```
#CALCULATION OF VaR and Expected Shortfall
import numpy as np,pandas as pd
import matplotlib.pyplot as plt
import yfinance as yf
import datetime as dt
from scipy.stats import norm
```

```
ticker="MSFT"
confidence_interval= 0.95
start_date=dt.datetime(2024,1,1)
end_date=dt.datetime(2024,12,31)
```

```
df=yf.download(ticker,start=start_date,end=end_date)
df
```

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

Price	Close	High	Low	Open	Volume
Ticker	MSFT	MSFT	MSFT	MSFT	MSFT
Date					
2024-01-02	367.380646	372.363320	363.319215	370.342505	25258600
2024-01-03	367.113159	369.748136	365.042827	365.538123	23083500
2024-01-04	364.478180	369.589635	363.715435	367.182505	20901500
2024-01-05	364.289978	368.559424	363.051739	365.498501	20987000
2024-01-08	371.164673	371.669884	365.538121	365.825371	23134000
...	...	...	...	...	...
2024-12-23	434.379028	436.774220	431.963858	435.866037	19152500
2024-12-24	438.450836	438.720315	433.321138	433.780209	7164500
2024-12-26	437.233276	440.057630	435.756258	438.201337	8194200
2024-12-27	429.668457	434.349074	425.496829	433.730320	18117700
2024-12-30	423.979858	426.694417	421.055729	425.207408	13158700

251 rows × 5 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
close=df["Close"]
close
```

Ticker	MSFT
Date	
2024-01-02	367.380646
2024-01-03	367.113159
2024-01-04	364.478180
2024-01-05	364.289978
2024-01-08	371.164673
...	...
2024-12-23	434.379028
2024-12-24	438.450836
2024-12-26	437.233276
2024-12-27	429.668457
2024-12-30	423.979858

251 rows × 1 columns

Next steps: Generate code with close | View recommended plots | New interactive sheet

```
return_MSFT=np.log(close/close.shift(1)).dropna()
return_MSFT
```

Ticker	MSFT
Date	
2024-01-03	-0.000728
2024-01-04	-0.007203
2024-01-05	-0.000516
2024-01-08	0.018696
2024-01-09	0.002931
...	...
2024-12-23	-0.003097
2024-12-24	0.009330
2024-12-26	-0.002781
2024-12-27	-0.017453
2024-12-30	-0.013328

250 rows × 1 columns

Next steps: Generate code with return_MSFT | View recommended plots | New interactive sheet

No results

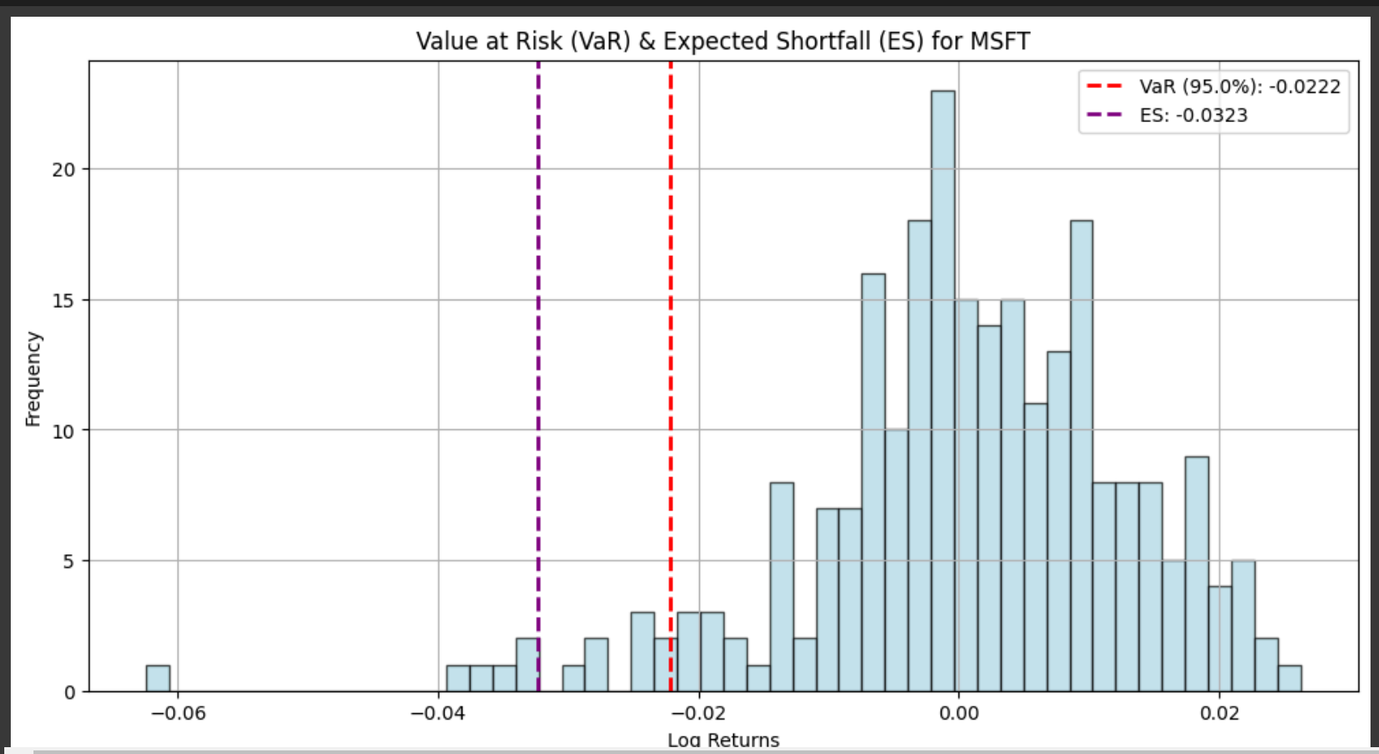
```
Var=np.percentile(return_MSFT,100*(1-confidence_interval))
Tail_loss=return_MSFT[return_MSFT<Var]
if len(Tail_loss)> 0:
  ETL=np.mean(Tail_loss)
else:
    ETL=0
print("Value at risk(Var) is:",Var)
print("Expected loss(ETL) is:",ETL)
```

```
Value at risk(Var) is: -0.022216232418296032
Expected loss(ETL) is: -0.03233013694534563
```

```
ticker = "MSFT"
confidence_interval = 0.95
start_date = dt.datetime(2024, 1, 1)
end_date = dt.datetime(2024, 12, 31)
```

```
plt.figure(figsize=(12, 6))
plt.hist(return_MSFT, bins=50, color='lightblue', edgecolor='black', alpha=0.7)
plt.axvline(Var, color='red', linestyle='dashed', linewidth=2, label=f'VaR ({confidence_interval * 100}%): {Var:.4f}')
plt.axvline(ETL, color='purple', linestyle='dashed', linewidth=2, label=f'ES: {ETL:.4f}')
plt.xlabel("Log Returns")
plt.ylabel("Frequency")
```

```
plt.title(f"Value at Risk (VaR) & Expected Shortfall (ES) for {ticker}")
plt.legend()
plt.grid()
plt.show()
```




```
#Backtest (Counting VaR Breaches)
df = yf.download(ticker, start=start_date, end=end_date)
close = df["Close"]
```

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

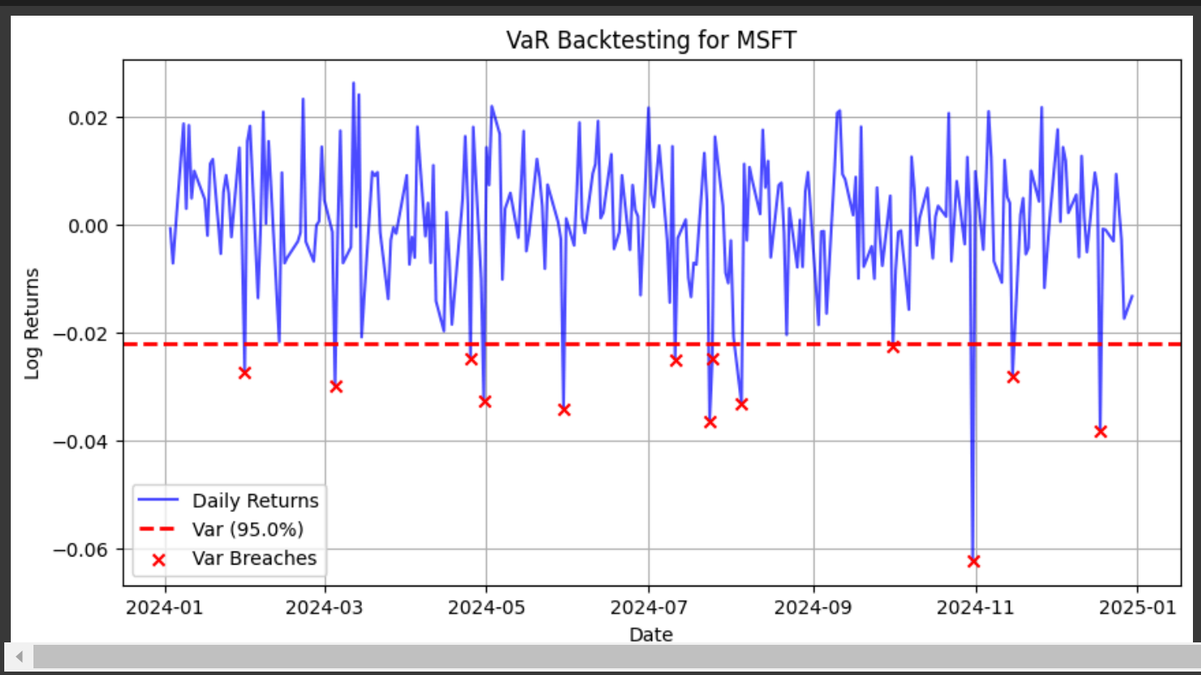
```
returns = np.log(close / close.shift(1)).dropna()
Var = np.percentile(returns, 100 * (1 - confidence_interval))
breaches = returns[returns < Var]
```

```
num_breaches = len(breaches)
total_days = len(returns)
expected_breaches = total_days * (1 - confidence_interval)

print(f"Total trading days: {total_days}")
print(f"Expected breaches (based on {confidence_interval*100}% confidence level): {expected_breaches:.2f}")
print(f"Actual breaches: {num_breaches}")
```

```
Total trading days: 250
Expected breaches (based on 95.0% confidence level): 12.50
Actual breaches: 250
```

```
plt.figure(figsize=(10, 5))
plt.plot(returns, label="Daily Returns", color="blue", alpha=0.7)
plt.axhline(Var, color="red", linestyle="dashed", linewidth=2, label=f'Var ({confidence_interval*100}%)')
plt.scatter(breaches.index, breaches, color="red", label="Var Breaches", marker="x", zorder=3)
plt.xlabel("Date")
plt.ylabel("Log Returns")
plt.title(f"VaR Backtesting for {ticker}")
plt.legend()
plt.grid()
plt.show()
```




```
Start coding or generate with AI.
```