19th 10am | Portfolio Risk Analytics

0.0.1 Day Trading | Portfolio Management

[73]:
```
import os
import time
import warnings
import datetime
import numpy as np
import pandas as pd
import scipy.stats as scs
from pylab import plt, mpl
import numpy.random as npr

plt.style.use('seaborn-v0_8')
mpl.rcParams['savefig.dpi'] = 300
mpl.rcParams['font.family'] = 'serif'
pd.set_option('mode.chained_assignment', None)
pd.set_option('display.float_format', '{:.4f}'.format)
np.set_printoptions(suppress=True, precision=4)
os.environ['PYTHONHASHSEED'] = '0'

warnings.filterwarnings("ignore")
```

Data

We will analyze several historical financial time series data.

[74]:
```
dataFrame = pd.read_csv('data/_all_risk_factors_data_1.csv',
                        index_col=0, parse_dates=True).dropna()
```

[75]:
```
columns = ['CDNS', 'CELH', 'CHRD', 'FICO', 'GLD',  'NVDA', '^HUI', '^IRX',␣
 ↪'^IXIC', '^NYA']

noOfAssets = len(columns)

dFrame = dataFrame[columns].dropna()
dFrame.info()# display dataset information 2015-08-05 to 2023-09-29
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1883 entries, 2016-01-04 to 2023-06-29
```

```
Data columns (total 10 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   CDNS    1883 non-null   float64
 1   CELH    1883 non-null   float64
 2   CHRD    1883 non-null   float64
 3   FICO    1883 non-null   float64
 4   GLD     1883 non-null   float64
 5   NVDA    1883 non-null   float64
 6   ^HUI    1883 non-null   float64
 7   ^IRX    1883 non-null   float64
 8   ^IXIC   1883 non-null   float64
 9   ^NYA    1883 non-null   float64
dtypes: float64(10)
memory usage: 161.8 KB
```

[76]:
```
dFrame.head() # display the first five rows
```

[76]:
```
              CDNS    CELH    CHRD    FICO      GLD    NVDA     ^HUI    ^IRX  \
Date
2016-01-04 20.5300 1.9700 7.5800 93.0600 102.8900 8.0925 114.6600 0.1550
2016-01-05 20.5400 2.0000 7.4500 94.0000 103.1800 8.2225 114.0500 0.2050
2016-01-06 20.2000 1.8700 6.9000 94.0000 104.6700 7.8825 116.7300 0.2050
2016-01-07 19.7500 1.8700 6.3800 91.4600 106.1500 7.5700 122.2400 0.1900
2016-01-08 19.2400 1.8700 6.4700 88.9600 105.6800 7.4075 118.5400 0.1900

               ^IXIC       ^NYA
Date
2016-01-04 4903.0898 10001.5596
2016-01-05 4891.4302 10028.0596
2016-01-06 4835.7598  9868.2598
2016-01-07 4689.4302  9650.4199
2016-01-08 4643.6299  9528.7695
```

0.0.2 Rename Columns

[77]:
```
dFrame.rename(columns={'CDNS': 'CDNS_Risk', 'CELH': 'CELH_Risk',   'CHRD':␣
 ↪'CHRD_Risk',
                       'FICO': 'FICO_Risk', 'GLD': 'GLD_ETF_Risk', 'NVDA':␣
 ↪'NVDA_Risk',
                       '^HUI': 'NYSE_ARCA_Risk', '^IRX': 'I_Risk', '^IXIC':␣
 ↪'Nasdaq_Index_Risk',
                       '^NYA': 'NYSE_Index_Risk'}, inplace=True)
```

Log returns

[78]:
```
riskF_log_returns = np.log(dFrame / dFrame.shift(1))
riskF_log_returns.dropna(inplace=True)
riskF_log_returns.head()
```

[78]:

Date	CDNS_Risk	CELH_Risk	CHRD_Risk	FICO_Risk	GLD_ETF_Risk	NVDA_Risk	NYSE_ARCA_Risk	I_Risk	Nasdaq_Index_Risk	NYSE_Index_Risk
2016-01-05	0.0005	0.0151	-0.0173	0.0101	0.0028	0.0159	-0.0053	0.2796	-0.0024	0.0026
2016-01-06	-0.0167	-0.0672	-0.0767	0.0000	0.0143	-0.0422	0.0232	0.0000	-0.0114	-0.0161
2016-01-07	-0.0225	0.0000	-0.0784	-0.0274	0.0140	-0.0405	0.0461	-0.0760	-0.0307	-0.0223
2016-01-08	-0.0262	0.0000	0.0140	-0.0277	-0.0044	-0.0217	-0.0307	0.0000	-0.0098	-0.0127
2016-01-11	0.0021	-0.0108	-0.0475	0.0043	-0.0089	0.0017	-0.0409	-0.0822	-0.0012	-0.0001

