# Fintech 545
# Assignment 4

# Problem 1

Use the data in problem1.csv. Fit a normal Distribution and a Generalized T distribution to this data. Calculate the VaR and the ES for both fiited distributions.

Overlay the graphs the distribution PDFs, VaR, and ES values. What do you notice? Explain the differences.

# Problem 1

**Fit Normal Distribution with 10000 simulations:**

```
sim_N = np.random.normal(np.mean(data1),np.std(data1),size=10000)
```

**Fit T Distribution with 10000 simulations using MLE:**

```
def MLE_T(p):
  return -1*np.sum(stats.t.logpdf(data1, df=p[0], loc = p[1],scale=p[2]))
constraints=({"type":"ineq", "fun":lambda x: x[0]-1},
             {"type":"ineq", "fun":lambda x: x[2]})
df, loc, scale = minimize(MLE_T, x0 =
(10,np.mean(data1),np.std(data1)),constraints=constraints).x
sim_T = stats.t(df=df, scale=scale).rvs(10000)
```

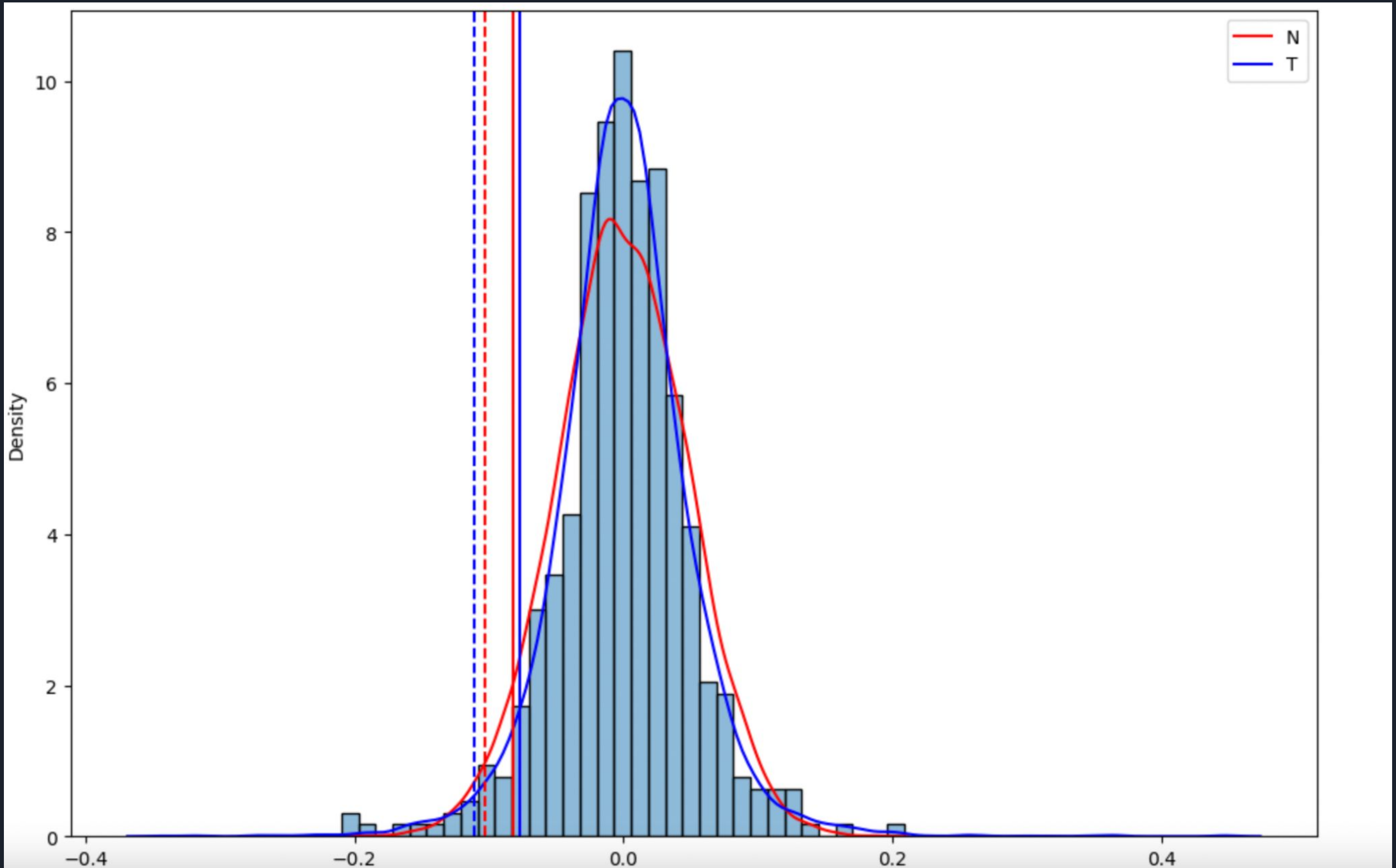
# Problem 1

Calculate the VaR and ES using functions in ft545 package:

```
print("VaR of Normal distribution is ",myfunctions.VaR(sim_N, 0))
print("VaR of T distribution is ",myfunctions.VaR(sim_T, 0))
print("ES of Normal distribution is ",myfunctions.es(sim_N))
print("ES of T distribution is ",myfunctions.es(sim_T))
```

```
VaR of Normal distribution is  0.08174392895998123
VaR of T distribution is  0.07446901701156956
ES of Normal distribution is  0.10296978366696842
ES of T distribution is  0.11009921509529187
```

# Here is the empirial data and fitted distributions



# Problem 1

**Conclusion:The solid lines are the VaR and the dotted lines are the ES. T distribution is a better fit of the graph because it has a greater kurtosis(fatter tail). The VaR of normal distribution is greater than that of T distribution and the ES of normal distribution is less than that of T distribution(loss is a positive number)**

# 2. Problem 2

In your main repository, create a Library for risk management. Create modules, classes, packages, etc as you see fit. Include all the functionality we have discussed so far in class. Make sure it includes

1. Covariance estimation techniques
2. Non PSD fixes for correlation matrices
3. Simulation Methods
4. VaR calculation methods(all discusses)
5. ES calculation

Create a test suite and show that each function performs as expected

# Problem 2

A library called ft545 is created and all the functions are in myfunctions.py. The detailed instruction is in README and here are the test results.

```
====================================== test session starts ======================================
