asset-class-return-correlations-part-2

November 1, 2026

0.1 Post Updates

Update 11/1/2026: Moved the total correlation metric, decile analysis, and trading strategy here from the original post, which is now part 1. Also added the 2x leveraged funds SSO, MVV, and SAA to the trading strategy, using the same signal as the original IVV, IJH, and IJR strategies.

0.2 Introduction

In part 1, we looked at historical return correlations between asset classes, and saw that the rolling correlations can abruptly increase or decrease in response to market shocks. In this post, we'll build a "total correlation" metric from those rolling correlations, look at future returns based on that metric, and develop a trading strategy that uses correlation as a market timing signal.

Python Imports

[1]:
```
# Standard Library
import os
import sys
import warnings

from IPython.display import display, Markdown
from itertools import combinations, count
from pathlib import Path

# Data Handling
import numpy as np
import pandas as pd

# Suppress warnings
warnings.filterwarnings("ignore")

# Pandas display options
pd.set_option("display.float_format", lambda x: f"%.2f" % x)
```

[2]:
```
# Add the source subdirectory to the system path to allow import config from␣
↪settings.py
current_directory = Path(os.getcwd())
BASE_DIR = current_directory.parent.parent.parent
src_directory = BASE_DIR / "src"
```

```
sys.path.append(str(src_directory)) if str(src_directory) not in sys.path else␣
↪None

# Import settings.py
from settings import config

# Add other configured directories
CONTENT_DIR = config("CONTENT_DIR")
POSTS_DIR = config("POSTS_DIR")
PAGES_DIR = config("PAGES_DIR")
PUBLIC_DIR = config("PUBLIC_DIR")
SOURCE_DIR = config("SOURCE_DIR")
DATA_DIR = config("DATA_DIR")
DATA_MANUAL_DIR = config("DATA_MANUAL_DIR")
```

0.3 Python Functions

Here are the functions needed for this project:

- load_data: Load data from a CSV, Excel, or Pickle file into a pandas DataFrame.
- plot_histogram: Plot the histogram of a data set from a DataFrame.
- plot_scatter: Plot the data from a DataFrame for a specified date range and columns.
- plot_time_series: Plot the timeseries data from a DataFrame for a specified date range and columns.
- summary_stats: Generate summary statistics for a series of returns.
- yf_pull_data: Download daily price data from Yahoo Finance and export it.

[3]:
```
from load_data import load_data
from plot_histogram import plot_histogram
from plot_scatter import plot_scatter
from plot_time_series import plot_time_series
from summary_stats import summary_stats
from yf_pull_data import yf_pull_data
```

0.4 Data Overview

The signal in this post comes from the rolling correlations between the following funds:

- Large Cap US Stocks / S&P 500 – IVV (iShares S&P 500 ETF)
- Mid Cap US Stocks / S&P MidCap 400 – IJH (iShares S&P MidCap 400 ETF)
- Small Cap US Stocks / S&P SmallCap 600 – IJR (iShares S&P SmallCap 600 ETF)

We'll also use the following 2x leveraged ETFs, but only in the trading strategy. They do not feed into the signal:

- 2x Large Cap US Stocks / S&P 500 – SSO (ProShares Ultra S&P500)
- 2x Mid Cap US Stocks / S&P MidCap 400 – MVV (ProShares Ultra MidCap400)
- 2x Small Cap US Stocks / S&P SmallCap 600 – SAA (ProShares Ultra SmallCap600)

We'll load the rest of the ETFs from part 1 as well, in order to keep the same sample of days.

That sample only includes the days where all of the ETFs have data, so it starts with the inception of GSG in July 2006. Using only IVV, IJH, and IJR would extend the sample back to 2000, and change the deciles and the resulting trades.

For all of these, we will use the adjusted closing price, which accounts for dividends and stock splits.

0.4.1 Acquire & Load Data

[4]:
```
figure_number = count(1)
table_number = count(1)

# Create list of tickers to pull data for
us_equity_tickers = ["IVV", "IJH", "IJR", "QQQ", "IWB", "IWD", "IWF", "IWM"]
intl_equity_tickers = ["EFA", "EEM", "IEV"]
equity_tickers = us_equity_tickers + intl_equity_tickers
bond_tickers = ["SHY", "IEF", "TLT", "AGG"]
commodity_tickers = ["GLD", "GSG"]
real_estate_tickers = ["IYR"]

# 2x leveraged funds, used only in the trading strategy. These are kept out of␣
 ↪etf_tickers
# so that SAA's later inception date doesn't change the sample used for the␣
 ↪correlations.
leveraged_tickers = ["SSO", "MVV", "SAA"]

etf_tickers = (
    us_equity_tickers
    + intl_equity_tickers
    + bond_tickers
    + commodity_tickers
    + real_estate_tickers
)
```

[5]:
```
for ticker in etf_tickers + leveraged_tickers:
    yf_pull_data(
        base_directory=DATA_DIR,
        ticker=ticker,
        adjusted=False,
        source="Yahoo_Finance",
        asset_class="Exchange_Traded_Funds",
        excel_export=True,
        pickle_export=True,
        output_confirmation=False,
    )
```

[6]:
```
fund_data = pd.DataFrame()

for ticker in etf_tickers + leveraged_tickers:
```

```
    data = load_data(
        base_directory=DATA_DIR,
        ticker=ticker,
        source="Yahoo_Finance",
        asset_class="Exchange_Traded_Funds",
        timeframe="Daily",
        file_format="pickle",
    )

    # Rename columns to "QQQ_Close", etc.
    data = data.rename(
        columns={
            "Adj Close": f"{ticker}_Adj_Close",
            "Close": f"{ticker}_Close",
            "High": f"{ticker}_High",
            "Low": f"{ticker}_Low",
            "Open": f"{ticker}_Open",
            "Volume": f"{ticker}_Volume",
        }
    )

    # Drop all columns except for the adjusted close price and date index
    data = data[[f"{ticker}_Adj_Close"]]

    # Calculate daily returns and add as new column
    data[f"{ticker}_Daily_Return"] = data[f"{ticker}_Adj_Close"].pct_change()

    # Concatenate the data for this ticker with the main fund_data DataFrame
    fund_data = pd.concat([fund_data, data], axis=1)

display(Markdown(f"Table {next(table_number)}: Fund Data"))
display(fund_data)
```

Table 1: Fund Data

Date	IVV_Adj_Close	IVV_Daily_Return	IJH_Adj_Close	IJH_Daily_Return
1999-03-10	NaN	NaN	NaN	NaN
1999-03-11	NaN	NaN	NaN	NaN
1999-03-12	NaN	NaN	NaN	NaN
1999-03-15	NaN	NaN	NaN	NaN
1999-03-16	NaN	NaN	NaN	NaN
…	…	…	…	…
2026-10-23	763.25	-0.01	72.37	-0.01
2026-10-26	768.69	0.01	72.91	0.01
2026-10-27	773.25	0.01	73.22	0.00
2026-10-28	775.65	0.00	73.35	0.00
2026-10-29	772.97	-0.00	74.20	0.01

Date	IJR_Adj_Close	IJR_Daily_Return	QQQ_Adj_Close	QQQ_Daily_Return	…
1999-03-10	NaN	NaN	42.98	NaN	…
1999-03-11	NaN	NaN	43.19	0.00	…
1999-03-12	NaN	NaN	42.14	-0.02	…
1999-03-15	NaN	NaN	43.35	0.03	…
1999-03-16	NaN	NaN	43.72	0.01	…
…	…	…	…	…	…
2026-10-23	138.22	-0.00	704.40	-0.00	…
2026-10-26	138.93	0.01	712.20	0.01	…
2026-10-27	139.68	0.01	721.25	0.01	…
2026-10-28	139.76	0.00	725.61	0.01	…
2026-10-29	142.01	0.02	715.34	-0.01	…

Date	IWB_Adj_Close	IWB_Daily_Return	…	GSG_Adj_Close	…
1999-03-10	NaN	NaN	…	NaN	…
1999-03-11	NaN	NaN	…	NaN	…
1999-03-12	NaN	NaN	…	NaN	…
1999-03-15	NaN	NaN	…	NaN	…
1999-03-16	NaN	NaN	…	NaN	…
…	…	…	…	…	…
2026-10-23	415.35	-0.00	…	37.70	…
2026-10-26	418.06	0.01	…	37.31	…
2026-10-27	420.41	0.01	…	36.99	…
2026-10-28	421.72	0.00	…	37.15	…
2026-10-29	420.52	-0.00	…	37.11	…

Date	GSG_Daily_Return	IYR_Adj_Close	IYR_Daily_Return	SSO_Adj_Close	…
1999-03-10	NaN	NaN	NaN	NaN	…
1999-03-11	NaN	NaN	NaN	NaN	…
1999-03-12	NaN	NaN	NaN	NaN	…
1999-03-15	NaN	NaN	NaN	NaN	…
1999-03-16	NaN	NaN	NaN	NaN	…
…	…	…	…	…	…
2026-10-23	0.02	98.40	0.00	68.52	…
2026-10-26	-0.01	98.83	0.00	69.48	…
2026-10-27	-0.01	97.97	-0.01	70.28	…
2026-10-28	0.00	97.29	-0.01	70.77	…
2026-10-29	-0.00	99.47	0.02	70.23	…

Date	SSO_Daily_Return	MVV_Adj_Close	MVV_Daily_Return	SAA_Adj_Close	…
1999-03-10	NaN	NaN	NaN	NaN	…
1999-03-11	NaN	NaN	NaN	NaN	…
1999-03-12	NaN	NaN	NaN	NaN	…

```
1999-03-15                NaN            NaN                NaN            NaN
1999-03-16                NaN            NaN                NaN            NaN
…                         …              …                  …              …
2026-10-23              -0.01          79.72              -0.01          32.37
2026-10-26               0.01          80.89               0.01          32.66
2026-10-27               0.01          81.51               0.01          33.05
2026-10-28               0.01          81.77               0.00          33.30
2026-10-29              -0.01          83.65               0.02          34.10

            SAA_Daily_Return
Date
1999-03-10               NaN
1999-03-11               NaN
1999-03-12               NaN
1999-03-15               NaN
1999-03-16               NaN
…                        …
2026-10-23             -0.01
2026-10-26              0.01
2026-10-27              0.01
2026-10-28              0.01
2026-10-29              0.02

[6954 rows x 42 columns]
```

[7]:
```
for ticker in etf_tickers + leveraged_tickers:
    print(f"First date for {ticker}: {fund_data[f'{ticker}_Daily_Return'].
 ↪first_valid_index()}")
```

```
First date for IVV: 2000-05-22 00:00:00
First date for IJH: 2000-05-30 00:00:00
First date for IJR: 2000-05-30 00:00:00
First date for QQQ: 1999-03-11 00:00:00
First date for IWB: 2000-05-22 00:00:00
First date for IWD: 2000-05-30 00:00:00
First date for IWF: 2000-05-30 00:00:00
First date for IWM: 2000-05-30 00:00:00
First date for EFA: 2001-08-28 00:00:00
First date for EEM: 2003-04-15 00:00:00
First date for IEV: 2000-07-31 00:00:00
First date for SHY: 2002-07-31 00:00:00
First date for IEF: 2002-07-31 00:00:00
First date for TLT: 2002-07-31 00:00:00
First date for AGG: 2003-09-30 00:00:00
First date for GLD: 2004-11-19 00:00:00
First date for GSG: 2006-07-24 00:00:00
First date for IYR: 2000-06-20 00:00:00
First date for SSO: 2006-06-22 00:00:00
```

```
First date for MVV: 2006-06-22 00:00:00
First date for SAA: 2007-01-26 00:00:00
```

[8]:
```python
# Keep only the daily return columns for the ETFs used in the correlations
daily_return_columns = [f"{ticker}_Daily_Return" for ticker in etf_tickers]
fund_data_daily_returns_all = fund_data[daily_return_columns]

# Drop the rows with missing data, which sets the start of the sample
fund_data_daily_returns = fund_data_daily_returns_all.dropna()

print(f"Shape of fund_data_daily_returns_all: {fund_data_daily_returns_all.
  ↪shape}")
print(f"Shape of fund_data_daily_returns: {fund_data_daily_returns.shape}")
print(f"First date in sample: {fund_data_daily_returns.index[0]}")
```

```
Shape of fund_data_daily_returns_all: (6954, 18)
Shape of fund_data_daily_returns: (5100, 18)
First date in sample: 2006-07-24 00:00:00
```

0.5 Calculate Rolling Correlations

Next, we will calculate the rolling correlations between IVV, IJH, and IJR for the same periods as in part 1.

[9]:
```python
# Define rolling windows in trading days
rolling_windows = {
    "3d": 3,  # 3 days (~3 trading days)
    "1w": 5,  # 1 week (~5 trading days)
    "2w": 10,  # 2 weeks (~10 trading days)
    "1m": 21,  # 1 month (~21 trading days)
    "3m": 63,  # 3 months (~63 trading days)
    "6m": 126,  # 6 months (~126 trading days)
    "1y": 252,  # 1 year (~252 trading days)
    "5y": 1260,  # 5 years (~1260 trading days)
    "10y": 2520,  # 10 years (~2520 trading days)
}
```

[10]:
```python
# Create temp list for tickers
tickers = ["IVV", "IJH", "IJR"]
pairs = list(combinations(tickers, 2))

# Create empty dictionary to store rolling correlation results
rolling_corr_results = {}

# Create empty DataFrame to store rolling correlation results
rolling_corr_results_df = pd.DataFrame()
```

```
# Calculate rolling correlations for each pair of tickers and each rolling␣
 ↪window
for ticker1, ticker2 in pairs:
    try:
        temp_df = fund_data_daily_returns[
            [f"{ticker1}_Daily_Return", f"{ticker2}_Daily_Return"]
        ].dropna()
    except Exception as e:
        print(f"Error creating temp_df for {ticker1} and {ticker2}: {e}")
        continue

    for window_name, window_size in rolling_windows.items():
        try:
            rolling_corr = (
                temp_df[f"{ticker1}_Daily_Return"]
                .rolling(window=window_size)
                .corr(temp_df[f"{ticker2}_Daily_Return"])
            )

            rolling_corr_results[
                f"{ticker1}_{ticker2}_{window_name}"
            ] = rolling_corr
            rolling_corr_results_df = pd.concat(
                [
                    rolling_corr_results_df,
                    rolling_corr.
 ↪to_frame(name=f"{ticker1}_{ticker2}_{window_name}"),
                ],
                axis=1,
            )

        except Exception as e:
            print(
                f"Error calculating rolling correlation for {ticker1} and␣
 ↪{ticker2} with window {window_name}: {e}"
            )

display(Markdown(f"Table {next(table_number)}: Rolling Correlations"))
display(rolling_corr_results_df)
```

Table 2: Rolling Correlations

Date	IVV_IJH_3d	IVV_IJH_1w	IVV_IJH_2w	IVV_IJH_1m	IVV_IJH_3m
2006-07-24	NaN	NaN	NaN	NaN	NaN
2006-07-25	NaN	NaN	NaN	NaN	NaN
2006-07-26	0.97	NaN	NaN	NaN	NaN
2006-07-27	0.99	NaN	NaN	NaN	NaN

```
2006-07-28        1.00        0.96         NaN         NaN         NaN
…                 …           …            …           …           …
2026-10-23        0.98        0.83        0.86        0.76        0.80
2026-10-26        0.98        0.91        0.89        0.78        0.80
2026-10-27        0.99        0.98        0.89        0.76        0.78
2026-10-28        0.95        0.98        0.85        0.76        0.79
2026-10-29       -0.86        0.32        0.71        0.68        0.76

            IVV_IJH_6m  IVV_IJH_1y  IVV_IJH_5y  IVV_IJH_10y  IVV_IJR_3d  …  \
Date                                                                       …
2006-07-24         NaN         NaN         NaN          NaN         NaN  …
2006-07-25         NaN         NaN         NaN          NaN         NaN  …
2006-07-26         NaN         NaN         NaN          NaN        0.96  …
2006-07-27         NaN         NaN         NaN          NaN        0.93  …
2006-07-28         NaN         NaN         NaN          NaN        0.99  …
…                  …           …           …            …           …    …
2026-10-23        0.78        0.80        0.88         0.90       -0.45  …
2026-10-26        0.78        0.80        0.88         0.90        0.07  …
2026-10-27        0.78        0.80        0.88         0.90        0.99  …
2026-10-28        0.78        0.80        0.88         0.90        0.94  …
2026-10-29        0.77        0.80        0.88         0.90       -0.82  …

            IVV_IJR_10y  IJH_IJR_3d  IJH_IJR_1w  IJH_IJR_2w  IJH_IJR_1m  \
Date
2006-07-24          NaN         NaN         NaN         NaN         NaN
2006-07-25          NaN         NaN         NaN         NaN         NaN
2006-07-26          NaN        1.00         NaN         NaN         NaN
2006-07-27          NaN        0.97         NaN         NaN         NaN
2006-07-28          NaN        0.98        0.99         NaN         NaN
…                   …          …           …            …           …
2026-10-23         0.83       -0.25        0.53        0.84        0.86
2026-10-26         0.83        0.28        0.54        0.83        0.86
2026-10-27         0.83        0.97        0.25        0.84        0.86
2026-10-28         0.83        0.80        0.29        0.75        0.86
2026-10-29         0.83        1.00        0.92        0.77        0.86

            IJH_IJR_3m  IJH_IJR_6m  IJH_IJR_1y  IJH_IJR_5y  IJH_IJR_10y
Date
2006-07-24         NaN         NaN         NaN         NaN          NaN
2006-07-25         NaN         NaN         NaN         NaN          NaN
2006-07-26         NaN         NaN         NaN         NaN          NaN
2006-07-27         NaN         NaN         NaN         NaN          NaN
2006-07-28         NaN         NaN         NaN         NaN          NaN
…                  …           …           …           …            …
2026-10-23        0.90        0.90        0.93        0.97         0.96
2026-10-26        0.91        0.90        0.93        0.97         0.96
2026-10-27        0.90        0.90        0.93        0.97         0.96
2026-10-28        0.90        0.90        0.93        0.97         0.96
```

```
2026-10-29        0.91        0.90        0.93        0.97        0.96

[5100 rows x 27 columns]
```

0.6 Total Correlation Metric

As we saw in part 1, there are periods of time when the correlations abruptly increase or decrease as a response to some kind of market shock or event.

As an attempt to find some kind of signal, let's simply add the correlations together to get a "total correlation" metric, and look at the distribution. Note that we normalize the correlations to account for the number of correlations being summed by dividing the total by the number of correlations, to keep the values between -1 and 1.

