Cao Yuchen, Zitong Cheng, Aaron Li

Yc3580@columbia.edu

Software Design Documentation

Group 2

Contents

[Introduction 1](#_Toc27043282)

[Software Designment 1](#_Toc27043283)

[Class: Portfolio 2](#_Toc27043284)

[Class: Historical 5](#_Toc27043285)

[Class: GBM 6](#_Toc27043286)

[Class: MonteCarlo 7](#_Toc27043287)

[Class: Backtesting 8](#_Toc27043288)

[Main function 9](#_Toc27043289)

# **Introduction**

This documentation contains the details in software designment, including portfolio establishment, Historical VaR and ES measurement, Parametric VaR and ES measurement, and Monte Carlo VaR and ES measurement.

The software is running consecutively by the following orders: first, it will automatically generate a portfolio given by the input that a user arbitrarily types in; second, it will calculate each different type of risk measurements; finally, the graph will be plotted respectively.

# **Software Designment**

Before the introduction of the class implemented into the model, it is necessary to illustrate the induction of the data for the portfolio establishment that entails information in equity and option. Those data are stored in ‘Equity Data.xlsx’ and ‘Option Data.xlsx’.

Equity Information contains Mark-to-Market Price of Ford, Xerox and S&P 500 Index, which is displayed in Table 1.

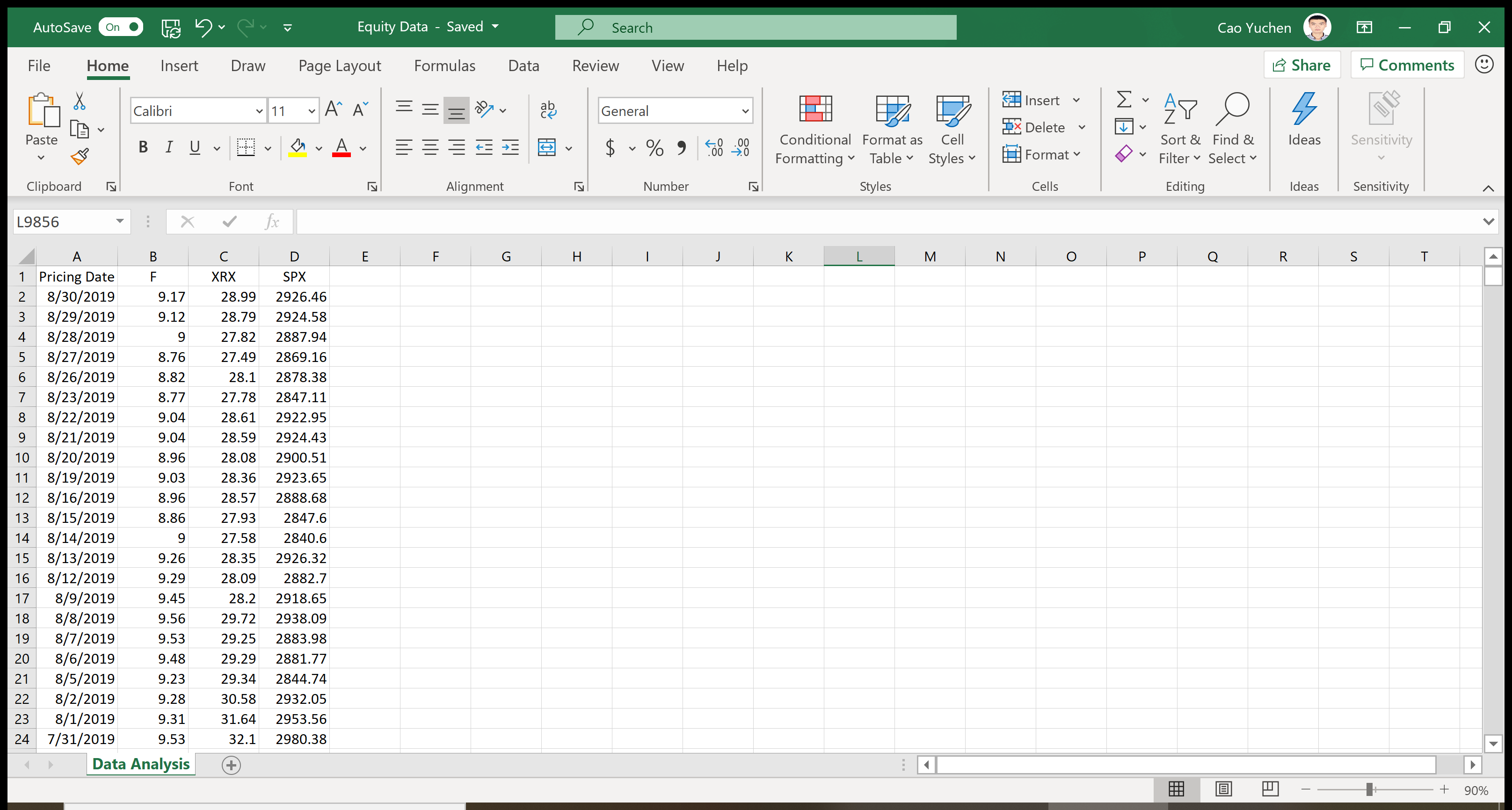


Table 1: data from Equity Data.xlsx

On the other hand, At-The-Money Option Information contains Strike Price and 12-month At-The-Money Implied Volatility of the option, which is displayed in Table 2. Please be aware that Implied Volatility yields the same result for either At-the-Money call or At-the-Money put.

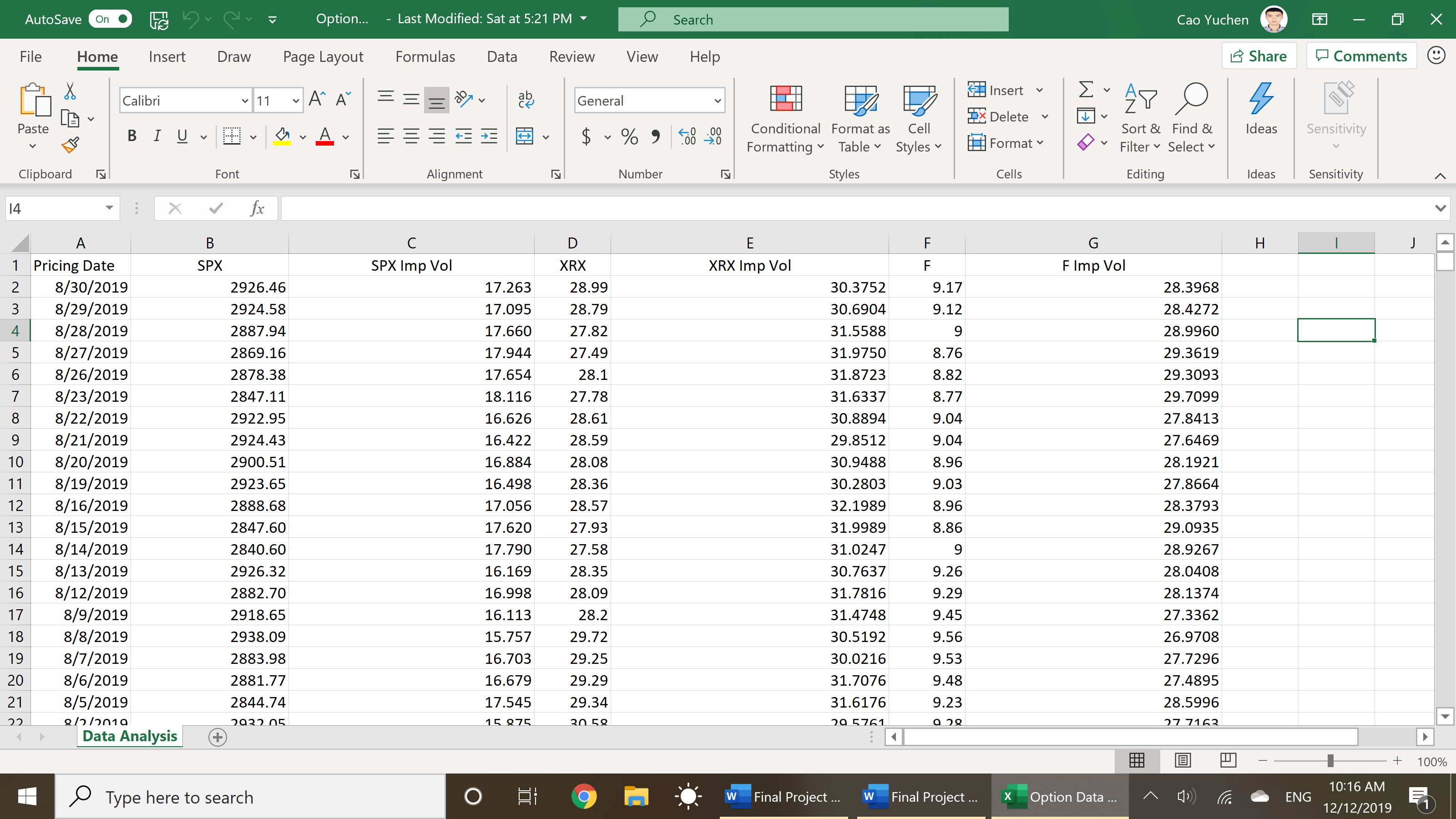


Table 2: data from Option Data.xlsx

A user may want to check the availability and accessibility of the those two files before the operation of the model, especially when verifying the path of the files.

