Problem 1

Test 1

```
  ... #1.1
  ...
>>> missing_cov(x, skipmiss = True, func = 'cov')
array([[ 1.34482861, -1.29280431,  0.06054652, -0.01611795],
       [-1.29280431,  1.50266133,  0.12431799, -0.10758295],
       [ 0.06054652,  0.12431799,  0.81174387, -0.41351547],
       [-0.01611795, -0.10758295, -0.41351547,  0.3639118 ]])
>>> #1.2
>>>
>>> missing_cov(x, skipmiss = True, func = 'cor')
          x1        x2        x3        x4        x5
x1  1.000000 -0.931618 -0.362959 -0.083616 -0.937042
x2 -0.931618  1.000000  0.344202  0.180583  0.800698
x3 -0.362959  0.344202  1.000000  0.925357  0.069333
x4 -0.083616  0.180583  0.925357  1.000000 -0.252163
x5 -0.937042  0.800698  0.069333 -0.252163  1.000000
>>> #1.3
>>>
>>> missing_cov(x, skipmiss = False, func = 'cov')
          x1        x2        x3        x4        x5
x1  1.173986 -0.629631 -0.278932 -0.081448 -0.735140
x2 -0.629631  1.318197  0.018090  0.446047  0.139309
x3 -0.278932  0.018090  0.918102  0.360836  0.258613
x4 -0.081448  0.446047  0.360836  0.894764 -0.235190
x5 -0.735140  0.139309  0.258613 -0.235190  0.522607
>>> #1.4
>>>
>>> missing_cov(x, skipmiss = False, func = 'cor')
          x1        x2        x3        x4        x5
x1  1.000000 -0.483199 -0.241787 -0.067767 -0.714761
x2 -0.483199  1.000000  0.015446  0.405660  0.178286
x3 -0.241787  0.015446  1.000000  0.488250  0.336248
```

Test 2

2.1

```
>>> EW_calCovariance(x, 0.97)
array([[ 0.85591056,  0.12755907,  0.18692897,  0.08141505,  0.05241202],
       [ 0.12755907,  1.08734957,  0.03271468,  0.11251514, -0.43272936],
       [ 0.18692897,  0.03271468,  0.74477054,  0.13106466,  0.06580594],
       [ 0.08141505,  0.11251514,  0.13106466,  0.86880951,  0.11383623],
       [ 0.05241202, -0.43272936,  0.06580594,  0.11383623,  1.13917985]])
```

2.2

```
>>> EW_calCorre(EW_calCovariance(x, 0.94))
array([[ 1.        ,  0.10971121,  0.21851066,  0.11690219,  0.05967711],
       [ 0.10971121,  1.        , -0.04671647,  0.19177348, -0.44489578],
       [ 0.21851066, -0.04671647,  1.        ,  0.18414813,  0.08992707],
       [ 0.11690219,  0.19177348,  0.18414813,  1.        ,  0.1220277 ],
       [ 0.05967711, -0.44489578,  0.08992707,  0.1220277 ,  1.        ]])
```

2.3

```
>>> cout
array([[ 0.85591056,  0.10583999,  0.17446088,  0.10080896,  0.05892752],
       [ 0.10583999,  1.08734957, -0.04204031,  0.1863955 , -0.49515256],
       [ 0.17446088, -0.04204031,  0.74477054,  0.14812925,  0.08283194],
       [ 0.10080896,  0.1863955 ,  0.14812925,  0.86880951,  0.12139949],
       [ 0.05892752, -0.49515256,  0.08283194,  0.12139949,  1.13917985]])
```

Test 3

3.1

```
>>> near_psd(x)
array([[ 1.17398583, -0.61798867, -0.284559  , -0.06515152, -0.68828687],
       [-0.61798867,  1.3181973 ,  0.01709196,  0.44569555,  0.13917594],
       [-0.284559  ,  0.01709196,  0.91810185,  0.35414666,  0.24605561],
       [-0.06515152,  0.44569555,  0.35414666,  0.89476398, -0.21871734],
       [-0.68828687,  0.13917594,  0.24605561, -0.21871734,  0.52260731]])
```

3.2

```
>>> near_psd(x)
array([[ 1.        , -0.48319932, -0.24178663, -0.06776693, -0.71476123],
       [-0.48319932,  1.        ,  0.01544631,  0.40565982,  0.17828625],
       [-0.24178663,  0.01544631,  1.        ,  0.48824985,  0.33624763],
       [-0.06776693,  0.40565982,  0.48824985,  1.        , -0.32213624],
       [-0.71476123,  0.17828625,  0.33624763, -0.32213624,  1.        ]])
```