[11]:
```
# Calculate total rolling correlation for each window
for window, days in rolling_windows.items():
    rolling_corr_results_df[f"total_rolling_correlation_{window}"] =␣
 ↪rolling_corr_results_df[[col for col in rolling_corr_results_df.columns if␣
 ↪col.endswith(f"_{window}")]].sum(axis=1) / len([col for col in␣
 ↪rolling_corr_results_df.columns if col.endswith(f"_{window}")])

# Reset any 0.00 values to NaN
rolling_corr_results_df.replace(0.00, np.nan, inplace=True)
```

[11]:

Date	IVV_IJH_3d	IVV_IJH_1w	IVV_IJH_2w	IVV_IJH_1m	IVV_IJH_3m	\
2006-07-24	NaN	NaN	NaN	NaN	NaN	
2006-07-25	NaN	NaN	NaN	NaN	NaN	
2006-07-26	0.97	NaN	NaN	NaN	NaN	
2006-07-27	0.99	NaN	NaN	NaN	NaN	
2006-07-28	1.00	0.96	NaN	NaN	NaN	
…	…	…	…	…	…	
2026-10-23	0.98	0.83	0.86	0.76	0.80	
2026-10-26	0.98	0.91	0.89	0.78	0.80	
2026-10-27	0.99	0.98	0.89	0.76	0.78	
2026-10-28	0.95	0.98	0.85	0.76	0.79	
2026-10-29	-0.86	0.32	0.71	0.68	0.76	

Date	IVV_IJH_6m	IVV_IJH_1y	IVV_IJH_5y	IVV_IJH_10y	IVV_IJR_3d	…	\
						…	
2006-07-24	NaN	NaN	NaN	NaN	NaN	…	
2006-07-25	NaN	NaN	NaN	NaN	NaN	…	
2006-07-26	NaN	NaN	NaN	NaN	0.96	…	
2006-07-27	NaN	NaN	NaN	NaN	0.93	…	
2006-07-28	NaN	NaN	NaN	NaN	0.99	…	
…	…	…	…	…	…	…	
2026-10-23	0.78	0.80	0.88	0.90	-0.45	…	
2026-10-26	0.78	0.80	0.88	0.90	0.07	…	

```
2026-10-27       0.78       0.80       0.88       0.90       0.99  …
2026-10-28       0.78       0.80       0.88       0.90       0.94  …
2026-10-29       0.77       0.80       0.88       0.90      -0.82  …

            IJH_IJR_10y  total_rolling_correlation_3d  \
Date
2006-07-24          NaN                          NaN
2006-07-25          NaN                          NaN
2006-07-26          NaN                         0.98
2006-07-27          NaN                         0.97
2006-07-28          NaN                         0.99
…                   …                            …
2026-10-23         0.96                         0.09
2026-10-26         0.96                         0.44
2026-10-27         0.96                         0.98
2026-10-28         0.96                         0.90
2026-10-29         0.96                        -0.23

            total_rolling_correlation_1w  total_rolling_correlation_2w  \
Date
2006-07-24                           NaN                           NaN
2006-07-25                           NaN                           NaN
2006-07-26                           NaN                           NaN
2006-07-27                           NaN                           NaN
2006-07-28                          0.97                           NaN
…                                      …                             …
2026-10-23                          0.45                          0.75
2026-10-26                          0.54                          0.76
2026-10-27                          0.44                          0.77
2026-10-28                          0.46                          0.67
2026-10-29                          0.41                          0.57

            total_rolling_correlation_1m  total_rolling_correlation_3m  \
Date
2006-07-24                           NaN                           NaN
2006-07-25                           NaN                           NaN
2006-07-26                           NaN                           NaN
2006-07-27                           NaN                           NaN
2006-07-28                           NaN                           NaN
…                                      …                             …
2026-10-23                          0.73                          0.80
2026-10-26                          0.73                          0.81
2026-10-27                          0.72                          0.80
2026-10-28                          0.72                          0.80
2026-10-29                          0.66                          0.77

            total_rolling_correlation_6m  total_rolling_correlation_1y  \
```

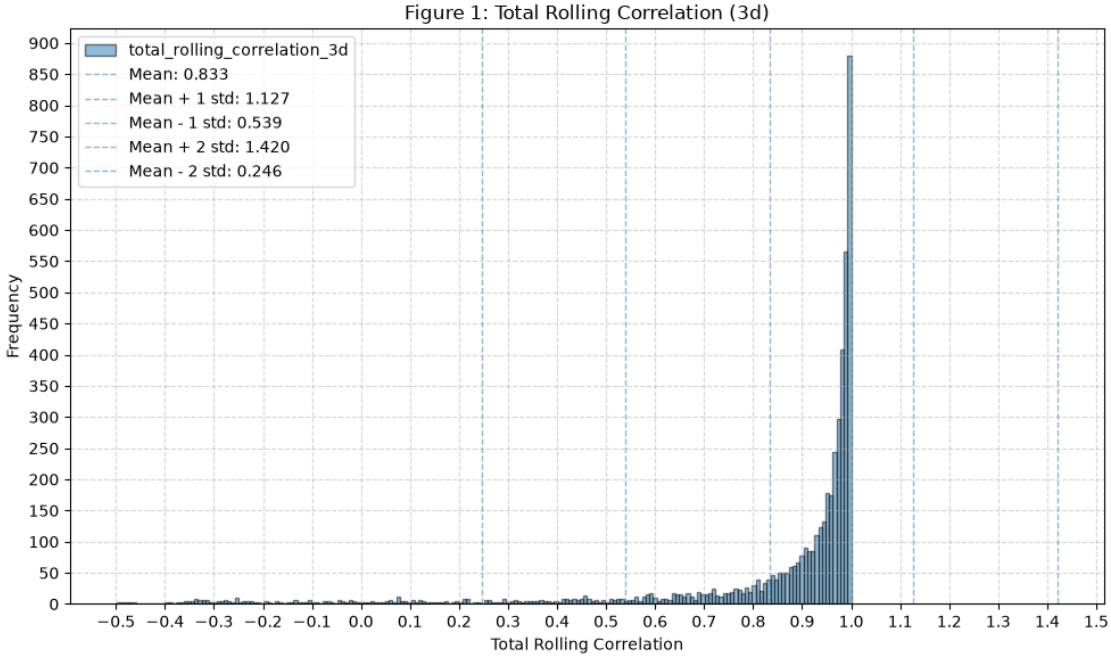
```
Date
2006-07-24                              NaN                              NaN
2006-07-25                              NaN                              NaN
2006-07-26                              NaN                              NaN
2006-07-27                              NaN                              NaN
2006-07-28                              NaN                              NaN
…                                       …                                …
2026-10-23                             0.79                             0.83
2026-10-26                             0.79                             0.83
2026-10-27                             0.79                             0.82
2026-10-28                             0.79                             0.83
2026-10-29                             0.77                             0.82

            total_rolling_correlation_5y  total_rolling_correlation_10y
Date
2006-07-24                              NaN                              NaN
2006-07-25                              NaN                              NaN
2006-07-26                              NaN                              NaN
2006-07-27                              NaN                              NaN
2006-07-28                              NaN                              NaN
…                                       …                                …
2026-10-23                             0.89                             0.90
2026-10-26                             0.89                             0.90
2026-10-27                             0.89                             0.90
2026-10-28                             0.89                             0.90
2026-10-29                             0.89                             0.90

[5100 rows x 36 columns]
```

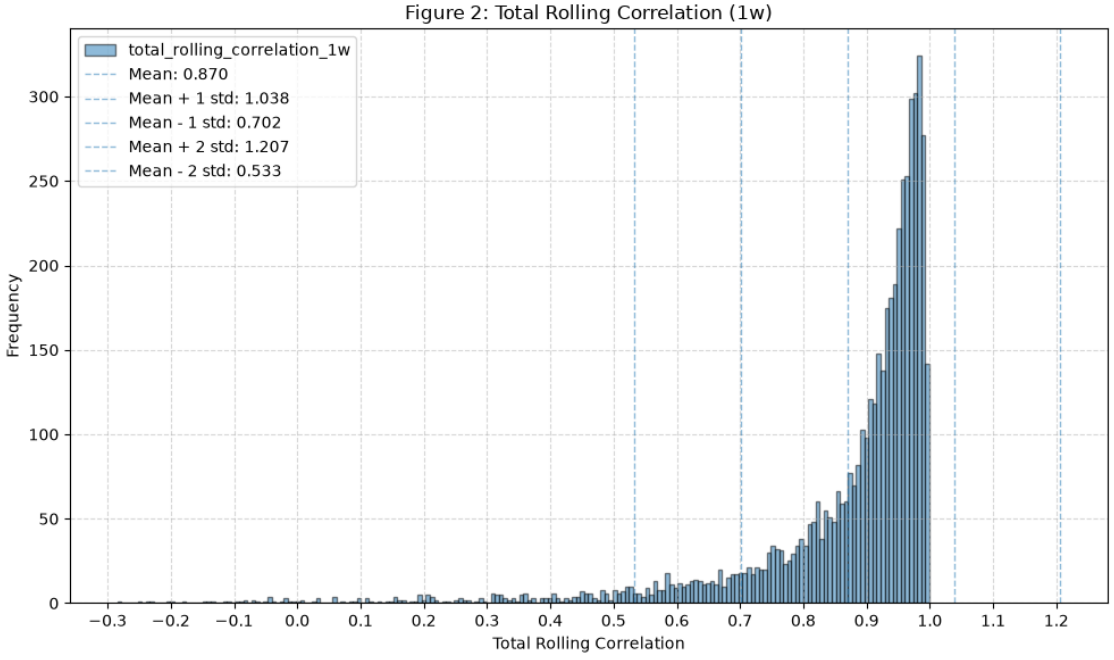
[12]:
```python
for window in rolling_windows.keys():
    plot_histogram(
        df=rolling_corr_results_df[[f"total_rolling_correlation_{window}"]].
 ↪dropna(),
        plot_columns=[f"total_rolling_correlation_{window}"],
        title=f"Figure {next(figure_number)}: Total Rolling Correlation␣
 ↪({window})",
        x_label="Total Rolling Correlation",
        x_tick_spacing="Auto",
        x_tick_rotation=0,
        y_label="Frequency",
        y_tick_spacing="Auto",
        y_tick_rotation=0,
        caption=f"Data Source: Yahoo Finance, Analysis by the Author",
    )
```

Figure 1: Total Rolling Correlation (3d)



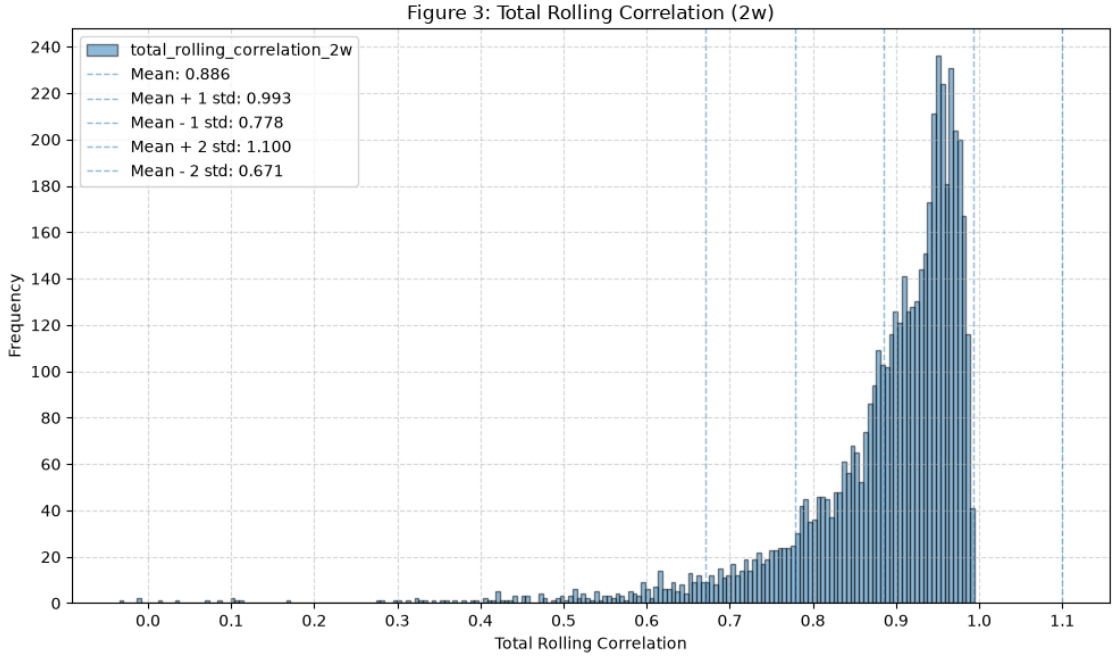
Data Source: Yahoo Finance, Analysis by the Author

Figure 2: Total Rolling Correlation (1w)



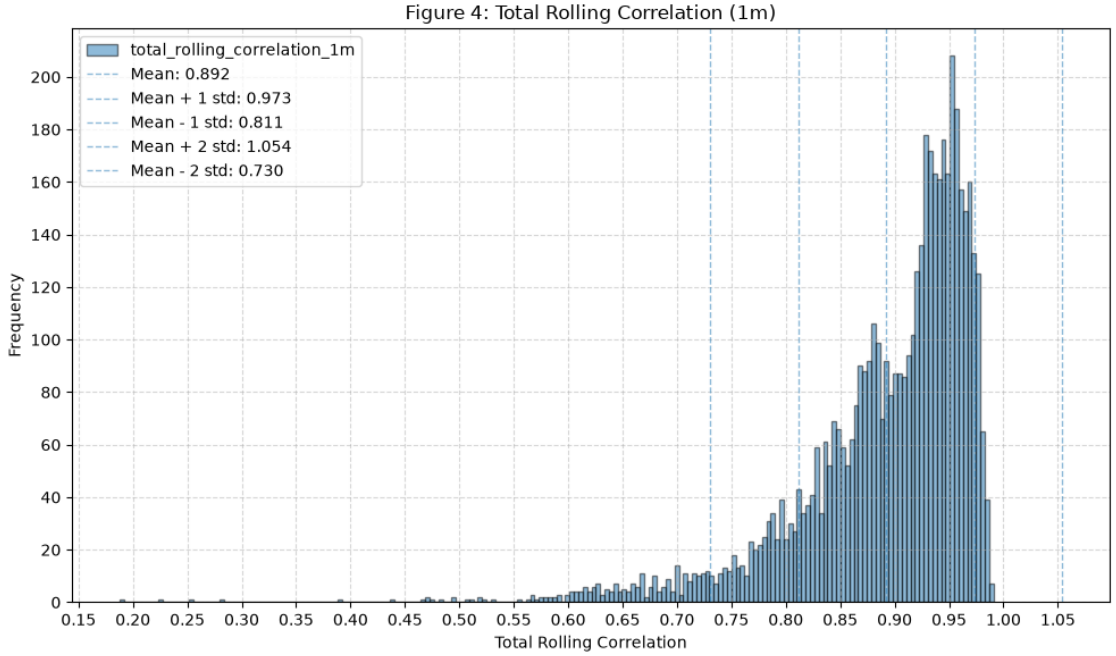
Data Source: Yahoo Finance, Analysis by the Author

Figure 3: Total Rolling Correlation (2w)



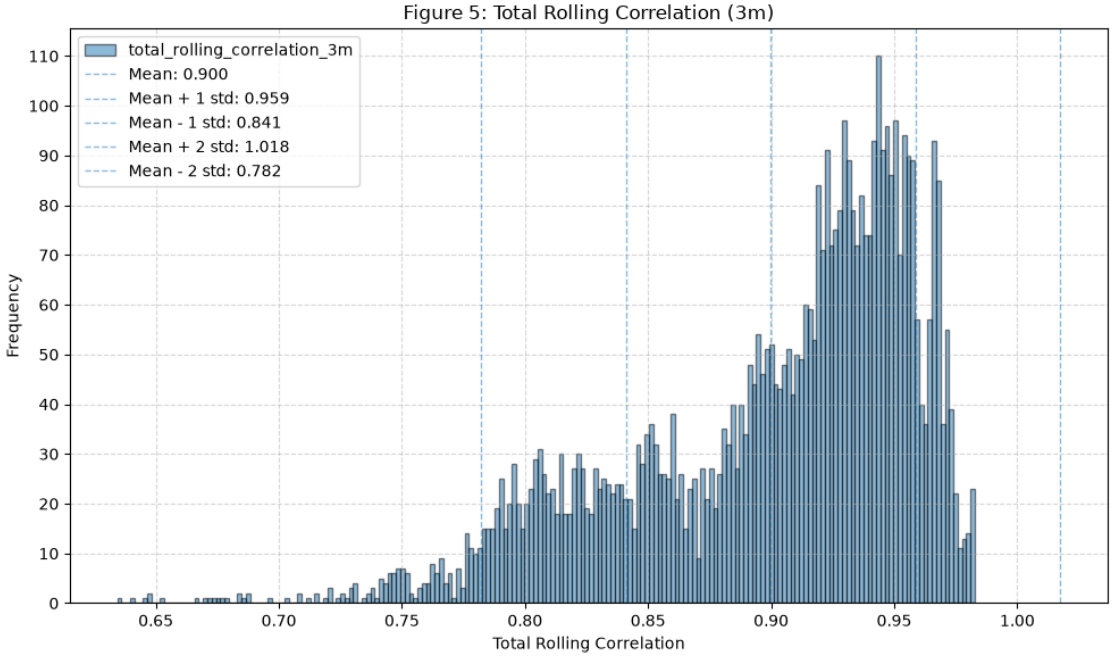
Data Source: Yahoo Finance, Analysis by the Author

Figure 4: Total Rolling Correlation (1m)



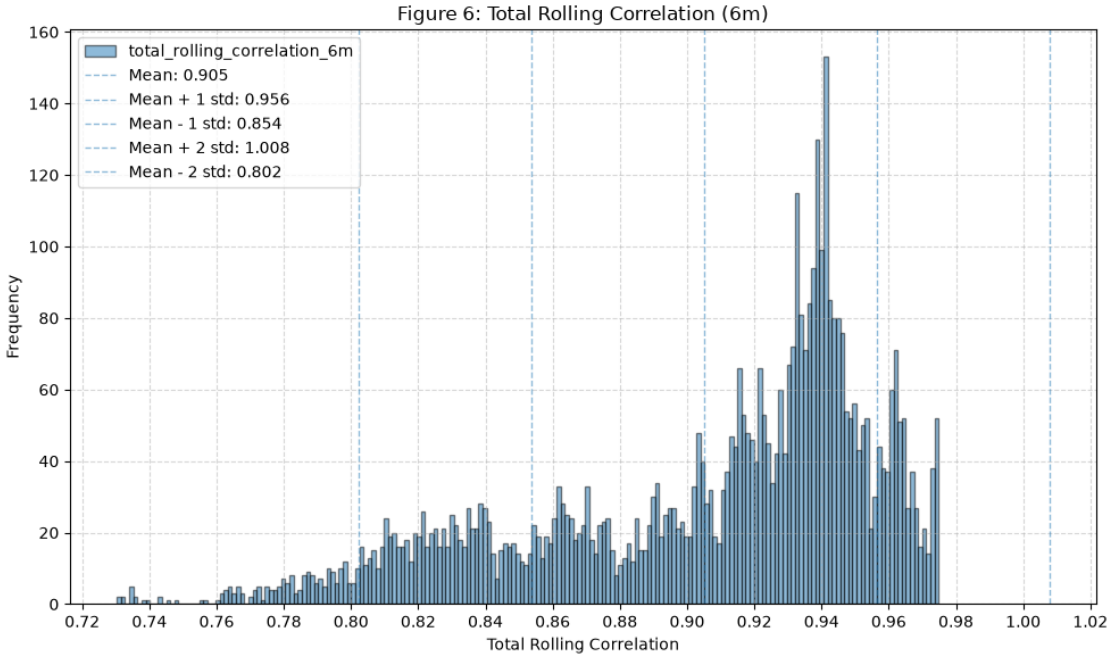
Data Source: Yahoo Finance, Analysis by the Author

Figure 5: Total Rolling Correlation (3m)



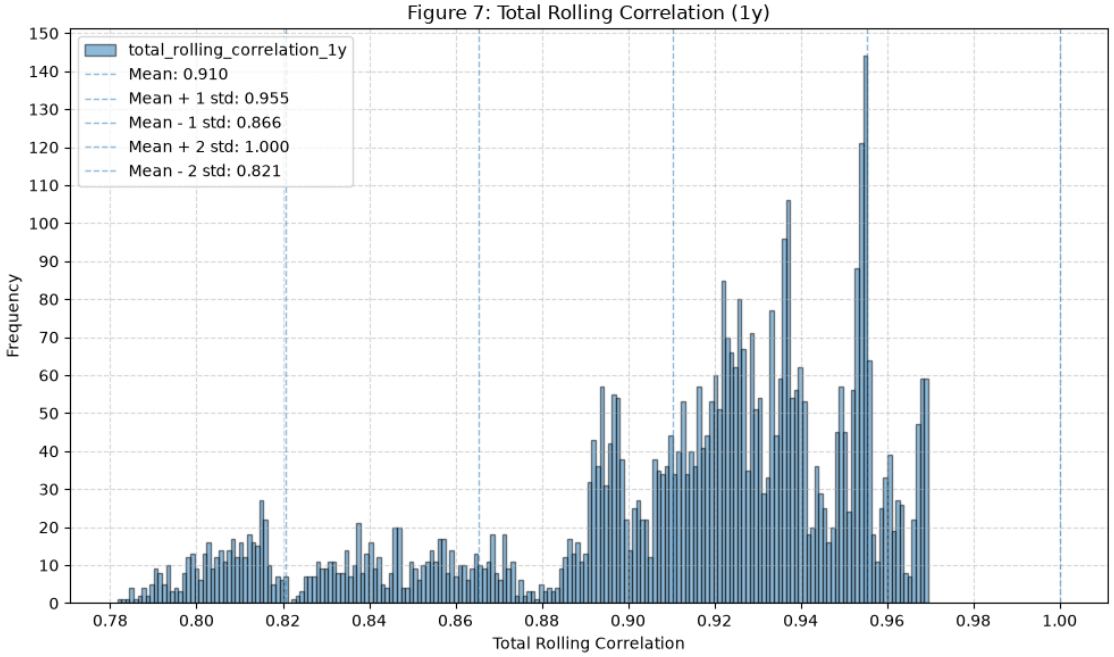
Data Source: Yahoo Finance, Analysis by the Author

