In [17]:
```
!pip install pandas
!pip install numpy
!pip install matplotlib
```

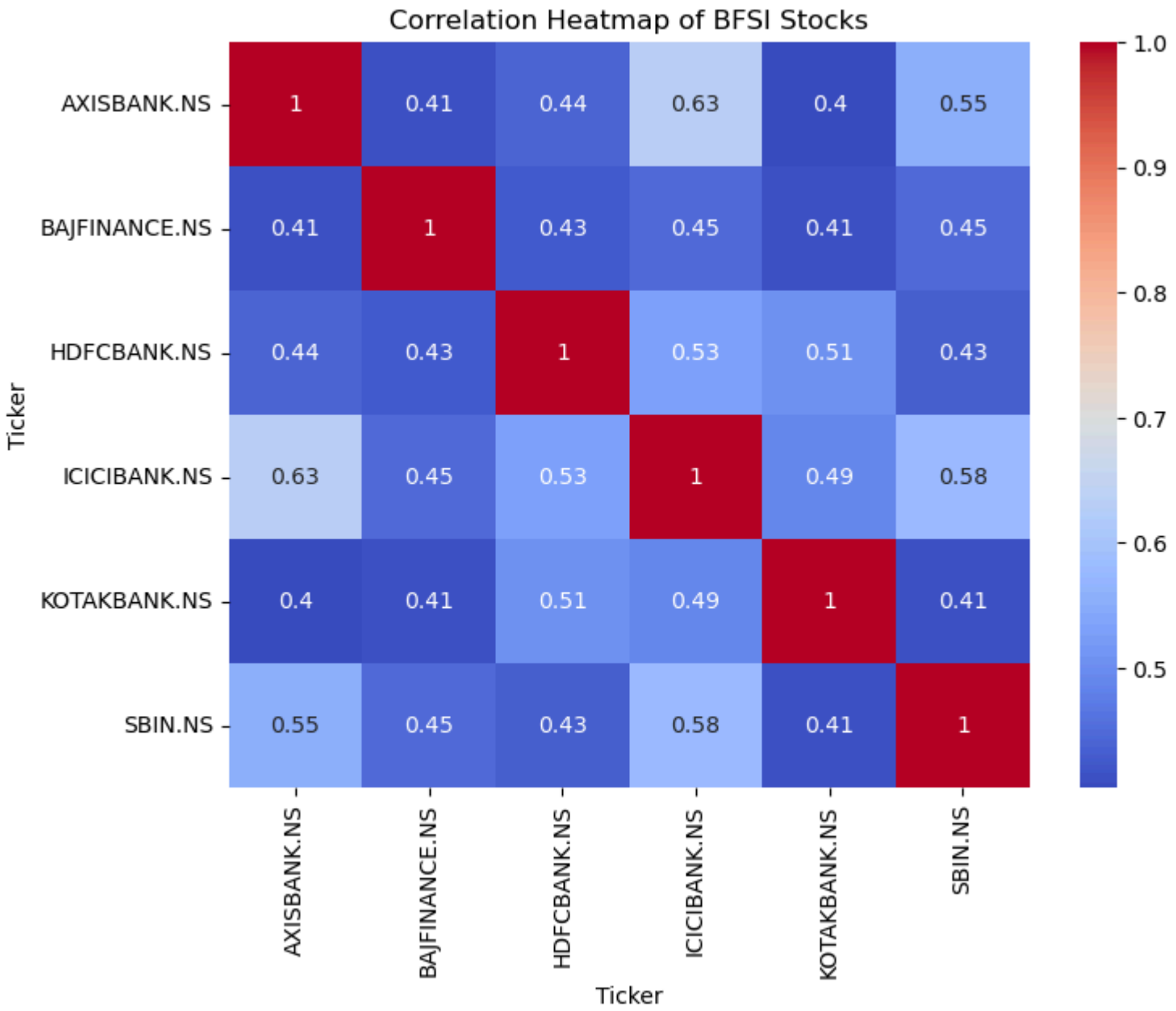
```
Requirement already satisfied: pandas in c:\users\aditya sakpal\anaconda3\lib\site-packages (2.2.2)
Requirement already satisfied: numpy>=1.26.0 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from pandas) (2.9.
0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pan
das) (1.16.0)
Requirement already satisfied: numpy in c:\users\aditya sakpal\anaconda3\lib\site-packages (1.26.4)
Requirement already satisfied: matplotlib in c:\users\aditya sakpal\anaconda3\lib\site-packages (3.8.4)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from matplotlib) (4.51.
