

Drawdown Risk: Measurement, Prediction, and Mitigation

Shane Conway

Table of contents

1	Executive Summary	4
2	Definitions and Measurement	6
2.1	Why definitions matter	6
2.2	Key metrics & formulas	6
2.2.1	Maximum Drawdown (MDD)	6
2.2.2	Average Drawdown (AvgDD)	6
2.2.3	Time Under Water (TUW)	6
2.2.4	Ulcer Index (UI)	7
2.2.5	Recovery Time	7
2.3	SPY example (downloaded with yfinance)	7
2.4	Practical notes	10
3	Literature Review	11
3.1	Drawdown mechanics and stylized models	11
3.2	Optimization under drawdown constraints	11
3.3	Risk measures beyond variance	11
3.4	Probability of drawdown and feedback control	12
3.5	Predicting large drawdowns: bubbles and criticality	12
3.6	When to start worrying during a drawdown	12
3.7	Stop-loss rules in theory and practice	12
3.8	The triple penance and behavioral reality of stops	13
3.9	Stops, momentum, and reflexivity	13
3.10	Correlation, contagion, and systemic warnings	13
3.11	From warnings to action: hedging and overlays	13
4	Simulation Framework	14
5	Simulation Framework	15
5.1	AR(1) process with trailing stop-loss	15
5.2	Implementation	16
5.3	Design choices	18
6	Empirical Measurement	19
7	Early Warning Signals	26

8	Stop-Loss, Momentum, and Hedging	29
8.1	Demos	29
8.2	Data-Driven Hedge Example (VIX proxy)	31
9	Results Synthesis	35
10	Appendices	36

1 Executive Summary

Drawdowns — sustained declines from prior peaks — are a central measure of investment risk. They capture not only the size of losses but also their duration and the psychological and liquidity pressures they impose on market participants. While volatility remains the industry’s standard risk metric, drawdowns often determine *actual* investor behavior, influencing capital withdrawals, mandate breaches, and risk appetite.

This study examines drawdown risk through three complementary lenses:

1. **Mechanics and measurement** — formal definitions of maximum drawdown (MDD), average drawdown, and time under water, along with their statistical properties under different return-generating processes.
2. **Prediction and early warnings** — the potential to forecast elevated drawdown risk using tools from both quantitative finance (e.g., correlation spikes, regime-switching probabilities, LPPL bubble detection) and market theory (e.g., Soros’s reflexivity, Mandelbrot’s multifractal markets).
3. **Mitigation strategies** — evaluating the performance of stop-loss rules, momentum-based overlays, and explicit hedges such as VIX futures and protective options.

A central theme is the **connection between stop-loss rules and momentum trading**. A trailing stop that re-enters only above the prior peak inherently captures part of a time-series momentum strategy: it cuts risk when recent returns are negative and adds exposure when trend resumes. This link suggests that the efficacy of stop-losses will vary with the degree of return autocorrelation and the persistence of regimes — a hypothesis we test both in simulations and on historical data.

Stop-losses are not the only path to drawdown control. We compare them with *explicit hedging* strategies, including: - **Volatility hedges**: buying VIX futures or calls when early-warning indicators trigger. - **Tail hedges**: purchasing out-of-the-money index puts or put spreads to cap downside. - **Dynamic allocation shifts**: moving capital to low-volatility or uncorrelated assets in high-risk regimes.

Our empirical analysis spans the S&P 500, US 10-year Treasury notes, and Bitcoin — assets with distinct volatility, correlation, and drawdown profiles. Simulation studies use AR(1), regime-switching, and stochastic volatility models to explore a wide parameter space and quantify expected performance differences across strategies.

Key takeaways (preview): - **Stop-loss effectiveness is conditional:** they tend to improve risk-adjusted returns when returns exhibit positive serial correlation and regimes persist, but can hurt performance in mean-reverting markets. - **Hedges offer complementary protection:** while they often cost carry in normal markets, they can sharply reduce maximum drawdown and time under water when timed with credible early-warning signals. - **Integrated approaches dominate:** blending signal-driven stop-loss logic with targeted hedging yields better drawdown control per unit of performance drag than either technique alone.

This report proceeds from measurement to prediction to mitigation, grounding each step in both simulation and real-world data. By the conclusion, we present a framework for active drawdown management that integrates statistical signals, behavioral insights, and cost-aware strategy design.

2 Definitions and Measurement

2.1 Why definitions matter

Consistent drawdown metrics allow fair comparison across strategies and time periods. While **Maximum Drawdown (MDD)** is common, it captures only the single worst episode. Complementary measures like **Average Drawdown (AvgDD)**, **Time Under Water (TUW)**, **Ulcer Index (UI)**, and **Recovery Time** help characterize the *shape* and *persistence* of losses.

2.2 Key metrics & formulas

2.2.1 Maximum Drawdown (MDD)

Definition: largest peak-to-trough loss.

$$\text{MDD} = \min_t \left(\frac{P_t}{\max_{\tau \leq t} P_\tau} - 1 \right)$$

2.2.2 Average Drawdown (AvgDD)

Definition: mean depth across all drawdowns.

$$\text{AvgDD} = \frac{1}{N_{\text{dd}}} \sum_{i=1}^{N_{\text{dd}}} \left(\frac{P_{\text{trough},i}}{P_{\text{peak},i}} - 1 \right)$$

2.2.3 Time Under Water (TUW)

Definition: average duration from a peak to full recovery.

$$\text{TUW} = \frac{1}{N_{\text{dd}}} \sum_{i=1}^{N_{\text{dd}}} (t_{\text{rec},i} - t_{\text{peak},i})$$

2.2.4 Ulcer Index (UI)

Definition: RMS of percentage drawdowns (emphasizes persistent pain).

$$UI = \sqrt{\frac{1}{T} \sum_{t=1}^T \left(100 \times \min \left(0, \frac{P_t}{\max_{\tau \leq t} P_{\tau}} - 1 \right) \right)^2}$$

2.2.5 Recovery Time

Definition: longest time from a peak to its recovery.

