Problem Set 3: Quantitative Asset Management

Name: Tanvi Johri
Student ID: 806555315
File: PS3_806555315.py

This project replicates the methodology and results presented in Daniel and Moskowitz (2016) by constructing and analyzing momentum portfolios using CRSP stock return data and Fama-French factors. The goal is to compute monthly momentum signals for U.S. equities, assign stocks to decile portfolios based on past return performance, and calculate value-weighted portfolio excess returns. The replication uses both the Daniel-Moskowitz (DM) equal-firm decile sorting method and the Ken French (KRF) NYSE-breakpoint-based sorting method. Each step is aligned to the procedures outlined in the paper, including the use of (12,2) momentum signals, proper treatment of delisting returns, and construction of WML (Winner Minus Loser) portfolios. The analysis culminates in the replication of Table 1 statistics for Daniel and Moskowitz momentum portfolios and Ken French momentum portfolios separately.

Question 1: Construct Monthly Stock Panel (Momentum Signal Preparation)

This question prepares the CRSP stock panel by calculating returns (including delisting returns), computing market capitalization, and generating the (12,2) momentum signal.

This PS3_Q1 function takes the raw CRSP stock return data and prepares it for portfolio formation by cleaning, filtering, computing total returns, calculating market cap, and generating the ranking return based on the (12,2) momentum signal. It returns a cleaned panel of stocks with computed returns and momentum signal for each stock-month.

Input – Data frame "CRSP_Stocks" with columns: PERMNO, date, SHRCD, EXCHCD, RET, DLRET, PRC, SHROUT

Output – Data frame "CRSP_Stocks_Momentum" with each row corresponding to a PERMNO/Year/Month, with columns Year, Month, PERMNO, EXCHCD, lag_Mkt_Cap, Ret, Ranking_Ret

Step 1: Copy the CRSP dataset to avoid modifying the original data directly.

Step 2: Convert key columns (RET, DLRET, PRC, SHROUT) to numeric, coercing errors to NaN to prevent data contamination.

Step 3: Compute total return using CRSP's standard formula: (1 + RET) * (1 + DLRET) - 1 if DLRET is available, else use RET.

Step 4: Filter for common stocks traded on NYSE, AMEX, or NASDAQ by checking SHRCD and EXCHCD values.

Step 5: Calculate market capitalization in millions of dollars using absolute price times shares outstanding, divided by 1,000.

Step 6: Extract year and month from the date column for later grouping. Also ensure the date is a proper datetime object and sort the data by PERMNO and date.

Step 7: Compute lagged market capitalization (t−1) and lagged price at t−13 for validation of the momentum window.

Step 8: Compute log returns using log1p for numerical stability and suppress runtime warnings when return equals −1 (total loss). Then compute the rolling sum of log returns from t−12 to t−2 using a window of 11 months. Lag by one period to exclude the most recent month.

Step 9: Apply final filters: ensure valid lagged price and restrict to stocks with lagged market cap > $15 million. Limit the sample to the years 1927 through 2024.

Step 10: Output the CRSP_Stocks_Momentum cleaned dataset with Year, Month, PERMNO, EXCHCD, lagged market cap, return, and ranking return

The resulting output is:

Notebook version:

	Year	Month	PERMNO	EXCHCD	lag_Mkt_Cap	Ret	Ranking_Ret
0	1991	12	10001	3.0	1.582675e+01	-0.006780	0.481465
1	1992	1	10001	3.0	1.558750e+01	-0.051724	0.461590
2	1992	10	10001	3.0	1.612500e+01	-0.025000	0.189838
3	1992	11	10001	3.0	1.572188e+01	-0.017094	0.038227
4	1992	12	10001	3.0	1.545312e+01	-0.015130	0.027788
...	...	...	...	...	...	...	...
2687040	2024	8	93436	3.0	7.413801e+05	-0.077390	-0.106230
2687041	2024	9	93436	3.0	6.840044e+05	0.221942	-0.155850
2687042	2024	10	93436	3.0	8.390474e+05	-0.045025	0.264423
2687043	2024	11	93436	3.0	8.020335e+05	0.381469	0.039890
2687044	2024	12	93436	3.0	1.107984e+06	0.170008	0.328647

