Subscribe: https://youtube.com/@ataglanceofficial

R Questions Solution (BDA)

1.

b) Consider the following data frame given below: [10]

subject	class	marks
1	1	56
2	2	75
3	1	48
4	2	69
5	1	84
6	2	53

i. Create a subset of subject less than 4 by using subset () function and demonstrate the output.

ii. Create a subset where the subject column is less than 3 and the class equals to 2 by using [ ] brackets and demonstrate the output.

Solution:

(i)

```
# Create a sample data frame with the given marks values
data <- data.frame(
  subject = c(1, 2, 3, 4, 5, 6),
  class = c(1, 2, 1, 2, 1, 2),
  marks = c(56, 75, 48, 69, 84, 53)
)

# Display the original data frame
print("Original Data Frame:")
print(data)

# Create a subset where subject is less than 4
subset_data <- subset(data, subject < 4)

# Display the subset data frame
print("Subset Data Frame (subject < 4):")
print(subset_data)
```

Subscribe: https://youtube.com/@ataglanceofficial

Output:

```
[1] "Original Data Frame:"
subject class marks
1       1     1    56
2       2     2    75
3       3     1    48
4       4     2    69
5       5     1    84
6       6     2    53
[1] "Subset Data Frame (subject < 4):"
subject class marks
1       1     1    56
2       2     2    75
3       3     1    48
```

This code creates a data frame with the specified marks values and then creates a subset where the "subject" is less than 4.

(ii)

```
# Create a sample data frame with the given marks values
data <- data.frame(
  subject = c(1, 2, 3, 4, 5, 6),
  class = c(1, 2, 1, 2, 1, 2),
  marks = c(56, 75, 48, 69, 84, 53)
)

# Display the original data frame
print("Original Data Frame:")
print(data)

# Create a subset where subject is less than 3 and class equals
to 2
subset_data <- data[data$subject < 3 & data$class == 2, ]

# Display the subset data frame
print("Subset Data Frame (subject < 3 and class == 2):")
print(subset_data)
```

Subscribe: https://youtube.com/@ataglanceofficial

Output:

```
[1] "Original Data Frame:"
subject class marks
1       1     1    56
2       2     2    75
3       3     1    48
4       4     2    69
5       5     1    84
6       6     2    53
[1] "Subset Data Frame (subject < 3 and class == 2):"
subject class marks
2       2     2    75
```

In this example, data$subject < 3 checks if the "subject" is less than 3, and data$class == 2 checks if the "class" is equal to 2. The resulting subset includes rows where both conditions are true. The subset is created using square brackets [].

Subscribe: https://youtube.com/@ataglanceofficial

2.

Q6 a) i. The following table shows the number of units of different products sold on different days: [10]

Product	Monday	Tuesday	Wednesday	Thursday	Friday
Bread	12	3	5	11	9
Milk	21	27	18	20	15
Cola Cans	10	1	33	6	12
Chocolate bars	6	7	4	13	12
Detergent	5	8	12	20	23

Create five sample numeric vectors from this data.

ii. Name and explain the operators used to form data subsets in R.

questionpanel

Solution:

(i)

```
# Create a data frame for the given sales data
sales_data <- data.frame(
  product = c("bread", "milk", "cola cans", "chocolate bars",
"detergent"),
  monday = c(12, 21, 10, 6, 5),
  tuesday = c(3, 27, 1, 7, 8),
  wednesday = c(5, 18, 33, 4, 12),
  thursday = c(11, 20, 6, 13, 20),
  friday = c(9, 15, 12, 12, 23)
)

# Display the sales data table
print("Sales Data Table:")
print(sales_data)

# Create five sample numeric vectors
sample_vector1 <- sales_data[sample(1:nrow(sales_data)),
"monday"]
sample_vector2 <- sales_data[sample(1:nrow(sales_data)),
"tuesday"]
sample_vector3 <- sales_data[sample(1:nrow(sales_data)),
"wednesday"]
```

```
sample_vector4 <- sales_data[sample(1:nrow(sales_data)),
"thursday"]
sample_vector5 <- sales_data[sample(1:nrow(sales_data)),
"friday"]

# Display the sample vectors
print("Sample Numeric Vectors:")
print(sample_vector1)
print(sample_vector2)
print(sample_vector3)
print(sample_vector4)
print(sample_vector5)
```

Output:

```
[1]"Sales Data Table:"
product monday tuesday wednesdaythursday friday
1          bread     12       3         5       11       9
2           milk     21      27        18       20      15
3      cola cans     10       1        33        6      12
4 chocolate bars      6       7         4       13      12
5      detergent      5       8        12       20      23
[1] "Sample Numeric Vectors:"
[1]  5 10 12 21  6
[1]  8  1 27  3  7
[1] 12 33  5 18  4
[1] 13  620 20 11
[1] 15  9 12 23 12
```

In the given sales data table, each row represents a different product, and each column from Monday to Friday represents the number of units sold for that product on each respective day. The sample numeric vectors are randomly selected columns from this table, representing the sales data for a particular day across all products. These vectors can be used for further analysis, such as calculating daily averages or comparing the sales performance of different products on a specific day.

