Unit -3

1. Create a Data Frame

```
name <- c("John", "Jane", "Doe", "Mary")

age <- c(25, 30, 35, 40)

gender <- c("M", "F", "M", "F")

data_frame <- data.frame(Name = name, Age = age, Gender = gender)

print(data_frame)
```

Output:

```
  Name Age Gender
1  John  25      M
2  Jane  30      F
3   Doe  35      M
4  Mary  40      F
```

2. Create data frame and extract height as vector

```
df <- data.frame(age = c(21, 25, 30), height = c(5.5, 6.0, 5.8))

height_vector <- df$height

print(height_vector)
```

Output:

```
[1] 5.5 6.0 5.8
```

3. Melt function on 'sale' data frame

```
library(reshape2)

sale <- data.frame(region = c("North", "South"), Q1 = c(100, 150), Q2 = c(110, 160), Q3 = c(120, 170),
Q4 = c(130, 180))

melted_sale <- melt(sale, id.vars = "region", variable.name = "quarter", value.name = "sales")

print(melted_sale)
```

Output:

```
  region quarter sales
1  North      Q1   100
2  South      Q1   150
```

```
3  North     Q2   110
4  South     Q2   160
5  North     Q3   120
6  South     Q3   170
7  North     Q4   130
8  South     Q4   180
```

4. Work with airquality dataset

```
data("airquality")
is.data.frame(airquality)
```

Output:

[1] TRUE

```
ordered_airquality <- airquality[order(airquality[,1], airquality[,2]), ]
modified_airquality <- subset(ordered_airquality, select = -c(Solar.R, Wind))
print(modified_airquality)
```

Output: Displays airquality data frame without 'Solar.R' and 'Wind' columns.

5. Create factor from 'women' dataset

```
women_height_factor <- factor(women$height)
print(women_height_factor)
```

Output:

```
 [1] 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
Levels: 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
```

6. EDA on Iris dataset

```
data(iris)
```

(i)

```
dim(iris)
```

Output:

[1] 150   5

```
str(iris)
```

Output: Structure of iris dataset

summary(iris)

Output: Summary statistics for all columns

sapply(iris[, 1:4], sd)

Output:

Sepal.Length Sepal.Width Petal.Length Petal.Width

0.8280661   0.4358663   1.7652982   0.7622377

(ii)

aggregate(. ~ Species, data = iris[, 1:5], mean)

Output: Mean values grouped by species

aggregate(. ~ Species, data = iris[, 1:5], sd)

Output: Standard deviation values grouped by species

(iii)

quantile(iris$Sepal.Width)

Output:

0% 25% 50% 75% 100%

2.0 2.8 3.0 3.3 4.4

quantile(iris$Sepal.Length)

Output:

0% 25% 50% 75% 100%

4.3 5.1 5.8 6.4 7.9

(iv)

iris$Sepal.Length.Cate <- cut(iris$Sepal.Length, quantile(iris$Sepal.Length), include.lowest = TRUE)

print(iris)

Output: Iris dataset with new Sepal.Length.Cate column

(v)

aggregate(. ~ Species + Sepal.Length.Cate, data = iris[, 1:5], mean)

Output: Average numerical values by Species and Sepal.Length.Cate

(vi)

```
aggregate(. ~ Species + Sepal.Length.Cate, data = iris[, 1:5], mean)
```

Output: Same as (v)

(vii)

```
library(dplyr)
iris %>% group_by(Species, Sepal.Length.Cate) %>% summarise(across(where(is.numeric), mean, na.rm = TRUE))
```

Output: Pivot table showing mean values

7. Logistic regression on Iris dataset

```
set.seed(123)
index <- sample(1:nrow(iris), 0.8 * nrow(iris))
train <- iris[index, ]
test <- iris[-index, ]
model <- glm(Species ~ Petal.Length + Petal.Width, data = train, family = "binomial")
predictions <- predict(model, test, type = "response")
confusion_matrix <- table(test$Species, predictions > 0.5)
print(confusion_matrix)
```

Output: Confusion matrix for test data

8. Explore airquality dataset

(i)

```
mean_temp <- sum(airquality$Temp, na.rm = TRUE) / sum(!is.na(airquality$Temp))
print(mean_temp)
```

Output:

```
[1] 77.88235
```

(ii)

head(airquality, 5)

Output: First five rows of airquality dataset

(iii)

subset(airquality, select = -c(Temp, Wind))

Output: Airquality dataset without Temp and Wind columns

(iv)

coldest_day <- airquality[which.min(airquality$Temp),]

print(coldest_day)

Output: Details of the coldest day

(v)

windy_days <- sum(airquality$Wind > 17, na.rm = TRUE)

print(windy_days)

Output:

[1] 21

9. Multi regression model

(a)

data(ChickWeight)

model <- lm(weight ~ Time + Diet, data = ChickWeight)

summary(model)

Output: Summary of the regression model

(b)

predict(model, data.frame(Time = 10, Diet = 1))

