R Basics

I. Calculator

A. Example

```
2+2
exp (-2)
rnorm (15)
```

B. Symbolic variables

```
x <- 2
x +x
```

1. Variable names
   a. Built from letter, digits, and period
   b. Do not start with period- special case – reserved

C. Data vectors

```
Weight <-  c(60, 72, 57, 90, 95, 72)
```

c (…) is construct

```
Weight
sum (Weight)
```

Calculator functions for calculating the mean:

```
sum (weight)/length (weight)
```

```
mean(weight)
sd(weight)
```

```
Height <- c(1.75, 1.80, 1.65, 1.90, 1.74, 1.91)
```

II. Vocabulary

A. Function Calls

```
log(x)   # log is a function
plot (Height, Weight)  # plot is a function
```

B. Arguments

```
log(x)   # x is an  argument
plot (Height, Weight)   # height and weight are arguments
plot (Height, Weight, pch=2)   # pch=2 – named actual argument
```

C. Vectors

1. Character vectors

   Names <- c (“Huey”, “Dewey”, “Louie”)

   Names

2. Logical vectors

   State <- c(T,T,F,T)

   State

D. Matrix and Arrays

1. Matrix- 2d vector
2. Array- 3d set of vectors
   a. I encourage you to import your matrix from excel so we will not be discussing how to construct a matrix in R.
   b. In most programming a 2d vector is called an array – confusing
3. Data frame- a data matrix or data set in other programs
   a. Data and labels

II. Missing Values

1. NA

III. Help

?<Function name>

? plot

help (plot)

help.search (“graph”) – when you do not know the function name

example (plot)

IV. Comments

# anything after the number sign is skipped – serves as a comment

V. Install a package

Demonstration

VI. Load a package

Library (ISwR)

VII. Set workspace

A. Make a directory called “Multi” on your drive

   Example: C:\Working\Multi

B. Make a directory in the “Multi” directory called “Lab02”

   Example: C:\Working\Multi\Lab02

VIII. Go to the lab webpage and download the data into the Lab2 directory that you just created

- A. Note the different links
  - 1. Directly to data
  - 2. To a linked webpage
- B. <u>FOR DIRECT LINKS:</u> Right click on the link and save the linked file to C:\Working\Multi\Lab02
- C. <u>FOR LINKED WEBPAGES:</u> Left click on the link to go to the next page and then right click on the data link and save the linked file to C:\Working\Multi\Lab02

IX. Edit data in Excel and export it for R

- A. Open the dataset Men.xlsx using Excel
- B. Inspect the dataset
  - 1. Case in rows, variables in columns
  - 2. First row variables name (remember rule for variable names)
    - a. No spaces
    - b. Names should be descriptive
    - c. Short names are better
  - 3. First column case names
  - 4. Missing values are NA
  - 5. Clean/delete empty space - that MIGHT have had data prior
- C. Export data as a tab- delimited
  - i. File → save as → save as type
    - a. Select text (tab delimited)
  - ii. Close Excel

X. Setting the working directory

- A. Start R
- B. Example
- C. `setwd("C:\\Working\\ Multi\\Lab2\\")`

XI. Importing dataset into R

```
Men <- read.table (file=" C:\Working\Multi\Lab02\\Men.txt", header = TRUE)
```

XII. Structure function

```
str(Men)
```

Checkup: Import the Energy.txt dataset into R. Then apply the structure function to confirm proper import. Call the file/dataset in R Energy. Ask for help if you have problems.

XIII. Data Frame and related functions

A. $ sign

```
Energy$expend
```

B. Unique function

```
unique (Energy$stature)
```

C. Subsetting data

```
ObeseL <- Energy$stature == "obese"
ObeseL
Obese <- Energy [ObeseL, ]
Obese
```

D. Joining data sets

1. Import part 1 name part1

   ```
   part1 <- read.table("C:\\Multi\\part1.txt", header =TRUE)
   ```

2. Import part 2 name part2

   ```
   part2 <- read.table("C:\\Multi\\part2.txt", header =TRUE)
   ```

3. Merge the two data sets based on variable “Sample”

   ```
   Whole <- merge (part1, part2, by = "Sample")
   ```

E. Exporting data (Explain line wrap in this portion)

```
write.table (Whole, file = "C:\\Working\\Multi\\something.txt", sep=",",
quote=TRUE, append = FALSE, na = "NA")

write.table (Whole,
file = "C:\\Multi\\something.txt",
sep = ",",
quote = TRUE,
append = FALSE,
na = "NA")
```

F. Basic function

1. Import file Vegetation.txt in into matrix “Veg”

   ```
   names (Veg)
   str(Veg)
   m <- mean(Veg$R)
   m

   m1 <- mean(Veg$R[Veg$Transect == 1])
   ```

   Show how to backspace to modify

   ```
   m2 <- mean(Veg$R[Veg$Transect == 2])
   ```

```
m3 <- mean(Veg$R[Veg$Transect == 3])
m4 <- mean(Veg$R[Veg$Transect == 4])
m5 <- mean(Veg$R[Veg$Transect == 5])
m6 <- mean(Veg$R[Veg$Transect == 6])
m7 <- mean(Veg$R[Veg$Transect == 7])
m8 <- mean(Veg$R[Veg$Transect == 8])
```

2. t apply

   More efficient

   ```
   tapply(Veg$R, Veg$Transect,mean)
   ```

   can be sd- standard deviation, var- variance, length- length more?