3.3

```
>>> higham_nearestPSD(x, 'covariance')
Converged in 20 iterations.
array([[ 1.17398583, -0.62386998, -0.2943348 , -0.05767708, -0.6938878 ],
       [-0.62386998,  1.3181973 ,  0.01644917,  0.44857883,  0.14370287],
       [-0.2943348 ,  0.01644917,  0.91810185,  0.35406689,  0.24686647],
       [-0.05767708,  0.44857883,  0.35406689,  0.89476398, -0.21706217],
       [-0.6938878 ,  0.14370287,  0.24686647, -0.21706217,  0.52260731]])
```

3.4

```
>>> higham_nearestPSD(x,'correlation')
array([[ 1.        , -0.48319932, -0.24178663, -0.06776693, -0.71476123],
       [-0.48319932,  1.        ,  0.01544631,  0.40565982,  0.17828625],
       [-0.24178663,  0.01544631,  1.        ,  0.48824985,  0.33624763],
       [-0.06776693,  0.40565982,  0.48824985,  1.        , -0.32213624],
       [-0.71476123,  0.17828625,  0.33624763, -0.32213624,  1.        ]])
```

Test 4

4.1

```
>>> chol_psd(root, x)
array([[ 1.08350627,  0.        ,  0.        ,  0.        ,  0.        ],
       [-0.57036003,  0.99643702,  0.        ,  0.        ,  0.        ],
       [-0.26262792, -0.13317501,  0.91180746,  0.        ,  0.        ],
       [-0.06013027,  0.4128707 ,  0.43138365,  0.73115953,  0.        ],
       [-0.63524033, -0.22393764,  0.05417906, -0.25689219,  0.        ]])
```

Test 5

5.1

```
>>> res = pd.DataFrame(simulate_normal(100000, x)).cov()
>>> res
          0         1         2         3         4
0  0.085090  0.087774  0.042453  0.008977  0.003887
1  0.087774  0.160986  0.058343  0.012360  0.005334
2  0.042453  0.058343  0.037501  0.005971  0.002581
3  0.008977  0.012360  0.005971  0.001687  0.000547
4  0.003887  0.005334  0.002581  0.000547  0.000315
```

5.2

```
>>> res = pd.DataFrame(simulate_normal(100000, x)).cov()
>>> res
          0         1         2         3         4
0  0.085090  0.116933  0.042446  0.008975  0.003886
1  0.116933  0.160694  0.058331  0.012334  0.005340
2  0.042446  0.058331  0.037485  0.005965  0.002581
3  0.008975  0.012334  0.005965  0.001686  0.000547
4  0.003886  0.005340  0.002581  0.000547  0.000316
```

5.3

```
>>> res = pd.DataFrame(simulate_normal(100000, x, fixMethod = 'near_psd')).cov()
>>> res
          0         1         2         3         4
0  0.085090  0.008845  0.037969  0.008029  0.003479
1  0.008845  0.161034  0.052182  0.011092  0.004784
2  0.037969  0.052182  0.037542  0.006028  0.002607
3  0.008029  0.011092  0.006028  0.001689  0.000552
4  0.003479  0.004784  0.002607  0.000552  0.000315
```

5.4

```
>>> res = pd.DataFrame(simulate_normal(100000, x, fixMethod = 'higham_nearestPSD')).cov()
Converged in 31 iterations.
>>> res
          0         1         2         3         4
0  0.085090  0.013250  0.039030  0.008257  0.003576
1  0.013250  0.161043  0.053651  0.011402  0.004918
2  0.039030  0.053651  0.037546  0.006242  0.002697
3  0.008257  0.011402  0.006242  0.001689  0.000572
4  0.003576  0.004918  0.002697  0.000572  0.000315
```

5.5

```
>>> res
          0         1         2         3         4
0  0.085037  0.116860  0.042491  0.009011  0.003877
1  0.116860  0.160593  0.058393  0.012383  0.005329
2  0.042491  0.058393  0.037730  0.006007  0.002585
3  0.009011  0.012383  0.006007  0.001696  0.000549
4  0.003877  0.005329  0.002585  0.000549  0.000315
```