This software contains four major classes: Portfolio, Historical, GBM, MonteCarlo and Backtesting. The following will describe each of the designment in details.

Initialization process for the running of the model is a requirement, where a user must type in the path of the Excel file correctly. The function in assistance is called initialization.

* **initialization** (Equity\_excel\_path,Option\_excel\_path):

‘Equity\_excel\_path’: path of the Equity data excel file

‘Option\_excel\_path’: path of the Equity data excel file

In addition, number inputs for the VaR and ES calculation should also be given before the model running, using numinput function:

* **numinput** (w1,w2,w3,portfolio\_position):

‘w1’: weight of position on Ford stock.

‘w2’: weight of position on Xerox stock

‘w3’: weight of position on S&P 500 options

‘portfolio\_position’: position of the whole portfolio weighted. ‘-1’ for short position and ‘1’ for long position.

A user may choose different weights for the instrument in the portfolio, and long/short position for the whole portfolio. Please be aware that the weight input should be a positive number between 0 and 1 in a closing set, and that ‘portfolio\_position’ is either 1 for a long position or -1 for a short position. Any other number inputs will raise a Type Error.

## **Class: Portfolio**

The “Portfolio” class serves as the initialization of the model by opening the database and reading data, with inputs given by users retrospectively. It will first open the two excels that contain Equity price information and At-The-Money Option strike price with its coordinating one-year implied volatility. The default initial position of the portfolio is $10,000. We assume that one investor exhausts the initial position and that only the portfolio as a whole can be long or short.

First, it will open data with its storage in different parameters:

* Equity: Excel file for “Equity Data.xlsx”
* Option: Excel file for “Option Data.xlsx”
* SPXOption: S&P 500 Index Option Strike Price
* SPX\_imp: S&P 500 Index Option 12-month Implied Volatility

The input for this class includes:

* ‘obs’: years of samples that a user arbitrarily chooses to observe. The total amount of observations will be ‘obs’ times 252 which represents a fiscal year for transactions.
* ‘w1’: weight of position on Ford stock.
* ‘w2’: weight of position on Xerox stock
* ‘w3’: weight of position on S&P 500 options
* ‘portfolio\_position’: position of the whole portfolio weighted. ‘-1’ for short position and ‘1’ for long position.

After running this class, another vital variables will be automatically generated:

* Fshares: stocks shares of Ford
* XRXshares: stock shares of Xerox
* Optionshares: option shares of S&P 500 options
* F0: initial stock price of Ford
* XRX0: initial stock price of Xerox
* SPX0: initial option price of S&P 500

Two Dataframe will be established, lookup and Return:

* lookup: information of the price of the instrument in the portfolio, with its P&L
* Return: information of the daily return of the instrument in the portfolio

Under this class, there are some functions that assists the calculation to the portfolio. Those functions are:

* **Option** (S, r=0.01, T=1.00, Type=’put’):

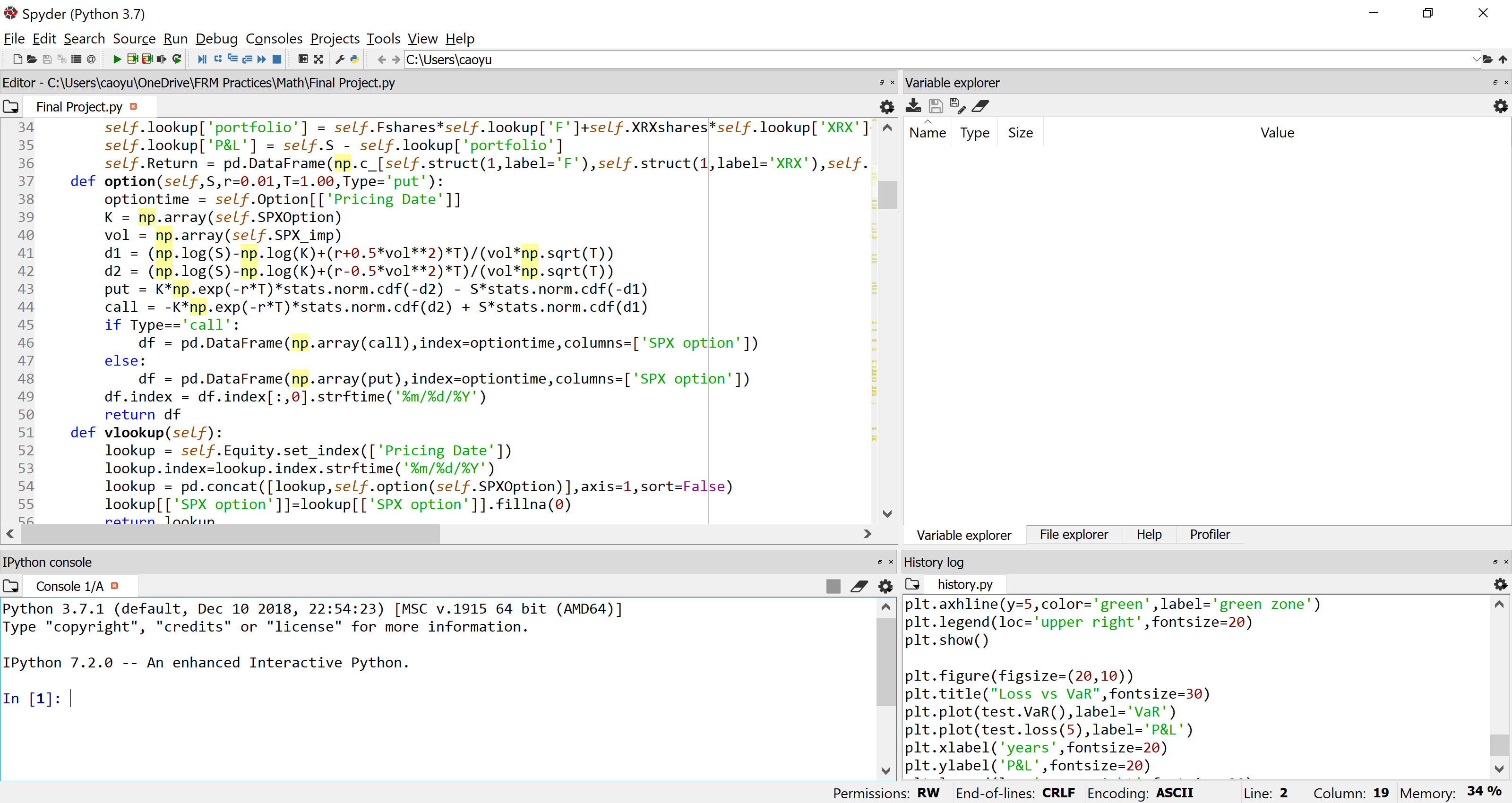
S: Spot price of S&P 500 Index

r: risk-free interest rate

T: Time-to-Maturity

Type: Type of the option

This is the function to calculate the price of At-The-Money option. We assume that the risk-free interest rate is 1%, and the Time-to-Maturity is always one year. This calculation sets default type of option as a put option. A user can change the Type to call option by setting Type as ‘call’.



* **Vlookup** ():