Figure 6: Total Rolling Correlation (6m)



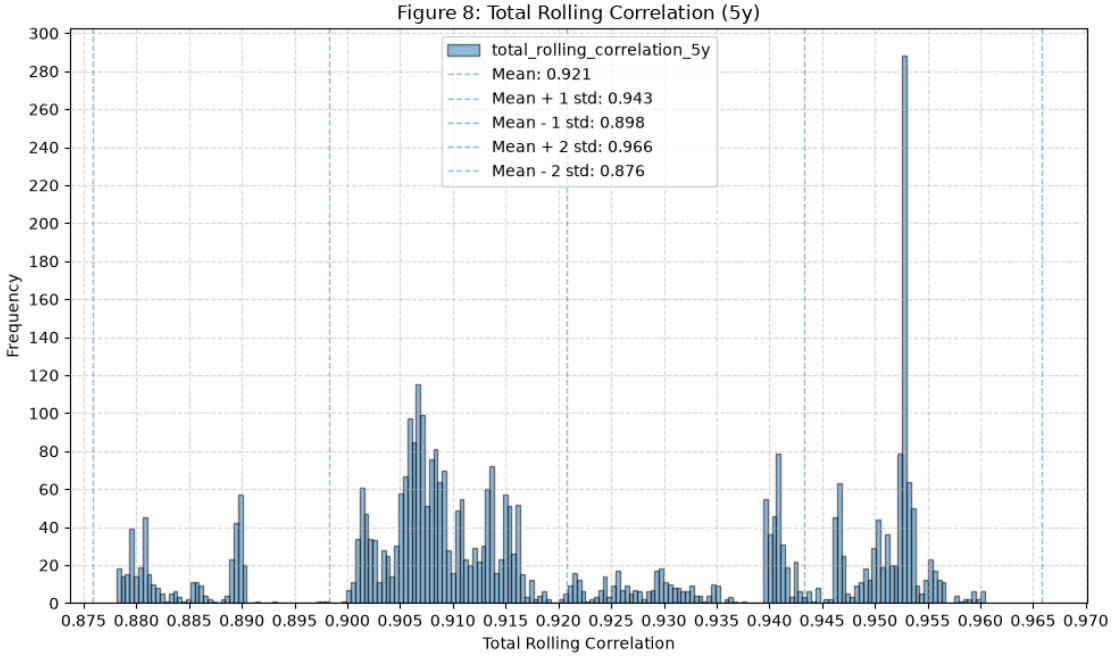
Data Source: Yahoo Finance, Analysis by the Author

Figure 7: Total Rolling Correlation (1y)



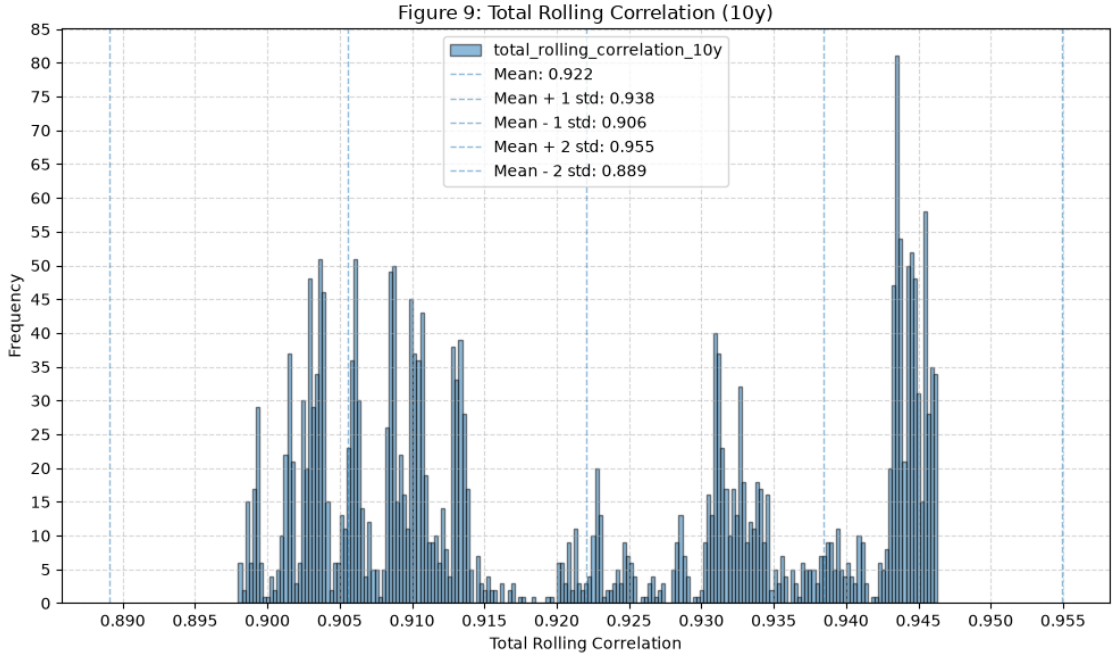
Data Source: Yahoo Finance, Analysis by the Author

Figure 8: Total Rolling Correlation (5y)



Data Source: Yahoo Finance, Analysis by the Author

Figure 9: Total Rolling Correlation (10y)



Data Source: Yahoo Finance, Analysis by the Author

Alright, so the distributions for 1w, 2w, and 1m look interesting. For each of those windows, let's look at the different deciles of the rolling correlations, and what that implies for future returns.

0.7 Future Returns Based On Total Rolling Correlation

First, we need to define the forward return windows that we want to look at.

[13]:
```
# Define rolling windows in trading days
return_windows = {
    "1d": 1,  # 1 day (~1 trading day)
    "3d": 3,  # 3 days (~3 trading days)
    "1w": 5,  # 1 week (~5 trading days)
    "2w": 10,  # 2 weeks (~10 trading days)
    "1m": 21,  # 1 month (~21 trading days)
    "2m": 42,  # 2 months (~42 trading days)
    "3m": 63,  # 3 months (~63 trading days)
}
```

Then, we'll merge in the returns and create the forward returns for each of the funds.

[14]:
```
# Merge daily returns in for the ETFs
total_rolling_corr_with_returns = rolling_corr_results_df.merge(
    fund_data_daily_returns_all[
        [f"{ft}_Daily_Return" for ft in tickers]
    ].dropna(),
    left_index=True,
```

```
    right_index=True,
    how="left",
)

# Calculate compounded rolling returns (CRR), and then shift to create forward returns
for rw in return_windows.keys():
    for ft in tickers:
        total_rolling_corr_with_returns[f"{ft}_Daily_Return_{rw}_CRR"] = (
            total_rolling_corr_with_returns[f"{ft}_Daily_Return"]
            .rolling(window=return_windows[rw])
            .apply(lambda x: (1 + x).prod() - 1, raw=True)
        )
        total_rolling_corr_with_returns[f"{ft}_Daily_Return_fwd_{rw}_CRR"] = total_rolling_corr_with_returns[
            f"{ft}_Daily_Return_{rw}_CRR"
        ].shift(-return_windows[rw])

display(Markdown(f"Table {next(table_number)}: Total Rolling Correlations With Forward Returns"))
display(total_rolling_corr_with_returns)
```

Table 3: Total Rolling Correlations With Forward Returns

	IVV_IJH_3d	IVV_IJH_1w	IVV_IJH_2w	IVV_IJH_1m	IVV_IJH_3m	\
Date						
2006-07-24	NaN	NaN	NaN	NaN	NaN	
2006-07-25	NaN	NaN	NaN	NaN	NaN	
2006-07-26	0.97	NaN	NaN	NaN	NaN	
2006-07-27	0.99	NaN	NaN	NaN	NaN	
2006-07-28	1.00	0.96	NaN	NaN	NaN	
…	…	…	…	…	…	
2026-10-23	0.98	0.83	0.86	0.76	0.80	
2026-10-26	0.98	0.91	0.89	0.78	0.80	
2026-10-27	0.99	0.98	0.89	0.76	0.78	
2026-10-28	0.95	0.98	0.85	0.76	0.79	
2026-10-29	-0.86	0.32	0.71	0.68	0.76	

	IVV_IJH_6m	IVV_IJH_1y	IVV_IJH_5y	IVV_IJH_10y	IVV_IJR_3d	…	\
Date						…	
2006-07-24	NaN	NaN	NaN	NaN	NaN	…	
2006-07-25	NaN	NaN	NaN	NaN	NaN	…	
2006-07-26	NaN	NaN	NaN	NaN	0.96	…	
2006-07-27	NaN	NaN	NaN	NaN	0.93	…	
2006-07-28	NaN	NaN	NaN	NaN	0.99	…	
…	…	…	…	…	…	…	
2026-10-23	0.78	0.80	0.88	0.90	-0.45	…	
2026-10-26	0.78	0.80	0.88	0.90	0.07	…	

```
2026-10-27        0.78        0.80        0.88         0.90        0.99  …
2026-10-28        0.78        0.80        0.88         0.90        0.94  …
2026-10-29        0.77        0.80        0.88         0.90       -0.82  …

            IJH_Daily_Return_2m_CRR  IJH_Daily_Return_fwd_2m_CRR  \
Date
2006-07-24                      NaN                         0.03
2006-07-25                      NaN                         0.01
2006-07-26                      NaN                         0.02
2006-07-27                      NaN                         0.04
2006-07-28                      NaN                         0.03
…                               …                           …
2026-10-23                    -0.05                          NaN
2026-10-26                    -0.05                          NaN
2026-10-27                    -0.04                          NaN
2026-10-28                    -0.03                          NaN
2026-10-29                    -0.01                          NaN

            IJR_Daily_Return_2m_CRR  IJR_Daily_Return_fwd_2m_CRR  \
Date
2006-07-24                      NaN                         0.04
2006-07-25                      NaN                         0.02
2006-07-26                      NaN                         0.04
2006-07-27                      NaN                         0.06
2006-07-28                      NaN                         0.05
…                               …                           …
2026-10-23                    -0.05                          NaN
2026-10-26                    -0.05                          NaN
2026-10-27                    -0.04                          NaN
2026-10-28                    -0.04                          NaN
2026-10-29                    -0.01                          NaN

            IVV_Daily_Return_3m_CRR  IVV_Daily_Return_fwd_3m_CRR  \
Date
2006-07-24                      NaN                         0.09
2006-07-25                      NaN                         0.09
2006-07-26                      NaN                         0.09
2006-07-27                      NaN                         0.10
2006-07-28                      NaN                         0.09
…                               …                           …
2026-10-23                     0.03                          NaN
2026-10-26                     0.04                          NaN
2026-10-27                     0.06                          NaN
2026-10-28                     0.04                          NaN
2026-10-29                     0.03                          NaN

            IJH_Daily_Return_3m_CRR  IJH_Daily_Return_fwd_3m_CRR  \
Date
```

```
2006-07-24                      NaN                          0.07
2006-07-25                      NaN                          0.07
2006-07-26                      NaN                          0.07
2006-07-27                      NaN                          0.09
2006-07-28                      NaN                          0.08
…                               …                            …
2026-10-23                    -0.04                           NaN
2026-10-26                    -0.04                           NaN
2026-10-27                    -0.02                           NaN
2026-10-28                    -0.02                           NaN
2026-10-29                    -0.01                           NaN

            IJR_Daily_Return_3m_CRR  IJR_Daily_Return_fwd_3m_CRR
Date
2006-07-24                      NaN                          0.08
2006-07-25                      NaN                          0.08
2006-07-26                      NaN                          0.09
2006-07-27                      NaN                          0.11
2006-07-28                      NaN                          0.10
…                               …                            …
2026-10-23                    -0.05                           NaN
2026-10-26                    -0.05                           NaN
2026-10-27                    -0.03                           NaN
2026-10-28                    -0.04                           NaN
2026-10-29                    -0.02                           NaN

[5100 rows x 81 columns]
```

0.7.1 Summary By Decile

Before looking at the individual plots, here is a summary of the forward returns for each decile of total rolling correlation compared with the baseline (all days). Values are in percent.

[15]:
```
analysis_windows = ["1w", "2w", "1m"]
decile_labels = [f"D{i}" for i in range(1, 11)]

decile_summary = {}

for window in analysis_windows:
    col = f"total_rolling_correlation_{window}"
    df = total_rolling_corr_with_returns.dropna(subset=[col])

    # Assign each day to a correlation decile and record each decile's range
    bucket = pd.qcut(df[col], 10, labels=decile_labels)
    ranges = df[col].groupby(bucket, observed=True).agg(["min", "max"])
    row_labels = ["Baseline (all days)"] + [
        f"{d} ({ranges.loc[d, 'min']:.2f} to {ranges.loc[d, 'max']:.2f})" for d␣
 ↪in decile_labels
```

```
    ]

    pct_positive = {}
    mean_return = {}
    for rw in return_windows.keys():
        for ft in tickers:
            fwd = df[f"{ft}_Daily_Return_fwd_{rw}_CRR"]
            valid = fwd.notna()
            fwd, b = fwd[valid], bucket[valid]
            pct_positive[(rw, ft)] = [(fwd > 0).mean() * 100] + list(fwd.gt(0).
↪groupby(b, observed=True).mean() * 100)
            mean_return[(rw, ft)] = [fwd.mean() * 100] + list(fwd.groupby(b,␣
↪observed=True).mean() * 100)

    decile_summary[window] = {
        "pct_positive": pd.DataFrame(pct_positive, index=row_labels),
        "mean_return": pd.DataFrame(mean_return, index=row_labels),
    }
```

[16]:
```
for window in analysis_windows:
    display(Markdown(f"Table {next(table_number)}: Percentage of Positive␣
↪Future Returns by {window} Total Rolling Correlation Decile"))
    display(decile_summary[window]["pct_positive"])
    display(Markdown(f"Table {next(table_number)}: Mean Future Return (%) by␣
↪{window} Total Rolling Correlation Decile"))
    display(decile_summary[window]["mean_return"])
```

Table 4: Percentage of Positive Future Returns by 1w Total Rolling Correlation Decile

	1d			3d			1w			\
	IVV	IJH	IJR	IVV	IJH	IJR	IVV	IJH	IJR	
Baseline (all days)	54.84	53.68	52.33	58.77	55.45	53.98	60.20	56.94	54.98	
D1 (-0.28 to 0.68)	54.03	48.13	47.15	57.79	50.30	47.34	60.79	54.26	49.50	
D2 (0.68 to 0.81)	56.47	54.71	51.96	57.84	55.10	50.78	57.45	54.71	50.98	
D3 (0.81 to 0.87)	54.42	49.71	50.49	57.37	53.44	52.46	60.51	55.01	54.22	
D4 (0.87 to 0.91)	56.08	59.02	58.24	62.16	57.25	56.67	63.14	60.39	59.61	
D5 (0.91 to 0.93)	59.33	58.35	55.80	61.30	57.76	56.97	60.51	55.80	53.44	
D6 (0.93 to 0.95)	49.41	49.41	48.24	54.31	52.16	51.37	56.67	54.71	53.92	
D7 (0.95 to 0.96)	54.22	54.62	53.63	61.30	60.31	59.14	63.65	60.71	58.15	
D8 (0.96 to 0.97)	53.33	52.94	50.98	58.63	55.29	53.73	59.41	58.82	54.90	
D9 (0.97 to 0.98)	55.80	56.39	54.03	59.14	57.17	55.60	59.33	57.56	56.78	
D10 (0.98 to 1.00)	55.29	53.53	52.75	57.84	55.69	55.69	60.59	57.45	58.24	

	2w	…	1m			2m			\
	IVV	…	IJR	IVV	IJH	IJR	IVV	IJH	IJR
Baseline (all days)	63.88	…	57.31	67.65	62.34	60.87	71.71	67.10	65.63
D1 (-0.28 to 0.68)	67.46	…	56.35	68.86	56.69	54.29	73.65	64.67	62.08
D2 (0.68 to 0.81)	64.24	…	53.44	67.26	59.76	57.59	74.40	67.80	66.00

D3 (0.81 to 0.87)	64.64	…	55.99	68.12	60.99	58.22	72.51	66.14	62.95
D4 (0.87 to 0.91)	59.33	…	55.01	67.32	60.83	58.66	69.31	65.35	66.34
D5 (0.91 to 0.93)	66.54	…	57.28	66.86	63.91	62.33	68.85	64.29	63.10
D6 (0.93 to 0.95)	57.76	…	55.40	62.67	61.49	60.12	66.54	63.58	62.80
D7 (0.95 to 0.96)	68.96	…	58.74	69.16	64.24	63.26	71.34	67.98	63.64
D8 (0.96 to 0.97)	62.35	…	58.04	68.63	65.29	64.71	73.33	69.22	68.63
D9 (0.97 to 0.98)	65.62	…	61.49	67.19	63.46	64.24	73.23	69.69	70.28
D10 (0.98 to 1.00)	61.96	…	61.37	70.39	66.67	65.10	73.92	72.16	70.39

	3m		
	IVV	IJH	IJR
Baseline (all days)	73.91	69.48	66.70
D1 (-0.28 to 0.68)	78.16	68.74	64.53
D2 (0.68 to 0.81)	76.88	71.81	68.36
D3 (0.81 to 0.87)	76.35	71.14	68.54
D4 (0.87 to 0.91)	75.20	68.25	66.87
D5 (0.91 to 0.93)	70.58	63.82	62.82
D6 (0.93 to 0.95)	66.93	63.76	60.59
D7 (0.95 to 0.96)	72.76	71.17	68.39
D8 (0.96 to 0.97)	72.10	71.91	68.37
D9 (0.97 to 0.98)	72.64	68.70	66.54
D10 (0.98 to 1.00)	77.65	75.49	71.96

[11 rows x 21 columns]

Table 5: Mean Future Return (%) by 1w Total Rolling Correlation Decile

	1d			3d			1w			2w	\
	IVV	IJH	IJR	IVV	IJH	IJR	IVV	IJH	IJR	IVV	
Baseline (all days)	0.05	0.05	0.05	0.15	0.14	0.14	0.24	0.23	0.23	0.48	
D1 (-0.28 to 0.68)	0.07	0.05	0.04	0.13	0.06	0.05	0.28	0.24	0.25	0.56	
D2 (0.68 to 0.81)	0.06	0.04	0.03	0.13	0.10	0.07	0.14	0.04	-0.01	0.37	
D3 (0.81 to 0.87)	-0.00	-0.03	-0.03	0.09	0.03	0.03	0.14	0.05	0.05	0.39	
D4 (0.87 to 0.91)	0.11	0.10	0.08	0.26	0.25	0.23	0.32	0.31	0.30	0.34	
D5 (0.91 to 0.93)	0.13	0.16	0.16	0.24	0.27	0.26	0.31	0.34	0.29	0.63	
D6 (0.93 to 0.95)	-0.07	-0.11	-0.10	-0.04	-0.07	-0.03	0.09	0.11	0.13	0.10	
D7 (0.95 to 0.96)	0.04	0.06	0.07	0.35	0.38	0.38	0.48	0.48	0.48	0.87	
D8 (0.96 to 0.97)	0.08	0.11	0.13	0.10	0.17	0.15	0.16	0.22	0.22	0.30	
D9 (0.97 to 0.98)	0.04	0.04	0.06	0.08	0.09	0.14	0.28	0.31	0.35	0.76	
D10 (0.98 to 1.00)	0.04	0.05	0.05	0.11	0.11	0.12	0.24	0.19	0.23	0.49	

