(Re) introduction to Linux

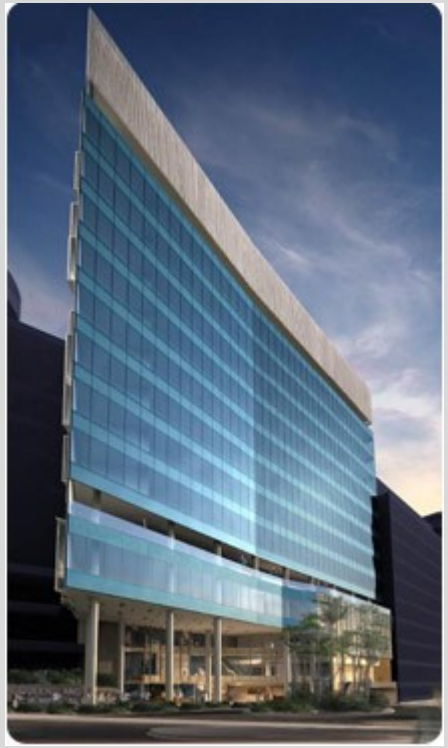
Sarah Medland

Boulder 2019

Getting the most out of the workshop

- Ask questions!!!
- Don’t sit next to someone you already know
- Work with someone with a different skillset and different experience level
- Use the workshop laptop
  - You will have access to your files after you leave
- Come to the social functions
- Ask questions!!!

I work in Brisbane at QIMR



Sarah Medland

not

Morning sessions

- Optional
  - Feel free to wander in and out/check email etc
- Topics
  - Shift in response to feedback
  - Tomorrow: PRS (plink & R)
  - Wednesday: Sex differences & X chromosome
  - Thursday: Python
  - Friday: Simulation

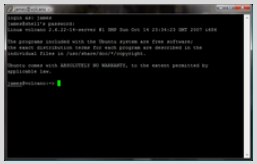
Superfast intro to Linux

This year's OS

- Debian (linux)
  - Free
  - Many free software packages available
    - Open office
    - R
    - PSPP
    - Terminal
- Based on Unix
  - long and venerable history
  - http://en.wikipedia.org/wiki/Unix

Close but not the same…

- Most basic shortcuts will work
  - crtl+C for copy crtl+V for paste etc
- Supports folder based navigation
- /\ BIG PROBLEM is \ vs /
- You will have used some version of unix previously



File hygiene is very important

- Files are stored in Unix format not DOS or Mac
  - Changes the line ending characters
  - Use dos2unix, unix2dos, mac2unix, unix2mac to change formats
  - Can use the file command to check format
- Unix systems are case sensitive!
- NO SPACES in your file/directory names!!
- Wildcards ie dos2unix *.dat

Working in the terminal
Input …. Output

- Input
  - Most commands don't need input signifiers
  - < can be used to specify
- Output
  - Without specifying most output will print to the screen
  - > can be used to direct
    - type: echo 'this is a dummy file'
    - echo 'this is a dummy file' > dummy.txt

    | (pipe) | more pauses the output after a screen worth of text has appeared hit the space bar to get the next screens worth

The manual

- The man command can be used in conjunction with other commands to put up some basic instructions

- type: man ls
  - ls is the list command it pulls up a list of the files in the directory

Also many many helpful webpages w examples

Permissions

the ability to read, write and execute files

- type: ls –l

```
Integlio@Lapis /cygdrive/c/wedtemp
$ ls -l
total 32
-rw-r--r-- 1 Integlio mkpasswd 21 Mar  4 13:25 dummy.txt
```


- These are the permissions
- 1st a directory flag (d or -)
- then 3 letters to define the owners permissions
- 3 letters to define the groups permissions
- 3 letters to define the everyone else's permissions

Permissions

the ability to read, write and execute files

- read access
- write access
- execute
  - to ‘run’ script or a program the file must be made executable

Permissions

the ability to **r**ead, **w**rite and e**x**ecute files

- To change the mode/permissions use chmod
  - a number of ways to do this
  - type: echo “this is a test” > dummy.txt
  - ls –l
  - chmod +x dummy.txt
  - ls –l
  - chmod -x dummy.txt
  - ls –l
  - what happened?

