Everything is better with friends: Executing SAS® code in Python scripts with SASPy, and turbocharging your SAS programming with open-source tooling

JOIN GITHUB

If you don't already have a GitHub account, please begin by visiting https://github.com/ and create one. We recommend a free account, which will be sufficient for the purposes of this class, but you can also sign up for a Pro account, which provides additional features.

We also recommend spending a few minutes working through each of the following to gain familiarity with the Git/GitHub vocabulary and workflows emphasized in the class:

- **GitHub "Hello World" Hands-on Exercise**
  https://guides.github.com/activities/hello-world/

- **GitHub flow overview**
  https://guides.github.com/introduction/flow/

- **GitHub Forking Project Hands-on Exercise**
  https://guides.github.com/activities/forking/

JOIN OUR GITTER COMMUNITY

Visit https://gitter.im/saspy-bffs/community. Then sign in using your GitHub account credentials. Then join the Community using the "Add" button in the upper-right corner of the Gitter interface. Once you've joined the community, email us with your GitHub/Gitter username, and we'll add you to a Private Room created specifically for this class.

While Gitter can be accessed in a web browser window, we recommend installing the dedicated desktop and mobile apps available at https://gitter.im/apps. This allows you to leave an app running in the background and receive push notifications for new messages without worrying about accidentally closing a browser tab.

INSTALL REQUIRED SOFTWARE

To setup the development environment required for hands-on practice during the class, please download and install the following completely free software in the order specified:

1. The latest version of Java SE 8 from https://www.java.com/
2. The latest version of Python 3.7 from https://www.python.org/downloads/
3. The latest version of Git from https://git-scm.com/downloads/
4. The latest version of PyCharm Community Edition (or, if preferred, the non-free Professional Edition) from https://www.jetbrains.com/pycharm/download/

Note: You will also need SAS 9.4 (not SAS University Edition) for SASPy examples, which we recommend be installed locally. In addition, we will be using SASHELP datasets.

CONFIGURE SASPY

To try out and configure the development environment required for hands-on practice during the class (especially SASPy), please complete these steps in order:

1. Start PyCharm, and when prompted, choose **Check out from Version Control → Git**, enter URL https://github.com/saspy-bffs/dataset-explorer, choose a directory to copy the files to, and click **Clone**.
2. Use the menu commands **File → Settings → Project: dataset-explorer → Project Interpreter**. Then click the **Gear Icon** (□) in the upper-right corner of the dialog box and select **Add…** This should prompt you to create a new virtual environment[1] as a subfolder named *venv* in your project folder. Once setup, click **OK** and, once processing has finished, click **OK** again to exit the dialog box.
3. Use the menu command **View → Tool Windows → Terminal** to open a terminal window. (Alternatively, click **Terminal** at the bottom of the PyCharm window.) Then type the following at the command prompt and press Enter: **pip install -r requirements.txt**
4. Use the project-navigation area in the left-hand panel to open the file *sascfg_personal-example.py* (i.e., double-click its name), and copy its contents to the system clipboard. Then use the menu command **File → New → Python File** to create a new file named *sascfg_personal.py*, paste the contents of the system clipboard into it, and update to match your SAS installation setup per the instructions at https://sassoftware.github.io/saspy/install.html.

   Warning: This step is **not** straightforward, and we can only provide support for configuring SASPy with a local installation of SAS running under Windows. See also https://github.com/sassoftware/saspy-examples/blob/master/SAS_contrib/autocfg.ipynb. You can also contact the developers of SASPy for installation support with other platforms using https://github.com/sassoftware/saspy/issues.
5. Finally, to verify SASPy has been configured correctly, use the menu command **Run → Run … → app**, and look for the text "Running on http://127.0.0.1:8000/" in the Run portion of the bottom panel. You can then either visit this URL in a web browser or click the link in the output[2]. (Alternatively, open the file *app.py* by double-clicking its name in the project-navigation area in the left-hand panel, right-click anywhere inside the code editor window, and choose the command **Run 'app'**).

   The class will focus on the skills and background knowledge needed to contribute to open-source projects like dataset-explorer on GitHub, or even to build your own.

WHAT ABOUT ANACONDA?

An alternative Python implementation for data-science applications can also be installed as part of the Anaconda distribution (https://www.anaconda.com/distribution/), which includes many popular data-science packages like pandas. However, the Anaconda distribution has its own separate form of virtual environment called a *conda environment*.

[1] A *virtual environment* (aka *venv*) is essentially a completely separate installation of Python. It's considered best practice to create a new venv for each project in order to keep its dependencies isolated.

[2] To try out the web app, try the directory **C:\Program Files\SASHome\SASFoundation\9.4\core\sashelp** if SAS was installed on your local machine with default options.