Statistical Modelling in R

Dr. Rakhee Chhibber

Introduction

Statistical modelling in R is enabled by Data Scientists to extract meaningful information from data and test hypotheses, ensuring that decision-making is efficient. Certainly, Data Scientists make use of different statistical modelling techniques that help in finding relationships between data.

What is Statistical Modelling?

Statistical modelling can be defined as the method of using different statistical techniques for describing, analysing and making predictions on the relationships within the data. It mainly involves creating representations or models for capturing underlying patterns, structures and associations in data, mathematically.

Process of statistical modelling

- **Problem Definition**
- **Data Collection**
- **Exploratory Data Analysis**
- **Model Selection**
- **Model Building**
- **Parameter Estimation**
- **Model Evaluation**
- **Inference and Interpretation**
- **Communication**

Types of Statistical Models in R

Linear Models

- At the core of statistical modelling, linear models form a cornerstone. They establish relationships between a dependent variable and one or more independent variables, assuming a linear connection. These models offer simplicity, interpretability, and a strong theoretical basis, making them invaluable for understanding data patterns and making predictions.
- **Linear Regression** is employed to predict a continuous numerical outcome based on one or more predictors. Its simplicity and interpretability make it a popular choice.
- **ANOVA (Analysis of Variance)** compares means across different groups which is particularly useful for experimental designs.
- **ANCOVA (Analysis of Covariance)** extends ANOVA by incorporating continuous covariates to account for their influence on the response variable.

Generalised Linear Models (GLMs)

Generalised Linear Models (GLMs) expand the capabilities of linear models by accommodating a wider range of response variable types. Traditional linear regression assumes a normal distribution for the outcome, whereas GLMs can handle response variables that follow different probability distributions.

+ **Logistic Regression** is tailored for predicting binary outcomes, making it invaluable for classification tasks.
+ **Poisson Regression** is suitable for counting data, modelling phenomena like the number of occurrences within a specific time period.

Nonlinear Models

- It represents complex relationships between variables that straight lines cannot adequately capture. These models offer greater flexibility to fit data exhibiting curves, peaks, or other non-linear patterns.
- By accommodating a wider range of functional forms, nonlinear models often provide more accurate and informative insights in comparison to their linear counterparts. We employ Nonlinear Least Squares to fit models with complex, non-linear patterns in the data.

Other Model Classes

- Beyond these fundamental models, R provides tools for a variety of statistical tasks.
- **Time Series Models** can analyse data collected sequentially over time, capturing patterns and trends.
- **Survival Analysis** focuses on predicting the time until an event occurs, such as patient survival or product failure.
- **Clustering** techniques, including K-means and hierarchical clustering, group similar data points together to uncover underlying structures.

Reasons for Learning Statistical Modelling

- Data Analysis and Interpretation
- Informed Decision-Making
- Hypothesis Testing
- Prediction and Forecasting
- Problem Solving
- Scientific Research
- Personalization and Recommendations
- Quality Improvement
- Risk Assessment
- Academic and Career Advancement
- Understanding Correlations

Correlation and regression Analysis

Dr. Rakhee Chhibber

Introduction

+ Correlation and regression analysis are two fundamental statistical techniques used to examine the relationships between variables. R Programming Language is a powerful programming language and environment for statistical computing and graphics, making it an excellent choice for conducting these analyses.

Correlation Analysis

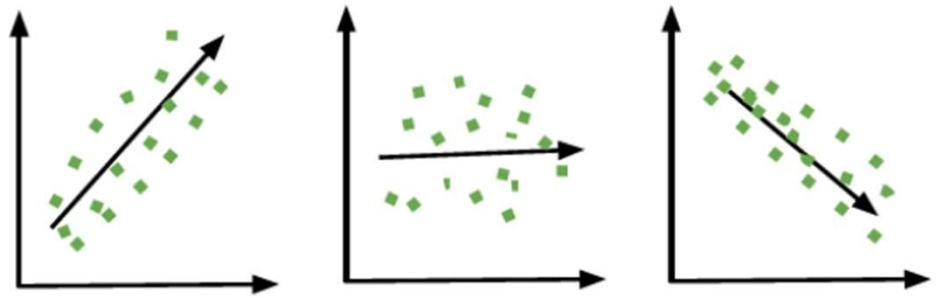
- Correlation analysis is a statistical technique used to measure the strength and direction of the relationship between two continuous variables. The most common measure of correlation is the Pearson correlation coefficient. It quantifies the linear relationship between two variables. The Pearson correlation coefficient, denoted as “r,” :

