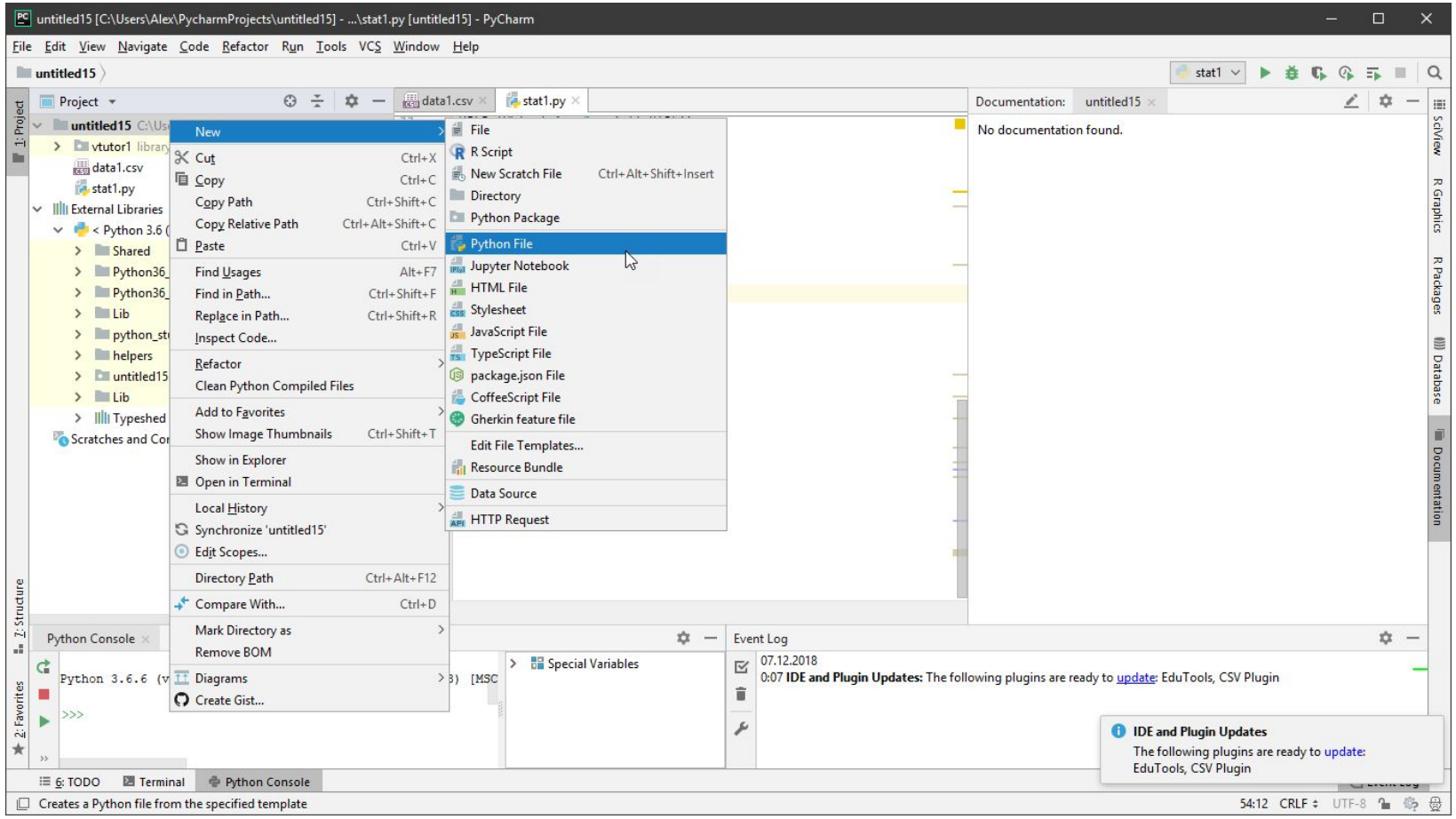
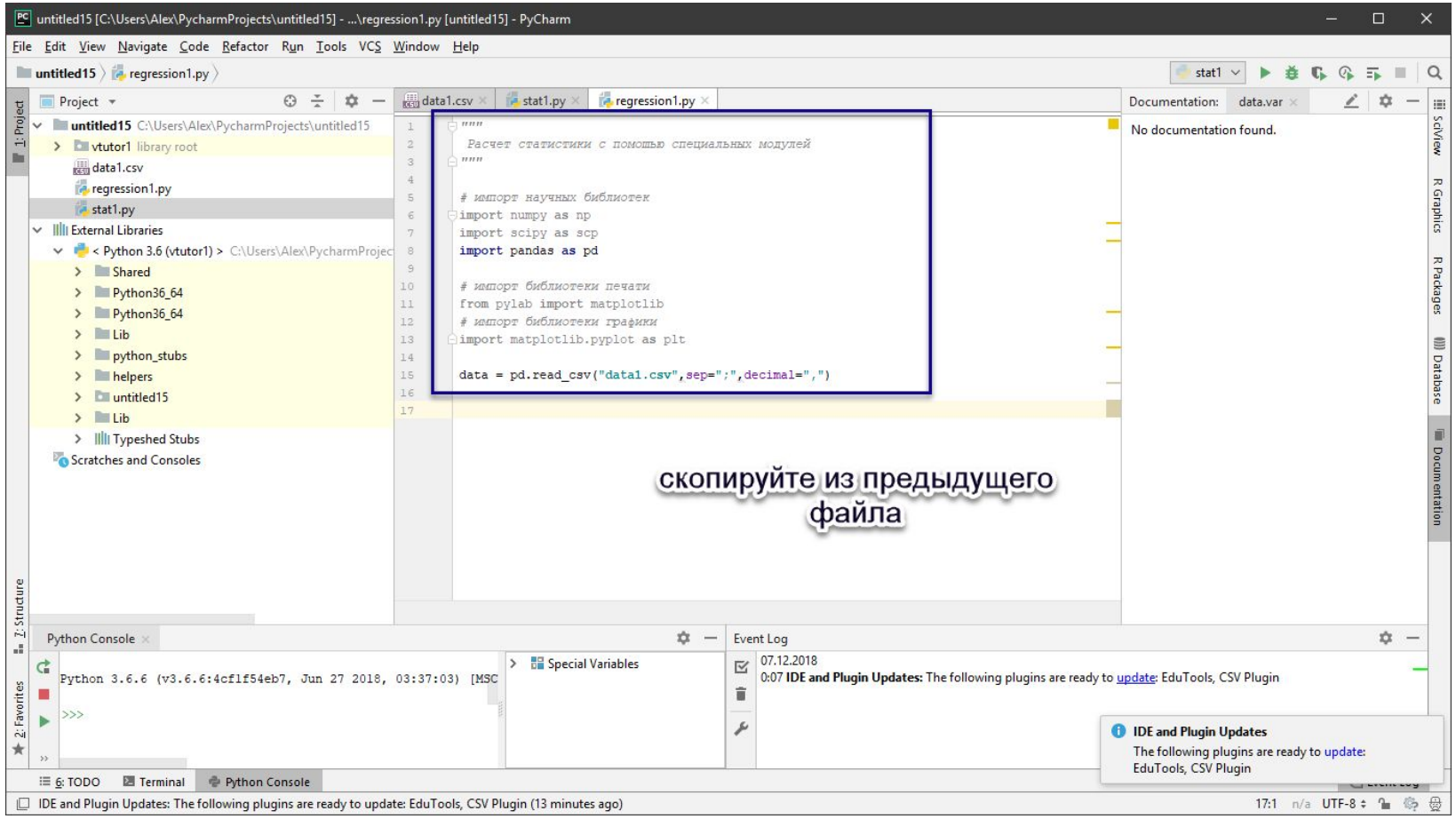

