

PRACTICAL ASSIGNMENT - 4

Aim :- Write R program to count the number of NA values in a data frame column. Write R program to find elements come only once that are common to both given data frames.

Objective :- To count the number of NA values in a data frame and also know about natural join, left outer join, right outer join, full outer join & cross join data frames in R.

Theory :-

How to count NA values in dataframes?

→ Count the number of NA values in a data frame using the `is.na()` function in combination with `sum()` function.

eg:-
`vec <- sum(is.na(df))`
`print("count of NA in dataframe:")`
`print(vec)`

→ Count the number of NA values in a data frame in a specific column:

eg:-
~~`a <- colSums(is.na(df))`~~
~~`print(a)`~~

→ Count the number of NA values in a data frame in a specific row:

For Educational Use Only

eg:- $b \leftarrow \text{rowSums}(\text{is.na}(df))$
print(b)

→ For checking NA values in data frame:

→ Using the any() function to check if there are any NA values in the data frame df. If there's at least one NA value in df, 'any(is.na(df))' will return TRUE; otherwise, it will return FALSE.

eg:- $a \leftarrow \text{any}(\text{is.na}(df))$
print(a)

Merge data frames:- All forms of melt

The R merge function allows merging

two data frames by common columns or by row names.

The syntax of the R merge function with a brief description of its arguments is shown in the following block.

of code in (1) and → now - p

eg:- $df \leftarrow \text{merge}(x=df1, y=df2, by="ab")$
↓
print(df)

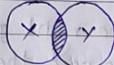
same
column name.

Note that the main method of the R merge function is for data frames. However, merge is a generic function that can be also used with other objects (like vectors or matrices), but they will be converted to data.frame class.

Inner join :-

An inner join (actually a natural join), is the most usual join of data sets that you can perform. It consists on merging two dataframes in one that contains the common elements of both, as described in the following illustration:

INNER JOIN



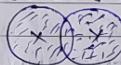
Syntax:-

```
merge(x = df_1, y = df_2, by = c("id", "name"))
```

Full outer join :-

The outer join, also known as full outer join or full join, merges all the columns of both data sets into one for all elements:

OUTER JOIN



In order to create a full outer join of the two data frames in R you have to set the argument `all` to TRUE:

Syntax:-

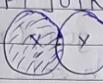
```
merge(x = df_1, y = df_2, all = TRUE)
```

Left (outer) join :-

The left join in R consist on matching all the rows in the first data frame with the corresponding values on the second.

For Educational Use Only

LEFT JOIN



select * from df1
left join df2 on df1.x = df2.y

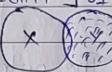
Syntax:-

merge (x = df1, y = df2, all, x = TRUE)

Right (Outer) Join :-

The right join in R is the opposite of the left outer join. In this case, the merge consists on joining all the rows in the second data frame with the corresponding rows in the first.

RIGHT JOIN



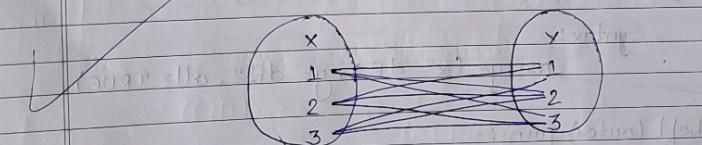
Syntax:-

merge (x = df1, y = df2, all, y = TRUE)

Cross join :-

The R cross join performs the Cartesian product

of the datasets df1 & df2.



Syntax : merge (x = df1, y = df2, by = NULL)

For Educational Use Only

Example:-

```
df1<-data.frame(ab=c(1,2,NA,4,NA),cd=c(NA,"hindi",  
"english","sst","phy"))
```

```
print(df1)
```

```
df2<-data.frame(ab=c(1,2,7,8,NA),de=c("it","comp",  
"mech","civil","cbs"))
```

```
print(df2)
```

```
df3<-merge(x=df1,y=df2,by="ab")  
print(df3)
```

```
df4<-merge(x=df1,y=df2,by="ab",all.x=TRUE)  
print(df4)
```

```
df5<-merge(x=df1,y=df2,by="ab",all.y=TRUE)  
print(df5)
```

```
df6<-merge(x=df1,y=df2,by="ab",all=TRUE)  
print(df6)
```

```
df7<-merge(x=df1,y=df2,by=NULL)  
print(df7).
```

For Educational Use Only