ITA0448-R PROGRAMMING

VISHWA.R

192124161(AI&DS)

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

CODE:

```
city <- c("Delhi","Bangalore","Chennai","Mumbai")
zipcode <- c(123456,789654,698748,456986)
addresses <- cbind(city,zipcode)
cat("# # # # The First data frame\n")
print(addresses)
new.address <- data.frame(
    city = c("Punjab","Kerala"),
    zipcode = c("456978","569875"),
    stringsAsFactors = FALSE
)
cat("# # # The Second data frame\n")
print(new.address)
all.addresses <- rbind(addresses,new.address)
cat("# # # The combined data frame\n")
print(all.addresses)
```

OUTPUT:

```
# # # # The First data frame
city              zipcode
[1,] "Delhi"        "123456"
[2,] "Bangalore" "789654"
```

[3,] "Chennai" "698748"

[4,] "Mumbai" "456986"

The Second data frame

city zipcode

1 Punjab 456978

2 Kerala 569875

The combined data frame

city zipcode

1 Delhi 123456

2 Bangalore 789654

3 Chennai 698748

4 Mumbai 456986

5 Punjab 456978

6 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.

CODE:

```
producers <- data.frame(
    surname =   c("Spielberg","Scorsese","Hitchcock","Tarantino","Polanski"),
    nationality = c("US","US","UK","US","Poland"),
```

```
        stringsAsFactors=FALSE)
movies <- data.frame(
        surname = c("Spielberg",
                    "Scorsese",
                        "Hitchcock",
                           "Hitchcock",
                        "Spielberg",
                        "Tarantino",
                        "Polanski"),
        title = c("Super 8",
                    "Taxi Driver",
                    "Psycho",
                    "North by Northwest",
                    "Catch Me If You Can",
                    "Reservoir Dogs","Chinatown"),
                    stringsAsFactors=FALSE)
m1 <- merge(producers, movies, by.x = "surname")
m1
dim(m1)
```

OUTPUT:

```
surname nationality                    title
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

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

CODE:

```
df = data.frame()

df = data.frame(matrix(nrow = 0, ncol = 0))

columns = c("id","name","dob")

df = data.frame(matrix(nrow = 0, ncol = length(columns)))

colnames(df) = columns

df = data.frame(id=numeric(0),name=character(0),dob=character(0))

df = data.frame(id=NA, name=NA, dob=NA)[numeric(0), ]

empty_df = df[FALSE,]
```

OUTPUT:

Empty dataset

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

CODE:

```
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\n', 'no\n', 'yes\n', 'no\n', 'no\n', 'yes\n', 'yes\n', 'no\n', 'no\n',
'yes\n')

print("Original data frame:")

print(name)

print(score)

print(attempts)

print(qualify)

df = data.frame(name, score, attempts, qualify)
```

print(df)

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\n" "no\n" "yes\n" "no\n" "no\n" "yes\n""yes\n" "no\n" "no\n"

[10] "yes\n"

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.

INPUT:

exam_data = data.frame(

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')
)
print("Original dataframe:")
print(exam_data)
print("Extract Specific columns:")
result <- data.frame(exam_data$name,exam_data$score)
print(result)
```

OUTPUT:

```
[1] "Original dataframe:"
namescoreattempts 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

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

CODE:

```
exam_data = data.frame(
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')
)
print("Original dataframe:")
print(exam_data)
print("Extract first two rows:")
result =  exam_data[1:2,]
print(result)
```

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

7. Write a R program to extract 3 rd and 5 th rows with 1 st and 3 rd columns from a

given data frame.

CODE:

```
exam_data = data.frame(

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')

)

print("Original dataframe:")

print(exam_data)

print("Extract 3rd and 5th rows with 1st and 3rd columns :")

result =  exam_data[c(3,5),c(1,3)]

print(result)
```

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

8. Write a R program to add a new column in a given data frame

CODE:

```
exam_data = data.frame(
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')
```

)

print("Original dataframe:")

print(exam_data)

print("New data frame after adding the 'country' column:")

exam_data$country = c("USA","USA","USA","USA","USA","USA","USA","USA","USA","USA")

print(exam_data)

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

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

CODE:

```
exam_data = data.frame(

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')

)

print("Original dataframe:")

print(exam_data)

new_exam_data = data.frame(

name = c('Robert', 'Sophia'),

score = c(10.5, 9),

attempts = c(1, 3),

qualify = c('yes', 'no')

)

exam_data =    rbind(exam_data, new_exam_data)

print("After adding new row(s) to an existing data frame:")

print(exam_data)
```

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

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

CODE:

```
exam_data = data.frame(
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')
)
print("Original dataframe:")
print(exam_data)
exam_data = subset(exam_data, select = -c(name, qualify))
print(exam_data)
```

Output:

```
[1] &quot;Original dataframe:&quot;
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

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

CODE:

```
exam_data = data.frame(
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')
)
print("Original dataframe:")
print(exam_data)
exam_data = subset(exam_data, select = -c(name, qualify))
print(exam_data)
```

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

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

CODE:

```
exam_data = data.frame(

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')

)
```

```
print("Original dataframe:")
print(exam_data)
print("dataframe after sorting 'name' and 'score' columns:")
exam_data = exam_data[with(exam_data, order(name, score)), ]
print(exam_data)
```

Output:

```
[1] &quot;Original dataframe:&quot;
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] &quot;dataframe after sorting &#39;name&#39; and &#39;score&#39; columns:&quot;
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.

CODE:

```
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:

```
[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.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.

CODE:

```
exam_data = data.frame(

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, NA, 2, NA, 2, NA, 1, NA, 2, 1),

qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')

)

print("Original dataframe:")

print(exam_data)

exam_data[is.na(exam_data)] = 3

print("After removing NA with 3, the said dataframe becomes:")

print(exam_data)
```

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] "After removing NA with 3, the said dataframe becomes:"

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 3 no

9 Kevin 8.0 2 no

10 Jonas 19.0 1 yes

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

CODE:

```
exam_data = data.frame(

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, NA, 2, NA, 2, NA, 1, NA, 2, 1),
```

```
qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')

)

print("Original dataframe:")

print(exam_data)

print("Change column-name 'name' to 'student_name' of the said dataframe:")

colnames(exam_data)[which(names(exam_data) == "name")] =
"student_name"

print(exam_data)
```

Output:

```
[1] &quot;Original dataframe:&quot;

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] &quot;Change column-name &#39;name&#39; to &#39;student_name&#39;
of the said dataframe:&quot;

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.

CODE:

```
exam_data = data.frame(

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, NA, 2, NA, 2, NA, 1, NA, 2, 1),

qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')

)

print("Original dataframe:")

print(exam_data)

print("Change more than one column name    of the said dataframe:")

colnames(exam_data)[which(names(exam_data) == "name")] = "student_name"

colnames(exam_data)[which(names(exam_data) == "score")] = "avg_score"

print(exam_data)
```

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 more than one column name of the said dataframe:"

student_nameavg_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.

CODE:

exam_data = data.frame(

```
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')

)

print("Original dataframe:")

print(exam_data)

print("Select three random rows of the said dataframe:")

print(exam_data[sample(nrow(exam_data), 3),])
```

Output:

```
[1] &quot;Original dataframe:&quot;

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] &quot;Select three random rows of the said dataframe:&quot;

name score attempts qualify

10 Jonas 19.0 1 yes

7 Matthew 14.5 1 yes
```

4 James 12.0 3 no

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

CODE:

```
exam_data = data.frame(
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')
)
print("Original dataframe:")
print(exam_data)
print("Reorder by column name:")
exam_data = exam_data[c("name", "attempts", "score", "qualify")]
print(exam_data)
```

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] "Reorder by column name:"

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.

