variable_datasets

January 31, 2026

0.1 Dependent and Independent Variables

This code creates the dependent and independent datasets for our project.

[41]:
```
#Go to repo
#%cd /home/jupyter-toomeyck/HelpHerInvest
```

[42]:
```
#Sync latest from GitHub before editing
#!git pull --rebase origin main
```

[43]:
```
# Imports
import time
import requests
import pandas as pd
#%pip install yfinance --quiet
import yfinance as yf
from pathlib import Path
import numpy as np
import warnings
import zipfile
from pathlib import Path

warnings.filterwarnings("ignore")
```

[ ]:
```
import os
repo_url = "https://github.com/tongyuguo/HelpHerInvest.git"
repo_dir = "HelpHerInvest"
if not os.path.exists(repo_dir):
    !git clone {repo_url}
else:
    !git -C {repo_dir} pull

# list available data files
!ls -lah {repo_dir}/Data
```

```
Already up to date.
total 9.5M
drwxr-xr-x 2 jupyter-toomeyck jupyter-toomeyck   68 Jan 31 00:24 .
drwxr-xr-x 8 jupyter-toomeyck jupyter-toomeyck 4.0K Jan 31 00:24 ..
```

```
-rw-r--r-- 1 jupyter-toomeyck jupyter-toomeyck 9.2M Jan 31 00:24
stock_symbols_new.csv.zip
-rw-r--r-- 1 jupyter-toomeyck jupyter-toomeyck 296K Jan 31 00:24
testing_small.csv.zip
```