	…	1m			2m			3m		
	…	IJR	IVV	IJH	IJR	IVV	IJH	IJR	IVV	IJH	IJR
Baseline (all days)	…	0.45	1.01	0.95	0.95	2.02	1.90	1.88	3.01	2.85	2.82
D1 (-0.28 to 0.68)	…	0.71	0.97	0.70	0.81	1.90	1.45	1.39	2.95	1.94	1.52
D2 (0.68 to 0.81)	…	0.14	0.85	0.43	0.35	1.82	1.41	1.43	2.93	2.37	2.41
D3 (0.81 to 0.87)	…	0.22	0.89	0.69	0.65	1.82	1.40	1.24	3.16	2.70	2.70
D4 (0.87 to 0.91)	…	0.34	0.67	0.57	0.62	1.71	1.52	1.68	2.62	2.34	2.46
D5 (0.91 to 0.93)	…	0.54	1.08	1.07	1.03	1.70	1.48	1.35	2.26	2.02	1.94

D6 (0.93 to 0.95)	…	0.09	0.56	0.73	0.73	0.85	0.78	0.82	1.71	1.77	1.85
D7 (0.95 to 0.96)	…	0.79	1.38	1.40	1.35	2.11	2.22	2.14	3.24	3.46	3.50
D8 (0.96 to 0.97)	…	0.29	0.93	0.98	0.93	2.34	2.47	2.45	3.32	3.40	3.35
D9 (0.97 to 0.98)	…	0.82	1.22	1.29	1.32	2.49	2.60	2.64	3.35	3.41	3.36
D10 (0.98 to 1.00)	…	0.56	1.59	1.67	1.67	3.43	3.65	3.67	4.56	5.06	5.08

[11 rows x 21 columns]

Table 6: Percentage of Positive Future Returns by 2w Total Rolling Correlation Decile

	1d			3d			1w			\
	IVV	IJH	IJR	IVV	IJH	IJR	IVV	IJH	IJR	
Baseline (all days)	54.85	53.69	52.32	58.77	55.44	53.97	60.20	56.98	55.01	
D1 (-0.03 to 0.76)	55.40	48.33	47.74	60.83	49.02	48.23	64.30	53.85	50.69	
D2 (0.76 to 0.83)	54.81	55.40	54.62	58.46	56.89	52.95	60.75	56.61	53.65	
D3 (0.83 to 0.87)	51.67	53.44	52.46	54.42	52.06	50.88	55.01	53.05	52.26	
D4 (0.87 to 0.90)	57.17	55.80	54.22	57.96	54.03	52.26	58.15	55.99	54.03	
D5 (0.90 to 0.92)	54.22	56.19	51.87	61.69	60.31	56.58	62.28	63.46	56.39	
D6 (0.92 to 0.94)	56.58	54.03	53.63	57.96	56.58	53.83	58.55	55.40	52.46	
D7 (0.94 to 0.95)	55.99	54.03	49.51	60.12	56.58	55.60	57.76	55.01	56.19	
D8 (0.95 to 0.96)	52.06	48.72	50.88	57.17	53.63	55.01	59.33	55.80	54.03	
D9 (0.96 to 0.97)	56.39	54.81	53.24	64.44	60.51	60.51	66.40	64.05	63.06	
D10 (0.97 to 0.99)	54.22	56.19	55.01	54.62	54.81	53.83	59.53	56.58	57.37	

	2w	…	1m			2m			\
	IVV	…	IJR	IVV	IJH	IJR	IVV	IJH	IJR	
Baseline (all days)	63.89	…	57.31	67.61	62.31	60.83	71.68	67.06	65.60	
D1 (-0.03 to 0.76)	69.37	…	55.53	73.11	58.96	55.98	75.85	63.87	64.07	
D2 (0.76 to 0.83)	66.80	…	51.69	70.88	60.44	57.83	74.74	66.32	62.63	
D3 (0.83 to 0.87)	57.17	…	51.08	64.57	59.65	58.07	72.11	68.53	66.33	
D4 (0.87 to 0.90)	63.85	…	57.37	63.78	58.07	57.48	67.98	65.61	61.46	
D5 (0.90 to 0.92)	66.60	…	59.33	70.53	66.60	64.44	75.20	69.69	67.91	
D6 (0.92 to 0.94)	61.30	…	55.80	62.28	60.31	58.35	67.19	63.65	63.65	
D7 (0.94 to 0.95)	57.76	…	56.39	62.67	59.33	58.74	64.64	59.14	58.55	
D8 (0.95 to 0.96)	63.26	…	58.35	61.69	62.28	60.71	66.60	63.46	61.69	
D9 (0.96 to 0.97)	66.60	…	62.67	72.30	66.40	64.05	73.87	71.12	70.73	
D10 (0.97 to 0.99)	66.21	…	64.83	74.46	70.92	72.50	78.78	79.17	78.78	

	3m		
	IVV	IJH	IJR
Baseline (all days)	73.89	69.45	66.67
D1 (-0.03 to 0.76)	79.04	67.27	65.27
D2 (0.76 to 0.83)	78.59	72.77	65.70
D3 (0.83 to 0.87)	77.02	72.18	68.95
D4 (0.87 to 0.90)	71.17	64.61	64.41
D5 (0.90 to 0.92)	75.90	69.72	67.13
D6 (0.92 to 0.94)	70.14	65.62	65.03
D7 (0.94 to 0.95)	63.26	61.89	59.53
D8 (0.95 to 0.96)	65.82	65.42	61.69

```
D9 (0.96 to 0.97)   75.83 74.07 69.55
D10 (0.97 to 0.99)  82.51 81.14 79.37

[11 rows x 21 columns]
```

Table 7: Mean Future Return (%) by 2w Total Rolling Correlation Decile

	1d			3d			1w			2w	\
	IVV	IJH	IJR	IVV	IJH	IJR	IVV	IJH	IJR	IVV	
Baseline (all days)	0.05	0.05	0.05	0.15	0.14	0.14	0.24	0.23	0.23	0.48	
D1 (-0.03 to 0.76)	0.06	0.00	0.00	0.19	0.09	0.13	0.31	0.20	0.31	0.63	
D2 (0.76 to 0.83)	0.05	0.05	0.07	0.16	0.15	0.14	0.24	0.18	0.14	0.55	
D3 (0.83 to 0.87)	0.01	0.01	0.03	0.03	-0.03	-0.02	0.08	-0.01	0.02	0.17	
D4 (0.87 to 0.90)	0.09	0.10	0.09	0.13	0.17	0.17	0.19	0.23	0.18	0.34	
D5 (0.90 to 0.92)	0.06	0.11	0.07	0.20	0.29	0.22	0.38	0.47	0.39	0.57	
D6 (0.92 to 0.94)	0.03	0.02	0.05	0.04	0.03	0.04	0.12	0.17	0.19	0.30	
D7 (0.94 to 0.95)	0.03	0.01	-0.03	0.16	0.13	0.07	0.09	0.01	-0.05	0.14	
D8 (0.95 to 0.96)	0.03	0.01	0.02	0.05	0.01	0.03	0.16	0.15	0.14	0.44	
D9 (0.96 to 0.97)	0.12	0.12	0.12	0.42	0.43	0.49	0.59	0.59	0.64	0.65	
D10 (0.97 to 0.99)	0.02	0.03	0.03	0.10	0.12	0.11	0.27	0.31	0.33	1.05	

	…	1m			2m			3m		
	…	IJR	IVV	IJH	IJR	IVV	IJH	IJR	IVV	IJH	IJR
Baseline (all days)	…	0.45	1.01	0.95	0.95	2.02	1.90	1.88	3.01	2.85	2.82
D1 (-0.03 to 0.76)	…	0.64	1.11	0.51	0.66	1.77	0.90	0.96	2.97	1.41	0.95
D2 (0.76 to 0.83)	…	0.21	1.02	0.63	0.45	1.77	1.21	1.03	2.91	2.03	1.91
D3 (0.83 to 0.87)	…	0.04	0.66	0.57	0.65	2.01	1.71	1.94	3.04	2.63	3.09
D4 (0.87 to 0.90)	…	0.37	0.73	0.95	0.98	1.35	1.39	1.31	2.15	2.09	2.18
D5 (0.90 to 0.92)	…	0.61	1.34	1.42	1.39	2.09	2.07	2.01	2.98	3.02	3.05
D6 (0.92 to 0.94)	…	0.35	0.59	0.66	0.63	1.03	0.88	0.82	1.96	2.01	1.93
D7 (0.94 to 0.95)	…	0.02	0.51	0.41	0.38	1.19	1.11	0.94	1.64	1.78	1.58
D8 (0.95 to 0.96)	…	0.42	0.60	0.60	0.58	1.38	1.46	1.49	2.56	2.74	2.77
D9 (0.96 to 0.97)	…	0.62	1.17	1.11	1.02	3.05	3.27	3.25	3.99	4.27	4.20
D10 (0.97 to 0.99)	…	1.24	2.39	2.66	2.71	4.50	4.97	5.03	5.87	6.44	6.43

```
[11 rows x 21 columns]
```

Table 8: Percentage of Positive Future Returns by 1m Total Rolling Correlation Decile

	1d			3d			1w			\
	IVV	IJH	IJR	IVV	IJH	IJR	IVV	IJH	IJR	
Baseline (all days)	54.85	53.71	52.33	58.76	55.49	53.99	60.22	56.99	55.01	
D1 (0.19 to 0.79)	55.82	51.68	50.30	59.01	52.67	49.11	58.65	52.49	49.90	
D2 (0.79 to 0.84)	55.12	52.17	51.38	59.45	50.98	49.02	61.22	51.38	49.80	
D3 (0.84 to 0.87)	55.31	54.53	51.57	58.46	53.74	50.79	61.22	56.10	51.18	
D4 (0.87 to 0.89)	53.74	54.72	51.38	58.27	57.87	52.95	59.65	61.02	57.28	
D5 (0.89 to 0.92)	56.10	55.31	54.92	60.43	58.46	58.66	59.06	57.68	56.89	
D6 (0.92 to 0.93)	53.35	52.17	51.38	58.86	57.48	55.51	60.63	60.24	59.65	
D7 (0.93 to 0.94)	55.71	53.94	52.56	54.53	50.79	50.98	55.31	52.36	52.56	
D8 (0.94 to 0.95)	55.71	53.74	51.97	59.06	56.50	58.86	64.76	60.04	57.28	

D9 (0.95 to 0.97)	54.33	55.71	54.33	62.20	58.86	55.91	60.43	58.46	56.10
D10 (0.97 to 0.99)	53.35	53.15	53.54	57.28	57.48	58.07	61.22	60.04	59.45

	2w	…	1m			2m			\
	IVV	…	IJR	IVV	IJH	IJR	IVV	IJH	IJR
Baseline (all days)	63.81	…	57.22	67.54	62.25	60.74	71.62	66.99	65.52
D1 (0.19 to 0.79)	63.20	…	54.20	68.02	59.11	55.06	74.39	62.60	59.96
D2 (0.79 to 0.84)	66.80	…	50.79	65.67	55.09	53.69	72.16	65.36	60.62
D3 (0.84 to 0.87)	65.16	…	51.57	72.44	61.02	58.46	78.22	68.91	63.17
D4 (0.87 to 0.89)	60.43	…	56.30	69.69	62.80	60.04	72.83	70.67	69.49
D5 (0.89 to 0.92)	64.76	…	57.09	67.91	63.98	60.63	70.67	64.37	63.98
D6 (0.92 to 0.93)	57.28	…	57.87	61.22	60.24	61.42	58.86	54.13	52.56
D7 (0.93 to 0.94)	59.06	…	55.51	63.98	59.84	58.86	62.80	61.81	61.81
D8 (0.94 to 0.95)	63.98	…	60.04	64.57	63.78	63.19	71.46	70.28	72.05
D9 (0.95 to 0.97)	70.08	…	62.60	69.09	66.93	65.75	74.80	73.62	72.83
D10 (0.97 to 0.99)	67.32	…	66.14	72.83	69.49	70.08	80.12	77.95	78.35

	3m		
	IVV	IJH	IJR
Baseline (all days)	73.83	69.38	66.59
D1 (0.19 to 0.79)	75.00	60.37	57.52
D2 (0.79 to 0.84)	80.58	74.38	70.04
D3 (0.84 to 0.87)	75.05	68.10	67.08
D4 (0.87 to 0.89)	80.75	71.83	71.83
D5 (0.89 to 0.92)	71.85	68.31	65.94
D6 (0.92 to 0.93)	58.86	54.72	51.38
D7 (0.93 to 0.94)	65.16	65.35	58.07
D8 (0.94 to 0.95)	74.02	73.43	71.06
D9 (0.95 to 0.97)	73.82	76.57	73.03
D10 (0.97 to 0.99)	83.66	80.71	79.92

[11 rows x 21 columns]

Table 9: Mean Future Return (%) by 1m Total Rolling Correlation Decile

	1d			3d			1w			2w	\
	IVV	IJH	IJR	IVV	IJH	IJR	IVV	IJH	IJR	IVV	
Baseline (all days)	0.05	0.05	0.05	0.15	0.14	0.14	0.24	0.23	0.23	0.48	
D1 (0.19 to 0.79)	0.02	-0.00	0.00	0.10	0.05	0.08	0.16	0.09	0.15	0.39	
D2 (0.79 to 0.84)	0.06	0.04	0.06	0.16	0.04	0.04	0.28	0.11	0.09	0.54	
D3 (0.84 to 0.87)	0.06	0.05	0.03	0.19	0.15	0.08	0.29	0.18	0.09	0.60	
D4 (0.87 to 0.89)	0.02	0.05	0.04	0.16	0.21	0.19	0.25	0.34	0.33	0.34	
D5 (0.89 to 0.92)	0.10	0.11	0.13	0.19	0.23	0.28	0.23	0.31	0.40	0.53	
D6 (0.92 to 0.93)	0.02	0.02	0.02	0.10	0.15	0.15	0.15	0.24	0.24	0.17	
D7 (0.93 to 0.94)	0.00	0.01	0.01	-0.15	-0.18	-0.16	-0.05	-0.11	-0.11	0.11	
D8 (0.94 to 0.95)	0.03	0.02	0.02	0.17	0.19	0.19	0.28	0.29	0.28	0.35	
D9 (0.95 to 0.97)	0.12	0.10	0.10	0.26	0.21	0.17	0.25	0.18	0.13	0.65	
D10 (0.97 to 0.99)	0.05	0.07	0.07	0.29	0.34	0.35	0.59	0.67	0.68	1.12	

```
                        …        1m                2m                3m
                        …  IJR  IVV  IJH   IJR   IVV   IJH   IJR  IVV  IJH   IJR
Baseline (all days)     … 0.45 1.01 0.95 0.94  2.01  1.89  1.87 3.00 2.84  2.80
D1 (0.19 to 0.79)       … 0.36 0.82 0.50 0.60  1.28  0.20 -0.03 2.50 0.58 -0.14
D2 (0.79 to 0.84)       … 0.25 0.67 0.33 0.33  1.81  1.09  1.08 3.34 2.52  2.66
D3 (0.84 to 0.87)       … 0.20 1.28 0.94 0.76  2.37  1.83  1.81 2.98 2.22  2.50
D4 (0.87 to 0.89)       … 0.49 1.24 1.21 1.27  2.47  2.34  2.54 3.45 3.56  4.09
D5 (0.89 to 0.92)       … 0.73 1.07 1.12 1.22  1.56  1.46  1.62 2.30 2.33  2.71
D6 (0.92 to 0.93)       … 0.37 0.18 0.37 0.36 -0.45 -0.36 -0.56 0.14 0.24 -0.10
D7 (0.93 to 0.94)       … 0.03 0.46 0.27 0.37  0.78  0.83  0.75 1.31 1.51  1.34
D8 (0.94 to 0.95)       … 0.39 0.55 0.64 0.58  2.23  2.60  2.70 3.26 3.72  3.68
D9 (0.95 to 0.97)       … 0.44 1.22 1.31 1.21  2.81  3.22  3.24 3.81 4.29  4.24
D10 (0.97 to 0.99)      … 1.24 2.59 2.82 2.71  5.18  5.64  5.49 6.87 7.32  6.95

[11 rows x 21 columns]
```

0.7.2 Plots By Decile

Now the plots to illustrate the tables above:

[17]:
```
for window in analysis_windows:
    for ticker in tickers:
        # Select this fund's columns (one per return window) and move the␣
↪decile labels into a column
        ticker_df = (
            decile_summary[window]["pct_positive"]
            .xs(ticker, axis=1, level=1)
            .reset_index(names="Decile")
        )

        plot_scatter(
            df=ticker_df,
            x_plot_column="Decile",
            y_plot_columns=list(return_windows.keys()),
            title=f"Figure {next(figure_number)}: {ticker} Percentage of␣
↪Positive Future Returns By Rolling {window} Total Correlation Decile +␣
↪Baseline",
            x_label=f"Rolling {window} Total Correlation Decile or Baseline",
            x_format="String",
            x_format_decimal_places="Auto",
            x_tick_spacing="Auto",
            x_tick_start=None,
            x_tick_rotation=30,
            y_label="Positive Future Returns (%)",
            y_format="Decimal",
            y_format_decimal_places=0,
            y_tick_spacing="Auto",
            y_tick_rotation=0,
```

```
        grid=True,
        legend=True,
        legend_location="upper left",
        legend_anchor=(1, 1),
        caption=f"Data Source: Yahoo Finance, Analysis by the Author",
    )
```

Figure 10: IVV Percentage of Positive Future Returns By Rolling 1w Total Correlation Decile + Baseline

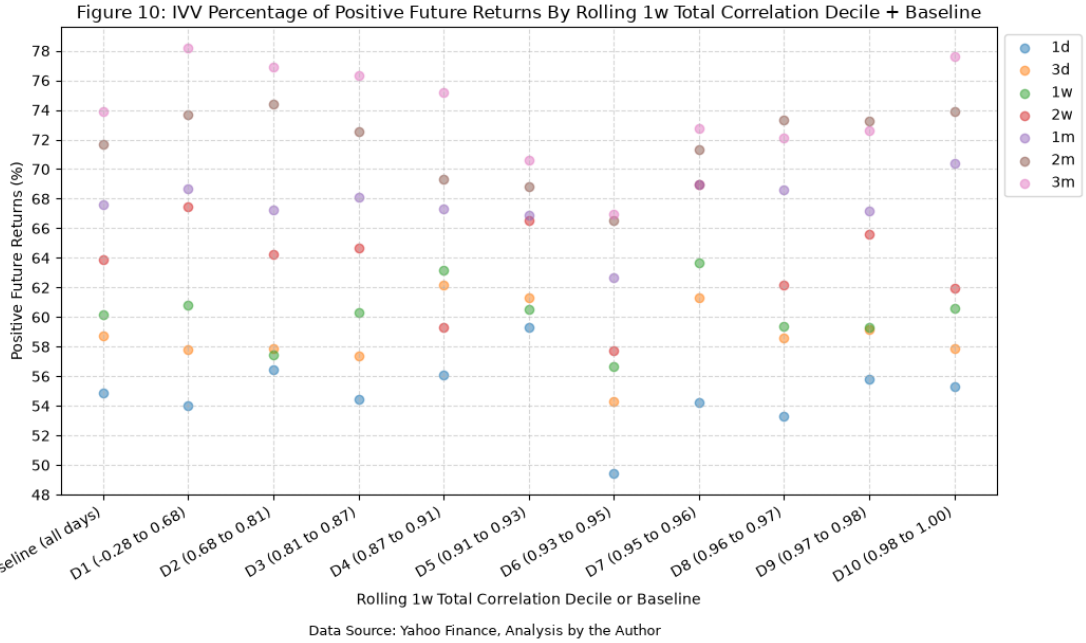


Figure 11: IJH Percentage of Positive Future Returns By Rolling 1w Total Correlation Decile + Baseline

