PyCharm

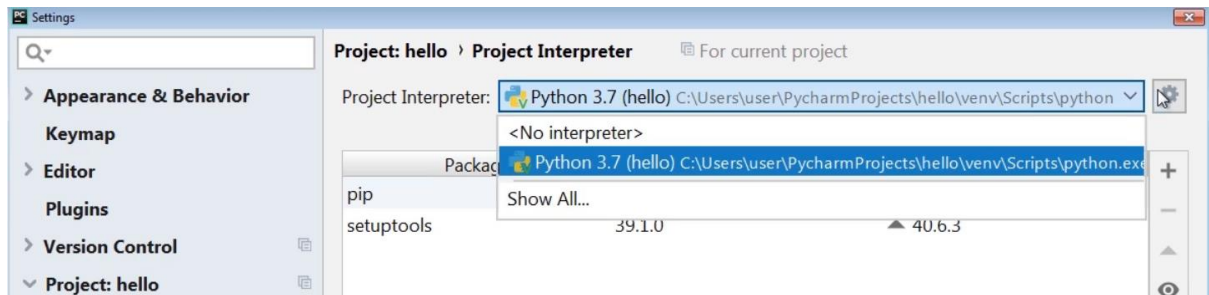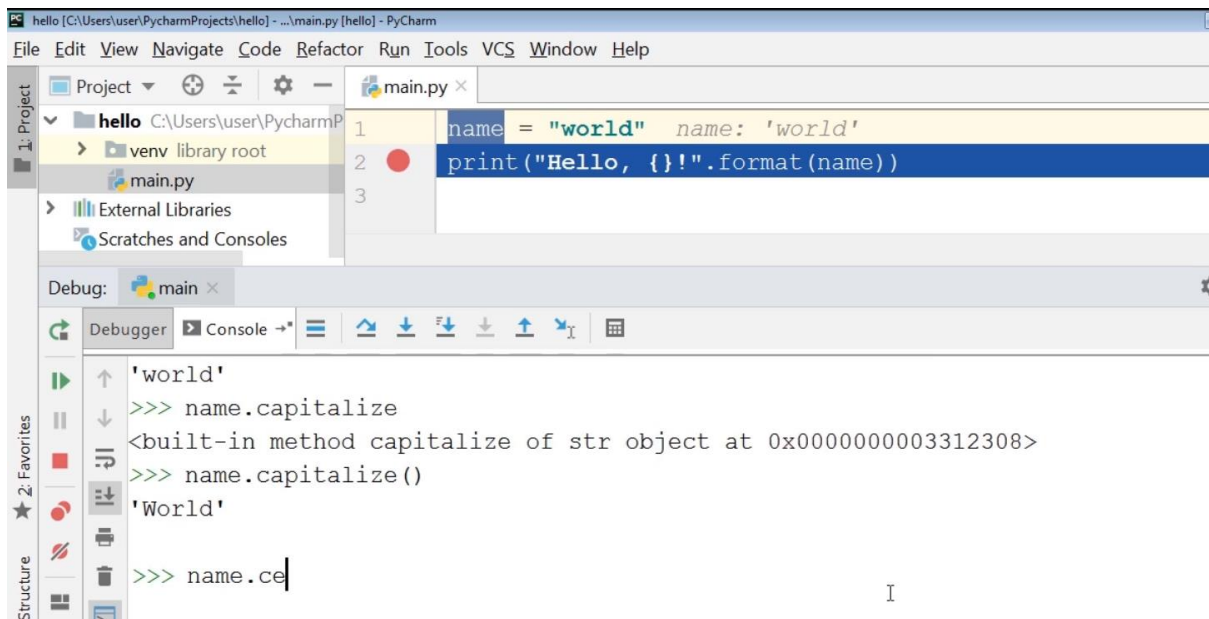## Boost Data Science Productivity with PyCharm

Getting Started

Pressing the escape key will always take to the editor code window.

To execute the python code PyCharm needs an interpreter which is basically the python executable, typically from some virtual environment.
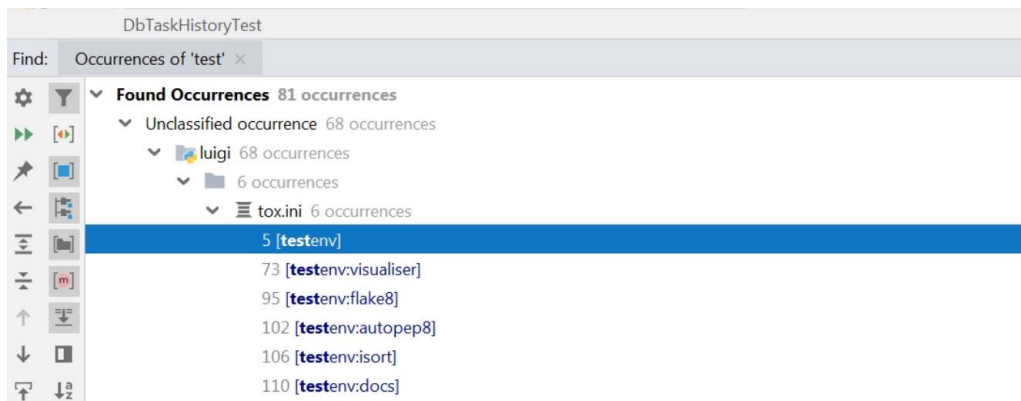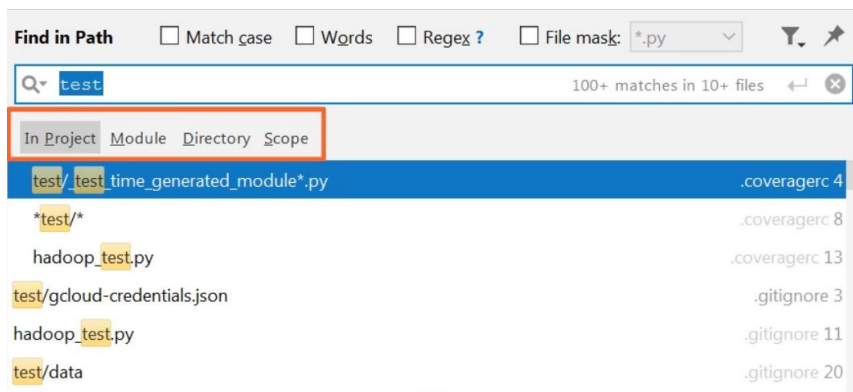

