Tutorial tools

National Cheng Kung University

IKM Lab

Outline

- Conda
- Github tutorial repositories
- Jupyter

Conda

- Conda
  - Open source package and environment management system for any language
- Install Miniconda
  - Links : https://conda.io/miniconda.html
  - Please download the Python3.6 version

Conda

	Windows	Mac OS X	Linux
Python 3.6	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer) 32-bit (bash installer)
Python 2.7	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer) 32-bit (bash installer)

- Install Miniconda
  - Windows
    - Download the exe installer and install it
  - Linux
    - Download the bash installer
    - run “bash Miniconda3-latest-Linux-x86_64.sh” in terminal
  - Mac OS X
    - Download the bash installer
    - run “bash Miniconda3-latest-MacOSX-x86_64.sh” in terminal

Conda

- Install Miniconda
  - Check installation
  - open a terminal and run “conda list”
    - if you see the following message, the installation is completed.

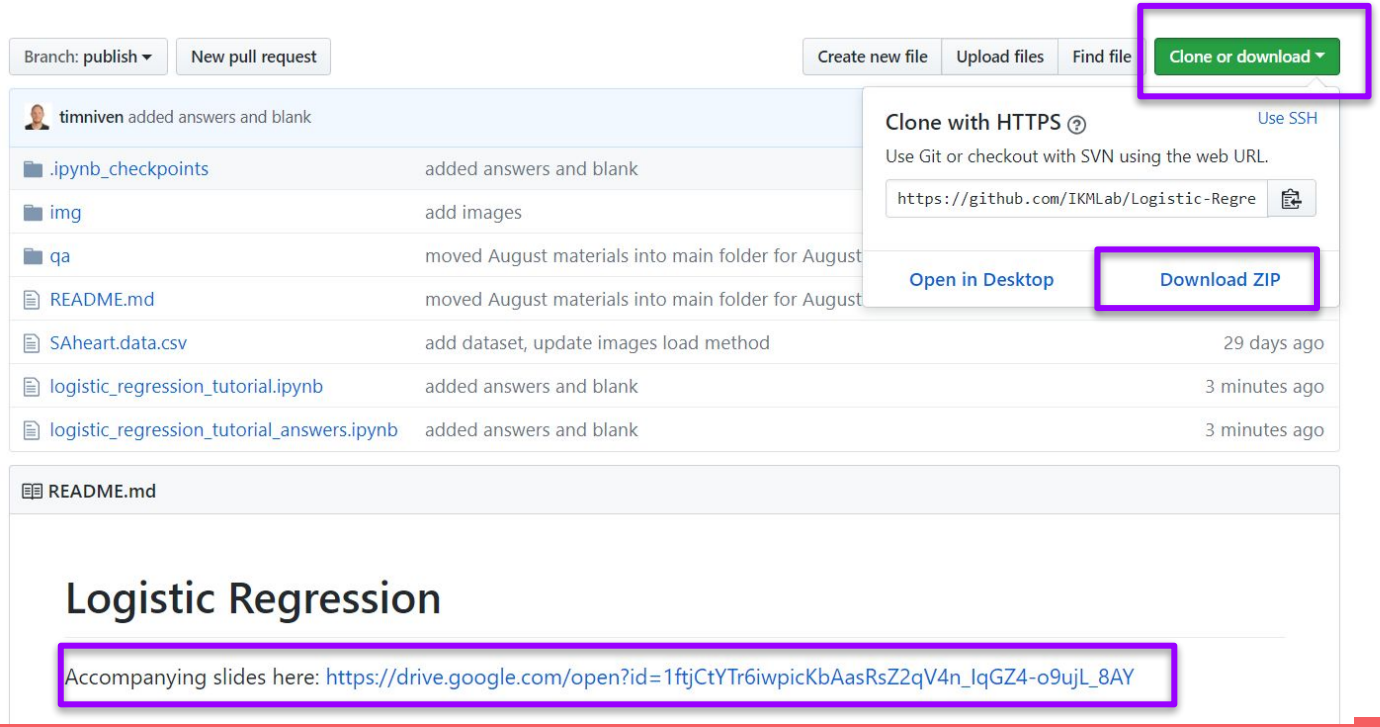
```
kjes89011@kjes89011-System-Product-Name:~/miniconda3/bin$ conda list
# packages in environment at /home/kjes89011/miniconda3:
#
# Name                    Version                   Build  Channel
asn1crypto                0.24.0                   py36_0
ca-certificates           2018.03.07                    0
certifi                   2018.4.16                py36_0
cffi                      1.11.5            py36h9745a5d_0
```

