 Sardar Patel Institute of Technology,Mumbai

Department of Electronics and Telecommunication Engineering

T.E. Sem-V (2018-2019)

ETL54-Statistical Computational Laboratory

**Lab-3: Regression Analysis and Modeling**

**Name:** Yash Patil **Roll No.** 2016120036

**Objective:**To carry out linear regression (including multiple regression) and build a regression model

**Outcomes:**

To carry out linear regression (including multiple regression)

To build a regression model using both forward and backward step wise processes

To plot regression models

To add lines of best-fit to regression plots

**System Requirements:** Ubuntu OS with R and RStudio installed

## **Introduction to Linear Regression**

**Regression analysis** is a statistical tool to determine relationships between different types of variables. Variables that remain unaffected by changes made in other variables are known as *independent variables*, also known as a *predictor* or *explanatory variables* while those that are affected are known as *dependent variables* also known as the *response variable*.

Linear regression is a statistical procedure which is used to predict the value of a response variable, on the basis of one or more predictor variables.

**There are two types of linear regressions in R:**

**Simple Linear Regression –** Value of response variable depends on a single explanatory variable.

**Multiple Linear Regression –** Value of response variable depends on more than 1 explanatory variables.

Some common examples of linear regression are calculating GDP, CAPM, oil and gas prices, medical diagnosis, capital asset pricing etc.

### **Simple Linear Regression in R**

R Simple linear regression enables us to find a relationship between a continuous dependent variable Y and a continuous independent variable X. It is assumed that values of X are controlled and not subject to measurement error and corresponding values of Y are observed.

The **general simple linear regression** **model** to evaluate the value of Y for a value of X:

*yi* = β0 + β1*x* + ε

Here, the ith data point, yi, is determined by the variable xi;

β0 and β1 are regression coefficients;

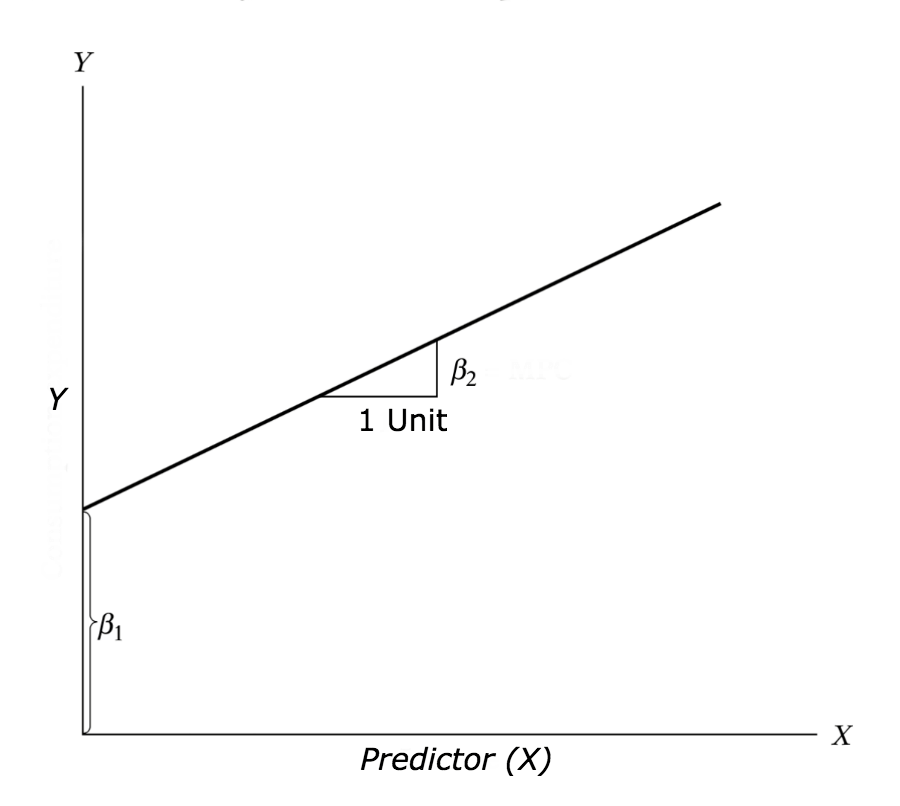
εi is the error in the measurement of the ith value of x.

Regression analysis is implemented to do the following:

Establish a relationship between independent (x) and dependent (y) variables.

Predict the value of y based on a set of values of x1, x2…xn.

Identify independent variables to understand which of them are important to explain the dependent variable, and thereby establishing a more precise and accurate causal relationship between the variables.



### **Multiple Linear Regression in R**

In the real world, you may find situations where you have to deal with more than 1 predictor variable to evaluate the value of response variable. In this case, simple linear models cannot be used and you need to use R multiple linear regressions to perform such analysis with multiple predictor variables.

**R multiple linear regression models** with two explanatory variables can be given as:

*yi* = β0 + β1*x1i* + β2*x1i* + εi

Here, the ith data point, yi, is determined by the levels of the two continuous explanatory variables x1i and x1i’ by the three parameters β0, β1, and β2 of the model, and by the residual ε1 of point i from the fitted surface.

General Multiple regression models can be represented as:

*yi* = Σβ1*x1i* + εi

**Procedure:**

Step-1: Open R Studio and go to R console (>)

>sessionInfo()

>install.packages("DAAG")

>library(lattice)

>library(DAAG)

>?cars # built-in data set in car

## **Example Problem**

For this analysis, we will use the *cars* dataset that comes with R by default. cars is a standard built-in dataset, that makes it convenient to demonstrate linear regression in a simple and easy to understand fashion. You can access this dataset simply by typing in cars in your R console. You will find that it consists of 50 observations(rows) and 2 variables (columns) – dist and speed. Lets print out the first six observations here..

***head(cars) # display the first 6 observations#>***

> head(cars) # display the first 6 observations

speed dist

1 4 2

2 4 10

3 7 4

4 7 22

5 8 16

6 9 10

Before we begin building the regression model, it is a good practice to analyze and understand the variables. The graphical analysis and correlation study below will help with this.

## **Graphical Analysis**

The aim of this exercise is to build a simple regression model that we can use to predict Distance (dist) by establishing a statistically significant linear relationship with Speed (speed). But before jumping in to the syntax, lets try to understand these variables graphically. Typically, for each of the independent variables (predictors), the following plots are drawn to visualize the following behavior:

**Scatter plot**: Visualize the linear relationship between the predictor and response

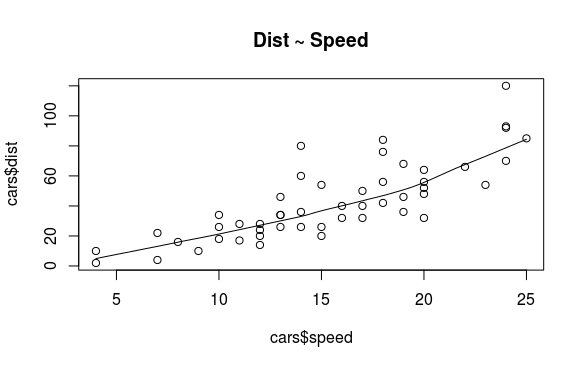
**Box plot**: To spot any outlier observations in the variable. Having outliers in your predictor can drastically affect the predictions as they can easily affect the direction/slope of the line of best fit.

**Density plot**: To see the distribution of the predictor variable. Ideally, a close to normal distribution (a bell shaped curve), without being skewed to the left or right is preferred. Let us see how to make each one of them.

### **Scatter Plot**

Scatter plots can help visualize any linear relationships between the dependent (response) variable and independent (predictor) variables. Ideally, if you are having multiple predictor variables, a scatter plot is drawn for each one of them against the response, along with the line of best as seen below.

***scatter.smooth(x=cars$speed, y=cars$dist, main="Dist ~ Speed") # scatterplot***



## **Correlation**

Correlation is a statistical measure that suggests the level of linear dependence between two variables, that occur in pair – just like what we have here in speed and dist. Correlation can take values between -1 to +1. If we observe for every instance where speed increases, the distance also increases along with it, then there is a high positive correlation between them and therefore the correlation between them will be closer to 1. The opposite is true for an inverse relationship, in which case, the correlation between the variables will be close to -1.

A value closer to 0 suggests a weak relationship between the variables. A low correlation (-0.2 < x < 0.2) probably suggests that much of variation of the response variable (*Y*) is unexplained by the predictor (*X*), in which case, we should probably look for better explanatory variables.

***cor(cars$speed, cars$dist) # calculate correlation between speed and*** distance #> ***[1] 0.8068949***

## **To Build Linear Model**

Refer the following online regression tutorial and perform all the steps and interpret.

<http://r-statistics.co/Linear-Regression.html>

&

2. Read the PPT shared on Google Classroom

Important Points to remember:

Understanding lm() function

Linear Regression Diagnostics using summary() function

Statistical Significance: The p-Value: Null and Alternate Hypothesis

To calculate the t Statistic and p-Values

To calculate AIC and BIC

To know if the model is best fit for your data:

The most common metrics to look at while selecting the model are:

|  |  |
| --- | --- |
| **STATISTIC** | **CRITERION** |
| R-Squared | Higher the better *(> 0.70)* |
| Adj R-Squared | Higher the better |
| F-Statistic | Higher the better |
| Std. Error | Closer to zero the better |
| t-statistic | Should be greater 1.96 for p-value to be less than 0.05 |
| AIC | Lower the better |
| BIC | Lower the better |
| Mallows cp | Should be close to the number of predictors in model |
| MAPE (Mean absolute percentage error) | Lower the better |
| MSE (Mean squared error) | Lower the better |
| Min\_Max Accuracy => mean(min(actual, predicted)/max(actual, predicted)) | Higher the better |

