A photograph of a curved glass facade of a modern building. The glass reflects the sky and surrounding structures, creating a distorted, curved view of the world. The text is overlaid on the upper right portion of the image.

(Re) introduction to Linux
Sarah Medland
Boulder 2016

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

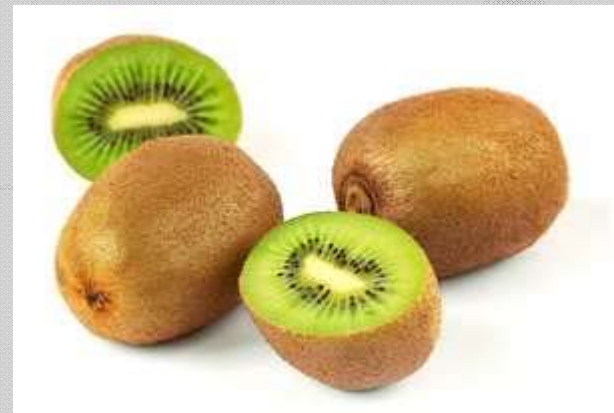


Sarah Medland





not



Morning sessions

- Optional
 - Feel free to wander in and out/check email etc
- Topics
 - Shift in response to feedback
 - Tomorrow: R intro
 - Wednesday: PCA
 - Thursday: Family Based association
 - Friday: TBA



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
 - ctrl+C for copy ctrl+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

A photograph of a modern building's curved glass facade. The glass reflects the sky and surrounding environment. A semi-transparent white rectangular box is overlaid on the lower half of the image, containing the text "Superfast intro to R".

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....

