APPLIED STATISTICS
MA 2540/4240

INSTRUCTOR: DR. Sameen Naqvi

Validating VAR Model using Hypothesis Testing

MA2540 Final Project Report

Hannnur Rahman| BT22BTECH11007

Himanshu Boora| BT22BTECH11008

Ansh Vardhan | ES21BTECH11008

Himani Agrawal| MA22BTECH11008

Meghavath Rajnikanth| ES20BTECH11018

Definition:

Value at Risk model is a statistical measure that estimates the maximum potential loss an investment might experience over a specific time period, given a certain level of confidence.

Summary of the procedure:

- Collecting stock data from yfinance python library.
- Data Analysis of the data we extract from yfinance.
- Separating data for Calculating VaR and Backtesting data.
- Calculate VaR.
- Backtesting to find the day the VaR fails to estimate the maximum loss incurred.
- Use Hypothesis Testing to validate the data we get after backtesting

About the data:

	Open	High	Low	Close	Volume
Date					
2004-08-24	284.171377	315.569996	236.654196	242.570557	3467822
2004-08-25	245.039557	245.412236	231.902450	238.331253	1436087
2004-08-26	240.380986	253.424916	236.654149	251.701248	965205
2004-08-27	252.586377	254.077117	243.176108	244.760025	368543
2004-08-30	245.971283	250.629830	245.039573	247.834702	155656
...	...	...	...	...	...
2024-04-03	10214.599609	10272.950195	9985.000000	9999.750000	14451
2024-04-04	10109.200195	10109.200195	9905.000000	10006.599609	3469
2024-04-05	10000.000000	10006.500000	9801.650391	9824.049805	8434

Description:

Our dataset includes daily stock market data for various firms across several years. Each of the ***5147*** rows depicts the percentage rise in share prices for a certain date across many firms. The dataset has ***31*** columns, with the first containing dates and the remaining representing unique firms.

In addition, we further use statistical techniques to add a new column that calculates net returns for each date using weighted averages of firm returns. This dataset serves as the foundation for our analysis of stock market trends.

]:

	ASIANPAINT.BO	AXISBANK.BO	BAJAJ-AUTO.BO	BAJFINANCE.BO	BAJAJFINSV.BO	BHARTIARTL.BO
Date						
2003-01-02	1.447696	-4.464266	0.000000	2.947349	0.000000	-1.091712
2003-01-03	-0.652766	-1.752373	0.000000	0.000000	0.000000	-0.441494
2003-01-06	0.764028	-0.356699	0.000000	-2.453981	0.000000	-1.330383
2003-01-07	0.288160	0.357976	0.000000	0.209651	0.000000	-0.898872
2003-01-08	-0.241961	2.259225	0.000000	1.673631	0.000000	-1.587304
...	...	...	...	...	...	...
2024-04-03	-0.127024	1.508214	-2.129001	1.395050	-1.079843	1.406819
2024-04-04	1.723145	-0.065833	0.694995	-0.355765	0.942214	-1.505637
2024-04-05	-1.183545	-0.423450	-1.461343	-1.469336	1.247585	-1.275935

This is the head of our data, and we can easily see that we have percentage changes in company stock prices as entries for various dates.

Exploratory Data Analysis

Introduction:

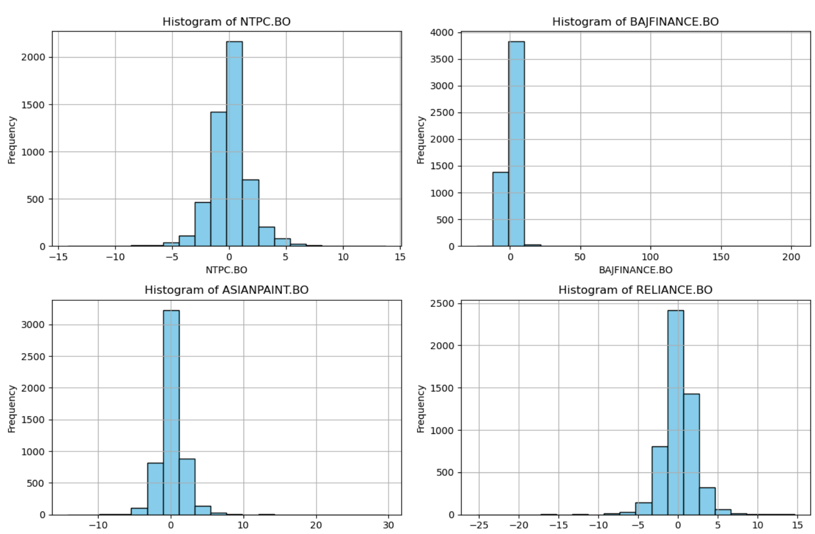
In this part, we use Exploratory Data Analysis (EDA) to acquire insights into our dataset, which includes daily stock market data from several firms. We use visuals and statistical summaries to comprehend the distribution of share price movements, detect trends and patterns across time, and analyze the portfolio's overall volatility and performance. EDA is an important early stage in our study, driving more inquiry and hypothesis testing to guide decision-making and model validation.

We'll use the following techniques to analyze the data:

- Histogram
- Trend Analysis through time
- Correlation Analysis

In addition to analyzing the raw data we'll also try to analyze the **return** variable which we'll calculate after we work through our VaR model.

1. Histogram:

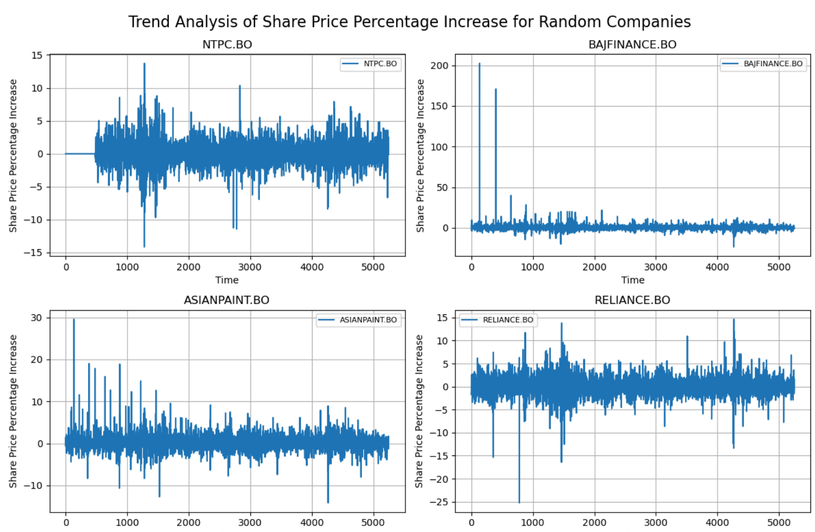


We took several random firms and showed share price histograms across the years to analyze how the stock prices are spread throughout the time interval for companies.

Inferences:

- Histograms for all four firms show a roughly bell-shaped distribution, suggesting a tendency toward normality.
- NTPC.BO and RELIANCE.BO histograms exhibit narrower distributions, indicating lower volatility in share price changes. and BAJFINANCE.BO histogram displays a wider spread, indicating higher volatility.
- Histograms for NTPC.BO, ASIANPAINT.BO, and RELIANCE.BO are centered around zero, indicating minimal changes in share prices on most days.

2. Trend Analysis through time:

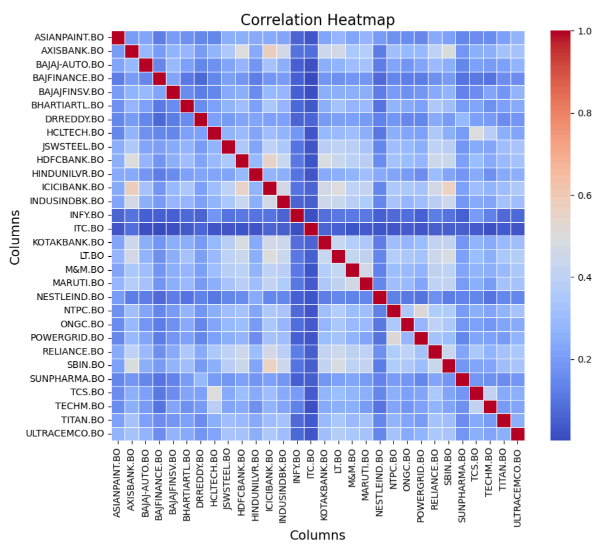


For the same Random Firms, we plotted stock prices over time to see how they fluctuate and how much volatility a specific firm demonstrated across the time period, in order to gain a better knowledge of future stock investment opportunities.