Test 6

6.1 result = test 6.1

```
>>> res
                SPY      AAPL      MSFT      AMZN      NVDA      GOOGL      TSLA  ...       PGR      SCHW      LRCX       ZTS         C       BSX       AMT
Date                                                                               ...
2022-09-02 -0.010544 -0.013611 -0.016667 -0.002425 -0.020808 -0.017223 -0.025076  ... -0.010428 -0.019242 -0.004236 -0.015244  0.001846 -0.012198 -0.026355
2022-09-06 -0.003773 -0.008215 -0.010974 -0.010980 -0.013336 -0.009643  0.015581  ...  0.000572  0.001848 -0.008019 -0.000892 -0.012695 -0.002717  0.013275
2022-09-07  0.017965  0.009254  0.019111  0.026723  0.018795  0.024717  0.033817  ...  0.038537  0.018731  0.012279  0.022698  0.008503  0.026994  0.020930
2022-09-08  0.006536 -0.009618  0.001666  0.002626  0.020126 -0.009776  0.019598  ...  0.015880  0.019083  0.016574 -0.011908  0.026116  0.029901  0.008362
2022-09-09  0.015535  0.018840  0.022977  0.026575  0.028377  0.020945  0.036023  ... -0.004179  0.018863  0.026460  0.036721  0.015431  0.005385 -0.000306
...              ...       ...       ...       ...       ...       ...       ...  ...       ...       ...       ...       ...       ...       ...       ...
2023-09-18  0.000586  0.016913 -0.003513 -0.002920  0.001503  0.005895 -0.033201  ...  0.013118 -0.006183  0.020125 -0.003329 -0.001639  0.001890 -0.003386
2023-09-19 -0.002074  0.006181 -0.001246 -0.016788 -0.010144 -0.001230  0.004599  ...  0.013589 -0.002247 -0.016519  0.012970  0.000938  0.000566 -0.012087
2023-09-20 -0.009193 -0.019992 -0.023977 -0.017002 -0.029435 -0.031150 -0.014672  ...  0.001544 -0.018361 -0.010062 -0.002748 -0.008903  0.020177  0.000282
2023-09-21 -0.016528 -0.008889 -0.003866 -0.044053 -0.028931 -0.024675 -0.026239  ... -0.002032 -0.011646 -0.013686 -0.026725 -0.013948 -0.002403 -0.045601
2023-09-22 -0.002249  0.004945 -0.007887 -0.001624  0.014457 -0.001457 -0.042315  ...  0.006039 -0.015354  0.014286  0.000283 -0.018940 -0.006856 -0.018368
```

6.1 result = test 6.2

```
>>> res = return_calculate(x,method = "LOG")
>>> res
                SPY      AAPL      MSFT      AMZN      NVDA      GOOGL      TSLA  ...       PGR      SCHW      LRCX       ZTS         C       BSX       AMT
Date                                                                               ...
2022-09-02 -0.010600 -0.013705 -0.016807 -0.002428 -0.021027 -0.017373 -0.025396  ... -0.010483 -0.019429 -0.004245 -0.015361  0.001844 -0.012273 -0.026708
2022-09-06 -0.003780 -0.008249 -0.011035 -0.011040 -0.013426 -0.009690  0.015460  ...  0.000572  0.001846 -0.008052 -0.000892 -0.012776 -0.002720  0.013187
2022-09-07  0.017806  0.009211  0.018931  0.026372  0.018621  0.024416  0.033258  ...  0.037813  0.018558  0.012204  0.022445  0.008467  0.026636  0.020714
2022-09-08  0.006515 -0.009664  0.001665  0.002623  0.019926 -0.009824  0.019409  ...  0.015756  0.018904  0.016438 -0.011979  0.025780  0.029463  0.008327
2022-09-09  0.015416  0.018664  0.022717  0.026228  0.027982  0.020729  0.035389  ... -0.004188  0.018687  0.026116  0.036063  0.015313  0.005371 -0.000306
...              ...       ...       ...       ...       ...       ...       ...  ...       ...       ...       ...       ...       ...       ...       ...
2023-09-18  0.000586  0.016772 -0.003519 -0.002925  0.001502  0.005878 -0.033765  ...  0.013033 -0.006203  0.019925 -0.003334 -0.001640  0.001889 -0.003392
2023-09-19 -0.002076  0.006162 -0.001247 -0.016931 -0.010196 -0.001231  0.004588  ...  0.013497 -0.002249 -0.016656  0.012886  0.000938  0.000566 -0.012161
2023-09-20 -0.009236 -0.020195 -0.024269 -0.017148 -0.029877 -0.031646 -0.014780  ...  0.001543 -0.018532 -0.010113 -0.002751 -0.008943  0.019976  0.000282
2023-09-21 -0.016666 -0.008929 -0.003873 -0.045053 -0.029357 -0.024984 -0.026589  ... -0.002035 -0.011715 -0.013781 -0.027088 -0.014046 -0.002406 -0.046673
2023-09-22 -0.002251  0.004932 -0.007918 -0.001625  0.014354 -0.001458 -0.043237  ...  0.006021 -0.015473  0.014185  0.000283 -0.019122 -0.006879 -0.018538
```

