**R-Lab 1-MATH U315**  
  
**What is R?**  
**R** is a programming language and free software environment for statistical computing and graphics.  
  
**Download R:** <https://www.r-project.org/>

**Start R:** double click R icon on your desktop or Start R   
  
When you start R it opens the console window. The console has a few basic menus at the top; check them out on your own. The console is also where you enter commands for R to execute interactively, meaning that the command is executed and the result is displayed as soon as you hit the Enter key.  
  
**1. Interactive calculations:**# Add two numbers  
> 2+2

[1] 4  
  
# Store the results from calculations using a variable  
  
> a=2+2

# Print the stored value:

> a

[1] 4  
  
# Round to 0 decimal  
> round(123.56)

R-Help for **round** function:  
>?round  
  
It gives the detail about this function including, the type of input, order of input and the output

[1] 124

# Round to 2 decimals  
> round(123.5645,2)

[1] 123.56  
  
# Round up to next whole number  
> ceiling(123.56)

[1] 124  
  
  
  
  
# Round down to next whole number  
> floor(123.56)

[1] 123  
  
###### Operation between two stored values ########  
> x=5

> y=2

> z1=x\*y

> z2=x/y

> z3=x^y

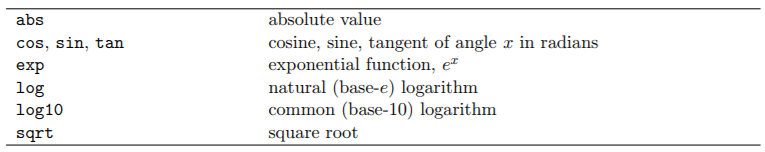
> z2

[1] 2.5

> z3

[1] 25

> z1

[1] 10  
  
**2. Some of the built-in mathematical functions in R:**  
 **For the detail functions :** [**https://www.statmethods.net/management/functions.html**](https://www.statmethods.net/management/functions.html)

> sqrt(16)

[1] 4

> abs(-10)

[1] 10

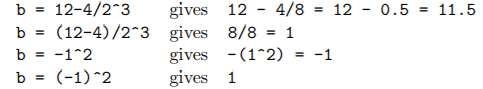
> exp(10)

[1] 22026.47 **Practice : Evaluate: for**

> x=3

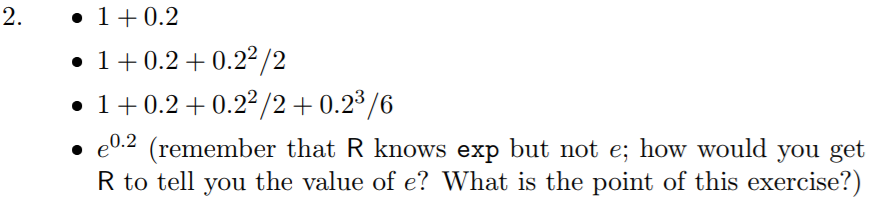
> C = (x+2\*sqrt(x))/(x+5\*sqrt(x))

> C

[1] 0.5543706  
  
**3. Order of mathematical operations:** The default order of operations is: (1) parentheses; (2) exponentiation, or powers, (3) multiplication and division, (4) addition and subtraction (“pretty please excuse my dear Aunt Sally”).  
  
  
  
Let’s calculate these in R

> b=12-4/2^3

> b

[1] 11.5  
 **Practice Exercise:** Compute the values of following using R  
  
1. and compare it with   
  
  
  
3. Evaluate for and . R knows ‘pi’.

**4. R-Help System:**  
You can get help on any R function by entering:   
> ?**function name**  
e.g.   
> ?sqrt

> ?sin  
  
**5. Creating Numeric Vector and perform few operations in it**  
Example: Consider you have a BMI (body mass index data): 36.8, 28.1, 18.1, 31.0, 17.2, 31.9, 19.6, 18.6, 21.9, 35.7, 32.7, 25.2, 25.2, 38.4, 35.2, 34.2, 26.5, 22.1, 39.2, 16.0.  
  