Use alt + shift + e to debug the python code – use IPython console.
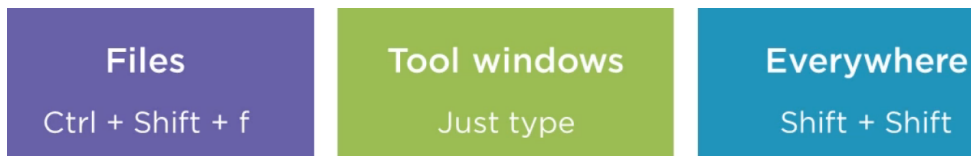

Understanding Code

Use shortcuts like alt + 1, alt + `, alt + 7, f3, shift + f3, ctrl + shift + f, ctrl + enter, shift + shift, ctrl + tab, ctrl + shift + tab
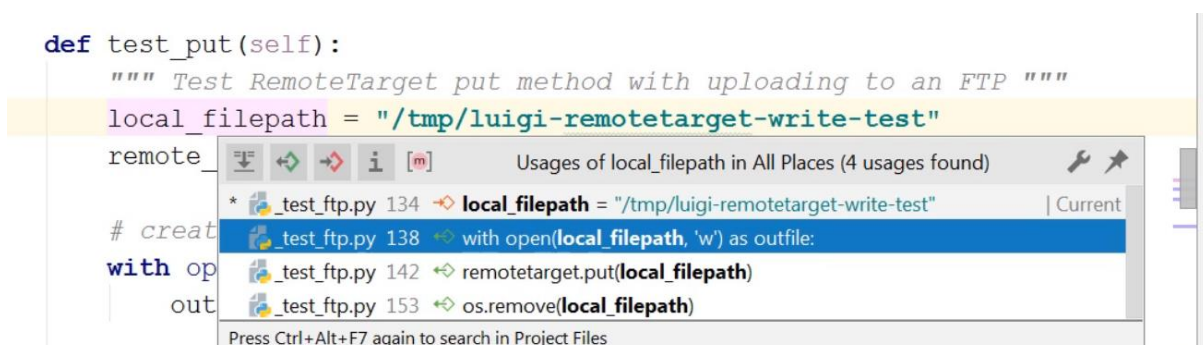
Use settings like auto scroll,

Using Luigi package to build complex pipelines of batch jobs. Use tox.ini to run tests in multiple environments.

Searching

Find usages – selection, ctrl + b, ctrl + shift + f7
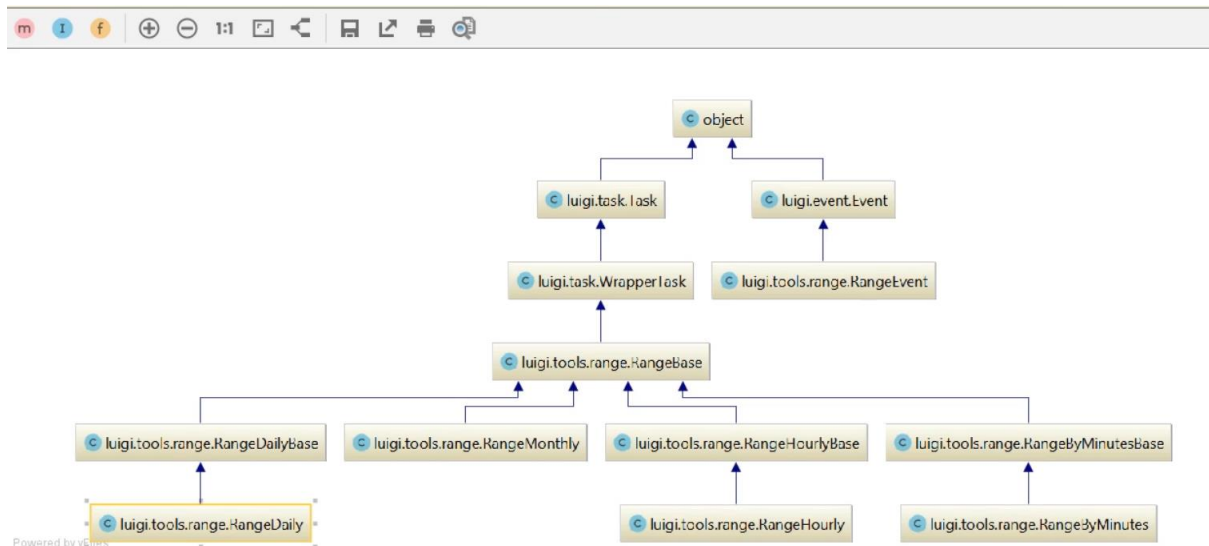