Test 7

7.1

```
model.error_model.mean()# 0.04602574
model.error_model.std() # 0.04677994
```

7.2

```
model.stat # [0.04594038004735414, 0.04544287220830122, 6.336866997308613]
```

7.3

```
>>> model = fit_regression_t(x['y'],x[['x1', 'x2', 'x3']])
print('mu: ', model.error_model.mean()) # 8.743006318923108e
>>> print('mu: ', model.error_model.mean()) # 8.743006318923108e
mu:  8.743006318923108e-18
>>> print('mu: ', model.error_model.mean()) # 8.743006318923108e-18
mu:  8.743006318923108e-18
>>> print('sigma: ', model.error_model.std()) # 0.06311884573136509
sigma:  0.0795599766664563
>>> print('nu: ', model.error_model) # 5.396697627453499
nu:  <scipy.stats._distn_infrastructure.rv_continuous_frozen object at 0x1728856d0>
>>> print('Alpha,beta: ', model.beta) #
Alpha,beta:  [0.04156686 1.02843062 2.18469665 3.17289129]
```

Test 8

8.1

```
>>> print("VaR Absolute", VaR_distribution(model.error_model)) # 0.03092042
VaR Absolute [0.03092042]
>>> print("VaR Diff from Mean",VaR_distribution(model.error_model)+ model.error_model.mean()) #
VaR Diff from Mean [0.07694615]
```

8.2

```
>>> print("VaR Absolute", VaR_distribution(model.error_model)) # 0.041529702716233574
VaR Absolute 0.041529702716233574
>>> print("VaR Diff from Mean", VaR_distribution(model.error_model)+ model.error_model.mean())
VaR Diff from Mean 0.08747008276358771
```

8.3

```
print("VaR Absolute", VaR(sim)) # 0.040364295532057046
print("VaR Diff from Mean",VaR(sim)+np.mean(sim)) #0.0868257807422640
```

8.4

```
>>> print("ES Absolute", ES_distribution(model.error_model))  # 0.05046784401426306
ES Absolute 0.05046784401426306
>>> print("ES Diff from Mean", ES_distribution(model.error_model)+ model.error_model.mean()) #
ES Diff from Mean [0.09649358]
```

8.5

```
>>> print("ES Absolute", ES_distribution(model.error_model)) # 0.07523208716011755
print("ES Diff from Mean", -ES_distribution(model.error_model)- model.error_model.mean()) #
ES Absolute 0.07523208716011755
>>> print("ES Diff from Mean", -ES_distribution(model.error_model)- model.error_model.mean(
ES Diff from Mean -0.12117246720747168
```

8.6

```
# Test 8.6 VaR Simulation
print("ES Absolute", ES(sim)) # 0.07329953615927434
print("ES Diff from Mean",ES(sim)+np.mean(sim)) # 0.11976102136948133
```

Test 9

```
>>> riskOut
             VaR95         ES95  VaR95_Pct  ES95_Pct
Stock
A        94.579515   118.537354   0.046144  0.057833
B       108.089527   152.346966   0.035089  0.049457
Total   153.900252   202.224016   0.030780  0.040445
```

Problem 2

```
VaR_weighted:  0.09028951366738855
ES_weighted:  0.0059592996180301
```

```
VaR_MLEt 0.07647602684516216
ES_MLEt 0.11321790139118341
```

```
VaR_his 0.075861511162783
ES_his 0.11677669788562187
```

Problem 3

```
PortfolioA
total VaR: 7811.662179038411
total ES: 10448.42307091131
total VaR_pct: 0.026043209336771886
total ES_pct: 0.03483387568974419
```

```
PortfolioB
total VaR: 6810.268704290856
total ES: 8919.413224608705
total VaR_pct: 0.023133838464632295
total ES_pct: 0.03029840285265055
```

```
PortfolioC
total VaR: 5714.146829028685
total ES: 7259.900811563228
total VaR_pct: 0.021843720975929976
total ES_pct: 0.02775274286531965
```