USC Viterbi
School of Engineering
Daniel J. Epstein Department of Industrial and Systems Engineering

Predictive Analytics (ISE529)

Introduction

Dr. Tao Ma
ma.tao@usc.edu

Tue/Thu, Aug 26 - Dec 6, 2024, Fall

USC University of Southern California

Course Management

- **Summary of take-aways**
  - Quantitative (numerical) Prediction
    - Simple/Multiple Linear Regression
    - Polynomial Regression
    - Ridge/Lasso Regression
    - Principal Components Regression
    - Partial Least Squares
  - Qualitative (categorical) Prediction, i.e., classification
    - Logistic Regression, Poisson Regression
    - Linear Discriminant Analysis
    - Quadratic Discriminant Analysis
    - Naive Bayes
    - Support Vector Machines
  - Tree-Based Methods
    - Decision Trees
    - Bagging, Random Forests, Boosting
  - Deep Learning
    - Multilayer Neural Networks
    - Recurrent Neural Networks
    - Convolutional Neural Networks

Course Management

- **Prerequisites**
  - Probability and Statistics
  - Linear Algebra
  - Working knowledge on Python programing
- **Programming**
  - Python
  - Libraries (NumPy, Pandas, Scikit Learn, and Statsmodels, etc.)
  - Development Environment (the place where you write/run script)
    - Jupyter Lab or Notebook
    - PyCharm (community or professional edition)
    - Many others …
  - Platforms running Python script
    - Personal laptop (Windows or Linux OS)

    Alternative Platforms:

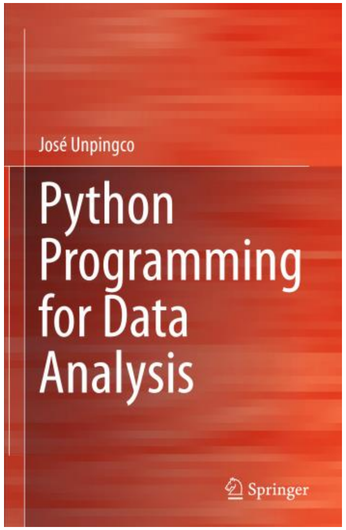
    - The Viterbi MyDesktop (VDI or Virtual Desktop Infrastructure)
      https://viterbiit.usc.edu/instructional-support/viterbi-mydesktop-vdi/
      connect via HTML access: https://mydesktop.vlab.usc.edu/ with your USCNetID account
    - HPC in USC-CARC (Center of Advanced Research Computing)
      - Linux platform without GUI (graphical user interface), submit script via Slurm
      https://www.carc.usc.edu/services/computing/hpc

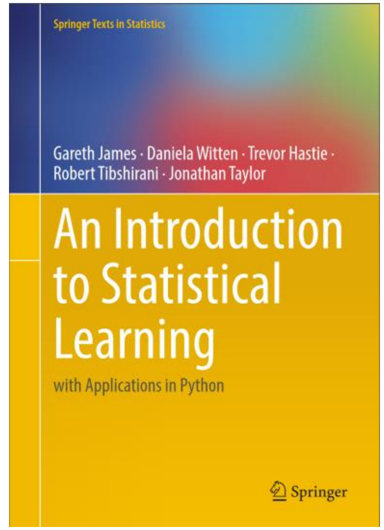
Course Management

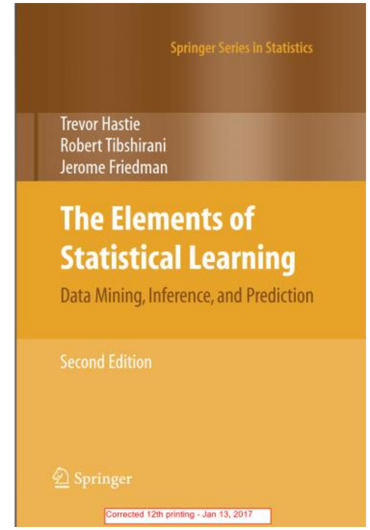
- **Further points**
  - All course materials (lecture slides, homework, labs/exercises, etc.) will be distributed via Brightspace
  - Submit your assignments to Brightspace. No late assignments will be accepted. Do NOT send your assignments via email to me after the due date.
  - Read textbook
    - James, Gareth et. al., An Introduction to Statistical Learning with Applications in Python (**ISLP**), Springer 2023, ISBN 978-3-031-38746-3. (free download)
    - José Unpingco, Python Programming for Data Analysis, Springer 2021, ISBN 978-3-030-68951-3
  - **Attend the lectures and take notes**. The formula derivation and solutions of homework assignments will mostly be presented on black/white board during the lecture time.
  - The distribution of final grade
    - Homework – 50%
    - Midterm exam – 20%
    - Final exam – 30%
    - Up to two points may be added to the overall grade based on class participation.

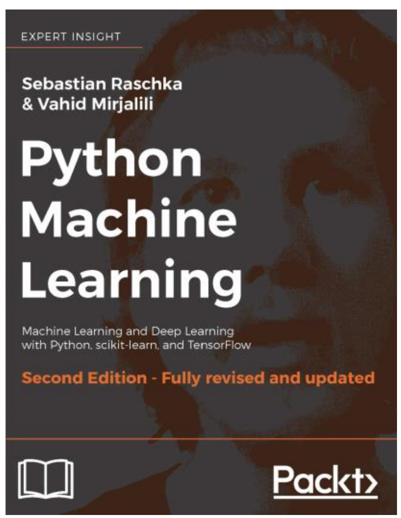
Course Texts

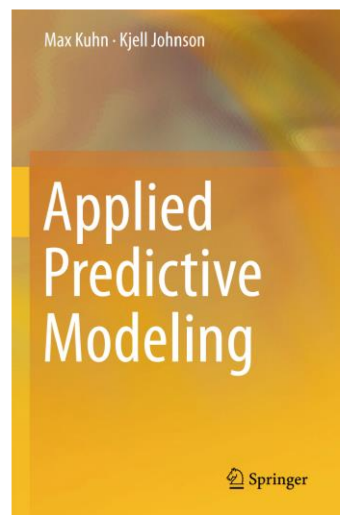
Required











PROGRAMING ENVIRONMENT

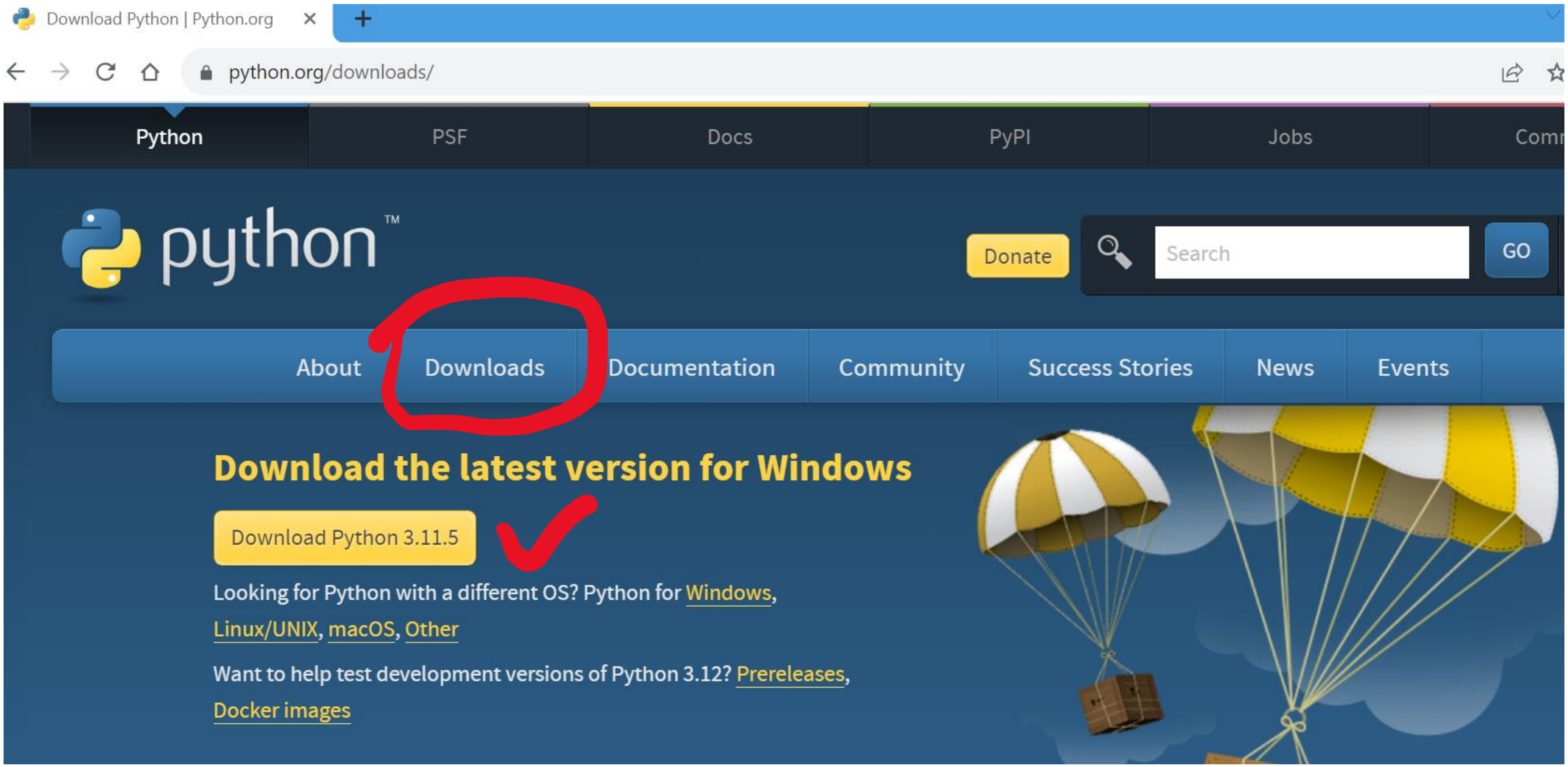
Quick Review on Python

- Installation
  - Install Python interpreter
  - Install third-party libraries (pip package manager)
- Programming Environment
  - Jupyter Lab/Notebook
  - PyCharm
- Syntax Examples (ISLP Chapter 2 *Lab*)
  - Use existing library
  - Read/write file
  - Build model and do computation
  - Show results: print and plot, write them to file
  - Create your own function (mostly you do need to do it.)

Install Python in Windows OS

Download Python from the website: https://www.python.org/

It is the power engine that interprets the scripts line by line.



Install Python in Windows OS

USC Viterbi
School of Engineering

You can also find older release and download. Choose the version to be compatible with the third-party libraries you need.

Looking for a specific release?

Python releases by version number:

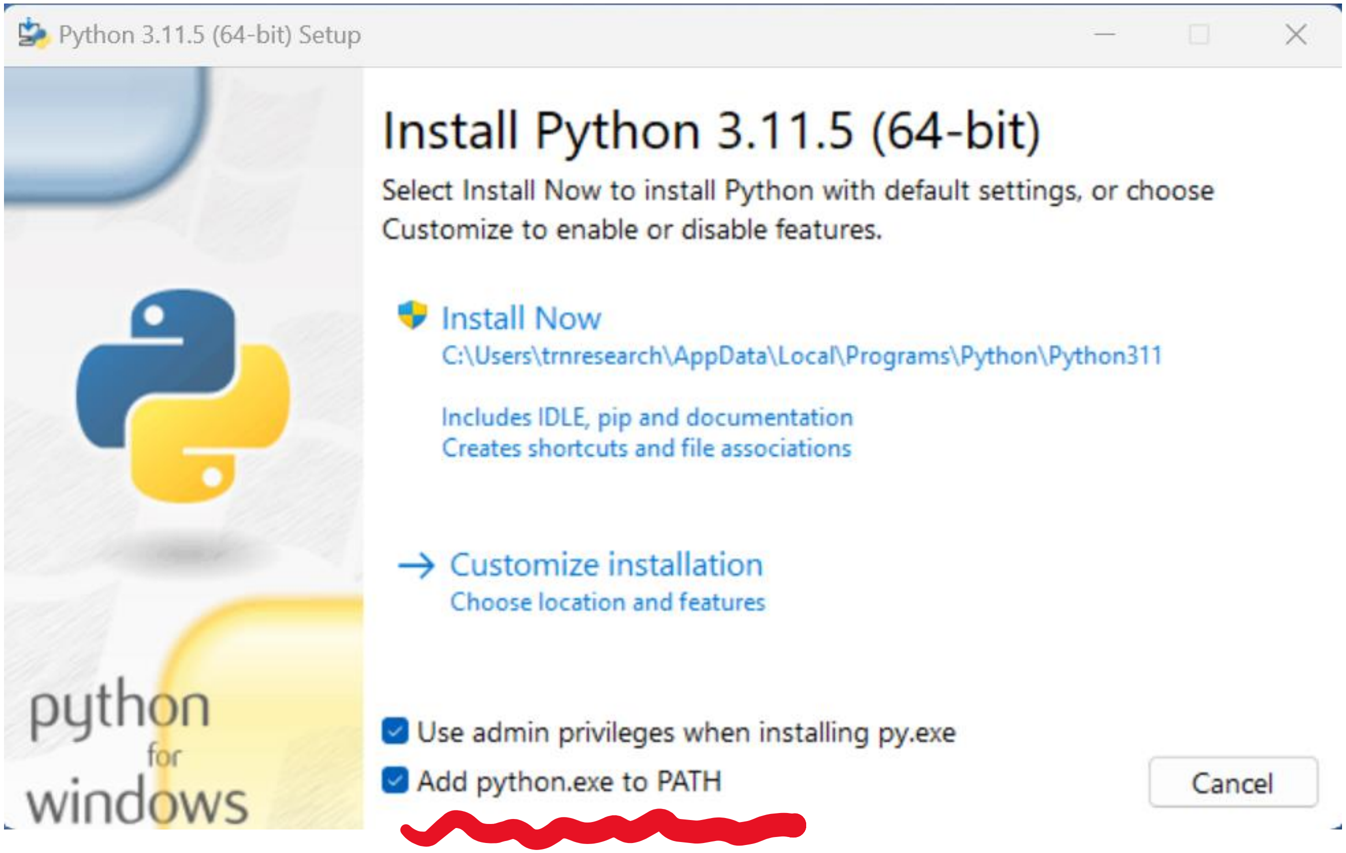
Release version	Release date		Click for more
Python 3.11.5	Aug. 24, 2023	Download	Release Notes
Python 3.10.13	Aug. 24, 2023	Download	Release Notes
Python 3.9.18	Aug. 24, 2023	Download	Release Notes
Python 3.8.18	Aug. 24, 2023	Download	Release Notes
Python 3.10.12	June 6, 2023	Download	Release Notes
Python 3.11.4	June 6, 2023	Download	Release Notes
Python 3.7.17	June 6, 2023	Download	Release Notes

