주식 분석 스크립트 설치, 사용법

# 설치

본 스크립트는 Python 3.7을 이용하여 PyCharm을 통해 개발하였으므로, PyCharm을 다운로드 받으시는 것이 좋습니다.

* PyCharm은 Visual Studio와 같은 프로그램 개발 도구(IDE)로써 <https://www.jetbrains.com/pycharm/download/> 에서 다운로드 받을 수 있습니다.
* Professional 버전은 이메일 등록이 필요하므로 Community 버전을 받으시면 됩니다.

컴퓨터에 Python 3.7이 설치되지 않은 경우 이 또한 설치하여야 합니다.

* <https://www.python.org/downloads/> 에서 다운로드 받을 수 있습니다.
* 3.7 버전을 설치하셔야 합니다.

PyCharm 설치 후, Github에 올라가 있는 스크립트를 통해 프로젝트를 생성할 수 있습니다.

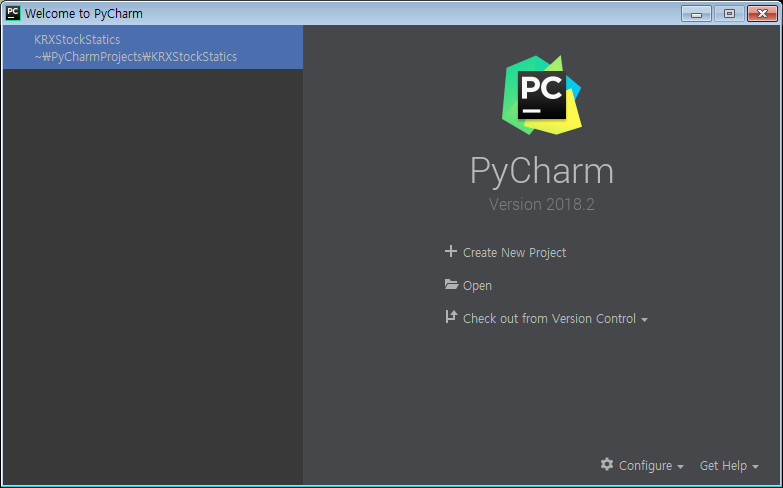


Figure 1 우측 메뉴 중 Check out from Version Control 선택 -> Git 선택

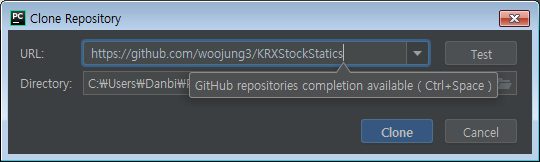


Figure 2스샷과 일치하게 URL 설정 후 Clone 클릭

Figure 1, 2 순으로 진행하고 프로젝트를 열면(Open) Figure 3과 같은 화면이 출력됩니다.

