Introduction

This Python script is designed to perform a comprehensive analysis of the Bank Nifty (NSE Index) stock data. The Bank Nifty index is a benchmark index for the Indian banking sector and is composed of the most liquid and large capitalized banking stocks listed on the National Stock Exchange of India (NSE). The script includes various analyses and predictions:

Time Series Forecasting of the closing prices of the Bank Nifty Index using a Long Short-Term Memory (LSTM) model. LSTMs are a type of recurrent neural network that are capable of learning patterns in sequences of data, such as time series data. Statistical Analysis of daily, weekly, and monthly returns, including their distributions. Trading Strategy Development, based on Chandelier Exit (a type of trailing stop loss order) and Average True Range (ATR), which are popular volatility-based indicators used by traders to set stop-loss orders.

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import math
import pandas as pd
import pandas_datareader as web
import yfinance as yf
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import LSTM, Dense
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.layers import Dropout
from tensorflow.keras.callbacks import EarlyStopping
plt.style.use('fivethirtyeight')
```

Data Collection

The script begins by importing necessary Python packages and libraries and then fetching the historical stock data of the Bank Nifty Index from Yahoo Finance using the yfinance library.

In [ ]:
```
#getting data for Bank Nifty

# Define the ticker symbol for Bank Nifty
ticker = "^NSEBANK"
```

```
# Set the start and end dates for the data
start_date = "2022-01-01"
end_date = "2023-05-30"

# Download the 15-minute data for Bank Nifty from Yahoo Finance
df = yf.download(ticker, start=start_date, end=end_date, interval="1h")
```

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

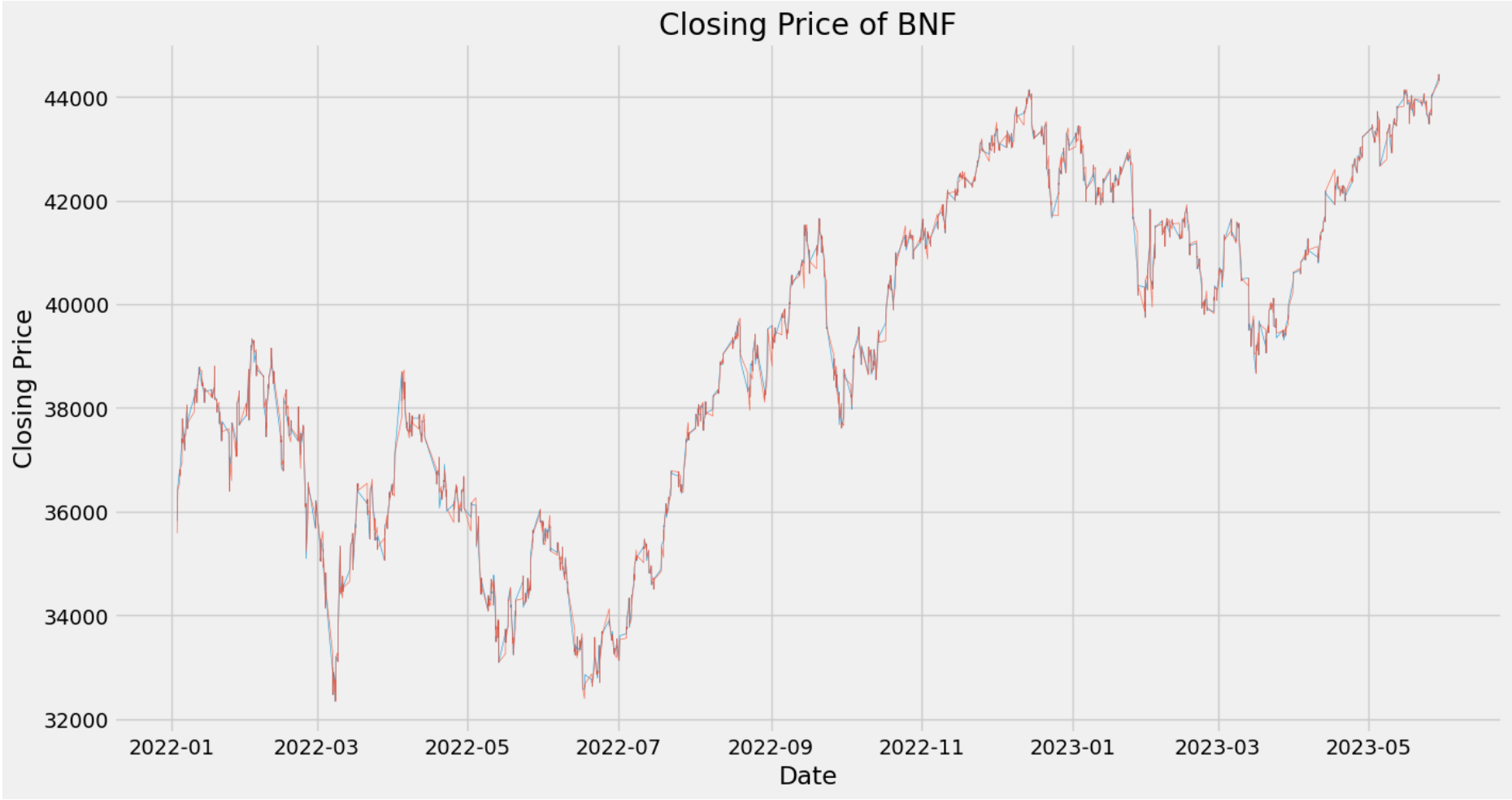
Data Preprocessing

The raw stock data is filtered to retain only the closing prices. The data is then split into training and test datasets, with 80% of the data used for training. The closing prices are normalized using the MinMaxScaler from the sklearn.preprocessing module.

In [ ]:
```
df.head(3)
df.shape
```

Out[ ]:
```
(2423, 6)
```

In [ ]:
```
plt.figure(figsize= (15, 8))
plt.title('Closing Price of BNF')
plt.plot(df.Close, linewidth=0.5, alpha=0.8, label='Close')
plt.plot(df.Open, linewidth=0.5, alpha=0.8, label='Open')
plt.xlabel('Date')
plt.ylabel('Closing Price')
plt.show()
```



In [ ]:
```
data = df.filter(['Close'])
# convert data into numpy array
dataset = data.values
# get the number of rows to train data on
training_data_len = math.ceil(len(dataset)*.8)
training_data_len
```

Out[ ]:
```
1939
```

In [ ]:
```
# scale the data
scaler = MinMaxScaler(feature_range=(0, 1))
```

