Multiple Linear Regression

Introduction – Part II

Content

Partitioning Total Variance

- Total Variation in dependent variables Y can be split into two: Explained and Unexplained.
- Explained variation is the summation of squared difference between estimated values of Y and the mean value of Y. Whereas, sum of squared difference between the actual values of Y and estimated values is considered to be unexplained.

Total Variation

$$\sum_{i=1}^{n}(Y_i - \bar{Y})^2$$

Explained Variation

$$\sum_{i=1}^{n}(\widehat{Y}_i - \bar{Y})^2$$

Unexplained Variation

$$\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$$

Global Testing – Using F Test

Testing **whether at least one variable is significant**

Objective	To test the **null hypothesis** that **all the parameters are simultaneously equal to zero**

Null Hypothesis (H_0): $b_1 = b_2 = \ldots = b_p = 0$

Alternate Hypothesis (H_1): At least one coefficient is not zero

Test Statistic	$F = \dfrac{\text{Mean Square of Regression}}{\text{Mean Square of Error}}$
Decision Criteria	Reject the null hypothesis **if p-value < 0.05**

Global Testing – Using F Test

ANOVA Table

Source	DF (Degrees of Freedom)	SS (Sum of Squares)	MSS =SS/DF (Mean Sum of Squares)	F Value	Pr > F
Regression(Explained)	p=4	2510.007	627.5017	49.8129	<0.0001
Error(Unexplained)	n-p-1=28	352.7208	12.5972		
Total	n-1=32	2862.728			

Reject the null hypothesis since p-value < 0.05

At least one variable has significant impact on performance index

* **Note : This slide is in continuation of the previous slide. So the data considered is the same, “Performance Index” data.**

Individual Testing – Using t Test

Testing **which variable is significant**

Objective	To test the **null hypothesis** that **parameters of individual variables are equal to zero**

Null Hypothesis (H_0): $b_i = 0$

Alternate Hypothesis (H_1): $b_i \neq 0$

where $i = 1,2,\ldots,p$

Test Statistic	$t = \dfrac{\text{Estimated } b_i}{\text{Standard Error of Estimated } b_i}$
Decision Criteria	Reject the null hypothesis **if p-value < 0.05**

Individual Testing – Using t Test

Parameters	Coefficients	Standard Error	t statistic	p-value
Intercept	-54.2822	7.3945	-7.3409	0.0000
aptitude	0.3236	0.0678	4.7737	0.0001
tol	0.0334	0.0712	0.4684	0.6431
technical	1.0955	0.1814	6.0395	0.0000
general	0.5368	0.1584	3.3890	0.0021

p-values for aptitude, technical and general are < 0.05

p-value for test of language (tol) is > 0.05

Therefore, tol is the only insignificant variable

Measure of Goodness of Fit – R Squared

R^2 is the proportion of variation in the dependent variable which is explained by the independent variables. Note that R^2 always increases if variable is added in the model

$$R^2 = \frac{\text{Explained Variation}}{\text{Total Variation}} = \frac{\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the modelling.

$$R_a^{\,2} = 1 - \frac{n-1}{n-p-1}(1 - R^2)$$

The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model.

Normally, R^2 greater than 0.7 is considered as a benchmark for accepting goodness of fit of a model.

Understanding Summary Output

```
#Model Summary
summary(jpimodel)
```

summary() generates a detailed description of the model.

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -54.28225    7.39453  -7.341 5.41e-08 ***
aptitude      0.32356    0.06778   4.774 5.15e-05 ***
tol           0.03337    0.07124   0.468   0.6431    
technical     1.09547    0.18138   6.039 1.65e-06 ***
general       0.53683    0.15840   3.389   0.0021 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.549 on 28 degrees of freedom
Multiple R-squared:  0.8768,    Adjusted R-squared:  0.8592 
F-statistic: 49.81 on 4 and 28 DF,  p-value: 2.467e-12
```

Interpretation :

- *Reject null hypothesis that no variables are significant as p-value is<0.05*
- *aptitude, technical, general are significant variables (p-values<0.05)*
- *tol is not significant (p-value>0.05)*

Summary of Findings

Significant variables → **Aptitude**, **Technical knowledge**, **General information**

Out of four dependent variables, **three affect job performance index positively**

R^2 → **0.88**

88% of the variation in job performance index is explained by the model & 12% is unexplained variation

Fitted Values and Residuals

- Fitted values (also called 'Predicted Values') are calculated using **estimated model parameters** and by substituting values of independent variables. The model now will include only the significant variables.

Estimated Model:

E(jpi)= **-54.40644** + **0.33335***aptitude + **1.11663***technical + **0.54316***general

Values of Independent Variables for First Employee	
aptitude	43.83
tol	55.92
technical	51.82
general	43.58

Fitted Values and Residuals

Values of Independent Variables for First Employee	
aptitude	43.83
technical	51.82
general	43.58

aptitude technical general

jpi= **-54.40644** + **0.33335***43.83 + **1.11663***51.82 + **0.54316***43.58

Predicted jpi= 41.73850

Residual= Observed jpi – Predicted jpi = 45.52 - 41.73850 = 3.781497

Fitted Values and Residuals

```
#Model Fitting after eliminating the insignificant variable
```

```
jpimodel_new<-lm(jpi~aptitude+technical+general,data=perindex)
jpimodel_new
```

The insignificant variable tol is not included in the new model

