Fintech 545 Assignment 6

Leo Zeng
Nov 5

Problem 1

-Current Stock Price 165

-Strike Price 165

-Current Date 03/13/2022

-Options Expiration Date 04/15/2022

-Risk Free Rate of 0.25%

-Continuously Compounding Coupon of 0.53%

Implement the close form greeks for GBSM. Implement a finite difference derivative calculation. Compare the values between two methods for both a call and a put.

Implement the binomial tree valuation for American options with and without discrete dividends.

Assume the stock above: -Pays dividend on 4/11/2022 of 0.88

Calculate the value of the call and the put. Calculate the Greeks of each.

What is the sensitivity of the put and call to a change in the dividend amount?

1.1 Implement the close form greeks

```
def d1(S,X,b,sigma,T):
    return (np.log(S/X)+(b+sigma**2/2)*T)/sigma/T**0.5
def delta_call(S,b,r,T,X,sigma):
    return np.exp((b-r)*T)*norm.cdf(d1(S,X,b,sigma,T))
def delta_put(S,b,r,T,X,sigma):
    return np.exp((b-r)*T)*(norm.cdf(d1(S,X,b,sigma,T))-1)
```

```
def gamma(S,b,r,T,X,sigma):
    return norm.pdf(d1(S,X,b,sigma,T))*np.exp((b-r)*T)/(S*sigma*np.sqrt(T))
```

```
def vega(S,b,r,T,X,sigma):
    return S*np.exp((b-r)*T)*norm.pdf(d1(S,X,b,sigma,T))*np.sqrt(T)
```

```
def d2(S,X,b,sigma,T):
    return d1(S,X,b,sigma,T)-sigma*T**0.5
def theta_call(S,b,r,T,X,sigma):
    return -S*np.exp((b-r)*T)*norm.pdf(d1(S,X,b,sigma,T))*sigma/2/np.sqrt(T)-
def theta_put(S,b,r,T,X,sigma):
    return -S*np.exp((b-r)*T)*norm.pdf(d1(S,X,b,sigma,T))*sigma/2/np.sqrt(T)+
```

```
def rho_call(S,b,r,T,X,sigma):
    return T*X*np.exp(-r*T)*norm.cdf(d2(S,X,b,sigma,T))
def rho_put(S,b,r,T,X,sigma):
    return -T*X*np.exp(-r*T)*norm.cdf(-d2(S,X,b,sigma,T))
```

```
def carry_rho_call(S,b,r,T,X,sigma):
    return T*S*np.exp((b-r)*T)*norm.cdf(d1(S,X,b,sigma,T))
def carry_rho_put(S,b,r,T,X,sigma):
    return -T*S*np.exp((b-r)*T)*norm.cdf(-d1(S,X,b,sigma,T))
```

