

ECON 1611: Big Data, Machine Learning and Society

How to use Python

SECTION 1: OPEN PYTHON IN ONE OF TWO WAYS

OPTION 1: THROUGH YOUR WEB BROWSER VIA GOOGLE COLAB

OPTION 2: DOWNLOAD ANACONDA

SECTION 2: CODING IN PYTHON

SECTION 3: HOW TO CREATE NEW ENVIRONMENTS IN ANACONDA

Section 1: Open Python in one of two ways

Option 1: Through your web browser via Google Colab

1. Open <https://colab.research.google.com/>
2. **To start a new Python notebook:** click File > New notebook. Start writing python code in the box with the play button. You can run the code by clicking the play button. Add new rows by clicking '+Code' at the top left of the screen.
 - To read data into your notebook use these commands:

```
from google.colab import files
uploades = files.upload()
df = pd.read_csv('baseball.csv')
```
 - **Note:** change the data file name `baseball.csv` to the name of the data file you want to read.
 - When you run these three commands a box will appear prompting you to 'Choose Files'. Click this button and select the data file you want to load.
3. **To open an existing Python notebook:** click File > Upload > Choose File. Navigate to the folder where your ipynb file is saved and click Open.
 - If your existing Python notebook calls on a dataset, find the command that reads in the data i.e. `df = pd.read_csv('baseball.csv')`

and add these top two command lines before reading in the data i.e.

```
from google.colab import files
uploades = files.upload()
df = pd.read_csv('baseball.csv')
```
 - **Note:** change the data file name `baseball.csv` to the name of the data file you want to read.
 - When you run these three commands a box will appear prompting you to 'Choose Files'. Click this button and select the data file you want to load.
4. **To open a notebook from GitHub:** click File > Open. Select the GitHub tab.
 - In the search bar area under "Enter a GitHub URL or search by organization or user", enter this URL: <https://github.com/annawzhu/class-exercises>
 - If you can't copy and paste the URL try typing 'annawzhu' in the search bar area and hit enter. Under Repository, click the drop-down arrow and select 'annawzhu/class-exercises'.
 - The weekly notebooks will appear under Path. Select the relevant week's notebook to open it.

Option 2: Download Anaconda

1. Download the program from one of these links:

Windows	64-Bit Graphical Installer (477 MB) 32-Bit Graphical Installer (409 MB)
MacOSX	64-Bit Graphical Installer (440 MB)

2. Double-click the .exe / .pkg file and follow the on-screen instructions.
3. Open Anaconda-Navigator in the Start Menu or Launchpad.
4. To use python, launch **Jupyter Notebook**. This will open a new tab in your web browser.
5. **To start a new Python notebook:**
 - a. In the new browser tab, click through your folders until you find a folder where you want to save your Jupyter Notebook file.
 - b. At the top right, click New > Python. The python shell will open in a new tab.
 - c. You can write python code in the row called 'In'.
 - d. At the top of the screen click 'Run' to execute your code.
 - e. Outputs will be displayed in a new row called 'Out'.
 - f. You can adjust the code in the first In row or you can add new rows by clicking the + sign at the top of the screen.
 - g. Save your file by clicking the floppy disk icon on the top left of the screen.
6. **To open an existing Python notebook:**
 - a. In the new browser tab after launching Jupyter Notebook, click through your folders until you find your Jupyter Notebook file. Click on the file to open it in a new browser.
7. For more detailed guides go to <https://docs.anaconda.com/anaconda/user-guide/getting-started/>

Section 2: Coding in Python

Note: Placing the # symbol in front of a command line changes it from a command to a comment. Python will not execute comment lines.

1. How to import packages

Import the Pandas package

```
import pandas
```

Import the Pandas package and refer to it as 'pd'

```
import pandas as pd
```

Import a single module (math) from the Scipy package

```
from scipy import math
```

Import a single module (math) from the Scipy package and refer to it as 'ma'

```
from scipy import math as ma
```

2. How to import data files

CSV files:

import pandas

```
import pandas as pd
```

import the csv and call the dataset 'baseball'

```
baseball = pd.read_csv('[enter file path]/baseball.csv')
```

view the first 5 rows of the data

```
print(baseball.head())
```

Excel files:

import pandas

```
import pandas as pd
```

import the excel and call the dataset 'levitt'

```
levitt = pd.read_excel('[enter file path]/levitt_ex.xlsx')
```

view the first 5 rows of the data

```
print(levitt.head())
```

dta (Stata) files:

import pandas

```
import pandas as pd
```

import the dta and call the dataset 'levitt_stata'

```
levitt_stata = pd.read_stata('[enter file path]/levitt_ex.dta')
```

view the first 5 rows of the data

```
print(levitt_stata.head())
```

3. How to produce sample statistics

```
# produce sample statistics of height using the baseball data  
print(baseball["Height"].describe())
```

4. How to produce graphs using the baseball data

```
# produce a scatter plot of height and weight
```

```
%matplotlib inline  
baseball.plot.scatter(x='Height', y='Weight')
```

```
# produce a line graph
```

```
baseball.plot.line(x='[enter variable name here]', y='[enter variable name here]')
```

```
# produce a bar graph
```

```
baseball.plot.bar(x='[enter variable name here]', y='[enter variable name here]')
```

