**NAME**: V.RAMYA SRI

**REG.NO:**192124039

**COURSE CODE:** ITA0448

**ASSESMENT DAY 3**

**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**

**rbind() function:**

**Output:**

**city zipcode**

**[1] delhi 123456**

**[2] bangalore 789654**

**[3] chennai 698748**

**[4] mumbai 456986**

**[5] punjab 456978**

**[6] kerala 569875**

**INPUT**

# Create vector objects.

city <- c(“delhi”,”banglore”,”chennai”,”Mumbai”)

state <- c("delhi","karnataka","tamilnadu","kolkata")

zipcode <- c(“123456”,”789654”,”698748”,”456986”)

addresses <- cbind(city,state,zipcode)

cat("# # # # The First data frame\n")

# Print the data frame.

print(addresses)

# Create another data frame with similar columns

new.address <- data.frame(

city = c("Punjab”,”kerala”),

state = c("gujarat","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)

**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:**

**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**

**INPUT**

# Create origin dataframe(

producers <- data.frame(

surname = c("Spielberg","Scorsese","Hitchcock","Tarantino","Polanski"),

nationality = c("US","US","UK","US","Poland"),

stringsAsFactors=FALSE)

# Create destination dataframe

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)

# Merge two datasets

m1 <- merge(producers, movies, by.x = "surname")

m1

dim(m1)

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

**Output:**

**[1] &quot;Structure of the empty dataframe:&quot;**

**&#39;data.frame&#39;: 0 obs. of 5 variables:**

**$ Ints : int**

**$ Doubles : num**

**$ Characters: chr**

**$ Logicals :logi**

**$ Factors : Factor w/ 0 levels:**

**NULL**

**INPUT**

df = data.frame(Ints=integer(),

Doubles=double(),

Characters=character(),

Logicals=logical(),

Factors=factor(),

stringsAsFactors=FALSE)

print("Structure of the empty dataframe:")

print(str(df))

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

**name = c(&#39;Anastasia&#39;, &#39;Dima&#39;, &#39;Katherine&#39;, &#39;James&#39;, &#39;Emily&#39;, &#39;Michael&#39;, &#39;Matthew&#39;,**

**&#39;Laura&#39;, &#39;Kevin&#39;, &#39;Jonas&#39;)**

**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(&#39;yes&#39;, &#39;no&#39;, &#39;yes&#39;, &#39;no&#39;, &#39;no&#39;, &#39;yes&#39;, &#39;yes&#39;, &#39;no&#39;, &#39;no&#39;, &#39;yes&#39;)**

**INPUT**

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 data frame:")

print(name)

print(score)

print(attempts)

print(qualify)

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

print(df)

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

**name.**

**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;Extract Specific columns:&quot;**

**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**

**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)

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

**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;Extract first two rows:&quot;**

**name score attempts qualify**

**1 Anastasia 12.5 1 yes**

**2 Dima 9.0 3 no**

**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 first two rows:")

result = exam\_data[1:2,]

print(result)

**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.**

**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;Extract 3rd and 5th rows with 1st and 3rd columns :&quot;**

**name attempts**

**3 Katherine 2**

**5 Emily 2**

**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 3rd and 5th rows with 1st and 3rd columns :")

result = exam\_data[c(3,5),c(1,3)]

print(result)

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

**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;New data frame after adding the &#39;country&#39; column:&quot;**

**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**

**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("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)

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

**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;After adding new row(s) to an existing data frame:&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**

**11 Robert 10.5 1 yes**

**12 Sophia 9.0 3 no**

**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)

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)

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

**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**

**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)

exam\_data = subset(exam\_data, select = -c(name, qualify))

print(exam\_data)

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

**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**

**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**

**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)

exam\_data <- exam\_data[-c(2, 4, 6), ]

print(exam\_data)

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

**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**

**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("dataframe after sorting 'name' and 'score' columns:")

exam\_data = exam\_data[with(exam\_data, order(name, score)), ]

print(exam\_data)

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

**data frames.**

**Output:**

**The output will be in pictorial presentation.**

**[1] &quot;Left outer Join:&quot;**

**numid**

**1 10**

**2 11**

**3 12**

**4 14**

**[1] &quot;Right outer Join:&quot;**

**numid**

**1 11**

**2 12**

**3 13**

**4 15**

**[1] &quot;Outer Join:&quot;**

**numid**

**1 10**

**2 11**

**3 12**

**4 13**

**5 14**

**6 15**

**[1] &quot;Cross Join:&quot;**

**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**

**INPUT**

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)

