Historically Stressed: How to Make Stress-Testing and Simulation More Intuitive and Insightful

Dane Rook [*,+], Dan Golosovker [+]

July 2023

[*] *Stanford University;* [+] *Addepar Research*

WELCOME

Simulation and stress-testing are best-practice tools for portfolio construction and management. Investors who lack these tools are at a competitive disadvantage, especially in times of abnormal market conditions - such as now. However, many of the existing techniques for simulation and stress-testing (hereafter, 'S&ST') don't properly fit investors' needs, because:

- They critically depend on assumptions about inputs - and relationships between those inputs - that can be difficult to formulate both precisely and confidently (e.g., probability distributions over scenarios, covariances between assets);[1]
- Their mechanics rely on embedded assumptions that are often unintuitive, unrealistic, or otherwise challenging to discern (or explain to clients); and
- Their outputs can be tricky to interpret, and may offer only shallow insights.

In this Primer, we introduce a novel paradigm, the *history-based multi-lens* (HBML) method, for S&ST that's assumptions-light and straightforward: it avoids black boxes, and its outputs are readily interpretable (and thus easily explained to clients), while still being insight-rich. HBML has wide applicability, and can be useful for portfolio construction, risk management, scenario exploration, back-testing, and wealth planning.

Our team at Addepar Research was specially equipped to develop HBML for three key reasons. First, HBML benefits from long historical time-series for a diverse swathe of assets; Addepar has

[1] Many S&ST methods also rely on assumptions that aren't inputs per se, but are nevertheless parameters that affect results. The number of runs used in a Monte Carlo simulation are an example of this class of parameters. We discuss Monte Carlo run counts later in this Primer.

extensive data resources, and was therefore in a privileged position to design, refine, and validate HBML. Second, although HBML itself is straightforward, testing its performance against other S&ST methods (like Monte Carlo) required some solid statistical acumen; our partnership with Stanford's Long-Term Investing initiative (SLTI) bolstered that expertise. Third, despite HBML's transparency, some components (e.g., submergence analytics) can't be practically implemented in spreadsheets; we've engineered an advanced toolkit for applying HBML that's both efficient and versatile. Collectively, these three factors let us pioneer an approach to S&ST that can help investors navigate the future with greater confidence, particularly in market environments they haven't directly encountered before.

This Primer's purpose is to: 1) discuss the deficiencies of current S&ST methods; 2) give a brief introduction to HBML; and 3) compare HBML's performance to other S&ST methods - Monte Carlo (MC) methods, in specific.[2] We find that, even though HBML requires fewer assumptions and is more straightforward than MC, it can be at least as accurate as MC, and often more so. In the course of our analysis, we demonstrate some of the insights HBML can supply, and discuss how these might aid investors. Together, these findings suggest that HBML should become part of the best-practice toolkit for investors.

KEY TAKEAWAYS

Here are the essential points that readers should glean from this Primer:

- Simulation and stress-testing capabilities are part of investment best-practice, but many methods for S&ST (such as Monte Carlo techniques) can present significant challenges, in terms of assumptions intensity, difficulty to explain to clients, or limited accuracy.
- We present a new paradigm, the history-based multi-lens (HBML) approach to S&ST, which is intuitive, insight-generating, and light on assumptions.
- Our tests show that HBML performs at least as well as Monte Carlo in various situations, which leads us to recommend that HBML become part of investors' standard analytical toolkits.

[2] Monte Carlo methods are a (very) broad class of techniques, but it's nevertheless commonplace to refer to both the entire class and specific instances of it as 'Monte Carlo'. We follow that convention in this Primer.

SIGNIFICANCE

Simulation and stress-testing are essential capabilities in modern investing, and they have broad applicability, for example in:

- Portfolio construction, where S&ST helps in identifying asset allocations and strategies that have an ideal balance of expected risk and return;
- Risk management, where S&ST can help reveal both the types and degrees of risk that candidate portfolios are likely to face under various market conditions;
- Back-testing, where S&ST can help investors study the behavior of portfolios in market environments that they themselves might not have directly experienced before;
- Scenario exploration, where S&ST can help investors envision the portfolio impacts of specific events (e.g., a hike in interest rates, rising inflation, or spiking unemployment); and
- Wealth planning, where S&ST can help investors uncover the likelihood of meeting their risk-adjusted return objectives, and map those to client needs (such as funding education, retirement, philanthropy, or other ambitions).

S&ST is distinct from 'point predictions' about the future (i.e. single values for expected return or risk) because it deals with asset dynamics over a period of time; it captures not just outcomes, but also paths by which those outcomes can be reached. In addition, (many) S&ST approaches yield a spectrum of possible outcomes with probabilities assigned to them, which arms investors with a fuller picture of risk.[3]

However, these analytical benefits generally come at a cost. Many popular S&ST methods are: 1) *assumptions-heavy* (both in terms of inputs required from users, and embedded assumptions on how these inputs interact over the course of the simulation); and 2) '*monocular*' in how they treat risk - i.e. they emphasize only one or two aspects of risk, but do a poor job at handling other aspects. The first shortcoming is problematic because many of the assumptions on which S&ST methods rely can be tough to formulate with precision (most S&ST approaches require explicitly specified probability distributions or covariance relationships).[4] This can result in users having

[3] In general, point predictions are just the most likely outcomes under these probability distributions.

[4] For asset classes that are well-established and well-populated (i.e., there are many individual assets in them), the problem of formulating assumptions can be less of a challenge than for younger, sparser asset classes (e.g. venture

meager confidence in their chosen inputs - a predicament that should lead them to have even less confidence in the method's outputs.

The second shortcoming (the monocular treatment of risk) is concerning because risk is almost always multidimensional. Many S&ST models (and their commercial implementations) presume that investors care only about volatility, when in reality they care about a host of other risks, such as the likelihood of extreme losses or large drawdowns, which aren't well-captured by volatility.[5] And while it's hypothetically possible to extend these models to capture other dimensions of risk, the assumptions required to do so can be precarious, and involve things like autocorrelation and conditional covariance (we discuss them below). These perils may prompt some investors to ask whether the benefits of S&ST genuinely outweigh the costs - i.e., is the juice worth the squeeze?

Yet, for many investors, that question is somewhat irrelevant, since - increasingly - many clients (and regulators) now demand that S&ST be integrated into an investor's core processes. Further, many investors are feeling pressure to use one family of approaches in particular: Monte Carlo (MC).[6] However, MC methods are assumptions-intensive - and often prohibitively so for many investors. Therefore, whenever clients request that MC methods be used, we urge investors to ask (themselves, or else their clients directly): what do they *expect* from MC? In many cases, the answer will be something along the lines of: "it gives an idea of the *range* of outcomes that can be anticipated, as well as an overall *probability* of hitting our objectives". Having such data can give clients comfort, but if MC (or any other S&ST) techniques are misapplied, then that comfort is undeserved, and both the investor and client may end up adopting more risk than they expect. So, investors can find themselves in a pickle (or whatever other deli snack they find undesirable) when it comes to S&ST.

capital or crypto-currency funds). For instance, it may be simple (and valid) to assume that a newly-IPO'd stock will have a similar return distribution to other stocks in its industry; but it might be shakier to assume that a first-time VC fund will behave like similar funds (indeed, even specifying "similar" can be tricky and highly subjective).

[5] That is, many risks apart from volatility aren't realistically generated from volatility alone, and other assumptions and modeling mechanisms are needed to incorporate them.

[6] For readers who may be unfamiliar with (or need a refresher on) MC methods, we give a concise overview below.

In response to these clear limitations, we developed a novel approach - the *history-based multi-lens* (HBML) method. But before we introduce HBML, let's delve deeper into several competing S&ST methods, and discuss why HBML improves upon them.

CONTEXT

Much of modern financial practice is based on ideas from the 1950s - chiefly, Modern Portfolio Theory (MPT).[7] MPT is elegant in its simplicity, and while some of its implications are timeless (e.g. diversification is useful - up to a point), some of its assumptions are dubious, particularly now that investors have access to far more data and computing power than they did back when MPT was conceived. Foremost among these assumptions is that investors only care about the first two statistical moments of returns distributions: mean and variance.[8] This assumption is demonstrably false: investors care about many other properties of distributions (e.g., skewness and tail risk), as well as risk phenomena that can't be derived from return distributions alone (e.g., autocorrelations that increase the likelihood of intense drawdowns). Nevertheless, the most prevalent approaches to simulation and stress-testing were architected around MPT, and share in its monocular view of risk-return relationships.

In fairness, all methods for S&ST are imperfect, partly because they're all subject to the same fundamental challenges - namely: 1) the forces and interactions that govern market activity are exceedingly complex; and 2) it's nearly impossible to enumerate all potential outcomes for the future *and* assign probabilities to each of them with high confidence, unless one greatly restricts the granularity of those outcomes (e.g., looking only at whether a stock's future price will be above or below its price today). The chief differentiator among S&ST approaches is how they address these two problems.