**Predicting Linear Models:**

#### **Step 1:** Create the training (development) and test (validation) data samples from original data.

# Create Training and Test data -  
set.seed(100) # setting seed to reproduce results of random sampling  
trainingRowIndex <- sample(1:nrow(cars), 0.8\*nrow(cars)) # row indices for training data  
trainingData <- cars[trainingRowIndex, ] # model training data  
testData <- cars[-trainingRowIndex, ] # test data

#### **Step 2:** Develop the model on the training data and use it to predict the distance on test data

# Build the model on training data -  
lmMod <- lm(dist ~ speed, data=trainingData) # build the model  
distPred <- predict(lmMod, testData) # predict distance

#### **Step 3:** Review diagnostic measures.

> summary (lmMod)

Call:

lm(formula = dist ~ speed, data = trainingData)

Residuals:

Min 1Q Median 3Q Max

-23.350 -10.771 -2.137 9.255 42.231

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -22.657 7.999 -2.833 0.00735 \*\*

speed 4.316 0.487 8.863 8.73e-11 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.84 on 38 degrees of freedom

Multiple R-squared: 0.674, Adjusted R-squared: 0.6654

F-statistic: 78.56 on 1 and 38 DF, p-value: 8.734e-11

#### **Step 4:** Calculate prediction accuracy and error rates

> actuals\_preds <- data.frame(cbind(actuals=testData$dist, predicteds=distPred))

> correlation\_accuracy <- cor(actuals\_preds)

> head(actuals\_preds)

actuals predicteds

1 2 -5.392776

4 22 7.555787

8 26 20.504349

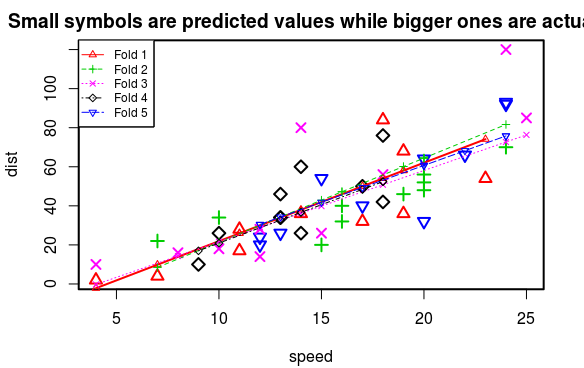
20 26 37.769100

26 54 42.085287

31 50 50.717663

## 8. Cross validation: k- Fold Cross validation

> cvResults <- suppressWarnings(CVlm(data=cars, form.lm=dist ~ speed, m=5, dots=FALSE, seed=29, legend.pos="topleft", printit=FALSE, main="Small symbols are predicted values while bigger ones are actuals."))



> attr(cvResults, 'ms')

[1] 251.2783

## **Describe the following with respect to Linear Regression and Building linear model and Prediction**

List types of regression

● Linear Regression

● Logistic Regression

● Polynomial Regression

● Stepwise Regression

● Ridge Regression

● Lasso Regression

● Elastic Net Regression

What is statistical significance test?

Once sample data has been gathered through an observational study or experiment,

statistical inference allows analysts to assess evidence in favor or some claim about the

population from which the sample has been drawn. The methods of inference used to

support or reject claims based on sample data are known as tests of significance.

Every test of significance begins with a null hypothesis H 0 . H 0 represents a theory that has been put forward, either because it is believed to be true or because it is to be used as a basis for argument, but has not been proved. For example, in a clinical trial of a new drug, the null hypothesis might be that the new drug is no better, on average, than

the current drug. We would write H 0 : there is no difference between the two drugs on

average.

The significance level for a given hypothesis test is a value for which a P-value

less than or equal to is considered statistically significant. Typical values for are 0.1,

0.05, and 0.01. These values correspond to the probability of observing such an extreme value by chance.

How to know if the model is best fit for your data?

If the probability of error is much less than the threshold value (0.05), the considered model is more significant as the best fit model. In the performed experiment, the summary of the linear model lmMod shows that the probability is around 10^(-11). Hence, the generated model is the best fit.

How to test model’s performance?

The amount of deviation of data from the regression line is the determining factor of a model’s performance. If this value is less, it means that amount of error is low and hence, performance is high.

**Conclusion:** (Write in own words)

Linear Regression is a technique of predictive analysis in which machine is trained using a given data set so as to obtain a prediction of other data.

**Note:**Complete your write-up with conclusion and upload your outputs on Google classroom.