Conda

- Install necessary python library in tutorial
  - Run the following commands in terminal
    - matplotlib :
      - “conda install matplotlib”
    - scikit-learn :
      - “conda install scikit-learn”
    - pandas :
      - “conda install pandas”
    - keras :
      - “conda install -c conda-forge keras”

Github tutorial

- Download and unzip tutorial repositories



Github tutorial

- Download and unzip tutorial repositories
  - Linear and Non Linear regression : https://github.com/IKMLab/Linear-Regression-Tutorial
  - Logistic regression : https://github.com/IKMLab/Logistic-Regression-Tutorial
  - Decision tree and Random Forest : https://github.com/IKMLab/decision-tree-and-random-forest-tutorial

Jupyter

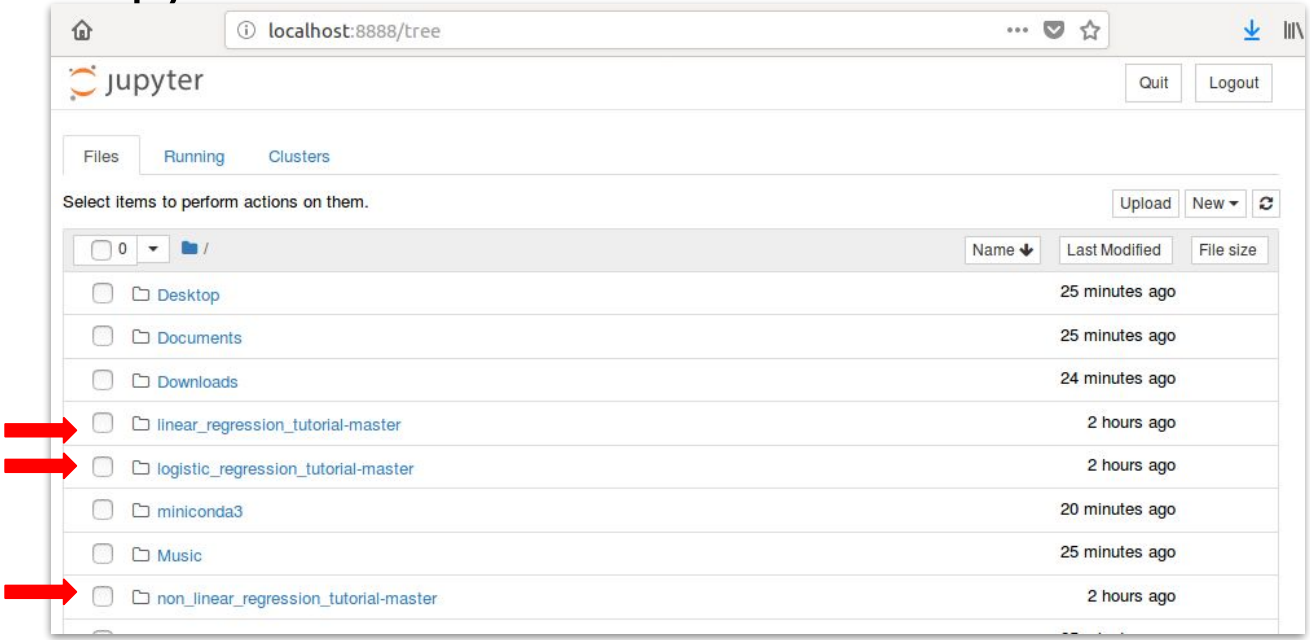
- A IDE on web which allow us to develop the program.
- Install jupyter
  - run “conda install jupyter” in terminal
  - if the program ask you to update package
    - The answer is “yes!”

```
The following packages will be UPDATED:

    asn1crypto:          0.24.0-py36_0           --> 0.24.0-py37_0
    certifi:             2018.4.16-py36_0        --> 2018.4.16-py37_0
    cffi:                1.11.5-py36h9745a5d_0   --> 1.11.5-py37h9745a5d_0
    chardet:             3.0.4-py36h0f667ec_1    --> 3.0.4-py37_1
Proceed ([y]/n)? y
```

