Student No: 3036383355

Name: Varun Aditya Balaji

FITE 7405 Assignment

Q1)

These values were observed using q1.py:

Call Option Price for S=100, K=100, T=1.0, sigma=0.3, r=0.03: 13.283308397880909

Put Option Price for S=100, K=100, T=1.0, sigma=0.3, r=0.03: 10.327861752731728

Call Option Price for S=100, K=110, T=1.0, sigma=0.3, r=0.03: 9.24002671364994

Put Option Price for S=100, K=110, T=1.0, sigma=0.3, r=0.03: 15.989035403985852

Call Option Price for S=100, K=100, T=1.5, sigma=0.3, r=0.03: 16.56103761888422

Put Option Price for S=100, K=100, T=1.5, sigma=0.3, r=0.03: 12.160785802194212

Call Option Price for S=100, K=100, T=1.0, sigma=0.4, r=0.03: 17.138735220515535

Put Option Price for S=100, K=100, T=1.0, sigma=0.4, r=0.03: 14.183288575366355

Call Option Price for S=100, K=100, T=1.0, sigma=0.3, r=0.05: 14.231254785985819

Put Option Price for S=100, K=100, T=1.0, sigma=0.3, r=0.05: 9.354197236057232

Call Option Price for S=110, K=100, T=1.0, sigma=0.3, r=0.03: 19.87301046965876

Put Option Price for S=110, K=100, T=1.0, sigma=0.3, r=0.03: 6.917563824509575

Trend Analysis

Option	Volatility (increases)	Maturity (increases)	Strike Price (increases)	Risk Free Rate (increases)
Call	**Increases**	**Increases**	**Decreases**	**Increases**
Put	**increases**	**Increases**	**Increases**	**Decreases**

Q2)

2.1

Q2. Given: $Cov(X,Y) = E(XY) - E(X)E(Y)$

$\rho(X,Y) = Cov(X,Y) / \sqrt{Var(X)\, Var(Y)}$

To prove: $Z = \rho X + \sqrt{1-\rho^2}\, Y \quad \rho \in [0,1]$ -> Given

$\rho(X',Z) = \rho.$

$\rho(X,Z) = Cov(X,Z) / \sqrt{Var(X)\, Var(Z)}$

$Cov(X,Z) = E(XZ) - E(X)E(Z) \quad \because E(X) = 0$ (std normal)

$= E(XZ)$

$E(XZ) = E(X^2\rho + XY\sqrt{1-\rho^2}) = E(X^2\rho) + E(XY\sqrt{1-\rho^2})$

$E(XZ) = \rho E(X^2) + \sqrt{1-\rho^2}\, E(XY)$

For standard normal variable X, Y

$E(X^2) = Var(X) + (E(X))^2 = 1$

$\therefore E(XY) = E(X)E(Y) \Rightarrow 0.$

$\therefore E(XZ) = \rho + 0 = \rho. \Rightarrow Cov(X,Z)$

Now to derive variance of Z:

$Var(Z) = Var(\rho X) + Var(\sqrt{1-\rho^2}\, Y) \Rightarrow \rho^2 Var(X) + (1-\rho^2) Var(Y)$

$\because Var(X) = Var(Y) = 1$ (std normal)

$Var Z = \rho^2 + 1 - \rho^2 = 1$

$\therefore \rho(X,Z) = \dfrac{\rho}{\sqrt{1.1}} = \rho$

2.2)

Please note for this question, seed used is: 10 (q2.py)

After running the code with 200 random samples of standard normal variables X and Y, we get the following results:

The correlation coefficient is: 0.4989406837688636

The difference between the correlation coefficient and the expected value is: -0.0010593162311364201

Mean of X: 0.07430692331668418

Variance of X: 0.9530511976376439

Mean of Y: 0.027842682934754306

Variance of Y: 0.8553175883347172

We can see the correlation coefficient is very similar to the value p. The small discrepancy comes from the fact that X and Y have small samples and hence cannot perfectly mimic a standard normal distribution.

If we take reference from **Central limit Theorem** and **law of large numbers**, if we generate more samples of X and Y, we should get closer to the theoretical p value.

Here is an observation with 200000 samples of X and Y, with seed 10:

The correlation coefficient is: 0.49947329965599535

The difference between the correlation coefficient and the expected value is: -0.0005267003440046492

Mean of X: -0.002445104430380625

Variance of X: 0.9980440929390706

Mean of Y: 0.002789587357785681

Variance of Y: 0.997899493855162

Q3)

1) Take a look at q31.py file:

The following values were used to test the implied volatility code implementation:

```
risk_free_rate = 0.03
spot_price = 2
strike_price = 2
time_to_maturity = 3
current_time = 0
volatility = 0.4
repo_rate = 0
```

The code produced the following output:

