Python初級數據分析員證書

（六）數據分析 及可視化專案

13. 數據分析專案 Demo1

Review

- Statistics
- Hypothesis testing
- Algebra
- Linear regression
- Propositional logic
- Python
- SQL
- Pandas, NumPy, SciPy
- Data Visualization, Matplotlib, Seaborn, Plotly
- Dashboard Visualization, Business Intelligence
- Storytelling

Chapter Summary

- DF Cleaning & Trimming
- Adding Features
- Outliers & Anomaly
- Distribution & QQ plot
- Log return and Volatility
- Autocorrelation
- Sharpe Ratio

Import data

財務數據分析是最常被研究的領域之一。通過學習工作流，我們也可以理解時間序列主題。今天我們探討基本的技術分析。

```
1  import yfinance as yf
2  import pandas as pd
3  import numpy as np
4  from decimal import ROUND_HALF_UP, Decimal
5  from tqdm import tqdm
6  import scipy.stats as scs
7  import statsmodels.api as sm
8  import plotly.express as px
9  import plotly.graph_objects as go
10 from plotly.subplots import make_subplots
11 import matplotlib.pyplot as plt
12 import seaborn as sns
13
14 df_raw = yf.download("^HSI", start="1987-01-01", end="2023-01-01")
```

```
[**********100%**********]  1 of 1 completed
```

Cleaning and trimming

我們在這項研究中刪除了這 3 個不必要的列，但這不代表它們在其他列中毫無用處。

```
1 data = df_raw.copy().drop(columns=['Adj Close','Volume']).dropna()
2 data.sample(3)
```

Date	Open	High	Low	Close
1995-11-06	9856.099609	9882.400391	9733.000000	9736.099609
2017-01-16	22895.359375	22908.859375	22657.349609	22718.150391
2022-09-26	17781.869141	18077.640625	17727.400391	17855.140625

Add features

在大多數情況下，原始數據集可能很簡單，不到 10 列。我們可以添加其他相關列，i.e. RSI, Stochastics, 為了進一步研究並考慮到machine learning.

```
def add_features(feats):
    for i in [20, 60]:
        feats[f"MA_{i}"] = (feats["Close"].rolling(i).mean())

    feats['simple_rtn'] = feats.Close.pct_change()
    feats['log_rtn'] = np.log(feats.Close/feats.Close.shift(1))
    feats["volatility_20"] = (np.log(feats["Close"]).diff().rolling(20).std())
    feats['log_rtn_20'] = np.log(feats.Close/feats.Close.shift(20))
    return feats

data = add_features(data).dropna()
```

Moving Average

移動平均線是分析特定時期趨勢的概念。由於每天或每周的變化對於決策來說可能是模糊的。移動平均線也適用於財務會計，即預測銷售額和現金流量。

幾十年來，投資者使用多條移動平均線以及均值回歸策略來預測股市趨勢。In Pandas, DF.rolling 提供 rolling window 計算。

```
data["MA_20"] = data["Close"].rolling(20).mean()
```

此小段code用於計算 之前20 天收盤的平均值。

Simple return and log return

Simple returns: 它們聚合了資產;投資組合的簡單回報是投資組合中單個資產的回報的加權總和。簡單退貨的定義如下：

$$R_t = (P_t - P_{t-1})/P_{t-1} = P_t/P_{t-1} - 1$$

```
data['simple_rtn'] = data.Close.pct_change()
```

Log returns: 它們會隨著時間的推移而聚集;借助示例，可以更容易地理解 - 給定月份的日誌返回是該月內天數的日誌返回量之和。日誌返回定義為:

$$r_t = log(P_t/P_{t-1}) = log(P_t) - log(P_{t-1})$$

```
data['log_rtn'] = np.log(data.Close/ data.Close.shift(1))
```

