Page No.
Date

Assignment No. 5

* **Aim :-** Perform different operations using R/python

* **Problem statement :-** Perform the following operations using R/python on the Amazon book reviews & feedback metrics data sets.

* **Objectives :-** ① To learn R/python programming
② To learn different data preprocessing techniques.

* **Theory :-**

- **Subsetting data :-** R has powerful indexing ~~feeter~~ feature for accessing object elements. These features can be used to select and element variables and observations. The following code snippets demonstrate ways to keep or delete variables and observations and to take random samples form of a dataset.

- Selecting (keeping) variables.

```
# select variables V1, V2, V3 ; myvars <- (V1, 'V2', 'V3')
newdata <- mydata [myvars]
```

- Excluding (dropping variables)

```
# exclude variables V1, V2, V3
```

myvars ← names(mydata) %in% c("V1", "V2", "V3")
newdata ← mydata[, myvars]

* selecting observations :-

#first 5 observations
newdata ← mydata[1:5,]

* selection using a subset function :-

The subset() function is easiest way to select variables and observations. In the following examples we select all rows that have a value of age greater than or equal to 20 or age less than 10. We keep the ID and weight columns.

#using subset function
newdata ← subset(mydata, age >= 20 | age < 10)
select = c(ID, weight)

- Merging data :-
To merge two data frame horizontally see the merge function. In the most cases you join two data frames by one or more common key vars.

merge two data frames by ID
total ← merge(dataFrameA, dataFrameB, by="ID")

* Sorting of data:
A sort a data frame in R, use the order() function. By default, sorting in ASCENDING. Prepend the sorting variables by a minus sign to indicate DECENDING order

Ex.
attach(mtcars)

newdata ← mtcars[order(mpg)]

- Transpose:
Use the t() function to transpose a matrix or a data frame. In the later case, rownames become variable (column) name

Example using build-in dataset
mtcars
t(mtcars).

Page No.
Date:

* The Reshape package
Basically you melt data so that each row is a unique id-variable combination. Then you cast the melted data into any shape you would like. Here is a very simple example

#Example of melt function

library(reshape)
mdata ← melt(mydata, id=c, ("id", "time"))

cast melted data
cast(data, formula, function)

subjmeans ← cast(mdata, id_variable, mean)
timemeans ← cast(mdata, time_variable, mean).

* Conclusion :- Thus we have studied & understand different data preprocessing technique