Introduction to R:

Using R for statistics and data analysis

BaRC Hot Topics – October 2011

George Bell, Ph.D.

http://iona.wi.mit.edu/bio/education/R2011/

Why use R?

- To perform inferential statistics (e.g., use a statistical test to calculate a p-value)
- To do real statistics (unlike in Excel)
- To create custom figures
- To automate analysis routines (and make them more reproducible)
- To reduce copying and pasting
  - But Unix commands may be easier – ask us
- To use up-to-date analysis algorithms
- Real statisticians use it
- It's free

Why not use R?

- A spreadsheet application already works fine
- You're already using another statistics package
  - Ex: Prism, MatLab
- It's hard to use at first
  - You have to know what commands to use
- Real statisticians use it
- You don't know how to get started
  - Irrelevant if you're here today

Getting started

- Log into tak

```
ssh –l USERNAME tak
```

- Start R

```
R
```

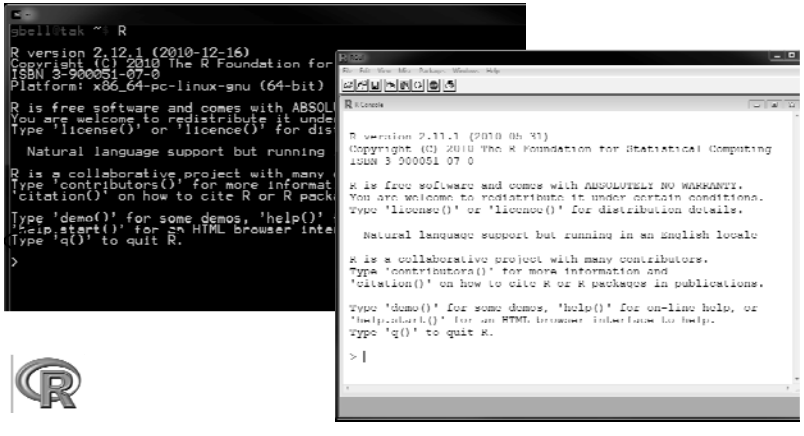
or

- Go to R (http://www.r-project.org/)
- Download "base" from CRAN and install it on your computer
- Open the program

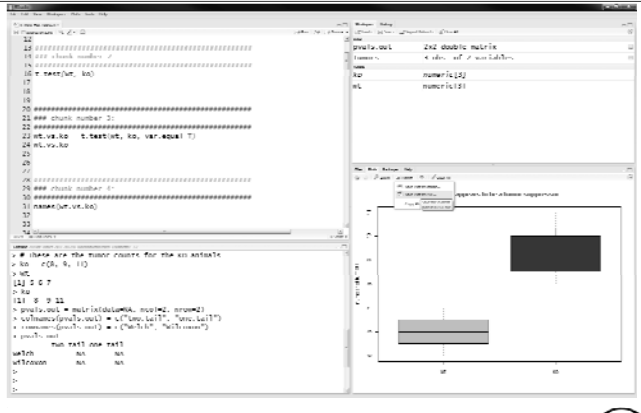
Start of an R session

On tak

On your own computer



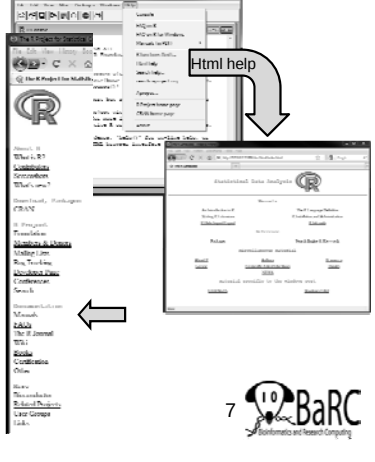
RStudio interface



Requires R; free download from http://rstudio.org/

Getting help

- Use the Help menu
- Check out “Manuals”
  - http://www.r-project.org/
  - contributed documentation
- Use R’s help
  `?median` [show info]
  `??median` [search docs]
- Search the web
  - “r-project median”
- Our favorite book:
  - Introductory Statistics with R (Peter Dalgard)



Handling data

- Data can be numerical or text
- Data can be organized into
  - Vectors (lists of values)
  - Matrices (2-dimensional tables of data)
  - Data frames (a combination of different types of data)
- Data can be entered
  - By typing (using the “c” command to combine things)
  - From files
- Names of data should start with letters
  - Uppercase + lowercase helps (myWTmice)
  - Can include dots (my.WT.mice)