Useful ‘one liners’

- cp copy
- mv move = rename
- rm remove
- ls list
- echo
- head looks at the top 10 lines
- tail looks at the last 10 lines
- wc counts number of lines, words, characters
- sed find and replace
- grep find and report
- awk restructure files

- pwd find where you are
- ~/ get to your home directory
- file reports type of file

Grep

- search **g**lobally for lines matching the **r**egular **e**xpression, and **p**rint them
  - For association output for chromosome 2
  - To extract the result for snp rs59831
  - Type: grep ‘rs59831’ output.txt > summary.txt

Grep

- Useful flags
  - -v
    - reverse grep select line that does not have the pattern
  - -C x
    - To x rows before and after the target
  - -n
    - Print the line number before the line
  - Many more…

Awk

- derived from the surnames of its authors — Alfred **A**ho, Peter **W**einberger, and Brian **K**ernighan
- Many functions
- Very useful for restructuring data

Awk

- Ozbmi2.rec

115	0	0.21	1	2	58	57	1.7	1.7	20.0692	19.7232	20.9943	20.8726
121	0	0.24	1	2	54	53	1.6299	1.6299	20.3244	19.9481	21.0828	20.9519
158	0	0.21	1	2	55	50	1.6499	1.6799	20.202	17.7154	21.0405	20.121
172	0	0.21	1	2	66	76	1.5698	1.6499	26.7759	27.9155	23.0125	23.3043
182	0	0.19	1	2	50	48	1.6099	1.6299	19.2894	18.0662	20.7169	20.2583
199	0	0.26	1	2	60	60	1.5999	1.5698	23.4375	24.3418	22.0804	22.3454
221	0	0.23	1	2	65	65	1.75	1.7698	21.2245	20.7476	21.3861	21.227
239	0	0.29	1	2	40	39	1.5598	1.5298	16.4366	16.6603	19.5966	19.6912
246	0	0.24	1	2	60	57	1.7598	1.7698	19.3698	18.194	20.746	20.3076

- awk '{ print $1, $10, $11, $4, $5 }' ozbmi2.rec > new.rec

```
115 20.0692 19.7232 1 2
121 20.3244 19.9481 1 2
158 20.202 17.7154 1 2
172 26.7759 27.9155 1 2
182 19.2894 18.0662 1 2
199 23.4375 24.3418 1 2
221 21.2245 20.7476 1 2
239 16.4366 16.6603 1 2
246 19.3698 18.194 1 2
```

Awk

- $1 = column 1
- Print $0 = print whole line
- add subtract multiply etc
- change number of decimals
- Many functions

Sort

- Useful flags
  - -f ignore case
  - -n numeric sort
  - -r reverse
  - -c check if a file is sorted
  - -u prints only unique lines
  - -k2 sort starting at column 2

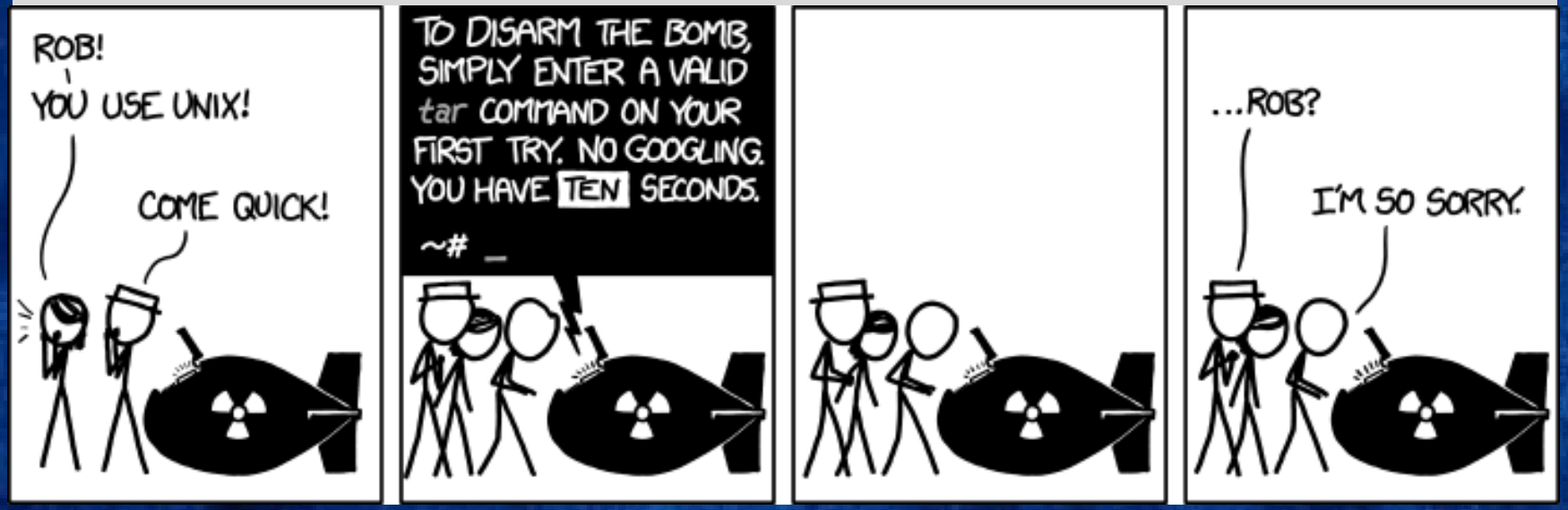
  - sort –fg –k 3 (sort in numeric order on column 3)