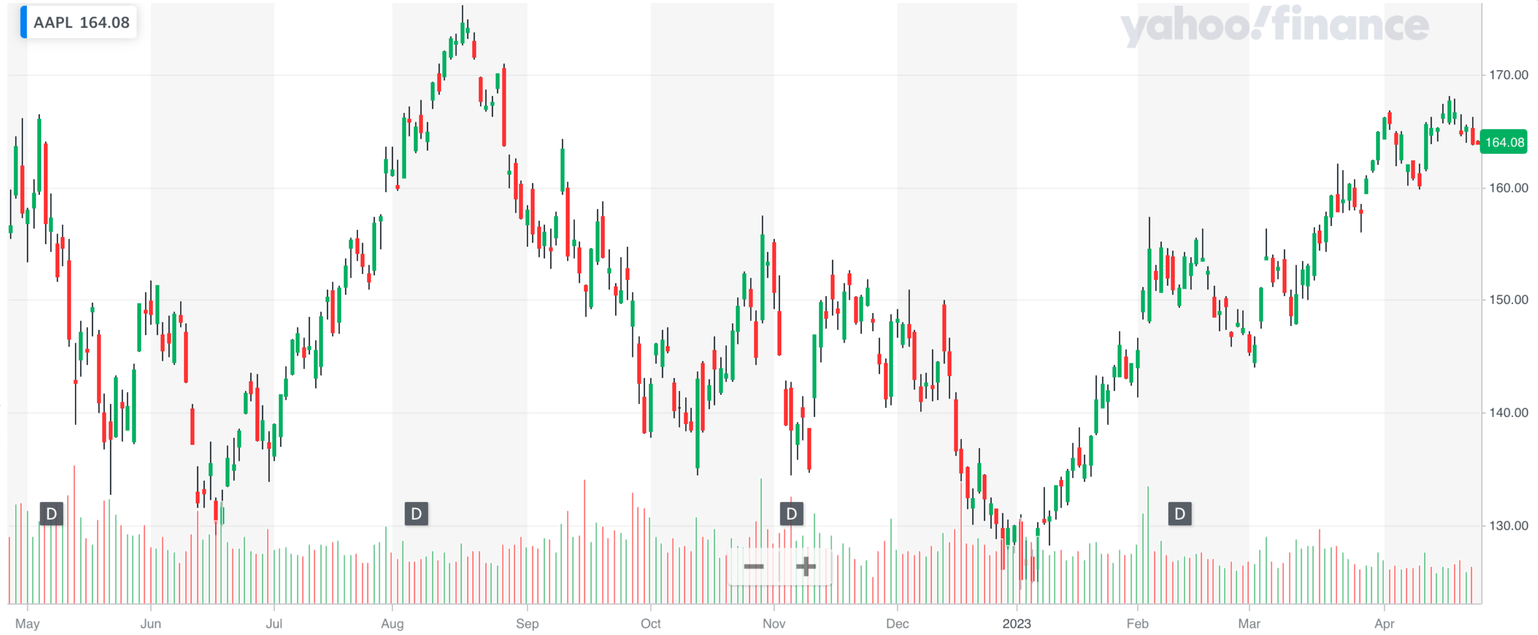
Simple return and log return

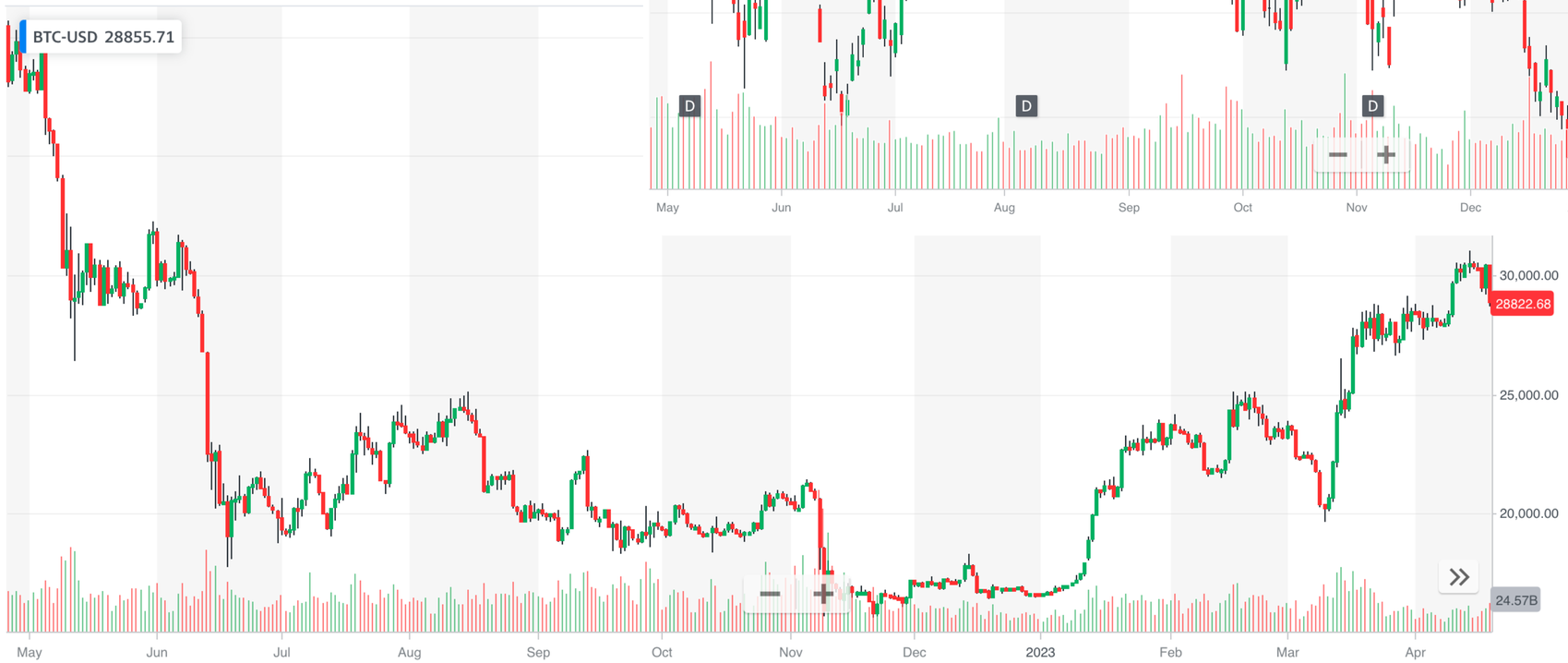
每日/日內的Simple and Log Return兩者的區別非常小，但是，一般Log Return的價值小於Simple Return。 (Simple Return 使人們感到高興，因為它增加得更多，下降得更少。但它並沒有一目了然地反映出本金價值回報。)

Period	Price	Simple Return	Log Return
Year 1	100	-	-
Year 2	200	100%	69%
Year 3	100	−50%	−69%

Measuring risk

Which asset do you think it is more risky?





Measuring risk

衡量資產風險的最簡單方法是衡量其Standard Deviation or volatility. 同樣的想法也適用於國家GDP, 公司月銷售額、匯率等. 想像一下，一種貨幣波動巨大，你會考慮與其國家進行貿易和業務往來嗎?

20 days volatility

```
data["volatility_20"] = (np.log(data["Close"]).diff().rolling(20).std())
```

20 days log return

```
data['log_rtn_20'] = np.log(data.Close/feats.Close.shift(20))
```

New DataFrame with added features

我們現在有了用於進一步分析的 DF。

```
1 data
```

Date	Open	High	Low	Close	MA_20	MA_60	simple_rtn	log_rtn	volatility_20	log_rtn_20
1987-03-30	2774.899902	2774.899902	2774.899902	2774.899902	2778.260010	2698.976664	-0.008504	-0.008540	0.017323	-0.042129
1987-03-31	2713.800049	2713.800049	2713.800049	2713.800049	2766.995007	2701.871663	-0.022019	-0.022265	0.017369	-0.079754
1987-04-01	2695.899902	2695.899902	2695.899902	2695.899902	2757.245007	2704.263330	-0.006596	-0.006618	0.017132	-0.069836
1987-04-02	2709.399902	2709.399902	2709.399902	2709.399902	2752.795007	2706.354997	0.005008	0.004995	0.015787	-0.032321
1987-04-03	2680.000000	2680.000000	2680.000000	2680.000000	2746.865002	2707.569995	-0.010851	-0.010910	0.015916	-0.043303

