Homework 2

Shi Bo

October 8, 2020

Task 1 : Sector ETF Factor Modeling

Problem 1

a) Download historical data on the Fama-French factors from Ken French's website and write a piece of code that parses them into a Python dataframe (or other similar structure). Validate and clean data.

Download Fama-French factors Data from Ken French's website and check whether there are data anomalies. The time range is from January 1st 2010 to today. Head of factor data is shown as follows:

Date	Mkt-RF	SMB	HML
2010-01-04	1.69	0.58	1.12
2010-01-05	0.31	-0.59	1.21
2010-01-06	0.13	-0.24	0.52
2010-01-07	0.4	0.09	0.94
2010-01-08	0.33	0.4	0.01

Figure 1: Head of F-F data

	data_anomalies
Mkt-RF	0
SMB	0
HML	0

Figure 2: validating data

Problem 2

(b) Calculate the daily covariance matrix of the factor returns over the entire time period. Are the factors highly correlated? Compare these correlations to the correlations of the sector ETFs that you calculated in HW1. Are they more or less correlated?

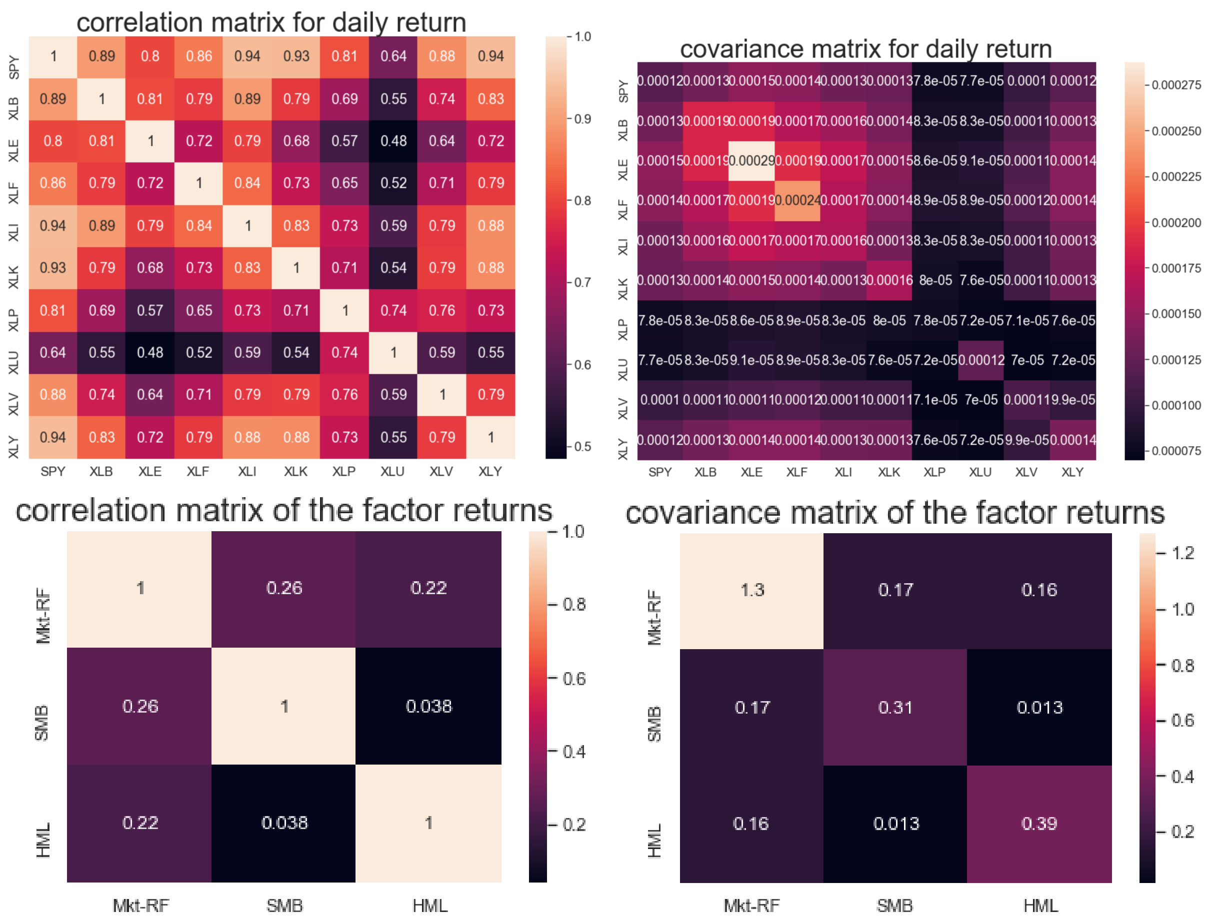


Figure 3: Daily and monthly return correlation and co-variance matrix

We can clearly see that factors are not highly correlated comparing to correlation and covariance for ETFs. If they are highly correlated, we can not do the regression in 3-factor Fama-French model since there will be multi-co linearity.

Problem 3

(c) Calculate rolling 90 day correlations for the factor returns. Are they stable over time? Are they more stable than the correlations of the ETFs from HW1?

From the charts below, we know that rolling correlations of F-F factors are more unstable than that of SP index with each sector ETF.

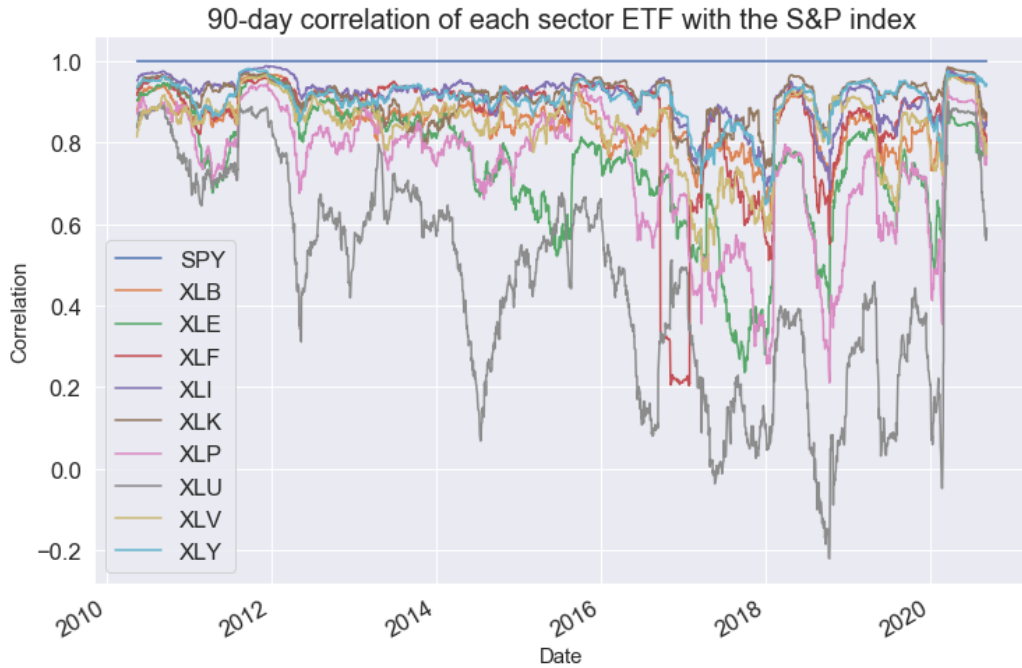


Figure 4: 90-days correlation of each sector

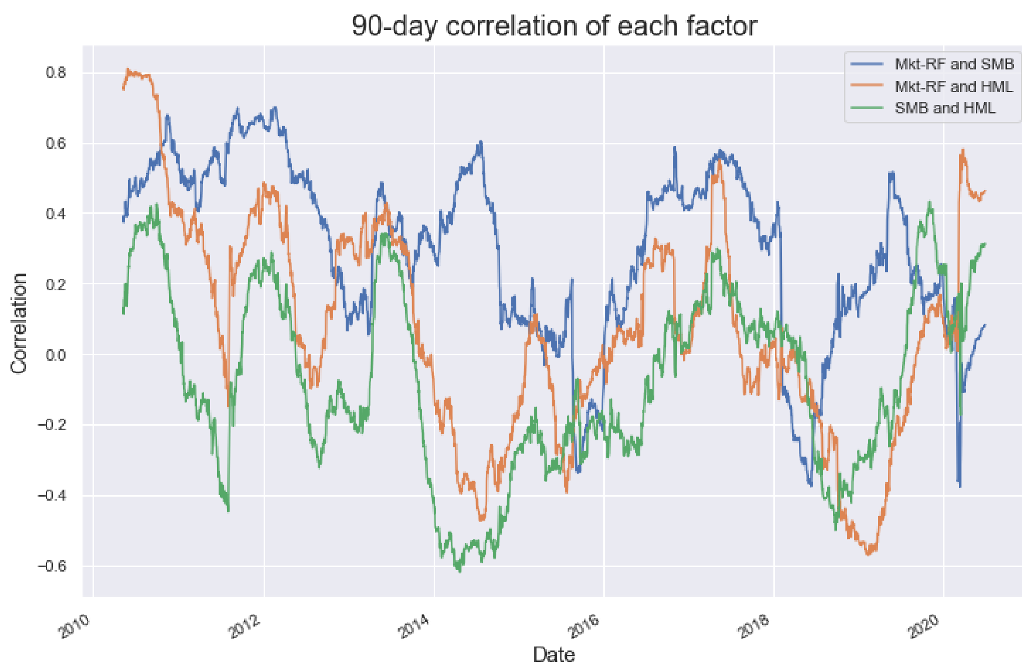


Figure 5: 90-days correlation of each factor

Problem 4

(d) Check the factor returns for normality using your favorite test. Do the factor returns appear normal?

By using the QQ-plot, it is lucid that there are a lot of outliers in the dataset. So, the factor returns do not appear normal in this regard.

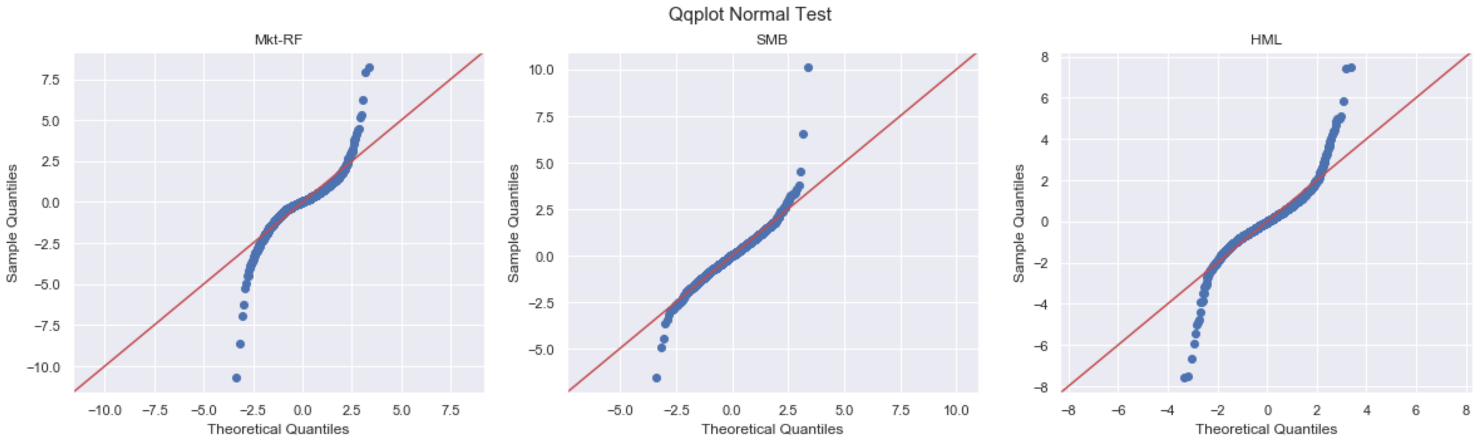


Figure 6: QQ-plot

Problem 5

(e)Consider a multi-factor model for the sector ETFs using the Fama French return data as variables. That is:

$$r_{i,t} = \beta_1 r_{\text{mkt},t} + \beta_2 r_{\text{size},t} + \beta_3 r_{\text{btm},t} + \epsilon_{i,t}$$

NOTE: This is the 3-factor Fama-French model. You may also choose to use a different set of Fama-French factors available on Ken French's website. For each sector ETF, compute it's to the Fama-French factors using the above model. Compute the for the entire historical period and also compute rolling 90-day 's. Are these 's more consistent than the single factor model 's that you obtained in your first homework?

	SPY	XLB	XLE	XLF	XLI	XLK	XLP	XLU	XLV	XLY
Mkt-RF	0.009815	0.010467	0.011136	0.011091	0.010339	0.011169	0.006815	0.006605	0.008876	0.009892
SMB	-0.00126	0.000816	0.000888	-0.0007	0.0003	-0.00185	-0.00362	-0.00414	-0.00185	0.000501
HML	7.2E-05	0.002643	0.007394	0.00736	0.002689	-0.00371	-0.00067	0.000841	-0.00299	-0.00106

Figure 7: whole period beta

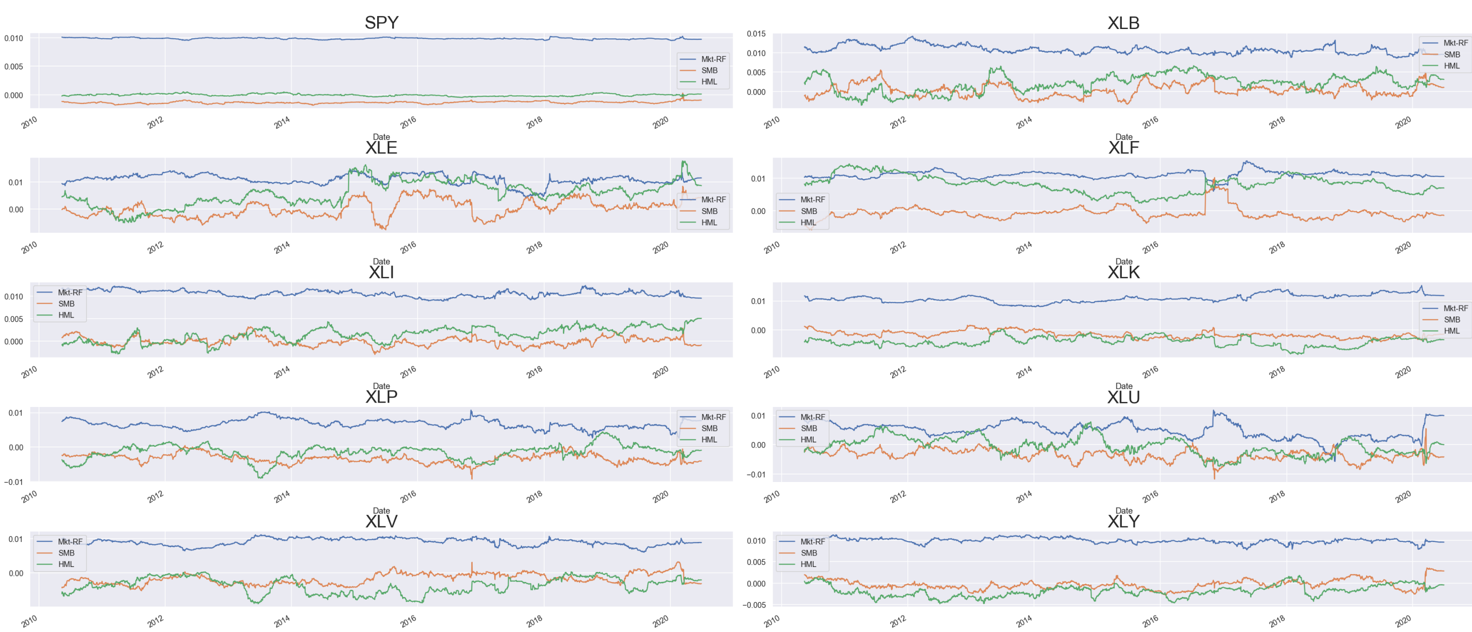


Figure 8: 90-days rolling beta

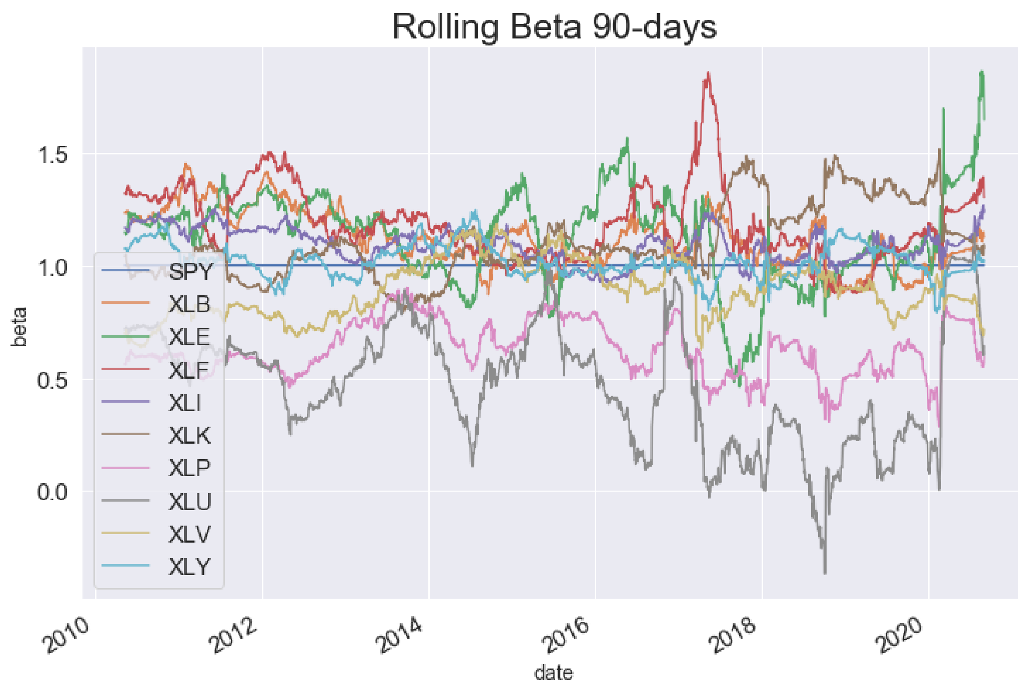
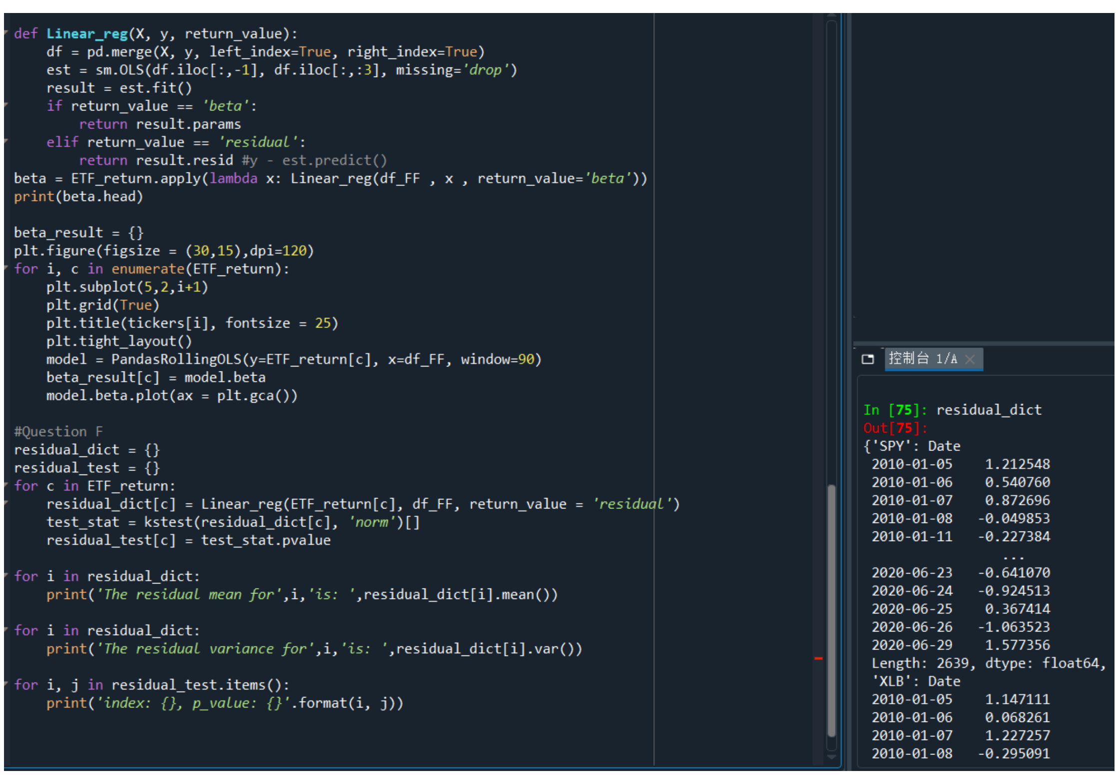


Figure 9: 90-days rolling beta

Obviously, these ’s more consistent than the single factor model ’s that I obtained in first homework.

Problem 6

(f) Compute the daily residuals *epsilon* **i,t in (1) for each sector ETF. What is their mean and variance? Do they appear to be normal? What does this tell you about the appropriateness of the model? Can you think of any other tests of the residuals that would help you judge the model?**



```
def Linear_reg(X, y, return_value):
    df = pd.merge(X, y, left_index=True, right_index=True)
    est = sm.OLS(df.iloc[:,-1], df.iloc[:,:3], missing='drop')
    result = est.fit()
    if return_value == 'beta':
        return result.params
    elif return_value == 'residual':
        return result.resid #y - est.predict()
beta = ETF_return.apply(lambda x: Linear_reg(df_FF , x , return_value='beta'))
print(beta.head)

beta_result = {}
plt.figure(figsize = (30,15),dpi=120)
for i, c in enumerate(ETF_return):
    plt.subplot(5,2,i+1)
    plt.grid(True)
    plt.title(tickers[i], fontsize = 25)
    plt.tight_layout()
    model = PandasRollingOLS(y=ETF_return[c], x=df_FF, window=90)
    beta_result[c] = model.beta
    model.beta.plot(ax = plt.gca())

#Question F
residual_dict = {}
residual_test = {}
for c in ETF_return:
    residual_dict[c] = Linear_reg(ETF_return[c], df_FF, return_value = 'residual')
    test_stat = kstest(residual_dict[c], 'norm')[]
    residual_test[c] = test_stat.pvalue

for i in residual_dict:
    print('The residual mean for',i,'is: ',residual_dict[i].mean())

for i in residual_dict:
    print('The residual variance for',i,'is: ',residual_dict[i].var())

for i, j in residual_test.items():
    print('index: {}, p_value: {}'.format(i, j))
```

```
控制台 1/A

In [75]: residual_dict
Out[75]:
{'SPY': Date
 2010-01-05    1.212548
 2010-01-06    0.540760
 2010-01-07    0.872696
 2010-01-08   -0.049853
 2010-01-11   -0.227384
                 ...
 2020-06-23   -0.641070
 2020-06-24   -0.924513
 2020-06-25    0.367414
 2020-06-26   -1.063523
 2020-06-29    1.577356
 Length: 2639, dtype: float64,
 'XLB': Date
 2010-01-05    1.147111
 2010-01-06    0.068261
 2010-01-07    1.227257
 2010-01-08   -0.295091
```

Figure 10: daily residual

```
The residual mean for SPY is:  -0.02614363262991394
The residual mean for XLB is:  -0.020897788599513238
The residual mean for XLE is:  -0.009722457394575678
The residual mean for XLF is:  -0.020028911799911574
The residual mean for XLI is:  -0.021974014817087287
The residual mean for XLK is:  -0.02256094144129758
The residual mean for XLP is:  -0.02776897592795237
The residual mean for XLU is:  -0.028446461573644808
The residual mean for XLV is:  -0.025509373093314837
The residual mean for XLY is:  -0.02567456901032466
The residual variance for SPY is:  0.3646052795586081
The residual variance for XLB is:  0.3395304978549624
The residual variance for XLE is:  0.2909750431404375
The residual variance for XLF is:  0.24987477639502664
The residual variance for XLI is:  0.31585872284710703
The residual variance for XLK is:  0.2718343510417592
The residual variance for XLP is:  0.36339259895946036
The residual variance for XLU is:  0.3644190275847817
The residual variance for XLV is:  0.32014987870297046
The residual variance for XLY is:  0.3563835311287331
```

```
In [78]: residual_test
Out[78]:
{'SPY': 2.3291396173261598e-70,
 'XLB': 1.7424881135510854e-70,
 'XLE': 1.5808117648281152e-78,
 'XLF': 1.8438732887214476e-123,
 'XLI': 6.960268743987578e-73,
 'XLK': 2.047453092246781e-93,
 'XLP': 5.990636949528242e-74,
 'XLU': 1.0604284183379211e-72,
 'XLV': 3.6195561830954894e-87,
 'XLY': 1.417102206044108e-76}
```

By using the p-value of kolmogorov-smirnov test, we can conclude that the residuals are not very normally distributed, so the model may not apply to the daily returns. Except Kstest, we can use like QQ-plot to test whether the results are normally distributed.

Task 2 : Exotic Option Pricing via Simulation

Problem 1

Consider again a one-year fixed strike lookback option which enables the buyer to choose the point of exercise for the option at its expiry. The Bachelier model can be written as:

$$dS_t = rdt + \sigma dW_t$$

Assume that r = 0, S 0 = 100 and = 10.0.
(a) Generate a series of normally distributed random numbers and use these to generate simulated paths for the underlying asset.

Set simulated time as 10000, step as 100. The plot of the simulated paths generated by Bachelier model is as follows:

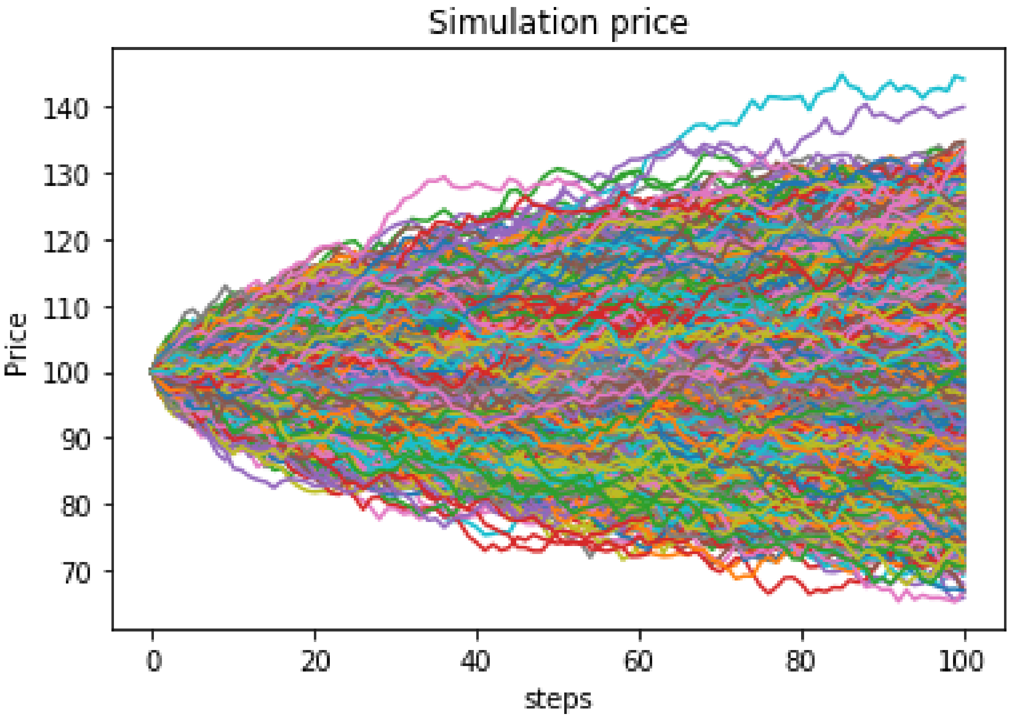


Figure 11: QQ-plot

Problem 2

(b) Plot a histogram of the ending values of the asset price along the simulated paths. Are the ending values of your simulated paths normally distributed? Check using your favorite normality test.