$$\max_i (t_{\text{rec},i} - t_{\text{peak},i})$$

2.3 SPY example (downloaded with yfinance)

```
import pandas as pd, numpy as np, matplotlib.pyplot as plt
import yfinance as yf
from datetime import datetime

spy = yf.download("SPY", start="2000-01-01", auto_adjust=True, progress=False)["Close"].squeeze()
rets = spy.pct_change().dropna()
wealth = (1+rets).cumprod()
dd = wealth/wealth.cummax() - 1

# Identify drawdown episodes for AvgDD and TUW
peaks = wealth.cummax()
in_dd = dd < 0
# Episode IDs increment when leaving water and re-entering
episode_id = ((~in_dd).astype(int).diff() == -1).cumsum()
episode_id = episode_id.where(in_dd, np.nan).ffill()

draws = []
if episode_id.notna().any():
    for eid, grp in wealth.groupby(episode_id.dropna()):
        # grp spans underwater segment including starting day after a peak
        depth = (grp/grp.cummax() - 1).min()
        # Recovery time: from peak to first recovery beyond that peak
        start_idx = grp.index[0]
```

```

    peak_level = peaks.loc[start_idx]
    # find recovery
    after = wealth.loc[start_idx:]
    rec_idx = after[after >= peak_level].index.min()
    if pd.isna(rec_idx):
        rec_days = np.nan
    else:
        rec_days = (rec_idx - start_idx).days
    draws.append((depth, rec_days))

avgdd = float(np.nanmean([d for d, _ in draws])) if draws else np.nan
tuw = float(np.nanmean([u for _, u in draws])) if draws else np.nan
recovery_time = float(np.nanmax([u for _, u in draws])) if draws else np.nan

mdd = float(dd.min())
vol = float(rets.std()*np.sqrt(252))
cagr = float((wealth.iloc[-1]/wealth.iloc[0])**((252/len(wealth))-1))
ui = float(np.sqrt((100*np.minimum(0, dd)).pow(2)).mean()))

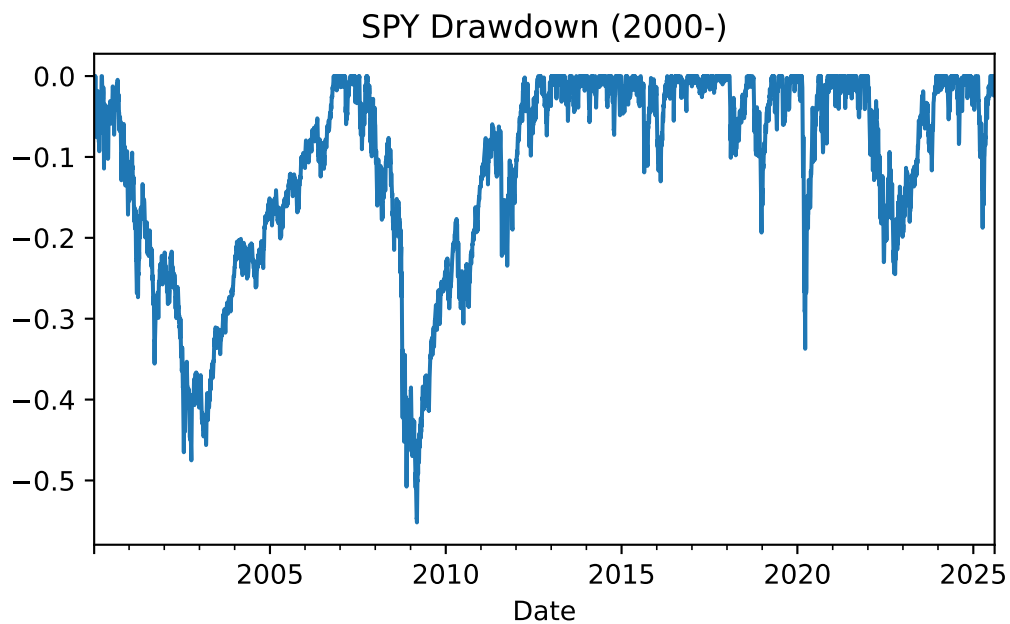
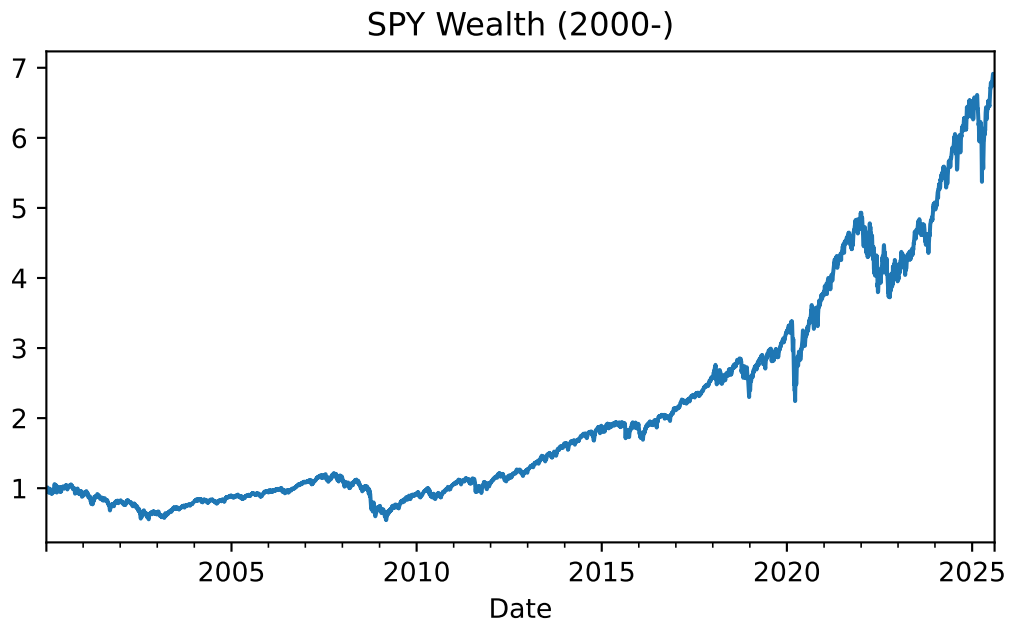
tbl = pd.DataFrame({
    "CAGR": [cagr],
    "Vol": [vol],
    "Sharpe": [cagr/vol if vol else np.nan],
    "MDD": [mdd],
    "AvgDD": [avgdd],
    "TUW_days": [tuw],
    "RecoveryTime_days": [recovery_time],
    "UlcerIndex": [ui]
})
display(tbl.round(4))
import os
os.makedirs("figures", exist_ok=True)
tbl.to_csv("figures/spy_drawdown_metrics.csv", index=False)
print("Saved to figures/spy_drawdown_metrics.csv")

```

	CAGR	Vol	Sharpe	MDD	AvgDD	TUW_days	RecoveryTime_days	UlcerIndex
0	0.0772	0.1914	0.4035	-0.5519	-0.0169	31.0366	2404.0	16.4171

Saved to figures/spy_drawdown_metrics.csv


```
# Quick plots
plt.figure(); wealth.plot(title="SPY Wealth (2000-)"); plt.tight_layout(); plt.show()
plt.figure(); dd.plot(title="SPY Drawdown (2000-)"); plt.tight_layout(); plt.show()
```



2.4 Practical notes

- MDD focuses on a single tail event; AvgDD and UI provide stability for optimization.
- TUW and Recovery Time reflect investor experience and liquidity needs.
- In stop-loss testing, AvgDD and TUW often show larger improvements than MDD.

3 Literature Review

This section surveys theory and evidence on drawdowns, their measurement, and interventions that aim to mitigate or anticipate them. We begin with analytical results for stylized processes, then review optimization under drawdown constraints, alternative risk measures, and the literature on early warnings and mitigation through stop-loss rules, momentum overlays, and hedging.

3.1 Drawdown mechanics and stylized models

Early work characterizes the maximum drawdown (MDD) of continuous-time diffusions and discrete-time random walks. Closed-form or semi-closed-form results exist for Brownian motion and related processes, providing benchmarks for how often and how deeply paths fall below previous peaks purely by chance (Magdon-Ismail et al. 2004; Hadjiliadis and Vecer 2006). These models are invaluable as “nulls” against which more realistic, autocorrelated, or regime-switching processes can be compared.

3.2 Optimization under drawdown constraints

A parallel literature designs policies that explicitly control drawdown rather than variance. Continuous-time portfolio problems with maximum-drawdown constraints yield optimal feedback rules that limit exposure after losses and scale back in after recoveries (Grossman and Zhou 1993). In discrete time, chance-constrained and dynamic-programming formulations treat drawdown as a state variable, often producing asymmetric risk-taking rules.

3.3 Risk measures beyond variance

Several practical measures complement or replace volatility for portfolios whose investors are especially sensitive to underwater periods. Conditional Expected Drawdown (CED) and Conditional Drawdown at Risk (CDaR) generalize VaR/CVaR logic from returns to drawdowns, enabling optimization and attribution frameworks that penalize depth and duration of losses

(Goldberg and Mahmoud 2014; Chekhlov, Uryasev, and Zabaranin 2005). Industry guidance likewise emphasizes “drawdown greeks” for understanding marginal contributions to risk (Harvey and Liu 2020).

3.4 Probability of drawdown and feedback control

Stochastic-control research analyzes the probability of breaching a given drawdown level and derives strategies that minimize it in diffusion settings (Angoshtari, Bayraktar, and Young 2015). In a related spirit, *drawdown-modulated feedback* adaptively scales positions based on current drawdown state, formalizing intuitive “pain-based” sizing rules (Hsieh and Barmish 2017).

3.5 Predicting large drawdowns: bubbles and criticality

The LPPL literature models bubbles as phases of faster-than-exponential growth decorated by log-periodic oscillations, culminating in a critical time t_c when crash risk is elevated (Sornette 2003; Johansen and Sornette 2000). Applications span indices, sectors, and single names, but calibration remains delicate—identifiability issues and overfitting are persistent challenges.

3.6 When to start worrying during a drawdown

Closer to the practitioner’s perspective, Rej, Seager, and Bouchaud (2017) analyze the distribution of drawdown lengths and depths under realistic return processes and propose alarm thresholds for when a running drawdown becomes statistically unusual. This helps distinguish expected drawdowns from anomalous events that may warrant de-risking or hedging.

3.7 Stop-loss rules in theory and practice

A substantial literature evaluates stop-loss rules—both fixed-percentage and trailing—under different data-generating processes and cost structures. Kaminski and Lo (2014) show that stops can improve risk-adjusted performance when returns display positive serial correlation and regime persistence, but often underperform buy-and-hold in mean-reverting markets or with high trading costs. Lo and Remorov (2017) extend this to regime-switching models with transaction costs.

3.8 The triple penance and behavioral reality of stops

Beyond mean–variance trade-offs, stops impose triple costs: (i) realizing losses, (ii) risking missed rebounds, and (iii) increasing trading and tax frictions—especially when whipsaws are frequent (Bailey and Lopez de Prado 2014). This perspective helps explain why stops may reduce MDD but still disappoint investors if re-entry logic is weak.

3.9 Stops, momentum, and reflexivity

Trailing stops that re-enter only after a prior peak mechanically embed a momentum filter: exposure is kept when trend is healthy and cut when it’s impaired, with re-risking contingent on trend resumption. This links stops to time-series momentum and moving-average rules, and to reflexivity in which losses tighten risk limits, reduce liquidity, and deepen trends.

3.10 Correlation, contagion, and systemic warnings

Large drawdowns often coincide with spikes in cross-asset correlations and concentration of variance in the first principal component; monitoring rolling eigenvalues of the correlation matrix can serve as a contemporaneous warning of systemic fragility (Conlon and Cotter 2010).

