|  |
| --- |
| **ITA0448 Statistics with R Programming for Vectorized Expressions** |
| **NAME : RAMYA B**  **REG NO : 192121038**  **DAY 3** |

**ASSESSMENT 1**

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

**Source Code:**

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

**Source Code:**

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"

1. **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:**

**surname nationality movies**

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

**Source Code:**

1. **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

Source Code:

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

1. **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

Source 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","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

1. **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

**Source 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","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

**Source 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","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 |
|  |
| |  | | --- | |  | |

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

**Source 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","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

1. **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

**Source 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', '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

**Source 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","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

**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

**Source 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', '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

1. **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

**Source 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', '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

1. **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

**Source 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', '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

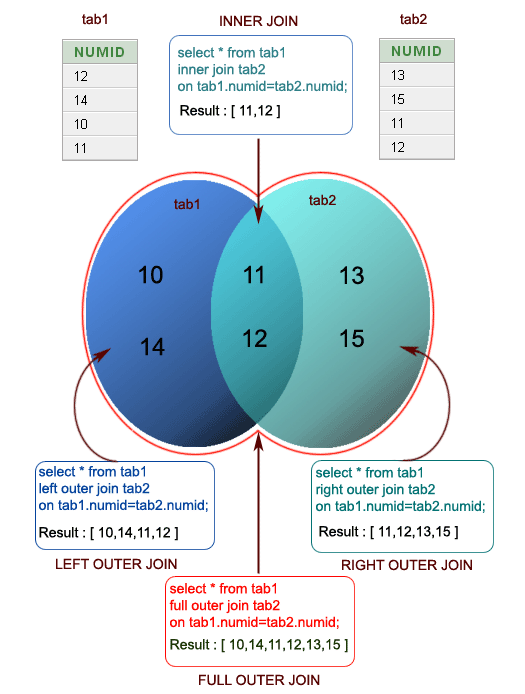
8 Laura 13.5 1 no

7 Matthew 14.5 1 yes

6 Michael 20.0 3 yes

**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

1. 11 12

Source 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:

|  |
| --- |
| 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 |
|  |
| |  | | --- | | > | |

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

**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

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

**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

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

**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

1. **Write a R program to select some random 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] "Select three random rows of the said dataframe:"

name score attempts qualify

10 Jonas 19.0 1 yes

7 Matthew 14.5 1 yes

4 James 12.0 3 no

1. **Write a R program to reorder an given data frame by 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] "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

1. 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.

**Output:**

[1] "Original Dataframes"

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

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

[1] "Data in first dataframe that are not present in second dataframe:"

[1] "a" "b" "c"

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

**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"

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

**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"

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

**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

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

**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

1. **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.**

**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

1. **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.**

**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

1. **Write a R program to call the (built-in) dataset airquality. Remove the variables 'Solar.R' and 'Wind' and display the data frame.**

**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