```
scaled_data = scaler.fit_transform(dataset)
scaled_data
```

Out[ ]:
```
array([[0.28794108],
       [0.29907785],
       [0.30134306],
       ...,
       [0.99918971],
       [0.98981013],
       [0.99065337]])
```

LSTM Model

A Sequential model is built using the Keras API, which is part of TensorFlow. The model architecture includes two LSTM layers with Dropout layers in between to prevent overfitting. It is then compiled using the Stochastic Gradient Descent (SGD) optimizer and the Mean Squared Error (MSE) as the loss function. Early stopping is used during training to halt training if the model's performance stops improving on a held-out validation dataset.

In [ ]:
```
# create the training dataset
# create the scaled training dataset
train_data = scaled_data[0:training_data_len, :]
# splitting the data into training and test
x_train = []
y_train = []
for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_data[i, 0])
    if i < 61:
        print(x_train)
        print(y_train)
        print()
```

```
[array([0.28794108, 0.29907785, 0.30134306, 0.31605946, 0.32662137,
       0.33678217, 0.34116402, 0.3696751 , 0.36189932, 0.36856317,
       0.35946457, 0.36200687, 0.36985693, 0.37333353, 0.418636  ,
       0.41227832, 0.42627937, 0.45037147, 0.44311662, 0.44479889,
       0.43849481, 0.40447767, 0.40039358, 0.40576335, 0.41600295,
       0.42476665, 0.42250113, 0.42407198, 0.47273124, 0.4625249 ,
       0.44116532, 0.43504307, 0.43491098, 0.44412101, 0.44760601,
       0.48709207, 0.47872108, 0.47474034, 0.47823762, 0.48624463,
       0.49734814, 0.49637249, 0.47640614, 0.49359863, 0.49337547,
       0.50381757, 0.504665  , 0.50476835, 0.50369775, 0.5332339 ,
       0.5312206 , 0.52736387, 0.53307662, 0.52127068, 0.52861629,
       0.52645863, 0.5041276 , 0.51070458, 0.51287063, 0.50610764])]
[0.5029039279203866]
```

In [ ]:
```
#convert into np.array
x_train, y_train = np.array(x_train), np.array(y_train)
```

In [ ]:
```
#reshape the data
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
x_train.shape
```

Out[ ]:
```
(1879, 60, 1)
```

In [ ]:
```
#build the LSTM model
model = Sequential()
model.add(LSTM(5000, return_sequences=True, input_shape=(x_train.shape[1], 1)))
model.add(Dropout(0.2))  # Add dropout layer
model.add(LSTM(5000, return_sequences=False))
model.add(Dropout(0.2))  # Add dropout layer
model.add(Dense(2500))
model.add(Dense(1))
```

In [ ]:
```
# compile the model
from tensorflow.keras.optimizers import SGD
model.compile(optimizer=SGD(learning_rate=0.00001), loss='mean_squared_error')
```

In [ ]:
```
# train the model
# Define early stopping callback
# Experiment with different optimizers, such as Adam, RMSprop, or SGD, and tune their hyperparameters.
early_stopping = EarlyStopping(patience=10, restore_best_weights=True)
```

```
# Compile the model with a different optimizer
model.compile(optimizer='SGD', loss='mean_squared_error')

# Train the model with early stopping
model.fit(x_train, y_train, batch_size=1, epochs=1, callbacks=[early_stopping]) # change the epochs to a higher number
```

```
1879/1879 [==============================] - ETA: 0s - loss: 0.0050
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not available. Available metrics are: loss
1879/1879 [==============================] - 347s 184ms/step - loss: 0.0050
```

Out[ ]:
```
<keras.callbacks.History at 0x7fd7f069ce50>
```

In [ ]:
```
# create the testing dataset
# create a new dataset containing sclaed values from index
test_data = scaled_data[training_data_len-60: , :]
# create the datasets x_test and y_test
x_test = []
y_test = dataset[training_data_len:, :]
for i in range(60, len(test_data)):
  x_test.append(test_data[i-60:i, 0])
```

In [ ]:
```
# convert the daata to np.array
x_test = np.array(x_test)
```

In [ ]:
```
# reshape the data
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
```

Model Prediction and Evaluation

The trained model is used to make predictions on the test data. The Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) between the predicted and actual closing prices are computed as metrics to evaluate the model's performance.

In [ ]:
```
# get the model's predicted price values
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)
```

