



Optimizing Logistics Distribution Routes: A Graph Theory Approach

Pre-Workshop Guide

June 2023



Contents

Introduction	2
Installation	3
• Installing Python using Anaconda	
System Verification	4
• Verify Python Installation	
• Verify Conda Installation	
• Verify Jupyter Installation	
Virtual Environment and Package Preparation	8

■ ■ ■



Introduction

This guide is a resource for students at Algoritma to use in setting up their laptop or environment prior to the scheduled workshops. In this guide, students can find a list of prerequisites that will be consistently used throughout the entire course. These prerequisites are required to be **completed before** the start of the workshop.

For new students, we will run through the installation process to ensure that the necessary programming languages and tools - such as Python - are installed. The next section will then talk about methods on how to verify whether the installs were completed successfully.

For recurring students, we recommend repeating the System Verification section once more to confirm past completed installations.

■ ■ ■



Installation

Installing Python using *Anaconda*

For our Python installation, we will be using and installing a package manager named **Anaconda**. With **Anaconda**, users will not only have Python installed but also will have the necessary packages (i.e. numpy, pandas) utilized in our workshops. Also, **Anaconda** would have included the installation of **Jupyter** - an open-source web application that allows you to create and share Python code. For alternative, you can also install **Miniconda**, the minimal Anaconda installer version that includes only conda, Python, the packages they depend on. Once opening the link below, **please choose Python version 3 for installation**.

Use this link: <https://www.anaconda.com/download>, or
<https://docs.conda.io/en/latest/miniconda.html> for Miniconda

More info on **Anaconda**:

<https://docs.continuum.io/anaconda/#anaconda-navigator-or-conda>

Warning: For Windows operating systems, if you can't find the conda command from your Command Prompt please add the **C:\User\Anaconda3** and the **C:\User\Anaconda3\Scripts** to the environment variable as shown here:

<https://superuser.com/questions/949560/how-do-i-set-system-environment-variables-in-windows-10>

■ ■ ■

System Verification

- For Mac OS X and Linux-based OS: Open “*Terminal*”
- For Windows: Open “*Command Prompt*”

Verify Python Installation:

1. Type the command `python`
2. If the installation was completed successfully, there should be a response which includes information on which Python version was installed as shown below. In this case, it appears the user installed Python version 2.7.13. **But make sure yours is 3.10.**
3. To exit, enter the command `quit()` or use Ctrl-D

```
[Matthews-MacBook-Pro:~ matthewhamdani$ python
Python 2.7.13 |Anaconda 4.4.0 (x86_64)| (default, Dec 20 2016, 23:05:08)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>>
```

Figure 1: python Response on Mac OS X Terminal

```
(base) C:\Users\dyahn>python
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figure 2: python Response on Windows Command Prompt

Verify Anaconda Installation

1. Type the command `conda list` in your “**Terminal**” or “**Command Prompt**”.
2. If the installation was completed successfully, your terminal will give a response of list of packages like the example below. In this case, the environment has Python version 3.10 pre-installed.
3. If your terminal do not give any response, please check the **Warning** in the installation section, if the problem still persist please contact mentor@algorit.ma via email for further help.

```
Matthews-MacBook-Pro:~ matthewhamdani$ conda list
# packages in environment at /Users/matthewhamdani/anaconda2:
#
_license                1.1                py27_1
alabaster                0.7.10            py27_0
anaconda                 4.4.0             np112py27_0
anaconda-client          1.6.3             py27_0
anaconda-navigator       1.6.2             py27_0
anaconda-project         0.6.0             py27_0
appnope                  0.1.0             py27_0
appscript                 1.0.1             py27_0
asn1crypto                0.22.0            py27_0
astroid                   1.4.9             py27_0
astropy                   1.3.2             np112py27_0
babel                     2.4.0             py27_0
backports                 1.0               py27_0
backports_abc             0.5               py27_0
beautifulsoup4            4.6.0             py27_0
bitarray                  0.8.1             py27_0
blaze                     0.10.1            py27_0
bleach                    1.5.0             py27_0
bokeh                     0.12.5            py27_1
boto                      2.46.1            py27_0
bottleneck                1.2.1             np112py27_0
```

Figure 3: conda list Response on Mac OS X Terminal

