Haochen Yang

Problem 1

Use the data in problem1.csv. Fit a Normal Distribution and a Generalized T distribution to this data. Calculate the VaR and ES for both fitted distributions.
Overlay the graphs the distribution PDFs, VaR, and ES values. What do you notice? Explain the differences.

Answer:

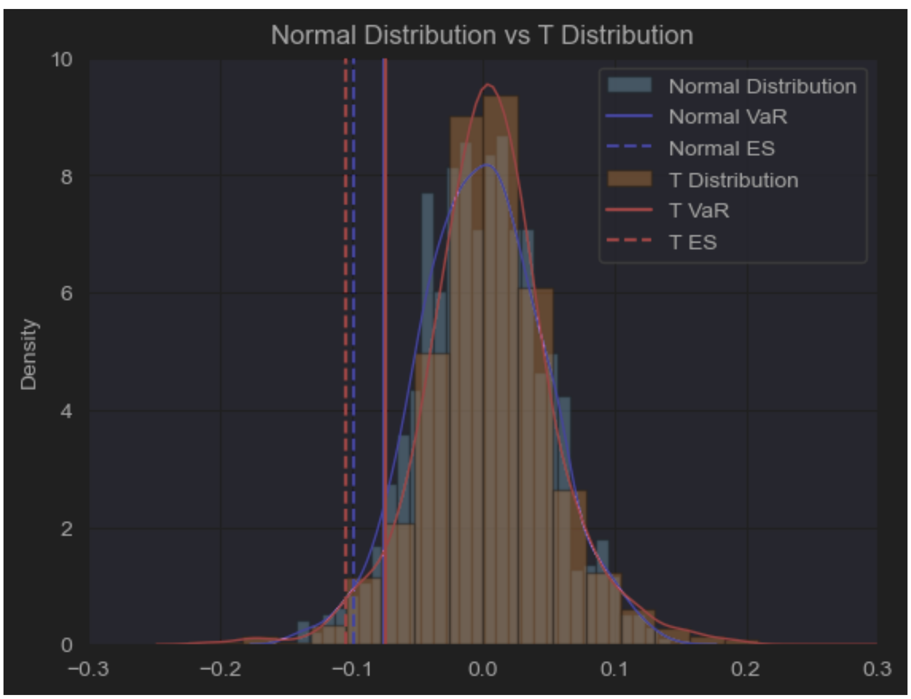
I fitted the data to a normal distribution and a generalized T distribution, and Value at Risk (VaR) and Expected Shortfall (ES) are below:

	Normal Distribution	Generalized T Distribution
VaR	0.0752	0.0737
ES	0.0983	0.1041

As can be obtained from the table, Var for normal distribution is higher than that of generalized T distribution, while ES for normal distribution is lower than that of generalized T distribution.

For T distribution itself having heavier tails, e.g., more extreme values on sides, extreme events are more likely, thus, the ES, which averages the losses in the worst α% cases, would generally be higher for the T distribution compared to a normal distribution for the same level of α. However, VaR depends on more parameters, such as the degrees of freedom, if the degrees of freedom are high, then T distribution should resemble normal distribution, thus yielding similar VaR; while the degrees of freedom are low, then VaR of T distribution would be higher than that of normal distribution.

Below is the graph that may more vividly illustrate the problem:



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 discussed)
5. ES calculation

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

Answer:

The package containing all required functions are in folder “lib_hcy_week5”, and shall be installed via “pip install” and called via “import lib_hcy_week5” or “from lib_hcy_week5 import myfunctions”.

Passes covariance estimation tests:

```
# test covariance estimation
df = pd.read_csv("DailyPrices.csv")
all_ret = mf.return_calculate(df)
all_ret.drop('Date', axis=1, inplace=True)

expo_w_cov = mf.expo_weighted_cov(all_ret)

ps_var_ps_cor = mf.cal_ps_var_ps_cor(all_ret)
ew_var_ew_cor = mf.cal_ew_var_ew_cor(all_ret)
ew_var_ps_cor = mf.cal_ew_var_ps_cor(all_ret)
ps_var_ew_cor = mf.cal_ps_var_ew_cor(all_ret)

print(ps_var_ps_cor.shape)
print(ew_var_ew_cor.shape)
print(ew_var_ps_cor.shape)
print(ps_var_ew_cor.shape)
Executed at 2023.10.07 12:32:56 in 160ms

(101, 101)
(101, 101)
(101, 101)
(101, 101)
```