1 X Variables (Independent) Dataset

[45]:
```python
# X-Variables #
DATA_DIR = Path(repo_dir) / "Data"
## Create Path to Dataset ##
dataset_path = f"{repo_dir}/Data/stock_symbols_new.csv.zip"
csv_file_name = "stock_symbols_new.csv"

# Open the zip file and then open the specific CSV file within it
with zipfile.ZipFile(dataset_path, 'r') as zf:
    with zf.open(csv_file_name) as file_handle:
        # Read the file-like object directly into pandas
        symbols = pd.read_csv(file_handle)

## Reading Symbols from Dataset ##
tickers = symbols['symbol'].tolist()
tickers.remove("SPY")
tickers = tickers[:50]  # limit to 50 for testing
benchmark = 'SPY'
all_tickers = tickers + [benchmark]

## Download Price Data ##
prices = yf.download(
    all_tickers,
    start='2010-01-01',
    auto_adjust=False,
    progress=False
)['Adj Close']

prices = prices.dropna(how='all')

print(prices.head(10))

## Compute Features ##
monthly_px = prices.resample('ME').last()  # month-end prices
mom_1m  = monthly_px / monthly_px.shift(1)  - 1  # 1-month momentum
mom_3m  = monthly_px / monthly_px.shift(3)  - 1  # 3-month momentum
mom_6m  = monthly_px / monthly_px.shift(6)  - 1  # 6-month momentum
mom_12m = monthly_px / monthly_px.shift(12) - 1  # 12-month momentum
mom_12m_ex_1m = (monthly_px.shift(1) / monthly_px.shift(12)) - 1  # 12-month␣
 ↪momentum excluding most recent month
```

```
rel_3m_spy  = mom_3m.sub(mom_3m["SPY"], axis=0)  # relative strength against␣
  ↪S&P 3-month
rel_6m_spy  = mom_6m.sub(mom_6m["SPY"], axis=0)  # relative strength against␣
  ↪S&P 6-month
rel_12m_spy = mom_12m.sub(mom_12m["SPY"], axis=0) # relative strength against␣
  ↪S&P 12-month

daily_ret = prices.pct_change() # daily returns

vol_3m = (daily_ret.rolling(63).std() * np.sqrt(252)).resample("M").last() #␣
  ↪3-month volatility
vol_6m = (daily_ret.rolling(126).std() * np.sqrt(252)).resample("M").last() #␣
  ↪6-month volatility


roll_max_6m  = monthly_px.rolling(6).max()  # 6-month rolling max
roll_max_12m = monthly_px.rolling(12).max() # 12-month rolling max

drawdown_6m  = monthly_px / roll_max_6m  - 1  # 6-month drawdown
drawdown_12m = monthly_px / roll_max_12m - 1  # 12-month drawdown

dma_200 = prices.rolling(200).mean().resample("M").last() # 200-day moving␣
  ↪average
pct_above_200dma = monthly_px / dma_200 - 1  # pct above 200-day moving average


## Combine Features ##
X = pd.concat(
    {
        "mom_1m": mom_1m[tickers],
        "mom_3m": mom_3m[tickers],
        "mom_6m": mom_6m[tickers],
        "mom_12m": mom_12m[tickers],
        "mom_12m_ex_1m": mom_12m_ex_1m[tickers],
        "rel_3m_spy": rel_3m_spy[tickers],
        "rel_6m_spy": rel_6m_spy[tickers],
        "rel_12m_spy": rel_12m_spy[tickers],
        "vol_3m": vol_3m[tickers],
        "vol_6m": vol_6m[tickers],
        #"downside_vol_6m": downside_vol_6m[tickers],
        "drawdown_6m": drawdown_6m[tickers],
        "drawdown_12m": drawdown_12m[tickers],
        "pct_above_200dma": pct_above_200dma[tickers],
    },
    axis=1
)


## Standardize Data Function - z score ##
def zscore_cs(row: pd.Series) -> pd.Series:
    # row contains values across tickers for a single feature at a single date
    mu = row.mean()
    sd = row.std(ddof=0)
    if sd == 0 or np.isnan(sd):
        return row * 0.0
    return (row - mu) / sd # calcs z-score


## Normalize per feature across tickers at each date ##
X_z = X.copy()
#print("X_z before normalization:")
#print(X_z.head(20))

for feat in X.columns.get_level_values(0).unique():
    X_z[feat] = X[feat].apply(zscore_cs, axis=1)

#print("X_z after normalization:")
#print(X_z.head(20))

## Flatten X_z table so each  ticker is a row ##
X_panel = (
    X_z.stack(level=1)              # index becomes (Date, Ticker)
      .rename_axis(index=["Date","Ticker"])
      .reset_index()
)


#print(X_panel.head())
print("-----  -----  -----")
print(X_panel.tail())
print("-----  -----  -----")

print("Size of dataset:",
"Rows:",X_panel.shape[0],
"Columns:",X_panel.shape[1])


output = DATA_DIR / "dependent_variables.csv"
X_panel.to_csv(output, index=False)
print(f"Output file saved to: {output}")