Finding usages vs. searching

Finding Usages	Searching
Code-oriented	Text-oriented
Imperfect	Perfect
Powerful	Powerful

Diagrams – high level overview of code

Gutter icons


```
232
233 ⬤↓ ⬤↑   def exists(self):
234              """
235                  Returns ``True`` if the path for this FileSyst
236
```
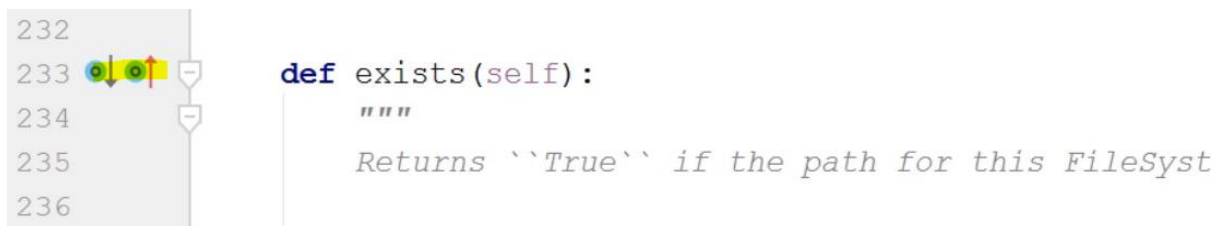
Documentation – ctrl + q for a quick overview, we can also show the documentation externally on browser window

```
656
657 ⬤↓   class S3Target(FileSystemTarget):
658              ""    luigi.contrib.s3
659         Ta   class S3Target(FileSystemTarget)
660              Target S3 file object              ⚙
661         :param kwargs: Keyword arguments are passed t
```
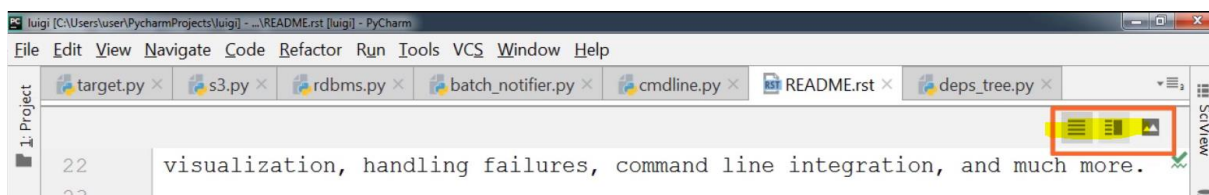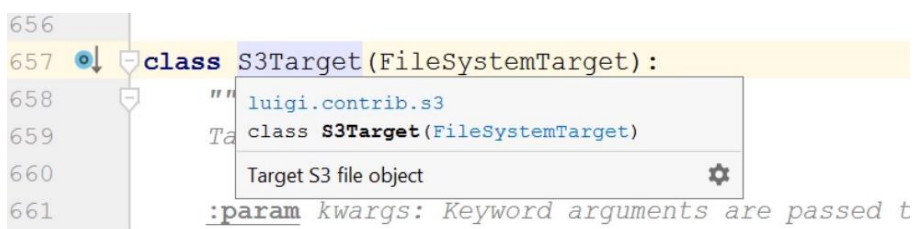

The tox.ini file will have list of dependencies for that project. Ctrl + shift + 10 to run tox file by pycharm.

Running tests – we should check we are using the right configuration for right task otherwise it will give an error like using Nose tests to run the tests.
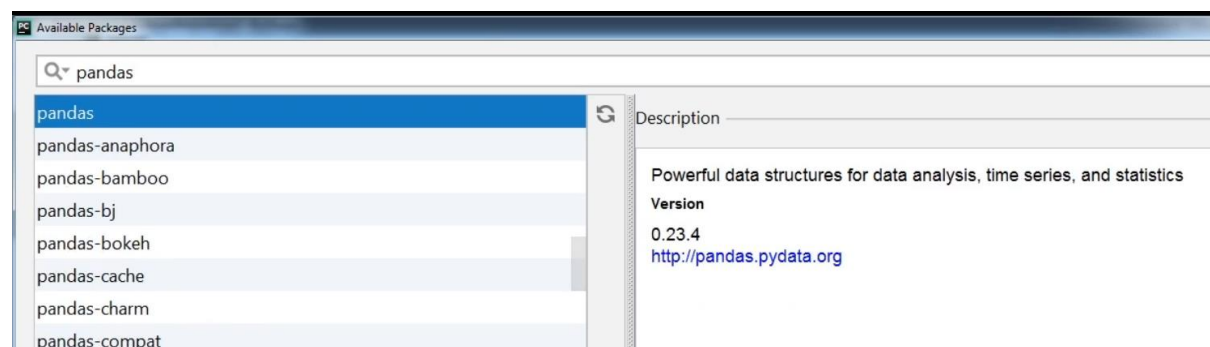
Writing Better Code

Code completion – use ctrl + q to see its documentation on intellisense, ctrl + shift + enter will take to the next line. Use alt + enter for opening the hint light bulb.

