**MODULE # 8**

**INPUT/OUTPUT, STRING MANIPULATION AND**

**PLYR PACKAGE**

This is an explanation about data input and output, string manipulation and using ‘plyr’ package.

‘read.table’ function can be used to read a table from a file.‘file.choose’ opens a dialog box which allows to choose file. ‘header=TRUE’ indicates that first row consists of column names. ‘sep=","’ code separates columns in the table by commas.

> #Imports dataset into R and assigns as ‘x’

> x=read.table(file.choose(),header=TRUE,sep=",")

> x

Name Age Sex Grade

1 Raul 25 Male 80

2 Booker 18 Male 83

3 Lauri 21 Female 90

4 Leonie 21 Female 91

5 Sherlyn 22 Female 85

6 Mikaela 20 Female 69

7 Raphael 23 Male 91

8 Aiko 24 Female 97

9 Tiffaney 21 Female 78

10 Corina 23 Female 81

11 Petronila 23 Female 98

12 Alecia 20 Female 87

13 Shemika 23 Female 97

14 Fallon 22 Female 90

15 Deloris 21 Female 67

16 Randee 23 Female 91

17 Eboni 20 Female 84

18 Delfina 19 Female 93

19 Ernestina 19 Female 93

20 Milo 19 Male 67

‘Plyr’ package allows to Split data, and run functions on that split section, then Combine it back with the original data set. The syntax is the same across all its functions, instead of having to load 5 packages for each data type. This makes plyr package the best choice.

> #installs 'plyr' package from CRAN

> install.packages("plyr")

> #loads package 'plyr'

> library(plyr)

Using the following code, the table "x" is splitted using the "ddply()" function from the "plyr" package. Subsets are evaluated using the variable "Sex", then results are transformed (grade average for each sex category) is assigned to a new column, and a new table is generated using the variable ‘y’.This gives mean for females as 86.9375 and mean for males as 80.2500.

> #run plyer generates for the mean of both Age and Grade split by gendor

> y = ddply(x,"Sex",transform, Grade.Average=mean(Grade))

> y

Name Age Sex Grade Grade.Average

1 Lauri 21 Female 90 86.9375

2 Leonie 21 Female 91 86.9375

3 Sherlyn 22 Female 85 86.9375

4 Mikaela 20 Female 69 86.9375

5 Aiko 24 Female 97 86.9375

6 Tiffaney 21 Female 78 86.9375

7 Corina 23 Female 81 86.9375

8 Petronila 23 Female 98 86.9375

9 Alecia 20 Female 87 86.9375

10 Shemika 23 Female 97 86.9375

11 Fallon 22 Female 90 86.9375

12 Deloris 21 Female 67 86.9375

13 Randee 23 Female 91 86.9375

14 Eboni 20 Female 84 86.9375

15 Delfina 19 Female 93 86.9375

16 Ernestina 19 Female 93 86.9375

17 Raul 25 Male 80 80.2500

18 Booker 18 Male 83 80.2500

19 Raphael 23 Male 91 80.2500

20 Milo 19 Male 67 80.2500

‘write.table’ function is used to write the table ‘y’ to file ‘Sorted\_Average’.

> #Print this to a file

> write.table(y,"Sorted\_Average")

‘sep=","’ code seperates columns in the table by commas.

> #Generate a CSV(comma-separated values)

> write.table(y,"Sorted\_Average",sep=",")

The following code creates a new table ‘newx’ that consists of rows of ‘x’ that has names with letter ‘I’ or ‘i’.‘subset()’function is used to get the rows and columns from the data frame. ‘grepl’function tests whether the pattern is found in each name.

> #Filter the names in the given list that contain the letter i or I.

> newx = subset(x,grepl("[iI]",x$Name))

‘write.table’ function is used to write the table ‘newx’ to file ‘DataSubset’

> #writes this subset to a file

> write.table(newx,"DataSubset",sep=",")

In this way, by following these codes we can import data and write table to file. In addition, we can filter the data and also get grade average from data based on requirements utilizing ‘plyr’ package.

