

R Data Frames

Sachin B.

Accessing elements from Dataframe

```
# To see first several rows from dataframe  
head(mtcars)
```

1. head() and tail()

```
##           mpg cyl  disp  hp  drat    wt  qsec vs am gear carb  
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46 0  1   4    4  
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02 0  1   4    4  
## Datsun 710     22.8   4  108  93 3.85 2.320 18.61 1  1   4    1  
## Hornet 4 Drive  21.4   6  258 110 3.08 3.215 19.44 1  0   3    1  
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02 0  0   3    2  
## Valiant        18.1   6  225 105 2.76 3.460 20.22 1  0   3    1
```

```
head(mtcars,1) # Only first row
```

```
##           mpg cyl  disp  hp  drat    wt  qsec vs am gear carb  
## Mazda RX4    21    6  160 110   3.9 2.62 16.46 0  1    4    4
```

```
head(mtcars,-1) # all rows except last row
```

```
##           mpg cyl  disp  hp  drat    wt  qsec vs am gear carb  
## Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46 0  1   4    4  
## Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02 0  1   4    4  
## Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61 1  1   4    1  
## Hornet 4 Drive  21.4   6 258.0 110 3.08 3.215 19.44 1  0   3    1  
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02 0  0   3    2  
## Valiant        18.1   6 225.0 105 2.76 3.460 20.22 1  0   3    1  
## Duster 360     14.3   8 360.0 245 3.21 3.570 15.84 0  0   3    4  
## Merc 240D      24.4   4 146.7  62 3.69 3.190 20.00 1  0   4    2  
## Merc 230       22.8   4 140.8  95 3.92 3.150 22.90 1  0   4    2  
## Merc 280       19.2   6 167.6 123 3.92 3.440 18.30 1  0   4    4  
## Merc 280C      17.8   6 167.6 123 3.92 3.440 18.90 1  0   4    4  
## Merc 450SE     16.4   8 275.8 180 3.07 4.070 17.40 0  0   3    3  
## Merc 450SL     17.3   8 275.8 180 3.07 3.730 17.60 0  0   3    3  
## Merc 450SLC    15.2   8 275.8 180 3.07 3.780 18.00 0  0   3    3  
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98 0  0   3    4  
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82 0  0   3    4  
## Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42 0  0   3    4
```

```
## Fiat 128      32.4  4  78.7  66 4.08 2.200 19.47  1  1   4   1
## Honda Civic  30.4  4  75.7  52 4.93 1.615 18.52  1  1   4   2
## Toyota Corolla 33.9  4  71.1  65 4.22 1.835 19.90  1  1   4   1
## Toyota Corona 21.5  4 120.1  97 3.70 2.465 20.01  1  0   3   1
## Dodge Challenger 15.5  8 318.0 150 2.76 3.520 16.87  0  0   3   2
## AMC Javelin  15.2  8 304.0 150 3.15 3.435 17.30  0  0   3   2
## Camaro Z28   13.3  8 350.0 245 3.73 3.840 15.41  0  0   3   4
## Pontiac Firebird 19.2  8 400.0 175 3.08 3.845 17.05  0  0   3   2
## Fiat X1-9    27.3  4  79.0  66 4.08 1.935 18.90  1  1   4   1
## Porsche 914-2 26.0  4 120.3  91 4.43 2.140 16.70  0  1   5   2
## Lotus Europa 30.4  4  95.1 113 3.77 1.513 16.90  1  1   5   2
## Ford Pantera L 15.8  8 351.0 264 4.22 3.170 14.50  0  1   5   4
## Ferrari Dino  19.7  6 145.0 175 3.62 2.770 15.50  0  1   5   6
## Maserati Bora 15.0  8 301.0 335 3.54 3.570 14.60  0  1   5   8
```

```
# To see last several rows from dataframe
tail(mtcars)
```

```
##          mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Porsche 914-2 26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
## Lotus Europa  30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
## Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
```

```
tail(mtcars,1) # Only Last row
```

```
##          mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Volvo 142E 21.4   4 121 109 4.11 2.78 18.6  1  1    4    2
```

```
tail(mtcars,-1) # all rows except first row
```

```
##          mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710    22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7  8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant       18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360    14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D     24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230      22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280      19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C     17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE    16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL    17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC   15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood 10.4  8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4  8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial 14.7  8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128      32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic   30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
```

```
## Toyota Corolla      33.9   4  71.1   65 4.22 1.835 19.90 1 1   4   1
## Toyota Corona      21.5   4 120.1   97 3.70 2.465 20.01 1 0   3   1
## Dodge Challenger    15.5   8 318.0  150 2.76 3.520 16.87 0 0   3   2
## AMC Javelin         15.2   8 304.0  150 3.15 3.435 17.30 0 0   3   2
## Camaro Z28          13.3   8 350.0  245 3.73 3.840 15.41 0 0   3   4
## Pontiac Firebird    19.2   8 400.0  175 3.08 3.845 17.05 0 0   3   2
## Fiat X1-9           27.3   4   79.0   66 4.08 1.935 18.90 1 1   4   1
## Porsche 914-2       26.0   4 120.3   91 4.43 2.140 16.70 0 1   5   2
## Lotus Europa        30.4   4   95.1  113 3.77 1.513 16.90 1 1   5   2
## Ford Pantera L      15.8   8 351.0  264 4.22 3.170 14.50 0 1   5   4
## Ferrari Dino        19.7   6 145.0  175 3.62 2.770 15.50 0 1   5   6
## Maserati Bora       15.0   8 301.0  335 3.54 3.570 14.60 0 1   5   8
## Volvo 142E          21.4   4 121.0  109 4.11 2.780 18.60 1 1   4   2
```

```
# Create Data Frame
```

```
df <- data.frame(
  Roll=1:4,
  Name=c('Ankit', 'Sara', 'Pravin', 'Kiran'),
  Gender=c("Male", "Female", "Male", "Female"),
  Pointer=c(8.23, 6.0, 4.2, 10),
  Rank=c(2, 3, 4, 1),
  stringsAsFactors = FALSE
)

print(df)
```

