Introduction to R Workshop
Presenters: Jon Wayland and Jeremy Yagle
Spring 2013

R is a free software downloadable at http://www.r-project.org/

Notes:	Code and Output:
01 R Console Setup: **>** prompts you for formula or function. The result appears on the next line(s).	
02 Current Working Directory Use the getwd() function to see your current working directory. You can manually change your working directory: "file" pull-down menu → "change dir…" option .	**> getwd()** [1] "C:/Windows/system32"
03 Warning R is case sensitive!	
04 Comments begin with # Anything on the line after the # will not be executed by R.	**> # This is a comment!**
05 Note: Rather than typing the R commands , you can copy the entire contents of the box and use the Edit → "Paste commands only" option in R You can also use the ↑ key to see and edit earlier commands	
06 Basic Operations: Addition /Subtraction: + and – Multiplication /Division: * and / Exponents: ^ Square root: Sqrt()	**> 1+2 #Simple Addition** [1] 3 **> 2*3 #Multiplication** [1] 6 **> 8/4 #Division** [1] 2 **> 2^3 #Exponent** [1] 8 **> (4*12)/(2*6) #Mixed Operations** [1] 4
07 Example of Basic Formula: The formula for calculating a person's BMI is $BMI = \frac{Weight*703}{Height^2}$ where weight is measured in pounds and height is measured in inches. Calculate the BMI for a 118 lb person who is 64.5 inches tall.	**> (118*703)/(64.5^2)** [1] 19.93967
08 Vector A data structure in R consisting of a collection of all numeric (or all character) values where the position in the vector is important . So 165 170 175 is different than 165 175 170	

09 Example Generating Vectors:

We have three people with the corresponding weights and heights:

Person	1	2	3
Weight	165	170	175
Height	56	58	60

We can create two vectors,
* w= the weights of the three subjects
* h= the heights of the three subjects
This can be done in two different ways:
using the c() function and using the scan() function.

For the **scan()** function used creating **h**, values are separated by spaces or going to the next line.

```
> w=c(165,170,175)
> w
[1] 165  170  175
> h=scan() #The values below are part of the command
56 58
60

> h
[1] 56  58  60
```

10 Vectors of character values

For the scan() command, what="" indicates that the values are character.

If values can be more than one word, put one value per line and use the sep="\n" to indicate that values end at the end of the line.

```
> g=c("m", "f", "m")
> g
[1] "m" "f" "m"
>state=scan(what="", sep="\n")
Pennsylvania
New York
West Virginia

> state
[[1] "Pennsylvania"  "New York"  "West Virginia"
```

11 Example Showing Use of Vectors:

From this we can calculate each person's BMI in a single step. This creates a new object that stores these values under bmi.

```
> bmi=(w*703)/(h^2)
> bmi
[1] 36.98820  35.52616  34.17361
```

12 Sub-setting Vectors

We can find the height, weight or BMI of a specific person.

Find the height of Person 2

Find the weight of Person 3.

We can find these values for more than one person at a time.

Find the BMI of Person 1 and Person 3.

```
> h[2]
[1] 58
> w[3]
[1] 175

> bmi[c(1,3)]
[1] 36.98820 34.17361
```

13 Data Frames

A two-dimensional R data structure used for individuals with more than one variable. Each row in the data frame contains the information for one individual and each column is a variable.

Data frames allow us to display multiple records and multiple variables in a tabular format.

14 Example Showing Use of Data Frames:

Let's store the heights, weights and BMI's together in a data frame.

```
> DFbmi=data.frame(h,w,bmi)
> DFbmi
   h   w       bmi
1 56 165 36.98820
2 58 170 35.52616
3 60 175 34.17361
```

15 Sub-setting data frames

You can subset data frames.

Find all information for the second person:

```
> DFbmi[2,]
   h   w      bmi
2 58 170 35.52616
```

Find the heights and weights for the second and third person:

```
> DFbmi[c(2,3),c(1,2)]   # or DFbmi[c(2,3),c("h","w")]
   h   w
2 58 170
3 60 175
```

16 Importing Data Files

Instead of creating vectors and combining them into a data frame, we can input existing data in tabular format.

17 Example of Reading in Data:

The comma-delimited excel file "body.csv" contains the Gender, height, and weight of 40 individuals (20 males and 20 females) . We input the file into R in two ways. The first way allows you to search for the file you want to work with.

The second way works only if the file is in your working directory.

```
> file=file.choose()
> body=read.csv(file)

> body=read.csv("body.csv")
```

18 Basic Data Frame Information

The **names()** function gives the names of the variables.

