Response for Project 6

Problem 1

Implement the closed form greeks for GBSM

Calculate the greeks according to the following tables.

Greek	Definition	Formula GBSM
Delta, Δ	First Derivative of Price with respect to underlying price, $\frac{\delta P}{\delta S}$	• Call: $e^{(b-r)T}\Phi(d_1)$ • Put: $e^{(b-r)T}\left(\Phi(d_1) - 1\right)$

Gamma, Γ	$\frac{\delta^2 P}{\delta S^2}$ Also called Convexity	$\frac{f(d_1)e^{(b-r)T}}{S\sigma\sqrt{T}}$, $f(x)$ is the normal PDF
Vega	$\frac{\delta P}{\delta \sigma}$	$Se^{(b-r)T}f(d_1)\sqrt{T}$
Theta, Θ	$-\frac{\delta P}{\delta T}$ Derivative is negative but often expressed as a positive number. Also called Theta Decay	• Call: $-\frac{Se^{(b-r)T}f(d_1)\sigma}{2\sqrt{T}} - (b-r)Se^{(b-r)T}\Phi(d_1) - rXe^{-rT}\Phi(d_2)$ • Put: $-\frac{Se^{(b-r)T}f(d_1)\sigma}{2\sqrt{T}} + (b-r)Se^{(b-r)T}\Phi(-d_1) + rXe^{-rT}\Phi(-d_2)$
Rho, ρ	$\frac{\delta P}{\delta r}$ Note, formulas are for Black Scholes where rf=b	• Call: $TXe^{-rT}\Phi(d_2)$ • Put: $-TXe^{-rT}\Phi(-d_2)$
Carry Rho	$\frac{\delta P}{\delta b}$	• Call: $TSe^{(b-r)T}\Phi(d_1)$ • Put: $-TSe^{(b-r)T}\Phi(-d_1)$

Implement a finite difference derivative calculation

I set the small increment to be 0.001 and use forward-diffence to calculate Delta, Vega, Theta, Rho, and Carry Rho.

$$f'(x) \approx \frac{f(x+\Delta) - f(x)}{\Delta}$$

I use the following finite difference function to calculate Gamma.

$$f'(x) \approx \frac{f(x+\Delta) + f(x-\Delta) - 2f(x)}{\Delta^2}$$

Compare the values between the two methods for both a call and a put

	GBSM_call	finite_diff_call	GBSM_put	finite_diff_put
delta	0.510071	0.510073	-0.489450	-0.489448
gamma	0.040173	0.040174	0.040173	0.040174
vega	19.776582	20.286657	19.776582	19.287136
theta	21.628607	21.622548	22.090281	22.084222
rho	NaN	-0.355829	NaN	-0.359604
carry_rho	7.609135	7.609616	-7.301527	-7.301113

Two methods have almost the same results. Deltas are ranged from -1 to 1. The sum of the two deltas approach to 1. Gammas are positive and very small. The sensitivities of time to maturity and implied volatility are large. The sensitivity of carry costs associated with American call is positive, while the sensitivity of carry costs associated with American put is negative.

Implement the binomial tree valuation for American options with and without discrete dividends. Calculate the value of the call and the put.

Steps for implementing the binomial tree valuation:

1. If there are no dividends during the life of the option, use the recombining tree method previously implemented.
2. Construct the nodes up to the first dividend payment. For each node
   - Value No Exercise: Recursively call the non-recombining tree with the stock price minus the dividend and any remaining dividends to be paid.
   - Value Exercise: Payoff function with the price before the dividend.
3. Using backward induction, find the value at the starting node.

Results:

Price of American call with dividend	Price of American put with dividend
3.856	4.417

Calculate the Greeks of each. What is the sensitivity of the put and call to a change in the dividend amount?

I set the small increment to be 0.001 and use forward-difference to calculate Delta, Vega, Theta, Rho, and sensitivity of dividend amount.

