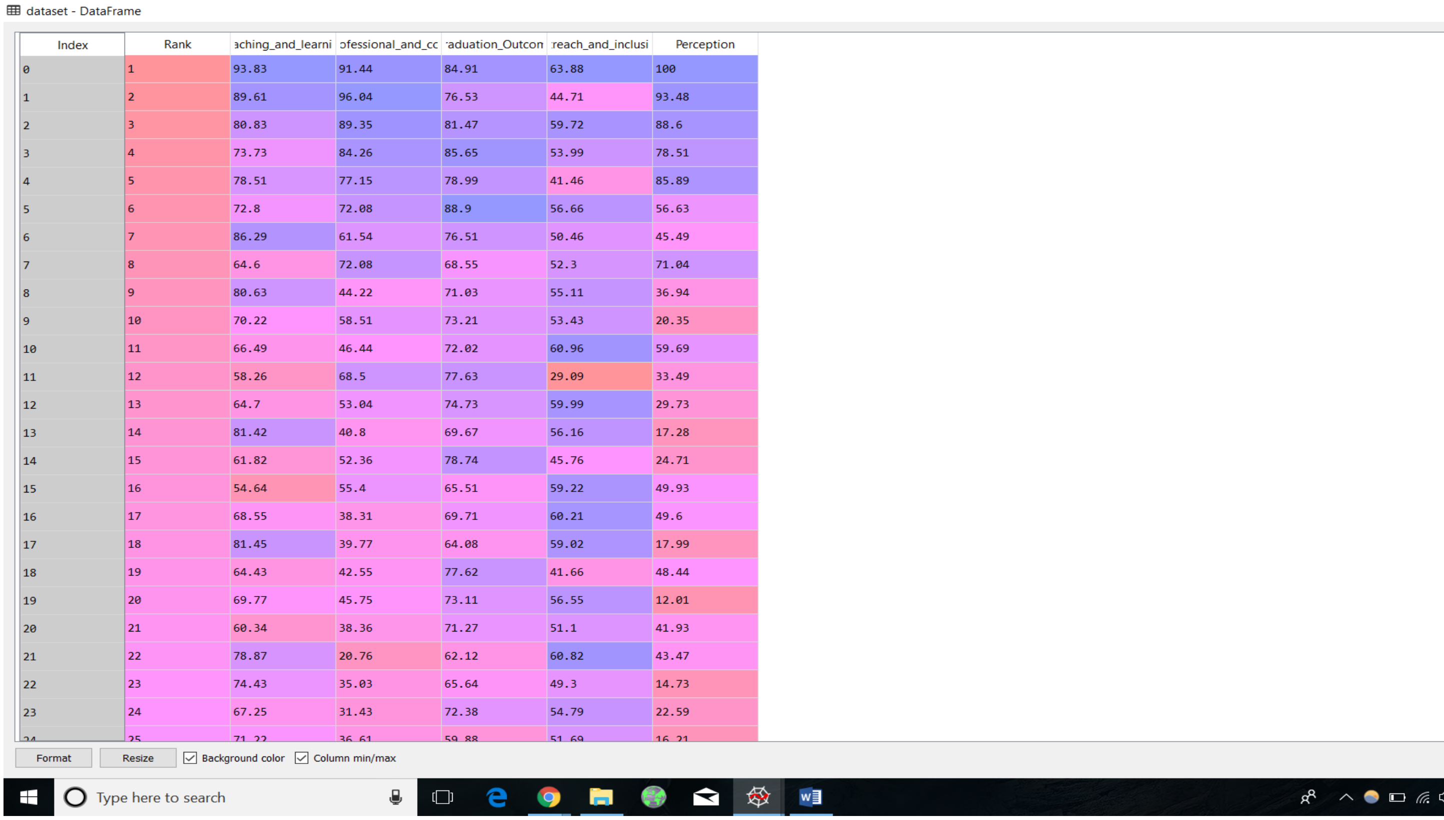
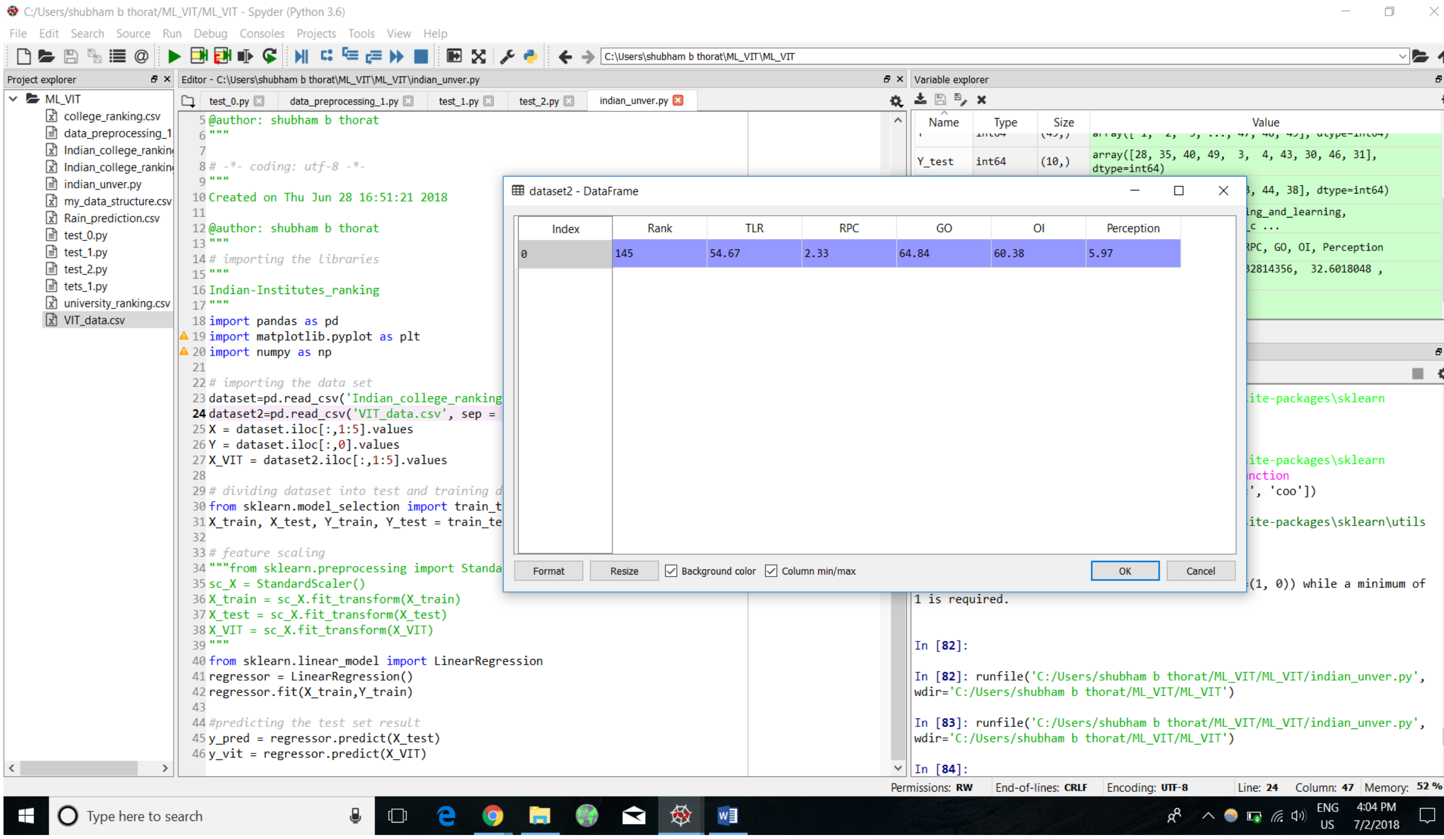
```
# -*- coding: utf-8 -*-
"""
Created on Mon Jul  2 15:27:32 2018

@author: shubham b thorat
"""


# -*- coding: utf-8 -*-
"""
Created on Thu Jun 28 16:51:21 2018
Indian-Institutes_ranking
@author: shubham b thorat
"""
# importing the libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# importing the data set
dataset=pd.read_csv('Indian_college_ranking.csv', sep = "\t")
dataset2=pd.read_csv('VIT_data.csv')
X = dataset.iloc[:,1:5].values
Y = dataset.iloc[:,0].values
X_VIT = dataset2.iloc[:,1:5].values

# dividing dataset into test and training dataset
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 1/5, random_state = 1)

# feature scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.fit_transform(X_test)
X_VIT = sc_X.fit_transform(X_VIT)

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train,Y_train)

#predicting the test set result
y_pred = regressor.predict(X_test)
y_vit = regressor.predict(X_VIT)

data_sheet =
```