2. dataframe[rows,columns]

```
##   Roll  Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    2
## 2    2   Sara Female    6.00    3
## 3    3 Pravin   Male    4.20    4
## 4    4  Kiran Female   10.00    1
```

```
# Accessing First Row and Second Column
```

```
df[1,2]
```

```
## [1] "Ankit"
```

```
# Accessing First Row and All Columns
```

```
df[1,]
```

```
##   Roll  Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    2
```

```
# Accessing First, Second and Third Rows and All Columns
```

```
df[1:3,]
```

```
##   Roll   Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    2
## 2    2   Sara Female    6.00    3
## 3    3 Pravin   Male    4.20    4
```

```
# Accessing Second and Forth Rows and All Columns
df[c(2,4),]
```

```
##   Roll   Name Gender Pointer Rank
## 2    2   Sara Female     6      3
## 4    4  Kiran Female    10      1
```

```
# Print Data Frame
print(df)
```

```
##   Roll   Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    2
## 2    2   Sara Female    6.00    3
## 3    3 Pravin   Male    4.20    4
## 4    4  Kiran Female   10.00    1
```

```
# Accessing All Rows and Second Column
df[,2]
```

```
## [1] "Ankit" "Sara" "Pravin" "Kiran"
```

```
df[2] # When only 1 parameter is passed; It considered as column.
```

```
##      Name
## 1  Ankit
## 2   Sara
## 3 Pravin
## 4  Kiran
```

```
# Accessing All Rows and Second, Third & Forth Columns
df[,2:4]
```

```
##      Name Gender Pointer
## 1  Ankit   Male    8.23
## 2   Sara Female    6.00
## 3 Pravin   Male    4.20
## 4  Kiran Female   10.00
```

```
# Accessing First and Forth Columns and All Rows
df[,c(1,4)]
```

```
##   Roll Pointer
## 1    1    8.23
## 2    2    6.00
## 3    3    4.20
## 4    4   10.00
```

```
# Select Rows 2 to 4 and columns 3 to 5
df[2:4, 3:5]
```

```
##   Gender Pointer Rank
## 2 Female      6.0    3
## 3   Male      4.2    4
## 4 Female     10.0    1
```

```
df["Name"]
```

3. Accessing Column using Column Name (Same as List)

```
##   Name
## 1 Ankit
## 2 Sara
## 3 Pravin
## 4 Kiran
```

```
df$Name
```

```
## [1] "Ankit" "Sara" "Pravin" "Kiran"
```

```
df[["Name"]]
```

```
## [1] "Ankit" "Sara" "Pravin" "Kiran"
```

```
df[[2]]
```

```
## [1] "Ankit" "Sara" "Pravin" "Kiran"
```

```
# Print Data Frame
print(df)
```

4. Slice with columns name

```
##   Roll  Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    2
## 2    2   Sara Female    6.00    3
## 3    3 Pravin   Male    4.20    4
## 4    4  Kiran Female   10.00    1
```

```
df[, c('Name', 'Rank')]
```

```
##      Name Rank
## 1  Ankit    2
## 2   Sara    3
## 3 Pravin    4
## 4  Kiran    1
```

```
df[c(2,4), c('Name', 'Rank')]
```

```
##      Name Rank
## 2   Sara    3
## 4  Kiran    1
```

```
new_df=data.frame(New_Name=df$Name,New_Rank=df$Rank)
new_df
```

```
##      New_Name New_Rank
## 1     Ankit         2
## 2      Sara         3
## 3    Pravin         4
## 4     Kiran         1
```

Subset a Data Frame

```
# Create Data Frame
df <- data.frame(
  Roll=1:5,
  Name=c('Ankit', 'Sara', 'Pravin', 'Kiran', 'Vishal'),
  Gender=c("Male", "Female", "Male", "Female", "Male"),
  Pointer=c(8.23, 6.0, 4.2, 10, 9.22),
  Rank=c(3, 4, 5, 1, 2),
  stringsAsFactors = FALSE
)
```

```
# Selecting the subset of the data frame
# where Pointer is less than 8
```

```
df_subset1 = subset(df, Pointer < 8)
```

```
df_subset1
```

```
##      Roll  Name Gender Pointer Rank
## 2      2   Sara Female     6.0    4
## 3      3 Pravin  Male     4.2    5
```

```
# Selecting the subset of the data frame
# where Gender is Male
# AND Pointer is less than 9
```

```
df_subset2 = subset(df, Pointer < 9 & Gender == "Male")
df_subset2
```

```
##   Roll   Name Gender Pointer Rank
## 1     1  Ankit   Male    8.23    3
## 3     3 Pravin   Male    4.20    5
```

```
df_subset2 = subset(df, Pointer < 9 & Gender == "Female") # Gender = "Female"
df_subset2
```

```
##   Roll Name Gender Pointer Rank
## 2     2 Sara Female        6    4
```

```
# Difference between "&" and "&&"
df_subset3 = subset(df, Pointer < 9 && Gender == "Male")
df_subset3
```

```
##   Roll   Name Gender Pointer Rank
## 1     1  Ankit   Male    8.23    3
## 2     2   Sara Female    6.00    4
## 3     3 Pravin   Male    4.20    5
## 4     4 Kiran Female   10.00    1
## 5     5 Vishal   Male    9.22    2
```

```
# Difference between "&" and "&&"
df_subset3 = subset(df, Pointer < 9 && Gender == "Female")
df_subset3
```

```
## [1] Roll   Name   Gender Pointer Rank
## <0 rows> (or 0-length row.names)
```

```
# Selecting Roll and Name from data frame
# where Gender is Male
# AND Pointer is less than 9
df_subset4 <- subset(df, select=c(Roll, Name), subset=(Gender=="Male" & Pointer < 9))
df_subset4
```

```
##   Roll   Name
## 1     1  Ankit
## 3     3 Pravin
```