From the figures below, we can observe that the ending value is approximately normally distributed, Since most of the data coincides with the red 45 degree line and p value of Kolmogorov-Smirnov test for ending value of simulations: 0.5078174714645691.

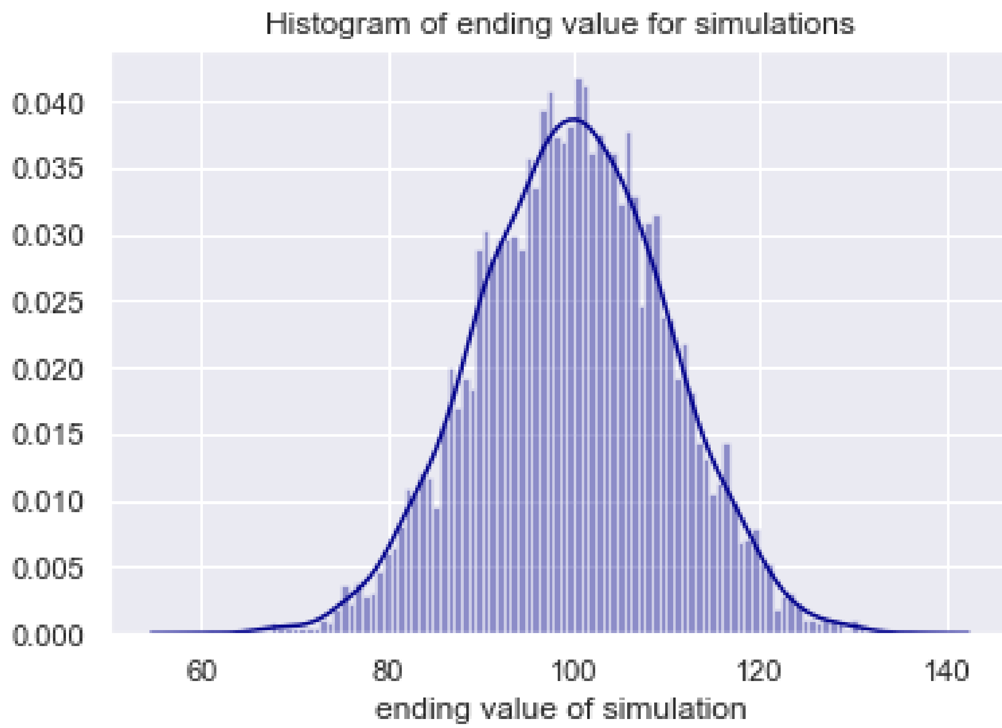


Figure 12: ending value

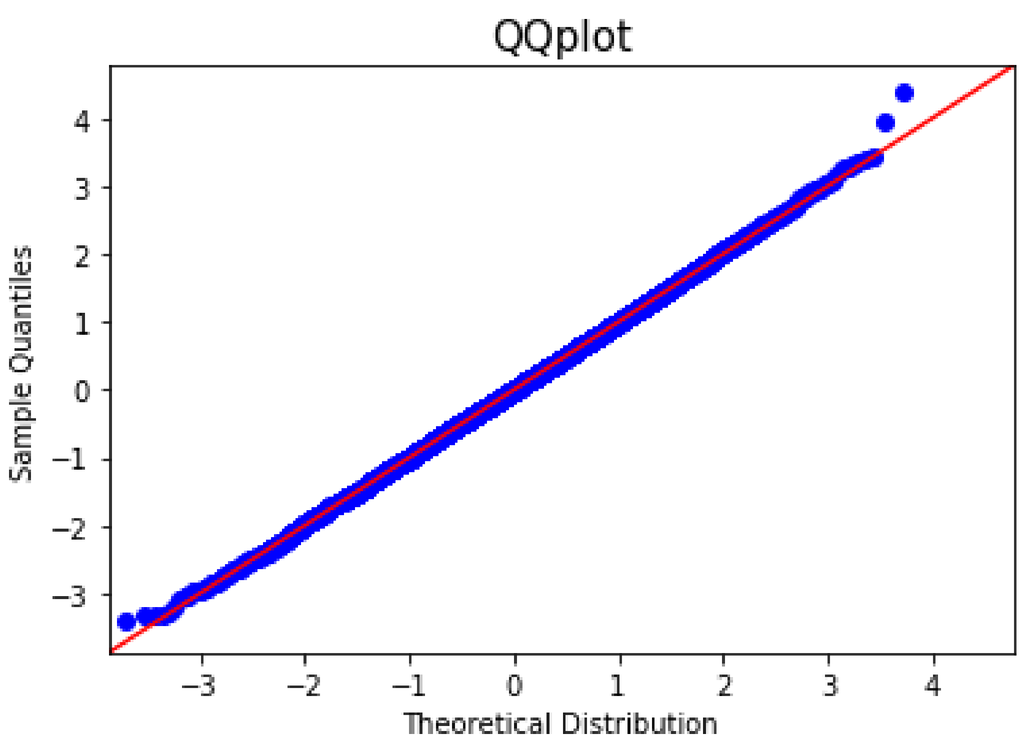


Figure 13: ending qqplot

Problem 3

(c) Calculate a simulation approximation to the price of a Lookback put option with strike 100 under the Bachelier model. Compare the price of the lookback option in the Bachelier model to the Black-Scholes model price obtained in HW1.

The price of a Lookback put option with : 7.5014913589047465 which is significantly less than the price of a Lookback put option with BSM: 9.94764496602258 because in this case the volatilty part of stock price is constant and is not amplified by St .

Problem 4

(d)Calculate the delta of the lookback option using finite differences as discussed in class. That is:

$$\Delta \approx \frac{c_0\left(S_0+\epsilon\right)-c_0\left(S_0-\epsilon\right)}{2\epsilon}$$

Try for several values of ϵ and plot the calculated Δ against the choice of ϵ. Comment on what you think is the optimal value of ϵ and what values lead to the largest amounts of error.

we choose positive epsilon from 0.01 to 10 here and the graph for delta as the following:

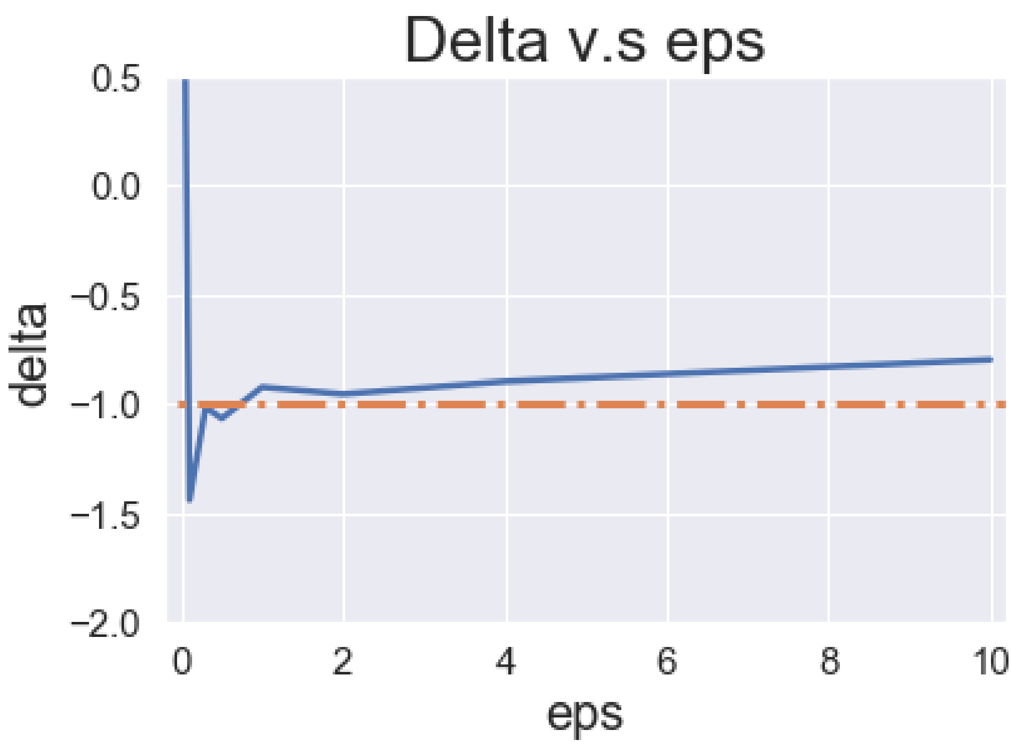


Figure 14: delta

From the graph, we can see that when ϵ is large, the value of is near -1. The reason is that the volatility of stock price is very limited in Bachelier model so the minimum stock price during the period is highly correlated with the initial stock price and we have extreme numbers around very small ϵ because c0 is not a smooth function. The value smaller than 1 will lead to larger errors and larger intervals still induce larger errors, and the range from 2 to 4 is much more precise than the other values.