Problem Statement

1.

```
df1 = data.frame(CustId = c(1:6), Product = c(rep("TV", 3), rep("Radio", 3)))
df2 = data.frame(CustId = c(2, 4, 6), State = c(rep("Texas", 2), rep("NYC",
1)))
df1 #left table
df2 #right table
```

For the above given data frames and tables perform the following operations:

- Return only the rows in which the left table have match.
- Returns all rows from both tables, join records from the left which have matching keys in the right table.
- Return all rows from the left table, and any rows with matching keys from the right table.
- Return all rows from the right table, and any rows with matching keys from the left table.

2. Perform the below operations on above given data frames and tables:

- Return a long format of the datasets without matching key.
- Keep only observations in df1 that match in df2.
- Drop all observations in df1 that match in df2.

Answers 1: R script is also attached for reference.

Step1: **Return library(dplyr)**

- Return only the rows in which the left table have match.

Here, we have to use **"Inner Join"** to get the rows of only the left table having a match.

```
inner_join(df1,df2, by=c("CustId"="CustId"))
```

```
> #Inner Join to return only the rows of the left table having match
> library(dplyr)
> inner_join(df1,df2, by=c("CustId"="CustId"))
  CustId Product State
1      2      TV Texas
2      4   Radio Texas
3      6   Radio   NYC
>
```

• Returns all rows from both tables, join records from the left which have matching keys in the right table.

In this scenario, we need to return all rows of both **df1** and **df2** and join the records of **df1,** which have matching keys in the **df2.** Use **full_Join** to do the same.

full_join(df1,df2,by=c("CustId"="CustId"))

```
> #FullJoin
> full_join(df1,df2,by=c("CustId"="CustId"))
  CustId Product State
1      1      TV  <NA>
2      2      TV Texas
3      3      TV  <NA>
4      4   Radio Texas
5      5   Radio  <NA>
6      6   Radio   NYC
>
```

• Return all rows from the left table, and any rows with matching keys from the right table.

To return all the rows from left table, we use **“Left Join”.**

left_join(df1, df2, by="CustId")

```
> #Left Join
> left_join(df1, df2, by="CustId")
  CustId Product State
1      1      TV  <NA>
2      2      TV Texas
3      3      TV  <NA>
4      4   Radio Texas
5      5   Radio  <NA>
6      6   Radio   NYC
>
```

• Return all rows from the right table, and any rows with matching keys from the left table.

To return all rows from the right table, we have to use **“right join’’.**

right_join(df1,df2, by="CustId")

```
> #right Join
> right_join(df1,df2, by="CustId")
  CustId Product State
1      2      TV Texas
2      4   Radio Texas
3      6   Radio   NYC
>
```

2. Perform the below operations on above given data frames and tables:

• Return a long format of the datasets without matching key.

Answer: We use product joins or Cartesian Joins to cross all the values of the dataset and form a long format. The output will 18 rows(6 from df1 and 3 from df2) , i.e. 6*3 =**18 rows**.

Merge(x=df1, y=df2, by=NULL)

```
> #Cartesian Join/Product Join
> merge(x=df1, y=df2, by=NULL)
   CustId.x Product CustId.y State
1         1      TV         2 Texas
2         2      TV         2 Texas
3         3      TV         2 Texas
4         4   Radio         2 Texas
5         5   Radio         2 Texas
6         6   Radio         2 Texas
7         1      TV         4 Texas
8         2      TV         4 Texas
9         3      TV         4 Texas
10        4   Radio         4 Texas
11        5   Radio         4 Texas
12        6   Radio         4 Texas
13        1      TV         6   NYC
14        2      TV         6   NYC
15        3      TV         6   NYC
16        4   Radio         6   NYC
17        5   Radio         6   NYC
18        6   Radio         6   NYC
```

• Keep only observations in df1 that match in df2.

Use **semi_join()** in dplyr package which will return only the observations from left table matching right table.

semi_join(df1,df2)

```
> semi_join(df1,df2)
Joining, by = "CustId"
  CustId Product
1      2      TV
2      4   Radio
3      6   Radio
>
```

• Drop all observations in df1 that match in df2.

Using ***anti_join()*** in dplyr package, it will **drop** the observations in df1 that match in df2

anti_join(df1, df2)

```
> anti_join(df1, df2)
Joining, by = "CustId"
  CustId Product
1      1      TV
2      3      TV
3      5   Radio
>
```