

1. How to use the `cbind()` and `rbind()` in data frame for the fields city and zipcodedatas using vector and data frame.

**Create a vectors:****cbind() function:****Output:**

	city	zipcode
[1]	delhi	123456
[2]	bangalore	789654
[3]	chennai	698748
[4]	mumbai	456986

**PROGRAM:**

```
city<-c("delhi","bangalore","chennai","mumbai")
zipcode<-c(123456,789654,698748,456986)
cbind(city,zipcode)
```

**Output:**

```
city  zipcode
[1,] "delhi"  "123456"
[2,] "bangalore" "789654"
[3,] "chennai" "698748"
[4,] "mumbai"  "456986"
```

**rbind() function:****Output:**

	city	zipcode
[1]	delhi	123456
[2]	bangalore	789654
[3]	chennai	698748
[4]	mumbai	456986
[5]	punjab	456978
[6]	kerala	569875

**PROGRAM:**

```
city<-c("delhi","bangalore","chennai","mumbai")
zipcode<-c(123456,789654,698748,456986)
```

```

cbind(city,zipcode)
r1<-c("delhi",123456)
r2<-c("bangalore",789654)
r3<-c("chennai",698748)
r4<-c("mumbai",456986)
r5<-c("punjab",456978)
r6<-c("kerala",569875)
rbind(r1,r2,r3,r4,r5,r6)

```

**Output:**

```

[,1]    [,2]
r1 "delhi"  "123456"
r2 "bangalore" "789654"
r3 "chennai" "698748"
r4 "mumbai"  "456986"
r5 "punjab"  "456978"
r6 "kerala"  "569875"

```

**2. Create First Dataset with variables**

- **surname**
- **nationality**

**Create Second Dataset with variables**

- **surname**
- **movies**

**The common key variable is surname. How to merge both data and check if the dimensionality is 7x3.**

**Output:**

<b>surname</b>	<b>nationality</b>	<b>movies</b>
1 Hitchcock	UK	Psycho
2 Hitchcock	UK	North by Northwest
3 Polanski	Poland	Chinatown
4 Scorsese	US	Taxi Driver
5 Spielberg	US	Super 8
6 Spielberg	US	Catch Me If You Can
7 Tarantino	US	Reservoir Dogs

**PROGRAM:**

**3. Write a R program to create an empty data frame.**

**Output:**

```

[1] "Structure of the empty dataframe:"
'data.frame': 0 obs. of 5 variables:
 $ Ints : int
 $ Doubles : num
 $ Characters: chr
 $ Logicals :logi
 $ Factors : Factor w/ 0 levels:
 NULL

```

**PROGRAM:**

```
df = data.frame(Ints=integer(),
               Doubles=double(),
               Characters=character(),
               Logicals=logical(),
               Factors=factor(),
               stringsAsFactors=FALSE)
print("Structure of the empty dataframe:")
print(str(df))
```

### Output:

```
'data.frame': 0 obs. of 5 variables:
 $ Ints : int
 $ Doubles : num
 $ Characters: chr
 $ Logicals : logi
 $ Factors : Factor w/ 0 levels:
 NULL
```

#### 4. Write a R program to create a data frame from four given vectors

```
name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael',
'Matthew', 'Laura', 'Kevin', 'Jonas')
score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19)
attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1)
qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
```

### Output:

```
[1] "Original data frame:"
[1] "Anastasia" "Dima" "Katherine" "James" "Emily" "Michael"
[7] "Matthew" "Laura" "Kevin" "Jonas"
[1] 12.5 9.0 16.5 12.0 9.0 20.0 14.5 13.5 8.0 19.0
[1] 1 3 2 3 2 3 1 1 2 1
[1] "yes" "no" "yes" "no" "no" "yes" "yes" "no" "no" "yes"
```

	name	score	attempts	qualify
1	Anastasia	12.5	1	yes
2	Dima	9.0	3	no
3	Katherine	16.5	2	yes
4	James	12.0	3	no
5	Emily	9.0	2	no
6	Michael	20.0	3	yes
7	Matthew	14.5	1	yes
8	Laura	13.5	1	no
9	Kevin	8.0	2	no
10	Jonas	19.0	1	yes

### PROGRAM:

```

name<-
c("Anastasia","Dima","Katherine","James","Emily","Michael","Matthew","Laura","Kevin","Jonas")
score<-c(12.5,9,16.5,12,9,20,14.5,13.5,8,19)
Attempts<-c(1,3,2,3,2,3,1,1,2,1)
Qualify<-c("yes","no","yes","no","no","yes","yes","no","no","yes")
df<-data.frame(name,score,Attempts,Qualify)
df

```

## Output:

```

name score Attempts Qualify
1 Anastasia 12.5    1   yes
2   Dima    9.0    3   no
3 Katherine 16.5    2   yes
4   James  12.0    3   no
5   Emily   9.0    2   no
6 Michael  20.0    3   yes
7 Matthew  14.5    1   yes
8   Laura  13.5    1   no
9   Kevin   8.0    2   no
10  Jonas  19.0    1   yes

```

## 5. Write a R program to extract specific column from a data frame using column name.

### Output:

```

[1] "Original dataframe:"
      name score attempts qualify
1 Anastasia 12.5     1   yes
2   Dima    9.0     3   no
3 Katherine 16.5     2   yes
4   James  12.0     3   no
5   Emily   9.0     2   no
6 Michael  20.0     3   yes
7 Matthew  14.5     1   yes
8   Laura  13.5     1   no
9   Kevin   8.0     2   no
10  Jonas  19.0     1   yes
[1] "Extract Specific columns:"
      exam_data.name exam_data.score
1   Anastasia      12.5
2     Dima        9.0
3 Katherine      16.5
4     James      12.0
5     Emily       9.0
6   Michael      20.0

```

7	Matthew	14.5
8	Laura	13.5
9	Kevin	8.0
10	Jonas	19.0

### PROGRAM:

```
name<-
c("Anastasia","Dima","Katherine","James","Emily","Michael","Matthew","Laura","Kevin","Jonas")
score<-c(12.5,9,16.5,12,9,20,14.5,13.5,8,19)
Attempts<-c(1,3,2,3,2,3,1,1,2,1)
Qualify<-c("yes","no","yes","no","no","yes","yes","no","no","yes")
df<-data.frame(name,score,Attempts,Qualify)
result<-(data.frame(df$name,df$score))
result
```

### Output:

```
df.name df.score
1 Anastasia 12.5
2 Dima 9.0
3 Katherine 16.5
4 James 12.0
5 Emily 9.0
6 Michael 20.0
7 Matthew 14.5
8 Laura 13.5
9 Kevin 8.0
10 Jonas 19.0
```

## 6. Write a R program to extract first two rows from a given data frame.

### Output:

```
[1] "Original dataframe:"
  name score attempts qualify
1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no
3 Katherine 16.5 2 yes
4 James 12.0 3 no
5 Emily 9.0 2 no
6 Michael 20.0 3 yes
7 Matthew 14.5 1 yes
8 Laura 13.5 1 no
9 Kevin 8.0 2 no
```

```

10 Jonas 19.0 1 yes
[1] "Extract first two rows:"
      name score attempts qualify
1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no

