A. Installation Guidance for Python

You should be able to do the exercise with Python 2.7 or 3.* . if you have already installed Python proceed to Development with Python

1. Check if Python is installed (on terminal)

Get the version number of Python installation else an error

- (Linux or MacOS): `$ python -V`
- (Windows): `$ python` or `$ py` (new python launcher implementation)

2. Instructions to install Python *(to install with recommended Anaconda installation see point 3.)*

- Official guide: https://wiki.python.org/moin/BeginnersGuide/Download
- Other useful
  - Mac: https://wsvincent.com/install-python3-mac/
  - Windows: https://datatofish.com/add-python-to-windows-path/

3. Instructions to install Python with Anaconda *(Recommended)*

- Anaconda is a distribution that simplifies installation and package management for Python
- Go to Anaconda download page (https://www.anaconda.com/distribution/) and find the installer file that matches your system. Open file, start the installation wizard and follow instructions

B. Development with Python

4. Virtual environment

A virtual environment is a '*self-contained directory tree that contains a Python installation for a particular version of Python, plus a number of additional packages*'

Since you will be installing helper packages to perform ML-related tasks, it is recommended to create a virtual environment wherein you will install the required packages instead of doing system-wide installation.

Below is how you create, activate, deactivate and delete virtual environment with Python

- Python
  - A tool for creating isolated Python environment is `venv` (Python 3) or `virtualenv` (Python 2)
  - Using `virtualenv` (Python 2) guide: https://virtualenv.pypa.io/en/stable/userguide/#usage
    - Create: `$ virtualenv ENV`
    - Activate: `$ source /path/to/ENV/bin/activate`
    - Deactivate: `$ deactivate`
  - Using `venv` (Python3) guide: https://docs.python.org/3/tutorial/venv.html

        - Create: `$ python3 -m venv [env-name]`
        - Activate: `$ [env-name]\Scripts\activate.bat`(Windows) or `$ source [env-name]/bin/activate` (Linux or MacOS)
        - Deactivate: `(env-name) $ deactivate`
- Anaconda
    - Using Anaconda Navigator
    - Using conda on Terminal
        - Create: `$ conda create --name [env-name]`
        - Activate: `$ activate [env-name]` (Windows) or `$ source activate [env-name]` (Linux or MacOS)
        - Deactivate: `$ deactivate` (Windows) or `$ source deactivate` (Linux or MacOS)

5. Installing packages *(ML-related packages)*
   Install the following ML-related packages that you will use for the exercise: *numpy, scipy, matplotib, scikit-learn, pandas.* Remember to activate the virtual environment before installing them.

- Python
    - Using pip:
      `$ pip install numpy scipy matplotib scikit-learn pandas`
- Anaconda
    - Using Anaconda Navigator or conda:
      `$ conda install numpy scipy matplotib scikit-learn pandas`

C. Performing ML tasks with Python

Lab1: Linear regression with one variable (*walkthrough of Python syntax)*

Procedure

1. Install Python
2. Create a virtual environment in which you will install ML related packages for the coming exercises
3. Install on the virtual environment (numpy, scipy, matplotib, scikit-learn, pandas, jupyter**): ** If preferred
4. Using your preferred IDE check Example code of Lab1 (see point 5) that performs ML tasks. You can run the script from terminal e.g., using python lab1.py

5. From Lab1 package and ex1data1.txt dataset and Example code is provided for simple linear regression

   Step 0: Import the required packages (os, scipy, sklearn, numpy, pandas, matplotlib.pyplot)

   Step 1: Load data using pandas by creating a DataFrame populated with data using read_csv(). Note our dataset does not have a header and you would need to add the header: https://pandas.pydata.org/pandas-docs/stable/reference/io.html#flat-file

Step 2: Explore and visualize data

- Explore (e.g., using shape, head(), describe(), scatter from matplotlib etc.): https://pandas.pydata.org/pandas-docs/stable/reference/frame.html#constructor
- Visualize (e.g., scatter): https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.scatter.html
- Separate feature variable (X) and target variable (y) by accessing values in Population and Profits respectively

Step 3: Train linear regression using sklearn

Split dataset

- Split dataset 80% for train and 20% for test using sklearn's train_test_split() remember to import it: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

Train model

- Get linear model: from sklearn.linear_model import LinearRegression: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- Access and fit linear model
- Can view theta values with (intercept_ and coef_)

Step 4: Evaluate model

- Plot linear model by drawing a line on scatter plot
- Get MSE value from metrics of sklearn: https://scikit-learn.org/stable/modules/model_evaluation.html#mean-squared-error