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

**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;After removing NA with 3, the said dataframe becomes:&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 3 no**

**9 Kevin 8.0 2 no**

**10 Jonas 19.0 1 yes**

**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, 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)

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

**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**

**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, 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)

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

**frame.**

**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 more than one column name of the said dataframe:&quot;**

**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**

**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, 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)

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

**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**

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

print(exam\_data[sample(nrow(exam\_data), 3),])

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

**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;Reorder by column name:&quot;**

**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**

**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("Reorder by column name:")

exam\_data = exam\_data[c("name", "attempts", "score", "qualify")]

print(exam\_data)

**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.**

**Output:**

**[1] &quot;Original Dataframes&quot;**

**[1] &quot;a&quot; &quot;b&quot; &quot;c&quot; &quot;d&quot; &quot;e&quot;**

**[1] &quot;d&quot; &quot;e&quot; &quot;f&quot; &quot;g&quot;**

**[1] &quot;Data in first dataframe that are not present in second dataframe:&quot;**

**[1] &quot;a&quot; &quot;b&quot; &quot;c&quot;**

**INPUT**

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))

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

**Output:**

**[1] &quot;Original Dataframes&quot;**

**[1] &quot;a&quot; &quot;b&quot; &quot;c&quot; &quot;d&quot; &quot;e&quot;**

**[1] &quot;d&quot; &quot;e&quot; &quot;f&quot; &quot;g&quot;**

**[1] &quot;Elements which are present in both dataframe:&quot;**

**[1] &quot;d&quot; &quot;e&quot;**

**INPUT**

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)

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

**given data frames.**

**Output:**

**[1] &quot;Original Dataframes&quot;**

**[1] &quot;a&quot; &quot;b&quot; &quot;c&quot; &quot;d&quot; &quot;e&quot;**

**[1] &quot;d&quot; &quot;e&quot; &quot;f&quot; &quot;g&quot;**

**[1] &quot;Find elements come only once that are common to both given dataframes:&quot;**

**[1] &quot;a&quot; &quot;b&quot; &quot;c&quot; &quot;d&quot; &quot;e&quot; &quot;f&quot; &quot;g&quot;**

**INPUT**

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)

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

**the information of the file.**

**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**

**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**

**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)

save(exam\_data,file="data.rda")

load("data.rda")

file.info("data.rda")

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

**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;The number of NA values in attempts column:&quot;**

**[1] 4**

**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, 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)))

**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.**

**Output:**

**[1] &quot;Original data frame:&quot;**

**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] &quot;Duplicate elements of the said data frame:&quot;**

**[1] FALSE FALSE TRUE FALSE FALSEFALSE TRUE FALSE**

**[1] &quot;Unique rows of the said data frame:&quot;**

**a b**

**1 10 10**

**2 20 30**

**4 10 20**

**5 40 0**

**6 50 50**

**8 30 30**

**INPUT**

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))

**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.**

**Output:**

**[1] &quot;Original data: Daily air quality measurements in New York, May to September**

**1973.&quot;**

**[1] &quot;data.frame&quot;**

**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] &quot;Order the entire data frame by the first and second column:&quot;**

**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**

**INPUT**

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)

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

**&#39;Solar.R&#39; and &#39;Wind&#39; and display the data frame.**

**Output:**

**[1] &quot;Original data: Daily air quality measurements in New York, May to September**

**1973.&quot;**

**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] &quot;data.frame after removing &#39;Solar.R&#39; and &#39;Wind&#39; variables:&quot;**

**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**

**INPUT**

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)

**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**

A data structure is a particular way of organizing data in a computer so that it can be used effectively. The idea is to reduce the space and time complexities of different tasks. Data structures in R programming are tools for holding multiple values.

R’s base data structures are often organized by their dimensionality (1D, 2D, or nD) and whether they’re homogeneous (all elements must be of the identical type) or heterogeneous (the elements are often of various types). This gives rise to the six data types which are most frequently utilized in data analysis.

The most essential data structures used in R include:

Vectors

Lists

Dataframes

Matrices

Arrays

Factors

Vectors

A vector is an ordered collection of basic data types of a given length. The only key thing here is all the elements of a vector must be of the identical data type e.g homogeneous data structures. Vectors are one-dimensional data structures.

Example:

# R program to illustrate Vector

# Vectors(ordered collection of same data type)

X = c(1, 3, 5, 7, 8)

# Printing those elements in console

print(X)

Output:

[1] 1 3 5 7 8

Lists

A list is a generic object consisting of an ordered collection of objects. Lists are heterogeneous data structures. These are also one-dimensional data structures. A list can be a list of vectors, list of matrices, a list of characters and a list of functions and so on.

