SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

ITA 0443 – STATISTICS WITH R PROGRAMMING FOR REAL TIME PROBLEM

DAY 2 – LAB EXERCISES

Reg No:192110529

Name:N M S N V ARUN

RESHAPE FUNCTION IN R

Exercise: 1

Construct the following data frame 'country'.

	countries	value.population_in_million	value.gdp_percapita
1	A	100	2000
2	B	200	7000
3	C	120	15000

Programm:

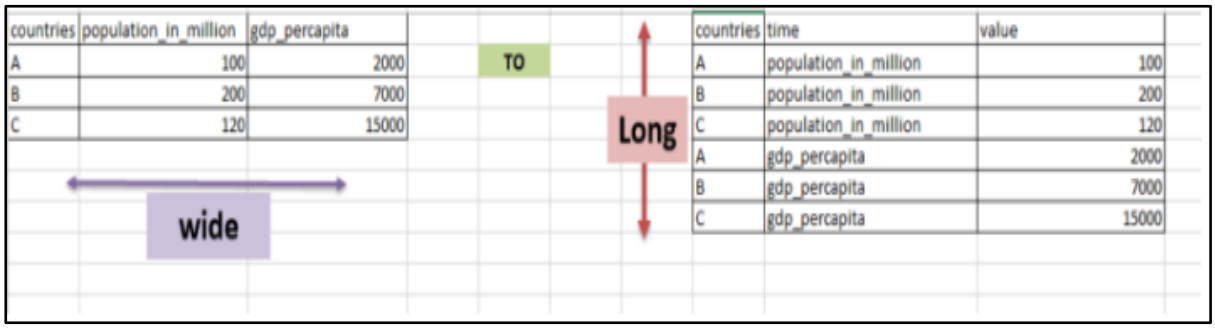
```
country<-data.frame(c("A","B","C"),c(100,200,120),c(2000,7000,15000))
colnames(country)<- c("countries","population_in_million","gdp_percapita")
print(country)
```

Output:

```
  countries population_in_million gdp_percapita
1         A                   100          2000
2         B                   200          7000
3         C                   120         15000
```

a) **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.
- new.row.names is used to assign row names to the resultant dataset
- direction is, to which format the data needs to be transformed

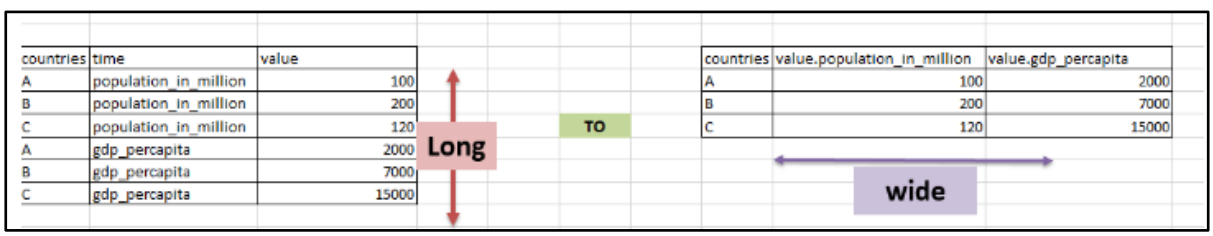
Program:

```
country1<- reshape(data=country, idvar="countries",
                   varying = c("population_in_million","gdp_percapita"),
                   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
```

b) **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

Program:

```
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

7. MELTING AND CASTING IN R

Exercises :

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

programm:

```
data("airquality")
> library(reshape2)
> airquality_melted <- melt(airquality, id.vars = c("Ozone", "Solar.R", "Wind", "Temp"))
> head(airquality_melted)
```

output:

```
Ozone Solar.R Wind Temp variable value

1    41     190  7.4   67    Month     5

2    36     118  8.0   72    Month     5

3    12     149 12.6   74    Month     5

4    18     313 11.5   62    Month     5

5    NA      NA 14.3   56    Month     5

6    28      NA 14.9   66    Month     5
```

2. Melt airquality data and specify month and day to be "ID variables" ?

programm:

```
library(reshape2)

> air_quality <- data.frame(month = c("Jan", "Feb", "Mar"),

+                           day = c("Mon", "Tue", "Wed"),

+                           Ozone = c(35, 40, 30),

+                           Solar.R = c(190, 200, 180),

+                           Wind = c(7.4, 8.0, 6.9))

> air_quality_melted <- melt(air_quality, id.vars = c("month", "

+ day"))

Error: id variables not found in data:

day

> air_quality_melted <- melt(air_quality, id.vars = c("month", "day"))
```

```
> print(air_quality_melted)
```

output:

```
month day variable value
1   Jan Mon    Ozone  35.0
2   Feb Tue    Ozone  40.0
3   Mar Wed    Ozone  30.0
4   Jan Mon  Solar.R 190.0
5   Feb Tue  Solar.R 200.0
6   Mar Wed  Solar.R 180.0
7   Jan Mon     Wind   7.4
8   Feb Tue     Wind   8.0
9   Mar Wed     Wind   6.9
```

3. Cast the molten airquality data set .

programm:

```
library(reshape2)
> air_quality_melted <- data.frame(month = c("Jan", "Feb", "Mar", "Jan", "Feb", "Mar", "Jan", "Feb", "Mar"),
+                                  day = c("Mon", "Tue", "Wed", "Mon", "Tue", "Wed", "Mon", "Tue", "Wed"),
+                                  variable = c("Ozone", "Ozone", "Ozone", "Solar.R", "Solar.R", "Solar.R",
"Wind", "Wind", "Wind"),
+                                  value = c(35, 40, 30, 190, 200, 180, 7.4, 8.0, 6.9))
> air_quality_casted <- dcast(air_quality_melted, month + day ~ variable, value.var = "value")
```

```
> print(air_quality_casted)
```

output:

```
month day Ozone Solar.R Wind
1   Feb Tue    40     200  8.0
2   Jan Mon    35     190  7.4
3   Mar Wed    30     180  6.9
```

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

programm:

```
 library(reshape2)
> air_quality <- data.frame(month = c("Jan", "Jan", "Jan", "Feb", "Feb", "Feb", "Mar", "Mar", "Mar"),
+                           day = c("Mon", "Tue", "Wed", "Mon", "Tue", "Wed", "Mon", "Tue", "Wed"),
+                           Ozone = c(35, 40, 30, 33, 36, 39, 32, 35, 38),
+                           Solar.R = c(190, 200, 180, 185, 195, 205, 170, 165, 160),
+                           Wind = c(7.4, 8.0, 6.9, 7.3, 7.5, 7.2, 6.8, 7.1, 6.9),
+                           Temperature = c(6, 7, 8, 5, 6, 7, 4, 5, 6))
>
> air_quality_melted <- melt(air_quality, id.vars = "month")
>
> air_quality_mean <- dcast(air_quality_melted, month ~ variable, fun.aggregate = mean)
There were 16 warnings (use warnings() to see them)
```

>

> print(air_quality_mean)

output:

```
month day Ozone Solar.R Wind Temperature
1   Feb  NA    NA      NA   NA          NA
2   Jan  NA    NA      NA   NA          NA
3   Mar  NA    NA      NA   NA          NA
>
```

8 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.

```
id,name,salary,start_date,dept
1,Rick,623.3,2012-01-01,IT
2,Dan,515.2,2013-09-23,Operations
3,Michelle,611,2014-11-15,IT
4,Ryan,729,2014-05-11,HR
5,Gary,843.25,2015-03-27,Finance
6,Nina,578,2013-05-21,IT
7,Simon,632.8,2013-07-30,Operations
8,Guru,722.5,2014-06-17,Finance
```

2. 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
```

3. Analyze the CSV File and compute the following.

a. Get the maximum salary

Program:

```
sal <- max(data$salary)
print(sal)
```

Output:

```
1] 843.25
```

b. 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
```

c. 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
```

d. 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
```

e. 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
```

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

rogram:

```
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
```