```

### PROGRAM:

```

name<-
c("Anastasia","Dima","Katherine","James","Emily","Michael","Matthew","Laura","Kevin","Jonas")
score<-c(12.5,9,16.5,12,9,20,14.5,13.5,8,19)
Attempts<-c(1,3,2,3,2,3,1,1,2,1)
Qualify<-c("yes","no","yes","no","no","yes","yes","no","no","yes")
df<-data.frame(name,score,Attempts,Qualify)
result =df[c(1,2),c(1,2,3,4)]
print(result)

```

### Output:

```

name score Attempts Qualify
1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no

```

**7. Write a R program to extract 3<sup>rd</sup> and 5<sup>th</sup> rows with 1<sup>st</sup> and 3<sup>rd</sup> columns from a given data frame.**

### Output:

```

[1] "Original dataframe:"
      name score attempts qualify
1 Anastasia 12.5 1 yes
2 Dima 9.0 3 no
3 Katherine 16.5 2 yes
4 James 12.0 3 no
5 Emily 9.0 2 no
6 Michael 20.0 3 yes
7 Matthew 14.5 1 yes
8 Laura 13.5 1 no
9 Kevin 8.0 2 no
10 Jonas 19.0 1 yes
[1] "Extract 3rd and 5th rows with 1st and 3rd columns : "

```

```

      name attempts
3 Katherine      2
5   Emily      2

```

### **PROGRAM:**

```

name<-
c("Anastasia","Dima","Katherine","James","Emily","Michael","Matthew","Laura","Kevin","Jonas")
score<-c(12.5,9,16.5,12,9,20,14.5,13.5,8,19)
Attempts<-c(1,3,2,3,2,3,1,1,2,1)
Qualify<-c("yes","no","yes","no","no","yes","yes","no","no","yes")
df<-data.frame(name,score,Attempts,Qualify)
result =df[c(3,5),c(1,3)]
print(result)

```

### **Output:**

```

name Attempts
3 Katherine      2
5   Emily      2

```

## **6. Write a R program to add a new column in a given data frame**

### **Output:**

```

[1] "Original dataframe:"
      name score attempts qualify
1 Anastasia 12.5      1    yes
2   Dima    9.0      3    no
3 Katherine 16.5      2    yes
4   James 12.0      3    no
5   Emily   9.0      2    no
6 Michael 20.0      3    yes
7 Matthew 14.5      1    yes
8   Laura 13.5      1    no
9   Kevin   8.0      2    no
10  Jonas 19.0      1    yes
[1] "New data frame after adding the 'country' column:"
      name score attempts qualify country
1 Anastasia 12.5      1    yes   USA
2   Dima    9.0      3    no   USA
3 Katherine 16.5      2    yes   USA
4   James 12.0      3    no   USA
5   Emily   9.0      2    no   USA
6 Michael 20.0      3    yes   USA
7 Matthew 14.5      1    yes   USA
8   Laura 13.5      1    no   USA
9   Kevin   8.0      2    no   USA
10  Jonas 19.0      1    yes   USA

```

### **PROGRAM:**

```

name<-
c('Anastasia','Dima','Katherine','James','Emily','Michael','Matthew','Laura','K
evin','Jonas')
score <-c(12.5,9,16.5,12,9,20,14.5,13.5,8,19)
attempts<-c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1)
qualify <-c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
exam<-data.frame(name,score,attempts,qualify)
print("New data frame after adding the 'country' column:")
exam$country =
c("USA","USA","USA","USA","USA","USA","USA","USA","USA","USA")
exam

```

**Output:**

```

name score attempts qualify country
1 Anastasia 12.5    1   yes   USA
2   Dima    9.0    3   no    USA
3 Katherine 16.5    2   yes   USA
4   James  12.0    3   no    USA
5   Emily   9.0    2   no    USA
6 Michael  20.0    3   yes   USA
7 Matthew  14.5    1   yes   USA
8   Laura  13.5    1   no    USA
9   Kevin   8.0    2   no    USA
10  Jonas  19.0    1   yes   USA

```

**8. Write a R program to add new row(s) to an existing data frame.**

**Output:**

```

[1] "Original dataframe:"
      name score attempts qualify
1 Anastasia 12.5     1   yes
2   Dima    9.0     3   no
3 Katherine 16.5     2   yes
4   James  12.0     3   no
5   Emily   9.0     2   no
6 Michael  20.0     3   yes
7 Matthew  14.5     1   yes
8   Laura  13.5     1   no
9   Kevin   8.0     2   no
10  Jonas  19.0     1   yes
[1] "After adding new row(s) to an existing data frame:"
      name score attempts qualify
1 Anastasia 12.5     1   yes
2   Dima    9.0     3   no
3 Katherine 16.5     2   yes
4   James  12.0     3   no
5   Emily   9.0     2   no
6 Michael  20.0     3   yes
7 Matthew  14.5     1   yes

```



8	Laura	13.5	1	no
9	Kevin	8.0	2	no
10	Jonas	19.0	1	yes
11	Robert	10.5	1	yes
12	Sophia	9.0	3	no

### PROGRAM:

```
name<-
c("Anastasia","Dima","Katherine","James","Emily","Michael","Matthew","Laura","Kevin","Jonas")

score<-c(12.5,9,16.5,12,9,20,14.5,13.5,8,19)

Attempts<-c(1,3,2,3,2,3,1,1,2,1)

Qualify<-c("yes","no","yes","no","no","yes","yes","no","no","yes")

df<-data.frame(name,score,Attempts,Qualify)

df

name<-c("Robert","Sophia")

score<-c(10.5,9)

Attempts<-c(1,3)

Qualify<-c("yes","no")

new<-data.frame(name,score,Attempts,Qualify)

a<-rbind(df,new)

print("After adding rows to an existing dataframe")

print(a)
```

### Output:

```
"After adding rows to an existing dataframe"
```

```
> print(a)
```

	name	score	Attempts	Qualify
1	Anastasia	12.5	1	yes
2	Dima	9.0	3	no
3	Katherine	16.5	2	yes

4	James	12.0	3	no
5	Emily	9.0	2	no
6	Michael	20.0	3	yes
7	Matthew	14.5	1	yes
8	Laura	13.5	1	no
9	Kevin	8.0	2	no
10	Jonas	19.0	1	yes
11	Robert	10.5	1	yes
12	Sophia	9.0	3	no

**10. Write a R program to drop column(s) by name from a given data frame.**

**Output:**

```
[1] "Original dataframe:"
  name score attempts qualify
1 Anastasia 12.5     1    yes
2   Dima    9.0     3    no
3 Katherine 16.5     2    yes
4   James  12.0     3    no
5   Emily   9.0     2    no
6 Michael  20.0     3    yes
7 Matthew 14.5     1    yes
8   Laura  13.5     1    no
9   Kevin   8.0     2    no
10 Jonas  19.0     1    yes
  score attempts
1  12.5       1
2   9.0       3
3  16.5       2
4  12.0       3
5   9.0       2
6  20.0       3
7  14.5       1
8  13.5       1
9   8.0       2
10 19.0       1
```