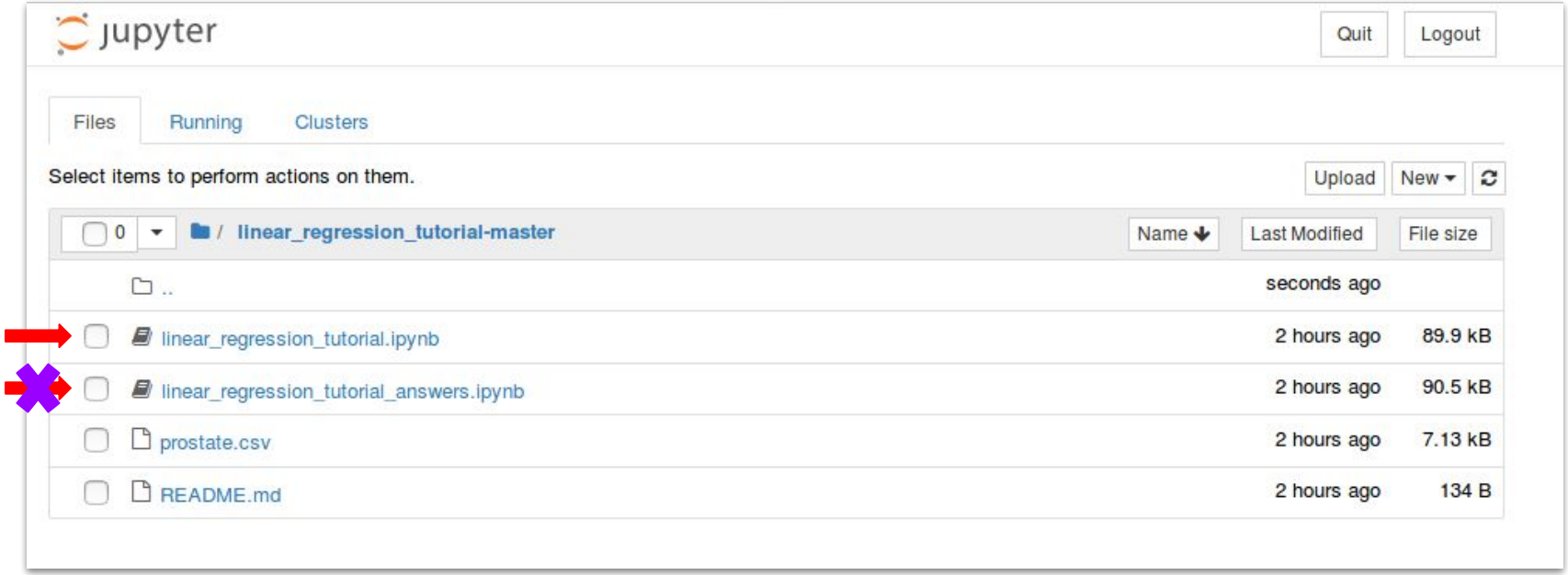
Jupyter

- Start jupyter
  - run “ipython notebook” in terminal



Jupyter

- Choose the “XXX_tutorial.ipython” file to start !
  - Don't chose the “XXX_tutorial_answers.ipython”



Jupyter

- Type your code in “cell.”