Zipping and unzipping

- zip
  - zip my1st.zip *txt
  - zip –mTr my1st.zip *txt
- unzip
  - unzip my1st.zip
- gzip
  - gzip example.txt
- Un-gzip
  - gzip –d example.txt.gz

tar

- Unzipping tar.gz files
  - tar -xzvf example.tar.gz
- Make Tar files
  - tar cvf MyProject.tar MyProject
- List contents
  - tar tvf my-archive.tar
  - tar tzvf my-archive.tar.gz



Looking at your data

- less *filename*
  - Allows you to scroll through your data
- less –S *filename*
  - Shows a screen width of data (stops text wrapping)
- zless –S *filename*
  - Allows you to look at a gz file without unzipping

Nano (text editor)

- nano *filename*
  - Commands at bottom of screen
  - Save = crtl+O
  - Exit = crtl +X

Putting it together

- Making a ‘shell’ script to automate analyses

<contents of imaginary file inefficient.sh>

```
pedstats –p 1.ped –d 1.dat –pdf --prefix:1
merlin –p 1.ped –d 1.dat –m 1.map --vc --pdf --prefix:1
pedstats –p 2.ped –d 2.dat –pdf --prefix:2
merlin –p 2.ped –d 2.dat –m 2.map --vc --pdf --prefix:2
pedstats –p 3.ped –d 3.dat –pdf --prefix:3
merlin –p 3.ped –d 3.dat –m 3.map --vc --pdf --prefix:3
```

To run this make inefficient.sh executable then type ./inefficient.sh

Loops 1

<contents of imaginary file loop_a.sh>

```
for $i in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
   21 22
do
   pedstats –p $i.ped –d $i.dat --pdf --prefix:$i
   merlin –p $i.ped –d $i.dat –m $i.map --vc --pdf --prefix:$i
done
```

Loops 2

<contents of imaginary file loop_b.sh>

```
for ((  i = 1 ;  i <= 22 ;  i++  ))
do
    pedstats –p $i.ped –d $i.dat --pdf --prefix:$i
    merlin –p $i.ped –d $i.dat –m $i.map --vc --pdf --prefix:$i
done
```

Other bits

- When working on servers
  - bg &
  - fg
  - nohup
  - crtl+c
  - crtl+z
  - which

Shutting down you unix session

- exit
- logout
- quit
- q

Superfast intro to R

What is it?

- R is an interpreted computer language.
  - System commands can be called from within R
- R is used for data manipulation, statistics, and graphics. It is made up of:
  - operators (+ - <- * %*% ...) for calculations on arrays & matrices
  - large, coherent, integrated collection of functions
  - facilities for making unlimited types of publication quality graphics
  - user written functions & sets of functions (packages); 800+ contributed packages so far & growing

Advantages

- Fast** and free.
- State of the art: Statistical researchers provide their methods as R packages. SPSS and SAS are years behind R!
- 2nd only to MATLAB for graphics.
- Active user community
- Excellent for simulation, programming, computer intensive analyses, etc.
- Forces you to *think* about your analysis.
- Interfaces with database storage software (SQL)

