

Resources

R home page:

<http://www.r-project.org/>

R search engine

(With a name like "R", Google has a high False Positive rate)

<http://www.rseek.org/>

Official R introduction:

<http://cran.r-project.org/doc/manuals/R-intro.html>

Official R faq:

<http://cran.r-project.org/faqs.html>

Lane Library's FAQ on R

<http://lane.stanford.edu/howto/index.html?id= 1790>

R commands organized by category

(Useful when you have a task but don't know what commands exist for it)

<http://www.stat.berkeley.edu/~epurdom/RCommands/>

Getting help on a function

These two commands are equivalent:

```
> help(function)
```

```
> ?function
```

Simple Calculations

```
> 1 + 2
```

```
[1] 3
```

```
> exp(2)
```

```
[1] 7.389056
```

Assignment

```
> x = 2
```

```
> x
```

```
[1] 2
```

```
> y = x * x
```

```
> y
```

```
[1] 4
```

```
# List the objects in the workspace
```

```
> ls()
```

```
[1] "x" "y"
```

Data types

Vectors, matrices, factors. lists, data frames, functions.

Vectors

```
> x = c ()
```

```
> x
```

```
NULL
```

```
> x = c(1, 2, 3)
```

```
> 1/x
```

```
[1] 1.0000000 0.5000000 0.3333333
```

```

# vector arithmetic
> z = 2 * x
> z
[1] 2 4 6

# available operators:
#: +, -, *, /, ^
# log, exp, sin, cos, tan, sqrt
# max, min, length, sort
# sum, prod
# mean, var
# ... and more.

> sum(x) / length(x)           # mean
[1] 2
> mean(x)                     # built-in function
[1] 2
> sum((x-mean(x))^2)/(length(x)-1) # variance
[1] 1
> var(x)                      # built-in function\
[1] 1

# sequences
> 1:5
[1] 1 2 3 4 5
> seq(length=5, from=0, by=2)
[1] 0 2 4 6 8
> rep(1, times=5)
[1] 1 1 1 1 1

# logic
> x = c(1, 2, 3)
> x > 2
[1] FALSE FALSE TRUE

# Missing values
> z = c(1:3, NA)
> z
[1] 1 2 3 NA
> z == NA
[1] NA NA NA NA
> is.na(z)
[1] FALSE FALSE FALSE TRUE

# Characters
> paste("hello", "world")
[1] "hello world"
> paste("hello", "world", sep = "")
[1] "helloworld"
> labs = paste(c("X","Y"), 1:5, sep="")
> labs
[1] "X1" "Y2" "X3" "Y4" "X5"

#Indexing
> x = c(10.4,5.6,3.1,6.4,21.7)
> x[1]
[1] 10.4
> x[1:3]
[1] 10.4 5.6 3.1

y = x[x > 5]
y
[1] 10.4 5.6 6.4 21.7
> length(x)
[1] 5
> length(y)

```

```

[1] 4

# set missing values to 0
> z = c(x, NA)
> z
[1] 10.4  5.6  3.1  6.4  21.7  NA
> z[is.na(z)] = 0
> z
[1] 10.4  5.6  3.1  6.4  21.7  0.0

# Index by name
> fruit = c(5, 10, 1, 20)
> names(fruit) = c("orange", "banana", "apple", "peach")
> lunch = fruit[c("apple", "orange")]
> lunch
apple orange
1      5

```

Matrices

```

> x = 1:20
> dim(x) = c(4,5)
> x
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20

```

```

> matrix(1:20, nrow=4)
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20

```

```

# names
> rownames(x) = LETTERS[1:4]
> x
      [,1] [,2] [,3] [,4] [,5]
A      1    5    9   13   17
B      2    6   10   14   18
C      3    7   11   15   19
D      4    8   12   16   20

```

```

# indexing
> x[,1]
[1] 1 2 3 4          # retrieve column 1

> x[1,]
[1] 1 5 9 13 17     # retrieve row 1

> x[,-5]
      [,1] [,2] [,3] [,4]
A      1    5    9   13
B      2    6   10   14
C      3    7   11   15
D      4    8   12   16
# remove the last column

> x[-1,]
      [,1] [,2] [,3] [,4] [,5]
B      2    6   10   14   18
C      3    7   11   15   19
D      4    8   12   16   20
# remove the first row