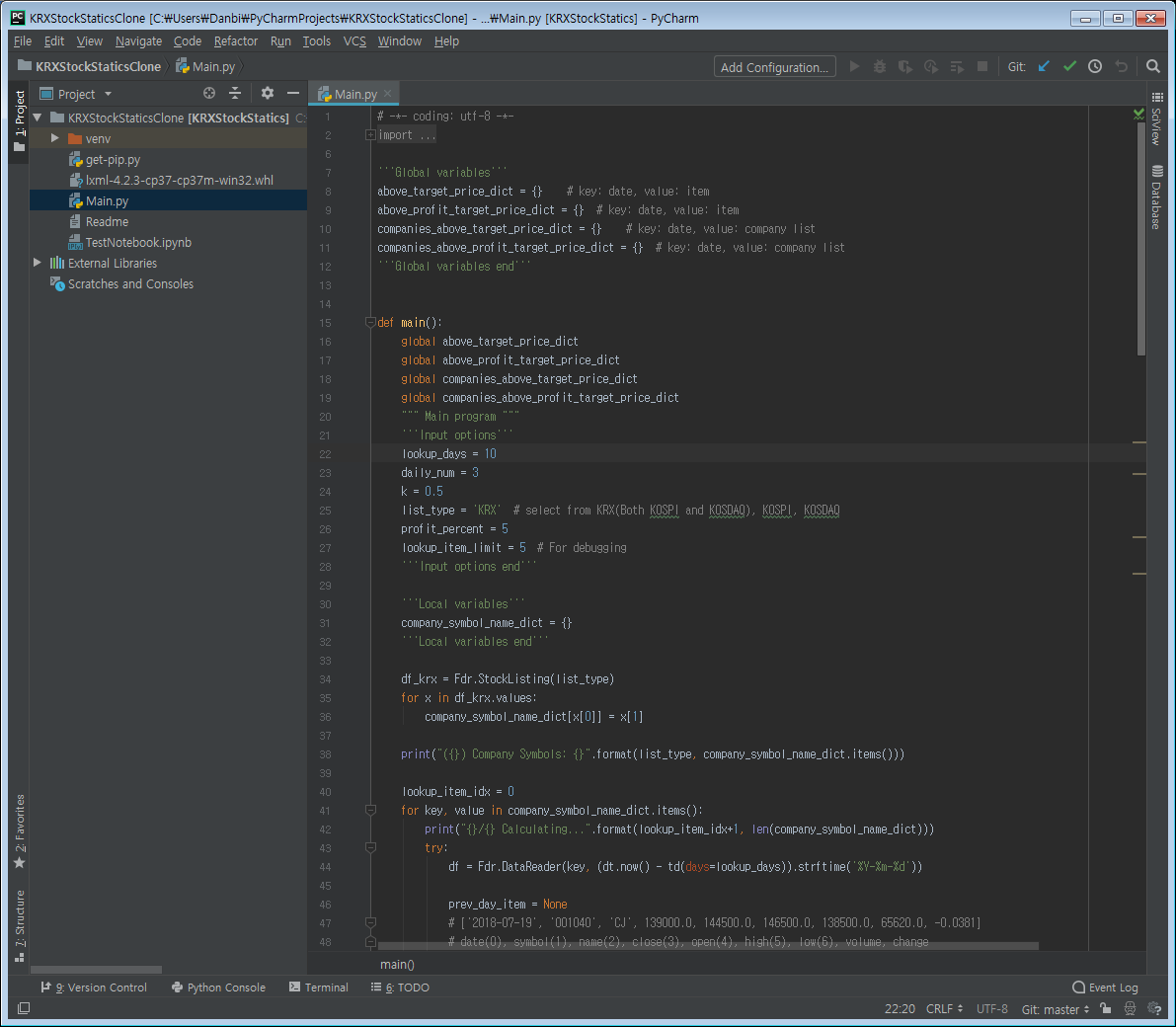


Figure 3 Github로부터 가져온 프로젝트

Figure 3상태에서는 필요 라이브러리가 설정되어 있지 않아 코드를 바로 실행할 수 없습니다.

File -> Settings에서 다음을 설정해주십시오.

