Install Python with Anaconda

Install Anaconda

- What is Anaconda?
  - Anaconda is a distribution of Python
  - It includes not only Python, but many libraries that we use in the course, as well as its own virtual environment system.
  - It's an all-in-one install that is extremely popular in data science and machine learning!

Install Anaconda

- Jupyter is an application that we can write code, display images and write down markdown notes.
- Interact via web browser and are easily shareable.
- It is the most popular IDE in data science for exploring and analyzing data!

Install Anaconda

- Let's download Anaconda!
- Go to:
  - https://www.anaconda.com/products/individual
  - Select the one matches your operating system

Install Anaconda

- Follow the tutorial with your corresponding OS:
  - For Windows OS:
    - https://docs.anaconda.com/anaconda/install/windows/
    - In the 8th steps of the tutorial, for advanced installation options, if you're new to python, you should click on the both options(Even it says not recommended)
    - This option will set this anaconda version of python as a default python in your computer. If you don't have another version of python installed before, it should be fine with option checked. If you does, this will new version of python will be your default.
  - For Mac OS:
    - https://docs.anaconda.com/anaconda/install/mac-os/
  - For Linux
    - https://docs.anaconda.com/anaconda/install/linux/

Create Virtual Environments

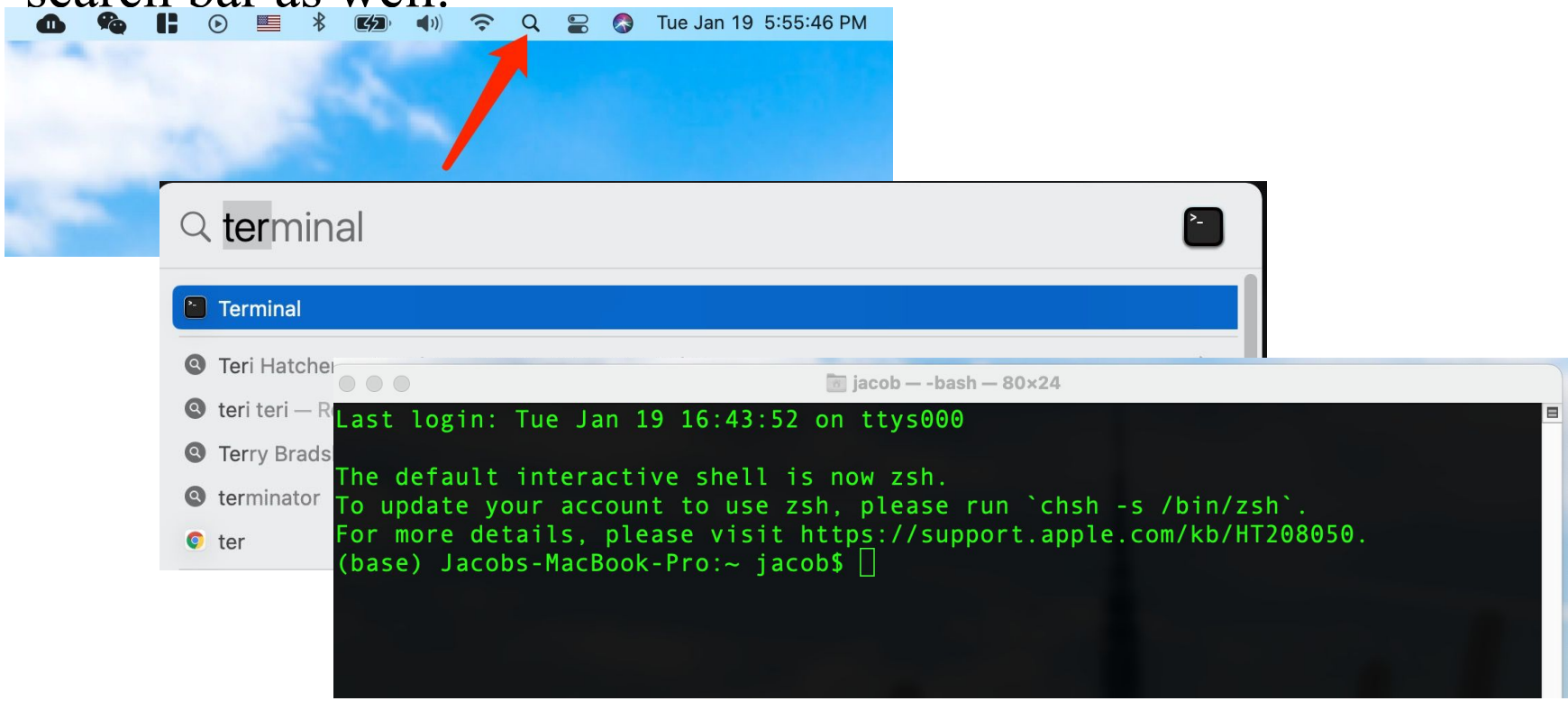
- Virtual Environments allow you to set up virtual installations of Python and libraries on your computer
- You can have multiple versions of Python or libraries and easily activate or deactivate these environments
- Why we want to do this?
  - For example:
    - You build a program with Tensorflow 1.0
    - Tensorflow 2.0 released
    - You want to see what're new features in 2.0 while still have your original program to run.
    - Solution: Created a new virtual environment with new version of Tensorflow

