Reliability assessment of scenarios generated for stock index returns incorporating momentum

Xiaoshi Guo*and Sarah M. Ryan†

Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, IA, 50011, USA

January 18, 2020

Abstract

Stochastic programming models for portfolio optimization rely on scenario paths for returns derived from stochastic process models. This paper investigates a variant of the geometric Brownian motion process for stock index returns that incorporates index momentum. Based on this model, three different processes for generating scenarios on a rolling basis are devised, that differ according to how frequently the momentum parameter is updated and whether it is estimated according to a simple moving average or an exponentially weighted moving average of returns. The reliability of scenario sets generated for multiple instances are assessed by applying a recently developed statistical tool. Backtesting is conducted in case studies of the Standard & Poor's 500 Index and the Financial Times Stock Exchange 100 Index for two different historical periods. The numerical results show that the frequency with which the expected return is updated does not significantly influence the performance of the scenario generation procedure, whereas how the expected return is calculated affects the autocorrelation and dispersion of generated scenarios drastically. All scenario generation schemes are highly sensitive to the estimated volatility of the index returns. Among the three processes tested, the algorithms that incorporate momentum estimates according to an exponentially weighted moving average can generate reliable scenarios when the volatility estimation error is small.

Keywords: index return, momentum, volatility, scenario generation, scenario reliability assessment

1 Introduction

Portfolio optimization is the problem of dynamically allocating investment funds among a set of assets or asset classes. This problem can be formulated as a mathematical program over a given time horizon by

*xsguo@iastate.edu

†smryan@iastate.edu

defining three major components of the model. First, decision variables represent the investment choices. Second, an objective function represents a goal such as maximizing future wealth, the returns generated by the portfolio in each period, and/or some measure of investment risk. Finally, constraints express limits on acceptable investment behavior and tolerable levels of risk while describing the links between decisions in one period and available funds to invest in the next. The uncertainty of future investment performance poses daunting challenges for modeling and computation. In some situations it is possible to formulate and solve a dynamic optimization problem where both time and outcomes are continuous (Koijen et al., 2009). However, it is more common to discretize the time horizon into periods and generate discrete probabilistic scenarios for asset performance (Fulton and Bastian, 2018). In practice, the resulting stochastic programming formulation may be updated and solved repeatedly to rebalance a portfolio in response to changing financial conditions.

To generate scenarios of investment performance over multiple future periods, a common approach is to identify and estimate parameters for a stochastic process model. Such a model may be derived by fitting to historical data (Høyland and Wallace, 2001; Fulton and Bastian, 2018) or be based on assumptions about markets and investor behavior (Hull, 2009). Based on the model specified, sample paths for future realizations can be generated by Monte Carlo methods. For stock prices, the classical lognormal or geometric Brownian motion model is appealing due to its simplicity and widespread use. It rests on simple assumptions on investor rationality in efficient markets and requires only two parameters, representing drift and volatility, to be estimated. However, a considerable amount of empirical research on stock market momentum and strategies to exploit it has raised questions concerning these fundamental assumptions (Jegadeesh and Titman, 1993; Conrad and Kaul, 1998; Lewellen, 2002). The various definitions of, and explanations for, momentum unite around the idea that trends in market performance persist over periods of up to a few months. Mathematically, the existence of momentum undermines the plausibility of a constant drift parameter that could be estimated once and used to generate sample paths extending far into the future. It also raises questions about how to estimate volatility in a nonstationary regime.

For an investment manager to place confidence in a stochastic programming model for portfolio optimization, convincing evidence must be provided that the probabilistic scenarios accurately reflect the range of possible financial outcomes over the investment horizon. Frequently, computational resources limit the number of scenarios that solvers can accommodate in reasonable solution times. The stochastic programming literature on scenario generation and reduction is rich in methods for discretizing continuous distributions and reducing large sets of generated scenarios to smaller representative sets for use in the optimization model. However, unlike in studies of investment strategies where backtesting over historical time periods is routine, it is rare for observational data on actual realizations to be used in testing the performance of scenario generation methods for use in stochastic programming models.

The goals of this paper are to investigate a variant of the geometric Brownian motion model that replaces the constant drift parameter with a time-dependent momentum parameter, and to explore different schemes for updating momentum estimates while generating sample paths. We use a recently developed tool for

assessing the reliability of sets of probabilistic scenarios against observational data. Originally motivated by work on verification of ensemble forecasts in meteorology, and successfully applied in the context of energy planning, a graphical and quantitative approach is applied to loosely test a hypothesis that the observed realization over a given period is indistinguishable from the collection of scenarios generated to represent possible realizations over that period. We find that the procedures that update the momentum estimate while emphasizing more recent observations outperform the alternative, as they generate reliable scenarios as long as the error in the estimate of volatility is small.

The remainder of this paper is organized as follows: Section 2 briefly reviews the relevant literature. Section 3 presents our methodology for generating scenarios of index returns by (1) defining the model of stock index returns and index levels, (2) detailing three alternative processes for scenario generation on a rolling basis, and (3) conducting the scenario assessment. In Section 4, the scenario generation method is applied to the Standard and Poor's 500 Index and the Financial Times Stock Exchange 100 Index in two separate time periods, and the results are analyzed. Section 5 concludes the paper with a summary of findings and directions for future research.

2 Literature Review

When determining how much of an investment fund should be allocated to equities, the market return is one of the primary concerns. Here we focus on the return of a market index, computed as the change in the index level over a period as a proportion of the level at the beginning of the period. Due to the uncertainty of the market, stochastic process models are employed to describe the index levels and returns. Typically, the form of the stochastic process is fit to historical data and its parameters are calibrated using some prior information (Hochreiter and Pflug, 2007).

Recent empirical research suggests that investors can improve the timing of their transactions by following momentum signals. Momentum can be defined as the tendency of stock prices to keep moving in the same direction for a period after an initial impulse. There are many forms of momentum, such as price momentum, return momentum, and earnings momentum, which are distinguished by the initial impulse used. Price momentum, where the initial impulse is simply a change in the price itself, is the most widely used, and it is stronger than the return momentum (Hong and Satchell, 2012). Index momentum is a type of price momentum, because the stock market index is a weighted average of the prices of several representative stocks or other investment vehicles within a sector of the stock market. It shares all the properties of price momentum. Even though the price momentum seems basic, there is little consensus in the literature about what drives it. Generally speaking, price momentum is attributed to behavioral explanations, which fall into two main categories. The first category addresses a process of gradual adjustment to news. Stock prices initially underreact to news, then adjust as time goes by so that the long-term response becomes appropriate and rational. The second category stresses the overreaction by irrational investors to stories of dubious

relevance. When overreaction develops gradually, the stock prices may display momentum for a period of time but will eventually reverse and return to their fundamental values. Attempting to exploit momentum, researchers have developed the relative strength strategy, which consists of buying past winners and selling past losers. Jegadeesh and Titman (1993) show that the past winners have continued to outperform past losers over horizons of 3-12 months, indicating that the return continuation lasts for at most a year. There are three sources for the expected profit of this relative strength strategy. This paper is concerned with only one source of momentum profits: stock returns might be positively autocorrelated (Moskowitz and Grinblatt, 1999), as a firm with a high return today is expected to have high returns in the future.

Admittedly, momentum can play an important role in estimating the future stock returns. Nevertheless, as momentum has various forms and controversial causes, models of returns and stock prices that incorporate momentum are few. Carhart (1997) extends Fama and French's (1993) three-factor model by incorporating a one-year momentum factor in the regression function of stock returns. Hong and Satchell (2012) define momentum as the ratio of today's price to its value n periods ago, and describe the asset price process as a log Ornstein-Uhlenbeck process so that momentum can manifest itself as either trend or autocorrelation. They validate this form of momentum for single asset prices; however, they do not discuss how to apply this formulation of momentum to simulate future asset prices. Koijen et al. (2009) propose a model of returns that accommodates both momentum and mean reversion with two state variables: the weighted average of past returns and the dividend yield.

A stochastic process model for stock index returns can be most accurately formulated with continuous marginal distributions. Nonetheless, it is difficult to solve a stochastic programming problem with continuous distributions for the uncertain parameters. Thus, the continuous distributions at future time points are often discretized in terms of a finite number of scenarios, the process of which is called scenario generation. Various scenario generation methods have been proposed and applied to portfolio optimization problems. Sampling from the continuous distributions is a straightforward approach that is widely used to generate scenarios. In previous studies (Bertsimas and Pachamanova, 2008; Dantzig and Infanger, 1993), the return has been assumed to follow a normal or lognormal distribution. Yu et al. (2003) illustrate three methods for generating asset return scenarios: 1) bootstrapping historical data, 2) statistical modeling with the value at risk approach, and 3) modeling economic factors and asset returns with vector autoregressive models. Bootstrapping is sampling with replacement. By bootstrapping, each scenario is obtained by sampling returns that were observed in the past. The second method is to use the historical data to estimate the volatility and covariance matrix of a multivariate normal distribution and then use Monte Carlo simulation to generate scenarios of returns of different asset classes. The third one is to apply a vector autoregressive model to accommodate the progress of time series while considering the correlation of asset returns. Şakar and Köksalan (2013) estimate the return through a regression equation for the single index model and generate scenarios of index returns from a random walk model. Gulten and Ruszczynski (2015) apply a multivariate generalized orthogonal GARCH model (Gülten and Ruszczyński, 2015; Van der Weide, 2002), which involves

the covariance of uncorrelated return ensembles based on both squared values and the covariance matrix of past uncorrelated return ensembles. Chen and Yang (2017) apply heuristic moment matching to approximate the joint distribution of risky asset returns for future periods. Fulton and Bastian (2018) employ Monte Carlo simulation to generate scenarios based on the estimate of sample means and covariance matrices from a multivariate normal distribution, remove the outlier data based on percentiles, and resample the remaining data to obtain three types of scenarios: positive, negative, or neutral outlook.