3.11 From warnings to action: hedging and overlays

Predictive signals must tie to concrete actions. Broadly, there are two levers: (i) exposure control via stops or momentum overlays, and (ii) explicit hedges (e.g., VIX futures or index puts). The most promising results come from integrated policies that tie hedge intensity or stop thresholds to early-warning probabilities, while accounting for costs and whipsaws.

4 Simulation Framework

5 Simulation Framework

We simulate synthetic return paths to test drawdown-mitigation rules under controlled statistical environments. This enables us to:

- Hold volatility constant while varying drift or autocorrelation.
- Quantify how stop-loss rules perform relative to buy-and-hold when return persistence changes.
- Observe how MDD, average drawdown, volatility, and Sharpe ratio respond to rule changes.

5.1 AR(1) process with trailing stop-loss

The base process for log-returns is:

$$r_t = \mu + \rho(r_{t-1} - \mu) + \sigma\varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, 1)$$

- μ : daily drift
- ρ : lag-1 autocorrelation coefficient
- σ : daily volatility
- We parameterize μ and σ in **annualized** terms for interpretability.

The stop-loss rule is:

- **Trigger**: 5% trailing peak-to-trough decline in portfolio value.
- **Exit**: Go to cash immediately upon breach.
- **Re-entry**: Only when price exceeds the prior peak (“stop-out peak”), ensuring no premature re-risking.

5.2 Implementation

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def simulate_ar1_paths(n_paths=100, n_days=252*5, mu_ann=0.0, sigma_ann=0.05, rho=0.0, seed=1234):
    np.random.seed(seed)
    mu = mu_ann / 252
    sigma = sigma_ann / np.sqrt(252)
    rets = np.zeros((n_days, n_paths))
    shocks = np.random.randn(n_days, n_paths)
    for t in range(1, n_days):
        rets[t] = mu + rho * (rets[t-1] - mu) + sigma * shocks[t]
    return rets

def apply_trailing_stop(returns, stop=0.05):
    """returns: 1D array of daily returns"""
    wealth = np.cumprod(1 + returns)
    peak = np.maximum.accumulate(wealth)
    in_market = np.ones(len(wealth), dtype=bool)
    cash = 0.0
    stop_peak = peak[0]
    for t in range(1, len(wealth)):
        if in_market[t-1]:
            # Check stop breach
            if (wealth[t] / peak[t] - 1) <= -stop:
                in_market[t] = False
                cash = wealth[t]
                stop_peak = peak[t]
            else:
                in_market[t] = True
        else:
            # Stay in cash until break above stop_peak
            if wealth[t] >= stop_peak:
                in_market[t] = True
                cash = 0.0
            else:
                in_market[t] = False
    strat_wealth = np.ones(len(wealth))
    strat_wealth[0] = 1.0
```



```

in_cash = False
for t in range(1, len(wealth)):
    if in_market[t]:
        strat_wealth[t] = strat_wealth[t-1] * (1 + returns[t])
    else:
        strat_wealth[t] = strat_wealth[t-1]
return strat_wealth

def metrics(wealth):
    rets = pd.Series(wealth).pct_change().dropna()
    vol = rets.std() * np.sqrt(252)
    cagr = (wealth[-1]/wealth[0])**((252/len(wealth))-1)
    sharpe = cagr/vol if vol>0 else np.nan
    dd = wealth/np.maximum.accumulate(wealth) - 1
    mdd = dd.min()
    avgdd = dd[dd<0].mean()
    return dict(CAGR=cagr, Vol=vol, Sharpe=sharpe, MDD=mdd, AvgDD=avgdd)

# Example: rho = 0.3, mu = 0
rets = simulate_ar1_paths(n_paths=1, rho=0.3, mu_ann=0.0)
rets = rets[:,0]
wealth_bh = np.cumprod(1+rets)
wealth_stop = apply_trailing_stop(rets)

# Metrics
print("Buy & Hold:", metrics(wealth_bh))
print("Stop-Loss:", metrics(wealth_stop))

# Plot
plt.figure(figsize=(8,4))
plt.plot(wealth_bh, label="Buy & Hold")
plt.plot(wealth_stop, label="5% Trailing Stop")
plt.title("Example Path - AR(1) =0.3, =0")
plt.legend()
plt.tight_layout()
plt.show()

```

```

Buy & Hold: {'CAGR': np.float64(0.04237321398912175), 'Vol': np.float64(0.05150267861021977)}
Stop-Loss: {'CAGR': np.float64(0.022248918103237925), 'Vol': np.float64(0.03398916770371736)}

```

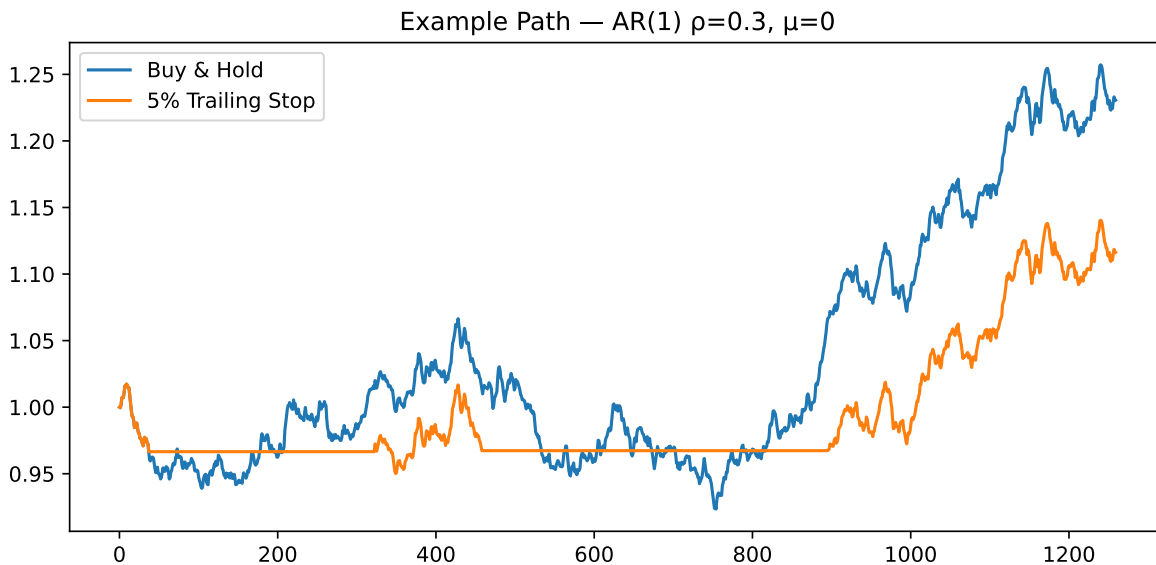


Figure 5.1: AR(1) simulation — Buy & Hold vs 5% trailing stop-loss

5.3 Design choices

Fixed stop threshold: Here set at 5% to illustrate; later sweeps will show sensitivity.

No transaction costs: This isolates pure statistical effects; costs will be added in empirical sections.

In-sample vs out-of-sample: Since AR(1) parameters are known here, there's no estimation error. This will not hold in real data.

In the next section, we will sweep ρ from -0.5 to 0.5 and drifts from -10% to +10% annually, holding volatility constant, to see when stops help/hurt on each performance dimension.

6 Empirical Measurement

We fetch daily data with **yfinance** and compute: returns, wealth & drawdowns, rolling β , and a summary table for SPY, IEF, and BTC.

```
import pandas as pd, numpy as np, matplotlib.pyplot as plt
from scripts.empirical_data import load_prices_yf, pct_returns, wealth_series, drawdown_series

tickers = {
    "S&P 500 (SPY adj close)": "SPY",
    "US 7-10Y Treasury (IEF)": "IEF",
    "Bitcoin (BTC-USD)": "BTC-USD",
    "VIX Index (proxy only)": "^VIX"
}

prices = {}
for label, t in tickers.items():
    try:
        prices[label] = load_prices_yf(t, start="2003-01-01")
    except Exception as e:
        print(f"Skipping {label}: {e}")

rets = {k: pct_returns(v) for k, v in prices.items() if v is not None}
summ = {k: summary_metrics(r) for k, r in rets.items() if k != "VIX Index (proxy only)"}
pd.DataFrame(summ).T
```

Ticker	SPY
Date	
2003-01-02	59.986370
2003-01-03	60.170784
2003-01-06	61.231274
2003-01-07	61.079792
2003-01-08	60.197128

Ticker	IEF
Date	
2003-01-02	44.324085