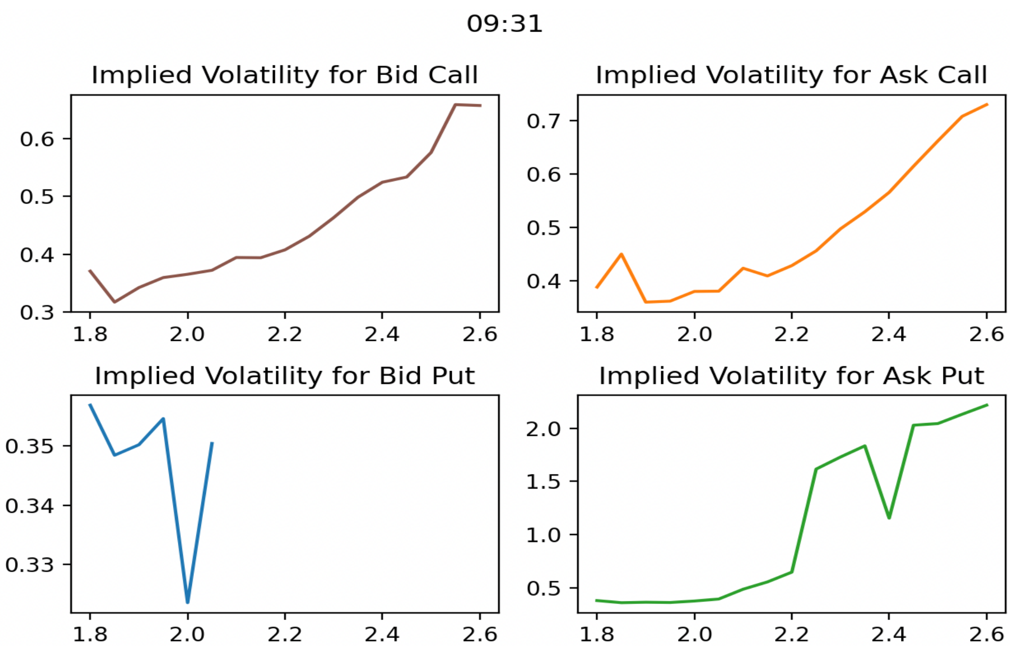
Actual Volatility is 0.4

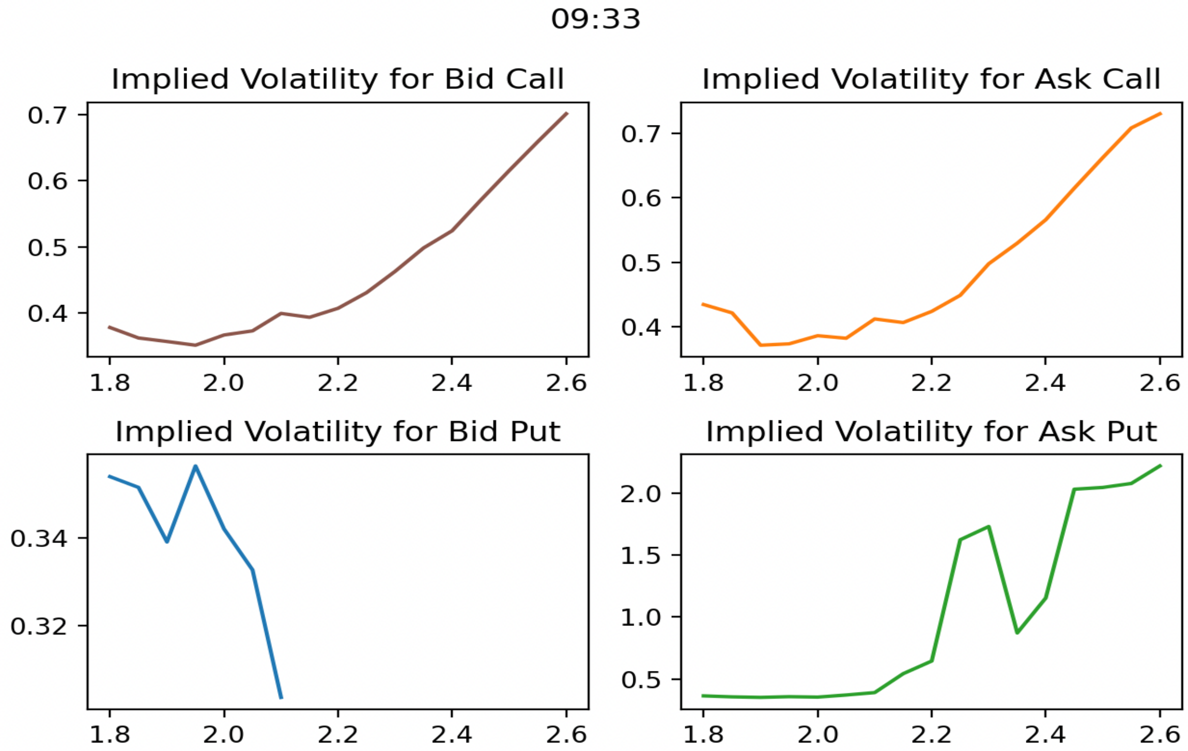
Implied Volatility from Calculation is 0.3999999999999999