Disadvantages

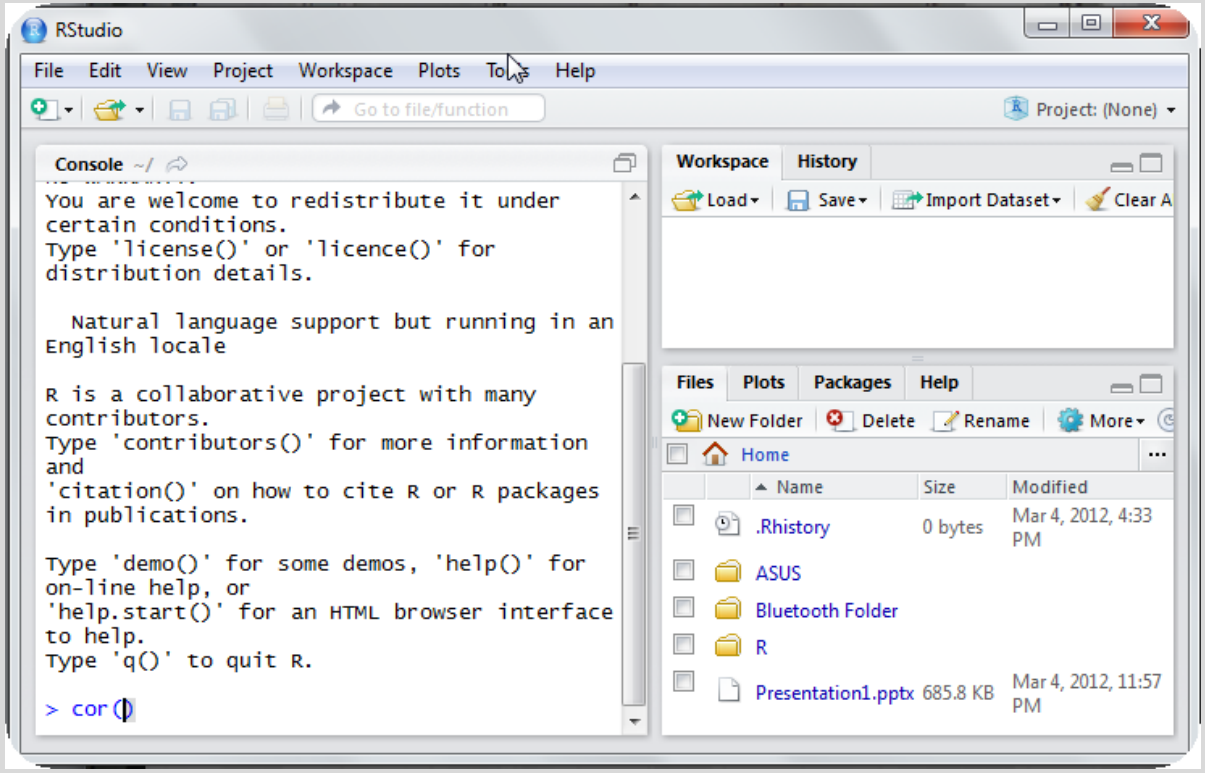
- Not user friendly @ start - steep learning curve, minimal GUI.
- No commercial support; figuring out correct methods or how to use a function on your own can be frustrating.
- Easy to make mistakes and not know.
- Working with large datasets is limited by RAM!!!
- Data prep & cleaning can be messier & more mistake prone in R vs. SPSS or SAS
- Hostility on the R listserve

Learning R....

R-help listserve....