```
16/16 [==============================] - 4s 200ms/step
```

In [ ]:
```
# get RMSE
rmse = np.sqrt(np.mean(((predictions - y_test)**2)))
rmse
```

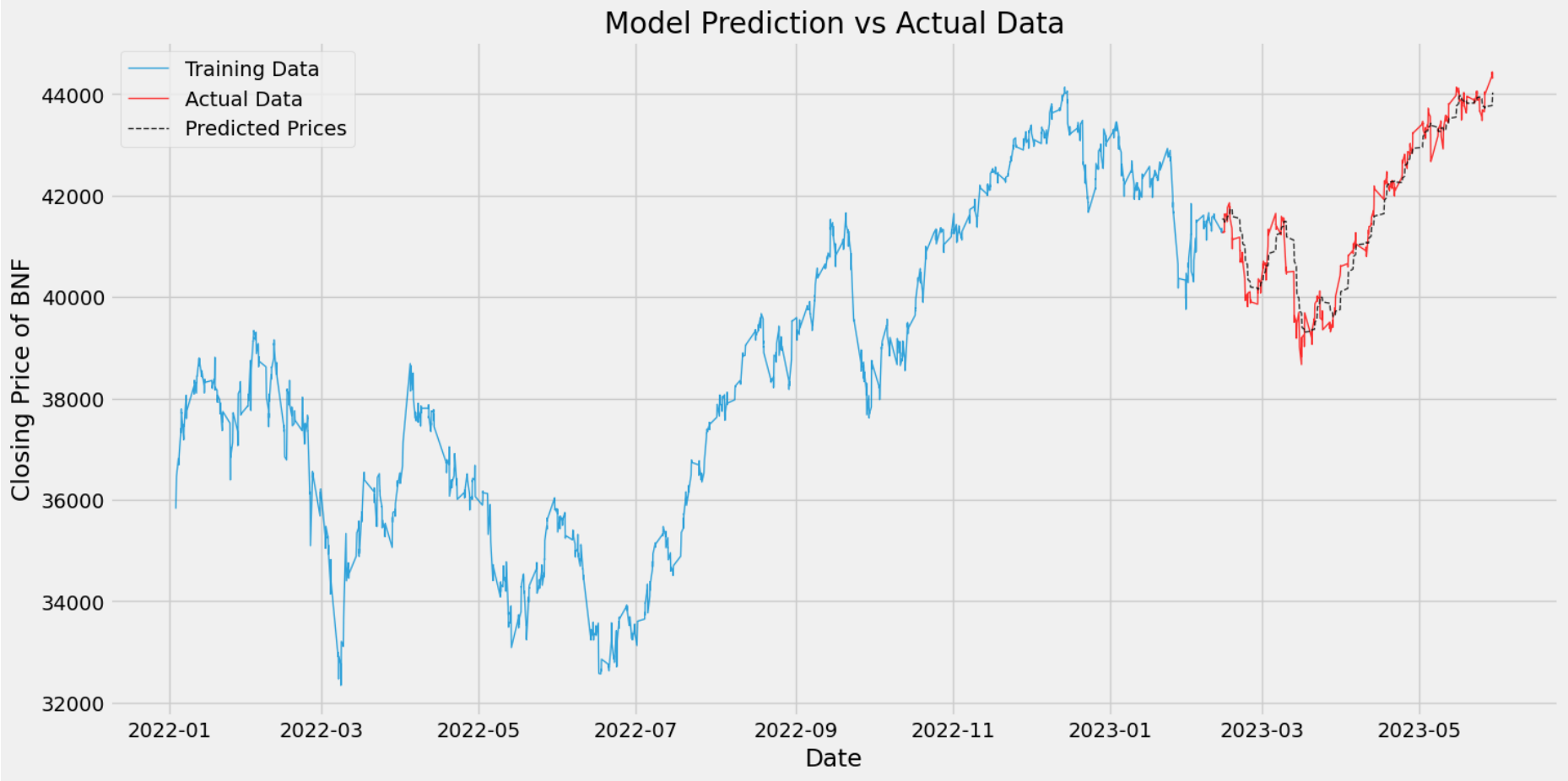
Out[ ]:
```
361.7366505425037
```

Statistical Analysis of Returns

The script computes daily, weekly, and monthly returns and plots the histogram of these returns. It also calculates certain percentiles (5th, 95th, 15th, and 85th) to visualize the range in which 90% and 70% of the returns lie. This analysis helps in understanding the volatility and risk associated with the Bank Nifty Index.

In [ ]:
```
# Plot the data
train = data[:training_data_len]
valid = data[training_data_len:]
valid['Predictions'] = predictions

plt.figure(figsize=(16, 8))
plt.title('Model Prediction vs Actual Data')
plt.xlabel('Date')
plt.ylabel('Closing Price of BNF')

# Plot training data
plt.plot(train['Close'], linewidth=1, alpha=0.8, label='Training Data')

# Plot validation data and predicted prices
plt.plot(valid['Close'], linewidth=1, alpha=0.8, color='red', label='Actual Data')
plt.plot(valid['Predictions'], linewidth=1, linestyle='--', alpha=0.8, color='black', label='Predicted Prices')

plt.legend()
plt.show()
```

```
<ipython-input-18-39522cd12c4d>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  valid['Predictions'] = predictions
```

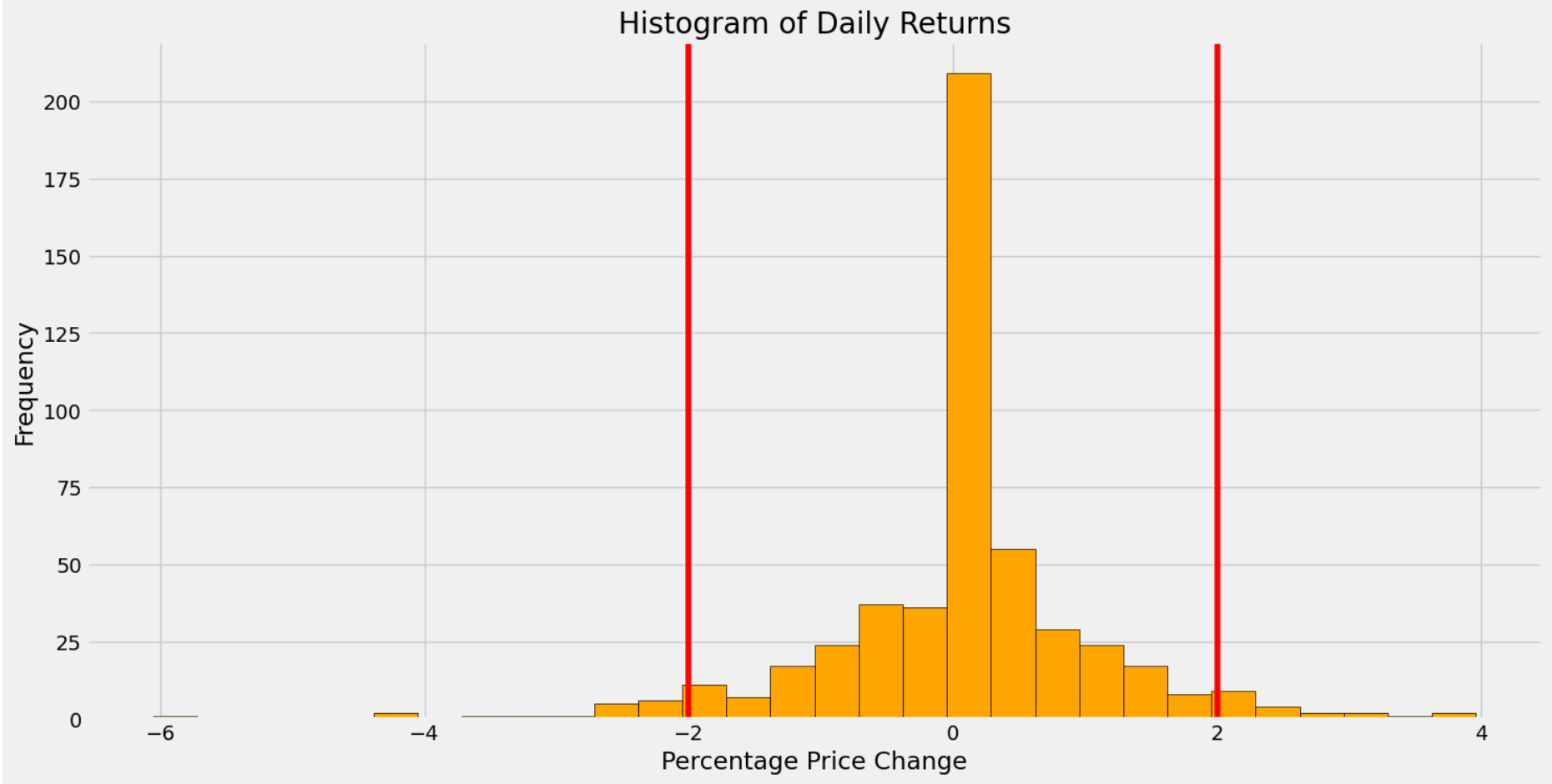


Interesting Observations: SGD continuously overpredicts the closing price while RMS prop and adam optimizers tend to underpredict the closing price.

Trading Strategy Development

The script then develops a trading strategy based on the Chandelier Exit and Average True Range (ATR) indicators. The Chandelier Exit sets a trailing stop-loss based on the ATR. The script performs a parameter optimization for the ATR calculation window and the multiplier used in the Chandelier Exit calculation. The objective of this optimization is to find the parameters that would have resulted in the best trading performance based on historical data.

In [ ]:
```
# Resample data to daily frequency and calculate daily returns as percentage price changes
df['Return'] = (df['Close'].resample('D').last().pct_change()*100) # somwhow this is not getting defined

