**Data Manipulations \_ DML \_ part\_1**

is the process of manipulating (creating, arranging, deleting) data points in a given data to get insights much easier

Steps involved,

**Create database \_**webscraping the data points unstructured to structured in database for processing

**Arrange the data\_** data preprocessed arranged for analysis

**Cleaning the data** \_ unnecessary data or outliers

The below are code chunks in R studio,

filtering, arranging , sorting, manipulation operations done.

The **DML part\_2** we will deep dive following week, more about **DML**

**install packages Dplyr**

library(tidyverse)

library(dplyr)

**create sample data**

data frame function used to create data

my\_data<-data.frame(y1=c(1:5),y2=letters[-1:-21],y3=5:1)  
my\_data

## y1 y2 y3  
## 1 1 v 5  
## 2 2 w 4  
## 3 3 x 3  
## 4 4 y 2  
## 5 5 z 1

**Creating the new variable**

variable name jothi\_y1 created , the vakues from addition of variables y1 and y3 the mutate

 function used.

new\_data <- my\_data %>% mutate(jothi\_y1=y1+y3)  
new\_data # print sample data

## y1 y2 y3 jothi\_y1  
## 1 1 v 5 6  
## 2 2 w 4 6  
## 3 3 x 3 6  
## 4 4 y 2 6  
## 5 5 z 1 6

**Rename the variable / column name**

variable name y1 rename as sabai\_y1 using renaming function

new\_data %>% rename(sabai\_y1=y1)

## sabai\_y1 y2 y3 jothi\_y1  
## 1 1 v 5 6  
## 2 2 w 4 6  
## 3 3 x 3 6  
## 4 4 y 2 6  
## 5 5 z 1 6

**Selecting the specific variables by our choice\_ select function used,**

selecting the variable names y1,jothi\_\_1 and y2

new\_data %>% select(y1,jothi\_y1,y2)

## y1 jothi\_y1 y2  
## 1 1 6 v  
## 2 2 6 w  
## 3 3 6 x  
## 4 4 6 y  
## 5 5 6 z

**Arrange** function used to rearrange the values of the variables

variable y1 values are manipulate in to descending orders

new\_data %>% arrange(desc(y1)) # descending order

## y1 y2 y3 jothi\_y1  
## 1 5 z 1 6  
## 2 4 y 2 6  
## 3 3 x 3 6  
## 4 2 w 4 6  
## 5 1 v 5 6

variable y1 values are manipulate in to descending orders

new\_data %>% arrange(y1) # ascending order

## y1 y2 y3 jothi\_y1  
## 1 1 v 5 6  
## 2 2 w 4 6  
## 3 3 x 3 6  
## 4 4 y 2 6  
## 5 5 z 1 6

**Order function** Alternate method for sorting ascending to the variable values

new\_data[order(new\_data$y2),]

## y1 y2 y3 jothi\_y1  
## 1 1 v 5 6  
## 2 2 w 4 6  
## 3 3 x 3 6  
## 4 4 y 2 6  
## 5 5 z 1 6

**Order function** Alternate method for sorting descending to the variable values

new\_data[rev(order(new\_data$y2)),]

## y1 y2 y3 jothi\_y1  
## 5 5 z 1 6  
## 4 4 y 2 6  
## 3 3 x 3 6  
## 2 2 w 4 6  
## 1 1 v 5 6

**SLICE** \_ selecting the rows , slice function used to show requested number of rows.

i am calling here only 2 rows here.

new\_data %>% slice(1:2)

## y1 y2 y3 jothi\_y1  
## 1 1 v 5 6  
## 2 2 w 4 6

**filtering** the rows or values of the variables by conditions

filtering the value of y1 greater then 3, and y3 is less then 2

new\_data %>% filter(y1>3 & y3<2)

## y1 y2 y3 jothi\_y1  
## 1 5 z 1 6

**summarize** the data by grouping

grouping the jothi\_y1 variable and summarise it for mean, new variable mean\_y1 created.

new\_data %>% group\_by(jothi\_y1) %>% summarise(mean\_y1=mean(y1))

## # A tibble: 1 × 2  
## jothi\_y1 mean\_y1  
## <int> <dbl>  
## 1 6 3

**counting** the rows by group

the number of times v ,w, …. occured shown in “n” column

new\_data %>% count(y2)

## y2 n  
## 1 v 1  
## 2 w 1  
## 3 x 1  
## 4 y 1  
## 5 z 1