0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from matplotlib) (1.4.
4)
Requirement already satisfied: numpy>=1.21 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=8 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from matplotlib) (2.
9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\aditya sakpal\anaconda3\lib\site-packages (from python-dateutil>=2.7->matpl
otlib) (1.16.0)
```

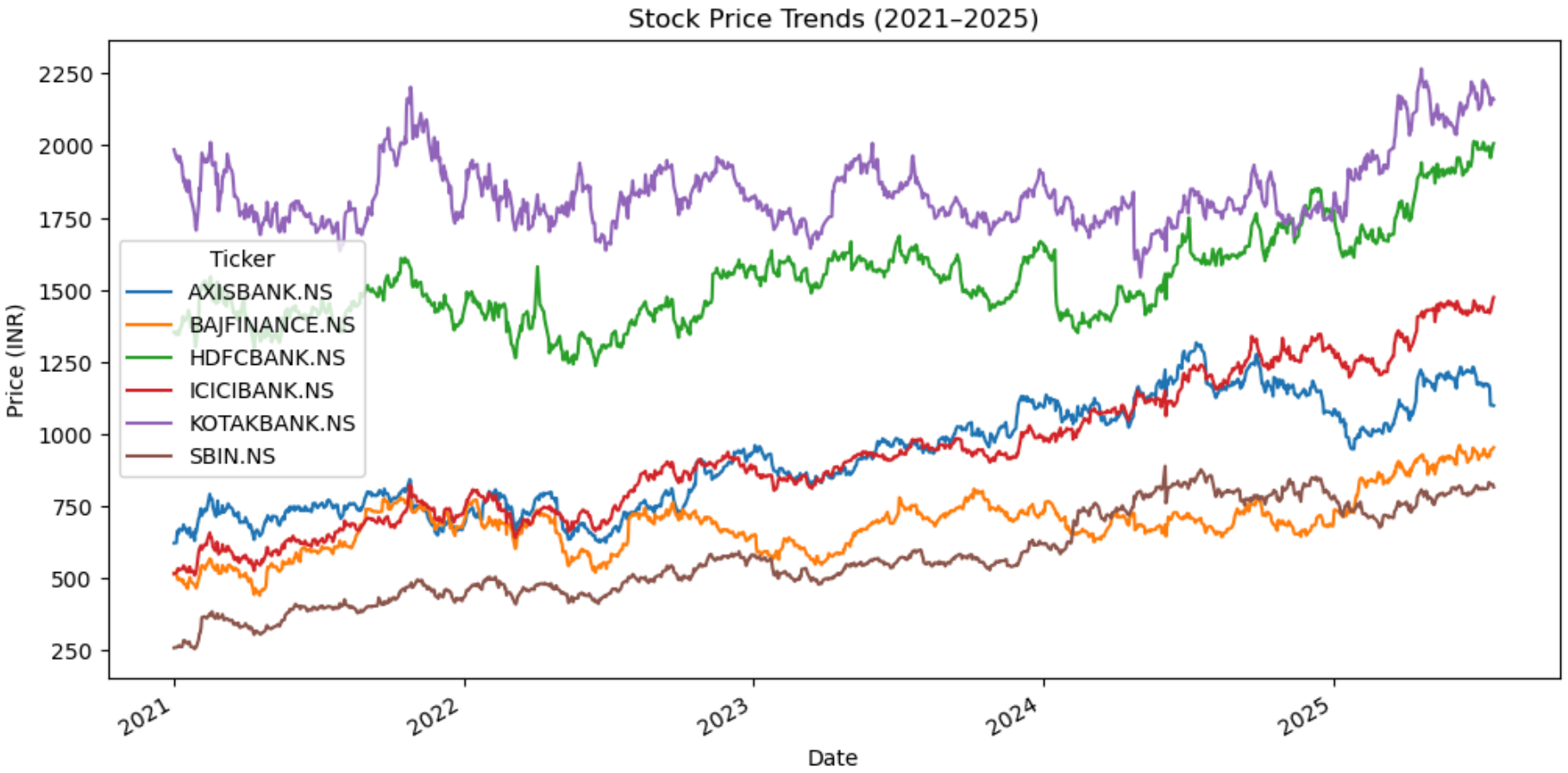
In [29]:
```
# Install these once if needed
# !pip install yfinance PyPortfolioOpt matplotlib seaborn

# STEP 1: Import Libraries
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pypfopt.efficient_frontier import EfficientFrontier
```

```
from pypfopt.expected_returns import mean_historical_return
from pypfopt.risk_models import CovarianceShrinkage
from pypfopt.plotting import plot_efficient_frontier

# STEP 2: Load BFSI stock data
tickers = ['HDFCBANK.NS', 'ICICIBANK.NS', 'KOTAKBANK.NS', 'SBIN.NS', 'AXISBANK.NS', 'BAJFINANCE.NS']
data = yf.download(tickers, start="2021-01-01", end="2025-07-23", auto_adjust=True)['Close']

