Practical 5

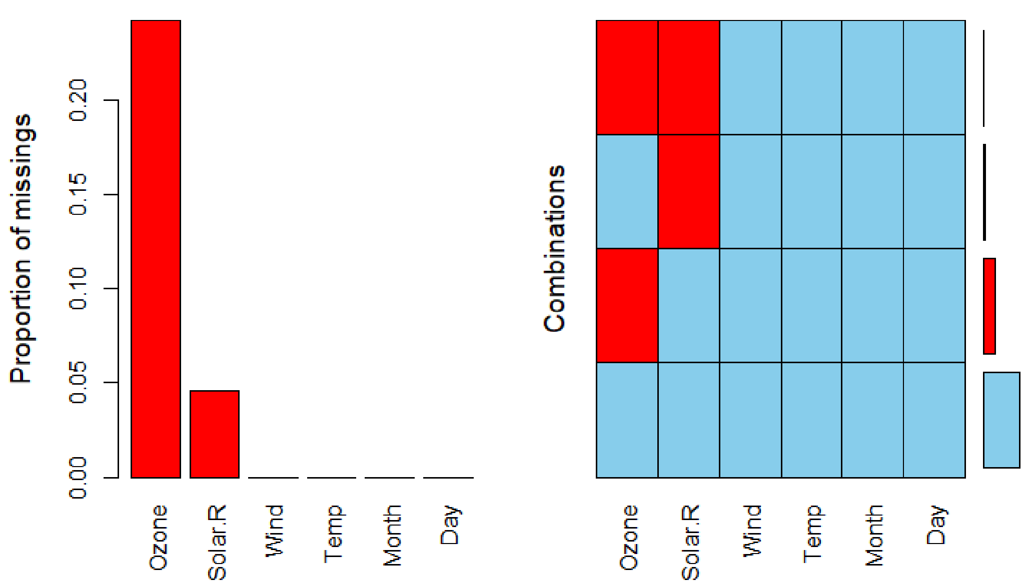
Aim : Implementation of data preprocessing/ Exploratory Data Analysis

CODE :

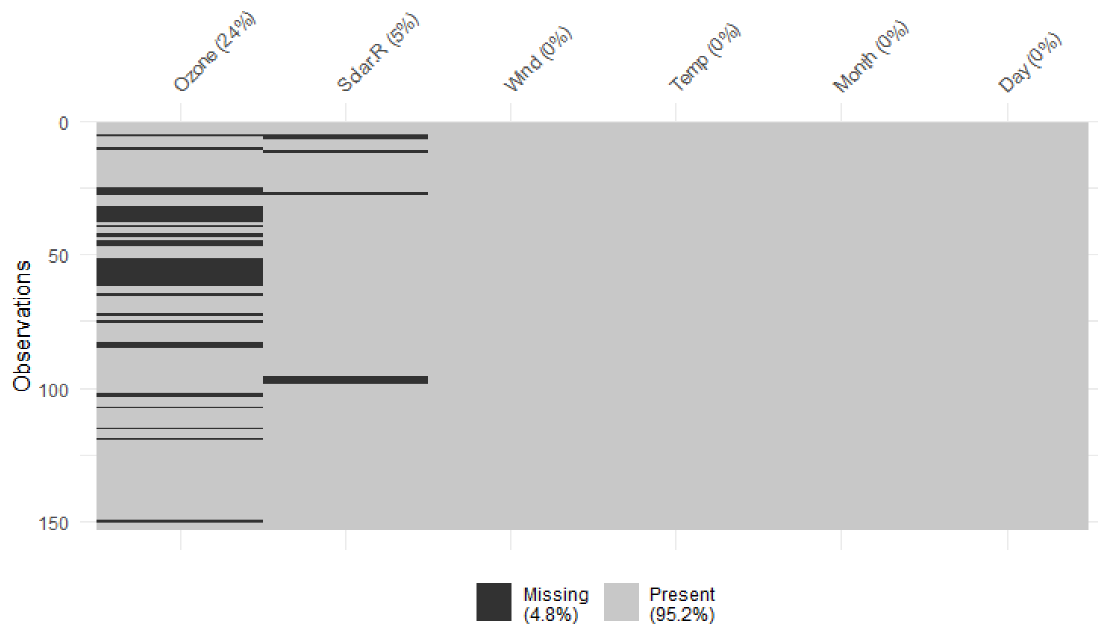
```
# Step 1: Loading the Dataset

data("airquality")  # Load the 'airquality' dataset
df <- as.data.frame(airquality)  # Convert it to a dataframe
View(df)

# Checking the dimensions and structure of the data

dim(df)  # Check the number of rows and columns
str(df)  # View the structure of the dataset
summary(df)  # Summary statistics of the dataset

# Step 2: Missing Values

# Identify missing values
sum(is.na(df))  # Total number of missing values in the dataset
sum(is.na(df$Ozone))  # Missing values in a specific column
```

```
> sum(is.na(df))  # Total number of missing values in the dataset
[1] 44
> sum(is.na(df$Ozone))  # Missing values in a specific column
[1] 37
```