New DataFrame with added features

看看 DF 基本統計數據。

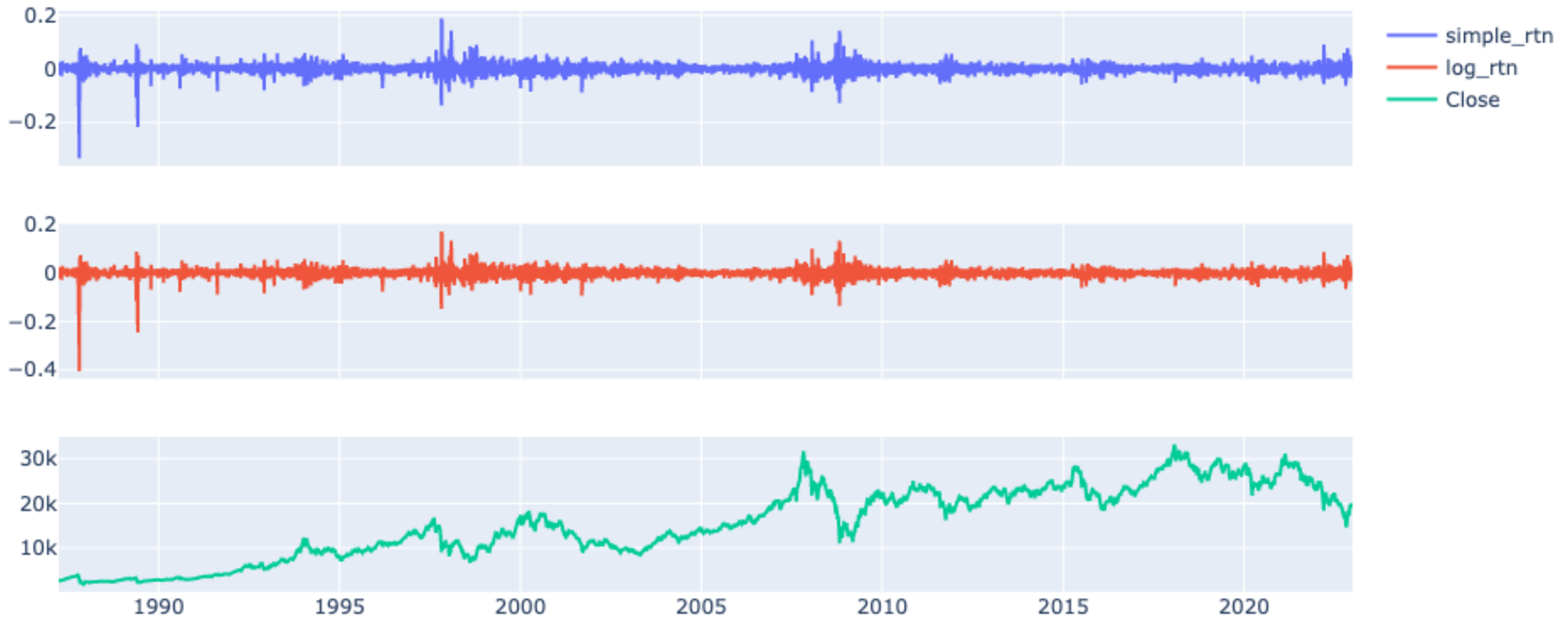
```
1 data.describe()
```

	Open	High	Low	Close	MA_20	MA_60	simple_rtn	log_rtn	volatility_20	log_rtn_20
count	8831.000000	8831.000000	8831.000000	8831.000000	8831.000000	8831.000000	8831.000000	8831.000000	8831.000000	8831.000000
mean	15618.668241	15720.067730	15497.632051	15612.443663	15594.402424	15560.239481	0.000355	0.000221	0.013847	0.004399
std	8078.948998	8121.273706	8020.202788	8070.968662	8068.342257	8065.845157	0.016228	0.016450	0.008838	0.075487
min	1950.500000	1950.500000	1894.900024	1894.900024	2085.339990	2220.491661	-0.333304	-0.405420	0.003966	-0.700236
25%	9494.599609	9569.250000	9416.470215	9493.165039	9471.997485	9499.276693	-0.006752	-0.006775	0.008813	-0.032728
50%	15075.780273	15192.919922	14936.900391	15070.559570	15058.706934	15019.605973	0.000551	0.000551	0.011381	0.009794
75%	22697.649414	22808.315430	22541.815430	22668.540039	22601.260791	22643.067643	0.007985	0.007953	0.015852	0.048447
max	33335.480469	33484.078125	32897.039062	33154.121094	32213.191504	31275.259961	0.188236	0.172470	0.100805	0.328689

First Plot

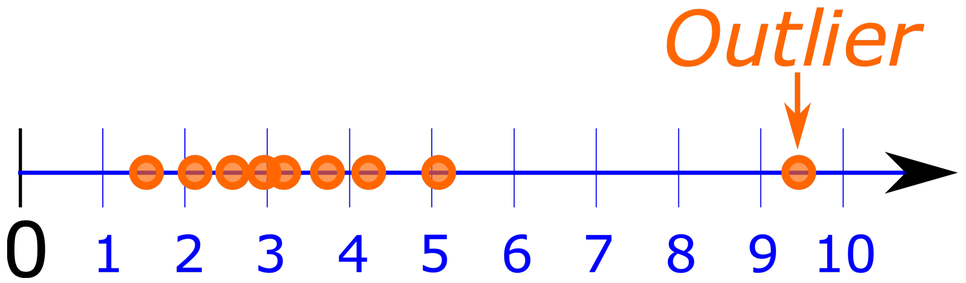
```
colors=px.colors.qualitative.Plotly

fig = make_subplots(rows=3, cols=1, shared_xaxes=True)
for i, j in enumerate([data.simple_rtn, data.log_rtn, data.Close]):
    fig.add_trace(go.Scatter(x=data.index, y=j, mode='lines',
                             name=j.name, marker_color=colors[i]), row=i+1, col=1)
fig.show()
```

