PYTHON: From Your Computer (Jupyter/Spyder)

Learn how to access WRDS data from Python on your computer

Introduction to Python at WRDS

Python is a widely-used high-level programming language that is both powerful and easy to use, and is proving to a major player in large-scale data analytics applications.

WRDS provides a direct interface for Python access, allowing native querying of WRDS data right within your Python program. All WRDS data is stored in a PostgreSQL database, and is available through Python through our in-house Python module, **wrds**, which is freely available on PyPI via a **pip install**.

Here is an example of a simple query against the Dow Jones Averages & Total Return Indexes using the **wrds** Python module:

```
1 import wrds
2 db = wrds.Connection(wrds_username='joe')
3 db.raw_sql('SELECT date, dji FROM djones.djdaily')
```

Full usage and many examples are given throughout this section for using Python to access WRDS data remotely from your computer.

Note: Class accounts and IPAuth / Daypass accounts are not permitted to access WRDS in this manner and will receive an error if trying this connection. You must have your own, dedicated WRDS account in order to access WRDS from Python.

Alternatively, if you are interested in using Python on the WRDS Cloud to access WRDS data directly on our high-performance computing cluster, please reference PYTHON: On the WRDS Cloud

PostgreSQL vs SAS

In the past, WRDS data was only available as SAS flat files. In this format, these files were accessible to non-SAS clients (such as Python) only via a JDBC interface. While this generally worked well, it proved problematic for memory-intensive programs, and proved impossible for extremely large datasets such as TAQ.

WRDS has now made almost all of our data additionally available as a series of PostgreSQL databases, which has significantly opened up research options for other, non-SAS programming languages such as Python, R, and MATLAB. Java and JDBC drivers are no longer required for Python connectivity, as a native Python **wrds** module exists to facilitate the connection. Compared to the previous JDBC connection method, the native method is faster, more robust, and capable of handling far larger queries.

Please let us know what you think of the new connection method, and how you're using it in your research!

Top of Section

Initial Setup - Installing Python and Jupyter / Spyder

To access WRDS data in Python, a minimal amount of setup is required. WRDS recommends **Python 3** and either **Jupyter** or **Spyder** for working with WRDS data in Python. While it is possible that your workstation already has a distribution of Python (especially if Mac OSX), WRDS recommends installing your own, dedicated version of Python to use as your research computing distribution, separate from your system's installation. Among other reasons, this ensures that you can maintain the packages in your distribution without your operating system overwriting, replacing, or upgrading your chosen setup, and potentially breaking your Python code.

While you could install Python itself on your own, WRDS highly recommends the **Anaconda Python Distribution**, available for free online, which includes most of the popular scientific and research programming Python modules that we'll need, such as *pandas, numpy, matplotlib, plotly, psycopg2*, etc. Further, Anaconda includes both Jupyter *and* Spyder, making the Anaconda installation far and away the easiest way to get started with Python and WRDS on your workstation. You're free to pursue a Python installation on your own if you wish, but this document will focus on installing Python, Jupyter, and Spyder via Anaconda.

Anaconda can be downloaded for free here:

- **Anaconda for Mac OSX:** Download
- **Anaconda for Windows:** Download

WRDS recommends the Python 3.6 version. Once downloaded, run the installer. When prompted to select a directory to install to, pick a central, permanent location. WRDS recommends a dedicated subdirectory in your *Documents* folder, such as the following:

- **OSX:** */Users/username/Documents/Anaconda*
- **Windows:** *C:\Users\username\Documents\Anaconda*

During the installation, you may be presented with other options for the installation. Leave all options at their default settings unless you specifically know you need to change them.

Once installed, you'll be able to use the Anaconda distribution of Python -- including many popular research computing Python modules as well as Jupyter and Spyder -- from the comfort of your own Workstation. For Mac OSX users, you can access this on the command line in *Terminal* by running: **python**. For Windows users, you can access this through the *Anaconda Prompt* application by running: **python**.

Since Anaconda includes Jupyter and Spyder already, we can move on to the configuration.