```

2003-01-03  44.397270
2003-01-06  44.287552
2003-01-07  44.428650
2003-01-08  44.533199
Ticker      BTC-USD
Date
2014-09-17  457.334015
2014-09-18  424.440002
2014-09-19  394.795990
2014-09-20  408.903992
2014-09-21  398.821014
Ticker      ^VIX
Date
2003-01-02  25.389999
2003-01-03  24.680000
2003-01-06  24.910000
2003-01-07  25.129999
2003-01-08  25.530001

```

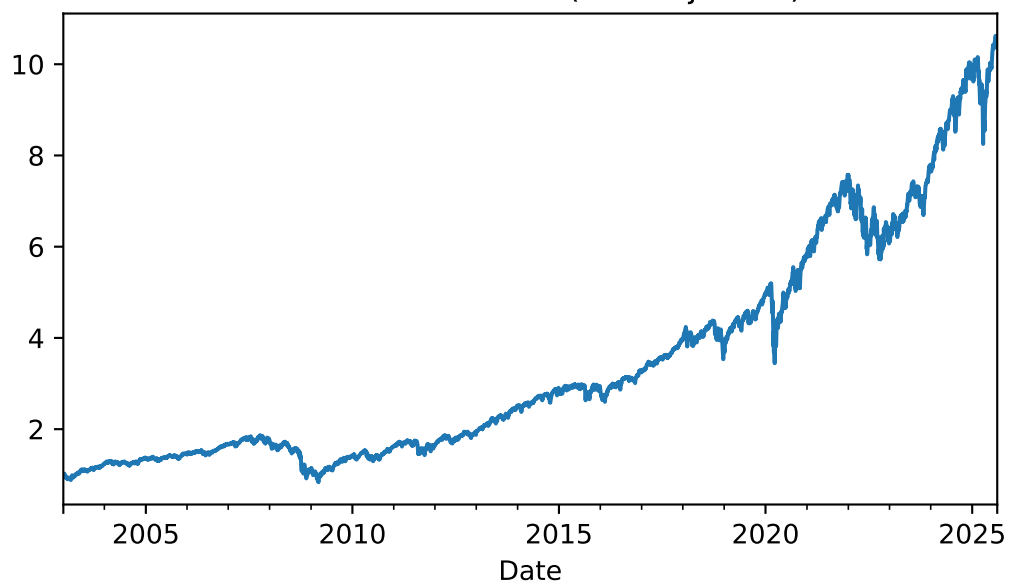
	CAGR	Vol	Sharpe	MDD	AvgDD	TUW_days
S&P 500 (SPY adj close)	0.106012	0.184301	0.575212	-0.551894	-0.078665	5175.0
US 7-10Y Treasury (IEF)	0.033232	0.067210	0.494447	-0.239248	-0.054519	5593.0
Bitcoin (BTC-USD)	0.647683	0.663650	0.975942	-0.830363	-0.370045	2695.0

```

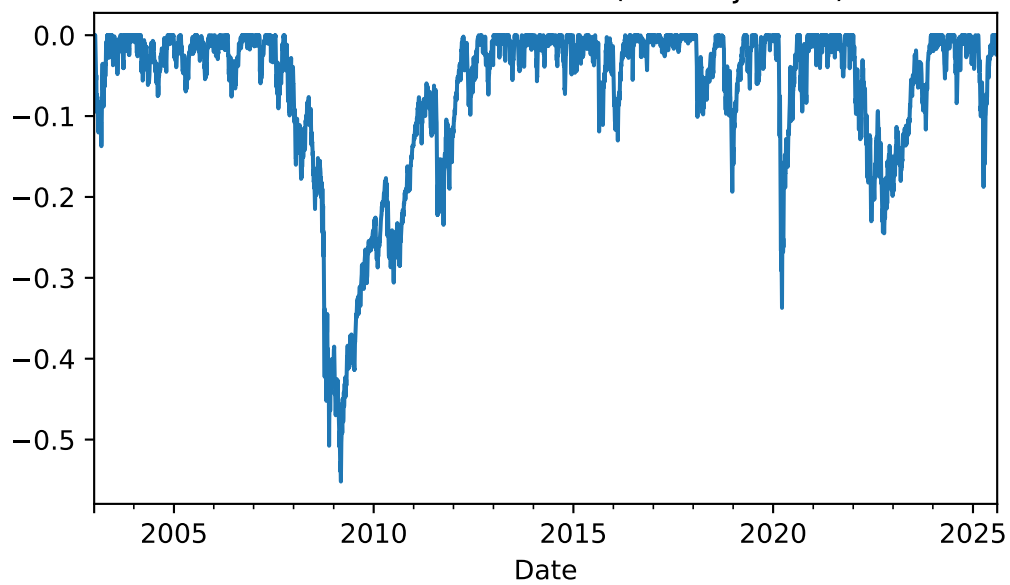
# Wealth and drawdowns
for label in ["S&P 500 (SPY adj close)", "US 7-10Y Treasury (IEF)", "Bitcoin (BTC-USD)"]:
    if label not in rets:
        continue
    r = rets[label]
    W = wealth_series(r)
    dd = drawdown_series(r)
    plt.figure(); W.plot(title=f"Wealth - {label}"); plt.tight_layout(); plt.show()
    plt.figure(); dd.plot(title=f"Drawdown - {label}"); plt.tight_layout(); plt.show()