Although numerous scenario generation methods have been proposed, no universally superior scenario generation method exists. Therefore, we need some techniques to evaluate the quality of generated scenarios. The verification rank histogram is one such recently-developed technique. The advantage of rank histograms over other scenario quality assessment methods is their ability to provide insights into how well the scenarios compare to the realizations. The verification rank histogram is constructed from the ranks of the observation within each group, consisting of a constant number of scenarios and the corresponding observation. Scalar rank histograms (Hamill and Colucci, 1997) originally were developed to assess univariate ensemble weather forecasts. The minimum spanning tree (MST) rank histogram, which tabulates the frequencies of the rank of the MST length for each scenario, within the group of such lengths that is obtained by replacing an observation for each of its scenario member in turn, is a multivariate extension of the scalar rank histogram (Wilks, 2004). One limitation of the MST histogram is that it is applicable only to equally likely scenarios. While scenarios generated by sampling are equally likely, the application of scenario reduction techniques (Hochreiter and Pflug, 2007)) results in unequally likely scenarios. Sarı et al. (2016) thus develop a new mass transportation distance (MTD) rank histogram, which can be applied to both equally and unequally likely scenarios. It substitutes the MTD (or Wasserstein distance) between the scenario members and the observation for the MST length as a pre-rank function. The principle behind the rank histogram is that the set of ideal scenario values at a given point and its associated observation can be regarded as random samples from the same distribution. Accordingly, the rank histogram of ideal scenarios is supposed to be flat (reflecting a discrete uniform distribution for the ranks), which would indicate that the generation method produces scenarios that well represent the underlying uncertainty. In contrast, overdispersion will present as an upward sloping rank histogram, and underdispersion in the scenarios will show up as a downward sloping histogram (Hamill, 2001). Sarı and Ryan (2018) demonstrate a connection between scenario reliability as assessed by the MTD rank histogram and quality of solutions to the resulting stochastic program instances in the context of short-term power system scheduling.

In summary, our paper focuses on identification, estimation and validation of the scenarios derived from a stochastic process model of stock index returns. In particular, we modify a model proposed by Koijen et al. (2009) to generate scenarios of stock market returns by Monte Carlo simulation. This model captures momentum as an exponentially weighted moving average of recent past index returns to predict future returns. On the basis of this model, three generation procedures are proposed, and the MTD rank histograms developed by Sarı et al. (2016) are plotted to assess the quality of the generated scenarios. Goodness of fit

tests are applied to test a hypothesis of uniformity, which could indicate scenario reliability.

3 Methodology

3.1 Momentum

Momentum refers to the tendency of stock prices to continue moving in the same direction for several months after an initial impulse. Technical analysts interpret momentum in terms of moving averages. An upward (bullish) trend appears when the current return rises above its moving average, while a downward (bearish) trend emerges when the current return falls below its moving average. Since the return is the price change relative to its current value, momentum can be used as an indicator for predicting future returns.

3.2 Model of stock index returns and levels

In this section, we introduce a model using momentum as the expected index return rate for a nonstationary variant of the geometric Brownian motion process of index levels. This model is built based on the work of Koijen et al. (2009). Their model accommodates both return continuation (momentum) over short horizons and return reversals (mean reversion) over longer horizons. Because our model is intended for use to generate scenarios over short time horizons with frequent parameter updates, the effect of return reversals is incorporated in the periodically renewed observations. For this reason, by keeping the short-run return continuation and neglecting the long-run return reversals, we can simplify the stock return model as

$$r_t = \frac{\Delta S_t}{S_t} = M_t \Delta t + \sigma \Delta Z_t \tag{1}$$

where r_t is the return of the index at time t, S_t is the index level at time t, M_t is the expected return over a short time interval, σ is the volatility of the index level, and Z_t is standard Brownian motion with mean 0 and variance t. We use Δ to denote a change over a time increment Δt (i.e. $\Delta S_t \equiv S_{t+\Delta t} - S_t$). According to the independent increments property of Brownian motion (Hull, 2009), ΔZ_t is normally distributed with mean 0 and standard deviation $\sqrt{\Delta t}$. By linear transformation of the normal distribution, the marginal distribution of stock market return at time t is normal with mean $M_t \Delta t$ and standard deviation $\sigma\sqrt{\Delta t}$:

$$r_t \sim N(M_t \Delta t, \sigma\sqrt{\Delta t}) \tag{2}$$

To specify this model for returns, the process of the expected return, M_t, and the volatility, σ, must be estimated.

Similar to Koijen et al. (2009), we approximate the expected return by a moving average model of momentum over the past T periods:

$$M_t = \sum_{i=1}^{T} \frac{e-1}{e^i} \frac{S_{t-i+1} - S_{t-i}}{S_{t-i}} \tag{3}$$

This model, reflecting the recent performance of the index levels, can also be viewed as an exponentially weighted moving average. It modifies the so-called performance variable in Koijen et al. (2009) so that the weights of returns used in the estimate sum to one and M_t is an unbiased estimate of the return. The exponentially diminishing weights place emphasis on the most recent return values, while the function of summation is to smooth the effect of random index level fluctuations.

Next, the volatility of the index level, σ, is a statistical measure of investors' uncertainty about market returns. A sector of the stock market with high volatility is riskier, but this risk cuts both ways: Investing in a volatile market will increase the chance of success as much as the risk of failure. Hull (2009) points out that, for the geometric Brownian motion model with constant drift, $\log S_t$ follows a Brownian motion process with a standard deviation of $\sigma\sqrt{t}$. Equivalently, $\log \frac{S_t}{S_{t-1}}$ follows a normal distribution with a standard deviation of σ, which suggests that the volatility can be estimated based on the standard deviation of the log ratios of successive index levels. Let $u_t \equiv \log \frac{S_t}{S_{t-1}}$ be this log ratio at time t, N be the number of historical periods used to calculate σ, and $\bar{u}$ be the mean of the sequence $\{u_1, ..., u_N\}$. Then the past volatility can be computed as

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}(u_k - \bar{u})^2}. \tag{4}$$

3.3 Scenario generation of stock index returns

To generate scenarios of the stock index returns according to the model in Section 3.2, we apply Monte Carlo simulation, described by Hull (2009), by sampling randomness for the process of the stock returns. To update momentum estimates and recalibrate the model with recent data, we repeatedly generate scenario paths over short time horizons on a rolling basis.

Let F be length of the forecast horizon. We use the observations up to time t to generate scenarios of returns for the following F periods. In each period t, J scenarios are generated. Returns over a single period are generated by sampling from the normal distribution with mean equal to the current momentum estimate and standard deviation equal to the estimated volatility. The momentum variable is an exponentially weighted sum of the returns over the past T periods, and the volatility, which is calculated as the standard deviation of the sequence of log ratios of index levels in the historical period, is a constant throughout the whole simulation horizon. A critical issue is that if this forecast horizon is greater than a year, the effect of momentum will gradually diminish while the effect of mean reversion will increase, which can cause a poor performance of the generated scenarios. Therefore, the forecast horizon should be set to less than one year. This leads us to apply the idea of rolling horizon: cut the horizon short, and consider the dynamic problem on a rolling basis. Specifically, at period t, scenarios of returns representing periods $t+1$ to $t+F$ are generated according to the realized index levels from period $t-T+1$ to t. These scenarios along with the associated observation from $t+1$ to $t+F$ compose one instance. To obtain non-overlapping instances, we proceed to period $t+F$ in the next step. The index level in period $t+F$ is observed, and scenarios over

periods $t+F+1$ to $t+2F$ are generated based on the historical data from period $t+F-T+1$ to $t+F$. By repeating this procedure, we not only maintain an effective forecast horizon but also generate relatively reliable scenarios with such frequent updates of observational data.

In this study, three different scenario generation processes are proposed: a process with daily momentum updated daily, a process with daily momentum updated monthly, and a process employing expected daily return updated monthly based on a simple moving average. The notation used to present the algorithms is defined as follows.

J	Number of scenarios
T	Number of days to calculate each estimate of momentum
S_d	Observed index level on day d
H	Total number of trading days in the simulation horizon
H'	Total number of trading months in the simulation horizon
F	Forecast horizon (days)
F'	Forecast horizon in months
σ	Volatility computed on a daily basis
t_d	Day d in month t
D	Number of trading days in each month
L	Number of months of data used to estimate the mean values of returns
M_d	Daily momentum on day d
$M^*_{d,j}$	Updated daily momentum on day d in scenario j
μ_d	Expected daily return computed by simple moving average on day d
$r^*_{d,j}$	Generated return on day d in scenario j
$S^*_{d,j}$	Generated index level on day d in scenario j
N	Number of non-overlapping instances

3.3.1 Process with daily momentum updated daily

In this subsection, daily momentum is updated each day upon generating the new index level. For example, on day d, the realized index level, S_d, can be observed. We use it, along with the past T periods' index levels, to compute the daily momentum, M_d. Based on the volatility and this momentum, the return on this day, $r^*_{d,j}$, can be derived for each scenario j. Then, we can forecast the index level for the next day in scenario j, $S^*_{d+1,j}$, according to the projected return, $r^*_{d,j}$. By taking account of this newly generated index level, the momentum in scenario j is updated to be $M^*_{d+1,j}$. We use this updated momentum to generate the return for the next day, $r^*_{d+1,j}$. By repeating this procedure, each scenario will have its own paths of daily momentum. To avoid overlapping generation, we obtain the observation data only once every F days. This procedure is detailed in Algorithm 1.

