*i*Journals: International Journal of Software & Hardware Research in Engineering
ISSN-2347-4890
Volume 7 Issue 2 February 2019

Machine Learning - Manually Calculating the Linear Regression Model Coefficients

Author: Satyanarayana Medicherla

Consultant, Canon U.S.A. Inc, 1 Canon Park, New York - 11747, email- satyams@hotmail.com

Abstract

Linear regression is a very simple statistical technique to determine the relationship between the input and output or target variables [1]. R provides a library to generate a linear regression model for the data that we want to study. Linear Model of R outputs a set of coefficients or values. Interpreting those coefficients is very essential for us to assess the model generated by R. There are different sections in the R output of linear regression. We will go through each of those sections and compute those coefficients manually. This gives us much more insight into the model. We will discuss the statistical significance of those coefficients. In this simple article, a modest attempt is made to provide simple hand-computation and interpretation of all the output provided by the summary of the linear model of R.

A look at the data set 'Women' provided by R

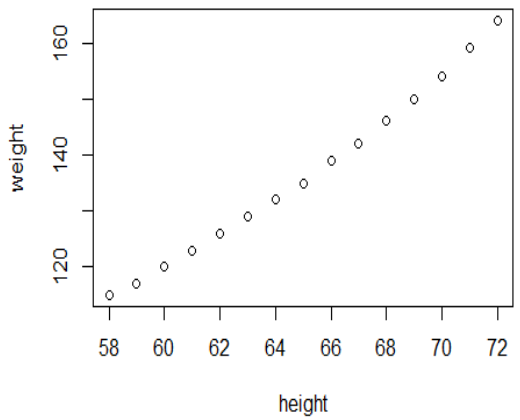
This simple and small training data set is one of the data sets provided by R by default [3]. Here is the summary of our data set.

```
summary(women)

##     height         weight
##  Min.   :58.0   Min.   :115.0
##  1st Qu.:61.5   1st Qu.:124.5
##  Median :65.0   Median :135.0
##  Mean   :65.0   Mean   :136.7
##  3rd Qu.:68.5   3rd Qu.:148.0
##  Max.   :72.0   Max.   :164.0
```

A glance at the summary would indicate height is between 58 and 72 and weight is between 115 and 164 and individual means and medians are also shown.

```
plot(women)
```



The above relation between height and the weight is very close to linear. We will use the lm function of R to fit a linear model and then print the summary of the model. The summary of the fit displays different regression coefficients. We will hand-compute different coefficients in different sections of the output and show that the displayed coefficients match the hand computed values.

```
n =length(women[,1])
m =lm(data=women,formula=weight~height)
s=summary(m)
s

##
## Call:
## lm(formula = weight ~ height, data = women)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7333 -1.1333 -0.3833  0.7417  3.1167
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -87.51667    5.93694  -14.74 1.71e-09
***
## height       3.45000    0.09114   37.85 1.09e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1
##
## Residual standard error: 1.525 on 13 degrees of
freedom
## Multiple R-squared:  0.991,  Adjusted R-squared:
0.9903
## F-statistic:  1433 on 1 and 13 DF,  p-value: 1.091e-
14
```

The model summary shown above contains the following sections

1. Call Section
2. Residuals Section
3. Coefficients Section
    a. Estimate
    b. Std. Error
    c. t-value
    d. Pr(>|t|)
4. Signif. codes Section
5. Residual Standard Error & degrees of freedom
6. Multiple R-squared and Adjusted R-squared
7. F-statistic
8. P-value

1. The Call: section of the summary is just the syntax of the lm command that we used to generate the model.

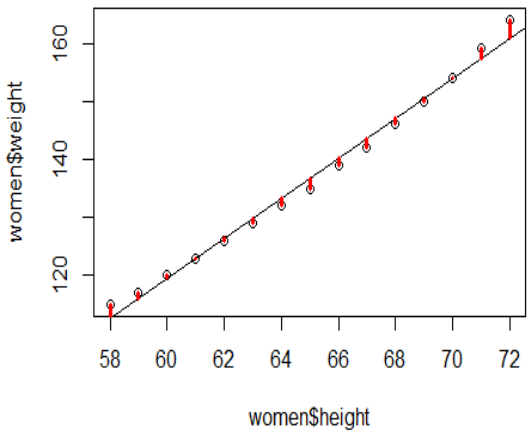
lm(formula = weight ~ height, data = women)