從圖中我們瞭解到，多年來有一些極端數據。



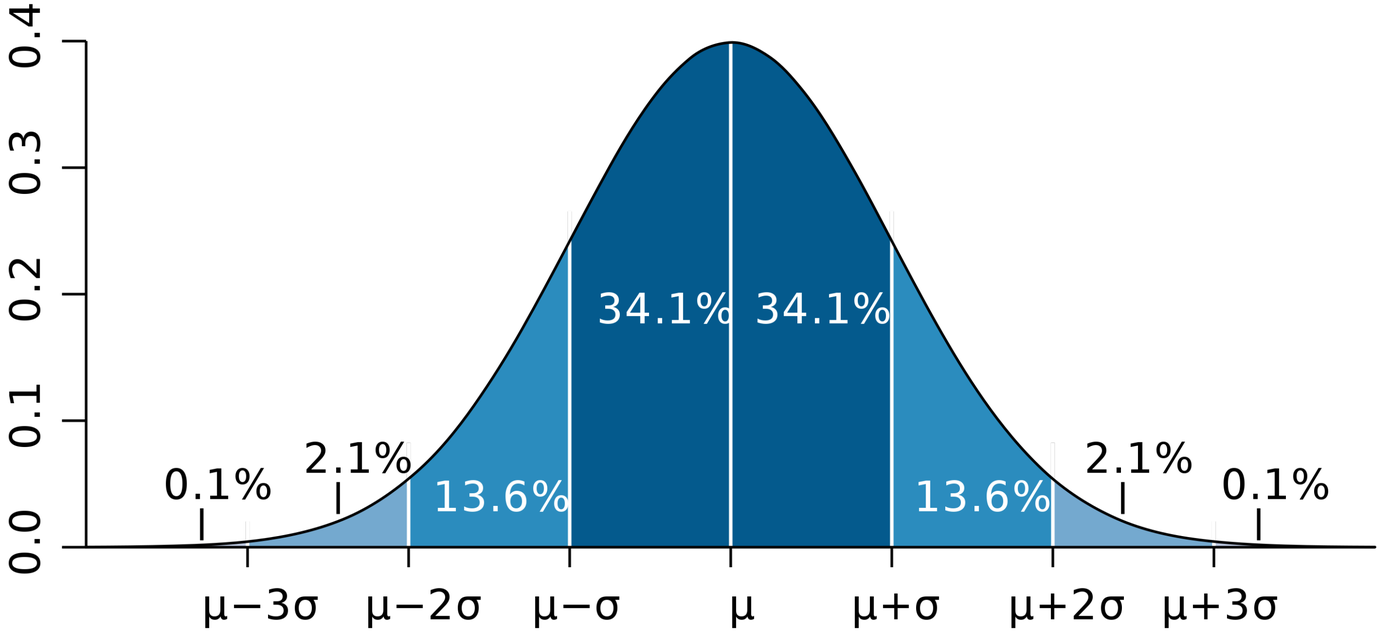
Investigate outliers

在統計學中，an outlier 是一個與大多數數據點有顯著差異的數據點。



在樣本分佈中，我們可以將異常值定義為

$> \mu+3\sigma$ 或 $< \mu-3\sigma$



Outliers Detection

Create a DF and function to detect outliers.

```
df_rolling = data[['simple_rtn']].rolling(20).agg(['mean', 'std'])
df_rolling.columns = df_rolling.columns.droplevel()
df_outliers = data.join(df_rolling)
```

```
def identify_outliers(row, n_sigmas=3):
    x = row['simple_rtn']
    mu = row['mean']
    sigma = row['std']
    if (x > mu + 3 * sigma) | (x < mu - 3 * sigma):
        return 1
    else:
        return 0
```

Outliers Detection

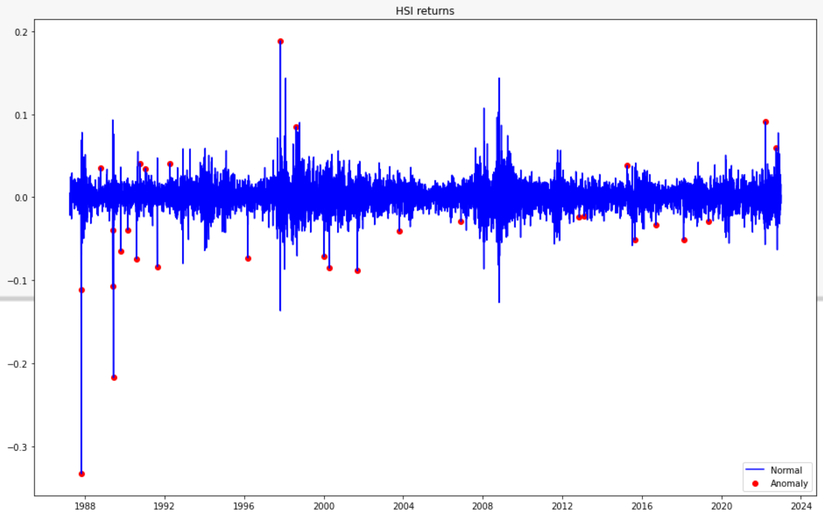
Maintain a outliers list for further study.

```
1 df_outliers['outlier'] = df_outliers.apply(identify_outliers,  axis=1)
2 outliers = df_outliers.loc[df_outliers['outlier'] == 1, ['simple_rtn']]
3 outliers.sample(5)
```

Date	simple_rtn
2012-11-08	-0.024115
2013-02-05	-0.022651
2003-10-23	-0.040973
2022-10-05	0.059045
1991-08-19	-0.083928

Plotting Anomaly

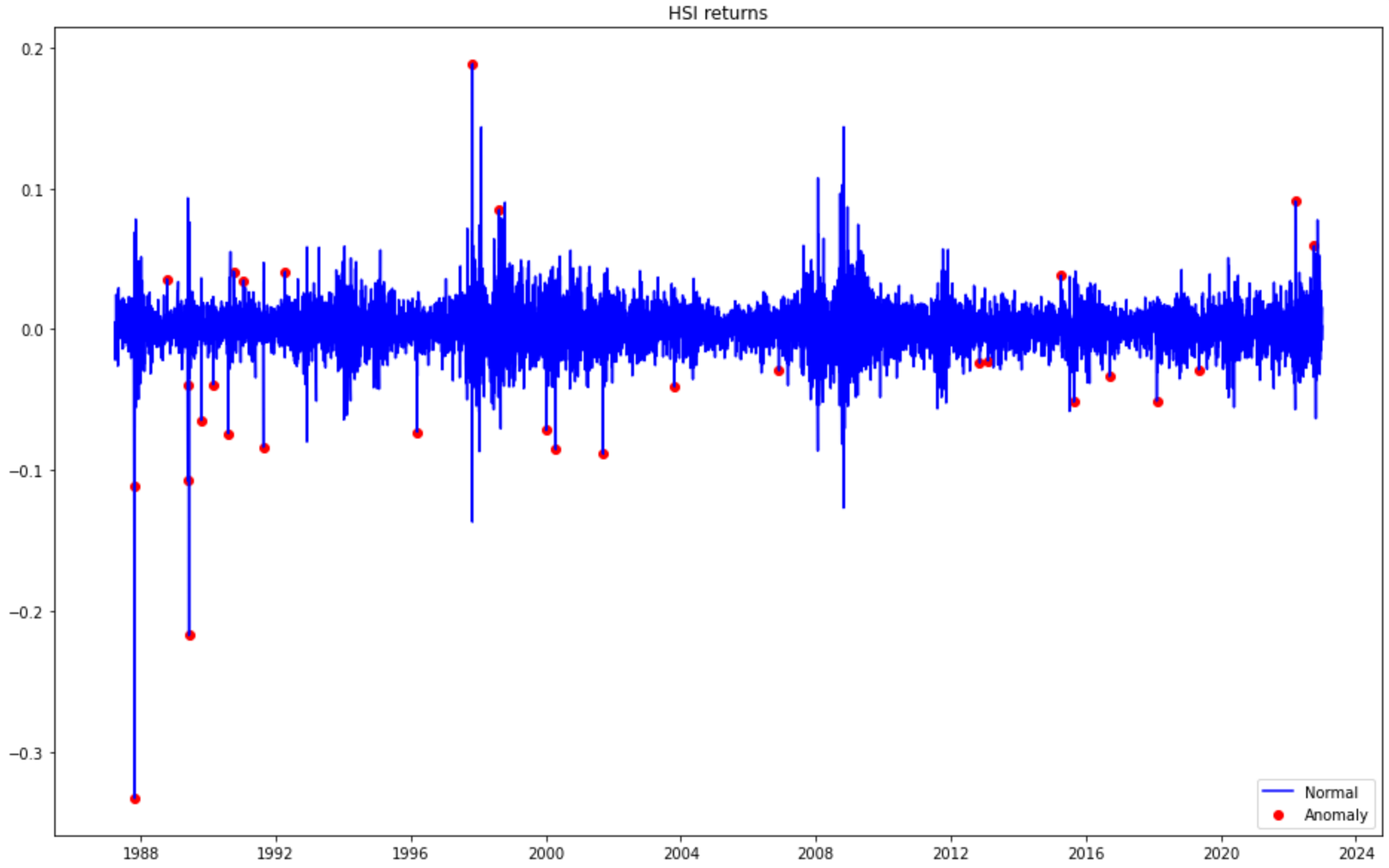
```
1  fig, ax = plt.subplots(figsize=(16,10))
2
3  ax.plot(df_outliers.index, df_outliers.simple_rtn,
4          color='blue', label='Normal')
5  ax.scatter(outliers.index, outliers.simple_rtn,
6             color='red', label='Anomaly')
7  ax.set_title("HSI returns")
8  ax.legend(loc='lower right')
9
10 plt.show()
```