Algorithm 1 Process with daily momentum updated daily

Input: J, T, H, F, σ, S_d for $d = 1, \cdots, H$

Output: $r^*_{d,j}$ for $d = T+1, \cdots, T+NF$ and $j = 1, \cdots, J$

begin:

$N \leftarrow \lfloor \frac{H-T}{F} \rfloor$ ▷ compute the number of non-overlapping instances

for $d \leftarrow 1$ to T **do**

  for $j \leftarrow 1$ **to** J **do** $S^*_{d,j} \leftarrow S_d$ **end for** ▷ initialize the scenarios of daily index levels up to day T

end for

for $d \leftarrow T+1, T+F+1, T+2F+1, \cdots, T+(N-1)F+1$ **do**

  for $j \leftarrow 1$ to J **do**

    $S^*_{d,j} \leftarrow S_d$ ▷ initialize all scenarios to the observed value on day d

    $M_d \leftarrow \frac{e-1}{e}\frac{S_d - S_{d-1}}{S_{d-1}} + ... + \frac{e-1}{e^T}\frac{S_{d-T+1} - S_{d-T}}{S_{d-T}}$ ▷ estimate momentum on day d

    $r^*_{d,j} \leftarrow rnorm(M_d, \sigma)$ ▷ generate scenarios for day d

    for $l \leftarrow d$ to $d+F-2$ **do** ▷ generate scenarios for day $d+1$ to $d+F-1$

      $S^*_{l+1,j} \leftarrow (1 + r^*_{l,j}) S^*_{l,j}$

      $M^*_{l+1,j} \leftarrow \frac{e-1}{e}\frac{S^*_{l+1,j} - S^*_{l,j}}{S^*_{l,j}} + ... + \frac{e-1}{e^T}\frac{S^*_{l-T+2,j} - S^*_{l-T+1,j}}{S^*_{l-T+1,j}}$

      $r^*_{l+1,j} \leftarrow rnorm(M^*_{l+1,j}, \sigma)$

    end for

  end for

  return $r^*_{l,j}$, where $l = d$ to $l = d+F-1$ and $j = 1$ to J.

end for

3.3.2 Process with daily momentum updated monthly

In this subsection, the daily momentum within the same month is held to a constant value and updated monthly. To illustrate, we denote the number of trading days in every month as D. The daily momentum in month t, M_{t_d}, is calculated according to the index level of the first trading day in that month, S_{t_1}, to the last trading day in that month, S_{t_D}. Daily returns in month $t+1$ for each scenario j, $r^*_{(t+1)_d,j}$, can be generated based on this momentum value and volatility. With the generated returns, we can forecast the index level for each day in the next month, $S^*_{(t+1)_d,j}$, and derive a new daily momentum in the next month for each scenario, $M^*_{(t+1)_d,j}$. Returns for each day in the next month, $r^*_{(t+2)_d,j}$, can be generated by this newly updated momentum. To acquire non-overlapping instances, we observe the realized data once every F' months. The specific procedure is demonstrated in Algorithm 2.

Algorithm 2 Process with daily momentum updated monthly

Input: J, H, F', D, σ, S_{t_d} for $t = 1, \cdots, H'$ and $d = 1, \cdots, D$

Output: $r^*_{t_d,j}$ for $t = 2, \cdots, 1 + NF', d = 1, \cdots, D$ and $j = 1, \cdots, J$

begin:

$H' \leftarrow \lfloor \frac{H}{D} \rfloor$, $N \leftarrow \lfloor \frac{H'-1}{F'} \rfloor$ ▷ compute the number of non-overlapping instances

for $t \leftarrow 1, 1+F', 1+2F', \cdots, 1+(N-1)F'$ **do**

  for $j \leftarrow 1$ to J **do**

    $S^*_{(t+1)_1,j} \leftarrow S_{(t+1)_1}$ ▷ initialize all scenarios to the observed value

    $r^*_{t_D,j} \leftarrow \frac{S_{(t+1)_1} - S_{t_D}}{S_{t_D}}$

    for $d \leftarrow 1$ to D **do**

      $M_{t_d} \leftarrow \frac{e-1}{e}\frac{S_{t_D}-S_{t_{D-1}}}{S_{t_{D-1}}} + \frac{e-1}{e^2}\frac{S_{t_{D-1}}-S_{t_{D-2}}}{S_{t_{D-2}}} + ... + \frac{e-1}{e^{D-1}}\frac{S_{t_2}-S_{t_1}}{S_{t_1}}$ ▷ get momentum in month t

      $r^*_{(t+1)_d,j} \leftarrow rnorm(M_{t_d}, \sigma)$ ▷ generate scenarios for month $t+1$

    end for

    for $f \leftarrow t$ to $t+F'-1$ **do** ▷ generate scenarios for month $t+2$ to $t+F'$

      for $d \leftarrow 1$ **to** $D-1$ **do** $S^*_{(f+1)_{(d+1)},j} \leftarrow (1 + r^*_{(f+1)_d,j}) S^*_{(f+1)_d,j}$ **end for**

      for $d \leftarrow 1$ to D **do**

        $M^*_{(f+1)_d,j} \leftarrow \frac{e-1}{e}\frac{S^*_{(f+1)_D,j} - S^*_{(f+1)_{D-1},j}}{S^*_{(f+1)_{D-1},j}} + ... + \frac{e-1}{e^{D-1}}\frac{S^*_{(f+1)_2,j} - S^*_{(f+1)_1,j}}{S^*_{(f+1)_1,j}}$

        $r^*_{(f+2)_d,j} \leftarrow rnorm(M^*_{(f+1)_d,j}, \sigma)$

      end for

    end for

  end for

  return $r^*_{f_d,j}$, where f_d is from $(t+1)_1$ to $(t+F')_D$ and j is from 1 to J

end for

3.3.3 Process with expected daily return updated monthly

To understand whether momentum improves the accuracy of forecast returns, we consider a third generation process using a simple, rather than exponentially weighted, average of recent returns. We replace the momentum estimates used in Subsection 3.3.2 with the equally weighted average of daily returns over the most recent L months to estimate the expected returns. To be more specific, we take month t as an illustration. On each day in month t, the expected return, μ_{t_d}, is computed as the mean of daily returns in the past L months. Then, the return of day d in month $t+1$ for scenario j, $r^*_{(t+1)_d,j}$, can be simulated by sampling from the normal distribution with mean equal to this expected return, μ_{t_d}, and standard deviation equal to the volatility, σ. Note that, besides the approach to calculate expected return, there is another difference between Algorithm 2 and Algorithm 3. In Algorithm 2, the expected return is updated when new stock index levels are generated; while for Algorithm 3, the expected return is always based on the observational data and remains the same throughout the instance horizon.

Algorithm 3 Process with expected daily return updated monthly

Input: J, L, H, F', D, σ, S_{t_d} for $t = 2, \cdots, L+H'$ and $d = 1, \cdots, D$

Output: $r^*_{t_d,j}$ for $t = L+2, \cdots, L+1+NF'$, $d = 1, \cdots, D$ and $j = 1, \cdots, J$

begin:

$H' \leftarrow \lfloor \frac{H}{D} \rfloor$, $N \leftarrow \lfloor \frac{H'-1}{F'} \rfloor$ ▷ compute the number of non-overlapping instances

for $t \leftarrow L+1, L+1+F', L+1+2F', \cdots, L+1+(N-1)F'$ **do**

  for $j \leftarrow 1$ to J **do**

    for $d \leftarrow 1$ to D **do**

      $\mu_{t_d} \leftarrow \frac{1}{DL}\left(\frac{S_{(t+1)_1}-S_{t_D}}{S_{t_D}} + \ldots\ldots + \frac{S_{(t-L+1)_2}-S_{(t-L+1)_1}}{S_{(t-L+1)_1}}\right)$ ▷ get the expected daily return using the past L-month index levels

    end for

    for $f \leftarrow t$ to $t+F'-1$ **do** ▷ generate scenarios of month $t+1$ to $t+F'$

      for $d \leftarrow 1$ **to** D **do** $r^*_{(f+1)_d,j} \leftarrow rnorm(\mu_{t_d}, \sigma)$ **end for**

    end for

  end for

  return $r^*_{f_d,j}$, where f_d is from $(t+1)_1$ to $(t+F')_D$ and j is from 1 to J

end for

3.4 Assessment of scenarios

To evaluate the reliability of a scenario generation method, we seek to quantify the extent to which scenarios and observations are interchangeable. A scenario set is called reliable or calibrated if the relative frequency of occurrence of scenario assigned a probability p tends to be close to p (Hsu and Murphy, 1986). In other

words, the more similar the distribution of generated scenarios and the distribution of real observations are, the more reliable the generation method is judged to be. Sarı et al. (2016) develop and test a novel mass transportation distance (MTD) rank histogram to assess whether the scenarios have similar patterns as the corresponding observations. Based on this work, Sarı and Ryan (2016) create a MTDrh package in R.

The mass transportation distance is the minimum total cost of transporting the probability of one distribution to the other. For two discrete distributions $P = \sum_i p_i \delta_{x_i}$ and $Q = \sum_j q_j \delta_{y_j}$, the mass transportation distance can be computed by solving an optimization problem

$$MTD = \min \left\{ \sum_{i,j} c(x_i, y_j)\eta_{i,j} : \eta_{i,j} \geq 0, \sum_i \eta_{i,j} = q_j, \sum_j \eta_{i,j} = p_i \right\}, \tag{5}$$

where $c(x_i, y_j)$ is the distance between mass points x_i and y_j. For example, $c(x_i, y_j)$ could be the Euclidean distance. In our case, suppose the number of instances is N, $\boldsymbol{r_d}$ is a vector of the observed (realized) values from day $d+1$ to day $d+F$ (i.e. $\boldsymbol{r_d} = (r_{d+1}, \cdots, r_{d+F})$), and $\boldsymbol{r^*_{d,j}}$ is the generated scenario j for day $d+1$ to day $d+F$ (i.e. $\boldsymbol{r^*_{d,j}} = (r^*_{d+1,j}, \cdots, r^*_{d+F,j})$). Then when calculating the mass transportation distance between the distribution of $\boldsymbol{r_d}$, denoted as P, and the distribution of $\boldsymbol{r^*_{d,j}}$, denoted as Q, the mass transportation minimization problem has a simple solution:

$$MTD_0^d = \sum_{j=1}^{J} c(\boldsymbol{r_d}, \boldsymbol{r^*_{d,j}}) p_j^d, \quad d = 1, \cdots, N, \tag{6}$$

where $P = \delta_{\boldsymbol{r_d}}$, $Q = \sum_j p_j^d \delta_{\boldsymbol{r^*_{d,j}}}$, and $c(\boldsymbol{r_d}, \boldsymbol{r^*_{d,j}}) = \sqrt{\sum_{k=d+1}^{d+F} (r_k - r^*_{k,j})^2}$.

Next, we compute MTD_j^d by replacing the scenario j in instance d with the observation during the same time period, and order these MTD values from the largest to the smallest to determine the rank of MTD_0^d in each instance. Finally, the rank histogram plots the frequency with which this rank attains each of its possible values, $1, \ldots, J+1$, over the N instances.