Other Ways to Subset A Data Frame in R

```
# Print Data Frame
print(df)
```

```
##   Roll   Name Gender Pointer Rank
## 1     1  Ankit   Male    8.23    3
## 2     2   Sara Female    6.00    4
## 3     3 Pravin   Male    4.20    5
## 4     4 Kiran Female   10.00    1
## 5     5 Vishal   Male    9.22    2
```

```
# subset in r - conditional indexing
df_subset5 <- df[df$Gender=="Female",]
df_subset5
```

```
##   Roll  Name Gender Pointer Rank
## 2    2  Sara Female      6     4
## 4    4 Kiran Female     10     1
```

```
# subset in r data frame multiple conditions
df_subset6 <- df[(df$Gender=="Male") & (df$Pointer<9),]
df_subset6
```

```
##   Roll  Name Gender Pointer Rank
## 1    1  Ankit  Male   8.23     3
## 3    3 Pravin  Male   4.20     5
```

```
# subset in r data frame multiple conditions and selected columns
df_subset7 <- df[(df$Gender=="Male") & (df$Pointer<9),c(2,3)]
df_subset7
```

```
##      Name Gender
## 1  Ankit  Male
## 3 Pravin  Male
```

```
# subset in r data frame multiple conditions and selected column using names
df_subset7 <- df[which((df$Gender=="Male") & (df$Pointer<9)),c("Roll","Name")] # Which command removes .
df_subset7
```

```
##   Roll  Name
## 1    1  Ankit
## 3    3 Pravin
```

Modifying elements from Dataframe

```
# Create Data Frame
df <- data.frame(
  Roll=1:5,
  Name=c('Ankit', 'Sara', 'Pravin', 'Kiran', 'Vishal'),
  Gender=c("Male", "Female", "Male", "Female", "Male"),
  Pointer=c(8.23, 6.0, 4.2, 10, 9.22),
  Rank=c(3, 4, 5, 1, 2),
  stringsAsFactors = FALSE
)

# Print Data Frame
print(df)
```

```
##   Roll  Name Gender Pointer Rank
## 1    1  Ankit  Male   8.23     3
```



```
## 2    2    Sara Female    6.00    4
## 3    3 Pravin   Male    4.20    5
## 4    4    Kiran Female   10.00    1
## 5    5 Vishal   Male    9.22    2
```

```
# Changing Element at 1st Row and 2nd Column
```

```
df[1,2] <- "Ravi"
```

```
print(df)
```

```
##   Roll   Name Gender Pointer Rank
## 1    1    Ravi   Male    8.23    3
## 2    2    Sara Female    6.00    4
## 3    3 Pravin   Male    4.20    5
## 4    4    Kiran Female   10.00    1
## 5    5 Vishal   Male    9.22    2
```

```
# Changing Element at 3rd Row and "Pointer" Column
```

```
df[3,"Pointer"] <- 5.99
```

```
print(df)
```

```
##   Roll   Name Gender Pointer Rank
## 1    1    Ravi   Male    8.23    3
## 2    2    Sara Female    6.00    4
## 3    3 Pravin   Male    5.99    5
## 4    4    Kiran Female   10.00    1
## 5    5 Vishal   Male    9.22    2
```

```
# Method 1 -> [row,column] : Changing Entire 3rd Column
```

```
df[,3] <- c("M","F","M","F","M")
```

```
print(df)
```

```
##   Roll   Name Gender Pointer Rank
## 1    1    Ravi      M    8.23    3
## 2    2    Sara      F    6.00    4
## 3    3 Pravin      M    5.99    5
## 4    4    Kiran      F   10.00    1
## 5    5 Vishal      M    9.22    2
```

```
# Method 1 -> [row,column][row] : Changing 2nd row of 3rd Column
```

```
df[,3][2] <- "Woman"
```

```
print(df)
```

```
##   Roll   Name Gender Pointer Rank
## 1    1    Ravi      M    8.23    3
## 2    2    Sara  Woman    6.00    4
```

```
## 3    3 Pravin      M    5.99    5
## 4    4 Kiran      F   10.00    1
## 5    5 Vishal     M    9.22    2
```

```
# Method 2 -> [[column]] : Changing Entire 3rd Column
df[[3]] <- c("Male","Female","Male","Female","Male")

print(df)
```

```
##   Roll  Name Gender Pointer Rank
## 1    1   Ravi  Male    8.23    3
## 2    2   Sara Female    6.00    4
## 3    3 Pravin  Male    5.99    5
## 4    4 Kiran Female   10.00    1
## 5    5 Vishal  Male    9.22    2
```

```
# Method 2 -> [[column]][row] : Changing 4th row of 3rd Column
df[[3]][4] <- "Man"

print(df)
```

```
##   Roll  Name Gender Pointer Rank
## 1    1   Ravi  Male    8.23    3
## 2    2   Sara Female    6.00    4
## 3    3 Pravin  Male    5.99    5
## 4    4 Kiran   Man   10.00    1
## 5    5 Vishal  Male    9.22    2
```

```
# Method 3 -> dataframe$column : Changing 5th Column by name "Rank"
df$Rank <-c("III","IV","V","I","II")

print(df)
```

```
##   Roll  Name Gender Pointer Rank
## 1    1   Ravi  Male    8.23  III
## 2    2   Sara Female    6.00  IV
## 3    3 Pravin  Male    5.99   V
## 4    4 Kiran   Man   10.00   I
## 5    5 Vishal  Male    9.22  II
```

```
# Method 3 -> dataframe$column[row] : Changing 4th row of 5th Column by name "Rank"
df$Rank[4] <- "One"

print(df)
```

```
##   Roll  Name Gender Pointer Rank
## 1    1   Ravi  Male    8.23  III
## 2    2   Sara Female    6.00  IV
## 3    3 Pravin  Male    5.99   V
## 4    4 Kiran   Man   10.00 One
## 5    5 Vishal  Male    9.22  II
```

```
# Changing Entire 4th Row
df[4,] <- list(4,"Karan","Male",9.8,"I")
print(df)
```

```
##   Roll   Name Gender Pointer Rank
## 1    1    Ravi   Male    8.23   III
## 2    2    Sara Female    6.00    IV
## 3    3 Pravin   Male    5.99     V
## 4    4    Karan   Male    9.80     I
## 5    5 Vishal   Male    9.22    II
```