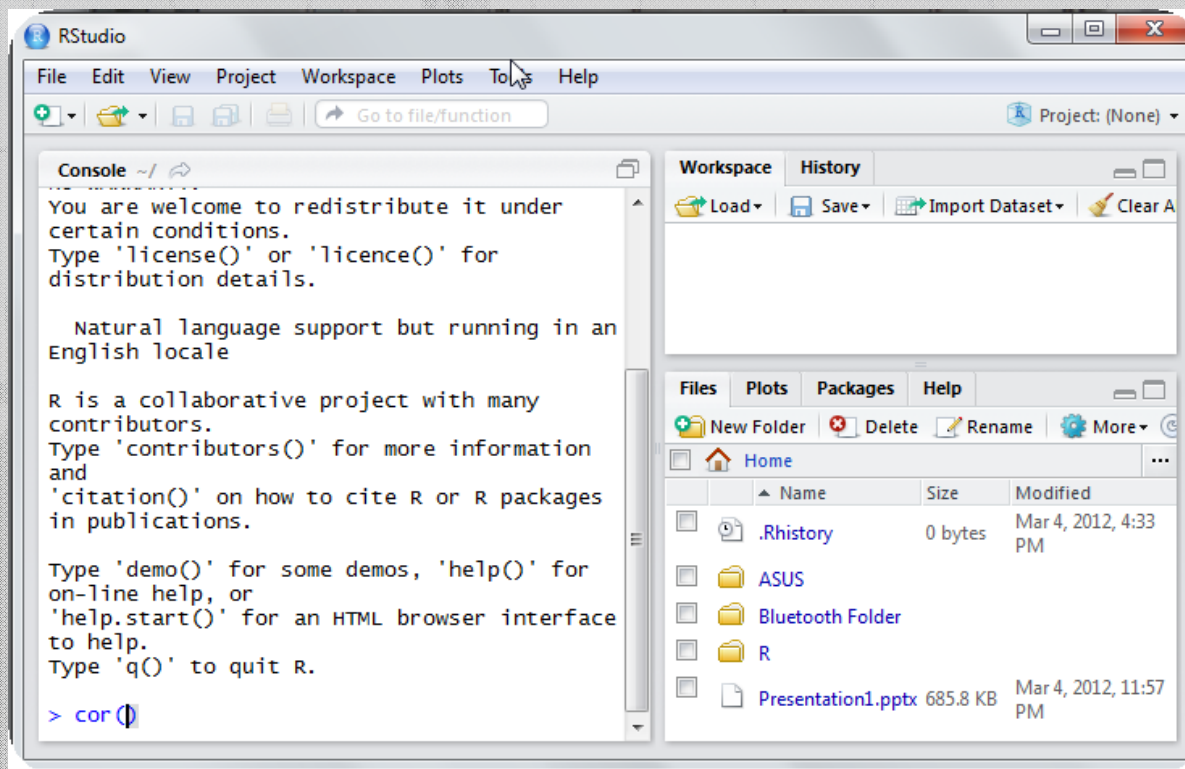


R-help listserve....



