ETFs Portfolio VaR and CoVaR of SPY

1.Abstract

The overall task Consider the system of 10 S&P select sector ETFs, XLB, XLE, XLF, XLI, XLK, XLP, XLU, XLV, XLY and SPY.

See the website http://www.sectorspdr.com/sectorspdr/for definitions and descriptions.

As for the core idea of this project, first create a portfolio based on 10 ETFs under the mean-variance optimization based on several rules. Then, we calculate returns from 2010-01-01 to 2022-09-30 for 10 ETFs, and portfolio Under the Mean-Variance mode. Next calculate the rolling VaR for each ETF, and portfolio. Finally, get the rolling CoVaR for each asset.

2. Risk Model Review

As we discussed in our lecture, it is important to know the CoVaR between our portfolio and the SPY.

Jianlin (2015) describes the background of CoVaR and the way to compute it. The VaR is a alpha-quantile of the return distribution and the CoVaR is just VaR of conditional distribution. Quantile regression is one of trackable and efficient ways to estimate CoVaR.

Kuan-Heng (2014) achieves a Copula method to calculate CoVaR for measuring systemic risk. He investigates the evolution of dependence structure and systemic risk in 10 S&P 500 sector indices in the U.S. stock market by estimating daily Copula $\Delta CoVaR$ and Copula $\Delta CoES$ from January 1, 1995, to July 31, 2013.

3. Data Preparation

3.1 Prepare the Portfolio

First, use XLB, XLE, XLF, XLI, XLK, XLP, XLU, XLV, XLY, 9 ETFs as a portfolio based on Mean-Variance Optimization Model.

Set constraints. I assume we could buy or sell (-1,1) each asset. The sum of three assets' weights should be 1. Implement constraints in python.

Figure 1: Portfolio Optimization with 9 ETFs

```python
# sigma_m = stock_cov
import scipy.optimize as sco
# from scipy.optimize import minimize

def objective(weights):
    weights = np.array(weights)
    stock_cov_2 = return_etfs.cov() * bf_day
    return weights.dot(stock_cov_2).dot(weights.T)


cons = ({"type":"eq", "fun": lambda x: np.sum(x)-1})
# Every stock can get any weight from -1 to 1
bounds = tuple((-1,1) for x in range(return_etfs.shape[1]))


# Initialize the weights with an even split
# In out case each stock will have 10% at the beginning
guess = [1./return_etfs.shape[1] for x in range(return_etfs.shape[1])]
```

```python
optimized_results = sco.minimize(objective, guess, method = "SLSQP", bounds=bounds, constraints=cons)
optimized_results
```

```
     fun: 0.2348773962394212
     jac: array([0.46950235, 0.47008601, 0.46941388, 0.46995582, 0.46946456,
       0.46967955, 0.46988058, 0.4695417 , 0.46975935])
 message: 'Optimization terminated successfully'
    nfev: 130
     nit: 13
    njev: 13
  status: 0
 success: True
       x: array([-0.0260997 ,  0.03245827, -0.16044157, -0.03426467, -0.12986182,
        0.75504151,  0.11269287,  0.28601166,  0.16446345])
```

3.2 Download Other ETFs

Then, download other ETFs from yahoo finance website from 2010-01-05 to 2022-09-29. Calculate the return and combine with the portfolio returns for each day.