untitled15 [C:\Users\Alex\PycharmProjects\untitled15] - ...\stat1.py [untitled15] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
untitled15
stat1
Project
data1.csv
stat1.py
Documentation: untitled15
No documentation found.
New
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy Relative Path Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Inspect Code...
Refactor
Clean Python Compiled Files
Add to Favorites
Show Image Thumbnails Ctrl+Shift+T
Show in Explorer
Open in Terminal
Local History
Synchronize 'untitled15'
Edit Scopes...
Directory Path Ctrl+Alt+F12
Compare With... Ctrl+D
Mark Directory as
Remove BOM
Diagrams
Create Gist...
File
R Script
New Scratch File Ctrl+Alt+Shift+Insert
Directory
Python Package
Python File
Jupyter Notebook
HTML File
Stylesheet
JavaScript File
TypeScript File
package.json File
CoffeeScript File
Gherkin feature file
Edit File Templates...
Resource Bundle
Data Source
HTTP Request
Python Console
Special Variables
Event Log
07.12.2018
0:07 IDE and Plugin Updates: The following plugins are ready to update: EduTools, CSV Plugin
IDE and Plugin Updates
The following plugins are ready to update:
EduTools, CSV Plugin
6: TODO
Terminal
Python Console
Creates a Python file from the specified template
54:12 CRLF UTF-8