Linear Regression Tutorial ¶

In [1]:

```
# packages we will be using
import matplotlib.pyplot as plt
from sklearn import linear_model, metrics, model_selection
import numpy as np
import pandas as pd
```

What is Linear Regression?

Finding a straight line of best fit through the data. This works well when the true underlying function is linear.

Example

We use features $\mathbf{x}$ to predict a "response" y. For example we might want to regress `num_hours_studied` onto `exam_score` - in other words we predict exam score from number of hours studied.

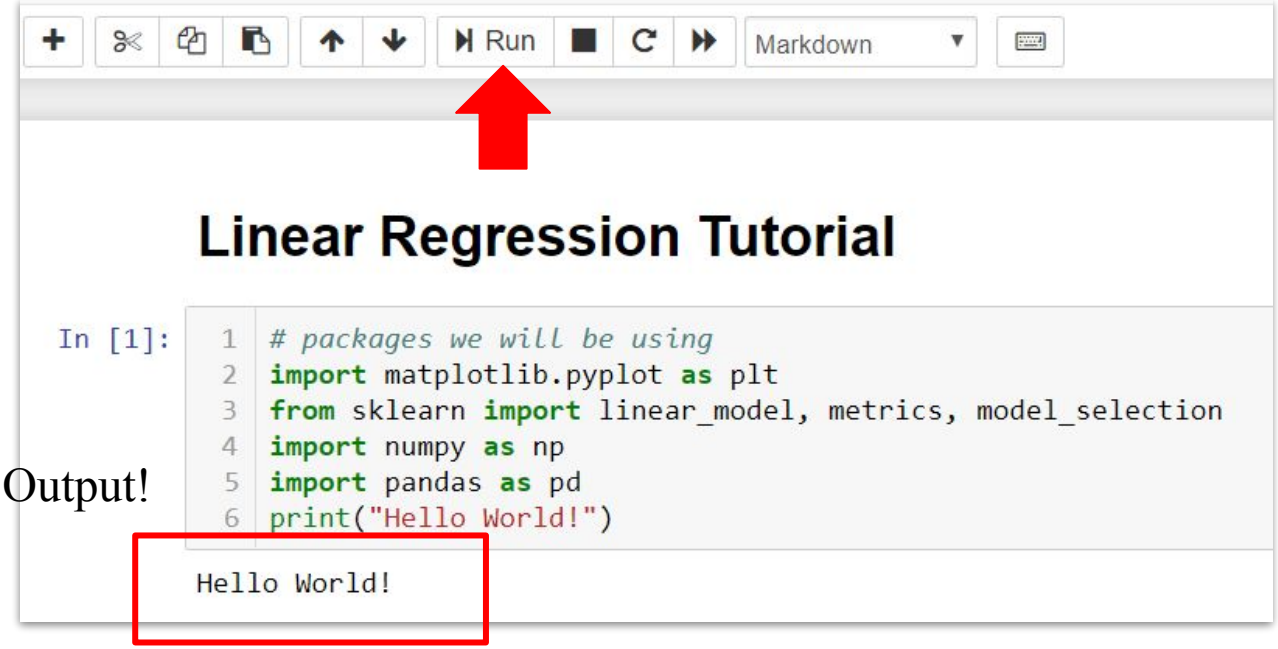
Let's generate some example data for this case and examine the relationship between $\mathbf{x}$ and y.

