# **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

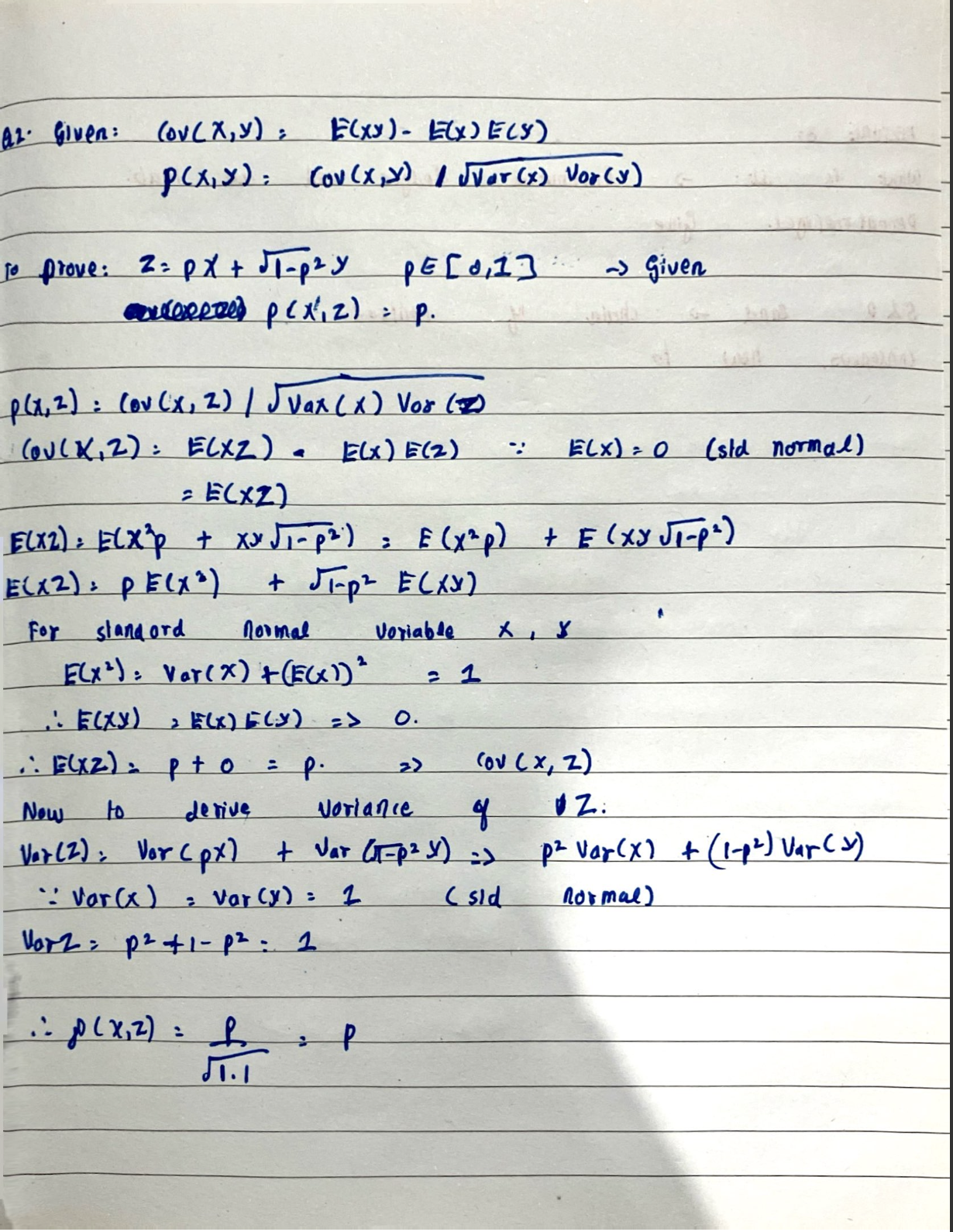
Put Option Price for S=110, K=100, T=1.0, sigma=0.3, r=0.03: 6.917563824509575