Output: Predicted weight

(c)

model$residuals

Output: Residuals from the regression model

10. Titanic dataset visualizations

library(ggplot2)

data(Titanic)

titanic_df <- as.data.frame(Titanic)

(a)

ggplot(titanic_df, aes(x = Class, fill = factor(Survived))) + geom_bar(position = "dodge")

Output: Bar chart of survival based on passenger class

(b)

ggplot(titanic_df, aes(x = Class, fill = factor(Survived))) + geom_bar(position = "dodge") + facet_wrap(~ Sex)

Output: Bar chart modified by gender

(c)

ggplot(titanic_df, aes(x = Age)) + geom_histogram(binwidth = 5)

Output: Histogram of Age distribution

11. Quartile and Box Plot

(a)

data <- c(6, 47, 49, 15, 43, 41, 7, 39, 43, 41, 36)

quartiles <- quantile(data)

print(quartiles)

Output:

0% 25% 50% 75% 100%

6.00 15.00 41.00 43.00 49.00

boxplot(data)

Output: Boxplot of the data

(b)

```r
temps <- c(35, 42, 38, 25, 28, 36, 40)
barplot(temps, names.arg = c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"), col = "blue")
legend("topright", legend = "Temperature", fill = "blue")
```

Output: Bar plot with legend

12. CSV File Analysis

(i)

```r
data <- read.csv("input.csv")
```

Output: Data from input.csv file

(ii)

```r
max_salary <- max(data$Salary)
print(max_salary)
```

Output: Maximum salary value

(iii)

```r
print(data[data$Salary == max_salary, ])
```

Output: Details of person with max salary

(iv)

```r
print(data[data$Department == "IT", ])
```

Output: People working in IT department

(v)

```r
print(data[data$Department == "IT" & data$Salary > 600, ])
```

Output: IT department employees with salary > 600

13. Merging Data Frames

```r
students <- data.frame(StudentID = 1:3, Name = c("Alice", "Bob", "Charlie"))
scores <- data.frame(StudentID = 1:3, MathScore = c(90, 85, 88), ScienceScore = c(92, 80, 87),
EnglishScore = c(88, 89, 90))
```

merged_df <- merge(students, scores, by = "StudentID")

print(merged_df)

Output: Merged data frame of students and scores

14. Titanic Dataset (Repeated)

Code is same as question 10

15. Product Data Frame

product_names <- c("Laptop", "Mouse", "Keyboard")

prices <- c(800, 20, 50)

quantities <- c(10, 50, 30)

product_data <- data.frame(Name = product_names, Price = prices, Quantity = quantities)

print(product_data)

Output:

```
    Name Price Quantity
1  Laptop   800      10
2   Mouse    20      50
3 Keyboard    50      30
```

average_price <- mean(product_data$Price)

print(average_price)

Output:

[1] 290

16. Customer and Purchase Data Merge

customers <- data.frame(CustomerID = 1:3, Name = c("John", "Jane", "Doe"))

purchases <- data.frame(CustomerID = 1:3, Purchase = c("Laptop", "Mouse", "Keyboard"))

merged_customers <- merge(customers, purchases, by = "CustomerID")

print(merged_customers)

Output: Merged data frame of customers and purchases

17. Sales Regression

sales_data <- data.frame(Spends = c(1000, 4000, 5000, 4500, 3000, 4000), Sales = c(9914, 40487, 54324, 50044, 34719, 42551))

model_sales <- lm(Sales ~ Spends, data = sales_data)

summary(model_sales)

Output: Summary of sales regression model

18. Height-Weight Linear Regression

height <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

weight <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

model_hw <- lm(weight ~ height)

predict(model_hw, data.frame(height = 170))

Output:

[1] 78.06

plot(height, weight)

abline(model_hw)

Output: Scatter plot with regression line

19. Salary Statistics

salary_data <- data.frame(Salary = c(400, 500, 600, 700, 800))

mean_salary <- mean(salary_data$Salary)

median_salary <- median(salary_data$Salary)

mode_salary <- as.numeric(names(which.max(table(salary_data$Salary))))

print(c(mean_salary, median_salary, mode_salary))

Output:

[1] 600 600 400

20. Univariate EDA

daily_temps <- c(30, 32, 35, 28, 29, 31, 33, 34, 36, 27, 30, 31, 32, 29, 28, 34, 35, 33, 31, 30, 29, 32, 33, 28, 27, 29, 31, 30, 34, 35)

mean_temp <- mean(daily_temps)

median_temp <- median(daily_temps)

mode_temp <- as.numeric(names(which.max(table(daily_temps)))) range_temp <- range(daily_temps)

iqr_temp <- IQR(daily_temps)

sd_temp <- sd(daily_temps)

print(c(mean_temp, median_temp, mode_temp, range_temp, iqr_temp, sd_temp))

Output:

[1] 31.06667 31.0 29.0 27.0 36.0 5.0 2.637969