# STEP 3: Calculate daily returns
returns = data.pct_change().dropna()

# STEP 4: Heatmap for correlation
plt.figure(figsize=(8,6))
sns.heatmap(returns.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap of BFSI Stocks")
plt.show()

# STEP 5: Line plot of stock prices
data.plot(figsize=(12,6))
plt.title("Stock Price Trends (2021–2025)")
plt.ylabel("Price (INR)")
plt.xlabel("Date")
plt.show()

# STEP 6: Financial metrics
mean_return = returns.mean()
volatility = returns.std()
sharpe = mean_return / volatility

# Display Sharpe ratios for individual stocks
sharpe_df = pd.DataFrame({
    'Mean Return': mean_return,
    'Volatility': volatility,
    'Sharpe Ratio': sharpe
})
print("\n Sharpe Ratio for Individual Stocks:\n")
print(sharpe_df)

# STEP 7: Portfolio Optimization - Max Sharpe
mu = mean_historical_return(data)
S = CovarianceShrinkage(data).ledoit_wolf()
```

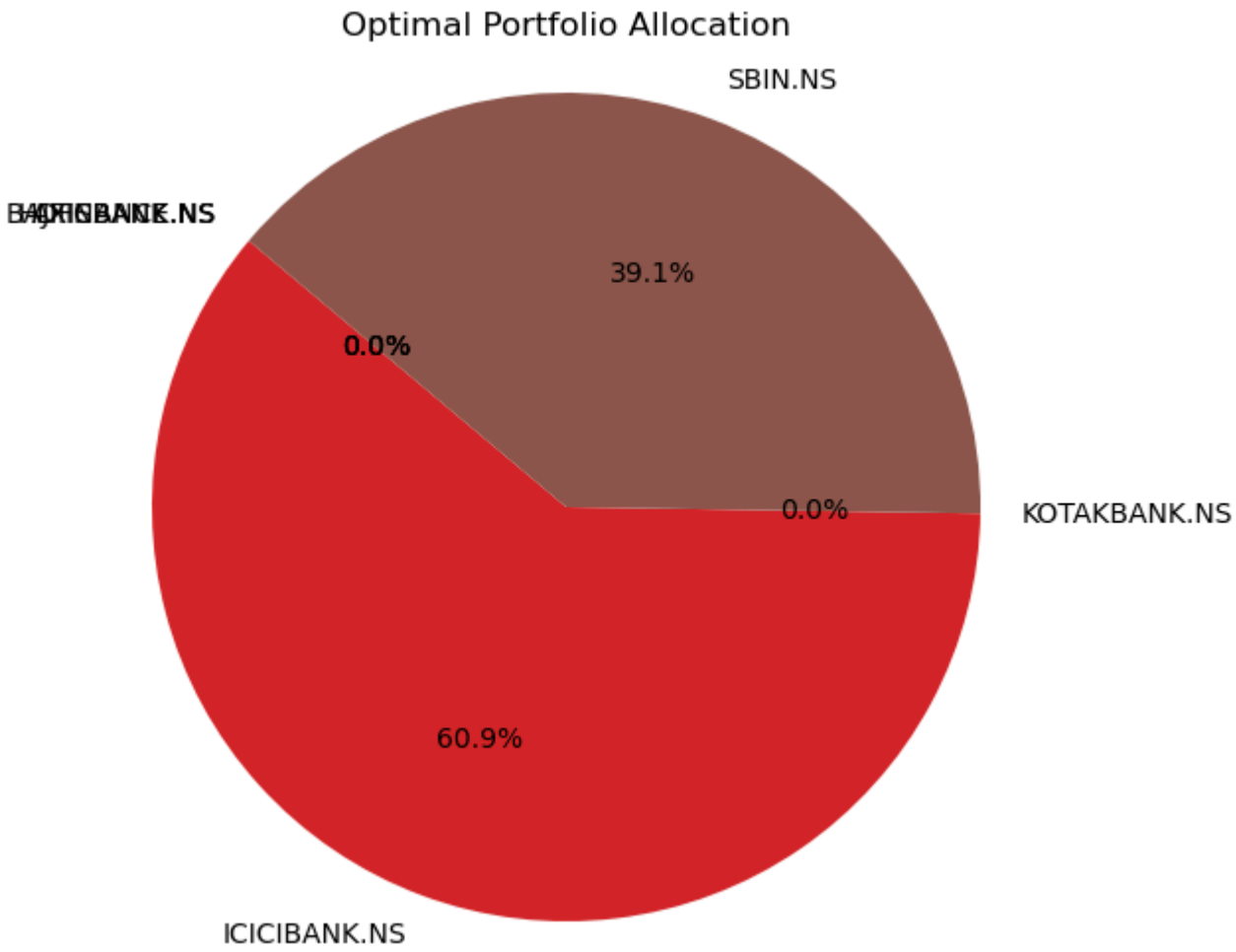
```
ef = EfficientFrontier(mu, S)
weights = ef.max_sharpe()
cleaned_weights = ef.clean_weights()

# Display optimized weights
print("\n Optimized Portfolio Allocation (Max Sharpe Ratio):")
for stock, weight in cleaned_weights.items():
