Assignment 2 - Data Analysis using R Programming

Group 5

2025-11-26

Data Loading

```
# Load the dataset df <-
read.csv("employee_salary_dataset.csv")
```

Structure and Overview

1. Print the structure of your dataset

```
str(df)
```

```
## 'data.frame':           50 obs. of 9 variables:
## $ EmployeeID           : int 1 2 3 4 5 6 7 8 9 10 ...
## $ Name                 : chr "Employee_1" "Employee_2" "Employee_3" "Employee_4" ...
## $ Department           : chr "Marketing" "Operations" "IT" "Operations" ...
## $ Experience_Years: int 15 7 12 8 15 3 14 17 4 18 ...
## $ Education_Level : chr "Master" "Bachelor" "High School" "PhD" ...
## $ Age                         : int 53 25 51 44 36 50 57 34 53 28 ...
## $ Gender                     : chr "Female" "Female" "Female" "Male" ...
## $ City                          : chr "Delhi" "Bangalore" "Hyderabad" "Delhi" ...
## $ Monthly_Salary : int 111416 95271 69064 95091 132450 65818 70525 44830 42429 31893 ...
```

2. List the variables in your dataset

```
names(df)
```

```
## [1] "EmployeeID"          "Name"              "Department"         "Experience_Years"
## [5] "Education_Level" "Age"                      "Gender"             "City"
## [9] "Monthly_Salary"
```

3. Print the top 15 rows of your dataset

```
head(df, 15)
```

```
##      EmployeeID                 Name Department Experience_Years Education_Level Age
## 1               1 Employee_1 Marketing                         15                  Master 53
## 2               2 Employee_2 Operations                         7                Bachelor 25
## 3             3 Employee_3              IT                         12        High School 51
## 4             4 Employee_4 Operations              8             PhD 44
## 5             5 Employee_5 Operations             15          Master 36
## 6            6 Employee_6       Finance            3      High School 50
## 7            7 Employee_7            IT           14             PhD 57
## 8            8 Employee_8            IT           17             PhD 34
## 9            9 Employee_9            IT            4        Bachelor 53
## 10           10 Employee_10 Operations            18      High School 28
## 11           11 Employee_11 Marketing              8             PhD 43
## 12           12 Employee_12          IT            4          Master 49
## 13           13 Employee_13 Operations             2          Master 23
## 14           14 Employee_14     Finance            6        Bachelor 27
## 15           15 Employee_15 Marketing             10             PhD 49
##        Gender      City Monthly_Salary

## 1 Female         Delhi           111416

## 2 Female Bangalore                95271

## 3 Female Hyderabad                69064

## 4      Male        Delhi            95091

## 5 Female           Delhi          132450

## 6      Male     Mumbai             65818

## 7      Male     Mumbai             70525

## 8 Female Bangalore                44830

## 9       Male Hyderabad              42429

## 10     Male     Mumbai             31893

## 11     Male        Delhi          141381

## 12 Female Hyderabad              104909

## 13 Female Hyderabad                72333

## 14     Male        Delhi            28436

## 15 Female      Mumbai              99290
```

User Defined Function

4. Write a user defined function using any of the variables from the data set.

```
# Function to categorize experience level
categorize_experience <- function(years) {
  if (years < 5) {
    return("Junior")
  } else if (years >= 5 & years <= 10) { return("Mid-Level")
  } else {
    return("Senior")
  }
}

# Apply the function to the first few rows to demonstrate sapply(head(df$Experience_Years), categorize_experience)
```

```
## [1] "Senior"          "Mid-Level" "Senior"        "Mid-Level" "Senior"       "Junior"
```

Data Manipulation and Filtering

5. Use data manipulation techniques and filter rows based on any logical criteria

```
# Filter employees with more than 10 years of experience and are from IT department
filtered_df <- df %>% filter(Experience_Years > 10 & Department
== "IT") head(filtered_df)
```

```
##    EmployeeID                  Name Department Experience_Years Education_Level Age Gender
## 1           3 Employee_3          IT               12      High School 51 Female
## 2           7 Employee_7          IT               14              PhD 57    Male
## 3           8 Employee_8          IT               17              PhD 34 Female
## 4          26 Employee_26         IT               14           Master 24    Male
## 5          31 Employee_31         IT               15         Bachelor 54 Female
## 6          35 Employee_35         IT               13           Master 53    Male
##          City Monthly_Salary
## 1 Hyderabad          69064
## 2      Mumbai          70525
## 3 Bangalore          44830
## 4 Hyderabad          30600
## 5 Hyderabad          70714
## 6 Bangalore         130983
```

