Problem 1

Assume you a call and a put option with the following

- Current Stock Price $165
- Current Date 03/03/2023
- Options Expiration Date 03/17/2023
- Risk Free Rate of 4.25%
- Continuously Compounding Coupon of 0.53%

Calculate the time to maturity using calendar days (not trading days).

For a range of implied volatilities between 10% and 80%, plot the value of the call and the put.

Discuss these graphs. How does the supply and demand affect the implied volatility?

In this question, I first calculated the time to maturity through dividing days to maturity by the number of calendar days in a year:

```
# calculate the time to maturity
current_day = date(2023,3,3)
expire_day = date(2023,3,17)
days_to_mat = (expire_day - current_day).days
ttm = days_to_mat / 365
print("Time to Maturity is {:.4f}".format(ttm))
```

```
Time to Maturity is 0.0384
```

Then, I define a function, like the one used in class, to calculate options' values with Black Scholes Formula. With the defined function, I set up the problem assuming that strikes for both options are the same, and called the function to compute put options' values and call options' values under different implied volatility, ranging from 0.1 to 0.8. Since both options' strikes are the same, I am able to check put call parity here with derived results with an absolute tolerance of 0.1:

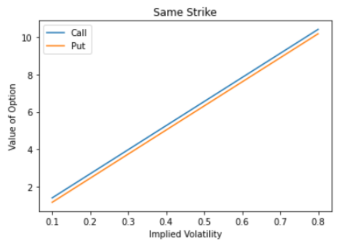
```
# checking put call parity
result = True
for i in range(len(gbsm_call_values)):
    if isclose(gbsm_call_values[i] + strike * exp(-rf*ttm), gbsm_put_values[i] + underlying, abs_tol = 0.1) == False:
        result = False
print(result)
```

```
True
```

As shown above, the put call parity holds in this case.

Eventually, I plotted derived values of both options with respect to different implied volatility:

```
# ploting values of put and call options with impied vols ranging from 0.1 to 0.8
plt.figure()
plt.plot(ivol, gbsm_call_values, label="Call")
plt.plot(ivol, gbsm_put_values, label="Put")
plt.xlabel("Implied Volatility")
plt.ylabel("Value of Option")
plt.legend()
plt.title("Same Strike")
plt.show()
```

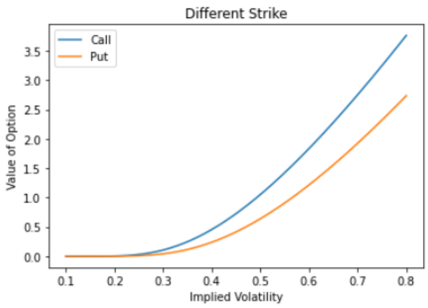


As we may notice from the plot above, values of options have linear relationships with implied volatility when strikes are the same; therefore, I would like to see what will happen when strikes are different:

```
# check what will happen if strike prices become difference for puts and calls
gbsm_call_values_diff = np.zeros(len(ivol))
gbsm_put_values_diff = np.zeros(len(ivol))

for i in range(len(ivol)):
    gbsm_call_values_diff[i] = gbsm(underlying, strike+20, ttm, rf, b, ivol[i], type="call")
    gbsm_put_values_diff[i] = gbsm(underlying, strike-20, ttm, rf, b, ivol[i], type="put")

# ploting values of put and call options with impied vols ranging from 0.1 to 0.8
plt.figure()
plt.plot(ivol, gbsm_call_values_diff, label="Call")
plt.plot(ivol, gbsm_put_values_diff, label="Put")
plt.xlabel("Implied Volatility")
plt.ylabel("Value of Option")
plt.legend()
plt.title("Different Strike")
plt.show()
```



As shown in two graphs above, when strikes of call options and put options of the same underlying are the same, their values seem to have linear relationships with implied volatility, and call options typically have slightly higher values than put options; when strikes are different, keeping all other conditions the same, their values are having nonlinear relationships with implied volatility, and options with higher strikes typically have higher values than the other type of options (in my case, call options have higher strikes). Across two graphs, as implied volatility increases, values of options increase. Upon the relationship between supply and demand and implied volatility, as demand increases and supply decreases, expectations of the market increase, and implied volatility correspondingly increases, leading to more premiums in options' values. Conversely, when demand decreases and supply increases, expectations of the market drop, and implied volatility correspondingly decreases, resulting in less premiums/more discounts in options' values.