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

where,

- r: Correlation coefficient
- x_i : i^th value first dataset X
- $\bar{x}$: Mean of first dataset X
- y_i : i^th value second dataset Y
- $\bar{y}$: Mean of second dataset Y

CORRELATION



Positive Correlation

Zero Correlation

Negative Correlation

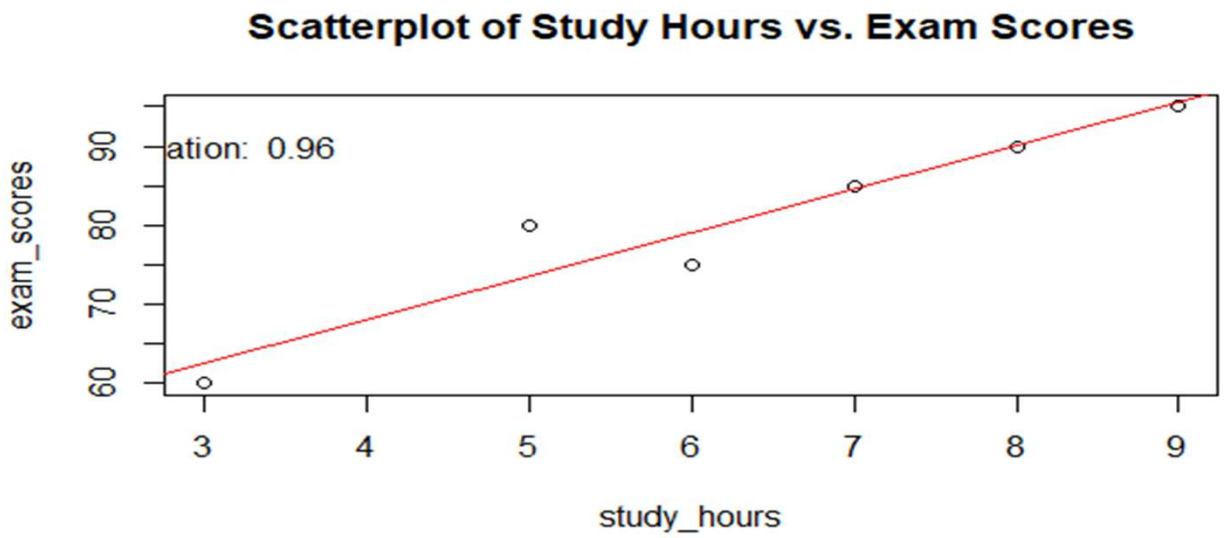
Correaltion

Correlation using R

```
# Sample data
study_hours <- c(5, 7, 3, 8, 6, 9)
exam_scores <- c(80, 85, 60, 90, 75, 95)

# Calculate Pearson correlation
correlation <- cor(study_hours, exam_scores)
correlation
```

Visualize the data and correlation

```
# Visualize the data and correlation
plot(study_hours, exam_scores, main = "Scatterplot of
Study Hours vs. Exam Scores")
# Add regression line
abline(lm(exam_scores ~ study_hours), col = "red")
text(3, 90, paste("Correlation: ", round(correlation, 2)))
```



In the scatterplot, you can see a positive linear trend, which means that as the number of study hours increases, exam scores tend to increase.