Good practices

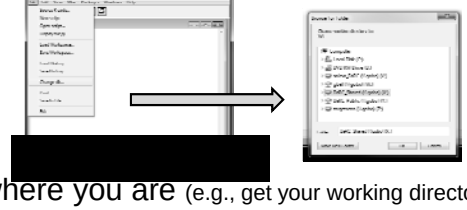
- Save all useful commands and rationale
  - Add comments (starting with "#")
  - Use history() to get previous commands
- Two approaches
  - Write commands in R and then paste into a text file, or
    - By convention, we end files of R commands with ".R"
    - Use a specific name for file (ex: compare_WT_KO_weights.R)
  - Write commands in a text editor and paste into R session.
- Use the up-arrow to get to previous command
  - Minimize typing, as this increases potential errors.
- To clear your R window, use Ctrl-L

Example commands

```
# Number of tumors (from litter 2 on 11 July 2010)
wt = c(5, 6, 7)
ko = c(8, 9, 11)
# Try default t-test settings (Welch's 2-sample t-test)
t.test(wt, ko)
# Do standard 2-sample t-test
t.test(wt, ko, var.equal=T)
# Save the results as a variable
wt.vs.ko = t.test(wt, ko, var.equal=T)
# What are the different parts of this data frame?
names(wt.vs.ko)
# Just print the p-value
wt.vs.ko$p.value
# What commands did we use?
history(max.show=Inf)
```


Reading files - intro

- Take R to your preferred directory ()



- Check where you are (e.g., get your working directory) and see what files are there

```
> getwd()
[1] "X:/bell/Hot_Topics/Intro_to_R"
> dir()
[1] "compare_WT_KO_weights.R"
```


Running a series of commands

- Copy and paste commands into R session, or
- Execute a script in R, or

  `source("compare_WT_KO_weights.R")`

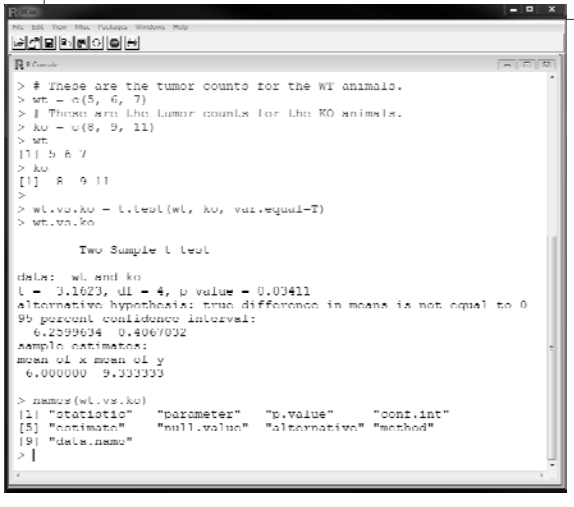
  [but not so useful in this case, since we aren't creating any files]
- [tak only]
  - Change to working directory with Unix command

    `cd /nfs/BaRC/Hot_Topics/Intro_to_R`
  - Run R, with script as input (print to screen), or

    `R --vanilla < compare_WT_KO_weights.R`
  - Run R, with script as input (save output)

    `R --vanilla < compare_WT_KO_weights.R > R_out.txt`

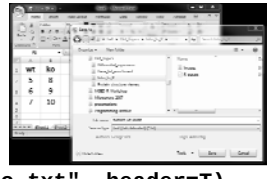
Command output



Partial output from R on tak, if saved as a file (R_out.txt from previous slide), also looks something like this (but without the colors).

Reading data files

- Usually it's easiest to read data from a file
  - Organize in Excel with one-word column names
  - Save as tab-delimited text



- Check that file is there
  `list.files()`
- Read file
  `tumors = read.delim("tumors_wt_ko.txt", header=T)`
- Check that it's OK

```
> tumors
  wt ko
1  5  8
2  6  9
3  7 11
```


Accessing data

```
> tumors
  wt ko
1  5  8
2  6  9
3  7 11
```

```
> tumors$wt       # Use the column name
[1] 5 6 7
> tumors[1:3,1]   # [rows, columns]
[1] 5 6 7
> tumors[,1]      # missing row or column => all
[1] 5 6 7
> tumors[1:2,1:2] # select a submatrix
  wt ko
1  5  8
2  6  9
> t.test(tumors$wt, tumors$ko)  # t-test as before
```


Creating an output table

- Most analyses involve several outputs
- You may want to create a matrix to hold it all
- Create an empty matrix
  - name rows and columns

	two.tail	one.tail
Welch		
Wilcoxon		

```
pvals.out = matrix(data=NA, ncol=2, nrow=2)
colnames(pvals.out) = c("two.tail", "one.tail")
rownames(pvals.out) = c("Welch", "Wilcoxon")
pvals.out
```

```
        two.tail one.tail
Welch         NA       NA
Wilcoxon      NA       NA
```

Filling the output table (matrix)