Subscribe: https://youtube.com/@ataglanceofficial

(ii) Name operators used to form data subsets

In R, the primary operators and functions for forming data subsets include:

1. Square Brackets [] Operator:
   - Syntax: `data[rows, columns]`
   - Explanation: Used to subset data frames or matrices based on specific row and column indices.
2. Logical Operators (&, |, !):
   - Syntax: `data[logical_condition, ]`
   - Explanation: Logical operators are employed to create conditions for subsetting data based on specific criteria.
3. Subset() Function:
   - Syntax: `subset(data, logical_condition, select = c(columns))`
   - Explanation: The `subset()` function is designed for concise subsetting of data frames using logical conditions.
4. %in% Operator:
   - Syntax: `data[data$column %in% c(values), ]`
   - Explanation: Checks if elements in a column are present in a specified set of values, commonly used for categorical variables.
5. Double Square Brackets [[]] and $ Operator for Extracting Columns:
   - Syntax: `data[[column_name]]` or `data$column_name`
   - Explanation: Used to extract a specific column from a data frame.

These operators and functions offer flexibility in manipulating and analyzing data, allowing users to efficiently subset datasets based on various conditions or criteria.

Subscribe: https://youtube.com/@ataglanceofficial

3.

b) The data analyst of Argon technology Mr. John needs to enter the salaries of 10 employees in R. The salaries of the employees are given in the following table: [10]

Sr. No.	Name of employees	Salaries
1	Vivek	21000
2	Karan	55000
3	James	67000
4	Soham	50000
5	Renu	54000
6	Farah	40000
7	Hetal	30000
8	Mary	70000
9	Ganesh	20000
10	Krish	15000

i. Which R command will Mr. John use to enter these values demonstrate the output.
ii. Now Mr. John wants to add the salaries of 5 new employees in the existing table, which command he will use to join datasets with new values in R. Demonstrate the

Solution:

(i)

```
# Create a data frame with the given records
employee_data <- data.frame(
 sr_number = 1:10,
 name = c("Vivek", "Karan", "James", "Soham", "Renu", "Farah",
"Hetal", "Mary", "Ganesh", "Krish")
)

print("Employee Dataset:")
print(employee_data)

salary <- c(21000, 55000, 67000, 50000, 54000, 40000, 30000,
70000, 20000, 15000)
```

Subscribe: https://youtube.com/@ataglanceofficial

```
employee_data$salary <- salary


# Display the dataset
print("Employee Dataset:")
print(employee_data)
```

Output:

```
[1] "Employee Dataset:"
   sr_number   name salary
1          1  Vivek  21000
2          2  Karan  55000
3          3  James  67000
4          4  Soham  50000
5          5   Renu  54000
6          6  Farah  40000
7          7  Hetal  30000
8          8   Mary  70000
9          9 Ganesh  20000
10        10  Krish  15000
```

(ii)

```
# Create a data frame with the salaries of 5 new employees
new_employees <- data.frame(
  sr_number = 11:15,
  name = c("Amit", "Neha", "Rahul", "Sara", "Rohit"),
  salary = c(60000, 45000, 58000, 52000, 48000)
)

# Join the new salaries with the existing dataset
combined_data <- rbind(employee_data, new_employees)
```

```
# Display the combined dataset
print("Combined Employee Dataset:")
print(combined_data)
```

In this demonstration, the rbind() function is used to combine the existing dataset (employee_data) with the data frame of salaries for 5 new employees (new_employees). The result is a combined dataset (combined_data) with the salaries of all 15 employees.

Output:

```
[1] "Combined Employee Dataset:"
   sr_number   name salary
1          1  Vivek  21000
2          2  Karan  55000
3          3  James  67000
4          4  Soham  50000
5          5   Renu  54000
6          6  Farah  40000
7          7  Hetal  30000
8          8   Mary  70000
9          9 Ganesh  20000
10        10  Krish  15000
11        11   Amit  60000
12        12   Neha  45000
13        13  Rahul  58000
14        14   Sara  52000
15        15  Rohit  48000
```

Subscribe: https://youtube.com/@ataglanceofficial

4.

b) Consider the following data frame given below: [10]

course	id	class	marks
1	11	1	56
2	12	2	75
3	13	1	48
4	14	2	69
5	15	1	84
6	16	2	53

i. Create a subset of course less than 3 by using [ ] brackets and demonstrate the output.

ii. Create a subset where the course column is less than 3 or the class equals to 2 by using subset () function and demonstrate the output.

Solution:

(i)

```
# Creating the data frame with the given information
course_data <- data.frame(
 course = c(1, 2, 3, 4, 5, 6),
 id = c(11, 12, 13, 14, 15, 16),
 class = c(1, 2, 1, 2, 1, 2),
 marks = c(56, 75, 48, 69, 84, 53)
)

# Displaying the data frame
print("Course Data Frame:")
print(course_data)


# Subset using []
subset_course_less_than_3 <- course_data[course_data$course < 3,
]

# Display the subset
print("Subset of Course less than 3 using [] brackets:")
print(subset_course_less_than_3)
```

Subscribe: https://youtube.com/@ataglanceofficial

Output:

```
[1] "Course Data Frame:"
  course id class marks
1      1 11     1    56
2      2 12     2    75
3      3 13     1    48
4      4 14     2    69
5      5 15     1    84
6      6 16     2    53
[1] "Subset of Course less than 3 using [] brackets:"
  course id class marks
1      1 11     1    56
2      2 12     2    75
```

Used square brackets ([]) to select rows where the "course" column is less than 3.

(ii)

```
# Subset using subset()
subset_course_class_condition <- subset(course_data, course < 3
| class == 2)

# Display the subset
print("Subset where course < 3 or class == 2 using subset():")
print(subset_course_class_condition)
```

```
[1] "Subset where course < 3 or class == 2 using subset():"
  course id class marks
1      1 11     1    56
2      2 12     2    75
4      4 14     2    69
6      6 16     2    53
```

Used the subset() function to select rows where either the "course" column is less than 3 or the "class" column equals 2.