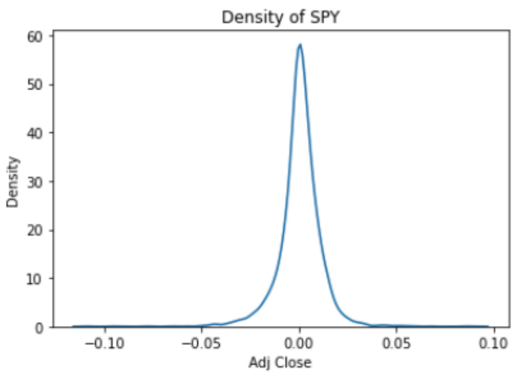
Figure 2: ETFs and Portfolio Returns

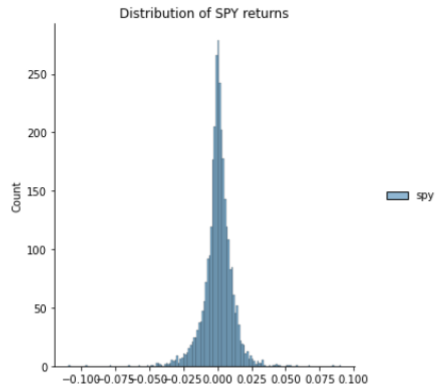
Date	XLB	XLE	XLF	XLI	XLK	XLP	XLU	XLV	XLY	spy	port
2010-01-05	0.003233	0.008162	0.018380	0.003531	-0.001289	0.000375	-0.011905	-0.009807	0.003667	0.002647	-0.005982
2010-01-06	0.016994	0.011975	0.002006	0.002111	-0.011188	-0.000750	0.005861	0.010224	0.001329	0.000704	0.004241
2010-01-07	-0.007779	-0.001500	0.021347	0.010885	-0.003916	0.000000	-0.004532	0.003479	0.008292	0.004221	-0.001287
2010-01-08	0.013937	0.006510	-0.005879	0.015977	0.006553	-0.003376	-0.000976	0.001576	-0.000329	0.003328	-0.002870
2010-01-11	-0.005441	-0.001327	0.000657	0.010941	-0.003906	0.002634	0.010417	0.005664	-0.001974	0.001397	0.004584
...	...	...	...	...	...	...	...	...	...	...	...
2022-09-23	-0.020470	-0.068956	-0.015848	-0.018757	-0.013080	-0.017002	-0.011570	-0.005107	-0.023020	-0.016755	-0.016208
2022-09-26	-0.016221	-0.024546	-0.015781	-0.010501	-0.006950	0.000721	-0.024108	-0.009533	-0.002254	-0.009893	-0.001849
2022-09-27	0.002228	0.011200	-0.004254	-0.003339	0.001302	-0.017284	-0.016850	-0.003044	0.003149	-0.002553	-0.014368
2022-09-28	0.025789	0.044304	0.019717	0.022015	0.010566	0.012018	0.010748	0.021866	0.027021	0.019676	0.016459
2022-09-29	-0.014738	0.000000	-0.012569	-0.017326	-0.026058	-0.016510	-0.040092	-0.008398	-0.034549	-0.020889	-0.018689

3207 rows × 11 columns

Here is the distribution of SPY returns and the density of SPY. We could figure out they are close to the normal distribution.

Figure 3: SPY returns distribution and density





3.3 Comparation between SPY and the Portfolio

Now is the result of comparation between SPY and this portfolio returns.

The portfolio shows less volatility, max drawdown 14.19%, and less cumulative sum returns than the SPY, but a larger Sharpe Ratio 5.39%. So, our portfolio is a good way to provide stable returns.

Figure 4: Portfolio & SPY

From this plot is a comparation between 10 ETFs, the blue line is SPY, and we could see the XLY, XLV, and XLK perform better than SPY.

Figure 5: 10 ETFs returns


```
print("Portfolio Cumsum Asset:",return_etfs1['port'].cumsum()[-1]*1000000)
print("SPY Cumsum Asset:",return_spy.cumsum()[-1]*1000000)

print("Portfolio Mean Asset",port.mean()*1000000)
print("SPY Mean Asset",return_spy.mean()*1000000)

print("Portfolio Sharpe Ratio:",port.mean()/port.std())
print("SPY Sharpe Ratio",return_spy.mean()/return_spy.std())

print("Portfolio Max Drawdown",get_max_drawdown_fast(list(port.values))[0])
print("SPY Max Drawdown",get_max_drawdown_fast(list(return_spy.values)))
```

```
Portfolio Cumsum Asset: 1480323.319807127
SPY Cumsum Asset: 1601328.408416149
Portfolio Mean Asset Portfolio    461.591306
dtype: float64
SPY Mean Asset 499.3228588762548
Portfolio Sharpe Ratio: Portfolio    0.053937
dtype: float64
SPY Sharpe Ratio 0.04546377315991613
Portfolio Max Drawdown 0.14129272049975874
SPY Max Drawdown 0.19490996455541743
```

4. VaR and CoVaR Calculation

4.1 VaR and CoVaR

In this calculation section, I choose the historical method to calculate the 10-day rolling VaR for 10 ETFs and Portfolio. Then use the quantile regression to determine the confidence level. Finally calculate the 10-day rolling and 100-day rolling CoVaR for each ETF and portfolio returns with SPY.