The **head()** function shows the names, as well as the first six values of each variable.

```
> names(body)
[1] "Gender"  "Weight"  "Height"
> head(body)
   Gender  Weight  Height
1  Female      118    64.5
2    Male       NA    72.5
3    Male      143    73.3
4    Male      172    68.8
5  Female      147    65.0
6  Female      146    69.0
```

19 Using Variables in Data Frames

You can call any variable in a data frame using the name of the data frame followed by **$** followed by the variable name.

```
> body$Gender
 [1] Female Male   Male   Male   Female Female Female
 [8] Female Male   Male   Female Male   Male   Female
[15] Female Female Female Male   Female Male
[21] Male   Male   Female Male   Female Male   Female
[28] Male   Female Female Female Male   Male   Male
[35] Female Female Male   Female Male   Male
Levels: Female Male
```

20 Attaching and Detaching

Using the **attach** function tells R that we will be working with this data set until we **detach** it.

Note: By using the attach(body) function, you can call the variables by their variable names without including the name of the data frame
i.e. Weight rather than body$Weight

```
> attach(body)
```

21 Example

Let's calculate the BMI of everyone in this data set. Let's also round these values to a single decimal place.

```
> BMI=(Weight*703)/(Height^2)
> BMI=round(BMI,digits=1)

> head(BMI)
[1] 19.9   NA 18.7 25.5 24.5 21.6
```

22 Adding The BMI Vector To The Current Data Frame:

Now add this vector of newly calculated BMI's to our data frame. Since we are altering the data frame, let's name this altered data set **newbody**.

Missing Values

Notice that some of the values in our data set are "NA". This means that they are missing values. If there are missing values for the Weight or Height, our BMI will be missing as well since our BMI depends on the Weight and Height.

```
> newbody=data.frame(body,BMI)
> head(newbody)
   Gender Weight Height  BMI
1  Female    118   64.5 19.9
2    Male     NA   72.5   NA
3    Male    143   73.3 18.7
4    Male    172   68.8 25.5
5  Female    147   65.0 24.5
6  Female    146   69.0 21.6
```

23 Descriptive Statistics:

To find the average height, weight or BMI, we can use the **summary()** command in R.
This also provides the number of missing data values for each variable. Also notice that our data set has an equal number of males as it does females. The Mean shows the average for each variable.

```
> summary(newbody)
```

```
   Gender       Weight          Height          BMI       
 Female:20   Min.   :106.0   Min.   :62.00   Min.   :18.70  
 Male  :20   1st Qu.:135.2   1st Qu.:66.00   1st Qu.:20.70  
             Median :146.5   Median :68.00   Median :22.30  
             Mean   :151.1   Mean   :68.53   Mean   :22.63  
             3rd Qu.:172.0   3rd Qu.:70.50   3rd Qu.:23.68  
             Max.   :192.0   Max.   :77.00   Max.   :29.90  
             NA's   :  2.0   NA's   : 1.00   NA's   : 2.00
```

To find the mean BMI score for this data set, we need to account for the missing values. The **na.rm** function removes those missing values.

```
> mean(BMI)
[1]NA

> mean(BMI,na.rm=T)
[1]22.62632
```

Find the standard deviation of the BMI's.

```
> sd(BMI,na.rm=T)
[1] 2.5721
```

24 Finding Descriptive Statistics of Subsets of Individuals

We can also find the mean BMI, height or weight for males or females. Let's find the mean height for males.

```
> mean(Height[Gender=="Male"],na.rm=T)
[1] 71.43158
```

Now let's find the mean BMI for both males and females separately. As we can see, the average BMI for the males in this particular data set is higher than the average BMI for the females.

```
> tapply(BMI, Gender, mean, na.rm=T)

Female     Male
22.275  23.01667
```

Let's see how many people have a BMI above 25 (overweight or obese).
Notice there are 7 people who are categorized as overweight or obese.

```
> sum(BMI>25,na.rm=T)
[1] 7
```

Let's also see what the BMI values above 25.

```
> BMI[BMI>25]
[1]   NA 25.5 28.2 25.5 25.8  NA 25.3 29.9 26.7
```

To exclude the missing values

```
> BMI[BMI>25 & !is.na(BMI)]
[1] 25.5  28.2  25.5  25.8  25.3  29.9  26.7
```

Note: Logical Operators

Less than	< than or = to	Exactly equal to	Not equal to	Not	and	Or
<	<=	==	!=	!	&	|

Let's see the complete records for individuals with a BMI exceeding 25.

Note the comma inside the subsetting brackets

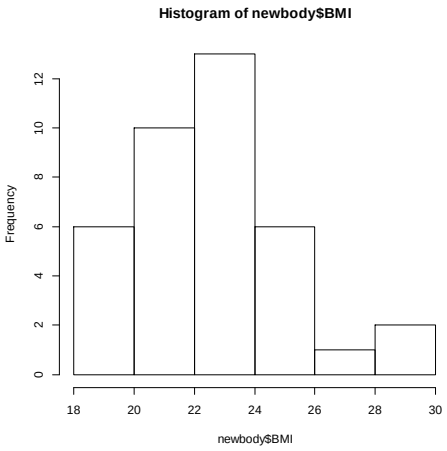
Before the comma: records
After the comma: variables

> newbody[BMI>25,]

