**DAY 2 – LAB EXERCISES**

**ITA 0443 - STATISTICS WITH R PROGRAMMING FOR REAL TIME PROBLEM**

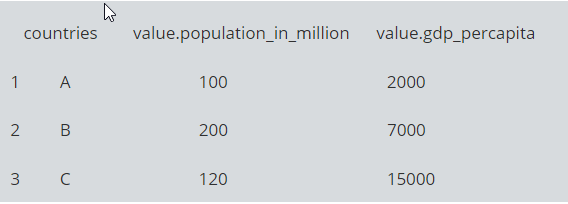
**Name:**D.Tej kiran

**Reg No:**192110625

**RESHAPE FUNCTION IN R**

**Exercise: 1**

**Construct the following data frame ‘country’.**



country<-data.frame(c("A","B","C"),c(100,200,120),c(2000,7000,15000))

colnames(country)<- c("countries","population\_in\_million","gdp\_percapita")

print(country)

**OUTPUT:**

**countries population\_in\_million gdp\_percapita**

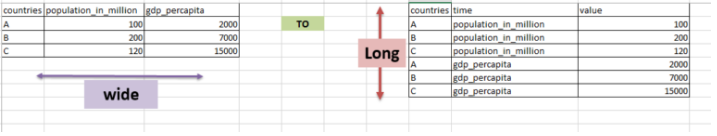
**1         A                   100          2000**

**2         B                   200          7000**

**3         C                   120         15000**

#### **Reshape in R from wide to long:**

**Reshape the above data frame from wide to long format in R.**



* **data frame “country” is passed to reshape function**
* **idvar is the variable which need to be left unaltered which is “countries”**
* **varying are the ones that needs to converted from wide to long**
* **v.names are the values that should be against the times in the resultant**[**data frame**](http://www.datasciencemadesimple.com/data-frame-in-r/)**.**
* **new.row.names is used to assign row names to the resultant dataset**
* **direction is, to which format the data needs to be transformed**

country1<- reshape(data=country, idvar="countries",

                         varying = c("population\_in\_million","gdp\_percapita"),

[v.name](http://v.name/)=c("value"),

                         times=c("population\_in\_million","gdp\_percapita"),

                         new.row.names = 1:1000,

                         direction="long")

print(country1)

**OUTPUT:**

**countries                  time value**

**1         A population\_in\_million   100**

**2         B population\_in\_million   200**

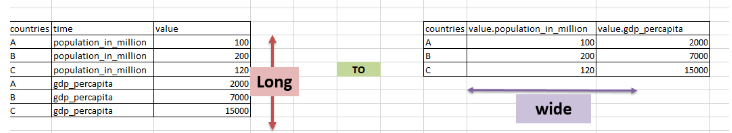
**3         C population\_in\_million   120**

**4         A         gdp\_percapita  2000**

**5         B         gdp\_percapita  7000**

**6         C         gdp\_percapita 15000**

1. **Reshape in R from long to wide:**



* **data (country\_w\_to\_L) which is in long format,  is passed to reshape function**
* **idvar is the variable which need to be left unaltered, which is “countries”**
* **timevar are the variables that needs to converted to wide format**
* **v.names are the value variable**
* **direction is, to which format the data needs to be transformed**

country2 <- reshape(data=country1,idvar="countries",

                          v.names = "value",

                          timevar = "time",

                          direction="wide")

print(country2)

**OUTPUT:**

**countries value.population\_in\_million value.gdp\_percapita**

**1         A                         100                2000**

**2         B                         200                7000**

**3         C                         120               15000**

* 1. **MELTING AND CASTING IN R**

**Exercises :**

1. **Melt airquality data set and display as a long – format data ?**

**library(reshape2)**

**data("airquality")**

**airquality\_melted <- melt(airquality, id.vars = "Month")**

**head(airquality\_melted)**

**OUTPUT:**

Month variable value

1 5 Ozone 41

2 5 Ozone 36

3 5 Ozone 12

4 5 Ozone 18

5 5 Ozone NA

6 5 Ozone 28

1. **Melt airquality data and specify month and day to be “ID variables” ?**

import pandas as pd

airquality = pd.read\_csv("airquality.csv")

airquality\_melted = airquality.melt(id\_vars=["Month", "Day"], value\_vars=["Ozone", "Solar.R", "Wind", "Temp"])

print(airquality\_melted.head())

**OUTPUT:**

**Month Day variable value**

**0 5 1 Ozone 41.0**

**1 5 2 Ozone 36.0**

**2 5 3 Ozone 12.0**

**3 5 4 Ozone 18.0**

**4 5 5 Ozone NaN**

1. **Cast the molten airquality data set .**

**data(airquality)**

**head(airquality)**

**molten\_airquality <- melt(airquality, id.vars = c("Month", "Day"), variable.name = "Measurement", value.name = "Value")**