View older releases

Install Python in Windows OS

USC Viterbi School of Engineering

In Windows OS, double click: python-3.11.5-amd64.exe



Install Python in Linux OS

Install Python in Linux OS (Ubuntu 22.04) from source file, follow the procedures below:

- Open a terminal
- Install system dependencies (required pre-requisite) first.

  $ sudo apt update

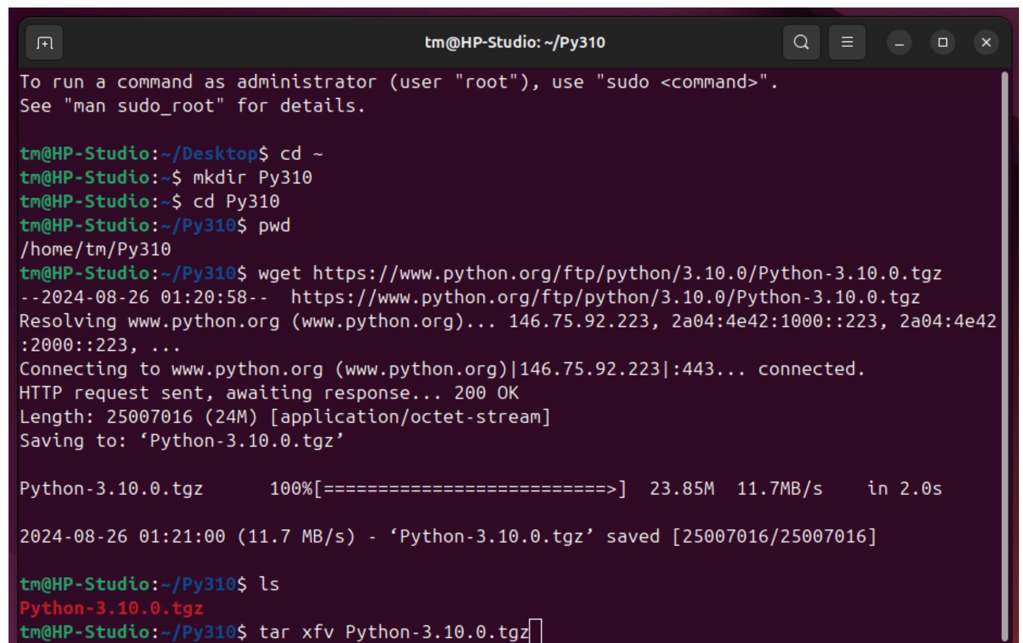
  $ sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev wget
- Download the source file to a folder

  $ mkdir /home/tm/Py310

  $ cd /home/tm/Py310

  $ wget https://www.python.org/ftp/python/3.10.0/Python-3.10.0.tgz
- Unzip the file :

  $ tar xfv Python-3.10.0.tgz



Install Python in Linux OS

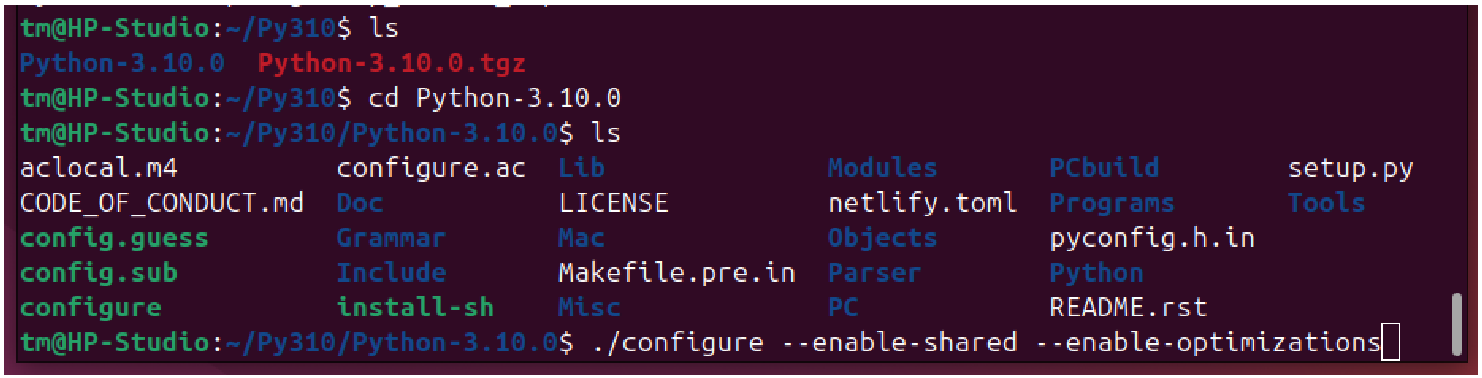
Install Python in Linux OS (Ubuntu 22.04) from source file, follow the procedures below:

- Install to system default folder: /usr/local/bin

  $ cd Python-3.10.0

  $ ls

  $ ./configure --enable-shared --enable-optimizations



  $ sudo make

  $ sudo make altinstall

- Add the line to .bashrc file

  export LD_LIBRARY_PATH="/usr/local/lib : LD_LIBRARY_PATH"

- Check your installation

  $ python --version

Install Python Library in Windows

What's pip?

The pip is a built-in packaging tool that is used to interact with a repository on the website of the Python Package Index (PYPI) https://pypi.org/, manage the installation or removal of third-party Python libraries (packages).

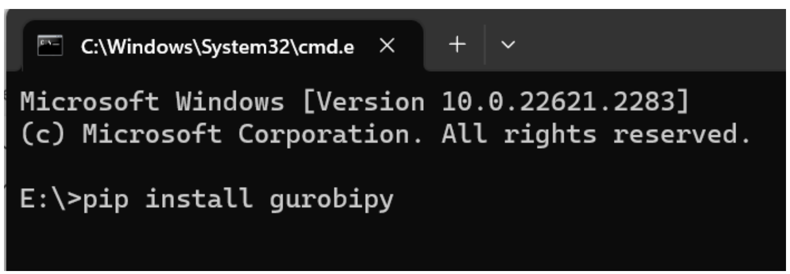
In Windows, open a command line terminal, using the command below to:

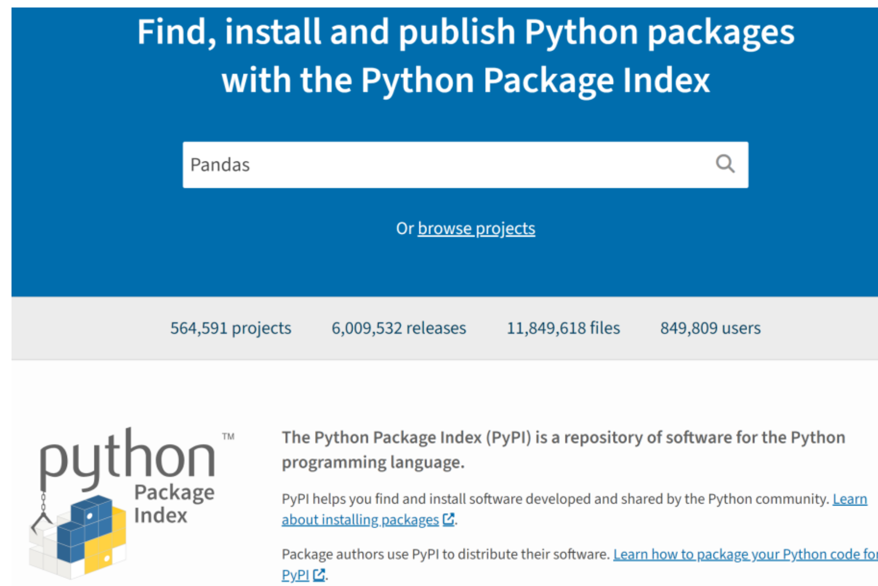
- install library:

> pip install NumPy, pandas, Matplotlib

- remove library:

> pip uninstall <name>





Install Python Libraries in Linux

Install third-party Python libraries in Linux OS (Ubuntu 22.04), follow the procedures below:

- Open a terminal

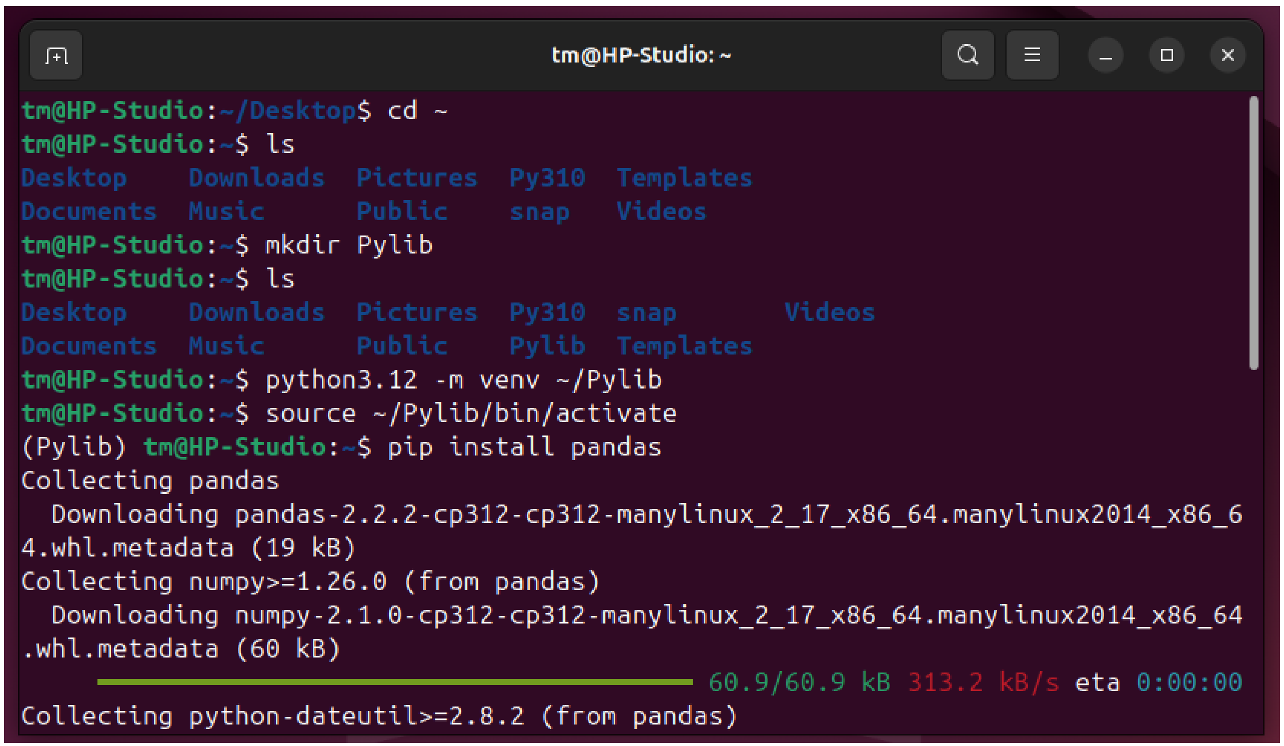
    $ mkdir Pylib

    $ python3.12 –m venv ~/Pylib

    $ source ~/Pylib/bin/activate

    $ pip install pandas

The packages are installed to: ~/Pylib/lib/python3.12/site-packages/



```
tm@HP-Studio:~/Desktop$ cd ~
tm@HP-Studio:~$ ls
Desktop    Downloads  Pictures  Py310  Templates
Documents  Music      Public    snap   Videos
tm@HP-Studio:~$ mkdir Pylib
tm@HP-Studio:~$ ls
Desktop    Downloads  Pictures  Py310  snap       Videos
Documents  Music      Public    Pylib  Templates
tm@HP-Studio:~$ python3.12 -m venv ~/Pylib
tm@HP-Studio:~$ source ~/Pylib/bin/activate
(Pylib) tm@HP-Studio:~$ pip install pandas
Collecting pandas
  Downloading pandas-2.2.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl.metadata (19 kB)
Collecting numpy>=1.26.0 (from pandas)
  Downloading numpy-2.1.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64
.whl.metadata (60 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.9/60.9 kB 313.2 kB/s eta 0:00:00
Collecting python-dateutil>=2.8.2 (from pandas)
```

Install Python and Libraries with Conda

USC Viterbi School of Engineering

What is Conda? https://conda.org/

Conda is an open-source package management system and environment management system. It allows you to install multiple versions of software packages and their dependencies and switch between them. It's particularly popular among data scientists because it makes it easy to manage Python and R packages.