**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**

****

**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

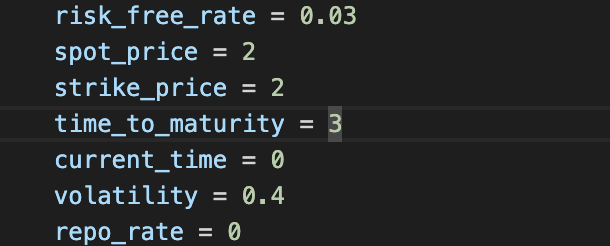
Variance of X: 0.9980440929390706

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:

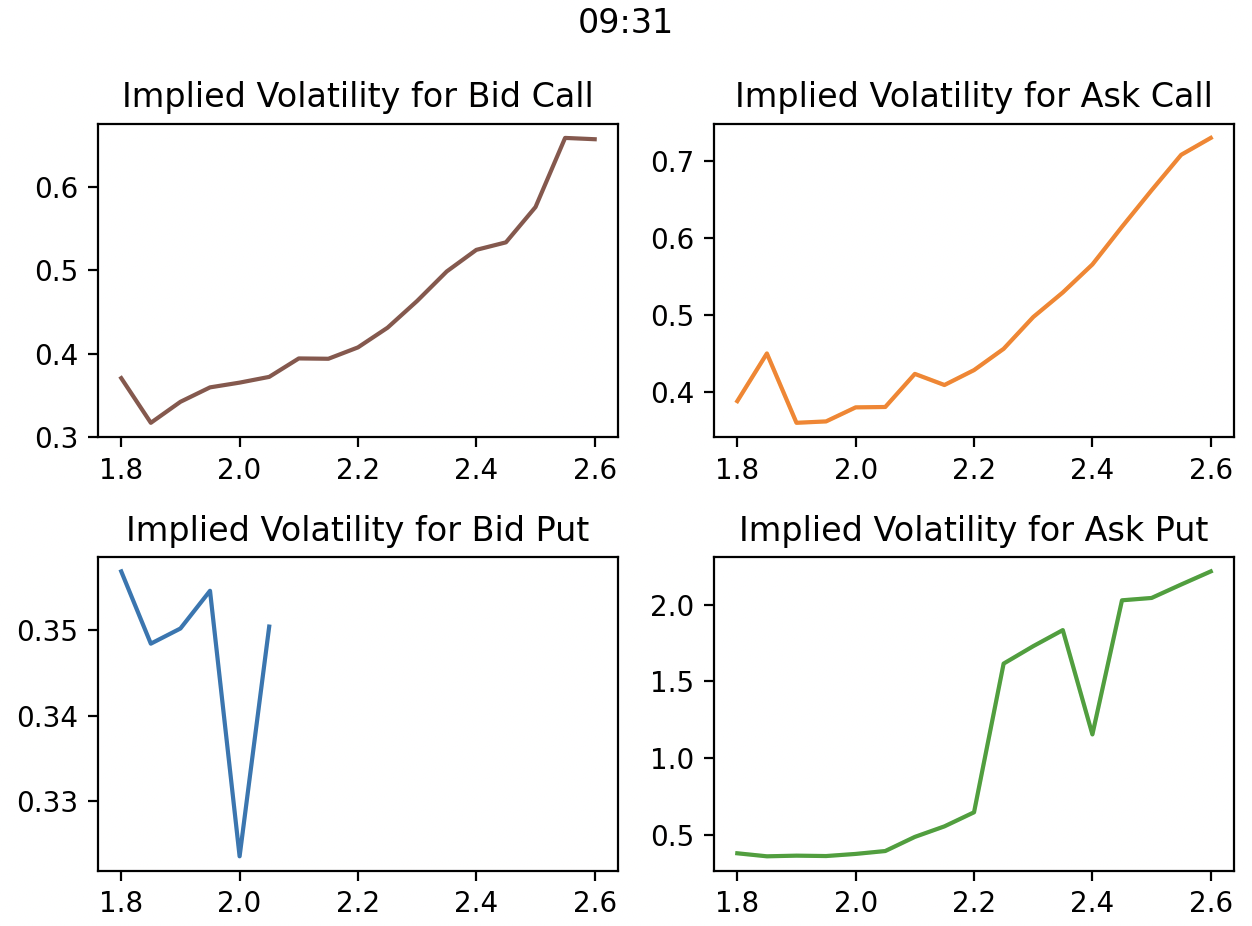
**The code produced the following output:**