Expand Dataframe

```
# Create Data Frame
df <- data.frame(
  Roll=1:5,
  Name=c('Ankit', 'Sara','Pravin', 'Kiran','Vishal'),
  Gender=c("Male","Female","Male","Female","Male"),
  Pointer=c(8.23, 6.0, 4.2, 10, 9.22),
  Rank=c(3, 4, 5, 1,2),
  stringsAsFactors = FALSE
)

# Print Data Frame
print(df)
```

1. Add New Row [rbind()]

```
##   Roll   Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    3
## 2    2   Sara Female    6.00    4
## 3    3 Pravin   Male    4.20    5
## 4    4  Kiran Female   10.00    1
## 5    5 Vishal   Male    9.22    2
```

```
# Adding new Row to Data Frame
df_new = rbind(df,list(6,"Sam","Male",3.9,6))

print(df_new)
```

```
##   Roll   Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    3
## 2    2   Sara Female    6.00    4
## 3    3 Pravin   Male    4.20    5
## 4    4  Kiran Female   10.00    1
## 5    5 Vishal   Male    9.22    2
## 6    6    Sam   Male    3.90    6
```

```
# Adding new Row to Data Frame
df_new2 = rbind(df, data.frame(Roll = 6, Name = "Sam", Gender = "Male", Pointer=3.9, Rank=6))

print(df_new)
```

```
##   Roll   Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    3
## 2    2   Sara Female    6.00    4
## 3    3 Pravin   Male    4.20    5
## 4    4  Kiran Female   10.00    1
## 5    5 Vishal   Male    9.22    2
## 6    6    Sam   Male    3.90    6
```

```
# Create the second data frame with single element
df2 <- data.frame(Roll = 6,
                  Name = "Sam",
                  Gender = "Male",
                  Pointer=3.9,
                  Rank=6,
                  stringsAsFactors = FALSE
                )
```

```
# Bind the two data frames.
df_new3 <- rbind(df, df2)
print(df_new3)
```

```
##   Roll   Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    3
## 2    2   Sara Female    6.00    4
## 3    3 Pravin   Male    4.20    5
## 4    4  Kiran Female   10.00    1
## 5    5 Vishal   Male    9.22    2
## 6    6    Sam   Male    3.90    6
```

```
# Create the second data frame with multiple elements
df2 <- data.frame(Roll = 6:7,
                  Name = c("Sam", "Siya"),
                  Gender = c("Male", "Female"),
                  Pointer=c(3.90, 4.00),
                  Rank=c(7, 6),
                  stringsAsFactors = FALSE
                )
```

```
# Bind the two data frames.
df_new4 <- rbind(df, df2)
print(df_new4)
```

```
##   Roll   Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    3
## 2    2   Sara Female    6.00    4
## 3    3 Pravin   Male    4.20    5
## 4    4  Kiran Female   10.00    1
```

```
## 5    5 Vishal   Male    9.22    2
## 6    6    Sam    Male    3.90    7
## 7    7    Siya  Female   4.00    6
```

```
# Create Data Frame
df <- data.frame(
  Roll=1:5,
  Name=c('Ankit', 'Sara', 'Pravin', 'Kiran', 'Vishal'),
  Gender=c("Male", "Female", "Male", "Female", "Male"),
  Pointer=c(8.23, 6.0, 4.2, 10, 9.22),
  Rank=c(3, 4, 5, 1, 2),
  stringsAsFactors = FALSE
)

# Print Data Frame
print(df)
```

2. Add New Column [cbind()]

```
##   Roll   Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    3
## 2    2   Sara Female    6.00    4
## 3    3 Pravin   Male    4.20    5
## 4    4  Kiran Female   10.00    1
## 5    5 Vishal   Male    9.22    2
```

```
# Adding new Column to Data Frame
df = cbind(df, Branch=c("CS", "IT", "IT", "IT", "CS"))

print(df)
```

```
##   Roll   Name Gender Pointer Rank Branch
## 1    1  Ankit   Male    8.23    3    CS
## 2    2   Sara Female    6.00    4    IT
## 3    3 Pravin   Male    4.20    5    IT
## 4    4  Kiran Female   10.00    1    IT
## 5    5 Vishal   Male    9.22    2    CS
```

```
df$Age <- c(20, 19, 21, 18, 22)
print(df)
```

```
##   Roll   Name Gender Pointer Rank Branch Age
## 1    1  Ankit   Male    8.23    3    CS  20
## 2    2   Sara Female    6.00    4    IT  19
## 3    3 Pravin   Male    4.20    5    IT  21
## 4    4  Kiran Female   10.00    1    IT  18
## 5    5 Vishal   Male    9.22    2    CS  22
```

Delete Rows and Columns in a data frame

```
# Create Data Frame
df <- data.frame(
  Roll=1:5,
  Name=c('Ankit', 'Sara', 'Pravin', 'Kiran', 'Vishal'),
  Gender=c("Male", "Female", "Male", "Female", "Male"),
  Pointer=c(8.23, 6.0, 4.2, 10, 9.22),
  Rank=c(3, 4, 5, 1, 2),
  stringsAsFactors = FALSE
)

# Print Data Frame
print(df)
```

```
##   Roll   Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23    3
## 2    2   Sara Female    6.00    4
## 3    3 Pravin   Male    4.20    5
## 4    4  Kiran Female   10.00    1
## 5    5 Vishal   Male    9.22    2
```

```
df$Rank <- NULL

print(df)
```

Delete Data Frame columns by assigning NULL

```
##   Roll   Name Gender Pointer
## 1    1  Ankit   Male    8.23
## 2    2   Sara Female    6.00
## 3    3 Pravin   Male    4.20
## 4    4  Kiran Female   10.00
## 5    5 Vishal   Male    9.22
```

```
# Delete Data Frame First Row by Re-assignment
df <- df[-1,]
print(df)
```