Index	Rank	aching_and_learni	ofessional_and_cc	raduation_Outcon	reach_and_inclusi	Perception
0	1	93.83	91.44	84.91	63.88	100
1	2	89.61	96.04	76.53	44.71	93.48
2	3	80.83	89.35	81.47	59.72	88.6
3	4	73.73	84.26	85.65	53.99	78.51
4	5	78.51	77.15	78.99	41.46	85.89
5	6	72.8	72.08	88.9	56.66	56.63
6	7	86.29	61.54	76.51	50.46	45.49
7	8	64.6	72.08	68.55	52.3	71.04
8	9	80.63	44.22	71.03	55.11	36.94
9	10	70.22	58.51	73.21	53.43	20.35
10	11	66.49	46.44	72.02	60.96	59.69
11	12	58.26	68.5	77.63	29.09	33.49
12	13	64.7	53.04	74.73	59.99	29.73
13	14	81.42	40.8	69.67	56.16	17.28
14	15	61.82	52.36	78.74	45.76	24.71
15	16	54.64	55.4	65.51	59.22	49.93
16	17	68.55	38.31	69.71	60.21	49.6
17	18	81.45	39.77	64.08	59.02	17.99
18	19	64.43	42.55	77.62	41.66	48.44
19	20	69.77	45.75	73.11	56.55	12.01
20	21	60.34	38.36	71.27	51.1	41.93
21	22	78.87	20.76	62.12	60.82	43.47
22	23	74.43	35.03	65.64	49.3	14.73
23	24	67.25	31.43	72.38	54.79	22.59
24	25	71.22	36.61	59.88	51.69	16.21