- Install miniconda3 in **Windows OS** (win10/11), follow the procedures below:

  Download Miniconda3-latest-Windows-x86_64.exe

  https://docs.anaconda.com/miniconda/#miniconda-latest-installer-links

  Double click and follow the prompt to install it.

- Install Python and third-party libraries with Conda
- Open command line terminal

  > conda create –prefix=E:\Py310

  > conda activate E:\Py310

  (Py310) > conda install python == 3.10.0

  (Py310) > conda install matplotlib

  > conda deactivate Py310

The both Python and packages are installed to: ~/Py310

```
Anaconda Prompt (minico
E:\>conda create --prefix=E:\Py310
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: E:\Py310


Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate E:\Py310
#
# To deactivate an active environment, use
#
#     $ conda deactivate

E:\>conda activate E:\Py310

(E:\Py310) E:\>conda deactivate

E:\>
```

Install Python and Libraries with Conda

Install miniconda in **Linux OS** (Ubuntu 22.04), follow the procedures below:

- Open a terminal

  $ mkdir miniconda

  $ cd miniconda

  $ wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh

  $ chmod 755 Miniconda3-latest-Linux-x86_64.sh

  $ ./Miniconda3-latest-Linux-x86_64.sh

Add a line to .bashrc file, and save the file

  export PATH="~/miniconda/bin:$PATH"

  $ source .bashrc

- Install Python and third-party libraries with Conda

  $ conda create -n Py310

  $ conda activate Py310

  (Py310) [tm@HP-Studio ~]$ $ conda install python == 3.10.0

  (Py310) [tm@HP-Studio ~]$ conda install matplotlib

The both Python and packages are installed to: ~/Py310

Programing Environment

JupyterLab/Jupyter Notebook is an application that we use to interact with Python interpreter (engine). It is an open-source integrated development environment (IDE) that allows users to create and run the python script, show numerical and graphical results and markdown documentation.

https://jupyter.org/install

Install JupyterLab in both Windows and Linux with the command below in terminal:
> pip install jupyterlab

To launch JupyterLab with:
> jupyter lab

Install the classic Jupyter Notebook with:
> pip install notebook

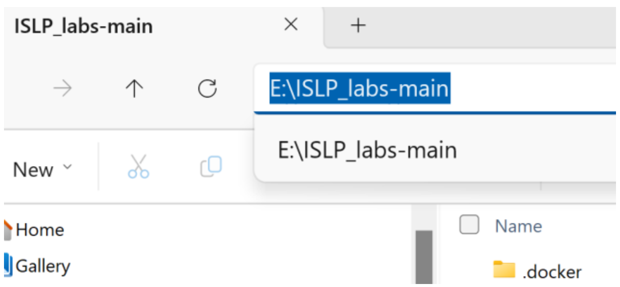
To launch the notebook with:
> jupyter notebook

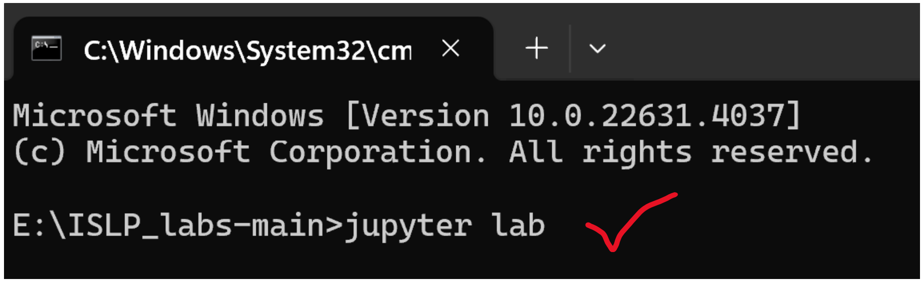
Programing Environment

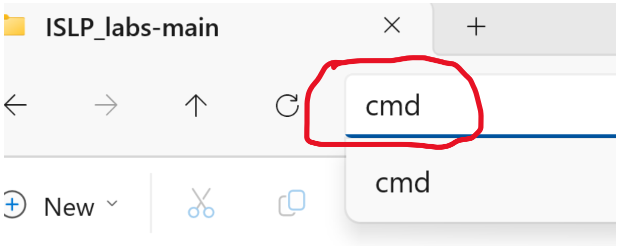
A few tips for **JupyterLab/Jupyter Notebook** :

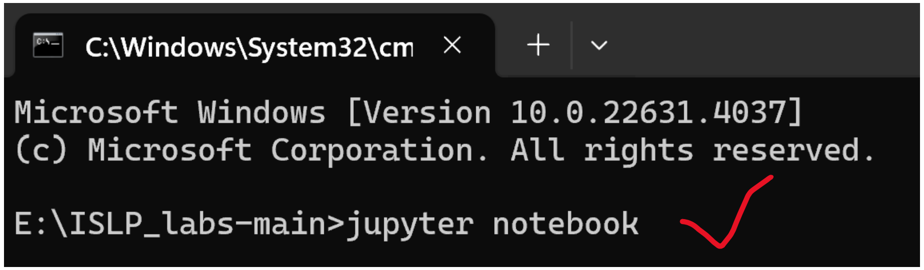
- Start JupyterLab/Notebook from the folder where you work on, i.e., move to the folder that includes all files you work on (data file, .ipynb, .py, etc.)

- In Windows OS, once move to the folder in **Windows Explorer**, open a command line terminal with:
  > cmd
  >Jupyter lab





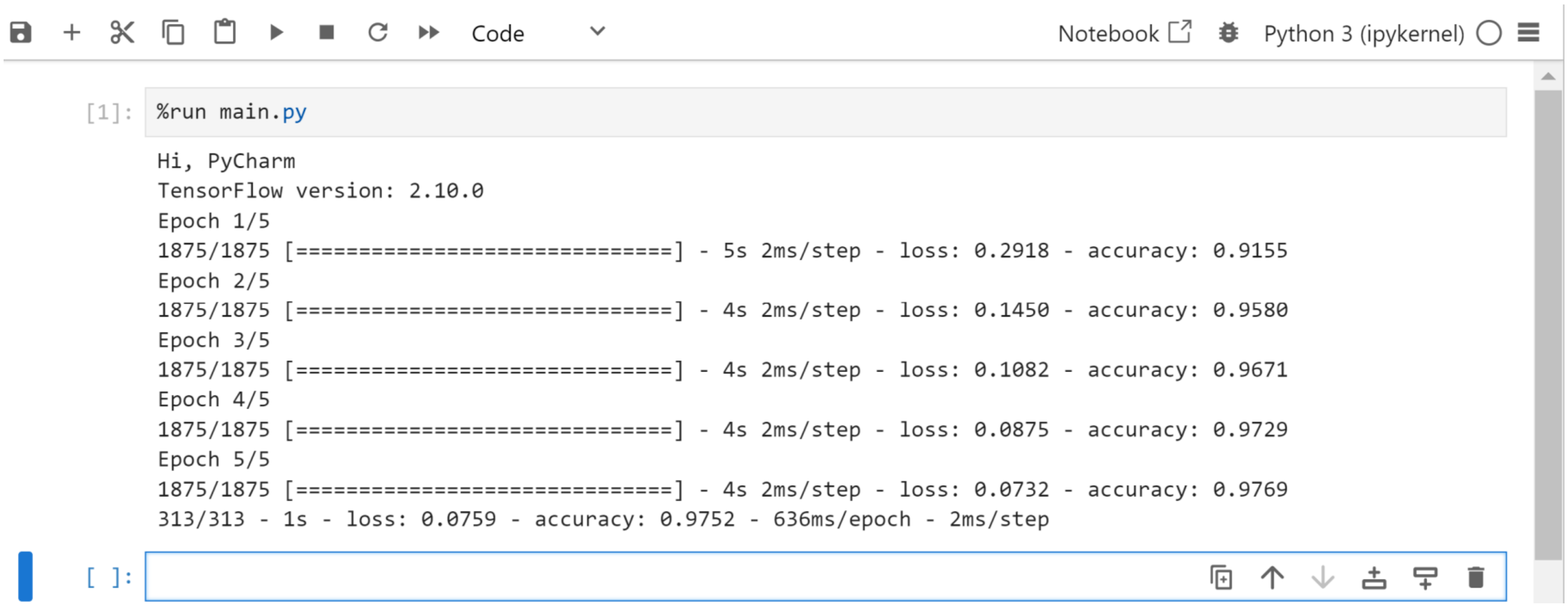




Programing Environment

A few tips for **JupyterLab/Jupyter Notebook** :

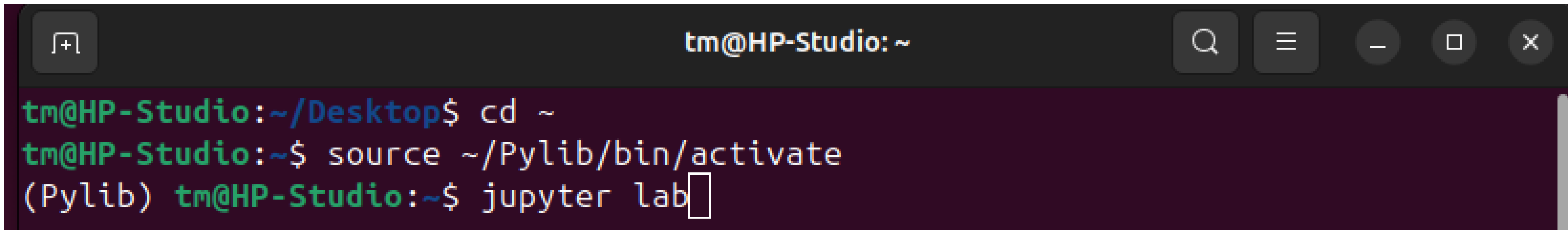
- To run .py file within JupyterLab or Notebook:
  %run <file name>.py

