Introduction to

Multiple Linear Regression II

Content

1. Global Testing – ANOVA
2. Individual Testing – t Test
3. Measure of Goodness of Fit – R Squared
4. Fitted values and Residuals
5. Predictions for New Dataset
6. Standardizing Coefficients

Partitioning Total Variance

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

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

Unexplained Variation

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

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

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**

Measure of Goodness of Fit – R Squared

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

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

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

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

The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. Normally, $\mathbf{R^2}$ greater than 0.7 is considered as the benchmark for accepting the goodness of fit of a model.

Understanding Summary Output

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

summary() generates a detailed description of the model.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    jpi   R-squared:                       0.877
Model:                            OLS   Adj. R-squared:                  0.859
Method:                 Least Squares   F-statistic:                     49.81
Date:                Wed, 23 Oct 2019   Prob (F-statistic):           2.47e-12
Time:                        14:01:20   Log-Likelihood:                -85.916
No. Observations:                  33   AIC:                             181.8
Df Residuals:                      28   BIC:                             189.3
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -54.2822      7.395     -7.341      0.000     -69.429     -39.135
tol            0.0334      0.071      0.468      0.643      -0.113       0.179
aptitude       0.3236      0.068      4.774      0.000       0.185       0.462
technical      1.0955      0.181      6.039      0.000       0.724       1.467
general        0.5368      0.158      3.389      0.002       0.212       0.861
==============================================================================
Omnibus:                        2.124   Durbin-Watson:                   1.379
Prob(Omnibus):                  0.346   Jarque-Bera (JB):                1.944
Skew:                          -0.544   Prob(JB):                        0.378
Kurtosis:                       2.518   Cond. No.                     1.25e+03
==============================================================================
```

Interpretation :

- *Reject Global Testing null hypothesis that no variables are significant as p-value is<0.05*
- *Intercept, 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

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

```
jpimodel_new=smf.ols('jpi ~ aptitude + technical +general',
data=perindex).fit()
jpimodel_new.params
```

The insignificant variable tol is not included in the new model

```
#Output
```

```
Intercept    -54.406443
aptitude       0.333346
technical      1.116627
general        0.543157
dtype: float64
```

Estimated values of the model parameters using the new model

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

Fitted Values and Residuals

#Adding Fitted Values and Residuals to the Original Dataset

```
perindex=perindex.assign(pred=pd.Series(jpimodel_new.fittedvalues))
perindex=perindex.assign(res=pd.Series(jpimodel_new.resid))
perindex.head()
```

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

#Output

	empid	jpi	aptitude	tol	technical	general	pred	res
0	1	45.52	43.83	55.92	51.82	43.58	41.738503	3.781497
1	2	40.10	32.71	32.56	51.49	51.03	41.709731	-1.609731
2	3	50.61	56.64	54.84	52.29	52.47	51.362151	-0.752151
3	4	38.97	51.53	59.69	47.48	47.69	41.691486	-2.721486
4	5	41.87	51.35	51.50	47.59	45.77	40.711451	1.158549

Interpretation :

- *pred values are calculated based on the values of the model parameters*
- *res 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 a New Dataset

- A 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=pd.read_csv("Performance Index new.csv")
perindex_new=perindex_new.assign(pred=pd.Series(jpimodel_new.predict(p
erindex_new)))
```

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

```
perindex_new.head()
```

	empid	jpi	tol	technical	general	aptitude	pred
0	34	66.35	59.20	57.18	54.98	66.74	61.552576
1	35	56.10	64.92	52.51	55.78	55.45	53.008978
2	36	48.95	63.59	57.76	52.08	51.73	55.621537
3	37	43.25	64.90	50.13	42.75	45.09	39.820600
4	38	41.20	51.50	47.89	45.77	50.85	40.879766

Predictions with Confidence Interval

```
#Predictions with Confidence Interval
```

```
result = jpimodel_new.get_prediction(perindex_new)
result.conf_int()
```

- *conf_int() generates 95% confidence intervals by default.*
- *Left hand side values in array gives lower confidence interval values, right gives upper.*

```
#Output
```

```
array([[59.00955719, 64.09559387],
       [50.67791702, 55.34003898],
       [53.65401364, 57.58906082],
       [37.73389546, 41.90730465],
       [39.23363549, 42.52589584],
       [45.41626758, 47.98650295]])
```

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 alpha= argument in the conf_int() function. For example, to calculate 90% confidence intervals,** alpha = 0.1

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 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).

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

	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 - Python code

Generation of standardized parameter estimate

```
import pandas as pd
import numpy as np
from scipy import stats
import statsmodels.formula.api as smf

# standardizing dataframe
df_z =
perindex.select_dtypes(include=[np.number]).dropna().apply(stats.zscore)

# fitting regression
formula = 'jpi ~ aptitude + technical + general'
std_coef = smf.ols(formula, data=df_z).fit()
std_coef.params
```

- *stats.zscore standardizes the specified variables.*
- *.dropna(), Otherwise, stats.zscore will return all NaN for a column if it has any missing values.*
- *.select_dtypes(include=[np.number]) selects the numeric columns from data frame*
- *.params gives the standardized coefficients.*

Standardized Coefficient - Python code

#Output

```
Intercept   -9.072604e-16
aptitude     3.543742e-01
technical    5.880966e-01
general      3.236793e-01
dtype: float64
```

Interpretation:

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

Quick Recap

Till now, we learnt the **basics of multiple linear regression**. Follow these simple steps to carry out your first analysis:

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 0.7 is considered to be a good indicator
Summary Output	• Summary of **ols()** output is exhaustive and gives t statistics, p-value, R^2 to draw fundamental conclusions about the model

Quick Recap

In this session, we learnt how to **perform basic multiple linear regression in R**:

Fitted Values and Errors	• `fitted()` and `resid()` 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	• `stats.zscore` function in package **scipy** gives the standardized coefficients. • It is used to compare the relative importance of independent variables when the variables are in different metric units