<https://www.statsmodels.org/dev/gettingstarted.html>

<https://mode.com/python-tutorial/libraries/statsmodels/>

# Statsmodels

As its name implies, statsmodels is a Python library built specifically for statistics. Statsmodels is built on top of [NumPy](https://mode.com/python-tutorial/libraries/numpy), [SciPy](https://mode.com/python-tutorial/libraries/scipy), and [matplotlib](https://mode.com/python-tutorial/libraries/matplotlib), but it contains more advanced functions for statistical testing and modeling that you won't find in numerical libraries like NumPy or SciPy.

## Statsmodels tutorials

The tutorials below cover a variety of statsmodels' features.

### Linear regression

* [A friendly introduction to linear regression (using Python)](https://github.com/justmarkham/DAT4/blob/master/notebooks/08_linear_regression.ipynb" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (Data School)
* [Linear Regression with Python](http://connor-johnson.com/2014/02/18/linear-regression-with-python/" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (Connor Johnson)
* [Using Python statsmodels for OLS linear regression](http://markthegraph.blogspot.com/2015/05/using-python-statsmodels-for-ols-linear.html" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (Mark the Graph)
* [Linear Regression](http://statsmodels.sourceforge.net/stable/regression.html" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (Official statsmodels documentation)

### Multiple regression

* [Multiple Regression using Statsmodels](https://www.datarobot.com/blog/multiple-regression-using-statsmodels/" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (DataRobot)

### Logistic regression

* [Logistic Regression in Python](http://blog.yhat.com/posts/logistic-regression-and-python.html" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (Yhat)

### Time series analysis

* [A Simple Time Series Analysis Of The S&P 500 Index](http://www.johnwittenauer.net/a-simple-time-series-analysis-of-the-sp-500-index/" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (John Wittenauer)
* [Time Series Analysis in Python with statsmodels](http://conference.scipy.org/scipy2011/slides/mckinney_time_series.pdf" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (Wes McKinney, Josef Perktold, and Skipper Seabold)
* [Time Series Analysis](http://statsmodels.sourceforge.net/stable/tsa.html" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (Official statsmodels documentation)

### Statistical tests

* [Regression Diagnostics and Specification Tests](http://statsmodels.sourceforge.net/stable/diagnostic.html" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (Official statsmodels documentation)

## Statsmodels resources

* [Chapter 11: Regression](http://greenteapress.com/thinkstats2/html/thinkstats2012.html" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) of [Think Stats](http://www.amazon.com/gp/product/1449307116/ref=as_li_qf_sp_asin_tl?ie=UTF8&tag=greenteapre01-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=1449307116" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (Allen B. Downey) - This chapter covers aspects of multiple and logistic regression in statsmodels. It explains the concepts behind the code, but you'll still need familiarity with basic statistics before diving in.
* The statsmodels section of [Cross Validated](http://stats.stackexchange.com/questions/tagged/statsmodels" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) - A question and answer site for people interested in statistics, machine learning, data analysis, data mining, and data visualization.
* [Logistic regression vs. multiple regression](https://cooldata.wordpress.com/2012/08/20/logistic-regression-vs-multiple-regression/" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank) (CoolData) - Not Python related, but this provides a helpful breakdown of the differences between logistic and multiple regression.
* [Official statsmodels documentation](http://statsmodels.sourceforge.net/stable/" \t "https://mode.com/python-tutorial/libraries/statsmodels/_blank)

### **Why StatsModels?**

We can work with statistics in a way that no other platform will allow us as the statsmodel itself is made, keeping the purpose of hardcore statistics in mind. It has more inclination towards R and is a perfect tool when analyzing statistical things. Most of the developers who program in R can make use of this and can easily make their move to Python using this package.