```
Anaconda Prompt (Anaconda3)
(base) C:\Users\indri>conda list
# packages in environment at C:\ProgramData\Anaconda3:
#
# Name                        Version      Build    Channel
_anaconda_depends            2022.05      py39_0
_ipyw_jlab_nb_ext_conf       0.1.0        py39haa95532_0
aiohttp                      3.8.1        py39h2bbff1b_1
aiosignal                    1.2.0        pyhd3eb1b0_0
alabaster                    0.7.12       pyhd3eb1b0_0
anaconda                     custom       py39_1
anaconda-client              1.9.0        py39haa95532_0
anaconda-navigator           2.2.0        py39haa95532_0
anaconda-project             0.10.2       pyhd3eb1b0_0
anyio                        3.5.0        py39haa95532_0
appdirs                      1.4.4        pyhd3eb1b0_0
argon2-cffi                  21.3.0       pyhd3eb1b0_0
argon2-cffi-bindings        21.2.0       py39h2bbff1b_0
arrow                        1.2.2        pyhd3eb1b0_0
astroid                      2.6.6        py39haa95532_0
astropy                      5.0.4        py39h080aedc_0
asttokens                    2.0.5        pyhd3eb1b0_0
async-timeout                4.0.1        pyhd3eb1b0_0
atomicwrites                 1.4.0        py_0
attrs                        21.4.0       pyhd3eb1b0_0
automat                      20.2.0       py_0
autopep8                     1.6.0        pyhd3eb1b0_0
babel                        2.9.1        pyhd3eb1b0_0
backcall                     0.2.0        pyhd3eb1b0_0
backports                    1.1          pyhd3eb1b0_0
backports.functools_lru_cache 1.6.4        pyhd3eb1b0_0
backports.tempfile           1.0          pyhd3eb1b0_1
backports.weakref            1.0.post1    py_1
bcrypt                       3.2.0        py39h196d8e1_0
beautifulsoup4               4.11.1       py39haa95532_0
binaryornot                  0.4.4        pyhd3eb1b0_1
```

Figure 4: conda list Response on Windows Command Prompt

Verify Jupyter Installation:

1. For Mac OS X or Linux-based OS: Type the command `jupyter notebook` in your **“Terminal”**. For windows OS find your jupyter notebook in the windows search or use **Anaconda Prompt** and type in `jupyter notebook`.
2. If the installation was completed successfully, Jupyter would have started a server connection and automatically opened a new window in your browser.
3. If it does not open automatically, the **Terminal** or **Command Prompt** would have provided a URL link for you to open in your browser manually.
4. If neither of these options occurred, repeat the **Anaconda** installation process.
5. To shutdown the server and exit, use Ctrl-C then type `y` to confirm or `n` to cancel

```
[Matthews-MacBook-Pro:~ matthewhamdani$ jupyter notebook
[I 11:06:53.460 NotebookApp] Serving notebooks from local directory: /Users/matthewhamdani
[I 11:06:53.460 NotebookApp] 0 active kernels
[I 11:06:53.460 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=61604128eaa7ddfff4048ce8a1e271959149f48584c04363
[I 11:06:53.460 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:06:53.461 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=61604128eaa7ddfff4048ce8a1e271959149f48584c04363
0:97: execution error: "http://localhost:8888/tree?token=50a7a657dceba621e9932c7b1ab4873bdf5073a646505819" doesn't
understand the "open location" message. (-1708)
```

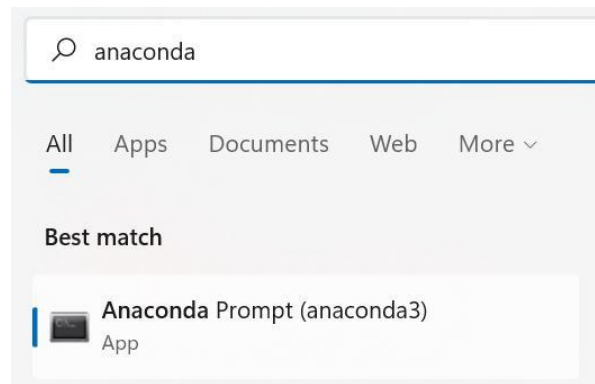
Figure 5: jupyter notebook Response on Mac OS X Terminal