### **PROGRAM:**

```
name<-  
c('Anastasia','Dima','Katherine','James','Emily','Michael','Matthew','Laura','K  
evin','Jonas')  
score <-c(12.5,9,16.5,12,9,20,14.5,13.5,8,19)  
attempts<-c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1)  
qualify <-c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')  
exam<-data.frame(name,score,attempts,qualify)  
exam<-subset(exam,select = -c(name, qualify))  
exam
```

### **Output:**

```
score attempts  
1 12.5      1  
2  9.0      3  
3 16.5      2  
4 12.0      3  
5  9.0      2  
6 20.0      3  
7 14.5      1  
8 13.5      1  
9  8.0      2  
10 19.0      1
```

**11. Write a R program to drop row(s) by number from a given data frame.**

### **Output:**

```
[1] "Original dataframe:"  
  name score attempts qualify  
1 Anastasia 12.5      1   yes  
2   Dima   9.0      3   no  
3 Katherine 16.5      2   yes  
4   James 12.0      3   no  
5   Emily   9.0      2   no  
6 Michael 20.0      3   yes  
7 Matthew 14.5      1   yes  
8   Laura 13.5      1   no  
9   Kevin   8.0      2   no  
10  Jonas 19.0      1   yes  
  name score attempts qualify  
1 Anastasia 12.5      1   yes  
3 Katherine 16.5      2   yes  
5   Emily   9.0      2   no  
7 Matthew 14.5      1   yes  
8   Laura 13.5      1   no  
9   Kevin   8.0      2   no  
10  Jonas 19.0      1   yes
```

**PROGRAM:**

```
name<-  
c('Anastasia','Dima','Katherine','James','Emily','Michael','Matthew','Laura','Kevin','Jonas')  
score <-c(12.5,9,16.5,12,9,20,14.5,13.5,8,19)  
attempts<-c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1)  
qualify <-c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')  
exam<-data.frame(name,score,attempts,qualify)  
exam<- exam[-c(2, 4, 6),]  
exam
```

**Output:**

```
name score attempts qualify  
1 Anastasia 12.5    1    yes  
3 Katherine 16.5    2    yes  
5   Emily   9.0     2    no  
7   Matthew 14.5    1    yes  
8    Laura  13.5    1    no  
9    Kevin   8.0     2    no  
10   Jonas  19.0    1    yes
```

**12. Write a R program to sort a given data frame by multiple column(s).****Output:**

```
[1] "Original dataframe:"  
   name score attempts qualify  
1 Anastasia 12.5    1    yes  
2    Dima   9.0     3    no  
3 Katherine 16.5    2    yes  
4    James 12.0     3    no  
5    Emily   9.0     2    no  
6 Michael 20.0     3    yes  
7 Matthew 14.5     1    yes  
8    Laura 13.5     1    no  
9    Kevin   8.0     2    no  
10   Jonas 19.0     1    yes  
[1] "dataframe after sorting 'name' and 'score' columns:"  
   name score attempts qualify  
1 Anastasia 12.5    1    yes  
2    Dima   9.0     3    no  
5    Emily   9.0     2    no  
4    James 12.0     3    no  
10   Jonas 19.0     1    yes  
3 Katherine 16.5    2    yes  
9    Kevin   8.0     2    no  
8    Laura 13.5     1    no  
7 Matthew 14.5     1    yes  
6 Michael 20.0     3    yes
```

**PROGRAM:**

```

name<-
c('Anastasia','Dima','Katherine','James','Emily','Michael','Matthew','Laura','K
evin','Jonas')
score <-c(12.5,9,16.5,12,9,20,14.5,13.5,8,19)
attempts<-c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1)
qualify <-c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
exam<-data.frame(name,score,attempts,qualify)
print("dataframe after sorting 'name' and 'score' columns:")
exam<-exam[with(exam,order(name, score)), ]
print(exam)

```

### Output:

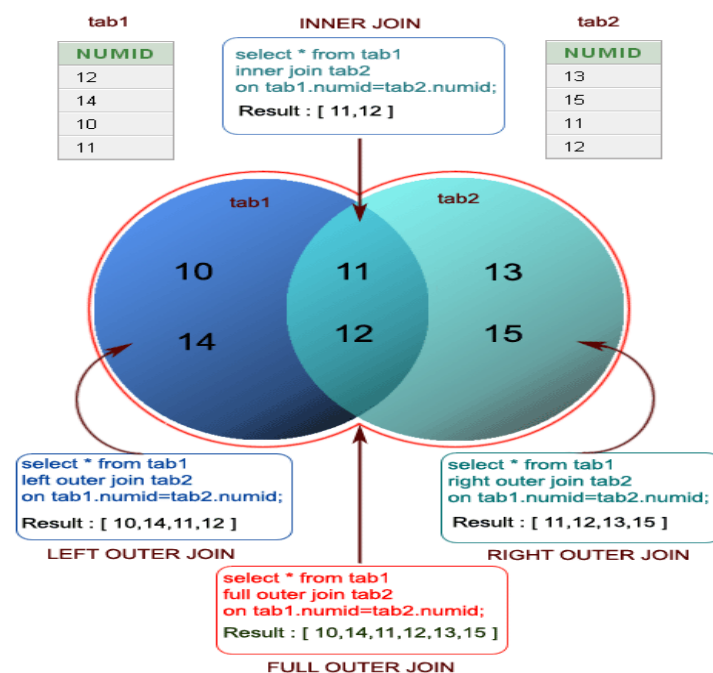
```

name score attempts qualify
1 Anastasia 12.5    1    yes
2   Dima    9.0     3     no
5   Emily    9.0     2     no
4   James  12.0     3     no
10  Jonas  19.0     1     yes
3 Katherine 16.5     2     yes
9   Kevin   8.0     2     no
8   Laura  13.5     1     no
7 Matthew  14.5     1     yes
6 Michael  20.0     3     yes

```

### 13. Write a R program to create inner, outer, left, right join(merge) from given two data frames.

### Output:



[1] "Left outer Join:"

```

numid
1  10

```

```

2  11
3  12
4  14
[1] "Right outer Join:"
numid
1  11
2  12
3  13
4  15
[1] "Outer Join:"
numid
1  10
2  11
3  12
4  13
5  14
6  15
[1] "Cross Join:"
numid.xnumid.y
1    12    13
2    14    13
3    10    13
4    11    13
5    12    15
6    14    15
7    10    15
8    11    15
9    12    11
10   14    11
11   10    11
12   11    11
13   12    12
14   14    12
15   10    12
16   11    12

```

### PROGRAM:

```

df1<-data.frame(numid = c(12, 14, 10, 11))
df2<-data.frame(numid = c(13, 15, 11, 12))
print("Left outer Join:")
result<-merge(df1, df2, by = "numid", all.x = TRUE)
print(result)
print("Right outer Join:")
result<-merge(df1, df2, by = "numid", all.y = TRUE)
print(result)
print("Outer Join:")

```

```
result<-merge(df1, df2, by = "numid", all = TRUE)
print(result)
print("Cross Join:")
result<-merge(df1, df2, by = NULL)
print(result)
```