# Plot the histogram of percentage price changes
plt.figure(figsize=(16, 8))
plt.hist((df['Close'].resample('D').last().pct_change()*100).dropna(), bins=30, color='orange', edgecolor='black')
plt.axvline(-2, color= 'red')
plt.axvline(2, color = 'red')
plt.xlabel('Percentage Price Change')
plt.ylabel('Frequency')
plt.title('Histogram of Daily Returns')
plt.show()
```



In [ ]:
```
#Define the ticker symbol for Bank Nifty
ticker = "^NSEBANK"

# Set the start and end dates for the data
start_date = "2017-01-01"
end_date = "2023-05-30"

# Download the daily data for Bank Nifty from Yahoo Finance
df_1 = yf.download(ticker, start=start_date, end=end_date, interval="1d")
```

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

In [ ]:
```
df_1.head(10)
```

Out[ ]:

Date	Open	High	Low	Close	Adj Close	Volume
2017-01-03	18002.750000	18115.050781	17830.949219	18035.599609	18035.390625	0
2017-01-04	18037.449219	18092.849609	17868.900391	17891.000000	17890.792969	0
2017-01-05	18000.750000	18164.050781	17977.800781	18115.949219	18115.738281	0
2017-01-06	18168.449219	18325.500000	18157.300781	18264.000000	18263.787109	0
2017-01-09	18314.250000	18373.099609	18256.150391	18286.650391	18286.437500	0
2017-01-10	18351.449219	18441.150391	18275.750000	18409.599609	18409.384766	0
2017-01-11	18535.349609	18889.449219	18515.250000	18830.000000	18829.781250	0
2017-01-12	18885.449219	18966.300781	18805.849609	18873.949219	18873.730469	0
2017-01-13	18949.699219	18952.250000	18781.250000	18912.099609	18911.878906	0
2017-01-16	18899.699219	19134.500000	18865.250000	19096.449219	19096.226562	0

In [ ]:

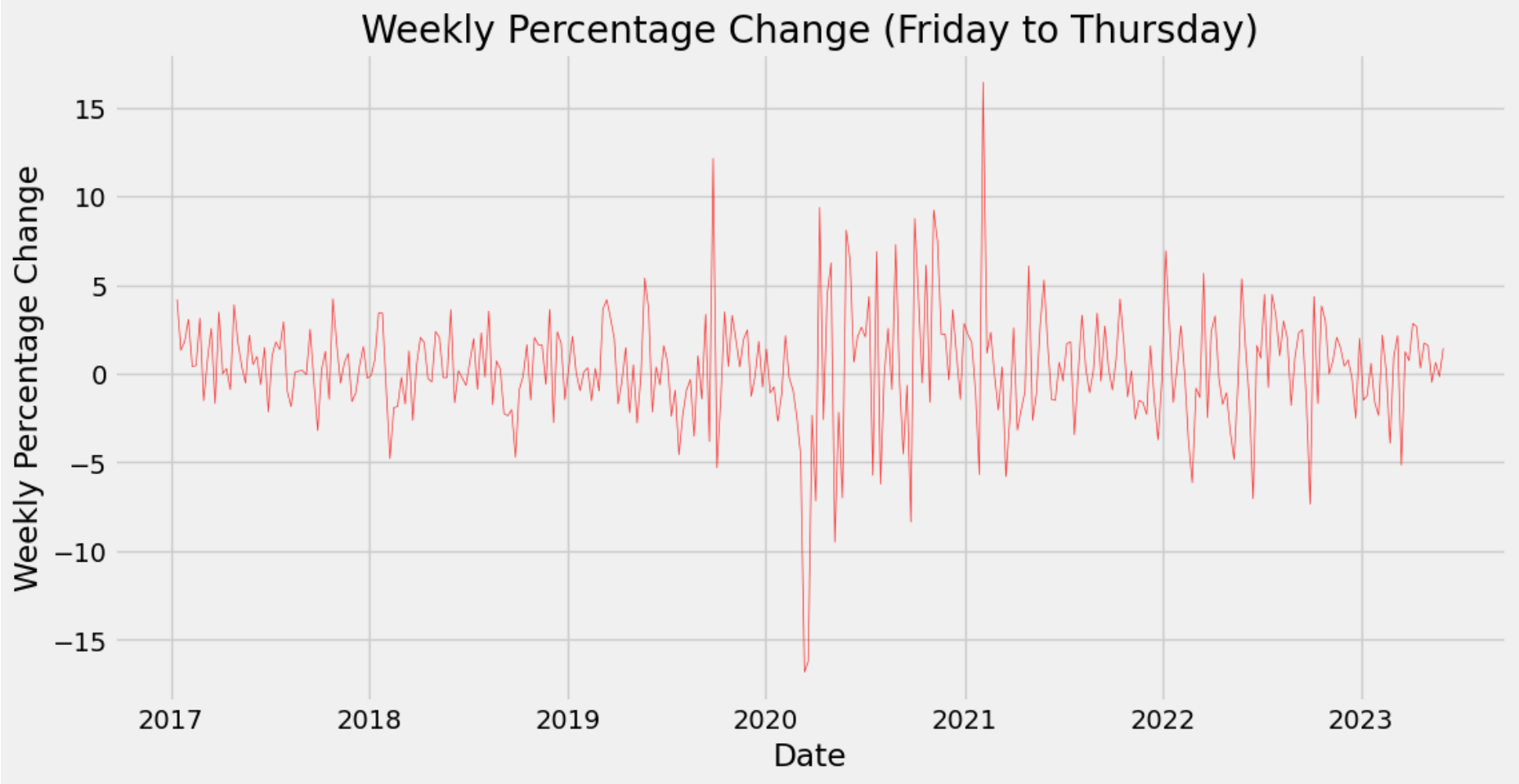
```
# Resample data to weekly frequency from Friday to Thursday
df_weekly = df_1.resample('W-THU').last()