Explanation of functions used in code

- **Calculate Pearson Correlation:** cor() function
- **Visualize the Data and Correlation:** plot(study_hours, exam_scores, main = “Scatterplot of Study Hours vs. Exam Scores”)
- **abline(lm(exam_scores ~ study_hours), col = “red”)**
- **text(3, 90, paste(“Correlation: “, round(correlation, 2)))**

Regression Analysis

Introduction

Regression analysis is used to model the relationship between one or more independent variables and a dependent variable. In simple linear regression, there is one independent variable, while in multiple regression, there are multiple independent variables. The goal is to find a linear equation that best fits the data.

There are two types of Regression analysis.

1. Simple Linear Regression
2. Multiple Linear Regression

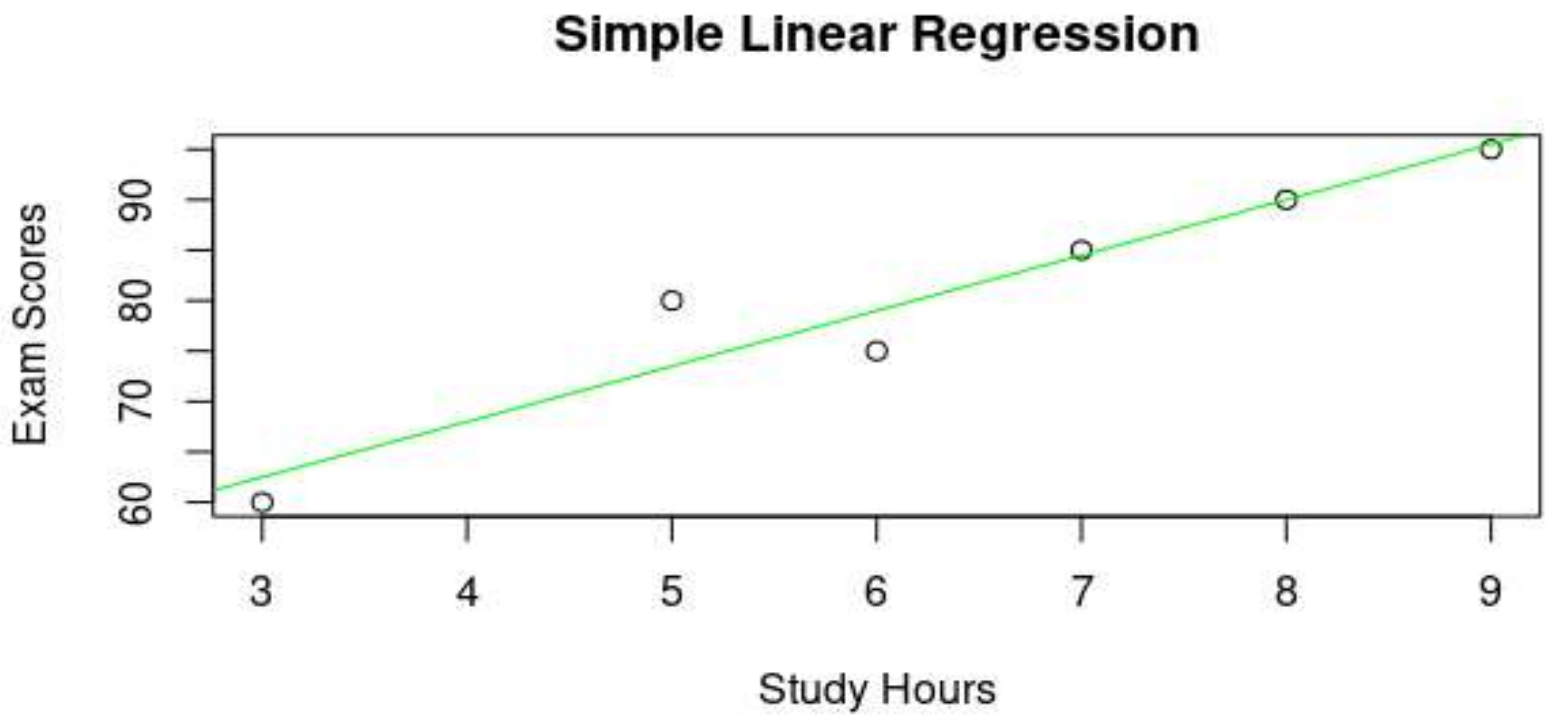
Simple Linear Regression in R

```
# Sample data
study_hours <- c(5, 7, 3, 8, 6, 9)
exam_scores <- c(80, 85, 60, 90, 75, 95)
# Perform simple linear regression
regression_model <- lm(exam_scores ~ study_hours)
# View the summary of the regression results
summary(regression_model)
```

```
Call:
lm(formula = exam_scores ~ study_hours)
Residuals:
        1         2         3         4         5         6
 6.50e+00  5.00e-01 -2.50e+00 -1.11e-15 -4.00e+00 -5.00e-01
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  46.0000     5.5356   8.310  0.00115 **
study_hours   5.5000     0.8345   6.591  0.00275 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 4.031 on 4 degrees of freedom
Multiple R-squared:  0.9157,    Adjusted R-squared:  0.8946
F-statistic: 43.44 on 1 and 4 DF,  p-value: 0.002745
```

Visualize the data and regression line

```
plot(study_hours, exam_scores, main = "Simple Linear
Regression", xlab = "Study Hours", ylab = "Exam Scores")
abline(regression_model, col = "Green")
```


Simple Linear Regression
Exam Scores
60
70
80
90
3
4
5
6
7
8
9
Study Hours

Interpretation

- In this example, a simple linear regression is performed to predict exam scores based on study hours. The lm() function creates a regression model, and summary() provides statistics. The scatterplot visually represents the relationship, with the red line indicating the best-fit linear model. The results in the summary reveal details about the intercept, slope, and model fit. This analysis helps us understand how study hours influence exam scores and provides a quantitative model for prediction.

Multiple Linear Regression Example in R

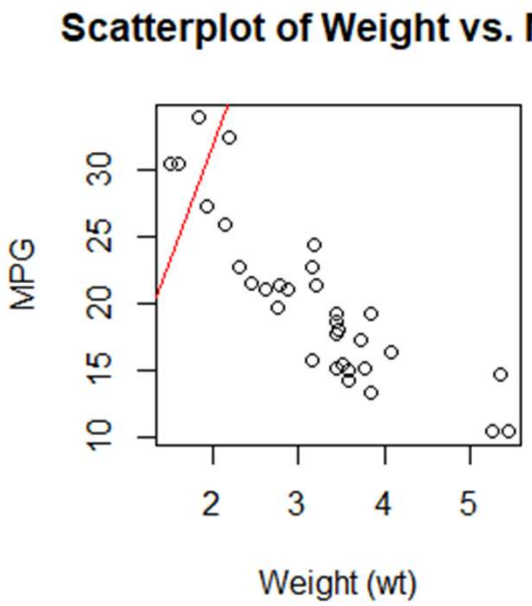
We'll use a dataset that contains information about the price of cars based on various attributes like engine size, horsepower, and the number of cylinders. Our goal is to build a multiple linear regression model to predict car prices based on these attributes. We'll use the mtcars dataset, which is built into R.