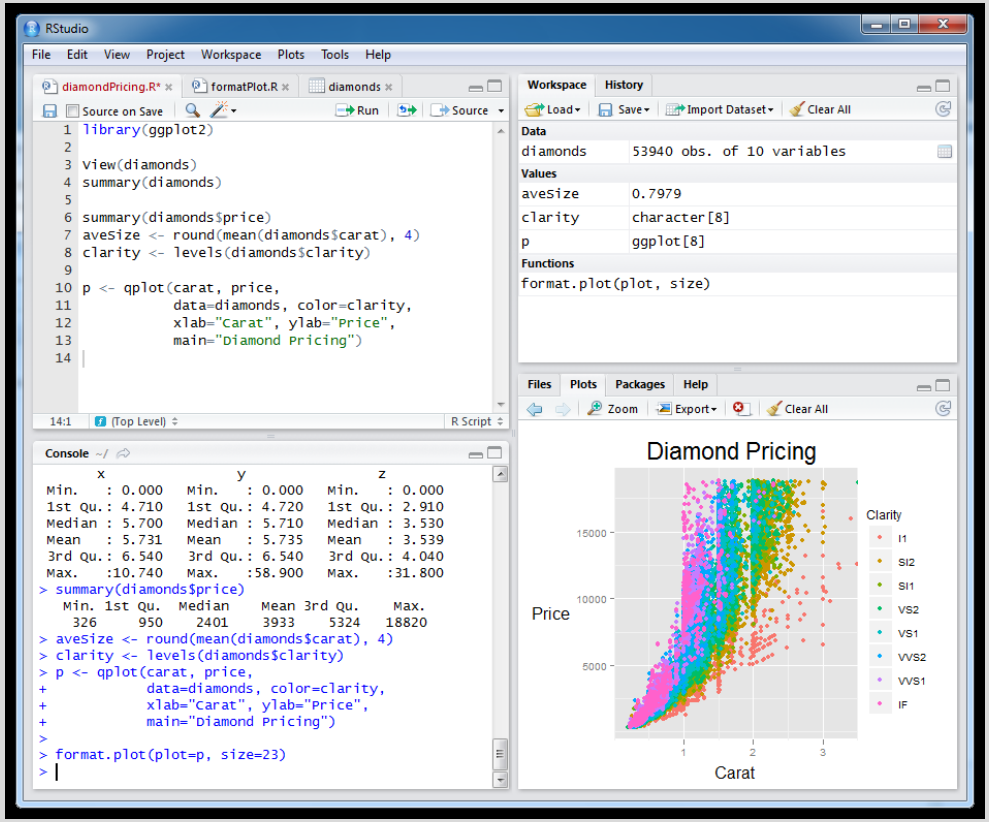
Using R this week

- R-studio http://rstudio.org/



Setting this up at home

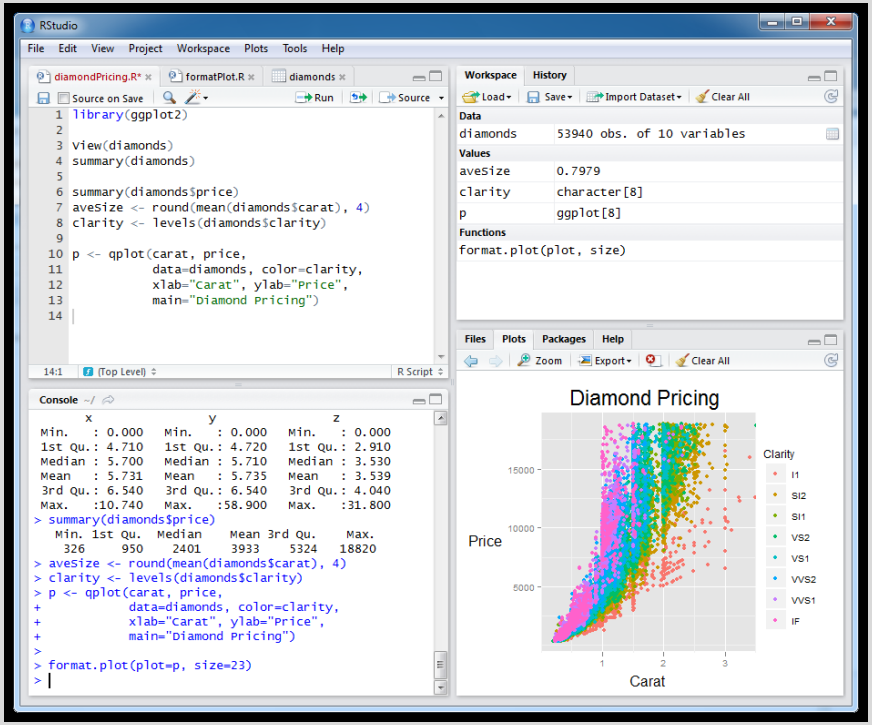
- Install R first
- Install R studio
- Install packages