VaR definition

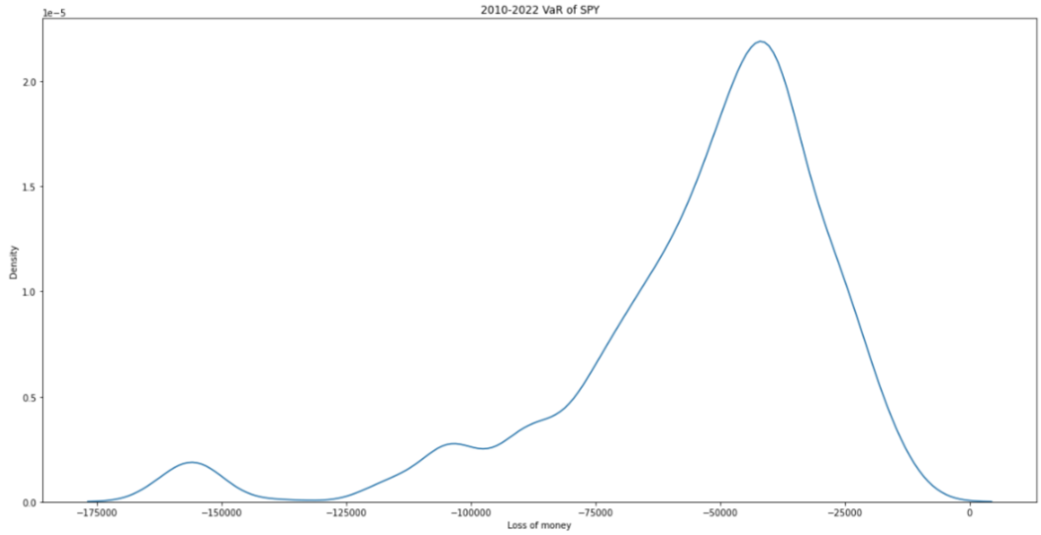
$$\Pr\left(X_t^i \leq VaR_{\alpha,t}^i\right) = \alpha$$