There are two main branches of S&ST methods: deterministic and stochastic. These branches differ in the role they assign to randomness; deterministic methods don't include randomness, whereas stochastic methods give priority to it. Essentially, deterministic methods can be seen as

---

[7] The keystone tenets of MPT come from Markowitz's seminal paper, "Portfolio Selection" [1952].

[8] MPT is sometimes referred to as the *mean-variance paradigm*. Many of the most popular performance metrics, such as Sharpe ratios, are rooted in MPT's (false) assumption that means and variances are pretty much all that investors care about.

being based on chains of 'hard-coded' (and possibly parameterized) *if-then* relations (which are assumptions), along the lines of "the S&P falls 2% for every 1% increase in inflation". Clearly, specifying these if-then relations is burdensome, and analytical error is likely to increase as more of these assumptions are added. However, assuming too few relationships handicaps an analysis, by reducing the specificity with which it's able to answer questions on risk and return. Further, most types of deterministic S&ST fluster attempts to assign probabilities to outcomes, since they don't take account of their own possible error.[9]

Stochastic methods attempt to resolve some of these deficiencies by granting a primary role to randomness: rather than assigning single values to key variables, stochastic methods assign them probability distributions. Values from these distributions are then drawn at random to generate potential outcomes. With many draws, it becomes feasible to characterize the distribution of potential outcomes - which facilitates answering questions on things like the odds of a portfolio underperforming a benchmark, or the range of outcomes that can be expected for a given level of confidence. In general, stochastic methods allow for richer analysis, but they are just as reliant on assumptions as their deterministic counterparts. Indeed, stochastic approaches often require more (and more complicated) assumptions than deterministic methods do. For instance, a deterministic model with a single variable requires only one value for that variable (per outcome explored); yet a stochastic model would require not only an assumption for the type of probability distribution for that variable (Gaussian, Pareto, hypergeometric, etc.), but also parameters for that distribution (mean and variance, in the case of the Gaussian distribution). That said, stochastic methods are what many professional investors commonly choose when conducting S&ST.

For many such investors, Monte Carlo (MC) is the darling child of stochastic S&ST methods: it's often presumed to be the default gold-standard - even though its supremacy is very much up for debate (as we'll soon discuss). In the Appendix, we give a simplified sketch of how MC is often used in simulating future portfolio behavior (this is only one implementation: there are other ways to set up this simplified MC application, but they're all fundamentally the same). Even for

[9] As an extreme example, consider one of the simplest deterministic methods: looking into the past and finding the period where market conditions most closely resemble those expected for the future horizon that's being simulated and/or stress-tested. From that single segment of history, it's not really possible to extract relevant probabilities, such as the likelihood of a portfolio underperforming its benchmark.

this ‘bare bones’ implementation, there are many *explicit* assumptions: an expected return for each stock in the portfolio; a co-variance for each pair of stocks; and a sampling distribution for each stock. In addition, there are numerous *embedded* assumptions - e.g., no correlation between returns across time steps. Formulating explicit assumptions and apprehending the embedded assumptions can be an onerous task, even in this simple version of MC. More intricate (and realistic) variations have even higher assumptions-intensity. And these assumptions are crucial: the quality of the simulation/stress-test is directly dependent on the quality of these assumptions.

Assumptions aside, there are other difficulties with MC. A not insignificant one is explaining both the procedure and its outputs to clients. Another challenge is insightfulness; even if a MC analysis manages to yield high-level statistics (e.g., expected overall return) that are acceptable, it isn’t necessarily true that the simulated returns-paths it generates are representative of actual likely paths. For example, many MC approaches assume that random draws are independent, and that a draw at one time step doesn’t affect subsequent draws. But this is unrealistic; many assets exhibit momentum effects, and other types of autocorrelation. Likewise, assets often behave differently in distinct market contexts; for instance, expected stock returns and covariances may be different in times of high and low inflation (a case of conditional covariance). Because MC methods often neglect these kinds of market dynamics, the insights that MC analysis actually provides may be limited, or even invalid.

Another challenge brought on by MC analysis is its time intensity. MC techniques only work (when they do work) through the power of volume: they require thousands, and often millions, of random samples to yield passable results. However, unless one is using advanced hardware or specially optimized algorithms, it can take many minutes, or even hours, to process all of those random observations.[10] Particularly in exploratory analyses, when quick turnaround times are valuable (to decide whether some novel idea is worth pursuing or not), MC can be prohibitively slow. In aggregate, these challenges wedge MC users between a proverbial rock and hard place:

[10] We encountered this problem in conducting the analysis that underpins this Primer. To get a sense of the differences in run-time for MC approaches and the HBML approach, we started with un-parallelized algorithms, so that the MC algorithm performed its random sampling one draw at a time (which is the standard format in which such algorithms are presented in textbooks, academic papers, tutorials, etc.). Most tests with HBML took only a few seconds on a standard laptop setup, whereas the MC tests often took an hour or more. By using parallelization (along with some tricks for matrix decomposition) we were able to reduce the MC run-time to the scale of seconds - but it was never faster than HBML.

ignoring them reduces analytical realism and quality; but addressing them demands additional (and usually hard-to-formulate) assumptions.

To resolve these challenges, we've invented an alternative approach, the history-based multi-lens (HBML) method, which we describe in the next section.

APPROACH

The driving force behind HBML is the fact that history is a rich source of assumptions. In fact, most MC and other stochastic approaches extract many of their (explicit) assumptions from historical data: for example, means, variances, and covariances of returns are often taken directly from empirical price histories. Yet historical time-series contain far more information than just periodic returns. They also capture (among other things):

- How the distribution of an asset's returns tends to change in response to various market and economic backdrops;
- How an asset's returns distribution tends to change according to its recent price history (i.e., autocorrelation effects, which can vary with market and economic conditions); and
- How covariances between returns on different assets change under distinct market and economic conditions.[11]

In essence, by looking only at distributions of returns (and statistics from those distributions), rather than time-series themselves, many stochastic methods throw away valuable contextual information.[12] HBML retains that information, and attempts to augment it, by segmenting it into *episodes*. Different segmentations amount to looking at time-series through distinct *lenses*; and applying multiple lenses can yield penetrating insights (in the sense that risk and return metrics can be applied to episodes after a time-series has had one or more lenses applied to it). Let's get more concrete on what we mean by all of this, by describing the three lenses we find most useful. (Visual representations of these lenses appear in Figures 1-3.)

- Rolling window - wherein time-series are segmented into fixed-length windows (e.g., 24 months), and the start of each window is staggered (so, for a time-series of monthly data

[11] Covariances here are essential. Omitting covariance relationships during simulation can massively distort results for portfolio-level analyses.

[12] HBML as we describe here is effectively a deterministic S&ST method, but one in which the 'if-then' relations are endogenous, and don't need to be externally specified. In the Coda, we mention some ways in which HBML can be enriched with elements of randomness.

for 1972-2022, the first window might begin in January 1972 and end in December 1974; then, the second window would run from February 1972 to January 1975, and so on, such that the final window would run from January 2020 to December 2022). Each window is an episode under this lens.[13]

Figure 1: Example of Rolling-Window Lens - Monthly Returns Split into 12-Month Windows