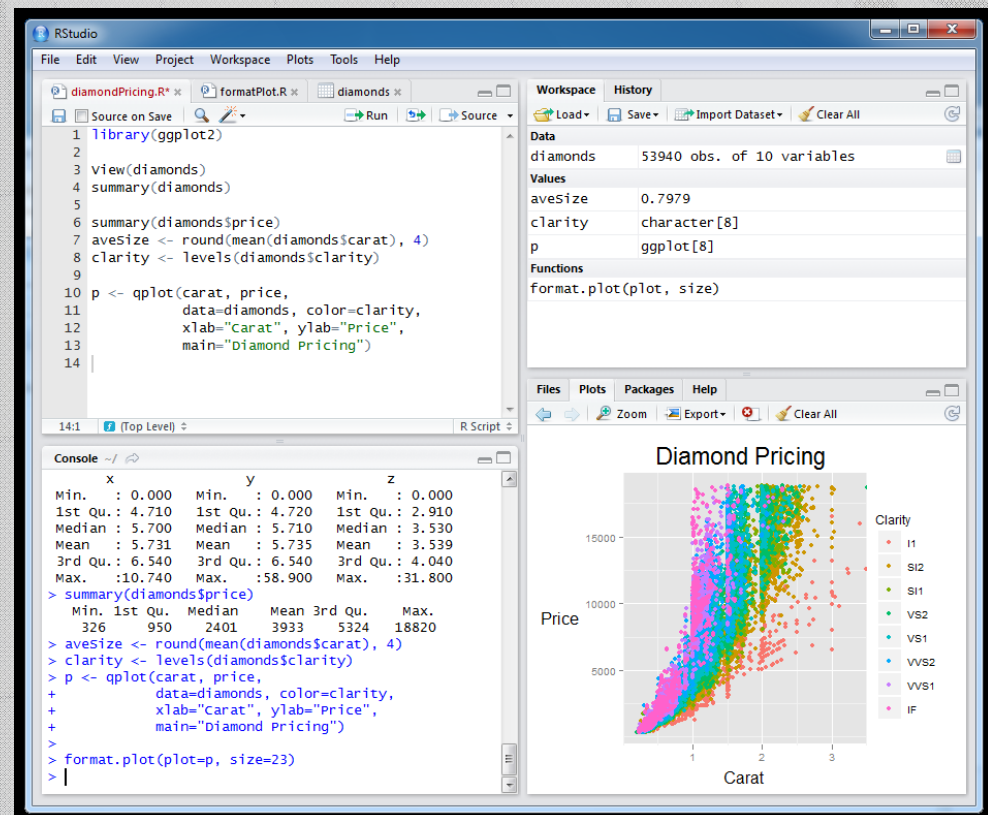
Using R this week

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



Setting this up at home

- Install R first
- Install R studio
- Install packages



Start up R via R studio

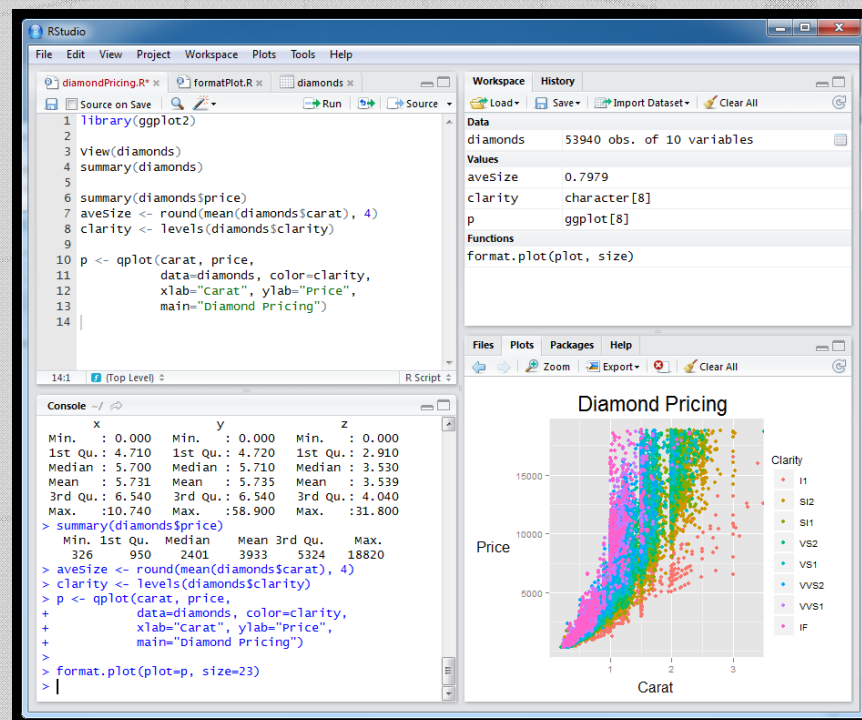
4 windows:

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*

The screenshot displays the RStudio interface with three main panels:

- Source Editor:** Contains an R script with the following code:

```
1 library(ggplot2)
2
3 view(diamonds)
4 summary(diamonds)
5
6 summary(diamonds$price)
7 aveSize <- round(mean(diamonds$carat), 4)
8 clarity <- levels(diamonds$clarity)
9
10 p <- qplot(carat, price,
11            data=diamonds, color=clarity,
12            xlab="carat", ylab="Price",
13            main="Diamond Pricing")
14
```
- Console:** Shows the execution output:

```
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)
> p <- qplot(carat, price,
+           data=diamonds, color=clarity,
+           xlab="carat", ylab="Price",
+           main="Diamond Pricing")
>
> format.plot(plot=p, size=23)
>
```
- Plots Panel:** Displays a scatter plot titled "Diamond Pricing". The x-axis is labeled "Carat" (ranging from 1 to 3) and the y-axis is labeled "Price" (ranging from 0 to 15000). The data points are colored by clarity, with a legend on the right showing categories: I1, SI2, SI1, VS2, VS1, VVS2, VVS1, and IF.



GETTING STARTED

How to use help in R?

- R has a help system built in.
- If you know which function you want help with simply 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")`.

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
```

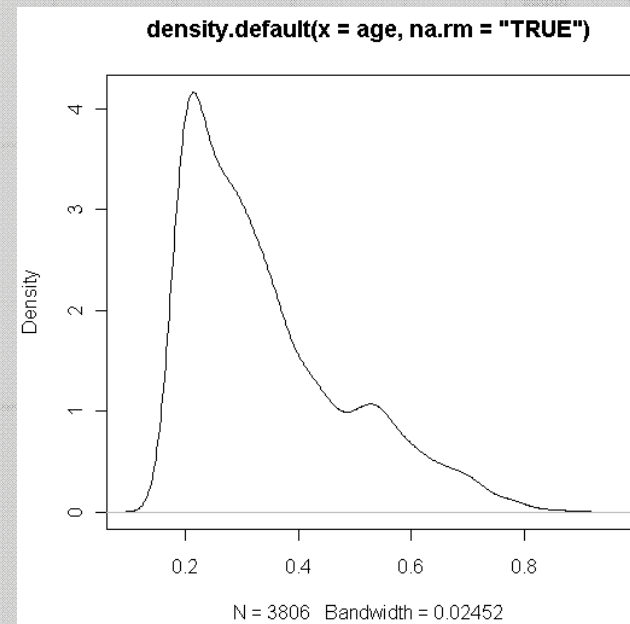
```
hist(age, breaks=12, col='red', xlab='age in  
years', main='Histogram of age')
```

Looking at your data...

#Kernel density plot

**d <- density(age, na.rm = "TRUE") # returns the
density data**

plot(d) # plots the results



Looking at your data...

#Kernel 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)
```