Market	Expectation	Implied Vol.	Values
D ↑ S ↓	↑	↑	↑
D ↓ S ↑	↓	↓	↓

(Note: " ↑ " stands for "increase", and " ↓ " stands for "decrease")

Problem 2

Use the options found in AAPL_Options.csv

- Current AAPL price is 151.03
- Current Date, Risk Free Rate and Dividend Rate are the same as problem #1.

Calculate the implied volatility for each option.

Plot the implied volatility vs the strike price for Puts and Calls. Discuss the shape of these graphs. What market dynamics could make these graphs?

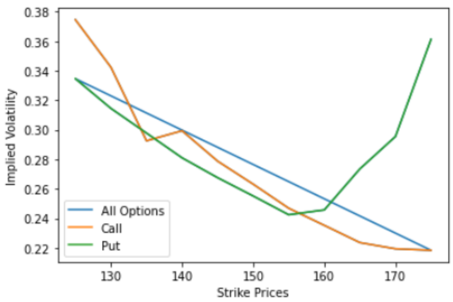
There are bonus points available on this question based on your discussion. Take some time to research if needed.

In this question, I copied the defined function from problem 1 for calculation of values of options. Then, I set up conditions, such as the underlying price, and started calculating implied volatilities through finding roots of defined functions of differences of AAPL options' values and prices, results are:

	Stock	Expiration	Type	Strike	Last Price	ivol
0	AAPL	4/21/2023	Call	125	27.300	0.374597
1	AAPL	4/21/2023	Call	130	22.575	0.342351
2	AAPL	4/21/2023	Call	135	17.750	0.292522
3	AAPL	4/21/2023	Call	140	13.850	0.299358
4	AAPL	4/21/2023	Call	145	9.975	0.278743
5	AAPL	4/21/2023	Call	150	6.700	0.263141
6	AAPL	4/21/2023	Call	155	4.050	0.246828
7	AAPL	4/21/2023	Call	160	2.210	0.235242
8	AAPL	4/21/2023	Call	165	1.035	0.223567
9	AAPL	4/21/2023	Call	170	0.460	0.219339
10	AAPL	4/21/2023	Call	175	0.195	0.218342
11	AAPL	4/21/2023	Put	125	0.405	0.334615
12	AAPL	4/21/2023	Put	130	0.665	0.314473
13	AAPL	4/21/2023	Put	135	1.120	0.297772
14	AAPL	4/21/2023	Put	140	1.840	0.280994
15	AAPL	4/21/2023	Put	145	3.010	0.267532
16	AAPL	4/21/2023	Put	150	4.750	0.255134
17	AAPL	4/21/2023	Put	155	7.150	0.242417
18	AAPL	4/21/2023	Put	160	10.575	0.245700
19	AAPL	4/21/2023	Put	165	14.925	0.273493
20	AAPL	4/21/2023	Put	170	19.425	0.295414
21	AAPL	4/21/2023	Put	175	24.625	0.361243

Then, I started plotting implied volatilities with respect to strike prices:

```python
# plot graphs of implied volatility with respect to strike prices
plt.figure()
plt.plot(strike, implied_vol, label="All Options")
plt.plot(call.Strike, call.ivol, label="Call")
plt.plot(put.Strike, put.ivol, label="Put")
plt.xlabel("Strike Prices")
plt.ylabel("Implied Volatility")
plt.legend()
plt.show()
```



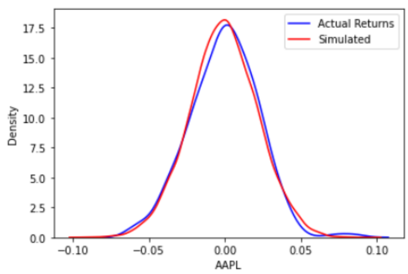
As we may notice from the graph, for call options of Apple stock and the general pattern of both options, as strike prices increase, especially after they go beyond underlying prices, implied volatilities of options decrease. However, put options' implied volatilities are in an approximately hyperbolic relationship with strike prices, which is abnormal. The pattern of implied volatilities with respect to strike prices of put options of Apple stock in this case is called volatility smile, indicating that out-of-the-money options have higher implied volatility than at-the-money or in-the-money options; in other words, the implied volatility of options tends to increase as the strike price moves further away from the current price of the underlying asset.

This phenomenon could be a result of several factors – 1. investors are uncertain about the future direction of the asset price, they will demand a higher premium to compensate for the increased risk, and implied volatility would increase correspondingly for options with strike prices further away from the current price; 2. when there is a surge in demand for options with a specific strike price, the price of those options will increase, causing their implied volatility to rise; 3. Unexpected market events or news could increase the implied volatility.