```

```

# transpose
> t(x)

```

```
      A B C D
[1,]  1 2 3 4
[2,]  5 6 7 8
[3,]  9 10 11 12
[4,] 13 14 15 16
[5,] 17 18 19 20
```

```
# row and column binding
```

```
> a = 1:5
```

```
> b = 6:10
```

```
> cbind(a,b)
```

```
      a b
```

```
[1,] 1 6
```

```
[2,] 2 7
```

```
[3,] 3 8
```

```
[4,] 4 9
```

```
[5,] 5 10
```

```
> rbind(a,b)
```

```
 [,1] [,2] [,3] [,4] [,5]
```

```
a    1    2    3    4    5
```

```
b    6    7    8    9   10
```

Factors

```
> pain = c(0, 3, 2, 2, 1)
```

```
> fpain = factor (pain, levels=0:3)
```

```
> fpain
```

```
[1] 0 3 2 2 1
```

```
Levels: 0 1 2 3
```

```
> levels (fpain) = c("none", "mild", "medium", "severe")
```

```
> fpain
```

```
[1] none severe medium medium mild
```

```
Levels: none mild medium severe
```

Conversions

```
# convert to characters
```

```
> as.character(fpain)
```

```
[1] "none" "severe" "medium" "medium" "mild"
```

```
> pain.char = as.character(pain)
```

```
> pain.char
```

```
[1] "0" "3" "2" "2" "1"
```

```
# convert to numeric
```

```
> as.numeric(pain.char)
```

```
[1] 0 3 2 2 1
```

Functions

```
> mult = function(x,y) {
```

```
+ x*y
```

```
+ }
```

```
> x = c(1:5)
```

```
> y = 2
```

```
> mult(x,y)
```

```
[1] 2 4 6 8 10
```

Loops

```
> y = c()
```

```
> y
```

```
NULL
```

```
> for( i in 1:5){
```

```
+   print(x[i])
```

```
+   y = c(y,x[i]*2)
```

```
+ }
```

```
[1] 1
[2] 2
[3] 3
[4] 4
[5] 5
```

```
y
[1] 2 4 6 8 10
```

Apply

```
> x = matrix(1:20, nrow = 4)
```

```
> x
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
```

```
> y = sum(x)
```

```
> y
```

```
[1] 210
```

```
> z = c()
```

```
> for( i in 1:ncol(x)){
```

```
+ z=c(z,sum(x[,i]))
```

```
+ }
```

```
> z
```

```
[1] 10 26 42 58 74
```

```
> a = apply(x,2,sum)
```

```
> a
```

```
[1] 10 26 42 58 74
```

```
> b = apply(x,1,sum)
```

```
> b
```

```
[1] 45 50 55 60
```

?apply for more information

Writing to files and reading from files

```
> surname = I(c("Tukey", "Venables", "Tierney", "Ripley", "McNeil"))
```

```
> nationality = c("US", "Australia", "US", "UK", "Australia")
```

```
> deceased = c("yes", rep("no",4))
```

```
> authors <- data.frame(surname, nationality, deceased)
```

```
> name = I(c("Tukey", "Venables", "Tierney", "Ripley", "Ripley", "McNeil", "R Core"))
```

```
> title = c("Exploratory Data Analysis", "Modern Applied Statistics ...", "LISP-STAT",  
"Spatial Statistics", "Stochastic Simulation",
```

```
"Interactive Data Analysis", "An Introduction to R")
```

```
> other.author = c(NA, "Ripley", NA, NA, NA, NA, "Venables & Smith")
```

```
> books <- data.frame(name,title,other.author)
```

```
> write.table(books,file='books.data',sep='\t')
```

```
> write.table(authors,file='authors.data',sep='\t')
```

```
> books = read.table('books.data', sep='\t', header = T, row.names = 1)
```

```
> books
```

	name	title	other.author
1	Tukey	Exploratory Data Analysis	<NA>
2	Venables	Modern Applied Statistics ...	Ripley
3	Tierney	LISP-STAT	<NA>
4	Ripley	Spatial Statistics	<NA>
5	Ripley	Stochastic Simulation	<NA>
6	McNeil	Interactive Data Analysis	<NA>
7	R Core	An Introduction to R	Venables & Smith

```
> books$name
```

```
[1] Tukey Venables Tierney Ripley Ripley McNeil R Core
```

```
Levels: McNeil R Core Ripley Tierney Tukey Venables
```

```

> attach(books)
> name
[1] Tukey Venables Tierney Ripley Ripley McNeil R Core
Levels: McNeil R Core Ripley Tierney Tukey Venables

# Remove the last column
> books = books[, -(length(books))]

# Read a second file
> authors = read.table('authors.data', sep = '\t', header = T, row.names=1)

save.image('savedData.RData')
load('savedData.RData')