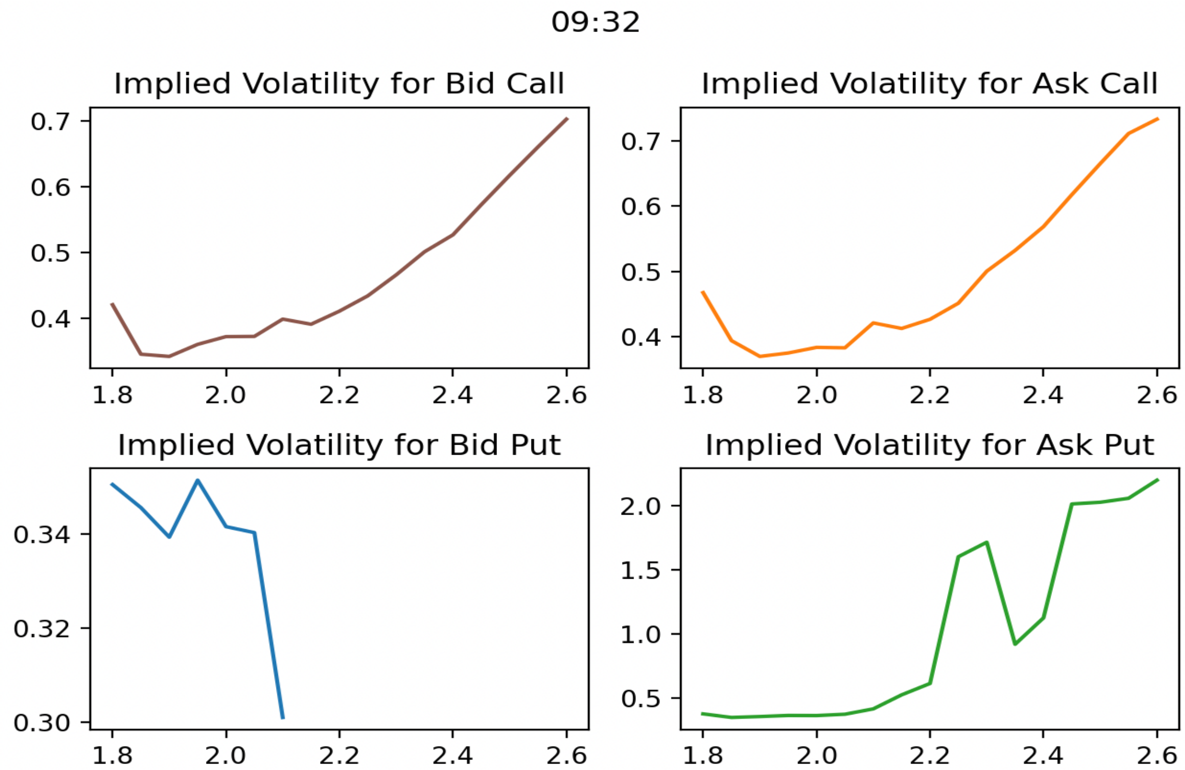
The difference between the actual and implied volatility is 1.1102230246251565e-16

2)

These are the following plots generated using q32.py







Please look at 31,32 and 33 csv files

3)

Program used: q33.py

The following categories of arbitrage were considered along with the corresponding action:

Type	Action
Put Call Parity (Overpriced Call)	Sell Call, Buy Put, Buy ETF, Sell Risk Free Bond
Put Call Parity (Overpriced Put)	Sell Put, Buy Call, Sell ETF, Buy Risk Free Bond
Call Lower Bound	Long risk free Bond and short Stock and Buy Call
Call Upper Bound	short the call and Long the stock and short risk free bond
Put Lower Bound	long the stock and put option and short the risk free bond
Put Upper Bound	Short put and long risk free bond

Two Situations were considered:

1) One without Transaction Fees
2) One With transaction fees

Two Separate csv files were generated for each situation and here is the information regarding each situation printed out by the program q33.py:

Total number of entries (without transaction fees): 521
Total number of entries (with transaction fees): 425

Total Profit (without transaction fees): 5563.399062440139
Total Profit (with transaction fees): 4020.7270606828106

Categorical Spread (without transaction fees):

Type and Action	
Put Call Parity, Sell Call and Buy Put and Buy the Stock and Sell Risk Free Bond	499
Put Call Parity, Sell Put and Buy Call and sell Stock and Buy Risk Free Bond	22

Categorical Spread (with transaction fees):

Type and Action	
Put Call Parity, Sell Call and Buy Put and Buy the Stock and Sell Risk Free Bond	424
Put Call Parity, Sell Put and Buy Call and sell Stock and Buy Risk Free Bond	1