RECASTING AND JOINING OF DATA FRAMES

```
>pd=data.frame("NAME"=c("senthil","senthil","sam","sam"),"MONTH"=c("jan","
feb","jan","feb"),"BS"=c(141.2,130.9,120,129),"BP"=c(90,78,85,89))
Warning messages:
1: package 'RMySQL' was built under R version 3.6.1
2: package 'DBI' was built under R version 3.6.1
3: package 'arules' was built under R version 3.6.1

> pd
     NAME MONTH    BS BP
1 senthil   jan 141.2 90
2 senthil   feb 130.9 78
3     sam   jan 120.0 85
4     sam   feb 129.0 89

> library(reshape2)
Warning message:
package 'reshape2' was built under R version 3.6.1


> df=melt(pd,id.vars=c("NAME","MONTH"),measure.vars = c("BS","BP"))


> df
     NAME MONTH variable value
1 senthil   jan       BS 141.2
2 senthil   feb       BS 130.9
3     sam   jan       BS 120.0
4     sam   feb       BS 129.0
5 senthil   jan       BP  90.0
6 senthil   feb       BP  78.0
7     sam   jan       BP  85.0
8     sam   feb       BP  89.0

>
```

JOIN IN R: HOW TO JOIN (MERGE) DATA FRAMES (INNER, OUTER, LEFT, RIGHT) IN R

We will have look at an example of

- Inner join using merge() function in R or inner_join() function of dplyr with example
- Outer join using merge() function or full_join() function of dplyr with example
- Left join using left_join() function of dplyr or merge() function
- Right join using right_join() function of dplyr or merge() function.
- Cross join with merge() function
- semi join and anti join in R using semi_join() function and anti_join() function.

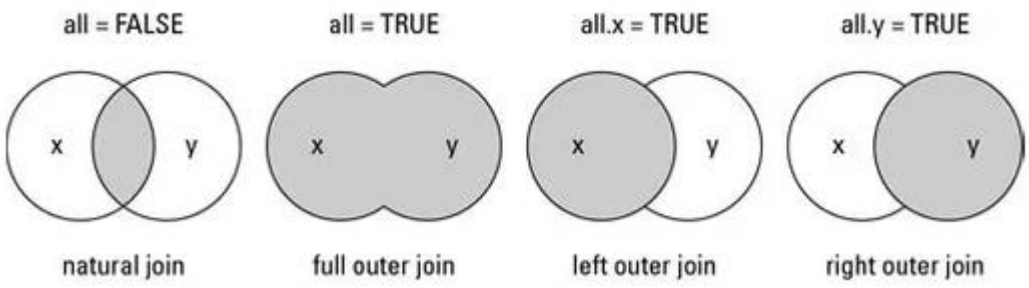
Syntax of merge() function in R

```
merge(x, y, by.x, by.y,all.x,all.y, sort = TRUE)
```

- **x:**data frame1.
- **y:**data frame2.
- **by,x, by.y:** The names of the columns that are common to both x and y. The default is to use the columns with common names between the two data frames.
- **all, all.x, all.y:**Logical values that specify the type of merge. The default value is all=FALSE (meaning that only the matching rows are returned).

UNDERSTANDING THE DIFFERENT TYPES OF MERGE IN R:

- **Natural join or Inner Join**: To keep only rows that match from the data frames, specify the argument all=FALSE.
- **Full outer join or Outer Join:**To keep all rows from both data frames, specify all=TRUE.
- **Left outer join or Left Join:**To include all the rows of your data frame x and only those from y that match, specify x=TRUE.
- **Right outer join or Right Join:**To include all the rows of your data frame y and only those from x that match, specify y=TRUE.