### **Install StatsModels**

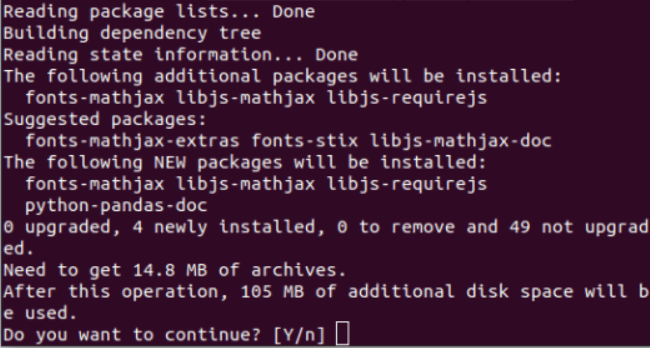
We can install the library of statsmodel by executing certain commands. We have to make sure that the following prerequisites are fulfilled –

* Numpy 1.6 package or later version
* Pandas 0.12 package or later version
* Cython 0.24 package or later version
* Patsy 0.2.1 package or later version
* Scipy 0.11 package or later version
* Python 2.6 package or later version

If you have all the above packages installed on your machine, you can go for installing statsmodel by using the terminal commands. For this, open the terminal, and by using the pip, you need to enter the following command on the terminal –

Sudo pip install statsmodel

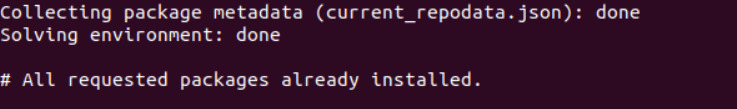
Executing the above command will give the following output –



As an alternative to the above, you can try installing statsmodel by using Conda, and the command will be –

Sudo conda install statsmodel

The execution of command on the terminal may result in the following –



The statsmodel package will be ready for use once you follow the above installation steps.

****Using StatsModels****

Once the statsmodel is installed, you can make use of the statsmodel package inside your Python program simply by importing the package at the top of the file by using the below import statement –

Import statsmodel

After that, you can use the functions and features of statsmodel in that Python program.

### **Example of Statsmodels Python**

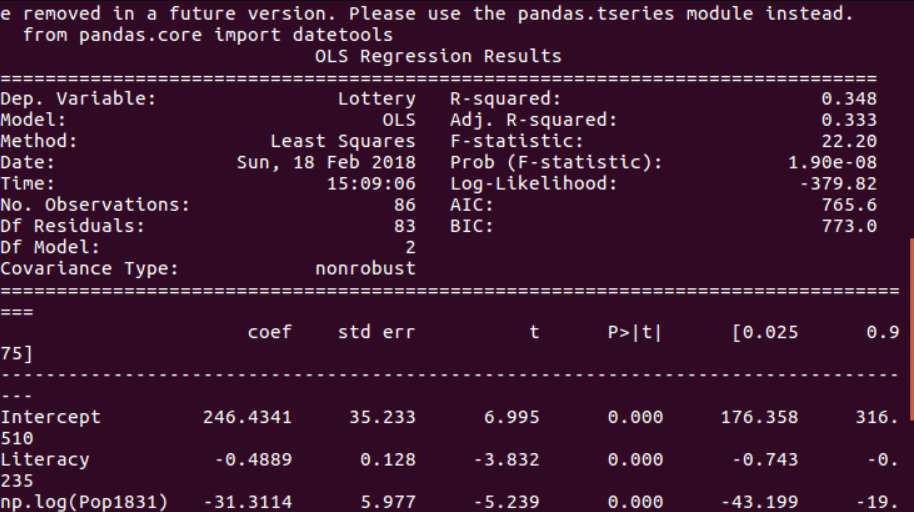
Let us consider a straightforward example to help you better understand the package use. The steps we will follow will be to import the necessary packages, load the data we want to use, create the regression model, and fit it. In our example, we will use a regressor with a natural log of one. The final step will involve the inspection of the obtained results. We will write the following code of Python–

//importing the required packagesimport numpy as npimport statsmodels.api as statsEducbaimport statsmodels.formula.api as statsEducbaModel// Providing the dataset for loading

sampleEducbaData= statsEducba.datasets.get\_rdataset ("Educba", "Payal").data// Fitting the model by using regression with the log of 1

sampleOutputRes = statsEducbaModel.ols('Lottery ~ Literacy + np.log(Pop1831)', data = sampleEducbaData).fit()// showing the final summary of output containing analysisprint (sampleOutputRes.summary()))

The output of executing the above code is as shown in the below image –



### **Linear Regression StatsModels**

After you have learned the basics of using the statsmodel, it’s time to turn to a more sophisticated part where we will implement the linear regression in the source data with the help of the statsmodel package. We will follow the same steps mentioned in the above example with one additional part for the OLS model. Let us directly jump to code and then try to understand it –