## Output:

```
print(result)
  numid
1   10
2   11
3   12
4   14
> print("Right outer Join:")
[1] "Right outer Join:"
> result<-merge(df1, df2, by = "numid", all.y = TRUE)
> print(result)
  numid
1   11
2   12
3   13
4   15
> print("Outer Join:")
[1] "Outer Join:"
> result<-merge(df1, df2, by = "numid", all = TRUE)
> print(result)
  numid
1   10
2   11
3   12
4   13
5   14
6   15
> print("Cross Join:")
[1] "Cross Join:"
> result<-merge(df1, df2, by = NULL)
> print(result)
  numid.x numid.y
1     12     13
2     14     13
3     10     13
```

4	11	13
5	12	15
6	14	15
7	10	15
8	11	15
9	12	11
10	14	11
11	10	11
12	11	11
13	12	12
14	14	12
15	10	12
16	11	12

**14. Write a R program to replace NA values with 3 in a given data frame.**

**PROGRAM:**

```
# create an example data frame
df <- data.frame(x = c(1, 2, NA, 4, 5), y = c("a", NA, "c", "d", NA))

# replace all NA values with 3
df[is.na(df)] <- 3

# print the updated data frame
Df
```

**15. Write a R program to change a column name of a given data frame.**

**PROGRAM:**

```
# create an example data frame
df <- data.frame(x = c(1, 2, 3), y = c("a", "b", "c"))

# print the original data frame
df
```



```
# change the column name
colnames(df)[1] <- "new_name"
```

```
# print the updated data frame
Df
```

## OUTPUT:

```
[1] "Original dataframe:"
name score attempts qualify
1 Anastasia 12.5 1 yes
2 Dima 9.0 NA no
3 Katherine 16.5 2 yes

4 James 12.0 NA no
5 Emily 9.0 2 no
6 Michael 20.0 NA yes
7 Matthew 14.5 1 yes
8 Laura 13.5 NA no
9 Kevin 8.0 2 no
10 Jonas 19.0 1 yes
[1] "Change column-name 'name' to 'student_name' of
the said dataframe:"
student_name score attempts qualify
1 Anastasia 12.5 1 yes
2 Dima 9.0 NA no
3 Katherine 16.5 2 yes
4 James 12.0 NA no
5 Emily 9.0 2 no
6 Michael 20.0 NA yes
7 Matthew 14.5 1 yes
8 Laura 13.5 NA no
9 Kevin 8.0 2 no
10 Jonas 19.0 1 yes
```

**16. Write a R program to change more than one column name of a given data frame.**

## PROGRAM:

```
# create an example data frame
df <- data.frame(name = c("Anastasia", "Dima", "Katherine", "James",
"Emily", "Michael", "Matthew", "Laura", "Kevin", "Jonas"),
score = c(12.5, 9.0, 16.5, 12.0, 9.0, 20.0, 14.5, 13.5, 8.0, 19.0),
attempts = c(1, NA, 2, NA, 2, NA, 1, NA, 2, 1),
qualify = c("yes", "no", "yes", "no", "no", "yes", "yes", "no", "no",
"yes"))
```

```
# print the original data frame
```

```
cat("Original dataframe:\n")
```

```
print(df)
```

```
# change the column names
```

```
names(df)[1:2] <- c("student_name", "avg_score")
```

```
# print the updated data frame
```

```
cat("\nChange more than one column name of the said dataframe:\n")
```

```
print(df)
```

## OUTPUT:

```
> # create an example data frame
> df <- data.frame(name = c("Anastasia", "Dima", "Katherine", "James", "Emily", "Michael", "Matthew", "Laura", "Kevin", "Jonas"),
+                   score = c(12.5, 9.0, 16.5, 12.0, 9.0, 20.0, 14.5, 13.5, 8.0, 19.0),
+                   attempts = c(1, NA, 2, NA, 2, NA, 1, NA, 2, 1),
+                   qualify = c("yes", "no", "yes", "no", "no", "yes", "yes", "no", "no", "yes"))
>
```

```
> # print the original data frame
```

```
> cat("Original dataframe:\n")
```

```
Original dataframe:
```

```
> print(df)
```

	name	score	attempts	qualify
1	Anastasia	12.5	1	yes
2	Dima	9.0	NA	no
3	Katherine	16.5	2	yes
4	James	12.0	NA	no
5	Emily	9.0	2	no
6	Michael	20.0	NA	yes
7	Matthew	14.5	1	yes
8	Laura	13.5	NA	no
9	Kevin	8.0	2	no
10	Jonas	19.0	1	yes

```
>
```

```
> # change the column names
```

```
> names(df)[1:2] <- c("student_name", "avg_score")
```

```
>
```

```
> # print the updated data frame
```

```
> cat("\nChange more than one column name of the said dataframe:\n")
```

```
Change more than one column name of the said dataframe:
```

```
> print(df)
```

	student_name	avg_score	attempts	qualify
1	Anastasia	12.5	1	yes
2	Dima	9.0	NA	no
3	Katherine	16.5	2	yes
4	James	12.0	NA	no
5	Emily	9.0	2	no
6	Michael	20.0	NA	yes
7	Matthew	14.5	1	yes
8	Laura	13.5	NA	no
9	Kevin	8.0	2	no
10	Jonas	19.0	1	yes

```
>
```

>

**17. Write a R program to select some random rows from a given data frame.**

**PROGRAM:**

```
# create an example data frame
df <- data.frame(name = c("Anastasia", "Dima", "Katherine", "James",
"Emily", "Michael", "Matthew", "Laura", "Kevin", "Jonas"),
                score = c(12.5, 9.0, 16.5, 12.0, 9.0, 20.0, 14.5, 13.5, 8.0, 19.0),
                attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
                qualify = c("yes", "no", "yes", "no", "no", "yes", "yes", "no", "no",
"yes"))

# print the original data frame
cat("Original dataframe:\n")
print(df)

# select random rows
set.seed(123) # set the seed for reproducibility
random_rows <- sample(nrow(df), 3)

# print the selected rows
cat("\nSelect three random rows of the said dataframe:\n")
print(df[random_rows,])
```