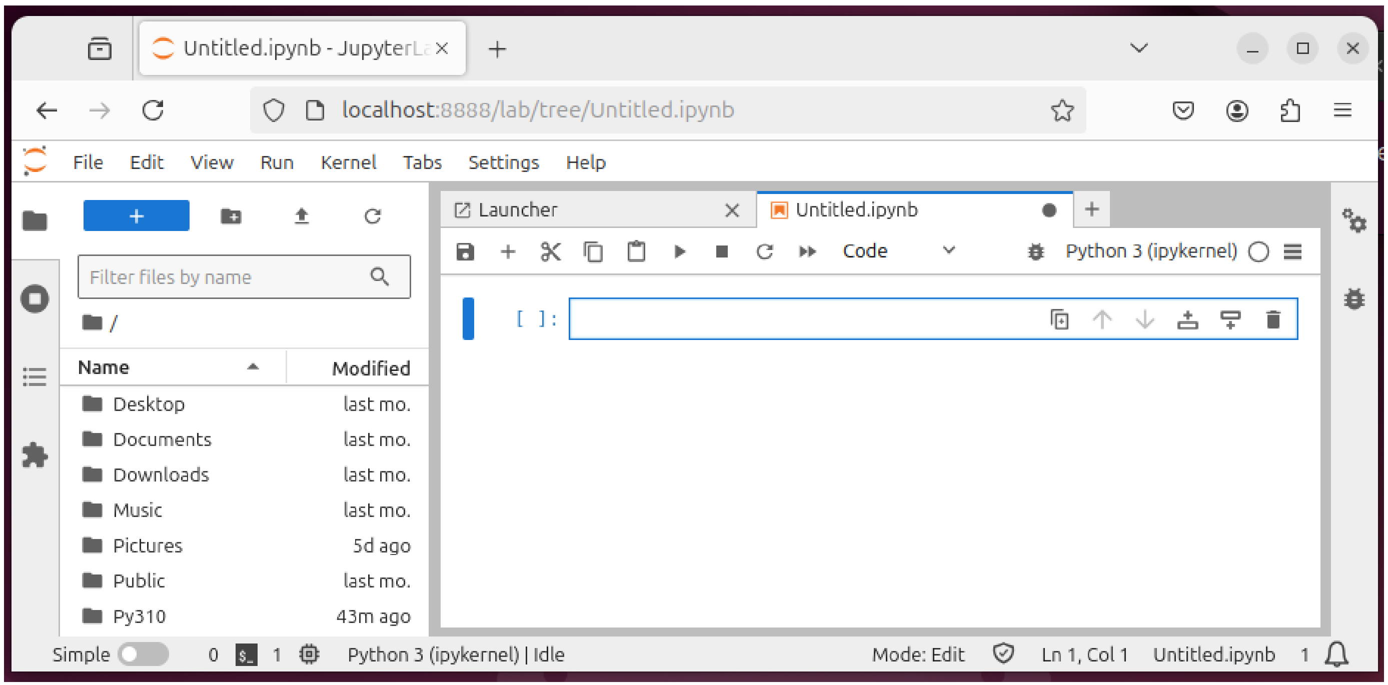


Programing Environment

A few tips for JupyterLab/Jupyter Notebook:

- In Linux OS, open a terminal, move to the folder and launch JupyterLab with:
  $Jupyter lab

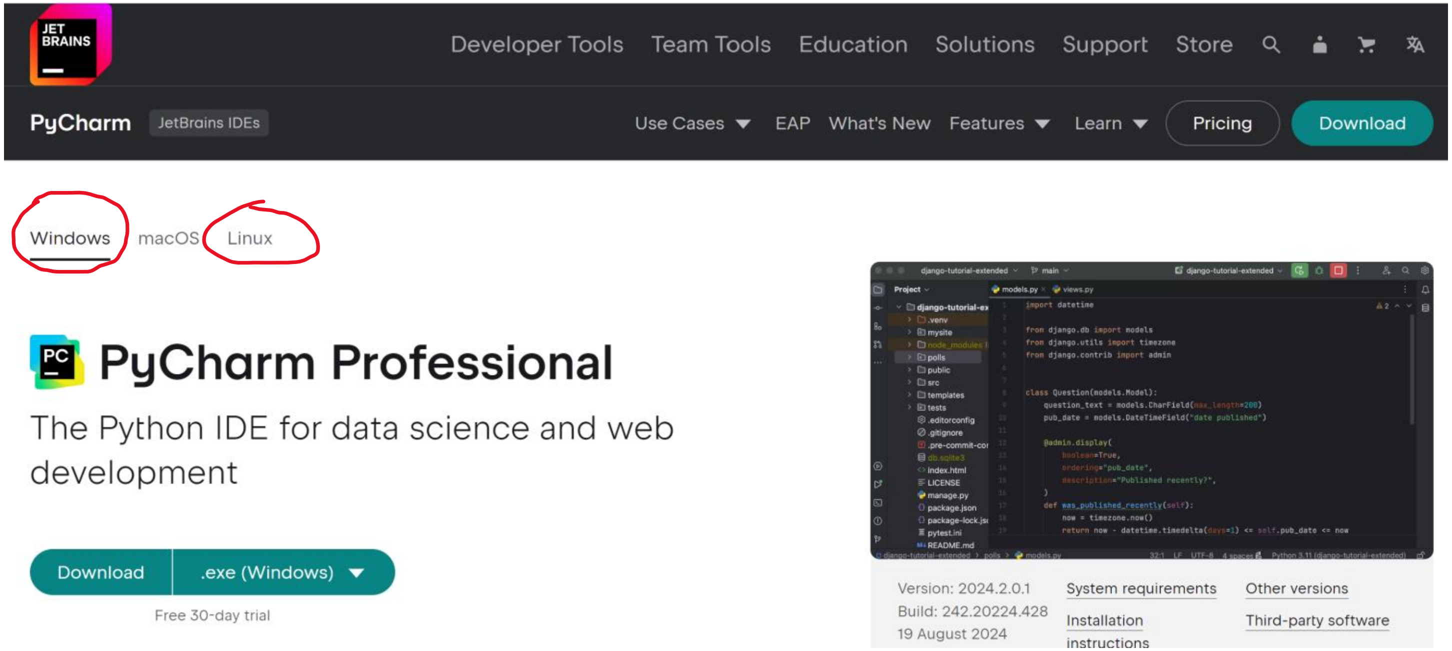




Programing Environment

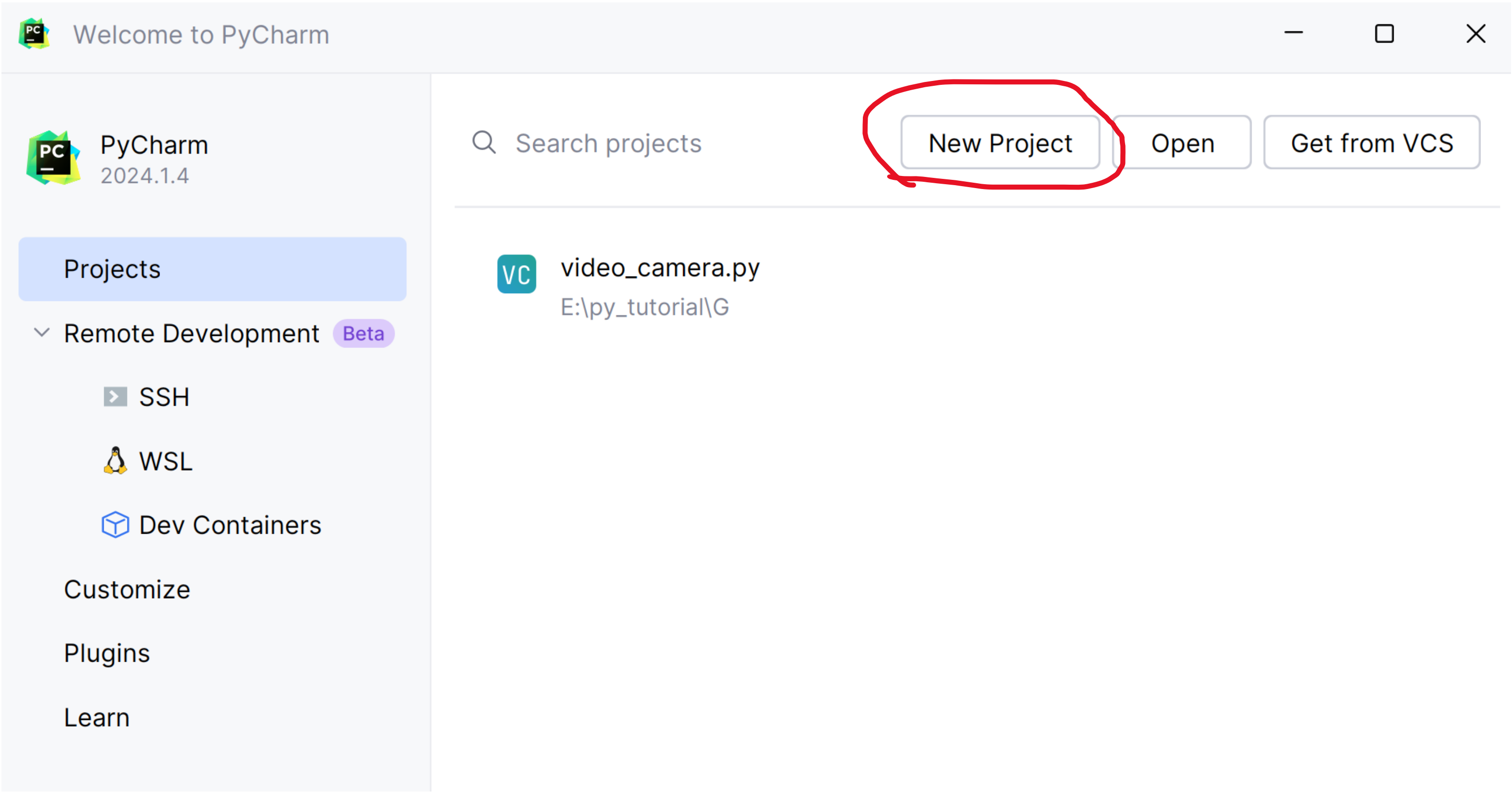
PyCharm is an excellent IDE (integrated development environment) for Python programing.

- You may need to register an academic/student account to get a free professional license (1 year, renewable).
- Download PyCharm
  https://www.jetbrains.com/pycharm/download/?section=windows
- Double click the .exe file to install