>BMI = c(36.8, 28.1, 18.1, 31.0, 17.2, 31.9, 19.6, 18.6, 21.9, 35.7, 32.7, 25.2, 25.2, 38.4, 35.2, 34.2, 26.5, 22.1, 39.2, 16.0)  
  
# To Print:   
>BMI

[1] 36.8 28.1 18.1 31.0 17.2 31.9 19.6 18.6 21.9 35.7 32.7 25.2 25.2 38.4 35.2

[16] 34.2 26.5 22.1 39.2 16.0  
  
# Length of vector: number of elements in the vector  
  
>length(BMI)  
  
# Multiply each element of a vector with 2  
  
> 2\*BMI

[1] 73.6 56.2 36.2 62.0 34.4 63.8 39.2 37.2 43.8 71.4 65.4 50.4 50.4 76.8 70.4

[16] 68.4 53.0 44.2 78.4 32.0  
  
# Multiply each element of a vector with 2 and add 5  
  
> 2\*BMI+5

[1] 78.6 61.2 41.2 67.0 39.4 68.8 44.2 42.2 48.8 76.4 70.4 55.4 55.4 81.8 75.4

[16] 73.4 58.0 49.2 83.4 37.0  
  
# Pull out specific values (say 3rd one) from the object, BMI  
  
> BMI[3]

[1] 18.1

# If you want values 3, 7, and 10 to 15.  
  
> BMI[c(3,7,10:15)]

[1] 18.1 19.6 35.7 32.7 25.2 25.2 38.4 35.2

# Add all the data values from BMI to obtain the sum.  
  
> sum(BMI)

[1] 553.6

# Sum of squared values of BMI  
  
> sum(BMI^2)

[1] 16436.04  
  
# Sum of square of the deviations of BMI values from 5.  
  
> sum((BMI-5)^2)

[1] 11400.04

#You can easily perform basic descriptive calculations (e.g., mean, median, and standard deviation) on your vector, using the built-in statistical functions in R:  
  
> mean(BMI)

[1] 27.68  
  
> median(BMI)

[1] 27.3  
  
> sd(BMI)

[1] 7.651598

# Convert vector “BMI” to a matrix of dimension with row=4, column = 5  
  
ccc = matrix(BMI, nrow=4, ncol=5, byrow=TRUE)  
> ccc

R-Help for **matrix** function:  
>?matrix  
  
It gives the detail about this function including, the type of inputs, order of inputs and the output

[,1] [,2] [,3] [,4] [,5]

[1,] 36.8 28.1 18.1 31.0 17.2

[2,] 31.9 19.6 18.6 21.9 35.7

[3,] 32.7 25.2 25.2 38.4 35.2

[4,] 34.2 26.5 22.1 39.2 16.0  
  
# Try to see the difference using  
  
ddd = matrix(BMI, nrow=4, ncol=5, byrow=FALSE)  
  
# Find row sums and column sums  
> rowSums(ccc)

[1] 131.2 127.7 156.7 138.0

> colSums(ccc)

[1] 135.6 99.4 84.0 130.5 104.1

**6. Creating Non-numeric Vectors and Factors**# First vector: names of 9 individuals

>subjectnames <- c("Bob", "Mary", "John", "Doug", "Lee", "Luke", "Leia", "Sarah", "Toby")

# To print:  
> subjectnames

[1] "Bob" "Mary" "John" "Doug" "Lee" "Luke" "Leia" "Sarah" "Toby"

##### Second vector: Gender of 20 individuals ######  
  
> sex <- c("M", "F", "M", "M", "M", "M", "F", "F", "M", "F", "M", "F", "M", "F", "F", "M", "F", "F", "F", "F")

# To print:  
> sex

[1] "M" "F" "M" "M" "M" "M" "F" "F" "M" "F" "M" "F" "M" "F" "F" "M" "F" "F" "F" "F"

# Return unique values of gender, which are “M” and “F”  
  
> unique(sex)

[1] "M" "F"  
  
# Check if vector “sex” is a factor type (or categorical type)  
  
> is.factor(sex)

[1] FALSE

# Encode (convert) a vector as a factor (or categories)  
  
> as.factor(sex)

[1] M F M M M M F F M F M F M F F M F F F F

Levels: F M

# Identify unique categories (or factor level) in the vector  
  
> levels(as.factor(sex))

[1] "F" "M"  
  
***Note: “unique” and “levels” function return the same things.***  
# How many males and females: frequency table  
> table(sex)

R-Help for **table** function:  
>?table  
  
It gives the detail about this function including, the type of input, order of input and the output

sex

F M

11 9

# If you want to store these frequencies  
> Fsex = table(sex)  
  
# Proportion of male and female  
  
> Fsex/sum(Fsex)

sex

F M

0.55 0.45