Reshaping and Joining

6. Identify the dependent & independent variables and use reshaping techniquesand create a new data frame by joining those variables from your dataset.

```
# Dependent variable: Monthly_Salary
# Independent variables: Experience_Years, Age
# Create two separate dataframes to demonstrate joining
df_salary <- df %>% select(EmployeeID, Monthly_Salary) df_details <- df %>% select(EmployeeID, Experience_Years,
Age) # Join them back together joined_df <- left_join(df_details, df_salary, by = "EmployeeID") head(joined_df)
```

```
##   EmployeeID Experience_Years Age Monthly_Salary
## 1          1               15  53         111416
## 2          2                7  25          95271
## 3          3               12  51          69064
## 4          4                8  44          95091
## 5          5               15  36         132450
## 6          6                3  50          65818
```

Data Cleaning

7. Remove missing values in your dataset.

```
# Check for missing values sum(is.na(df))
```

```
## [1] 0
```

```
# Remove missing values (if any) df_clean <-
na.omit(df)
```

8. Identify and remove duplicated data in your dataset

```
# Check for duplicates sum(duplicated(df_clean))
```

```
## [1] 0
```

```
# Remove duplicates df_clean <- df_clean
%>% distinct()
```

Reordering and Renaming

9. Reorder multiple rows in descending order

```
# Reorder by Monthly_Salary in descending order
df_sorted <- df_clean %>% arrange(desc(Monthly_Salary)) head(df_sorted)
```

```
##   EmployeeID        Name Department Experience_Years Education_Level Age Gender
## 1         38 Employee_38 Operations                9          Master  23   Male
## 2         11 Employee_11  Marketing                8             PhD  43   Male
## 3         34 Employee_34         HR               15        Bachelor  53 Female
## 4            25 Employee_25          HR               8      High School 34 Female
## 5             5 Employee_5 Operations               15                Master 36 Female
## 6            35 Employee_35          IT             13              Master 53      Male
##          City Monthly_Salary

## 1      Mumbai           149123

## 2        Delhi          141381
## 3        Delhi          134616
## 4 Bangalore             132455
## 5        Delhi          132450
## 6 Bangalore             130983
```

10. Rename some of the column names in your dataset

```
# Rename 'Monthly_Salary' to 'Salary' and 'Experience_Years' to 'Experience'
df_renamed <- df_sorted %>% rename(Salary =
    Monthly_Salary,
              Experience = Experience_Years)

names(df_renamed)
```

```
## [1] "EmployeeID"         "Name"               "Department"         "Experience"
## [5] "Education_Level" "Age"                  "Gender"             "City"
## [9] "Salary"
```

New Variables

11. Add new variables in your data frame by using a mathematical function

```
# Add a new variable 'Annual_Salary' (Monthly_Salary * 12)
df_final <- df_renamed %>% mutate(Annual_Salary =
Salary * 12) head(df_final)
```

```
##     EmployeeID                    Name Department Experience Education_Level Age Gender
## 1             38 Employee_38 Operations              9              Master 23      Male
## 2             11 Employee_11 Marketing               8                 PhD 43      Male
## 3              34 Employee_34          HR          15            Bachelor 53 Female
## 4              25 Employee_25          HR           8      High School 34 Female
## 5               5 Employee_5 Operations            15                Master 36 Female
## 6              35 Employee_35          IT          13              Master 53      Male
##          City Salary Annual_Salary

## 1      Mumbai 149123          1789476

## 2      Delhi 141381           1696572

## 3      Delhi 134616           1615392
## 4 Bangalore 132455            1589460
## 5        Delhi 132450         1589400
## 6 Bangalore 130983            1571796
```

Training Set

12. Create a training set using random number generator engine.

```
set.seed(123) # Set seed for reproducibility
sample_index <- sample(1:nrow(df_final), 0.7 * nrow(df_final)) training_set <-
df_final[sample_index, ] testing_set <- df_final[-sample_index, ] dim(training_set)
```

```
## [1] 35 10
```

Summary Statistics

13. Print the summary statistics of your dataset

```
summary(df_final)
```

```
##       EmployeeID        Name             Department           Experience
## Min.     : 1.00     Length:50          Length:50            Min.     : 1.00
## 1st Qu.:13.25         Class :character  Class :character    1st Qu.: 5.25
## Median :25.50         Mode :character   Mode :character      Median :10.00
## Mean     :25.50                                              Mean     : 9.90

## 3rd Qu.:37.75                                                3rd Qu.:14.75

## Max.     :50.00                                              Max.     :19.00

## Education_Level              Age              Gender              City
## Length:50               Min.     :22.00     Length:50            Length:50
## Class :character          1st Qu.:28.25     Class :character     Class :character
## Mode :character           Median :43.50     Mode :character      Mode :character
##                         Mean     :39.76

##                           3rd Qu.:49.00

##                         Max.     :57.00

##           Salary        Annual_Salary

## Min.     : 28420        Min.    : 341040

## 1st Qu.: 59424          1st Qu.: 713088

## Median : 73890          Median : 886686

## Mean     : 82289        Mean    : 987466

## 3rd Qu.:107219          3rd Qu.:1286628

## Max.     :149123        Max.    :1789476
```