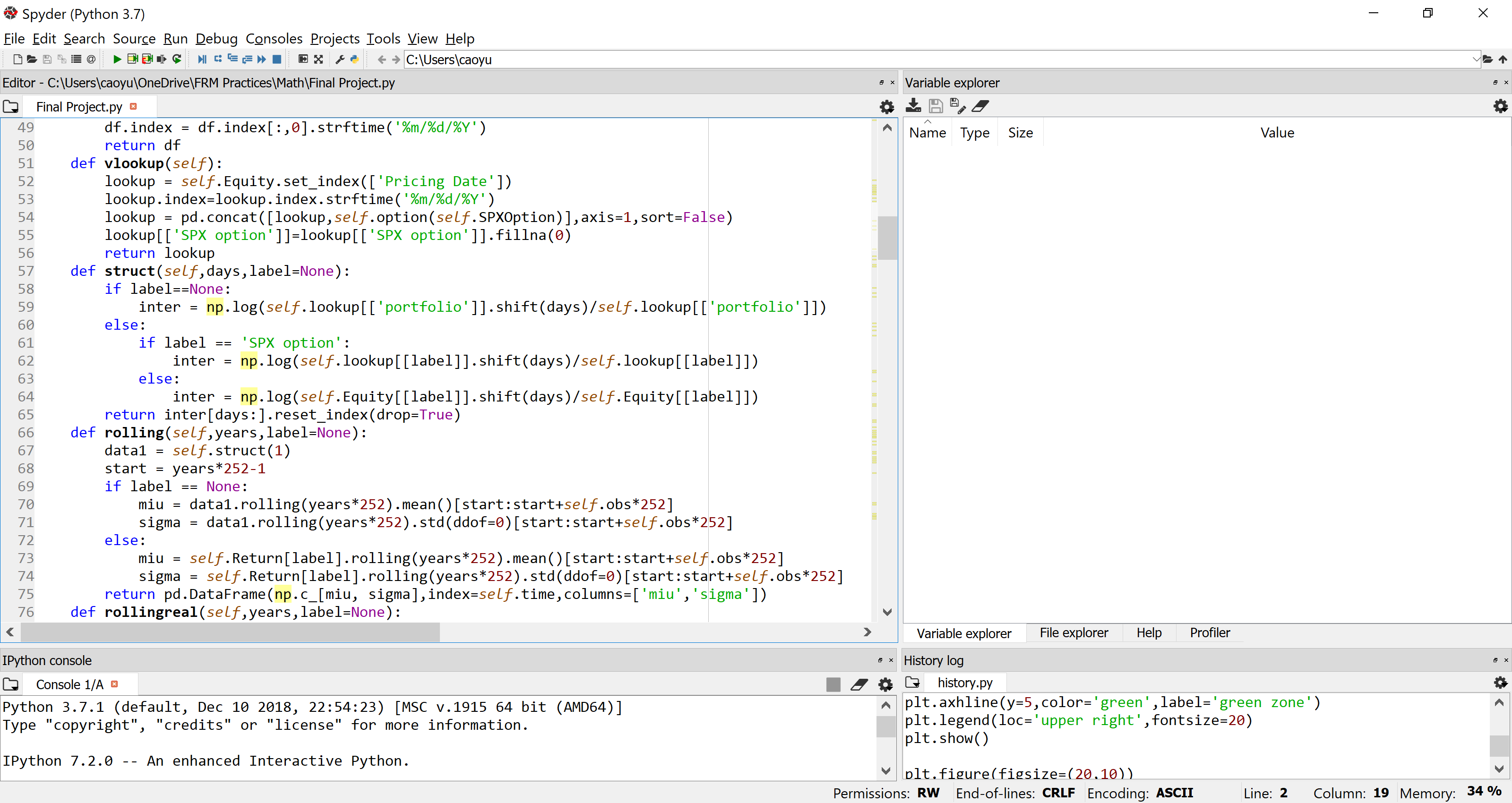
This is the function to set up a dataframe called “lookup”, where it contains information about stock price, S&P 500 Index and its option price, as well as the price of the portfolio and its coordinating Profit and Loss (P&L).

* **Struct** (days, label=None):

days: day of delay for the return calculation

label: a specific instrument in the portfolio from which the result is calculated

This function will calculate return based on the days delay that a user determines. Label set default value as None for the portfolio. A user can change the label within four key arguments: ‘F’, ‘XRX’, ‘SPX’ and ‘SPX option’.

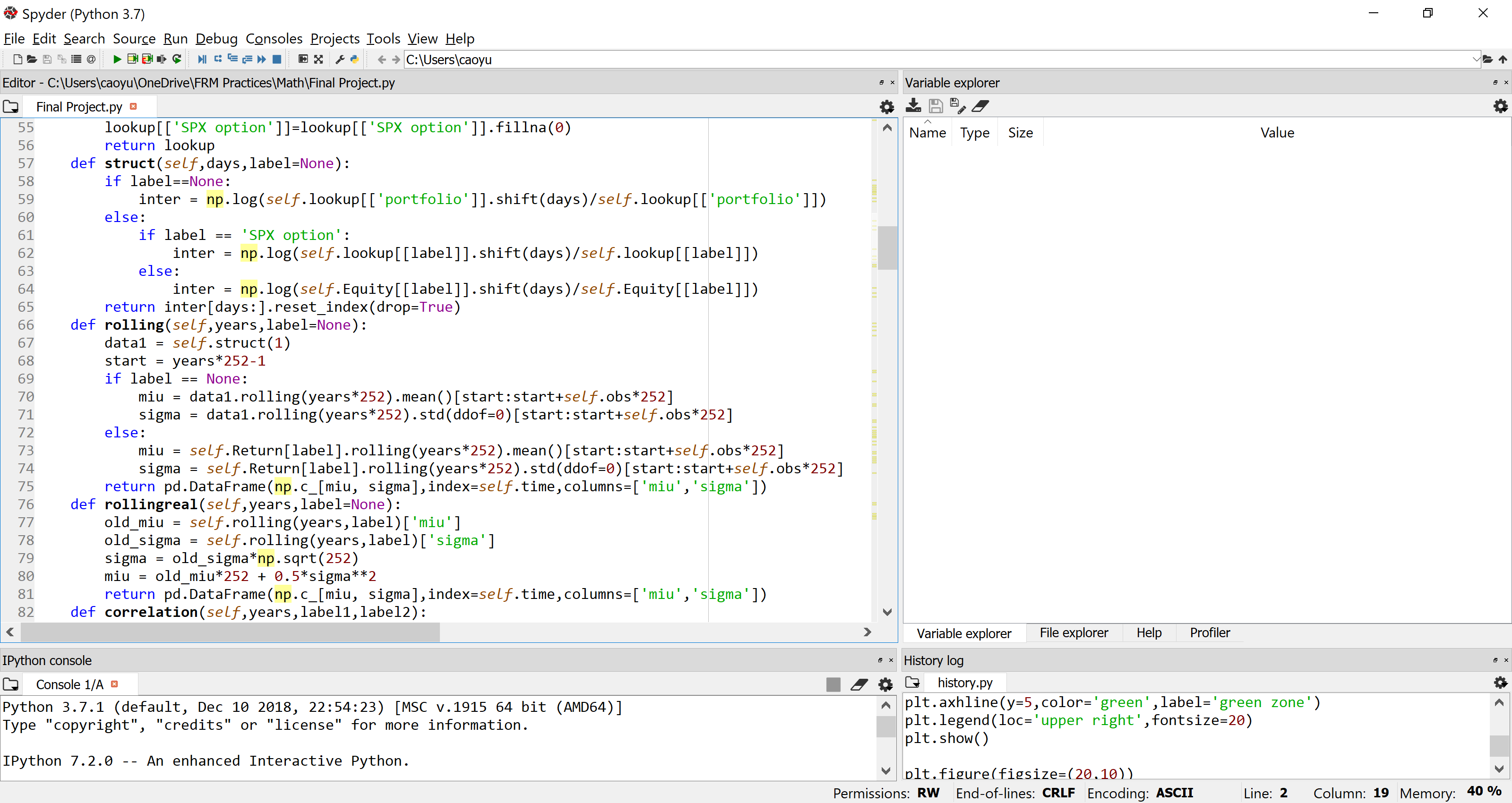


* **Rolling** (years, label=None):

years: years of the rolling window

label: a specific instrument in the portfolio from which the result is calculated

This function will use a rolling window predetermined by the user to calculate sample mean and sample standard deviation of the return. Label set default value as None for the portfolio. A user can change the label within four key arguments: ‘F’, ‘XRX’, ‘SPX’ and ‘SPX option’.

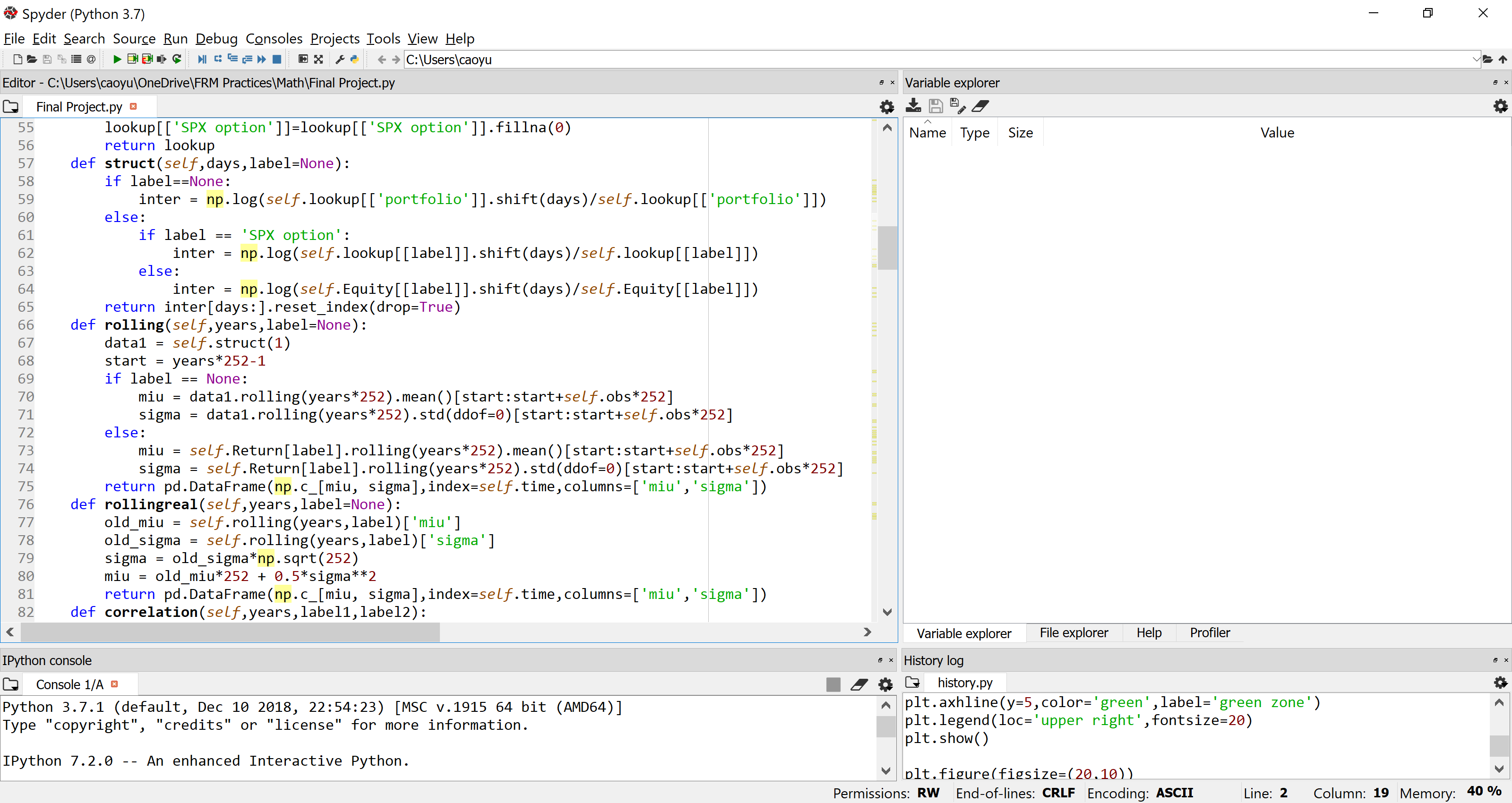


* **Rollingreal** (years, label=None):

years: years of the rolling window

label: a specific instrument in the portfolio from which the result is calculated

This function is the calibration of the “rolling” function from which it will adjust into a geometric Brownian motion, where drift and volatility are determined. Label set default value as None for the portfolio. A user can change the label within four key arguments: ‘F’, ‘XRX’, ‘SPX’ and ‘SPX option’.



Please be aware that Option contracts does not, but its underlying asset does, follow a GBM, and thus does not follow a log-normal distribution either. However, since we assume that the portfolio follows a GBM, it is applicable to calibrate.

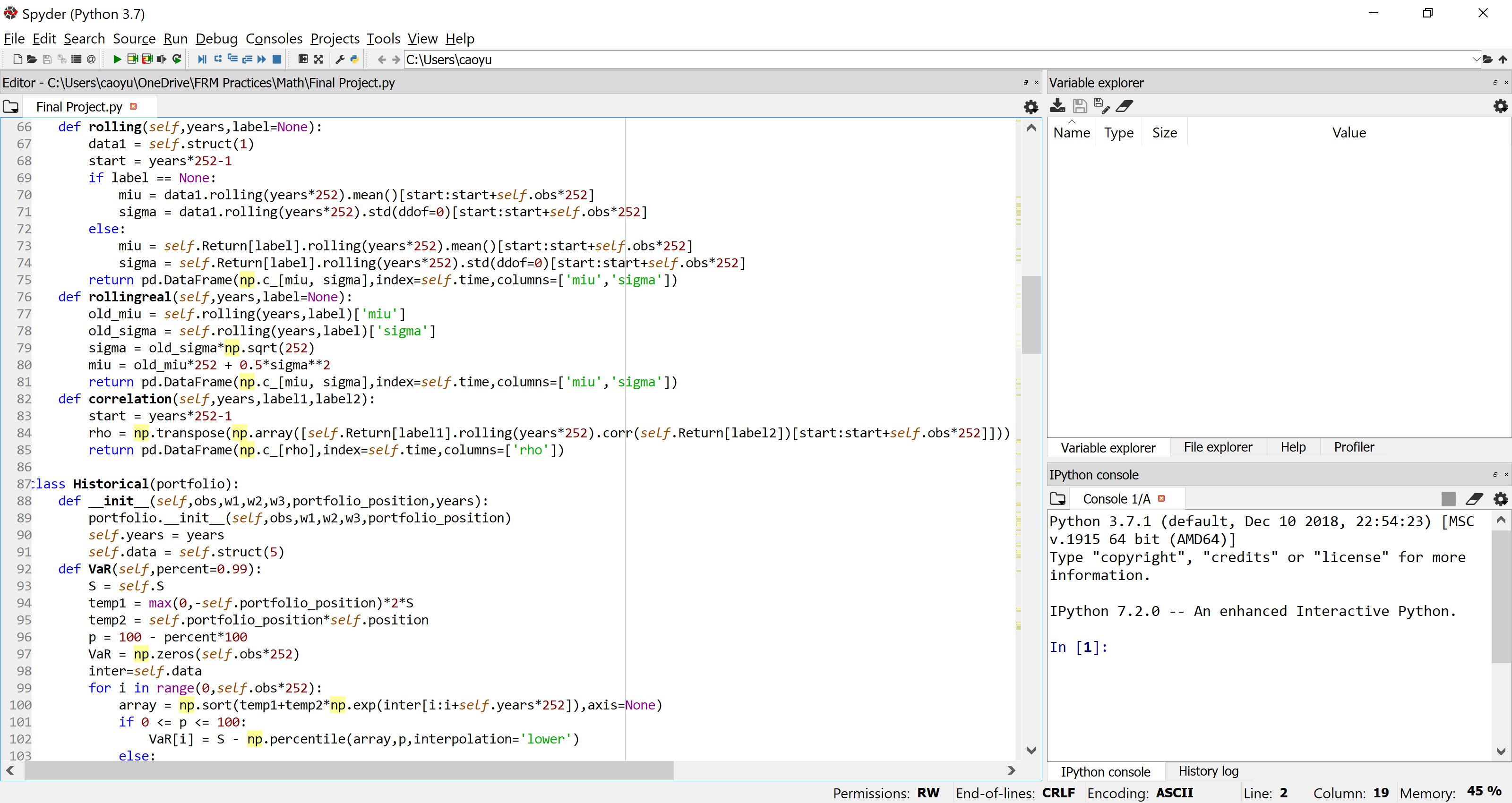
* **Correlation** (years, label1, label2):

years: years of the rolling window

label1: one of the logreturn data

label2: ome of the logreturn data

This function serves as the calculation of the correlation with a rolling window. A user can type three key arguments: ‘F’, ‘XRX’ and ‘SPX’.



## **Class: Historical**

The “Historical” class serves as the risk measurement using historical simulation methodology. It will use a 5-day return to measure the Value-at-Risk (VaR) and Expected Shortfall (ES) of the whole portfolio.

The input for this class includes:

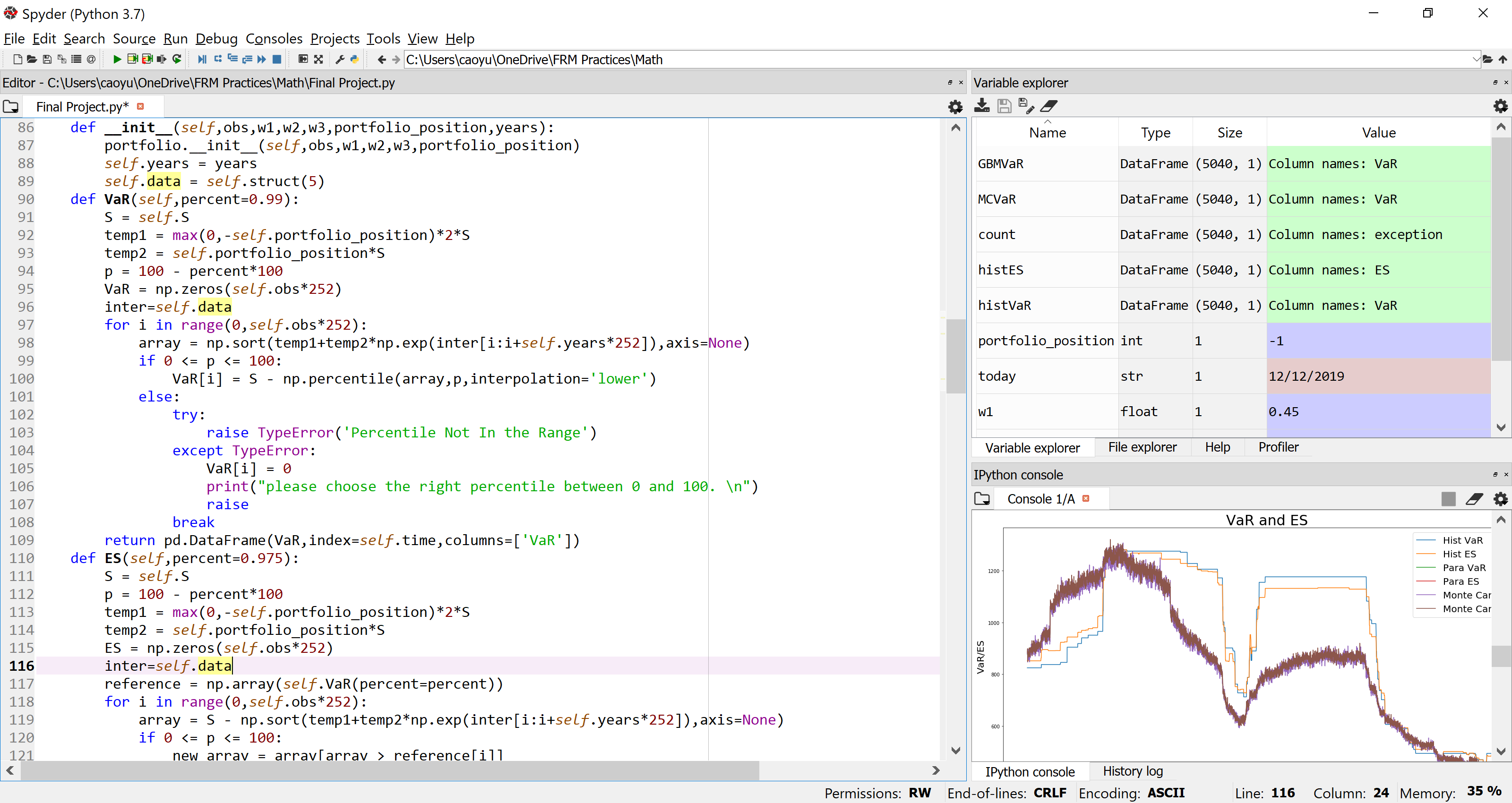
* ‘obs’: years of samples that a user arbitrarily chooses to observe. The total amount of observations will be ‘obs’ times 252 which represents a fiscal year for transactions.
* ‘w1’: weight of position on Ford stock.
* ‘w2’: weight of position on Xerox stock
* ‘w3’: weight of position on S&P 500 options
* ‘portfolio\_position’: position of the whole portfolio weighted. ‘-1’ for short position and ‘1’ for long position.
* ‘years’: the length of rolling window for calculating the VaR and ES.

Under this class, there are some functions in measuring the risk, which are called VaR and ES for VaR and ES calculation respectively.

* **VaR** (percent=0.99):

percent: the percentage of which VaR is calculated

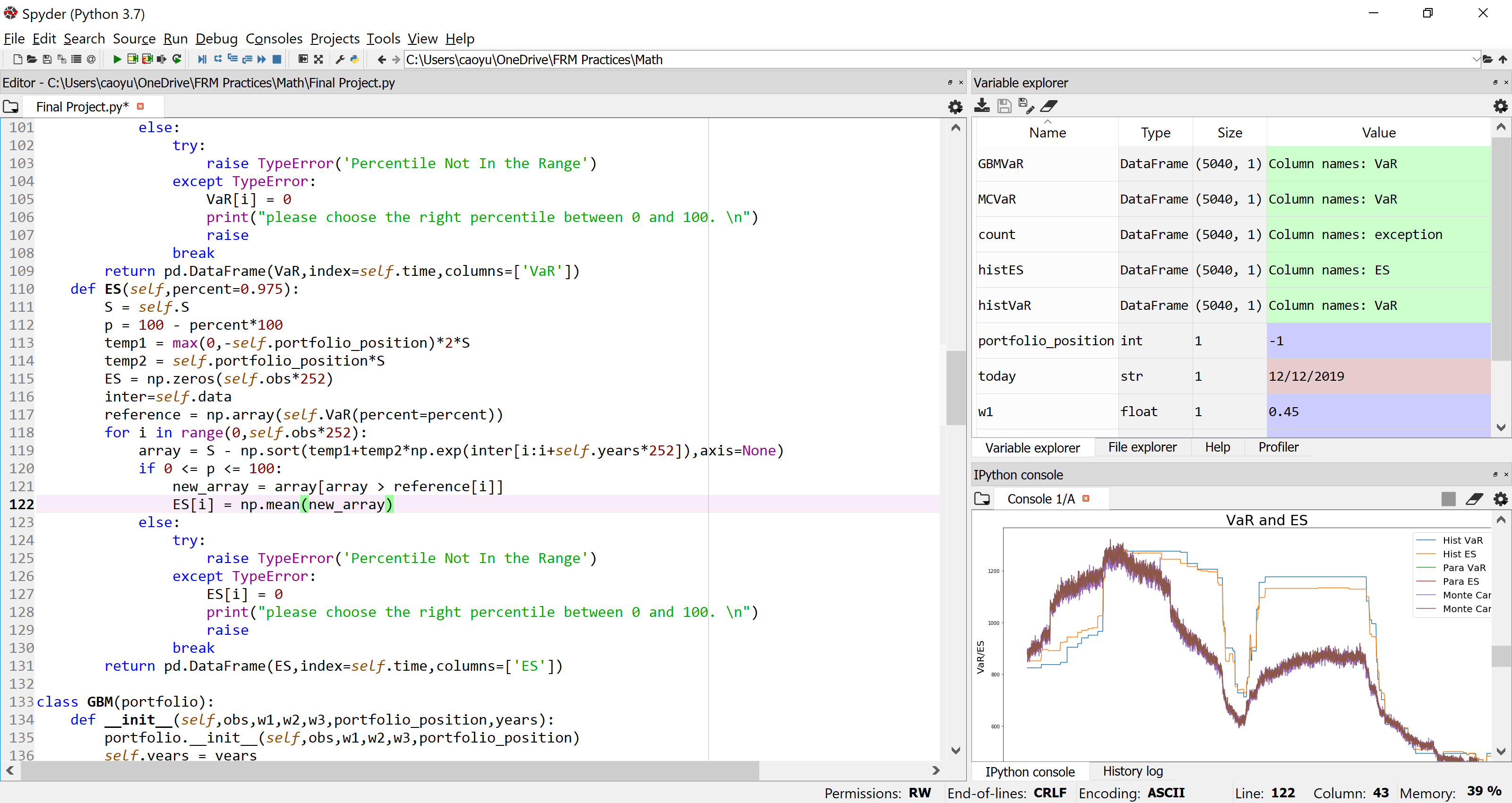
This VaR calculations specifies VaR measurement with a 5-day return under a rolling window predetermined by ‘years’ attributes. It will simulate the portfolio’s price by the historical return, and sort the result A user can change the ‘percent’ in (0,1).



* **ES** (percent=0.975)

percent: the percentage of which ES is calculated

This ES calculations specifies ES measurement with a 5-day return under a rolling window predetermined by ‘years’ attributes. It will average the losses from historical simulation that exceed VaR at 97.5%. A user can change the ‘percent’ in (0,1).



## **Class: GBM**

The “GBM” class serves as the risk measurement using Geometric Brownian Motion (GBM) methodology. It will use portfolio’s drift and volatility to measure the Value-at-Risk (VaR) and Expected Shortfall (ES) of the whole portfolio. Notice that in the Model Documentation, it is mentioned that Parametric VaR/ES is interchangeable under this model with GBM VaR/ES due to the nonlinearity payoff in option contracts.

The input for this class includes:

* ‘obs’: years of samples that a user arbitrarily chooses to observe. The total amount of observations will be ‘obs’ times 252 which represents a fiscal year for transactions.
* ‘w1’: weight of position on Ford stock.
* ‘w2’: weight of position on Xerox stock
* ‘w3’: weight of position on S&P 500 options
* ‘portfolio\_position’: position of the whole portfolio weighted. ‘-1’ for short position and ‘1’ for long position.
* ‘years’: the length of rolling window for calculating the VaR and ES.

Under this class, same as class “Historical”, there are some functions in measuring the risk, which are called VaR and ES for VaR and ES calculation respectively.