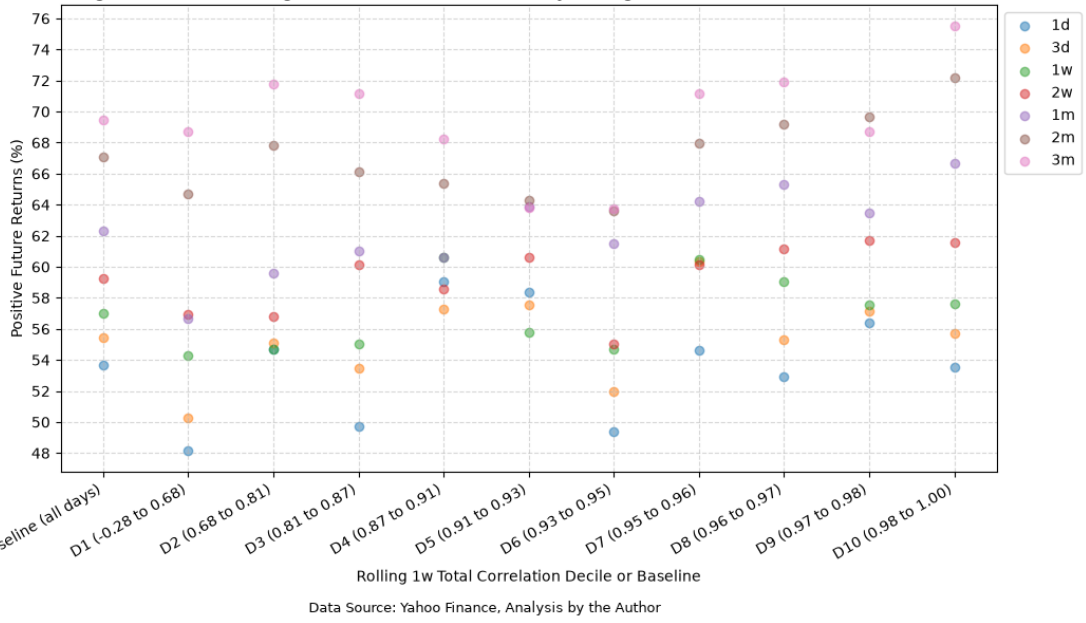
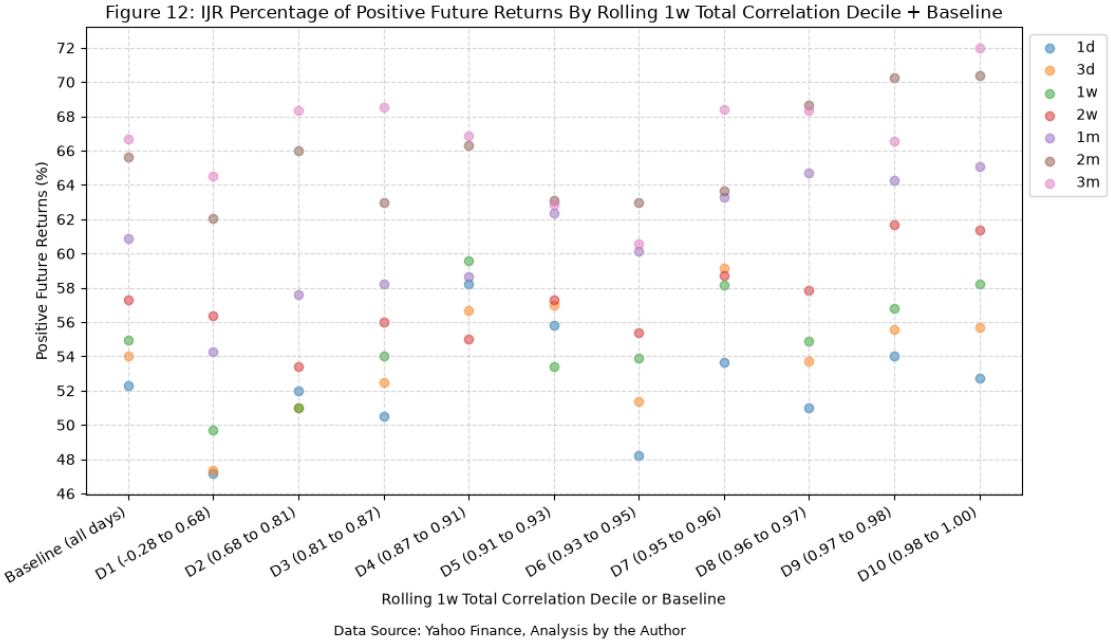
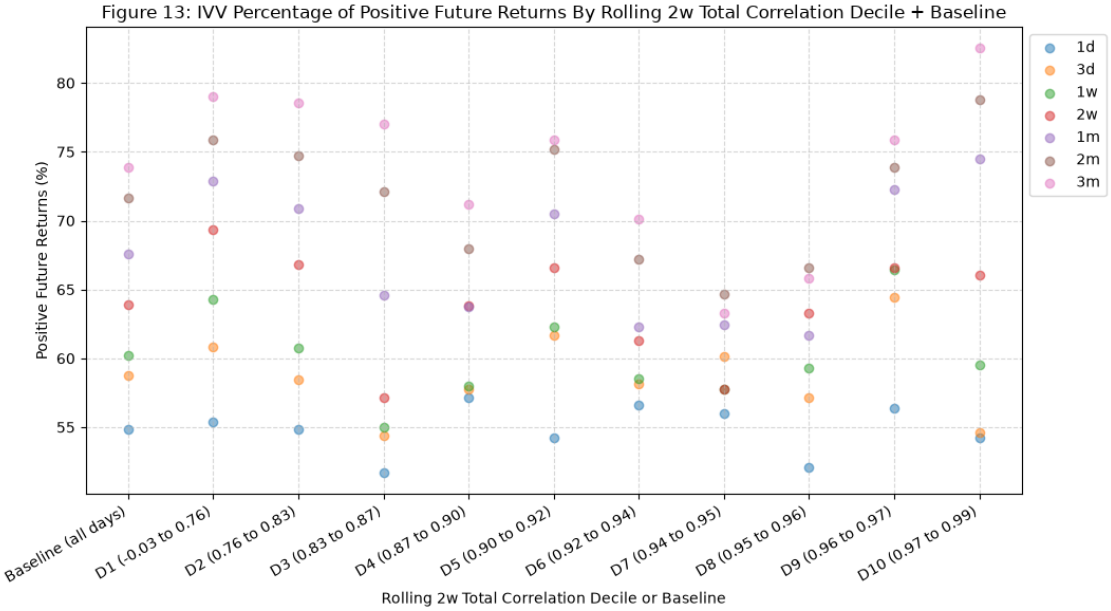


Figure 12: IJR Percentage of Positive Future Returns By Rolling 1w Total Correlation Decile + Baseline



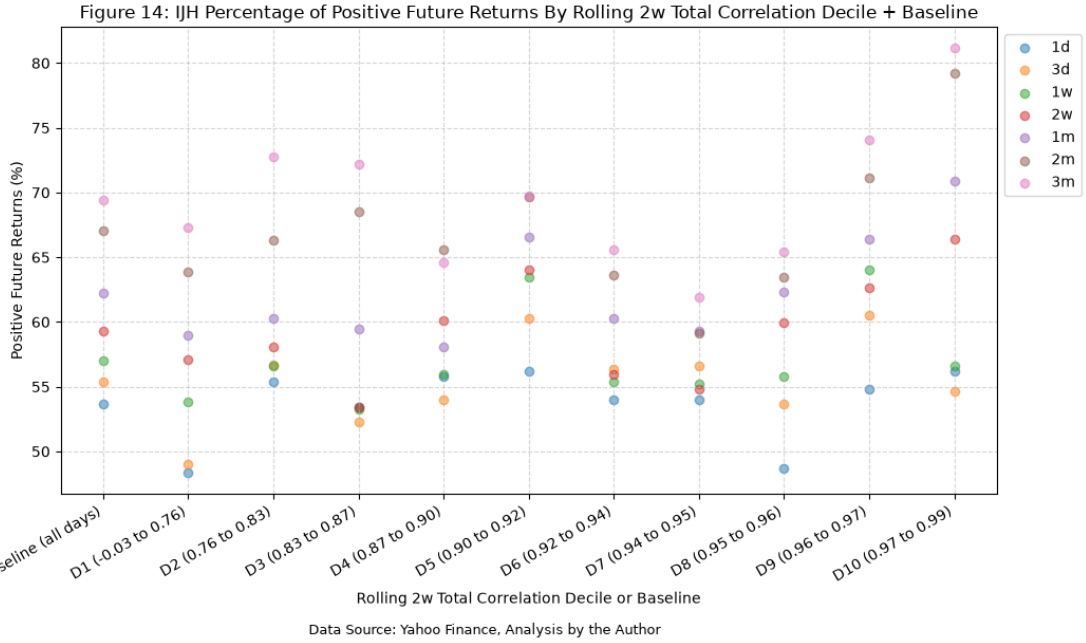
Data Source: Yahoo Finance, Analysis by the Author

Figure 13: IVV Percentage of Positive Future Returns By Rolling 2w Total Correlation Decile + Baseline



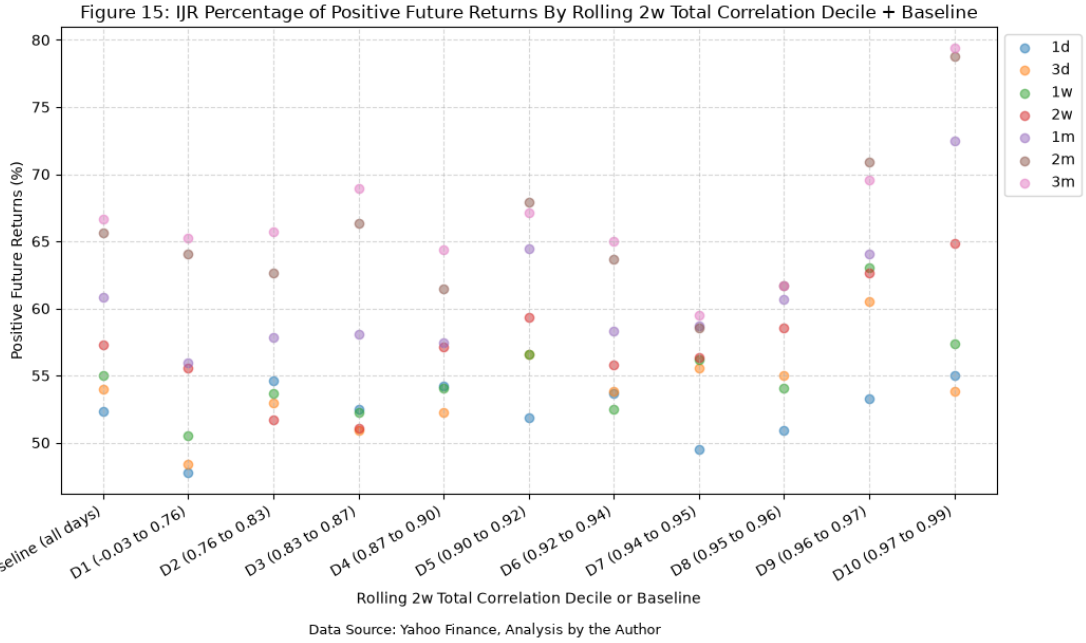
Data Source: Yahoo Finance, Analysis by the Author

Figure 14: IJH Percentage of Positive Future Returns By Rolling 2w Total Correlation Decile + Baseline



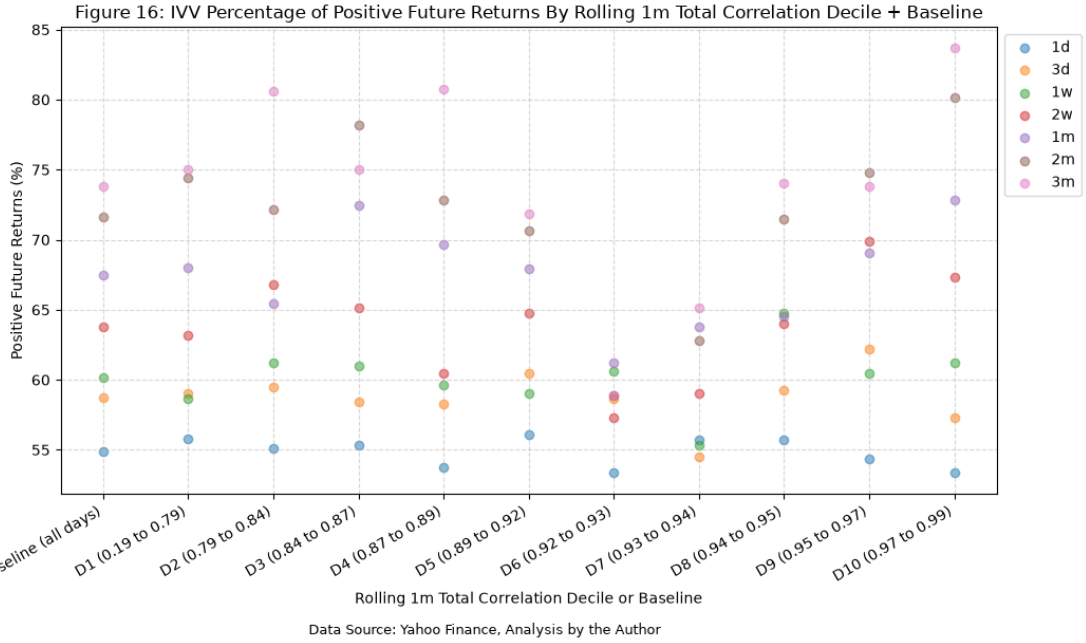
Data Source: Yahoo Finance, Analysis by the Author

Figure 15: IJR Percentage of Positive Future Returns By Rolling 2w Total Correlation Decile + Baseline



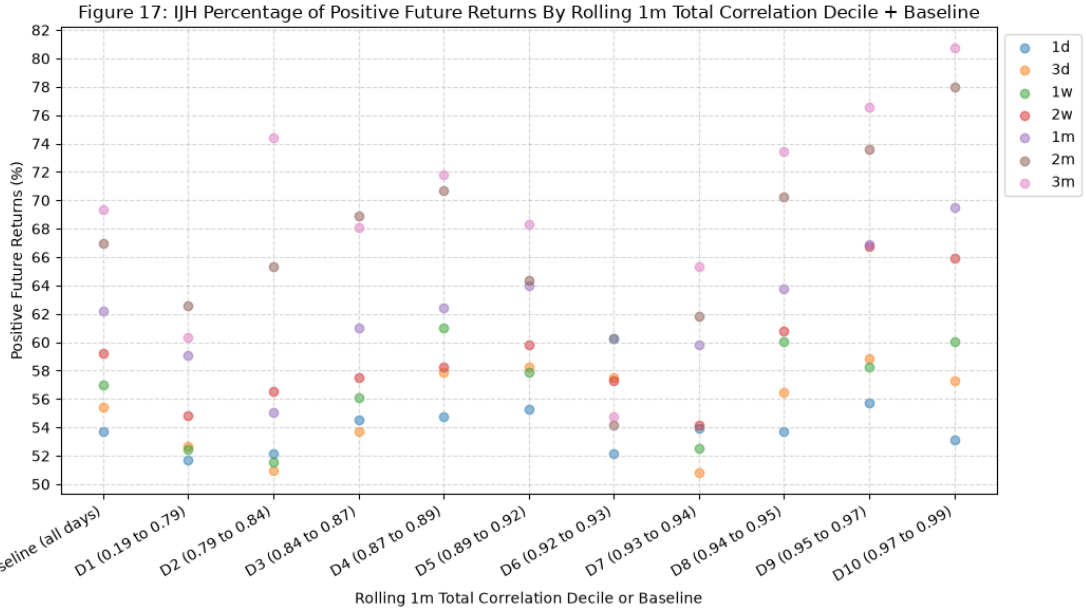
Data Source: Yahoo Finance, Analysis by the Author

Figure 16: IVV Percentage of Positive Future Returns By Rolling 1m Total Correlation Decile + Baseline

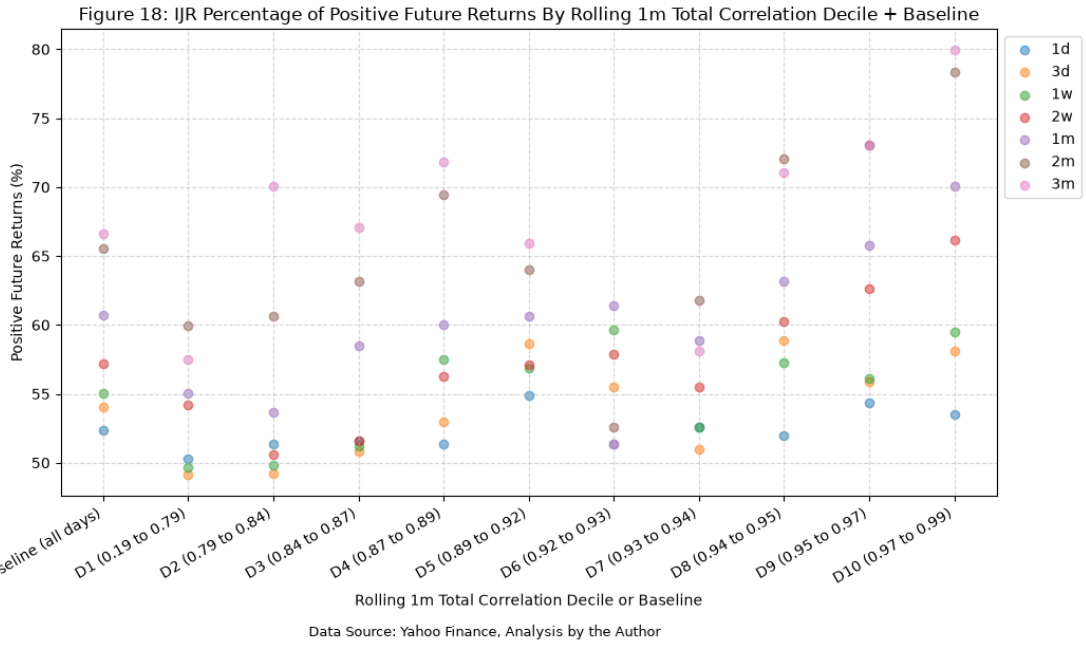


Data Source: Yahoo Finance, Analysis by the Author

Figure 17: IJH Percentage of Positive Future Returns By Rolling 1m Total Correlation Decile + Baseline



Data Source: Yahoo Finance, Analysis by the Author



It's interesting what the baseline (all days) columns reveal about future returns. For *every* plot, the positive future return percentage increases as time increases. We see a similar, but slightly less pronounced, trend for the total rolling correlation deciles, and this suggests that, on average, the likelihood of positive future returns grows over longer time horizons, regardless of the rolling correlation decile. There is certainly a "belly" present in some of the plots, particularly at the 6th and 7th deciles.

The 10th decile consistently shows the highest positive future return percentages across all time horizons, indicating that assets in this decile tend to perform better in the future compared to those in lower deciles, so let's cherry pick the 2 week total rolling correlation for IVV, IJH, and IJR, at the 10th decile, and the 3 month forward return to examine more closely.

0.8 2w Total Rolling Correlation, 10th Decile, 3m Forward Return for IVV, IJH, and IJR

As mentioned above, we will take a closer look at the 2 week total rolling correlation for IVV, IJH, and IJR, at the 10th decile, and the 3 month forward return.