CODE:

```
df_90 = data.frame(
   "item" = c("item1", "item2", "item3"),
   "Jan_sale" = c(12, 14, 12),
   "Feb_sale" = c(11, 12, 15),
   "Mar_sale" = c(12, 14, 15)
)
df_91 = data.frame(
   "item" = c("item1", "item2", "item3"),
   "Jan_sale" = c(12, 14, 12),
```

```
  "Feb_sale" = c(11, 12, 15),
  "Mar_sale" = c(12, 15, 18)
)
print("Original Dataframes:")
print(df_90)
print(df_91)
print("Row(s) in first data frame that are not present in second data frame:")
print(setdiff(df_90,df_91))
```

Output:

```
[1] "Original Dataframes:"
    item Jan_sale Feb_sale Mar_sale
1 item1       12       11       12
2 item2       14       12       14
3 item3       12       15       15
    item Jan_sale Feb_sale Mar_sale
1 item1       12       11       12
2 item2       14       12       15
3 item3       12       15       18
[1] "Row(s) in first data frame that are not present in second data frame:"
  Mar_sale
1       12
2       14
3       15
```

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

CODE:

```
a = c("a", "b", "c", "d", "e")
```

```
b = c("d", "e", "f", "g")

print("Original Dataframes")

print(a)

print(b)

print("Elements which are present in both dataframe:")

result = intersect(a, b)

print(result)
```

Output:

```
[1] "Original Dataframes"

[1] "a" "b" "c" "d" "e"

[1] "d" "e" "f" "g"

[1] "Elements which are present in both dataframe:"

[1] "d" "e"
```

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

given data frames.

CODE:

```
a = c("a", "b", "c", "d", "e")

b = c("d", "e", "f", "g")

print("Original Dataframes")

print(a)

print(b)

print("Find elements come only once that are common to both given dataframes:")

result = union(a, b)

print(result)
```

Output:

[1] "Original Dataframes"

[1] "a" "b" "c" "d" "e"

[1] "d" "e" "f" "g"

[1] "Find elements come only once that are common to both given dataframes:"

[1] "a" "b" "c" "d" "e" "f" "g"

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

the information of the file.

CODE:

```
exam_data = data.frame(

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')

)

print("Original dataframe:")

print(exam_data)

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

load("data.rda")

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.

CODE:

```
exam_data = data.frame(
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, NA, 2, NA, 2, NA, 1, NA, 2, 1),
qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')
)
print("Original dataframe:")
print(exam_data)
print("The number of NA values in attempts column:")
print(sum(is.na(exam_data$attempts)))
```

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] "The number of NA values in attempts column:"

[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.

CODE:

```
a = c(10,20,10,10,40,50,20,30)

b = c(10,30,10,20,0,50,30,30)

print("Original data frame:")

ab = data.frame(a,b)

print(ab)

print("Duplicate elements of the said data frame:")

print(duplicated(ab))

print("Unique rows of the said data frame:")

print(unique(ab))
```

Output:

[1] "Original data frame:"

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

[1] "Duplicate elements of the said data frame:"

[1] FALSE FALSE TRUE FALSE FALSEFALSE TRUE FALSE

[1] Unique rows of the said data frame:

a b

1 10 10

2 20 30

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.

CODE:

data = airquality

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

print(class(data))

print(head(data,10))

result = data[order(data[,1]),]

print("Order the entire data frame by the first and second column:")

print(result)
```

Output:

```
[1] "Original data: Daily air quality measurements in New York, May to September

1973".

[1] "data.frame"

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

7 23 299 8.6 65 5 7

8 19 99 13.8 59 5 8

9 8 19 20.1 61 5 9

10 NA 194 8.6 69 5 10

[1]"Order the entire data frame by the first and second column:"

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

.....

119 NA 153 5.7 88 8 27

150 NA 145 13.2 77 9 27

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

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

CODE:

```
data = airquality

print("Original data: Daily air quality measurements in New York, May to September 1973.")

print(data)

data[,c("Solar.R")]=NULL

data[,c("Wind")]=NULL

print("data.frame after removing 'Solar.R'    and 'Wind' variables:")

