

PyStata intro — Introduction to using Python and Stata together[Description](#) [Also see](#)

Description

Stata provides two ways for Python and Stata to interact, and we refer to these mechanisms collectively as PyStata.

First, Python can be invoked from a running Stata session so that Python's extensive language features can be leveraged from within Stata. We call this Python integration. With Python integration, you can embed and execute Python code interactively, or in do-files and ado-files. You can read more about using Python from within Stata in [\[P\] PyStata integration](#).

Second, Stata can be invoked from a standalone Python environment via the [pystata](#) Python package. It includes three [IPython](#) (interactive Python) magic commands and a suite of API functions for interacting with Stata from within Python. With these tools, you can access Stata and Mata conveniently in an IPython kernel-based environment (for example, Jupyter Notebook and console, and Jupyter Lab and console), in other environments that support the IPython kernel (for example, Spyder IDE and PyCharm IDE), or when accessing Python from a command line (for example, Windows Command Prompt, macOS terminal, Unix terminal, and Python's IDLE). See [\[P\] PyStata module](#) for more details about calling Stata from within Python.

Whether you are integrating Python into Stata or Stata into Python, you can use the [sfi](#) (Stata Function Interface) module to interact Python's capabilities with Stata's core features. Within the module, classes are defined to provide access to Stata's current dataset, frames, macros, scalars, matrices, value labels, characteristics, global Mata matrices, and more. Refer to Stata's [Python API documentation](#) for more details.

Also see

[\[P\] PyStata integration](#) — Call Python from Stata

[\[P\] PyStata module](#) — Python package `pystata` to call Stata from Python