5. How to run OLS regression using the baseball data

```
# import the statsmodel package
```

```
import statsmodels.formula.api as smf
```

```
# drop missing data if necessary
```

```
baseball_na = baseball.dropna()
```

```
# regress age on weight
```

```
ols_1 = smf.ols('Weight ~ Age', data=baseball_na).fit()
```

```
# show the regression table
```

```
print(ols_1.summary().tables[1])
```

```
# regress age and height on weight
```

```
ols_2 = smf.ols('Weight ~ Age + Height', data=baseball_na).fit()on weight
```

```
# show regression table
```

```
print(ols_2.summary().tables[1])
```

6. How to create data frames

```
# Import the Pandas package with the following command:
```

```
import pandas as pd
```

Input the data:

```
dataframe = pd.DataFrame({'A': ['A0', 'A1', 'A2', 'A3'],  
                          'B': ['B0', 'B1', 'B2', 'B3'],  
                          'C': ['C0', 'C1', 'C2', 'C3'],  
                          'D': ['D0', 'D1', 'D2', 'D3']},  
                          index=[0, 1, 2, 3])
```

Result:

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

View a subset of your data frame

```
print(dataframe.head())
```

View the data types of the variables in your data frame

```
dataframe.dtypes  
dataframe.info()
```

Pandas dtype	Python type	Usage
object	str or mixed	Text or mixed numeric and non-numeric values
int64	int, uint	Integer numbers
float64	float	Floating point numbers
bool	bool	True/False values

Convert a yes/no variable (i.e. Accident) into a binary variable

```
dataframe['Accident'] = dataframe['Accident'].replace({'No':0,'Yes':1})
```

Note: use this code for any binary string variable i.e. true/false, reject/accept, pass/fail

Convert a categorical variable (i.e. State) into individual dummy variables

```
dataframe_state = pd.get_dummies(dataframe.State)
```

Note: dataframe_state is a new data frame

Merge new dummy variables (i.e. saved in dataframe_state) into original data frame

```
dataframe = pd.concat([dataframe, dataframe_state], axis=1)
```

Change an object variable (i.e. Driver ID) or float variable into an integer variable

```
dataframe['Driver ID'] = dataframe['Driver ID'].astype('int')
```

Change an object variable (i.e. Date) into a date variable

```
dataframe['Date'] = pd.to_datetime(dataframe['Date']).dt.date
```

Change an object variable (i.e. Time) into a time variable

```
dataframe['Time'] = pd.to_datetime(dataframe['Time']).dt.time
```

Change the name of a variable

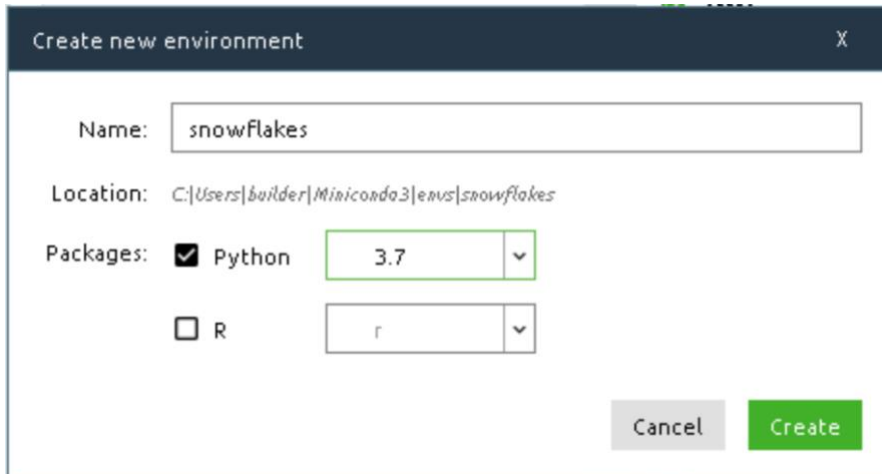
```
dataframe.rename(columns={'old_var_name': 'new_var_name'}, inplace=True)
```

Section 3: How to create new environments in Anaconda

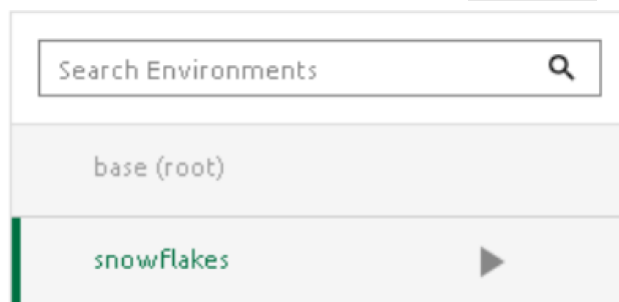
Create a new environment named **snowflakes** and install a package in it

Anaconda Navigator can handle separate environments containing files, packages, and their dependencies that will not interact with other environments.

1. In Navigator, click the **Environments** tab, then click the Create button. The **Create new environment** dialog box appears.
2. In the **Environment** name field, type a descriptive name for your environment.



