Introduction to R

BIOINF 525 Lab2-1

w16.bioinfquiz.org

Login: class

Password: c1a55!

What is R?

- Powerful programming language designed for statistical analysis and graphics.
- R is very popular in the field of bioinformatics.
- Available for Mac, Windows and Unix.
- Free but commercial quality.

- www.r-project.org
- www.rstudio.com

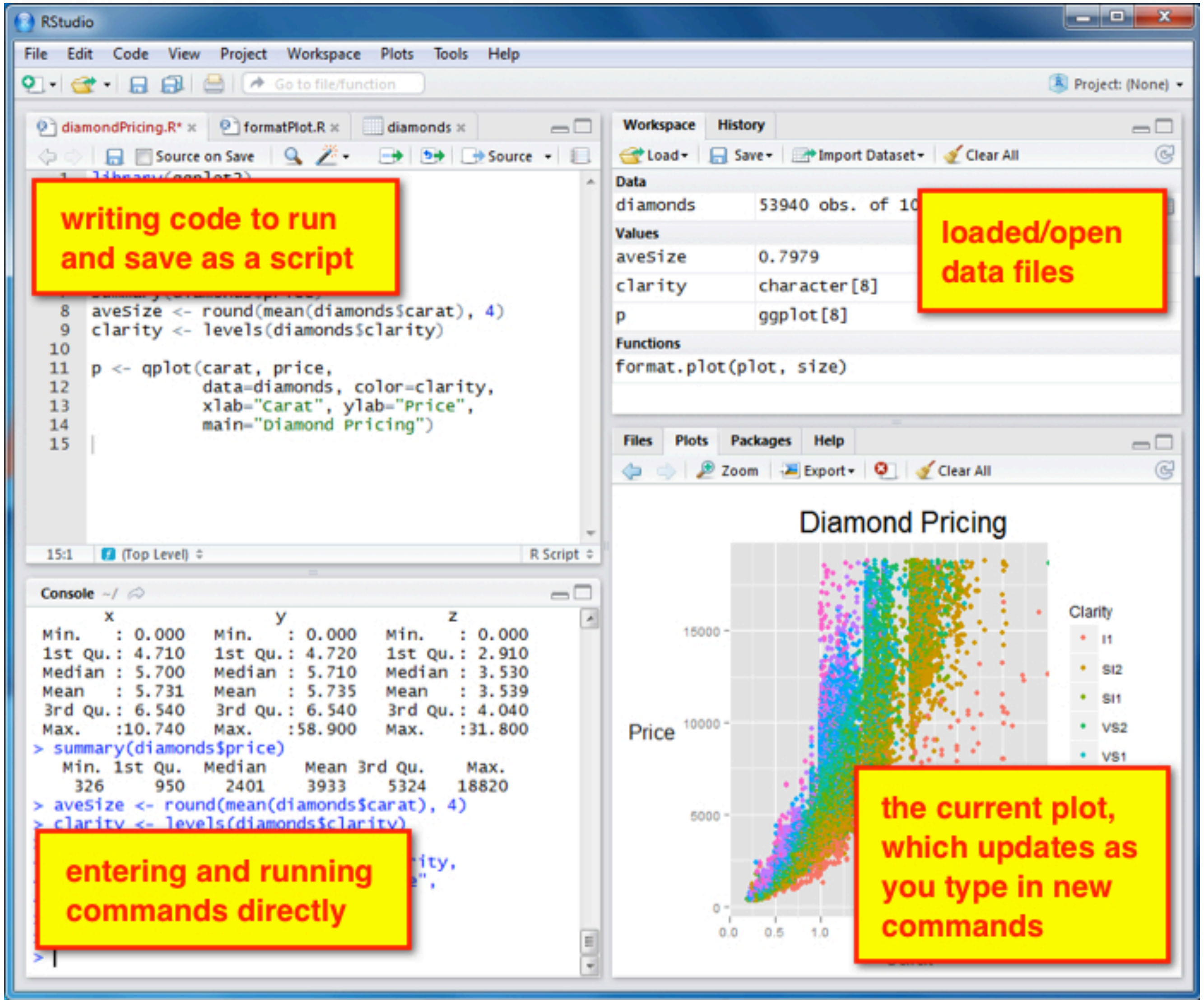
Preparation

Click the black terminal icon
on the top toolbar to open a terminal.