2687045 rows × 7 columns

.py version:

```
-----------PS3 - Output 1------------
         Year  Month  PERMNO  EXCHCD   lag_Mkt_Cap       Ret  Ranking_Ret
0        1991     12   10001     3.0  1.582675e+01 -0.006780     0.481465
1        1992      1   10001     3.0  1.558750e+01 -0.051724     0.461590
2        1992     10   10001     3.0  1.612500e+01 -0.025000     0.189838
3        1992     11   10001     3.0  1.572188e+01 -0.017094     0.038227
4        1992     12   10001     3.0  1.545312e+01 -0.015130     0.027788
...       ...    ...     ...     ...           ...       ...          ...
2687040  2024      8   93436     3.0  7.413801e+05 -0.077390    -0.106230
2687041  2024      9   93436     3.0  6.840044e+05  0.221942    -0.155850
2687042  2024     10   93436     3.0  8.390474e+05 -0.045025     0.264423
2687043  2024     11   93436     3.0  8.020335e+05  0.381469     0.039890
2687044  2024     12   93436     3.0  1.107984e+06  0.170008     0.328647

[2687045 rows x 7 columns]
```

<u>Question 2: Assign Momentum Deciles (DM and KRF)</u>

This question sorts firms into momentum deciles using two methods: Daniel-Moskowitz (DM) and Ken French (KRF).

This is done through the PS3_Q2 function that assigns each stock-month to a decile based on its past (12,2) momentum signal, using two sorting methodologies: Daniel-Moskowitz (DM) using equal-firm quantiles, and Ken French (KRF) using NYSE breakpoints. It returns the decile labels for each firm.

Input – Data frame "CRSP_Stocks_Momentum" the output of PS3 Q1

Output – Data frame "CRSP_Stocks_Momentum_decile" with each row corresponding to a stock-year-month with columns Year, Month, PERMNO, lag_Mkt_Cap, Ret, DM_decile, KRF_decile, EXCHCD

Step 1: Generate a datetime column for monthly grouping. Sort the dataset by year, month, and ranking return.

Step 2: Group data by month. For each month:

- Drop observations with missing ranking return.
- Create the investable universe using lagged market cap > $15M.
- Use pd.qcut on the investable universe to create 10 DM quantiles, then assign decile labels to all firms based on the resulting breakpoints.
- For KRF deciles, use only NYSE stocks from the investable universe to create breakpoints, then apply these breakpoints to the full cross-section.

Step 3: Store decile labels for each date and PERMNO. Concatenate the monthly results and merge them back into the main dataframe.

Step 4: Return the CRSP_Stocks_Momentum_decile full dataset with lagged market cap, return, EXCHCD, and assigned decile labels.

The resulting output is:

Notebook version:

	Year	Month	PERMNO	lag_Mkt_Cap	Ret	DM_decile	KRF_decile	EXCHCD
0	1927	1	15245	15.750000	-0.047619	1	1.0	1.0
1	1927	1	10524	16.750000	-0.089552	1	1.0	1.0
2	1927	1	10196	19.095750	-0.018519	1	1.0	1.0
3	1927	1	11789	31.728125	-0.048951	1	1.0	1.0
4	1927	1	11578	21.145500	-0.101351	1	1.0	1.0
...	...	...	...	...	...	...	...	...
2687040	2024	12	20975	1104.525240	-0.119173	10	NaN	3.0
2687041	2024	12	19142	118.913340	-0.023524	10	NaN	3.0
2687042	2024	12	21090	918.354250	0.091759	10	NaN	3.0
2687043	2024	12	24100	2377.536210	-0.396741	10	NaN	3.0
2687044	2024	12	19956	2153.451690	-0.019518	10	NaN	3.0

2687045 rows × 8 columns

.py version:

```
-----------PS3 - Output 2------------
         Year  Month  PERMNO  lag_Mkt_Cap       Ret  DM_decile  KRF_decile  EXCHCD
0        1927      1   15245    15.750000 -0.047619          1         1.0     1.0
1        1927      1   10524    16.750000 -0.089552          1         1.0     1.0
2        1927      1   10196    19.095750 -0.018519          1         1.0     1.0
3        1927      1   11789    31.728125 -0.048951          1         1.0     1.0
4        1927      1   11578    21.145500 -0.101351          1         1.0     1.0
...       ...    ...     ...          ...       ...        ...         ...     ...
2687040  2024     12   20975  1104.525240 -0.119173         10         NaN     3.0
2687041  2024     12   19142   118.913340 -0.023524         10         NaN     3.0
2687042  2024     12   21090   918.354250  0.091759         10         NaN     3.0
2687043  2024     12   24100  2377.536210 -0.396741         10         NaN     3.0
2687044  2024     12   19956  2153.451690 -0.019518         10         NaN     3.0

[2687045 rows x 8 columns]
```

<u>Question 3: Compute Value-Weighted Excess Returns by Decile</u>

This question constructs value-weighted momentum portfolio returns.