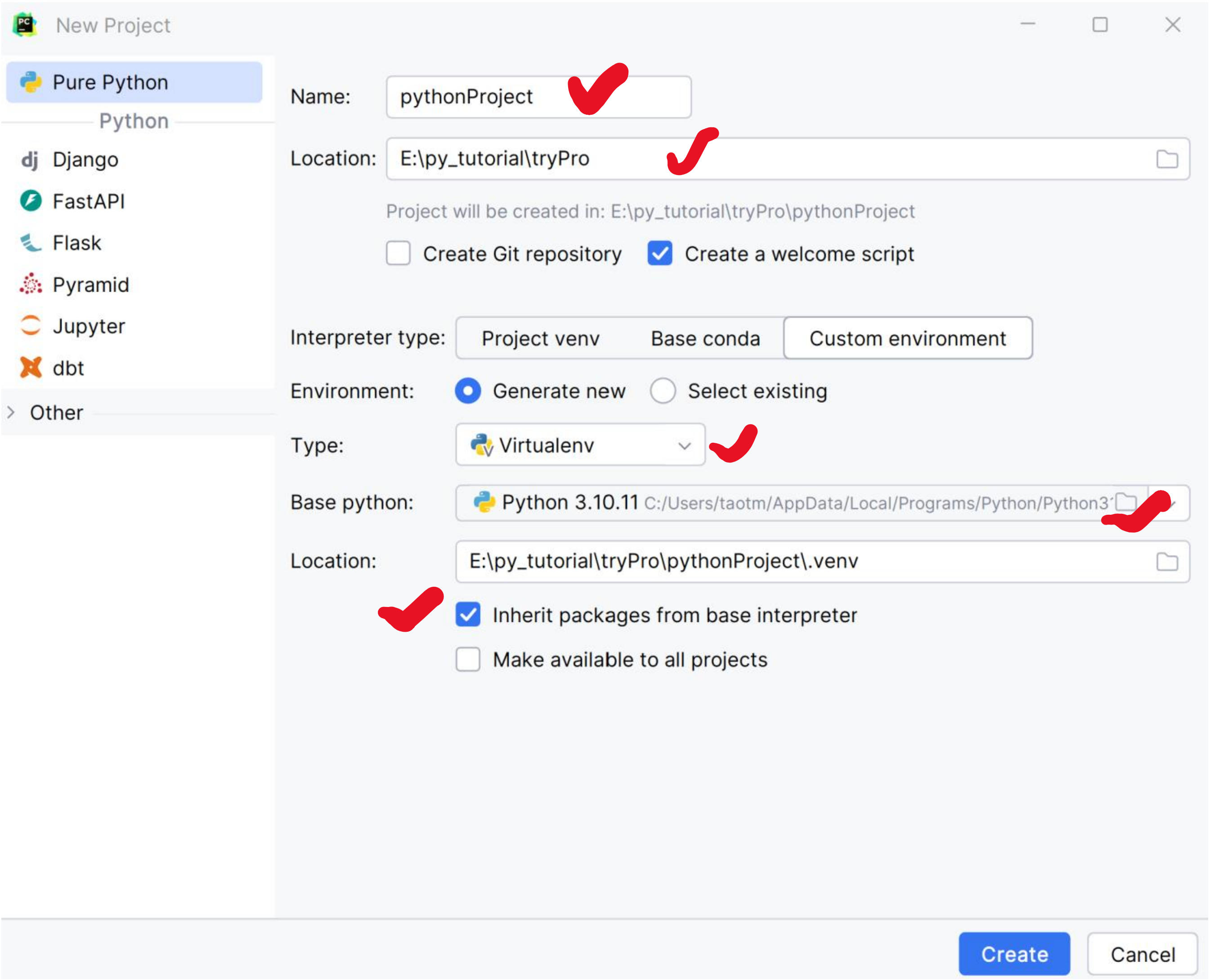
Create a new project with PyCharm

Click New Project to start



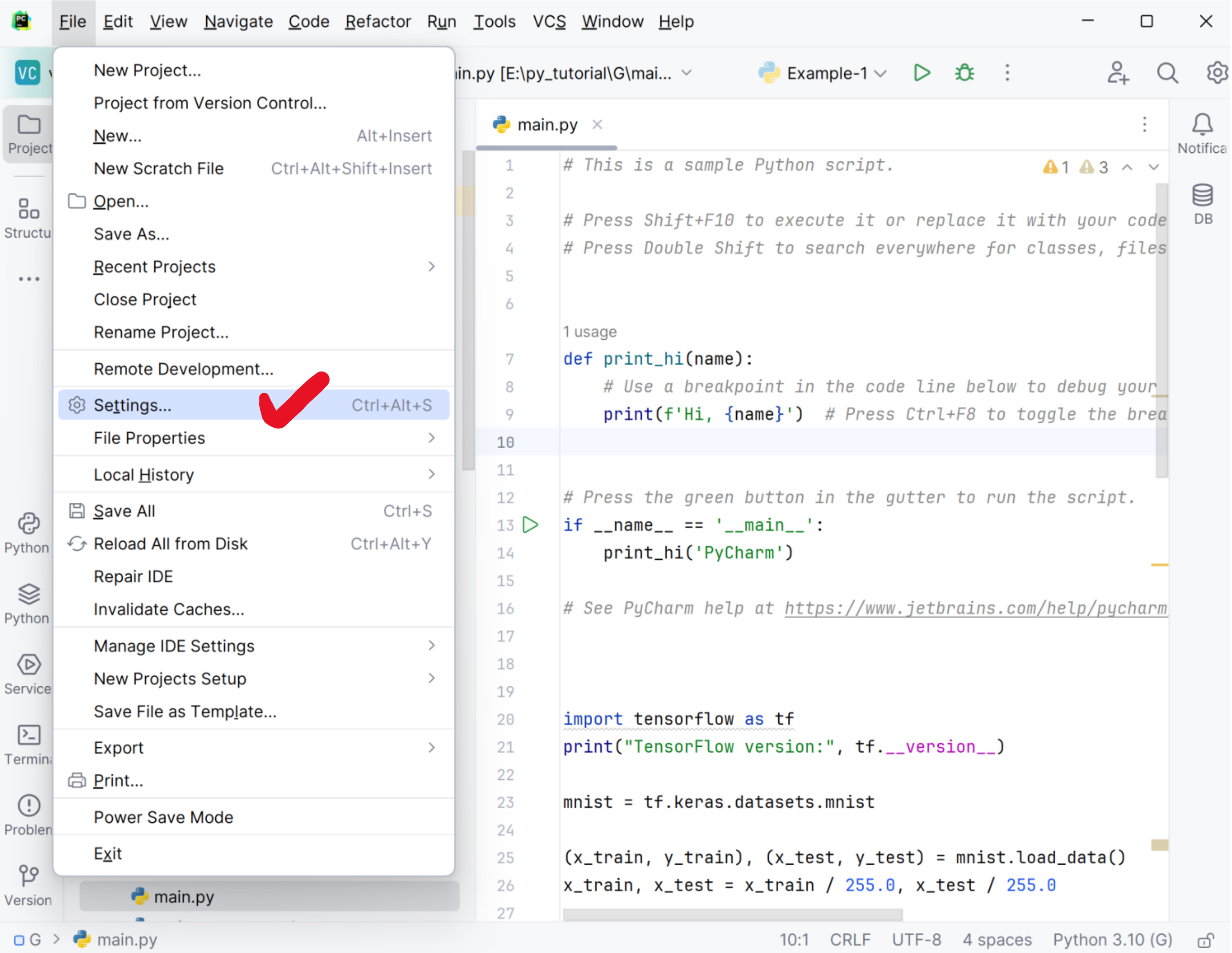
Create a new project with PyCharm

Choose the settings for your project.



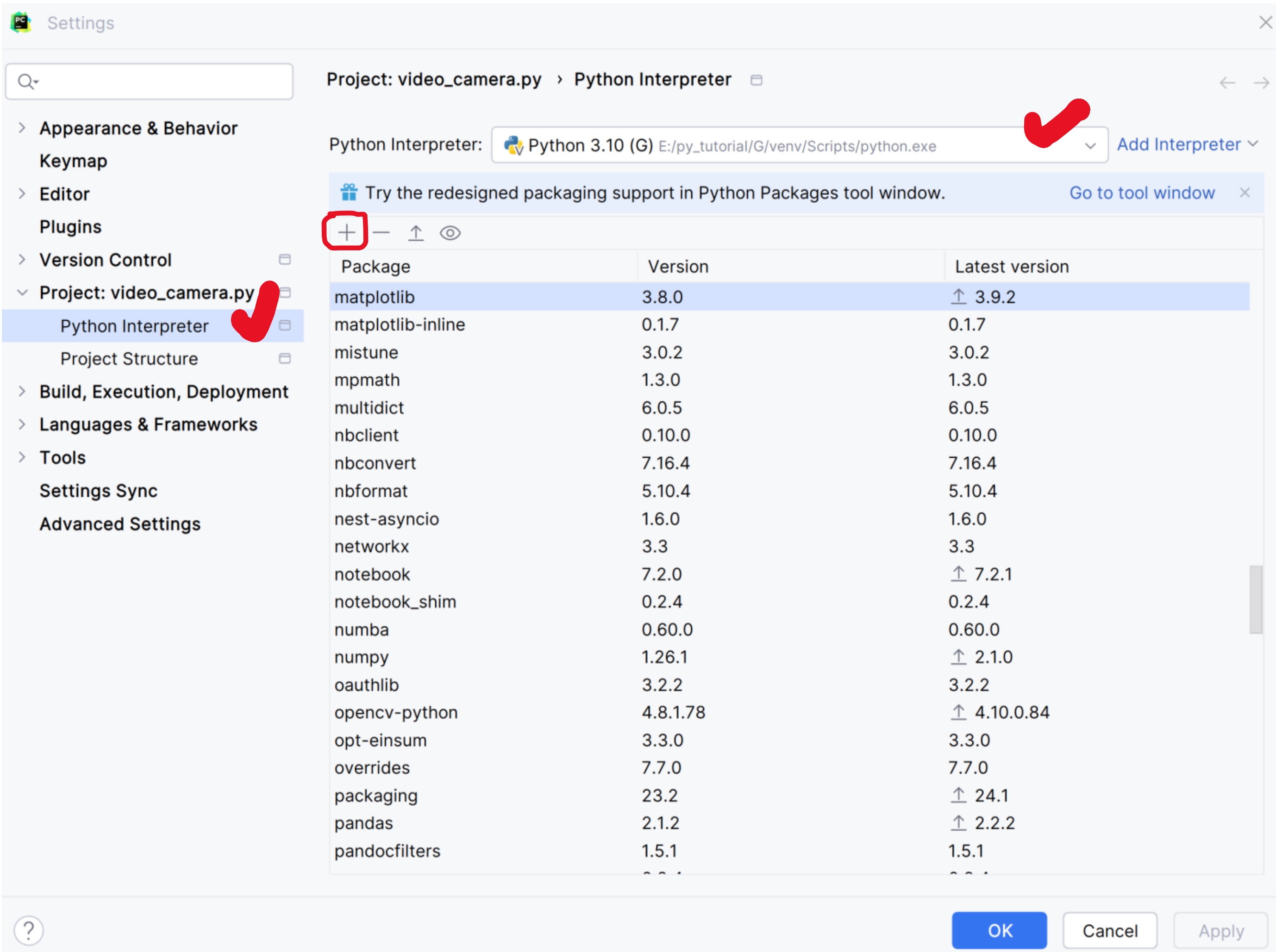
Create a new project with PyCharm

Go to: File -> Settings… to customize your project



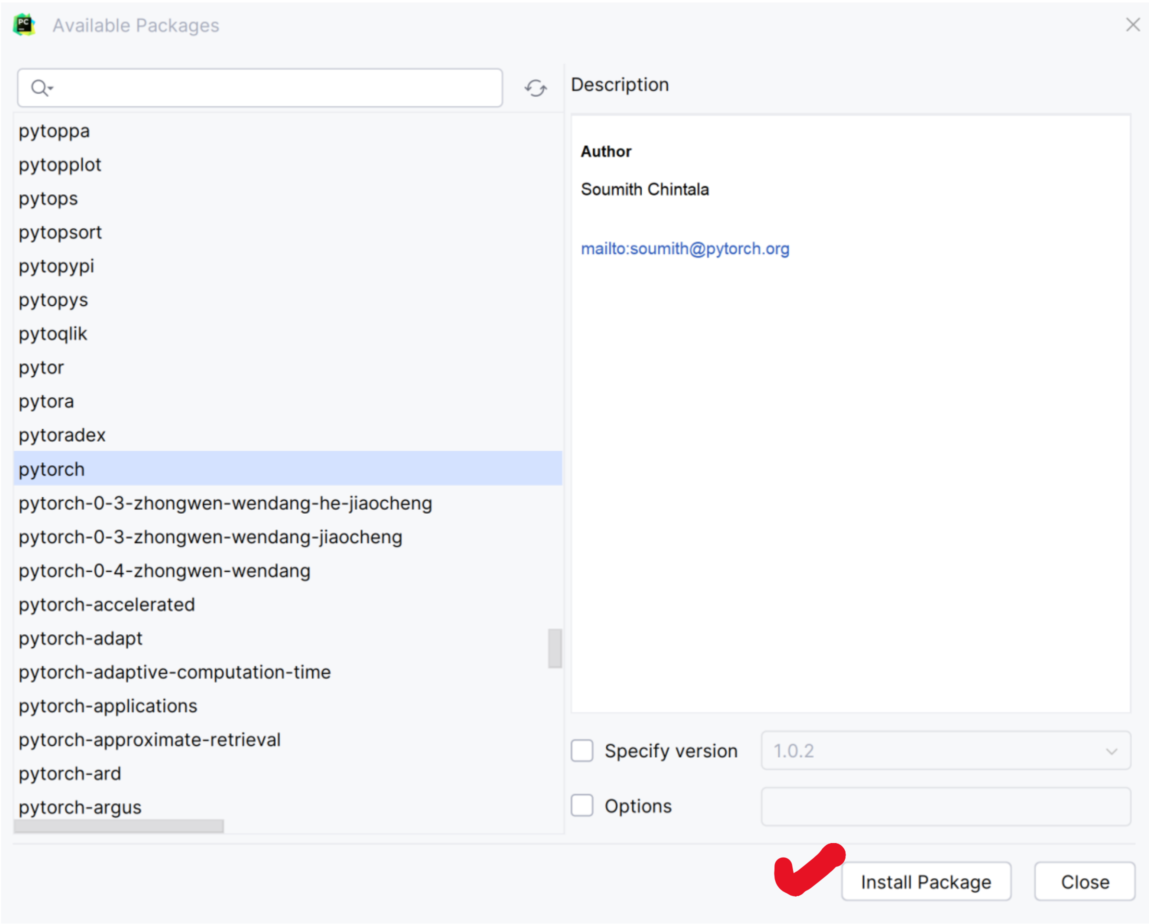
Create a new project with PyCharm

Set up environment in PyCharm, choose interpreter, show list of libraries, click "+" to add more libraries.