```

```
Ticker          AAPL  ABBV   AMD    AMZN       ASML      AVGO         AXP  \
Date
```

```
2010-01-04  6.418383   NaN  9.70  6.6950  32.425728  1.328563  32.483322
2010-01-05  6.429481   NaN  9.71  6.7345  32.678314  1.338425  32.411892
2010-01-06  6.327210   NaN  9.57  6.6125  32.977699  1.348991  32.935795
2010-01-07  6.315514   NaN  9.47  6.5000  32.060875  1.340538  33.469971
2010-01-08  6.357502   NaN  9.43  6.6760  31.293720  1.350401  33.446064
2010-01-11  6.301420   NaN  9.14  6.5155  30.629494  1.358853  33.063351
2010-01-12  6.229738   NaN  8.65  6.3675  30.676279  1.338425  33.501862
2010-01-13  6.317612   NaN  9.15  6.4555  31.527605  1.297568  33.605526
2010-01-14  6.281024   NaN  9.00  6.3675  31.312433  1.298272  34.028069
2010-01-15  6.176055   NaN  8.84  6.3570  30.620132  1.284888  33.796867

Ticker            AZN  BABA        BAC  …        RTX        SAP        SPY  \
Date                                    …
2010-01-04  12.797744   NaN  12.169184  …  30.905077  36.360821  85.027954
2010-01-05  12.538607   NaN  12.564739  …  30.443419  36.136993  85.253014
2010-01-06  12.438726   NaN  12.712106  …  30.283791  36.978291  85.313080
2010-01-07  12.573697   NaN  13.130935  …  30.413216  37.950787  85.673203
2010-01-08  12.627685   NaN  13.014595  …  30.473629  38.236374  85.958298
2010-01-11  12.854429   NaN  13.130935  …  31.133776  39.085373  86.078323
2010-01-12  12.795050   NaN  12.688840  …  30.952539  38.244087  85.275551
2010-01-13  13.094677   NaN  12.890500  …  31.306345  38.892414  85.995804
2010-01-14  13.332224   NaN  13.045622  …  31.414202  38.714893  86.228394
2010-01-15  13.186459   NaN  12.611284  …  31.060413  37.780994  85.260536

Ticker             TM  TSLA        UNH          V        WFC        WMT  \
Date
2010-01-04  57.091167   NaN  24.597445  19.645657  18.025999  12.960374
2010-01-05  56.212109   NaN  24.558432  19.420536  18.520853  12.831315
2010-01-06  56.930119   NaN  24.800282  19.159750  18.547249  12.802631
2010-01-07  56.225548   NaN  25.752035  19.338068  19.220251  12.809811
2010-01-08  57.547474   NaN  25.510201  19.391567  19.042110  12.745275
2010-01-11  57.842724   NaN  25.681814  19.335835  19.002522  12.955583
2010-01-12  60.057102   NaN  25.003107  19.244448  18.527456  13.079862
2010-01-13  59.849106   NaN  25.385376  19.422766  18.857365  13.146778
2010-01-14  61.016705   NaN  25.993872  19.489624  19.127882  12.955583
2010-01-15  61.157600   NaN  26.329332  19.199867  18.527456  12.828929

Ticker            XOM
Date
2010-01-04  37.881809
2010-01-05  38.029697
2010-01-06  38.358418
2010-01-07  38.237873
2010-01-08  38.084488
2010-01-11  38.511795
2010-01-12  38.320049
2010-01-13  38.166668
2010-01-14  38.172142
```

```
2010-01-15  37.859886

[10 rows x 51 columns]
-----  -----  -----
           Date Ticker    mom_1m    mom_3m    mom_6m   mom_12m  mom_12m_ex_1m  \
9156 2026-01-31    NVS  0.302008  0.683322  0.092443  0.170972       0.187299
9157 2026-01-31    AXP -0.700591 -0.434786 -0.172078 -0.408221      -0.280782
9158 2026-01-31    NVO  1.024695  0.685888  0.028152 -1.064248      -1.648084
9159 2026-01-31     PM  0.626897  0.952194 -0.316095  0.095610      -0.034148
9160 2026-01-31    RTX  0.440492  0.326942  0.047290  0.359920       0.391959

      rel_3m_spy  rel_6m_spy  rel_12m_spy    vol_3m    vol_6m  drawdown_6m  \
9156    0.683322    0.092443     0.170972 -0.844804 -0.761584     0.624286
9157   -0.434786   -0.172078    -0.408221 -0.484395 -0.409385     0.115909
9158    0.685888    0.028152    -1.064248  1.286949  0.930018     0.624286
9159    0.952194   -0.316095     0.095610 -0.583738 -0.528087     0.624286
9160    0.326942    0.047290     0.359920 -0.746751 -0.589330     0.624286

      drawdown_12m  pct_above_200dma
9156      0.650648          0.138737
9157      0.258516         -0.280700
9158     -2.134469         -0.527479
9159      0.650648         -0.205271
9160      0.650648          0.424437
-----  -----  -----
Size of dataset: Rows: 9161 Columns: 15
Output file saved to: HelpHerInvest/Data/dependent_variables.csv
```