```

Merge (join)

```

> merge(x=authors,y=books, by.x = "surname", by.y="name")
  surname nationality deceased title other.author
1  McNeil  Australia      no  Interactive Data Analysis <NA>
2  Ripley      UK        no    Spatial Statistics <NA>
3  Ripley      UK        no    Stochastic Simulation <NA>
4  Tierney     US        no      LISP-STAT <NA>
5   Tukey     US        yes  Exploratory Data Analysis <NA>
6 Venables  Australia     no Modern Applied Statistics ... Ripley

```

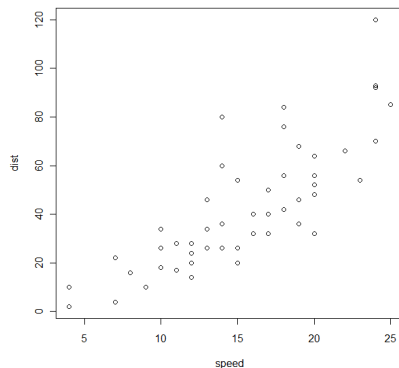
Plotting and hypothesis testing.

If you don't use the R GUI, you must install X11 in cygwin for pc

```

# Pre-fabricated data 'cars' comes with every R installation
> data(cars)
> ?cars
> attach(cars)
> names(cars)
[1] "speed" "dist"
> plot(x = speed, y = dist)

```



```

# Correlation
> cor.test(dist,speed)

```

Pearson's product-moment correlation

```

data: dist and speed
t = 9.464, df = 48, p-value = 1.49e-12
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6816422 0.8862036
sample estimates:
 cor
0.8068949

```

```

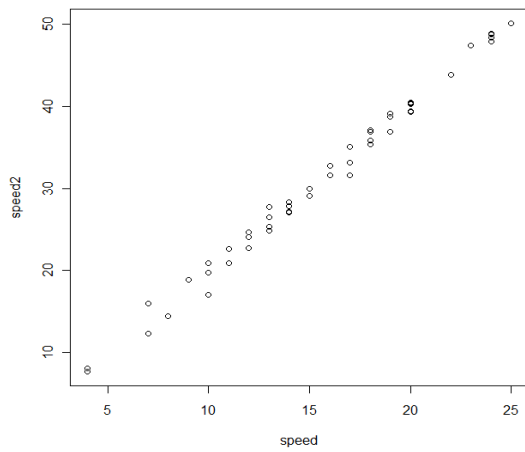
# T-test

```

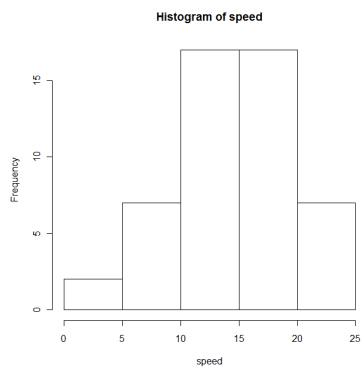
```
# make another set of cars that goes twice as fast
```

```
> speed2 = speed * 2 + rnorm(length(speed))
```

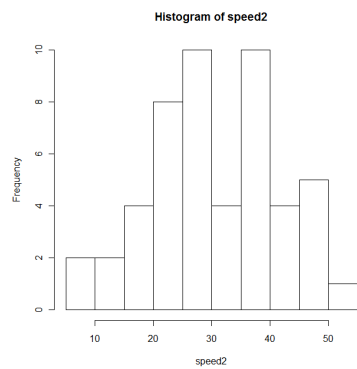
```
# twice the speed plus some variation
```



```
> hist(speed)
```



```
> hist(speed2)
```