Create Virtual Environments

- For example, you can also create different versions of Python.
- You want one environment with Python 2.7 and another with Python 3.5.

Create Virtual Environments

- Once we finished installation, for MacOS, you can open up the terminal.
- For Windows OS, you can search ***Anaconda Prompt*** in the search bar as well.



Create Virtual Environments

- First, we download the *requirements.txt* file from Coursework, and move it to Desktop
- Terminal Command to navigate around
- List all the files in current working directory:

```
(base) Jacobs-MacBook-Pro:~ jacob$ ls
Anacoda3                Movies                  images
Applications            Music                   js
Check                   Pictures                mapsnyc
Desktop                 Public                  newmasp
Documents               Server                  nltk_data
Downloads               backup-rstudio-desktop  opt
Library                 css                     seaborn-data
Maps                    iCloud Drive (Archive)  shinyplot
```

- Change the current working directory to Desktop:
  - **cd** Desktop/
  - Then using **ls** to list all the files in Desktop, make sure *requirements.txt* .

```
(base) Jacobs-MacBook-Pro:Desktop ja
(base) Jacobs-MacBook-Pro:~ jacob$ cd Desktop/
(base) Jacobs-MacBook-Pro:Desktop jacob$ ls
2101.05490.pdf          Ph.D. Applications      requirements.txt
Columbia                Udemy
Miscellaneous           eBook
(base) Jacobs-MacBook-Pro:Desktop jacob$
```

Create Virtual Environments

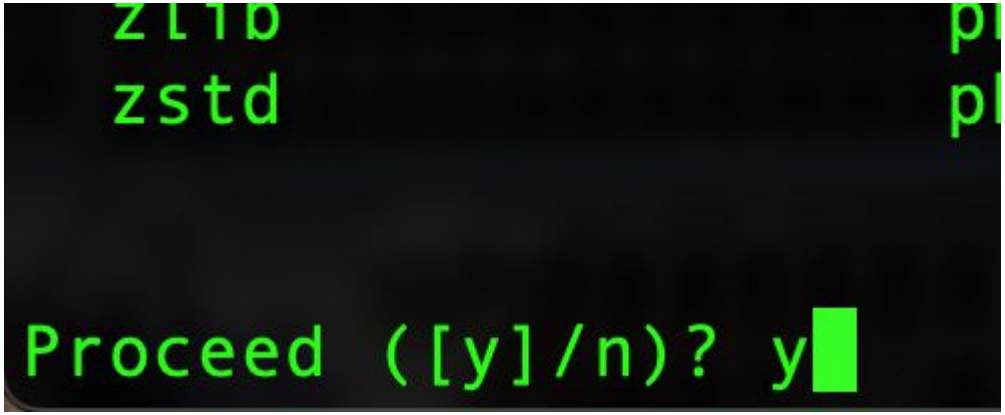
- We will following this website to create a virtual environment for this course:
  - https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#creating-an-environment-with-commands
  - Created an environment with name: sml_s21
    - conda create -n sml_s21 --file requirements.txt

```
(base) Jacobs-MacBook-Pro:~ jacob$ conda create -n sml-s21 --file Desktop/requirements.txt

Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next
 repodata source.
Collecting package metadata (repodata.json): done
Solving environment: \
Warning: 2 possible package resolutions (only showing differing packages):
  - defaults/noarch::parso-0.8.1-pyhd3eb1b0_0, defaults/osx-64::jedi-0.17.0-py38_0
  - defaults/noarch::parso-0.7.0-py_0, defaults/osx-64::jedi-0.17.2-py38hecd8cb5done


==> WARNING: A newer version of conda exists. <==
  current version: 4.8.4
  latest version: 4.9.2
```

