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:Manish Dsilva **Roll No.15**

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

Outcomes:

1. To carry out linear regression (including multiple regression)
2. To build a regression model using both forward and backward step wise processes
3. To plot regression models
4. 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:

$y_i = \beta_0 + \beta_1 x + \varepsilon$

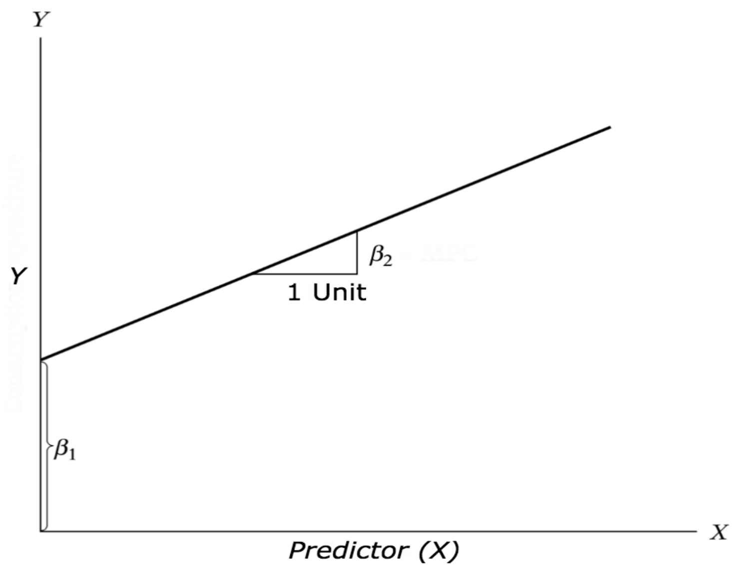
Here, the i^{th} data point, y_i, is determined by the variable x_i;

β_0 and β_1 are regression coefficients;

ε_i is the error in the measurement of the i^{th} 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:

$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{1i} + \varepsilon_i$

Here, the i^{th} data point, y_i, is determined by the levels of the two continuous explanatory variables x_{1i} and x_{1i}' by the three parameters β_0, β_1, and β_2 of the model, and by the residual ε_1 of point i from the fitted surface.

General Multiple regression models can be represented as:

$y_i = \Sigma\beta_1 x_{1i} + \varepsilon_i$

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#>

	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

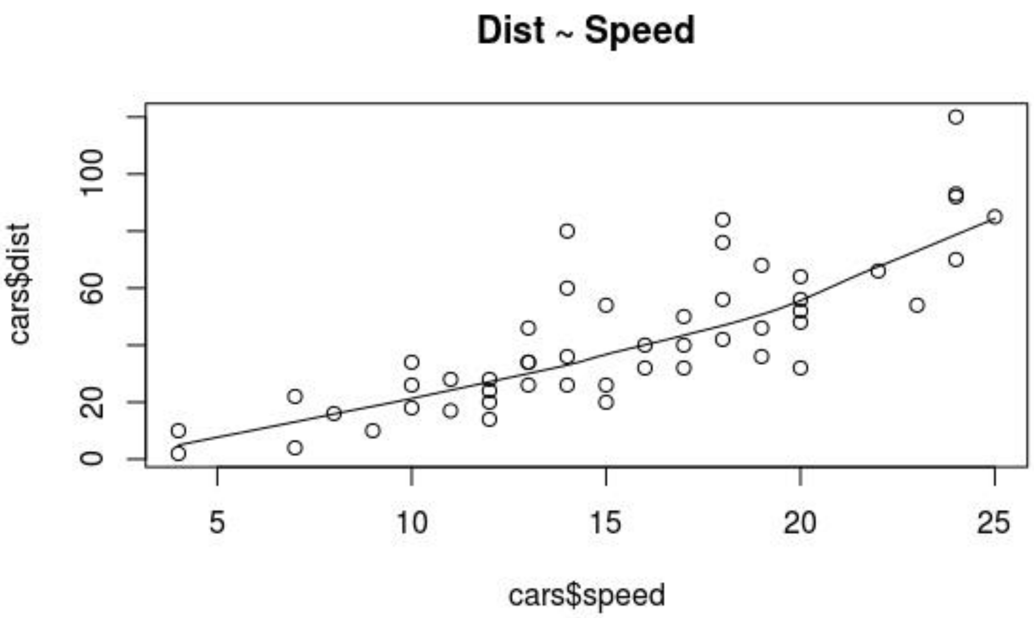
The aim of this exercise is to build a simple regression model that we can use to predict Distance (dist) by establishing a statistically significant linear relationship with Speed (speed). But before jumping in to the syntax, lets try to understand these variables graphically. Typically, for each of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