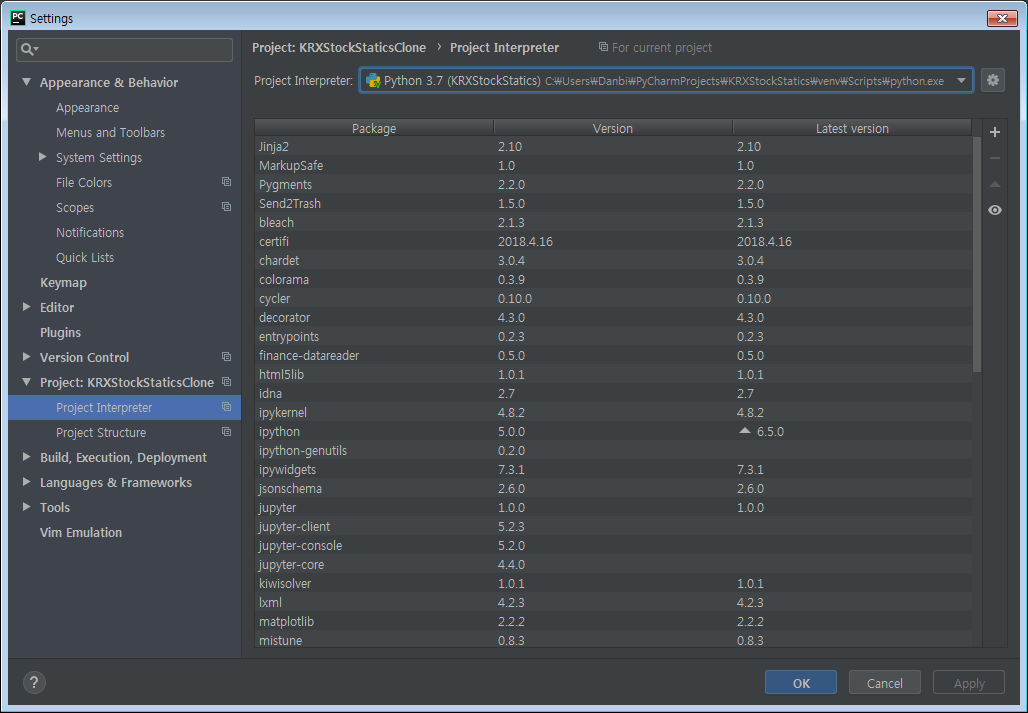


Figure 4 File - Settings - Project - Project Interpreter

Figure 4의 메뉴로 들어가 우상단의 톱니바퀴를 클릭 – Add를 선택하여 Figure 5의 화면을 엽니다.

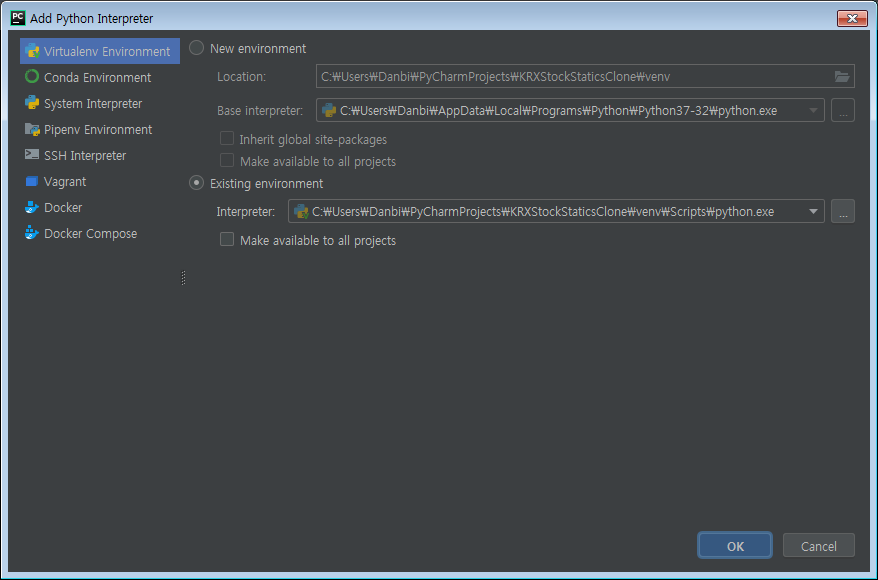


Figure 5 Exisiting environment 선택 - ..\venv\Scripts\python.exe 선택

Figure 5에서 interpreter를 설정하였다면 컴퓨터가 ‘윈도우 7/10, 64비트’ 인 경우 코드를 바로 실행시킬 수 있습니다. Figure 6를 참조해주세요.