# now we print and check if we have closing price on thursday (indeed we do)
df_weekly.tail(10)
```

Out[ ]:

Date	Open	High	Low	Close	Adj Close	Volume
2023-03-30	39611.550781	40055.000000	39609.550781	39910.148438	39910.148438	259600
2023-04-06	40940.699219	41274.699219	40820.550781	41041.000000	41041.000000	196500
2023-04-13	41680.101562	42196.199219	41502.648438	42132.550781	42132.550781	232600
2023-04-20	42218.500000	42378.148438	42108.851562	42269.500000	42269.500000	180200
2023-04-27	42753.898438	43043.398438	42736.601562	43000.851562	43000.851562	181400
2023-05-04	43236.101562	43739.800781	43213.949219	43685.449219	43685.449219	175400
2023-05-11	43535.101562	43774.250000	43367.250000	43475.300781	43475.300781	190700
2023-05-18	44006.898438	44079.199219	43673.699219	43752.300781	43752.300781	226500
2023-05-25	43630.250000	43719.851562	43390.300781	43681.398438	43681.398438	170200
2023-06-01	44276.800781	44483.351562	44193.949219	44311.898438	44311.898438	148600

In [ ]:

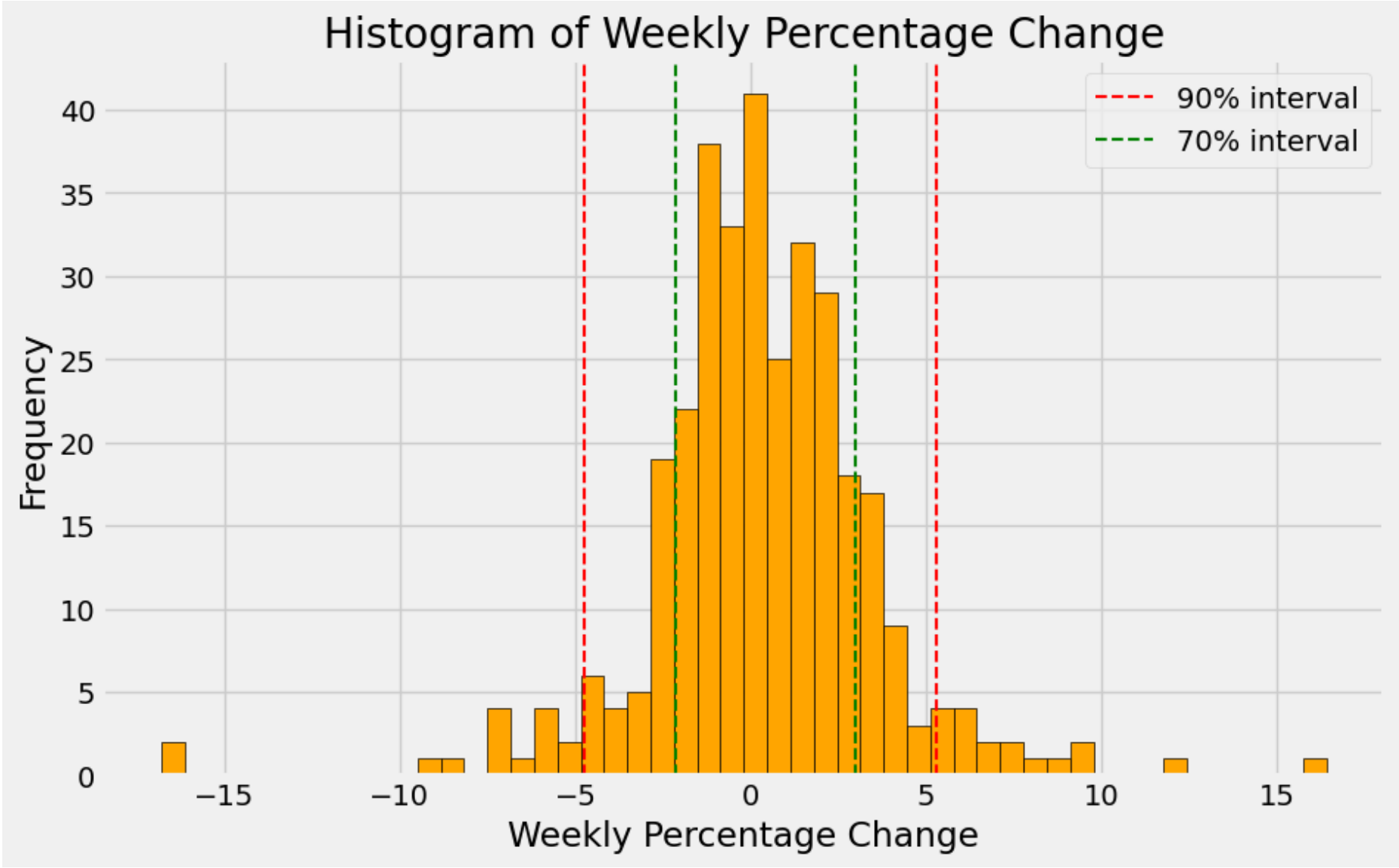
```
# Calculate weekly percentage change
df_weekly['Weekly_Return'] = df_weekly['Close'].pct_change() * 100

# Plot the weekly percentage change
plt.figure(figsize=(12, 6))
plt.plot(df_weekly['Weekly_Return'], color='red', linewidth=0.5, alpha=0.8)
plt.xlabel('Date')
plt.ylabel('Weekly Percentage Change')
plt.title('Weekly Percentage Change (Friday to Thursday)')
plt.show()
```



In [ ]:

```
# Plot the histogram of weekly percentage change
plt.figure(figsize=(10, 6))
plt.hist(df_weekly['Weekly_Return'].dropna(), bins=50, color='orange', edgecolor='black')
plt.xlabel('Weekly Percentage Change')
plt.ylabel('Frequency')
plt.title('Histogram of Weekly Percentage Change')

# Calculate the 5th and 95th percentiles
pct_5 = np.percentile(df_weekly['Weekly_Return'].dropna(), 5)
pct_95 = np.percentile(df_weekly['Weekly_Return'].dropna(), 95)
pct_15 = np.percentile(df_weekly['Weekly_Return'].dropna(), 15)
pct_85 = np.percentile(df_weekly['Weekly_Return'].dropna(), 85)
```

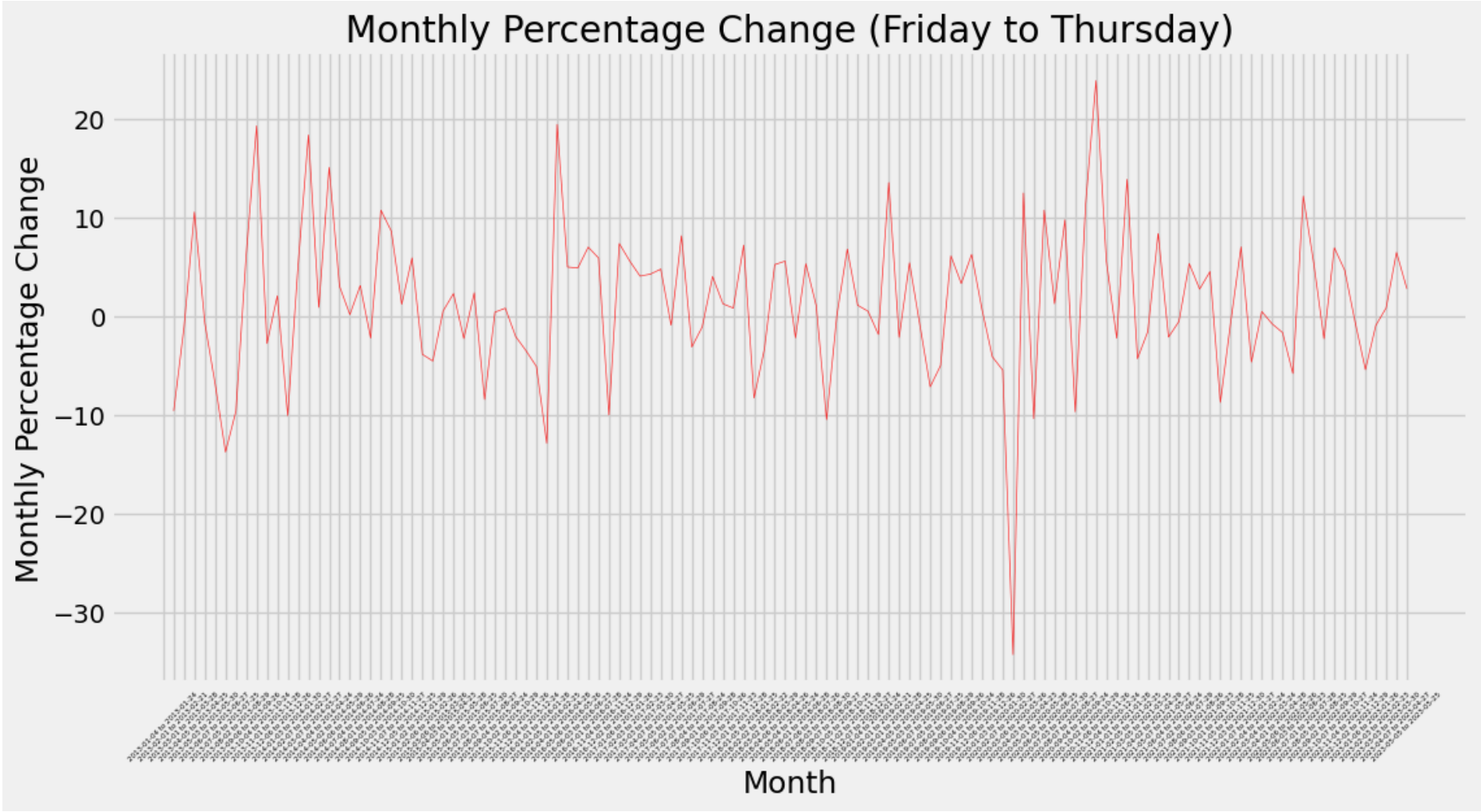
```
# Draw lines for the 90% and 70% intervals around 0
plt.axvline(pct_5, color='red', linestyle='--', linewidth=1.5, label='90% interval')
plt.axvline(pct_95, color='red', linestyle='--', linewidth=1.5)
plt.axvline(pct_15, color='green', linestyle='--', linewidth=1.5, label='70% interval')
plt.axvline(pct_85, color='green', linestyle='--', linewidth=1.5)

plt.legend()
plt.show()
```



In [ ]:
```
# Define the ticker symbol for Bank Nifty
ticker = "^NSEBANK"