If the scenario sets are reliable, then their distributions should be similar to the distribution of the observation, which means the mass transportation distance from the scenarios to the observation should not be distinguishable from the distances to each scenario. Moreover, by examining the relationship between the shape of the MTD histogram and the difference between the autocorrelation of scenarios and observations, Sarı et al. (2016) point out that if the MTD rank histogram exhibits uniformity, it indicates that the autocorrelation of scenarios and observations are similar and the observation is indistinguishable from the scenarios. If the histogram displays a U-shape or a downward trend, it means the autocorrelations of the scenarios are less than that of the observation and the scenarios are under-dispersed; if it shows an upward trend, it suggests that the autocorrelation of the scenario is greater than that of the observation and the scenarios are over-dispersed.

Note that even if the histogram exhibits uniformity, it cannot be guaranteed that the ranks indeed follow a uniform distribution because the effect of bias can offset the effect of overdispersion when being displayed in the MTD histogram. In order to prevent misdiagnosis, a bias-correction technique should be utilized. The

scenario data are de-biased according to the following formula (Sarı et al., 2016):

$$\tilde{\boldsymbol{r}}^*_{\boldsymbol{d,j}} = \boldsymbol{r}^*_{\boldsymbol{d,j}} - \frac{1}{N}\sum_{d=1}^{N}\left(\frac{1}{J}\sum_{j=1}^{J}\boldsymbol{r}^*_{\boldsymbol{d,j}} - \boldsymbol{r_d}\right), \quad d = 1, \cdots, N, \quad j = 1, \cdots, J. \tag{7}$$

Additionally, we are interested in the effects of correlations within scenarios on effective distances between scenarios. Therefore, the variances of the marginal distributions are equalized by applying the Mahalanobis transformation (Wilks, 2004) for both the observation, $\boldsymbol{r_d}$, and the de-biased scenarios, $\tilde{\boldsymbol{r}}^*_{\boldsymbol{d,j}}$, to derive the transformed observation, $\boldsymbol{z_{d,o}}$, and the transformed de-biased scenario, $\boldsymbol{z_{d,j}}$:

$$\boldsymbol{z_{d,o}} = \boldsymbol{S_d^{-1/2}}(\boldsymbol{r_d} - \bar{\tilde{\boldsymbol{r}}}^*_{\boldsymbol{d}}), \quad d = 1, \cdots, N \tag{8}$$

$$\boldsymbol{z_{d,j}} = \boldsymbol{S_d^{-1/2}}(\tilde{\boldsymbol{r}}^*_{\boldsymbol{d,j}} - \bar{\tilde{\boldsymbol{r}}}^*_{\boldsymbol{d}}), \quad d = 1, \cdots, N, \quad j = 1, \cdots, J, \tag{9}$$

where

$$\bar{\tilde{\boldsymbol{r}}}^*_{\boldsymbol{d}} = \frac{1}{J+1}\left(\boldsymbol{r_d} + \sum_{j=1}^{J}\tilde{\boldsymbol{r}}^*_{\boldsymbol{d,j}}\right), \quad d = 1, \cdots, N \tag{10}$$

and

$$\boldsymbol{S_d} = \frac{1}{J}\left[(\boldsymbol{r_d} - \bar{\tilde{\boldsymbol{r}}}^*_{\boldsymbol{d}})(\boldsymbol{r_d} - \bar{\tilde{\boldsymbol{r}}}^*_{\boldsymbol{d}})^T + \sum_{j=1}^{J}(\tilde{\boldsymbol{r}}^*_{\boldsymbol{d,j}} - \bar{\tilde{\boldsymbol{r}}}^*_{\boldsymbol{d}})(\tilde{\boldsymbol{r}}^*_{\boldsymbol{d,j}} - \bar{\tilde{\boldsymbol{r}}}^*_{\boldsymbol{d}})^T\right], \quad d = 1, \cdots, N \tag{11}$$

After de-biasing and transforming, we use the Cramér-von Mises goodness-of-fit test to assess the nearness to uniformity. The Cramér–von Mises statistic, W^2, can be computed to measure the distance between the uniform distribution and the distribution of the rank of MTD_0. Let F^* be the uniform distribution with possible outcomes of $\{1, 2, \ldots, J+1\}$ that are equally likely to occur, and F be the distribution of the rank of MTD_0. Then W^2 is computed as (Choulakian et al., 1994)

$$W^2 = \frac{N}{J+1}\sum_{x=1}^{J+1}(F(x) - F^*(x))^2 \tag{12}$$

4 Computational results

The index levels for the S&P 500 index and the FTSE 100 index are obtained from the CRSP (`www.crsp.com`) and Yahoo Finance (`https://finance.yahoo.com`) databases, respectively. To test the model, we consider two time intervals for simulation. The first runs from July 1, 1999, to December 27, 2006, which follows the trend of up-down, and the second goes from January 3, 2011, to June 29, 2018, which has an upward trend (Figure 1). These two data sets each contains a total of 1886 trading days. Therefore, the value of H in Algorithm 1 is 1886, while $H' = 94$ in Algorithms 2 and 3. All the following statistical simulations and analyses are conducted in R.

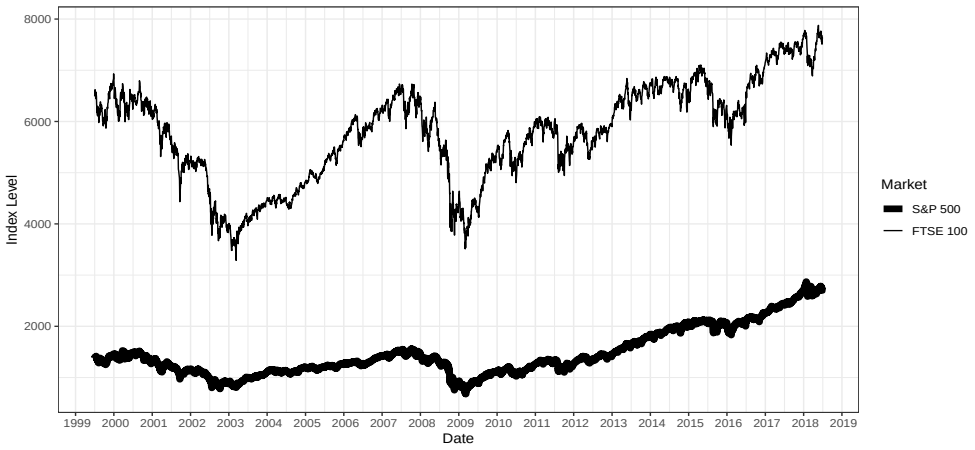


Figure 1: Index levels of S&P 500 and the FTSE 100 from July 1, 1999, to June 29, 2018

In the experiments, $J = 25$ scenarios are generated for each instance. By calculation, the number of trading days in each month, D, is around 20. For simplicity, we assume that $D = 20$ for all months. The forecast horizon, F, in Algorithm 1 is set to be 20 (days) and F' in Algorithms 2 and 3 is set to be 1 (month). The number of days to calculate each daily momentum estimate, T, is 20. As a result, N is 93 in all three algorithms. To match the length of the simulation horizon, we set L in Algorithm 3 to be 94 (trading months). Moreover, because of the financial crisis in 2008, the data between January 3, 2007, and December 31, 2009, are omitted to obtain a relatively valid expected return for the second simulation period. To obtain the candidate values of σ, we compute the daily volatility of roughly every non-overlapping sequence of 1886 trading days based on the historical data back to the very beginning of the S&P 500 index (Table 1) and the FTSE 100 index (Table 2) for consistency with the length of the simulation horizon. Based on these two tables, we test the maximum volatility, the minimum volatility, and some other values in between. The candidate values of daily volatility are listed in Table 3. The three algorithms to generate scenarios are compared on the basis of the performance of their de-biased and transformed MTD rank histogram through the W^2 value of the Cramér–von Mises test. As the scenarios generated by the Monte Carlo simulation are random, each trial is performed thirty times, with common random numbers used across algorithms for each trial, to ensure the comparability of results.

Figures 2 - 4 show the performance of Algorithms 1 - 3, respectively, using S&P 500 index levels from January 3, 2000, to December 31, 2018. Each panel depicts the results of non-overlapping instances, consisting of the observation and 25 scenarios, each of which is a time series of length 20. Note that, within in each figure, the difference in the estimated value of σ is the only source of changes in the shapes of the histograms. In Figures 2 and 3, the observation ranks exhibit a downward trend when daily volatility is 0.006 or 0.007, and an upward trend when daily volatility is 0.010 or 0.011. When daily volatility is 0.0085, the histogram for index returns is nearly flat. These results indicate that when estimated daily volatility is low, the autocorrelation of scenarios is less than that of the observation and scenarios are under-dispersed.

Table 1: Historical daily volatility of the S&P 500 returns

Period	Daily volatility
01/1963 - 06/1970	0.0060
07/1970 - 12/1977	0.0087
01/1978 - 06/1985	0.0088
07/1986 - 12/1993	0.0110
01/1994 - 06/2001	0.0095
07/2001 - 12/2006	0.0110

Table 2: Historical daily volatility of the FTSE 100 returns

Period	Daily volatility
01/1984 - 06/1991	0.010
07/1991 - 12/1998	0.008
01/1999 - 06/2006	0.012

Table 3: Candidate Values of Daily Volatility

Candidate daily volatility (σ)	Period 1 (07/01/1999-12/26/2006) & Period 2 (01/03/2011-06/29/2018)				
S&P 500	0.0060	0.0070	0.0085	0.0100	0.0110
FTSE 100	0.0080	0.0090	0.0100	0.0110	0.0120

When estimated daily volatility is high, the autocorrelation of scenarios becomes greater than that of the observation, and scenarios are over-dispersed. When daily volatility is 0.0085, the autocorrelation of scenarios and observations are similar and the observation is indistinguishable from the scenarios. In Figure 4, we observe a downward trend for index returns when daily volatility is set to be 0.006, flatness when volatility is 0.007, and an upward trend in other cases. Comparing Algorithms 1 and 2, both the volatility of generated returns and the autocorrelation of observations in each column of figures are controlled to be the same, and the frequency with which momentum estimates are updated causes the difference in the shape of histograms, indicating that this frequency has trivial influence on the autocorrelation or dispersion of scenarios. Comparing Algorithms 2 and 3, we notice that even when the frequency with which the expected return is updated is the same, using exponentially weighted average or simple moving average to calculate can cause the differences in the ensembles of index return scenarios. The rank histograms of the other datasets, including the return of the S&P 500 index in the first simulation period, and the FTSE 100 in both simulation periods, shown in the Appendix, exhibit similar behavior.

