

New in

STATA®

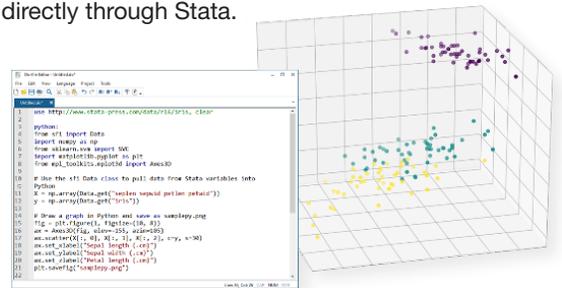
16

PYTHON INTEGRATION

- **Python integration with Stata**
 - » Invoke Python interactively
 - » Embed Python code in a do-file
 - » Run a Python script file within Stata
 - » Embed Python code in an ado-file
 - » Run Stata commands within Python
- **Use any Python package within Stata such as**
 - » Matplotlib for 3-D graphs
 - » Scrapy for scraping data
 - » Scikit-learn for machine learning
 - » Much more
- **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

Stata's new **python** command allows you to easily embed and execute Python code from within Stata and to output Python results in Stata and store results from Python in Stata.

You can invoke Python interactively or in scripts (do-files) and programs (ado-files) so that you can leverage Python's extensive features. You can also execute a Python script file (.py) directly through Stata.



Invoke Python interactively

You can 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
```

Embed Python code in a Stata program

Python code can be embedded and executed in Stata programs (ado-files) too.

```
Do-file Editor - mysvm.ado
File Edit View Language Project Tools
mysvm.ado x
1 program mysvm
2 version 16
3 syntax varlist, predict(name)
4
5 gettoken label feature : varlist
6
7 //call the Python function
8 python: dosvm("`label'", "`feature'", "`predict'")
9 end
10
11 version 16
12 python:
13 from sfi import Data
14 import numpy as np
15 from sklearn.svm import SVC
16
17 def dosvm(label, features, predict):
18     X = np.array(Data.get(features))
19     y = np.array(Data.get(label))
20
21     svc_clf = SVC(gamma='auto')
22     svc_clf.fit(X, y)
23
24     y_pred = svc_clf.predict(X)
25
26     Data.addVarByte(predict)
27     Data.store(predict, None, y_pred)
28
29 end
Line: 32, Col: 1 CAP NUM OVR
```

Embed Python code in a Stata script

You can also embed Python code in a Stata script (do-file) to intermix Python code with Stata code.

```
Do-file Editor - mpyy.do
File Edit View Language Project Tools
mpyy.do x
1 use http://www.stata-press.com/data/r16/iris, clear
2
3 python:
4 from sfi import Data
5 import numpy as np
6 from sklearn.svm import SVC
7
8 X = np.array(Data.get("seplen sepwid petlen petwid"))
9 y = np.array(Data.get("iris"))
10
11 svc_clf = SVC(gamma='auto')
12 svc_clf.fit(X, y)
13
14 y_pred = svc_clf.predict(X)
15
16 Data.addVarByte("irispr")
17 Data.store(irispr, None, y_pred)
18 end
19
20 tabulate iris irispr
Line: 23, Col: 1 CAP NUM OVR
```

```
. use http://www.stata-press.com/data/r16/iris, clear
(Iris data)

. mysvm iris seplen sepwid petlen petwid, predict(irispr)

. tabulate iris irispr
```

Stata Function Interface (sfi) Python module

The **sfi** module provides a bidirectional connection between Stata and Python. It allows you to read from and write to Stata's current dataset, frames, macros, scalars, matrices, value labels, characteristics, global Mata matrices, and more from within Python. This module can be used interactively, in scripts (do-files), in programs (ado-files), and in Python script files.

```
Do-file Editor - mypy2.do
File Edit View Language Project Tools

mypy2.do x
1 clear
2
3 python:
4 import urllib.request
5 from bs4 import BeautifulSoup
6 from sfi import Data
7
8 url = "https://www.usclimatedata.com/climate/" \
9       "college-station/texas/united-states/ustx2165/2018/2"
10 page = urllib.request.urlopen(url)
11
12 soup = BeautifulSoup(page.read(),'lxml')
13 table = soup.findAll('table')[5]
14
15 Data.addVarStr("Day", 11)
16 Data.addVarFloat("High")
17 Data.addVarFloat("Low")
18
19 count = 0
20 for tr in table.findAll('tr'):
21     td = tr.findAll('td')
22     if len(td)>0:
23         Data.setObsTotal(count+1)
24         Data.storeAt(0, count, td[0].getText())
25         Data.storeAt(1, count, float(td[1].getText()))
26         Data.storeAt(2, count, float(td[2].getText()))
27         count = count + 1
28 end
29
30 summarize High Low
```

```
Viewer - view ex3.smcl
File Edit History Help

view ex3.smcl x
Dialog Also see Jump to

python:
python (type end to exit)
>>> import urllib.request
>>> from bs4 import BeautifulSoup
>>> from sfi import Data
>>>
>>> url = "https://www.usclimatedata.com/climate/" \
...      "college-station/texas/united-states/ustx2165/2018/2"
>>> page = urllib.request.urlopen(url)
>>>
>>> soup = BeautifulSoup(page.read(),'lxml')
>>> table = soup.findAll('table')[5]
>>>
>>> Data.addVarStr("Day", 11)
>>> Data.addVarFloat("High")
>>> Data.addVarFloat("Low")
>>>
>>> count = 0
>>> for tr in table.findAll('tr'):
...     td = tr.findAll('td')
...     if len(td)>0:
...         Data.setObsTotal(count+1)
...         Data.storeAt(0, count, td[0].getText())
...         Data.storeAt(1, count, float(td[1].getText()))
...         Data.storeAt(2, count, float(td[2].getText()))
...         count = count + 1
>>> end

. summarize High Low

Variable | Obs   Mean   Std. Dev.   Min   Max
-----+-----+-----+-----+-----+-----
High    | 28  64.07143  12.60484   37.9   82.9
Low     | 28  48.39286  11.1752   30.2   70
```

Run a Python script file within Stata

You can use **python script** to run a Python script file (.py) directly from Stata.

```
Do-file Editor - mypy.py
File Edit View Language Project Tools

mypy.py x
1 from sfi import Data
2 import numpy as np
3 import matplotlib
4 matplotlib.use('agg')
5 import matplotlib.pyplot as plt
6 from mpl_toolkits.mplot3d import Axes3D
7
8 X = np.array(Data.get("seplen sepwid petlen petwid"))
9 y = np.array(Data.get("iris"))
10
11 fig = plt.figure(1, figsize=(10, 8))
12 ax = Axes3D(fig, elev=-155, azim=105)
13 ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=y, s=30)
14 ax.set_xlabel("Sepal length (.cm)")
15 ax.set_ylabel("Sepal width (.cm)")
16 ax.set_zlabel("Petal length (.cm)")
17 plt.savefig("samplepy.png")
18
```

- use `http://www.stata-press.com/data/r16/iris`, clear (Iris data)
- `python script mypy.py`