There are also several essential implications of volatility smile: 1. traders and investors will need to pay a higher premium to purchase out-of-the-money options, reflecting the increased perceived risk; 2. when investors are attempting to hedge risks of options, they need to consider different volatilities across different strike prices; 3. volatility smile could be seen as an indicator of market sentiment – when deviations from expected volatility smile exist, investors should consider adjusting their positions correspondingly; 4. Investors should choose to buy or sell options based on their perceived value relative to the implied volatility.

What's more, volatility smile is often seen as evidence of a deficiency in the Black-Scholes model. The Black-Scholes model assumes that the volatility of the underlying asset is constant across all strike prices and expiration dates, and that the distribution of returns is log-normal. However, the existence of a volatility smile indicates that this assumption is not accurate, and that the market's perceived volatility of the underlying asset can vary depending on the strike price and expiration date of the option.

Through calculating Apple stock's log returns and plotting the distribution with a log normal distribution, as below:



A nuanced difference could be identified from the graph (the actual returns have another protrusion between 0.05 and 0.1), proving that the assumption behind the Black Scholes model is broken in the real market.

Problem 3

Use the portfolios found in problem3.csv

- Current AAPL price is 151.03
- Current Date, Risk Free Rate and Dividend Rate are the same as problem #1.

For each of the portfolios, graph the portfolio value over a range of underlying values. Plot the portfolio values and discuss the shapes. Bonus points available for tying these graphs to other topics discussed in the lecture.

Using DailyPrices.csv. Calculate the log returns of AAPL. Demean the series so there is 0 mean. Fit an AR(1) model to AAPL returns. Simulate AAPL returns 10 days ahead and apply those returns to the current AAPL price (above). Calculate Mean, VaR and ES. Discuss.

Hints:

- you will need to calculate the implied volatility - might not be the same as #2
- you need to take into account the change in dates for option valuations. You are simulating forward in time and options valuations are a function of time
- Calculate the PL from the current portfolio value using Current Date

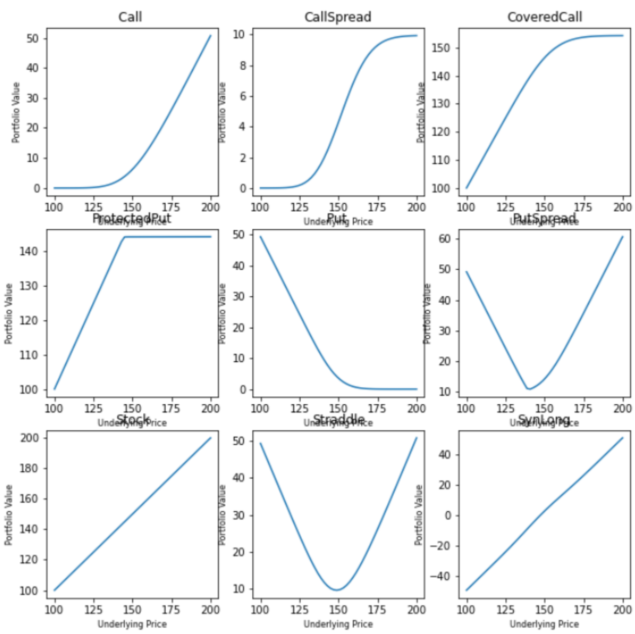
Part 1:

In this question, I first defined a new function that could be used to calculate simulated portfolio values under different underlying prices. Then, I derived an array of implied volatilities corresponding to each option (derived implied volatilities of stocks are incorrect), and applied the defined function to derive portfolio values from various underlying prices. Part of the resulting matrix is screenshot below:

```
simulated_vals
```