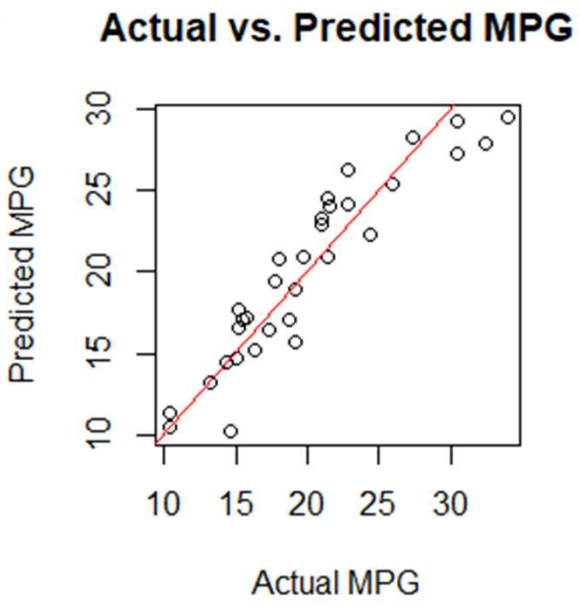
```
# Load the mtcars dataset
data(mtcars)

# Perform multiple linear regression
regression_model <- lm(mpg ~ wt + hp + qsec + am, data
= mtcars)

# View the summary of the regression results
summary(regression_model)
```

```
Call:
lm(formula = mpg ~ wt + hp + qsec + am, data = mtcars)
Residuals:
    Min      1Q  Median      3Q     Max
-3.4975 -1.5902 -0.1122  1.1795  4.5404
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 17.44019    9.31887   1.871  0.07215 .
wt          -3.23810    0.88990  -3.639  0.00114 **
hp          -0.01765    0.01415  -1.247  0.22309
qsec         0.81060    0.43887   1.847  0.07573 .
am           2.92550    1.39715   2.094  0.04579 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 2.435 on 27 degrees of freedom
Multiple R-squared:  0.8579,    Adjusted R-squared:  0.8368
F-statistic: 40.74 on 4 and 27 DF,  p-value: 4.589e-11
```

Visualize the data and regression line

```
# Visualize the data and regression line for one variable (wt) and the
#actual vs.predicted values
# Create a 1x2 grid of plots
par(mfrow = c(1, 2))
# Plot 1: Scatterplot of Weight (wt) vs. MPG
plot(mtcars$wt, mtcars$mpg, main = "Scatterplot of Weight vs. MPG",  xlab = "Weight (wt)", ylab = "MPG")
abline(regression_model$coefficients["wt"], regression_model$coefficients["(Intercept)"], col = "red")

# Plot 2: Actual vs. Predicted MPG
predicted_mpg <- predict(regression_model, newdata = mtcars)
plot(mtcars$mpg, predicted_mpg, main = "Actual vs. Predicted MPG", xlab = "Actual MPG", ylab =
"Predicted MPG")
abline(0, 1, col = "red")
```


Scatterplot of Weight vs. MPG
MPG
10
15
20
25
30
2
3
4
5
Weight (wt)


Actual vs. Predicted MPG
Predicted MPG
10
15
20
25
30
10
15
20
25
30
Actual MPG

Analysis

- We load the mtcars dataset, which contains data on various car attributes, including miles per gallon (mpg), weight (wt), horsepower (hp), quarter mile time (qsec), and transmission type (am).
- We perform a multiple linear regression using the lm() function. In this example, we predict mpg based on weight (wt), horsepower (hp), quarter mile time (qsec), and transmission type (am) as independent variables.
- We view the summary of the regression results to analyze the model coefficients, including the intercept and coefficients for each independent variable. This summary provides information about the strength and significance of each variable in predicting mpg.

Two plots:

- Plot 1: A scatterplot of weight (wt) vs. MPG, along with the regression line. This shows the relationship between weight and MPG. The red line represents the linear relationship found by the regression model.
- Plot 2: A scatterplot of actual MPG vs. predicted MPG. This plot helps us assess how well our model's predictions align with the actual data. The red line represents a perfect fit (actual equals predicted).

Difference between Correlation and Regression Analysis

Correlation Analysis	Regression Analysis
It is used to measure and quantify the strength and direction of the association between two or more variables.	Regression is used for prediction and understanding the causal relationships between variables.
The primary output is a correlation coefficient that quantifies the strength and direction of the relationship between variables.	The output includes regression coefficients, which provide information about the intercept and the slopes of the independent variables
It is often used when you want to understand the degree of association between variables and explore patterns in data.	It is employed when you want to make predictions, understand how one variable affects another, and control for the influence of other variables.