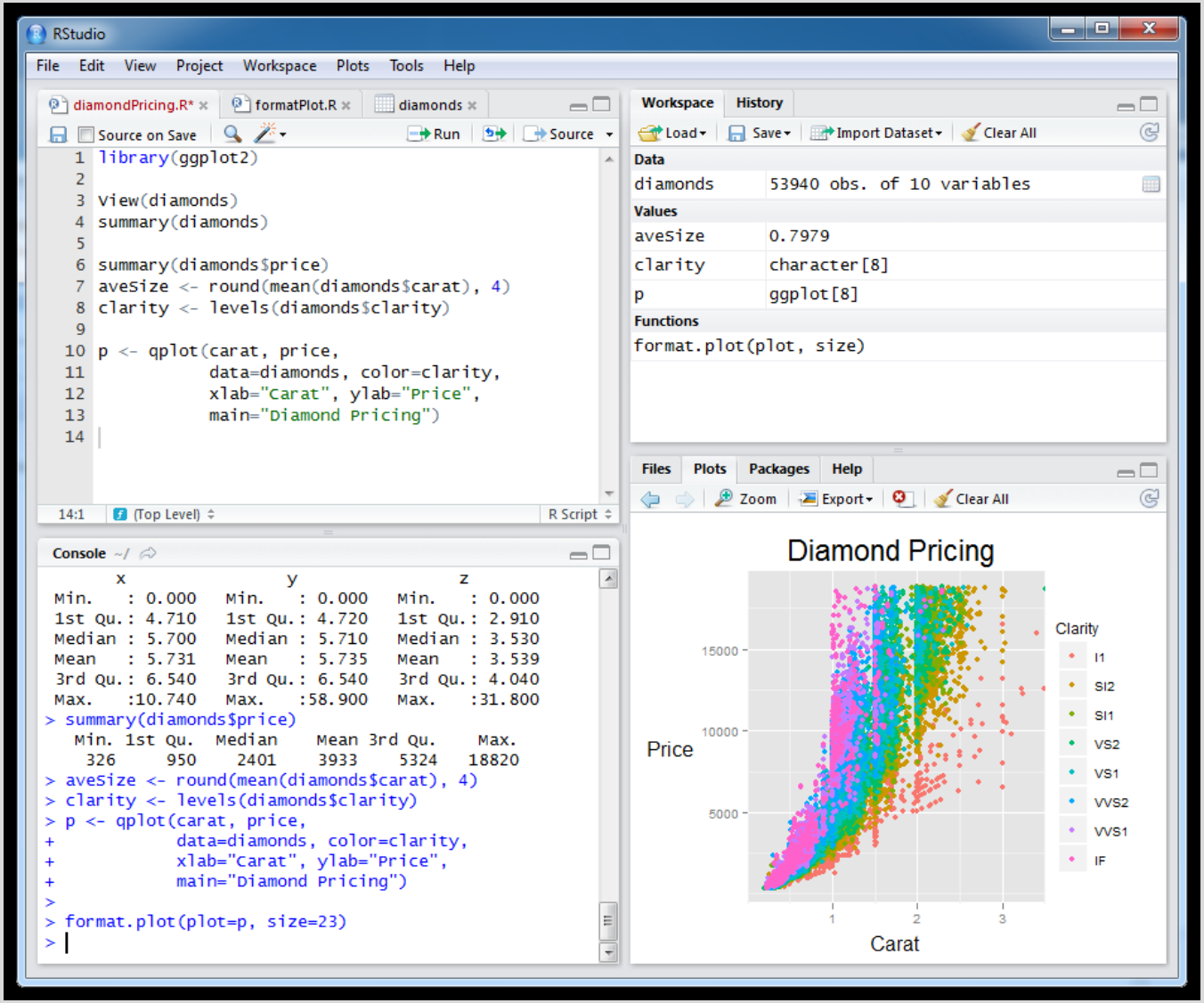
Start up R via R studio

4 windows:

- Syntax – can be opened in regular txt file - saved
- Terminal – output & temporary input - usually unsaved
- Data manager – details of data sets and variables
- Plots etc



R sessions are *interactive*



GETTING STARTED

How to use help in R?

- R has a help system built in.
- If you know which function you want help with use ?_____ or help(_____) with the function in the blank.
  - ?hist.
  - help(hist)
- If you don't know which function to use, then use help.search("_____").
  - help.search("histogram").
- QuickR webpage
  - https://www.statmethods.net/

Importing Data

First make sure your data is in an easy to read format such as space, tab or CSV

Use code:

```
D <- read.table("ozbmi2.txt",header=TRUE)
D <-read.table("ozbmi2.txt",na.strings="-99",header=TRUE)
D <- read.table("ozbmi2.csv", sep="," header=TRUE)
D <- read.csv("ozbmi2.csv", header=TRUE)
```

Exporting Data

Tab delimited

```
write.table(D, "newdata.txt",sep="\t")
```

To xls

```
library(xlsReadWrite)
write.xls(D, "newdata.xls")
```

Checking data

```
#list the variables in D
   names(D)
# dimensions of D
   dim(D)
# print the first 10 rows of D
   head(D, n=10)
#referring to variables in D
   #format is Object$variable
   head(D$age, n=10)
```

Basic Manipulation

#You can make new variables within an existing object

```
D$newage<- D$age*100
```

#Or overwrite a variable

```
D$age<- D$age*100
```

#Or recode a variable

```
#D$catage <- ifelse(D$age > 30,
c("older"), c("younger"))
```