Portfolio	0	1	2	3	4	5	6	7	8	9	...	40	41	42	43
Call	0.000059	0.000148	0.000352	0.000791	0.001693	0.003457	0.006753	0.012651	0.022781	0.039516	...	32.493216	34.496725	36.509130	38.528389
CallSpread	0.000059	0.000148	0.000351	0.000788	0.001684	0.003433	0.006689	0.012489	0.022393	0.038636	...	9.678299	9.746264	9.798402	9.837885
CoveredCall	99.999998	102.040811	104.081616	106.122404	108.163148	110.203797	112.244240	114.284269	116.323509	118.361311	...	154.036169	154.094285	154.138186	154.171121
ProtectedPut	100.000000	102.040816	104.081633	106.122449	108.163265	110.204082	112.244898	114.285714	116.326531	118.367347	...	144.304248	144.305700	144.307151	144.308603
Put	49.217740	47.178375	45.139011	43.099646	41.060282	39.020919	36.981562	34.942226	32.902952	30.863855	...	0.003888	0.002101	0.001114	0.000580
PutSpread	49.217740	47.178375	45.139011	43.099646	41.060282	39.020919	36.981562	34.942226	32.902952	30.863855	...	42.303847	44.341425	46.379802	48.418633
Stock	100.000000	102.040816	104.081633	106.122449	108.163265	110.204082	112.244898	114.285714	116.326531	118.367347	...	181.632653	183.673469	185.714286	187.755102
Straddle	49.217799	47.178524	45.139362	43.100437	41.061974	39.024376	36.988315	34.954877	32.925733	30.903372	...	32.497104	34.498826	36.510244	38.528969
SynLong	-49.217681	-47.178227	-45.138659	-43.098855	-41.058589	-39.017462	-36.974809	-34.929575	-32.880171	-30.824339	...	32.489328	34.494624	36.508016	38.527809

Plotting resulting simulated values with respect to underlying prices:



As shown above, for portfolio Call and portfolio Put, only basic options exist in these portfolios, and investors would make money as underlying prices go up or down beyond strike prices. For portfolio CallSpread and portfolio PutSpread, same type of options with different strike prices are shorted in addition to basic options in these portfolios, but patterns of returns are quite different. Portfolio PutSpread would generate increase portfolio values regardless of which direction underlying prices are moving, while portfolio CallSpread has a pattern of portfolio values that follows an "S" shape – portfolio values would increase dramatically when underlying prices are close to the current price. For portfolio CoveredCall and portfolio PortectedPut, a combination of stock and option is employed. Patterns of portfolio values are similar to the ones associated with basic options. For portfolio SynLong and portfolio Stock, their patterns of portfolio values are similar, but portfolio SynLong is attempting to maximize returns with an exposure of extremely high risk. For portfolio Straddle, the relationship between portfolio value and underlying price is basically hyperbolic, and it involves both options, resulting in a phenomenon that portfolio values would increase regardless of whether underlying price decreases or increases.

Part 2:

Then, I calculated log returns of Apple stock with price data from the DailyPrices.csv. I demeaned those log returns, and fit them with an AR(1) model. With the fitted model, I am able to simulate log returns, and correspondingly prices, of Apple Stock 10 days ahead for 10000 times, and simulated results are below:

```
ar1_sim_p
```

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
0	155.230208	156.521007	155.089551	158.260523	161.457132	158.824996	155.942800	158.124687	153.938777	157.608823
1	152.669172	151.231875	148.657619	145.866314	143.389117	140.774847	135.167819	133.714093	135.606869	139.791292
2	151.959508	150.502639	151.760672	146.687039	148.295926	148.087458	149.235930	152.882734	156.874725	153.480372
3	149.597711	149.088732	144.542077	147.650773	146.833522	146.808196	148.163201	150.635714	148.896236	152.409802
4	157.921715	156.240770	154.899554	155.035529	158.185890	156.810806	156.373084	162.473897	169.274052	166.413661
...	...	...	...	...	...	...	...	...	...	...
9995	150.596075	158.243371	159.644230	162.695672	154.428185	151.341948	149.776770	145.587643	145.222855	143.454305
9996	151.626925	153.556948	153.053261	152.140186	153.205905	153.321604	151.550599	145.304550	149.494375	147.374270
9997	148.690956	144.991274	153.034144	153.212927	155.428046	158.789773	158.565972	157.462568	153.236489	155.782473
9998	150.012856	144.453515	143.501883	141.738701	142.476620	143.846694	146.050930	152.352463	155.224101	157.052329
9999	148.826289	144.726005	144.234901	146.174335	150.218932	149.318461	155.112153	158.786181	157.200521	155.109345

With these simulated prices, I derived simulated portfolio values on the 10th day ahead for each portfolio respectively (part of the resulting matrix) (note: here, I take time into account while calculating portfolio values):

```
ar1_sim_10port
```