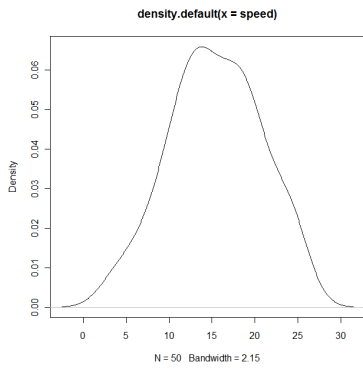
```
> mean(speed)
```

```
[1] 15.4
```

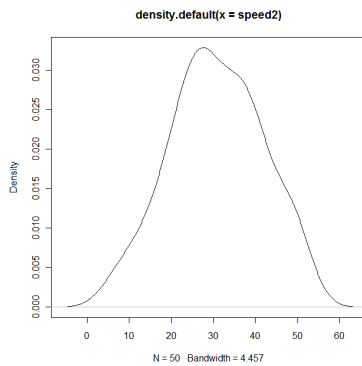
```
> mean(speed2)
```

```
[1] 30.74978
```

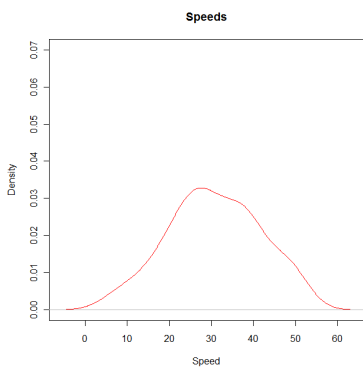
```
> plot(density(speed))
```



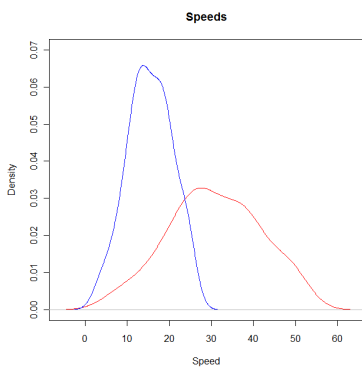
```
> plot(density(speed2))
```



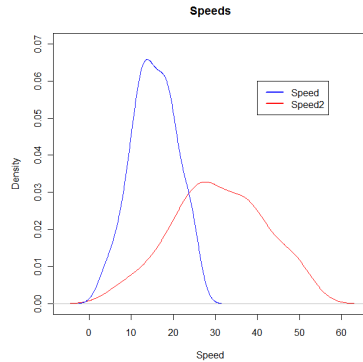
```
> plot(density(speed2), col = 'red', xlab = 'Speed', ylim = c(0, 0.07), main = 'Speeds')
```



```
> lines(density(speed), col='blue')
```



```
> legend(x=40, y = 0.06, c('Speed', 'Speed2'), col=c('blue', 'red'), lty =1, lwd =2)
```



```
# Are the means of the two distributions significantly different?
> t.test(speed,speed2)
```

Welch Two Sample t-test

```
data: speed and speed2
t = -9.0064, df = 71.107, p-value = 2.256e-13
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -18.74800 -11.95156
sample estimates:
mean of x mean of y
 15.40000  30.74978
```

```
# Saving plot to a file
pdf(file = 'speeds.pdf')

> pdf(file='speeds.pdf')
> plot(density(speed2), col = 'red', xlab = 'Speed', ylim = c(0, 0.07), main = 'Speeds')
> lines(density(speed), col='blue')
> legend(x=40, y = 0.06, c('Speed', 'Speed2'), col=c('blue', 'red'), lty =1, lwd =2)
> dev.off()
```

Installing packages (e.g. SAM)

R packages contain contributed functions. Load an R package from a library (directory). A library

may contain multiple packages (or only one).

Bioconductor: A collection of packages useful for bioinformatics.

<http://www.bioconductor.org>

Bioconductor installation instructions:

<http://www.bioconductor.org/docs/install/howto.html>

```
> ?install.packages # for help on installing packages
> install.packages('package.name') # e.g. samr for SAM
```

```
> library ( 'samr' )
Error in library("samr") : there is no package called 'samr'
```

```
# Load the bioconductor function 'biocLite'
> source("http://www.bioconductor.org/biocLite.R")
> biocLite() # gets a default set of packages
```

```
> install.packages('samr')
```

```
# Type ?samr to learn how to use samr
```