[18]:
```
# Assign each day to a correlation decile and record each decile's range
bucket = pd.qcut(total_rolling_corr_with_returns.
 ↪dropna(subset=["total_rolling_correlation_2w"])["total_rolling_correlation_2w"],␣
 ↪10, labels=decile_labels)

# Merge the decile bucket back into the original DataFrame
total_rolling_corr_with_returns = total_rolling_corr_with_returns.
 ↪dropna(subset=["total_rolling_correlation_2w"])
total_rolling_corr_with_returns["total_rolling_correlation_2w_decile"] = bucket
```

```
display(Markdown(f"Table {next(table_number)}: Total Rolling Correlations With 2w Decile Assignments"))
display(total_rolling_corr_with_returns)
```

Table 10: Total Rolling Correlations With 2w Decile Assignments

Date	IVV_IJH_3d	IVV_IJH_1w	IVV_IJH_2w	IVV_IJH_1m	IVV_IJH_3m	\
2006-08-04	0.98	0.98	0.96	NaN	NaN	
2006-08-07	0.79	0.89	0.91	NaN	NaN	
2006-08-08	0.32	0.93	0.92	NaN	NaN	
2006-08-09	-0.01	0.80	0.93	NaN	NaN	
2006-08-10	0.99	0.89	0.95	NaN	NaN	
…	…	…	…	…	…	
2026-10-23	0.98	0.83	0.86	0.76	0.80	
2026-10-26	0.98	0.91	0.89	0.78	0.80	
2026-10-27	0.99	0.98	0.89	0.76	0.78	
2026-10-28	0.95	0.98	0.85	0.76	0.79	
2026-10-29	-0.86	0.32	0.71	0.68	0.76	

Date	IVV_IJH_6m	IVV_IJH_1y	IVV_IJH_5y	IVV_IJH_10y	IVV_IJR_3d	…	\
							…
2006-08-04	NaN	NaN	NaN	NaN	0.69	…	
2006-08-07	NaN	NaN	NaN	NaN	0.97	…	
2006-08-08	NaN	NaN	NaN	NaN	1.00	…	
2006-08-09	NaN	NaN	NaN	NaN	0.74	…	
2006-08-10	NaN	NaN	NaN	NaN	0.96	…	
…	…	…	…	…	…	…	
2026-10-23	0.78	0.80	0.88	0.90	-0.45	…	
2026-10-26	0.78	0.80	0.88	0.90	0.07	…	
2026-10-27	0.78	0.80	0.88	0.90	0.99	…	
2026-10-28	0.78	0.80	0.88	0.90	0.94	…	
2026-10-29	0.77	0.80	0.88	0.90	-0.82	…	

Date	IJH_Daily_Return_fwd_2m_CRR	IJR_Daily_Return_2m_CRR	\
2006-08-04	0.02	NaN	
2006-08-07	0.04	NaN	
2006-08-08	0.05	NaN	
2006-08-09	0.06	NaN	
2006-08-10	0.06	NaN	
…	…	…	
2026-10-23	NaN	-0.05	
2026-10-26	NaN	-0.05	
2026-10-27	NaN	-0.04	
2026-10-28	NaN	-0.04	
2026-10-29	NaN	-0.01	

Date	IJR_Daily_Return_fwd_2m_CRR	IVV_Daily_Return_3m_CRR	\
2006-08-04	0.03	NaN	
2006-08-07	0.05	NaN	
2006-08-08	0.06	NaN	
2006-08-09	0.07	NaN	
2006-08-10	0.07	NaN	
…	…	…	
2026-10-23	NaN	0.03	
2026-10-26	NaN	0.04	
2026-10-27	NaN	0.06	
2026-10-28	NaN	0.04	
2026-10-29	NaN	0.03	

Date	IVV_Daily_Return_fwd_3m_CRR	IJH_Daily_Return_3m_CRR	\
2006-08-04	0.07	NaN	
2006-08-07	0.07	NaN	
2006-08-08	0.09	NaN	
2006-08-09	0.10	NaN	
2006-08-10	0.09	NaN	
…	…	…	
2026-10-23	NaN	-0.04	
2026-10-26	NaN	-0.04	
2026-10-27	NaN	-0.02	
2026-10-28	NaN	-0.02	
2026-10-29	NaN	-0.01	

Date	IJH_Daily_Return_fwd_3m_CRR	IJR_Daily_Return_3m_CRR	\
2006-08-04	0.04	NaN	
2006-08-07	0.05	NaN	
2006-08-08	0.07	NaN	
2006-08-09	0.09	NaN	
2006-08-10	0.09	NaN	
…	…	…	
2026-10-23	NaN	-0.05	
2026-10-26	NaN	-0.05	
2026-10-27	NaN	-0.03	
2026-10-28	NaN	-0.04	
2026-10-29	NaN	-0.02	

Date	IJR_Daily_Return_fwd_3m_CRR	total_rolling_correlation_2w_decile
2006-08-04	0.05	D7
2006-08-07	0.06	D4
2006-08-08	0.09	D5

```
2006-08-09                      0.10                            D6
2006-08-10                      0.10                            D6
…                               …                               …
2026-10-23                       NaN                            D1
2026-10-26                       NaN                            D2
2026-10-27                       NaN                            D2
2026-10-28                       NaN                            D1
2026-10-29                       NaN                            D1

[5091 rows x 82 columns]
```

[19]:
```
for ticker in tickers:
    plot_scatter(
        ␣
 ↪df=total_rolling_corr_with_returns[total_rolling_corr_with_returns["total_rolling_correlati
 ↪== "D10"],
        x_plot_column="total_rolling_correlation_2w",
        y_plot_columns=[f"{ticker}_Daily_Return_fwd_3m_CRR"],
        title=f"Figure {next(figure_number)}: {ticker} 3 Month Future Returns␣
 ↪vs 2 Week Total Rolling Correlation in Decile 10",
        x_label="2 Week Total Rolling Correlation (Normalized)",
        x_format="Decimal",
        x_format_decimal_places="Auto",
        x_tick_spacing="Auto",
        x_tick_start=None,
        x_tick_rotation=30,
        y_label=f"Future 3 Month Cumulative Rolling Return for {ticker}",
        y_format="Decimal",
        y_format_decimal_places=2,
        y_tick_spacing="Auto",
        y_tick_rotation=0,
        grid=True,
        legend=True,
        export_plot=False,
        plot_file_name=None,
        caption=f"Data Source: Yahoo Finance, Analysis by the Author",
    )

    plot_histogram(
        ␣
 ↪df=total_rolling_corr_with_returns[total_rolling_corr_with_returns["total_rolling_correlati
 ↪== "D10"],
        plot_columns=[f"{ticker}_Daily_Return_fwd_3m_CRR"],
        title=f"Figure {next(figure_number)}: {ticker} Distribution of Future 3␣
 ↪Month Cumulative Rolling Returns from 2 Week Total Rolling Correlation in␣
 ↪Decile 10",
        x_label="Future 3 Month Cumulative Rolling Return",
```

```
        x_tick_spacing="Auto",
        x_tick_rotation=30,
        y_label="Frequency",
        y_tick_spacing="Auto",
        y_tick_rotation=0,
        grid=True,
        legend=True,
        export_plot=False,
        plot_file_name=None,
        caption=f"Data Source: Yahoo Finance, Analysis by the Author",
    )
```

Figure 19: IVV 3 Month Future Returns vs 2 Week Total Rolling Correlation in Decile 10

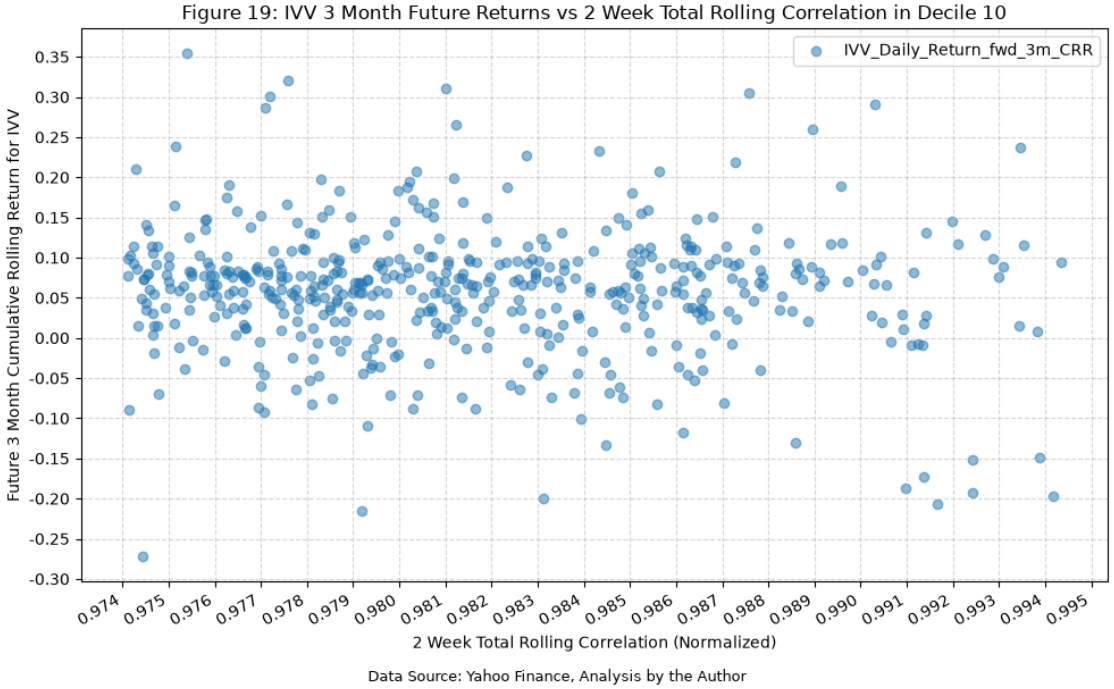
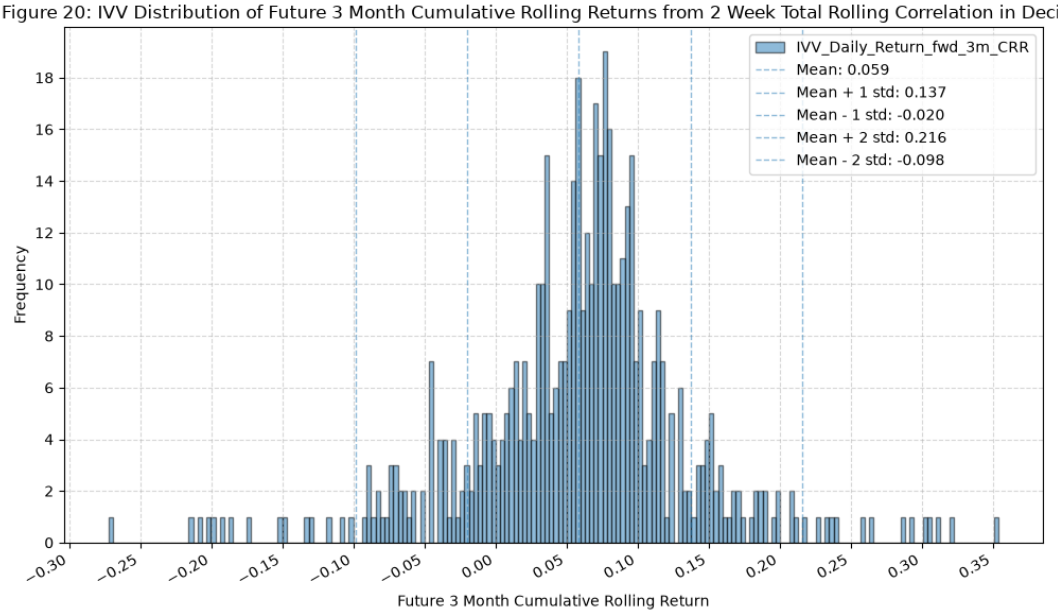
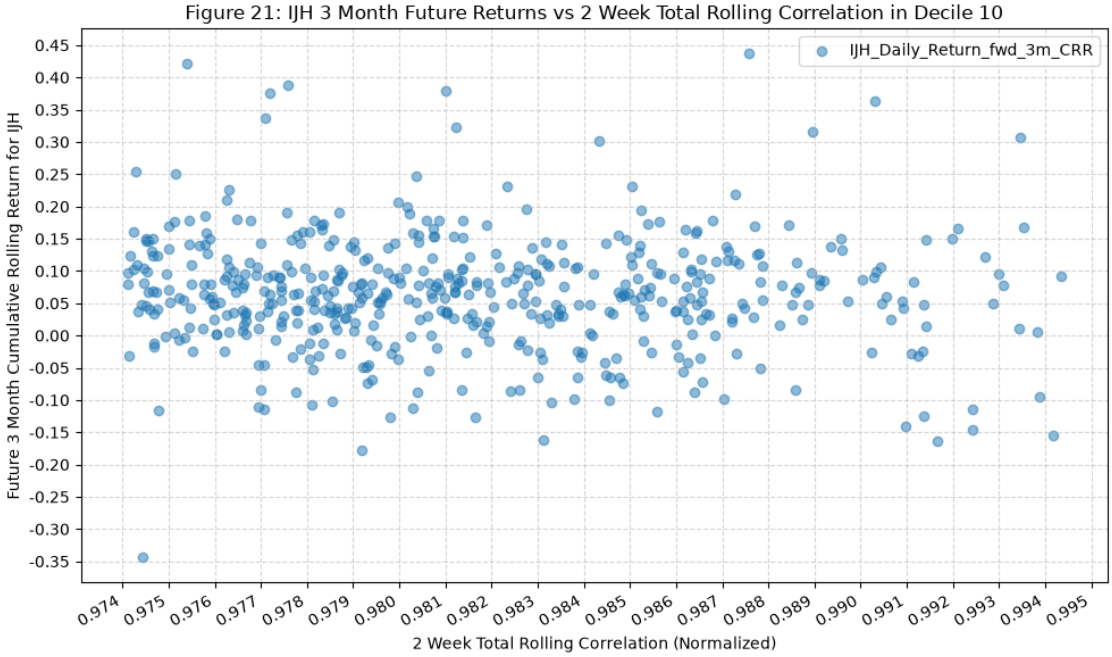


Figure 20: IVV Distribution of Future 3 Month Cumulative Rolling Returns from 2 Week Total Rolling Correlation in Decile 10



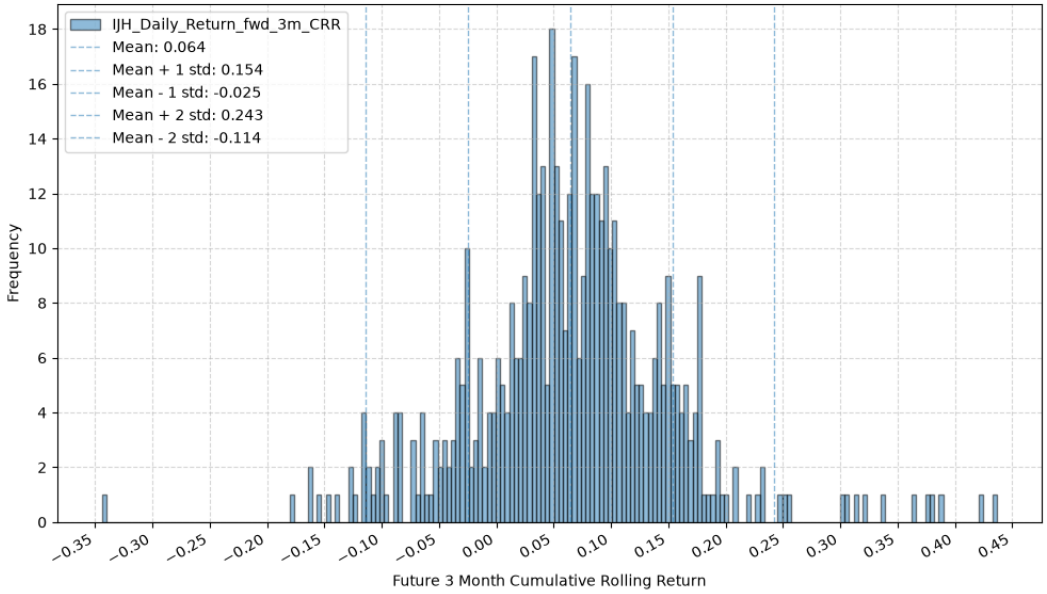
Data Source: Yahoo Finance, Analysis by the Author

Figure 21: IJH 3 Month Future Returns vs 2 Week Total Rolling Correlation in Decile 10



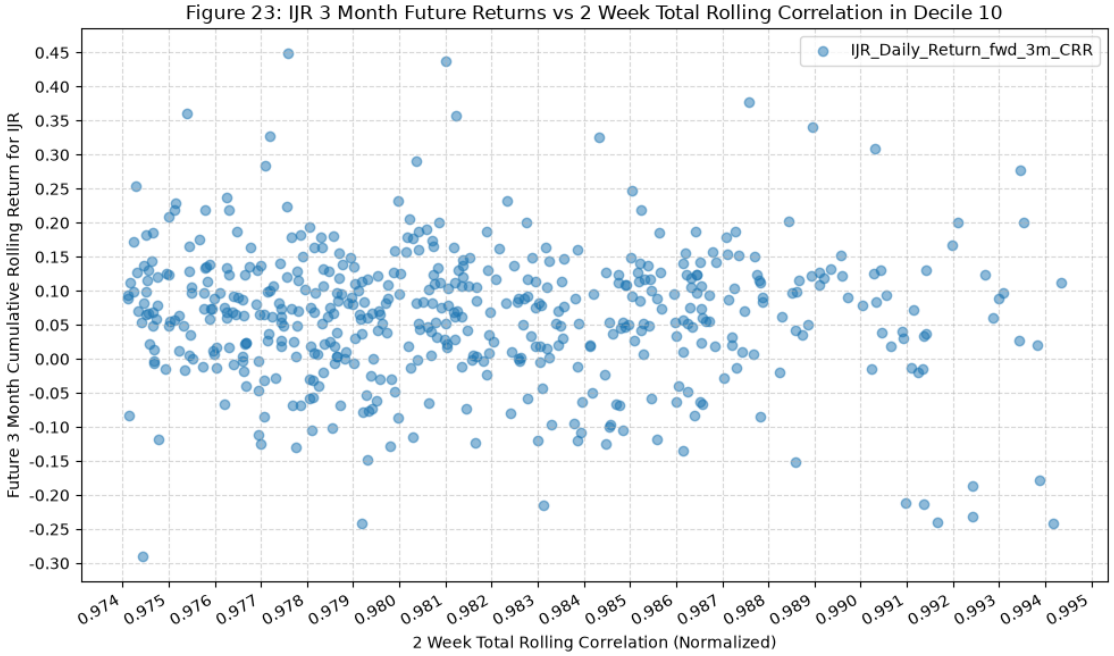
Data Source: Yahoo Finance, Analysis by the Author

Figure 22: IJH Distribution of Future 3 Month Cumulative Rolling Returns from 2 Week Total Rolling Correlation in Decile 10



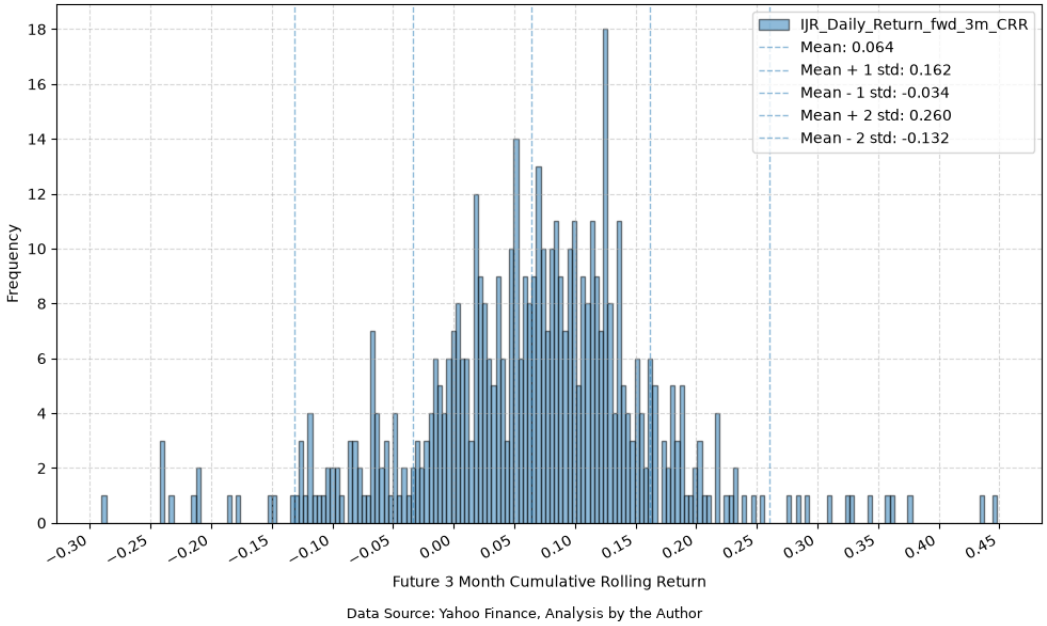
Data Source: Yahoo Finance, Analysis by the Author

Figure 23: IJR 3 Month Future Returns vs 2 Week Total Rolling Correlation in Decile 10



Data Source: Yahoo Finance, Analysis by the Author

Figure 24: IJR Distribution of Future 3 Month Cumulative Rolling Returns from 2 Week Total Rolling Correlation in Decile 10



With all three of these, we have a mean forward return of 3%, with the -1 standard deviation being around -2/-3%, and the +1 standard deviation being around 13-16%. This shows that while the average expected return is modestly positive, there is a range of potential outcomes.