1.2 Implement a finite difference derivative calculation

```
def bs_call(S,b,r,T,X,sigma):
    return S*np.exp((b-r)*T)*norm.cdf(d1(S,X,b,sigma,T))-X*np.exp(-r*T)*norm.cdf(d2(S,X,b,sigma,T))
def bs_put(S,b,r,T,X,sigma):
    return X*np.exp(-r*T)*norm.cdf(-d2(S,X,b,sigma,T))-S*np.exp((b-r)*T)*norm.cdf(-d1(S,X,b,sigma,T))
```

```
def fd_delta_call(S,b,r,T,X,sigma, d = 0.001*S):
    return (bs_call(S+d,b,r,T,X,sigma)-bs_call(S-d,b,r,T,X,sigma))/2/d
def fd_delta_put(S,b,r,T,X,sigma, d = 0.001*S):
    return (bs_put(S+d,b,r,T,X,sigma)-bs_put(S-d,b,r,T,X,sigma))/2/d
```

```
def fd_gamma(S,b,r,T,X,sigma, d = 0.001*S):
    return (bs_call(S+d,b,r,T,X,sigma)+bs_call(S-d,b,r,T,X,sigma)-2*bs_call(S,b,r,T,X,sigma))/d/d
```

```
def fd_vega(S,b,r,T,X,sigma, d = 0.001*sigma):
    return (bs_call(S,b,r,T,X,sigma+d)-bs_call(S,b,r,T,X,sigma-d))/2/d
```

```
def fd_theta_call(S,b,r,T,X,sigma, d = 0.001*T):
    return -(bs_call(S,b,r,T+d,X,sigma)-bs_call(S,b,r,T-d,X,sigma))/2/d
def fd_theta_put(S,b,r,T,X,sigma, d = 0.001*T):
    return -(bs_put(S,b,r,T+d,X,sigma)-bs_put(S,b,r,T-d,X,sigma))/2/d
```

```
def fd_rho_call(S,b,r,T,X,sigma, d = 0.001*r):
    return (bs_call(S,b+d,r+d,T,X,sigma)-bs_call(S,b-d,r-d,T,X,sigma))/2/d
def fd_rho_put(S,b,r,T,X,sigma, d = 0.001*r):
    return (bs_put(S,b+d,r+d,T,X,sigma)-bs_put(S,b-d,r-d,T,X,sigma))/2/d
```

```
def fd_carry_rho_call(S,b,r,T,X,sigma, d = 0.001*q):
    return (bs_call(S,b+d,r,T,X,sigma)-bs_call(S,b-d,r,T,X,sigma))/2/d
def fd_carry_rho_put(S,b,r,T,X,sigma, d = 0.001*q):
    return (bs_put(S,b+d,r,T,X,sigma)-bs_put(S,b-d,r,T,X,sigma))/2/d
```

1.3 Compare Values

```
delta call closed form vs delta call finite difference: 0.510070560620057 0.5100689809347269
delta put closed form vs delta put finite difference: -0.48945037608523323 -0.4894519557705156
gamma closed form vs gamma finite difference: 0.04017281658573558 0.0401719025425916
vega closed form vs vega finite difference: 19.776582323857255 19.776582320680802
theta call closed form vs theta call finite difference: -21.62860677878208 -21.62860951656207
theta put closed form vs theta put finite difference: -22.090281063696036 -22.09028380148204
rho call closed form vs rho call finite difference: 7.253304276901479 7.253304278265204
rho put closed form vs rho put finite difference: -7.661132489946645 -7.6611324942632555
carry rho call closed form vs carry rho call finite difference: 7.609134801578659 7.609134800484856
carry rho put closed form vs carry rho put finite difference: -7.301526843244096 -7.301526844438534
```

1.4 Implement the binomial tree valuation for American options with and without discrete dividends.

```
def bt_american_continous(is_call, S,X,T,r,b,sigma, N=200):
    dt = T/N
    u = np.exp(sigma*np.sqrt(dt))
    d = 1/u
    pu = (np.exp(b*dt)-d)/(u-d)
    pd = 1-pu
    df = np.exp(-r*dt)
    z = is_call
    def nNodeFunc(n):
        return (n+2)*(n+1)//2
    def idxFunc(i,j):
        return nNodeFunc(j-1)+i
    nNodes = nNodeFunc(N)
    optionValues = np.empty(nNodes, dtype = float)

    for j in range(N, -1, -1):
        for i in range(j, -1, -1):
            idx = idxFunc(i,j)
            price = S*u**i*d**(j-i)
            optionValues[idx] = max(0,z*(price-X))
            if j < N:
                optionValues[idx] = max(optionValues[idx], df*(pu*optionValues[idxFunc(i+1,j+1)] + pd*optionValues[
    return optionValues[0]
```

1.5 Calculate the value of the call and the put. Calculate the Greeks of each. Calculate the sensitivity

the value of Call with dividend is: 3.817994047321665

the value of Put with dividend is: 4.442979957979094

```
delta call: 0.5119851469040851
delta put: -0.5173627304436692
gamma call: 0.040424651516273356
gamma put: 0.040424651516273356
vega call: 19.566465699959455
vega put: 19.566465699959455
theta call: -21.448080912971747
theta put: -22.075813821093327
rho call: -0.1105640272669461З
rho put: -0.8066777397175428
sensitivity to dividend amount call -0.0918969874209989
sensitivity to dividend amount put 0.5385491011947438
```

2. Problem 2

Using the options portfolio From Problem3 last week(named problem2.csv in this week's repo) and assuming:

- American Options
- Current Date 02/25/2022
- Current AAPL price is 164.85
- Risk Free Rate of 0.25%
- Dividend Payment of 1.00 on 3/15/2022

Using DailyReturn.csv. 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.