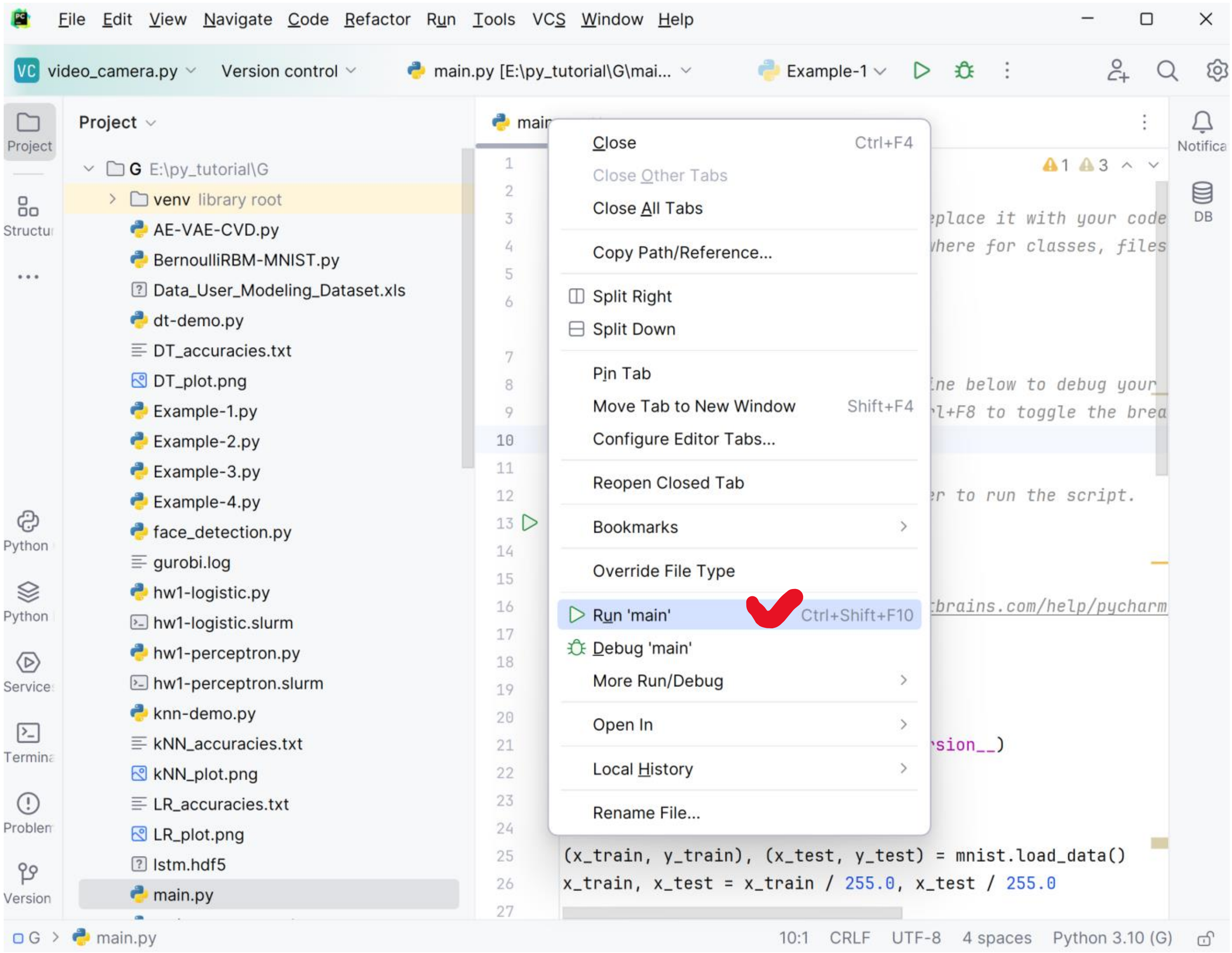
Create a new project with PyCharm

The left window shows the list of all libraries in the PYPI repository. Choose and install libraries you need.



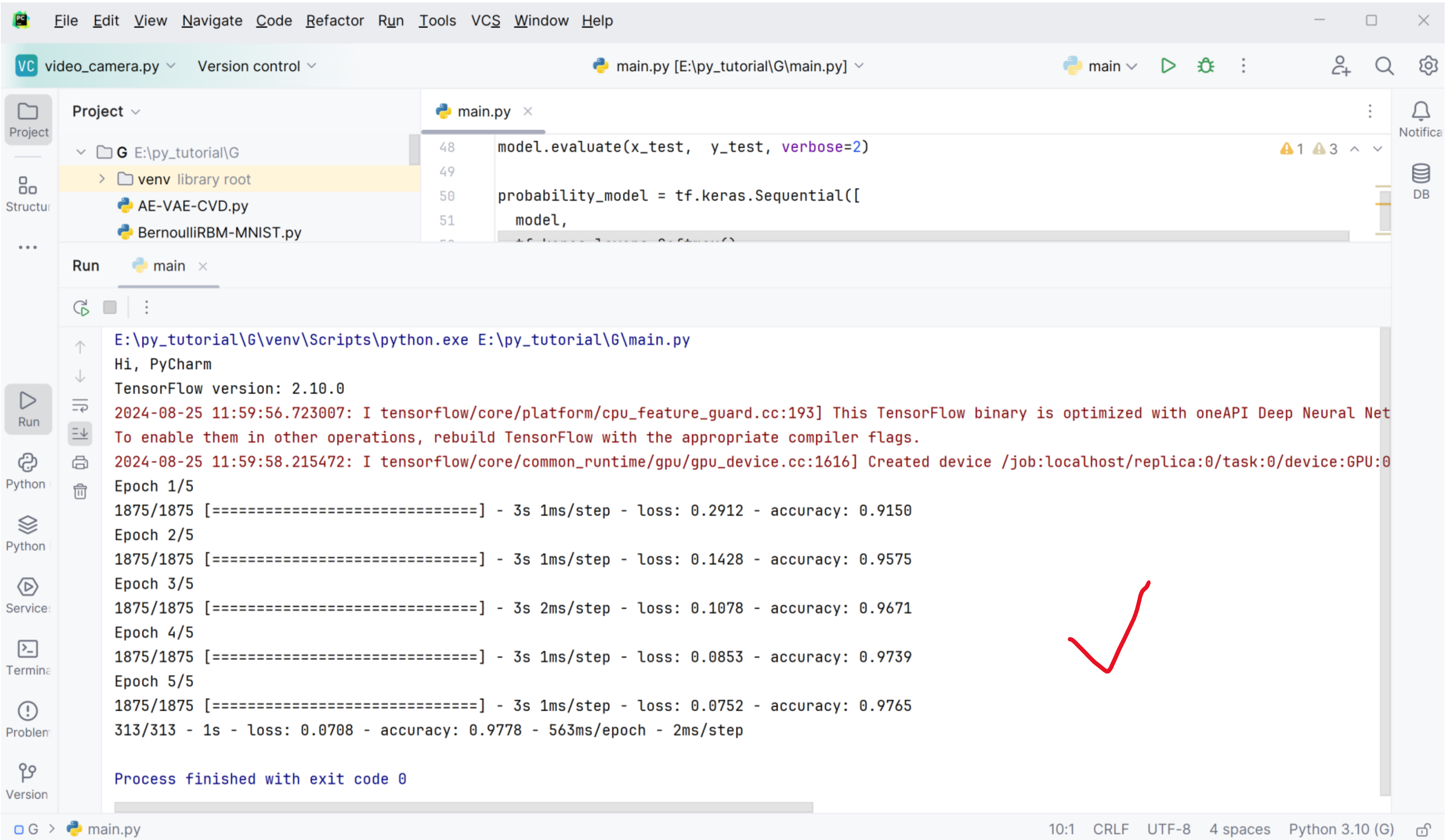
Create a new project with PyCharm

Move the cursor to main.py, right click your mouse, then from the menu click "Run"



Create a new project with PyCharm

Show the result in the window below.



Python Basic Syntax

- **ISLP** Chapter 2.3 Lab: Introduction to Python
- Learn Python Programming
  https://docs.python.org/3/tutorial/
  https://www.programiz.com/python-programming
  https://www.datacamp.com/courses/intro-to-python-for-data-science

At least you need to know the following operations **via Python script**:

- Use existing library (load/import)
- Read/write file
  - load your data to a variable (e.g., dataframe, matrix)
  - output and save the modeling results
- Call functions from existing libraries to specify your models and do computation
- Show results:
  - Print numerical results
  - Plot graphs, curves, lines
- Create your own function (mostly you do need to do it.)

Example

- Use existing library (OS, Pandas)
- Read/write file (load data/output result)

```
[1]: import os

[2]: os.getcwd()

[2]: 'C:\\Users\\taotm'

[3]: os.chdir("E:/py_tutorial/Gurobi")

[4]: os.getcwd()

[4]: 'E:\\py_tutorial\\Gurobi'

[5]: fobj = open('test.txt', 'w')
     fobj.write('foo\n')
     fobj.write('bar\n')
     fobj.close()

[6]: f = open("test.txt","r")
     eachLine = f.read()
     print(eachLine, "\n")
     f.close()

     foo
     bar
```

\+ ✂ Code

```
[10]: import numpy as np
      from matplotlib.pyplot import subplots
      import pandas as pd
      Auto = pd.read_csv('Auto.csv')
      Auto
```

[10]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite

Example

- Call functions from existing libraries to specify your models and do computation

[12]:
```python
import numpy as np
import pandas as pd
from matplotlib.pyplot import subplots
import statsmodels.api as sm
from statsmodels.stats.outliers_influence \
     import variance_inflation_factor as VIF
from statsmodels.stats.anova import anova_lm
from ISLP import load_data
from ISLP.models import (ModelSpec as MS,
                         summarize,
                         poly)
```

[14]:
```python
Boston = load_data("Boston")
Boston
```

[14]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	2.94	33.4

[24]:
```python
X = pd.DataFrame({'intercept': np.ones(Boston.shape[0]),
                  'lstat': Boston['lstat']})
X[:4]
```

[24]:

	intercept	lstat
0	1.0	4.98
1	1.0	9.14
2	1.0	4.03
3	1.0	2.94

[19]:
```python
y = Boston['medv']
model = sm.OLS(y, X)
results = model.fit()
summarize(results)
```

[19]:

	coef	std err	t	P>|t|
intercept	34.5538	0.563	61.415	0.0
lstat	-0.9500	0.039	-24.528	0.0

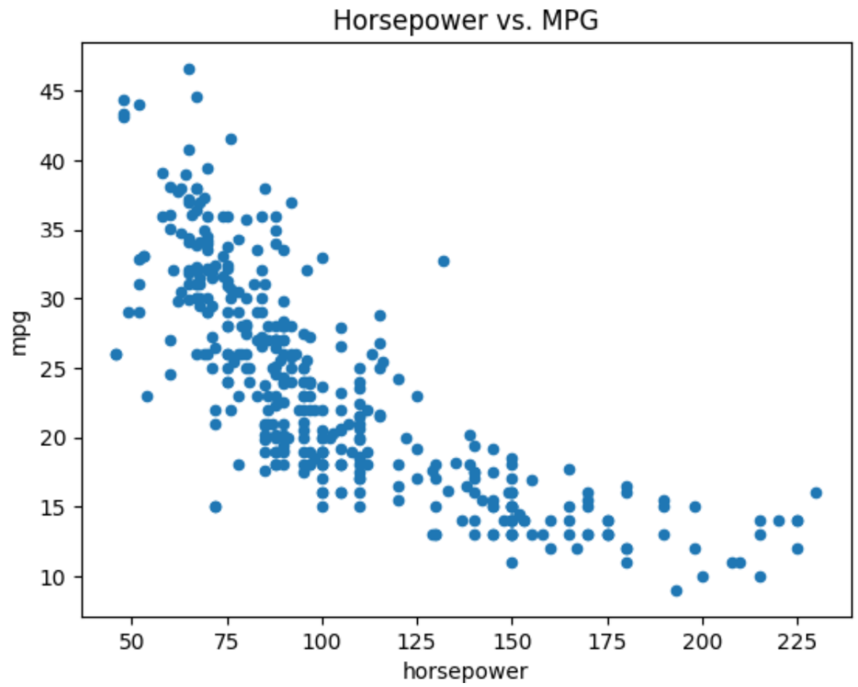
Example

- Show graphical and numerical results:
  - Print numerical results
  - Plot graphs, curves, lines

```
[5]: Auto['horsepower'].sum()
```

```
[5]: 40952
```

```
[7]: ax = Auto.plot.scatter('horsepower', 'mpg')
     ax.set_title('Horsepower vs. MPG');
```




```
[9]: Auto[['mpg', 'weight']].describe()
```

[9]:

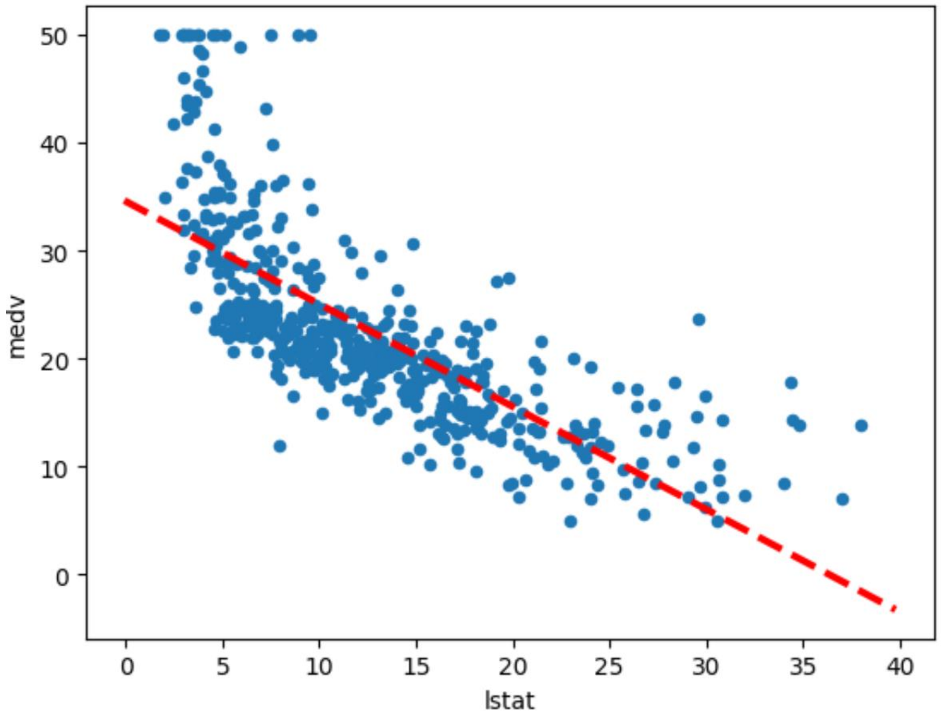
	mpg	weight
count	392.000000	392.000000
mean	23.445918	2977.584184
std	7.805007	849.402560
min	9.000000	1613.000000
25%	17.000000	2225.250000
50%	22.750000	2803.500000
75%	29.000000	3614.750000
max	46.600000	5140.000000

