# Programming in Data science
# Intro to Functions

Asad Waqar Malik

Dept. of Information Systems
Faculty of Computer Science and Information Technology
University of Malaya
Malaysia

Oct 24, 2019

# Lecture Outline

1. Online class exercise
2. Scope
3. Quiz Instructions

# Outline

# Exercise

## Questions

```
1  Write a program that computes the average on a given ↩
       column. The function takes a data frame and column ↩
       name as an input and returns the mean of the mentioned↩
        column.
2
3  Revisit the above question and add that the mentioned ↩
       column should be numeric for the average calculation.
4
5  Complete the following function
6  f.sum <- function (x, y) {
7  # Write code
8  }
9  > f.sum(5, 10)
10 [1] 15
```

# Exercise

## Questions

Consider the function $f(x) = 2x^2 - 0.9x - 1$, and calculate $f(0)$ and $f(1.5)$

# Exercise

Consider an experiment where the biomass has been measured for 8 random plants grown under certain conditions. The sample values are denoted y1,..., yn. Assume that we are interested in an estimate of the population average, denoted $\mu$. It is well known that the sample mean $\bar{y}$ is an estimate of the population average. The sample mean is also the least squares estimate: Consider the sum of squared deviations from $\mu$, regarded as a function of $\mu$:

$$f(\mu) = (y1 - \mu)^2 + \cdots + (yn - \mu)^2$$

This function has its minimum for $\mu = \bar{y}$. Consider in the following the sample consisting of 24.7 32.5 22.6 23.9 19.6 21.6 19.9 20.9

# Exercise

$\Rightarrow$ Make a vector with the sample values, denoted y. Then make a function that takes $\mu$ (mu) as argument and calculates the f($\mu$). Call the function f.
$\Rightarrow$ Recall that the sample standard variance is defined as:

$$s^2 = \frac{1}{n-1}\left((y_1 - \bar{y})^2 + ... + (y_n - \bar{y})^2\right) \quad (1)$$

# Exercise

1. Write a function to calculate the mean and standard deviation for a vector? The function should return the mean and standard deviation value. Hint see mean(..) and sd(..).
2. Write a function make.cueb that take one argument as an input and return the cube of a number?
3. Create a function to accept an employee data frame(Name, Gender, Age, Designation & SSN) and print the Names & the Designation of all the employees in the function?
4. Create a user defined function to create a matrix and return the same?
5. Write a function var(x) that computes the variance of a sample?

# Self explore

Variable scope using Functions in R - A sample file is uploaded on Spectrum.

# Outline

# Global variables

- Global variables are those variables which exists throughout the execution of a program. It can be changed and accessed from any part of the program..
- However, global variables also depend upon the perspective of a function.
- For example, in the below example, from the perspective of inner_func(), both a and b are global variables

```
1  outer_func <- function(){
2  b <- 20
3  inner_func <- function(){
4  c <- 30
5  }
6  }
7  a <- 10
```

# Global variables

- However, from the perspective of outer_func(), b is a local variable and only a is global variable. The variable c is completely invisible to outer_func().

```
1  outer_func <- function(){
2  b <- 20
3  inner_func <- function(){
4  c <- 30
5  }
6  }
7  a <- 10
```

# Local variables

- On the other hand, Local variables are those variables which exist only within a certain part of a program like a function, and is released when the function call ends.
- In the above program the variable c is called a local variable.
- If we assign a value to a variable with the function inner_func(), the change will only be local and cannot be accessed outside the function.
- This is also the same even if names of both global variable and local variables matches, e.g.

```
1  outer_func <- function(){
2  a <- 20
3  inner_func <- function(){
4  a <- 30
5  print(a) }
6  inner_func()
7  print(a) }
```

# Local variables

- When we call it,

```
> a <- 10
> outer_func()
[1] 30
[1] 20
> print(a)
[1] 10
```

- We see that the variable a is created locally within the environment frame of both the functions and is different to that of the global environment frame.

# Accessing global variables

- Global variables can be read but when we try to assign to it, a new local variable is created instead.
- To make assignments to global variables, superassignment operator, `<<-` is used.
- When using this operator within a function, it searches for the variable in the parent environment frame, if not found it keeps on searching the next level until it reaches the global environment.
- If the variable is still not found, it is created and assigned at the global level.

```
1  outer_func <- function(){
2  inner_func <- function(){
3  a <<- 30
4  print(a)}
5  inner_func()
6  print(a)}
```

# Accessing global variables

- On running this function,

```
1  > outer_func()
2  [1] 30
3  [1] 30
4  > print(a)
5  [1] 30
```

- When the statement a $<<-$ 30 is encountered within inner_func(), it looks for the variable a in outer_func() environment.
- When the search fails, it searches in R_GlobalEnv.
- Since, a is not defined in this global environment as well, it is created and assigned there which is now referenced and printed from within inner_func() as well as outer_func().

# Variable Scope

- There is working code with comments available on "Variable Scope". Go through with that code for better understanding.

# Outline

# Online Quiz

- Will start at 8:00, time duration is 6 mins.