Импортируйте модули (скопируйте с прошлой работы)



```
# импорт научных библиотек
import numpy as np
import scipy as scp
import pandas as pd

# импорт библиотеки печати
from pylab import matplotlib
# импорт библиотеки графики
import matplotlib.pyplot as plt

data = pd.read_csv("data1.csv",sep=";",decimal=",")
```

См. следующий слайд

```
14
15      data = pd.read_csv("data1.csv",sep=";",decimal=",")
16
17      # новый модуль :)
18      # не забудьте установить его
19      import statsmodels.api as sm
20
```

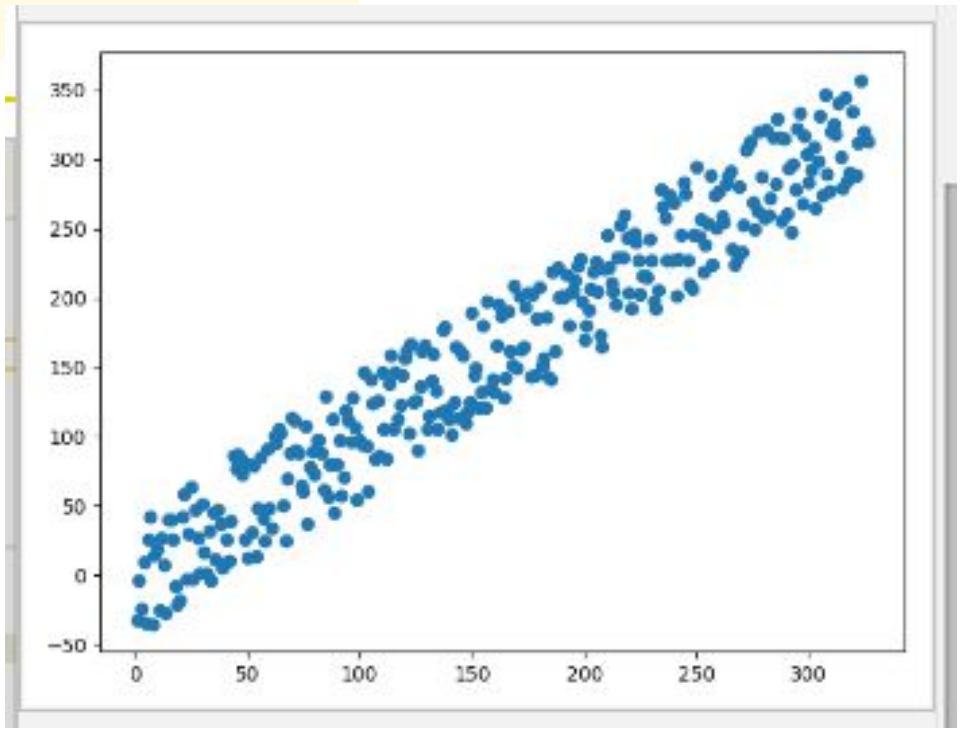
Не «подцепился» модуль из зависимостей, надо его установить

```
# новый модуль :)
# не забудьте установить его
import patsy
import statsmodels.api as sm
```

- Для расчета уравнения регрессии надо указать
  - вид модели (простая регрессия)
  - Аргумент X
  - Отклик – Y

- Конкретно ДАННЫЙ пакет требует дополнительных усилий для расчета константы, поэтому мы сделаем две версии – с константой и без

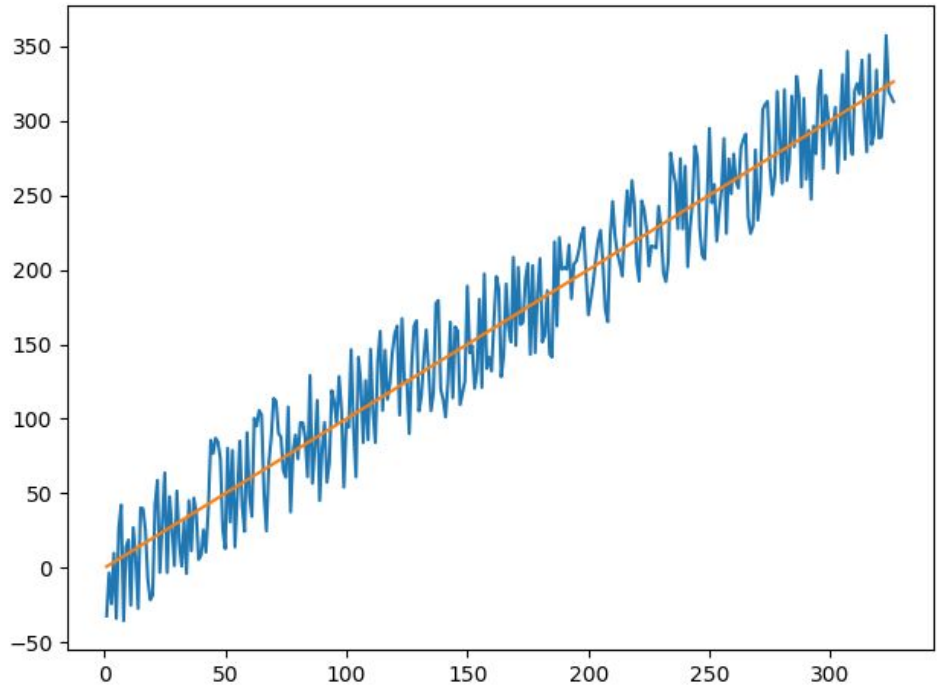
```
19  import patsy
20  import statsmodels.api as sm
21
22  x = data["N"]
23  y = data["z"]
24
25  plt.figure()
26  plt.scatter(x,y)
27  plt.show()
```



