**CT5102: Programing for Data Analytics**

Assignment 4: Creating a my_tapply() function

Maximum Marks (10)

The goal of this assignment is to create a replica to the **tapply()** function - called **my_tapply()** - and test it against the actual **tapply()** function. As a reminder, the format of the **tapply()** function is **tapply(x,f,g),** where:

- x is a vector
- f is a factor (same length as x). More than one factor can be used
- g is a function

Write a **my_tapply()** function with the same set of arguments as **tapply().** Test it for two cases: (1) where the factor (f) is a single vector, and (2) where the factor (f) is a list of two vectors. An if/else statement will be needed within the function to distinguish between case (1) and case (2).The overall algorithm has the following structure.

- Use the **split()** function to split the vector x into factor-based groups, and then process the data (hint: sapply() can be used).
- If **f** is not a list, return a named vector
- If **f** is a list, return a two dimensional matrix. The first vector on the list can be referenced by f[[1]], and the second vector reference by f[[2]].
- The row and column names for a 2D matrix **m** can be set by:
  dimnames(m)[[1]]<- some.vector # for rows
  dimnames(m)[[2]]<- some.vector # for columns

Sample output is shown below:

```
> t1<-tapply(ages,affils,mean)
> t2<-tapply(ages,list(affils,gender),mean)
> r1<-my_tapply(ages,affils,mean)
> r2<-my_tapply(ages,list(affils,gender),mean)

> t1
 D  R  U
41 31 21
> r1
 D  R  U
41 31 21

> t2
   F  M
D 55 34
R 37 25
U 21 NA
> r2
   F  M
D 55 34
R 37 25
U 21 NA
```