Basic Ctrl + Space	**Statement** Ctrl + Shift + Enter	**Intentions** Alt + Enter

Use refactor menu for refactoring the code –
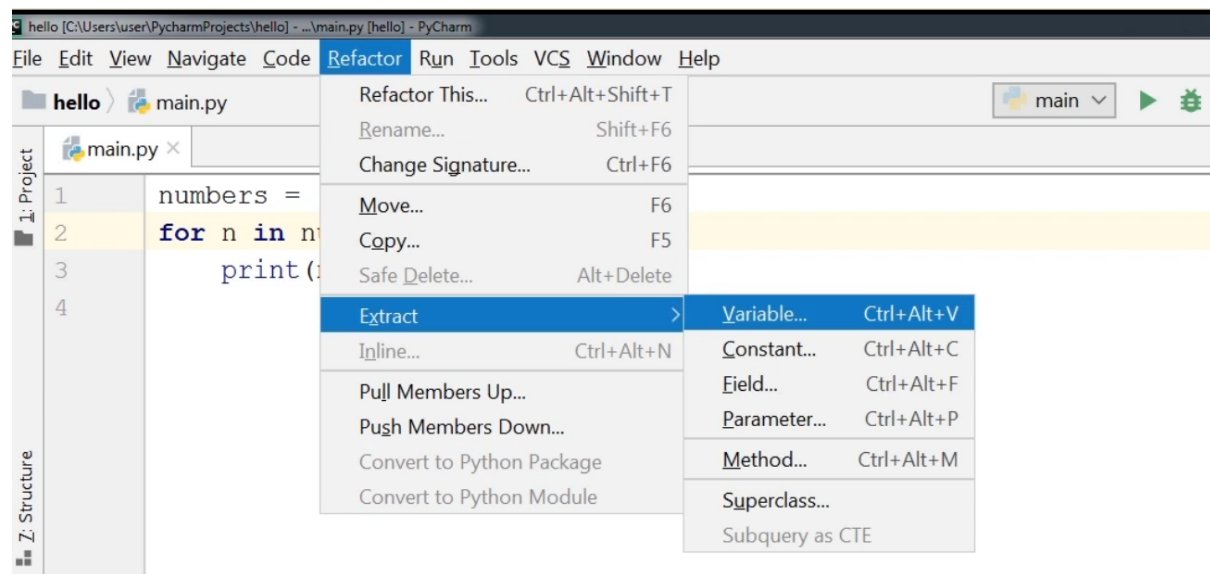
Install libraries like below – ctrl + alt + s

If code completion doesn't work then use the debugger window to see them by debugging it.

Inspections – automated code reviews – on the fly and on demand – ctrl + f1

```
   5
 6     def download_iris(csv_file='output.csv'):
 7         link = 'https://en.wikipedia.org/wiki/Iris_flower_data
 8         response = requests.get(link)
 9         soup = BeautifulSoup(response.text)
10         iris_table = soup.find_all('table')[0]
11         iris_dataframe = pd.read_html(str(iris_table))[0]
12             Typo: In word 'dataframe' more... (Ctrl+F1)   csv(csv_file)
13
```

Use f2 to go the error and then again for next error then ctrl + f1.