	0	1	2	3	4	5	6	7	8	9	...	9990	9991	9992	9993
Portfolio															
Call	10.422555	1.628822	7.567097	6.901377	17.723044	3.820564	5.794547	7.155993	3.737720	2.242324	...	4.866883	1.282386	7.000784	0.928187
CallSpread	6.392016	1.421322	5.150778	4.816629	8.437977	3.012782	4.219501	4.946578	2.957799	1.897447	...	3.676295	1.140304	4.867684	0.842873
CoveredCall	150.817326	139.167877	148.968779	148.400861	153.164080	144.584861	147.294387	148.625754	144.441588	141.148613	...	146.166064	137.738717	148.489850	135.884013
ProtectedPut	144.432263	139.791292	144.429925	144.429319	144.437248	144.425949	144.428236	144.429555	144.425837	142.090189	...	144.427233	138.196744	144.429412	136.186053
Put	0.862550	10.135007	1.841201	2.197449	0.113519	5.099249	2.955277	2.053553	5.221847	8.217941	...	3.803911	11.548990	2.140078	13.406526
PutSpread	19.016456	10.514246	15.868994	15.155277	27.067278	12.108093	14.001175	15.426823	12.035327	10.861999	...	13.078332	11.549014	15.261067	13.406526
Stock	157.608823	139.791292	153.480372	152.409802	166.413661	146.457447	150.496789	152.825479	146.261973	142.090189	...	148.724308	138.196744	152.573055	136.186053
Straddle	11.285106	11.763829	9.408299	9.098825	17.836562	8.919814	8.749823	9.209546	8.959567	10.460265	...	8.670794	12.831376	9.140862	14.334714
SynLong	9.560005	-8.506185	5.725896	4.703928	17.609525	-1.278685	2.839270	5.102440	-1.484127	-5.975617	...	1.062972	-10.266603	4.860707	-12.478339

9 rows × 10000 columns

Eventually, I computed corresponding mean of simulated portfolio values, 5% VaR, and 5% ES for each portfolio, and expressed latter two in both numerical (in columns "VaR" and "ES") and percentage (in columns "VaR%" and "ES%") forms, the resulting matrix is:

```
resulting_mat
```

	Mean of Portfolio Value($)	Mean of Losses/Gains($)	VaR($)	ES($)	VaR(%)	ES(%)	Current Value (on 2023/3/3)
Call	7.750295	-0.950295	6.016988	6.360480	88.485122	93.536470	6.80
CallSpread	4.549168	4.460832	8.292276	8.599723	92.034140	95.446423	9.01
CoveredCall	146.364309	8.715691	20.135761	23.879030	12.984112	15.397879	155.08
ProtectedPut	142.965842	11.074158	18.853101	22.718202	12.239095	14.748249	154.04
Put	4.424179	0.425821	4.795894	4.829382	98.884420	99.574879	4.85
PutSpread	16.987650	-10.297650	-4.022981	-3.904941	-60.134244	-58.369819	6.69
Stock	151.468970	-0.438970	15.843101	19.708202	10.490036	13.049197	151.03
Straddle	12.174474	-0.524474	2.966320	2.977172	25.461971	25.555123	11.65
SynLong	3.326117	8.323883	25.220350	29.341559	216.483688	251.858878	11.65

Note that in this matrix (excluding column "Mean" and "Current Value (on 2023/3/3)"), positive values represent losses, and negative values represent gains, since I subtracted simulated portfolio values from portfolio values on current date before displaying results in the matrix.

From calculated 5% VaRs, we could notice that portfolio SynLong is the one with highest risk – on a 5% bad day, investors would at least lose 25.22 dollars, equally 216.48% of the original value. Followingly, portfolio CallSpread and portfolios with two basic options have roughly the same level of risk – investors would experience a minimum of 88.49% of loss on a 5% bad day. Then, portfolio Straddle, though generally generating a small increase in portfolio values (mean of losses/gains is -0.52 dollars; note that negative number indicates a gain), would lead investors to lose at least 25.46% of their money on a 5% bad day. Another three portfolios, CoveredCall, ProtectedPut, and Stock, have generally small 5% VaRs – investors would lose a minimum of 10% of their original values on the current date on a 5% bad day. Eventually, portfolio PutSpread is the one that has the highest mean of losses/gains, -10.30 dollars, and its 5% VaR is -60.13%, indicating that investors would typically gain from investing in PutSpread.

Overall, portfolios with pure stocks have a lower level of risk than portfolios with pure options at costs of returns. Portfolios with a mix of stocks and options have a level of risk between those with pure stocks and those with pure options. However, when investors would like to short options for higher expected returns, they would be exposed to much higher risks that may lead them to lose more than expected.