Datasheet2 = (Info Of VIT)



C:/Users/shubham b thorat/ML_VIT/ML_VIT - Spyder (Python 3.6)

Editor - C:\Users\shubham b thorat\ML_VIT\ML_VIT\indian_unver.py

```
@author: shubham b thorat
"""

# -*- coding: utf-8 -*-
"""
Created on Thu Jun 28 16:51:21 2018

@author: shubham b thorat
"""
# importing the libraries
"""
Indian-Institutes_ranking
"""
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# importing the data set
dataset=pd.read_csv('Indian_college_ranking
dataset2=pd.read_csv('VIT_data.csv', sep =
X = dataset.iloc[:,1:5].values
Y = dataset.iloc[:,0].values
X_VIT = dataset2.iloc[:,1:5].values

# dividing dataset into test and training d
from sklearn.model_selection import train_t
X_train, X_test, Y_train, Y_test = train_te

# feature scaling
"""from sklearn.preprocessing import Standa
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.fit_transform(X_test)
X_VIT = sc_X.fit_transform(X_VIT)
"""
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train,Y_train)

#predicting the test set result
y_pred = regressor.predict(X_test)
y_vit = regressor.predict(X_VIT)
```

dataset2 - DataFrame

Index	Rank	TLR	RPC	GO	OI	Perception
0	145	54.67	2.33	64.84	60.38	5.97

☑ Background color ☑ Column min/max

Variable explorer

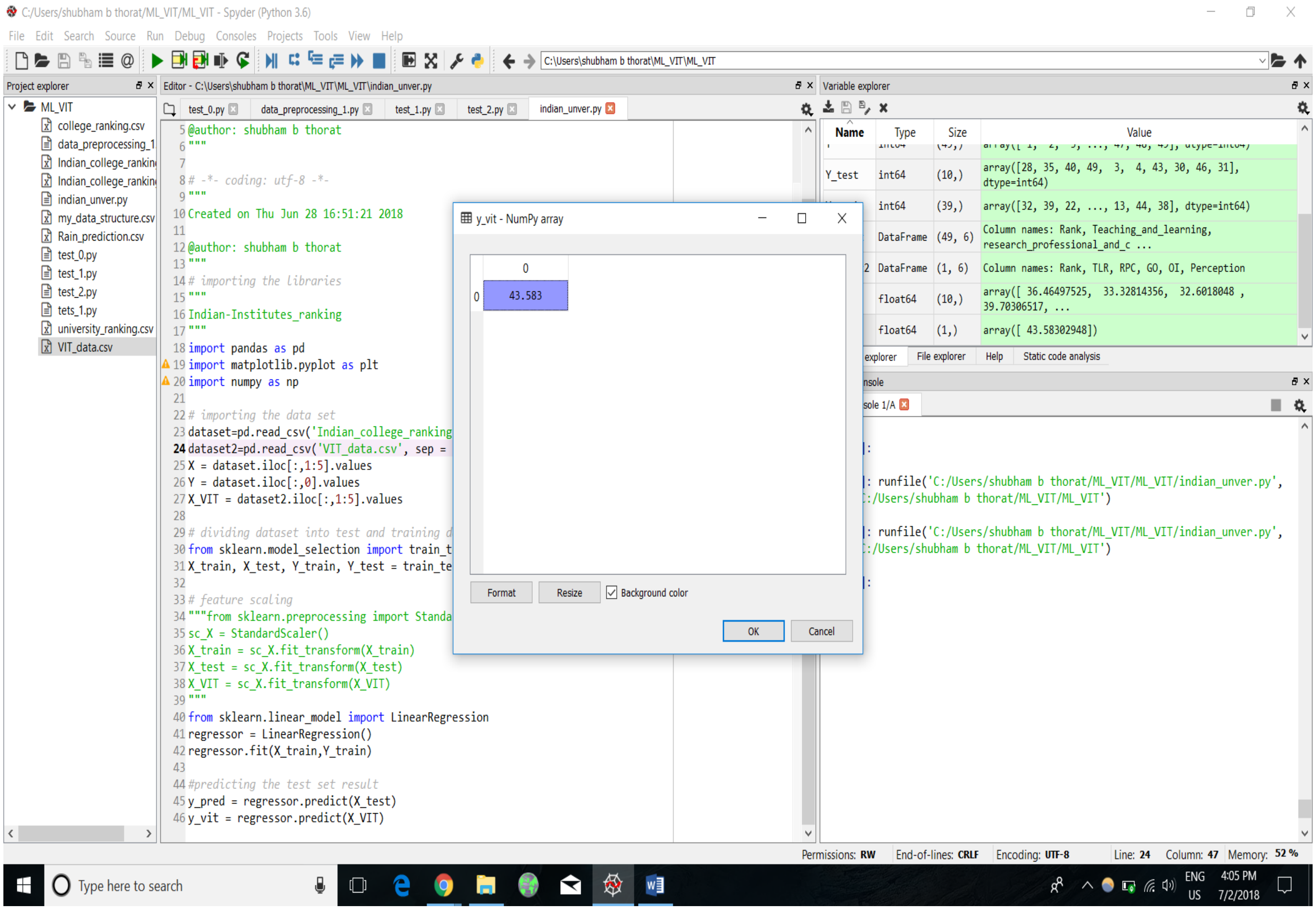
Name	Type	Size	Value
Y_test	int64	(10,)	array([28, 35, 40, 49, 3, 4, 43, 30, 46, 31], dtype=int64)

```
(1, 0)) while a minimum of
1 is required.


In [82]:

In [82]: runfile('C:/Users/shubham b thorat/ML_VIT/ML_VIT/indian_unver.py',
wdir='C:/Users/shubham b thorat/ML_VIT/ML_VIT')

In [83]: runfile('C:/Users/shubham b thorat/ML_VIT/ML_VIT/indian_unver.py',
wdir='C:/Users/shubham b thorat/ML_VIT/ML_VIT')

In [84]:
```