Passes non PSD fixes for correlation matrices tests:

```
# test non PSD fixes
n = 500
sigma = np.full((n,n),0.9)
for i in range(n):
    sigma[i,i]=1.0
sigma[0,1] = 0.7357
sigma[1,0] = 0.7357

is_psd1 = mf.is_psd(sigma)
print(is_psd1)

near_sigma = mf.near_psd(sigma)
is_psd2 = mf.is_psd(near_sigma)
print(is_psd2)

weight = np.identity(sigma.shape[0])
higham_sigma = mf.higham_psd(sigma, weight)
is_psd3 = mf.is_psd(higham_sigma)
print(is_psd3)
Executed at 2023.10.07 12:33:03 in 6s 964ms

False
True
True
```

Passes simulation methods tests:

```
# test simulation methods
simu_from_cov = mf.simu_from_cov(ew_var_ew_cor, 1000)
simu_from_pca = mf.simu_from_pca(ew_var_ew_cor, 1000, 0.97)

print(simu_from_cov.shape)
print(simu_from_pca.shape)
```

Executed at 2023.10.07 12:33:03 in 40ms

```
(1000, 101)
(1000, 101)
```

Passes VaR calculation methods tests:

```
# test VaR calculation methods
META = all_ret['META']
META_centered = META - META.mean()

norm_VaR, _ = mf.norm_VaR(META_centered)
norm_ew_VaR, _ = mf.norm_VaR(META_centered)
MLE_T_VaR, _ = mf.MLE_T_VaR(META_centered)
ar1_VaR, _ = mf.ar1_VaR(META_centered)
his_VaR, _ = mf.his_VaR(META_centered)
print(norm_VaR)
print(norm_ew_VaR)
print(MLE_T_VaR)
print(ar1_VaR)
print(his_VaR)
```

Executed at 2023.10.07 12:33:03 in 115ms

```
0.05326282065091955
0.053455081154041725
0.04747854671003253
0.05551525873249308
0.03948424995533789
```

Passes ES calculation tests:

```
var_t, dist_t = mf.MLE_T_VaR(p1_data)
es_t = mf.cal_ES(var_t, dist_t)
print("Under Generalized T Distribution, VaR is {:.4f}, ES is {:.4f}".format(round(var_t, 4), round(es_t, 4)))
```

Executed at 2023.10.07 15:11:38 in 32ms

```
Under Generalized T Distribution, VaR is 0.0737, ES is 0.1041
```

Problem 3

Use your repository from #2.
Using Portfolio.csv and DailyPrices.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 from this to your VaR form Problem 3 from Week 4.

Answer:

I fitted the stock returns through generalized T distribution via Gaussian Copula, and the VaR and ES for four portfolios are below:

	Portfolio A	Portfolio B	Portfolio C	Portfolio Total
VaR	19945.822	11678.926	27079.722	53930.945
ES	29852.888	16039.015	37654.669	74463.148

Below are the results for generalized t VaR:

	VaR in Dollar Amount
Portfolio A	19945.822
Portfolio B	11678.926
Portfolio C	27079.722
Portfolio Total	53930.945

Below are the results for delta normal VaR:

	VaR in Dollar Amount
Portfolio A	15426.968
Portfolio B	8082.572
Portfolio C	18163.292
Portfolio Total	38941.376

Below are the results for historical simulated VaR:

	VaR in Dollar Amount
Portfolio A	16525.829
Portfolio B	10757.91
Portfolio C	21981.393
Portfolio Total	47618.778

As can be obtained from the tables above, historical simulated VaR is very similar to generalized t VaR, this could probably be due to the fact that historical distribution of asset returns is more similar to simulated t distribution of asset returns, which means actual asset returns have more extreme values, outliers that happened to be better captured by t distribution. However, we can also see that generalized t VaR is a little bigger than historical simulated VaR, this is probably because generalized t distribution has overestimated the extreme values than they actually are, e.g., fatter tail than actual. But to conclude, clearly t distribution is a better estimator than normal distribution for the given data set.