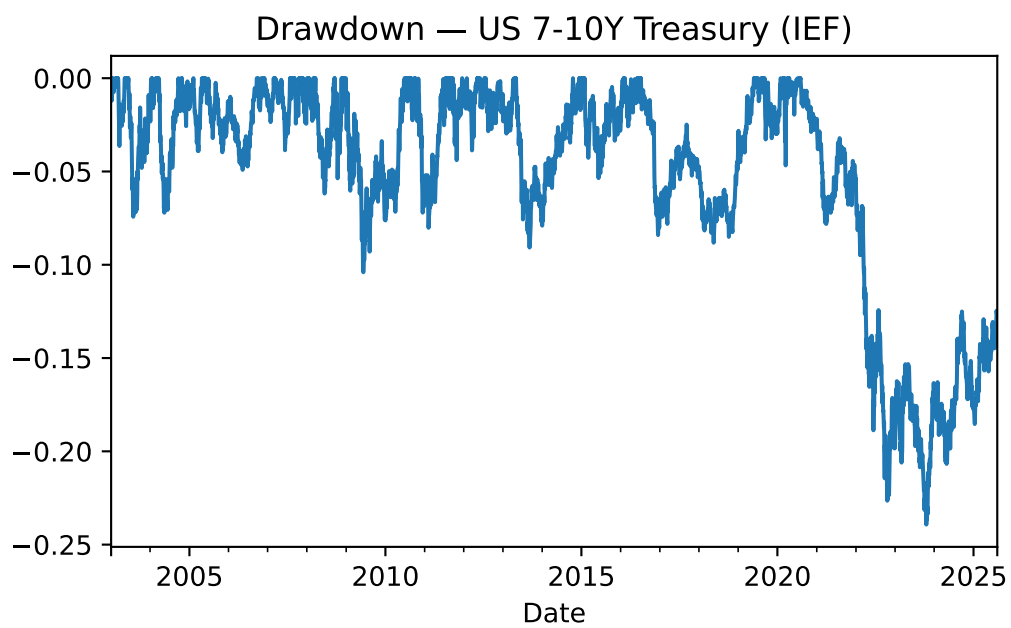
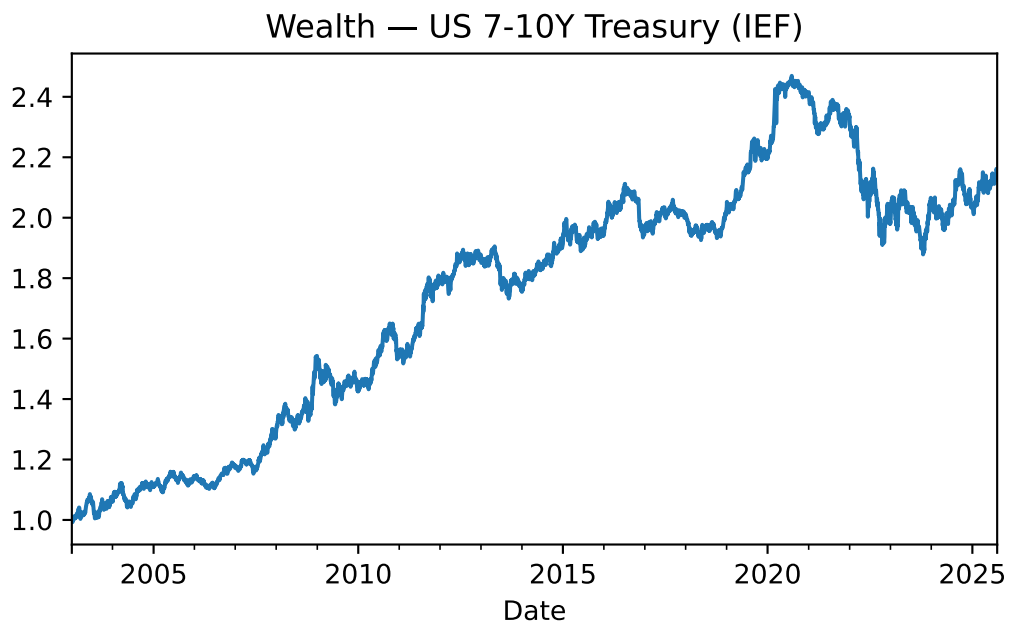
```

Wealth — S&P 500 (SPY adj close)

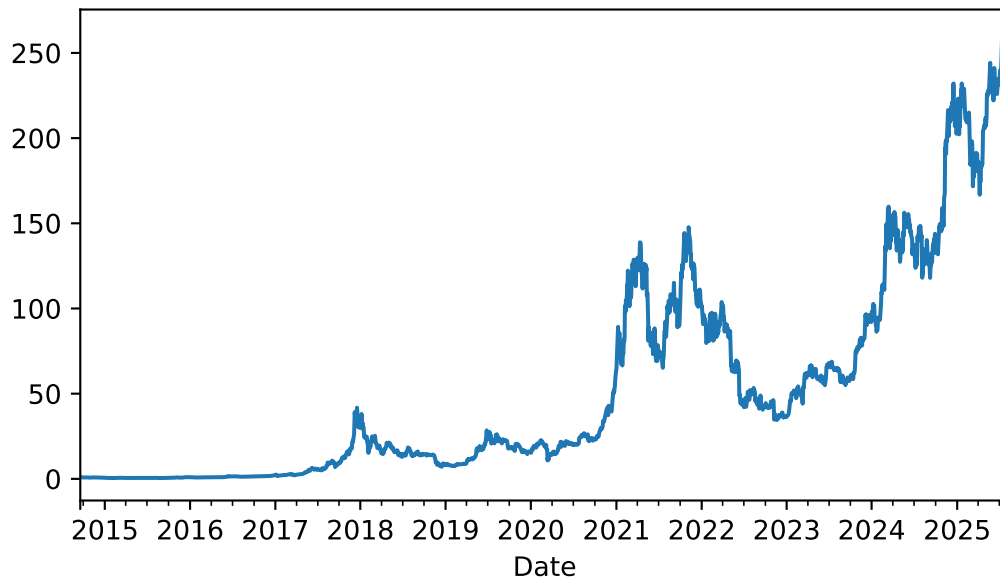


Drawdown — S&P 500 (SPY adj close)

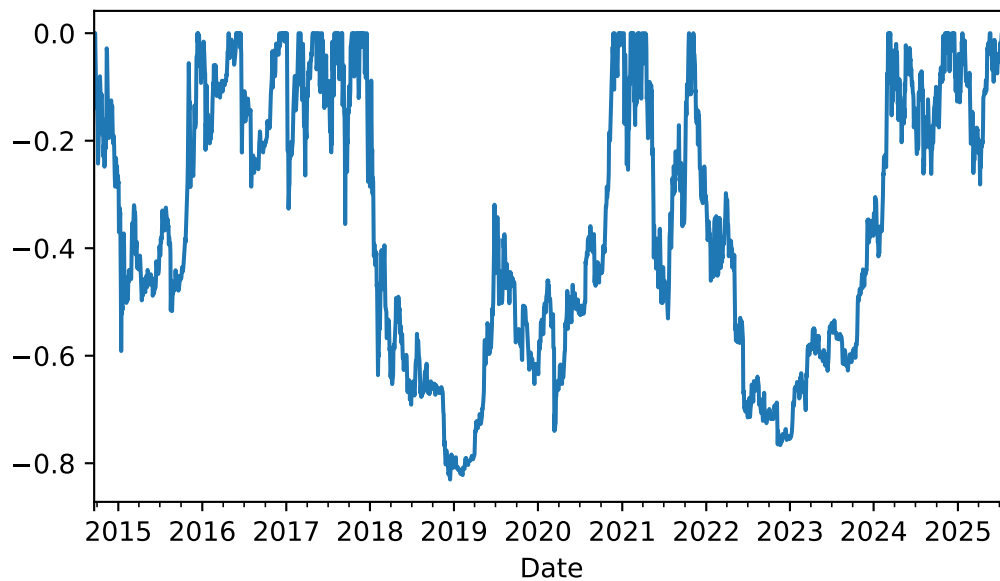




Wealth — Bitcoin (BTC-USD)

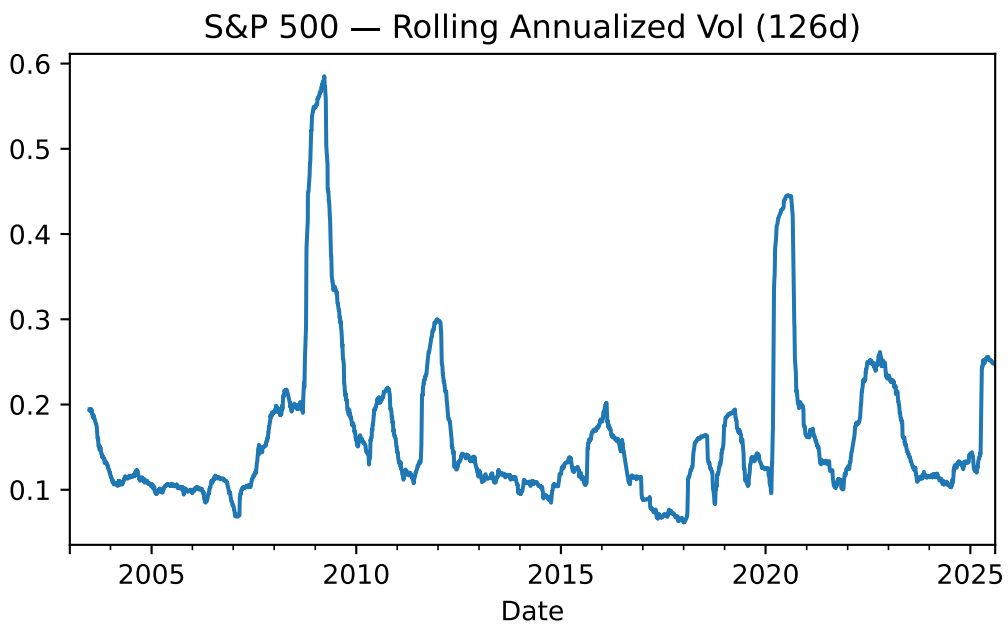
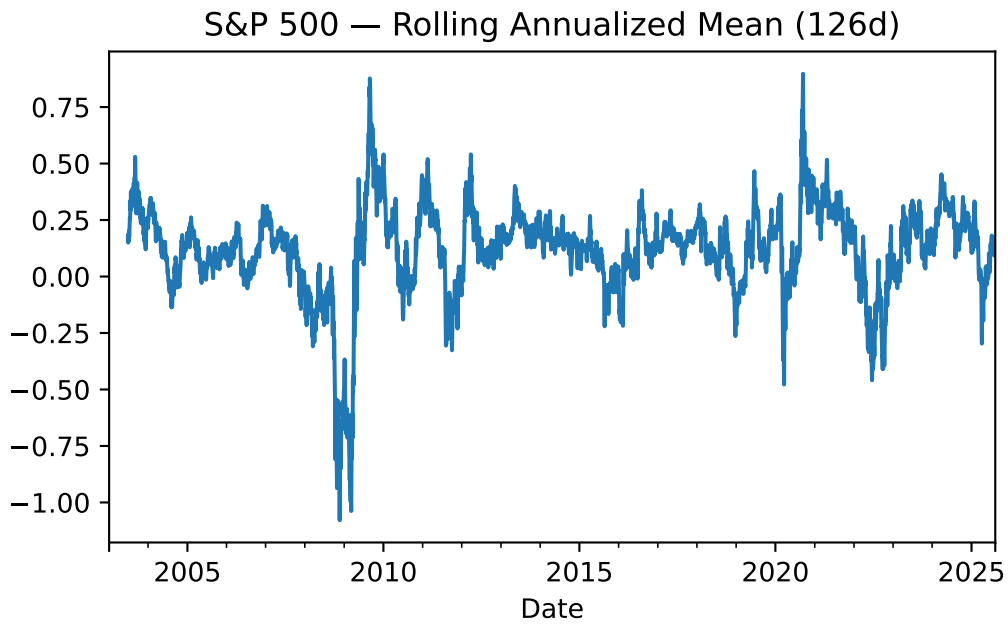


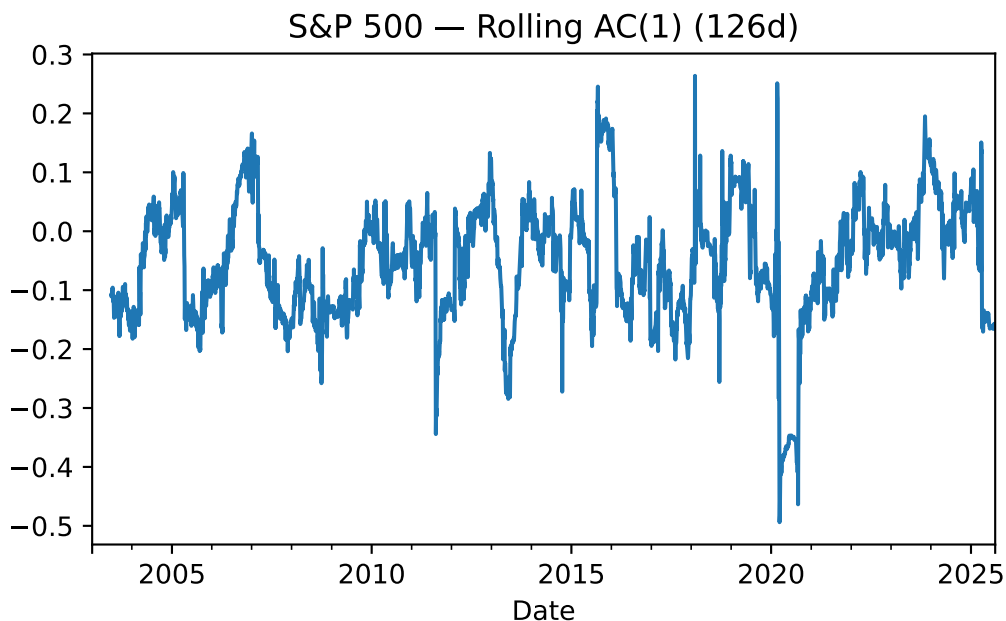
Drawdown — Bitcoin (BTC-USD)