0.0.3 Risk factor contribution

[79]:
```
riskF_log_returns = np.log(dFrame / dFrame.shift(1))
riskF_log_returns.dropna(inplace=True)
riskF_log_returns.head()

risk_factor_1 = riskF_log_returns['CDNS_Risk'].std()
risk_factor_2 = riskF_log_returns['CELH_Risk'].std()
risk_factor_3 = riskF_log_returns['CHRD_Risk'].std()
risk_factor_4 = riskF_log_returns['FICO_Risk'].std()
risk_factor_5 = riskF_log_returns['GLD_ETF_Risk'].std()
risk_factor_6 = riskF_log_returns['NVDA_Risk'].std()
risk_factor_7 = riskF_log_returns['NYSE_ARCA_Risk'].std()
risk_factor_8 = riskF_log_returns['I_Risk'].std()
risk_factor_9 = riskF_log_returns['Nasdaq_Index_Risk'].std()
risk_factor_10 = riskF_log_returns['NYSE_Index_Risk'].std()
```

```
position_weights =  [0.119, 0.0819, 0.0021, 0.0301, 0.3066, 0.1603, 0.3]


positions = [
    {"position_weights": position_weights[0], "risk_factor_1": risk_factor_1,␣
 ↪"risk_factor_2": 0.0, "risk_factor_3": 0.0, "risk_factor_4": 0.0,␣
 ↪"risk_factor_5": 0.0, "risk_factor_6": 0.0, "risk_factor_7": 0.0,␣
 ↪"risk_factor_8": 0.0, "risk_factor_9": risk_factor_9, "risk_factor_10": 0.0},
    {"position_weights": position_weights[1], "risk_factor_1": 0.0,␣
 ↪"risk_factor_2": risk_factor_2, "risk_factor_3": 0.0, "risk_factor_4": 0.0,␣
 ↪"risk_factor_5": 0.0, "risk_factor_6": 0.0, "risk_factor_7": 0.0,␣
 ↪"risk_factor_8": 0.0, "risk_factor_9": risk_factor_9, "risk_factor_10": 0.0},
    {"position_weights": position_weights[2], "risk_factor_1": 0.0,␣
 ↪"risk_factor_2": 0.0, "risk_factor_3": risk_factor_3, "risk_factor_4": 0.0,␣
 ↪"risk_factor_5": 0.0, "risk_factor_6": 0.0, "risk_factor_7": 0.0,␣
 ↪"risk_factor_8": 0.0, "risk_factor_9": risk_factor_9, "risk_factor_10": 0.0},
    {"position_weights": position_weights[3], "risk_factor_1": 0.0,␣
 ↪"risk_factor_2": 0.0, "risk_factor_3": 0.0, "risk_factor_4": risk_factor_4,␣
 ↪"risk_factor_5": 0.0, "risk_factor_6": 0.0, "risk_factor_7": 0.0,␣
 ↪"risk_factor_8": 0.0, "risk_factor_9": 0.0, "risk_factor_10":␣
 ↪risk_factor_10},
    {"position_weights": position_weights[4], "risk_factor_1": 0.0,␣
 ↪"risk_factor_2": 0.0, "risk_factor_3": 0.0, "risk_factor_4": 0.0,␣
 ↪"risk_factor_5": risk_factor_5, "risk_factor_6": 0.0, "risk_factor_7":␣
 ↪risk_factor_7, "risk_factor_8": 0.0, "risk_factor_9": 0.0, "risk_factor_10":␣
 ↪0.0},
    {"position_weights": position_weights[5], "risk_factor_1": 0.0,␣
 ↪"risk_factor_2": 0.0, "risk_factor_3": 0.0, "risk_factor_4": 0.0,␣
 ↪"risk_factor_5": 0.0, "risk_factor_6": risk_factor_6, "risk_factor_7": 0.0,␣
 ↪"risk_factor_8": 0.0, "risk_factor_9": risk_factor_9, "risk_factor_10": 0.0},
    {"position_weights": position_weights[6], "risk_factor_1": 0.0,␣
 ↪"risk_factor_2": 0.0, "risk_factor_3": 0.0, "risk_factor_4": 0.0,␣
 ↪"risk_factor_5": 0.0, "risk_factor_6": 0.0, "risk_factor_7": 0.0,␣
 ↪"risk_factor_8": risk_factor_8, "risk_factor_9": 0.0, "risk_factor_10": 0.0},

]

positions

# Risk Factor matrix
risk_factors_matrix = np.array([[position[f"risk_factor_{i+1}"] for i in␣
 ↪range(10)] for position in positions])

# Position weights
weights = np.array([position["position_weights"] for position in positions])
```

```
# Risk Factor covariance  matrix
covariance_matrix = np.cov(risk_factors_matrix)

# Overal Portfolio Volatility
portfolio_volatility = np.sqrt(np.dot(weights, np.dot(covariance_matrix,␣
 ↪weights.T)))

print("")
print("Portfolio Daily Volatility: ", (portfolio_volatility.round(5)))
```

```
Portfolio Daily Volatility:  0.01339
```

[80]:
```
# Calculating Portfolio VaR: 5% chance of understating the realized loss
z_score = 1.645  # Z-score for a 5%
VaR = z_score * portfolio_volatility

print("")
print("Portfolio VaR: ", (VaR.round(5)))
```

```
Portfolio VaR:  0.02202
```

[ ]:
```

```