2. Residuals Section

Residuals are the differences between the observed value of the dependent variable (y) and the estimated value. Each of the data points in the training set has one residual. These residuals are normally distributed with a mean of zero. In this section, we are using pulling the residuals from the R model that we generated. In the next sections, we will calculate these manually.

```
plot(women$height, women$weight,
xlim=c(min(women$height), max(women$height)),
ylim=c(min(women$weight), max(women$weight)))
abline(m)

# calculate residuals and predicted values
res <-signif(residuals(m), 5)
pre <-predict(m) # plot distances between points and
the regression line
segments(women$height, women$weight,
women$height, pre, col="red",lwd=3)
```



Since the residuals are very small, it is a bit difficult to clearly see the residuals on the line segments that show the magnitude of the residuals.

Now let us see the summary of the residuals.

```
calc_res <-women$weight -pre
summary(calc_res)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.7330 -1.1330 -0.3833  0.0000  0.7417  3.1170
```

This matches the Residuals section of the R model summary. In the next sections we will compute the

predicted values using Gradient Descent method.

3. Model Coefficients Section

3a. Estimate

Under coefficients section, there are two lines, one for the Intercept and one for the independent variable height. The Estimate column of the Intercept, the y-intercept -87.51667, which is the weight of the woman of height 0 inches. As we know in some cases, the intercept does not have any meaning or interpretation. And the estimate of the height 3.45000, which is the slope of the regression line. This is the weight increase per an inch increase in height.

We can use the gradient descent or normal equation method to determine the coefficients of the least squares line.

Let us use the normal equation method to find the values of the coefficients [5].

The estimated fit is

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

where $\hat{\beta}_0$ and $\hat{\beta}_1$ are the estimated coefficients.

The residual is the difference between the estimated value and the observed value of the dependent variable. The error function is defined as the sum of the squared residuals.

The Error Function or the Residual sum of squares,

$$J = \sum_{i=1}^{n} (y_i - \hat{y})^2$$

$$J = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

Now let us compute the partial derivatives of the error function and equate the derivative to zero to find out the values for β_0 and β_1 that minimize the error function.

$$\frac{\partial J}{\partial \hat{\beta}_0} = 0$$

,

$$\frac{\partial J}{\partial \hat{\beta}_1} = 0$$

Derivative with respect to β_0

$$\sum_{i=1}^{n} 2\,(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)(-1) = 0$$

$$\sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0$$

$$n\bar{y} - n\hat{\beta}_0 - n\hat{\beta}_1 \bar{x} = 0$$

Derivative with respect to β_1

$$\sum_{i=1}^{n} 2\,(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)(-x_i) = 0$$

$$\sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) x_i = 0$$

substitute for $\hat{\beta}_0$ from equation (1)

$$\sum_{i=1}^{n} (y_i - \bar{y} + \hat{\beta}_1 \bar{x} - \hat{\beta}_1 x_i) x_i = 0$$

$$\sum_{i=1}^{n} (y_i x_i - \bar{y} x_i + \hat{\beta}_1 \bar{x} x_i - \hat{\beta}_1 x_i^2) = 0$$

$$\sum_{i=1}^{n} (y_i x_i - \bar{y} x_i + \hat{\beta}_1 \bar{x}^2 - \hat{\beta}_1 x_i^2) = 0$$

$$\sum_{i=1}^{n} (y_i x_i - \bar{y} x_i) = \sum_{n=1}^{n} (\hat{\beta}_1 x_i^2 - \hat{\beta}_1 \bar{x}^2)$$

$$\sum_{i=1}^{n} (y_i x_i - \bar{y} x_i) = \sum_{n=1}^{n} \hat{\beta}_1 (x_i^2 - \bar{x}^2)$$

$$\sum_{i=1}^{n} (y_i x_i - \bar{y} x_i) = \hat{\beta}_1 \sum_{n=1}^{n} (x_i^2 - \bar{x}^2)$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (y_i x_i - \bar{y} x_i)}{\sum_{i=1}^{n} (x_i^2 - \bar{x}^2)}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (y_i x_i - \bar{y} x_i + \bar{y} x_i - \bar{y} x_i)}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (y_i x_i - \bar{y} x_i + \bar{y}\bar{x} - y_i \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\hat{\beta_1} = \frac{\sum_{i=1}^{n}((y_i - \overline{y})x_i - (y_i - \overline{y})\overline{x}))}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

$$\hat{\beta_1} = \frac{\sum_{i=1}^{n}(y_i - \overline{y})(x_i - \overline{x})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

$$\hat{\beta_0} = \overline{y} - \hat{\beta_1}\overline{x}$$

Now let us use the above equations for computing the coefficients

```
xbar =mean(women$height)
ybar =mean(women$weight)


b1 <-sum((women$weight-ybar)*(women$height-
xbar))/sum((women$height-xbar)^2)

b0 <-ybar -b1*xbar
b0

## [1] -87.51667

b1

## [1] 3.45
```