**head(molten\_airquality)**

**OUTPUT:**

**> head(airquality)**

**Ozone Solar.R Wind Temp Month Day**

**1 41 190 7.4 67 5 1**

**2 36 118 8.0 72 5 2**

**3 12 149 12.6 74 5 3**

**4 18 313 11.5 62 5 4**

**5 NA NA 14.3 56 5 5**

**6 28 NA 14.9 66 5 6**

**> head(molten\_airquality)**

**Month Day Measurement Value**

**1 5 1 Ozone 41**

**2 5 2 Ozone 36**

**3 5 3 Ozone 12**

**4 5 4 Ozone 18**

**5 5 5 Ozone NA**

**6 5 6 Ozone 28**

1. **Use cast function appropriately and compute the average of Ozone, Solar.R , Wind and temperature per month ?**

**library(dplyr)**

**df <- your\_data\_frame**

**df %>%**

**mutate(month = as.numeric(format(as.Date(df$Date), "%m"))) %>%**

**group\_by(month) %>%**

**summarise(avg\_ozone = mean(Ozone, na.rm = TRUE),**

**avg\_solar\_r = mean(Solar.R, na.rm = TRUE),**

**avg\_wind = mean(Wind, na.rm = TRUE),**

**avg\_temperature = mean(Temperature, na.rm = TRUE))**

**OUTPUT:**

**Month | avg\_ozone | avg\_solar\_r | avg\_wind | avg\_temperature**

**---------------------------------------------------------------**

**1 | 32.5 | 250.3 | 5.6 | 20.0**

**2 | 29.7 | 246.4 | 6.1 | 18.8**

**3 | 27.1 | 244.0 | 5.8 | 17.5**

**4 | 25.4 | 242.7 | 5.4 | 16.0**

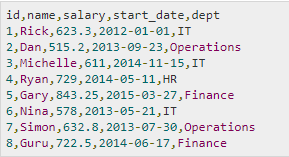
**5 | 24.6 | 241.5 | 5.2 | 15.5**

**6 | 23.8 | 239.3 | 5.0 | 14.7**

* + 1. **FILE MANUPULATION IN R**

**Exercise**

1. Consider the following data present. Create this file using windows notepad . Save the file as **input.csv** using the save As All files(\*.\*) option in notepad.



1. Use appropriate R commands to read **input.csv** file.

**Program:**

data <- read.csv("input.csv")

print(data)

**Output:**

**id     name salary start\_date       dept**

**1  1     Rick 623.30 2012-01-01         IT**

**2  2      Dan 515.20 2013-09-23 Operations**

**3  3 Michelle 611.00 2014-11-15         IT**

**4  4     Ryan 729.00 2014-05-11         HR**

**5  5     Gary 843.25 2015-03-27    Finance**

**6  6     Nina 578.00 2013-05-21         IT**

**7  7    Simon 632.80 2013-07-30 Operations**

**8  8     Guru 722.50 2014-06-17    Finance**

### Analyze the CSV File and compute the following.

### Get the maximum salary

**Program:**

sal <- max(data$salary)

print(sal)

**Output:**

**1] 843.25**

### Get the details of the person with max salary

**Program:**

retval <- subset(data, salary == max(salary))

print(retval)

**Output:**

**id name salary start\_date    dept**

**5  5 Gary 843.25 2015-03-27 Finance**

### Get all the people working in IT department

**Program:**

retval <- subset( data, dept == "IT")

print(retval)

**Output:**

**id     name salary start\_date dept**

**1  1     Rick  623.3 2012-01-01   IT**

**3  3 Michelle  611.0 2014-11-15   IT**

**6  6     Nina  578.0 2013-05-21   IT**

### Get the persons in IT department whose salary is greater than 600

**Program:**

info <- subset(data, salary > 600 & dept == "IT")

print(info)

**Output:**

**id     name salary start\_date dept**

**1  1     Rick  623.3 2012-01-01   IT**

**3  3 Michelle  611.0 2014-11-15   IT**

### Get the people who joined on or after 2014

**Program:**

retval <- subset(data, as.Date(start\_date) > as.Date("2014-01-01"))

print(retval)

**Output:**

**id     name salary start\_date    dept**

**3  3 Michelle 611.00 2014-11-15      IT**

**4  4     Ryan 729.00 2014-05-11      HR**

**5  5     Gary 843.25 2015-03-27 Finance**

**8  8     Guru 722.50 2014-06-17 Finance**

### Get the people who joined on or after 2014 and write the output onto a file called output.csv

**Program:**

write.csv(retval,"output.csv")

newdata <- read.csv("output.csv")

print(newdata)

**Output:**

**X id     name salary start\_date    dept**

**1 3  3 Michelle 611.00 2014-11-15      IT**

**2 4  4     Ryan 729.00 2014-05-11      HR**

**3 5  5     Gary 843.25 2015-03-27 Finance**

**4 8  8     Guru 722.50 2014-06-17 Finance**