    print(f"{stock}: {weight:.2%}")

# STEP 8: Pie Chart of Allocation
plt.figure(figsize=(8,6))
plt.pie(cleaned_weights.values(), labels=cleaned_weights.keys(), autopct='%1.1f%%', startangle=140)
plt.title("Optimal Portfolio Allocation")
plt.axis('equal')
plt.show()

# STEP 9: Plot Efficient Frontier using a new ef instance
ef_plot = EfficientFrontier(mu, S)
plot_efficient_frontier(ef_plot, show_assets=True)
plt.title("Efficient Frontier (Risk vs Return)")
plt.show()

# STEP 10: Portfolio Performance (Return, Risk, Sharpe Ratio)
expected_return, risk, sharpe_ratio = ef.portfolio_performance(verbose=True)

# Optional: Display just the Sharpe Ratio
print(f"\n Optimized Portfolio Sharpe Ratio: {sharpe_ratio:.2f}")
```

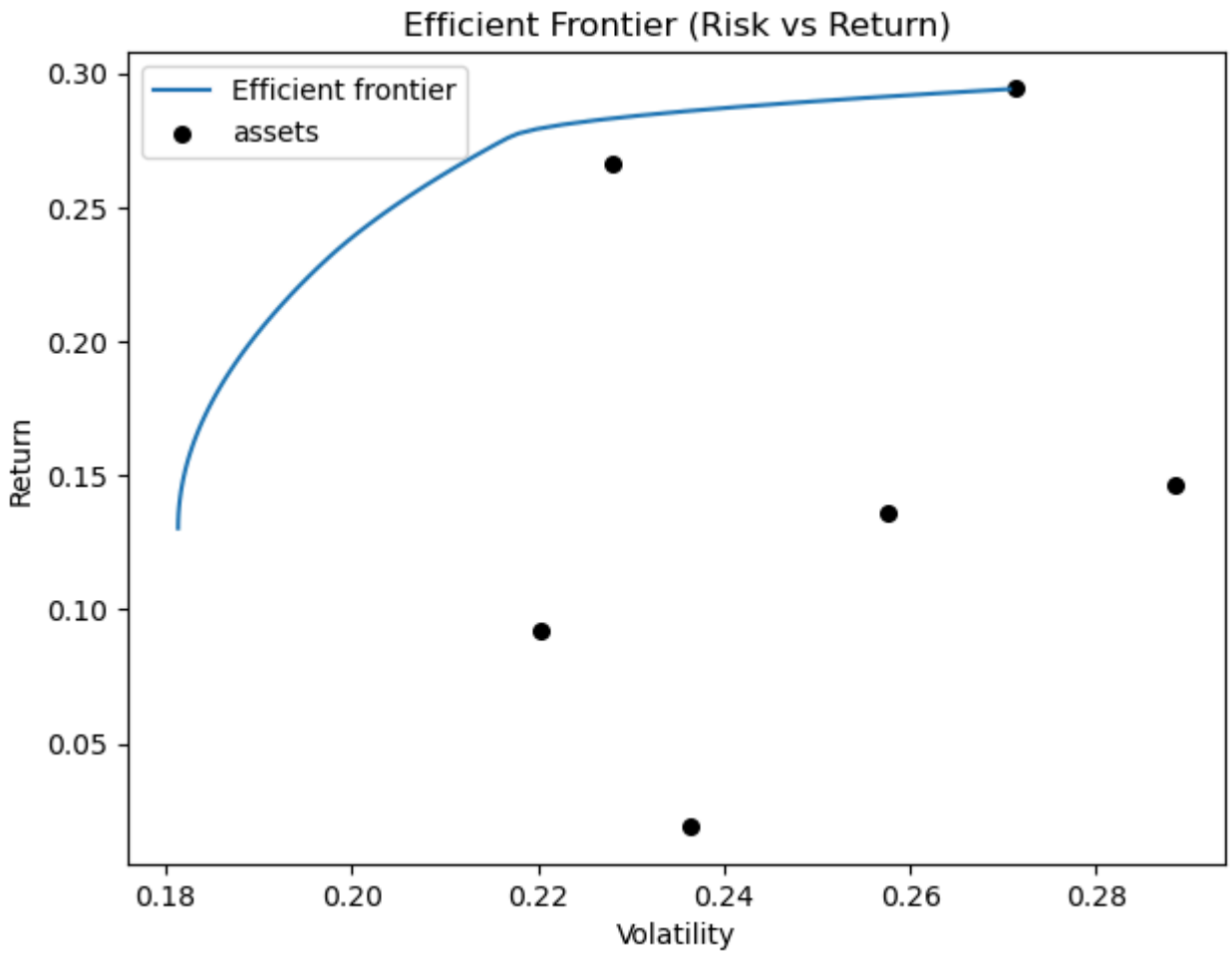
```
[**********************100%**********************]  6 of 6 completed
```


Correlation Heatmap of BFSI Stocks
AXISBANK.NS
BAJFINANCE.NS
HDFCBANK.NS
ICICIBANK.NS
KOTAKBANK.NS
SBIN.NS
Ticker
1 0.41 0.44 0.63 0.4 0.55
0.41 1 0.43 0.45 0.41 0.45
0.44 0.43 1 0.53 0.51 0.43
0.63 0.45 0.53 1 0.49 0.58
0.4 0.41 0.51 0.49 1 0.41
0.55 0.45 0.43 0.58 0.41 1