The uniformity of the de-biased and transformed MTD rank histogram is tested using the Cramér-von Mises (CvM) statistic. According to Choulakian et al. (1994), the critical values of W^2 are 0.871 for the 1% significance level, 0.743 for the 2% significance level, and 0.581 for the 5% significance level in a two-sided hypothesis test. Tables 4 and 5 display the 98% t-statistic confidence intervals for W^2 values of the ranks of observed returns of the S&P 500 and the FTSE 100. In the table bodies, underlining is used to indicate that we cannot reject the null hypothesis of uniformity at the significance level of 1%, while italic and boldface correspond to the 2% and the 5% significance level, respectively. The value of σ shown in bold in each table heading is the one closest to the observed value for that index and time period. From Tables 4 and 5, it can be seen that the performance of Algorithms 1 and 2 is similar: we cannot reject the null hypothesis of uniformity at the 5% (or lower) significance level when daily volatility is set close to the observed value of 0.0110 in Period 1 or 0.0090 in Period 2, for either index. For Algorithm 3, we fail to reject the null hypothesis of uniformity at the 5% significance level only when the estimated volatility is lower than the observed value. Scenarios generated by any of these three algorithms can be considered reliable at the 5% significance level as long as the volatility is set appropriately. The appropriate volatility for Algorithm 1 is always the largest among the three algorithms, and the appropriate volatility for Algorithm 3 is always the smallest. If we can estimate the volatility of observation correctly, the performance of Algorithm 1 and 2 will be reliable. However, even if σ is set equal to the true volatility, Algorithm 3 cannot provide reliable scenarios in any of these four settings.

For additional insight into the results, Figure 5 shows the time-series plots of de-biased and transformed S&P 500 index returns over a small portion of the simulation horizon, from January 2, 2013, to April 29, 2013, for each algorithm when the daily volatility is set to 0.0085 for scenario generation. In these plots, the

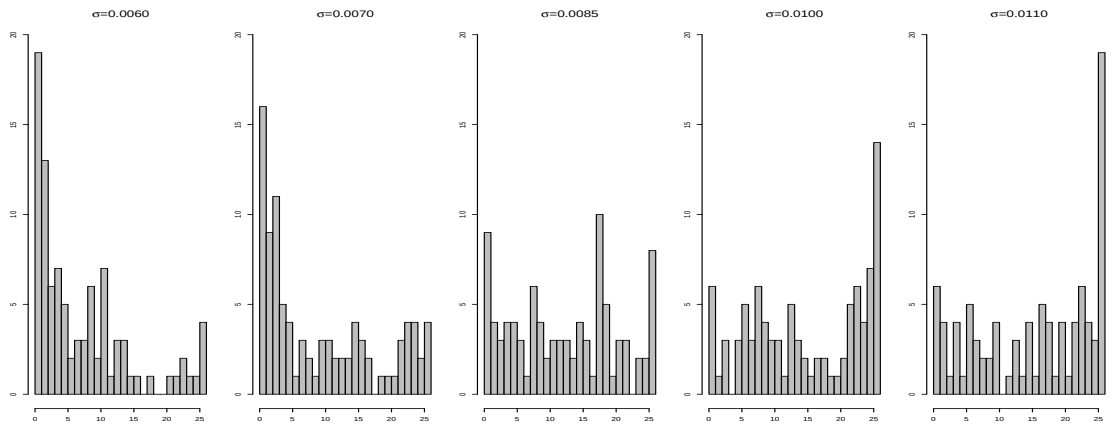


Figure 2: Performance of de-biased and transformed returns generated by Algorithm 1 for index return of S&P 500 in the second simulation period.

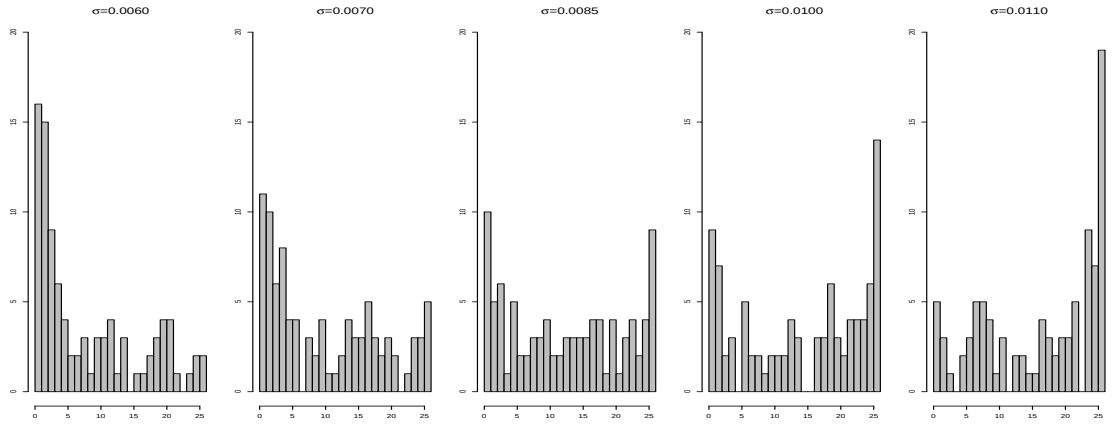


Figure 3: Performance of de-biased and transformed returns generated by Algorithm 2 for index return of S&P 500 in the second simulation period.

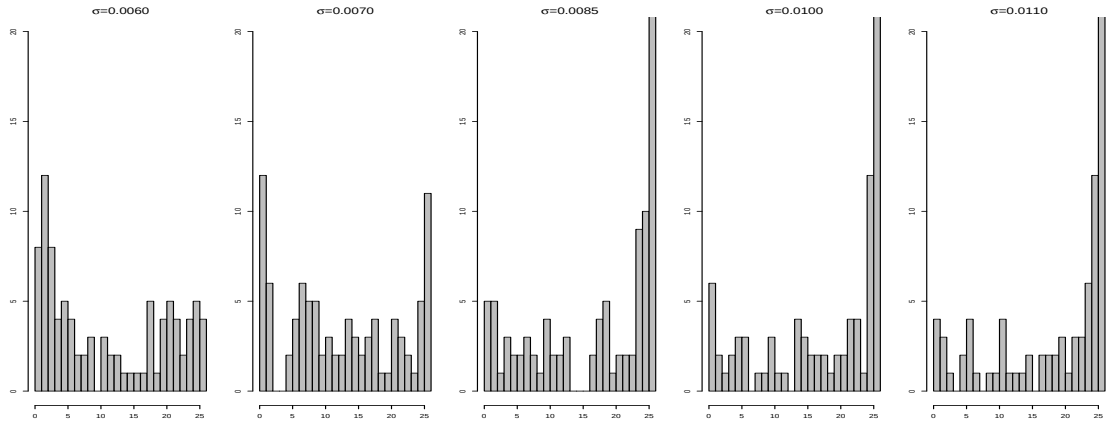


Figure 4: Performance of de-biased and transformed returns generated by Algorithm 3 for index return of S&P 500 in the second simulation period.

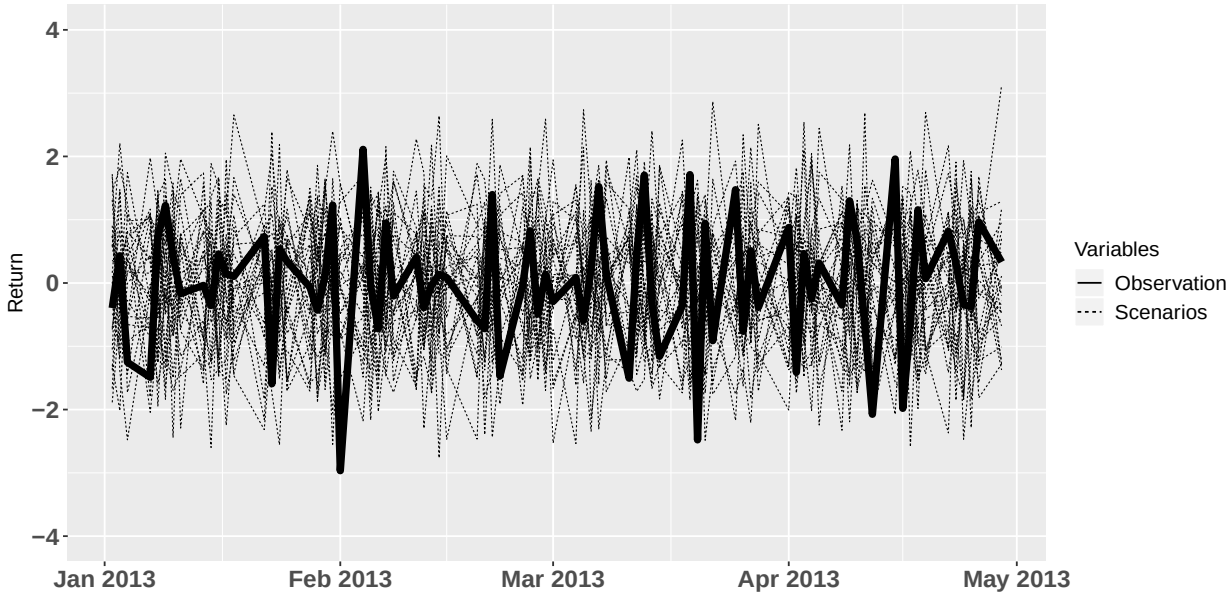
Table 4: 98% confidence interval of W^2 for de-biased and transformed S&P 500 index returns. Underline is used to indicate that the null hypothesis of uniformity cannot be rejected at the significance level of 1%, and italic and bold correspond to the 2% and the 5% significance level, respectively.