Lets look at with some examples

```
# data frame 1
df1 = data.frame(CustomerId = c(1:6), Product =
c("Oven","Television","Mobile","WashingMachine","Lightings","Ipad"))
df1

# data frame 2
df2 = data.frame(CustomerId = c(2, 4, 6, 7, 8), State =
c("California","Newyork","Santiago","Texas","Indiana"))
df2
```

so we will get following two data frames

df1 will be

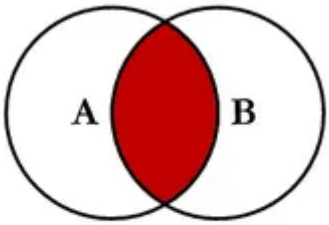
	CustomerId	Product
1	1	Oven
2	2	Television
3	3	Mobile
4	4	WashingMachine
5	5	Lightings
6	6	Ipad

df2 will be

	CustomerId	State
1	2	California
2	4	Newyork
3	6	Santiago
4	7	Texas
5	8	Indiana

INNER JOIN Explained

Inner Join in R is the simplest and most common type of join. It is also known as simple join or Natural Join. Inner join returns the rows when matching condition is met.



Inner Join

```
df = merge(x=df1,y=df2,by="CustomerId")
df
```

the resultant inner joined dataframe df will be

	CustomerId	Product	State
1	2	Television	California
2	4	WashingMachine	Newyork
3	6	Ipad	Santiago

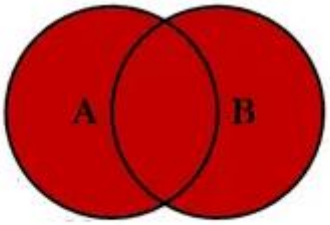
Inner join in R using inner_join() function of dplyr:

dplyr() package has inner_join() function which performs inner join of two dataframes by "CustomerId" as shown below.

```
library(dplyr)
df= df1 %>% inner_join(df2,by="CustomerId")
df
```

OUTER JOIN Explained

Outer Join in R combines the results of both left and right outer joins. The joined table will contain all records from both the tables



Outer Join

Outer join in R using merge() function*:* merge() function takes df1 and df2 as argument along with all=TRUE there by returns all rows from both tables, join records from the left which have matching keys in the right table.

```
###### outer join in R using merge() function
df = merge(x=df1,y=df2,by="CustomerId",all=TRUE)
df
```

the resultant data frame df will be

	CustomerId	Product	State
1	1	Oven	<NA>
2	2	Television	California
3	3	Mobile	<NA>
4	4	WashingMachine	Newyork
5	5	Lightings	<NA>
6	6	Ipad	Santiago
7	7	<NA>	Texas
8	8	<NA>	Indiana

outer join in R using full_join() function of dplyr:

dplyr() package has full_join() function which performs outer join of two dataframes by "CustomerId" as shown below.

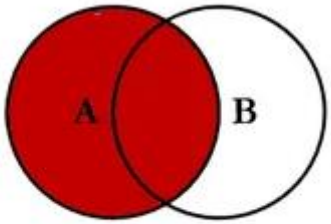
```
###### outer join in R using outer_join() function
library(dplyr)
df= df1 %>% full_join(df2,by="CustomerId")
df
```

	CustomerId	Product	State
1	1	Oven	<NA>
2	2	Television	California
3	3	Mobile	<NA>
4	4	WashingMachine	Newyork
5	5	Lightings	<NA>
6	6	Ipad	Santiago
7	7	<NA>	Texas
8	8	<NA>	Indiana

LEFT JOIN Explained:

The **LEFT JOIN in R** returns all records from the **left** dataframe (A), and the matched records from the right dataframe (B)

Left join in R: merge() function takes df1 and df2 as argument along with all.x=TRUE there by returns all rows from the left table, and any rows with matching keys from the right table.