Highlighting level of inspection or setting up custom inspection
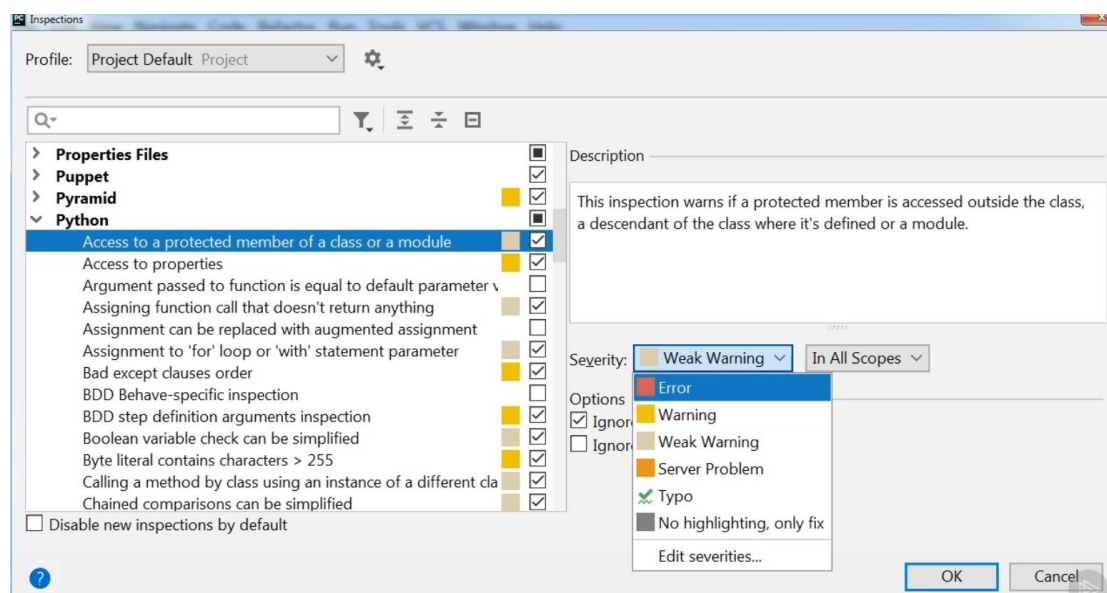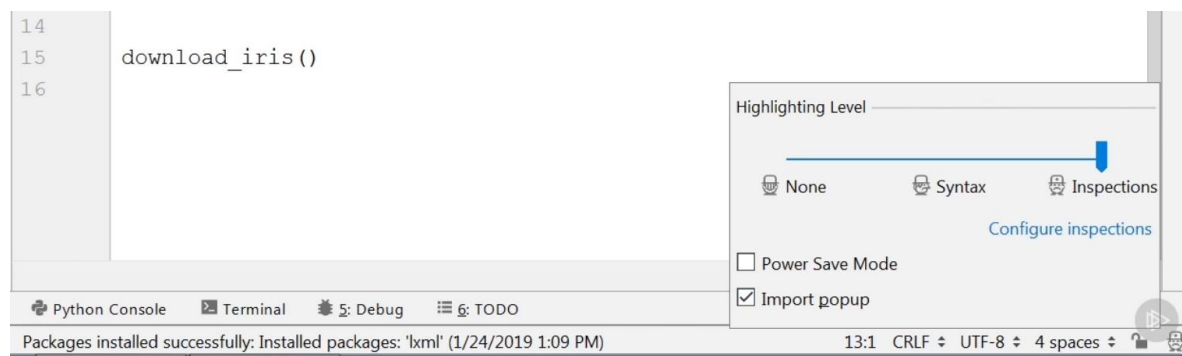
On demand inspection
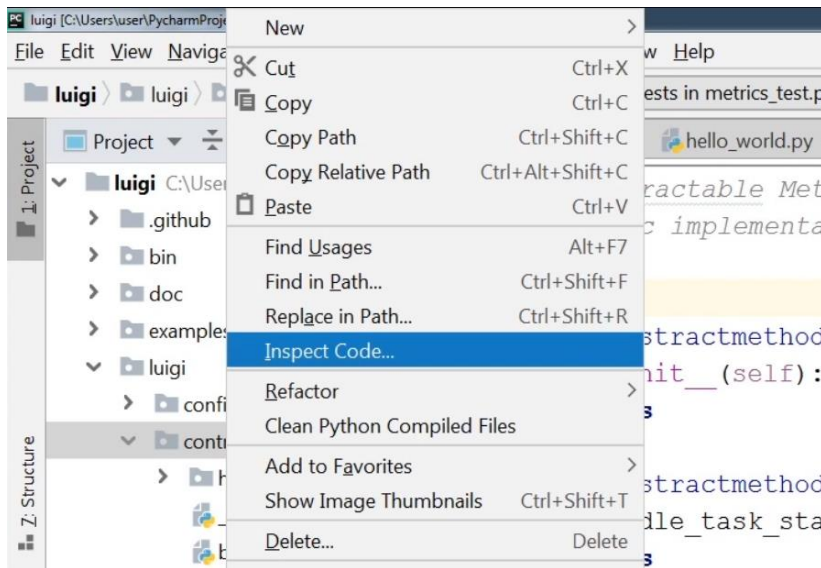
Enable version control system with Git.
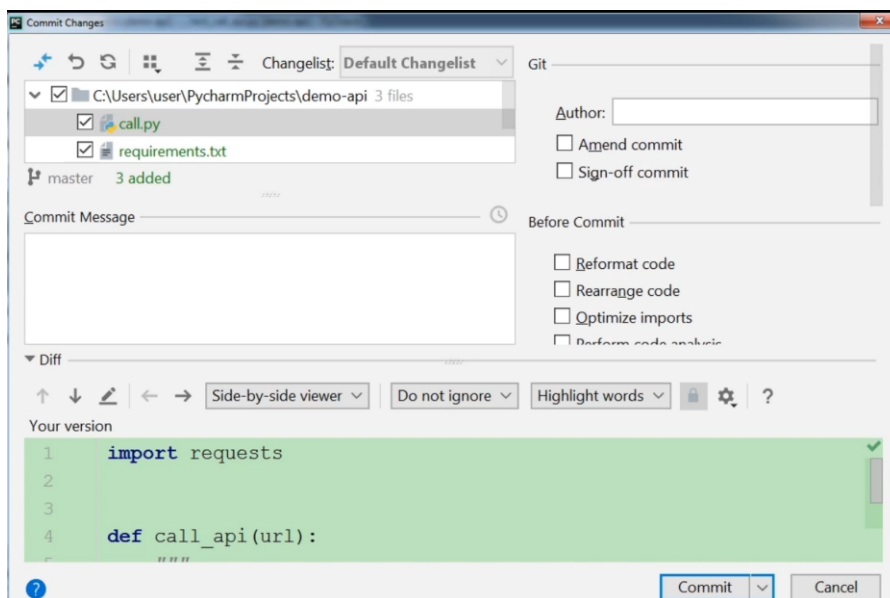
Debugging Code

