**Stock Market Trader Bot**

Software Operation Manual

*List of Functions*

by Daniel Leal

Icon

Description automatically generated with low confidence

# TraderBot.py

## **Dataframes Functions**

### IndividualHistoricalData(symbol, startDate, endDate)

**Description**: Gets historical data from yahoo finance of one stock.

**Params**:

symbol (str) – Acronym of stock

startDate(str) – Start date of historical data in format YYYY-MM-DD

endDate(str) – endDate of historical data YYYY-MM-DD

**Returns**: Single dataframe

### HistoricalData(symbolsArray, startDate, endDate)

**Description**: Gets historical dataframes from yahoo finance given a time range.

**Params**:

symbolsArray (list) – Array of stock acronyms

startDate (str) – Start date of historical data in format YYYY-MM-DD

endDate (str) – End date of historical data YYYY-MM-DD

**Returns**: List of dataframes

### NormalizeDfs(dfArray)

**Description**: Normalizes all dataframes inputted.

**Params**:

dfArray (pd.DataFrame or list) – single dataframe or array of dataframes.

**Returns**: List of dataframes

### CombineDfs(dfArray, startDate, endDate)

**Description**: Combines dataframes.

**Params**:

dfArray (list) – Array of dataframes to be combined

startDate (str) – Start date of combined dataframe in format YYYY-MM-DD

endDate (str) – End date of combined dataframe in format YYYY-MM-DD

**Returns**: Dataframe

### StockReturns(dfArray, dailyOrMonthly)

**Description**: Computes the daily, or monthly, return of all dataframes inputted.

**Params**:

dfArray (pd.DataFrame or list) – Single dataframe or array of dataframes

dailyOrMonthly (str) – ‘daily’ or ‘monthly’

**Returns**: List of dataframes

### **Cum**ulativeReturns(dfArray, dailyOrMonthly)

**Description**: Computes the daily, or monthly, cumulative returns of all dataframes inputted.

**Params**:

dfArray (pd.DataFrame or list) – Single dataframe or array of dataframes

dailyOrMonthly (str) – ‘daily’ or ‘monthly’

**Returns**: List of dataframes

## **Statistical Functions**

### GetMaxClose(dfArray)

**Description**: Gets the max closing prices of a stock or set of stocks.

**Params**:

dfArray (pd.DataFrame or list) – Single dataframe or array of dataframes

**Returns**: Dict (e.g., symbol: MaxPrice)

## **Slice Functions**

### SliceRow(dfArray, startDate, endDate)

**Description**: Slices dataframes based on specific rows.

**Params**:

dfArray (pd.DataFrame or list) – Single dataframe or array of dataframes

startDate (str) – Start date of sliced dataframe in format YYYY-MM-DD

endDate (str) – End date of sliced dataframe in format YYYY-MM-DD

**Returns**: List of dataframes.

### SliceColumn(dfArray, colArray)

**Description**: Slices dataframes based on specific columns.

**Params**:

dfArray (pd.DataFrame or list) – Single dataframe or array of dataframes

colArray (str or list) – Single string or list of strings of column names.

**Returns**: List of dataframes.

## **Validation Functions**

### ValidateDates(startDate, endDate)

**Description**: Checks if startDate to endDate is a valid timeframe.

**Params**:

startDate (str) – Start date in format YYYY-MM-DD

endDate (str) – End date in format YYYY-MM-DD

**Returns**: Invalid or Valid output message

## **Plot Functions**

### PlotData(dfArray, title, x, y)

**Description**: Plots any inputted dataframes on one graph.

**Params**:

dfArray (pd.DataFrame or list) – Single dataframe or array of dataframes

title (str) – Title for graph

x (str) – Label for x-axis

y (str) – Label for y-axis

**Returns**: One graph

### PlotRollingMean(dfArray, window)

**Description**: Plots rolling mean along with standard plot of all inputted dataframes. Will print out an individual graph for each dataframe.

**Params**:

dfArray (pd.DataFrame or list) – Single dataframe or array of dataframes

window (int) – Frequency of mean computation

**Returns**: Individual graph for each dataframe inputted.

### PlotBollingerBands(dfArray, window)

**Description**: Plots standard plot, Bollinger bands, and moving average of all dataframes inputted.

**Params**:

dfArray (pd.DataFrame or list) – Single dataframe or array of dataframes

window (int) – Frequency of mean computation

**Returns**: Individual graph for each dataframe inputted.

### PlotHistogram(dfArray, plotStatisticsYesOrNo, bin)

**Description**: Plots histograms of multiple dataframes at once. If less than two dataframes is inputted, statistics such as mean, standard deviation, and kurtosis can be printed.

**Params**:

dfArray (pd.DataFrame or list) – Single dataframe or array of dataframes

plotStatisticsYesOrNo (str) – ‘yes’ or ‘no’

bin (int) – Frequency in which bars are computed

**Returns**: One graph with all dataframes in it.

### PlotScatter(dfArray)

**Description**: Plots a scatter plot of two dataframes.

**Params**:

dfArray (pd.DataFrame) – Single, combined dataframe

**Returns**: Scatter plot with a dataframe symbol represented on each axis.

### PlotCorrelationMatrix(dfArray)

**Description**: Plots a correlation matrix of all dataframes inputted.

**Params**:

dfArray (pd.DataFrame) – Single, combined dataframe

**Returns**: One correlation matrix image.

## **Portfolio Functions**

### ComputePortfolioValue(startValue, startDate, endDate, symbols, allocations)

**Description**: Computes the value of a portfolio over a given time frame.

**Params**:

startValue (int or float) – Value in dollars of total invested in portfolio

startDate (str) – Start date of portfolio investment in format YYYY-MM-DD

endDate (str) – End date of portfolio investment in format YYYY-MM-DD

symbols (list) – Array of symbols in portfolio

allocations (list) – Array of allocated numbers of representing percentage of startValue

per respective symbol. When elements of array are added together, they must equal to 1.

**Returns**: Dataframe

### ComputeSharpeRatio(historicalData, k, startDate, endDate, allocations)

**Description**: Computes either the annual, weekly, or daily sharpe ratio.

**Params**:

historicalData (pd.DataFrame or list) – Single historical dataframe or array of historical

dataframes

startDate (str) – Start date in format YYYY-MM-DD

endDate (str) – End date in format YYYY-MM-DD

allocations (list) – Array of allocations respective to symbols in historicalData dataframes. When elements of the array are added together, theymust equal to 1.

**Returns**: List of symbol and sharpe ratio.

### OptimizePortfolio(symbols, numOfSim, startDate, endDate)

**Description**: Computes the best set of allocations to effectively optimize portfolio profits using Monte Carlo Simulations.

**Params**:

Symbols (list) – List of strings of the stock symbol acronyms

numOfSim (int) – Number of simulations

startDate (str) – Start date in format YYYY-MM-DD

endDate (str) – End date in format YYYY-MM-DD

**Returns**: List of optimized allocations

### ScipyOptimizePortfolio(symbols, startDate, endDate)

**Description**: Computes the best set of allocations to effectively optimize portfolio profits using Scipy.optimize.minimize function.

**Params**:

Symbols (list) – List of strings of the stock symbol acronyms

startDate (str) – Start date in format YYYY-MM-DD

endDate (str) – End date in format YYYY-MM-DD

**Returns**: List of optimized allocations