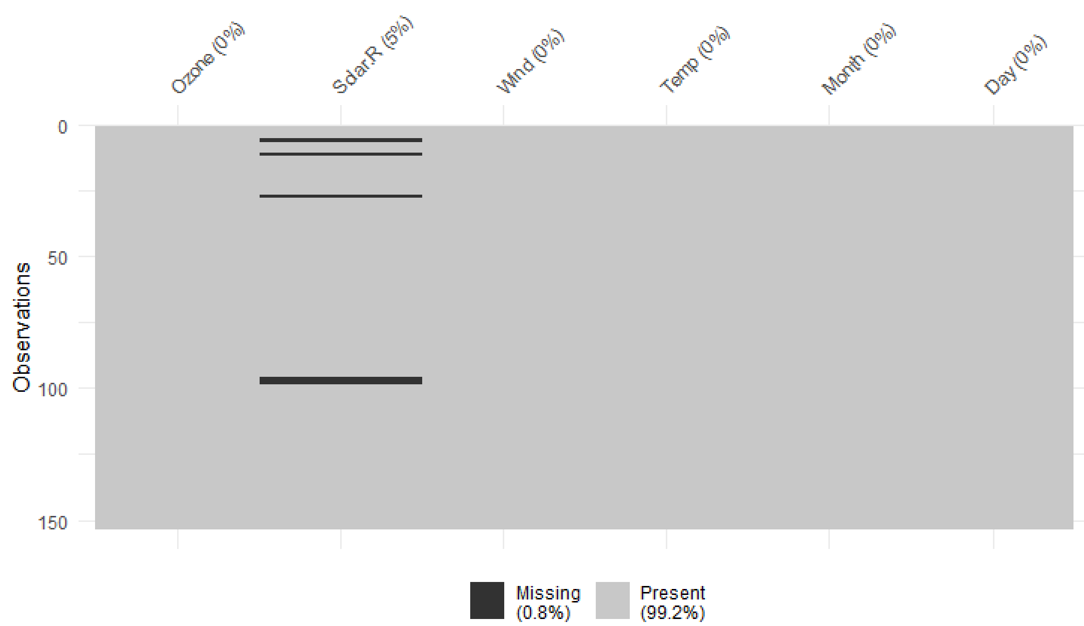
```
# Visualize missing values

library(VIM)
aggr(df)  # Missing value aggregation plot
```



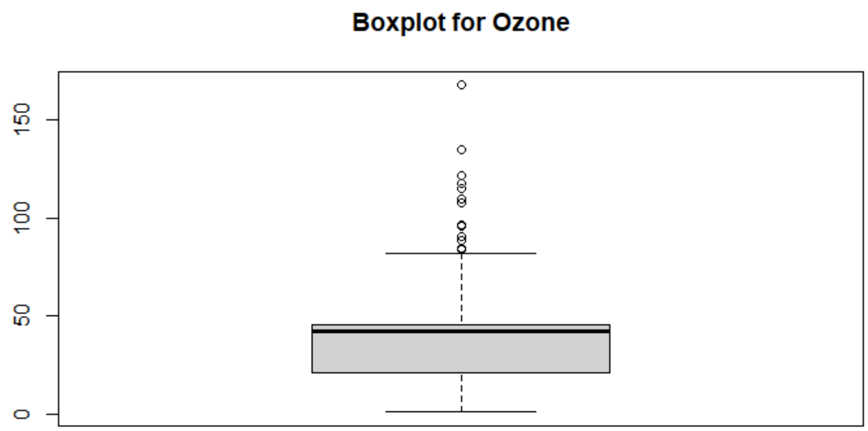
```
library(visdat)
vis_miss(df)  # Visualizing missing values using visdat
```