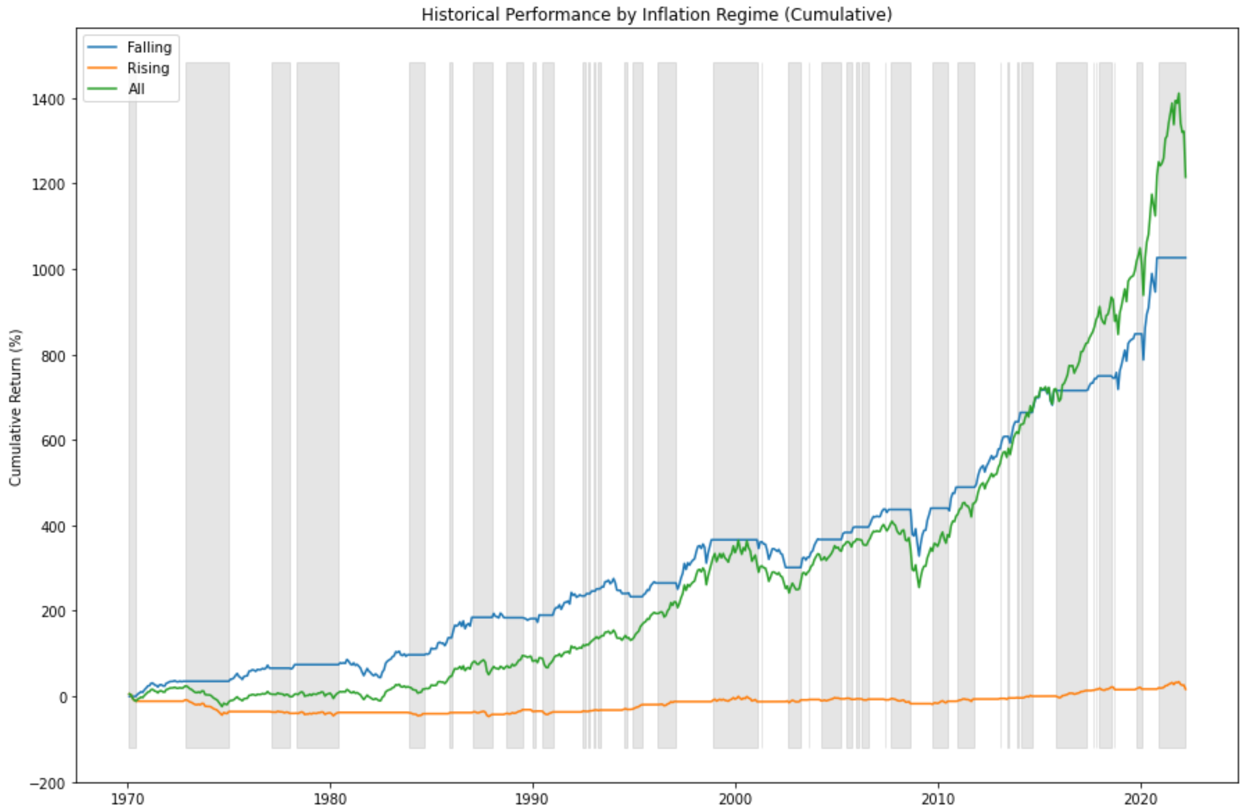
PERIOD	1	2	3	4	5	6	7	8	9	10	11	12
1970-02-01	0.055213	-0.011349	-0.085682	-0.044625	-0.026855	0.050657	0.023969	0.031131	-0.014121	0.052808	0.034642	0.041856
1970-03-01	-0.011349	-0.085682	-0.044625	-0.026855	0.050657	0.023969	0.031131	-0.014121	0.052808	0.034642	0.041856	0.007577
1970-04-01	-0.085682	-0.044625	-0.026855	0.050657	0.023969	0.031131	-0.014121	0.052808	0.034642	0.041856	0.007577	0.044427
1970-05-01	-0.044625	-0.026855	0.050657	0.023969	0.031131	-0.014121	0.052808	0.034642	0.041856	0.007577	0.044427	0.003521
1970-06-01	-0.026855	0.050657	0.023969	0.031131	-0.014121	0.052808	0.034642	0.041856	0.007577	0.044427	0.003521	-0.031743
...	...	...	...	...	...	...	...	...	...	...	...	...
2021-01-01	-0.006619	0.004856	0.008118	0.033451	0.004856	0.021848	0.015963	0.015644	-0.033878	0.039297	-0.004337	0.015742
2021-02-01	0.004856	0.008118	0.033451	0.004856	0.021848	0.015963	0.015644	-0.033878	0.039297	-0.004337	0.015742	-0.046756
2021-03-01	0.008118	0.033451	0.004856	0.021848	0.015963	0.015644	-0.033878	0.039297	-0.004337	0.015742	-0.046756	-0.01464
2021-04-01	0.033451	0.004856	0.021848	0.015963	0.015644	-0.033878	0.039297	-0.004337	0.015742	-0.046756	-0.01464	0.001802
2021-05-01	0.004856	0.021848	0.015963	0.015644	-0.033878	0.039297	-0.004337	0.015742	-0.046756	-0.01464	0.001802	-0.075295

Source: Addepar Research
Notes: Figure depicts 12-month episodes that result from splitting a time-series of monthly excess returns on a portfolio composed of 60% stocks and 40% bonds.

- Regime - wherein a time-series is segmented into intervals that correspond to particular economic or market conditions (e.g., rising and falling interest rates). In the usual case, those intervals from the same regime are ‘stitched together’ - for example, if one uses rising and falling interest rates as the two regimes, there’d be two resulting time-series: one constructed from intervals of rising rates, and the other from intervals of falling rates. Under this lens (as we typically use it), these regime-specific time-series are the episodes. The regime lens is especially useful in studying asset behavior under conditions that haven’t been experienced in recent times, but have occurred at points further back in history; as such, the regime lens can help investors better understand what to expect from unfamiliar market situations that they themselves haven’t directly encountered before.

[13] Many investors look to stress-testing to probe how portfolios might perform in adverse scenarios. However, it can be hard to know which scenarios to test (out of the infinitely many that might be imagined). Rolling windows are excellent candidates for scenarios, especially those that contain extreme market conditions.

Figure 2: Example of Regime Lens Using Rising and Falling Inflation as Regimes

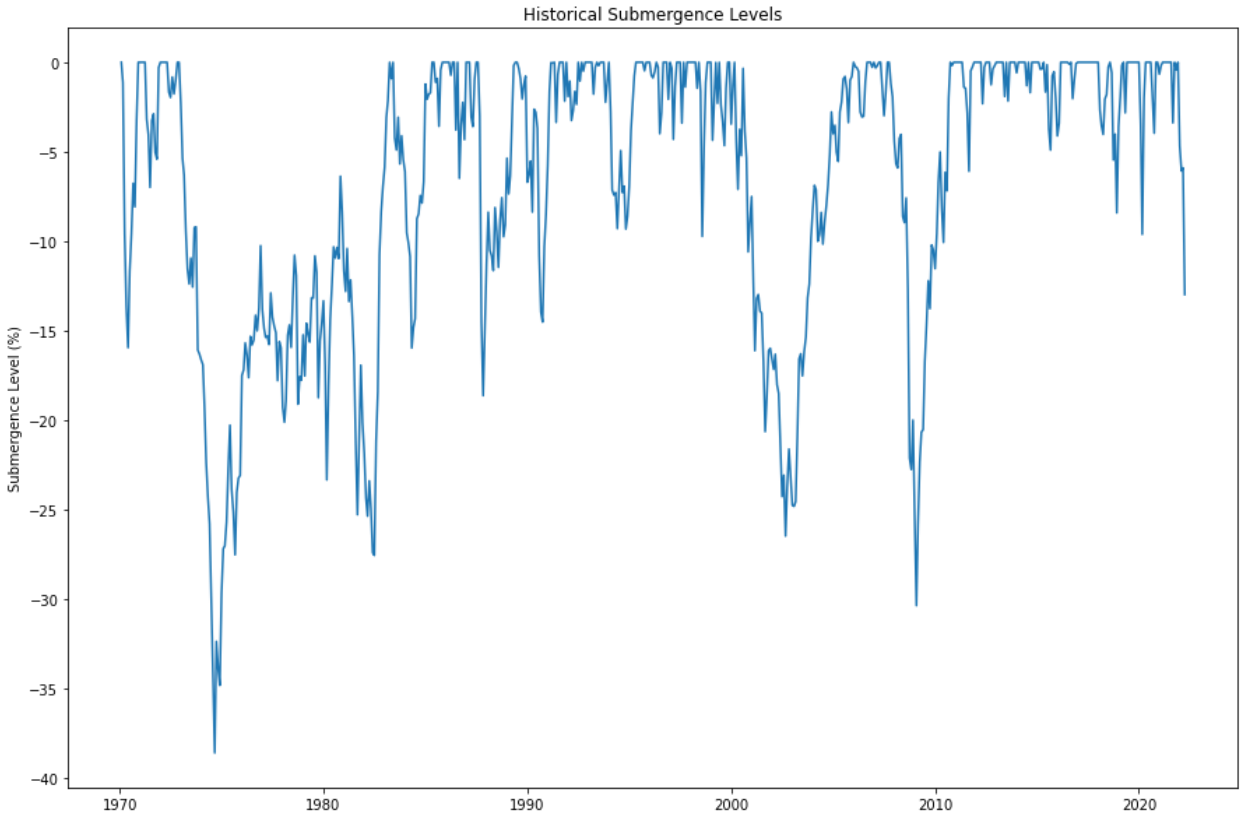


Source: Addepar Research
Notes: Figure depicts cumulative excess returns on a portfolio composed of 60% stocks and 40% bonds, where returns are split into two regimes: returns that are accumulated in periods of rising inflation; and returns that are accumulated in periods of falling inflation. Cumulative excess returns across both regimes ('All') are included for reference. Gray vertical bands indicate periods during which inflation is rising.

