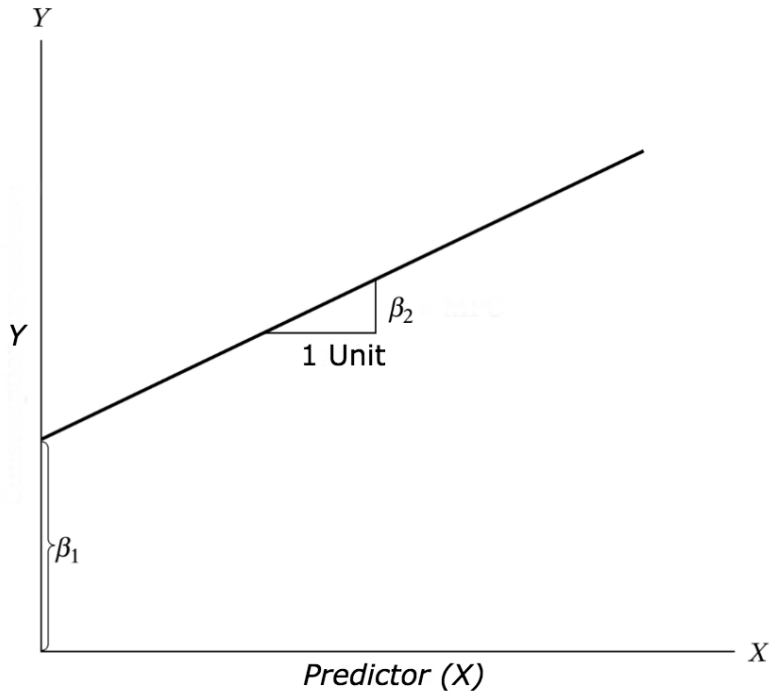

Linear Regression in R

Alok

Steps to Establish a Regression

- The steps to create the relationship is −
- Carry out the experiment of gathering a sample of observed values of height and corresponding weight.
- Create a relationship model using the **lm()** functions in R.
- Find the coefficients from the model created and create the mathematical equation using these
- Get a summary of the relationship model to know the average error in prediction. Also called **residuals**.
- To predict the weight of new persons, use the **predict()** function in R.

Graphical analysis

- 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.
- 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*

BoxPlot – Check for outliers

- Generally, any datapoint that lies outside the 1.5 * interquartile-range (1.5 * *IQR*) is considered an outlier, where, IQR is calculated as the distance between the 25th percentile and 75th percentile values for that variable.
- **par**(mfrow=**c**(1, 2)) *# divide graph area in 2 columns*
- **boxplot**(cars$speed, main="Speed", sub=**paste**("Outlier rows: ", **boxplot.stats**(cars$speed)$out)) *# box plot for 'speed'*
- **boxplot**(cars$dist, main="Distance", sub=**paste**("Outlier rows: ", **boxplot.stats**(cars$dist)$out)) *# box plot for 'distance'*

Density plot – Check if the response variable is close to normality

- **library**(e1071)
- **par**(mfrow=**c**(1, 2)) *# divide graph area in 2 columns*
- **plot**(**density**(cars$speed), main="Density Plot: Speed", ylab="Frequency", sub=**paste**("Skewness:", **round**(e1071::**skewness**(cars$speed), 2))) *# density plot for 'speed'*
- **polygon**(**density**(cars$speed), col="red")
- **plot**(**density**(cars$dist), main="Density Plot: Distance", ylab="Frequency", sub=**paste**("Skewness:", **round**(e1071::**skewness**(cars$dist), 2))) *# density plot for 'dist'*
- **polygon**(**density**(cars$dist), col="red")

- lm() Function
- This function creates the relationship model between the predictor and the response variable.
- Syntax
  - lm(formula,data)
- **formula** is a symbol presenting the relation between x and y.
- **data** is the vector on which the formula will be applied.

```
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Apply the lm() function.
relation <- lm(y~x)

print(relation)
```

```
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
   -38.4551       0.6746
```

Linear Regression Diagnostics

- Now the linear model is built and we have a formula that we can use to predict the dist value if a corresponding speed is known.
- Is this enough to actually use this model?
- NO! Before using a regression model, you have to ensure that it is statistically significant.
- How do you ensure this?
- Lets begin by printing the summary statistics for linearMod.