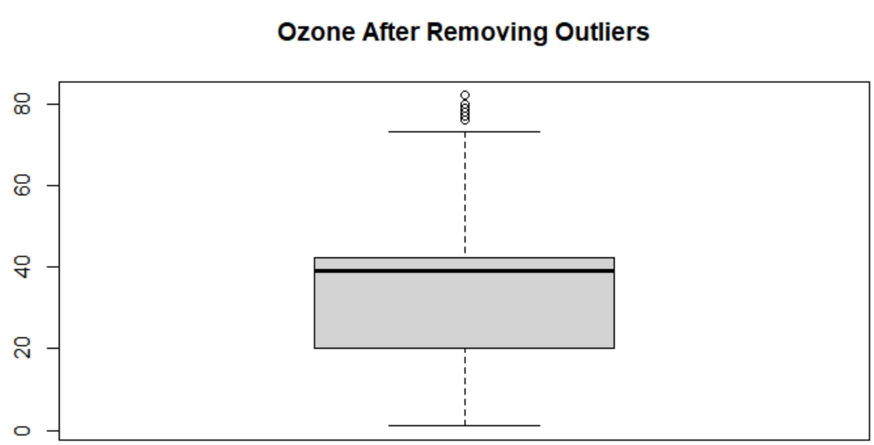

```
# Handling missing values by imputing with mean
mean_ozone <- mean(df$Ozone, na.rm = TRUE)
df$Ozone[is.na(df$Ozone)] <- mean_ozone  # Replace missing values
with mean
```



```
# Step 3: Outlier Detection and Handling

# Boxplot to detect outliers
boxplot(df$Ozone, main = "Boxplot for Ozone")
```




```
# Detecting outliers using the IQR method

Q1 <- quantile(df$Ozone, 0.25)
Q3 <- quantile(df$Ozone, 0.75)
IQR_value <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR_value
upper_bound <- Q3 + 1.5 * IQR_value

# Identifying rows with outliers

outliers <- df$Ozone < lower_bound | df$Ozone > upper_bound
df_with_no_outliers <- df[!outliers, ]  # Removing outliers
boxplot(df_with_no_outliers$Ozone, main = "Ozone After Removing
Outliers")
```



```
# Winsorization for Outlier Handling

library(DescTools)
df$Ozone <- Winsorize(df$Ozone, probs = c(0.05, 0.95))  # Winsorizing data to handle extreme outliers
```

 # Step 4: Feature Encoding

```
# Load 'Salaries' dataset for feature encoding example
View(Salaries)
```

	Age	Gender	Education.Level	Job.Title	Years.of.Experience	Salary
1	32	Male	Bachelor's	Software Engineer	5	90000
2	28	Female	Master's	Data Analyst	3	65000
3	45	Male	PhD	Senior Manager	15	150000
4	36	Female	Bachelor's	Sales Associate	7	60000

```
# Label encoding using factor()
Salaries$Gender <- as.numeric(factor(Salaries$Gender))
```

	Age	Gender	Education.Level	Job.Title	Years.of.Experience	Salary
1	32	3	Bachelor's	Software Engineer	5.0	90000
2	28	2	Master's	Data Analyst	3.0	65000
3	45	3	PhD	Senior Manager	15.0	150000
4	36	2	Bachelor's	Sales Associate	7.0	60000

```
# One-hot encoding
one_hot <- model.matrix(~Salaries$Gender - 1)
Salaries <- cbind(Salaries, one_hot)  # Combine the one-hot encoding with the original data
View(Salaries)
```