14. Use any of the numerical variables from the dataset and perform the following statistical functions

```
# Using 'Salary' variable
salary_mean <- mean(df_final$Salary) salary_median <-
median(df_final$Salary) salary_range <- range(df_final$Salary)

# Calculate Mode
get_mode <- function(v) { uniqv <-
    unique(v)

    uniqv[which.max(tabulate(match(v, uniqv)))]
} salary_mode <- get_mode(df_final$Salary)
cat("Mean Salary:", salary_mean, "\n")
```

```
## Mean Salary: 82288.8
```

```
cat("Median Salary:", salary_median, "\n")
```

```
## Median Salary: 73890.5
```

```
cat("Mode Salary:", salary_mode, "\n")
```

```
## Mode Salary: 149123
```

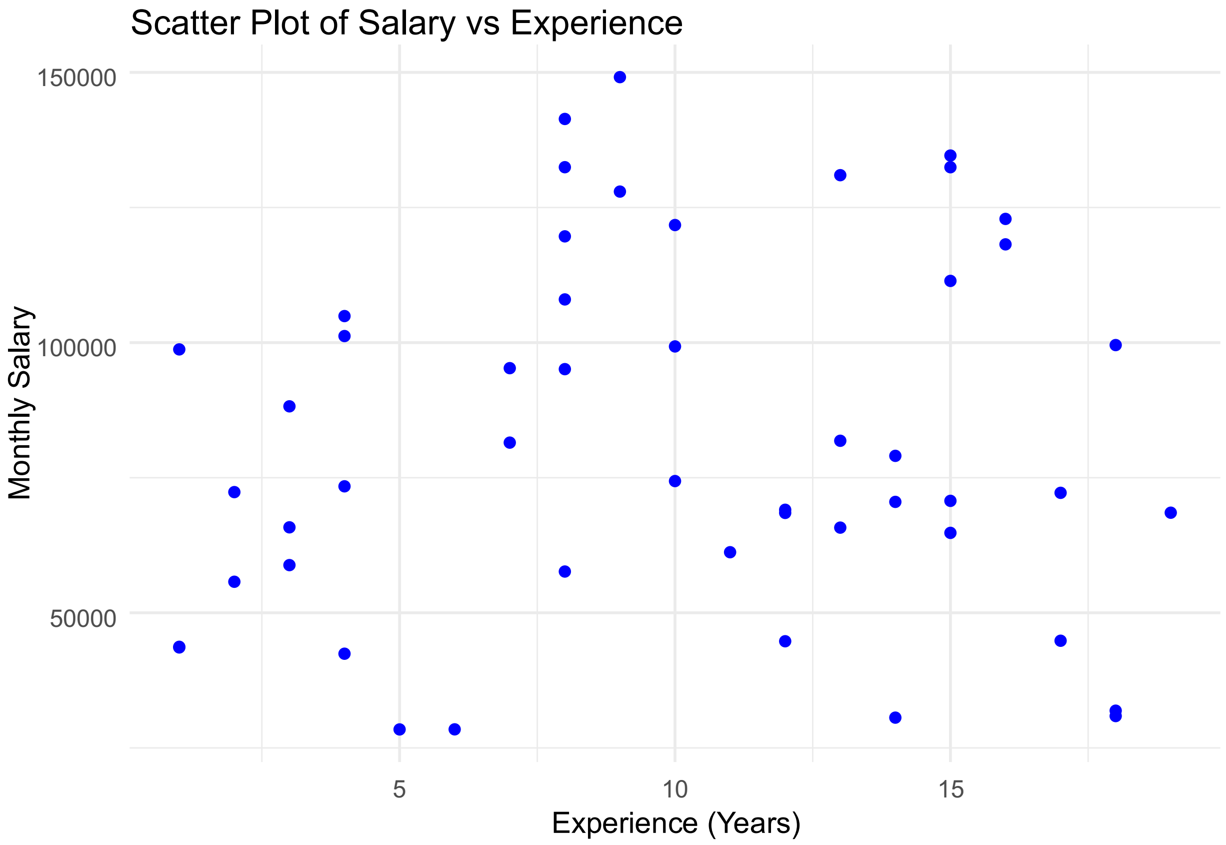
```
cat("Range Salary:", salary_range, "\n")
```

```
## Range Salary: 28420 149123
```