Here is a generalization of the normal equation method - We can use the following Normal Equation [5]

$$\Theta = (X^T X)^{-1} X^T Y$$

where Θ is a vector of θ_0 and θ_1

```
X <-c(women$height)

ones <-rep(1, length(women$height))

X <-cbind(ones,X)
Y =women$weight

THETA <-solve(t(X)%*%X)%*%t(X)%*%Y

rownames(THETA) <-c("intercept","slope")
intercept <-THETA[1,1]
slope <-THETA[2,1]

intercept

## intercept
## -87.51667

slope

## slope
##  3.45
```

Gradient Descent Method to determine the coefficients

The gradient descent method uses the technique of iteratively achieving the convergence after initializing the theta values to an arbitrary set of values. It uses a constant alpha called the learning rate of the algorithm which controls the rate of convergence of the gradient descent. Here is the R source code for gradient descent [4]

```
gradient_descent <-function(x, y, alpha, n, max_limit)
{
 slope <-1
 intercept <-1
 iterations <-0
 y_new <-slope *x +intercept
 squared_error <-sum((y_new -y) ^2) /n
 squared_error_new <-0

 while(TRUE)
 {
  slope_new <-slope -alpha *((1 /n) *(sum((y_new -y)
*x)))
  intercept_new <-intercept -alpha *((1 /n)
*(sum(y_new -y)))
  slope <-slope_new
  intercept <-intercept_new
  y_new <-slope_new *x +intercept_new
  squared_error <-squared_error_new
  squared_error_new <-sum((y -y_new) ^2) /n

  iterations =iterations +1

  if(iterations >max_limit) {
print(iterations)
print(squared_error)
print(squared_error_new)
print(intercept)
print(slope)
return(paste(intercept, ":", slope))
  }
 }
}


n =length(women[,1])
x =women[,1]
```

```
y =women[,2]

#  Slope = 3.45000  Intercept - 87.51667
a=gradient_descent(x, y, 0.0001, n, 25000000)

## [1] 2.5e+07
## [1] 2.015556
## [1] 2.015556
## [1] -87.51518
## [1] 3.449977
```

The final model is as follows:

weight = 3.45000 * height - 87.51667

Now let us calculate the residuals for the height values provided in our training data set using the above calculated coefficients.

```
#intercept <- s$coefficients[ 1, 1]
intercept <--87.51667
#height_coef <- s$coefficients[ 2, 1]
height_coef <-3.45000
fit_weights =height_coef *women[,1] +intercept
res_calc =women[,2] -fit_weights
res_sq =res_calc^2
summary(res_calc)

##      Min.    1st Qu.     Median       Mean    3rd Qu.
Max.
## -1.7330000 -1.1330000 -0.3833000  0.0000033
0.7417000  3.1170000
```

Now let us pull the residuals from our model and see if they match.

```
res <-resid(m)
sd1 =sd(res)

summary(res)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.7330 -1.1330 -0.3833  0.0000  0.7417  3.1170
```

The above output is exactly matching the manually calculated residuals above.

3b. Standard Error

Now let us turn our attention to the "Std. Error"" Column of the coefficients section. Standard Error is defined as the standard deviation of the sampling distributions of a given population [6]. In this case, we are talking about how much the coefficients differ between different samples of the same population. It is calculated using the following equation

Std. Error = standard deviation of the sample/ sqrt(sample size)

Different samples drawn from the same population will have different have slightly different intercepts and slopes. The standard deviation of these different intercepts is called the Standard Error of the intercept and similarly the standard deviation of these different slopes is called the SE of the slope.