- First make a new directory and open RStudio

```
$ mkdir ~/lab2.1
$ cd ~/lab2.1
$ rstudio
```

RStudio


RStudio
File Edit Code View Project Workspace Plots Tools Help
Go to file/function
Project: (None)
diamondPricing.R* formatPlot.R diamonds
Source on Save
Source
Workspace History
Load Save Import Dataset Clear All
Files Plots Packages Help
Zoom Export Clear All
Diamond Pricing
Price
Clarity
I1
SI2
SI1
VS2
VS1
writing code to run and save as a script
loaded/open data files
the current plot, which updates as you type in new commands
entering and running commands directly
15:1
(Top Level)
R Script
Console ~/

```r
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")
```

Data	
diamonds	53940 obs. of 10
Values	
aveSize	0.7979
clarity	character[8]
p	ggplot[8]
Functions	
format.plot(plot, size)	

```
       x                y                z
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    326     950    2401    3933    5324   18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
>
```

As a calculator

```
> # This is a comment (begins with hash #)
> 10+10
> 10*10
> 10**10
OR
> 10^10

> # Follows order of operations
> 10+10/2
> (10+10)/2

> # Comparisons
> # logical values TRUE or FALSE
> 10 == 5
> 10 != 5
> 10 > 5
> 10 <= 5
```

Saving variables

```
> a <- 10+10
```

OR

```
> a = 10+10
```

```
> a
[1] 20
```

What happens if you type A instead of a?

Vectors

```
> b <- c(1,2,3,4,5)

> b2 <- 1:5

> d <- seq(10,50,by=10)

> e <- rep(0,10)
```

`What do b, b2, d, and e look like?`

Getting help

```
> # if you know the function

> help(mean)
OR
> ?mean


> # if you don’t know the function

> help.search(“variance”)
OR
> ??variance
```

More Vectors

```
> # logical vectors (TRUE or FALSE)
> d > 30

> # You can extract part of a vector
> d[1:2]
> d[d>30]

> # Everything except for a selection
> d[-(2:4)]
```

More Vectors

```
> # character/string vectors

> words=c(‘a’, ‘second’, ‘asdf’)

> words[1]

> words[2]
```

Matrices

```
> cbind(b,d)
```

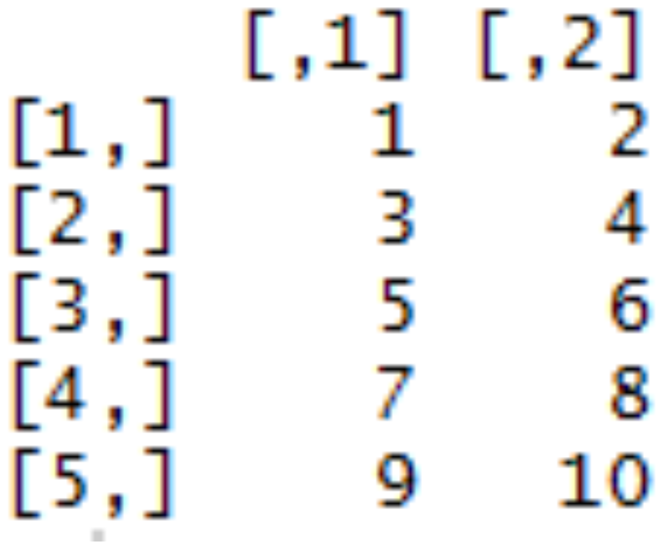
How would you concatenate rows instead of columns?

```
> mat1=matrix(1:10, nrow=5, ncol=2, byrow=TRUE)
> mat2=matrix(1:10, nrow=5, ncol=2, byrow=FALSE)
```

What’s the difference between these two?

More Matrices

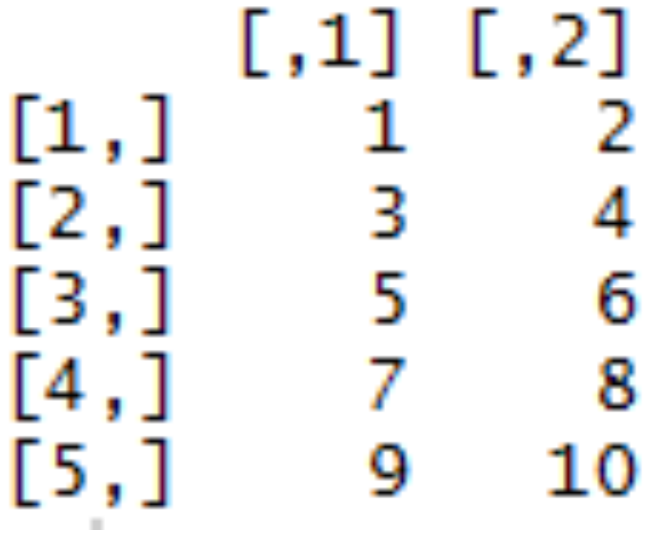
```
> # Just like vectors, we can select part of a matrix

> # What do each of these do?

> mat1[1,2]

> mat1[2:3,1]

> mat1[,1]

> mat1[2,]

> mat1[mat1>3]
```



```
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
[5,]    9   10
```