Calculate VaR and ES using Delta-Normal.

Present all VaR and ES values as loss, not percentages.

Compare these results to last week's results.

2.1 Add the implied vol using binomial tree with dividend

We are using binomial tree here to calculate implied volatility because there is a dividend and BSM model itself does not capture that

	Portfolio	Type	Underlying	Holding	OptionType	ExpirationDate	Strike	CurrentPrice	Implied Volatility
0	Straddle	Option	AAPL	1	Call	3/18/2022	165.0	4.50	0.299747
1	Straddle	Option	AAPL	1	Put	3/18/2022	165.0	4.40	0.238282
2	SynLong	Option	AAPL	1	Call	3/18/2022	165.0	4.50	0.299747
3	SynLong	Option	AAPL	-1	Put	3/18/2022	165.0	4.40	0.238282
4	CallSpread	Option	AAPL	1	Call	3/18/2022	165.0	4.50	0.299747
5	CallSpread	Option	AAPL	-1	Call	3/18/2022	175.0	0.72	0.245446
6	PutSpread	Option	AAPL	1	Put	3/18/2022	165.0	4.40	0.238282
7	PutSpread	Option	AAPL	-1	Put	3/18/2022	155.0	1.60	0.310073
8	Stock	Stock	AAPL	1	NaN	NaN	NaN	164.85	NaN
9	Call	Option	AAPL	1	Call	3/18/2022	165.0	4.50	0.299747
10	Put	Option	AAPL	1	Put	3/18/2022	165.0	4.40	0.238282
11	CoveredCall	Stock	AAPL	1	NaN	NaN	NaN	164.85	NaN
12	CoveredCall	Option	AAPL	-1	Call	3/18/2022	165.0	4.50	0.299747
13	ProtectedPut	Stock	AAPL	1	NaN	NaN	NaN	164.85	NaN
14	ProtectedPut	Option	AAPL	1	Put	3/18/2022	165.0	4.40	0.238282

2.2 Calculate the mean, VaR and ES using Monte Carlo

Use Monte Carlo Simulation to calculate mean, VaR and ES with our new implied volatility

4]:

	Mean	VaR	ES
Straddle	0.223322	2.735163	2.748453
SynLong	0.172584	13.644135	17.350743
CallSpread	-0.193031	3.615680	3.707631
PutSpread	0.247894	2.748180	2.776602
Stock	-0.061948	13.522990	17.162870
Call	0.197953	4.335143	4.427485
Put	0.025369	4.338751	4.372083
CoveredCall	-0.259901	9.187848	12.735385
ProtectedPut	-0.036579	4.213998	4.239612

2.3 Add delta column

Add the delta for each portfolio by using the formula we learned in week 4. Stock has delta of 1.

	Portfolio	Type	Underlying	Holding	OptionType	ExpirationDate	Strike	CurrentPrice	Implied Volatility	delta
0	Straddle	Option	AAPL	1	Call	3/18/2022	165.0	4.50	0.299747	0.488306
1	Straddle	Option	AAPL	1	Put	3/18/2022	165.0	4.40	0.238282	-0.517168
2	SynLong	Option	AAPL	1	Call	3/18/2022	165.0	4.50	0.299747	0.488306
3	SynLong	Option	AAPL	-1	Put	3/18/2022	165.0	4.40	0.238282	-0.517168
4	CallSpread	Option	AAPL	1	Call	3/18/2022	165.0	4.50	0.299747	0.488306
5	CallSpread	Option	AAPL	-1	Call	3/18/2022	175.0	0.72	0.245446	0.149071
6	PutSpread	Option	AAPL	1	Put	3/18/2022	165.0	4.40	0.238282	-0.517168
7	PutSpread	Option	AAPL	-1	Put	3/18/2022	155.0	1.60	0.310073	-0.207275
8	Stock	Stock	AAPL	1	NaN	NaN	NaN	164.85	NaN	1.000000
9	Call	Option	AAPL	1	Call	3/18/2022	165.0	4.50	0.299747	0.488306
10	Put	Option	AAPL	1	Put	3/18/2022	165.0	4.40	0.238282	-0.517168
11	CoveredCall	Stock	AAPL	1	NaN	NaN	NaN	164.85	NaN	1.000000
12	CoveredCall	Option	AAPL	-1	Call	3/18/2022	165.0	4.50	0.299747	0.488306
13	ProtectedPut	Stock	AAPL	1	NaN	NaN	NaN	164.85	NaN	1.000000
14	ProtectedPut	Option	AAPL	1	Put	3/18/2022	165.0	4.40	0.238282	-0.517168