Period	Period 1				
Daily Volatility	0.0060	0.0070	0.0085	0.0100	**0.0110**
Algorithm 1	[11.05,12.17]	[7.02, 8.06]	[2.57, 3.63]	[1.08, 1.49]	***[0.43, 0.77]***
Algorithm 2	[10.18, 11.49]	[6.13, 7.22]	[2.38, 3.04]	[0.80, 1.10]	***[0.53, 0.76]***
Algorithm 3	[6.14, 7.32]	[2.92, 3.83]	*[0.66, 1.06]*	***[0.53, 0.78]***	[0.88, 1.34]
Period	Period 2				
Daily Volatility	0.0060	0.0070	**0.0085**	0.0100	0.0110
Algorithm 1	[4.65, 5.58]	[1.62, 2.21]	***[0.34, 0.48]***	[0.77, 1.25]	[1.96, 2.58]
Algorithm 2	[4.45, 5.31]	[1.86, 2.46]	***[0.46, 0.72]***	[0.89, 1.31]	[1.70, 2.29]
Algorithm 3	[1.33, 1.82]	***[0.36, 0.56]***	[0.91, 1.57]	[3.47, 4.23]	[5.08, 6.07]

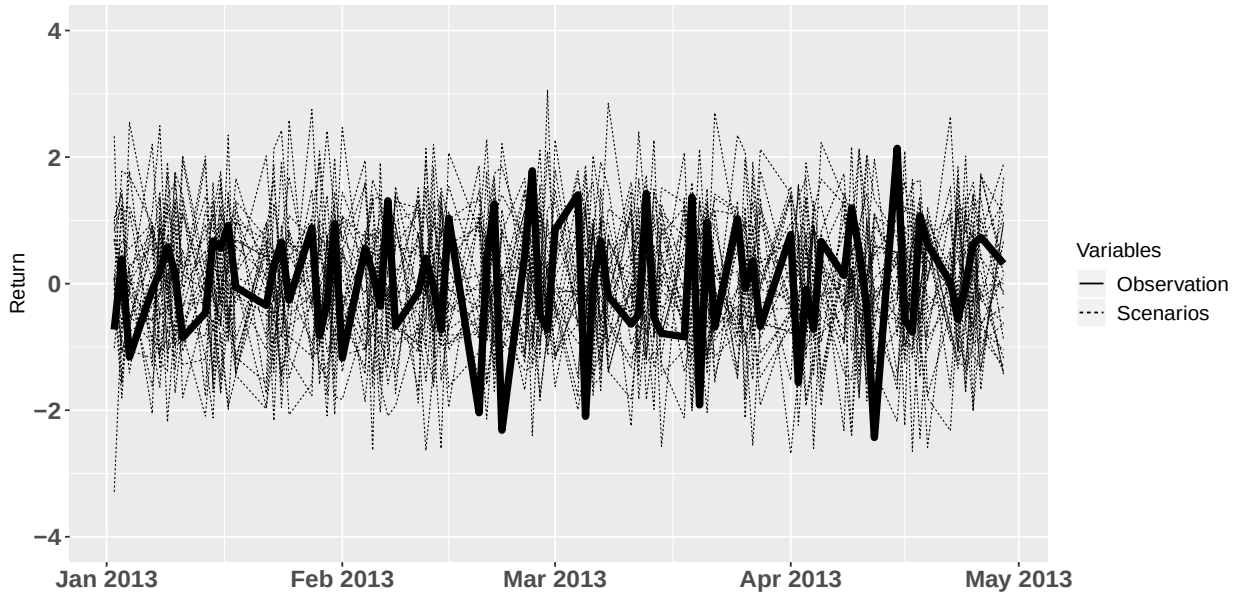
Table 5: 98% confidence interval of W^2 for de-biased and transformed FTSE 100 index returns. Underline is used to indicate that the null hypothesis of uniformity cannot be rejected at the significance level of 1%, and italic and bold correspond to the 2% and the 5% significance level, respectively.

Period	Period 1				
Daily Volatility	0.0080	0.0090	0.0100	**0.0110**	0.0120
Algorithm 1	[2.96, 3.76]	[1.42, 1.86]	*[0.68, 0.96]*	***[0.50, 0.75]***	*[0.72, 1.08]*
Algorithm 2	[2.26, 2.90]	[1.02, 1.37]	***[0.57, 0.80]***	*[0.62, 0.92]*	[1.09, 1.52]
Algorithm 3	[0.80, 1.17]	***[0.52, 0.68]***	[0.76, 1.12]	[1.47, 2.09]	[2.67, 3.51]
Period	Period 2				
Daily Volatility	0.0080	**0.0090**	0.0100	0.0110	0.0120
Algorithm 1	[1.08, 1.69]	***[0.38, 0.55]***	***[0.39, 0.63]***	[1.16, 1.57]	[2.02, 2.68]
Algorithm 2	[0.85, 1.29]	***[0.40, 0.61]***	***[0.57, 0.79]***	[1.54, 2.08]	[2.15, 3.05]
Algorithm 3	***[0.23, 0.38]***	[0.91, 1.41]	[1.88, 2.60]	[4.12, 5.19]	[5.30, 6.79]

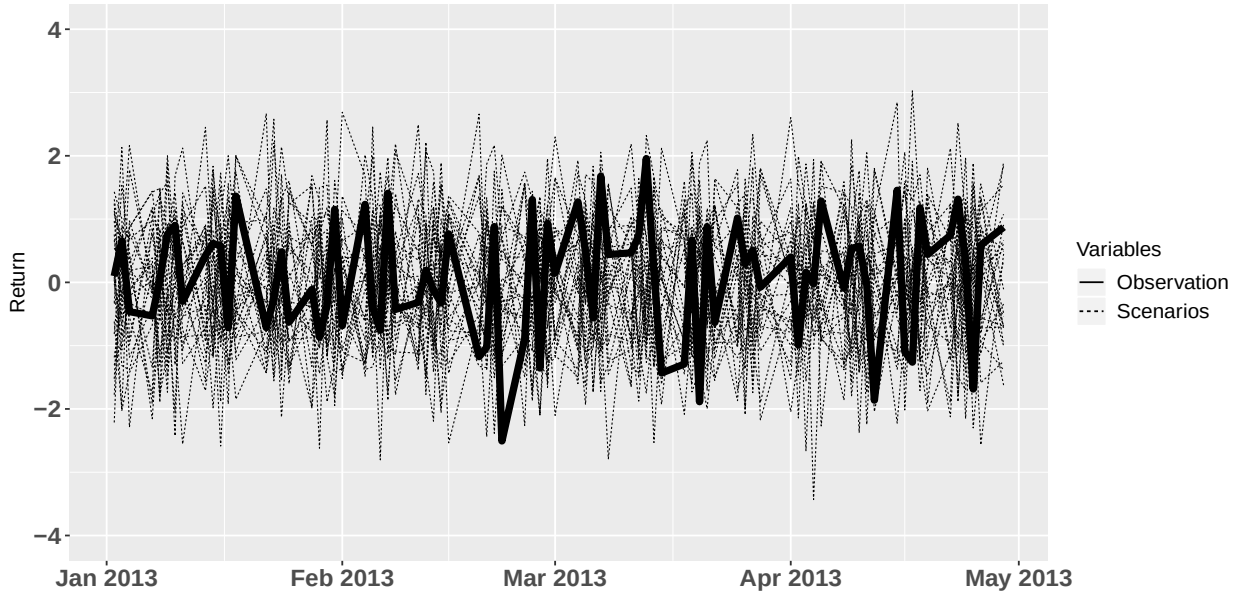
thicker line represents the path of observation data, and other dashed lines are for the 25 generated scenario. From Figure 5a, we can see that the scenarios and observations are intertwined with each other, and it is hard to differentiate between them as the red thicker line is sometimes above most scenarios, sometimes below the majority of scenarios, and sometimes in between the scenarios. From Figure 5b, it can be observed that the scenarios almost always envelop the observation data before mid February but the relative position of the observation oscillates drastically in the latter portion of the time horizon, which explains why the rank histogram for Algorithm 2 is less flat than that for Algorithm 1 in this setting. Compared with the other two time series plots, the observation data in Figure 5c spends the least time being either above or below all the scenarios. The scenario data mostly surround it, indicating that the generated scenarios are over-dispersed, which is consistent with the upward slope of the MTD rank histogram for Algorithm 3 in this setting.



a: De-biased and transformed index returns from Algorithm 1 when $\sigma = 0.0085$



b: De-biased and transformed index returns from Algorithm 2 when $\sigma = 0.0085$



c: De-biased and transformed index returns from Algorithm 3 when $\sigma = 0.0085$

Figure 5: Time-series plot: 25 generated scenarios of S&P 500 index returns, represented by the dashed lines, along with the corresponding observations, represented by the thicker solid line, from Feb 2, 2011, to April 28, 2011

5 Conclusion

In this study, we propose a model that substitutes short-term momentum for the drift in a geometric Brownian motion process for index levels to produce a nonstationary variant of this widely used model. While the volatility can be estimated by computing the standard deviation of log ratios of index levels when the drift is constant, it is not straightforward to produce a reliable estimate of volatility to use in our momentum model. Therefore, we test different volatility estimates along with various schemes for updating the momentum or drift estimates in rolling horizon procedures for scenario generation. The performance of algorithms with different values of volatility is compared based on the MTD rank histograms.

By carrying out computational studies on the S&P 500 and the FTSE 100 indices in two different time periods, we find that the approach to update the expected index returns has a large effect on the reliability of scenarios generated, while the frequency to update the expected return does not influence the generation significantly. Besides the expectation of index returns, the volatility of returns also plays an important role in the scenario generation process. All algorithms are able to produce reliable scenarios when the volatility is set appropriately. For Algorithm 1 and 2, as long as the volatility is set close to the true observed volatility, the results will be reliable. For Algorithm 3, the volatility parameter value that produces reliable scenarios is always smaller than the true volatility. Thus, even if we estimate the volatility accurately, Algorithm 3 does not provide reliable scenarios for any combination of stock indices and time periods tested. Therefore, Algorithms 1 and 2 seem to be more reliable in that they provide reliable scenarios as long as the volatility estimation error is small. These results provide indirect evidence for the existence of momentum in index returns, as weighting recent observations more heavily when updating expected return estimates produces better scenarios for future returns.