Summary of the Relationship

- x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
- y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
-
- # Apply the lm() function.
- relation <- lm(y~x)
-
- print(summary(relation))

```
Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3002  -1.6629  0.0412  1.8944  3.9775

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -38.45509    8.04901  -4.778  0.00139 **
x             0.67461    0.05191  12.997 1.16e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06
```

The p Value: Checking for statistical significance

- The summary statistics above tells us a number of things.
- One of them is the model p-Value (bottom last line) and the p-Value of individual predictor variables (extreme right column under ‘Coefficients’).
- The p-Values are very important because, We can consider a linear model to be statistically significant only when both these p-Values are less that the pre-determined statistical significance level, which is ideally 0.05.
- This is visually interpreted by the significance stars at the end of the row. The more the stars beside the variable’s p-Value, the more significant the variable.

Null and alternate hypothesis

- When there is a p-value, there is a null and alternative hypothesis associated with it.
- In Linear Regression, the Null Hypothesis is that the coefficients associated with the variables is equal to zero.
- The alternate hypothesis is that the coefficients are not equal to zero (i.e. there exists a relationship between the independent variable in question and the dependent variable).

t-value

- We can interpret the t-value something like this.
- A larger *t-value* indicates that it is less likely that the coefficient is not equal to zero purely by chance.
- So, higher the t-value, the better.
- *Pr(>|t|)* or *p-value* is the probability that you get a t-value as high or higher than the observed value when the Null Hypothesis (the β coefficient is equal to zero or that there is no relationship) is true.
- So if the *Pr(>|t|)* is low, the coefficients are significant (significantly different from zero). If the *Pr(>|t|)* is high, the coefficients are not significant.

- What this means to us? when p Value is less than significance level (< 0.05), we can safely reject the null hypothesis that the co-efficient β of the predictor is zero.
- In our case, linearMod, both these p-Values are well below the 0.05 threshold, so we can conclude our model is indeed statistically significant.
- It is absolutely important for the model to be statistically significant before we can go ahead and use it to predict (or estimate) the dependent variable, otherwise, the confidence in predicted values from that model reduces and may be construed as an event of chance.

How to calculate the t Statistic and p-Values?

- When the model co-efficients and standard error are known, the formula for calculating t Statistic and p-Value is as follows:

$$t - Statistic = \frac{\beta - coefficient}{Std.\ Error}$$

- modelSummary <- **summary**(linearMod) *# capture model summary as an object*
- modelCoeffs <- modelSummary$coefficients *# model coefficients*
- beta.estimate <- modelCoeffs["speed", "Estimate"] *# get beta estimate for speed*
- std.error <- modelCoeffs["speed", "Std. Error"] *# get std.error for speed*
- t_value <- beta.estimate/std.error *# calc t statistic*
- p_value <- 2***pt**(-**abs**(t_value), df=**nrow**(cars)-**ncol**(cars)) *# calc p Value*
- f_statistic <- linearMod$fstatistic[1] *# fstatistic*
- f <- **summary**(linearMod)$fstatistic *# parameters for model p-value calc*
- model_p <- **pf**(f[1], f[2], f[3], lower=FALSE)

R-Squared and Adj R-Squared

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

Standard Error and F-Statistic

predict() Function

- Syntax
- The basic syntax for predict() in linear regression is −
- predict(object, newdata)
- **object** is the formula which is already created using the lm() function.
- **newdata** is the vector containing the new value for predictor variable.

- Live Demo

- # The predictor vector.
- x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
-
- # The resposne vector.
- y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
-
- # Apply the lm() function.
- relation <- lm(y~x)
-
- # Find weight of a person with height 170.
- a <- data.frame(x = 170)
- result <- predict(relation,a)