- Submergence - wherein a time-series is segmented into distinct submergence events, and the periods of growth that intervene.[14] So, under this lens, the submergence events are the episodes.[15]

[14] Elsewhere (see Monk and Rook [forthcoming] and Rook et al. [2021]), we've defined a submergence as being a drawdown *plus* the recovery that follows. So, a submergence event is a sub-interval of a time-series in which the price of some asset is some specified amount below its running high. Most time-series will contain several such sub-intervals, and each can be treated as a separate submergence event. Sub-intervals that aren't submergence events can be treated separately as well (we call these *growth events*). Essentially, applying this lens amounts to converting a time-series from 'return-space' to 'submergence-space'.
[15] Like rolling windows that encompass extreme market conditions, submergence events can also serve as excellent 'scenarios' for stress tests (whether using the HBML method, or other techniques).

Figure 3: Example of Submergence Lens, Showing Historical Submergence Levels



Source: Addepar Research
Notes: Figure depicts degree of submergence for a portfolio that's composed 60% of stocks and 40% of bonds. Submergence levels are based on running highs for monthly excess returns, based on a start date of January 1970.

We've called our method 'multi-lens' not merely because multiple lenses can be applied, but also because it's most powerful when multiple lenses are applied *simultaneously* - for instance, using both the rolling-window and submergence lenses together to probe a portfolio's behavior during severe drawdowns.

Applying each of these lenses requires a long time-series that stretches sufficiently far back into history (usually multiple decades, at a minimum), and reflects the returns dynamics of assets in the portfolio being simulated or stress-tested.[16] These lengthy histories aren't always available; yet when they are, HBML can work by 'back-testing' portfolios through relevant slices of the past.

---

[16] In some cases, long return histories aren't available for specific portfolio assets, and suitable proxies must be used.

Strong advantages of this approach to S&ST include the facts that: 1) returns distributions don't need to be specified; 2) autocorrelations are automatically accounted for; and 3) covariances don't need to be separately assumed (regardless of whether they're conditional of unconditional). Each of these 'assumptions' is already embedded within the data itself. The only true lynchpin assumption behind HBML is that the past is (to some extent) representative of the distribution of possible outcomes for the future. This is a credible assumption when inputted time-series are very long, and the window for simulation and testing isn't especially lengthy.[17]

While these theoretical advantages of HBML are appealing, it's vital to ask how well HBML performs in practice, especially in comparison with other S&ST methods. In the next section, we test the validity of HBML, both by itself and in relation to Monte Carlo.

FINDINGS

Here, we test the performance of HBML against MC, and use two portfolios to do so:

- A *simple* portfolio that's 60% stocks and 40% bonds;[18]
- A *complex* portfolio that's 40% US stocks, 10% international stocks, 10% commodities, 10% corporate bonds, 10% municipal bonds, and 20% US treasuries.[19]

We based our analysis on monthly excess (net of a risk-free rate) total (inclusive of dividends, coupons, etc.) returns, and assume that each of these portfolios is rebalanced monthly.

Let's begin by applying the rolling-window lens to the simple portfolio, for a window-size of 12 months. Doing so yields a sample of 616 *episodes* for HBML, each of which can be thought of as a 'run' from a historical simulation. Together, these episodes yield various meaningful statistics (as reflected in Figure 4), for example:

---

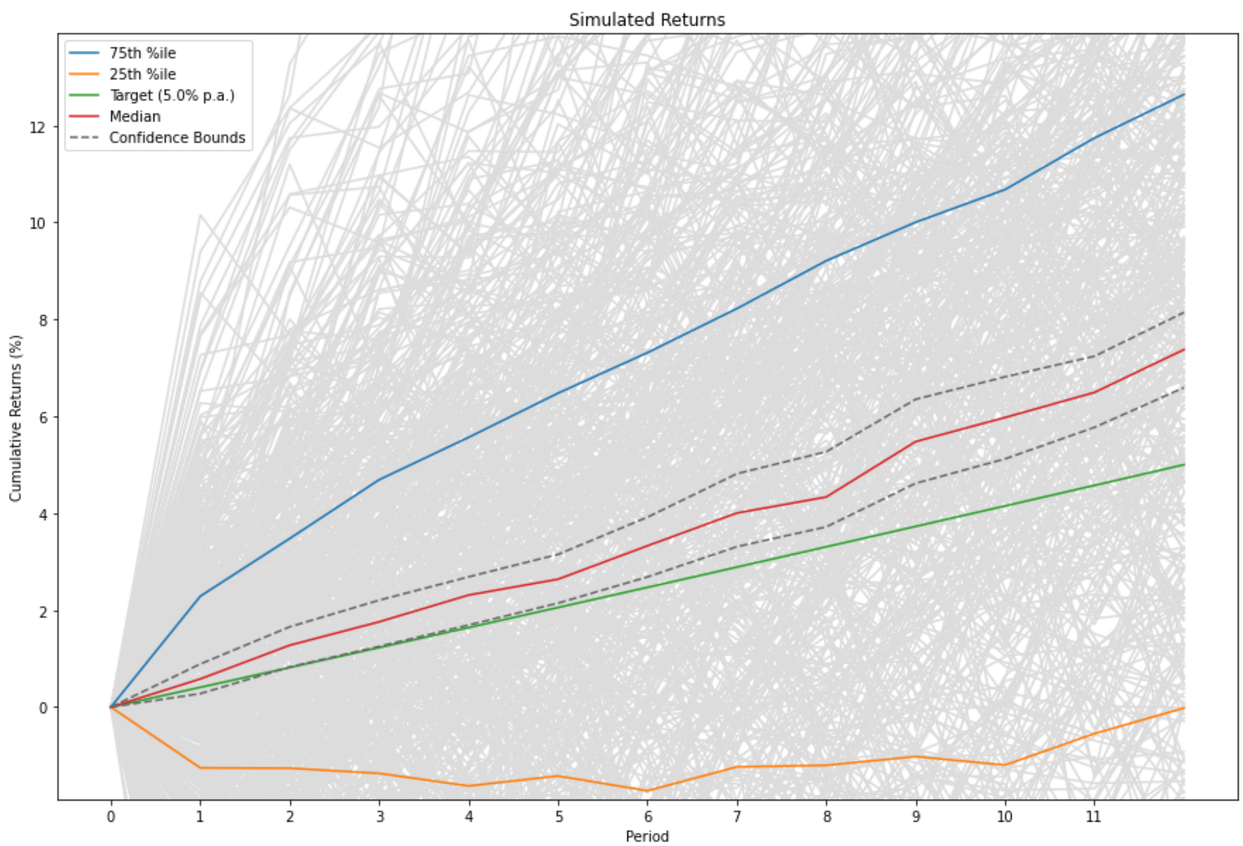
[17] Note, however, that no assumptions at all are needed if one treats HBML as a tool for extracting insights from past tendencies - insights that can be used in probing expectations for future risk and return (after all, such explorations are the very point of S&ST).

[18] As proxies for stocks and bonds, we use monthly excess returns on the XXX and YYY indices, respectively, from January 1970 through April 2022.

[19] As proxies for assets in the complex portfolio, we use monthly excess total returns on the following indices from January 1980 through November 2022: S&P-500 (US equities); the MSCI World-ex-US equities index (international stocks); the S&P/Goldman Sachs Commodities index (commodities); the ICE BofAML US Corporate Bond index (corporate bonds); the Bloomberg Municipal Bond index (municipal bonds); and the Bloomberg US Treasury index (US treasuries).

- The annualized mean return is 5.96%, while the annualized median return 7.37%;[20]
- The cutoff for top-quartile 1-year returns is 12.6%, and the cutoff for the bottom quartile is essentially a null return: -0.02%;
- Over the course of 12 months, the simple portfolio has a 60.2% chance of beating a 5.0% annualized return target.[21]

Figure 4: Simulated Performance of a 60/40 Portfolio with 12-Month Rolling Windows



Source: Addepar Research
Notes: Figure depicts 12-month episodes that result from splitting a time-series of monthly excess returns on a portfolio composed of 60% stocks and 40% bonds.

Undoubtedly, statistics like these are valuable to many investors, because they convey how the portfolio would have performed in 616 distinct, inherently realistic (in the sense that they *truly*

[20] Figure 4 shows a 99% confidence interval around the monthly median return. This interval can easily be derived under HBML through bootstrapping approaches (we give a brief overview of bootstrapping in the Appendix). The proper interpretation of this interval is that, given the data on which HBML is based, there's a 99% chance that the true value for the median return falls inside that interval.
[21] This probability is a straightforward frequentist probability - i.e., it's the fraction of episodes in which year-end returns are greater than 5.0%.

occurred) scenarios. But how useful are these scenarios from a predictive standpoint, i.e., can they be relied upon to accurately indicate future performance? To answer this, we make use of an iterative evaluation method (details of which appear in the Appendix).

