Quantitative Portfolio Management

Assignment #8

Raman Uppal
EDHEC Business School

November 2023

Instructions for each assignment . . . I

- Assignment #1 should be done individually.
- The other assignments are to be done in groups of 4 or 5 students.
  - This means that groups of 1, 2, 3, 6, etc. are not allowed.
  - Diversity in groups is strongly encouraged
    (people from different countries, different genders, different finance knowledge, and different coding ability, etc.)

Instructions for each assignment . . . II

- Each assignment should be emailed as a Jupyter file
  - To `Raman.Uppal@edhec.edu`
  - The subject line of the email should be: "QPM: Assignment n," where $n = \{1, 2, \ldots, 8\}$.
  - Assignment n is due before Lecture n, where $n = \{1, 2, \ldots, 8\}$.
  - Assignments submitted late will not be accepted (grade $= 0$), so please do not email me assignments after the deadline.

Instructions for each assignment . . . III

- The Jupyter file should include the following (use Markdown):
  - Section "0" with information about your submission:
    - Line 1: QPM: Assignment n
    - Line 2: Group members: listed alphabetically by last name, where the last name is written in CAPITAL letters
    - Line 3: Any comments/challenges about the assignment
  - Section "k" where $k = \{1, 2, \dots\}$.
    - First type Question k of Assignment n.
    - Then, below the question, provide your answer.
    - Your code should include any packages that need to be imported.

Data for Assignment 8

- The data file for this assignment has monthly returns for nine firm-specific characteristics: Market, SMB, HML, RMW, CMA, UMD, ROE, IA, BAB.
  - This data is the same as the one for the last assignment.
- The first five characteristics (Market, SMB, HML, RMW, CMA) are from Fama and French (2015), the sixth (UMD) is from Carhart (1997), the profitability (ROE) and investment (IA) factors are from Hou, Xue, and Zhang (2015), and the betting-against-beta (BAB) factor is from Frazzini and Pedersen (2014).
- All factors are returns in excess of the risk-free rate.
  - In particular, every factor (besides MKT and BAB) is the return of a long-short portfolio of stocks with $1 in the long leg and $1 in the short leg, and thus, their returns equal their excess returns.
  - The MKT and BAB factors are also long-short portfolios because they are returns in excess of the risk-free rate.

Instructions for Assignment 8

- Use an estimation window of 120 months. Therefore, to facilitate comparison, the in-sample and out-of-sample performance should be evaluated from January 1977 to December 2020.
- In the original papers on volatility timing, volatility is computed using daily returns data. Because I have not given you daily data, please estimate current volatility using monthly data for the last 12 months.
- Define f_{t+1} to be an excess return
- Construct a new volatility-managed factor, whose return is

$$f_{t+1}^{\sigma} = \frac{c}{\sigma_t^2(f)} \times f_{t+1}, \quad \text{where}$$

  - $\sigma_t(f)$ is the previous 12 month's realized volatility, estimated using monthly data
  - choose c so f^{σ} has the same unconditional volatility as f; (if it is difficult to understand how to compute c, set $c = 1$).

List of questions

Q8.1 Please use mean-variance optimization to combine

- The original (without timing) factor, f_{t+1};
- The volatility-managed version of this factor, f^{σ}_{t+1}.

Q8.2 Compare the Sharpe ratios of the combined position that includes

- the portfolio with just the original factor and
- the portfolio that includes the volatility-timed factor.

Q8.3 What do you conclude from your analysis above?

Q8.4 Please list the limitations of your analysis. Could one implement this volatility-timing policy in practice?