```
# Rolling stats (S&P 500 example)
label = "S&P 500 (SPY adj close)"
if label in rets:
    r = rets[label]
    roll = rolling_stats(r, w_short=20, w_long=126)
```

```
plt.figure(); roll["mu_ann"].plot(title="S&P 500 - Rolling Annualized Mean (126d)"); plt
plt.figure(); roll["vol_ann"].plot(title="S&P 500 - Rolling Annualized Vol (126d)"); plt
plt.figure(); roll["rho1"].plot(title="S&P 500 - Rolling AC(1) (126d)"); plt.tight_layout
```





```
from scripts.reporting import compare_strategies, save_table
rets_named = {k: v for k, v in rets.items() if k in ["S&P 500 (SPY adj close)", "US 7-10Y Treasury (IEF)", "Bitcoin (BTC-USD)"]}
if len(rets_named) >= 1:
    table = compare_strategies(rets_named)
    display(table)
    out_csv = save_table(table, "figures/empirical_summary.csv")
    print("Saved summary CSV to:", out_csv)
```

Strategy	CAGR	Vol	Sharpe	MDD	AvgDD	TUW_days
S&P 500 (SPY adj close)	0.106012	0.184301	0.575212	-0.551894	-0.078665	5175
US 7-10Y Treasury (IEF)	0.033232	0.067210	0.494447	-0.239248	-0.054519	5593
Bitcoin (BTC-USD)	0.647683	0.663650	0.975942	-0.830363	-0.370045	2695

Empirical summary exported to CSV

Saved summary CSV to: figures/empirical_summary.csv

7 Early Warning Signals

We build features from price-only data (volatility ratio, drawdown slope, SMA deviation, and a lightweight LPPL risk score) and predict whether a large drawdown occurs within a forward horizon.

```
import pandas as pd, numpy as np, matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score, roc_curve, classification_report
from scripts.empirical_data import load_prices_yf
from scripts.early_warnings import build_ews_features_from_prices, label_future_drawdown_from_prices

spy = load_prices_yf("SPY", start="2000-01-01")
X = build_ews_features_from_prices(spy)
y = label_future_drawdown_from_prices(spy, horizon=60, threshold=-0.15)

df = pd.concat([X, y], axis=1).dropna()
split = int(len(df)*0.7)
X_train, X_test = df.iloc[:split, :-1], df.iloc[split:, :-1]
y_train, y_test = df.iloc[:split, -1], df.iloc[split:, -1]

clf = LogisticRegression(max_iter=2000)
clf.fit(X_train, y_train)
proba = clf.predict_proba(X_test)[:,-1]
auc = roc_auc_score(y_test, proba)
print(f"AUC: {auc:.3f}")

# ROC
fpr, tpr, _ = roc_curve(y_test, proba)
plt.figure(); plt.plot(fpr, tpr, label=f"AUC={auc:.3f}"); plt.plot([0,1],[0,1], 'k--')
plt.xlabel("False Positive Rate"); plt.ylabel("True Positive Rate")
plt.title("ROC - EWS with LPPL (SPY)"); plt.legend(); plt.tight_layout(); plt.show()

# Feature importances (coef magnitude)
imp = pd.Series(clf.coef_[0], index=X.columns).sort_values(key=np.abs, ascending=False)
print("Top features:\n", imp.head(10))
```

```
# Classification report at 0.5 threshold
pred = (proba >= 0.5).astype(int)
print(classification_report(y_test, pred, digits=3))

# Export predictions
out = pd.DataFrame({"proba": proba, "y_test": y_test.values}, index=y_test.index)
out.to_csv("figures/ews_lppl_spy_predictions.csv")
print("Saved predictions to figures/ews_lppl_spy_predictions.csv")
```

Ticker	SPY
Date	
2000-01-03	92.142494
2000-01-04	88.539185
2000-01-05	88.697586
2000-01-06	87.272102
2000-01-07	92.340508
AUC:	0.636

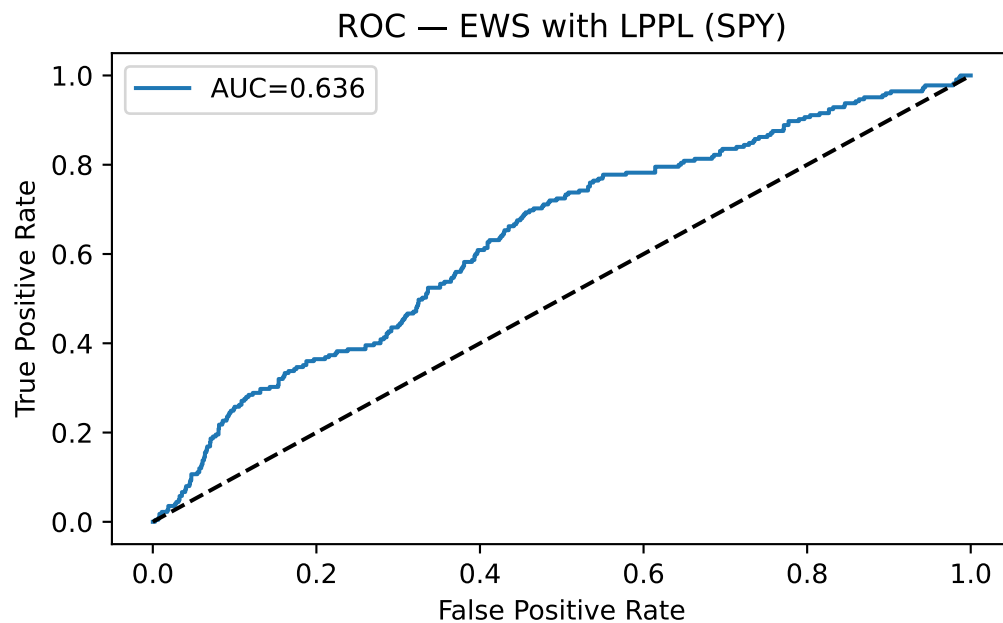


Figure 7.1: SPY early-warning model with LPPL risk

Top features:

sma_dev	-10.152640
---------	------------

```

vol_ratio      0.399041
lppl_risk      -0.094796
dd_slope       0.080637
dtype: float64

```

	precision	recall	f1-score	support
0.0	0.886	0.997	0.938	1742
1.0	0.143	0.004	0.009	225
accuracy			0.883	1967
macro avg	0.514	0.501	0.473	1967
weighted avg	0.801	0.883	0.832	1967

Saved predictions to figures/ews_lppl_spy_predictions.csv

8 Stop-Loss, Momentum, and Hedging

8.1 Demos