Let's start with HBML's ability to forecast success rates (i.e., the likelihood of a portfolio beating some target over a specified horizon). For the 60/40 portfolio with 12-month windows and a 5% return target, HBML has a 4.1% median error, which means that 50% of the time, there's a 4.1% or less difference between HBML's prediction of the success rate and the actual success rate in a test sample of 12-month windows (using 1,000 iterations of our validation method). Bear in mind that the version of HBML we're exploring here is basic, and there are extensions of the method that can generate higher predictive capacity. Our main reason for using this bare-bones version of HBML is that it allows a fairer comparison with bare-bones Monte Carlo; and, in this specific case the median error for HBML is less than half of that for MC (which is 8.4%, using 100,000 MC runs). Indeed, HBML was more accurate than MC 74.2% of the time in our validation tests, for this 60/40, 12-month case with a 5% performance target. We observe similar results over both shorter (6 month) and longer (24, 60, 120 month) horizons, as is shown in Figure 5. That figure also displays results for both lower (3%) and higher (7%, 15%) returns targets, for which we note the higher accuracy of HBML is general, although not universal: for a target annualized return of 7%, MC is more accurate than HBML over 24 and 60-month horizons. However, even in these instances, MC's outperformance is only slight: in the 60-month case (for a 7% return target), the median error for HBML is 4.13%, versus 3.87% for MC; and in the 24-month case (again, for a 7% return target), the median error for HBML is actually slightly less than that for MC (4.76%, versus 4.80%). The key takeaway here is that, in our tests, HBML usually outperformed MC at point predictions for success rates.[22]

[22] It's worth noting here that prediction of success rates has importance not only prior to investment: once a portfolio (or asset) is owned, predicting success rates during the ownership cycle can be just as important. For example if one owns a stake in a private equity fund that is underperforming, knowing the probability of still achieving a target rate of return can be a crucial determinant in whether or not it's advisable to sell that stake on a secondary market, rather than hold it until the end of the fund's life.

Figure 5: Frequencies with which HBML Outperforms MC at Predicting Success Rates for 60/40 Portfolio Performance, for Various Horizons and Returns Targets

	Target Return (Annualized)			
Horizon	5%	3%	7%	15%
6 months	61.8%	64.1%	60.4%	61.7%
12 months	74.2%	73.0%	71.0%	68.0%
24 months	63.5%	60.6%	34.2%	70.6%
60 months	60.5%	55.7%	21.9%	98.7%
120 months	70.2%	69.9%	66.1%	100.0%

Source: Addepar Research
Notes: Results are based on 1,000 iterations of our validation testing procedure (described in the Appendix) for rolling-windows of excess monthly returns on a 60/40 portfolio. Monte Carlo forecasts are based on 100,000 runs.

That said, point predictions aren't all investors should care about in simulation and stress-testing, and a good S&ST method should be able to reliably forecast intervals within which statistics of interest - such as success probabilities - are likely to fall, with a stipulated degree of confidence. Using a bootstrapping method (which we explain in the Appendix), we find that HBML predicts (with 99% confidence) that the chance of the simple 60/40 portfolio earning at least a 5% return over 12 months is between 55.2% and 65.2% (this is equivalent to saying the predicted success rate is 60.2% $\pm$ 5.1%).[23] How accurate is this prediction? According to our iterative validation procedure (using 10,000 iterations), it's 57.5% accurate: i.e., more than half the time, the actual success rate falls within the interval predicted by HBML.

How does this level of accuracy compare with Monte Carlo? The answer depends on the number of runs one performs for MC, which is effectively a procedural assumption chosen by the user.[24] But it proves to be a tricky and consequential assumption that deserves a brief discussion. Often, there's an impression that "more runs is better" when it comes to MC, and that the only downside to conducting more runs is the added compute time (it's common to encounter analyses that use

[23] 99% is a usual confidence level in many applications, and we use it throughout this Primer.
[24] The number of MC runs is a parameter choice that's distinct from the choice of a confidence level. For HBML, there's no need to choose the number of runs.

millions of MC runs without justifying that amount).[25] But this more-is-better logic is unfounded, particularly when it comes to generating confidence intervals: for the case at hand, increasing the number of runs decreases the accuracy of the confidence intervals that MC generates (holding confidence level fixed). Phrased another way: using too many runs can result in ‘over-confident’ confidence intervals, as Figure 6 indicates.[26]

Figure 6: Hit Rates and Confidence Intervals for Various Monte Carlo Run Quantities

	100 Runs	1,000 Runs	10,000 Runs	616 Runs
Hit Rate	72.9%	24.5%	6.8%	30.3%
Confidence Interval	39.8% - 64.9%	48.4% - 56.4%	51.1% - 53.6%	47.3% - 57.5%

Source: Addepar Research
Notes: Results are based on 10,000 iterations of our validation testing procedure for 12-month rolling-windows of excess monthly returns on a 60/40 portfolio. Results assume a 5.0% target return and 99% confidence level. Confidence intervals were constructed by use of bootstrapping.

Even at the level of 10,000 runs (which is 1-2 orders of magnitude fewer than is frequently used for MC analysis), MC leads to very narrow confidence intervals, which can give false comfort in the accuracy of its predictions, as evidenced by the low hit rate - that is, the frequency with which the actual rate of success (probability of the portfolio reaching its benchmark return) falls inside the confidence interval that MC forecasts. The rightmost column of Figure 7 shows the hit rate for 616 runs, which is equal to the number of episodes that HBML uses in forecasting confidence intervals (CIs) for a 12-month horizon (for this particular case and dataset). From our validation testing, the average CI length for HBML in this example is 10.6%, which is very close to the observed length of 10.2% for MC with 616 runs. This phenomenon (similar-sized CIs for MC and HBML, when the number of MC runs matches the number of episodes HBML uses to make its predictions, with confidence level kept constant) holds over various horizons, as Figure 7 shows. Matching the number of MC runs to the number of HBML episodes allows for a more ‘apples-to-apples’ comparison of the accuracy of CIs created by MC and HBML, respectively. In doing this, we can observe (in Figure 7) that HBML consistently outperforms HBML, in terms of its hit rate in our validation tests.

[25] There are a significant number of peer-reviewed academic papers that derive the optimal number of runs for MC under various conditions. However, those findings seem to be rarely used in practice.
[26] [Here again we use bootstrapping for generating confidence intervals; Shapiro tests indicated non-normality of the success probabilities predicted by MC]

Figure 7: Average Hit Rates and CI Lengths for HBML and MC (60/40 Portfolio)

		Hit Rate		Confidence Interval Length	
Horizon	Runs	MC	HBML	MC	HBML
6 months	622	48.7%	58.1%	10.1%	10.7%
12 months	616	30.3%	57.5%	10.2%	10.6%
24 months	604	52.6%	59.3%	10.3%	10.8%
60 months	568	48.9%	59.6%	10.6%	11.2%
120 months	508	39.8%	56.6%	11.2%	11.5%

Source: Addepar Research
Notes: Results are based on 10,000 iterations of our validation testing procedure for excess monthly returns on a 60/40 portfolio. Results assume a 5.0% target return and 99% confidence level. Confidence intervals were constructed by use of bootstrapping.

The foregoing are all tests of accuracy for a 5.0% target return. Do things change if we switch to different targets? Figure 8 shows that the answer is both yes and no.[27] Generally, HBML has a higher hit rate than MC, with similar-sized confidence intervals (when confidence level is fixed at 99%, and the number of runs for MC matches the number of HBML episodes). In some cases, HBML vastly outperforms MC; although this outperformance isn't universal - there are cases in which MC has a higher hit rate than HBML (yet in those instances the performance gap between the two is fairly negligible). What's interesting to note here is how consistent HBML's hit rate is relative to that for MC.

Figure 8: Outperformance of HBML (Relative to MC) in Predicting Success Rates for 60/40 Portfolio, under Various Returns Targets

		Target Return (Annualized)		
Horizon	Runs	3%	7%	15%
6 months	622	+9.2% (57.8%)	+3.8% (57.2%)	+12.5% (58.9%)
12 months	616	+25.0% (60.2%)	+20.2% (59.4%)	+13.1% (60.9%)

[27] We chose to display values here that are both slightly below and above the 5% level (i.e. 3% and 7%), as well as significantly above the 5% mark (i.e. 15%). Results weren't significantly different from what appears here for the many other target return values we tested.

24 months	604	+6.6% (56.2%)	+1.3% (58.7%)	+26.5% (64.9%)
60 months	568	-0.2% (55.5%)	-2.9% (54.3%)	+7.8% (81.3%)
120 months	508	+17.0% (57.3%)	+8.8% (53.5%)	+9.3% (100.0%)

Source: Addepar Research
Notes: Values in parentheses indicate hit rates for HBML. Results are based on 10,000 iterations of our validation testing procedure for 12-month rolling-windows of excess monthly returns on a 60/40 portfolio. Results assume a 99% confidence level. Confidence intervals were constructed by use of bootstrapping.