The PS3_Q3 function calculates monthly value-weighted excess returns for each momentum decile using the assigned deciles and the Fama-French risk-free rate. It produces a long-format dataset of excess returns by decile for both DM and KRF methods.

Input – Data frame “CRSP_Stocks_Momentum_decile”, the output of PS3 Q2

Dataframe FF mkt (as defined in PS1 Q2), with columns Year, Month, Market minus Rf, SMB, HML, Rf

Output – Data frame “CRSP_Stocks_Momentum_returns” with each row corresponding to a year-month-decile with columns Year, Month, decile, DM_Ret, KRF_Ret, Rf

Step 1: Drop rows with missing required fields (return, lagged market cap, or decile labels). Convert DM and KRF decile columns to integer type for grouping. Merge the risk-free rate from the Fama-French dataset using Year and Month.

Step 2: Compute excess return as the firm’s return minus the risk-free rate. Group the dataset by year, month, and decile for both DM and KRF.

Step 3: Within each decile group, calculate the value-weighted average of excess returns using lagged market cap as weights.

Step 4: Merge the DM and KRF value-weighted return series into a single dataset. Also merge the risk-free rate again to keep it available in the final output, “CRSP_Stocks_Momentum_returns”. Sort by Year, Month, and decile.

The resulting output is:

Notebook version:

	Year	Month	decile	DM_Ret	KRF_Ret	Rf
0	1927	1	1	-0.022619	-0.022619	0.0025
1	1927	1	2	-0.008032	-0.008032	0.0025
2	1927	1	3	-0.004554	-0.004554	0.0025
3	1927	1	4	-0.007277	-0.007277	0.0025
4	1927	1	5	0.003399	0.003399	0.0025
...	...	...	...	...	...	...
11755	2024	12	6	-0.025927	-0.002948	0.0037
11756	2024	12	7	-0.028186	-0.069167	0.0037
11757	2024	12	8	0.013648	0.027621	0.0037
11758	2024	12	9	-0.050267	-0.048984	0.0037
11759	2024	12	10	-0.050475	-0.052058	0.0037

11760 rows × 6 columns

.py version:

```
-----------PS3 - Output 3------------
       Year  Month  decile    DM_Ret   KRF_Ret      Rf
0      1927      1       1 -0.022619 -0.022619  0.0025
1      1927      1       2 -0.008067 -0.008067  0.0025
2      1927      1       3 -0.004554 -0.004554  0.0025
3      1927      1       4 -0.007277 -0.007277  0.0025
4      1927      1       5  0.003399  0.003399  0.0025
...     ...    ...     ...       ...       ...     ...
11755  2024     12       6 -0.025927 -0.002948  0.0037
11756  2024     12       7 -0.028186 -0.069167  0.0037
11757  2024     12       8  0.013648  0.027621  0.0037
11758  2024     12       9 -0.050267 -0.048984  0.0037
11759  2024     12      10 -0.050475 -0.052058  0.0037

[11760 rows x 6 columns]
```

<u>Question 4: Replicate Table 1 for DM Portfolios</u>

This question replicates Table 1 statistics using decile portfolios formed via the DM method.

The PS3_Q4 function aggregates the DM decile return series to compute annualized return, volatility, Sharpe ratio, skewness, and correlation against a benchmark. It replicates the Daniel & Moskowitz Table 1 for DM deciles and WML.

Input – Dataframe "CRSP_Stocks_Momentum_returns", the output of PS3 Q3, Dataframe "DM_returns" (momentum portfolio returns from Daniel's website) where each row corresponds to a year-month-decile, with columns Year, Month, decile and DM_Ret.

Output – 5×11 numeric matrix,/dataframe reproducing part of Table 1 in Daniel & Moskowitz (2016).