```
import numpy as np, pandas as pd, matplotlib.pyplot as plt
from scripts.hedging import rolling_put_hedge, vix_overlay
from scripts.simulations import simulate_ar1

price = simulate_ar1(mu_ann=0.00, rho=0.2, sigma_ann=0.15, years=3, seed=123)
rets = price.pct_change().dropna()

hedged, roll_flags = rolling_put_hedge(price, moneyness=0.95, tenor_days=21, iv_annual=0.20)

wealth_asset = (1+rets).cumprod()
wealth_hedged = (1+hedged).cumprod()

plt.figure(); wealth_asset.plot(label="Asset"); wealth_hedged.plot(label="Put-hedged")
plt.title("Wealth: Asset vs Rolling 95% Put Hedge (toy)"); plt.legend(); plt.tight_layout();

vix_proxy = -rets * 3.0
trigger = rets.rolling(10).std().shift(1) > rets.rolling(60).std().shift(1)
vix_ov = vix_overlay(rets, vix_proxy, trigger, hedge_weight=0.2).dropna()
plt.figure(); (1+rets).cumprod().plot(label="Asset"); (1+vix_ov).cumprod().plot(label="VIX ov")
plt.title("Wealth: Asset vs VIX Overlay (toy)"); plt.legend(); plt.tight_layout(); plt.show()
```

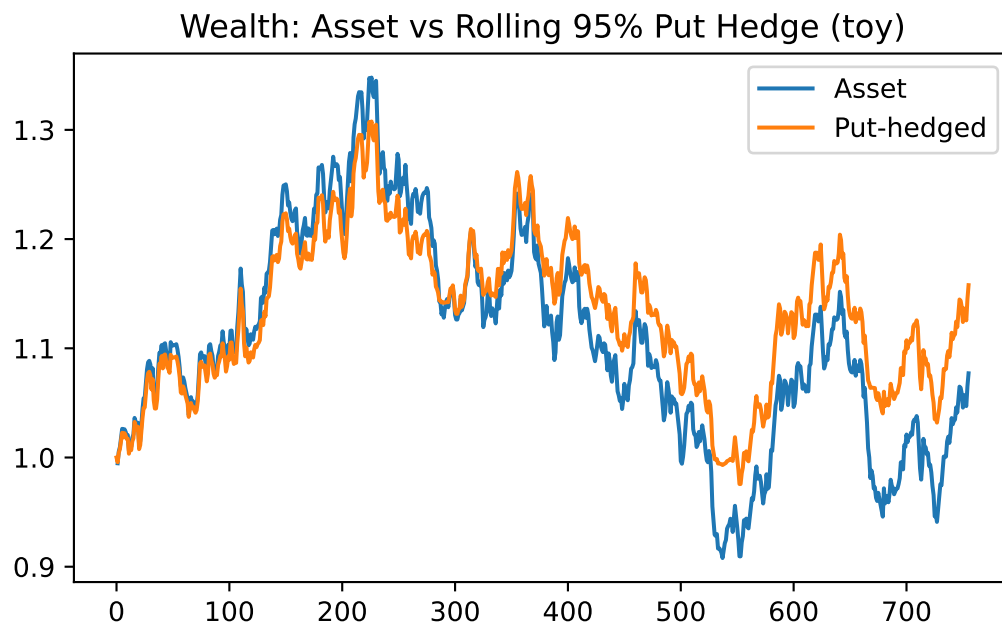
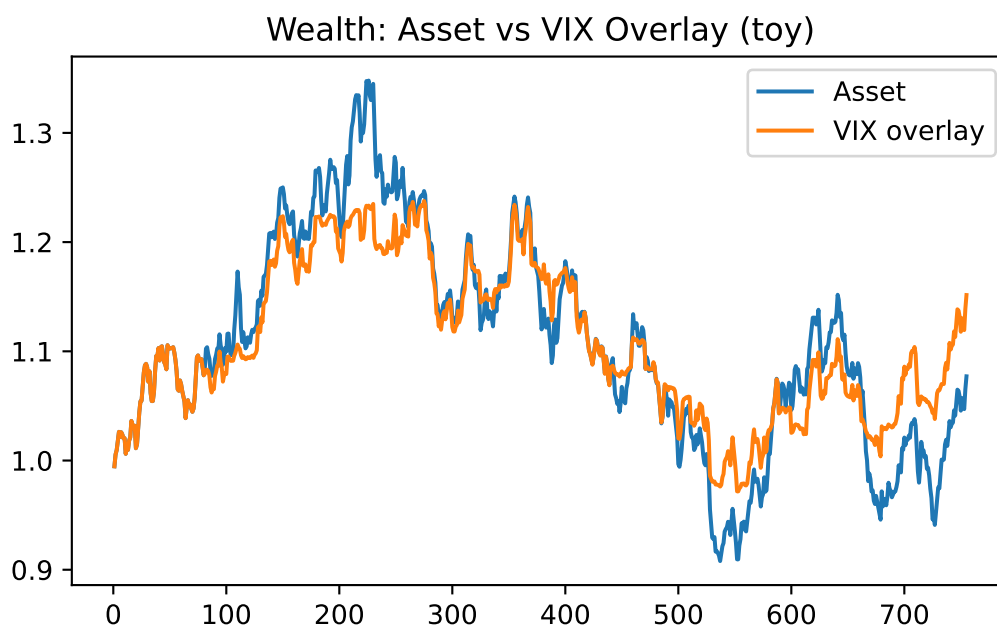


Figure 8.1: Illustrative hedging overlays (toy examples)



```
import numpy as np, pandas as pd, matplotlib.pyplot as plt
from scripts.simulations import simulate_ar1
```

```

from scripts.stoploss import apply_trailing_stop, apply_costs

price = simulate_ar1(mu_ann=0.04, rho=0.1, sigma_ann=0.15, years=5, seed=7)
rets = price.pct_change().dropna()

sl_rets, pos, trade, tovr = apply_trailing_stop(price, stop_pct=0.05, reenter_above_peak=True)
sl_costed = apply_costs(sl_rets, trade, tovr, fixed_bps=1.0, slippage_bps=2.0, spread_bps=5.0)

wealth_bh = (1+rets).cumprod()
wealth_sl = (1+sl_rets).cumprod()
wealth_sl_cost = (1+sl_costed).cumprod()

plt.figure(); wealth_bh.plot(label="Buy & Hold"); wealth_sl.plot(label="Stop 5% (no costs)");
plt.title("Wealth with Stop-Loss and Transaction Costs (toy)"); plt.legend(); plt.tight_layout()

```

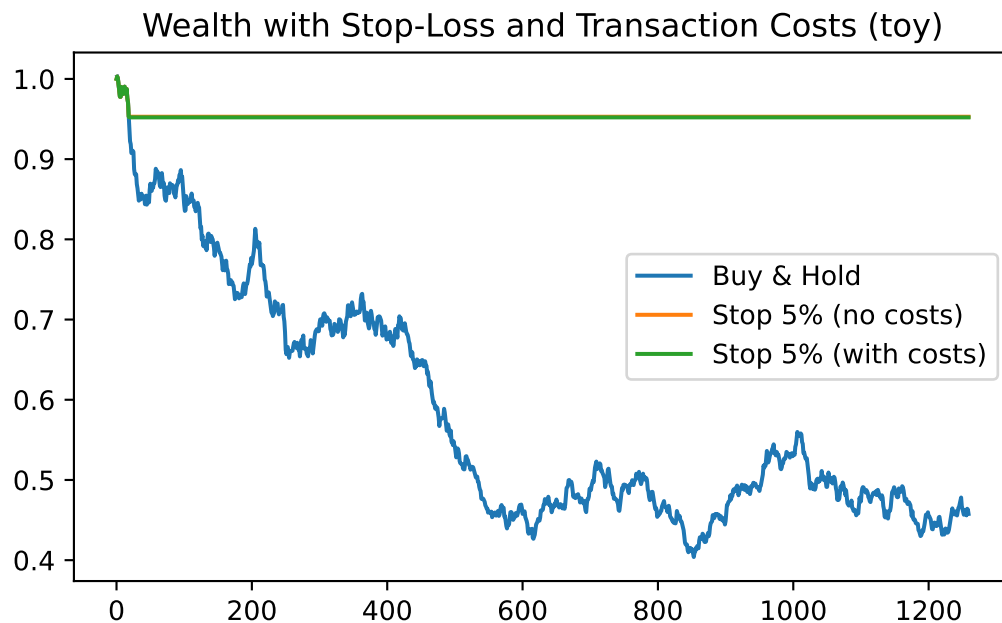


Figure 8.2: Stop-loss with transaction costs (toy example)

8.2 Data-Driven Hedge Example (VIX proxy)

```

import pandas as pd, numpy as np, matplotlib.pyplot as plt
from scripts.empirical_data import load_prices_yf, pct_returns, wealth_series
from scripts.hedging import vix_overlay
from scripts.stoploss import apply_trailing_stop, apply_costs

spy = load_prices_yf("SPY", start="2006-01-01")
vix = load_prices_yf("^VIX", start="2006-01-01")

r_spy = pct_returns(spy)
r_vix = pct_returns(vix)

vol_s = r_spy.rolling(20).std().shift(1)
vol_l = r_spy.rolling(60).std().shift(1)
trigger = (vol_s > vol_l).reindex_like(r_spy).fillna(False)

sl_rets, pos, trade, tovr = apply_trailing_stop(spy, stop_pct=0.05, reenter_above_peak=True)
sl_rets = sl_rets.reindex_like(r_spy).fillna(0.0)
sl_costed = apply_costs(sl_rets, trade, tovr, fixed_bps=1.0, slippage_bps=2.0, spread_bps=5.0)

combo = vix_overlay(sl_costed, r_vix, trigger, hedge_weight=0.20).dropna()