1.1 Y Variable (Dependent) Dataset

[ ]:
```python
# Y-Variable #

DATA_DIR = Path(repo_dir) / "Data"
## Create Path to Dataset ##
dataset_path = f"{repo_dir}/Data/stock_symbols_new.csv.zip"
csv_file_name = "stock_symbols_new.csv"

# Open the zip file and then open the specific CSV file within it
with zipfile.ZipFile(dataset_path, 'r') as zf:
    with zf.open(csv_file_name) as file_handle:
        # Read the file-like object directly into pandas
        symbols = pd.read_csv(file_handle)

## Reading Symbols from Dataset ##
tickers = symbols['symbol'].tolist()
tickers.remove("SPY")
tickers = tickers[:50]  # limit to 50 for testing
```

```python
benchmark = 'SPY'
all_tickers = tickers + [benchmark]

def forward_excess_return_monthly(tickers, benchmark="SPY", start="2010-01-01",␣
 ↪end=None, horizon_months=3):

    universe = list(dict.fromkeys(list(tickers) + [benchmark]))

    px_daily = yf.download(
        universe, start=start, end=end, auto_adjust=False, progress=False
    )["Adj Close"]

    px_m = px_daily.resample("M").last().dropna(subset=[benchmark])

    fwd_ret = px_m.shift(-horizon_months) / px_m - 1.0
    bench_fwd = fwd_ret[benchmark].rename("bench_fwd_return")

    # Wide to long in one go
    out = pd.DataFrame(index=px_m.index)
    out["bench_fwd_return"] = bench_fwd

    for t in tickers:
        out[f"adj_close_{t}"] = px_m[t]
        out[f"fwd_return_{t}"] = fwd_ret[t]
        out[f"fwd_excess_{t}"] = fwd_ret[t] - bench_fwd

    return out

df_y_m = forward_excess_return_monthly(all_tickers, benchmark="SPY",␣
 ↪start="2010-01-01", horizon_months=6)
print(df_y_m.head(10))
print(df_y_m.tail(10))

bench_df = (
    df_y_m[["bench_fwd_return"]]
    .rename(columns={"bench_fwd_return": "bench_fwd_return"})
    .reset_index()
)

# Keep only stock-level columns
stock_cols = [c for c in df_y_m.columns if "_" in c and not c.
 ↪startswith("fwd_ret_bench")]

y_long = (
    df_y_m[stock_cols]
    .reset_index()
    .melt(id_vars="Date", var_name="metric_ticker", value_name="value")
)

# Split "adj_close_AAPL" -> metric="adj_close", ticker="AAPL"
y_long[["metric", "ticker"]] = y_long["metric_ticker"].str.rsplit("_", n=1,␣
 ↪expand=True)

y_long = y_long.drop(columns="metric_ticker")

df_y_m_output = (
    y_long
    .pivot(index=["Date", "ticker"], columns="metric", values="value")
    .reset_index()
)

df_y_m_output = df_y_m_output[["Date", "ticker", "adj_close", "fwd_excess",␣
 ↪"fwd_return"]]
df_y_m_output = df_y_m_output[df_y_m_output["ticker"] != "return"]

print(df_y_m_output.head(10))
print(df_y_m_output.tail(10))

print("Size of dataset:",
"Rows:",df_y_m_output.shape[0],
"Columns:",df_y_m_output.shape[1])

# Save to CSV
y_output = DATA_DIR / "independent_variables.csv"
df_y_m_output.to_csv(y_output, index=False)
print(f"Output file saved to: {y_output}")
```

```
            bench_fwd_return  adj_close_NVDA  fwd_return_NVDA  \
Date
2010-01-31          0.035952        0.352752        -0.402859
2010-02-28         -0.040575        0.371318        -0.424074
2010-03-31         -0.014641        0.398823        -0.328735
2010-04-30          0.007416        0.360087        -0.234882
2010-05-31          0.094369        0.301180         0.035768
2010-06-30          0.231236        0.234022         0.508325
2010-07-31          0.179371        0.210643         1.602829
2010-08-31          0.277816        0.213852         1.428724
2010-09-30          0.172928        0.267716         0.580479
2010-10-31          0.162488        0.275509         0.663894

            fwd_excess_NVDA  adj_close_GOOGL  fwd_return_GOOGL  \
Date
2010-01-31        -0.438811        13.162311         -0.085085
2010-02-28        -0.383499        13.084322         -0.145748
```

```
2010-03-31        -0.314094        14.085764         -0.072877
2010-04-30        -0.242298        13.057001          0.167396
2010-05-31        -0.058600        12.061768          0.144307
2010-06-30         0.277089        11.051385          0.334914
2010-07-31         1.423458        12.042394          0.238239
2010-08-31         1.150908        11.177309          0.363051
2010-09-30         0.407551        13.059237          0.115959
2010-10-31         0.501405        15.242688         -0.113410

            fwd_excess_GOOGL  adj_close_AAPL  fwd_return_AAPL  \
Date
2010-01-31         -0.121037        5.760079         0.339425