Breakpoints type s- line, conditional, exception. For exception breakpoint in case of custom exception use ctrl + shift + 8 and add it like below with 'On raise' setting
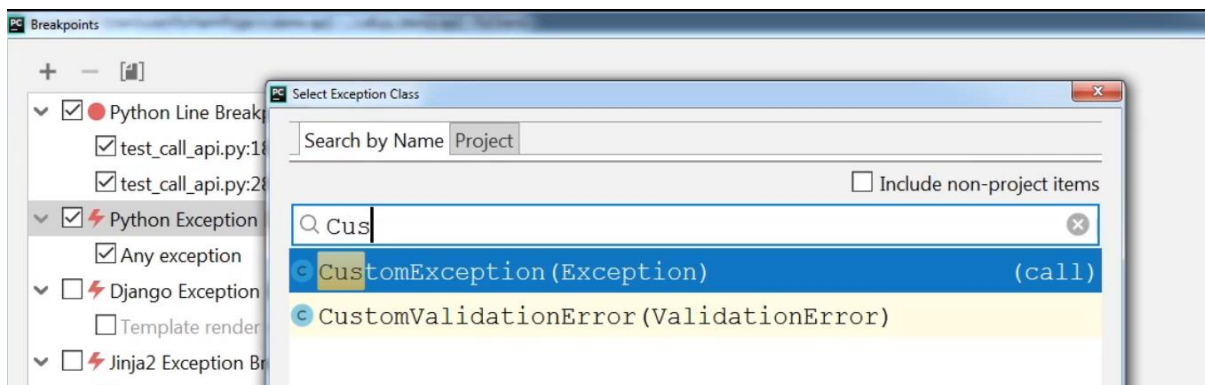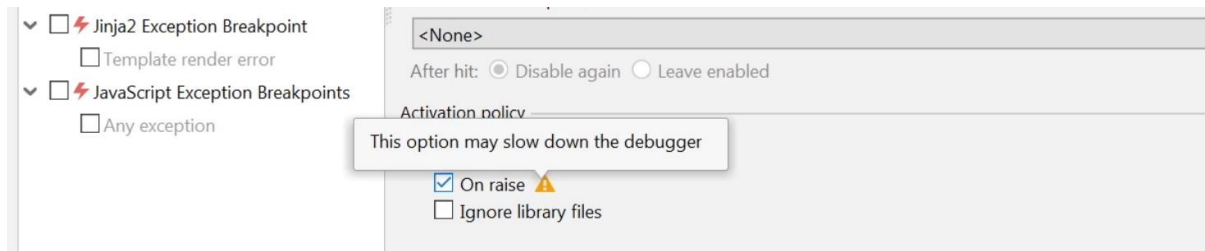
While debugging the code it is a good practice to avoid adding print statements into our code as it pollutes and obfuscate code.

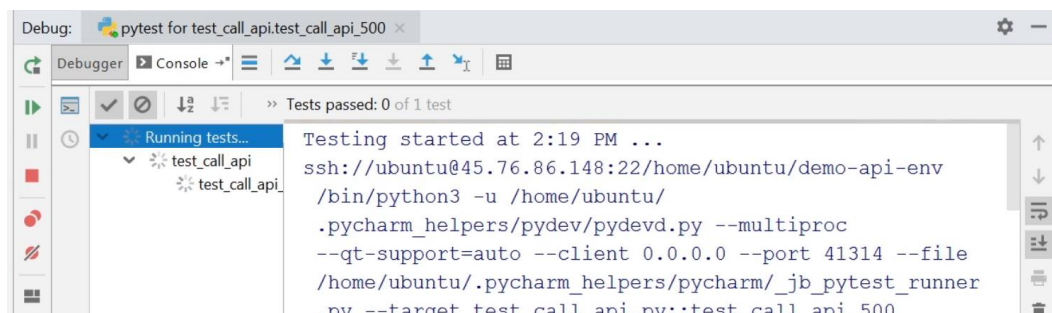Use frames pane in debugger which debugging to go back in time.

Stepping

Attaching to a process – if python code is started outside of the pycharm. Also apply below filter to see only relevant process
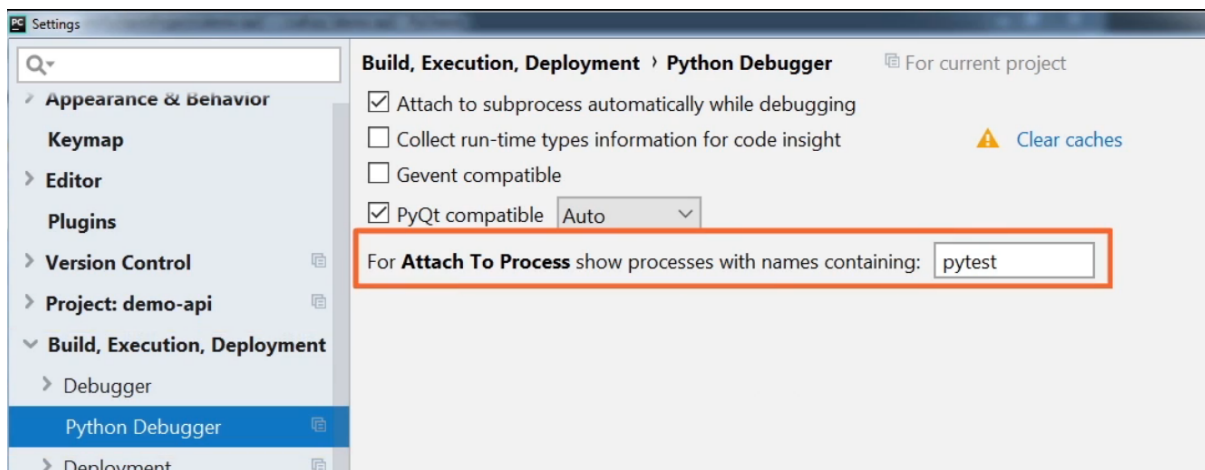
Remote debugging – to debug if our code is running on some other server.