Create Virtual Environments

- Type y to proceed
- 

- Then wait roughly several minutes ~

Activating an environment

- To activate the environment we just created:
  - conda activate sml_s21
  - After activated, we see the left side changed from ***base*** to our environment name - sml_s21.

```
# To activate this environment, use
#
#     $ conda activate sml_s21
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) Jacobs-MacBook-Pro:Desktop jacob$ conda activate sml_s21
(sml_s21) Jacobs-MacBook-Pro:Desktop jacob$ jupte
```

- To deactivate the environment we just created:
  - conda deactivate

Install a new kernel in Jupyter

- (base)######## $ conda activate sml_s21
- (sml_s21)#### $ python -m ipykernel install --user --name ipykernel

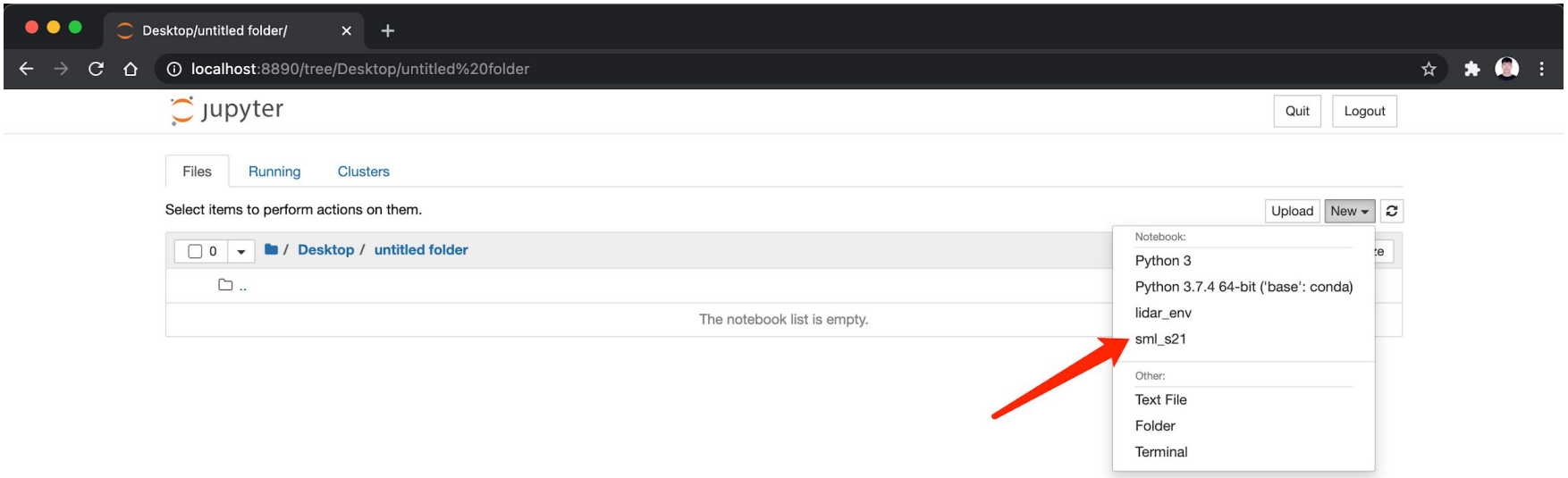
```
# All requested packages already installed.

(sml_s21) Jacobs-MacBook-Pro:Desktop jacob$ python -m ipykernel install --us
er --name sml_s21
Installed kernelspec sml_s21 in /Users/jacob/Library/Jupyter/kernels/sml_s2
(sml_s21) Jacobs-MacBook-Pro:Desktop jacob$ jupyter notebook
```

Opening Jupyter Notebook

- (base)###### $ conda activate sml_s21
- (sml_s21) #### $ jupyter notebook

```
(sml_s21) Jacobs-MacBook-Pro:Desktop jacob$ jupyter notebook
```



Make sure you have sml_s21 in this place.

Testing Packages

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear_model import LinearRegression

In [3]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
```

Installing New Packages

- Open Terminal
- Conda Install:
  a. conda install -n [env_name] [package]
- Conda-forge
  a. conda install -n [env_name] -c conda-forge [package]
- Pip Install:
  a. conda activate [env_name]
  b. pip install [package]

Lastly

- If due to some technical difficulties, you can not finish the above process. It's totally fine, you can use google colab as well. It installed common libraries for us already.
  - https://colab.research.google.com/notebooks/intro.ipynb#recent=true