The lack of uniformity of the rank histograms produced even by the most suitable method tested, if the volatility is wrong, suggests that the model and scenario generation methods could be improved. Some possibilities to be investigated are to incorporate nonstationary volatility as well as momentum in the model, to obtain better estimates of volatility, or to include other factors such as mean reversion. The relationship between scenario reliability and performance of the solutions to corresponding stochastic programs also must be explored thoroughly. In this paper, the forecast horizon of 20 trading days (1 month) has been tested. Comparing the performance of the algorithms with a longer forecast horizon could be another avenue to explore.

Fama and French (1993)

References

Bertsimas, D. and Pachamanova, D. (2008). Robust multiperiod portfolio management in the presence of transaction costs. *Computers & Operations Research*, 35(1):3–17.

Carhart, M. M. (1997). On persistence in mutual fund performance. *The Journal of Finance*, 52(1):57–82.

Chen, H.-H. and Yang, C.-B. (2017). Multiperiod portfolio investment using stochastic programming with conditional value at risk. *Computers & Operations Research*, 81:305–321.

Choulakian, V., Lockhart, R. A., and Stephens, M. A. (1994). Cramér-von Mises statistics for discrete distributions. *Canadian Journal of Statistics*, 22(1):125–137.

Conrad, J. and Kaul, G. (1998). An anatomy of trading strategies. *The Review of Financial Studies*, 11(3):489–519.

Dantzig, G. B. and Infanger, G. (1993). Multi-stage stochastic linear programs for portfolio optimization. *Annals of Operations Research*, 45(1):59–76.

Fama, E. F. and French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33:3–56.

Fulton, L. V. and Bastian, N. D. (2018). Multiperiod stochastic programming portfolio optimization for diversified funds. *International Journal of Finance & Economics*.

Gülten, S. and Ruszczyński, A. (2015). Two-stage portfolio optimization with higher-order conditional measures of risk. *Annals of Operations Research*, 229(1):409–427.

Hamill, T. M. (2001). Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, 129(3):550–560.

Hamill, T. M. and Colucci, S. J. (1997). Verification of Eta–RSM short-range ensemble forecasts. *Monthly Weather Review*, 125(6):1312–1327.

Hochreiter, R. and Pflug, G. C. (2007). Financial scenario generation for stochastic multi-stage decision processes as facility location problems. *Annals of Operations Research*, 152(1):257–272.

Hong, K. J. and Satchell, S. (2012). Defining single asset price momentum in terms of a stochastic process. *Theoretical Economics Letters*, 2(03):274.

Høyland, K. and Wallace, S. W. (2001). Generating scenario trees for multistage decision problems. *Management Science*, 47(2):295–307.

Hsu, W.-r. and Murphy, A. H. (1986). The attributes diagram a geometrical framework for assessing the quality of probability forecasts. *International Journal of Forecasting*, 2(3):285–293.

Hull, J. C. (2009). *Option, Futures and other Derivatives (7th ed.)*. Upper Saddle River, NJ: Prentice Hall.

Jegadeesh, N. and Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of finance*, 48(1):65–91.

Koijen, R. S. J., Rodríguez, J. C., and Sbuelz, A. (2009). Momentum and mean reversion in strategic asset allocation. *Management Science*, 55(7):1199–1213.

Lewellen, J. (2002). Momentum and autocorrelation in stock returns. *Review of Financial Studies*, 15(2):533–564.

Moskowitz, T. J. and Grinblatt, M. (1999). Do industries explain momentum? *The Journal of Finance*, 54(4):1249–1290.

Şakar, C. T. and Köksalan, M. (2013). A stochastic programming approach to multicriteria portfolio optimization. *Journal of Global Optimization*, 57(2):299–314.

Sarı, D., Lee, Y., Ryan, S., and Woodruff, D. (2016). Statistical metrics for assessing the quality of wind power scenarios for stochastic unit commitment. *Wind Energy*, 19(5):873–893.

Sarı, D. and Ryan, S. M. (2016). MTDrh: Mass transportation distance rank histogram. `https://cran.r-project.org/web/packages/MTDrh/index.html`.

Sarı, D. and Ryan, S. M. (2018). Statistical reliability of wind power scenarios and stochastic unit commitment cost. *Energy Systems*, 9(4):873–898.

Van der Weide, R. (2002). GO-GARCH: a multivariate generalized orthogonal GARCH model. *Journal of Applied Econometrics*, 17(5):549–564.

Wilks, D. S. (2004). The minimum spanning tree histogram as a verification tool for multidimensional ensemble forecasts. *Monthly Weather Review*, 132(6):1329–1340.

Yu, L.-Y., Ji, X.-D., and Wang, S.-Y. (2003). Stochastic programming models in financial optimization: A survey. *AMO - Advanced Modeling and Optimization*, 5(1):1–26.

Appendix

Figures A1-A9 show the performance of the generated scenarios using Algorithms 1, 2 and 3 for the S&P 500 in the first period, and the FTSE 100 in both periods.

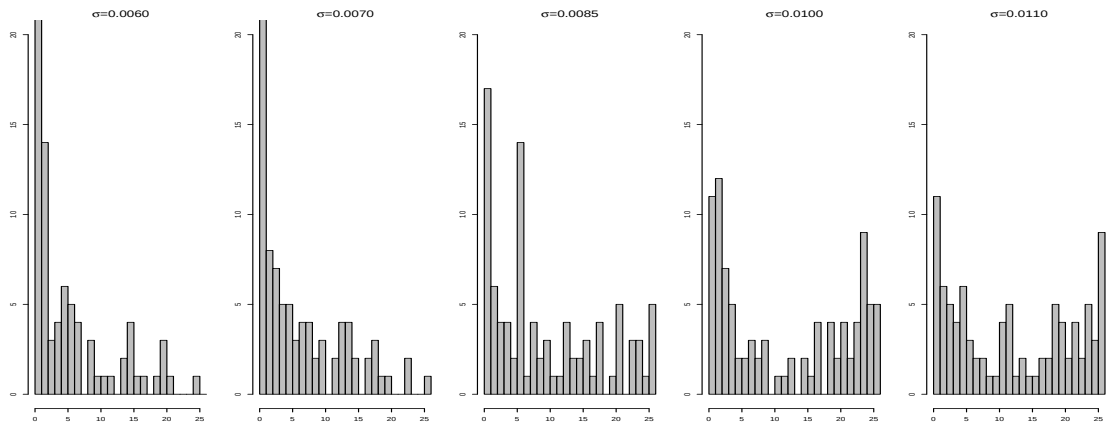


Figure A1: Performance of de-biased and transformed returns generated by Algorithm 1 for index return of the S&P 500 in the first simulation period.

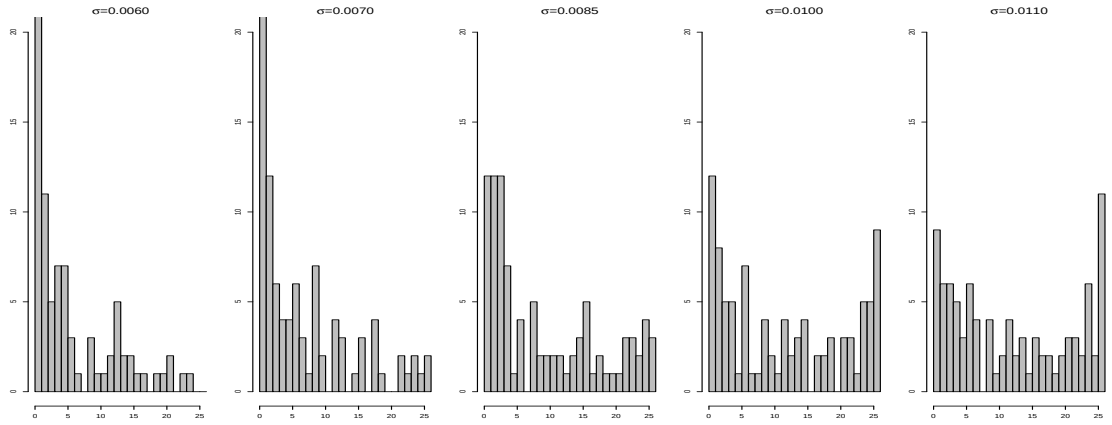


Figure A2: Performance of de-biased and transformed returns generated by Algorithm 2 for index return of the S&P 500 in the first simulation period.

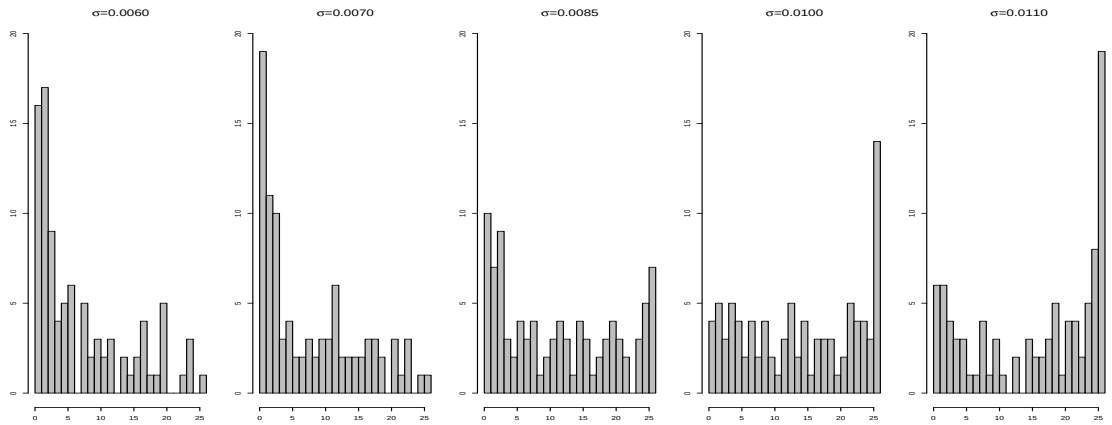


Figure A3: Performance of de-biased and transformed returns generated by Algorithm 3 for index return of the S&P 500 in the first simulation period.