Exploring Data

Using scientific mode which is more of data oriented –
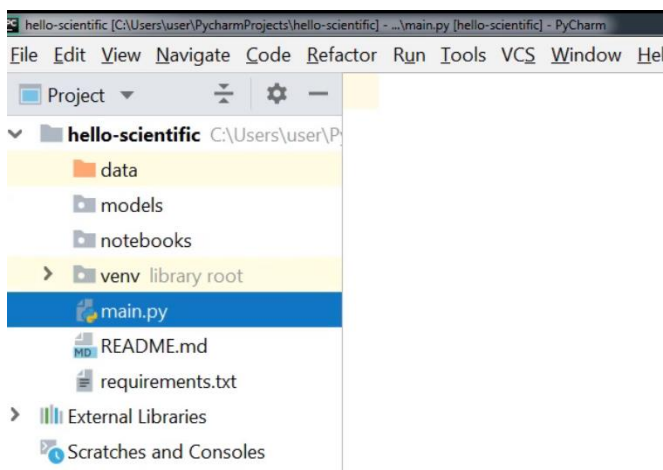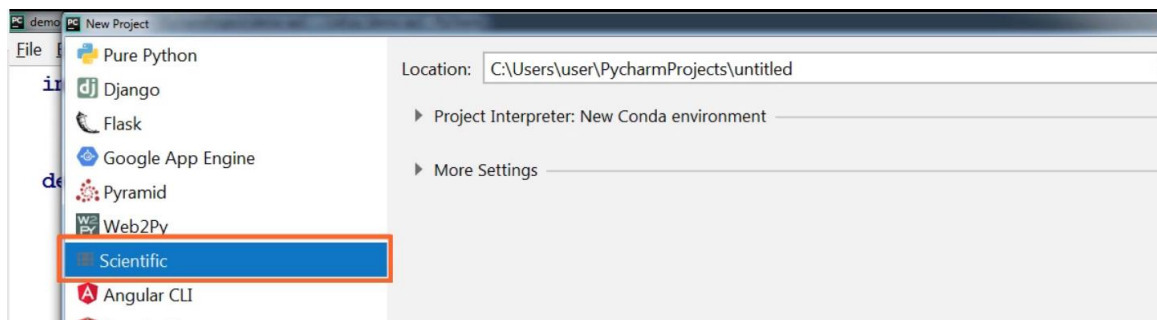
Also we can use code cells using # %%
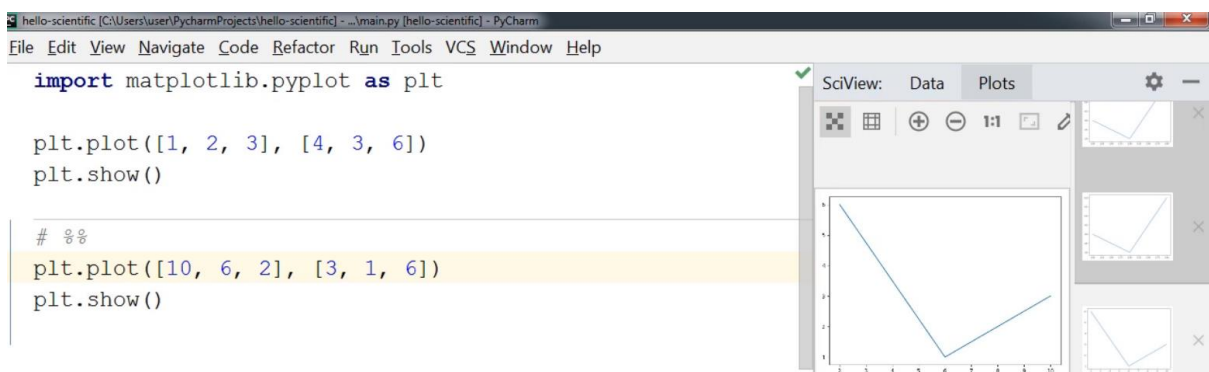
To install packages, create requirements.txt file and client install requirement link:
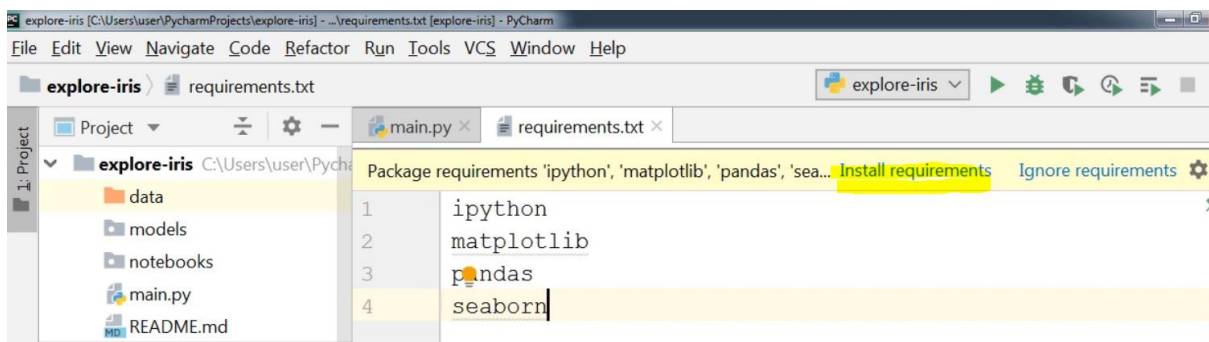
Use alt + shift + e to execute.
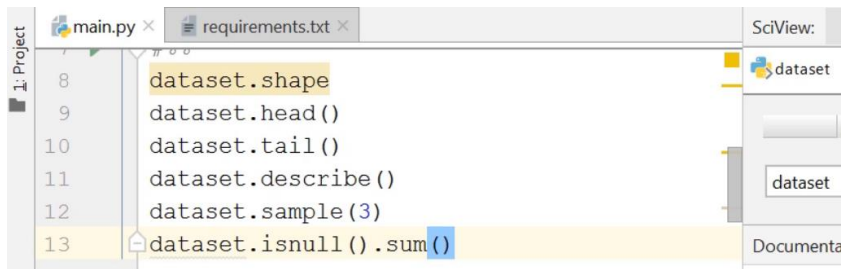
Jupyter Notebooks – we can install jupyter package and ad jupyter notebook in pycharm also it provides various features like code completion, cells debugging and code refactoring.
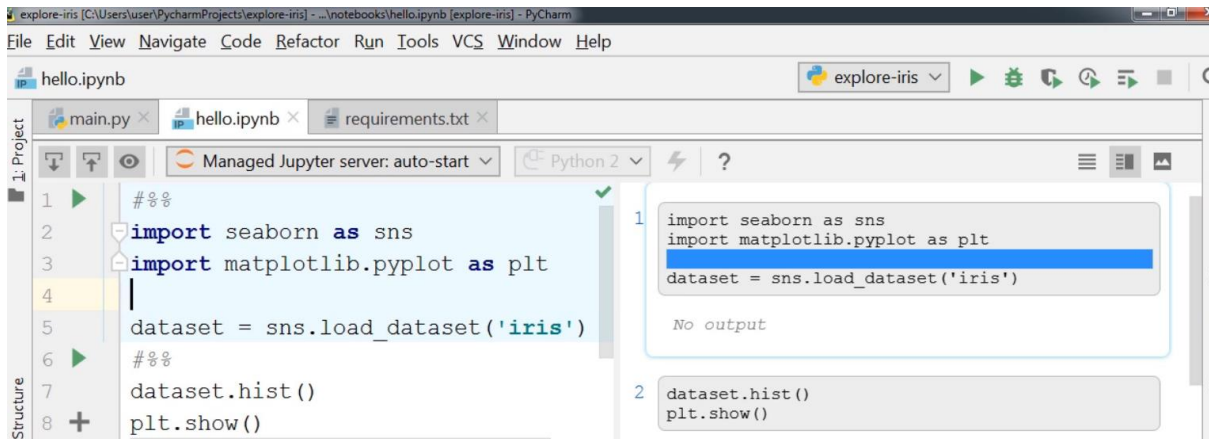