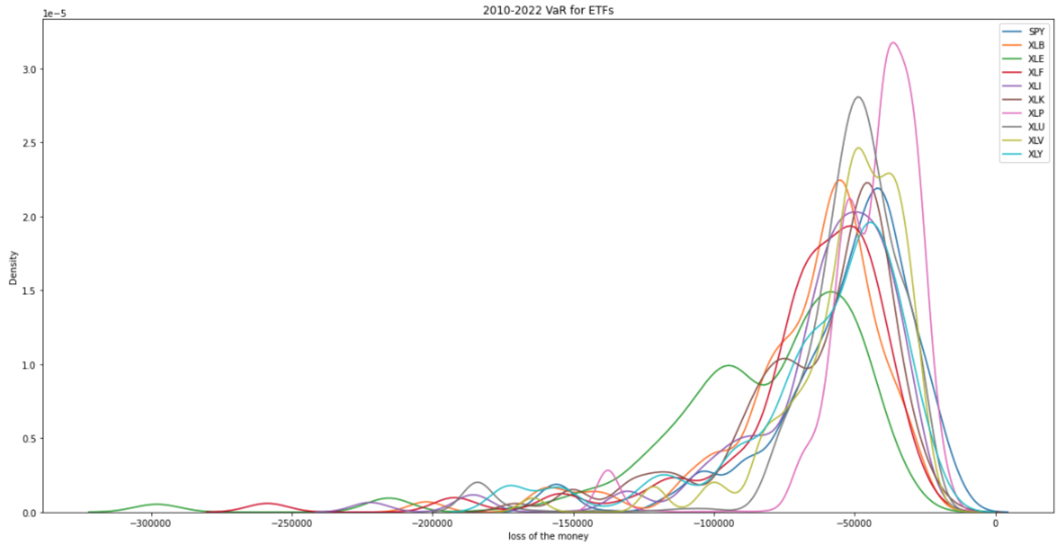
CoVaR definition

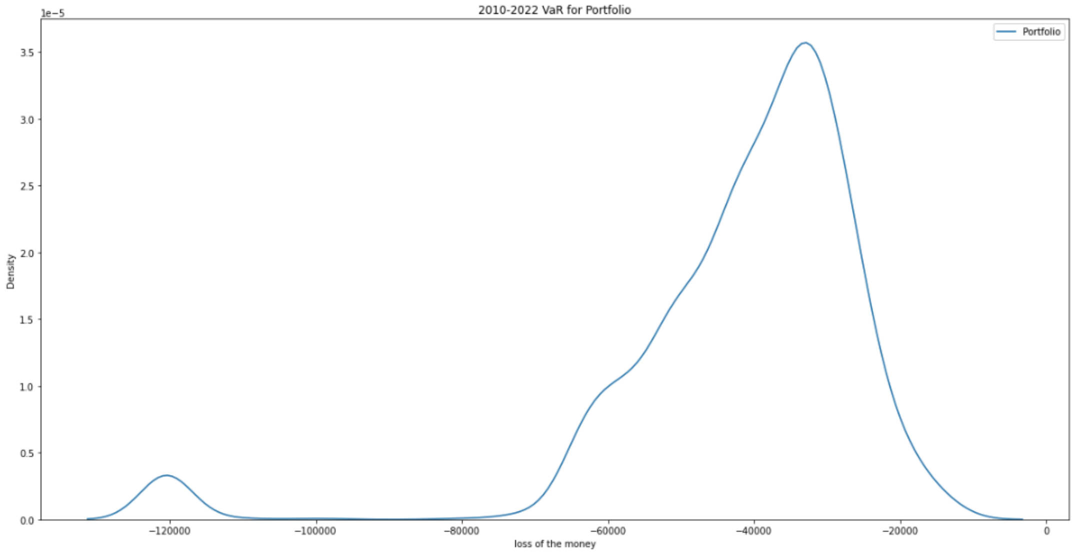
$$\Pr\left(R_t^i \leq CoVaR_{\beta,t}^{i|J} \mid X_t^j = VaR_{\alpha,t}^i\right) = \beta$$

First assume the confidential level is 5%. VaR under 5% means max loss given 5% over a 10-day time. Here are the 10-day rolling VaR density plots for ETFs and Portfolio.

Figure 6: 10 ETFs VaR and Portfolio VaR







4.2 Determine Confidential Level

Here is a quantile regression between portfolio and SPY. We follow this quantile regression process and determine which confidential level is the most appropriate level.

$$X_t^i = \alpha^i + \beta^i M_{t-1} + \varepsilon_t^i$$

$$X_t^j = \alpha^{j|i} + X_t^j + \beta^{j|i} M_{t-1} + \varepsilon_t^{j|i}$$

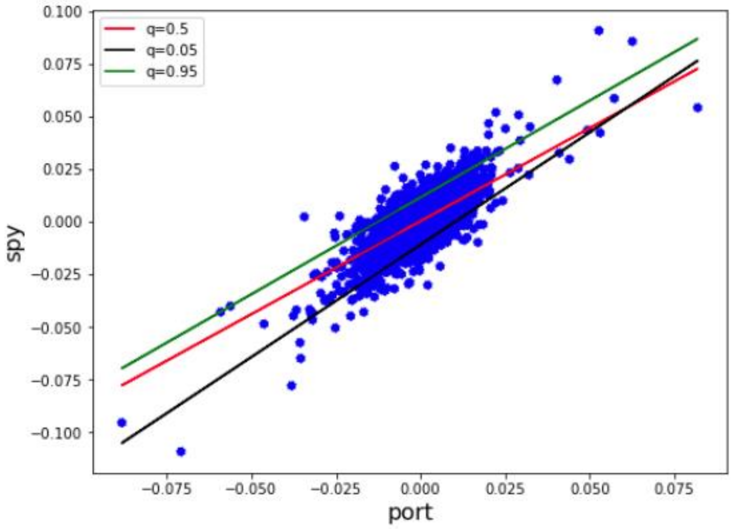
$$VaR_{\alpha,t}^i = \widehat{\alpha}^i + \widehat{\beta^i} M_{t-1}$$

$$CoVaR_{q,t}^{i|J} = VaR_{q,t}^i + \widehat{\alpha}^{j|i} + \widehat{\beta^{j|i}} M_{t-1}$$

