

how.To → INSTALL JUPYTER NOTEBOOK

Overview

Programming assignments are performed in [Jupyter Notebook](#) because it enables one to create and share documents with active code, equations, visualizations, and narrative text.

Jupyter Notebooks are an open-source web application allowing users to create and share documents that contain live code, equations, visualizations, and narrative text. They're a go-to data science resource, given their ability to communicate outcomes and methods in an interactive and reproducible manner, especially when data sources are web-based, such as with GitHub repositories and government statistics.

- **Interactive data analysis:** Jupyter Notebooks allow data analysts to interact with data and visualize results in real-time, making exploring datasets and identifying patterns easier.
- **Collaboration:** Jupyter Notebooks are shareable amongst users facilitating collaboration activities. Students can work together on assignments. The software is currently in beta testing for real-time, seamless web-browser collaboration.
- **Reproducibility:** Because Jupyter Notebooks contain live code and data, they make it easy to reproduce and verify results, which is essential for scientific research.
- **Multiple program language support:** Notebook languages included Python, R, and Julia, making them flexible and adaptable to different research needs.
- **Documentation and communication:** Jupyter Notebooks allow data analysts to document their work and communicate outcomes clearly and concisely, which is vital in academic and professional settings.

BENEFITS, FEATURES, AND USEFULNESS

Benefits

- **Interactive:** Jupyter Notebooks provide an interactive computing environment that allows users to test and experiment with code and data in real time.
- **Reproducible:** Jupyter Notebooks make it easy to reproduce and share code and data, which can help with collaboration and replication of results.
- **Flexible:** Jupyter Notebooks support various programming languages, including Python, R, and Julia.
- **Visual:** Jupyter Notebooks allow users to create visualizations and graphs within the notebook, making it easy to analyze and explore data.

Features

- **Code execution:** Jupyter Notebooks allow users to execute code in real time, which makes it easy to test and experiment with different programming techniques and algorithms.
- **Markdown support:** Jupyter Notebooks support Markdown, which allows users to create rich text documents with headings, bullet points, links, images, and more.
- **Visualization:** Jupyter Notebooks support various visualization libraries, such as Matplotlib, Seaborn, and Bokeh, allowing users to create interactive visualizations and graphs.

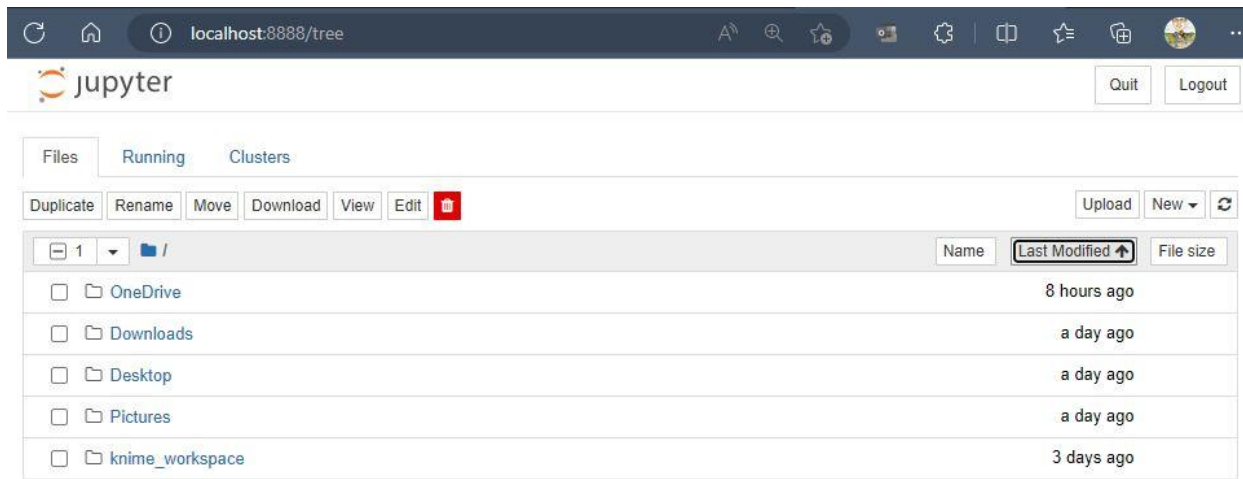
Usefulness

- **Data exploration:** Jupyter Notebooks are a powerful tool for exploring and analyzing data, allowing users to visualize and manipulate data in real time.
- **Education:** Jupyter Notebooks are a popular tool for teaching and learning programming, as they allow students to experiment with code and see the results in real time.
- **Research:** Jupyter Notebooks are a popular tool for researchers, as they allow for easy sharing and collaboration of code, data, and generation of similar results.
- **In summary,** Jupyter Notebooks are a powerful data mining tool that enables data analysts to explore, analyze, and communicate their findings in an interactive and reproducible way.