The pandas library is great when working with relatively small datasets, however for large datasets pandas might require more memory than what is available on our machine, for this scenario SQL scripts need to use as they are highly optimized for working with large dataset. By SQL, we will extract a data subset and analyse it with pandas.
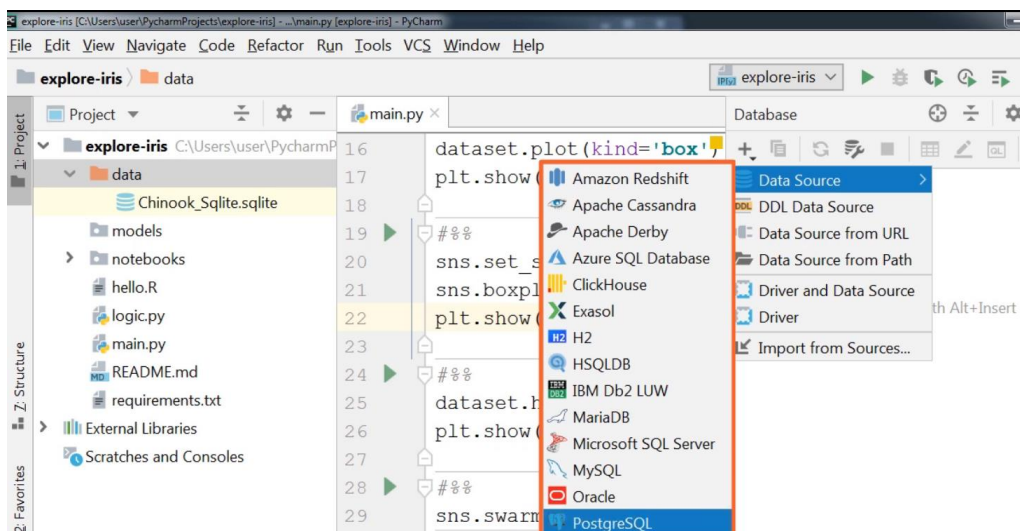
Using alt + enter we can inject a language in a string for intellisense inside a string and also execute it
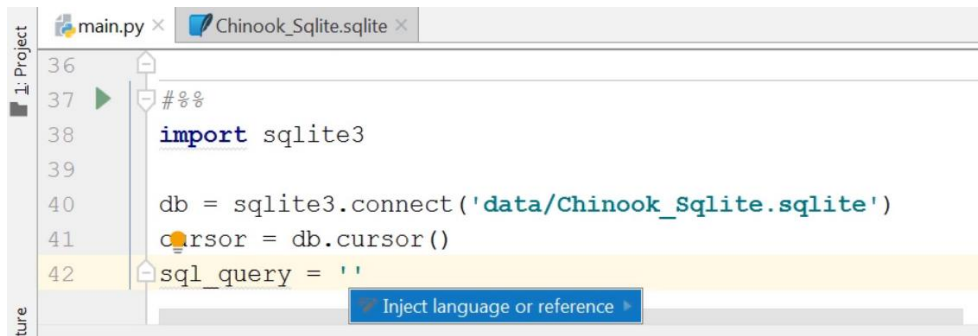
Keep pycharm keymap reference sheet handy and key promoter helper for shortcut helps