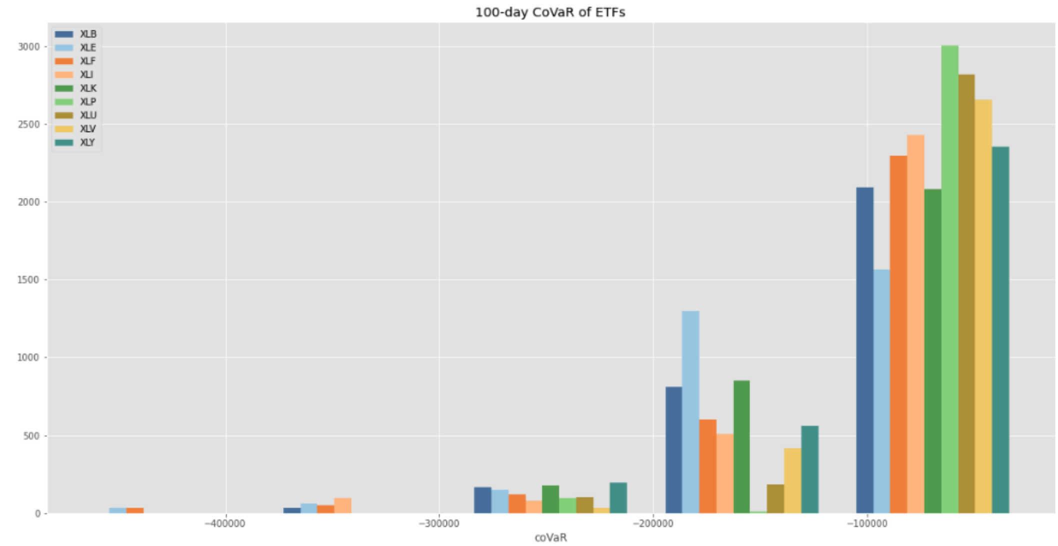
Figure 7: Quantile Regression

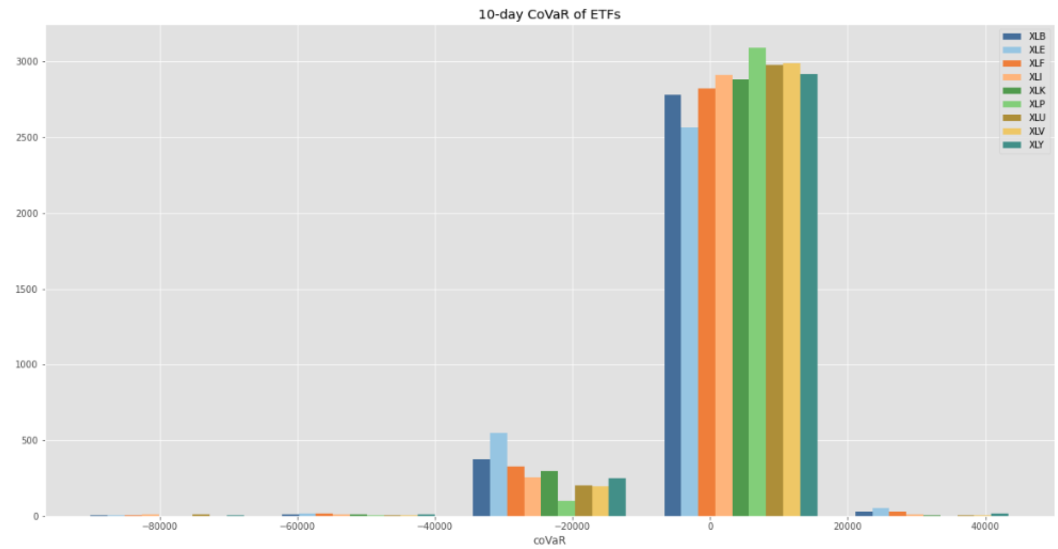
```
import statsmodels.formula.api as smf
q = 0.5
q2 = 0.05
q3 = 0.95
mod = smf.quantreg("spy ~ port",data)
res = mod.fit(q)
res2 = mod.fit(q2)
res3 = mod.fit(q3)
print(res.summary())
```