Example:

# R program to illustrate a List

# The first attributes is a numeric vector

# containing the employee IDs which is

# created using the 'c' command here

empId = c(1, 2, 3, 4)

# The second attribute is the employee name

# which is created using this line of code here

# which is the character vector

empName = c("Debi", "Sandeep", "Subham", "Shiba")

# The third attribute is the number of employees

# which is a single numeric variable.

numberOfEmp = 4

# We can combine all these three different

# data types into a list

# containing the details of employees

# which can be done using a list command

empList = list(empId, empName, numberOfEmp)

print(empList)

Output:

[[1]]

[1] 1 2 3 4

[[2]]

[1] "Debi" "Sandeep" "Subham" "Shiba"

[[3]]

[1] 4

Dataframes

Dataframes are generic data objects of R which are used to store the tabular data. Dataframes are the foremost popular data objects in R programming because we are comfortable in seeing the data within the tabular form. They are two-dimensional, heterogeneous data structures. These are lists of vectors of equal lengths.

Data frames have the following constraints placed upon them:

A data-frame must have column names and every row should have a unique name.

Each column must have the identical number of items.

Each item in a single column must be of the same data type.

Different columns may have different data types.

To create a data frame we use the data.frame() function.

Example:

# R program to illustrate dataframe

# A vector which is a character vector

Name = c("Amiya", "Raj", "Asish")

# A vector which is a character vector

Language = c("R", "Python", "Java")

# A vector which is a numeric vector

Age = c(22, 25, 45)

# To create dataframe use data.frame command

# and then pass each of the vectors

# we have created as arguments

# to the function data.frame()

df = data.frame(Name, Language, Age)

print(df)

Output:

Name Language Age

1 Amiya R 22

2 Raj Python 25

3 Asish Java 45

Matrices

A matrix is a rectangular arrangement of numbers in rows and columns. In a matrix, as we know rows are the ones that run horizontally and columns are the ones that run vertically. Matrices are two-dimensional, homogeneous data structures.

Now, let’s see how to create a matrix in R. To create a matrix in R you need to use the function called matrix. The arguments to this matrix() are the set of elements in the vector. You have to pass how many numbers of rows and how many numbers of columns you want to have in your matrix and this is the important point you have to remember that by default, matrices are in column-wise order.

Example:

# R program to illustrate a matrix

A = matrix(

# Taking sequence of elements

c(1, 2, 3, 4, 5, 6, 7, 8, 9),

# No of rows and columns

nrow = 3, ncol = 3,

# By default matrices are

# in column-wise order

# So this parameter decides

# how to arrange the matrix

byrow = TRUE

)

print(A)

Output:

[,1] [,2] [,3]

[1,] 1 2 3

[2,] 4 5 6

[3,] 7 8 9

Arrays

Arrays are the R data objects which store the data in more than two dimensions. Arrays are n-dimensional data structures. For example, if we create an array of dimensions (2, 3, 3) then it creates 3 rectangular matrices each with 2 rows and 3 columns. They are homogeneous data structures.

Now, let’s see how to create arrays in R. To create an array in R you need to use the function called array(). The arguments to this array() are the set of elements in vectors and you have to pass a vector containing the dimensions of the array.

Example:

# R program to illustrate an array

A = array(

# Taking sequence of elements

c(1, 2, 3, 4, 5, 6, 7, 8),

# Creating two rectangular matrices

# each with two rows and two columns

dim = c(2, 2, 2)

)

print(A)

Output:

, , 1

[,1] [,2]

[1,] 1 3

[2,] 2 4

, , 2

[,1] [,2]

[1,] 5 7

[2,] 6 8

Factors

Factors are the data objects which are used to categorize the data and store it as levels. They are useful for storing categorical data. They can store both strings and integers. They are useful to categorize unique values in columns like “TRUE” or “FALSE”, or “MALE” or “FEMALE”, etc.. They are useful in data analysis for statistical modeling.

Now, let’s see how to create factors in R. To create a factor in R you need to use the function called factor(). The argument to this factor() is the vector.