```
(base) C:\Users\dyahn>jupyter notebook
[W 13:12:20.273 NotebookApp] Error loading server extension jupyter_http_over_ws
Traceback (most recent call last):
  File "C:\Users\dyahn\anaconda3\lib\site-packages\notebook\notebookapp.py", line 1670, in init_server_extensions
    mod = importlib.import_module(modulename)
  File "C:\Users\dyahn\anaconda3\lib\importlib\_init_.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1014, in _gcd_import
  File "<frozen importlib._bootstrap>", line 991, in _find_and_load
  File "<frozen importlib._bootstrap>", line 973, in _find_and_load_unlocked
ModuleNotFoundError: No module named 'jupyter_http_over_ws'
[I 13:12:20.312 NotebookApp] [jupyter_nbextensions_configurator] enabled 0.4.1
[I 13:12:20.424 NotebookApp] JupyterLab extension loaded from C:\Users\dyahn\anaconda3\lib\site-packages\jupyterlab
[I 13:12:20.424 NotebookApp] JupyterLab application directory is C:\Users\dyahn\anaconda3\share\jupyter\lab
[I 13:12:20.427 NotebookApp] Serving notebooks from local directory: C:\Users\dyahn
[I 13:12:20.427 NotebookApp] The Jupyter Notebook is running at:
[I 13:12:20.427 NotebookApp] http://localhost:8888/?token=0648ecf70743a074c963e220d7027792e0c4f2e4b55b90f6
[I 13:12:20.427 NotebookApp] or http://127.0.0.1:8888/?token=0648ecf70743a074c963e220d7027792e0c4f2e4b55b90f6
[I 13:12:20.428 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 13:12:20.461 NotebookApp]
```

Figure 6: jupyter notebook Response on Windows Command Prompt

Virtual Environment and Package Preparation *

1. Search and open **Anaconda Prompt (anaconda3)** application for windows, or **Terminal** for Mac



2. Create a new virtual environment named **dss_june** using Python version **3.10**

```
conda create -n dss_june python=3.10
```

3. Proceed the installation by typing **y**


```
The following NEW packages will be INSTALLED:

ca-certificates      pkgs/main/win-64::ca-certificates-2021.10.26-haa95532_4
certifi              pkgs/main/win-64::certifi-2021.10.8-py39haa95532_2
openssl              pkgs/main/win-64::openssl-1.1.1m-h2bbff1b_0
pip                  pkgs/main/win-64::pip-21.2.4-py39haa95532_0
python               pkgs/main/win-64::python-3.9.7-h6244533_1
setuptools           pkgs/main/win-64::setuptools-58.0.4-py39haa95532_0
sqlite               pkgs/main/win-64::sqlite-3.37.0-h2bbff1b_0
tzdata               pkgs/main/noarch::tzdata-2021e-hda174b7_0
vc                   pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime       pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel                pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0
wincertstore         pkgs/main/win-64::wincertstore-0.2-py39haa95532_2

Proceed ([y]/n)? y
```

4. Activate the newly created virtual environment, namely **dss_june** `conda`

```
activate dss_june
```

- 
5. Change your terminal directory to the path where the **requirements.txt** is located.
For example, if your txt file is in the Downloads directory, use: `cd Downloads`

6. For standardization packages and libraries installation, please install using the **requirements.txt** shared to you.

```
pip install -r requirements.txt
```

Wait until the installation is complete.

7. Install **kernel** to connect the virtual environment to the Jupyter Notebook.

```
pip install ipykernel
```

```
python -m ipykernel install --user --name=dss_june
```

8. You are good to go. You can now open up your **Jupyter Notebook** by typing
`jupyter notebook` or `python3 -m notebook` on your Anaconda Prompt.

Notes: If you choose not to install the package via the shared requirements.txt, make sure you have the following package version for standardization:

1. folium==0.14.0
2. matplotlib==3.7.1
3. networkx==3.1
4. numpy==1.24.2
5. osmnx==1.3.1.post0
6. pandas==2.0.0
7. scikit-learn==1.2.2

* will be demonstrated later in pre class session

■ ■ ■