**OUTPUT:**

```
> # create an example data frame
> df <- data.frame(name = c("Anastasia", "Dima", "Katherine", "James", "
Emily", "Michael", "Matthew", "Laura", "Kevin", "Jonas"),
+                 score = c(12.5, 9.0, 16.5, 12.0, 9.0, 20.0, 14.5, 13.
5, 8.0, 19.0),
+                 attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
+                 qualify = c("yes", "no", "yes", "no", "no", "yes", "yes", "no", "y
es", "no", "no", "yes"))
>
> # print the original data frame
> cat("Original dataframe:\n")
Original dataframe:
> print(df)
  name score attempts qualify
1 Anastasia 12.5         1    yes
2 Dima      9.0         3     no
3 Katherine 16.5         2    yes
4 James    12.0         3     no
5 Emily     9.0         2     no
6 Michael  20.0         3    yes
```

```

7   Matthew 14.5      1   yes
8   Laura   13.5      1   no
9   Kevin   8.0       2   no
10  Jonas   19.0      1   yes
>
> # select random rows
> set.seed(123) # set the seed for reproducibility
> random_rows <- sample(nrow(df), 3)
>
> # print the selected rows
> cat("\nSelect three random rows of the said dataframe:\n")

Select three random rows of the said dataframe:
> print(df[random_rows,])
      name score attempts qualify
3 Katherine 16.5         2     yes
10 Jonas    19.0         1     yes
2  Dima     9.0         3     no
> # create an example data frame
> df <- data.frame(name = c("Anastasia", "Dima", "Katherine", "James", "
Emily", "Michael", "Matthew", "Laura", "Kevin", "Jonas"),
+                   score = c(12.5, 9.0, 16.5, 12.0, 9.0, 20.0, 14.5, 13.
5, 8.0, 19.0),
+                   attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
+                   qualify = c("yes", "no", "yes", "no", "no", "yes", "y
es", "no", "no", "yes"))
>
> # print the original data frame
> cat("Original dataframe:\n")
Original dataframe:
> print(df)
      name score attempts qualify
1 Anastasia 12.5         1     yes
2 Dima      9.0         3     no
3 Katherine 16.5         2     yes
4 James     12.0         3     no
5 Emily      9.0         2     no
6 Michael   20.0         3     yes
7 Matthew   14.5         1     yes
8 Laura     13.5         1     no
9 Kevin      8.0         2     no
10 Jonas    19.0         1     yes
>
> # select random rows
> set.seed(123) # set the seed for reproducibility
> random_rows <- sample(nrow(df), 3)
>
> # print the selected rows
> cat("\nSelect three random rows of the said dataframe:\n")

Select three random rows of the said dataframe:
> print(df[random_rows,])
      name score attempts qualify
3 Katherine 16.5         2     yes
10 Jonas    19.0         1     yes
2  Dima     9.0         3     no

```

**18. Write a R program to reorder an given data frame by column name.**

**PROGRAM:**

\

```
# Create a sample data frame
df <- data.frame(name = c("Anastasia", "Dima", "Katherine", "James", "Emily",
"Michael", "Matthew", "Laura", "Kevin", "Jonas"),
  score = c(12.5, 9.0, 16.5, 12.0, 9.0, 20.0, 14.5, 13.5, 8.0, 19.0),
  attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c("yes", "no", "yes", "no", "no", "yes", "yes", "no", "no", "yes"))
```

```
# Print the original data frame
cat("Original dataframe:\n")
print(df)
```

```
# Reorder the data frame by column name
cat("Reorder by column name:\n")
df <- df[, c("name", "attempts", "score", "qualify")]
print(df)
```

## OUTPUT:

```
> # Create a sample data frame
> df <- data.frame(name = c("Anastasia", "Dima", "Katherine", "James", "
Emily", "Michael", "Matthew", "Laura", "Kevin", "Jonas"),
+               score = c(12.5, 9.0, 16.5, 12.0, 9.0, 20.0, 14.5, 13.
5, 8.0, 19.0),
+               attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
+               qualify = c("yes", "no", "yes", "no", "no", "yes", "yes", "no", "no", "yes"))
>
> # Print the original data frame
> cat("Original dataframe:\n")
Original dataframe:
> print(df)
  name score attempts qualify
1 Anastasia 12.5      1    yes
2 Dima      9.0      3     no
3 Katherine 16.5      2    yes
4 James    12.0      3     no
5 Emily     9.0      2     no
6 Michael  20.0      3    yes
7 Matthew  14.5      1    yes
8 Laura    13.5      1     no
9 Kevin     8.0      2     no
10 Jonas   19.0      1    yes
>
> # Reorder the data frame by column name
> cat("Reorder by column name:\n")
Reorder by column name:
> df <- df[, c("name", "attempts", "score", "qualify")]
> print(df)
  name attempts score qualify
1 Anastasia      1 12.5    yes
2 Dima          3  9.0     no
3 Katherine      2 16.5    yes
4 James          3 12.0     no
5 Emily          2  9.0     no
6 Michael        3 20.0    yes
```

7	Matthew	1	14.5	yes
8	Laura	1	13.5	no
9	Kevin	2	8.0	no
10	Jonas	1	19.0	yes

**19. Write a R program to compare two data frames to find the elements in first data frame that are not present in second data frame.**

**PROGRAM:**

```
# Create two data frames
df1 <- data.frame(a = c(1, 2, 3, 4, 5), b = c("A", "B", "C", "D", "E"))
df2 <- data.frame(a = c(2, 3, 6), b = c("B", "C", "F"))

# Find elements in df1 not present in df2
df1_not_in_df2 <- df1[!(paste(df1$a, df1$b) %in% paste(df2$a, df2$b)), ]

# Print the original data frames and the result
cat("df1:\n")
print(df1)
cat("\ndf2:\n")
print(df2)
cat("\nElements in df1 not present in df2:\n")
print(df1_not_in_df2)
```

**OUTPUT**

```
> # Create two data frames
> df1 <- data.frame(a = c(1, 2, 3, 4, 5), b = c("A", "B", "C", "D", "E"))
> df2 <- data.frame(a = c(2, 3, 6), b = c("B", "C", "F"))
> # Find elements in df1 not present in df2
> df1_not_in_df2 <- df1[!(paste(df1$a, df1$b) %in% paste(df2$a, df2$b)), ]
> # Print the original data frames and the result
> cat("df1:\n")
df1:
> print(df1)
  a b
1 1 A
2 2 B
3 3 C
4 4 D
5 5 E
> cat("\ndf2:\n")
df2:
> print(df2)
  a b
1 2 B
2 3 C
3 6 F
> cat("\nElements in df1 not present in df2:\n")
Elements in df1 not present in df2:
> print(df1_not_in_df2)
  a b
1 1 A
4 4 D
5 5 E
```

20. Write a R program to find elements which are present in two given data frames.

**PROGRAM:**

**# create two data frames**

```
df1 <- data.frame(a=c(1,2,3), b=c(4,5,6), c=c(7,8,9), d=c(10,11,12),  
e=c(13,14,15))
```

```
df2 <- data.frame(d=c(10,11,12), e=c(13,14,15), f=c(16,17,18),  
g=c(19,20,21))
```

**# display original data frames**

```
cat("Original Dataframes\n")
```

```
print(df1)
```

```
print(df2)
```

**# find elements present in both data frames**

```
common_elements <- intersect(names(df1), names(df2))
```

**# display common elements**

```
cat("\nElements which are present in both dataframe:\n")
```

```
print(common_elements)
```

**OUTPUT:**

```
> # create two data frames  
> df1 <- data.frame(a=c(1,2,3), b=c(4,5,6), c=c(7,8,9), d=c(10,11,12), e  
=c(13,14,15))  
> df2 <- data.frame(d=c(10,11,12), e=c(13,14,15), f=c(16,17,18), g=c(19,  
20,21))  
>  
> # display original data frames  
> cat("Original Dataframes\n")  
Original Dataframes  
> print(df1)  
  a b c  d  e  
1 1 4 7 10 13  
2 2 5 8 11 14  
3 3 6 9 12 15  
> print(df2)  
  d e f  g  
1 10 13 16 19  
2 11 14 17 20  
3 12 15 18 21  
>  
> # find elements present in both data frames  
> common_elements <- intersect(names(df1), names(df2))  
>  
> # display common elements  
> cat("\nElements which are present in both dataframe:\n")
```

```
Elements which are present in both dataframe:  
> print(common_elements)  
[1] "d" "e"
```