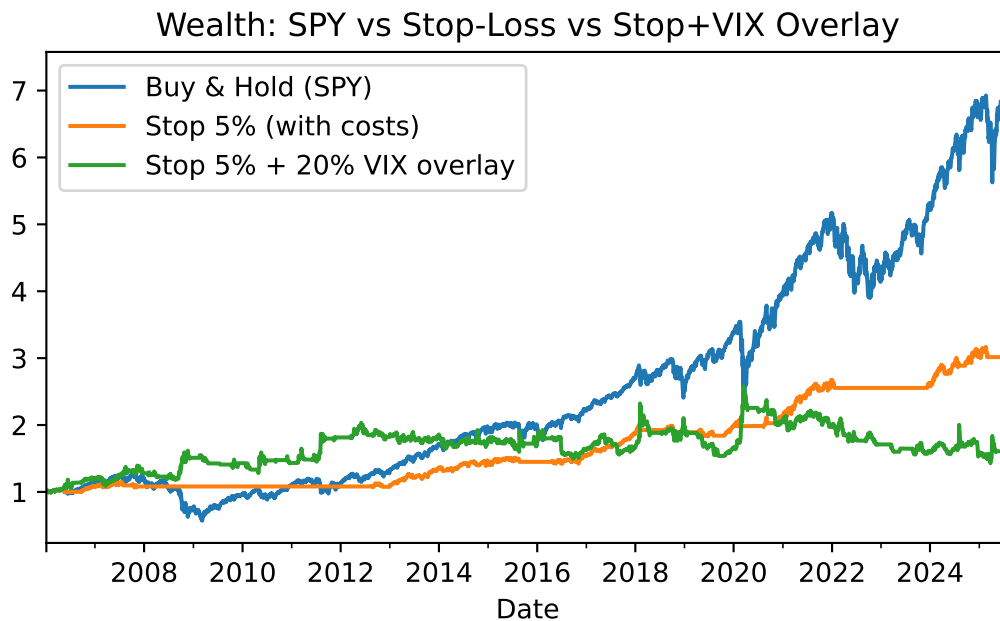
W_bh = wealth_series(r_spy)
W_sl = wealth_series(sl_costed)
W_combo = wealth_series(combo)

plt.figure(); W_bh.plot(label="Buy & Hold (SPY)"); W_sl.plot(label="Stop 5% (with costs)"); W_combo.plot(label="Stop 5% + VIX Overlay");
plt.title("Wealth: SPY vs Stop-Loss vs Stop+VIX Overlay"); plt.legend(); plt.tight_layout();

```

Ticker	SPY
Date	
2006-01-03	87.964905
2006-01-04	88.381447
2006-01-05	88.437012
2006-01-06	89.172943
2006-01-09	89.402100

Ticker	^VIX
Date	
2006-01-03	11.14
2006-01-04	11.37
2006-01-05	11.31
2006-01-06	11.00



```
from scripts.reporting import compare_strategies, save_table, save_figure
from scripts.empirical_data import wealth_series

comparables = {
    "Buy & Hold (SPY)": r_spy,
    "Stop 5% (with costs)": sl_costed,
    "Stop 5% + 20% VIX overlay": combo
}
table = compare_strategies(comparables)
display(table)

csv_path = save_table(table, "figures/spy_strategy_comparison.csv")
print("Saved strategy metrics to:", csv_path)

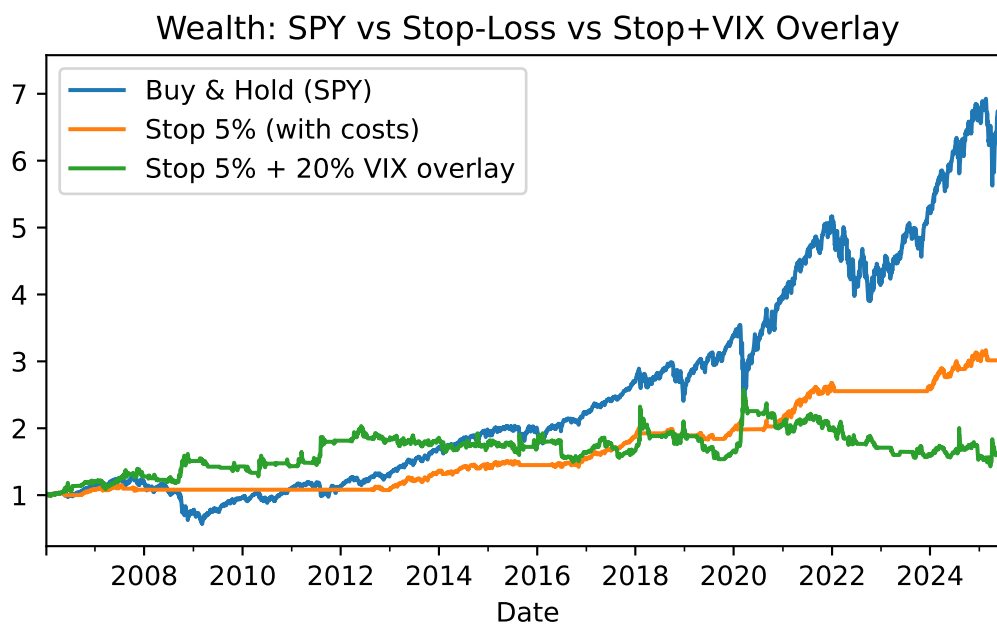
plt.figure()
wealth_series(r_spy).plot(label="Buy & Hold (SPY)")
wealth_series(sl_costed).plot(label="Stop 5% (with costs)")
wealth_series(combo).plot(label="Stop 5% + 20% VIX overlay")
plt.title("Wealth: SPY vs Stop-Loss vs Stop+VIX Overlay"); plt.legend()
png_path = save_figure("figures/spy_wealth_strategies.png")
print("Saved figure to:", png_path)
```

Strategy	CAGR	Vol	Sharpe	MDD	AvgDD	TUW_days
Buy & Hold (SPY)	0.102128	0.191638	0.532922	-0.551895	-0.086434	4495
Stop 5% (with costs)	0.058013	0.072148	0.804087	-0.087379	-0.039348	4739
Stop 5% + 20% VIX overlay	0.025733	0.179518	0.143343	-0.445388	-0.168309	5034

SPY: Buy&Hold vs Stop vs Stop+VIX — metrics and exports

Saved strategy metrics to: figures/spy_strategy_comparison.csv

Saved figure to: figures/spy_wealth_strategies.png



9 Results Synthesis

(Write-up and comparison tables to be added.)

10 Appendices

(Math notes, parameter grids, and full tables to be added.)

- Angoshtari, Bahman, Erhan Bayraktar, and Virginia R. Young. 2015. “Minimizing the Probability of Drawdown in Diffusion Models.” *Stochastic Processes and Their Applications* 125 (5): 1810–65.
- Bailey, David H., and Marcos Lopez de Prado. 2014. “Stop-Loss Strategies, Serial Correlation, and the Triple Penance Rule.” *Journal of Risk* 17 (2): 91–113.
- Chekhlov, Alexei, Stanislav Uryasev, and Roman Zabarankin. 2005. “Drawdown Measure in Portfolio Optimization.” *International Journal of Theoretical and Applied Finance* 8 (1): 13–58.
- Conlon, Thomas, and John Cotter. 2010. “Cross-Market Linkages and Financial Contagion.” *International Review of Financial Analysis* 19 (3): 190–202.
- Goldberg, Lisa R., and Owais Mahmoud. 2014. “Drawdown: From Practice to Theory and Back Again.” *Journal of Investment Strategies* 3 (4): 29–46.
- Grossman, Sanford J., and Zhongquan Zhou. 1993. “Optimal Investment Strategies for Controlling Drawdowns.” *Mathematical Finance* 3 (3): 241–76.
- Hadjiliadis, Olympia, and Jan Vecer. 2006. “Drawdowns and the Speed of Market Recovery.” *Stochastic Processes and Their Applications*.
- Harvey, Campbell R., and Yan Liu. 2020. “Drawdown: Measures, Risk and Mitigation.” *Journal of Portfolio Management* 46 (6): 28–39.
- Hsieh, Chuangyin, and B. Ross Barmish. 2017. “On Drawdown-Modulated Feedback Control in Stock Trading.” *Quantitative Finance* 17 (3): 329–40.
- Johansen, Anders, and Didier Sornette. 2000. “Critical Crashes.” *Risk* 12 (1): 91–94.
- Kaminski, Kathryn M., and Andrew W. Lo. 2014. “When Do Stop-Loss Rules Stop Losses?” *Journal of Portfolio Management* 40 (2): 45–61.
- Lo, Andrew W., and Alexander Remorov. 2017. “Stop-Loss Strategies with Serial Correlation, Regime Switching, and Transactions Costs.” *Journal of Portfolio Management* 43 (2): 59–75.
- Magdon-Ismael, Malik, Amir F. Atiya, Ashraf Pratap, and Yaser S. Abu-Mostafa. 2004. “Maximum Drawdown Under Brownian Motion.” *Risk*.
- Rej, Adam, Philip Seager, and Jean-Philippe Bouchaud. 2017. “You Are in a Drawdown. When Should You Start Worrying?” *arXiv Preprint*.
- Sornette, Didier. 2003. *Why Stock Markets Crash: Critical Events in Complex Financial Systems*. Princeton University Press.