0.9 Trading Strategy

Next, let's develop a simple trading strategy. When the total rolling correlation hits the 10th decile, we will open a long position in the fund, hold the position for exactly 3 months, and then close it. We'll run the strategy separately for each of six funds: IVV, IJH, and IJR, along with their 2x leveraged counterparts SSO, MVV, and SAA. Here's the complete details:

- Use the 10th decile of the total rolling correlation as the signal for entering a trade.
- Position sizing: we always go fully long on the fund when a signal occurs, and we do not scale in or out.
- Entry: when the 2w total correlation decile is D10 on day i, buy the fund at that day's close.
- Exit: hold for exactly 63 trading days (return_windows["3m"]), then sell at that day's close.
- Overlapping signals: a D10 day that comes while a position is open is ignored. If the exit day is itself a D10 day, the strategy buys again, so the position just continues.
- Shared signal: the signal comes from the total correlation across IVV, IJH, and IJR, so the entry and exit dates are the same for all six funds. Only the returns differ. The leveraged funds are only traded, and do not feed into the signal.
- Inception dates: SAA's return history starts on January 26, 2007, after the first trade closes, so its statistics start from that date and cover 43 trades instead of 44.
- Settings: strategy_tickers, signal_decile and hold_days are variables at the top, so we can easily modify them to try other combinations.

We'll start with the strategy, and the complete list of trades.

[20]:
```
# Strategy parameters
strategy_tickers = tickers + leveraged_tickers
signal_column = "total_rolling_correlation_2w_decile"
signal_decile = "D10"
hold_days = return_windows["3m"]  # 63 trading days

# Returns stay NaN before a fund's inception date (SAA starts after the first␣
 ↪signals), so that those
# days and trades are left out of that fund's statistics rather than being␣
 ↪counted as flat.
strategy_df = total_rolling_corr_with_returns[["total_rolling_correlation_2w",␣
 ↪signal_column]].join(
    fund_data[[f"{ticker}_Daily_Return" for ticker in strategy_tickers]]
)
strategy_df["Signal"] = strategy_df[signal_column] == signal_decile

# Walk forward through the days. A signal at the close of day i opens a␣
 ↪position at that close,
# which earns the returns of days i+1 through i+hold_days and is then closed.␣
 ↪Signals that occur
# while a position is already open are ignored. The signal comes from the total␣
 ↪correlation alone,
# so the entry and exit dates are the same for every fund and only the returns␣
 ↪differ.
signals = strategy_df["Signal"].to_numpy()
position = np.zeros(len(strategy_df))
trades = []

i = 0
while i < len(strategy_df) - 1:  # A signal on the last day has no following␣
 ↪days to trade
    if not signals[i]:
        i += 1
        continue

    entry_idx = i
    exit_idx = min(i + hold_days, len(strategy_df) - 1)
    position[entry_idx + 1 : exit_idx + 1] = 1

    trade = {
        "Entry Date": strategy_df.index[entry_idx],
        "Exit Date": strategy_df.index[exit_idx],
        "Entry Correlation": strategy_df["total_rolling_correlation_2w"].
 ↪iloc[entry_idx],
        "Days Held": exit_idx - entry_idx,
```

```
        "Open": exit_idx - entry_idx < hold_days,  # True if the data ends␣
 ↪before the hold period does
    }
    for ticker in strategy_tickers:
        daily_returns = strategy_df[f"{ticker}_Daily_Return"].to_numpy()
        trade[f"{ticker} Return"] = np.prod(1 + daily_returns[entry_idx + 1 :␣
 ↪exit_idx + 1]) - 1
    trades.append(trade)

    i = exit_idx

strategy_df["Position"] = position
for ticker in strategy_tickers:
    strategy_df[f"{ticker}_Strategy_Return"] = strategy_df["Position"] *␣
 ↪strategy_df[f"{ticker}_Daily_Return"]

display(Markdown(f"Table {next(table_number)}: Days in D10 With an Open␣
 ↪Position"))
display(strategy_df[
    (strategy_df["total_rolling_correlation_2w_decile"] == "D10") &
    (strategy_df["Position"] == 1)
])

trades_df = pd.DataFrame(trades)
display(Markdown(f"Table {next(table_number)}: Strategy Trades"))
display(trades_df)
```

Table 11: Days in D10 With an Open Position

Date	total_rolling_correlation_2w	total_rolling_correlation_2w_decile	\
2006-08-21	0.98	D10	
2007-02-28	0.98	D10	
2007-03-01	0.98	D10	
2007-03-02	0.97	D10	
2007-03-06	0.97	D10	
…	…	…	
2025-04-22	0.99	D10	
2025-04-23	0.99	D10	
2026-04-09	0.99	D10	
2026-04-10	0.99	D10	
2026-04-13	0.98	D10	

Date	IVV_Daily_Return	IJH_Daily_Return	IJR_Daily_Return	\
2006-08-21	-0.00	-0.01	-0.01	
2007-02-28	0.01	0.01	0.01	
2007-03-01	-0.00	-0.00	-0.00	

```
2007-03-02             -0.01             -0.02             -0.02
2007-03-06              0.02              0.02              0.02
…                          …                 …                 …
2025-04-22              0.03              0.02              0.02
2025-04-23              0.02              0.01              0.01
2026-04-09              0.01              0.00              0.01
2026-04-10             -0.00             -0.00             -0.00
2026-04-13              0.01              0.01              0.01

            SSO_Daily_Return  MVV_Daily_Return  SAA_Daily_Return  Signal  \
Date
2006-08-21             -0.00             -0.02               NaN    True
2007-02-28              0.02              0.01              0.01    True
2007-03-01             -0.00             -0.01             -0.00    True
2007-03-02             -0.03             -0.03             -0.04    True
2007-03-06              0.03              0.04              0.05    True
…                          …                 …                 …     …
2025-04-22              0.05              0.05              0.05    True
2025-04-23              0.03              0.02              0.03    True
2026-04-09              0.01              0.00              0.02    True
2026-04-10             -0.00             -0.01             -0.02    True
2026-04-13              0.02              0.02              0.02    True

            Position  IVV_Strategy_Return  IJH_Strategy_Return  \
Date
2006-08-21      1.00                -0.00                -0.01
2007-02-28      1.00                 0.01                 0.01
2007-03-01      1.00                -0.00                -0.00
2007-03-02      1.00                -0.01                -0.02
2007-03-06      1.00                 0.02                 0.02
…                  …                    …                    …
2025-04-22      1.00                 0.03                 0.02
2025-04-23      1.00                 0.02                 0.01
2026-04-09      1.00                 0.01                 0.00
2026-04-10      1.00                -0.00                -0.00
2026-04-13      1.00                 0.01                 0.01

            IJR_Strategy_Return  SSO_Strategy_Return  MVV_Strategy_Return  \
Date
2006-08-21                -0.01                -0.00                -0.02
2007-02-28                 0.01                 0.02                 0.01
2007-03-01                -0.00                -0.00                -0.01
2007-03-02                -0.02                -0.03                -0.03
2007-03-06                 0.02                 0.03                 0.04
…                             …                    …                    …
2025-04-22                 0.02                 0.05                 0.05
2025-04-23                 0.01                 0.03                 0.02
2026-04-09                 0.01                 0.01                 0.00
```

2026-04-10	-0.00	-0.00	-0.01
2026-04-13	0.01	0.02	0.02

Date	SAA_Strategy_Return
2006-08-21	NaN
2007-02-28	0.01
2007-03-01	-0.00
2007-03-02	-0.04
2007-03-06	0.05
…	…
2025-04-22	0.05
2025-04-23	0.03
2026-04-09	0.02
2026-04-10	-0.02
2026-04-13	0.02

[472 rows x 16 columns]

Table 12: Strategy Trades

	Entry Date	Exit Date	Entry Correlation	Days Held	Open	IVV Return	\
0	2006-08-17	2006-11-15	0.98	63	False	0.08	
1	2007-02-27	2007-05-29	0.98	63	False	0.10	
2	2007-06-14	2007-09-13	0.98	63	False	-0.02	
3	2007-09-27	2007-12-27	0.98	63	False	-0.03	
4	2008-03-07	2008-06-06	0.98	63	False	0.06	
5	2008-06-06	2008-09-05	0.99	63	False	-0.08	
6	2008-11-14	2009-02-18	0.97	63	False	-0.09	
7	2009-02-24	2009-05-26	0.98	63	False	0.19	
8	2009-05-27	2009-08-25	0.98	63	False	0.16	
9	2009-08-25	2009-11-23	0.99	63	False	0.08	
10	2009-11-23	2010-02-25	0.98	63	False	0.00	
11	2010-05-10	2010-08-09	0.98	63	False	-0.02	
12	2010-09-03	2010-12-03	0.97	63	False	0.11	
13	2011-02-09	2011-05-11	0.99	63	False	0.02	
14	2011-06-01	2011-08-30	0.98	63	False	-0.07	
15	2011-08-30	2011-11-29	0.99	63	False	-0.01	
16	2011-11-30	2012-03-02	0.99	63	False	0.11	
17	2012-04-18	2012-07-18	0.98	63	False	-0.00	
18	2012-11-19	2013-02-21	0.99	63	False	0.09	
19	2013-02-21	2013-05-22	0.98	63	False	0.11	
20	2013-06-20	2013-09-19	0.98	63	False	0.09	
21	2013-10-14	2014-01-14	0.97	63	False	0.08	
22	2014-01-24	2014-04-25	0.99	63	False	0.05	
23	2015-07-13	2015-10-09	0.98	63	False	-0.04	
24	2015-10-16	2016-01-19	0.97	63	False	-0.07	
25	2016-02-03	2016-05-04	0.98	63	False	0.07	
26	2016-06-13	2016-09-12	0.97	63	False	0.04	

27	2016-09-21	2016-12-20	0.98	63	False	0.06
28	2017-08-30	2017-11-29	0.97	63	False	0.07
29	2017-12-29	2018-04-03	0.97	63	False	-0.02
30	2018-04-03	2018-07-02	0.99	63	False	0.05
31	2018-12-12	2019-03-15	0.98	63	False	0.07
32	2019-03-22	2019-06-21	0.98	63	False	0.06
33	2019-08-13	2019-11-11	0.98	63	False	0.06
34	2020-03-04	2020-06-03	0.98	63	False	0.00
35	2020-06-24	2020-09-23	0.99	63	False	0.07
36	2022-02-28	2022-05-27	0.99	63	False	-0.05
37	2022-06-09	2022-09-09	0.98	63	False	0.02
38	2022-09-20	2022-12-19	0.98	63	False	-0.01
39	2022-12-19	2023-03-22	0.98	63	False	0.04
40	2024-08-14	2024-11-12	0.98	63	False	0.10
41	2024-12-23	2025-03-27	0.98	63	False	-0.04
42	2025-04-03	2025-07-07	0.98	63	False	0.16
43	2026-04-08	2026-07-09	0.99	63	False	0.12

	IJH Return	IJR Return	SSO Return	MVV Return	SAA Return
0	0.08	0.10	0.15	0.15	NaN
1	0.09	0.08	0.18	0.15	0.15
2	-0.05	-0.05	-0.07	-0.12	-0.12
3	-0.03	-0.07	-0.08	-0.07	-0.15
4	0.15	0.11	0.09	0.30	0.21
5	-0.10	-0.03	-0.17	-0.21	-0.08
6	-0.03	-0.08	-0.22	-0.14	-0.23
7	0.23	0.23	0.36	0.43	0.43
8	0.18	0.19	0.32	0.37	0.39
9	0.05	0.03	0.16	0.10	0.04
10	0.07	0.07	-0.00	0.13	0.13
11	-0.02	-0.05	-0.06	-0.06	-0.11
12	0.16	0.17	0.23	0.34	0.36
13	0.05	0.05	0.04	0.09	0.09
14	-0.10	-0.10	-0.16	-0.23	-0.21
15	-0.03	-0.02	-0.04	-0.09	-0.08
16	0.11	0.09	0.22	0.22	0.18
17	-0.03	0.00	-0.02	-0.06	-0.00
18	0.13	0.14	0.18	0.27	0.29
19	0.10	0.09	0.22	0.18	0.18
20	0.10	0.12	0.18	0.21	0.25
21	0.07	0.07	0.16	0.13	0.14
22	0.03	0.01	0.09	0.04	0.00
23	-0.04	-0.05	-0.08	-0.10	-0.10
24	-0.12	-0.12	-0.14	-0.23	-0.24
25	0.12	0.12	0.15	0.24	0.32
26	0.05	0.06	0.08	0.09	0.09
27	0.10	0.14	0.11	0.19	0.31
28	0.10	0.14	0.14	0.21	0.24

29	-0.02	-0.00	-0.06	-0.05	-0.06
30	0.05	0.11	0.09	0.10	0.21
31	0.07	0.05	0.12	0.13	0.07
32	0.04	0.02	0.11	0.08	0.03
33	0.06	0.06	0.11	0.10	0.11
34	-0.02	-0.07	-0.09	-0.17	-0.26
35	0.04	0.03	0.12	0.06	0.05
36	-0.04	-0.05	-0.11	-0.11	-0.12
37	0.02	-0.02	0.01	0.01	-0.06
38	0.01	0.01	-0.04	-0.01	-0.02
39	0.01	-0.00	0.05	-0.01	-0.04
40	0.12	0.13	0.19	0.22	0.24
41	-0.05	-0.08	-0.11	-0.12	-0.17
42	0.14	0.13	0.29	0.25	0.22
43	0.08	0.12	0.22	0.14	0.24

Next, here's the statistics for win rate, average return, max gain, and max loss.

[21]:
```
# Trade summary
closed_trades = trades_df[~trades_df["Open"]]

print(f"Total trades: {len(trades_df)} ({len(closed_trades)} closed)")
print(f"Time in market: {strategy_df['Position'].mean() * 100:.2f}%")

for ticker in strategy_tickers:
    trade_returns = closed_trades[f"{ticker} Return"].dropna()
    print(f"\n{ticker}:")
    print(f"  Closed trades: {len(trade_returns)}")
    print(f"  Win rate (closed trades): {(trade_returns > 0).mean() * 100:.
 ↪2f}%")
    print(f"  Average return per closed trade: {trade_returns.mean() * 100:.
 ↪2f}%")
    print(f"  Max trade gain: {trade_returns.max() * 100:.2f}%")
    print(f"  Max trade loss: {trade_returns.min() * 100:.2f}%")
```

```
Total trades: 44 (44 closed)
Time in market: 54.45%

IVV:
  Closed trades: 44
  Win rate (closed trades): 68.18%
  Average return per closed trade: 3.96%
  Max trade gain: 18.70%
  Max trade loss: -9.01%

IJH:
  Closed trades: 44
  Win rate (closed trades): 68.18%
```

```
  Average return per closed trade: 4.37%
  Max trade gain: 23.09%
  Max trade loss: -11.54%

IJR:
  Closed trades: 44
  Win rate (closed trades): 65.91%
  Average return per closed trade: 4.32%
  Max trade gain: 23.17%
  Max trade loss: -11.87%

SSO:
  Closed trades: 44
  Win rate (closed trades): 63.64%
  Average return per closed trade: 6.60%
  Max trade gain: 35.67%
  Max trade loss: -21.76%

MVV:
  Closed trades: 44
  Win rate (closed trades): 63.64%
  Average return per closed trade: 7.14%
  Max trade gain: 43.41%
  Max trade loss: -22.68%

SAA:
  Closed trades: 43
  Win rate (closed trades): 60.47%
  Average return per closed trade: 6.75%
  Max trade gain: 43.02%
  Max trade loss: -25.61%
```

Now we can compare the summary stats for the different funds and see how the strategy performed across all six funds.

[22]:
```
# Compare each strategy against buy and hold for the same period
strategy_stats = pd.concat(
    [
        summary_stats(
            fund_list=[ticker],
            df=strategy_df[[return_column]].dropna(),
            period="Daily",
            use_calendar_days=False,
            excel_export=False,
            pickle_export=False,
            output_confirmation=False,
        )
        for ticker in strategy_tickers
```

```
        for return_column in [f"{ticker}_Strategy_Return",␣
 ↪f"{ticker}_Daily_Return"]
    ]
)

display(Markdown(f"Table {next(table_number)}: Strategy vs Buy and Hold Summary␣
 ↪Statistics"))
display(strategy_stats)
```

Table 13: Strategy vs Buy and Hold Summary Statistics

	Number of Observations	Data Start Date	Data End Date	\
IVV_Strategy_Return	5091	2006-08-04	2026-10-29	
IVV_Daily_Return	5091	2006-08-04	2026-10-29	
IJH_Strategy_Return	5091	2006-08-04	2026-10-29	
IJH_Daily_Return	5091	2006-08-04	2026-10-29	
IJR_Strategy_Return	5091	2006-08-04	2026-10-29	
IJR_Daily_Return	5091	2006-08-04	2026-10-29	
SSO_Strategy_Return	5091	2006-08-04	2026-10-29	
SSO_Daily_Return	5091	2006-08-04	2026-10-29	
MVV_Strategy_Return	5091	2006-08-04	2026-10-29	
MVV_Daily_Return	5091	2006-08-04	2026-10-29	
SAA_Strategy_Return	4972	2007-01-26	2026-10-29	
SAA_Daily_Return	4972	2007-01-26	2026-10-29	

	Annual Mean Return (Arithmetic)	Annualized Volatility	\
IVV_Strategy_Return	0.09	0.15	
IVV_Daily_Return	0.13	0.19	
IJH_Strategy_Return	0.10	0.18	
IJH_Daily_Return	0.12	0.22	
IJR_Strategy_Return	0.10	0.19	
IJR_Daily_Return	0.12	0.24	
SSO_Strategy_Return	0.17	0.31	
SSO_Daily_Return	0.22	0.38	
MVV_Strategy_Return	0.19	0.36	
MVV_Daily_Return	0.20	0.44	
SAA_Strategy_Return	0.18	0.38	
SAA_Daily_Return	0.20	0.48	

	Annualized Sharpe Ratio	CAGR (Geometric)	\
IVV_Strategy_Return	0.60	0.08	
IVV_Daily_Return	0.65	0.11	
IJH_Strategy_Return	0.58	0.09	
IJH_Daily_Return	0.54	0.10	
IJR_Strategy_Return	0.54	0.09	
IJR_Daily_Return	0.50	0.09	
SSO_Strategy_Return	0.55	0.13	
SSO_Daily_Return	0.57	0.16	

MVV_Strategy_Return	0.53	0.13
MVV_Daily_Return	0.46	0.11
SAA_Strategy_Return	0.48	0.12
SAA_Daily_Return	0.42	0.09

	Daily Max Return	Daily Max Return (Date)
IVV_Strategy_Return	0.10	2025-04-09
IVV_Daily_Return	0.11	2008-10-28
IJH_Strategy_Return	0.11	2020-03-24
IJH_Daily_Return	0.11	2020-03-24
IJR_Strategy_Return	0.09	2025-04-09
IJR_Daily_Return	0.09	2025-04-09
SSO_Strategy_Return	0.19	2020-03-24
SSO_Daily_Return	0.22	2008-10-13
MVV_Strategy_Return	0.20	2025-04-09
MVV_Daily_Return	0.25	2008-10-13
SAA_Strategy_Return	0.18	2025-04-09
SAA_Daily_Return	0.18	2025-04-09