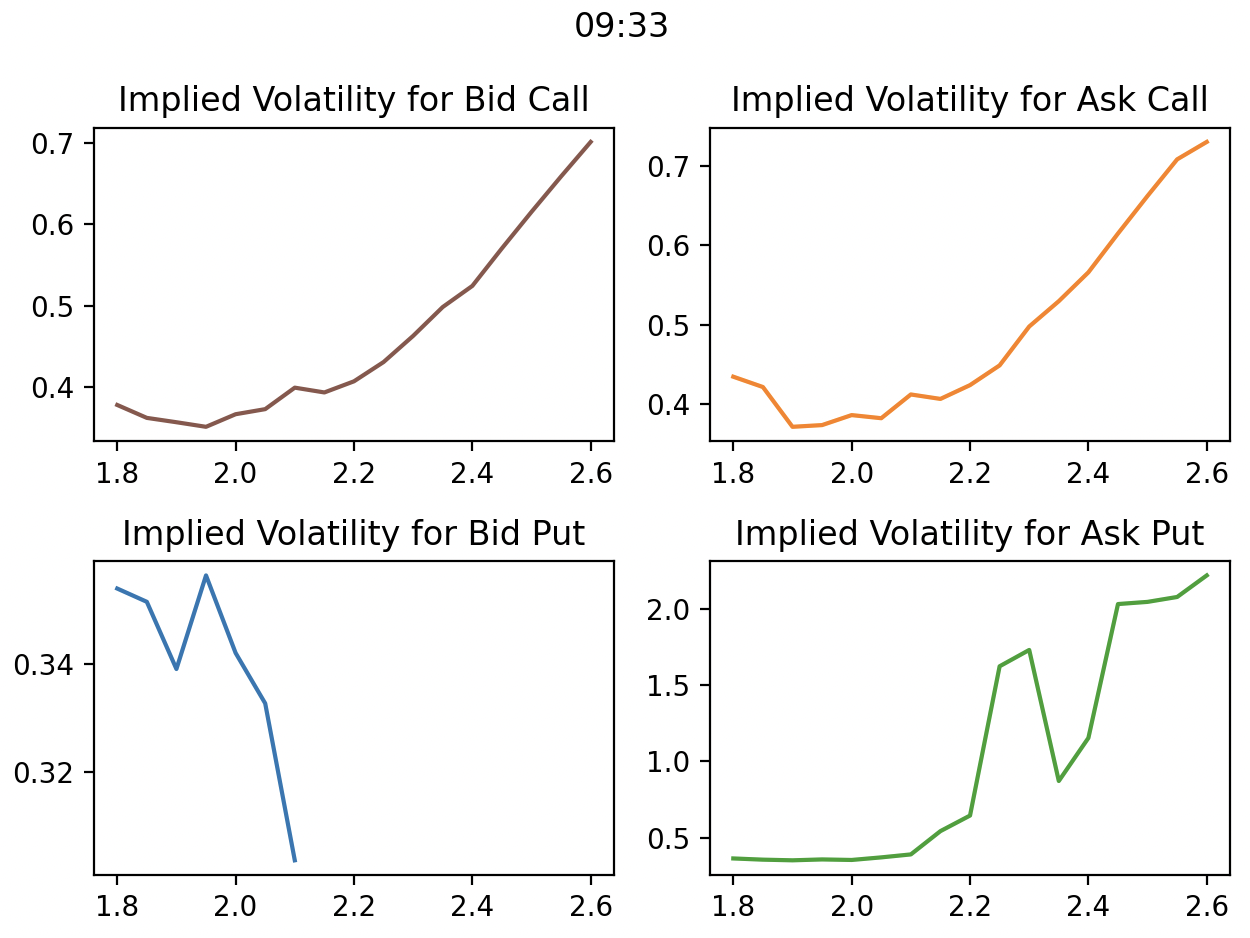
Actual Volatility is 0.4

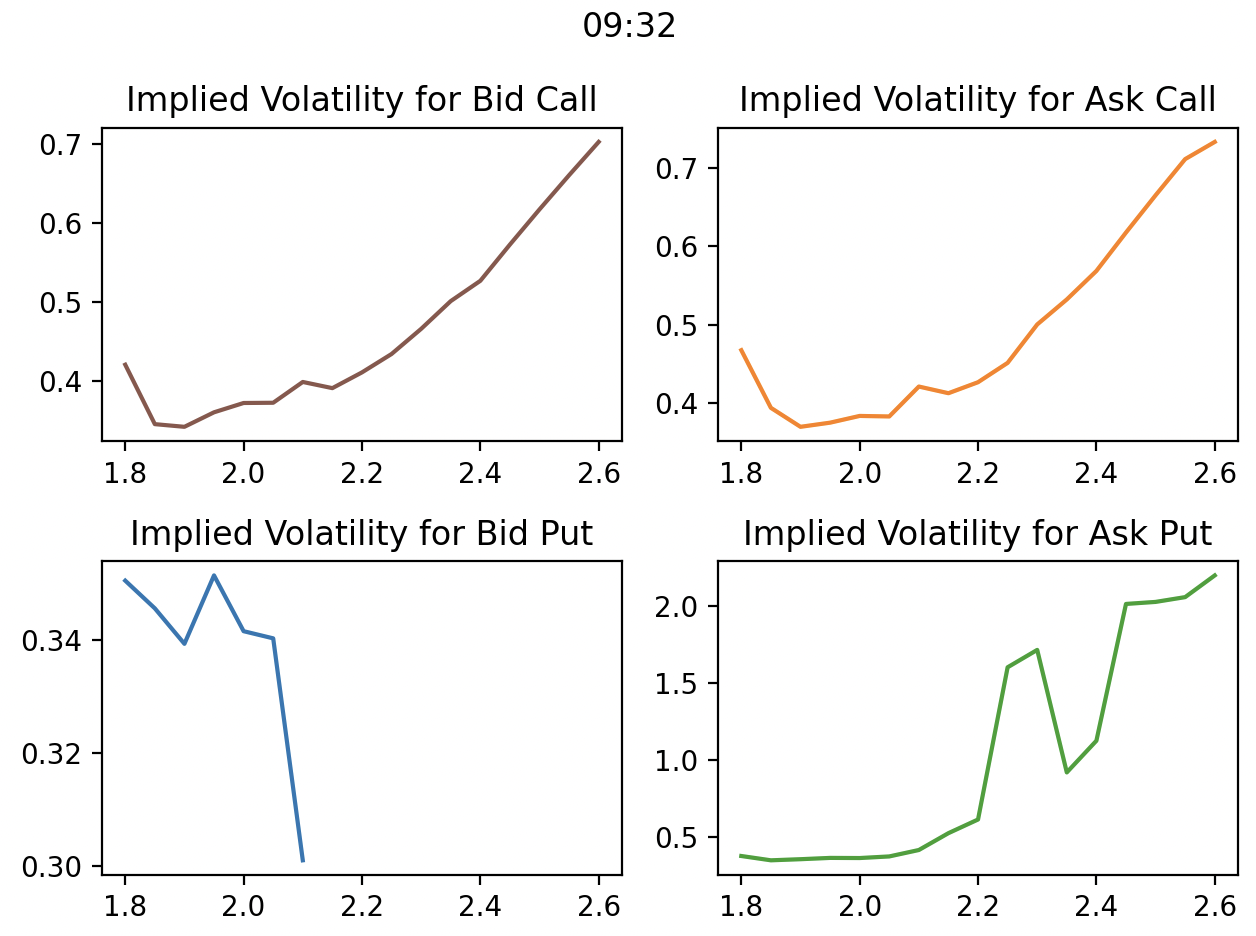
Implied Volatility from Calculation is 0.3999999999999999

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

2)

 These are the following plots generated using q32.py





Please take a look at 31,32 and 33 csv files

3)