Delete Data Frame Data by Re-assignment

```
##   Roll   Name Gender Pointer
## 2    2   Sara Female    6.00
## 3    3 Pravin   Male    4.20
## 4    4  Kiran Female   10.00
## 5    5 Vishal   Male    9.22
```

```
# Delete Data Frame Column "Gender" by Re-assignment
df <- df[,-3]
print(df)
```

```
##   Roll   Name Pointer
## 2    2   Sara    6.00
## 3    3 Pravin    4.20
## 4    4  Kiran   10.00
## 5    5 Vishal    9.22
```

Recasting (Melt & Cast)

```
# Create Data Frame
df <- data.frame(
  Name=c('A', 'A', 'B', 'B'),
  Month=c("Jan", "Feb", "Jan", "Feb"),
  BS=c(141.3, 139.3, 135.2, 160.1),
  BP=c(90, 78, 80, 81),
  stringsAsFactors = FALSE
)

# Print Data Frame
print(df)
```

```
##   Name Month   BS BP
## 1    A   Jan 141.3 90
## 2    A   Feb 139.3 78
## 3    B   Jan 135.2 80
## 4    B   Feb 160.1 81
```

```
# Install Package: "reshape2"

install.packages("reshape2")
```

```
library(reshape2) # Load "reshape2" Package
```

```
## Warning: package 'reshape2' was built under R version 4.0.3
```

```
df_melt <- melt(df, id.vars = c("Name", "Month"), measure.vars = c("BS", "BP"))
print(df_melt)
```

Melt

```
##   Name Month variable value
## 1    A   Jan        BS 141.3
```

```
## 2    A    Feb      BS 139.3
## 3    B    Jan      BS 135.2
## 4    B    Feb      BS 160.1
## 5    A    Jan      BP  90.0
## 6    A    Feb      BP  78.0
## 7    B    Jan      BP  80.0
## 8    B    Feb      BP  81.0
```

```
df_cast <- dcast(df_melt,variable+Month~Name,value.var = "value")

print(df_cast)
```

Cast

```
##   variable Month      A      B
## 1      BS   Feb 139.3 160.1
## 2      BS   Jan 141.3 135.2
## 3      BP   Feb  78.0  81.0
## 4      BP   Jan  90.0  80.0
```

Recasting -> recast()

recast

```
df_recast <- recast(df,variable+Month~Name,id.var = c("Name","Month"),measure.var = c("BS","BP"))

print(df_recast)
```

```
##   variable Month      A      B
## 1      BS   Feb 139.3 160.1
## 2      BS   Jan 141.3 135.2
## 3      BP   Feb  78.0  81.0
## 4      BP   Jan  90.0  80.0
```

Recasting (Practice)

```
# Data Frame "mtcars" from Sample Data
data("mtcars")

# viewing head() and tail() of data

head(mtcars)
```

Using dataframe / Creating Data frame


```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46 0 1    4    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02 0 1    4    4
## Datsun 710     22.8   4  108  93 3.85 2.320 18.61 1 1    4    1
## Hornet 4 Drive  21.4   6  258 110 3.08 3.215 19.44 1 0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02 0 0    3    2
## Valiant        18.1   6  225 105 2.76 3.460 20.22 1 0    3    1
```

```
tail(mtcars)
```

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0 1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1 1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0 1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0 1    5    6
## Maserati Bora   15.0   8 301.0 335 3.54 3.570 14.6  0 1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1 1    4    2
```

```
# Creating new Column "name" and add row names vector to that column
mtcars$name <- rownames(mtcars)
head(mtcars)
```

```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46 0 1    4    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02 0 1    4    4
## Datsun 710     22.8   4  108  93 3.85 2.320 18.61 1 1    4    1
## Hornet 4 Drive  21.4   6  258 110 3.08 3.215 19.44 1 0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02 0 0    3    2
## Valiant        18.1   6  225 105 2.76 3.460 20.22 1 0    3    1
##                name
## Mazda RX4        Mazda RX4
## Mazda RX4 Wag    Mazda RX4 Wag
## Datsun 710        Datsun 710
## Hornet 4 Drive    Hornet 4 Drive
## Hornet Sportabout Hornet Sportabout
## Valiant           Valiant
```

```
# Melt
```

```
mtcars_melt <- melt(mtcars, id.vars = c("name", "gear", "cyl"),
  measure.vars = c("mpg", "hp")
)
print(mtcars_melt)
```

Melting dataframe