Discussion of Assignment 8: Initial setup and data download

Code to import libraries and download data

```
# Import the libraries we need
import numpy as np
import pandas as pd

# To format numbers in pandas dataframes set up format for entire file
pd.options.display.float_format = '{:,.4f}'.format

# Import the data we will use
data = pd.read_excel("QPM-FactorsData.xlsx", header=0, index_col=0)
data.index = pd.to_datetime(data.index, format='%Y%m')
```

Discussion of Assignment 8: Preliminary analysis

- To set the stage, we compute the unconditional Sharpe ratios of the individual factors (i.e., without conditioning on volatility)

Code to compute unconditional Sharpe ratios of the factors

```
# Unconditional Sharpe ratio (i.e., without conditioning on volatility)
start_date = pd.to_datetime('1977-02-01')
SR_uncon = data.loc[data.index >= start_date, :].mean() / data.loc[data.
    index >= start_date, :].std() * np.sqrt(12)

# Put the results in a dataframe
summary = pd.DataFrame({"SR\_unconditional":SR_uncon})
summary = summary.T
summary
```

	Market	SMB	HML	RMW	CMA	UMD	ROE	IA	BAB
SR_unconditional	0.5305	0.2078	0.1696	0.5055	0.3985	0.4740	0.7224	0.5080	0.8796

Discussion of Assignment: Q8.1

Q8.1 Please use mean-variance optimization to combine

- The original (without timing) factor, f_{t+1};
- The volatility-managed version of this factor, f^{σ}_{t+1}.

Code for in-sample analysis conditioning on volatility

```
# We denote ‘realized variance‘ for the market return as ‘RV‘
# Note that here we are using realized variance based on monthly returns
# In the original papers, it was realized variance based on DAILY
    returns

RV = data.rolling(12,closed = "left").var()

# Calculate the constant
c = 1/(1/RV).mean()

# Calculate weights for the risky assets (factors)
weight_is = c/RV # in-sample volatility multiplier

# Calculate return of the strategy
r_str_is = weight_is*data            # ... code continues on next page
```

Discussion of Assignment: Q8.1 (continued)

Code to combine factor with its volatility-timed counterpart

```
# Create an empty dataframe
combine_r_is = pd.DataFrame()

# Combine optimally each factor with its volatility-timed counterpart
for i in r_str_is.columns:
    gamma = 5
    combine_df = pd.merge(data.loc[data.index >= start_date, :][i],
     r_str_is[i],left_index=True,right_index=True)
    mu_df = combine_df.mean()
    combine_V = combine_df.cov()
    w_df = (1/gamma) * (mu_df @ np.linalg.inv(combine_V))
    combine_r_is[i] = w_df[0] * data[i] + w_df[1] * r_str_is[i]

# Calculate in-sample Sharpe ratio
SR_combine_is = combine_r_is.mean()/combine_r_is.std() * np.sqrt(12)
```

Output for Q8.1

Code to put output in a dataframe

```
# Put results in a dataframe
summary = pd.DataFrame({"SR\_unconditional":SR_uncon,"SR\_combined\_in\
    _sample":SR_combine_is})
summary = summary.T
summary
```

	Market	SMB	HML	RMW	CMA	UMD	ROE	IA	BAB
SR_combined_in_sample	0.4405	0.1430	0.3091	0.5141	0.5015	0.7176	0.9971	0.6464	1.0010

Discussion of Assignment: Q8.2

Q8.2 Compare the Sharpe ratios of

- the portfolio with just the original factor and
- the portfolio that includes the volatility-timed factor.

Code for Sharpe ratios of unconditional and combined (in-sample) strategy

```
# Compare Sharpe ratios of unconditional & combined (in-sample) strategy
summary = pd.DataFrame({"SR\_unconditional":SR_uncon,"SR\_combined\_in\
    _sample":SR_combine_is})
summary = summary.T
summary
```

	Market	SMB	HML	RMW	CMA	UMD	ROE	IA	BAB
SR_unconditional	0.5305	0.2078	0.1696	0.5055	0.3985	0.4740	0.7224	0.5080	0.8796
SR_combined_in_sample	0.4405	0.1430	0.3091	0.5141	0.5015	0.7176	0.9971	0.6464	1.0010

Discussion of Assignment: Q8.3

Q8.3 What do you conclude from your analysis above?