**21. Write a R program to find elements come only once that are common to both given data frames.**

**PROGRAM:**

```
# create two data frames  
df1 <- data.frame(a = 1:5, b = 6:10, c = 11:15, d = 16:20, e = 21:25)  
df2 <- data.frame(d = 16:18, e = 21:23, f = 26:28, g = 29:31)  
  
# combine data frames  
combined <- c(df1, df2)  
  
# count frequency of each element  
freq <- table(unlist(combined))  
  
# subset elements that appear exactly once  
result <- names(freq[freq == 1])  
  
# print the result  
cat("Original Dataframes\n")  
print(df1)  
print(df2)  
cat("Find elements come only once that are common to both given  
dataframes:\n")  
print(result)
```

**OUTPUT:**

```
> # create two data frames  
> df1 <- data.frame(a = 1:5, b = 6:10, c = 11:15, d = 16:20, e = 21:25)  
> df2 <- data.frame(d = 16:18, e = 21:23, f = 26:28, g = 29:31)  
>  
> # combine data frames  
> combined <- c(df1, df2)  
>  
> # count frequency of each element  
> freq <- table(unlist(combined))  
>
```



```

> # subset elements that appear exactly once
> result <- names(freq[freq == 1])
>
> # print the result
> cat("Original Dataframes\n")
Original Dataframes
> print(df1)
  a  b  c  d  e
1 1  6 11 16 21
2 2  7 12 17 22
3 3  8 13 18 23
4 4  9 14 19 24
5 5 10 15 20 25
> print(df2)
  d  e  f  g
1 16 21 26 29
2 17 22 27 30
3 18 23 28 31
> cat("Find elements come only once that are common to both given dataframes:\n")
Find elements come only once that are common to both given dataframes:
> print(result)
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13"
[14] "14" "15" "19" "20" "24" "25" "26" "27" "28" "29" "30" "31"

```

**22. Write a R program to save the information of a data frame in a file and display the information of the file.**

**PROGRAM:**

**# Create a sample data frame**

```

df <- data.frame(
  name = c("Anastasia", "Dima", "Katherine", "James", "Emily", "Michael",
"Matthew", "Laura", "Kevin", "Jonas"),
  score = c(12.5, 9.0, 16.5, 12.0, 9.0, 20.0, 14.5, 13.5, 8.0, 19.0),
  attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),
  qualify = c("yes", "no", "yes", "no", "no", "yes", "yes", "no", "no", "yes")
)

```

**# Save the data frame to a file**

```

save(df, file = "data.rda")

```

**# Display information about the file**

```

cat("File info:\n")
print(file.info("data.rda"))

```

**OUTPUT:**

**[1] "Original dataframe:"**

**name score attempts qualify**

**1 Anastasia 12.5 1 yes**

**2 Dima 9.0 3 no**

**3 Katherine 16.5 2 yes**

**4 James 12.0 3 no**

**5 Emily 9.0 2 no**

**6 Michael 20.0 3 yes**

**7 Matthew 14.5 1 yes**

**8 Laura 13.5 1 no**

**9 Kevin 8.0 2 no**

**10 Jonas 19.0 1 yes**

**size isdir mode mtimectime**

**data.rda 344 FALSE 644 2018-10-25 12:06:09 2018-10-25 12:06:09**

**atimeuidgidunamegrname**

**data.rda 2018-10-25 12:06:09 1000 1000 trinket trinket**

**23. Write a R program to count the number of NA values in a data frame column.**

**PROGRAM:**

**# create a sample dataframe**

**df <- data.frame(**

**name = c("Anastasia", "Dima", "Katherine", "James", "Emily",  
"Michael", "Matthew", "Laura", "Kevin", "Jonas"),**

**score = c(12.5, 9.0, 16.5, 12.0, 9.0, 20.0, 14.5, 13.5, 8.0, 19.0),**

**attempts = c(1, NA, 2, NA, 2, NA, 1, NA, 2, 1),**

**qualify = c("yes", "no", "yes", "no", "no", "yes", "yes", "no", "no", "yes")**

**)**

**# print the original dataframe**

**cat("Original dataframe:\n")**

**print(df)**

**# count the number of NA values in attempts column**

**cat("The number of NA values in attempts column:\n")**

**sum(is.na(df\$attempts))**

## OUTPUT:

```
> # create a sample dataframe
> df <- data.frame(
+   name = c("Anastasia", "Dima", "Katherine", "James", "Emily",
+           "Michael", "Matthew", "Laura", "Kevin", "Jonas"),
+   score = c(12.5, 9.0, 16.5, 12.0, 9.0, 20.0, 14.5, 13.5, 8.0, 19.0)
+ ,
+   attempts = c(1, NA, 2, NA, 2, NA, 1, NA, 2, 1),
+   qualify = c("yes", "no", "yes", "no", "no", "yes", "yes", "no", "no", "yes")
+ )
>
> # print the original dataframe
> cat("Original dataframe:\n")
Original dataframe:
> print(df)
  name score attempts qualify
1 Anastasia 12.5      1    yes
2 Dima      9.0     NA     no
3 Katherine 16.5      2    yes
4 James    12.0     NA     no
5 Emily     9.0      2     no
6 Michael  20.0     NA    yes
7 Matthew  14.5      1    yes
8 Laura    13.5     NA     no
9 Kevin     8.0      2     no
10 Jonas   19.0      1    yes
>
> # count the number of NA values in attempts column
> cat("The number of NA values in attempts column:\n")
The number of NA values in attempts column:
> sum(is.na(df$attempts))
[1] 4
```

**24. Write a R program to create a data frame using two given vectors and display the duplicated elements and unique rows of the said data frame.**

## PROGRAM:

```
# create two vectors
vec1 <- c(10, 20, 10, 10, 40, 50, 20, 30)
vec2 <- c(10, 30, 10, 20, 0, 50, 30, 30)

# create data frame using the two vectors
df <- data.frame(a = vec1, b = vec2)

# print original data frame
cat("Original data frame:\n")
print(df)

# find duplicated elements in data frame
```

```
cat("Duplicate elements of the said data frame:\n")
duplicated_rows <- duplicated(df) | duplicated(df, fromLast = TRUE)
print(duplicated_rows)
```

```
# find unique rows in data frame
cat("Unique rows of the said data frame:\n")
unique_rows <- df[!duplicated_rows, ]
print(unique_rows)
```

## OUTPUT:

```
> # create two vectors
> vec1 <- c(10, 20, 10, 10, 40, 50, 20, 30)
> vec2 <- c(10, 30, 10, 20, 0, 50, 30, 30)
>
> # create data frame using the two vectors
> df <- data.frame(a = vec1, b = vec2)
>
> # print original data frame
> cat("Original data frame:\n")
Original data frame:
> print(df)
  a  b
1 10 10
2 20 30
3 10 10
4 10 20
5 40  0
6 50 50
7 20 30
8 30 30
>
> # find duplicated elements in data frame
> cat("Duplicate elements of the said data frame:\n")
Duplicate elements of the said data frame:
> duplicated_rows <- duplicated(df) | duplicated(df, fromLast = TRUE)
> print(duplicated_rows)
[1] TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE
>
> # find unique rows in data frame
> cat("Unique rows of the said data frame:\n")
Unique rows of the said data frame:
> unique_rows <- df[!duplicated_rows, ]
> print(unique_rows)
  a  b
4 10 20
5 40  0
6 50 50
8 30 30
```