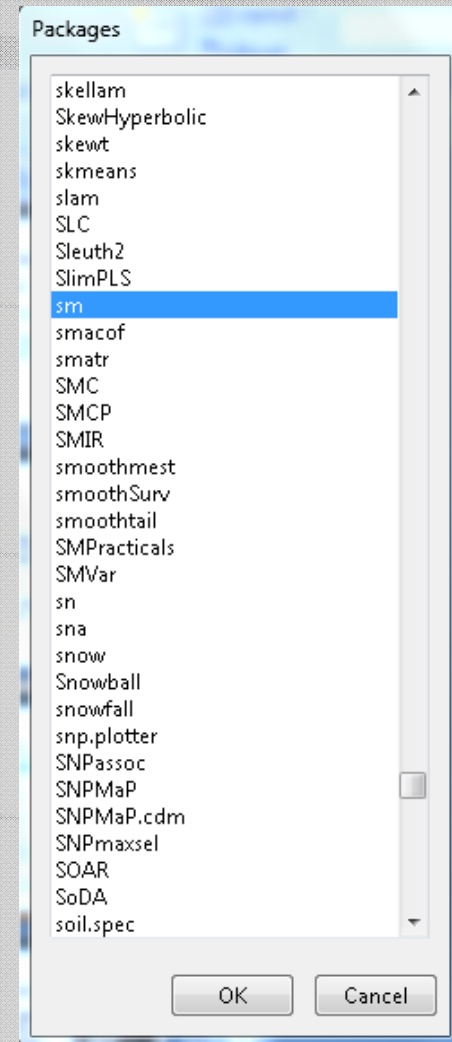
```
Error in library(sm) : there is no package called 'sm'
```

```
> 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\RtmpR6K1MN\downloaded_packages
```

Looking at your data...

#Kernel 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
```

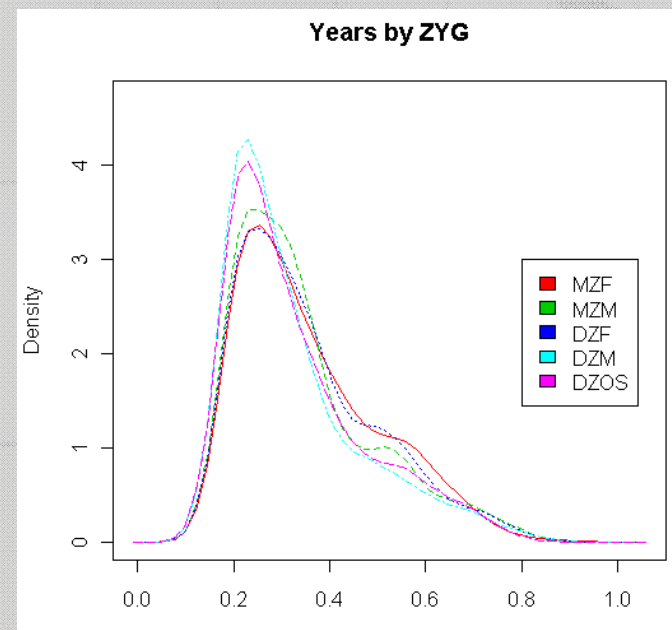
```
sm.density.compare(age, zyg, xlab="Years")
```

```
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()
```

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



Time for coffee





Vienna