The SE of the Regression Coefficients is given by the formula

The standard error of the coefficients $\beta_0 and \beta_1$ can be computed using the following equations [2]

$$SE(\overline{\beta}_0)^2 = \sigma^2(\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2})$$

$$SE(\overline{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

As we can see from the above equation, the standard error goes lower and lower as the sample size increases. It is intuitive that if we take a much bigger sample, the deviation in sampling parameters like mean etc. will decrease.

```
s =summary(m)
names(s)

##  [1] "call"          "terms"         "residuals"
"coefficients"
##  [5] "aliased"       "sigma"         "df"
"r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"

s["sigma"]

## $sigma
## [1] 1.525005

s$coefficients[ , 1]

## (Intercept)      height
##   -87.51667     3.45000

# calculating the standard error of the estimate

#sd1 = 1.525
sd1 =sqrt(sum(res^2)/13)

# calculating the standard error of the coefficients
```

```
seb0 =sqrt(sd1^2 *(1.0/length(women[,1])
+(xbar^2/sum((women$height -xbar)^2))))
seb0

## [1] 5.936944

seb1 =sqrt((sd1^2)/sum((women$height -xbar)^2))
seb1

## [1] 0.0911365

seb0

## [1] 5.936944

seb1

## [1] 0.0911365
```

3c. t-value or t-statistic

For any of the parameters THETA, if the Standard Error is very low compared to the value of that estimated coefficient values, then the possibility of the estimated value being zero is very less for different samples of the population. The t-statistic is obtained by dividing the Estimated parameter value by the standard error of the parameter. This way we are making sure that the absolute value of the Estimated value of the coefficient sufficiently large when compared to the possible error in determining the value of the coefficient and hence the chances of the coefficient becoming zero is remote. In other words, we are making sure that the coefficient is significant.

So the t-statistic is the ratio of the estimate of the coefficient to the standard error of the coefficient.

The t-statistic for the intercept is calculated using - 87.51667/5.93694 = -14.74104

and the t-statistic of the height is calculated using 3.45000/0.09114 = 37.85385

3d. Pr(> |t|)

This is the compliment of the t-value, this talks about the probability of how many cases the coefficient values is zero. For example, let us say the SE is 2 and the estimated coefficient value is 6, we can see that 6 is equal to 3 SEs. As we know in the normal distribution, 3 standard deviation convers 99.7% indicating that 99.7% of the samples from the population will have non-zero parameter estimate.

4. Significant Codes

Let us take a look at the significant codes section

The significant codes are shown below - *** very close to zero and so on.

0 '' ***0.001*** '' *0.01* '' 0.05 '.' 0.1 ' ' 1

5. Residual Standard Error & degrees of freedom

It is the Standard Error of the residuals - for different samples of the same population, the sum of the squared residuals will be different - if we calculate the standard deviation of these sum of the squared residuals for different possible samples, it is the Residual Standard Error. The less the residual standard error, the better is our fit.

It is the square root of the sum of the squared residuals averaged over the degrees of freedom.

```
res <-s$residuals
res^2

##          1          2          3          4          5
## 5.8402777778 0.9344444444 0.2669444444
0.0044444444 0.1469444444
##          6          7          8          9         10
## 0.6944444444 1.6469444444 3.0044444444
1.4002777778 2.6677777778
##         11         12         13         14         15
## 1.1736111111 0.2844444444 0.0002777778
2.4544444444 9.7136111111

sum_ressq =sum(res^2)
sum_ressq

## [1] 30.23333

stderr =sqrt(sum_ressq/(length(women$height)-2))
stderr

## [1] 1.525005

s["sigma"]

## $sigma
## [1] 1.525005
```

As you can see above both the calculated and extracted value from the model are matching for Residual Standard Error.

6. Multiple R-squared and Adjusted R-squared.

R-squared

R-squared [7] tells us how good is our fit when compared to the intercept-only of NULL model.

a) find the sum of the squared distances of each of our data points from the regression line (Our model)
b) find the sum of the squared distances of each of our data points from the mean of the dependent variable (NULL model)
c) divide the result in step-a by the result in step-b - this tells us how good is our fit when compared to NULL model
d) subtract the result in step-c from 1 and convert it into percentage - this tells how much of the variation in dependent variable is explained by the independent variable - the higher the R-squared, the better is the fit

If we say that R-squared is 60%, it means that 60% of the variance in dependent variable explained by the independent variable.