The outperformance of HBML - in terms of predicting success rates and their CIs - is perhaps surprising, given how few assumptions HBML requires (as well as its mechanical simplicity), relative to MC. However, success rates aren't the only statistic that investors care about when it comes to S&ST. They will usually also be interested in (e.g.) a portfolio's expected mean return, volatility, and Sharpe ratio. Figure 9 shows the frequency with which HBML outperforms MC in terms of its ability to accurately predict these statistics in our validation tests, over a variety of horizons (once again, using the simple 60/40 portfolio).[28]

Figure 9: Frequencies with which HBML Outperforms MC at Predicting Various Performance Metrics for a 60/40 Portfolio, over Various Horizons

	Mean	Volatility	Sharpe Ratio
6 months	47.6%	65.1%	50.2%
12 months	44.5%	61.8%	55.2%
24 months	50.5%	58.1%	55.4%
60 months	53.0%	49.6%	48.0%
120 months	53.1%	62.5%	53.4%

Source: Addepar Research
Notes: Results are based on 1,000 iterations of our validation testing procedure for 12-month rolling-windows of excess monthly returns on a 60/40 portfolio. Monte Carlo results are derived from 100,000 runs each.

[28] We use 1,000 validation tests, and 100,000 MC runs. Unlike the previous case of success rates, this is a situation in which 'more runs is better' for the accuracy of MC results. 100,000 runs was the level at which results began to stabilize - that is, further increasing the number of runs failed to produce significantly different values, and simply added to compute time.

What's noticeable is that, overall, MC and HBML perform about equally well at predicting mean portfolio returns, but HBML is a better predictor of volatility, which allows it to be slightly better at forecasting risk-adjusted returns (as measured through Sharpe ratios). The outperformance of HBML in forecasting volatility seems to be a direct result of HBML using *actual* sequences of volatility, rather than inducing volatility at the portfolio level via assuming covariances and standard deviations at the asset level (as MC does). Use of actual sequences allows HBML to retain volatility information that gets discarded by MC, such as autocorrelation effects, which can be significant drivers of both risk and risk-adjusted returns.[29]

One might wonder whether all of these results are being influenced by the simplicity of the test portfolio - perhaps a more sophisticated portfolio might favor MC. Yet, re-performing the above tests with our more complex portfolio (described at the start of this section) shows that isn't so. Figures 10 through 13 below show outcomes of these re-performed tests, and it can be seen that increasing portfolio complexity appears to increase HBML's outperformance, relative to MC.

<u>Figure 10: Frequencies with which HBML Outperforms MC at Predicting Success Rates for a Complex Portfolio's Performance, for Various Horizons and Returns Targets</u>

	Target Return (Annualized)			
Horizon	5%	3%	7%	15%
6 months	67.0	63.7	55.1	63.4
12 months	57.7	72.8	67.2	69.5
24 months	62.7	81.3	49.1	79.9
60 months	60.4	62.0	51.2	100.0
120 months	54.2	89.4	80.7	100.0

Source: Addepar Research
Notes: Results are based on 1,000 iterations of our validation testing procedure (described in the Appendix) for rolling-windows of excess monthly returns on a 'complex' portfolio (described earlier in this section). Monte Carlo forecasts are based on 100,000 runs.

[29] There are other variants of Monte Carlo that are specially designed to include elements of autocorrelation. These often take the form of algorithms known as Markov Chain Monte Carlo processes (MCMC). However, MCMC is more assumption-laden than plain-vanilla MC.

Figure 11: Average Hit Rates and CI Lengths for HBML and MC (Complex Portfolio)

		Hit Rate		Confidence Interval Length	
Horizon	Runs	MC	HBML	MC	HBML
6 months	509	46.7%	55.7%	11.3%	11.7%
12 months	503	49.6%	55.8%	11.3%	11.8%
24 months	491	44.2%	60.1%	11.4%	11.8%
60 months	455	46.0%	56.4%	11.8%	12.3%
120 months	395	55.6%	59.2%	12.8%	13.5%

Source: Addepar Research
Notes: Results are based on 10,000 iterations of our validation testing procedure for excess monthly returns on a 'complex' portfolio (described earlier in this section). Results assume a 5.0% target return (annualized) and 99% confidence level. Confidence intervals were constructed by use of bootstrapping.

Figure 12: Outperformance of HBML (Relative to MC) in Predicting Success Rates for a Complex Portfolio, under Various Returns Targets

		Target Return (Annualized)		
Horizon	Runs	3%	7%	15%
6 months	509	+15.5% (58.1%)	+0.4% (54.7%)	+7.9% (53.8%)
12 months	503	+20.0% (56.2%)	+5.0% (55.0%)	+19.8% (61.7%)
24 months	491	+34.1% (52.9%)	+3.1% (59.7%)	+17.4% (49.4%)
60 months	455	+7.6% (59.3%)	+1.7% (58.5%)	+81.5% (100.0%)
120 months	395	+49.7% (58.9%)	+35.6% (60.2%)	0.0% (100.0%)

Source: Addepar Research
Notes: Values in parentheses indicate hit rates for HBML. Results are based on 10,000 iterations of our validation testing procedure for 12-month rolling-windows of excess monthly returns on a 'complex' portfolio (described earlier in this section). Results assume a 99% confidence level. Confidence intervals were constructed by use of bootstrapping.

Figure 13: Frequencies with which HBML Outperforms MC at Predicting Various Performance Metrics for a Complex Portfolio, over Various Horizons

	Mean	Volatility	Sharpe Ratio
6 months	45.5%	66.6%	50.7%
12 months	45.5%	65.7%	53.7%
24 months	47.1%	64.9%	54.3%
60 months	48.7%	57.2%	57.5%
120 months	54.8%	63.7%	50.1%

Source: Addepar Research
Notes: Results are based on 1,000 iterations of our validation testing procedure for 12-month rolling-windows of excess monthly returns on a 'complex' portfolio (described earlier in this section). Monte Carlo results are derived from 100,000 runs each.

Our analysis thus far has focused only on the rolling-window lens of HBML. Let's now explore properties of HBML under the regime lens. As an example, we use HBML to segment the simple 60/40 portfolio's performance into two regimes: rising and falling inflation (a graphic illustration of this segmentation appears in the previous section). Figure 14 presents aggregate performance statistics for these two regimes from 1970-2022.[30] Clearly, performance is more appealing during regimes of falling inflation.

Figure 14: Performance Statistics for a 60/40 Portfolio Under Two Different Inflation Regimes

Regime	Return (annualized)	Return (cumulative)	Volatility (annualized)	Sharpe Ratio (annualized)	% of Months in Regime
Rising	1.14%	16.74%	10.23%	0.11	48.6%
Falling	9.57%	1,025.89%	10.13%	0.94	51.4%
All	5.47%	1,214.41%	10.24%	0.53	100.0%

Source: Addepar Research
Notes: Results are based on excess monthly returns on a portfolio that is composed of 60% stocks and 40% bonds.

These statistics are interesting from a historical standpoint, but investors may be more interested in examining how a portfolio will likely behave over specific horizons under each regime. For instance, they might wish to know the likelihood of achieving a particular target return over 24

[30] This is the table-based representation of data in Figure 2.

months, if inflation is rising, on average. Combining the regime and rolling-window lenses can help investors do just that. Figure 15 displays the probability (as calculated by HBML) that the 60/40 portfolio will hit a 5.0% annualized return target over various horizons, when inflation is either rising or falling, on average.[31]

Figure 15: Predicted Success Probabilities for 60/40 Portfolio Under Different Inflation Regimes

Regime	6 months	12 months	24 months	60 months	120 months
Rising	47.3%	51.2%	43.4%	36.6%	46.8%
Falling	65.6%	70.1%	68.5%	68.4%	75.7%

Source: Addepar Research
Notes: Results are based on rolling-windows of excess monthly returns on a portfolio that is composed of 60% stocks and 40% bonds. Regimes of rising inflation are those for which inflation was increasing during at least half of the months in the specified horizon. Regimes of falling inflation are those for which inflation was increasing during less than half of the months in the specified horizon.

As before, HBML can also predict confidence intervals for these success probabilities. It's worth noting that MC can do these things as well: a simple way of allowing it to do so involves using parameters (for expected return and covariance) taken from conditional monthly returns (that is, returns during months in which inflation is either rising or falling). Given that HBML and MC are both capable of predicting regime-specific CIs, one may ask: which method is more reliable?

This question can be answered via the same tests that we used previously. Figure 16 displays the respective hit rates (for a 5% annualized target return) of HBML and MC in forecasting 99% CIs over various time horizons, for regimes in which inflation is either rising or falling, on average. Here, HBML is generally more accurate than MC - and especially so during times when inflation is falling. (We re-performed this analysis for different return targets, and the results were similar; to save space, we don't report those results here.) Figure 17 demonstrates that moving to a more complex portfolio doesn't seem to alter this conclusion - although it does change the accuracies of both HBML and MC (more so for the latter).