platform darwin -- Python 3.10.7, pytest-7.1.3, pluggy-1.0.0
rootdir: /Users/zengboyuan/Downloads/Fintech545-Quantitive-Rist-Management-main/lib
collected 7 items

tests/test_myfunctions.py .......                                                          [100%]

======================================= 7 passed in 9.40s =======================================
```

# Problem 3

Use your repository from #2. Using Portfolio.csv and Daily Prices.csv. Assume the expected return on all stocks is 0. This file contains the stock holdings of 3 portfolios. You own each of these portfolios. Fit a Generalized T model to each stock and calculate the VaR and ES of each portfolio as well as your total VaR and ES. Compare the results form this to your VaR from Problem3 from Week4.

# Problem 3

Since each stock is of T distribution instead of normal distribution, we need to fit each stock using T distribution first and then use Copula simulation.

```
for i in p['Stock']:
    holdings.append(p[p['Stock']==i]['Holding'].iloc[0])
    cps.append(dp.loc[:,i].iloc[-1])
    total+=dp.loc[:,i].iloc[-1]*p[p['Stock']==i]['Holding'].iloc[0]
    returns = myfunctions.return_calculate(dp.loc[:,i])
    def MLE_T(p):
        return -1*np.sum(stats.t.logpdf(returns, df=p[0], loc = p[1],scale=p[2]))
    constraints=({"type":"ineq", "fun":lambda x: x[0]-2},
             {"type":"ineq", "fun":lambda x: x[2]})
    df, loc, scale = minimize(MLE_T, x0 = (2,np.mean(returns),np.std(returns)),constraints=constraints).x
    params.append([df,loc,scale])
    cdf.append(stats.t.cdf(returns, df = df, loc = loc, scale = scale))
sim = myfunctions.multi_normal_sim(stats.spearmanr(np.array(cdf), axis=1)[0],1000)
rt = []
for i in range(len(sim)):
    rt.append(stats.t.ppf(stats.norm.cdf(sim[i]), df = params[i][0], loc = params[i][1], scale = params[i][2]))
```

# Problem 3

Results:

```
the VaR of portfolio A is 6088.051472116343
the ES of portfolio A is 7827.26676477154
the VaR of portfolio B is 4588.4359311812295
the ES of portfolio B is 5905.84762118043
the VaR of portfolio C is 3386.636384165054
the ES of portfolio C is 4916.092085448166
the VaR of portfolio total is 13415.849435841787
the ES of portfolio total is 18168.745384693135
```

# Problem 3

**Conclusion: There are only little difference between the results here and the results from week4. However, simulation using Copulus should be a better fit if we have the right distribution of each stock because it does not require normality assumption.**