- Rows: r − rf , σ, SR, sκ(m), correlations.
- Columns: Decile 1, Decile 2, Decile 3, Decile 4, Decile 5, Decile 6, Decile 7, Decile 8, Decile 9, Decile 10, WML

Step 1: Pivot the DM return table into wide format with date as index and deciles as columns. Compute WML (Winner Minus Loser) as the difference between decile 10 and decile 1.

Step 2: For each column (1–10 and WML), compute the following:

- Annualized excess return = mean monthly return × 12 × 100
- Annualized volatility = monthly std dev × √12 × 100
- Sharpe ratio = mean / std × √12
- Skewness = skewness of log(1 + return), ignoring returns ≤ −1

Step 3: Load the reference replication table, reshape into wide format, and compute WML from D10 – D1.

Step 4: Calculate Pearson correlation between your return series and the benchmark across overlapping dates.

Step 5: Return a 5×11 summary matrix including the four metrics and correlation for each decile and WML.

The resulting output is:

Notebook version:

	Metric	1	2	3	4	5	6	7	8	9	10	WML
0	Excess Return	2.59	5.49	6.50	6.99	8.14	7.21	8.81	9.78	10.12	13.68	11.09
1	Volatility	34.06	27.07	22.52	20.23	19.89	18.60	18.62	18.17	19.43	23.21	28.00
2	Sharpe Ratio	0.08	0.20	0.29	0.35	0.41	0.39	0.47	0.54	0.52	0.59	0.40
3	Skewness	-0.27	-0.36	-0.35	-0.51	0.22	-0.70	-0.25	-0.72	-0.67	-0.73	-3.47
4	Correlation	0.95	0.94	0.94	0.94	0.93	0.95	0.94	0.95	0.96	0.97	0.89

.py version:

```
-----------PS3 - Output 4------------
         Metric      1      2      3      4      5      6      7      8      9     10    WML
0  Excess Return   2.59   5.48   6.50   6.99   8.14   7.21   8.81   9.78  10.12  13.68  11.08
1     Volatility  34.06  27.07  22.52  20.23  19.89  18.60  18.62  18.17  19.43  23.21  28.00
2   Sharpe Ratio   0.08   0.20   0.29   0.35   0.41   0.39   0.47   0.54   0.52   0.59   0.40
3       Skewness  -0.27  -0.36  -0.35  -0.51   0.22  -0.70  -0.25  -0.72  -0.67  -0.73  -3.47
4    Correlation   0.95   0.94   0.94   0.94   0.93   0.95   0.94   0.95   0.96   0.97   0.89
```

<u>Question 5: Replicate Table 1 for KRF Portfolios</u>

This question performs the same calculations as Q4, but using the KRF decile returns.

The PS3_Q5 function replicates Table 1 statistics for the KRF deciles using Ken French's benchmark portfolios. It computes annualized metrics and the correlation of each decile and WML against the reference data.

Input – Dataframe "CRSP_Stocks_Momentum_returns", the output of PS3 Q3, Dataframe "KRF_returns" (momentum portfolio returns from French's website) where each row corresponds to a year-month-decile, with columns Year, Month, decile and KRF_Ret.

Output – 5 × 11 numeric matrix/dataframe, with the correlations between your estimated KRF momentum portfolio returns and the KRF momentum portfolio returns on French's website.

- Rows: r − rf , σ, SR, sκ(m), correlations.
- Columns: Decile 1, Decile 2, Decile 3, Decile 4, Decile 5, Decile 6, Decile 7, Decile 8, Decile 9, Decile 10, WML

Step 1: Subtract the risk-free rate from KRF_Ret to compute excess return. Pivot data to wide format and compute WML as decile 10 minus decile 1.

Step 2: For each column (1–10 and WML), compute the same four statistics as in Q4: annualized return, volatility, Sharpe ratio, and log skewness.

Step 3: Load the benchmark KRF decile return file from Ken French's website, reshape into wide format, and compute WML.

Step 4: Compute correlations between your return series and the benchmark series across all common dates.

Step 5: Return the final 5×11 matrix containing excess return, volatility, Sharpe, skewness, and correlation for all deciles and WML.

