STATA Features

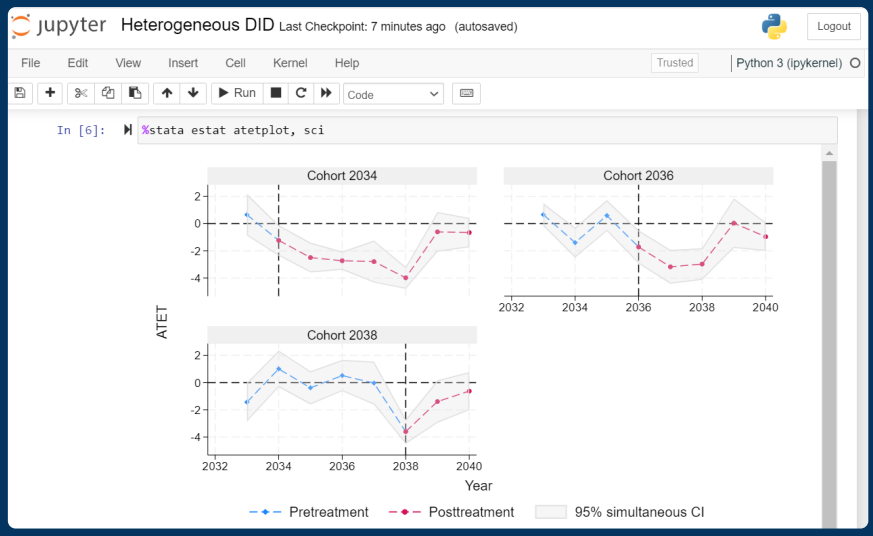
PyStata—Python and Stata Integration

Call Stata from Python and call Python from Stata

Call Stata from Python

Access Stata from Python environments such as

- Jupyter Notebook and Lab
- Spyder IDE
- PyCharm IDE
- Windows Command Prompt
- macOS terminal
- Unix terminal



Work with IPython magic commands or API functions

Use magic commands in any IPython kernel-based environment

Run one Stata command with line magic

```
%stata regress y x1 x2 i.a##i.b
```

Or run a block of Stata commands with cell magic

```
%%stata
regress y x1 x2 i.a##i.b
margins a#b
predict yhat
```

Add options to cell magic to transfer data and results and alter settings.

Load Python data into Stata

```
%%stata -d mydata
```

Load Stata data into Python

```
%%stata -doutd newdata
```

Load Stata results into Python

```
%%stata -ret r -eret e
```

Use API functions anywhere you access Python

Load Python data into Stata

```
stata.pdataframe_to_data(mydata)
```

Run Stata commands in Python

```
stata.run("regress y x1 x2 i.a##i.b")
```

Or run a block of Stata commands

```
stata.run('''
regress y x1 x2 i.a##i.b
margins a#b
predict yhat
''')
```

Load Stata data into Python

```
newdata = stata.pdataframe_from_data()
```

Load Stata results into Python

```
r = stata.get_return()
e = stata.get_ereturn()
```

Add options to suppress output, use a subset of observations or variables, and more!

Call Python from Stata

- **Use Python packages within Stata**
  - Matplotlib for 3-D graphs
  - Scrapy for scraping data
  - scikit-learn for machine learning
  - TensorFlow for deep learning
  - Much more
- **Use the Stata Function Interface (sfi) Python module**
  - Access Stata's core features in Python, including data, frames, macros, scalars, matrices, value labels, characteristics, and more
  - Store Python results back into Stata