Example

- Create your own function

```
[22]: def abline(ax, b, m, *args, **kwargs):
          "Add a line with slope m and intercept b to ax"
          xlim = ax.get_xlim()
          ylim = [m * xlim[0] + b, m * xlim[1] + b]
          ax.plot(xlim, ylim, *args, **kwargs)
```

```
[25]: ax = Boston.plot.scatter('lstat', 'medv')
      abline(ax, results.params[0], results.params[1], 'r--', linewidth=3)
```



PROBABILITY REVIEW

Bernoulli trials

USC Viterbi School of Engineering

Definition
A random variable is called ***Bernoulli random variable*** *with parameter p if its probability mass function is given by equation*

$$p(x)=\begin{cases}1-p\equiv q & \textbf{\textit{if}}\ \ x=0\\ p & \textbf{\textit{if}}\ \ x=1\\ 0 & \textbf{\textit{otherwise}}\end{cases}$$

$$p(x)=\begin{cases}p^x(1-p)^{1-x} & \textbf{\textit{if}}\ \ x=0\ \textbf{\textit{or}}\ 1\\ 0 & \textbf{\textit{otherwise}}\end{cases}$$

For a Bernoulli random variable X with parameter p, $0 < p < 1$,

$$E(X)=p,\qquad Var(X)=p(1-p)$$

Example

If in a throw of a fair die the event of obtaining 4 or 6 is called a success, and the event of obtaining 1, 2, 3, or 5 is called a failure, then X is a Bernoulli random variable. Determine its probability mass function, its expected value $E(X)$ and variance Var(X).

Solution:

$$X = \begin{cases} 1 & \text{if 4 or 6 is obtained} \\ 0 & \text{otherwise} \end{cases}$$

The parameter p = 1/3. Therefore, its probability mass function is

$$p(x) = \begin{cases} 2/3 & \text{if } x = 0 \\ 1/3 & \text{if } x = 1 \\ 0 & \text{elsewhere.} \end{cases}$$

$E(X) = p = 1/3$, and $Var(X) = 1/3(1 - 1/3) = 2/9$.

Binomial Distribution

If ***n*** Bernoulli trials all with probability of success ***p*** are performed independently, then *X*, the number of successes, is called a **binomial** with parameters ***n*** and ***p***.

Theorem *Let X be a binomial random variable with parameters n and p. Then p(x), the probability mass function of X, is*

$$p(x) = P(X = x) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x} & \textbf{\textit{if}}\ x = 0,1,2,\cdots,n \\ 0 & \textbf{\textit{elsewhere}} \end{cases}$$

If *X* is a binomial random variable with parameters *n* and *p*, then

$$E(X) = np \qquad \textbf{\textit{and}} \qquad Var(X) = np(1-p)$$

Example

A restaurant serves 8 entrées of fish, 12 of beef, and 10 of poultry. If customers select from these entrées randomly, what is the probability that two of the next four customers order fish entrées?

Solution:

Let X denote the number of fish entrées (successes) ordered by the next four customers. Then X is binomial with the parameters (n=4, p=8/30 = 4/15). Thus

$$P(X=2)=\binom{4}{2}\left(\frac{4}{15}\right)^2\left(\frac{11}{15}\right)^2=0.23.$$

Poisson Distribution

By the French mathematician Simeon-Denis Poisson in 1837

Definition

A discrete random variable X with possible values 0, 1, 2, 3, . . . *is called* ***Poisson random variable*** *with parameter* **λ**, λ> 0, *if*

$$P\left(X = x\right) = \frac{e^{-\lambda}\lambda^{x}}{x!}, \qquad x = 0, 1, 2, 3, \cdots$$

If *X* is a Poisson random variable with parameter λ, then

$$E\left(X\right) = Var\left(X\right) = \lambda$$

Example

Suppose that, on average, in every three pages of a book there is one typographical error. If the number of typographical errors on a single page of the book is a Poisson random variable, what is the probability of at least one error on a specific page of the book?

Solution:

Let X be the number of errors on the page we are interested in. Then X is a Poisson random variable with $E(X) = 1/3$. Hence $\lambda = E(X) = 1/3$, and thus

$$P(X = n) = \frac{(1/3)^n e^{-1/3}}{n!}.$$

Therefore,

$$P(X \geq 1) = 1 - P(X = 0) = 1 - e^{-1/3} \approx 0.28.$$

Poisson Processes

If random events occur during time $\boldsymbol{t}$ in a way that for $t = 0$, $N(0) = 0$ and, for all $t > 0$, $0 < P[N(t) = 0] < 1$, then there exists a positive number $\boldsymbol{\lambda}$ such that

$$P\left(N\left(t\right)=n\right)=\frac{\left(\lambda t\right)^{n}e^{-\lambda t}}{n!}$$

That is, for all $t > 0$, **$N(t)$ is a Poisson random variable** with parameter $\boldsymbol{\lambda t}$.

Hence $E[N(t)] = \lambda t$ and for unit time $t = 1$, $E[N(1)] = \lambda$.

Example

Suppose that children are born at a Poisson rate of **five** per day in a certain hospital. What is the probability that

(a) at least two babies are born during the next six hours;

(b) no babies are born during the next two days?

Solution:

Let $N(t)$ denote the number of babies born at or prior to t. If we choose one day as time unit, then $\lambda = E[N(1)] = 5$. Therefore,

$$P\big(N(t) = n\big) = \frac{(5t)^n e^{-5t}}{n!}.$$

Hence for (a),

$$P\big(N(1/4) \geq 2\big) = 1 - P\big(N(1/4) = 0\big) - P\big(N(1/4) = 1\big)$$

$$= 1 - \frac{(5/4)^0 e^{-5/4}}{0!} - \frac{(5/4)^1 e^{-5/4}}{1!} \approx 0.36,$$

for (b),

$$P\big(N(2) = 0\big) = \frac{(10)^0 e^{-10}}{0!} \approx 4.54 \times 10^{-5}.$$

Where is normal distribution from?

In search of formulas to approximate binomial probabilities, Poisson was not alone. In 1718, before Poisson, De Moivre had discovered approximation to a binomial random variable with parameters n and $p = 1/2$, which is completely different from Poisson's. In 1812 Laplace generalized it to binomial random variables with parameters n and p.

Theorem (De Moivre-Laplace Theorem) *Let X be a binomial random variable with parameters n and p. Then for any numbers a and b, a < b,*

$$\lim_{n\to\infty} P\left(a < \frac{X - np}{\sqrt{np(1-p)}} < b \right) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2} dx$$

The De Moivre-Laplace formula yields excellent approximations for values of n and p for which $np(1 - p) \geq \mathbf{10}$.

Where is normal distribution from?

By this theorem, if X is a binomial random variable with parameters (n, p), the sequence of probabilities

$$P\left(\frac{X - np}{\sqrt{np(1-p)}} \leq t\right), \qquad n = 1, 2, 3, 4, \ldots ,$$

converges to $\dfrac{1}{\sqrt{2\pi}} \displaystyle\int_{-\infty}^{t} e^{-x^2/2} dx$

thus,

$$\lim_{n\to\infty} P\left(\frac{X - np}{\sqrt{np(1-p)}} < t\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-x^2/2} dx$$

Where is normal distribution from?

Such that the function

$$\Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-x^2/2} dx$$

should be a distribution function (CDF) itself.

To prove that $\Phi(t)$ is a distribution function, **Gauss** showed that

$$I = \int_{-\infty}^{\infty} e^{-x^2/2}\, dx = \sqrt{2\pi},$$

and hence

$$\Phi(\infty) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-x^2/2} dx = 1$$

Therefore, $\Phi(t)$ is a distribution function.

Standard Normal Distribution

Definition *A random variable X is called* ***standard normal*** *if its distribution function is Φ, that is, if*

$$\Phi(t) = P(X < t) \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-x^2/2} dx$$

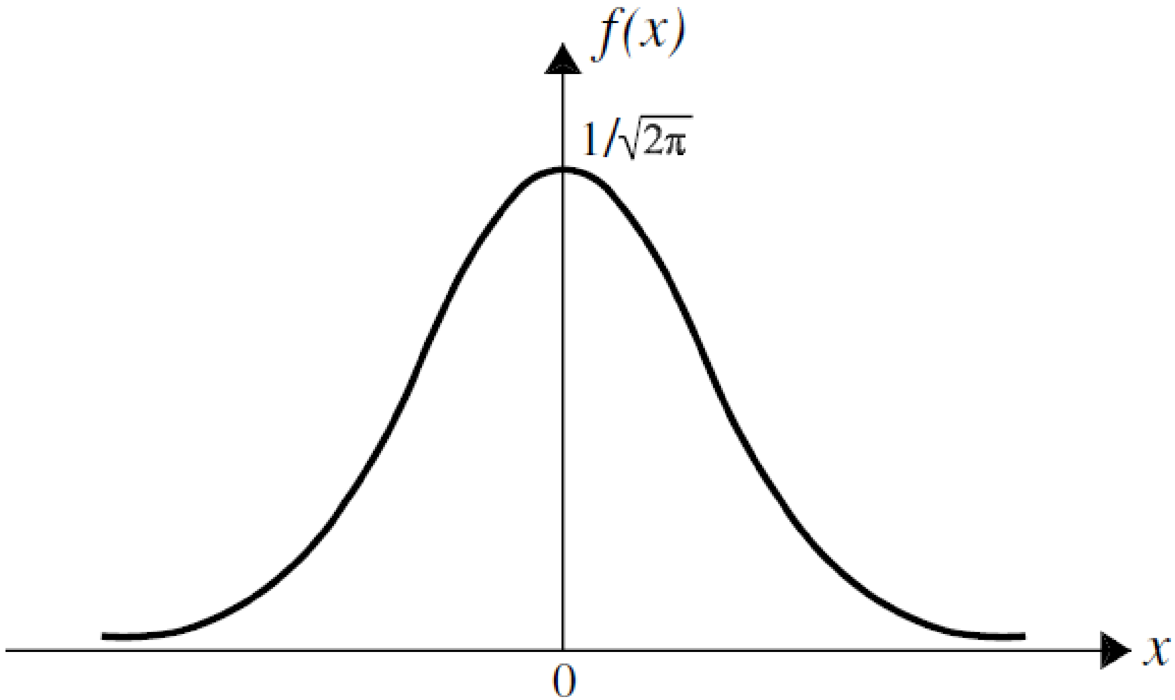
The density function *f* of a standard normal random variable, is given by

$$f(x) = \Phi'(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

Standard Normal Distribution

USC Viterbi School of Engineering

The standard normal density function is a bell-shaped curve that is symmetric about the y-axis.



$$\Phi(0)=1/2, \quad \Phi(\infty)=1, \quad \Phi(-t)=1-\Phi(t)$$

$$E(X)=0 \quad Var(X)=1$$

Normal Distribution

Definition

A random variable X is called ***normal****, with parameters μ and σ, if its density function is given by*

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{2\sigma^2}\right], \qquad -\infty < x < \infty.$$

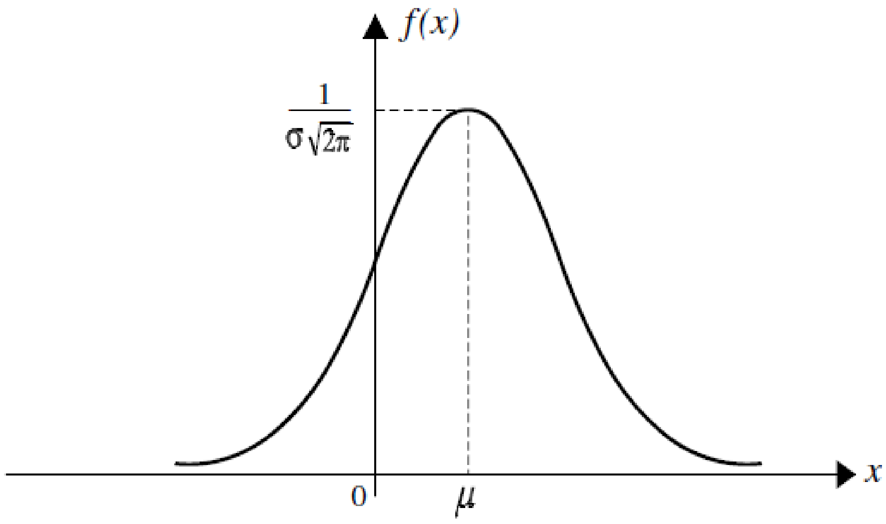
If X is a normal random variable with parameters μ and σ, we write $X \sim N(\mu, \sigma^2)$.