# Set the start and end dates for the data
start_date = "2013-01-01"
end_date = "2023-05-31"
```

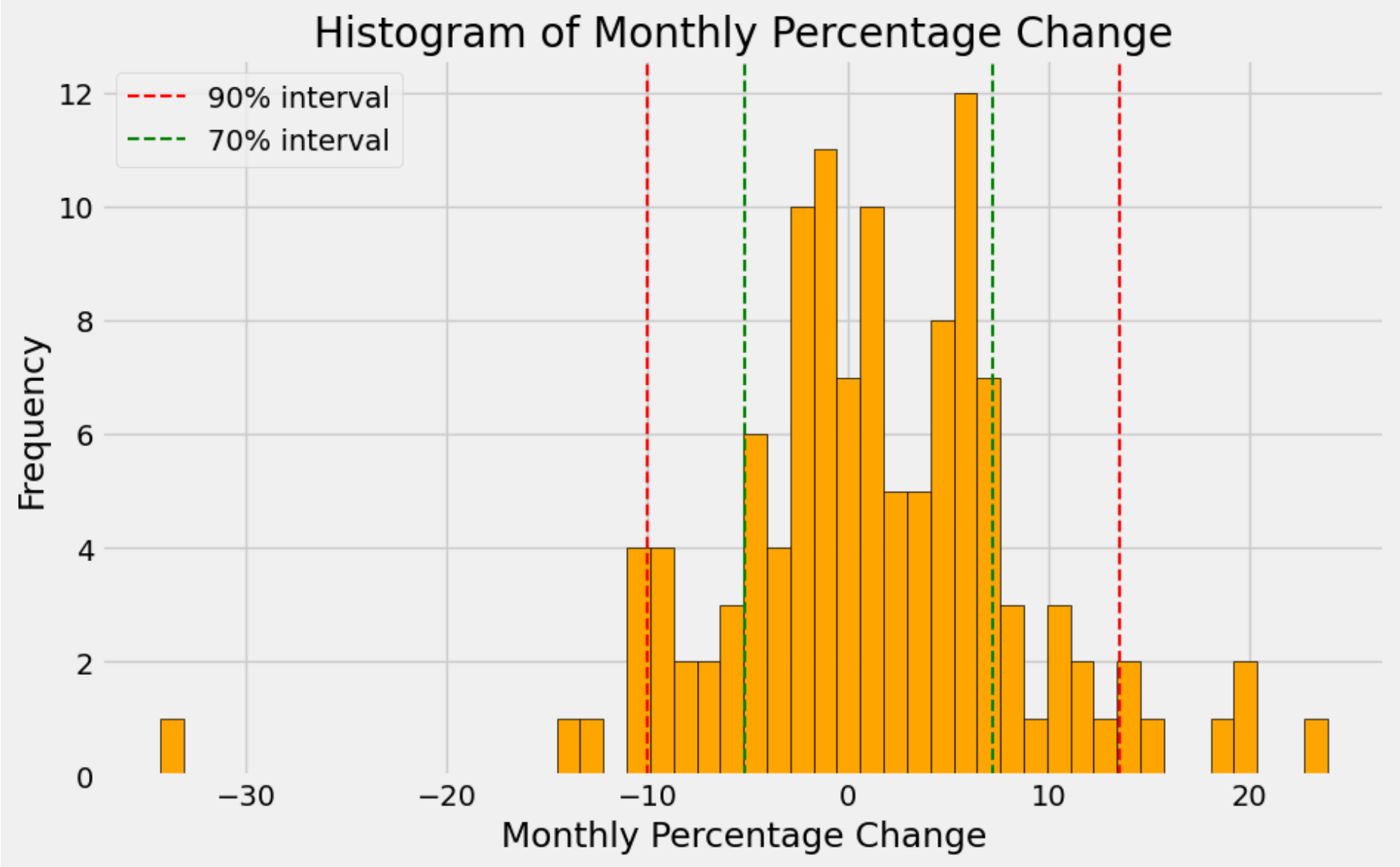
```python
# Download the daily data for Bank Nifty from Yahoo Finance
df_2 = yf.download(ticker, start=start_date, end=end_date)

# Resample data to monthly frequency with custom month definition
def custom_month_labels(date):
    year = date.year
    month = date.month
    first_day = pd.Timestamp(year, month, 1)
    first_friday = first_day + pd.DateOffset(days=(4 - first_day.dayofweek) % 7)
    last_day = pd.Timestamp(year, month, pd.Timestamp(year, month, 1).days_in_month)
    last_thursday = last_day - pd.DateOffset(days=(last_day.dayofweek + 3) % 7)
    if last_thursday >= first_friday:
        last_thursday -= pd.DateOffset(days=1)
    return f"{first_friday.strftime('%Y-%m-%d')} to {last_thursday.strftime('%Y-%m-%d')}"

df_monthly = df_2.groupby(custom_month_labels).last()

# Calculate monthly percentage change
df_monthly['Monthly_Return'] = df_monthly['Close'].pct_change() * 100

# Plot the monthly percentage change as a line plot
plt.figure(figsize=(12, 6))
plt.plot(df_monthly['Monthly_Return'], color='red', linewidth=0.5, alpha=0.8)
plt.xlabel('Month')
plt.ylabel('Monthly Percentage Change')
plt.title('Monthly Percentage Change (Friday to Thursday)')
plt.xticks(rotation=45, size = 4)
plt.show()
```

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



In [ ]:
```
# Plot the histogram of monthly percentage change
plt.figure(figsize=(10, 6))
plt.hist(df_monthly['Monthly_Return'].dropna(), bins=50, color='orange', edgecolor='black')
plt.xlabel('Monthly Percentage Change')
plt.ylabel('Frequency')
plt.title('Histogram of Monthly Percentage Change')

# Calculate the 5th and 95th percentiles
pct_5 = np.percentile(df_monthly['Monthly_Return'].dropna(), 5)
pct_95 = np.percentile(df_monthly['Monthly_Return'].dropna(), 95)
pct_15 = np.percentile(df_monthly['Monthly_Return'].dropna(), 15)
```

```
pct_85 = np.percentile(df_monthly['Monthly_Return'].dropna(), 85)

# Draw lines for the 90% and 70% intervals around 0
plt.axvline(pct_5, color='red', linestyle='--', linewidth=1.5, label='90% interval')
plt.axvline(pct_95, color='red', linestyle='--', linewidth=1.5)
plt.axvline(pct_15, color='green', linestyle='--', linewidth=1.5, label='70% interval')
plt.axvline(pct_85, color='green', linestyle='--', linewidth=1.5)

plt.legend()
plt.show()
```



In [ ]:
```
df_monthly.tail(10)
```

Out[ ]:

Date	Open	High	Low	Close	Adj Close	Volume	Monthly_Return
2022-08-05 to 2022-08-25	38516.949219	39606.550781	38472.699219	39536.750000	39536.289062	176800	5.455522
2022-09-02 to 2022-09-29	37660.000000	38811.000000	37386.351562	38631.949219	38631.500000	246100	-2.288506
2022-10-07 to 2022-10-27	41265.699219	41353.699219	41106.101562	41307.898438	41307.417969	237200	6.926778
2022-11-04 to 2022-11-24	43122.750000	43332.300781	42880.050781	43231.000000	43230.496094	202000	4.655530
2022-12-02 to 2022-12-29	43401.699219	43422.949219	42833.101562	42986.449219	42985.949219	258700	-0.565684
2023-01-06 to 2023-01-26	40563.851562	40811.648438	40167.699219	40655.050781	40655.050781	289900	-5.423566
2023-02-03 to 2023-02-23	40302.699219	40391.449219	40073.000000	40269.050781	40269.050781	327800	-0.949452
2023-03-03 to 2023-03-30	40231.250000	40690.398438	40180.199219	40608.648438	40608.648438	188000	0.843322
2023-04-07 to 2023-04-27	43045.500000	43302.050781	42810.351562	43233.898438	43233.898438	238700	6.464756
2023-05-05 to 2023-05-25	44277.351562	44498.601562	44207.500000	44436.351562	44436.351562	157600	2.781274

Now we will try to develop a good trading strategy for BNF using chandilier exit and contraction zones.

Backtesting

Finally, the script performs a backtest of the trading strategy, which involves simulating what the returns would have been if the strategy had been followed with an initial capital of 100,000 units. The script assumes that we can take a long position (buy) when the buy signal is triggered and that we should exit the position (sell) when the sell signal is triggered.

In [ ]:
```
!pip install ta
import ta
import yfinance as yf
from itertools import product