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 24 Column: 47 Memory: 52 %

Rank_of_VIT

Y_VIT = 43.583



```
@author: shubham b thorat
"""

# -*- coding: utf-8 -*-
"""
Created on Thu Jun 28 16:51:21 2018

@author: shubham b thorat
"""
# importing the libraries
"""
Indian-Institutes_ranking
"""
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# importing the data set
dataset=pd.read_csv('Indian_college_ranking
dataset2=pd.read_csv('VIT_data.csv', sep =
X = dataset.iloc[:,1:5].values
Y = dataset.iloc[:,0].values
X_VIT = dataset2.iloc[:,1:5].values

# dividing dataset into test and training d
from sklearn.model_selection import train_t
X_train, X_test, Y_train, Y_test = train_te

# feature scaling
"""from sklearn.preprocessing import Standa
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.fit_transform(X_test)
X_VIT = sc_X.fit_transform(X_VIT)
"""
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train,Y_train)

#predicting the test set result
y_pred = regressor.predict(X_test)
y_vit = regressor.predict(X_VIT)
```

y_vit - NumPy array

	0
0	43.583

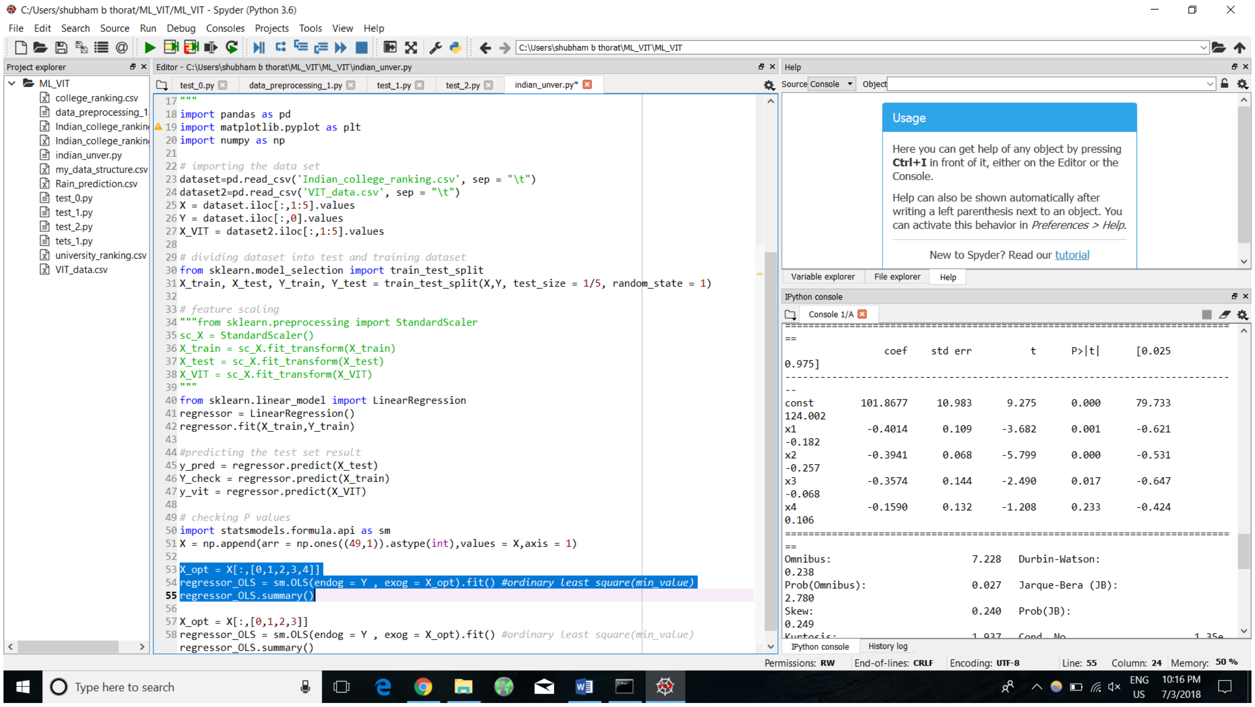
Format | Resize | ☑ Background color | OK | Cancel