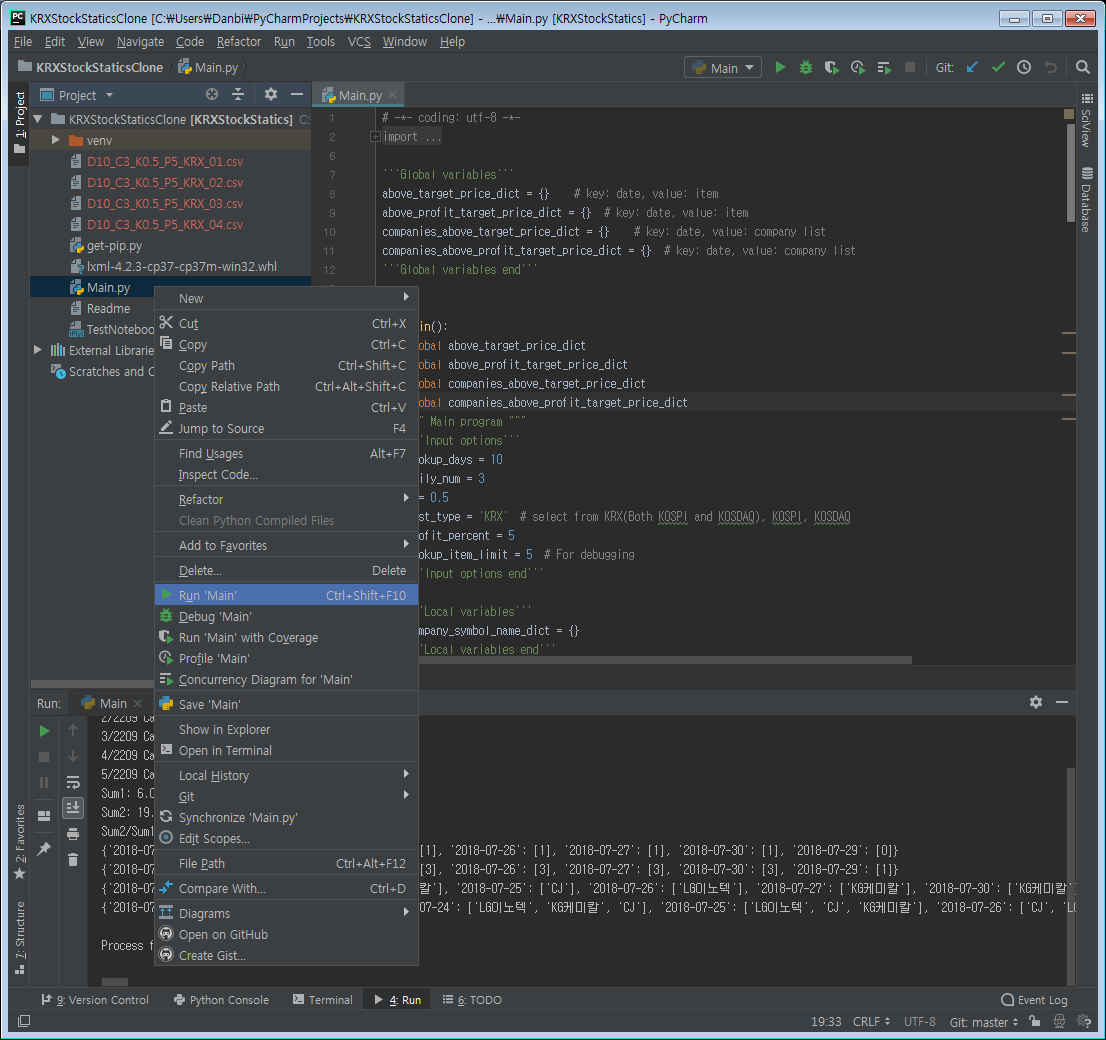


Figure 6 Main.py 우클릭 -> Run Main 클릭 으로 실행

Figure 6과 같이 실행시키면, Figure 6의 하단에 출력된 것 처럼 결과를 확인할 수 있습니다. 결과 중 Dictionary ({}로 표기된 마지막 4개 항목)은 D\*\_C\*\_K\*\_P\*\_\*\*\*\_01.csv 와 비슷한 이름으로 폴더에 저장됩니다.

만일 컴퓨터가 윈도우 7/10, 64비트가 아닐 경우 Run Main 하더라도 에러만 발생할 수 있습니다. 이런 경우에는 제가 미리 설치한 라이브러리가 해당 컴퓨터와 호환이 되지 않는 경우입니다.

<https://github.com/woojung3/KRXStockStatics> 의 Readme 파일을 확인하셔서 필요 라이브러리를 설치해주세요.