2.4 Calculate VaR and ES using Delta Normal

	VaR	ES
Straddle	0.393763	0.493794
SynLong	13.717957	17.202871
CallSpread	4.628281	5.804051
PutSpread	4.227954	5.302025
Stock	13.643272	17.109214
Call	6.662097	8.354539
Put	7.055860	8.848333
CoveredCall	6.981175	8.754675
ProtectedPut	6.587412	8.260881

2.5 Conclusion

Compared to last week's result, the values calculated by Monte Carlo are very close with minor difference in mean values. The VaR and ES calculated by delta normal have greater values in many strategies such as ProtectedPut, Call, Put. It has lower values in Straddle and CoveredCall. It's surprising to see Straddle Strategy has VaR and ES smaller than 1.

3. Problem 3

Use the Fama French 3 factor return time series(F-F_Research_Data_Factors_daily.CSV) as well as the Carhart Momentum time series(F-F_momentum_Factor_daily.CSV) to fit a 4 factor model to the following stocks.

Fama stores values as percentages, you will need to divide by 100(or multiply the stock returns by 100) to get like units.

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

Construct an annual covariance matrix for the 10 stocks.

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

3.1 Formula

$$r_s - r_{rf} = \alpha + \beta_1 * (r_{mkt} - r_{rf}) + \beta_2 * SMB + \beta_3 * HML + \beta_4 * UMD$$

We use the formula learned in class to create a model.

3.2 mean and covariance

1. fit the linear model
2. Use the data from recent 10 years to calculate the daily returns
3. Calculate the annual return by taking the average of annual returns in 10 years
4. Calculate the covariance using the data of 10 years annual returns

```
AAPL 1.5454551657130924
FB 0.233548147458683
UNH 0.5248855870608644
MA 0.13470309841474062
MSFT 0.449706709979263
NVDA 4.352215996863421
HD 0.250051481790147
PFE 2.0834606203164605
AMZN 0.10312548020738274
BRK-B 0.227364505360541
PG 0.4180087108855499
XOM 0.24836803174157182
TSLA 5.5693243322882715
JPM -0.45276930284894573
V -0.29594769275583255
DIS -0.3411675887954086
GOOGL 0.13853850809514495
JNJ -0.008845584573422337
BAC -0.2461782542730492
CSCO 0.5010356246775759
```