3. Click **Create**. Navigator creates the new environment and activates it. Now you have two environments, the default environment **base (root)**, and **snowflakes**.



4. Switch between them (activate and deactivate environments) by clicking the name of the environment you want to use. **Tip:** The active environment is the one with the arrow next to its name.
5. Return to the other environment by clicking its name.

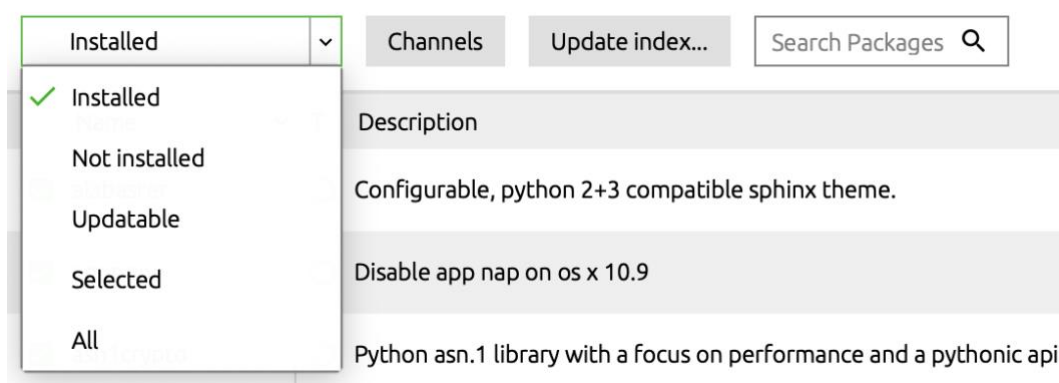
Create a new environment named “snakes” that contains Python 3.5

When you create a new environment, Navigator installs the same Python version used when you downloaded and installed Anaconda. If you want to use a different version of Python, for example Python 3.5, simply create a new environment and specify the version of Python that you want in that environment.

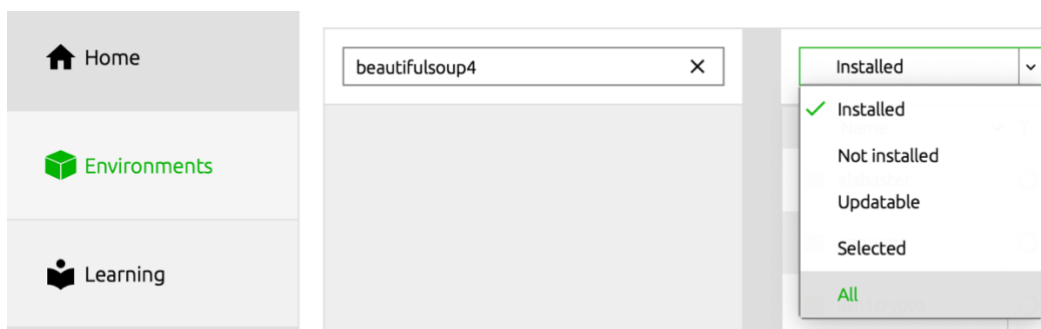
1. In Navigator, click the **Environments** tab, then click the Create button. The Create new environment dialog box appears.
2. In the Environment name field, type the descriptive name “snakes” and select the version of Python you want to use from the Python Packages box (3.8, 3.7, 3.6, 3.5, or 2.7). Select a different version of Python than is in your other environments, base or snowflakes.
3. Click the Create button.
4. Activate the version of Python you want to use by clicking the name of that environment.

How to check which packages you have installed, which are available, and how to install other packages

1. To find a package you have already installed, click the name of the environment you want to search. The installed packages are displayed in the right pane.
2. You can change the selection of packages displayed in the right pane at any time by clicking the drop-down box above it and selecting Installed, Not Installed, Updatable, Selected, or All.



3. Check to see if a package you have not installed named “beautifulsoup4” is available from the Anaconda repository (must be connected to the Internet). On the Environments tab, in the Search Packages box, type “beautifulsoup4”, and from the Search Subset box select All or Not Installed.



- To install the package into the current environment, check the checkbox next to the package name, then click the bottom Apply button. The newly installed program will be displayed in your list of installed programs.

Not installed

Channels

Update index...

Search Packages

Name	T	Description	Version
<input type="checkbox"/> _ipyw_jlab_nb_ex...		A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
<input checked="" type="checkbox"/> _mutex_mxnnet			0.0.40
<input type="checkbox"/> _nb_ext_conf			0.4.0
<input type="checkbox"/> _py-xgboost-mutex			2.0
<input checked="" type="checkbox"/> _r-mutex			1.0.0
<input type="checkbox"/> _r-xgboost-mutex			2.0
<input type="checkbox"/> _tflow_1100_select			0.0.2

1783 packages available 2 packages selected

ApplyClear

How to import an environment (a yml file)

- Launch Anaconda
- Click the Environments tab at the left of the screen
- Click Import at the bottom of the screen
- A pop-up window will appear. Click on the folder icon under Local drive and navigate to where your yml file is saved. Click Open.
- Give the environment a name in the dialogue box under 'New environment name'
- Click Import