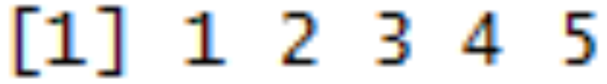
Vector and Matrix Math

```
> mat1
> t(mat1)
> mat1+mat1
> mat1 + 1
> mat1 * 4
```



```
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
[5,]    9   10
```

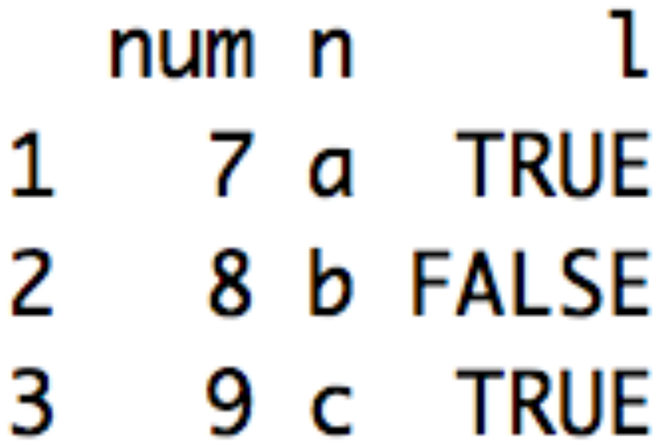
```
> b*b
> mat1*mat1
```



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

data.frames

```
> # A data frame is a list of vectors of the
same length.

> num = c(7,8,9)
> n = c("a","b","c")
> l = c(TRUE,FALSE,TRUE)
> df = data.frame(num,n,l)

> df[2,]
> df[,"n"]
```



```
  num n     l
1   7 a  TRUE
2   8 b FALSE
3   9 c  TRUE
```

Generating normal random numbers

```
> rand=rnorm(10,mean=10,sd=4)

> mean(rand)
> sd(rand)
```

Do these match our inputs?

```
> min(rand)
> max(rand)
```

What are some other distributions?

How could we sample from those instead?

Other Distributions

```
> runif(n, min, max)
> rpois(n, lambda)
> rbinom(n, size, prob)

> # Many more
> ?distributions
```

Opening a file

What if we want to work with outside data?

read.csv, read.delim, read.table

```
> a = read.csv(filename,header,sep)
```

The filename is a string indicating which file to open (including the folders, etc.)

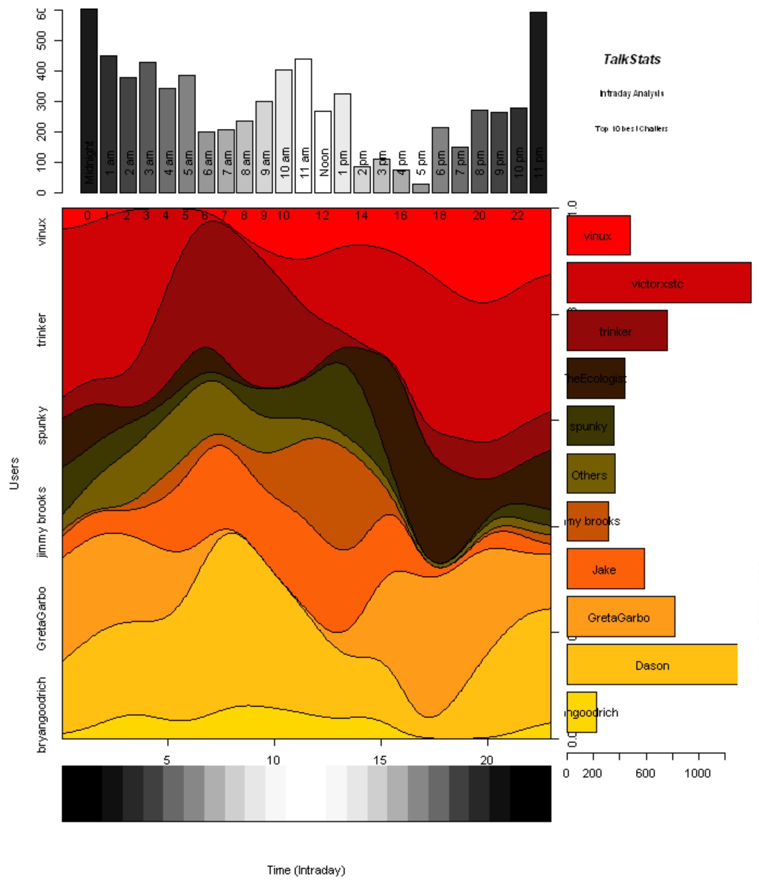
Example filename: “~/Downloads/TROPHY.csv”