	AAPL	FB	UNH	MA	MSFT	NVDA	HD	PFE	AMZN	BRK-B	PG	XOM	TSLA	JPI
AAPL	0.070909	0.020138	0.020219	-0.024050	0.052252	0.419699	0.025480	0.005241	0.020083	-0.013200	-0.005843	-0.015845	0.392962	-0.00522
FB	0.020138	0.036112	0.016064	0.046652	0.005924	0.028632	-0.004687	-0.129722	0.024455	0.015685	0.008025	0.026969	0.049766	0.00823
UNH	0.020219	0.016064	0.015158	0.021267	0.015769	0.110604	0.007311	-0.037133	0.013541	0.011758	0.010009	0.016181	0.035864	0.00539
MA	-0.024050	0.046652	0.021267	0.129122	-0.025606	-0.220263	-0.019030	-0.206692	0.026348	0.058830	0.036178	0.086963	-0.311556	0.02783
MSFT	0.052252	0.005924	0.015769	-0.025606	0.044009	0.356881	0.025297	0.049707	0.010137	-0.009377	-0.001516	-0.013875	0.273108	-0.00423
NVDA	0.419699	0.028632	0.110604	-0.220263	0.356881	3.046392	0.216914	0.526265	0.059237	-0.083944	-0.021351	-0.114300	2.406311	-0.03555
HD	0.025480	-0.004687	0.007311	-0.019030	0.025297	0.216914	0.018294	0.061364	-0.000196	-0.003704	0.001372	-0.006715	0.136169	-0.00177
PFE	0.005241	-0.129722	-0.037133	-0.206692	0.049707	0.526265	0.061364	0.616180	-0.079726	-0.073183	-0.032208	-0.120634	0.271660	-0.03721
AMZN	0.020083	0.024455	0.013541	0.026348	0.010137	0.059237	-0.000196	-0.079726	0.018345	0.008699	0.005358	0.014341	0.053473	0.00439
BRK-B	-0.013200	0.015685	0.011758	0.058830	-0.009377	-0.083944	-0.003704	-0.073183	0.008699	0.032626	0.022467	0.045537	-0.182069	0.01504
PG	-0.005843	0.008025	0.010009	0.036178	-0.001516	-0.021351	0.001372	-0.032208	0.005358	0.022467	0.016969	0.029871	-0.122333	0.01005
XOM	-0.015845	0.026969	0.016181	0.086963	-0.013875	-0.114300	-0.006715	-0.120634	0.014341	0.045537	0.029871	0.065842	-0.220260	0.02152
TSLA	0.392962	0.049766	0.035864	-0.311556	0.273108	2.406311	0.136169	0.271660	0.053473	-0.182069	-0.122333	-0.220260	2.889486	-0.07419
JPM	-0.005220	0.008231	0.005397	0.027830	-0.004235	-0.035550	-0.001773	-0.037218	0.004391	0.015045	0.010058	0.021523	-0.074199	0.00707
V	-0.005318	0.020668	0.008993	0.049220	-0.008194	-0.072641	-0.007121	-0.089543	0.011955	0.021903	0.013047	0.033151	-0.093871	0.01057
DIS	-0.003014	0.017835	0.006651	0.036819	-0.006838	-0.065209	-0.007263	-0.078390	0.010739	0.014781	0.008158	0.022924	-0.063347	0.00719
GOOGL	0.031817	0.010716	0.015572	0.003055	0.027046	0.209574	0.015200	0.002864	0.011346	0.004441	0.006533	0.005319	0.118967	0.00198
JNJ	-0.005304	0.007765	0.007399	0.028708	-0.002765	-0.034445	-0.001190	-0.033738	0.005393	0.016223	0.011938	0.021591	-0.099955	0.00719
BAC	-0.012898	0.016601	0.009824	0.056933	-0.010806	-0.089919	-0.005325	-0.077704	0.008624	0.029819	0.019547	0.042834	-0.159157	0.01400
CSCO	0.009936	0.019910	0.015059	0.039153	0.006590	0.037467	0.002123	-0.065906	0.014368	0.020190	0.014584	0.028770	-0.035355	0.00943

3.3 Super efficient portfolio

```
AAPL has weight of 40.17 %
FB has weight of 0.0 %
UNH has weight of 0.0 %
MA has weight of 26.47 %
MSFT has weight of 0.0 %
NVDA has weight of 0.0 %
HD has weight of 0.0 %
PFE has weight of 14.3 %
AMZN has weight of 0.0 %
BRK-B has weight of 0.0 %
PG has weight of 19.07 %
XOM has weight of 0.0 %
TSLA has weight of 0.0 %
JPM has weight of 0.0 %
V has weight of 0.0 %
DIS has weight of 0.0 %
GOOGL has weight of 0.0 %
JNJ has weight of 0.0 %
BAC has weight of 0.0 %
CSCO has weight of 0.0 %
The Sharpe ratio is 8.553140176049073
```

A super efficient portfolio should have the highest sharpe ratio among all the possible portfolios. Therefore, we can get the super efficient portfolio by maximizing the sharpe function.

The super efficient portfolio has a sharpe ratio of 8.55 and it contains AAPL, MA, PFE and PG