# Define parameter ranges
atr_periods = range(1, 21)  # ATR periods to test
atr_multipliers = [i / 2.0 for i in range(21)]  # ATR multipliers to test

# Define performance metrics
```

```python
metrics = ['metric1', 'metric2', 'metric3']  # Add your desired performance metrics here

# Initialize results dictionary
results = {metric: [] for metric in metrics}

# Iterate over parameter combinations
for period, multiplier in product(atr_periods, atr_multipliers):
    # Calculate ATR
    df_2['atr'] = ta.volatility.average_true_range(df_2['High'], df_2['Low'], df_2['Close'], window=period)

    # Calculate Chandelier Exit
    df_2['long_stop'] = df_2['High'].rolling(period).max() - (multiplier * df_2['atr'])
    df_2['short_stop'] = df_2['Low'].rolling(period).min() + (multiplier * df_2['atr'])

    # Generate buy/sell signals
    df_2['buy_signal'] = (df_2['Close'] > df_2['short_stop'].shift()) & (df_2['Close'].shift() < df_2['long_stop'])
    df_2['sell_signal'] = (df_2['Close'] < df_2['long_stop'].shift()) & (df_2['Close'].shift() > df_2['short_stop'])

    # Calculate performance metrics
    # Replace the code below with your own calculations for the desired performance metrics
    metric1 = df_2['buy_signal'].sum()
    metric2 = df_2['sell_signal'].sum()
    metric3 = metric1 / metric2  # Example calculation

    # Append results to the respective metric lists
    results['metric1'].append(metric1)
    results['metric2'].append(metric2)
    results['metric3'].append(metric3)

# Convert results to a DataFrame for analysis
results_df = pd.DataFrame(results)

# Find the parameter combination with the best performance for each metric
best_params = {metric: results_df[metric].idxmax() for metric in metrics}

# Print the optimal parameters for each metric
for metric, best_param in best_params.items():
    print(f"Optimal {metric}: ATR Period={atr_periods[best_param // len(atr_multipliers)]}, "
          f"ATR Multiplier={atr_multipliers[best_param % len(atr_multipliers)]}")
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting ta
  Downloading ta-0.10.2.tar.gz (25 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from ta) (1.22.4)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from ta) (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas->ta) (2.
8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->ta) (2022.7.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->panda
s->ta) (1.16.0)
Building wheels for collected packages: ta
  Building wheel for ta (setup.py) ... done
  Created wheel for ta: filename=ta-0.10.2-py3-none-any.whl size=29088 sha256=d049fdc93f59c6355d7ce7db5feb49bfba079a80
14d17131323b3e32570ebbaa
  Stored in directory: /root/.cache/pip/wheels/47/51/06/380dc516ea78621870b93ff65527c251afdfdc5fa9d7f4d248
Successfully built ta
Installing collected packages: ta
Successfully installed ta-0.10.2
```

```
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
```

```
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
```

```
Optimal metric1: ATR Period=20, ATR Multiplier=0.0
Optimal metric2: ATR Period=20, ATR Multiplier=0.0
Optimal metric3: ATR Period=3, ATR Multiplier=2.0
```

```
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: divide by zero encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
<ipython-input-28-5096a38b8f1a>:33: RuntimeWarning: invalid value encountered in long_scalars
  metric3 = metric1 / metric2  # Example calculation
```

In [ ]:

```
# Define the ticker symbol for Bank Nifty
ticker = "^NSEBANK"

# Set the start and end dates for the data
```

```
start_date = "2013-01-01"
end_date = "2023-05-31"

# Download the daily data for Bank Nifty from Yahoo Finance
df_2 = yf.download(ticker, start=start_date, end=end_date)
# Set the optimal parameters obtained from parameter optimization
optimal_period = 7
optimal_multiplier = 2.5

# Calculate ATR
df_2['atr'] = ta.volatility.average_true_range(df_2['High'], df_2['Low'], df_2['Close'], window=optimal_period)

# Calculate Chandelier Exit
df_2['long_stop'] = df_2['High'].rolling(optimal_period).max() - (optimal_multiplier * df_2['atr'])
df_2['short_stop'] = df_2['Low'].rolling(optimal_period).min() + (optimal_multiplier * df_2['atr'])

# Generate buy/sell signals
df_2['buy_signal'] = (df_2['Close'] > df_2['short_stop'].shift()) & (df_2['Close'].shift() < df_2['long_stop'])
df_2['sell_signal'] = (df_2['Close'] < df_2['long_stop'].shift()) & (df_2['Close'].shift() > df_2['short_stop'])

# Backtest simulation
capital = 100000  # Initial capital
position = 0  # Position in the market (0 for no position, 1 for long, -1 for short)
shares = 0  # Number of shares held
portfolio = []  # Portfolio value over time

for i in range(len(df_2)):
    if df_2['buy_signal'].iloc[i]:
        if position == 0:  # Enter long position
            shares = capital / df_2['Close'].iloc[i]
            capital = 0
            position = 1
    elif df_2['sell_signal'].iloc[i]:
        if position == 1:  # Exit long position
            capital = shares * df_2['Close'].iloc[i]
            shares = 0
            position = 0
    portfolio.append(capital + shares * df_2['Close'].iloc[i])

# Calculate portfolio returns
returns = (portfolio[-1] - portfolio[0]) / portfolio[0] * 100

print(f"Backtest Returns: {returns:.2f}%")
```

```
[**********************100%**********************]  1 of 1 completed
Backtest Returns: 288.24%
```

Conclusion

This script provides a comprehensive workflow for analyzing stock market data, developing trading strategies based on technical indicators, and backtesting those strategies to assess their potential profitability. It is a useful tool for anyone interested in quantitative finance, algorithmic trading, or stock market analysis.

In [ ]: