Introduction to R

Part II

Part II

Please interrupt with questions

Review

Working with Scripts and Writing Functions

Libraries and Packages

More Graphics

Working with Scripts and Writing Functions

R scripts

Scripts are a little different from libraries

They are simple text files and don't follow any special requirements

Write scripts for often used procedures

Scripts are saved as .r or .R files

To load a script

> source("myScript.R")

Last time we wrote a script for reading in numeric files, but we copied and pasted the code

Now let us use `source()`

Making a function

Right now you have to go into the script and enter the file name

Everytime you load a data file it will overwrite myData within the R console

Let's make the process more efficient by the process a function

What should the input variable be?

What should the output be?

```
FuncName <-
function(input1,input2,...){

    DO STUFF

    return(output)

}
```

Open your editor and copy the script into an empty file

C:\Users\juli\Desktop\INBRE_workshops\Functions.R - Notepad++

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

GradStudent.txt | petalRun.r | petalRun.r | Histogram.R | UploadingFile.R | controltreatment_sample.txt | Functions.R

```r


readinTabText <- function(fileName){
# make sure the file is in table format
# line 1 should have the same length as line 2

    m1 = scan(fileName, sep = "\t", what = "", nlines = 1)
    m2 = scan(fileName, sep = "\t", what = "", nlines = 1, skip = 1)

    tmpM= matrix(scan(fileName, what = "", sep = "\t", skip = 1), ncol = length(m1), byrow = TRUE)
    myData= matrix(as.numeric(tmpM[, -1]), ncol = length(m1) -1, byrow = FALSE)
    colnames(myData) = m1[-1]
    rownames(myData) = tmpM[, 1]

    # check if the file was read in correctly
    print(paste("Dimension of input file: ", dim(myData)[1], " by ", dim(myData)[2], sep=""))
    return(myData)
}

# myData is a numeric matrix object
```

R programming language | length : 688 lines : 20 | Ln : 17 Col : 19 Sel : 0 | 0 | Dos\Windows | ANSI as UTF-8 | INS

Libraries and Packages

Libraries

The basic R functions can be extended by libraries or packages

Packages on CRAN and Bioconducter have to follow specific regulations

Packages outside of these archives can still be valuable, but possibily do not follow CRAN's standard

To install and use a package

```
> install.packages("<PackageName>")
```

```
> library("<PackageName>")
```

Let us install ggplot2 - a powerful graphics library

```
> install.packages("ggplot2")
```

R will prompt you to select a mirror, I often chose CA1

After installaing the library, it has to be loaded to enable its functions

```
> library(ggplot2)
```

Now all of ggplot2's functions are available

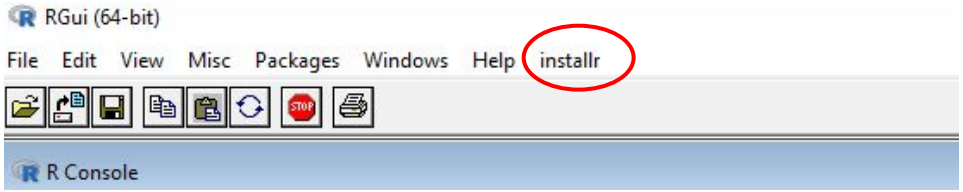
Updating R on Windows

```
> install.packages("installr")

> library(installr)

> updateR()
```

Click on `installr`

Select `Update R`

Go through Windows' Wizard Setup



Updating Libraries

Just like R, libraries consistently upgrade as well

To check which R and library versions you are currently using

```
> sessionInfo()
```

To update R libraries

```
> update.packages()
```

R Console

```
>
>
> sessionInfo()
R version 3.3.1 (2016-06-21)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 14393)

locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] ggplot2_2.2.1

loaded via a namespace (and not attached):
 [1] colorspace_1.3-2 scales_0.4.1     assertthat_0.1   lazyeval_0.2.0
 [5] plyr_1.8.4       tools_3.3.1      gtable_0.2.0     tibble_1.2
 [9] Rcpp_0.12.9      grid_3.3.1       munsell_0.4.3
>
```

More Graphics

Lines, Venn Diagrams

Library: ggplot2

```
> library(ggplot2)
```

Set the file name to "OldvsYoung.txt"

```
> source("UploadingFile.R")

> plot(myData[1,], myData[2,])
> colnames(myData)
> group = c(rep("Old",18), rep("Young",19))
```

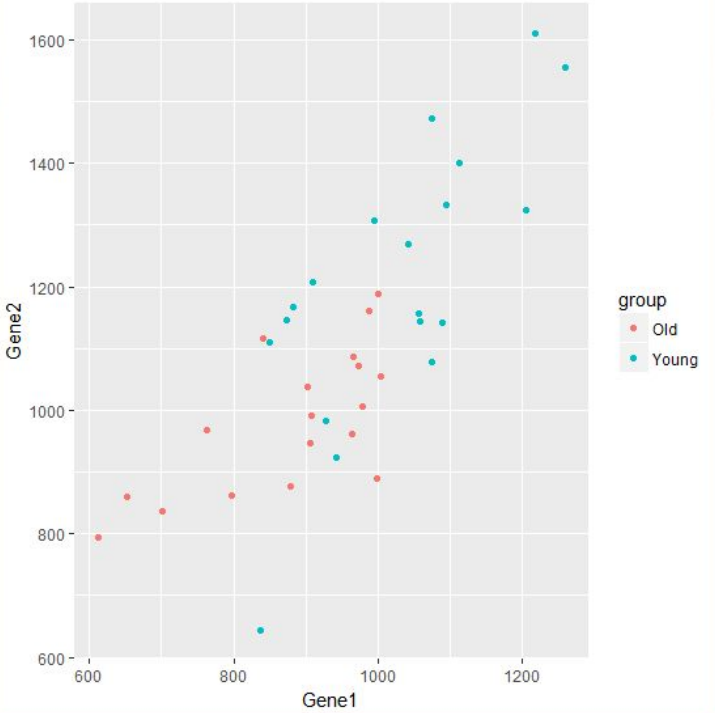
Resources:

http://ggplot2.org/book/

http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html

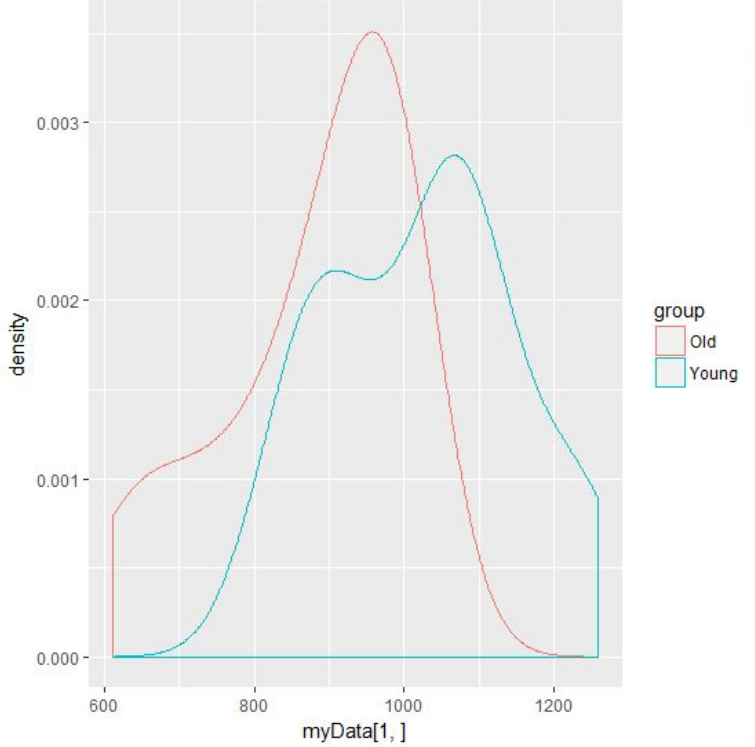
Library: ggplot2

```
> qplot(myData[1,], myData[2,], colour=group, xlab="Gene1", ylab="Gene2")
```

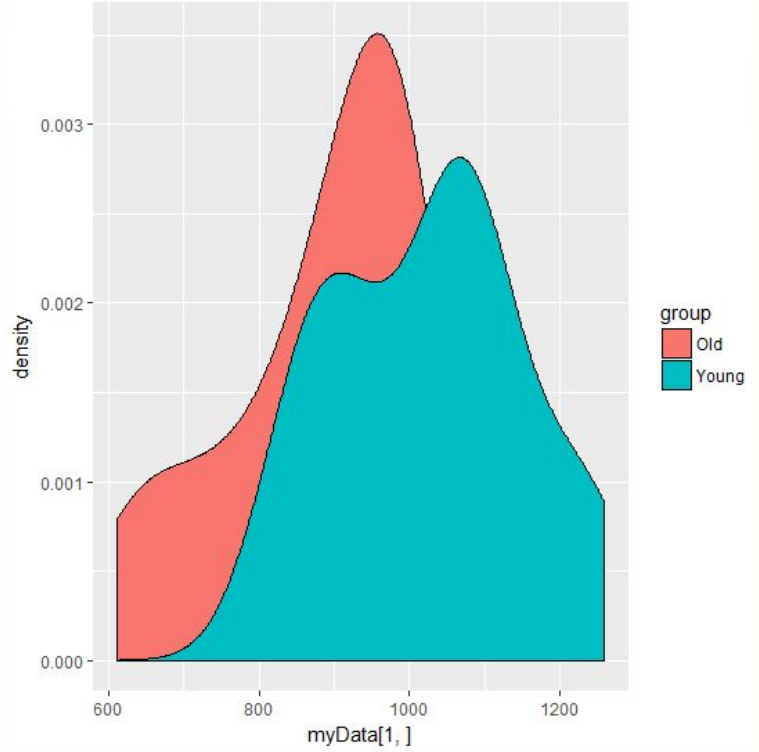


Library: ggplot2

```
qplot(myData[1,], geom="density", colour=group)
```

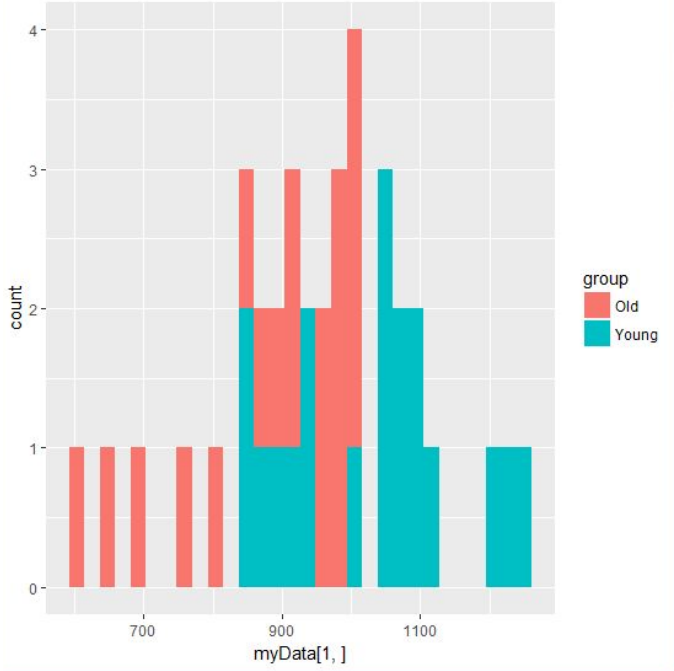


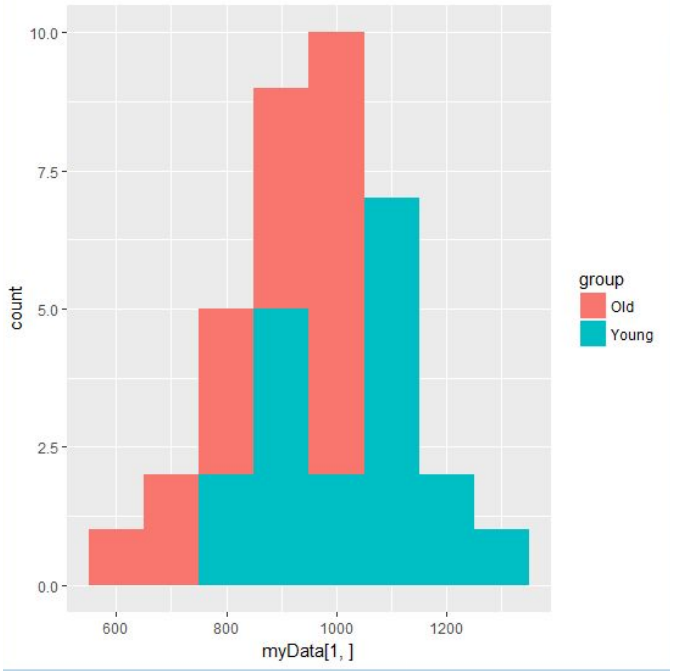

```
qplot(myData[1,], geom="density", fill=group)
```



Library: ggplot2

```
qplot(myData[1,], geom="histogram", fill=group)
qplot(myData[1,], geom="histogram", fill=group, binwidth=100)
qplot(myData[1,], geom="histogram", fill=group, binwidth=100, xlab=rownames(myData)[1])
```