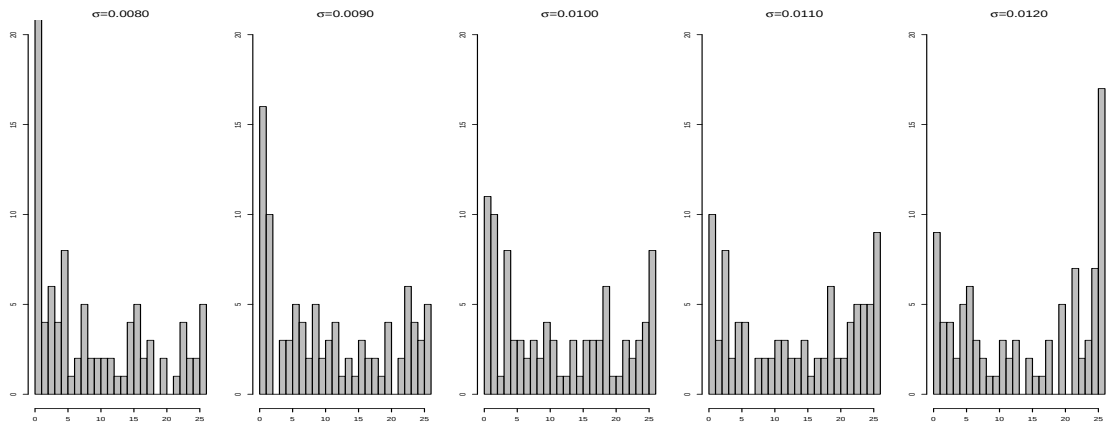


Figure A4: Performance of de-biased and transformed returns generated by Algorithm 1 for index return of the FTSE 100 in the first simulation period.

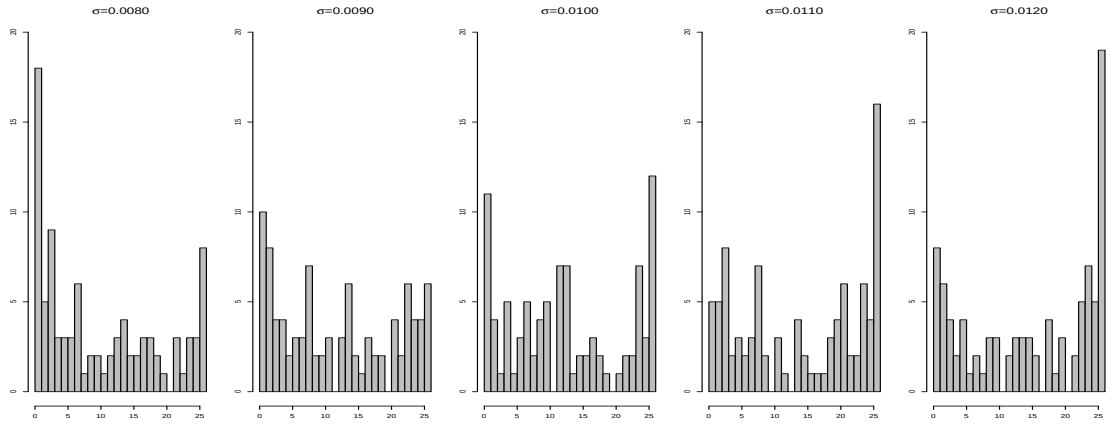


Figure A5: Performance of de-biased and transformed returns generated by Algorithm 2 for index return of the FTSE 100 in the first simulation period.

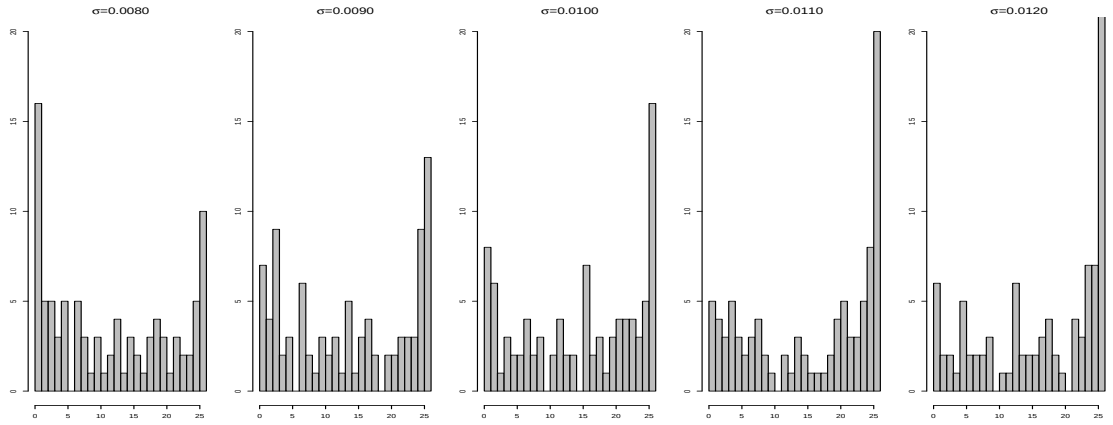


Figure A6: Performance of de-biased and transformed returns generated by Algorithm 3 for index return of the FTSE 100 in the first simulation period.

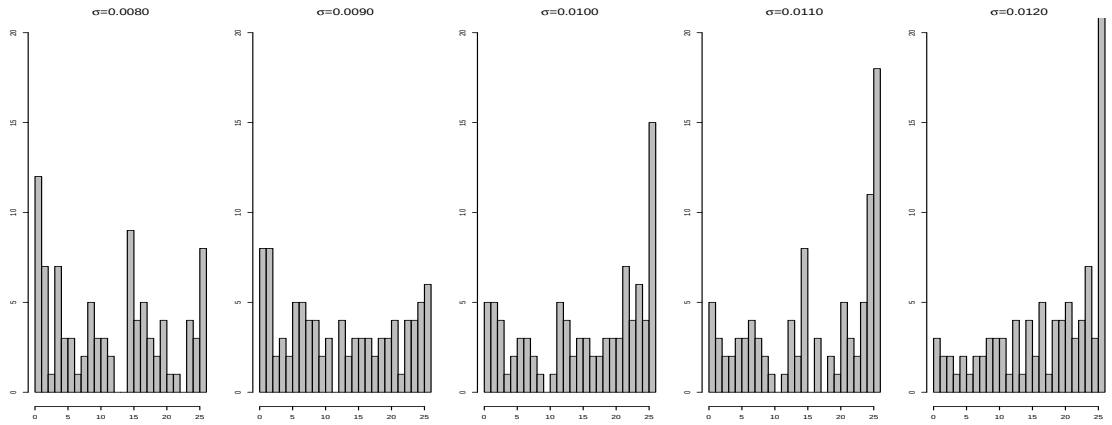


Figure A7: Performance of de-biased and transformed returns generated by Algorithm 1 for index return of the FTSE 100 in the second simulation period.

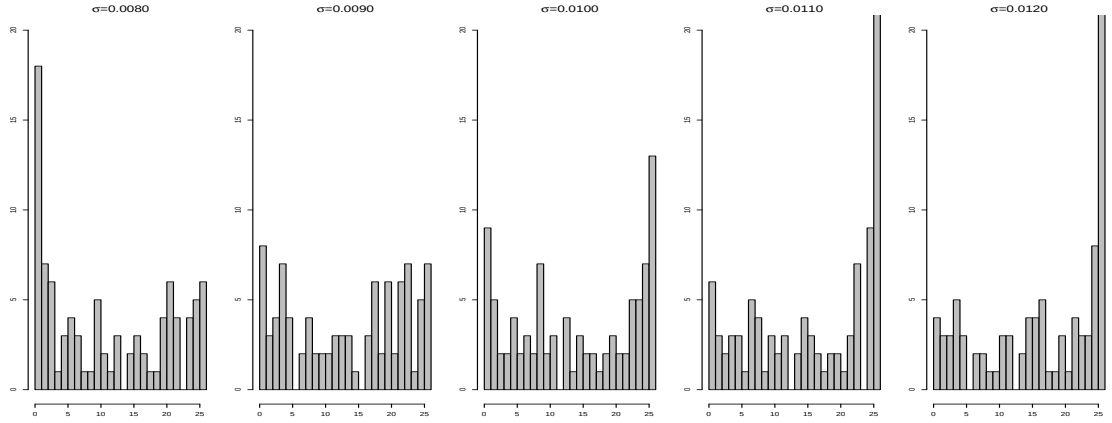


Figure A8: Performance of de-biased and transformed returns generated by Algorithm 2 for index return of the FTSE 100 in the second simulation period.

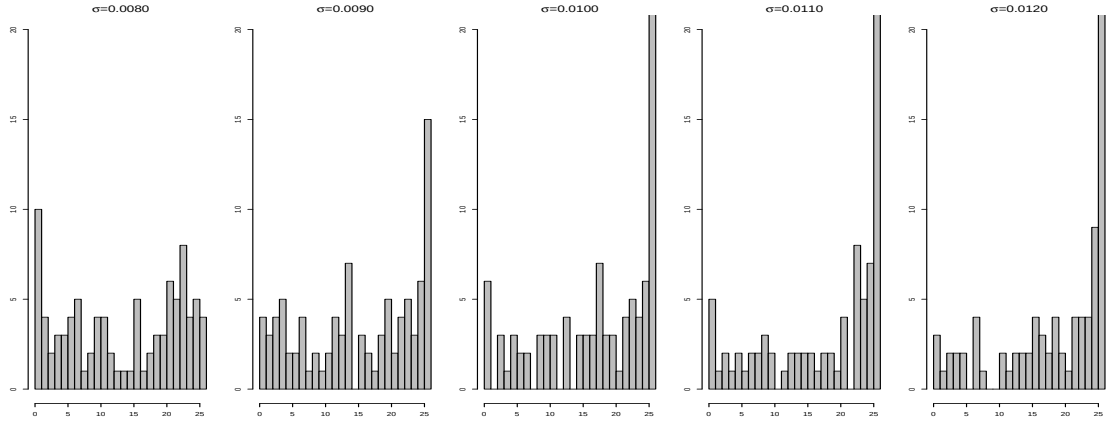


Figure A9: Performance of de-biased and transformed returns generated by Algorithm 3 for index return of the FTSE 100 in the second simulation period.