2010-02-28         -0.105173        6.136767         0.188056
2010-03-31         -0.058236        7.047895         0.207446
2010-04-30          0.159980        7.830363         0.152783
2010-05-31          0.049939        7.704100         0.211266
2010-06-30          0.103678        7.543651         0.282391
2010-07-31          0.058868        7.715196         0.319028
2010-08-31          0.085235        7.290825         0.452941
2010-09-30         -0.056969        8.509956         0.228229
2010-10-31         -0.275899        9.026706         0.163300

            fwd_excess_AAPL  …  fwd_excess_NVO  adj_close_PM  fwd_return_PM  \
Date                         …
2010-01-31         0.303473  …        0.262986     21.525187       0.148335
2010-02-28         0.228631  …        0.266249     23.166422       0.073670
2010-03-31         0.222088  …        0.291094     24.944031       0.100195
2010-04-30         0.145366  …        0.269076     23.471107       0.221209
2010-05-31         0.116897  …        0.199825     21.099121       0.320885
2010-06-30         0.051155  …        0.158174     22.199821       0.305611
2010-07-31         0.139657  …        0.135635     24.718128       0.146749
2010-08-31         0.175125  …        0.201511     24.873098       0.249902
2010-09-30         0.055301  …        0.118159     27.443302       0.196433
2010-10-31         0.000812  …        0.071164     28.663118       0.212017

            fwd_excess_PM  adj_close_RTX  fwd_return_RTX  fwd_excess_RTX  \
Date
2010-01-31       0.112383      29.114555        0.066639        0.030687
2010-02-28       0.114245      29.810360       -0.038763        0.001812
2010-03-31       0.114837      31.964167       -0.020774       -0.006132
2010-04-30       0.213793      32.546055        0.009515        0.002099
2010-05-31       0.226516      29.429911        0.130349        0.035980
2010-06-30       0.074375      28.351074        0.227143       -0.004093
2010-07-31      -0.032622      31.054720        0.157025       -0.022347
2010-08-31      -0.027914      28.654823        0.294956        0.017140
2010-09-30       0.023505      31.300150        0.201265        0.028337
2010-10-31       0.049529      32.855721        0.211040        0.048551
```

```
           adj_close_SPY  fwd_return_SPY  fwd_excess_SPY
Date
2010-01-31      80.571365        0.035952             0.0
2010-02-28      83.084755       -0.040575             0.0
2010-03-31      88.142929       -0.014641             0.0
2010-04-30      89.506523        0.007416             0.0
2010-05-31      82.394806        0.094369             0.0
2010-06-30      78.131592        0.231236             0.0
2010-07-31      83.468063        0.179371             0.0
2010-08-31      79.713631        0.277816             0.0
2010-09-30      86.852394        0.172928             0.0
2010-10-31      90.170311        0.162488             0.0

[10 rows x 154 columns]
           bench_fwd_return  adj_close_NVDA  fwd_return_NVDA  \
Date
2025-04-30          0.237012      108.900230         0.859305
2025-05-31          0.166139      135.105484         0.310014
2025-06-30          0.110029      157.972305         0.180587
2025-07-31          0.101030      177.850067         0.074669
2025-08-31               NaN      174.160477              NaN
2025-09-30               NaN      186.569611              NaN
2025-10-31               NaN      202.478729              NaN
2025-11-30               NaN      176.990143              NaN
2025-12-31               NaN      186.500000              NaN
2026-01-31               NaN      191.130005              NaN

           fwd_excess_NVDA  adj_close_GOOGL  fwd_return_GOOGL  \
Date
2025-04-30         0.622292       158.362671          0.774447
2025-05-31         0.143876       171.267044          0.868256
2025-06-30         0.070558       175.957413          0.778840
2025-07-31        -0.026360       191.603180          0.764063
2025-08-31              NaN       212.580704               NaN
2025-09-30              NaN       242.941101               NaN
2025-10-31              NaN       281.006195               NaN
2025-11-30              NaN       319.970703               NaN
2025-12-31              NaN       313.000000               NaN
2026-01-31              NaN       338.000000               NaN

           fwd_excess_GOOGL  adj_close_AAPL  fwd_return_AAPL  \