Invoke Python interactively

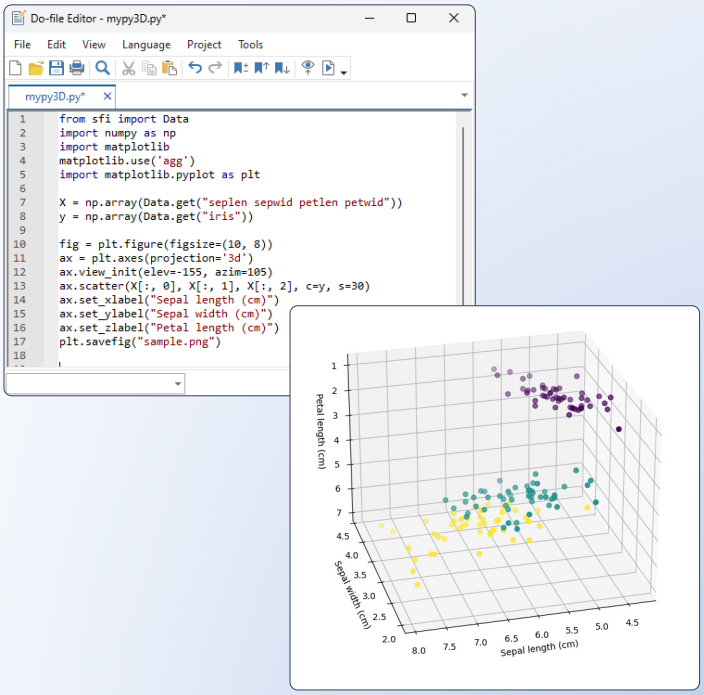
Type **python** in the Stata Command Window to enter the Python environment.

```
. python
                                    python (type end to exit)
>>> import numpy as np
>>> b = np.arange(6).reshape(2,3)
>>> b
array([[0, 1, 2],
       [3, 4, 5]])
>>> b.sum(axis=0)
array([3, 5, 7])
>>> end
```

Run a Python script within Stata

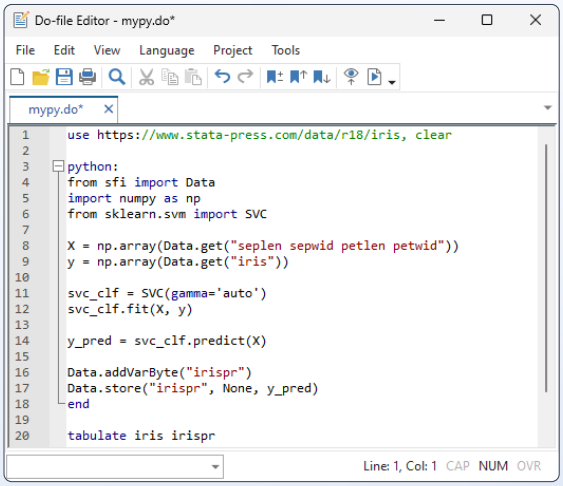
Use the **python script** command to run a Python script directly from Stata.

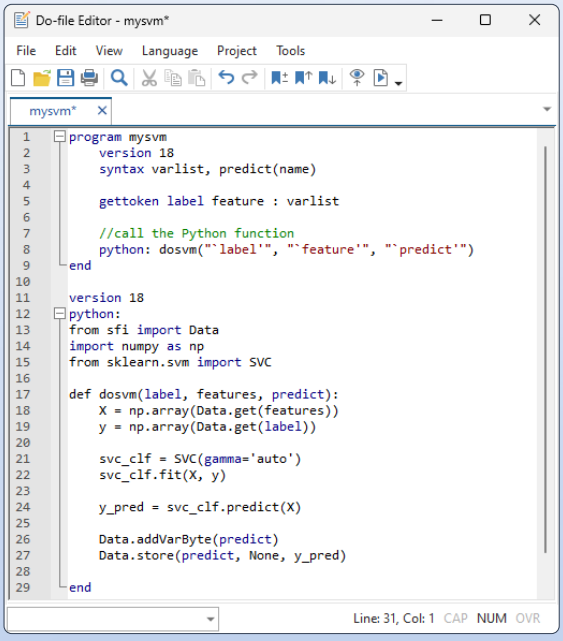
```
. use https://www.stata-press.com/data/r18/iris.dta
. python script mypy3D.py
```



Embed Python code in a Stata script (do-file) or program (ado-file)

Combine Stata code with Python code in your do-files and ado-files.






```
. use https://www.stata-press.com/data/r18/iris.dta
. mysvm iris seplen sepwid petlen petwid,
        predict(irispr)
. tabulate iris irispr
```

stata.com/pystata