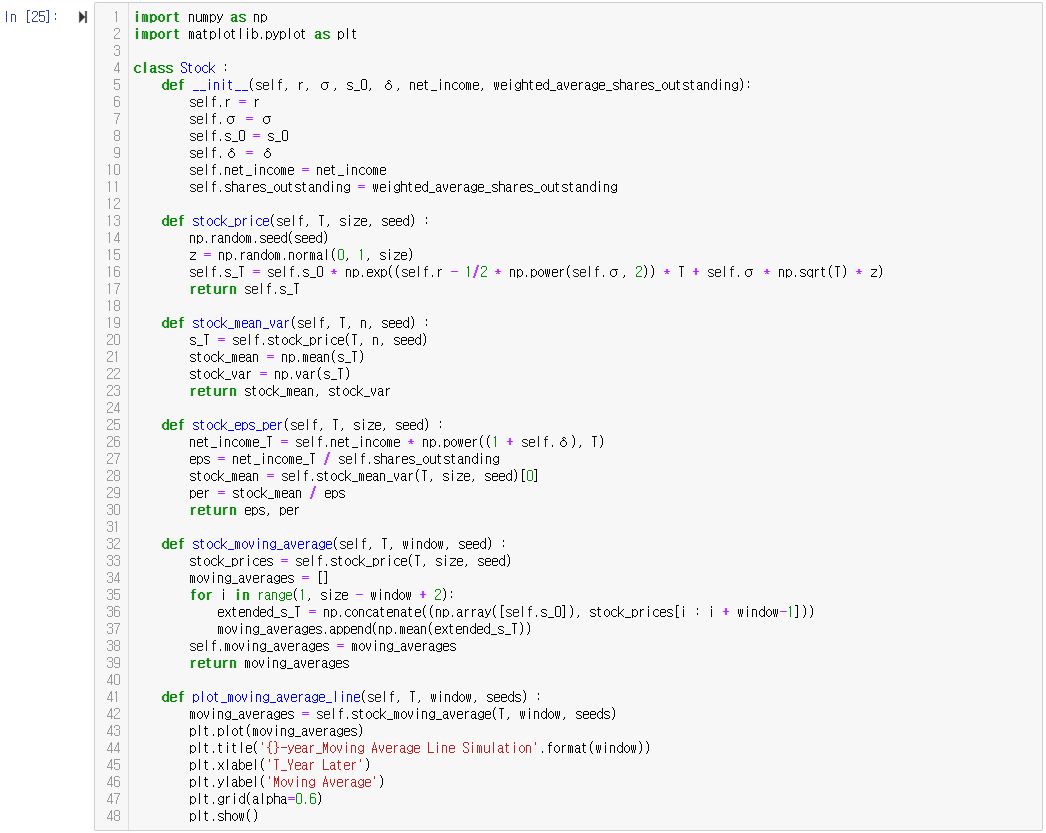
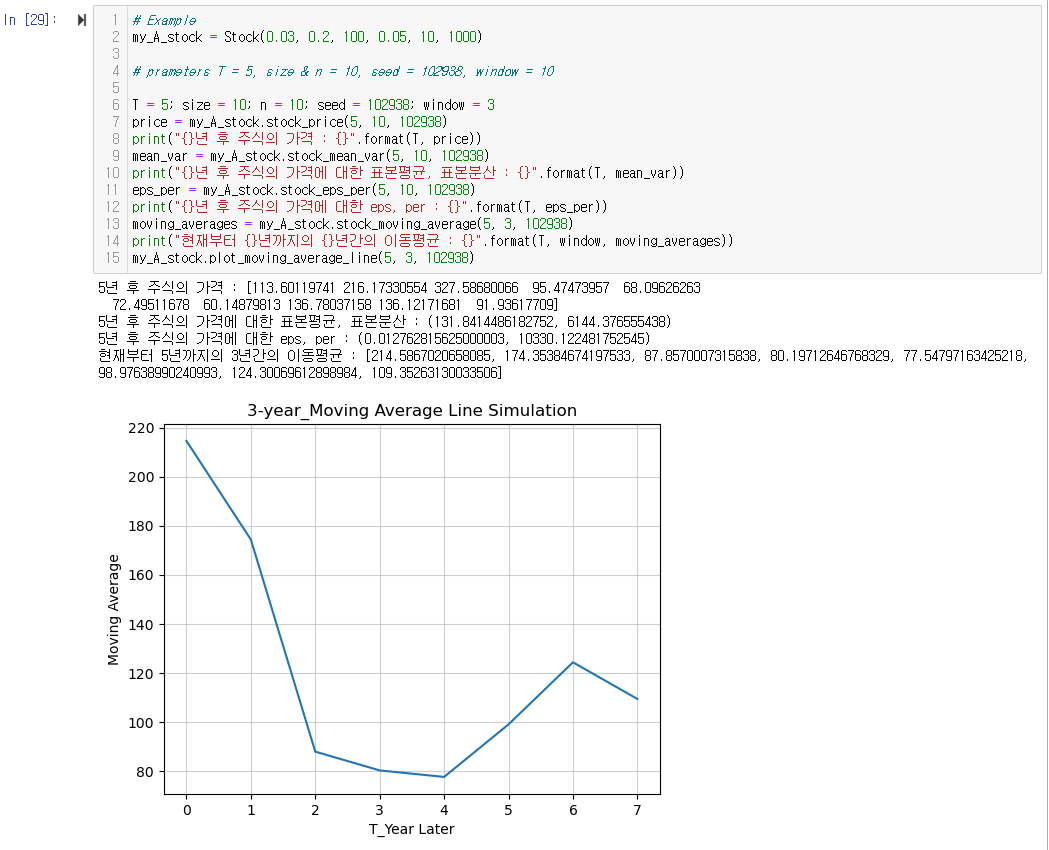
금융프로그래밍1 기말프로젝트

202221520 여지성

코드 :



예제 :



주석 :

Stock class

ㆍ멤버변수

r = (1년간) 주식 수익률

σ = (1년간) 주식 변동성

s\_0 = 현재 주식의 가격

δ = (1년간) 당기순이익 변동성

net\_income = 현재 당기순이익

weighted\_average\_shares\_outstanding = 현재 발행 주식수

ㆍ라이브러리

Import 라이브러리명 as A : 라이브러리를 불러와 사용하되 본문에서는 A라고 지칭

사용한 라이브러리 :

numpy : 숫자와 관련된 클래스 및 함수들이 효율적으로 구현되어 저장되어있는 라이브러리

matplotlib : 파이썬에서 그래프를 그리기 위한 라이브러리로 특히 pyplot은 matplotlib 내의

하위 모듈로 데이터 시각화를 더욱 쉽게하며

그래프 종류로는 선그래프, 막대 그래프, 점 등이 존재

ㆍ멤버함수

1. stock\_price : T년 후 주식의 가격 시뮬레이션

가정 : s\_T = s\_0 × e^{ [ (r - 1/2 × σ^2) x T ] + σ × sqrt(T) × z } | z ~ N(0, 1)

매개변수 T : 주가를 계산할 시간(년)

size : 주가의 로그 변동을 모델링하기 위한 랜덤 변수(z)의 크기

seed : 난수 생성시 사용할 시드값

설명 :

블랙-숄즈-머튼 모형 등의 옵션 가격 모형에서 사용되는 모형과정 중 하나로 주가의

로그변동이 정규분포를 따른다는 가정하에 주가의 로그 변동을 모델링 하기 위해

정규 분포를 사용하여 랜덤 변수 z 생성한 뒤, T년 후 주가 시뮬레이션하여 리턴

사용된 코드 :

np.random.seed(seed number) : 알고리즘 PRNG와 seed number를 사용한 난수 생성

np.random.normal(m, s, size) : 평균 = m, 분산 = s인 정규분포에서 size개의 샘플 생성

self. : 인스턴스 메서드 내에서 인스턴스 변수 설정, 인스턴스 변수 값에 접근 가능

1. stock\_mean\_var : T년 후 주가 샘플 n개에 대한

표본평균(stock\_mean)과 표본분산(stock\_var) 계산

가정 : E[S\_T] = S\_0 × e^{r × T} , Var[S\_T] = S\_0^2 × e^{2 × r × T} × (e^{σ^2 × T} - 1)

매개변수 T : 주식 가치를 계산할 대상 시간(년)

n : 시뮬레이션 실행 횟수

seed : 난수 생성시 사용할 시드값

설명 :

stock\_price를 호출하여 T년 후의 주가를 계산한 뒤,

n개의 s\_T의 표본평균(mean)과 표본분산(var) 계산하여 리턴

사용된 코드 :

np.mean( ) : ( )안의 표본평균 계산

np.var( ) : ( )안의 표본분산 계산

self. : 인스턴스 메서드 내에서 인스턴스 변수 설정, 인스턴스 변수 값에 접근 가능

1. stock\_eps\_per : T년 후 주가 샘플 n개에 대한 표본평균의 EPS와 PER 계산

가정 : 당기순이익은 δ에 따라 일정하게 변동 -> net\_income\_T = net\_income \* (1 + δ)^T,

weighted\_average\_shares\_outstanding(현재 발행 주식수)는 일정하게 유지 ,

per 계산 시, 주가의 로그변동이 정규분포를 따른다는 가정하에 계산한 T년 후

주가 샘플 n개의 평균 사용

매개변수 T : 주식 가치를 계산할 대상 시간(년)

size : 주가의 로그 변동을 모델링하기 위한 랜덤 변수(z)의 크기

seed : 난수 생성시 사용할 시드값

설명 :

가정에 따라 T년 후의 당기순이익인 net\_income\_T를 변동성 δ로 계산하고

Stock\_mean\_var함수를 호출하여 stock\_mean 계산한 뒤

‘eps = 당기순이익 / 총 발행 주식수, per = 주가 / eps ‘공식에 따라

T년 후 주가 샘플 n개에 대한 표본평균의 EPS와 PER 계산하여 리턴

사용된 코드 :

np.power(밑 , 지수) : 밑^지수 값을 나타냄

self. : 인스턴스 메서드 내에서 인스턴스 변수 설정, 인스턴스 변수 값에 접근 가능

1. stock\_moving\_average : 현재부터 T년까지의 window년 간의 주가의 이동평균 계산

가정 : window일이 아닌 winow년 간의 이동평균선

년당 하나의 종가를 가지며 종가의 이동평균 계산

기하 브라운 운동체 모델을 사용한 주식의 가격 모형에 따라 미래의 주가 계산

매개변수 T : 주식 가치를 계산할 대상 시간(년)

window : 이동평균 기간(년)

seed : 난수 생성시 사용할 시드값

설명 :

현재 주가를 포함해 window년 동안의 주가의 이동평균을 계산하기 위해 stock\_price 함수를 호출하여 미래의 주가를 계산하고 for문을 이용해 현재 주가를 포함한 s\_T 값들을 돌면서 각각의 이동평균 계산 후 이동평균의 값을 moving\_average에 담아 리턴

사용된 코드 :

for문 : 리스트 값, 범위, 딕셔너리 값, 문자열 등을 순회하며 작업 수행 -> 반복문 수행

np.concatenate(a, b) : a와 b 접합 (list에서의 +와 같은 역할)

np.mean( ) : ( )안의 표본평균 계산

.append( ) : list의 마지막에 원소를 추가하는 함수

self. : 인스턴스 메서드 내에서 인스턴스 변수 설정, 인스턴스 변수 값에 접근 가능

1. stock\_moving\_average\_line : 현재부터 T년까지의 window년 간의 주가의 이동평균선 생성

가정 : 종가의 window년 이동평균에 대한 시계열 데이터를 통해 이동평균 선 생성

매개변수 T : 주식 가치를 계산할 대상 시간(년)

window : 이동평균 기간(일)

seeds :난수 생성시 사용할 시드값 리스트

설명 :

Stock\_moving\_average 함수를 호출하여 현재부터 T년까지의 주가의 이동평균을 계산한 후, matplotlib.pyplot을 사용해 이동평균선 생성

사용된 코드 :

plt.plot( ) : ( )에 대한 선 그래프 생성

plt.title( ) : 그래프의 제목 설정

plt.xlabel( ) : x축 레이블 설정

plt.ylabel( ) : y축 레이블 설정

plt.grid(alpha = ) : 그래프 내 그리드 추가 ( alpha: 그리드 투명도 설정)

plt.show(): 그래프를 화면에 표시

self. : 인스턴스 메서드 내에서 인스턴스 변수 설정, 인스턴스 변수 값에 접근