```
##           name gear cyl variable value
## 1      Mazda RX4    4   6      mpg  21.0
## 2  Mazda RX4 Wag    4   6      mpg  21.0
```

## 3	Datsun 710	4	4	mpg	22.8
## 4	Hornet 4 Drive	3	6	mpg	21.4
## 5	Hornet Sportabout	3	8	mpg	18.7
## 6	Valiant	3	6	mpg	18.1
## 7	Duster 360	3	8	mpg	14.3
## 8	Merc 240D	4	4	mpg	24.4
## 9	Merc 230	4	4	mpg	22.8
## 10	Merc 280	4	6	mpg	19.2
## 11	Merc 280C	4	6	mpg	17.8
## 12	Merc 450SE	3	8	mpg	16.4
## 13	Merc 450SL	3	8	mpg	17.3
## 14	Merc 450SLC	3	8	mpg	15.2
## 15	Cadillac Fleetwood	3	8	mpg	10.4
## 16	Lincoln Continental	3	8	mpg	10.4
## 17	Chrysler Imperial	3	8	mpg	14.7
## 18	Fiat 128	4	4	mpg	32.4
## 19	Honda Civic	4	4	mpg	30.4
## 20	Toyota Corolla	4	4	mpg	33.9
## 21	Toyota Corona	3	4	mpg	21.5
## 22	Dodge Challenger	3	8	mpg	15.5
## 23	AMC Javelin	3	8	mpg	15.2
## 24	Camaro Z28	3	8	mpg	13.3
## 25	Pontiac Firebird	3	8	mpg	19.2
## 26	Fiat X1-9	4	4	mpg	27.3
## 27	Porsche 914-2	5	4	mpg	26.0
## 28	Lotus Europa	5	4	mpg	30.4
## 29	Ford Pantera L	5	8	mpg	15.8
## 30	Ferrari Dino	5	6	mpg	19.7
## 31	Maserati Bora	5	8	mpg	15.0
## 32	Volvo 142E	4	4	mpg	21.4
## 33	Mazda RX4	4	6	hp	110.0
## 34	Mazda RX4 Wag	4	6	hp	110.0
## 35	Datsun 710	4	4	hp	93.0
## 36	Hornet 4 Drive	3	6	hp	110.0
## 37	Hornet Sportabout	3	8	hp	175.0
## 38	Valiant	3	6	hp	105.0
## 39	Duster 360	3	8	hp	245.0
## 40	Merc 240D	4	4	hp	62.0
## 41	Merc 230	4	4	hp	95.0
## 42	Merc 280	4	6	hp	123.0
## 43	Merc 280C	4	6	hp	123.0
## 44	Merc 450SE	3	8	hp	180.0
## 45	Merc 450SL	3	8	hp	180.0
## 46	Merc 450SLC	3	8	hp	180.0
## 47	Cadillac Fleetwood	3	8	hp	205.0
## 48	Lincoln Continental	3	8	hp	215.0
## 49	Chrysler Imperial	3	8	hp	230.0
## 50	Fiat 128	4	4	hp	66.0
## 51	Honda Civic	4	4	hp	52.0
## 52	Toyota Corolla	4	4	hp	65.0
## 53	Toyota Corona	3	4	hp	97.0
## 54	Dodge Challenger	3	8	hp	150.0
## 55	AMC Javelin	3	8	hp	150.0
## 56	Camaro Z28	3	8	hp	245.0

```
## 57 Pontiac Firebird 3 8 hp 175.0
## 58 Fiat X1-9 4 4 hp 66.0
## 59 Porsche 914-2 5 4 hp 91.0
## 60 Lotus Europa 5 4 hp 113.0
## 61 Ford Pantera L 5 8 hp 264.0
## 62 Ferrari Dino 5 6 hp 175.0
## 63 Maserati Bora 5 8 hp 335.0
## 64 Volvo 142E 4 4 hp 109.0
```

```
# Cast Length
mtcars_Cyl_length <- dcast(mtcars_melt,cyl~variable,length)

print(mtcars_Cyl_length)
```

Casting dataframe

```
## cyl mpg hp
## 1 4 11 11
## 2 6 7 7
## 3 8 14 14
```

```
# Cast Mean
mtcars_Cyl_mean <- dcast(mtcars_melt,cyl~variable,mean)

print(mtcars_Cyl_mean)
```

```
## cyl mpg hp
## 1 4 26.66364 82.63636
## 2 6 19.74286 122.28571
## 3 8 15.10000 209.21429
```

```
# Cast Median
mtcars_Cyl_median <- dcast(mtcars_melt,cyl~variable,median)

print(mtcars_Cyl_median)
```

```
## cyl mpg hp
## 1 4 26.0 91.0
## 2 6 19.7 110.0
## 3 8 15.2 192.5
```

```
# Cast Name Length
mtcars_name_length <- dcast(mtcars_melt,name~variable,length)

print(mtcars_name_length)
```

```
## name mpg hp
## 1 AMC Javelin 1 1
## 2 Cadillac Fleetwood 1 1
```

```
## 3      Camaro Z28      1  1
## 4    Chrysler Imperial  1  1
## 5      Datsun 710      1  1
## 6    Dodge Challenger  1  1
## 7      Duster 360      1  1
## 8      Ferrari Dino    1  1
## 9      Fiat 128        1  1
## 10     Fiat X1-9       1  1
## 11     Ford Pantera L   1  1
## 12     Honda Civic      1  1
## 13     Hornet 4 Drive   1  1
## 14    Hornet Sportabout 1  1
## 15 Lincoln Continental  1  1
## 16     Lotus Europa     1  1
## 17     Maserati Bora     1  1
## 18     Mazda RX4        1  1
## 19     Mazda RX4 Wag    1  1
## 20      Merc 230         1  1
## 21      Merc 240D        1  1
## 22      Merc 280         1  1
## 23      Merc 280C        1  1
## 24      Merc 450SE       1  1
## 25      Merc 450SL       1  1
## 26      Merc 450SLC      1  1
## 27    Pontiac Firebird   1  1
## 28     Porsche 914-2     1  1
## 29     Toyota Corolla    1  1
## 30     Toyota Corona     1  1
## 31      Valiant          1  1
## 32     Volvo 142E        1  1
```

```
# Cast Gear Median
mtcars_gear_median <- dcast(mtcars_melt, gear ~ variable, median)

print(mtcars_gear_median)
```

```
##   gear  mpg  hp
## 1    3 15.5 180
## 2    4 22.8  94
## 3    5 19.7 175
```

Merge

```
# Create Data Frame
df_x <- data.frame(
  id=c(3,6,1,1,4),
  post=c("Engineer", "Trainee", "Founder", "CEO", "Office Boy"),
  stringsAsFactors = FALSE
)
```

```
# Print Data Frame
print(df_x)
```

Creating Dataframe

```
##   id      post
## 1  3  Engineer
## 2  6   Trainee
## 3  1   Founder
## 4  1        CEO
## 5  4 Office Boy
```

```
# Create Data Frame
df_y <- data.frame(
  eid=c(1:5),
  name=c("Alex","Ben","Cavin","Dan","Ellis"),
  stringsAsFactors = FALSE
)

# Print Data Frame
print(df_y)
```

```
##   eid  name
## 1   1  Alex
## 2   2   Ben
## 3   3 Cavin
## 4   4   Dan
## 5   5  Ellis
```

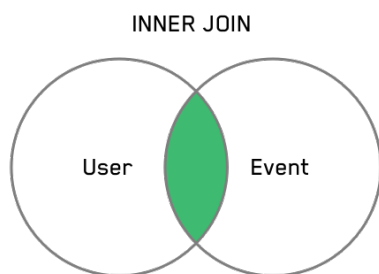


Figure 1:

1. Inner Join