Name	Type	Size	Value
Y_test	int64	(10,)	array([28, 35, 40, 49, 3, 4, 43, 30, 46, 31], dtype=int64)
	int64	(39,)	array([32, 39, 22, ..., 13, 44, 38], dtype=int64)
	DataFrame	(49, 6)	Column names: Rank, Teaching_and_learning, research_professional_and_c ...
2	DataFrame	(1, 6)	Column names: Rank, TLR, RPC, GO, OI, Perception
	float64	(10,)	array([36.46497525, 33.32814356, 32.6018048 , 39.70306517, ...
	float64	(1,)	array([43.58302948])

```
]:

]: runfile('C:/Users/shubham b thorat/ML_VIT/ML_VIT/indian_unver.py',
:/Users/shubham b thorat/ML_VIT/ML_VIT')

]: runfile('C:/Users/shubham b thorat/ML_VIT/ML_VIT/indian_unver.py',
:/Users/shubham b thorat/ML_VIT/ML_VIT')

]:
```

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 24 Column: 47 Memory: 52 %

```
# checking P values

import statsmodels.formula.api as sm

X = np.append(arr = np.ones((49,1)).astype(int),values = X,axis = 1)


X_opt = X[:,[0,1,2,3,4]]

regressor_OLS = sm.OLS(endog = Y , exog = X_opt).fit() #ordinary least square(min_value)

regressor_OLS.summary()
```



```
"""
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# importing the data set
dataset=pd.read_csv('Indian_college_ranking.csv', sep = "\t")
dataset2=pd.read_csv('VIT_data.csv', sep = "\t")
X = dataset.iloc[:,1:5].values
Y = dataset.iloc[:,0].values
X_VIT = dataset2.iloc[:,1:5].values

# dividing dataset into test and training dataset
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 1/5, random_state = 1)

# feature scaling
"""from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.fit_transform(X_test)
X_VIT = sc_X.fit_transform(X_VIT)
"""
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train,Y_train)

#predicting the test set result
y_pred = regressor.predict(X_test)
Y_check = regressor.predict(X_train)
y_vit = regressor.predict(X_VIT)

# checking P values
import statsmodels.formula.api as sm
X = np.append(arr = np.ones((49,1)).astype(int),values = X,axis = 1)

X_opt = X[:,[0,1,2,3,4]]
regressor_OLS = sm.OLS(endog = Y , exog = X_opt).fit() #ordinary least square(min_value)
regressor_OLS.summary()

X_opt = X[:,[0,1,2,3]]
regressor_OLS = sm.OLS(endog = Y , exog = X_opt).fit() #ordinary least square(min_value)
regressor_OLS.summary()
```