Checking data

#Mean and variance

```
mean(D$age, na.rm =TRUE)
var(D$age , na.rm =TRUE)
```

#For a number of variables

```
lapply(D, mean, na.rm=TRUE)
sapply(D, mean, na.rm=TRUE)
```

Checking data

A bit more info

```
summary(D$age)
summary(D$age[which(D$agecat==1)])
```

What about a categorical variable

```
table(D$agecat)
table(D$agecat,D$zyg)
```

Some basic analysis

typing D$ is getting annoying so we can attach the data

```
attach(D)
table(agecat,zyg)
#detach(D)
```

Correlations anyone?

```
cor(wt1,bmi1, use="complete")
cor(ht1,bmi1, use="complete")
```

regression

Multiple Linear Regression

```
fit <- lm(bmi1 ~ age + zyg, data=D)
summary(fit)
```

Other useful functions

```
coefficients(fit) # model coefficients
confint(fit, level=0.95) # CIs for model
parameters
anova(fit) # anova table
vcov(fit) # covariance matrix for model parameters
```

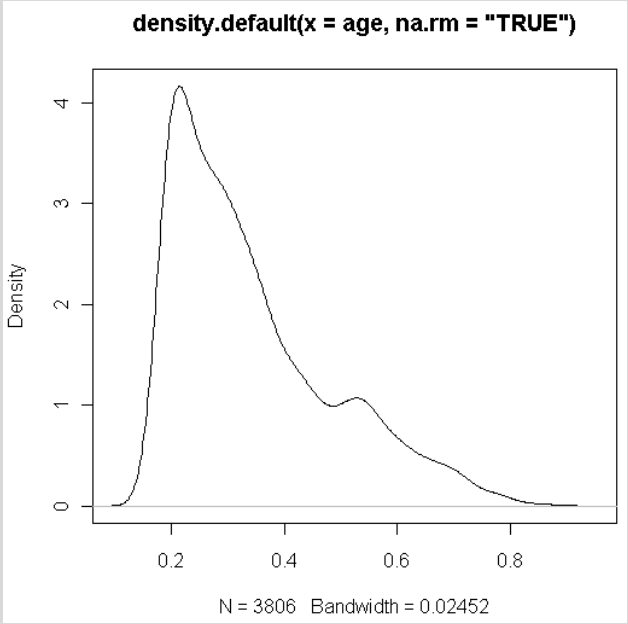
Basic plots

Histogram

```
#basic
   hist(age)
#basic
   hist(age, breaks=12, col=‘red’)
# Add labels
   hist(age, breaks=12, col='red', xlab='age in
   years',main='Histogram of age‘)
```

Looking at your data…

```
#Kernal density plot
d <- density(age, na.rm = "TRUE") # returns the
    density data
plot(d) # plots the results
```



Looking at your data…

#Kernal density plot by zyg?

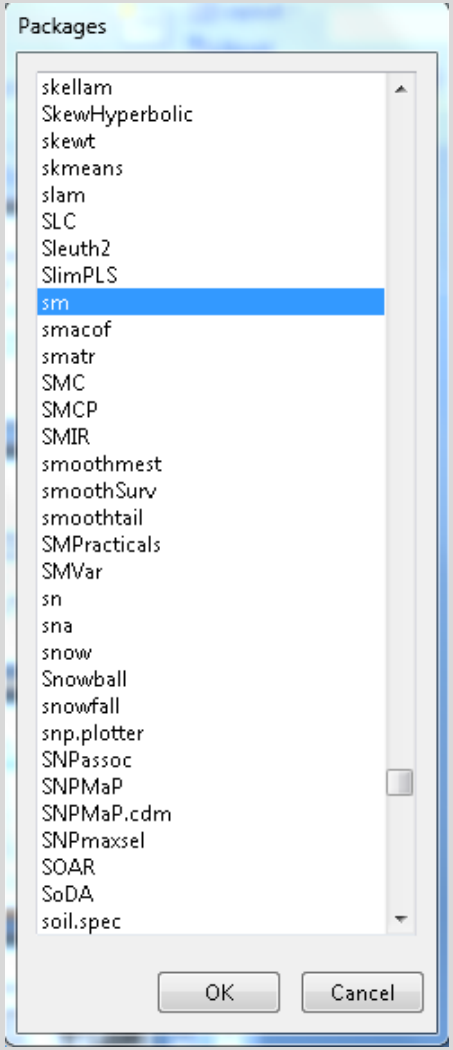
```
library(sm)
# create value labels
zyg.f <- factor(zyg, levels= seq(1,5),
  labels = c("MZF", "MZM", "DZF", "DZM", "DZOS"))

# plot densities
sm.density.compare(age, zyg, xlab="Years")
title(main="Years by ZYG")

# add legend
colfill<-c(2:(2+length(levels(zyg.f))))
legend(.8,3, levels(zyg.f), fill=colfill)
```