```
# Print Data Frames
print(df_x)
```

When column names are different

```
##   id      post
## 1  3   Engineer
## 2  6   Trainee
## 3  1   Founder
## 4  1      CEO
## 5  4 Office Boy
```

```
print(df_y)
```

```
##   eid name
## 1  1  Alex
## 2  2   Ben
## 3  3  Cavin
## 4  4   Dan
## 5  5  Ellis
```

```
# Inner Join Operation
df_inner <- merge(df_x,df_y,by.x="id",by.y="eid")
print(df_inner)
```

```
##   id      post name
## 1  1   Founder Alex
## 2  1      CEO  Alex
## 3  3   Engineer Cavin
## 4  4 Office Boy  Dan
```

```
# Data frame column name changed from "eid" -> "id"
names(df_y)[1] <- "id"

# Print Data Frames
print(df_x)
```

When column names are same

```
##   id      post
## 1  3   Engineer
## 2  6   Trainee
## 3  1   Founder
## 4  1      CEO
## 5  4 Office Boy
```

```
print(df_y)
```

```
##   id name
## 1  1  Alex
## 2  2   Ben
## 3  3  Cavin
## 4  4   Dan
## 5  5  Ellis
```

```

# Inner Join Operation
df_natural <- merge(df_x,df_y,by="id")

print(df_natural)

##   id      post  name
## 1  1   Founder  Alex
## 2  1      CEO  Alex
## 3  3 Engineer  Cavin
## 4  4 Office Boy  Dan

# Data frame column name changed from "id" -> "eid"
names(df_y)[1] <- "eid"

```

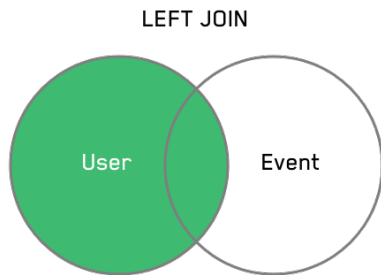


Figure 2:

```

# Print Data Frames
print(df_x)

```

2. Left Outer Join

```

##   id      post
## 1  3 Engineer
## 2  6  Trainee
## 3  1  Founder
## 4  1      CEO
## 5  4 Office Boy

```

```

print(df_y)

```

```

##   eid  name
## 1   1  Alex
## 2   2   Ben
## 3   3 Cavin
## 4   4   Dan
## 5   5 Ellis

```

```
# Left Outer Join Operation
```

```
df_left_outer <- merge(df_x,df_y,by.x="id",by.y="eid",all.x = TRUE)  
print(df_left_outer)
```

```
##   id      post  name  
## 1 1    Founder  Alex  
## 2 1      CEO    Alex  
## 3 3  Engineer  Cavin  
## 4 4 Office Boy  Dan  
## 5 6   Trainee  <NA>
```

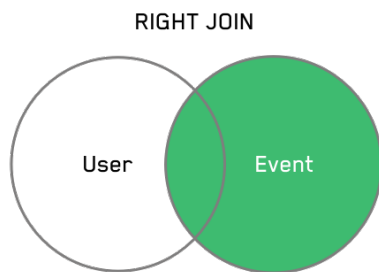


Figure 3:

```
# Print Data Frames
```

```
print(df_x)
```

3. Right Outer Join

```
##   id      post  
## 1 3  Engineer  
## 2 6  Trainee  
## 3 1  Founder  
## 4 1      CEO  
## 5 4 Office Boy
```

```
print(df_y)
```

```
##   eid  name  
## 1   1  Alex  
## 2   2   Ben  
## 3   3 Cavin  
## 4   4   Dan  
## 5   5 Ellis
```

```
# Right Outer Join Operation
```

```
df_right_outer <- merge(df_x,df_y,by.x="id",by.y="eid",all.y = TRUE)  
print(df_right_outer)
```



```
##   id      post  name
## 1  1    Founder Alex
## 2  1        CEO Alex
## 3  2      <NA>  Ben
## 4  3  Engineer  Cavin
## 5  4 Office Boy  Dan
## 6  5      <NA> Ellis
```

OUTER JOIN
(with UNION)

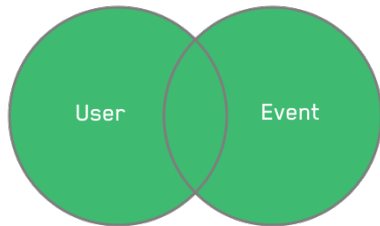


Figure 4:

```
# Print Data Frames
print(df_x)
```

4. Full Outer Join

```
##   id      post
## 1  3  Engineer
## 2  6   Trainee
## 3  1   Founder
## 4  1        CEO
## 5  4 Office Boy
```

```
print(df_y)
```

```
##   eid  name
## 1   1  Alex
## 2   2   Ben
## 3   3 Cavin
## 4   4   Dan
## 5   5  Ellis
```

```
# Full Outer Join Operation
```

```
df_outer <- merge(df_x,df_y,by.x="id",by.y="eid",all = TRUE)
print(df_outer)
```

```
##   id      post  name
## 1  1    Founder Alex
```

```
## 2 1      CEO  Alex
## 3 2      <NA> Ben
## 4 3  Engineer  Cavin
## 5 4 Office Boy  Dan
## 6 5      <NA> Ellis
## 7 6  Trainee  <NA>
```

```
# Print Data Frames
print(df_x)
```

5. Cross/Cartesian Join

```
##   id      post
## 1 3  Engineer
## 2 6  Trainee
## 3 1  Founder
## 4 1      CEO
## 5 4 Office Boy
```

```
print(df_y)
```

```
##   eid  name
## 1   1  Alex
## 2   2   Ben
## 3   3 Cavin
## 4   4   Dan
## 5   5 Ellis
```

```
# Cross Join Operation
df_cross <- merge(df_x,df_y,by = NULL)
print(df_cross)
```