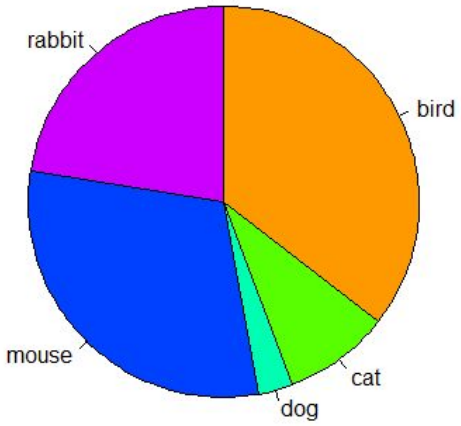


Plotting Lines script

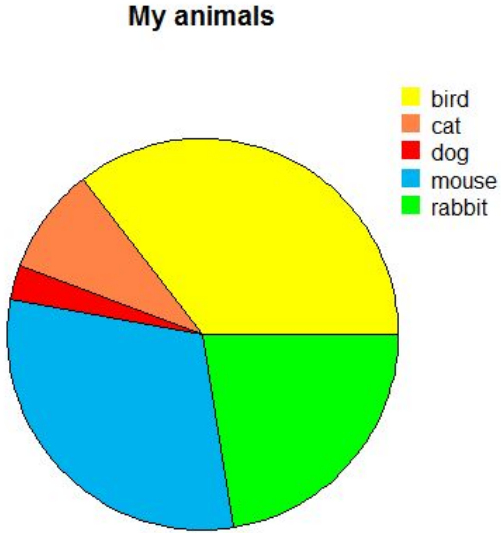
```
# Plots a line graph for all rows in data frame 'myData'
# 'split.screen()' function is used to overlay several line graphs in the same plot

MIN = round(min(as.vector(myData))-0.5,digits=0)
MAX = round(max(as.vector(myData))+0.5,digits=0)

split.screen(c(1,1))
plot(myData[1,], ylim=c(MIN,MAX), xlab="Measurement", ylab="Intensity", type="l", lwd=2, col=1)
for(i in 2:length(myData[,1])){
    screen(1, new=FALSE)
    plot(myData[i,], ylim=c(MIN,MAX), type="l", lwd=2, col="blue", xaxt="n", yaxt="n", ylab="",
xlab="", main="", bty="n")
    }
screen(1, new=FALSE)
plot(rep(500, 37), ylim=c(MIN,MAX), type="l", lwd=2, col="red", xaxt="n", yaxt="n", ylab="",
xlab="", main="", bty="n")
close.screen(all=TRUE)
```

Pie Chart

```
> animal = c("cat", "mouse", "dog", "bird", "rabbit")
> count = c(109,378,36,443,280)
> inputT = table(rep(animal,count))

> pie(inputT, col=rainbow(length(animal), start=0.1, end=0.8),
clockwise=T)
```



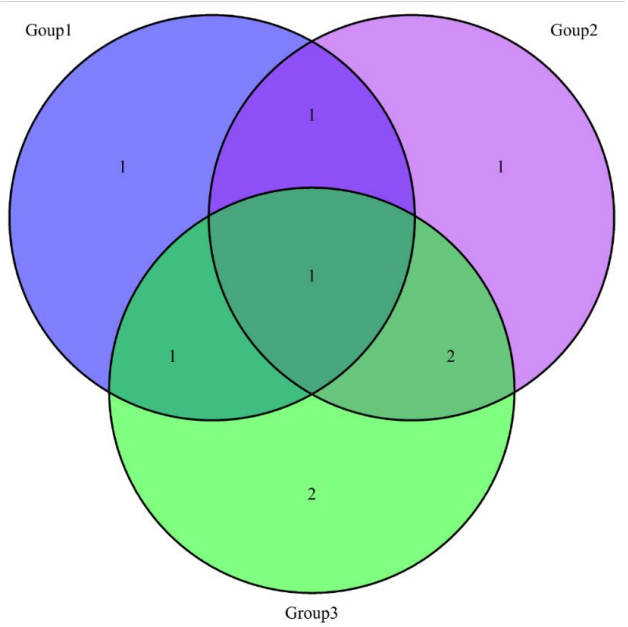

```
> myColors = c("yellow1", "sienna1", "red1", "DeepSkyBlue2",
"green")

> pie(inputT, col=myColors, labels=NA, main="My animals")
> legend("topright", legend=row.names(inputT), cex=1, bty="n",
pch=15, pt.cex=1.8, col=myColors, ncol=1)
```



Venn Diagram

```
> install.packages("VennDiagram")
```

```
> g1 = c("Marie", "Claus", "Stef", "Anna")

> g2 = c("Tom", "Claus", "Laura", "Stef", "Betty")

> g3 = c("Marie", "Tom", "Claus", "Clair", "Laura", "Beth")
```




```
> venn.diagram(list(G1 = g1,G2 = g2), fill=c("blue", "purple"), filename="Venn2sets.png")

> venn.diagram(list(Goup1 = g1,Goup2 = g2, Group3 = g3), fill=c("blue", "purple",
"green"), filename="Venn3sets.png")
```

Handy functions

```
> objects()
```
does the same as `ls()`

```
> rm()

> typeof(x)

> class()

> is.numeric()        > as.numeric

> is.null()

> is.matrix()         > as.matrix
```