	Daily Min Return	Daily Min Return (Date)	Max Drawdown
IVV_Strategy_Return	-0.12	2020-03-16	-0.30
IVV_Daily_Return	-0.12	2020-03-16	-0.55
IJH_Strategy_Return	-0.14	2020-03-16	-0.36
IJH_Daily_Return	-0.14	2020-03-16	-0.55
IJR_Strategy_Return	-0.12	2020-03-16	-0.35
IJR_Daily_Return	-0.12	2020-03-16	-0.58
SSO_Strategy_Return	-0.23	2020-03-16	-0.55
SSO_Daily_Return	-0.23	2020-03-16	-0.85
MVV_Strategy_Return	-0.29	2020-03-16	-0.62
MVV_Daily_Return	-0.29	2020-03-16	-0.86
SAA_Strategy_Return	-0.25	2020-03-16	-0.61
SAA_Daily_Return	-0.25	2020-03-16	-0.87

	Peak	Trough	Recovery Date
IVV_Strategy_Return	2008-05-19	2009-03-09	2009-06-01
IVV_Daily_Return	2007-10-09	2009-03-09	2012-08-16
IJH_Strategy_Return	2019-11-08	2020-03-23	2020-06-30
IJH_Daily_Return	2007-06-04	2009-03-09	2010-12-02
IJR_Strategy_Return	2019-11-05	2020-03-23	2020-07-21
IJR_Daily_Return	2007-07-19	2009-03-09	2011-02-07
SSO_Strategy_Return	2007-07-13	2009-03-09	2009-08-07
SSO_Daily_Return	2007-10-09	2009-03-09	2013-10-22
MVV_Strategy_Return	2019-11-05	2020-03-23	2020-08-05
MVV_Daily_Return	2007-06-04	2009-03-09	2013-04-30
SAA_Strategy_Return	2019-11-05	2020-03-23	NaT
SAA_Daily_Return	2007-06-04	2009-03-09	2013-07-09

	Calendar Days to Recovery	MAR Ratio

```
IVV_Strategy_Return                  84.00       0.28
IVV_Daily_Return                   1256.00       0.20
IJH_Strategy_Return                  99.00       0.25
IJH_Daily_Return                    633.00       0.18
IJR_Strategy_Return                 120.00       0.25
IJR_Daily_Return                    700.00       0.16
SSO_Strategy_Return                 151.00       0.23
SSO_Daily_Return                   1688.00       0.18
MVV_Strategy_Return                 135.00       0.21
MVV_Daily_Return                   1513.00       0.13
SAA_Strategy_Return                    NaN       0.19
SAA_Daily_Return                   1583.00       0.10
```

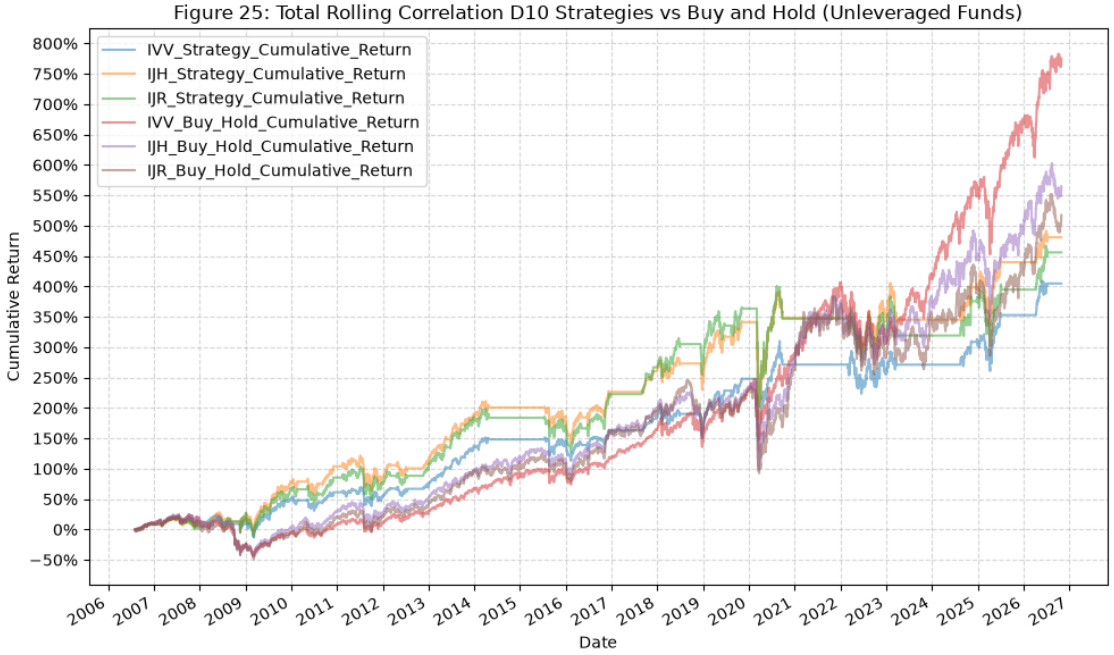
And finally, the plots for the cumulative returns and drawdowns.

[23]:
```
# Cumulative returns for each strategy and buy and hold
for ticker in strategy_tickers:
    strategy_df[f"{ticker}_Strategy_Cumulative_Return"] = (1 +␣
 ↪strategy_df[f"{ticker}_Strategy_Return"]).cumprod() - 1
    strategy_df[f"{ticker}_Buy_Hold_Cumulative_Return"] = (1 +␣
 ↪strategy_df[f"{ticker}_Daily_Return"]).cumprod() - 1

for fund_group, group_tickers in {"Unleveraged": tickers, "2x Leveraged":␣
 ↪leveraged_tickers}.items():
    plot_time_series(
        df=strategy_df,
        plot_start_date=None,
        plot_end_date=None,
        plot_columns=[f"{ticker}_Strategy_Cumulative_Return" for ticker in␣
 ↪group_tickers]
        + [f"{ticker}_Buy_Hold_Cumulative_Return" for ticker in group_tickers],
        title=f"Figure {next(figure_number)}: Total Rolling Correlation␣
 ↪{signal_decile} Strategies vs Buy and Hold ({fund_group} Funds)",
        x_label="Date",
        x_format="Year",
        x_tick_spacing=1,
        x_tick_start=None,
        x_tick_rotation=30,
        y_label="Cumulative Return",
        y_format="Percentage",
        y_format_decimal_places=0,
        y_tick_spacing="Auto",
        y_tick_rotation=0,
        grid=True,
        legend=True,
        export_plot=False,
        plot_file_name=None,
```

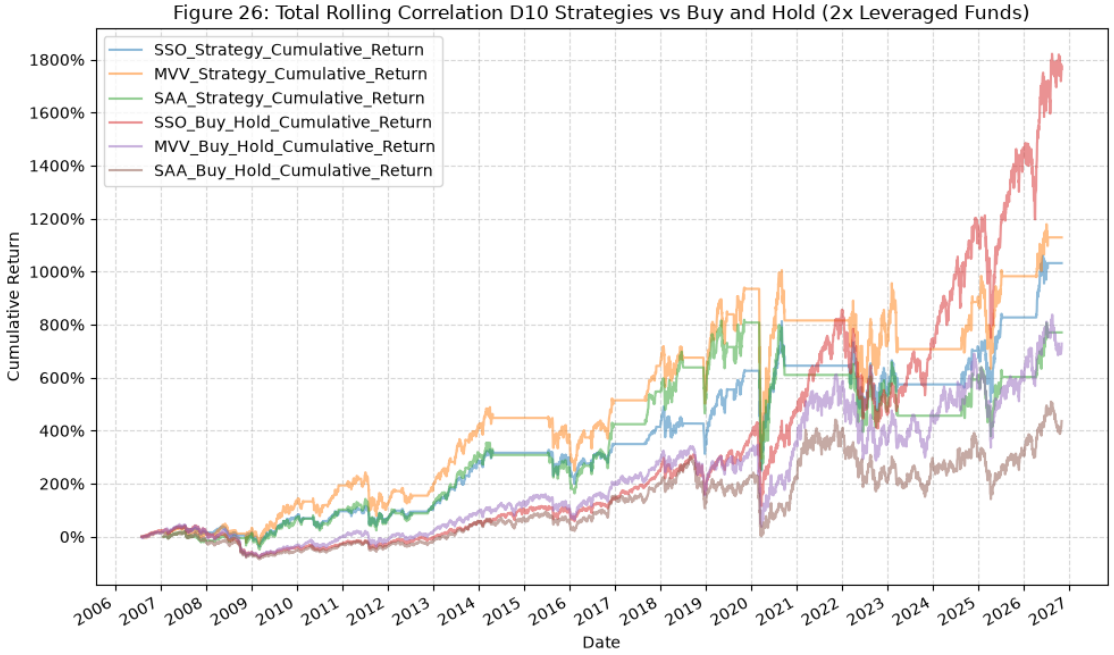
```
        caption=f"Data Source: Yahoo Finance, Analysis by the Author",
    )
```

Figure 25: Total Rolling Correlation D10 Strategies vs Buy and Hold (Unleveraged Funds)



Data Source: Yahoo Finance, Analysis by the Author

Figure 26: Total Rolling Correlation D10 Strategies vs Buy and Hold (2x Leveraged Funds)



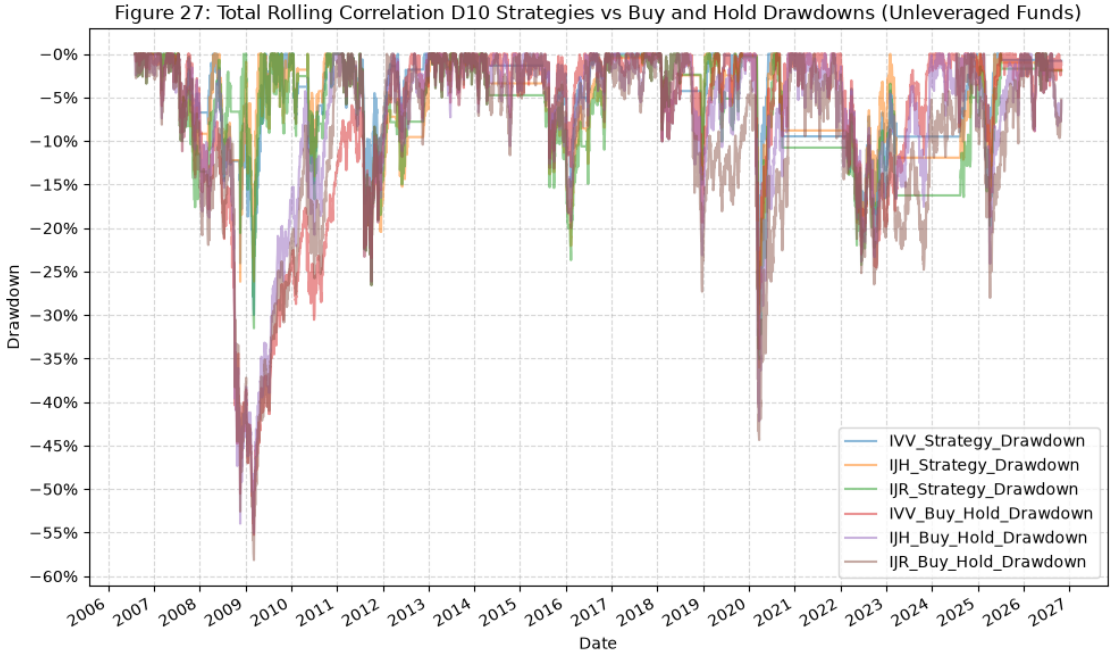
Data Source: Yahoo Finance, Analysis by the Author

[24]:
```
# Drawdowns for each strategy and buy and hold, measured from the running peak␣
 ↪of each equity curve
for ticker in strategy_tickers:
    strategy_equity = (1 + strategy_df[f"{ticker}_Strategy_Return"]).cumprod()
    buy_hold_equity = (1 + strategy_df[f"{ticker}_Daily_Return"]).cumprod()

    strategy_df[f"{ticker}_Strategy_Drawdown"] = strategy_equity /␣
 ↪strategy_equity.cummax() - 1
    strategy_df[f"{ticker}_Buy_Hold_Drawdown"] = buy_hold_equity /␣
 ↪buy_hold_equity.cummax() - 1

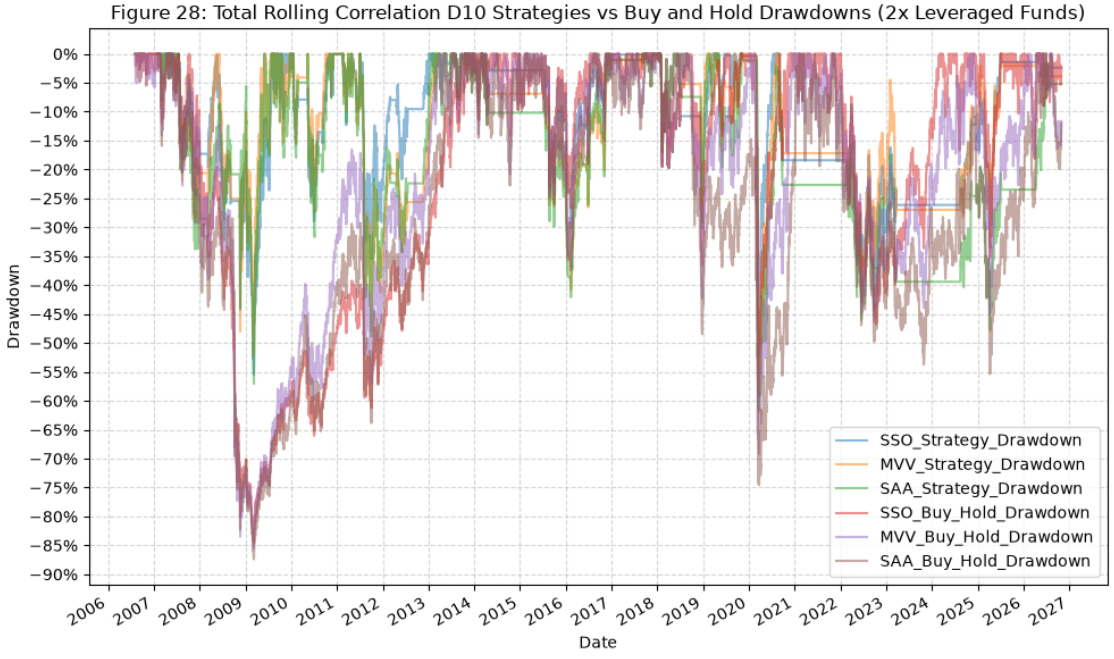
for fund_group, group_tickers in {"Unleveraged": tickers, "2x Leveraged":␣
 ↪leveraged_tickers}.items():
    plot_time_series(
        df=strategy_df,
        plot_start_date=None,
        plot_end_date=None,
        plot_columns=[f"{ticker}_Strategy_Drawdown" for ticker in group_tickers]
        + [f"{ticker}_Buy_Hold_Drawdown" for ticker in group_tickers],
        title=f"Figure {next(figure_number)}: Total Rolling Correlation␣
 ↪{signal_decile} Strategies vs Buy and Hold Drawdowns ({fund_group} Funds)",
        x_label="Date",
        x_format="Year",
        x_tick_spacing=1,
        x_tick_start=None,
        x_tick_rotation=30,
        y_label="Drawdown",
        y_format="Percentage",
        y_format_decimal_places=0,
        y_tick_spacing="Auto",
        y_tick_rotation=0,
        grid=True,
        legend=True,
        export_plot=False,
        plot_file_name=None,
        caption=f"Data Source: Yahoo Finance, Analysis by the Author",
    )
```

Figure 27: Total Rolling Correlation D10 Strategies vs Buy and Hold Drawdowns (Unleveraged Funds)



Data Source: Yahoo Finance, Analysis by the Author

Figure 28: Total Rolling Correlation D10 Strategies vs Buy and Hold Drawdowns (2x Leveraged Funds)



Data Source: Yahoo Finance, Analysis by the Author

Starting with the unleveraged funds, we end up with 44 trades over the 20 year period, with a win rate of 68% for IVV and IJH, and 66% for IJR, not bad overall. Since the signal is shared, all three

strategies spend just under 55% of their time in the market, which really gives us an opportunity to use our capital the other 45% of the time, either in a risk free asset or another strategy altogether.

For IVV, with a CAGR of 8%, we underperform the buy-and-hold strategy over the same period, but we also experience lower volatility (15% vs 19%), and much lower max drawdowns (30% vs 55%). Our strategy gives us a MAR ratio of 0.28, vs 0.20 for the buy-and-hold strategy.

IJH and IJR tell a similar story, but the gap in CAGR is much smaller:

- IJH: a CAGR of 9.1% vs 9.8% for buy-and-hold, with lower volatility (18% vs 22%), lower max drawdowns (36% vs 55%), and a MAR ratio of 0.25 vs 0.18.
- IJR: a CAGR of 8.9% vs 9.4% for buy-and-hold, with lower volatility (19% vs 24%), lower max drawdowns (35% vs 58%), and a MAR ratio of 0.25 vs 0.16.

For IJH and IJR, we give up less than 1% of CAGR per year in exchange for the reduced risk, and both strategies also have a higher Sharpe ratio than their buy-and-hold counterparts, which is not the case for IVV.

The difference in CAGR does add up over the long term, the effect of which we can see in the cumulative return plot, particularly for IVV.

The drawdown plot further illustrates the reduced risk of the strategies compared to the buy-and-hold approach, highlighting the lower maximum drawdowns experienced over the same period. During the GFC, the IVV strategy experienced its max drawdown of 30%, which was significantly lower than that of the buy-and-hold strategy at 55%. Interestingly, the max drawdowns for the IJH and IJR strategies came during the COVID crash in early 2020 rather than the GFC, while all three buy-and-hold strategies saw their max drawdowns during the GFC.

Moving on to the 2x leveraged funds, the win rates are lower (64% for SSO and MVV, and 60% for SAA over its 43 trades), but the results relative to buy-and-hold are more interesting:

- SSO: a CAGR of 12.8% vs 15.6% for buy-and-hold, with lower volatility (31% vs 38%), lower max drawdowns (55% vs 85%), and a MAR ratio of 0.23 vs 0.18.
- MVV: a CAGR of 13.2% vs 11.0% for buy-and-hold, with lower volatility (36% vs 44%), lower max drawdowns (62% vs 86%), and a MAR ratio of 0.21 vs 0.13.
- SAA: a CAGR of 11.6% vs 8.9% for buy-and-hold, with lower volatility (38% vs 48%), lower max drawdowns (61% vs 87%), and a MAR ratio of 0.19 vs 0.10.

For MVV and SAA, the strategy beats buy-and-hold on CAGR, despite being out of the market about 45% of the time. As with IVV, SSO gives up CAGR and has a slightly lower Sharpe ratio than buy-and-hold, while MVV and SAA have higher Sharpe ratios. Even with the strategy, the leveraged funds still see max drawdowns of more than 50%, so the reduced risk is relative. The max drawdown for the SSO strategy came during the GFC, and for the MVV and SAA strategies it came during the COVID crash, matching the pattern we see with the unleveraged funds.

0.10 Conclusion

Overall, we delved into the different deciles of a "total correlation" metric, assembled from the rolling correlations of 3 well known US equity ETFs. This analysis, including comparison to a baseline, gave us insight into where the total correlations could be used to improve our ability to time market entries and exits. Finally, we built a simple trading strategy using a single total correlation window and decile and a single future return window, as a proof of concept for how these

insights could be applied in practice. This strategy, while underperforming buy-and-hold over the long-term for IVV, IJH, IJR, and SSO, did have lower volatility and drawdowns for all six funds, and outperformed buy-and-hold for MVV and SAA, demonstrating the potential risk management benefits of incorporating total correlation information into trading decisions.

0.11 Future Investigation

None

0.12 References

None