R-squared can be obtained from the fitted model as follows

R-squred obtained from model is s$r.squared

R-squared <- (1.0 - SSE/SST)

Where SST is the sum of squared residuals of Intercept-only model and SSE is the sum of squared residuals of our fit

Sum of Squared Error, SSE = sum($residiaul^2$) Sum of Sqaured Total, SST is the sum of the errors from the mean of the dependent variable

```
ybar =mean(women$weight)
ybar

## [1] 136.7333

SSE <-sum(res_calc^2)
SSE

## [1] 30.23333

SST <-sum((women$weight -ybar)^2)
SST

## [1] 3362.933

rsquared <-1-(SSE/SST)

rsquared

## [1] 0.9910098

s$r.squared

## [1] 0.9910098
```

As you can see above, the computed r-squared and r-squared from the model matched

Adjusted R-squared

It is always not true that the higher the R-squared, the better the fit. Particularly in multi-variate linear models, we can increase the R-squared by adding more and more independent variables. When we add more and more independent variable, we are decreasing the degrees of freedom (df=n-k-1) as a result R-squared will increase. That does not mean our fit is getting better with the addition of more and more independent variables. This is where the adjusted R-squared comes into rescue. This is done by penalizing for each of the added features or variables.

Adjusted R-squared [8] is calculated as follows

```
n <-length(women$height) # sample size
k <-length(women[1,]) -1# k is the number of predictors
adj_rsquared =1 -(1 -rsquared) *(n -1)/(n -k -1)
adj_rsquared

## [1] 0.9903183

s$adj.r.squared

## [1] 0.9903183
```

7. F-statistic:

F-statistic determines the quality of the fit - it compares the model with variables with intercept-only (or no-variables model) [10]. Here the null hypothesis is the fit with variables is no way better than the intercept-only model and the alternative hypothesis is our fit with variables is significantly better than the intercept only model.

State the null hypothesis is and the alternate hypothesis. Calculate the F value. The F Value is

calculated using the formula [9] F = Mean Squared Model/Mean Squared Error

F = (Sum of Squared Residuals with no predictors - Sum of Squared Residual with predictors)/Sum of Squared Residuals with predictors

Where the number of restrictions on the numerator is P (number of variables) and that of the denominator is N-P-1.

When the model explains most of the variation in the dependent variable, then the numerator is much more than the denominator resulting in a high value of F-statistic indicating that it is a good model.

where SSE = residual sum of squares, m = number of restrictions and k = number of independent variables. Find the F Statistic (the critical value for this test).Sep 12, 2013

$F = (R^{2/(k-1)})/((1-R2)/(n-k))$

SSE = SST ybar = mean(women[,2]) rsquared_mean = sum((women[,2] - ybar)^2)

F = ((SSE-SST)/1) / (SST / (n - k))

$$F = \frac{(R^2}{(k-1)} / \frac{(1-R^2)}{(n-k)}$$

where n = number of training examples = 15, k = number of variables + 1 = 2

```
k =2

F =((rsquared)/(2-1))/((1-rsquared)/(n-k))

F

## [1] 1433.024

# pull the F-statistic from the model

s$fstatistic

##    value    numdf    dendf
## 1433.024    1.000   13.000
```

8. P-value

The higher the F value, the lower the p-value. When the F-statistic is high, the p-value is the area under the F-curve to the right of the F value. When the p-value is higher than your alpha level, then then reject the NULL hypothesis. If the p-value is below your alpha level, go ahead and check the individual p-values. If there is one predictor, the p-value of the predictor and the p-value of the overall model is the same.

```
pf(q=1433, df1=1, df2=13, lower.tail=FALSE)

## [1] 1.091092e-14
```

Conclusion

As we know, R displays a set of coefficients for linear regression models. We have discussed how to calculate manually compute these coefficients. We showed that the hand computed regression coefficients match the regression coefficients displayed by R. This helped us gain more insight into our model.

References

1. https://en.wikipedia.org/wiki/Simple_linear_regression
2. Introduction to Statistical Learning, Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Springer Texts in Statistics
3. https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/women.html
4. https://www.r-bloggers.com/implementing-the-gradient-descent-algorithm-in-r/
5. http://cs229.stanford.edu/notes/cs229-notes1.pdf
6. https://stattrek.com/estimation/standard-error.aspx
7. http://blog.minitab.com/blog/statistics-and-quality-data-analysis/r-squared-sometimes-a-square-is-just-a-square
8. https://www.listendata.com/2014/08/adjusted-r-squared.html
9. http://www.philender.com/courses/linearmodels/notes1/nopredict.html
10. http://www.statisticshowto.com/probability-and-statistics/f-statistic-value-test/