The George Washington University
Department of Statistics

STAT 6197 – Spring 2020

Week 14 – April 24, 2020

Major Topic: SASPy and Other Python Packages/Modules

Detailed Topics:

1) JupyterLab
2) SAS Kernel
3) SASPy
4) Pandas

SASPy and Jupyter Notebook

A Complete Introduction to SASPy and Jupyter Notebooks. By Jason Phillips, 2019

Python-izing the SAS Programmer. By Mike Molter. PharmaSUG 2019 - Paper AP-212

Connecting to Datasets through Python and SAS® By Matise, J. MWSUG, 2019

Integrate Python with SAS® using SASPy for a Simpler, More Effective Script By Vickery J. SESUG Paper 152-2019

Everything is better with friends: Executing SAS® code in Python scripts with SASPy. By Lankham, I and Slaughter, M. WUSS. 2019

Machine learning with SASPy: Exploring and preparing your data (part 1) SAS Blogs, Sep 9, 2019

Machine learning with SASPy: Exploring and preparing your data (part 2), SAS Blogs, Sep 27, 2019

Learning Data Science with SAS® University Edition and JupyterLab (SGF, 2019)

STEPPING UP YOUR SAS® GAME WITH JUPYTER NOTEBOOKS (SGF, 2019)

JupyterLab

- A product of an integrated environment for interactive programming - single interface that can tie many things including
  - Code
  - Text (markdown)
  - Equation
  - Output
  - Visualization, etc.

- A product of Project Jupyter
- A successor of Jupyter Notebook (formally known as IPython Notebook) with more functionality
  - originally supporting open source scientific languages like **Julia, Python and R**
  - now supporting **SAS** via SAS kernel and many other languages
- Web-based and typically used from a single machine
- An alternative to the SAS Studio Interface in SAS University Edition
- An alternative to the SAS Windowing Environment Interface

See JupyterLab User Cheatsheet here.

See JupyterLab Documentation here.

SASPy

- Python Application Programming Interface (API) to the SAS System
- Can load a SAS data set into a Pandas dataframe by using the SASPy module
- Can load a Pandas dataframe into SAS data set
- Communicates between Jupyter and SAS when using the SAS kernel
- Can create a SAS session and send code to it for execution provided you have SAS installed in your local machine
- Can generate SAS code
- Can imitate the SAS macro facility
- Can map many SAS procedures to Python methods

See the SASPy Documentation here.

Important Notes:

If you are using a licensed version of SAS, you need to install

- SASPy
- SAS Kernel
- Anaconda Distribution of Python that includes JupyterLab

Here are the instructions for Installing SASPy Kernel for Jupyter Notebooks and Jupyter Lab.

If you are using the SAS University Edition, you don't need to install the above Python packages, they are already included.

Here is SAS University Edition: Help Center.

Loading a SAS data set into a Python

- Import saspy
- Create a connection with SAS, authenticating and spinning up a SAS session
- Create python object
- Run descriptive statistics

If you have installed SASPy, SAS Kernel, and Jupyter Notebook for Licensed SAS in your computer, use the following codes

```
import saspy
sas = saspy.SASsession(cfgname='winlocal')
iris = sas.sasdata("iris","SASHELP")
iris.describe()
```

If you are using "JupyterLab in SAS University Edition", use the following code block. Notice the null argument in the second line of the code block below.

```
import saspy
sas = saspy.SASsession()
iris = sas.sasdata("iris","SASHELP")
iris.describe()
```