// importing the necessary packagesimport numpy as educbaSampleNumpyimport stateducbaSampleStatsodels.api as educbaSampleStats// Loading the source data set

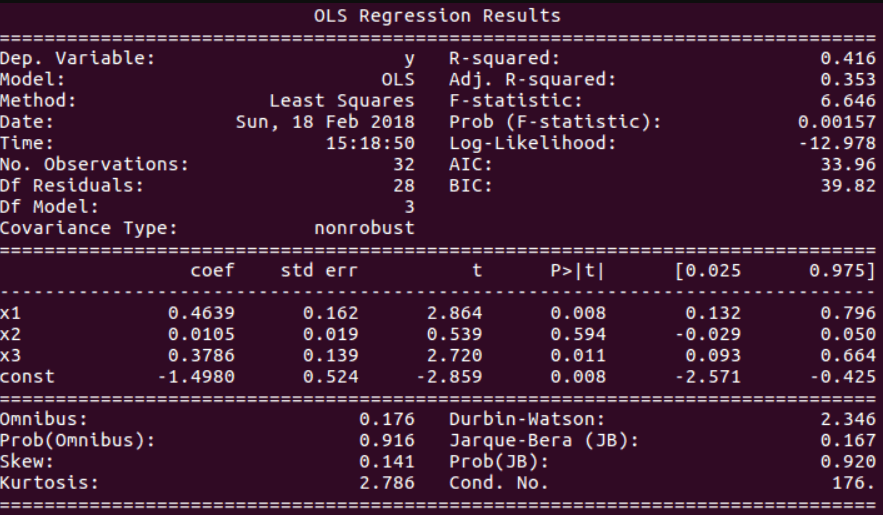
educba\_data = educbaSampleStats.datasets.spector.load()// Adding constants to the data file

educba\_data.exog = educbaSampleStats.add\_constant(educba\_data.exog, prepend=False)//Fitting the model which is in OLS

educbaModel = educbaSampleStats.OLS(educba\_data.endog, educba\_data.exog)

res = educbaModel.fit()// Summarize the statistical results and printing the same on console print(res.summary())

We can easily read the details of the result from the output. Execution of the above code gives the following output –



Let us take one more example that will implement the generalized linear models, also denoted as GLMs, with the support of exponential families with one parameter for estimation. It can be implemented simply by observing the below example –

// Importing required library of statsmodelimport statsmodels.api as educba\_stat// Loading the data

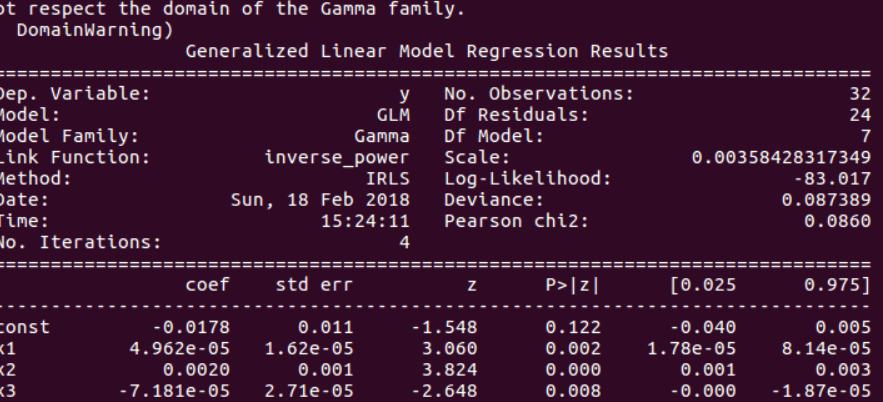
educba\_sample\_data = educba\_stat.datasets.scotland.load()// Creating the data file along with addition of constants to it

educba\_sample\_data.exog = educba\_stat.add\_constant(educba\_sample\_data.exog)// Using the link function provided by default try to initialize the gamma family model

educba\_GAMA = educba\_stat.GLM(educba\_sample\_data.endog, educba\_sample\_data.exog, family=educba\_stat.families.Gamma())// Fit the model

educba\_gamma\_results = educba\_GAMA.fit()// print the summaryprint(educba\_gamma\_results.summary())

The output of the above code is as shown in the below image –



### **Conclusion**

We can use statsmodel to perform the statistical analysis and create the new models in just a few lines of code without much hassle, giving us an obvious and easy-to-understand summary output.