Inferences:

- NTPC.BO and RELIANCE.BO show relatively stable fluctuations over time, with consistent bands of values indicating minimal volatility.
- BAJFINANCE.BO exhibits significant spikes, suggesting drastic price changes on specific days possibly due to extraordinary events.
- BAJFINANCE.BO displays the highest volatility with pronounced spikes, while NTPC.BO shows the least volatility.

3. Correlation Analysis:



This heatmap will help us understand the link between different corporations and how variations in one stock price may impact another.

Inferences:

- From the Map we can see the following have the highest Correlation.

```
Top 5 most correlated pairs:
AXISBANK.BO   ICICIBANK.BO   0.578972
ICICIBANK.BO  SBIN.BO        0.570657
HDFCBANK.BO   ICICIBANK.BO   0.547557
NTPC.BO       POWERGRID.BO   0.515657
ICICIBANK.BO  LT.BO          0.504071
```

These data indicate that there are certain firms whose stocks depend on each other.

Value at Risk Model Explanation.

Introduction:

The VaR (Value at Risk) model is a popular risk management method in finance, used to predict the possible loss in value of a portfolio or investment over a certain time horizon and confidence level. VaR assists investors and financial institutions in assessing and managing their market risk exposure by calculating the greatest possible loss under normal market circumstances. In this section, we delve into the application of the VaR model to our dataset, exploring its effectiveness in predicting and mitigating downside risk in our portfolio of stocks.

Workflow:

- We have collected 30 stocks data from yfinance python lib.
- Data will be in form of stock price , we will convert that data into return formats
- We will then clean the data i.e. replacing null place with zero

The code below will assist us in computing the return variable that we described before; this return variable is the real return value we receive for a certain day if we have a specific portfolio. So this is our net profit or loss percentage change for a specific day.

```
def calculate_returns(stock, start, end):
    data = yf.download(stock, start, end, auto_adjust=True)['Close']
    returns = data.pct_change()  # Calculate percentage changes
    returns*=100
    return returns.dropna()
```

```
start =  datetime.date(2003, 1, 1)
end =  datetime.date(2024, 4, 10)
```

```
for stock in stock_list:
    stock_list[stock] = calculate_returns(stock, start, end)
```

To calculate the net return for each day we will:

- We will declare the weightage of investment we will invest in each stock
- We will separate the clean data for backtesting purpose

This weightages indicates the amount of stocks we have for each company so to simulate that we have randomly assigned weights to each firm. We use the following code to calculate the return value.

Value Allocation code

```
no_of_stock = 30
list=[]
for i in range(30):
    list.append(random.randint(1, 100))
sum_of_list = sum(list)
normalised_list = [round((num/sum_of_list),5) for num in list]
```

So after running the above code we'll have a new column in our dataset called as return variable which as stated will be the net actual profit or loss for that particular date.

We will now use the VaR Model to compute the expected return on a given day. This will essentially be the value that VaR claims is the least return value for the day, and since we are using a 99% confidence interval, we can declare that

'We are 99% convinced that the return value for your portfolio will not go below this level on a specific date'

When calculating the expected return value, we provide a number of days that we will evaluate to determine the return value for that day; these days are referred to as ***Backtesting Days***.

Now the final code to calculate the expected return value from VaR model:

Separating data that we want to backtest

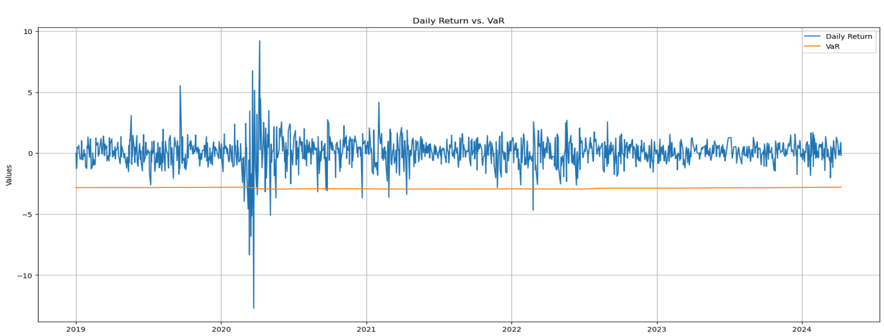
```
start_index = datetime.date(2019, 1, 1)
end_index = datetime.date(2024, 4, 11)
backtest_df = stock_list.loc[start_index:end_index]
```

```
position_vector = np.array([normalised_list])   # this is just the alloca
```

```
#calcualting z_score here we take conf=0.99 of stock data we are calculat
confidence_level = 0.99
alpha = 1 - confidence_level
alpha = round((alpha),3)
z_score = norm.ppf(alpha)
```

```
def VaR(start,end,position_vector,z_score,sensex,stock_list):
    stock_list = stock_list.loc[start:end]
    stock_list = stock_list.iloc[:, :30]
    covariance_matrix = stock_list.cov()
    portfolio_Var = np.dot(position_vector,np.dot(covariance_matrix,posit
    portfolio_Var_absolute = np.sqrt(portfolio_Var)*z_score
    return round(portfolio_Var_absolute[0][0],5)
```

We now have our real return value for a certain date, as well as the VaR model value calculated using a 99% confidence interval. Now we'll compare our model's performance to the real numbers. To do this, we'll create a graph.



In this graph, the orange line reflects the VaR model for our portfolio, while the blue line displays the actual values.

As you can see, in some circumstances, our model fails to provide accurate values for that day, and the value for that day falls below the VaR model values.

We will calculate the number of days where the values go below the VaR model values.

Calculation of total no exceeding days and dates at which it exceed(dates are not important for the calculation)

```
excedding_days = 0
for index, row in backtest_df.iterrows():
    if (backtest_df.loc[index,'daily_return']<= backtest_df.loc[index,'VaR']):
        print(index)
        excedding_days+=1
```

```
2020-02-28 00:00:00
2020-03-09 00:00:00
2020-03-12 00:00:00
2020-03-16 00:00:00
2020-03-18 00:00:00
2020-03-23 00:00:00
2020-04-01 00:00:00
2020-04-21 00:00:00
2020-05-04 00:00:00
2020-05-18 00:00:00
2020-09-24 00:00:00
2020-12-21 00:00:00
2021-02-26 00:00:00
2021-04-12 00:00:00
2022-02-24 00:00:00
```

```
excedding_days
```

```
15
```

As we can see from the above code, the total number of days where the values exceeded were 15 days.

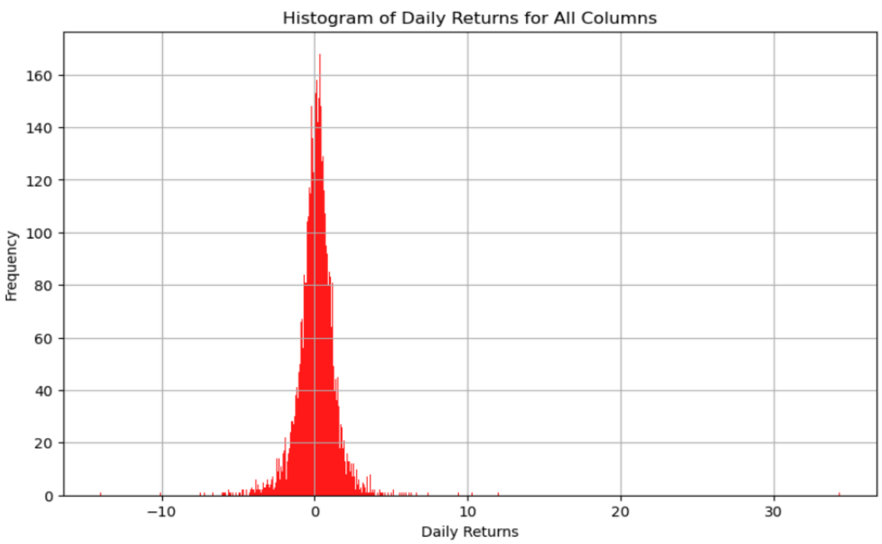
Analysis of the Return Variable

Introduction:

The Return variable represents the percentage change in the closing price of a financial instrument between two consecutive trading days.

This is a histogram plotted for the daily returns for all the columns. This histogram resembles the a bell curve indicating it could follow normal distribution

The measure of central tendency of the about plotted is as follows:

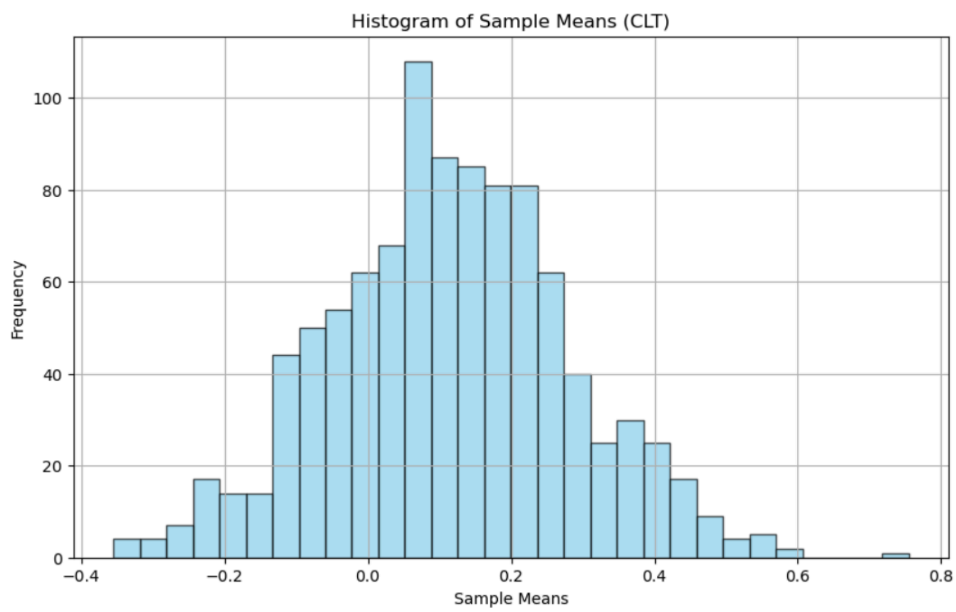



```
Measure of Central Tendency for Returns:
Mean: 0.1129518813667678
Median: 0.15543238295227996
Standard Deviation: 1.199390451005923
Mode: 0.0
```

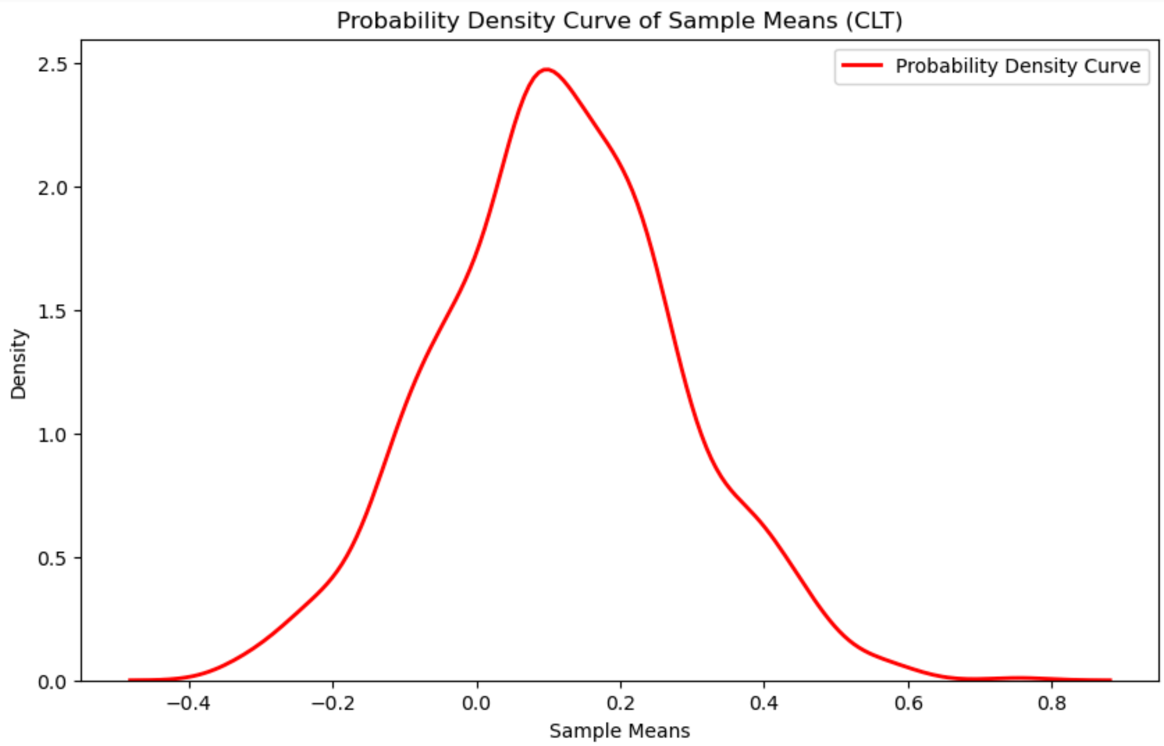
Proof of CLT :

A histogram is created by taking 1000 random samples with a sample size of 50 and calculating the sample mean.

```
# Number of samples to draw
num_samples = 1000
sample_size = 50
sample_means = []
for _ in range(num_samples):
    sample = np.random.choice(stock_list['daily_return'], size=sample_size, replace=False)
    sample_mean = np.mean(sample)
    sample_means.append(sample_mean)
```



Plotting the kernel density for the above histogram shows us that the distribution is approaching normal distribution.


Probability Density Curve of Sample Means (CLT)
Probability Density Curve
Density
2.5
2.0
1.5
1.0
0.5
0.0
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
Sample Means

Hypothesis Testing

Introduction:

To determine whether the VaR estimates produced by the model are consistent with the actual losses observed in historical data.According to the model we claim that at 1% of days our model should fail. We back tested the model on 1282 days and found out that the number of days our losses exceeded the VaR value was 15.we want to find out weather our VaR model is accurate and provide reliable estimate of potential losses within specified confidence level

Hypothesis:

Null hypothesis (H_0) : p= 0.01

Alternative hypothesis (H_a) : p ≠ 0.01

Level of significance:

α = 0.05

Test Statistic:

$$Z^* = \frac{\hat{p} - p}{\sigma_{\hat{p}}}$$

Here,

p0=0.01

p^=0.0117

$$\sigma_{\hat{p}} = \sqrt{\frac{p_0(1 - p_0)}{n}}$$

Substituting the values we have,

Z*=0.611

Z-score approach:

Zα/2=1.959

|Z*|<Zα/2

We fail to reject the Null hypothesis

P-value approach:

$$p\text{-value}=2*P(Z>|Z^*|)=0.563$$

$$p\text{-value} > \alpha$$

We fail to reject the Null hypothesis

Conclusion:

Since we fail to reject the Null-Hypothesis

The observed number of exceedances is less extreme than what would be expected under the null hypothesis.

Therefore,the VaR model is performing adequately within the specified confidence level.(i.e 95%)

Ultimately, there is enough data to draw the conclusion that the VaR model is accurate and offers trustworthy estimates of possible losses within the given confidence level.