Visualization

15. Plot a scatter plot for any 2 variables in your dataset

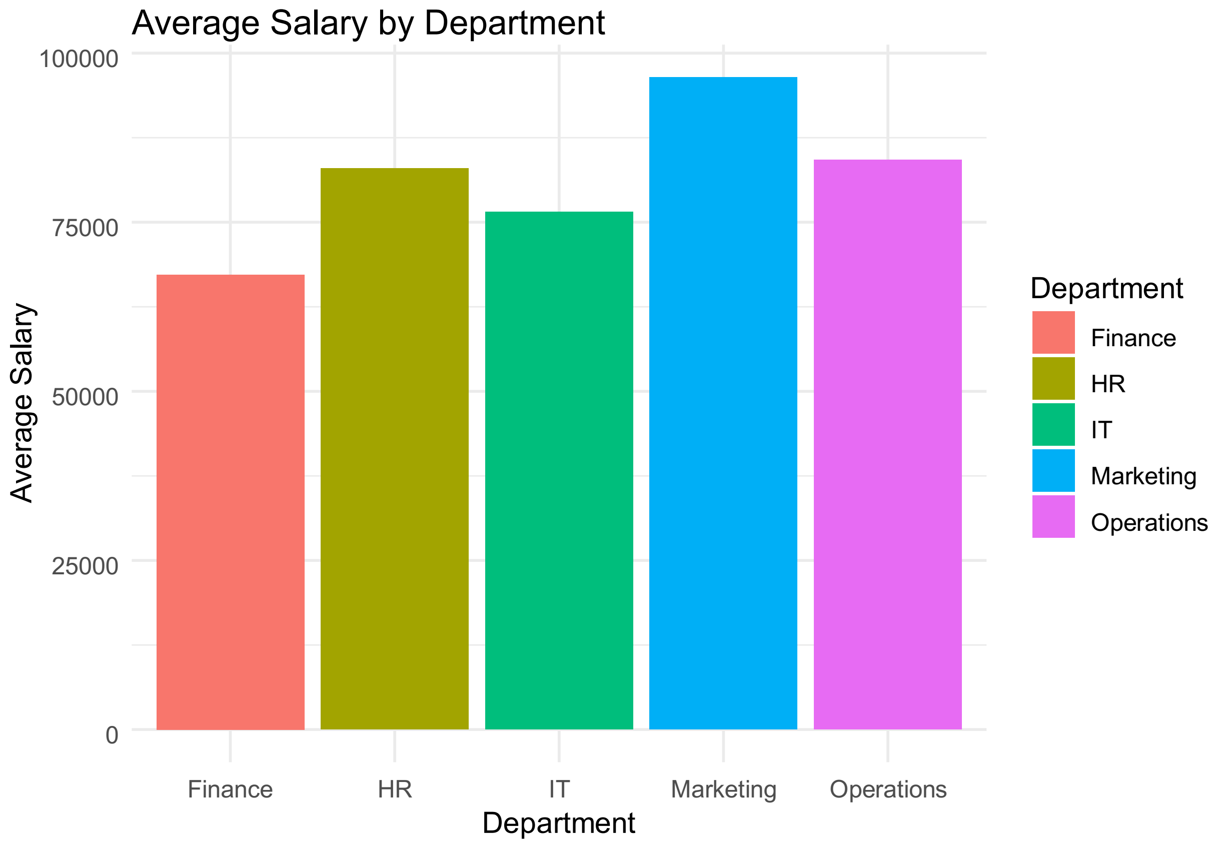
```
ggplot(df_final, aes(x = Experience, y = Salary)) + geom_point(color =
  "blue") + labs(title = "Scatter Plot of Salary vs Experience", x =
  "Experience (Years)", y = "Monthly Salary") +
  theme_minimal()
```



16. Plot a bar plot for any 2 variables in your dataset

```
# Average salary by Department
avg_salary_dept <- df_final %>% group_by(Department)
  %>% summarise(Avg_Salary = mean(Salary))

ggplot(avg_salary_dept, aes(x = Department, y = Avg_Salary, fill = Department)) + geom_bar(stat = "identity") +
  labs(title = "Average Salary by Department", x =
       "Department", y = "Average Salary") +
  theme_minimal()
```



Correlation

17. Find the correlation between any 2 variables by applying Pearson correlation

```
correlation <- cor(df_final$Experience, df_final$Salary, method = "pearson") cat("Pearson correlation between
Experience and Salary:", correlation, "\n")
```

```
## Pearson correlation between Experience and Salary: 0.07422086
```