Top of Section

Initial Setup - Installing the WRDS Python Module

Now that we've installed Python, the next step is to install the Python **wrds** module. The **wrds** module is available for free on PyPI, the online, central repository of all Python packages. Installing it is easy!

- **For Mac OSX:** open Terminal like you did before.
- **For Windows:** launch the *Anaconda Prompt* app you just installed.

Once you have this open, run the following command:

```
1  pip install wrds
```

This command will download and install the **wrds** Python module, as well as any required dependancies.

The **wrds** module is open source! Check out our GitHub Repository.

If you run into any errors with the installation, please take note of your error and contact us at WRDS Support.

Top of Section

Initial Setup - The pgpass File

Now that we've installed Python and the **wrds** module, the next step is to setup a **pgpass** file on your workstation. The **pgpass** file includes your WRDS username and password so that you do not need to enter them each time you wish to connect to WRDS within Python. With the **wrds** module, creating this file is easy!

First, start **python** on your workstation. Then enter the following Python steps:

```
1  import wrds
2  db = wrds.Connection(wrds_username='joe')
3  db.create_pgpass_file()
```

Where `wrds_username` is your own WRDS username (the same as your login to the WRDS website). You will be prompted once for your WRDS username and password on your first login, at the `Connection()` step, but then, after running `create_pgpass_file()` once, you should be able to connect from then on without needing to do so. Test this by disconnecting and reconnecting, using the following:

```
1  db.close()
2  db = wrds.Connection(wrds_username='joe')
```

You should be connected directly without needing to authenticate. If your WRDS username and your local computer username (that you're running Python as) happen to be the same (in this case, both *joe*), you may omit the `wrds_username` argument to `Connection()`.

Top of Section

Verifying your Setup

Before we move on to accessing WRDS data, let's briefly verify that everything is installed and set up properly.

Having installed Anaconda, we have the **python** shell itself available to us, as well as two IDEs on top of Python: **Jupyter** and **Spyder**. You should be able to call all three from the respective command line app on your platform. Let's try that now as a test:

For Mac OSX users, you use the *Terminal*.
For Windows users, you use the *Anaconda Prompt* application.

Regardless of platform, the commands are the same. Try running the following commands to test that they work:

```
python
jupyter notebook
spyder
```

You will need to quit out of each and return to the command line interface to test the next one.

Once you've verified that these all work, in *any one of them*, try importing the **wrds** module to make sure it works as well:

```
import wrds
```

You must be in a Python shell to use the above Python command. Here's a quick example of a Mac OSX user testing each of the above, and testing the **import wrds** command in the standard Python shell as the final step:

```
mylaptop$ python
Python 3.7.0 | Anaconda custom (x86_64) ...
>>> exit()
mylaptop$ jupyter notebook
[I 17:14:10.815 NotebookApp] Serving notebooks from ...
^C
Shutdown this notebook server (y/[n])? y
mylaptop$ spyder
<Spyder opens successfully, user quits it>
mylaptop$ python
Python 3.7.0 | Anaconda custom (x86_64) ...
>>> import wrds
>>>
```

Here, the user starts a **python** shell successfully, then exits it with the **exit()** Python command. Next, the user starts up a **jupyter notebook**, which opens up in a web browser, then cancels it by issuing the command **^C**, or CTL-C. Next, the user starts up **spyder**, which launches the Spyder graphical IDE, then quits the program. Lastly, the user starts a **python** shell once more, and runs the **import wrds** Python command, to prove it works. As the shell returns to an empty prompt without any errors, the **import** step succeeded.

Using Jupyter and Spyder, as well as the **wrds** module, are covered later in this documentation.

Top of Section

Next Steps

Now that you've installed Python, installed the **wrds** module, and setup your **pgpass** file, you're ready to connect to WRDS and start accessing data.

Please proceed to the next section to learn how to write Python programs that make use of WRDS data remotely from your workstation.

Next section: Querying WRDS Data using Python.

Additional Resources:

- Using Python on WRDS Platform

Top of Section

Top

Table of Contents

- Next Steps

Related Information

- PYTHON: On the WRDS Cloud
- Querying WRDS Data
- Graphing Data
- Example Workflow
- Python Replications