```
##   id      post eid  name
## 1  3  Engineer   1  Alex
## 2  6  Trainee   1  Alex
## 3  1  Founder   1  Alex
## 4  1      CEO   1  Alex
## 5  4 Office Boy   1  Alex
## 6  3  Engineer   2   Ben
## 7  6  Trainee   2   Ben
## 8  1  Founder   2   Ben
## 9  1      CEO   2   Ben
## 10 4 Office Boy   2   Ben
## 11 3  Engineer   3 Cavin
## 12 6  Trainee   3 Cavin
## 13 1  Founder   3 Cavin
## 14 1      CEO   3 Cavin
## 15 4 Office Boy   3 Cavin
## 16 3  Engineer   4   Dan
```

Split

##	gender	height
## 1	MALE	170
## 2	FEMALE	165
## 3	FEMALE	167
## 4	FEMALE	166
## 5	MALE	180
## 6	MALE	175
## 7	FEMALE	165
## 8	MALE	180
## 9	FEMALE	167
## 10	FEMALE	165
## 11	MALE	170

```
## $FEMALE
##      gender height
## 2  FEMALE    165
## 3  FEMALE    167
## 4  FEMALE    166
## 7  FEMALE    165
## 9  FEMALE    167
## 10 FEMALE    165
##
## $MALE
```

```
##      gender height
## 1      MALE      170
## 5      MALE      180
## 6      MALE      175
## 8      MALE      180
## 11     MALE      170
```

```
# 2. Split data frame by height
st <- split(df, df$height)

# Print split data
print(st)
```

```
## $'165'
##      gender height
## 2  FEMALE      165
## 7  FEMALE      165
## 10 FEMALE      165
##
## $'166'
##      gender height
## 4  FEMALE      166
##
## $'167'
##      gender height
## 3  FEMALE      167
## 9  FEMALE      167
##
## $'170'
##      gender height
## 1      MALE      170
## 11     MALE      170
##
## $'175'
##      gender height
## 6      MALE      175
##
## $'180'
##      gender height
## 5      MALE      180
## 8      MALE      180
```

```
# 3. Split data frame by gender and height
st <- split(df$gender, df$height)

# Print split data
print(st)
```

```
## $'165'
## [1] "FEMALE" "FEMALE" "FEMALE"
##
## $'166'
## [1] "FEMALE"
```

```
##
## $'167'
## [1] "FEMALE" "FEMALE"
##
## $'170'
## [1] "MALE" "MALE"
##
## $'175'
## [1] "MALE"
##
## $'180'
## [1] "MALE" "MALE"
```

```
# 4. Split data frame by height and gender
st <- split(df$height,df$gender)

# Print split data
print(st)
```

```
## $FEMALE
## [1] 165 167 166 165 167 165
##
## $MALE
## [1] 170 180 175 180 170
```

Split (Practice)

```
data("InsectSprays")
head(InsectSprays)
```

```
##      count spray
## 1      10      A
## 2       7      A
## 3      20      A
## 4      14      A
## 5      14      A
## 6      12      A
```

```
tail(InsectSprays)
```

```
##      count spray
## 67      13      F
## 68      10      F
## 69      26      F
## 70      26      F
## 71      24      F
## 72      13      F
```

```
split(InsectSprays$count, InsectSprays$spray)
```

```
## $A
## [1] 10  7 20 14 14 12 10 23 17 20 14 13
##
## $B
## [1] 11 17 21 11 16 14 17 17 19 21  7 13
##
## $C
## [1] 0 1 7 2 3 1 2 1 3 0 1 4
##
## $D
## [1]  3  5 12  6  4  3  5  5  5  5  2  4
##
## $E
## [1] 3 5 3 5 3 6 1 1 3 2 6 4
##
## $F
## [1] 11  9 15 22 15 16 13 10 26 26 24 13
```

Transpose Data Frame

```
# Create Data Frame

df <- data.frame(
  Roll=1:5,
  Name=c('Ankit', 'Sara', 'Pravin', 'Kiran', 'Vishal'),
  Gender=c("Male", "Female", "Male", "Female", "Male"),
  Pointer=c(8.23, 6.0, 4.2, 10, 9.22),
  Rank=c(3, 4, 5, 1, 2),
  stringsAsFactors = FALSE
)

# Print Data Frame
print(df)
```

Create Data Frame

```
##   Roll  Name Gender Pointer Rank
## 1    1  Ankit   Male    8.23   3
## 2    2   Sara Female    6.00   4
## 3    3 Pravin   Male    4.20   5
## 4    4  Kiran Female   10.00   1
## 5    5 Vishal   Male    9.22   2
```

```
# Adding Row Name to Data Frame
rownames(df) <- paste0("Student-", 1:5)

# Print Data Frame
print(df)
```

```
##           Roll  Name Gender Pointer Rank
## Student-1    1  Ankit   Male    8.23    3
## Student-2    2   Sara Female    6.00    4
## Student-3    3 Pravin   Male    4.20    5
## Student-4    4  Kiran Female   10.00    1
## Student-5    5 Vishal   Male    9.22    2
```

```
# Transpose Dataframe
df_transpose<-as.data.frame(t(df))

# Print Data Frame
print(df_transpose)
```

Transpose

```
##           Student-1 Student-2 Student-3 Student-4 Student-5
## Roll              1         2         3         4         5
## Name             Ankit        Sara   Pravin    Kiran    Vishal
## Gender            Male      Female    Male     Female    Male
## Pointer           8.23       6.00    4.20    10.00    9.22
## Rank              3         4         5         1         2
```

```
# Back to Original -> Transpose Dataframe

df_Original<-as.data.frame(t(df_transpose))

# Print Data Frame
print(df_Original)
```

```
##           Roll  Name Gender Pointer Rank
## Student-1    1  Ankit   Male    8.23    3
## Student-2    2   Sara Female    6.00    4
## Student-3    3 Pravin   Male    4.20    5
## Student-4    4  Kiran Female   10.00    1
## Student-5    5 Vishal   Male    9.22    2
```