Huh what?

```
> library(sm)
Error in library(sm) : there is no package called 'sm'
> sm.density.compare(age, zyg, xlab="Years")
Error: could not find function "sm.density.compare"
```

Adding a package...

install.packages()

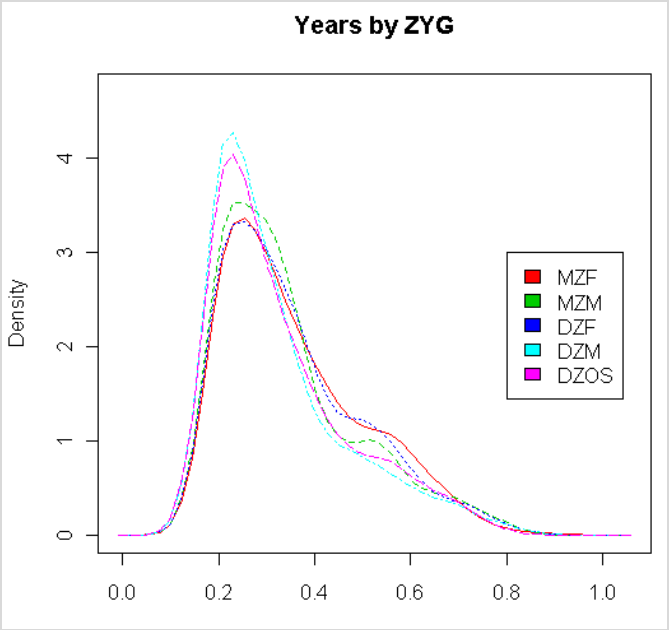


```
> install.packages()
--- Please select a CRAN mirror for use in this session ---
Warning: unable to access index for repository http://www.stats.ox.ac.uk/pub/RW$
Warning in install.packages() :
  argument 'lib' is missing: using 'C:\Users\Indigo\Documents/R/win-library/2.1$
trying URL 'http://www.ibiblio.org/pub/languages/R/CRAN/bin/windows/contrib/2.1$
Content type 'application/zip' length 341341 bytes (333 Kb)
opened URL
downloaded 333 Kb

package 'sm' successfully unpacked and MD5 sums checked

The downloaded packages are in
        C:\Users\Indigo\AppData\Local\Temp\Rtmpr6KlMN\downloaded_packages
```

Looking at your data…

#Kernal density plot by zyg?

```
library(sm)
# create value labels
zyg.f <- factor(zyg, levels= seq(1,5),
  labels = c("MZF", "MZM", "DZF", "DZM", "DZOS"))

# plot densities
sm.density.compare(age, zyg, xlab="Years")
title(main="Years by ZYG")

# add legend
colfill<-c(2:(2+length(levels(zyg.f))))
legend(.8,3, levels(zyg.f), fill=colfill)
```



That's great but how do I save it?

```
# make a png file to hold the plot
png("zygdensity.png")

# create value labels
zyg.f <- factor(zyg, levels= seq(1,5),
  labels = c("MZF", "MZM", "DZF", "DZM", "DZOS"))
# plot densities
sm.density.compare(age, zyg, xlab="Years")
title(main="Years by ZYG")
# add legend via mouse click
colfill<-c(2:(2+length(levels(zyg.f))))
legend(.8,3, levels(zyg.f), fill=colfill)

# close the png file to allow viewing
dev.off()
```

pdf("mygraph.pdf")	pdf file
win.metafile("mygraph.wmf")	windows metafile
png("mygraph.png")	png file
jpeg("mygraph.jpg")	jpeg file
bmp("mygraph.bmp")	bmp file
postscript("mygraph.ps")	postscript file

Final Words of Warning

"Using R is a bit akin to smoking. The beginning is difficult, one may get headaches and even gag the first few times. But in the long run, it becomes pleasurable and even addictive. Yet, deep down, for those willing to be honest, there is something not fully healthy in it." --Francois Pinard

explodingdog.com

Time for coffee