The resulting output is:

Notebook version:

	Metric	1	2	3	4	5	6	7	8	9	10	WML
0	Excess Return	1.59	3.15	4.64	4.30	4.35	4.10	5.75	6.35	6.27	9.35	7.76
1	Volatility	32.58	25.37	21.60	19.81	19.74	18.52	18.68	17.81	18.92	22.00	26.14
2	Sharpe Ratio	0.05	0.12	0.21	0.22	0.22	0.22	0.31	0.36	0.33	0.42	0.30
3	Skewness	-0.01	-0.16	-0.27	-0.41	0.26	-0.68	-0.20	-0.72	-0.71	-0.85	-4.18
4	Correlation	0.95	0.95	0.94	0.93	0.92	0.94	0.94	0.94	0.95	0.97	0.89

.py version:

```
-----------PS3 - Output 5------------
          Metric      1      2      3      4      5      6      7      8      9     10    WML
0  Excess Return   1.59   3.15   4.64   4.30   4.35   4.10   5.75   6.35   6.27   9.35   7.76
1     Volatility  32.58  25.37  21.60  19.81  19.74  18.52  18.68  17.81  18.92  22.00  26.14
2   Sharpe Ratio   0.05   0.12   0.21   0.22   0.22   0.22   0.31   0.36   0.33   0.42   0.30
3       Skewness  -0.01  -0.16  -0.27  -0.41   0.26  -0.68  -0.20  -0.72  -0.71  -0.85  -4.18
4    Correlation   0.95   0.95   0.94   0.93   0.92   0.94   0.94   0.94   0.95   0.97   0.89
```

All 5 outputs when run on terminal:

```
(base) PS C:\Users\johri_vuo0ktq\Desktop\UCLA MFE\Term 3\Quant AM\HW 3> python PS3.py
-----------PS3 - Output 1------------
         Year  Month  PERMNO  EXCHCD   lag_Mkt_Cap       Ret  Ranking_Ret
0        1991     12   10001     3.0  1.582675e+01 -0.006780     0.481465
1        1992      1   10001     3.0  1.558750e+01 -0.051724     0.461590
2        1992     10   10001     3.0  1.612500e+01 -0.025000     0.189838
3        1992     11   10001     3.0  1.572188e+01 -0.017094     0.038227
4        1992     12   10001     3.0  1.545312e+01 -0.015130     0.027788
...       ...    ...     ...     ...           ...       ...          ...
2687040  2024      8   93436     3.0  7.413801e+05 -0.077390    -0.106230
2687041  2024      9   93436     3.0  6.840044e+05  0.221942    -0.155850
2687042  2024     10   93436     3.0  8.390474e+05 -0.045025     0.264423
2687043  2024     11   93436     3.0  8.020335e+05  0.381469     0.039890
2687044  2024     12   93436     3.0  1.107984e+06  0.170008     0.328647

[2687045 rows x 7 columns]
-----------PS3 - Output 2------------
         Year  Month  PERMNO  lag_Mkt_Cap       Ret  DM_decile  KRF_decile  EXCHCD
0        1927      1   15245    15.750000 -0.047619          1         1.0     1.0
1        1927      1   10524    16.750000 -0.089552          1         1.0     1.0
2        1927      1   10196    19.095750 -0.018519          1         1.0     1.0
3        1927      1   11789    31.728125 -0.048951          1         1.0     1.0
4        1927      1   11578    21.145500 -0.101351          1         1.0     1.0
...       ...    ...     ...          ...       ...        ...         ...     ...
2687040  2024     12   20975  1104.525240 -0.119173         10         NaN     3.0
2687041  2024     12   19142   118.913340 -0.023524         10         NaN     3.0
2687042  2024     12   21090   918.354250  0.091759         10         NaN     3.0
2687043  2024     12   24100  2377.536210 -0.396741         10         NaN     3.0
2687044  2024     12   19956  2153.451690 -0.019518         10         NaN     3.0

[2687045 rows x 8 columns]
```

```
-----------PS3 - Output 3------------
       Year  Month  decile    DM_Ret   KRF_Ret      Rf
0      1927      1       1 -0.022619 -0.022619  0.0025
1      1927      1       2 -0.008067 -0.008067  0.0025
2      1927      1       3 -0.004554 -0.004554  0.0025
3      1927      1       4 -0.007277 -0.007277  0.0025
4      1927      1       5  0.003399  0.003399  0.0025
...     ...    ...     ...       ...       ...     ...
11755  2024     12       6 -0.025927 -0.002948  0.0037
11756  2024     12       7 -0.028186 -0.069167  0.0037
11757  2024     12       8  0.013648  0.027621  0.0037
11758  2024     12       9 -0.050267 -0.048984  0.0037
11759  2024     12      10 -0.050475 -0.052058  0.0037

[11760 rows x 6 columns]
C:\Users\johri_vuo0ktq\Desktop\UCLA MFE\Term 3\Quant AM\HW 3\PS3.py:255: FutureWarning: The 'delim_whitespace' keyword in pd.read_csv is deprecated and will
 be removed in a future version. Use ``sep='\s+'`` instead
  dm_ref_df = pd.read_csv("m_m_pt_tot.txt", delim_whitespace=True, header=None)
-----------PS3 - Output 4------------
          Metric      1      2      3      4      5      6      7      8      9     10    WML
0  Excess Return   2.59   5.48   6.50   6.99   8.14   7.21   8.81   9.78  10.12  13.68  11.08
1     Volatility  34.06  27.07  22.52  20.23  19.89  18.60  18.62  18.17  19.43  23.21  28.00
2   Sharpe Ratio   0.08   0.20   0.29   0.35   0.41   0.39   0.47   0.54   0.52   0.59   0.40
3       Skewness  -0.27  -0.36  -0.35  -0.51   0.22  -0.70  -0.25  -0.72  -0.67  -0.73  -3.47
4    Correlation   0.95   0.94   0.94   0.94   0.93   0.95   0.94   0.95   0.96   0.97   0.89
-----------PS3 - Output 5------------
          Metric      1      2      3      4      5      6      7      8      9     10    WML
0  Excess Return   1.59   3.15   4.64   4.30   4.35   4.10   5.75   6.35   6.27   9.35   7.76
1     Volatility  32.58  25.37  21.60  19.81  19.74  18.52  18.68  17.81  18.92  22.00  26.14
2   Sharpe Ratio   0.05   0.12   0.21   0.22   0.22   0.22   0.31   0.36   0.33   0.42   0.30
3       Skewness  -0.01  -0.16  -0.27  -0.41   0.26  -0.68  -0.20  -0.72  -0.71  -0.85  -4.18
4    Correlation   0.95   0.95   0.94   0.93   0.92   0.94   0.94   0.94   0.95   0.97   0.89
(base) PS C:\Users\johri_vuo0ktq\Desktop\UCLA MFE\Term 3\Quant AM\HW 3>
```