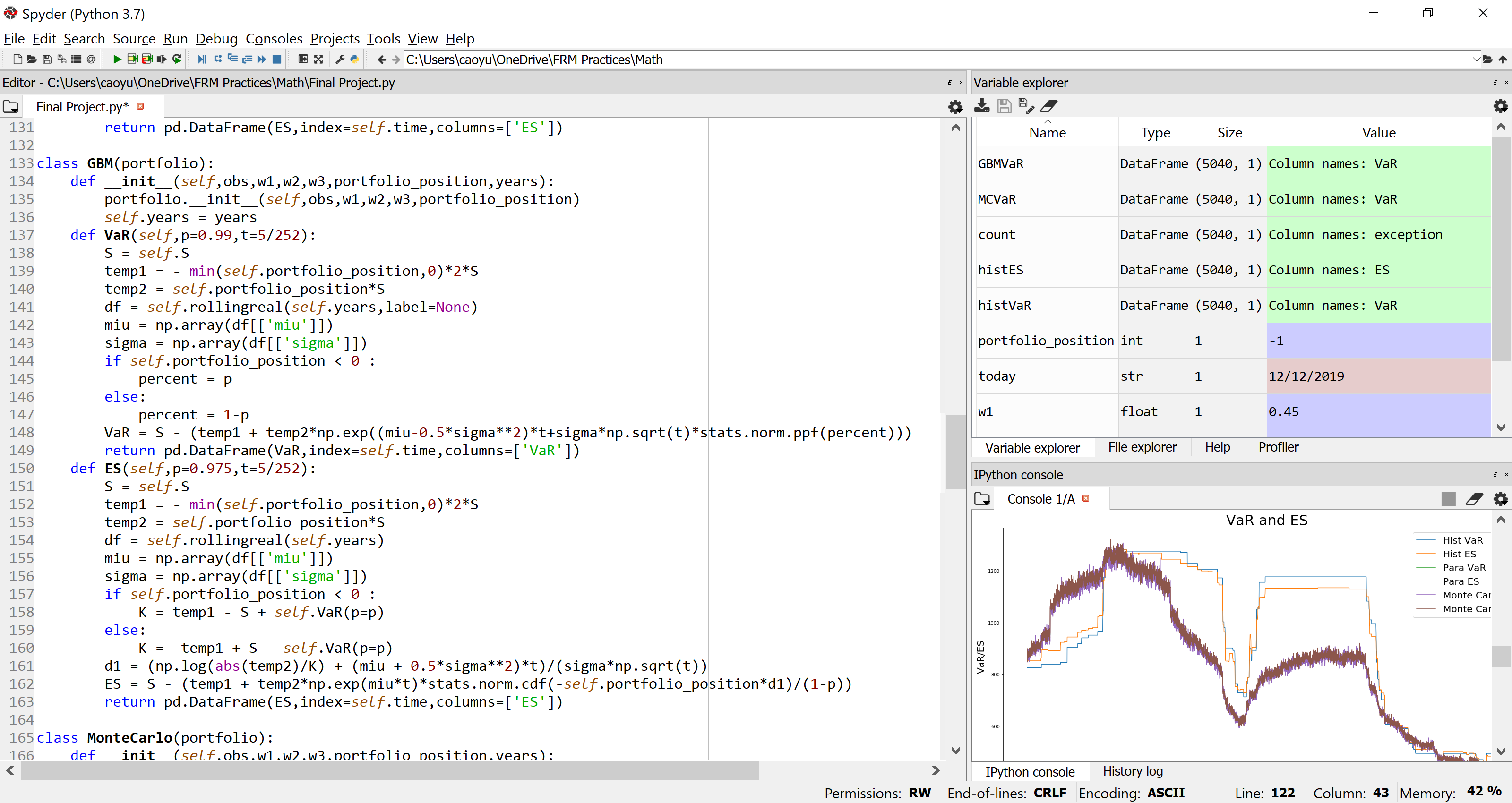
* **VaR** (p=0.99,t=5/252):

p: the percentage of which VaR is calculated

t: days used to calculate GBM VaR

This VaR calculations specifies VaR measurement of the portfolio following GBM in a time period t whose default value is 5/252 (that is 5-day). Drift and volatility have been calculated before in the class “portfolio”, given a predetermined rolling window “years”.

A user can change the ‘percent’ in (0,1) and t in a positive real number set.



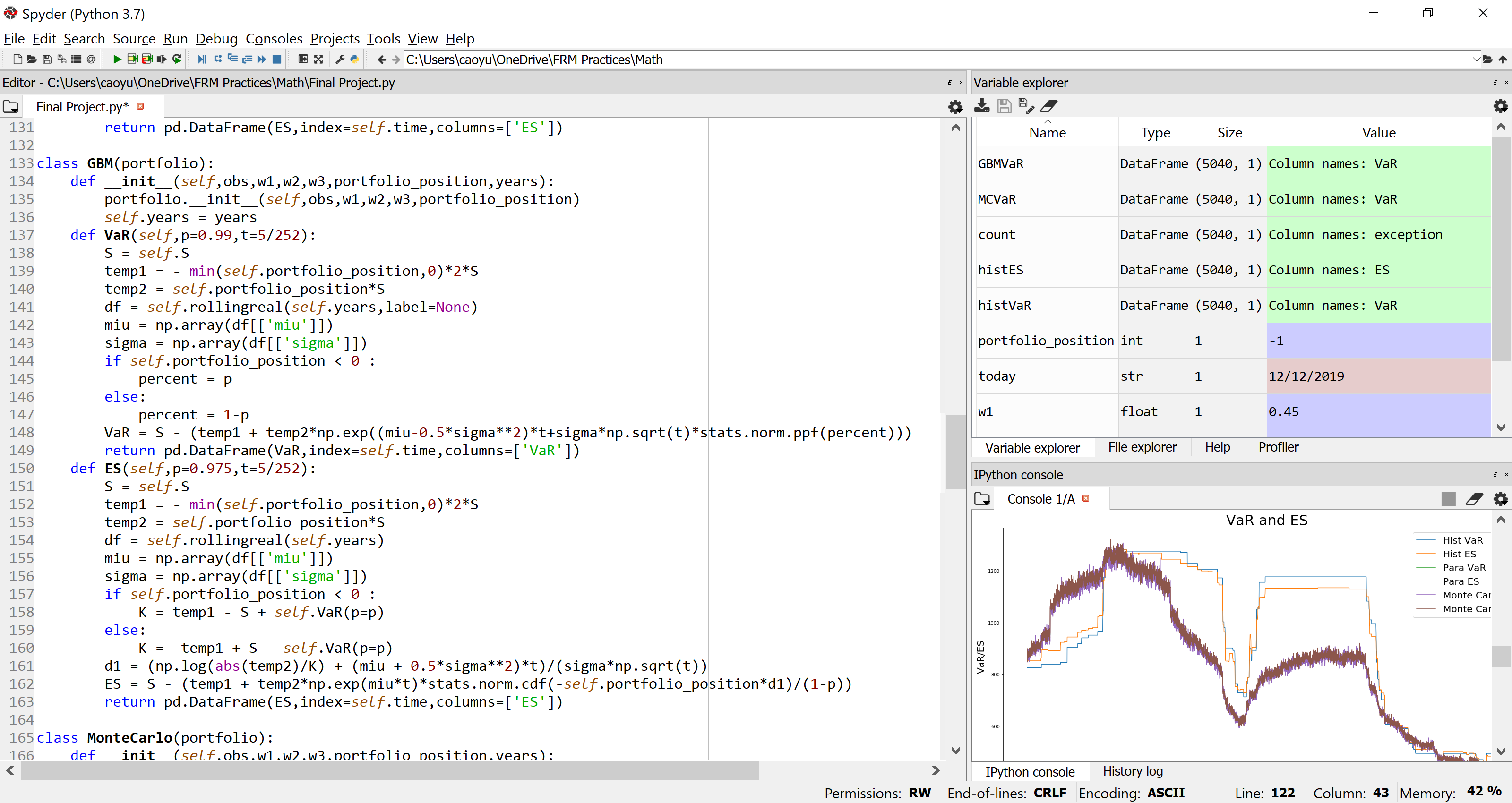
* **ES** (percent=0.975,t=5/252)

percent: the percentage of which ES is calculated

t: days used to calculate GBM ES

This ES calculations specifies ES measurement of the portfolio following GBM in a time period t whose default value is 5/252 (that is 5-day). Drift and volatility have been calculated before in the class “portfolio”, given a predetermined rolling window “years”. GBM ES takes average of losses that exceed GBM VaR at 97.5%.

A user can change the ‘percent’ in (0,1) and t in a positive real number set.



## **Class: MonteCarlo**

Monte Carlo Simulation serves as the simulation for the whole portfolio given their drift and volatility of the whole portfolio was calculated. It still applies GBM process for the whole portfolio in a time period.

The input for this class includes:

* ‘obs’: years of samples that a user arbitrarily chooses to observe. The total amount of observations will be ‘obs’ times 252 which represents a fiscal year for transactions.
* ‘w1’: weight of position on Ford stock.
* ‘w2’: weight of position on Xerox stock
* ‘w3’: weight of position on S&P 500 options
* ‘portfolio\_position’: position of the whole portfolio weighted. ‘-1’ for short position and ‘1’ for long position.
* ‘years’: the length of rolling window for calculating the VaR and ES.

Under this class, there are also two functions: VaRsimulation and ESsimulation which calculates VaR and ES based on simulated results for each day in the observation.