Без коэффициента

```
# создаем модель и рассчитываем ее
model = sm.OLS(y,x).fit()
# рассчитываем "предсказания по модели"
yl = model.predict(x)

plt.plot(x,y,x,yl)
plt.show()
```



… и небольшой отчет

```
# создаем модель и рассчитываем ее
model = sm.OLS(y,x).fit()
# рассчитываем "предсказания по модел
y1 = model.predict(x)

plt.plot(x,y,x,y1)
plt.show()

print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      z   R-squared:                       0.981
Model:                            OLS   Adj. R-squared:                  0.981
Method:                 Least Squares   F-statistic:                 1.695e+04
Date:                Fri, 07 Dec 2018   Prob (F-statistic):          1.82e-282
Time:                        10:21:57   Log-Likelihood:                -1526.3
No. Observations:                 326   AIC:                             3055.
Df Residuals:                     325   BIC:                             3058.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
N              1.0001      0.008    130.183      0.000       0.985       1.015
==============================================================================
Omnibus:                      252.856   Durbin-Watson:                   1.995
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               21.219
Skew:                           0.032   Prob(JB):                     2.47e-05
Kurtosis:                       1.752   Cond. No.                         1.00
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

>>>
```

```
# добавляем константу
x2 = x # серия с константой будет немного испорчена, поэтому делаем же
x2 = sm.add_constant(x2)
modell = sm.OLS(y,x2).fit()
y2 = modell.predict(x2)

plt.plot(x,y,x,y1,x,y2)
plt.title("x2,y2")
plt.show()

# найдите разницу в отчетах
print(modell.summary())

plt.plot(x,y1-y2)
plt.title("Ошибка в расчетах по моделям с константой и без")
plt.show()
```

Отчет по модели с константой

OLS Regression Results

Dep. Variable:	z	R-squared:	0.928
Model:	OLS	Adj. R-squared:	0.928
Method:	Least Squares	F-statistic:	4192.
Date:	Fri, 07 Dec 2018	Prob (F-statistic):	2.03e-187
Time:	10:47:03	Log-Likelihood:	-1526.3
No. Observations:	326	AIC:	3057.
Df Residuals:	324	BIC:	3064.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.3387	2.910	0.116	0.907	-5.386	6.063
N	0.9986	0.015	64.742	0.000	0.968	1.029

Omnibus:	251.446	Durbin-Watson:	1.995
Prob(Omnibus):	0.000	Jarque-Bera (JB):	21.190
Skew:	0.031	Prob(JB):	2.50e-05
Kurtosis:	1.753	Cond. No.	378.

График ошибки

И еще один модуль

- Полное название модуля: scikit-learn
- Но импортное имя – sklearn
- Модуль предназначен для решения задач машинного обучения и регрессионные модели – только часть его возможностей

Импорт модуля расчета линейной регрессии

```
54      plt.show()
55
56      # и еще немного :)
57      from sklearn import linear_model
58
```

- К сожалению, новый пакет ожидает новых данных, точнее данных в новом формате, поэтому мы «трансформируем» иксы в правильном порядке.
- Если быть пунктуальным, то это связано с тем, что модуль более мощный, и он рассчитывает, фактически, множественную регрессию

```
# создаем объект, способный рассчитать регрессию
lm = linear_model.LinearRegression()
# рассчитываем модель
xx = x.values.reshape(-1,1)
model3 = lm.fit(xx,y)

print(model3.coef_)

plt.plot(xx,model3.predict(xx),xx,y)
plt.show()
```