AXISBANK.NS
BAJFINANCE.NS
HDFCBANK.NS
ICICIBANK.NS
KOTAKBANK.NS
SBIN.NS
Ticker
1.0
0.9
0.8
0.7
0.6
0.5


Stock Price Trends (2021–2025)
Price (INR)
2250
2000
1750
1500
1250
1000
750
500
250
Ticker
AXISBANK.NS
BAJFINANCE.NS
HDFCBANK.NS
ICICIBANK.NS
KOTAKBANK.NS
SBIN.NS
2021
2022
2023
2024
2025
Date

```
 Sharpe Ratio for Individual Stocks:

              Mean Return  Volatility  Sharpe Ratio
Ticker
AXISBANK.NS       0.000638    0.016237      0.039293
BAJFINANCE.NS     0.000707    0.018220      0.038812
HDFCBANK.NS       0.000446    0.013853      0.032222
ICICIBANK.NS      0.001038    0.014344      0.072396
KOTAKBANK.NS      0.000186    0.014880      0.012473
SBIN.NS           0.001170    0.017117      0.068355

 Optimized Portfolio Allocation (Max Sharpe Ratio):
AXISBANK.NS: 0.00%
BAJFINANCE.NS: 0.00%
HDFCBANK.NS: 0.00%
ICICIBANK.NS: 60.87%
KOTAKBANK.NS: 0.00%
SBIN.NS: 39.13%
```


Optimal Portfolio Allocation
SBIN.NS
39.1%
0.0%
0.0%
KOTAKBANK.NS
60.9%
ICICIBANK.NS




```
Expected annual return: 27.7%
Annual volatility: 21.8%
Sharpe Ratio: 1.27

 Optimized Portfolio Sharpe Ratio: 1.27
```