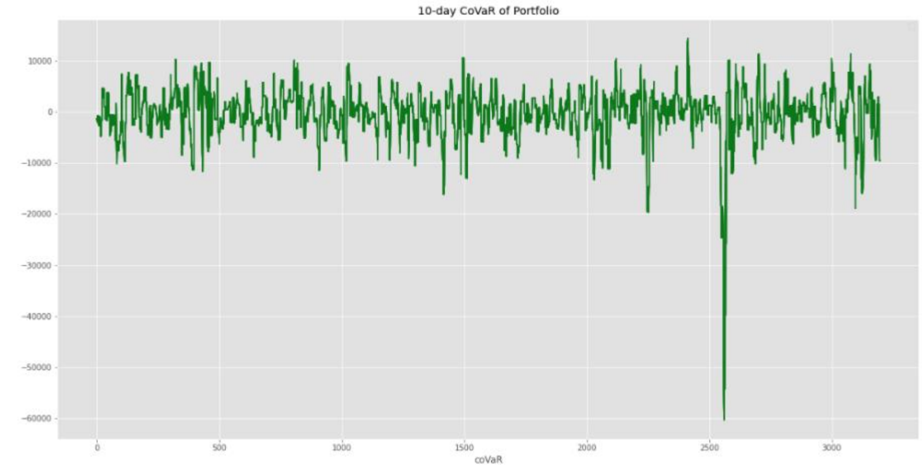
```
                         QuantReg Regression Results
==============================================================================
Dep. Variable:                    spy   Pseudo R-squared:               0.2866
Model:                       QuantReg   Bandwidth:                    0.001849
Method:                 Least Squares   Sparsity:                      0.01337
Date:                Tue, 29 Nov 2022   No. Observations:                 3207
Time:                        08:57:50   Df Residuals:                     3205
                                        Df Model:                            1
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0001      0.000      0.873      0.383      -0.000       0.000
port           0.8847      0.014     64.117      0.000       0.858       0.912
==============================================================================
```

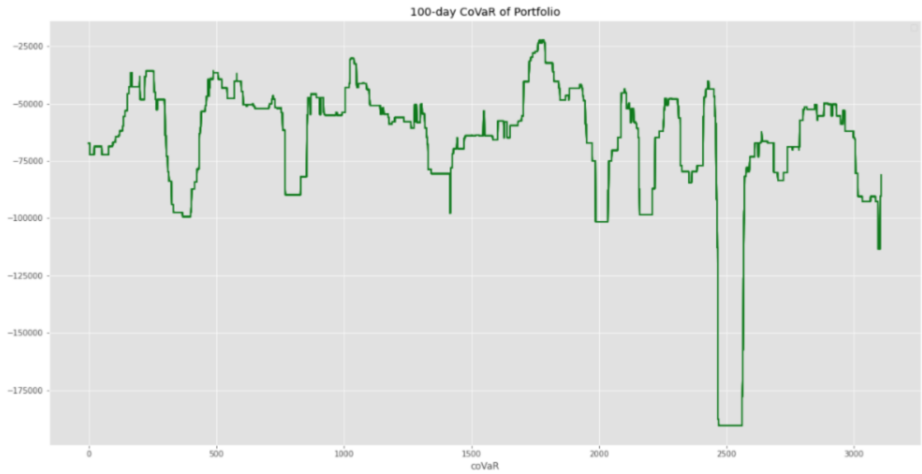


So based on the confidential line, we choose 5% as our confidential level. These are the 10-day rolling, 100-day-rolling, CoVaR for ETFs, Portfolio and SPY.









6. Improvements with relevant topic

1. Try to use GANs to simulate new test data (similar with previous training data) for VaR and CoVaR calculation
2. Combine Copula to the joint distribution of returns for the 10 sectors.
3. Improve the portfolio with Robust Optimization method to give a higher returns result.

7.References

1. Systemic Risk Measure, CoVaR and Copula Master Thesis from Humboldt-Universit¨at zu Berlin, Jianlin Zhang, 2015
2. Measuring Systemic Risk, Copula CoVaR, from KuanHeng Chen, Khaldoun Khashanah, 2014

For the project codes please visit:
https://github.com/zongrui991007/ETFs-Portfolio-VaR-and-CoVaR-of-SPY/tree/main/ETFs%20Portfolio%20VaR%20and%20CoVaR%20of%20SPY