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

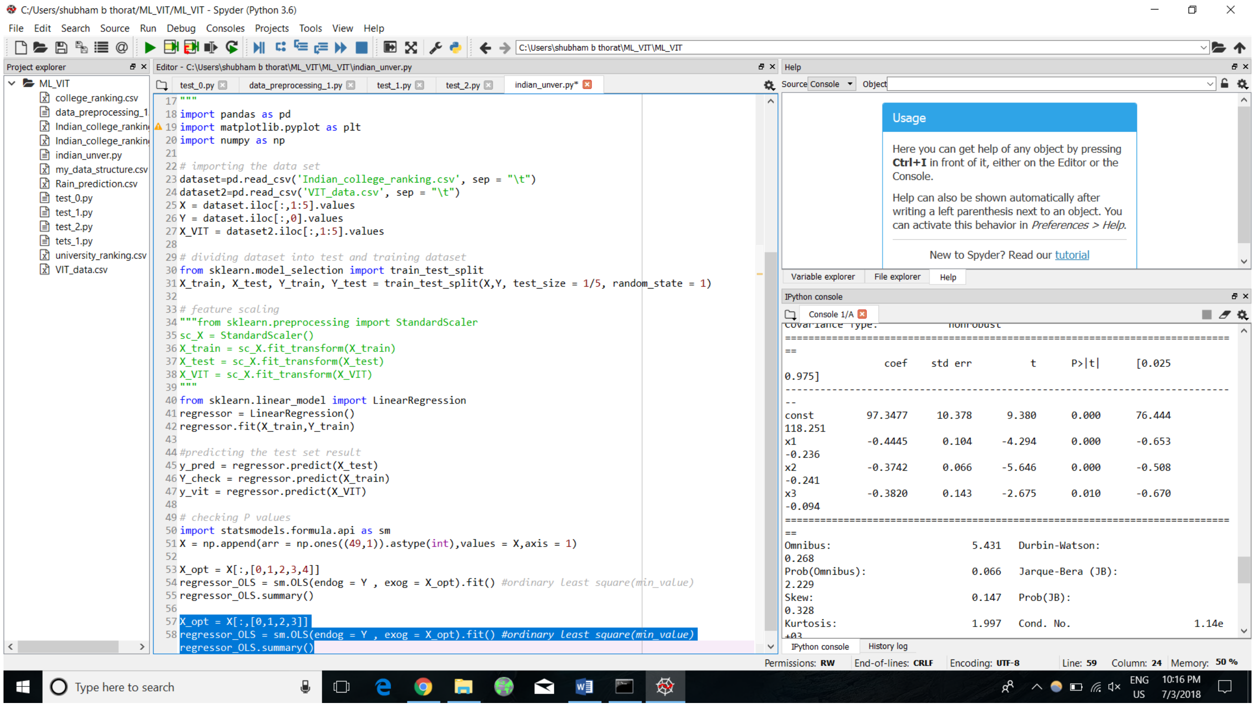
```
==============================================================================
==
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--
const        101.8677     10.983      9.275      0.000      79.733
124.002
x1            -0.4014      0.109     -3.682      0.001      -0.621
-0.182
x2            -0.3941      0.068     -5.799      0.000      -0.531
-0.257
x3            -0.3574      0.144     -2.490      0.017      -0.647
-0.068
x4            -0.1590      0.132     -1.208      0.233      -0.424
0.106
==============================================================================
==
Omnibus:                        7.228   Durbin-Watson:
0.238
Prob(Omnibus):                  0.027   Jarque-Bera (JB):
2.780
Skew:                           0.240   Prob(JB):
0.249
Kurtosis:                       1.937   Cond. No.                       1.35e
```

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 55 Column: 24 Memory: 50 %

X_opt = X[:,[0,1,2,3]]

regressor_OLS = sm.OLS(endog = Y , exog = X_opt).fit() #ordinary least square(min_value)

regressor_OLS.summary()



```
"""
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# importing the data set
dataset=pd.read_csv('Indian_college_ranking.csv', sep = "\t")
dataset2=pd.read_csv('VIT_data.csv', sep = "\t")
X = dataset.iloc[:,1:5].values
Y = dataset.iloc[:,0].values
X_VIT = dataset2.iloc[:,1:5].values

# dividing dataset into test and training dataset
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 1/5, random_state = 1)

# feature scaling
"""from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.fit_transform(X_test)
X_VIT = sc_X.fit_transform(X_VIT)
"""
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train,Y_train)

#predicting the test set result
y_pred = regressor.predict(X_test)
Y_check = regressor.predict(X_train)
y_vit = regressor.predict(X_VIT)

# checking P values
import statsmodels.formula.api as sm
X = np.append(arr = np.ones((49,1)).astype(int),values = X,axis = 1)

X_opt = X[:,[0,1,2,3,4]]
regressor_OLS = sm.OLS(endog = Y , exog = X_opt).fit() #ordinary least square(min_value)
regressor_OLS.summary()

X_opt = X[:,[0,1,2,3]]
regressor_OLS = sm.OLS(endog = Y , exog = X_opt).fit() #ordinary least square(min_value)
regressor_OLS.summary()
```

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Covariance Type:            nonrobust
==============================================================================
==
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--
const         97.3477     10.378      9.380      0.000      76.444
118.251
x1            -0.4445      0.104     -4.294      0.000      -0.653
-0.236
x2            -0.3742      0.066     -5.646      0.000      -0.508
-0.241
x3            -0.3820      0.143     -2.675      0.010      -0.670
-0.094
==============================================================================
==
Omnibus:                        5.431   Durbin-Watson:
0.268
Prob(Omnibus):                  0.066   Jarque-Bera (JB):
2.229
Skew:                           0.147   Prob(JB):
0.328
Kurtosis:                       1.997   Cond. No.                       1.14e
+03
```

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 59 Column: 24 Memory: 50 %