MAIN MENUS

When installed, two browser window views include:

- A front view visualizes the Jupyter home page with a user-defined folder structure.
- A rear view is an individual notebook opened in a separate browser window.



jupyter 1.7.2.week1.basicwalkthrough Last Checkpoint: 01/09/2020 (autosaved) Python 3 (ipykernel) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted

```
In [2]: # fit linear regression models by ordinary least squares
set_I_design_matrix = sm.add_constant(anscombe['x1'])
set_I_model = sm.OLS(anscombe['y1'], set_I_design_matrix)
print(set_I_model.fit().summary())

set_II_design_matrix = sm.add_constant(anscombe['x2'])
set_II_model = sm.OLS(anscombe['y2'], set_II_design_matrix)
print(set_II_model.fit().summary())

set_III_design_matrix = sm.add_constant(anscombe['x3'])
set_III_model = sm.OLS(anscombe['y3'], set_III_design_matrix)
print(set_III_model.fit().summary())

set_IV_design_matrix = sm.add_constant(anscombe['x4'])
set_IV_model = sm.OLS(anscombe['y4'], set_IV_design_matrix)
print(set_IV_model.fit().summary())
```

OLS Regression Results

Dep. Variable:	y1	R-squared:	0.667
Model:	OLS	Adj. R-squared:	0.629
Method:	Least Squares	F-statistic:	17.99
Date:	Tue, 24 Oct 2017	Prob (F-statistic):	0.00217
Time:	15:35:58	Log-Likelihood:	-16.841
No. Observations:	11	AIC:	37.68
Df Residuals:	9	BIC:	38.48
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	3.0001	1.125	2.667	0.026	0.456	5.544
x1	0.5001	0.118	4.241	0.002	0.233	0.767

Omnibus: 0.082 Durbin-Watson: 3.212
 Prob(Omnibus): 0.960 Jarque-Bera (JB): 0.289
 Skew: -0.122 Prob(JB): 0.865
 Kurtosis: 2.244 Cond. No. 29.1



SOFTWARE AND SCIENTIFIC PACKAGE INSTALLATION

Now that you have a comprehensive understanding of what we're trying to accomplish with data mining, let's get to it! In this course, we will rely on a selection of software programs to complete a range of data mining tasks and assignments. This page will guide you through the installation process for Anaconda and show you how to access Jupyter Notebooks.

Anaconda [H3]

[Anaconda](#) is a powerful tool for installing multiple software packages and environments for Python, which we will be using extensively in this course, as well as many Python packages like PySpark, NumPy, scikit-learn, and pandas. These packages are collections of modules, functions, classes, and other code assets used by programmers and analysts to perform computations, organize information, and generate graphs. Using Anaconda will also ensure that all students are using the same environment.

Installing Anaconda

Select your operating system from the following options to navigate to the applicable installation page.

<Button>[Install for Windows](#)

<Button>[Install for Mac](#)

<Button>[Install for Linux](#)

Installing Python Libraries [H4]

Once you have successfully installed Anaconda, launch the Anaconda Navigator interface from the Start Menu and familiarize yourself with this interface. Inspect the "Environments" tab to view software packages installed. You should see NumPy, pandas, Matplotlib, and PySpark. If you do not see these packages, launch the CMD.exe to open a black window known as the terminal, and type the following to install each:

- `pip install pandas`

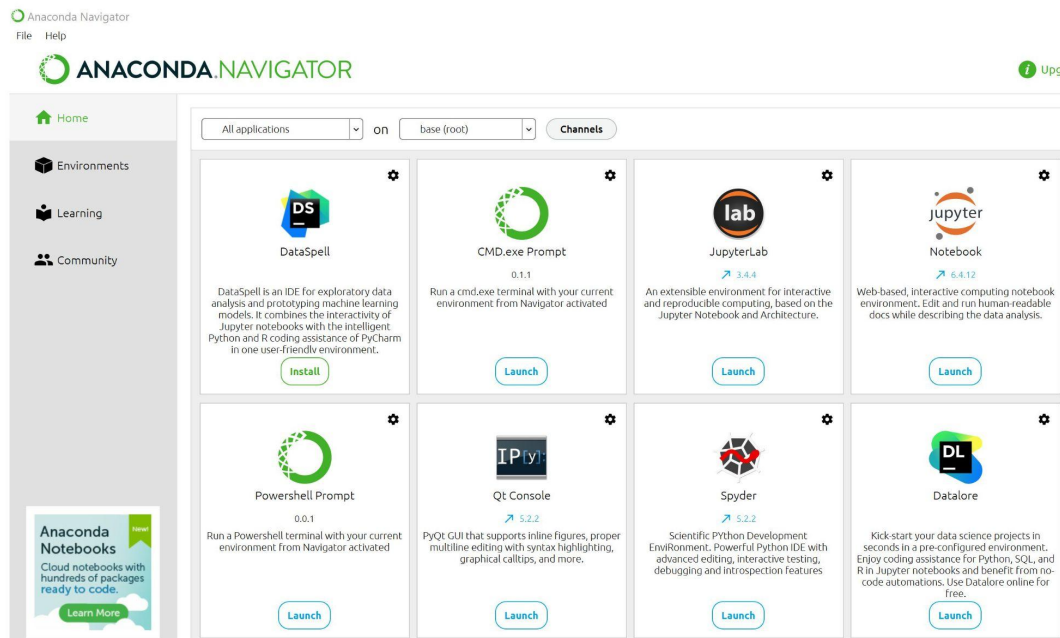
- pip install numpy
- pip install matplotlib
- pip install pyspark

Jupyter Notebook

Programming assignments are done in [Jupyter Notebook](#). Jupyter Notebook will allow you to create and share documents that contain live code, equations, visualizations, and narrative text. For each module, you will receive a Jupyter Notebook that contains the problems that you must solve along with some template code to build on.

Launching Jupyter Notebook [H4]

From the Anaconda Navigator, select Jupyter Notebook (not JupyterLab) to launch it. This will start a local web server. Jupyter Notebook's main interface will open in the computer's default browser.



<Insert Image UTK_COSC_526_M1_G01, alt text="Anaconda.Navigator control panel; select upper right Jupyter Notebook=" middle">

This interface will be used to:

- Create new Jupyter nNotebooks
- Upload and download notebooks for sharing
- Upload and download data files for performing analysis within a nNotebook

Note: Some students enjoy the experience of an integrated development environment (IDE), such as JupyterLab or Spyder, included in this installation environment. Use the [Jupyter Discourse Forum](#) to locate resources for additional self-learning.

RESOURCES

- [Anaconda for Windows](#)
- [Anaconda for macOS](#)
- [Anaconda for Linux](#)
- [Install scientific packages](#)

Additional Resources

- Anaconda installation [documentation](#)
- Jupyter Notebook [documentation](#) (including [get started](#) guides)
- Jupyter Discourse [Forum](#)
 - Search here for tips, tricks, and solutions
- Python Package Index ([pypi](#))
- [Spyder IDE](#)—an alternative programming environment to Notebooks called an integrated development environment ([IDE](#)). Spyder is a sister environment to Notebook providing an interactive console to view data, variables, and outcomes. It is not covered in the course but works alongside Jupyter Notebook.
- [GitHub](#)—a place for storing files, searching for ideas, and framing interactive Jupyter Notebook environments
 - All data mining and machine learning scientists should have a page!
- [Jupyter Project Documentation](#)
- [Get Started with Jupyter Notebook](#)
- Jupyter Discourse [Forum](#).
 - Search for tips, tricks, and solutions.
- [howTo.jupyter.Notebook.guide.Bryn.Mawr.college.ipynb](#)
- [Architecture—Jupyter ecosystem](#)
- Python crash course: [B.py.crash.course.pdf](#)

<End Shadow Box>