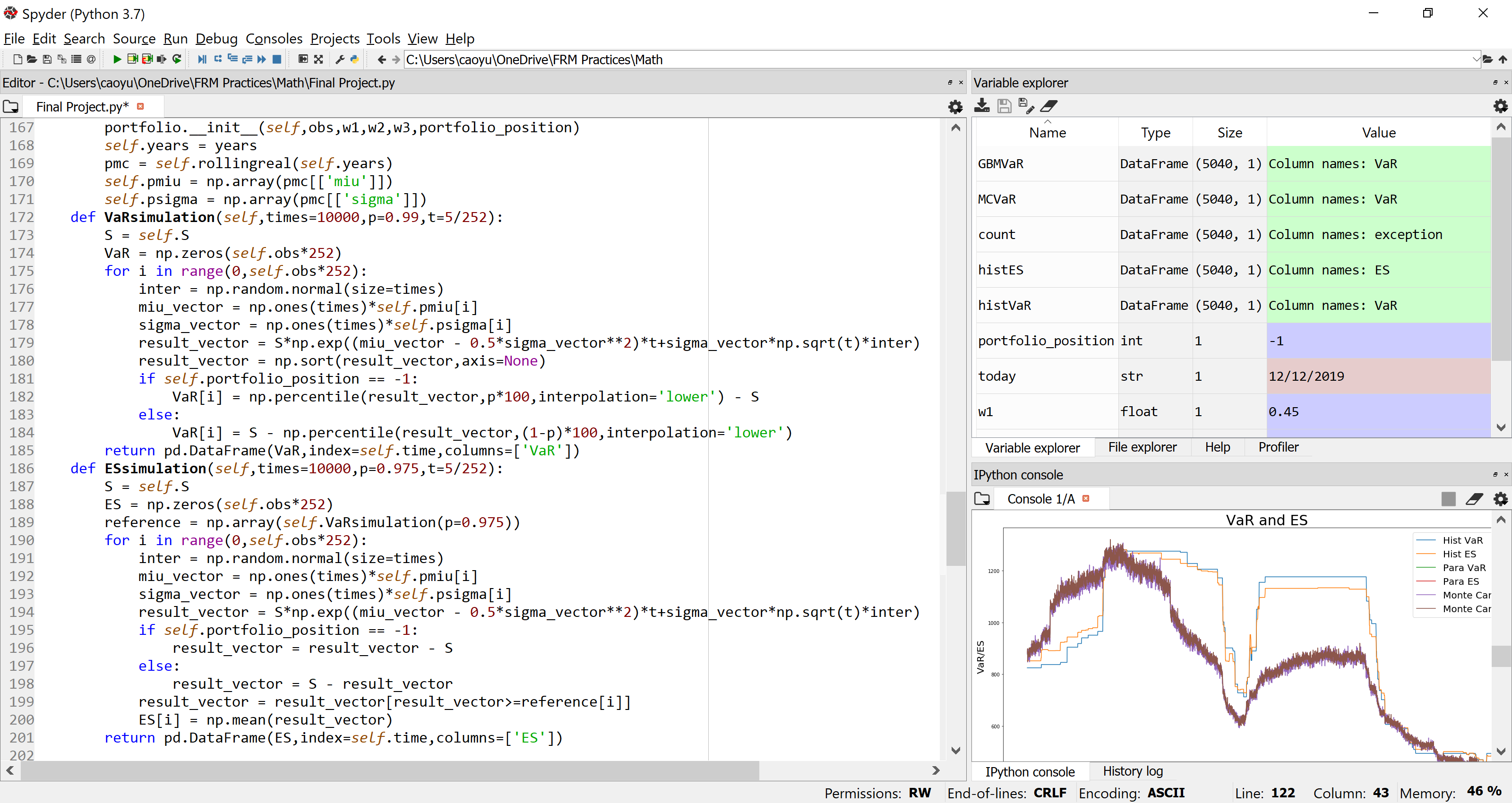
* **VaRsimulation** (times=10000, p=0.99, t=5/252):

times: times for Monte Carlo simulation on each day on the observation

p: the percentage of which VaR is calculated

t: days used to calculate GBM VaR

The default value of iteration for the creating path is set to 10,000. Each day drift and volatility are calculated that can be used to determine the new portfolio’s price under GBM process. Default time for the price movement is 5/252 (that is 5 days). Therefore, 10,000 5-day P&L is automatically computed. By sorting it, a certain percentile—default as 99%—of P&L is Monte Carlo VaR.



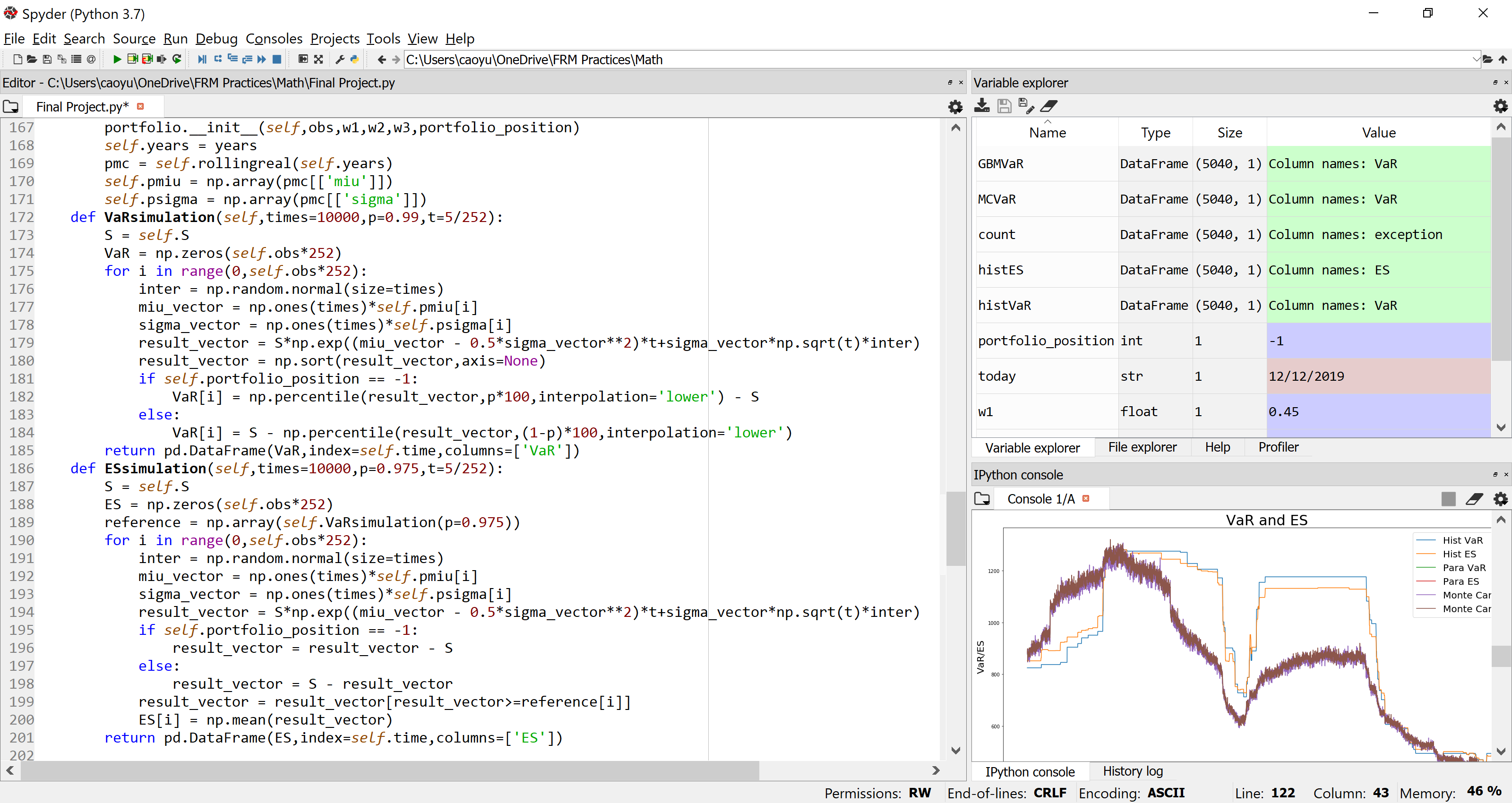
* **ESsimulation** (times=10000, p=0.975, t=5/252):

times: times for Monte Carlo simulation on each day on the observation

p: the percentage of which ES is calculated

t: days used to calculate Monte Carlo ES

The default value of iteration for the creating simulation is set to 10,000. Each day drift and volatility are calculated that can be used to determine the new portfolio’s price under GBM process. Default time for the price movement is 5/252 (that is 5 days). Therefore, 10,000 5-day P&L is automatically computed. Default percentage as 97.5%, Monte Carlo ES will average the losses that exceed 97.5% Monte Carlo VaR.



## **Class: Backtesting**

Backtesting is a methodology to testify the validity of a VaR model. It will compare the real losses incurred to the VaR calculated. Then it will count the exception where the loss exceeds VaR. This model will apply a one-year rolling window to count total exceptions for a year. According to BASEL Accord, if the exception is lower than five, the model is in the Green Zone, and there will be no punishment for the capital requirement for the market risk.

The input for this class includes:

* ‘obs’: years of samples that a user arbitrarily chooses to observe. The total amount of observations will be ‘obs’ times 252 which represents a fiscal year for transactions.
* ‘w1’: weight of position on Ford stock.
* ‘w2’: weight of position on Xerox stock
* ‘w3’: weight of position on S&P 500 options
* ‘portfolio\_position’: position of the whole portfolio weighted. ‘-1’ for short position and ‘1’ for long position.
* ‘years’: the length of rolling window for calculating the VaR and ES.
* ‘typeVaR’: Type of the VaR model to validate

Please be aware that a user can customize ‘Type’ in one out of the four options: ‘GBM’ for Parametric VaR, ‘MC’ for Monte Carlo VaR, and ‘Hist’ for Historical VaR.