**25. Write a R program to call the (built-in) dataset airquality. Check whether it is a data frame or not? Order the entire data frame by the first and second column.**

## PROGRAM:

**# Load the airquality dataset**

```
data("airquality")
```

**# Check whether it is a data frame or not**

```
cat("Original data: Daily air quality measurements in New York, May to  
September 1973.\n")
```

```
if (is.data.frame(airquality)) {  
  cat("data.frame\n")  
}
```

**# Order the entire data frame by the first and second column**

```
cat("\nOrder the entire data frame by the first and second column:\n")  
airquality <- airquality[order(airquality$Ozone, airquality$Solar.R), ]  
print(head(airquality))
```

## OUTPUT:

```
> # Load the airquality dataset  
> data("airquality")  
>  
> # Check whether it is a data frame or not  
> cat("Original data: Daily air quality measurements in New York, May to  
September 1973.\n")  
Original data: Daily air quality measurements in New York, May to Septem  
ber 1973.  
> if (is.data.frame(airquality)) {  
+   cat("data.frame\n")  
+ }  
data.frame  
>  
> # Order the entire data frame by the first and second column  
> cat("\norder the entire data frame by the first and second column:\n")  
  
Order the entire data frame by the first and second column:  
> airquality <- airquality[order(airquality$Ozone, airquality$Solar.R),  
]  
> print(head(airquality))  
   Ozone Solar.R Wind Temp Month Day  
21     1      8  9.7  59     5  21  
23     4     25  9.7  61     5  23  
18     6     78 18.4  57     5  18  
76     7     48 14.3  80     7  15  
147    7     49 10.3  69     9  24  
11     7     NA  6.9  74     5  11
```

**26. Write a R program to call the (built-in) dataset airquality.**

**Remove the variables**

**'Solar.R' and 'Wind' and display the data frame.**

### **PROGRAM:**

```
# Load the airquality dataset
data(airquality)

# Display the original dataset
cat("Original data: Daily air quality measurements in New York, May to
September 1973.\n")
head(airquality)

# Remove the 'Solar.R' and 'Wind' variables
airquality_new <- airquality[, c("Ozone", "Temp", "Month", "Day")]

# Display the resulting data frame
cat("\ndata.frame after removing 'Solar.R' and 'Wind' variables:\n")
head(airquality_new)
```

### **OUTPUT:**

```
> # Load the airquality dataset
> data(airquality)
>
> # Display the original dataset
> cat("Original data: Daily air quality measurements in New York, May to
September 1973.\n")
Original data: Daily air quality measurements in New York, May to Septem
1973.
> head(airquality)
  Ozone Solar.R Wind Temp Month Day
1   41     190  7.4   67     5    1
2   36     118  8.0   72     5    2
3   12     149 12.6   74     5    3
4   18     313 11.5   62     5    4
5   NA       NA 14.3   56     5    5
6   28       NA 14.9   66     5    6
>
> # Remove the 'Solar.R' and 'Wind' variables
> airquality_new <- airquality[, c("Ozone", "Temp", "Month", "Day")]
>
> # Display the resulting data frame
> cat("\ndata.frame after removing 'Solar.R' and 'Wind' variables:\n")
```

data.frame after removing 'solar.R' and 'wind' variables:

```
> head(airquality_new)
  Ozone Temp Month Day
1    41   67     5   1
2    36   72     5   2
3    12   74     5   3
4    18   62     5   4
5     NA   56     5   5
6    28   66     5   6
>
>
```

**27. Find the difference between Data Frames and other Data Structures with example.**

**Solution:**

**Data Structure:**

There is also an array data structure that extends this idea to more than two

dimensions. A collection of vectors that all have the same length.

This is like a matrix,

except that each column can contain a different data type.

Eg:Array, Linked Lists, Stack, Queues, Trees, Graphs, Sets, Hash Tables.

**Data Frame:**

A data frame can be used to represent an entire data set. A data frame is a table or a

two-dimensional array-like structure in which each column contains values of one

variable and each row contains one set of values from each column.

Eg: Matrices

**28. How to create the data frame and print it for the employee data set.**

**PROGRAM:**

```

# Create the employee data frame
employee_df <- data.frame(
  Emp_id = 1:5,
  Emp_name = c("Ricky", "Danish", "Mini", "Ryan", "Gary"),
  Salary = c(643.3, 515.2, 671.0, 729.0, 943.25),
  Start_date = as.Date(c("2022-01-01", "2021-09-23", "2020-11-15",
"2021-05-11", "2022-03-27"))
)

```

```

# Print the employee data frame
print(employee_df)

```

## OUTPUT:

```

> # Create the employee data frame
> employee_df <- data.frame(
+   Emp_id = 1:5,
+   Emp_name = c("Ricky", "Danish", "Mini", "Ryan", "Gary"),
+   Salary = c(643.3, 515.2, 671.0, 729.0, 943.25),
+   Start_date = as.Date(c("2022-01-01", "2021-09-23", "2020-11-15", "
2021-05-11", "2022-03-27"))
+ )
>
> # Print the employee data frame
> print(employee_df)
  Emp_id Emp_name Salary Start_date
1      1   Ricky 643.30 2022-01-01
2      2  Danish 515.20 2021-09-23
3      3   Mini 671.00 2020-11-15
4      4   Ryan 729.00 2021-05-11
5      5   Gary 943.25 2022-03-27

```

## 29. Write the code to get the Structure of the R Data Frame.

### PROGRAM:

```

# Create a sample data frame
df <- data.frame(
  A = c(1, 2, 3),
  B = c("foo", "bar", "baz"),
  C = c(TRUE, FALSE, TRUE)
)

```



```
# Get the structure of the data frame  
str(df)
```

**OUTPUT:**

```
> # Create a sample data frame  
> df <- data.frame(  
+   A = c(1, 2, 3),  
+   B = c("foo", "bar", "baz"),  
+   C = c(TRUE, FALSE, TRUE)  
+ )  
>  
> # Get the structure of the data frame  
> str(df)  
'data.frame': 3 obs. of 3 variables:  
 $ A: num  1 2 3  
 $ B: chr  "foo" "bar" "baz"  
 $ C: logi  TRUE FALSE TRUE
```

**30. How to extract data from data frame for the above employee dataset.**

**PROGRAM:**

```
# Create the employee data frame  
employee_df <- data.frame(  
  Emp_id = 1:5,  
  Emp_name = c("Ricky", "Danish", "Mini", "Ryan", "Gary"),  
  Salary = c(643.3, 515.2, 671.0, 729.0, 943.25),  
  Start_date = as.Date(c("2022-01-01", "2021-09-23", "2020-11-15",  
"2021-05-11", "2022-03-27"))  
)  
  
# Extract the Emp_name and Salary columns  
emp_data <- data.frame(  
  emp_name = employee_df$Emp_name,  
  salary = employee_df$Salary  
)  
  
# Display the extracted data
```

```
print(emp_data)
```