print(data)
```

Output:

[1] "Original data: Daily air quality measurements in New York, May to September

1973."

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

.....

152 18 131 8.0 76 9 29

153 20 223 11.5 68 9 30

[1] "data.frame after removing 'Solar.R' and 'Wind' variables:"

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

.....

152 18 76 9 29

153 20 68 9 30

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

SOLN:

Tables, Spreadsheets, Database tables.

Example:

Let's consider an example to understand the difference between Data Frames and other Data Structures. Suppose we have a dataset containing information about students in a class, including their names, ages, grades, and subjects. We want to analyze this data and find out which students are performing well in which subjects. Here are some ways we can represent this data:

Array: We can use a three-dimensional array to represent this data, where the first dimension represents the student, the second dimension represents the subject, and the third dimension represents the variable (name, age, grade). However, this can be difficult to work with, and we would need to use complex indexing to access specific values.

Linked List: We can use a linked list to represent each student, where each node in the list contains the student's information. However, this would not

allow us to easily compare or analyze data across multiple students.

Data Frame: We can use a data frame to represent this data, where each column represents a variable (name, age, grade, subject), and each row represents a student. This would allow us to easily compare and analyze data across multiple students and subjects.

In summary, while other data structures like arrays and linked lists can be used to represent data, they may not be as efficient or convenient for analyzing complex data sets like those found in a data frame.

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

Emp_id = 1:5

CODE:

```
> 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 = c("2022-01-01", "2021-09-23", "2020-11-15", "2021-05-11","2022-03-27")
+ )
> # print the data frame
> employee_df
```

Output:

	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.

CODE:

```
> df <- data.frame(
+     x = c(1, 2, 3),
+     y = c("A", "B", "C"),
+     z = c(TRUE, FALSE, TRUE)
+ )
> # get the structure of the data frame
> str(df)
```

Output:

```
data.frame':   3 obs. of    3 variables:
 $ x: num    1 2 3
 $ y: chr    "A" "B" "C"
 $ z: logi    TRUE FALSE TRUE
```

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

CODE:

```
> 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 = c("2022-01-01", "2021-09-23", "2020-11-15",
"2021-05-11","2022-03-27")
+ )
> # extract employee names and salaries
> emp_names <- employee_df$Emp_name
> emp_salaries <- employee_df$Salary
> # create a data frame with the extracted data
> emp_data <- data.frame(emp_name = emp_names, salary = emp_salaries)
> # print the data frame
```

> emp_data

Output:

	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.

CODE:

> employee_df[1:2,]

Output:

	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.

CODE:

> employee_df[c(3,5), c(2,4)]

	Emp_name	Start_date
3	Mini	2020-11-15
5	Gary	2022-03-27

Data Reshaping:

Data reshaping means changing how data is represented in rows and column. It includes

splitting, merging or interchanging the rows and columns.

Reshaping functions:

- cbind()
- rbind()
- mergr()

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

employee data set.

CODE:

emp_id = 6:8,

emp_name = "Rasmi","Pranab","Tusar",

salary =578.0,722.5,632.8,

start_date = "2022-05-21","2020-07-30","2019-06-17",

dept = "IT","Operations","Fianance",

Expected Output:

emp_id = 6:8,

emp_name = "Rasmi","Pranab","Tusar",

salary =578.0,722.5,632.8,

start_date = "2022-05-21","2020-07-30","2019-06-17",

dept = "IT","Operations","Fianance&quot

34. Write a R program to compare two data frames to find the row(s) in first data frame that

are not present in second data frame.

CODE:

```
> 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")
+ )
>
> 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:

```
a = c("a", "b", "c", "d", "e")
b = c("d", "e", "f", "g")
print("Original Dataframes")
print(a)
print(b)
print("Find elements come only once that are common to both given dataframes:")
```

```
result = union(a, b)

print(result)
```

Output:

```
[1] "Original Dataframes"

[1] "a" "b" "c" "d" "e"

[1] "d" "e" "f" "g"

[1] "Find elements come only once that are common to both given dataframes:"

[1] "a" "b" "c" "d" "e" "f" "g"
```

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.

CODE:

```
a = c(10,20,10,10,40,50,20,30)

b = c(10,30,10,20,0,50,30,30)

print("Original data frame:")

ab = data.frame(a,b)

print(ab)

print("Duplicate elements of the said data frame:")

print(duplicated(ab))

print("Unique rows of the said data frame:")

print(unique(ab))
```

Output:

```
[1] "Original data frame:"

    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

[1] "Duplicate elements of the said data frame:"

[1] FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE

[1] "Unique rows of the said data frame:"

a b

1 10 10

2 20 30

4 10 20

5 40 0

6 50 50

8 30 30