	Market	SMB	HML	RMW	CMA	UMD	ROE	IA	BAB
SR_unconditional	0.5305	0.2078	0.1696	0.5055	0.3985	0.4740	0.7224	0.5080	0.8796
SR_conditional_in_sample	0.4405	0.1430	0.3091	0.5141	0.5015	0.7176	0.9971	0.6464	1.0010

- From the table above, reproduced from the previous page, we see that volatility timing has the potential to improve the Sharpe ratio;
  - for many of the factors, the combined portfolio of the unconditional and conditional factors, has a higher Sharpe ratio than the unconditional factor (HML, RMW, CMA, UMD, ROE, IA, and BAB).
  - Performance can be further improved by conditioning not on monthly realized volatility for the last 12 months (as we have done here), but by conditioning on daily realized volatility for the last month.

Discussion of Assignment: Q8.4

Q8.4 Please list the limitations of your analysis. Could one implement this volatility-timing policy in practice?

- There are several limitations of the analysis presented above:
  1. Conditioning on monthly instead of daily realized volatility is not good;
  2. The entire exercise is done in-sample; that is, the choice of c suffers from look-ahead bias
  3. The entire exercise ignores transaction costs.
  4. Similarly, there is look-ahead bias when choosing the weights in the "combined" strategy, because the moments of asset returns are computed using all of the data, not just historical data.
     - The next page shows that out-of-sample performance is less good.

Discussion of Assignment: Q8.4 (code for out-of-sample analysis)

- Note you were not expected to compute out-of-sample Sharpe ratio.

Code for computing out-of-sample Sharpe ratio

```
combine_r_oos = pd.DataFrame()

for i in data.index[120:]: #start from the 120th month
    start_month = i - pd.DateOffset(months = 120)
    end_month = i - pd.DateOffset(months = 1)

    Ret_df = data.loc[start_month:end_month] #rolling original return
    r_str_df = r_str_is.loc[start_month:end_month] # rolling vol-
     strategy return

    for j in Ret_df.columns:
        gamma = 5
        combine_df = pd.merge(Ret_df[j], r_str_df[j], left_index = True,
      right_index = True)
        combine_V = combine_df.cov()
        mu_df = combine_df.mean()
        w_df = (1/gamma) * (mu_df @ np.linalg.inv(combine_V))
        combine_r_oos.loc[i,j] = w_df[0] * data.loc[i,j] + w_df[1] *
     r_str_is.loc[i,j]

SR_OOS = combine_r_oos.mean()/combine_r_oos.std() * np.sqrt(12)
```

Discussion of Assignment: Q8.4 (output of out-of-sample analysis)

Code for printing the results of the three strategies

```
# Summary performance of nine factors for the three strategies
summary = pd.DataFrame({"SR\_unconditional":SR_uncon,
                        "SR\_combined\_in\_sample":SR_combine_is,
                        "SR\_combined\_out\_of\_sample":SR_OOS})
summary = summary.T
summary
```

	Market	SMB	HML	RMW	CMA	UMD	ROE	IA	BAB
SR_unconditional	0.5305	0.2078	0.1696	0.5055	0.3985	0.4740	0.7224	0.5080	0.8796
SR_combined_in_sample	0.4405	0.1430	0.3091	0.5141	0.5015	0.7176	0.9971	0.6464	1.0010
SR_combined_out_of_sample	0.3865	-0.1770	0.5530	0.2447	0.2206	0.6134	0.9657	0.5456	0.7336

- Other than for HML, performance is much poorer out of sample.
- Performance will be even weaker after we net transaction costs.

Bibliography

Carhart, M. M. 1997. On persistence in mutual fund performance. *Journal of Finance* 52, no. 1 (March): 57–82. (Cited on page 5).

Fama, E. F., and K. R. French. 2015. A five-factor asset pricing model. *Journal of Financial Economics* 116 (1): 1–22. (Cited on page 5).

Frazzini, A., and L. H. Pedersen. 2014. Betting against beta. *Journal of Financial Economics* 111 (1): 1–25. (Cited on page 5).

Hou, K., C. Xue, and L. Zhang. 2015. Digesting anomalies: An investment approach. *Review of Financial Studies* 28 (3): 650–705. (Cited on page 5).

End of assignment