[31] Here, "on average" refers to the fraction of months in which inflation is rising or falling. For simplicity, we count an episode as one of rising inflation if inflation is rising in at least half of the months it contains; otherwise we count it as an episode of falling inflation.

Figure 16: Average Hit Rates for HBML and MC (60/40 Portfolio) Under Different Regimes

	Rising Inflation		Falling Inflation	
Horizon	MC	HBML	MC	HBML
6 months	54.3%	55.2%	20.3%	54.1%
12 months	56.3%	54.3%	9.4%	61.2%
24 months	41.9%	58.5%	13.9%	59.5%
60 months	14.9%	56.2%	18.2%	50.3%
120 months	49.0%	57.6%	4.3%%	63.4%

Source: Addepar Research
Notes: Results are based on 10,000 iterations of our validation testing procedure for excess monthly returns on a portfolio that is composed of 60% stocks and 40% bonds. Results assume a 5.0% target return (annualized) and 99% confidence level. Confidence intervals were constructed by use of bootstrapping. Monte Carlo simulations in each horizon use a number of runs that matches the number of episodes for HBML in that horizon. Regimes of rising inflation are those for which inflation was increasing during at least half of the months in the specified horizon. Regimes of falling inflation are those for which inflation was increasing during less than half of the months in the specified horizon.

Figure 17: Average Hit Rates for HBML and MC (60/40 Portfolio) Under Different Regimes

	Rising Inflation		Falling Inflation	
Horizon	MC	HBML	MC	HBML
6 months	57.2%	59.0%	35.9%	59.9%
12 months	39.4%	51.9%	53.4%	50.9%
24 months	41.0%	57.0%	51.1%	50.0%
60 months	50.2%	50.8%	7.3%	58.0%
120 months	5.5%	58.0%	5.7%	51.5%

Source: Addepar Research
Notes: Results are based on 10,000 iterations of our validation testing procedure for excess monthly returns on a 'complex' portfolio (described earlier in this section). Results assume a 5.0% target return (annualized) and 99% confidence level. Confidence intervals were constructed by use of bootstrapping. Monte Carlo simulations in each horizon use a number of runs that matches the number of episodes for HBML in that horizon. Regimes of rising inflation are those for which inflation was increasing during at least half of the months in the specified horizon. Regimes of falling inflation are those for which inflation was increasing during less than half of the months in the specified horizon.

It's also possible to create horizon-specific predictions for returns, volatility, and Sharpe ratios under the different regimes. Figures 18 and 19 show these for the 60/40 and complex portfolios, respectively. Yet again, HBML seems slightly better than MC in predicting volatility and Sharpe ratios (this outperformance appears stronger for Sharpes in regimes of falling inflation).

Figure 18: Frequencies with which HBML Outperforms MC at Predicting Various Performance Metrics for a 60/40 Portfolio, over Various Horizons (Under Different Inflation Regimes)

	Rising Inflation			Falling Inflation		
Horizon	Mean	Volatility	Sharpe	Mean	Volatility	Sharpe
6 month	49.9%	64.7%	54.4%	57.6%	64.3%	58.4%
12 month	46.8%	64.3%	44.9%	60.2%	55.7%	60.6%
24 month	55.7%	55.6%	48.3%	56.2%	57.7%	61.8%
60 month	63.5%	50.2%	60.7%	59.5%	44.7%	56.0%
120 month	54.1%	83.3%	48.6%	51.8%	49.3%	58.9%

Source: Addepar Research
Notes: Results are based on 1,000 iterations of our validation testing procedure for 12-month rolling-windows of excess monthly returns on a portfolio that is composed of 60% stocks and 40% bonds. Monte Carlo results are derived from 100,000 runs each. Regimes of rising inflation are those for which inflation was increasing during at least half of the months in the specified horizon. Regimes of falling inflation are those for which inflation was increasing during less than half of the months in the specified horizon.

Figure 19: Frequencies with which HBML Outperforms MC at Predicting Various Performance Metrics for a Complex Portfolio, over Various Horizons (Under Different Inflation Regimes)

	Rising Inflation			Falling Inflation		
Horizon	Mean	Volatility	Sharpe	Mean	Volatility	Sharpe
6 month	47.1%	66.8%	49.2%	56.6%	63.7%	54.6%
12 month	47.4%	63.9%	54.5%	49.3%	62.4%	53.8%
24 month	50.4%	59.3%	56.3%	46.7%	66.3%	57.7%
60 month	53.9%	55.6%	46.1%	58.5%	56.2%	71.7%
120 month	68.9%	50.9%	57.1%	67.4%	73.9%	70.5%

Source: Addepar Research
Notes: Results are based on 1,000 iterations of our validation testing procedure for 12-month rolling-windows of excess monthly returns on a 'complex' portfolio (described earlier in this section). Monte Carlo results are derived from 100,000 runs each. Regimes of rising inflation are those for which inflation was increasing during at least half of the months in the specified horizon. Regimes of falling inflation are those for which inflation was increasing during less than half of the months in the specified horizon.

We hasten to note that HBML can handle far more regimes than inflation: the regime lens works with any dataset that allows a segmentation of history. In fact, sometimes no additional datasets (apart from historical returns) are even needed to apply the regime lens. For example, an investor might be curious about returns to the 60/40 portfolio under different volatility regimes. Applying this lens requires only returns data and an ex ante application of the rolling-window lens (which can be used to split history into rolling, fixed-length episodes, from which volatility quartiles can be calculated, and episodes can be matched to quartiles). Figure 20 shows the annualized returns predicted by HBML for 'high' and 'low' volatility regimes, for various horizons (where high and low correspond to top-quartile and bottom-quartile volatility, respectively).

Figure 20: Annualized Returns Predicted by HBML under Different Volatility Regimes

	Horizon				
Regime	6 months	12 months	24 months	60 months	120 months
High Volatility	2.3%	3.7%	2.4%	3.9%	3.9%
Low Volatility	8.8%	8.5%	7.8%	8.1%	7.3%

Source: Addepar Research
Notes: Results are based on rolling-windows of excess monthly returns on a portfolio that is composed of 60% stocks and 40% bonds. Predictions for the high volatility regime come from rolling-window episodes that had top-quartile volatility, historically. Predictions for the low volatility regime come from rolling-window episodes that had bottom-quartile volatility, historically.

The final HBML lens we'll cover here is the submergence lens (recall from the previous section that a submergence is an event made up of a drawdown and the subsequent recovery, such that the value of a portfolio remains 'submerged' below its running high throughout the event). The submergence lens is useful for answering questions about risk that aren't specifically rooted in volatility. For example: what fraction of time can a portfolio's value be expected to be 10% or more below its running high? The answer for our simple 60/40 portfolio (and underlying dataset) is 29.7% of the time. These kinds of insights can be significant in managing risk and liquidity.

Yet the submergence lens becomes even more powerful when coupled with the rolling-window lens. This combination easily permits answers to questions such as: what is the probability that a given portfolio will be at least 10% below its starting value at some point during a 1-year period? Per HBML, the answer for a simple 60/40 portfolio is 16.9%.[32] Additionally, HBML is able to predict CIs for this probability: 13.1% to 20.8% with 99% confidence (that is, 16.9% $\pm$ 3.9%).[33]

[32] This is based on monthly granularity in measurement.
[33] This calculation uses 10,000 bootstrapped resamples.

Once more we ask: how accurate are these CIs relative to those MC is able to generate? Figure 21 shows the hit rates for MC and HBML in predicting CIs for the chance that a 60/40 portfolio falls 5%, 10%, or 20% below its starting value over horizons of various lengths. Figure 22 shows the same analysis for the more complex portfolio (the same one we've studied above).[34] Overall, we see that HBML tends to perform better than MC - and more so for over longer horizons and for larger levels of decline. We take this as further evidence that HBML is able to match MC's performance as a risk-testing engine - and in many cases surpass it.

Figure 21: Hit Rates in Predicting CIs for the Probability that a 60/40 Portfolio Experiences a Specified Decline in Value over a Given Horizon

	Decline in Value over the Specified Horizon					
	> 5%		> 10%		> 20%	
	MC	HBML	MC	HBML	MC	HBML
6 month	53.7%	57.3%	49.2%	53.4%	58.8%	64.1%
12 month	47.5%	58.5%	46.2%	53.5%	18.9%	49.9%
24 month	55.3%	56.7%	42.6%	57.2%	15.6%	57.7%
60 month	42.6%	56.5%	3.8%	54.6%	3.2%	62.0%
120 month	16.7%	57.6%	1.1%	55.6%	5.6%	61.8%

Source: Addepar Research
Notes: Results are based on 10,000 iterations of our validation testing procedure for excess monthly returns on a portfolio that is composed of 60% stocks and 40% bonds. Results assume a 99% confidence level. Confidence intervals were constructed by use of bootstrapping. Monte Carlo simulations in each horizon use a number of runs that matches the number of episodes for HBML in that horizon.