Since Gamma is too small to be identified if I set the increment to be 0.001, I adjust the increment to be 1 for Gamma.

	AmericanCallWithDiv	AmericanPutWithDiv
delta	0.521444	-0.486405
gamma	0.038660	0.507978
vega	19.859877	19.978669
theta	22.143119	21.882027
rho	6.617093	-7.575673
div_amount	-0.153617	0.507545

The sensitivity of the put and call to a change in the dividend amount is 0.507 and -0.15 respectively.

Problem 2

Fit a Normal distribution to AAPL returns – assume 0 mean return. Simulate AAPL returns 10 days ahead and apply those returns to the current AAPL price (above). Calculate Mean, VaR and ES.

Steps:

1. For each option, calculate the implied volatility given the underlying inputs.

```
currUnderlying = 164.85
current = pd.to_datetime("2022-02-25")
rf = 0.0025
coupon = 0
b = rf - coupon
initVol = 0.5
daysInYear = 365
divAmounts = [1]
divTime = pd.to_datetime("2022-03-15")
divTimes = [(divTime - current).days]
N = days
```

2. Set the increment to be 0.001 and calculate Delta for each option in the portfolio using forward difference and the Binary Tree model for American option with dividends.
3. Calculate each portfolio's present value.
4. Calculate each portfolio's gradient using the following formula.

$$\frac{dR}{dr_i} = \frac{P_i}{PV}\Sigma_{j=1}^{m} h_j \delta_j$$

5. Calculate each portfolio's standard deviation using the following formula.

$$\sigma_p = \sqrt{\Delta R^T \Sigma R}$$

6. Simulate portfolio returns 10 days ahead assuming normal distribution with location equaling to 0 and scale equaling to each porfolio's standard deviation.

7. Calculate the mean, VaR, and ES using the following fomulas

$$VaR_T(\alpha) = -PV * F_x^{-1}(\alpha) * \sqrt{T}\sigma_p$$

$$ES_T(\alpha) = -PV * E(X|x \leq -F_x^{-1}(\alpha)) * \sqrt{T}\sigma_p$$

Compare these results to last week's results.

Delta normal:

	Mean	VaR	ES
Straddle	8.895647	0.666924	0.832880
SynLong	0.105911	14.078145	17.854311
CallSpread	3.771095	5.433957	6.862093
PutSpread	2.807378	4.215210	5.264990
Stock	164.856765	13.758407	17.302377
Call	4.489631	7.372534	9.175613
Put	4.434907	6.705610	8.389813
CoveredCall	160.288056	6.385873	7.811404
ProtectedPut	169.180734	7.052797	8.702303

Last week's results:

	Mean	VaR	ES
Straddle	9.238795	2.444581	2.452591
SynLong	-0.053501	13.477768	16.689049
CallSpread	3.529414	3.627343	3.710826
PutSpread	3.184107	2.666481	2.740758
Stock	164.820570	13.258781	16.449877
Call	4.592647	4.346952	4.430713
Put	4.646148	4.245900	4.331782
CoveredCall	160.227924	8.911828	12.019164
ProtectedPut	169.466718	4.127965	4.191541

The results are quite similar.

Problem 3

Use the Fama French 3 factor return time series as well as the Carhart Momentum time series to fit a 4 factor model to the following stocks.

1. For each stock, fit a 4 factor FF model with an alpha to find betas considering the last 60 days returns.

$$r_s - R_{r_f} = \alpha + \beta_{mkt}(r_{mkt} - r_{r_f}) + \beta_{SMB}SMB + \beta_{HML}HML + \beta_{MOM}MOM + \epsilon_s$$

Based on the past 10 years of factor returns, find the expected annual return of each stock.