In [2]:

```
num_hours_studied = np.array([1, 3, 3, 4, 5, 6, 7, 7, 8, 8, 10])
exam_score = np.array([18, 26, 31, 40, 55, 62, 71, 70, 75, 85, 97])
plt.scatter(num_hours_studied, exam_score)
plt.xlabel('num_hours_studied')
plt.ylabel('exam_score')
plt.show()
```

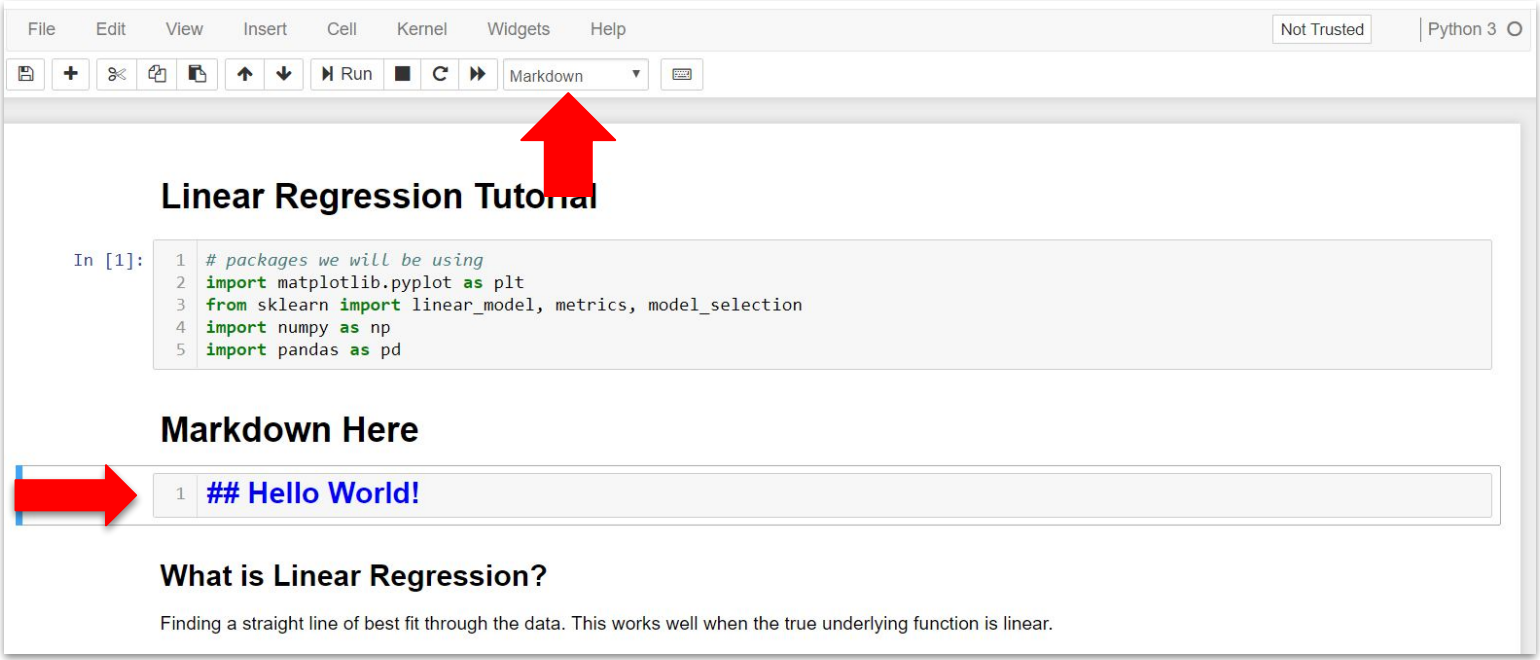
Jupyter

- Run your code in current cell



Jupyter

- You can also write Markdown syntax in the cell



Jupyter

- Run the Markdown cell to get the output

Linear Regression Tutorial

In [1]:

```
# packages we will be using
import matplotlib.pyplot as plt
from sklearn import linear_model, metrics, model_selection
import numpy as np
import pandas as pd
```

Markdown Here

Hello World!

What is Linear Regression?

Finding a straight line of best fit through the data. This works well when the true underlying function is linear.