Figure 22: Hit Rates in Predicting CIs for the Probability that a Complex Portfolio Experiences a Specified Decline in Value over a Given Horizon

	Submergence Depth					
	-5%		-10%		-20%	
	MC	HBML	MC	HBML	MC	HBML

[34] In all cases, we match the number of MC runs to the number of HBML rolling-window episodes for the horizon in question. We also use a confidence level of 99%.

6 month	56.2%	56.2%	43.9%	65.9%	53.4%	38.0%
12 month	55.4%	55.3%	52.2%	62.8%	25.4%	61.5%
24 month	55.8%	58.6%	55.4%	56.4%	10.1%	48.5%
60 month	55.7%	57.7%	36.0%	54.3%	3.5%	50.5%
120 month	46.6%	56.3%	24.8%	61.6%	4.9%	50.6%

Source: Addepar Research
Notes: Results are based on 10,000 iterations of our validation testing procedure for excess monthly returns on a 'complex' portfolio (described earlier in this section). Results assume a 99% confidence level. Confidence intervals were constructed via bootstrapping. Monte Carlo simulations in each horizon use a number of runs that matches the number of episodes for HBML in that horizon.

SUMMARY IMPLICATIONS

In the foregoing, we've demonstrated that our novel technique for simulation and stress-testing, the history-based multi-lens (HBML) approach, is competitive with a Monte Carlo approach. Indeed, HBML's performance relative to MC leads us to recommend it as a new best-practice for S&ST - not necessarily to replace MC and related methods, but at least as a supplement to what those methods have to offer. From what we've analyzed thus far, HBML is able to retain a good deal of contextualized information from runs of true history - such as elements of autocorrelation and conditional covariance - that other methods (namely, MC) can struggle to incorporate. Due to this, HBML appears to be a worthwhile addition to any investor's S&ST toolbelt, especially when it comes to exploring risks that go beyond simple volatility.

A point which we've mentioned, but tried not to belabor, is the powerful role that S&ST methods can play in investor's interactions with clients. The universe of investing is rife with uncertainty, and even savvy clients may routinely find themselves desperate for guideposts on what the future holds, both in terms of hazards and opportunities. Appropriate S&ST methods can deliver these guideposts, by providing insights on possibilities and probabilities that, while imperfect, can help both investors and clients familiarize themselves with otherwise unfamiliar situations. In light of the straightforward, intuitive nature of HBML, we think it could go far in boosting the quality of dialogue that investors can have with their clients, when it comes to venturing into new terrain - whether that entails new asset classes, new market environments, or simply new levels of mutual understanding.

CODA

Simulation and stress-testing (S&ST) are clear best-practice activities for any investor, but many popular S&ST methods suffer from a number of deficiencies, such as being assumptions-heavy, opaque, or challenging to explain to clients. In this Primer, we introduced a new S&ST approach, the history-based multi-lens (HBML) paradigm, that is straightforward and assumptions-light, but also capable of furnishing rich insights about the space of probable outcomes for portfolios. Through a suite of tests, we showed that HBML can perform on-par with, or even exceed, the accuracy of standard Monte Carlo techniques. This performance gain promises sharper insights to investors, with a higher dose of confidence.

That said, the version of HBML we've introduced here might be seen as 'bare-bones', as it uses un-modified empirical data, and is largely deterministic (its sole use of randomness comes in the way it generates confidence intervals). There are many extensions to HBML that we're currently pursuing, in order to improve both accuracy and richness. Foremost among these next steps are developing:

- Techniques to incorporate stochasticity, for the sake of expanding datasets in ways that are realistic, but allow HBML to operate when data might otherwise be sparse.
- Approaches that allow HBML to go far 'out-of-sample'. In general, the worst that has occurred isn't the worst that *could have* occurred; (otherwise back-testing would be the only type of stress-test necessary!). We're investigating approaches that let us look farther out into the tails of distributions, to identify extremal outcomes that represent legitimate threats to portfolios, and therefore should factor into HBML analysis - even if they are counterfactual.
- Methods for proxy-ing assets that have short returns histories (e.g., venture capital funds have, from a statistical standpoint, comparatively small datasets compared to stocks or treasuries).

We expect HBML to be a meaningful fixture of our research in coming years.

Nevertheless, some readers may be wondering not so much about future extensions to HBML as present implementations. We've mentioned that HBML can have far shorter runtime and use less computing resources than Monte Carlo methods, but this doesn't detract from the fact that it can sometimes be unclear how to implement HBML in spreadsheets - or if doing so is even feasible for some cases. We've built prototype HBML toolkits in Python, and we're continually refining their versatility and performance profiles. We hope someday soon to make these available to our readership, but - in the meantime - we encourage interested readers to come talk to us about how we're building these things, and what their future capabilities might be.

REFERENCES

Markowitz, H. (1952) "Portfolio Selection", *Journal of Finance*, 7(1):77-91.

Monk, A., and D. Rook (forthcoming) *The Long-Term Investor: Navigating Markets with Resilience and Knowledge*, Stanford University Press.

Rook, D., S-Y. In, and A. Monk (2021) "The Resilience Manifesto: Integrating ESG and Risk Management, in Investment Portfolios and Organizations," *Social Science Research Network*, available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3942909.

APPENDIX

A1) *Example Recipe for Applying Monte Carlo to a Stock Portfolio*

1. For each stock, **assume** an expected return for each time increment (i.e. if the analysis is for daily performance of a portfolio over a 1-year horizon, then expected daily return for each stock in the portfolio would be needed).
2. **Assume** a covariance matrix that describes the co-movements in prices for stocks in the portfolio.
3. **Assume** a sampling distribution for each stock in the portfolio (for example. a Gaussian distribution) that represents the volatility of its returns.
4. For each time step in a single run of the simulation, take a single random draw from the sampling distribution for each stock (from #3), and apply the covariance matrix to this random sample, and then add the resulting values to the expected returns (from #1) and apply portfolio weights.

5. Repeat #4 for each time step. This results in a simulated returns path (for example, one year's worth of simulated daily returns for the portfolio).
6. Repeat steps 4 and 5 many times (often tens or hundreds of thousands of times, if not more).
7. Calculate appropriate statistics (e.g. mean and standard deviation) across all simulated returns paths (usually this is done based on cumulative returns for the simulated horizon).

For our analysis, we follow the MC implementation described in the Appendix, and assume that: mean returns equal the average monthly returns for our example time-series; the variances and covariances for stocks and bonds in the portfolio is the same as what's observed in the time-series; and at each time-step in the 24-month prediction window, two random values are drawn from a 0,1-Gaussian distribution. This implementation is very similar to those that many investors use in practice, because it amounts to drawing returns at every time-step from a multivariate Gaussian distribution.

A2) *Explanation of Our Iterative Validation Procedure*

The validation procedure we use throughout our analysis is based on a common technique used in machine learning: k-fold cross-validation. For each iteration of the procedure, we create a random split in the dataset of returns, whereby 10% of monthly returns are designated as a 'test' set and the remaining 90% are treated as a 'training' set. The training set is used as inputs to the predictions made by HBML and Monte Carlo (respectively), and these predictions are then tested against the test set. Many iterations are run, with a new random split each time, and performance statistics (in terms of the accuracy of predictions for each test set) are then aggregated (typically by taking an average value). In our analysis, we typically ran either 1,000 or 10,000 iterations, depending on the validation case. We use these quantities of iteration because we found them to produce stable results.

A3) *Explanation of Bootstrapped Confidence Intervals*

Throughout our analysis, we make use of bootstrapped confidence intervals. Bootstrapping is a common technique in statistical analysis, and is most commonly used when making assumptions about the distribution of the associated statistic proves challenging. For example, if one is trying to calculate a confidence interval for the median of a sample, but feels uncomfortable making an assumption about the distribution of the median (i.e., the distribution of median values that one could expect if repeated, distinct samples of the same size were re-taken), then bootstrapping is generally an applicable technique, so long as the sample size is sufficiently large (in many cases, a hundred observations is adequate). The procedure works by randomly resampling (with equal probability and replacement) values from the observed data, such that each random re-sample is of the same size as the dataset. One re-calculates the statistic of interest - e.g., the median - for each re-sample, and then calculates the confidence intervals based on relevant quantiles for this pool of statistics. For example, to produce a 90% confidence interval for the median, one would use the 5th-percentile value from the pool of medians as the lower end of the confidence interval, and the 95th-percentile value would be used as the upper end. Generally, bootstrapping must use a few hundred or a few thousand random re-samples. In our analysis, we use 1,000 re-samples, as we found this amount to yield stable values for confidence bounds.