Calculate the geographic annualized expected return based on 10 years of factor returns.

$$E(r_s - r_f) = log(1 + \Sigma(\beta_i * E(r_i))) * 255$$

stock	expected_annual_return
AAPL	0.143640
FB	0.228551
UNH	0.141535
MA	0.277833
MSFT	0.172325
NVDA	0.320119
HD	0.096481
PFE	-0.163102
AMZN	0.195017
BRK_B	0.116525
PG	0.079122
XOM	0.184257
TSLA	0.177962
JPM	0.135935
V	0.200537
DIS	0.159945
GOOGL	0.197218
JNJ	0.079525
BAC	0.177901
CSCO	0.142407

Construct an annual covariance matrix for the 10 stocks.

Calculate the geographic annualized covariance based on 10 years of factor returns.

$$\Sigma_s = cov(log(1 + r_s)) * 255$$

```
            AAPL        FB       UNH  ...       JNJ       BAC      CSCO
AAPL    0.065441  0.031073  0.020744  ... -0.003238  0.005305  0.011880
FB      0.031073  0.104613  0.008503  ...  0.005036  0.008607  0.016598
UNH     0.020744  0.008503  0.044678  ...  0.008787  0.003671  0.015559
MA      0.010865  0.040435  0.025495  ...  0.013416  0.040533  0.017995
MSFT    0.039993  0.037940  0.022884  ... -0.001555 -0.000343  0.022863
NVDA    0.081306  0.071235  0.037212  ... -0.013139  0.017036  0.013277
HD      0.020419  0.007342  0.016155  ...  0.002178  0.007904  0.003715
PFE    -0.021341 -0.034076 -0.006501  ...  0.006779 -0.034283 -0.002152
AMZN    0.041714  0.039219  0.018861  ... -0.003340  0.002833  0.002798
BRK_B   0.000136  0.009335  0.002269  ...  0.009128  0.028815  0.007915
PG     -0.002699  0.000548  0.011324  ...  0.014588  0.008013  0.008888
XOM     0.008210  0.016634  0.009566  ...  0.002300  0.047495  0.014596
TSLA    0.081903  0.070407  0.024662  ... -0.017519  0.027099  0.013973
JPM     0.005251  0.003429  0.003660  ...  0.006472  0.051533  0.012274
V       0.010712  0.034626  0.018395  ...  0.007931  0.034852  0.012377
DIS     0.012826  0.029683  0.010282  ...  0.003500  0.019232  0.019587
GOOGL   0.031423  0.036417  0.021946  ...  0.001318  0.005473  0.019005
JNJ    -0.003238  0.005036  0.008787  ...  0.022242  0.005681  0.005765
BAC     0.005305  0.008607  0.003671  ...  0.005681  0.064756  0.014238
CSCO    0.011880  0.016598  0.015559  ...  0.005765  0.014238  0.054527
```

Assume the risk free rate is 0.0025. Find the super efficient portfolio.

Steps:

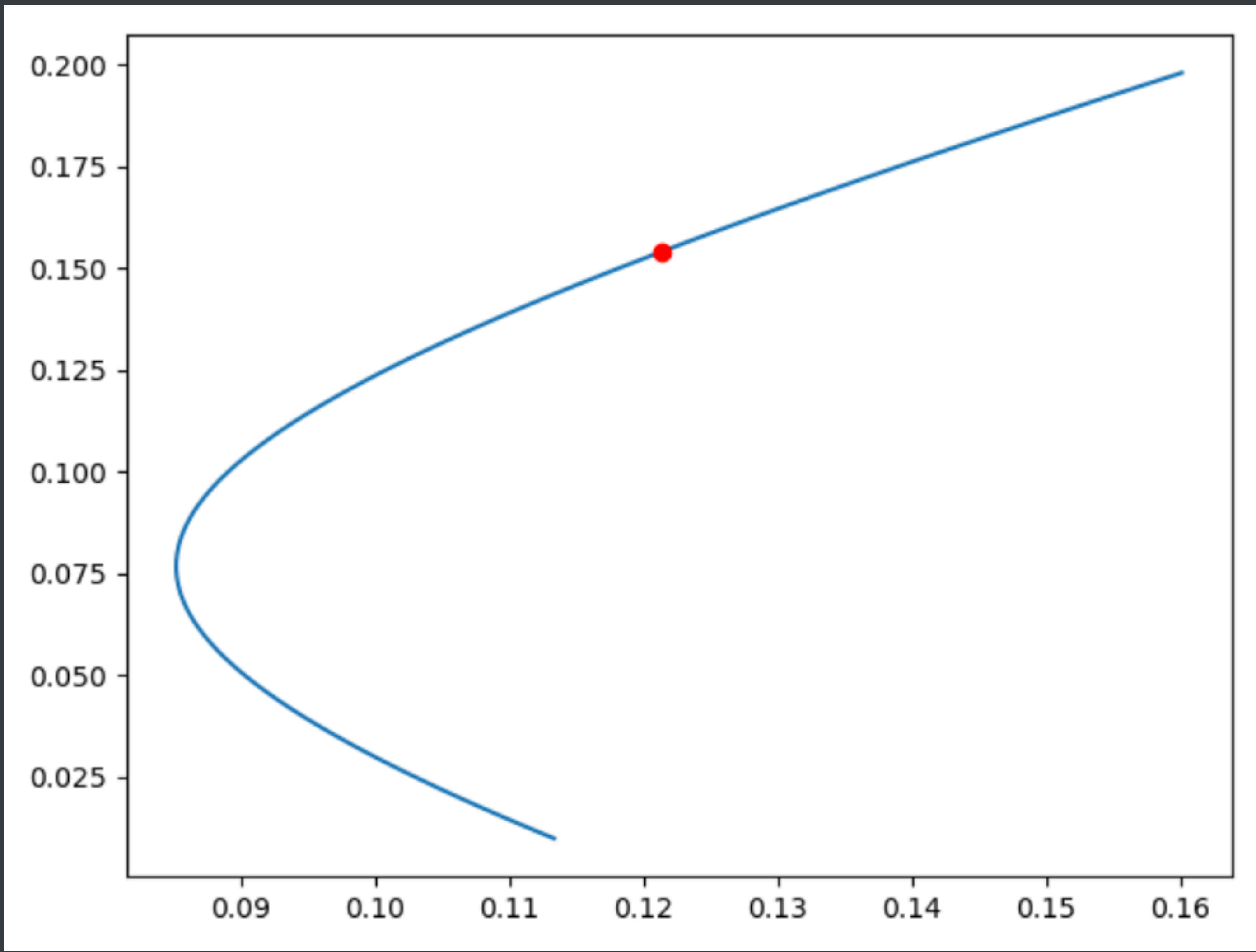
1. Define the optimization function to be

$$min\ \sigma^2 = w^T \Sigma w$$
$$s.t.\ \Sigma_{i=1}^{n} w_i = 1,\ w^T \mu = R$$

2. Set initial weights to be equally weighted decimals and pass them into the optimization function. Let the optimization function find the minimum weights given each specified R.

3. Pass the minimum weights into the objective function to find the minimized standard deviation. Draw the efficient frontier.
4. Compare the sharpe ratio for each R, find the supter efficient portfolio.

Results:



The best efficient portfolio is

	stock	weights
0	AAPL	-0.041149
1	FB	0.023992
2	UNH	0.049279
3	MA	0.072491
4	MSFT	0.089708
5	NVDA	-0.011556
6	HD	0.021025
7	PFE	-0.009398
8	AMZN	0.209739
9	BRK_B	0.255620
10	PG	0.028510
11	XOM	0.002188
12	TSLA	-0.001467
13	JPM	0.014959
14	V	-0.011077
15	DIS	0.015932
16	GOOGL	0.037162
17	JNJ	0.133667
18	BAC	0.040659
19	CSCO	0.079718

Return	Volatility	Sharpe
0.154	0.122	1.246