 # Step 5: Standardization (Z-score normalization)

```
# Standardize using manual calculation
View(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1

```r
mean_disp <- mean(mtcars$disp)
sd_disp <- sd(mtcars$disp)
mtcars$std_disp <- (mtcars$disp - mean_disp) / sd_disp  # Standardized disp
View(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	std_disp
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4	-0.57061982
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4	-0.57061982
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1	-0.99018209
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1	0.22009369

```r
# Standardization using scale function

mtcars_scaled <- scale(mtcars[, 2:ncol(mtcars)])
View(mtcars_scaled)
```

	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	std_disp
Mazda RX4	-0.1049878	-0.57061982	-0.53509284	0.56751369	-0.610399567	-0.77716515	-0.8680278	1.1899014	0.4235542	0.7352031	-0.57061982
Mazda RX4 Wag	-0.1049878	-0.57061982	-0.53509284	0.56751369	-0.349785269	-0.46378082	-0.8680278	1.1899014	0.4235542	0.7352031	-0.57061982
Datsun 710	-1.2248578	-0.99018209	-0.78304046	0.47399959	-0.917004624	0.42600682	1.1160357	1.1899014	0.4235542	-1.1221521	-0.99018209
Hornet 4 Drive	-0.1049878	0.22009369	-0.53509284	-0.96611753	-0.002299538	0.89048716	1.1160357	-0.8141431	-0.9318192	-1.1221521	0.22009369

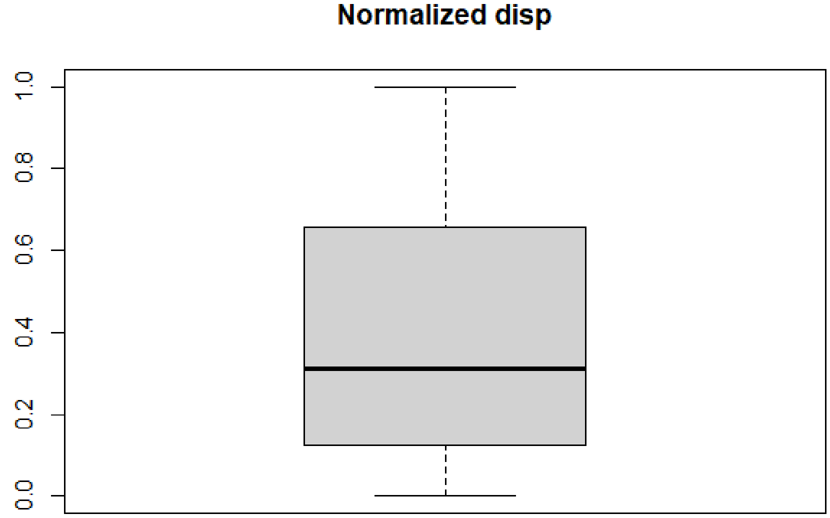
```r
# Step 6: Normalization (Min-Max scaling)

# Define normalization function
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

# Apply normalization on mtcars dataset
mtcars_normalized <- as.data.frame(lapply(mtcars, normalize))
View(mtcars_normalized)
```

Untitled1* mtcars_normalized

Filter

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	std_disp
1	0.4510638	0.5	0.22175106	0.20494700	0.52534562	0.28304781	0.23333333	0	1	0.5	0.4285714	0.22175106
2	0.4510638	0.5	0.22175106	0.20494700	0.52534562	0.34824853	0.30000000	0	1	0.5	0.4285714	0.22175106
3	0.5276596	0.0	0.09204290	0.14487633	0.50230415	0.20634109	0.48928571	1	1	0.5	0.0000000	0.09204290
4	0.4680851	0.5	0.46620105	0.20494700	0.14746544	0.43518282	0.58809524	1	0	0.0	0.0000000	0.46620105

```
# Step 7: Visualizing the final results
# Boxplot visualization after normalization
boxplot(mtcars_normalized$disp, main = "Normalized disp")
```




```
# Visualize density
plot(density(mtcars_normalized$disp), main = "Density plot of
normalized disp")
```