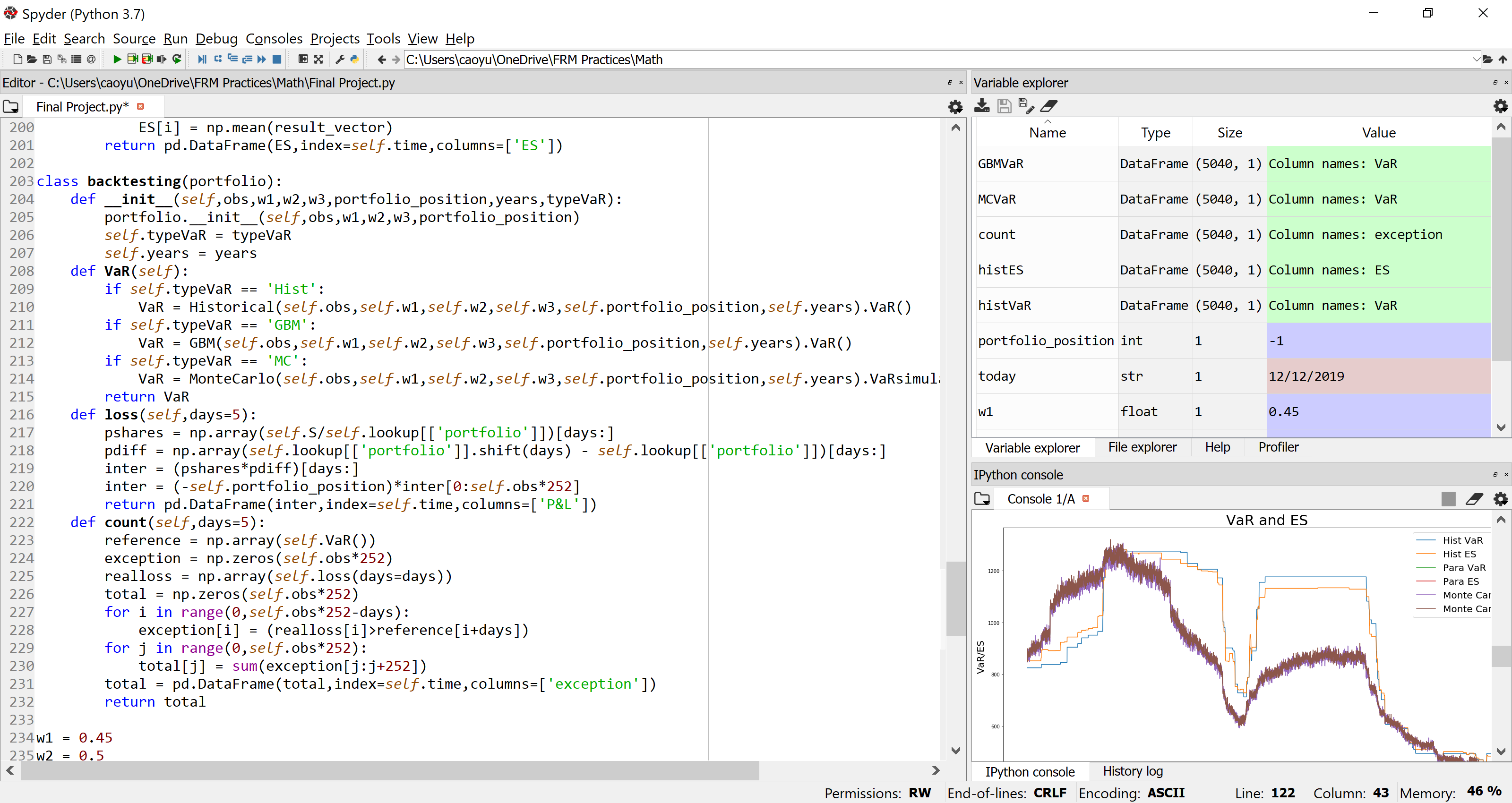
Under this class, three functions are established for assistance in the backtesting:

* **VaR** ():

this function serves as the establishment of the VaR given in the ‘typeVaR’.

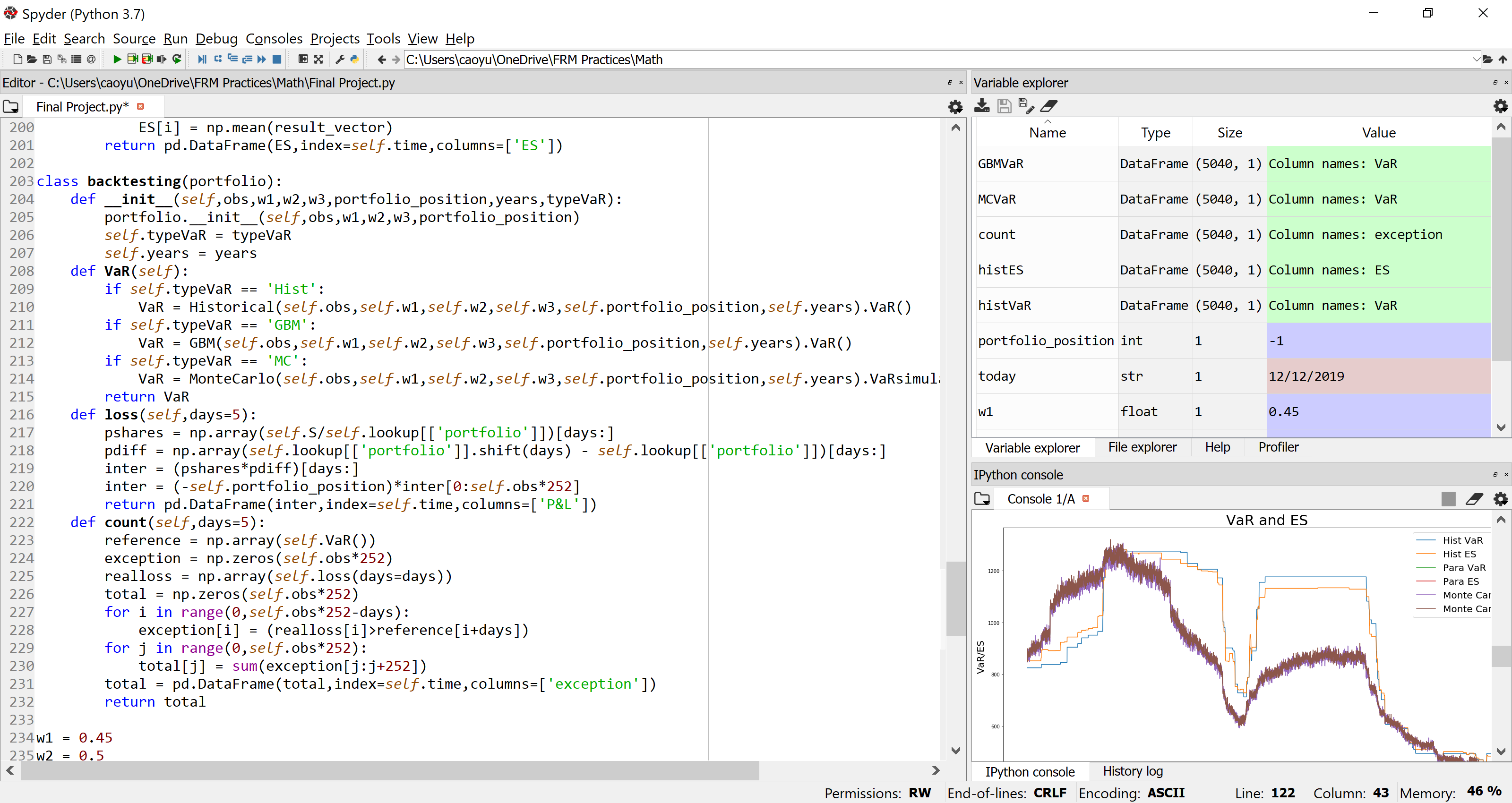
* **loss** (days=5):

This function serves as the calculation of the real loss of the portfolio given by the day delay whose default value is five day.



* **Count** (days=5):

This function serves as the counting process of the exception over a one-year rolling window. This exception detection is conducted by comparing the VaR and real loss that incurs in a given day—indicated as ‘days’ in the function whose default value is five— after VaR is computed. Then we use a one-year rolling window to cumulate the result of the detection for out exception counting.



# **Main function**

In the main function, seven major dataframes will be established and some graphs will be plotted.

Dataframes consists of Historical VaR and historical ES, parametric VaR and parametric ES, Monte Carlo VaR and Monte Carlo ES, and exception counting results. All dataframes shares the same index as dates over the observation period that a user has already chosen.

Graphs includes the VaR and ES results, Exception counting results for different VaR, and Real loss VS different VaR comparison.