Date
2025-04-30          0.537435      211.775833         0.275444
2025-05-31          0.702117      200.428024         0.391273
2025-06-30          0.668811      204.738937         0.327837
2025-07-31          0.663033      207.133911         0.252716
2025-08-31               NaN      231.915176              NaN
2025-09-30               NaN      254.383408              NaN
```

```
2025-10-31                NaN      270.108154             NaN
2025-11-30                NaN      278.850006             NaN
2025-12-31                NaN      271.859985             NaN
2026-01-31                NaN      259.480011             NaN

            fwd_excess_AAPL  …  fwd_excess_NVO  adj_close_PM  fwd_return_PM  \
Date                         …
2025-04-30         0.038431  …       -0.484306    166.973267      -0.143423
2025-05-31         0.225134  …       -0.468151    175.966995      -0.113148
2025-06-30         0.217809  …       -0.364544    178.803696      -0.102927
2025-07-31         0.151687  …        0.175786    161.053909       0.114161
2025-08-31              NaN  …             NaN    164.077652            NaN
2025-09-30              NaN  …             NaN    159.237671            NaN
2025-10-31              NaN  …             NaN    143.025497            NaN
2025-11-30              NaN  …             NaN    156.056625            NaN
2025-12-31              NaN  …             NaN    160.399994            NaN
2026-01-31              NaN  …             NaN    179.440002            NaN

            fwd_excess_PM  adj_close_RTX  fwd_return_RTX  fwd_excess_RTX  \
Date
2025-04-30      -0.380435     124.448151        0.428686        0.191673
2025-05-31      -0.279287     135.346619        0.292312        0.126173
2025-06-30      -0.212955     144.807404        0.266510        0.156481
2025-07-31       0.013132     156.261490        0.285857        0.184828
2025-08-31            NaN     157.975632             NaN             NaN
2025-09-30            NaN     166.671265             NaN             NaN
2025-10-31            NaN     177.797287             NaN             NaN
2025-11-30            NaN     174.910004             NaN             NaN
2025-12-31            NaN     183.399994             NaN             NaN
2026-01-31            NaN     200.929993             NaN             NaN

            adj_close_SPY  fwd_return_SPY  fwd_excess_SPY
Date
2025-04-30     549.752380        0.237012             0.0
2025-05-31     584.301453        0.166139             0.0
2025-06-30     614.326538        0.110029             0.0
2025-07-31     628.475403        0.101030             0.0
2025-08-31     641.371399             NaN             NaN
2025-09-30     664.217285             NaN             NaN
2025-10-31     680.050537             NaN             NaN
2025-11-30     681.376587             NaN             NaN
2025-12-31     681.919983             NaN             NaN
2026-01-31     691.969971             NaN             NaN

[10 rows x 154 columns]
metric       Date ticker  adj_close  fwd_excess  fwd_return
0      2010-01-31   AAPL   5.760079    0.303473    0.339425
1      2010-01-31   ABBV        NaN         NaN         NaN
```

```
2      2010-01-31    AMD   7.460000   -0.031931    0.004021
3      2010-01-31   AMZN   6.270500   -0.095915   -0.059963
4      2010-01-31   ASML  29.235538    0.002494    0.038446
5      2010-01-31   AVGO   1.224306    0.216062    0.252014
6      2010-01-31    AXP  30.025711    0.159926    0.195878
7      2010-01-31    AZN  12.549404    0.090095    0.126047
8      2010-01-31   BABA        NaN         NaN         NaN
9      2010-01-31    BAC  11.773627   -0.109888   -0.073936
metric       Date ticker   adj_close  fwd_excess  fwd_return
10025  2026-01-31    RTX  200.929993         NaN         NaN
10026  2026-01-31    SAP  201.039993         NaN         NaN
10027  2026-01-31    SPY  691.969971         NaN         NaN
10028  2026-01-31     TM  226.860001         NaN         NaN
10029  2026-01-31   TSLA  430.410004         NaN         NaN
10030  2026-01-31    UNH  286.929993         NaN         NaN
10031  2026-01-31      V  321.829987         NaN         NaN
10032  2026-01-31    WFC   90.489998         NaN         NaN
10033  2026-01-31    WMT  119.139999         NaN         NaN
10034  2026-01-31    XOM  141.399994         NaN         NaN
Size of dataset: Rows: 9843 Columns: 5
```