1. **Scatter plot**: Visualize the linear relationship between the predictor and response
2. **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.
3. **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.

1. http://r-statistics.co/Linear-Regression.html

&

2. Read the PPT shared on Google Classroom

Important Points to remember:

1. Understanding lm() function
2. Linear Regression Diagnostics using summary() function
3. Statistical Significance: The p-Value: Null and Alternate Hypothesis
4. To calculate the t Statistic and p-Values
5. To calculate AIC and BIC
6. 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

7. 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.

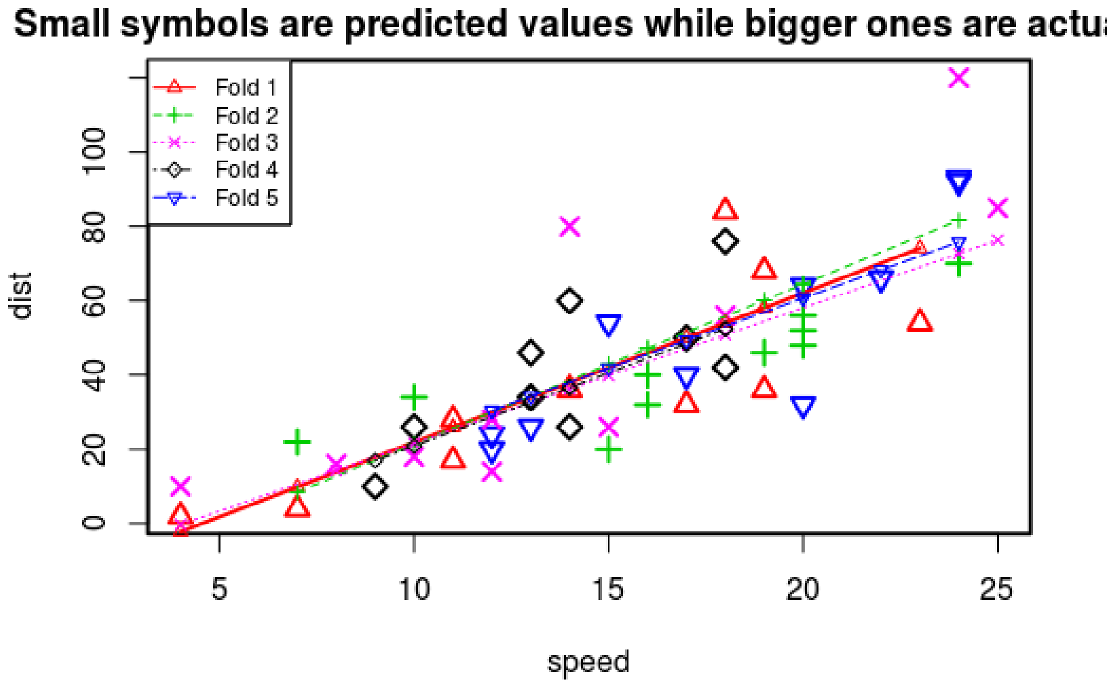
```
> summary (lmMod)

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."))



Build a regression model using the forward stepwise procedure.

1. Look at the mtcars data item. This is built into R.

> ***str(mtcars)***

2. Start by creating a blank model using mpg as the response variable:

> ***mtcars.lm = lm(mpg ~ 1, data = mtcars)***

3. Determine which predictor variable is the best starting candidate:

> ***add1(mtcars.lm, mtcars, test = 'F')***

4. Add the best predictor variable to the blank model:

> ***mtcars.lm = lm(mpg ~ wt, data = mtcars)***

5.Do a quick check of the model summary:

> ***summary(mtcars.lm)***

6.Now look again at the remaining candidate predictor variables:

> ***add1(mtcars.lm, mtcars, test = 'F')***

7.Add the next best predictor variable to your regression model:

> **mtcars.lm = lm(mpg ~ wt + cyl, data = mtcars)**

8.Now check the model summary once more:

> ***summary(mtcars.lm)***

9.Check the remaining variables to see if there are any other candidate predictors to add:

> ***add1(mtcars.lm, mtcars, test = 'F')***

10. The current model remains the most adequate.

Output:

```
'data.frame':	32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
Single term additions

Model:
mpg ~ 1
       Df Sum of Sq     RSS     AIC F value    Pr(>F)
<none>              1126.05 115.943
cyl     1    817.71  308.33  76.494 79.5610 6.113e-10 ***
disp    1    808.89  317.16  77.397 76.5127 9.380e-10 ***
hp      1    678.37  447.67  88.427 45.4598 1.788e-07 ***
drat    1    522.48  603.57  97.988 25.9696 1.776e-05 ***
wt      1    847.73  278.32  73.217 91.3753 1.294e-10 ***
qsec    1    197.39  928.66 111.776  6.3767  0.017082 *
vs      1    496.53  629.52  99.335 23.6622 3.416e-05 ***
am      1    405.15  720.90 103.672 16.8603  0.000285 ***
gear    1    259.75  866.30 109.552  8.9951  0.005401 **
carb    1    341.78  784.27 106.369 13.0736  0.001084 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Call:
lm(formula = mpg ~ wt, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-4.5432 -2.3647 -0.1252  1.4096  6.8727

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
```

```
wt          -5.3445     0.5591  -9.559 1.29e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared:  0.7528,	Adjusted R-squared:  0.7446
F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10

Single term additions

Model:
mpg ~ wt
       Df Sum of Sq    RSS    AIC F value   Pr(>F)
<none>              278.32 73.217
cyl     1    87.150 191.17 63.198 13.2203 0.001064 **
disp    1    31.639 246.68 71.356  3.7195 0.063620 .
hp      1    83.274 195.05 63.840 12.3813 0.001451 **
drat    1     9.081 269.24 74.156  0.9781 0.330854
qsec    1    82.858 195.46 63.908 12.2933 0.001500 **
vs      1    54.228 224.09 68.283  7.0177 0.012926 *
am      1     0.002 278.32 75.217  0.0002 0.987915
gear    1     1.137 277.19 75.086  0.1189 0.732668
carb    1    44.602 233.72 69.628  5.5343 0.025646 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Call:
lm(formula = mpg ~ wt + cyl, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2893 -1.5512 -0.4684  1.5743  6.1004

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  39.6863     1.7150  23.141  < 2e-16 ***
wt           -3.1910     0.7569  -4.216 0.000222 ***
cyl          -1.5078     0.4147  -3.636 0.001064 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.568 on 29 degrees of freedom
Multiple R-squared:  0.8302,	Adjusted R-squared:  0.8185
F-statistic: 70.91 on 2 and 29 DF,  p-value: 6.809e-12
```