Lemma

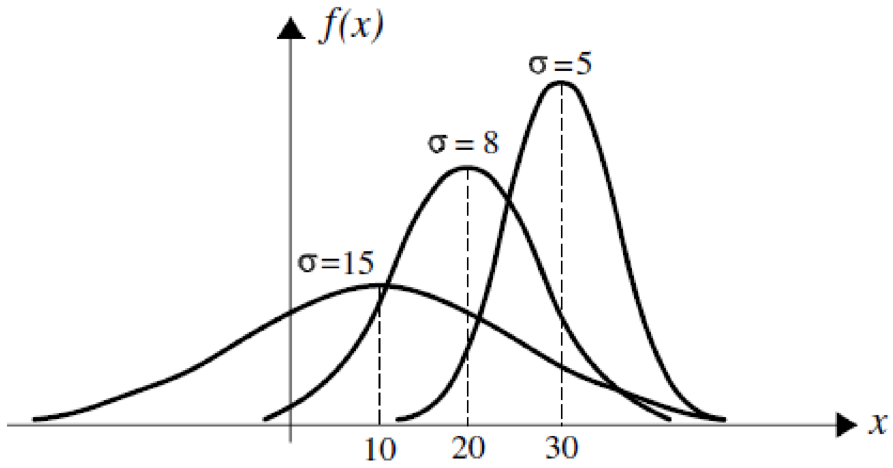
If $X \sim N(\mu, \sigma^2)$, then $Z = (X - \mu)/\sigma$ is $N(0, 1)$. That is, if $X \sim N(\mu, \sigma^2)$, the ***standardized*** *X is $N(0, 1)$.*

Normal Distribution

One of the first applications of $N(\mu, \sigma^2)$ was given by Gauss in 1809. Gauss used $N(\mu, \sigma^2)$ to model the errors of observations in astronomy. For this reason, the normal distribution is sometimes called the ***Gaussian distribution****.*



Density of $N(\mu, \sigma^2)$



Different normal densities with specified parameters

Example

Suppose that a Scottish soldier's chest size is normally distributed with mean 39.8 and standard deviation 2.05 inches, respectively. What is the probability that of 20 randomly selected Scottish soldiers, five have a chest of at least 40 inches?

Solution:
Let p be the probability that a randomly selected Scottish soldier has a chest of 40 or more inches. If X is the normal random variable with mean 39.8 and standard deviation 2.05, then

$$p = P(X \geq 40) = P\left(\frac{X - 39.8}{2.05} \geq \frac{40 - 39.8}{2.05}\right) = P\left(\frac{X - 39.8}{2.05} \geq 0.10\right)$$
$$= P(Z \geq 0.10) = 1 - \Phi(0.1) \approx 1 - 0.5398 \approx 0.46.$$

Example

Suppose that a Scottish soldier's chest size is normally distributed with mean 39.8 and standard deviation 2.05 inches, respectively. What is the probability that of 20 randomly selected Scottish soldiers, five have a chest of at least 40 inches?

Solution:
Therefore, the probability that of 20 randomly selected Scottish soldiers, five have a chest of at least 40 inches is

$$\binom{20}{5}(0.46)^5(0.54)^{15} \approx 0.03.$$

Conditional probability

Definition

If $P(B) > 0$, the conditional probability of A given B, denoted by $P(A|B)$, is

$$P(A \mid B) = \frac{P(AB)}{P(B)}.$$

Example

Suppose that all of the freshmen of an engineering college took calculus and discrete math last semester. Suppose that 70% of the students passed calculus, 55% passed discrete math, and 45% passed both. If a randomly selected freshman is found to have passed calculus last semester, what is the probability that he or she also passed discrete math last semester?

Bayes' formula

It is study of the calculation of *P(B|A)* in terms of *P(A|B)*

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} = \frac{P(A \mid B)P(B)}{P(A)}$$

Example

In a bolt factory, 30, 50, and 20% of production is manufactured by machines I, II, and III, respectively. If 4, 5, and 3% of the output of these respective machines is defective, what is the probability that a randomly selected bolt that is found to be defective is manufactured by machine III?

Bayes' formula

USC Viterbi School of Engineering

Solution

Let A be the event that a random bolt is defective.
Let B1 be the event that it is manufactured by machine I.
Let B2 be the event that it is manufactured by machine II.
Let B3 be the event that it is manufactured by machine III.

Solve P (B3 | A)

$$P(B_3 \mid A) = \frac{P(B_3 A)}{P(A)}, \qquad P(B_3 A) = P(A \mid B_3) P(B_3).$$

To calculate P (A), we must use the law of total probability.

$$P(A) = P(A \mid B_1) P(B_1) + P(A \mid B_2) P(B_2) + P(A \mid B_3) P(B_3).$$

Use Bayes' formula:

$$\begin{aligned} P(B_3 \mid A) &= \frac{P(B_3 A)}{P(A)} \\ &= \frac{P(A \mid B_3) P(B_3)}{P(A \mid B_1) P(B_1) + P(A \mid B_2) P(B_2) + P(A \mid B_3) P(B_3)} \\ &= \frac{(0.03)(0.20)}{(0.04)(0.30) + (0.05)(0.50) + (0.03)(0.20)} \approx 0.14. \end{aligned}$$

STATISTICS REVIEW

Expectation

Consider a casino game in which the probability of losing \$1 per game is 0.6, and the probabilities of winning \$1, \$2, and \$3 per game are 0.3, 0.08, and 0.02, respectively.

The gain or loss of a gambler who plays this game only a few times depends on his luck more than anything else.

However, if a gambler decides to play the game a large number of times, his loss or gain depends more on the number of plays than on his luck. If he plays the game n times, for a large n, then in approximately $(0.6)n$ games he will lose \$1 per game, and in approximately $(0.3)n$, $(0.08)n$, and $(0.02)n$ games he will win \$1, \$2, and \$3, respectively. Therefore, his total gain is

$$(0.6)n \cdot (-1) + (0.3)n \cdot 1 + (0.08)n \cdot 2 + (0.02)n \cdot 3 = (-0.08)n.$$

This gives an average of $\$-0.08$, or about 8 cents of loss per game.

Expectation

If X is the random variable denoting the gain in one play, then the number -0.08 is called the **expected value** of X. We write $E(X) = -0.08$. $E(X)$ is the average value of X. That is, if we play the game n times and find the average of the values of X, then as $n \to \infty$, $E(X)$ is obtained.

In this example, X is a discrete random variable with the set of possible values $\{-1, 1, 2, 3\}$. The probability mass function of X, $p(x)$, is given by

i	-1	1	2	3
$p(i) = P(X = i)$	0.6	0.3	0.08	0.02

$$(0.6) \cdot (-1) + (0.3) \cdot 1 + (0.08) \cdot 2 + (0.02) \cdot 3 = -0.08.$$

Hence,

$$-1 \cdot p(-1) + 1 \cdot p(1) + 2 \cdot p(2) + 3 \cdot p(3) = -0.08,$$

a relation showing that the **expected value** of X can be calculated directly by summing up the product of possible values of X by their probabilities.

Expectation

Definition

*The **expected value** of a discrete random variable X with the set of possible values* A *and probability mass function p*(*x*) *is defined by*

$$E(X) = \sum_{x \in A} x p(x)$$

We say that E(X) exists if this sum converges absolutely.

The expected value of a random variable X is also called the **mean**, or the **mathematical expectation**, or simply the **expectation** of X. It is also occasionally denoted by $E[X]$, EX, μ_X or μ.

Variance

Definition

Let X be a discrete random variable with a set of possible values A, *probability mass function p*(*x*), *and E*(*X*) = *μ. Then* σ_X *and Var*(X), *called the* ***standard deviation*** *and the* ***variance*** *of X, respectively, are defined by*

$$\sigma_X = \sqrt{E\left[\left(X-\mu\right)^2\right]} \quad \textbf{\textit{and}} \quad Var\left(X\right) = E\left[\left(X-\mu\right)^2\right]$$

Note that by this definition and Theorem 1

$$Var\left(X\right) = E\left[\left(X-\mu\right)^2\right] = \sum_{x\in A}\left(x-\mu\right)^2 p\left(x\right)$$

Variance

Example 9
Karen is interested in two games, Keno and Bolita. To play Bolita, she buys a ticket for \$1, draws a ball at random from a box of 100 balls numbered 1 to 100. If the ball drawn matches the number on her ticket, she wins \$75; otherwise, she loses. To play Keno, Karen bets \$1 on a single number that has a 25% chance to win. If she wins, they will return her dollar plus two dollars more; otherwise, they keep the dollar.

Let B and K be the amounts that Karen gains in one play of Bolita and Keno, respectively. Then

$$E(B) = (74)(0.01) + (-1)(0.99) = -0.25,$$
$$E(K) = (2)(0.25) + (-1)(0.75) = -0.25.$$

Variance

Example 9

Therefore, in the long run, it does not matter which of the two games Karen plays. Her gain would be about the same. However,

$$\mathrm{Var}(B) = E\left[(B-\mu)^2\right] = (74+0.25)^2(0.01) + (-1+0.25)^2(0.99) = 55.69$$

$$\mathrm{Var}(K) = E\left[(K-\mu)^2\right] = (2+0.25)^2(0.25) + (-1+0.25)^2(0.75) = 1.6875,$$

In Bolita, on average, the deviation of the gain from the expectation is much higher than in Keno. In other words, the risk with Keno is far less than the risk with Bolita. In Bolita, the probability of winning is very small, but the amount one might win is high. In Keno, players win more often but in smaller amounts.

Variance

Theorem 2

$$Var(X) = E(X^2) - [E(X)]^2$$

$$= \sum_{x \in A} x^2 p(x) - \mu^2$$

Theorem 3

Let X *be a discrete random variable, then for constants* **a** *and* **b** *we have that*

$$Var(aX + b) = a^2 Var(X)$$

$$\sigma_{aX+b} = |a| \sigma_X$$

Example

What is the expected number of floods a year?

What is the variance of RV X?

Number of flood	Probability	Cumulative Prob	xi * p(xi)	(xi-mean)^2*p(xi)
0	0.00	0.00	0	0
1	0.06	0.06	0.06	0.511584
2	0.18	0.24	0.36	0.663552
3	0.20	0.44	0.6	0.16928
4	0.26	0.70	1.04	0.001664
5	0.12	0.82	0.6	0.139968
6	0.03	0.85	0.18	0.129792
7	0.12	0.97	0.84	1.138368
8	0.03	1.00	0.24	0.499392
> 8	0.00	1.00	0	0
sum	1.00		3.92	3.2536

Expectation

Definition *If X is a continuous random variable with probability density function f , the expected value of X is defined by*

$$E(X) = \int_{-\infty}^{\infty} x f(x)\, dx$$

The expected value of X is also called the **mean**, or the **mathematical expectation**, or simply the **expectation** of X. It is denoted by $E[X]$, EX, μ_X, or μ.

Variance

Definition *If X is a continuous random variable with E(X) = μ, then Var(X) and* σ_X*, called the variance and standard deviation of X, respectively, are defined by*

$$Var(X) = E\left[(X-\mu)^2\right]$$

$$\sigma_X = \sqrt{E\left[(X-\mu)^2\right]}$$

Therefore, if *f* is the density function of *X*, then

$$Var(X) = E\left[(X-\mu)^2\right] = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)\,dx$$

Variance

The following important relations are analogous to those in the discrete case

$$Var(X) = E(X^2) - [E(X)]^2$$

$$= \int_{-\infty}^{\infty} x^2 f(x)\,dx - \mu^2$$

Let X be a continuous random variable, then for constants **a** *and* **b** *we have that*

$$Var(aX + b) = a^2 Var(X)$$

$$\sigma_{aX+b} = |a|\sigma_X$$

Standardized random variable

Let X be a random variable with mean μ and standard deviation σ. The random variable

$$X^* = \frac{(X - \mu)}{\sigma}$$

is called the **standardized** X.

When standardizing a random variable X, we change the origin to μ and the scale to the units of standard deviation. The value that is obtained for X^* is independent of the units in which X is measured. It is the number of standard deviation units by which X differs from $E(X)$.