   ```
   Means <- tapply (Veg$R, Veg$Transect, mean)
   SDeviations <- tapply (Veg$R, Veg$Transect, sd)
   SSizes <- tapply (Veg$R, Veg$Transect, length)

   Means
   SDeviations
   SSizes
   ```

XIV. Bar Graph

```
Color.Table <- table(Men$Color)

barplot (Color.Table,
beside = TRUE,
legend = FALSE,
ylim = c(0,140),# Length of y axis
ylab = "Frequency",  # Y axis label
main = "")# graph title
```

Notice the switch in focus to the graphics window.

```
box()


barplot (Color.Table, beside = TRUE, legend = FALSE, ylim = c(0,140), ylab =
"Frequency",  main = "")

? barplot
Example(barplot)
```

XV. Histogram

```
hist(Men$Height, main = "", xlab = "Height (cm)", ylim = c(0,80))

box()
```

```
? hist
example (hist)
```

XVI. Box Plot

A. Also known as a box and whisker plot
B. Center line represents the median
C. Upper and lower parts of the box present the upper and lower quartile limits. (IQR)
D. Lower whisker – The lowest value that is no smaller than 1.5 IQR
E. Upper whisker – The largest value that is no larger than 1.5 IQR
F. Region between the whiskers should contain 99% of data if normally distributed

```
boxplot (Class$Height ~ Class$Sex,  # Variable by species
data = Class,                       # Dataset
ylab = " Height (cm)",              # y axis label
main = "Class Measurements",        # Graph title
boxwex = 0.5)                       # Width of boxes
```

```
boxplot (Class$Height ~ Class$Sex, data = Class, ylab = "Height (cm)", main =
"Class Measurements", boxwex = 0.5)
```

```
? boxplot
example (boxplot)
```

XVII. Scatterplot

```
plot (x = Class$Height, y = Class$Mass,
xlab = "Height (cm)",
ylab = "Mass (kg)",
main = "Class Measurements",
xlim = c(0,200),
ylim = c(0,120),
pch = 16,                #controls data symbol on plot (1-25)
cex = 1.25,              # controls data symbol size on plot (multiples)
col = Class$Sex)         # controls data color on plot (1-8)
```

```
plot (x = Class$Height, y = Class$Mass, xlab = "Height (cm)", ylab = "Mass (kg)",
main = "Class Measurements", xlim = c(0,200), ylim = c(0,120), pch = 16, cex = 1.25,
col = Class$Sex)
```

```
? plot
example (plot)
```

XVIII. Line Graph

1. Import the CellBio dataset into R and name it CellBio

```
# Determines the number of treatments = the number of lines in the graph
    nTreatment <- max(CellBio$Treatment)
    # Determines the range of the data for the X and Y axes
            xrange <- range(CellBio$Time)
            yrange <- range(CellBio$Weight)

    # Establishes the axes for the graph
            plot(xrange, yrange, type="n", xlab="Time (mins)", ylab="Percent
            Change in Mass")

    # Sets the graph parameters (line color and type)
            colors <- rainbow(nTreatment)
            linetype <- c(1:nTreatment)

    # Plots the data and lines in the graph by iterating through the different #
    treaments
            plotchar <- seq(18, 18+nTreatment, 1)
            for (i in 1:nTreatment) {
            treatment <- subset(CellBio, Treatment==i)
            lines(treatment$Time, treatment$Weight,
            type="b",
            lwd=2,
            lty=linetype[i],
            col=colors[i],
            pch=plotchar[i]
            )
            }
            title("")

    # Builds the legend for the graph
           legend(xrange[1], yrange[2], unique(CellBio$Solution), cex=0.8,
           col=colors,
           pch=plotchar,
           lty=linetype,
           title="Solution"
           )
```

XIX. Saving graphics as a file

A. Right click on the graph
B. Select save as enhanced metafile (.emf)
C. Navigate to where you want to save the file.
D. Give the file a name.
E. Click save.

XX. Quitting R

```
q()
```

XXI. Recording Work

1. A record of commands -> can recreate the workspace later