Week 8 - Introduction to Regression Models

Lab Session: Simple and Multiple Linear Regression Using R Studio

Objective:

In this lab session, we will learn how to perform simple and multiple linear regression analyses using R Studio. We will use the built-in datasets to understand the concepts and implement them practically.

Load Required Libraries:

```
# Load necessary libraries
library(MASS)
library(ISLR)
library(ggplot2)  # For data visualization
```

Load Dataset:

For this lab, we'll use the built-in dataset "mtcars," which contains data about various car models.

```
# Load dataset
data(mtcars)
```

Explore Dataset:

Before performing regression, let's understand our dataset.

```
# Display the structure of the dataset
str(mtcars)

# Display the first few rows of the dataset
head(mtcars)
```

Compute Correlations:

We will calculate the correlations between variables in the dataset using the cor() function.

```
# Compute correlations
correlations <- cor(mtcars)

# Print correlations
print(correlations)
```

Visualize Correlations:

Visualizing correlations can provide a clearer understanding of relationships between variables.

Heatmap of correlations

```
library(ggplot2)
ggplot(data = reshape2::melt(correlations), aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 10, hjust = 1)) +
  coord_fixed()
```

Correlation Interpretation:

Correlation values range from -1 to 1, where:

- 1 indicates a perfect positive correlation.
- -1 indicates a perfect negative correlation.
- 0 indicates no correlation.
- Positive values indicate a direct relationship, while negative values indicate an inverse relationship.
- Strong correlations (close to -1 or 1) suggest a strong relationship between variables.

Simple Linear Regression:

Let's start with a simple linear regression between two variables, say, "mpg" (miles per gallon) and "wt" (weight of the car).

```
# Simple linear regression
model_simple <- lm(mpg ~ wt, data = mtcars)

# Summary of the model
summary(model_simple)
```

Visualize Results:

Visualizing the regression line with the actual data helps in understanding the relationship.

```
# Plotting
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Simple Linear Regression", x = "Weight", y = "Miles per Gallon")
```

Multiple Linear Regression:

Now, let's perform multiple linear regression using "mpg" as the response variable and "wt", "hp" (horsepower), and "disp" (displacement) as predictors.

```
# Multiple linear regression
model_multiple <- lm(mpg ~ wt + hp + disp, data = mtcars)

# Summary of the model
summary(model_multiple)
```

Interpretation:

In simple linear regression, the coefficient of "wt" indicates that for every unit increase in weight, the miles per gallon decrease by the coefficient value. In multiple linear regression, coefficients for "wt", "hp", and "disp" indicate the impact of each variable on miles per gallon while holding other variables constant.

Conclusion:

Simple and multiple linear regression are powerful techniques to understand relationships between variables. R provides convenient functions like lm() for regression analysis. Visualization aids in interpreting the results effectively.

Now you:

The Boston dataset in R is a built-in dataset that contains information about housing in the Boston area. It is often used as a standard dataset for regression analysis and predictive modeling tasks. This dataset is commonly used to demonstrate various statistical techniques, especially regression modeling, due to its rich set of variables and relatively small size.

Here's a description of the variables in the Boston dataset:

- crim: Per capita crime rate by town.
- zn: Proportion of residential land zoned for lots over 25,000 sq.ft.
- indus: Proportion of non-retail business acres per town.
- chas: Charles River dummy variable (1 if tract bounds river; 0 otherwise).
- nox: Nitrogen oxide concentration (parts per 10 million).
- rm: Average number of rooms per dwelling.
- age: Proportion of owner-occupied units built prior to 1940.
- dis: Weighted distances to five Boston employment centers.
- rad: Index of accessibility to radial highways.
- tax: Full-value property tax rate per $10,000.
- ptratio: Pupil-teacher ratio by town.
- black: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town.
- lstat: Percentage of lower status of the population.

- medv: Median value of owner-occupied homes in $1000s (the target variable).

This dataset is often used for predicting median house values (medv) based on the other variables. Researchers and data analysts use it to study relationships between various socio-economic factors and housing prices, as well as to demonstrate regression techniques and machine learning algorithms in educational settings.

To access the Boston dataset in R, you can simply call it by name:

```
data(Boston)
```

Questions

- Using simple linear regression, can you predict the median value of owner-occupied homes based on a single predictor variable? Which variable are you going to use? Is the result better if you use multiple linear regression? Which variables give the best results?
- Can you determine which variable has the strongest correlation with median home value in the Boston dataset?

Write your results in an rmarkdown document