# 상세 설치 과정

직접 설치시에는 venv(Virtualenv Environment)가 아닌 System Interpreter를 이용합니다. File -> Setting 에서 Project Interpreter를 선택하고 우상한 톱니바퀴를 눌러 Add Python Interpreter로 들어가 System Interpreter로 컴퓨터에 설치된 python37의 경로를 선택하십시오.

보통 C:\Users\UserName\AppData\Local\Programs\Python\Python37\python.exe 에 있습니다.

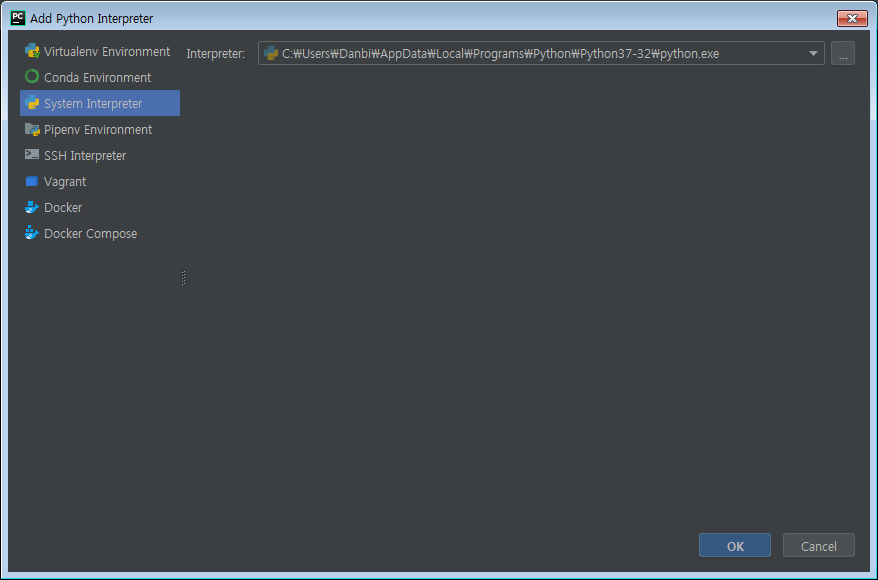


Figure 7 System Interpreter 설정

만일 cmd 창에서 python 명령어나 pip 명령어가 동작하지 않는다면 두 명령어를 시스템 환경변수로 등록시킬 필요가 있습니다.

<https://medium.com/@psychet_learn/python-%EA%B8%B0%EC%B4%88-2%EC%9E%A5-python-%EC%84%A4%EC%B9%98-%EB%B0%8F-%ED%99%98%EA%B2%BD%EC%84%A4%EC%A0%95-windows-ver-b030d96bcbd0>

상기 링크를 참고해서

C:\Users\%UserName\AppData\Local\Programs\Python\Python36\Scripts\;  
C:\Users\%UserName\AppData\Local\Programs\Python\Python36;

두 줄을 환경변수에 추가해주세요.

환경변수 추가가 끝난 뒤에는 pip를 업그레이드 해야 합니다. 프로젝트에 첨부되어 있는 get-pip.py를 적당한 위치로 복사한 뒤,

python get-pip.py

하여 업그레이드를 수행합니다. 2018-08-01 기준 18.0 버전으로 업그레이드 될 것입니다.

이후 필수 라이브러리 두 개 (matplotlib, FinanceDataReader) 를 설치합니다.

1. matplotlib: pip install matplotlib
2. FinanaceDataReader: pip install –U finance-datareader
   1. Microsoft Visual Studio가 설치되어 있지 않은 경우:
      1. Microsoft Visual Studio가 설치되지 않은 경우 Microsoft Visual C++ 14.0을 별도로 설치해야 함.
      2. 다운로드 링크: <https://visualstudio.microsoft.com/downloads/#build-tools-for-visual-studio-2017> (“**Visual Studio 2017용 Build Tools” 를 검색하여 다운로드)**
   2. Lxml이 설치되어 있지 않은 경우:
      1. 다운로드 링크: <https://www.lfd.uci.edu/~gohlke/pythonlibs/#lxml>
      2. 본인의 os에 맞는 lxml 파일을 다운로드 한 뒤 (lxml-4.2.4-cp37xxx 의 이름)
      3. pip install 다운로드\_파일명
      4. 하여 설치.