```
#Output
```

```
Coefficients:
(Intercept)     aptitude    technical      general
   -54.4064       0.3333       1.1166       0.5432
```

Estimated values of the model parameters using the new model

* To get the fitted values and the residuals values, the model should include only the significant variables

Fitted Values and Residuals

```
#Adding Fitted Values and Residuals to the Original Dataset
perindex$pred<-fitted(jpimodel_new)

perindex$resi<-residuals(jpimodel_new)
```

fitted() and residuals() fetch fitted values and residuals respectively.

#Output

	empid	jpi	aptitude	tol	technical	general	pred	resi
1	1	45.52	43.83	55.92	51.82	43.58	41.73850	3.781497
2	2	40.10	32.71	32.56	51.49	51.03	41.70973	-1.609731
3	3	50.61	56.64	54.84	52.29	52.47	51.36215	-0.752151
4	4	38.97	51.53	59.69	47.48	47.69	41.69149	-2.721486
5	5	41.87	51.35	51.50	47.59	45.77	40.71145	1.158549
6	6	38.71	39.60	43.63	48.34	42.06	35.61699	3.093010

Interpretation :

- *pred values are calculated based on the values of the model parameters*
- *resi is the difference between the actual jpi values and the pred values.*
- *Lower the residuals, lesser is the difference between fitted and observed and better is the model.*

Predictions for New Dataset

- New data set should have all the independent variables used in the model
- Column names of all common variables in the new and old datasets should be identical
- Note that missing values will be taken as 0 (which can be incorrect)

```
#Importing New Dataset
perindex_new<-read.csv("Performance Index new.csv", header=TRUE)

perindex_new$pred<-predict(jpimodel_new,perindex_new)

head(perindex_new)
```

predict() returns predicted values. Fitted model is the first argument and new dataset object is the second argument. This ensures R uses parameters from the fitted model for predictions on new data.

	empid	jpi	tol	technical	general	aptitude	pred
1	34	66.35	59.20	57.18	54.98	66.74	61.55258
2	35	56.10	64.92	52.51	55.78	55.45	53.00898
3	36	48.95	63.59	57.76	52.08	51.73	55.62154
4	37	43.25	64.90	50.13	42.75	45.09	39.82060
5	38	41.20	51.50	47.89	45.77	50.85	40.87977
6	39	50.24	55.77	51.13	47.98	53.86	46.70139

Predictions with Confidence Interval

```
#Predictions with Confidence Interval
predict(jpimodel_new, perindex_new, interval="confidence")
```

interval ="confidence" generates 95% confidence intervals by default

```
#Output
       fit      lwr      upr
1 61.55258 59.00956 64.09559
2 53.00898 50.67792 55.34004
3 55.62154 53.65401 57.58906
4 39.82060 37.73390 41.90730
5 40.87977 39.23364 42.52590
6 46.70139 45.41627 47.98650
```

Q. Why are confidence intervals needed for predictions?

A. The point estimate is the best guess of the true value of the parameter, while the interval estimate gives a measure of accuracy of that point estimate by providing an interval that contains plausible values.

* **If you wish to specify the level of tolerance/confidence, use level= argument in the predict() function. For example, to calculate 90% confidence intervals, level=0.90**

Standardized Coefficients

How to determine relative importance of predictors?

One possible answer is standardized regression coefficient

Predictors can have very different types of units, which make comparing the regression coefficients meaningless. One solution is to standardize all variables before performing regression analysis.

standardization refers to the process of subtracting the mean (μ) from each value and dividing by the standard deviation (σ).

$$\boxed{Z = \frac{X - \mu}{\sigma}}$$

	X1	X2	Standardized X1	Standardized X2
	32	1052	-0.20	-1.74
	37	1237	0.46	-1.06
	25	1672	-1.12	0.54
	39	1724	0.72	0.74
	23	1555	-1.38	0.11
	41	1423	0.99	-0.37
	43	1870	1.25	1.27
	28	1661	-0.72	0.50
Mean	33.5	1524.25		
SD	7.60	271.69		

Standardized Coefficient - R code

Generation of standardized parameter estimate

#Install and load package lm.beta

```
install.packages("lm.beta")
library(lm.beta)

lm.beta(jpimodel_new)
```

❑ *lm.beta function in "lm.beta" is used to generate the standardized parameter estimate*

#Output

```
Standardized Coefficients::
(Intercept)    aptitude   technical     general
  0.0000000   0.3543742   0.5880966   0.3236793
```

Interpretation:

- *technical has highest impact on job performance index followed by aptitude*

Quick Recap

Check Variable Significance	• Undertake global and individual testing
Measure Goodness of Fit	• Check R-squared, Adjusted R-squared to see how much variation is explained by the model • Generally, R-squared greater than 0.6 is considered to be a good indicator
Summary Output	• Summary of **lm()** output is exhaustive and gives t statistics, p-value, R^2 to draw fundamental conclusions about the model

Quick Recap

Fitted Values and Errors	• **`fitted()`** and **`residuals()`** are used to fetch fitted values and residuals respectively
Predictions	• **`predict()`** function predicts values for new data • Predictions can be obtained as either point estimates or as confidence intervals
Standardizing Coefficients	• **lm.beta()** function in package **lm.beta** gives the standardized coefficients. • It is used to compare the relative importance of independent variables when the variables are in different metric units