	Gender	Weight	Height	BMI
NA	<NA>	NA	NA	NA
4	Male	172	68.8	25.54506
8	Female	175	66.0	28.24265
10	Male	172	68.8	25.54506
20	Male	180	70.0	25.82449
NA.1	<NA>	NA	NA	NA
30	Female	159	66.5	25.27605
32	Male	191	67.0	29.91156
34	Male	181	69.0	26.72611

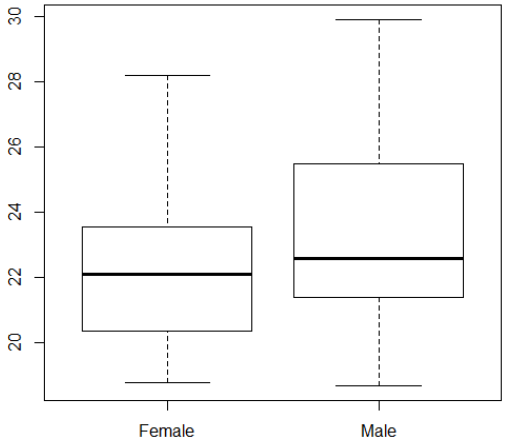
25 Graphical Interpretation:
Now let's look at the BMI's graphically using a histogram.

> hist(BMI)

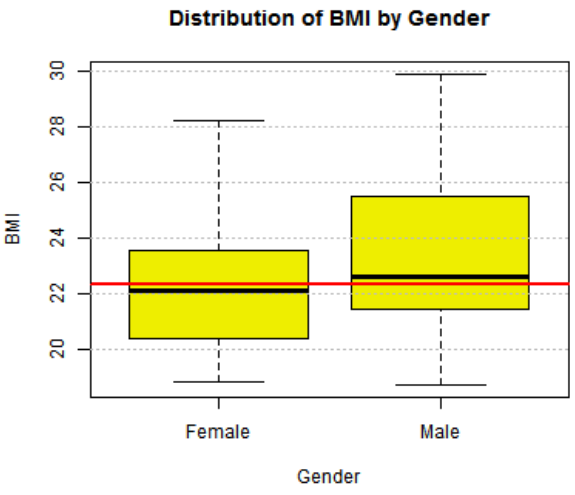


Let's make a **boxplot** of the BMI's for males and females and compare. Here we can verify that BMI tends to be a little higher for males than for females.

> boxplot(BMI~Gender)



Or even better: (Come to the R Workshop in Graphs!)



>boxplot(BMI~Gender, main="Distribution of BMI by Gender", xlab="Gender", ylab="BMI", col="yellow2")

>abline(h=median(BMI, na.rm=T), col="red", lwd=2)

>grid(NA, NULL, col="grey")

26 Packages There are thousands of functions and datasets that are accessible through packages. For example, the MASS package comes with R but can only be accessed if the library command is used. Other packages can be installed and then activated through the **library ()** command. See documentation on MASS at http://cran.r-project.org/web/packages/MASS/MASS.pdf	**> Animals** Error: object 'Animals' not found **> library(MASS)** **> Animals**
27 Getting Help: Use the **help()** function to see documentation on an R function, including a list of function arguments. Must have internet access.	**> help(summary)**
28 Saving Your Workspace Objects Select "File" and then "Save to file" which allows you to save this as a text file. R will also prompt you to save your workspace when you quit R	
29 Saving Your Workspace Commands and Results You may also save your workspace (on your h drive!) and save it as a Word Document or Note Pad (recommended). This will show commands and results. It will not save your data objects. You can later copy it and use the "Paste commands only" option in R, but if you used the scan() function to input data, it will not execute properly.	
30 Saving and loading an individual object To save objects as a .RData file, use the save() command. You can move this .RData file to the working directory on another computer and load the objects there.	**> Save(Animals, file="Animals.RData")** **> load("Animals.RData")**
31 The history() function Opens up a "R History window with the most recent commands that you submitted. This will not show the results and it does not include the R prompts.	> history(max.lines=60)
32 Quit R When you are finished with R, you can either exit out manually through the "File" pull down menu or you can use the exit function **q()**.	**>q()**

Appendix the body data set

Gender	Weight	Height
Female	118	64.5
Male		72.5
Male	143	73.3
Male	172	68.8
Female	147	65
Female	146	69
Female	138	64.5
Female	175	66
Male	134	66.3
Male	172	68.8
Female	118	64.5
Male	151	70
Male	155	69
Female	155	70.5
Female	146	66
Female	135	68
Female	127	68.5
Male	178	73.5
Female	136	66.3
Male	180	70
Male		
Male	186	76.5
Female	122	62
Male	132	68
Female	114	63
Male	171	72
Female	140	68
Male	187	77
Female	106	63
Female	159	66.5
Female	127	62.5
Male	191	67
Male	192	75.5
Male	181	69
Female	143	66.5
Female	153	66.5
Male	144	70.5
Female	139	64.5
Male	148	74
Male	179	75.5