<u>Question 6: Has the momentum anomaly worked in the past few years?</u>

Yes, the momentum anomaly has continued to work, though with somewhat lower performance in our replication. Importantly, both our sample and the professor’s benchmark use the same date range (1927–2024), so differences in performance cannot be attributed to a different or more recent sample.

Instead, the differences in WML returns (11.08% vs. 17.65% for DM, and 7.76% vs. 13.64% for KRF) and Sharpe ratios (0.40 vs. 0.58 for DM) likely stem from technical differences in implementation — such as sorting methods, treatment of small-cap or delisted stocks, or weighting mechanisms. Nevertheless, our replication still shows strong monotonic increases in returns from decile 1 to 10, consistent with the presence of a momentum premium. This confirms that the momentum effect has persisted over the long run, albeit potentially with slightly weaker expression under our method.

Question 7: Would you implement this trading strategy if you were running your own fund?

Yes, I would consider implementing a momentum strategy, but only with proper risk controls in place. The strategy has demonstrated long-term profitability, both in my replication and in the professor's benchmark results, using the same sample period (1927–2024). While my replicated WML returns are somewhat lower, the monotonic increase in decile returns and positive Sharpe ratios still support the viability of momentum investing.

That said, momentum strategies face several implementation challenges:

- Crash Risk: As highlighted by Daniel and Moskowitz (2016), momentum strategies are prone to sudden and severe drawdowns, especially following periods of strong performance. Managing exposure during these crash-prone regimes is critical.
- High Turnover: Monthly rebalancing causes frequent trading, which can erode returns through transaction costs and market impact—particularly in less liquid names.
- Crowding: Momentum is widely followed and implemented, increasing the risk of strategy crowding and correlated liquidations.
- Data Quality: Proper treatment of delistings and survivorship bias (e.g., via DLRET) is essential for performance replication and live trading.

To implement this in practice, I would integrate momentum into a broader multi-factor framework (e.g., alongside value, quality, and low risk), apply turnover constraints, and incorporate signals like volatility scaling or crash indicators to adapt position sizing. This would make the strategy more robust to market regimes and execution realities.