Plotting Anomaly

Question:

- 我們應該使用這些異常值來構建回歸模型嗎？
- 我們應該在 DF 中擦除這些異常值嗎？



Dealing anomaly

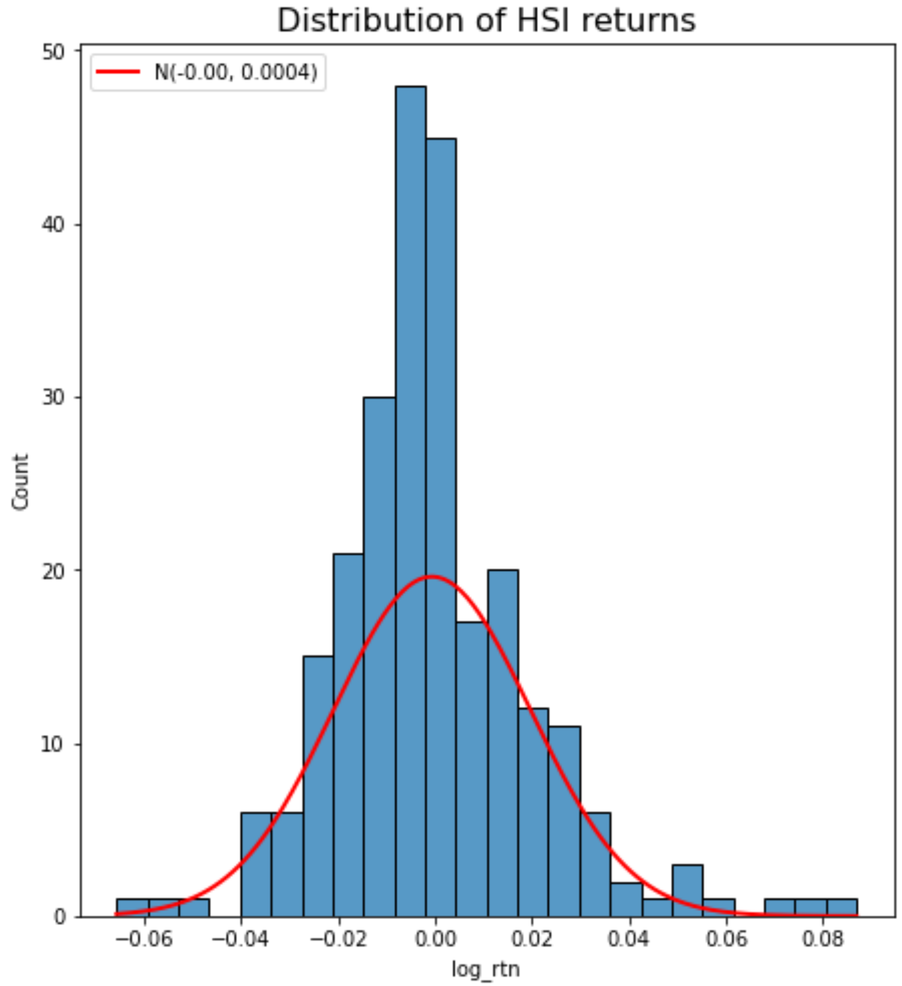
對於嚴肅的研究：

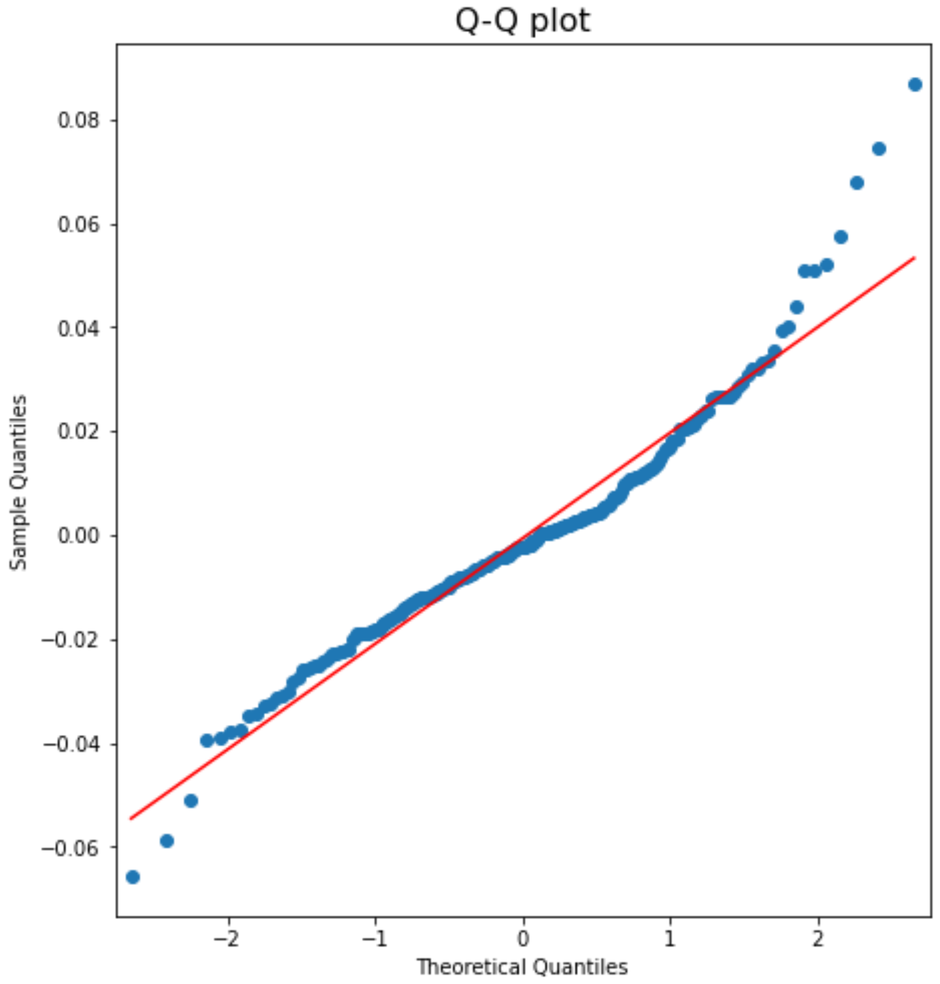
我們應獨立調查這些異常值事件。

- 異常值可能會嚴重影響多數數據，因此不應將其保留為樣本變數來構建回歸模型。
- 研究這些異常數據可能更為重要。著名對沖基金經理（Michael Burry）做到了。

Distribution and QQ plot

To analyse log return and its quantile, we may plot these two graphs.