```
Single term additions

Model:
mpg ~ wt + cyl
       Df Sum of Sq    RSS    AIC F value Pr(>F)
<none>              191.17 63.198
disp    1    2.6796 188.49 64.746  0.3980 0.5332
hp      1   14.5514 176.62 62.665  2.3069 0.1400
drat    1    0.0010 191.17 65.198  0.0001 0.9903
qsec    1   10.5674 180.60 63.378  1.6383 0.2111
vs      1    0.7059 190.47 65.080  0.1038 0.7497
am      1    0.1249 191.05 65.177  0.0183 0.8933
gear    1    3.0281 188.14 64.687  0.4507 0.5075
carb    1   13.7724 177.40 62.805  2.1738 0.1515
```

Comments on Result:

We have built a Linear Model on dataset of Distance and Speed of Dataset ‘Cars’. It makes a hypothesis and fits a best fit straight line on the given data. We now predict the dependent variable with the help of independent variable and the hypothesis formulated.

To Evaluate Performance of our trained model.

> t_value

[1] 9.46399

> p_value

[1] 1.489836e-12

> f

  value   numdf   dendf

89.56711 1.00000 48.00000

AIC(linearMod) # AIC => 419.1569

[1] 419.1569

> BIC(linearMod) # BIC => 424.8929

[1] 424.8929

correlation_accuracy

           actuals predicteds

actuals   1.0000000 0.8277535

predicteds 0.8277535 1.0000000

min_max_accuracy
[1] 0.3800489

> mape

[1] 0.6995032

Part-II: Logistic Regression

We use the logistic regression equation to predict the probability of a dependent variable taking the dichotomy values 0 or 1. Suppose x1, x2, ..., xp are the independent variables, α and βk (k = 1, 2, ..., p) are the parameters, and E(y) is the expected value of the dependent variable y, then the logistic regression equation is:

$$\boxed{E(y) = 1/(1 + e^{-(\alpha + \sum_k \beta_k x_k)})}$$

For example, in the built-in data set ***mtcars,*** the data column am represents the transmission type of the automobile model (0 = automatic, 1 = manual).
With the logistic regression equation, we can model the probability of a manual transmission in a vehicle based on its engine horsepower and weight data.

$$P(Manual\ Transmission) = 1/(1 + e^{-(\alpha + \beta_1 * Horsepower + \beta_2 * Weight)})$$

Estimated Logistic Regression Equation
Using the generalized linear model, an estimated logistic regression equation can be formulated as below. The coefficients a and bk (k = 1, 2, ..., p) are determined according to a maximum likelihood approach, and it allows us to estimate the probability of the dependent variable y taking on the value 1 for given values of xk (k = 1, 2, ..., p).

$$Estimate\ of\ P(y = 1 \mid x_1, ...x_p) = 1/(1 + e^{-(a + \sum_k b_k x_k)})$$

Example Problem:
By use of the logistic regression equation of vehicle transmission in the data set ***mtcars***, estimate the probability of a vehicle being fitted with a manual transmission if it has a 120hp engine and weights 2800 lbs.

Solution:
We apply the function ***glm*** to a formula that describes the transmission type (am) by the horsepower (hp) and weight (wt). This creates a generalized linear model (GLM) in the binomial family.
In R:

```
#Build a model:
am.glm = glm(formula=am ~ hp + wt, data=mtcars, family=binomial)
#Test data
newdata = data.frame(hp=120, wt=2.8)
#Predict
predict(am.glm, newdata, type="response")
```

Answer
For an automobile with 120hp engine and 2800 lbs weight, the probability of it being fitted with a manual transmission is about **64.18125%**

Further detail of the function predict for generalized linear model can be found in the R documentation.
> *help(predict.glm)*
Significance Test for Logistic Regression
We can decide whether there is any significant relationship between the dependent variable y and the independent variables xk (k = 1, 2, ..., p) in the logistic regression equation. In particular, if any of the null hypothesis that βk = 0 (k = 1, 2, ..., p) is valid, then xk is statistically insignificant in the logistic regression model.
Problem
At .05 significance level, decide if any of the independent variables in the logistic regression model of vehicle transmission in data set mtcars is statistically insignificant.
Solution
We apply the function glm to a formula that describes the transmission type (am) by the horsepower (hp) and weight (wt). This creates a generalized linear model (GLM) in the binomial family.