필수 라이브러리 설치 후 pip install –U finance-datareader 하면 FinanceDataReader가 설치됩니다.

# 실행

입력 변수는 다음과 같습니다.

lookup\_days = 10  
daily\_num = 3  
k = 0.5  
list\_type = 'KRX' # select from KRX(Both KOSPI and KOSDAQ), KOSPI, KOSDAQ  
profit\_percent = 5  
lookup\_item\_limit = 5 # For debugging

lookup\_days는 오늘로부터 며칠 전 까지의 데이터를 볼 것인가를 결정합니다. 7월 30일 기준 10일 전인 7월 20일부터의 데이터를 얻고 싶다면 10을 적습니다.

daily\_num 은 매일 몇 개의 종목을 확인할 것인지를 설정하는 값입니다.

k 는 0과 1 사이로 지정하시면 되겠습니다.

list\_type은 KRX, KOSPI, KOSDAQ 중 하나를 ‘’ 안에 적어넣으시면 됩니다. KRX 선택시 코스피와 코스닥 모두를 검색합니다.

profit\_percent 는 수익률 값입니다. 0~100 사이로 지정하시면 되겠습니다.

lookup\_item\_limit 는 제가 디버깅을 위해 임의로 넣은 입력 변수입니다. 스크린샷과 같이 5를 입력하면 2200여개의 종목 중 5개만 확인합니다.

상기 수치를 넣고 Run Main 하면 다음 결과가 나옵니다.

1/2209 Calculating...

2/2209 Calculating...

3/2209 Calculating...

4/2209 Calculating...

5/2209 Calculating...

Sum1: 6.00000000001

Sum2: 1e-11

Sum2/Sum1: 1.6666666666638888e-12

{'2018-07-23': [0], '2018-07-24': [2], '2018-07-25': [1], '2018-07-26': [1], '2018-07-27': [1], '2018-07-30': [1], '2018-07-29': [0]}

{'2018-07-23': [0], '2018-07-24': [0], '2018-07-25': [0], '2018-07-26': [0], '2018-07-27': [0], '2018-07-30': [0], '2018-07-29': [0]}

{'2018-07-23': [], '2018-07-24': ['LG이노텍', 'KG케미칼'], '2018-07-25': ['CJ'], '2018-07-26': ['LG이노텍'], '2018-07-27': ['KG케미칼'], '2018-07-30': ['KG케미칼'], '2018-07-29': []}

{'2018-07-23': [], '2018-07-24': [], '2018-07-25': [], '2018-07-26': [], '2018-07-27': [], '2018-07-30': [], '2018-07-29': []}

Sum1과 Sum2는 Divided by zero 예외를 막기 위해 초기값으로 0.00000000001이 잡혀있습니다. 따라서 1e-11이 출력되면 0이라 간주하시면 되고, Sum2/Sum1이 매우 작으면 Sum2가 0, 매우 크면 Sum1이 0이라고 간주하시면 되겠습니다.

결과 중 일부는 파일로 저장되며, 내용은 최하단 Figure와 같습니다.

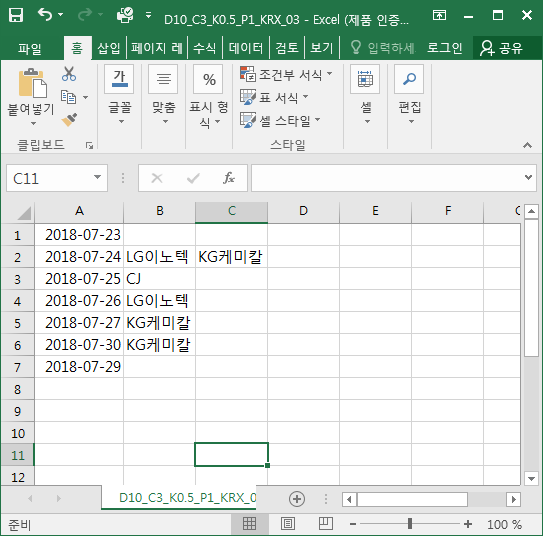


Figure 8 저장된 결과