Left join

left join in R using merge() function
df = merge(x=df1,y=df2,by="CustomerId",all.x=TRUE)
df

the resultant data frame df will be

	CustomerId	Product	State
1	1	Oven	<NA>
2	2	Television	California
3	3	Mobile	<NA>
4	4	WashingMachine	Newyork
5	5	Lightings	<NA>
6	6	Ipad	Santiago

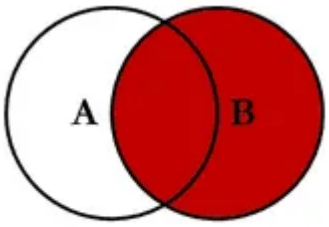
Left join in R using left_join() function of dplyr:

dplyr() package has left_join() function which performs left join of two dataframes by "CustomerId" as shown below.

left join in R using left_join() function
library(dplyr)
df= df1 %>% left_join(df2,by="CustomerId")
df

RIGHT JOIN Explained:

The **RIGHT JOIN in R** returns all records from the **right** dataframe (B), and the matched records from the left dataframe (A)
Right join in R: merge() function takes df1 and df2 as argument along with all.y=TRUE and thereby returns all rows from the right table, and any rows with matching keys from the left table.



Right Join

right join in R using merge() function
df = merge(x=df1,y=df2,by="CustomerId",all.y=TRUE)
df

the resultant data frame df will be

	CustomerId	Product	State
1	2	Television	California
2	4	WashingMachine	Newyork
3	6	Ipad	Santiago
4	7	<NA>	Texas
5	8	<NA>	Indiana

Right join in R using right_join() function of dplyr:
dplyr() package has right_join() function which performs outer join of two dataframes by "CustomerId" as shown below.

right join in R using merge() function

library(dplyr)

df= df1 %>% right_join(df2,by="CustomerId")
df

Cross join in R: A Cross Join (also sometimes known as a Cartesian Join) results in every row of one table being joined to every row of another table

cross join in R

df = merge(x = df1, y = df2, by = NULL)
df

the resultant data frame df will be

	CustomerId.x	Product	CustomerId.y	State
1	1	Oven	2	California
2	2	Television	2	California
3	3	Mobile	2	California
4	4	WashingMachine	2	California
5	5	Lightings	2	California
6	6	Ipad	2	California
7	1	Oven	4	Newyork
8	2	Television	4	Newyork
9	3	Mobile	4	Newyork
10	4	WashingMachine	4	Newyork
11	5	Lightings	4	Newyork
12	6	Ipad	4	Newyork
13	1	Oven	6	Santiago
14	2	Television	6	Santiago
15	3	Mobile	6	Santiago
16	4	WashingMachine	6	Santiago
17	5	Lightings	6	Santiago
18	6	Ipad	6	Santiago
19	1	Oven	7	Texas
20	2	Television	7	Texas
21	3	Mobile	7	Texas
22	4	WashingMachine	7	Texas
23	5	Lightings	7	Texas
24	6	Ipad	7	Texas
25	1	Oven	8	Indiana
26	2	Television	8	Indiana
27	3	Mobile	8	Indiana
28	4	WashingMachine	8	Indiana
29	5	Lightings	8	Indiana
30	6	Ipad	8	Indiana

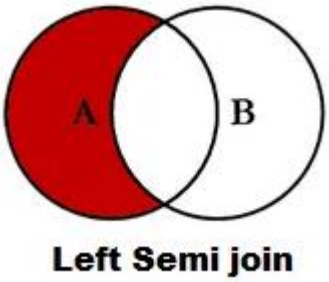
SEMI JOIN in R using dplyr:

This is like inner join, with only the left dataframe columns and values are selected

Semi join in R

```
library(dplyr)

df= df1 %>% semi_join(df2,by="CustomerId")
df
```



Left Semi join

the resultant data frame df will be

	CustomerId	Product
1	2	Television
2	4	WashingMachine
3	6	Ipad

ANTI JOIN in R using dplyr:

This join is like df1-df2, as it selects all rows from df1 that are not present in df2.

anti join in R

```
library(dplyr)
df= df1 %>% anti_join(df2,by="CustomerId")
```

df

the resultant data frame df will be

	CustomerId	Product
1	1	Oven
2	3	Mobile
3	5	Lightings