```
> am.glm = glm(formula=am ~ hp + wt, data=mtcars, family=binomial)
  Coefficients:
              Estimate Std. Error z value Pr(>|z|)
  (Intercept) 18.86630    7.44356   2.535  0.01126 *
  hp           0.03626    0.01773   2.044  0.04091 *
  wt          -8.08348    3.06868  -2.634  0.00843 **
```

We then print out the summary of the generalized linear model and check for the p-values of the hp and wt variables.
> *summary(am.glm)*
Answer
As the p-values of the hp and wt variables are both less than 0.05 and 0.01 neither hp or wt is insignificant in the logistic regression model.

Further detail of the function summary for the generalized linear model can be found in the R documentation.
> help(summary.glm)

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

1. List types of regression

Types of Regression –

- Linear regression
- Logistic regression
- Polynomial regression
- Stepwise regression
- Stepwise regression
- Ridge regression
- Lasso regression
- ElasticNet regression

2. What is statistical significance test?

Statistical significance is the likelihood that a relationship between two or more variables is caused by something other than chance. Statistical hypothesis testing is used to determine whether the result of a data set is statistically significant. This test provides a p-value, representing the probability that random chance could explain the result. In general, a p-value of 5% or lower is considered to be statistically significant.

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

In statistics, a model is meant to provide a similarly condensed description, but for data rather than for a physical structure. Like physical models, a statistical model is generally much simpler than the data being described; it is meant to capture the structure of the data as simply as possible. In both cases, we realize that the model is a convenient fiction that necessarily glosses over some of the details of the actual thing being modeled. As the statistician George Box famously said: “All models are wrong but some are useful.”

This expresses the idea that the data can be described by a statistical model, which describes what we expect to occur in the data, along with the difference between the model and the data, which we refer to as the *error*.

4. How to test model’s performance?

Metrics that can be used for evaluation a classification model:

- Percent correction classification (PCC): measures overall accuracy. Every error has the same weight.

- Confusion matrix: also measures accuracy but distinguished between errors, i.e false positives, false negatives and correct predictions.

- Area Under the ROC Curve (AUC – ROC): is one of the most widely used metrics for evaluation. Popular because it ranks the positive predictions higher than the negative. Also, ROC curve it is independent of the change in proportion of responders.

- Lift and Gain charts: both charts measure the effectiveness of a model by calculating the ratio between the results obtained with and without the predictive model. In other words, these metrics examine if using predictive models has any positive effects or not.

  Regression Problems

- R-squared: indicate how many variables compared to the total variables the model predicted. R-squared does not take into consideration any biases that might be present in the data. Therefore, a good model might have a low R-squared value, or a model that does not fit the data might have a high R-squared value.

- Average error: the numerical difference between the predicted value and the actual value.

- Mean Square Error (MSE): good to use if you have a lot of outliers in the data. Median error: the average of all difference between the predicted and the actual values.

- Average absolute error: similar to the average error, only you use the absolute value of the difference to balance out the outliers in the data.

- Median absolute error: represents the average of the absolute differences between prediction and actual observation. All individual differences have equal weight, and big outliers can therefore affect the final evaluation of the model.

Conclusion:

- Linear Regression is a model where we predict a numerical data(dependent variable) with respect to one or more features(independent variables)
- Logistic Regression is a binary classification model where we predict based on independent variables whether the classification is ‘0’ or ‘1’
- We have learnt about multi-variate(multiple independent variables) and single-variable regression.
- We have learnt about various parameters which tell us about the accuracy of our predicted model.
- We have learnt how to build a model for our data,train it and then use it for prediction purposes.