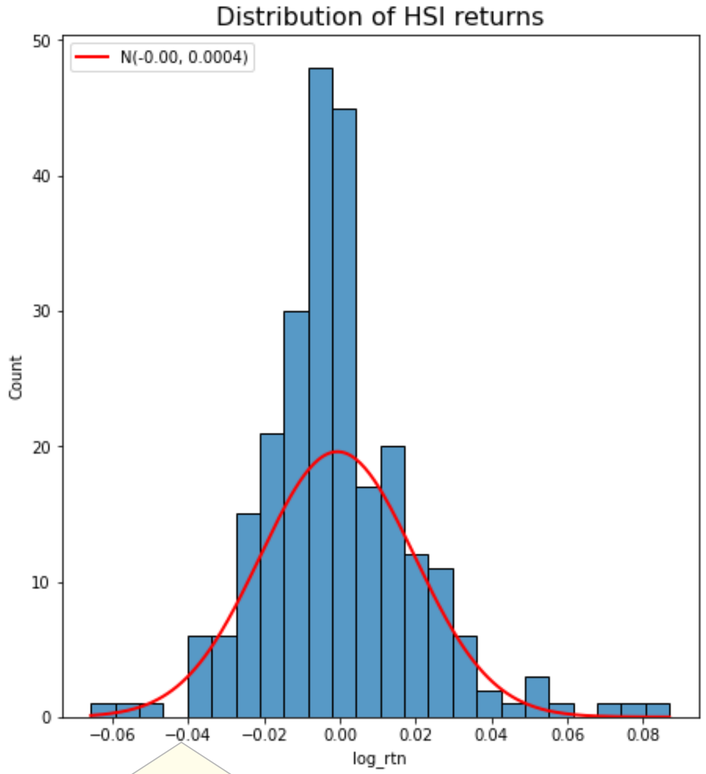
Distribution and QQ plot in latest 250 days

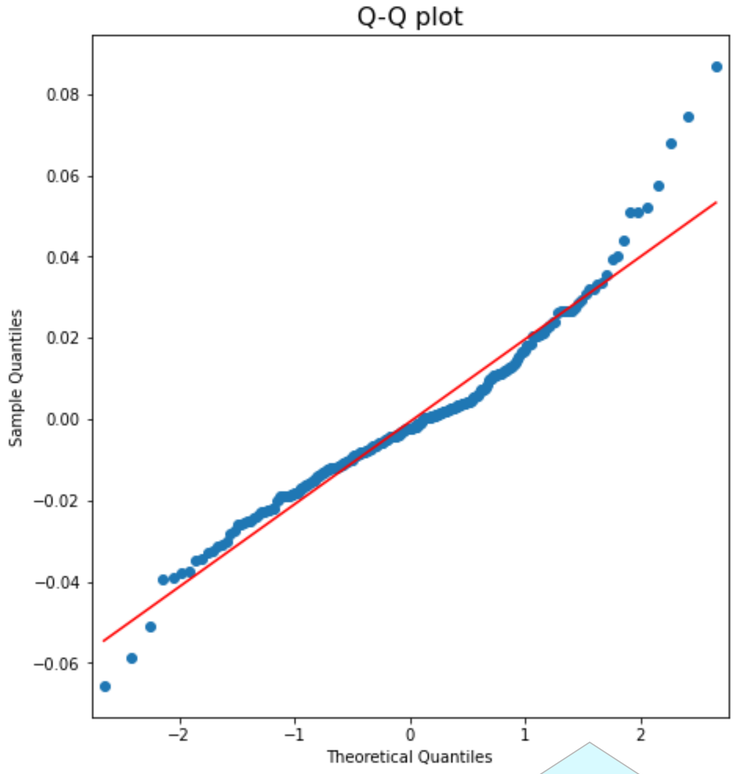
為了將對數返回分佈與正態分佈進行比較，我們可以使用對數返回的平均值、標準差、樣本大小來創建一個概率密度函數 PDF。

```
r_range = np.linspace(min(data.iloc[-250:].log_rtn), max(data.iloc[-250:].log_rtn), num=250)
mu = data.iloc[-250:].log_rtn.mean()
sigma = data.iloc[-250:].log_rtn.std()
norm_pdf = scs.norm.pdf(r_range, loc=mu, scale=sigma)
```

```
fig, ax = plt.subplots(1, 2, figsize=(16, 8))

# histogram
ax[0].set_title('Distribution of HSI returns', fontsize=16)
sns.histplot(data=data.iloc[-250:], x='log_rtn', kde=False, ax=ax[0])
ax[0].plot(r_range, norm_pdf, 'r', lw=2, label=f'N({mu:.2f}, {sigma**2:.4f})')
ax[0].legend(loc='upper left');

# Q-Q plot
qq = sm.qqplot(data.iloc[-250:].log_rtn.values, line='s', ax=ax[1])
ax[1].set_title('Q-Q plot', fontsize = 16)

plt.show()
```

Distribution and QQ plot





The red line on distribution plot is PDF we generated.

QQ plot also assess the normality of the log return. The red line is the fit line.

Descriptive Statistics

Generate stats figures.

```
1  print('---------- Descriptive Statistics ----------')
2  print('Range of dates:', min(data.iloc[-250:].index.date), '-', max(data.iloc[-250:].index.date))
3  print('Number of observations:', data.iloc[-250:].shape[0])
4  print(f'Mean: {data.iloc[-250:].log_rtn.mean():.4f}')
5  print(f'Median: {data.iloc[-250:].log_rtn.median():.4f}')
6  print(f'Min: {data.iloc[-250:].log_rtn.min():.4f}')
7  print(f'Max: {data.iloc[-250:].log_rtn.max():.4f}')
8  print(f'Standard Deviation: {data.iloc[-250:].log_rtn.std():.4f}')
9  print(f'Skewness: {data.iloc[-250:].log_rtn.skew():.4f}')
10 print(f'Kurtosis: {data.iloc[-250:].log_rtn.kurtosis():.4f}')
```