**OUTPUT:**

```
> # Create the employee data frame
> employee_df <- data.frame(
+   Emp_id = 1:5,
+   Emp_name = c("Ricky", "Danish", "Mini", "Ryan", "Gary"),
+   Salary = c(643.3, 515.2, 671.0, 729.0, 943.25),
+   Start_date = as.Date(c("2022-01-01", "2021-09-23", "2020-11-15", "
2021-05-11", "2022-03-27"))
+ )
>
> # Extract the Emp_name and Salary columns
> emp_data <- data.frame(
+   emp_name = employee_df$Emp_name,
+   salary = employee_df$Salary
+ )
>
> # Display the extracted data
> print(emp_data)
  emp_name salary
1   Ricky 643.30
2  Danish 515.20
3    Mini 671.00
4    Ryan 729.00
5    Gary 943.25
```

**31. How to extract the first two rows and then all columns in employee data frame.**

**PROGRAM:**

```
# Create the employee data frame
```

```
employee_df <- data.frame(
  Emp_id = 1:5,
  Emp_name = c("Ricky", "Danish", "Mini", "Ryan", "Gary"),
  Salary = c(643.3, 515.2, 671.0, 729.0, 943.25),
  Start_date = as.Date(c("2022-01-01", "2021-09-23", "2020-11-15",
"2021-05-11", "2022-03-27"))
)
```

```
# Extract the first two rows and all columns
```

```
emp_data <- employee_df[1:2, ]
```

```
# Display the extracted data
```

```
print(emp_data)
```

## OUTPUT:

```
> # Create the employee data frame
> employee_df <- data.frame(
+   Emp_id = 1:5,
+   Emp_name = c("Ricky", "Danish", "Mini", "Ryan", "Gary"),
+   Salary = c(643.3, 515.2, 671.0, 729.0, 943.25),
+   Start_date = as.Date(c("2022-01-01", "2021-09-23", "2020-11-15", "20
21-05-11", "2022-03-27"))
+ )
>
> # Extract the first two rows and all columns
> emp_data <- employee_df[1:2, ]
>
> # Display the extracted data
> print(emp_data)
  Emp_id Emp_name Salary Start_date
1      1   Ricky  643.3 2022-01-01
2      2  Danish  515.2 2021-09-23
```

**32. Write a code to extract 3 rd and 5 th row with 2 nd and 4 th column of the employee data.**

## PROGRAM:

```
# Create the employee data frame
employee_df <- data.frame(
  Emp_id = 1:5,
  Emp_name = c("Ricky", "Danish", "Mini", "Ryan", "Gary"),
  Salary = c(643.3, 515.2, 671.0, 729.0, 943.25),
  Start_date = as.Date(c("2022-01-01", "2021-09-23", "2020-11-15",
"2021-05-11", "2022-03-27"))
)

# Extract the 3rd and 5th rows with the 2nd and 4th columns
emp_data <- employee_df[c(3, 5), c(2, 4)]

# Display the extracted data
print(emp_data)
```

## OUTPUT:

```
> # Create the employee data frame
> employee_df <- data.frame(
+   Emp_id = 1:5,
+   Emp_name = c("Ricky", "Danish", "Mini", "Ryan", "Gary"),
+   Salary = c(643.3, 515.2, 671.0, 729.0, 943.25),
+   Start_date = as.Date(c("2022-01-01", "2021-09-23", "2020-11-15", "20
21-05-11", "2022-03-27"))
+ )
>
```

```

> # Extract the 3rd and 5th rows with the 2nd and 4th columns
> emp_data <- employee_df[c(3, 5), c(2, 4)]
>
> # Display the extracted data
> print(emp_data)
  Emp_name Start_date
3     Mini 2020-11-15
5      Gary 2022-03-27

```

### 33. How to expand the data frame by adding rows and columns in data frame for

employee data set.

**Add Column: dept**

**( "IT", "Operations", "IT", "HR", "Finance" )**

#### PROGRAM:

```

# create the first data frame
> df1 <- data.frame(
+   ID = c(1, 2, 3, 4, 5),
+   Name = c("John", "Sara", "David", "Sarah", "Mike")
+ )
> # create the second data frame
> df2 <- data.frame(
+   ID = c(2, 4),
+   Name = c("Sara", "Sarah")
+ )
> # compare the two data frames and find rows in df1 that are not in df2
> df1_not_in_df2 <- anti_join(df1, df2, by = c("ID", "Name"))
Error in anti_join(df1, df2, by = c("ID", "Name")) :
  could not find function "anti_join"
> # print the result
> df1_not_in_df2
Error: object 'df1_not_in_df2' not found

```

35. Write a R program to find elements come only once that are common to both given data

frames.

#### CODE:

```

> # create two example data frames
> df1 <- data.frame(A = c(1, 2, 3, 4, 5),
+   B = c("apple", "banana", "cherry", "banana", "apple"))
> df2 <- data.frame(A = c(2, 4, 6),
+   B = c("banana", "apple", "orange"))
> # find elements that occur only once and are common to both data frames
> common <- intersect(df1$B, df2$B)
> result <- unique(df1$B[duplicated(df1$B) & df1$B %in% common])
> # print the result

```

```
> print(result)
[1] "banana" "apple"
```

36. Write a R program to create a data frame using two given vectors and display the duplicated elements and unique rows of the said data frame.

Practice Probs

File Read and Write Functions in R

```
Readline()
con <- file("Sample.txt", "r")
w<-readLines(con)
close(con)
w[1]
w[2]
w[3]
writeline()
sample<-c("Class,Alcohol,Malic
acid,Ash",1,14.23,1.71,2.43,"1,13.2,1.78,2.14")
writeLines(sample,"sample.csv")
```

```
dput() and dget():
# Create a data frame
x <- data.frame(Name = "Mr. A", Gender = "Male", Age=35)
#Print 'dput' output to your R console
dput(x)
#Write the 'dput' output to a file
dput(x, file = "w.R")
# Now read in 'dput' output from the file
y <- dget("w.R")
```

```
y
dump()
x<-1:10
d <- data.frame(Name = "Mr. A", Gender = "Male", Age=35)
dump(c("x","d"), file = "dump_data.R")
rm(x, d) #After dumping just remove the variables from environment.
source("dump_data.R")
```

```

x
d
str(d)
read & Write
> data <- read.csv("employee_data.csv", header =
TRUE, sep=",")
> is.data.frame(data)
[1] TRUE
> ncol(data)
[1] 9
> nrow(data)
[1] 1000
> sal <- max(data$salary)
> sal
[1] 106905
> retval <- subset(data, gender=="M")
> write.csv(retval, "output.csv")
> dim(retval)
[1] 610 9

```

#### CODE:

```

# create two vectors
> vec1 <- c("A", "B", "C", "D", "E", "F")
> vec2 <- c(1, 2, 3, 4, 5, 6)
> # create a data frame from the vectors
> df <- data.frame(vec1, vec2)
> # display the duplicated elements
> duplicated_elements <- df[duplicated(df),]
> cat("Duplicated elements:\n")
Duplicated elements:
> print(duplicated_elements)
[1] vec1 vec2
<0 rows> (or 0-length row.names)
> # display the unique rows
> unique_rows <- unique(df)
> cat("\nUnique rows:\n")

```

Unique rows:

```

> print(unique_rows)
  vec1 vec2
1  A    1
2  B    2
3  C    3
4  D    4
5  E    5

```

6 F 6