R Data Frame

A data frame is a two-dimensional data structure which can store data in tabular format. Data frames have rows and columns and each column can be a different vector. And different vectors can be of different data types.

Before we learn about Data Frames, make sure you know about *R vector*.

Create a Data Frame in R

In R, we use the `data.frame()` function to create a Data Frame.

The syntax of the `data.frame()` function is

```
dataframe1 <- data.frame(
   first_col  = c(val1, val2, ...),
   second_col = c(val1, val2, ...),
   ...
)
```

Here,

`first_col` - a vector with values `val1, val2, ...` of same data type

`second_col` - another vector with values `val1, val2, ...` of same data type and so on

Let's see an example,

```
# Create a data frame
dataframe1 <- data.frame (
  Name = c("Juan", "Alcaraz", "Simantha"),
  Age = c(22, 15, 19),
  Vote = c(TRUE, FALSE, TRUE)
)

print(dataframe1)
```

Output

```
      Name   Age    Vote
1     Juan    22    TRUE
2  Alcaraz    15   FALSE
3 Simantha    19    TRUE
```

In the above example, we have used the `data.frame()` function to create a data frame named `dataframe1`. Notice the arguments passed inside `data.frame()`,

```
data.frame (
  Name = c("Juan", "Alcaraz", "Simantha"),
  Age = c(22, 15, 19),
  Vote = c(TRUE, FALSE, TRUE)
)
```

Here, `Name`, `Age`, and `Vote` are column names for vectors of `String`, `Numeric`, and `Boolean` type respectively.

And finally the datas represented in tabular format are printed.

Access Data Frame Columns

There are different ways to extract columns from a data frame. We can use `[ ]`, `[[ ]]`, or `$` to access specific column of a data frame in R. For example,

```
# Create a data frame
```

```r
dataframe1 <- data.frame (
  Name = c("Juan", "Alcaraz", "Simantha"),
  Age = c(22, 15, 19),
  Vote = c(TRUE, FALSE, TRUE)
)

# pass index number inside [ ]
print(dataframe1[1])

# pass column name inside [[  ]]
print(dataframe1[["Name"]])

# use $ operator and column name
print(dataframe1$Name)
```

Output

```
     Name
1    Juan
2  Alcaraz
3 Simantha
[1] "Juan"     "Alcaraz"  "Simantha"
[1] "Juan"     "Alcaraz"  "Simantha"
```

In the above example, we have created a data frame named `dataframe1` with three columns `Name`, `Age`, `Vote`.

Here, we have used different operators to access `Name` column of `dataframe1`.

Accessing with `[[ ]]` or `$` is similar. However, it differs for `[ ]`, `[ ]` will return us a data frame but the other two will reduce it into a vector and return a vector.

Combine Data Frames

In R, we use the `rbind()` and the `cbind()` function to combine two data frames together.

`rbind()` - combines two data frames vertically

`cbind()` - combines two data frames horizontally

Combine Vertically Using rbind()

If we want to combine two data frames vertically, the column name of the two data frames must be the same. For example,

```r
# create a data frame
dataframe1 <- data.frame (
  Name = c("Juan", "Alcaraz"),
  Age = c(22, 15)
)

# create another data frame
dataframe2 <- data.frame (
  Name = c("Yiruma", "Bach"),
  Age = c(46, 89)
)

# combine two data frames vertically
updated <- rbind(dataframe1, dataframe2)
print(updated)
```

Output

```
       Name   Age
1      Juan    22
2   Alcaraz    15
3    Yiruma    46
4      Bach    89
```

Here, we have used the `rbind()` function to combine the two data frames: `dataframe1` and `dataframe2` vertically.

Combine Horizontally Using cbind()

The `cbind()` function combines two or more data frames horizontally. For example,

```
# create a data frame
dataframe1 <- data.frame (
  Name = c("Juan", "Alcaraz"),
  Age = c(22, 15)
)

# create another data frame
dataframe2 <- data.frame (
  Hobby = c("Tennis", "Piano")
)

# combine two data frames horizontally
updated <- cbind(dataframe1, dataframe2)
print(updated)
```

Output

```
      Name   Age   Hobby
1     Juan    22  Tennis
2 Alcaraz     15   Piano
```

Here, we have used `cbind()` to combine two data frames horizontally.

Note: The number of items on each vector of two or more combining data frames must be equal otherwise we will get an error: `arguments imply differing number of rows or columns`.

Length of a Data Frame in R

In R, we use the `length()` function to find the number of columns in a data frame. For example,

```
# Create a data frame
dataframe1 <- data.frame (
  Name = c("Juan", "Alcaraz", "Simantha"),
  Age = c(22, 15, 19),
  Vote = c(TRUE, FALSE, TRUE)
)

cat("Total Elements:", length(dataframe1))
```

Output

```
Total Elements: 3
```

Here, we have used `length()` to find the total number of columns in `dataframe1`. Since there are **3** columns, the `length()` function returns **3**.