- Do the stats

```
# Welch's test (t-test with pooled variance)
pvals.out[1,1] = t.test(tumors$wt, tumors$ko)$p.value
pvals.out[1,2] = t.test(tumors$wt, tumors$ko,
alt="less")$p.value
# Wilcoxon rank sum test (non-parametric alternative to
t-test)
pvals.out[2,1] = wilcox.test(tumors$wt,
tumors$ko)$p.value
pvals.out[2,2] = wilcox.test(tumors$wt, tumors$ko,
alt="less")$p.value
pvals.out
```

```
           two.tail   one.tail
Welch    0.04191452 0.02095726
Wilcoxon 0.10000000 0.05000000
```


Printing the output table

- We may want to round the p-values

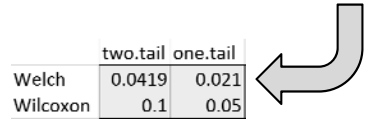
```
pvals.out.rounded = round(pvals.out, 4)
```

- Print the matrix (table)

```
write.table(pvals.out.rounded,
  file="Tumor_pvals.txt", quote=F, sep="\t")
```

- Warning: output column names are shifted by 1 when read in Excel

two.tail	one.tail	
Welch	0.0419	0.021
Wilcoxon	0.1	0.05



	two.tail	one.tail
Welch	0.0419	0.021
Wilcoxon	0.1	0.05

Introduction to figures

- R is very powerful and very flexible with its figure generation
- Any aspect of a figure should be modifiable
- Some figures aren't available in spreadsheets
- Boxplot example

```
boxplot(tumors)      # Simplest case
# Add some more details
boxplot(tumors, col=c("gray", "red"), main="MFG
  appears to be a tumor suppressor", ylab="number
  of tumors")
```


Boxplot description

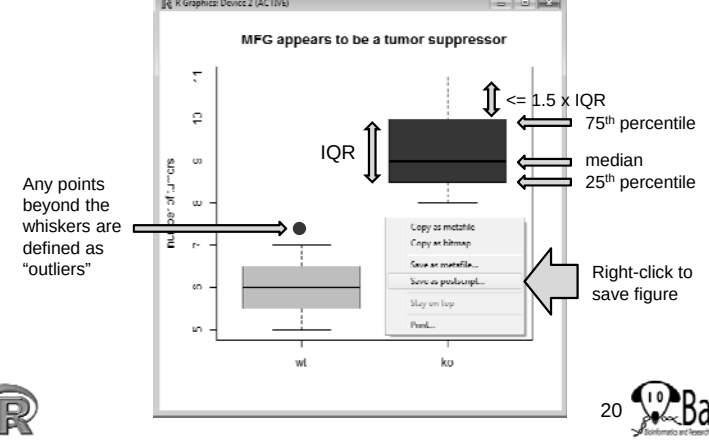


Figure formats and sizes

- By default, figures on tak are saved as “Rplots.pdf”
- Helpful figure names can be included in code

- To select name and size (in inches) of pdf file

```
pdf(“tumor_boxplot.pdf”, w=11, h=8.5)
boxplot(tumors) # can have >1 page
dev.off()       # tell R that we’re done
```

- To create another format (with size in pixels)

```
png(“tumor_boxplot.png”, w=1800, h=1200)
boxplot(tumors)
dev.off()
```

Bioconductor and other packages

- Many statisticians have extended R by creating packages (libraries) containing a set of commands to do something special
  - Ex: affy, limma, edgeR, made4
- For a huge list of Bioconductor packages, see http://www.bioconductor.org/packages/release/Software.html
- All require the package to be installed AND explicitly called, for example,

```
library(limma)
```

- Install what you need on your computer or, for tak, ask the IT group to install packages via
  http://tak.wi.mit.edu/trac/newticket

Other useful commands

`library()`	`mean()`	`round(x, n)`
`dir()`	`median()`	`min()`
`length()`	`sd()`	`max()`
`dim()`	`rbind()`	`paste()`
`nrow()`	`cbind()`	`x[x>0]`
`ncol()`	`sort()`	`x[c(1,3,5)]`
`unique()`	`rev()`	`seq(from, to, by)`
`t()`	`log(x, base)`	`commandArgs()`

More resources from BaRC

- “Statistics for Biologists” course:
  - http://iona.wi.mit.edu/bio/education/stats2007/
- “Microarray Analysis” course
  - http://jura.wi.mit.edu/bio/education/bioinfo2007/arrays/
- R scripts for Bioinformatics
  - http://iona.wi.mit.edu/bio/bioinfo/Rscripts/
- List of R modules installed on tak
  - http://tak/trac/wiki/R
- We’re glad to share commands and/or scripts to get you started

Upcoming Hot Topics

- Introduction to R Graphics (tomorrow)
- Introduction to Bioconductor - microarray and RNA-Seq analysis (Thursday)

- Unix, Perl, and Perl modules (short course)
- Quality control for high-throughput data
- RNA-Seq analysis
- Gene list enrichment analysis
- Galaxy
- Sequence alignment: pairwise and multiple

- See http://iona.wi.mit.edu/bio/hot_topics/
- Other ideas? Let us know.