Example:

# R program to illustrate factors

# Creating factor using factor()

fac = factor(c("Male", "Female", "Male",

"Male", "Female", "Male", "Female"))

print(fac)

Output:

[1] Male Female Male Male Female Male Female

Levels: Female Male

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

**Emp\_id = 1:5**

**Emp\_name = &quot;Ricky&quot;,&quot;Danish&quot;,&quot;Mini&quot;,&quot;Ryan&quot;,&quot;Gary&quot;**

**Salary = 643.3,515.2,671.0,729.0,943.25**

**Start\_date = &quot;2022-01-01&quot;, &quot;2021-09-23&quot;, &quot;2020-11-15&quot;, &quot;2021-05-11&quot;,&quot;2022-03-**

**27&quot;**

**INPUT**

Employees = data.frame(Name=c("Anastasia S","Dima R","Katherine S", "JAMES A","LAURA MARTIN"),

Gender=c("M","M","F","F","M"),

Age=c(23,22,25,26,32),

Designation=c("Clerk","Manager","Exective","CEO","ASSISTANT"),

SSN=c("123-34-2346","123-44-779","556-24-433","123-98-987","679-77-576")

)

print("Details of the employees:")

print(Employees)

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

**INPUT**

# Create the data frame.

emp.data <- data.frame(

emp\_id = c (1:5),

emp\_name = c("Rick","Dan","Michelle","Ryan","Gary"),

salary = c(623.3,515.2,611.0,729.0,843.25),

start\_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",

"2015-03-27")),

stringsAsFactors = FALSE

)

# Print the data frame.

print(emp.data)

OUTPUT

emp\_id emp\_name salary start\_date

1 1 Rick 623.30 2012-01-01

2 2 Dan 515.20 2013-09-23

3 3 Michelle 611.00 2014-11-15

4 4 Ryan 729.00 2014-05-11

5 5 Gary 843.25 2015-03-27

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

**Expected Output:**

**emp.data.emp\_name. emp.data.salary**

**1 Ricky 643.30**

**2 Danish 515.20**

**3 Mini 671.00**

**4 Ryan 729.00**

**5 Gary 943.25**

**INPUT**

# Create the data frame.

emp.data <- data.frame(

emp\_id = c (1:5),

emp\_name = c("Rick","Dan","Michelle","Ryan","Gary"),

salary = c(623.3,515.2,611.0,729.0,843.25),

start\_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",

"2015-03-27")),

stringsAsFactors = FALSE

)

# Print the data frame.

print(emp.data)

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

**Expected Output:**

**emp\_idemp\_name salary start\_date**

**1 Ricky 643.3 2012-01-01**

**2 Danish 515.2 2013-09-23**

**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 first two rows:")

result = exam\_data[1:2,]

print(result)

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

**data.**

**Expected Output:**

**emp\_namestart\_date**

**3 Mini 2014-11-15**

**5 Gary 2015-03-27**

**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()**

**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 3rd and 5th rows with 1st and 3rd columns :")

result = exam\_data[c(3,5),c(1,3)]

print(result)

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

**employee data set.**

**Add Column: dept&lt;- c(&quot;IT&quot;,&quot;Operations&quot;,&quot;IT&quot;,&quot;HR&quot;,&quot;Finance&quot;)**

**Expected Output:**

**emp\_idemp\_name salary start\_date dept**

**1 Ricky 643.30 2012-01-01 IT**

**2 Danish 515.20 2013-09-23 Operations**

**3 Mini 671.00 2014-11-15**

**4 Ryan 729.00 2014-05-11 HR**

**5 Gary 943.25 2015-03-27 Finance**

**Add Row using the second dataframe given below:**

**emp\_id = 6:8,**

**emp\_name = &quot;Rasmi&quot;,&quot;Pranab&quot;,&quot;Tusar&quot;,**

**salary =578.0,722.5,632.8,**

**start\_date = &quot;2022-05-21&quot;,&quot;2020-07-30&quot;,&quot;2019-06-17&quot;,**

**dept = &quot;IT&quot;,&quot;Operations&quot;,&quot;Fianance&quot;,**

**Expected Output:**

**emp\_idemp\_name salary start\_date dept**

**1 Ricky 643.30 2012-01-01 IT**

**2 Danish 515.20 2013-09-23 Operations**

**3 Mini 671.00 2014-11-15 IT**

**4 Ryan 729.00 2014-05-11 HR**

**5 Gary 943.25 2015-03-27 Finance**

**6 Rasmi 578.00 2013-05-21 IT**

**7 Pranab 722.50 2013-07-30 Operations**

**8 Tusar 632.80 2014-06-17 Fianance**

**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("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)

**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.**

**INPUT**

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

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

**frames.**

**INPUT**

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.**

**Input**

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