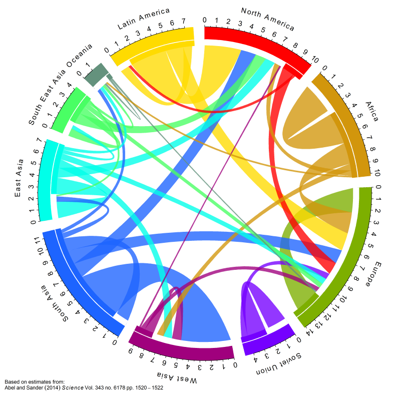
For read.csv(), what do header and sep do?

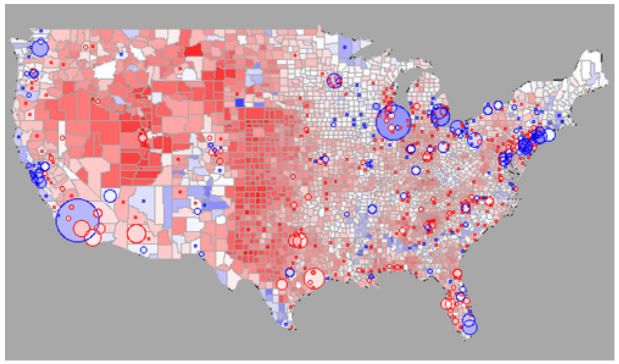
Plotting

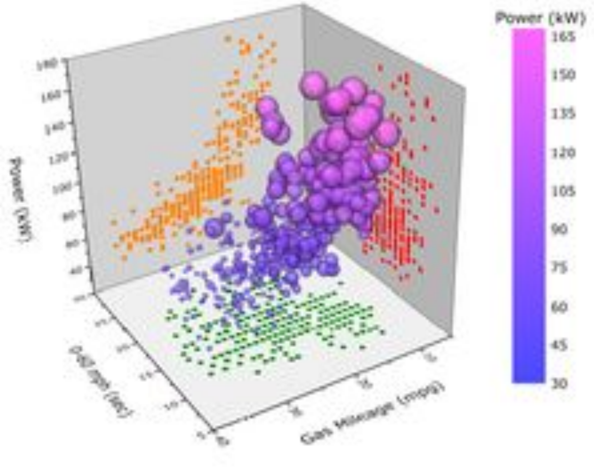
```
> # Create 2 vectors of 1000 random
> # numbers from a normal distribution
> # with mean of 0 and sd of 10

> rand1=rnorm(1000,mean=0,sd=10)
> rand2=rnorm(1000,mean=0,sd=10)
> plot(rand1,rand2)
> hist(rand1)
> boxplot(rand1)
```

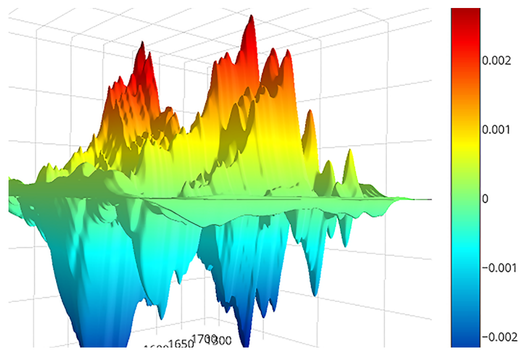
Virtually endless plotting capabilities….











Tutorials and references

- http://cran.r-project.org/doc/manuals/R-intro.html
- http://www.statmethods.net/
- http://bioinformatics.knowledgeblog.org/2011/06/21/using-r-a-guide-for-complete-beginners/
- http://www.cyclismo.org/tutorial/R/
- Many, many more on the internet.