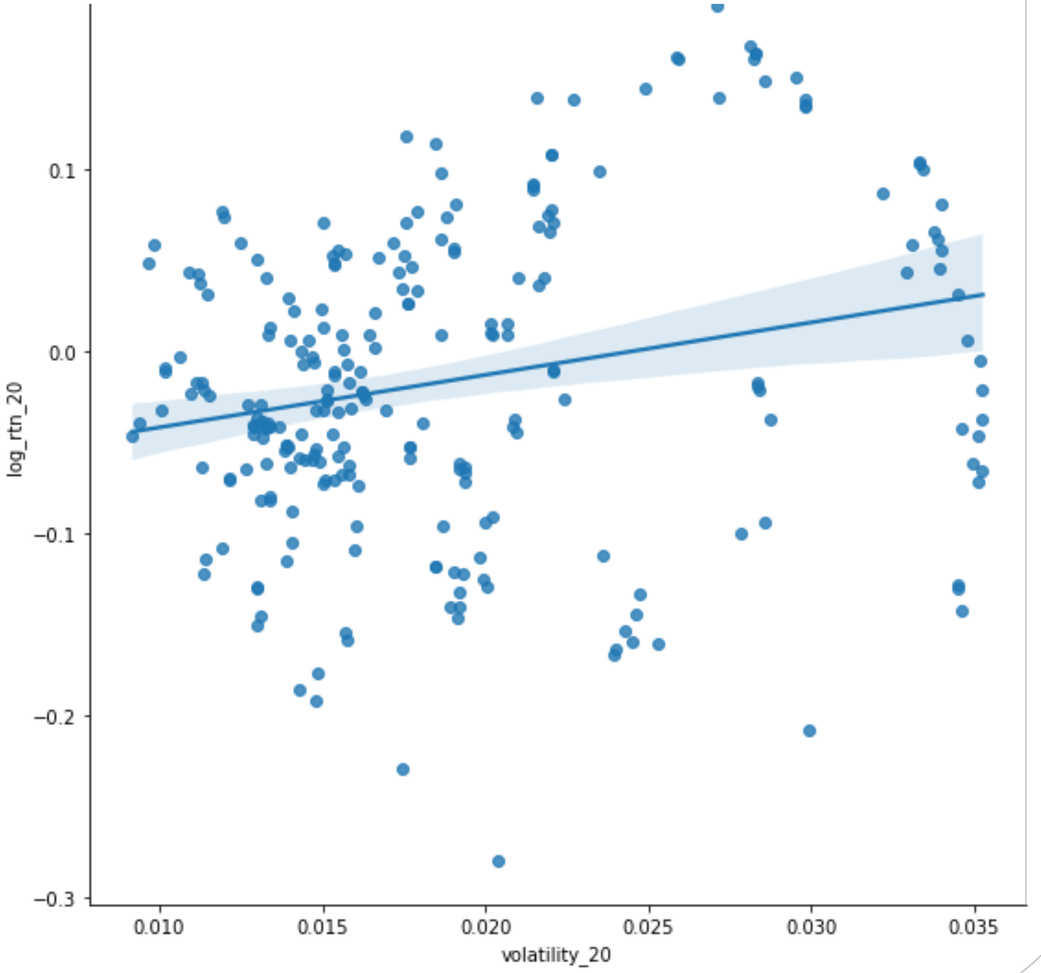
```
---------- Descriptive Statistics ----------
Range of dates: 2021-12-28 - 2022-12-30
Number of observations: 250
Mean: -0.0006
Median: -0.0022
Min: -0.0657
Max: 0.0869
Standard Deviation: 0.0204
Skewness: 0.7131
Kurtosis: 2.5186
```

Log return vs Volatility

```
1  sns.lmplot(data=data.iloc[-250:], x="volatility_20", y="log_rtn_20", height=8)
```

我們經常聽到有人說「*風險越大，回報越高」。真的嗎？*為了證明它在資產上是正確的，它應該是一條乾淨的線性回歸線，具有較小的標準差。

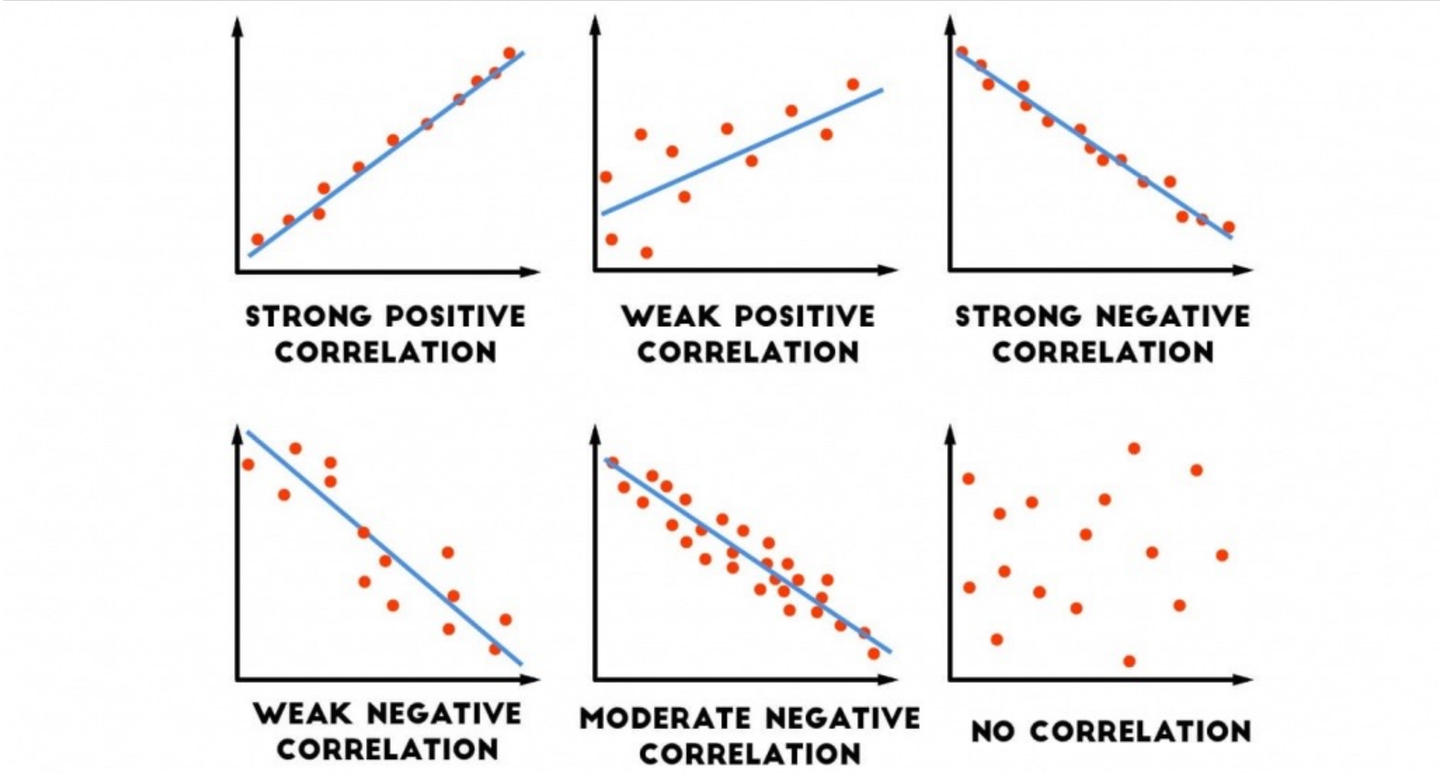
R^2 (correlation) 應該足夠大。



Log return vs Volatility

```
1 print("Correlation between return and volatility:",
2       data.iloc[-250:].volatility_20.corr(data.iloc[-250:].log_rtn_20))
```

```
Correlation between return and volatility: 0.22993256556828845
```



Autocorrelation

根據我們的共同觀察：

- 如果今天下雨，明天可能會多雲。
- 如果一個國家本月在出口方面獲得正增長，那麼下個月可能會有正增長。

Autocorrelation 是調查當前和過去的事物關係。

Correlation 是調查同一時間序列中 2 個事項的關係。

Autocorrelation

An autocorrelation of +1 代表著完美 positive correlation（在一個時間序列中看到的增加會導致另一個時間序列的相應增加）表示當前和過去。

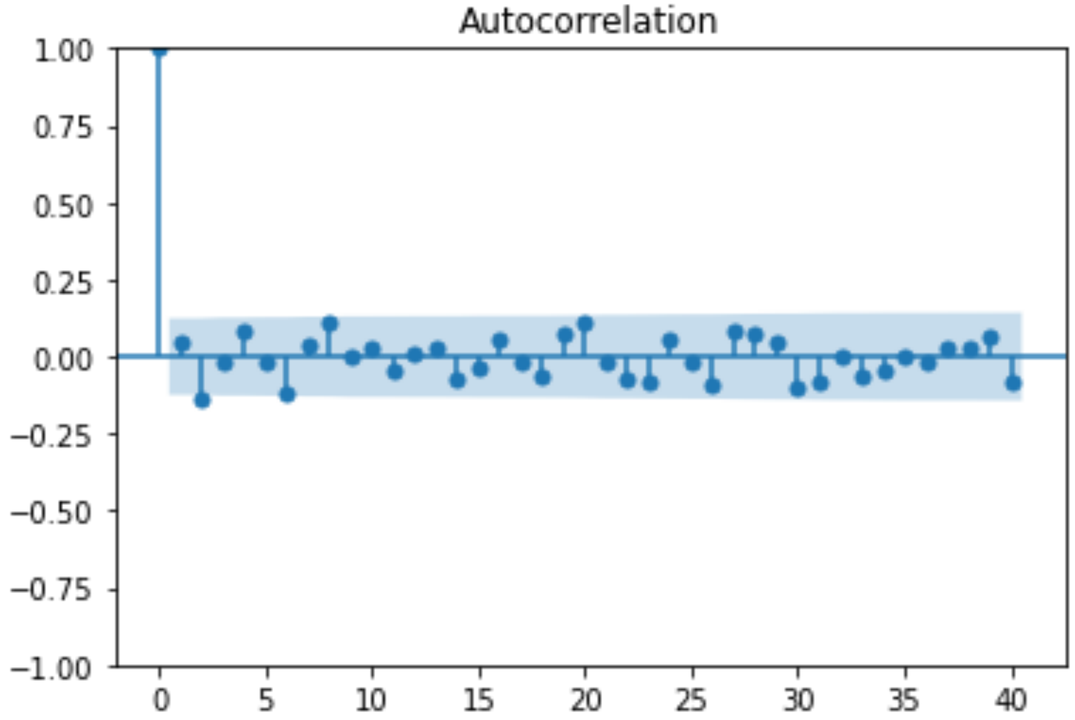
-1 是完全否定該相關。

Autocorrelation

```
1  import statsmodels.tsa.api as smt
2  N_LAGS = 40
3  SIGNIFICANCE_LEVEL = 0.05
```

```
1  acf = smt.graphics.plot_acf(data.iloc[-250:].log_rtn,
2                              lags=N_LAGS, alpha=SIGNIFICANCE_LEVEL)
```

只有少數值位於置信區間之外（我們不查看滯後 0），可以考慮statistically significant. 我們可以假設我們已經驗證了存在no autocorrelation 在log returns series.



Autocorrelation

為了計算實際數位，我們可以使用df.series.autocorr() function in Pandas. 這個數字告訴我們，數量不多 correlated, 但勉強保持正值。

```
1 print("Autocorrelation of log return inlast 250-days: ",
2       data.iloc[-250:].log_rtn.autocorr(lag=1))
```

```
Autocorrelation of log return inlast 250-days:  0.05060587450032967
```

```
1 print("Autocorrelation of log return in all years: ",
2       data.log_rtn.autocorr(lag=1))
```

```
Autocorrelation of log return in all years:  0.018825447245086 62
```

Autocorrelation

Open Question:

- 如果自相關比率低，我們還應該相信移動平均線策略嗎？
- 如果是這樣，歷史不能預測未來嗎？
- 在什麼情況下，這個比例可能會更大？

Compare stocks return and risk

Stock	Annual Return
HSBC	3.56%
CITIC	2.04%

如果您只能選擇一隻股票進行投資，如何公平地比較它們的回報與風險？

Sharpe Ratio

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$$

where:

R_p = return of portfolio

R_f = risk-free rate

σ_p = standard deviation of the portfolio's excess return

Sharpe Ratio 是單位風險的淨回報率。它可以以簡單的方式公平地比較資產。但在一些罕見的情況下，比如股票被停牌並且沒有交易，SD會非常小，那麼夏普比率就會變得毫無意義的巨大。

Sharpe Ratio

```
hsbc = yf.download("0005.HK", start="2022-01-01", end="2023-01-01")
citic = yf.download("0998.HK", start="2022-01-01", end="2023-01-01")
```

```
[**********************100%**********************]  1 of 1 completed
[**********************100%**********************]  1 of 1 completed
```

Assume risk free rate is 1%

```
def cal_sharpe(df):
    rf_rate = 0.01
    log_ret_annual = np.log(df.Close[df.index[-1]] / df.Close[df.index[0]])
    df['log_ret_daily'] = np.log(df.Close/df.Close.shift(1))
    df = df.dropna()
    sharpe = (log_ret_annual - rf_rate) / df.log_ret_daily.std()
    return f"Sharpe Ratio: {sharpe:.4f}, Annual Log Return: {log_ret_annual:.4f}"
```

```
print("HSBC: ", cal_sharpe(hsbc))
print("CITIC: ", cal_sharpe(citic))
```

```
HSBC:  Sharpe Ratio: 1.3495, Annual Log Return: 0.0356
CITIC:  Sharpe Ratio: 0.7583, Annual Log Return: 0.0204
```

Sharpe Ratio Range

A Sharpe ratio less than 1 is considered bad.

From 1 to 1.99 is considered adequate/good,

From 2 to 2.99 is considered very good,

and greater than 3 is considered excellent.

但是，當它大於 3 時，也可能是一些奇怪的事情，例如新的 IPO 股票或停牌股票或公司重組或資產清算等。

Academic Resources

Google Scholar - https://scholar.google.com/

Research Gate - https://www.researchgate.net/

JSTOR - https://www.jstor.org/

Research SPJ - https://spj.science.org/

APA 6th Edition - https://libguides.library.cityu.edu.hk/citing/apa