Financial Econometrics Project

Analysis of High-Frequency Data, Volatility, and Value at Risk (VaR)

Authors: Chaker Meraihi & Shakil Rohimun
Degree Program: Master's in Quantitative Finance (M2QF)
Instructor: Prof. Juhyun Park

Date: December 2024

Declaration of Contributions

Chaker Meraihi was responsible for implementing the Brownian Bridge to address the irregular time intervals in high-frequency price arrivals and performed the Functional Principal Component Analysis (FPCA) for data dimensionality reduction. Shakil Rohimun developed the GARCH(1,1) model and conducted the Value at Risk (VaR) analysis under non-stationary conditions.

Introduction

Analyzing high-frequency financial data is essential to understanding market behavior and managing financial risks effectively. This project tackles key challenges, such as irregular time intervals, modeling conditional volatility, and estimating Value at Risk (VaR) from high-frequency price data. By leveraging advanced mathematical tools like the Brownian Bridge, alongside econometric models such as GARCH(1,1) and Functional Principal Component Analysis (FPCA), it establishes a robust framework for examining both daily and intraday volatility. The results of this study could be highly relevant for risk management teams, offering practical methods to estimate and forecast VaR using high-frequency data, which can enhance decision-making in highly volatile market environments.

Keywords

High-Frequency Data, Value at Risk (VaR), GARCH(1,1), Functional Principal Component Analysis (FPCA), Brownian Bridge, Microstructure Noise, Log Returns, Conditional Volatility, Time Series Analysis, Financial Econometrics, Risk Management, Volatility Forecasting.

1.1 Handling Irregular Time Intervals in High-Frequency Data

In high-frequency financial data, the timing of price observations is irregular, as trades occur at varying intervals. This irregularity poses a problem for return calculations : without consistent time intervals, returns calculated over different periods are not directly comparable. Thus, a consistent method is needed to normalize the timing of observations, enabling accurate analysis of log returns. To address this, we propose four methods for handling irregular time intervals:

1. **Time-Weighted Returns:** This approach weights log returns by the time between observations, accounting for the variability in intervals. For two consecutive log prices, p_i and p_{i+1}, observed at times t_i and t_{i+1}, respectively, the time-weighted return is given by:

   $$r_i = (p_{i+1} - p_i) \times (t_{i+1} - t_i).$$

   This method provides a weighted measure of returns, reducing the distortion caused by irregular intervals.

2. **Regular Sampling:** This method selects prices at regular intervals (e.g., every 10 ticks). By resampling the data, we achieve a standardized interval, allowing for consistent comparisons of returns. However, this approach may discard valuable information by skipping intermediate prices.

3. **Interpolation Methods (Linear and Spline):** Interpolation fills in missing values on a regular time grid by estimating prices at predefined intervals. Linear interpolation assumes a constant rate of change between two points, while spline interpolation uses piecewise polynomials for smoother transitions. For linear interpolation, given two prices p_i and p_{i+1} at times t_i and t_{i+1}, the interpolated price at time t is:

   $$p(t) = p_i + \frac{p_{i+1} - p_i}{t_{i+1} - t_i} \times (t - t_i).$$

   While effective, interpolation may introduce artificial trends in high-frequency data. This limitation paves the way for more sophisticated approaches, such as the Brownian Bridge, which better captures the dynamics of high-frequency data.

1.1.1 Brownian Bridge for Handling Irregular Time Intervals

High-frequency trading data often has irregular time intervals due to factors like fragmented liquidity across exchanges, dark pools, and algorithmic trading. This irregularity poses challenges for modeling returns, as trades are not evenly spaced. The Brownian Bridge offers a stochastic interpolation approach, creating a realistic path between observed prices while preserving the statistical structure.

Given two observed log prices p_i and p_{i+1} at times t_i and t_{i+1}, the Brownian Bridge price $p(t)$ for $t \in [t_i, t_{i+1}]$ is:

$$p(t) = p_i + \frac{t - t_i}{t_{i+1} - t_i}(p_{i+1} - p_i) + Z\sqrt{\sigma^2 \frac{(t_{i+1} - t)(t - t_i)}{t_{i+1} - t_i}}$$

The mean and variance at any intermediate time t are:

$$\mathbb{E}[p(t)] = p_i + \frac{t - t_i}{t_{i+1} - t_i}(p_{i+1} - p_i) \qquad \mathrm{Var}[p(t)] \quad = \sigma^2 \frac{(t_{i+1} - t)(t - t_i)}{t_{i+1} - t_i}.$$

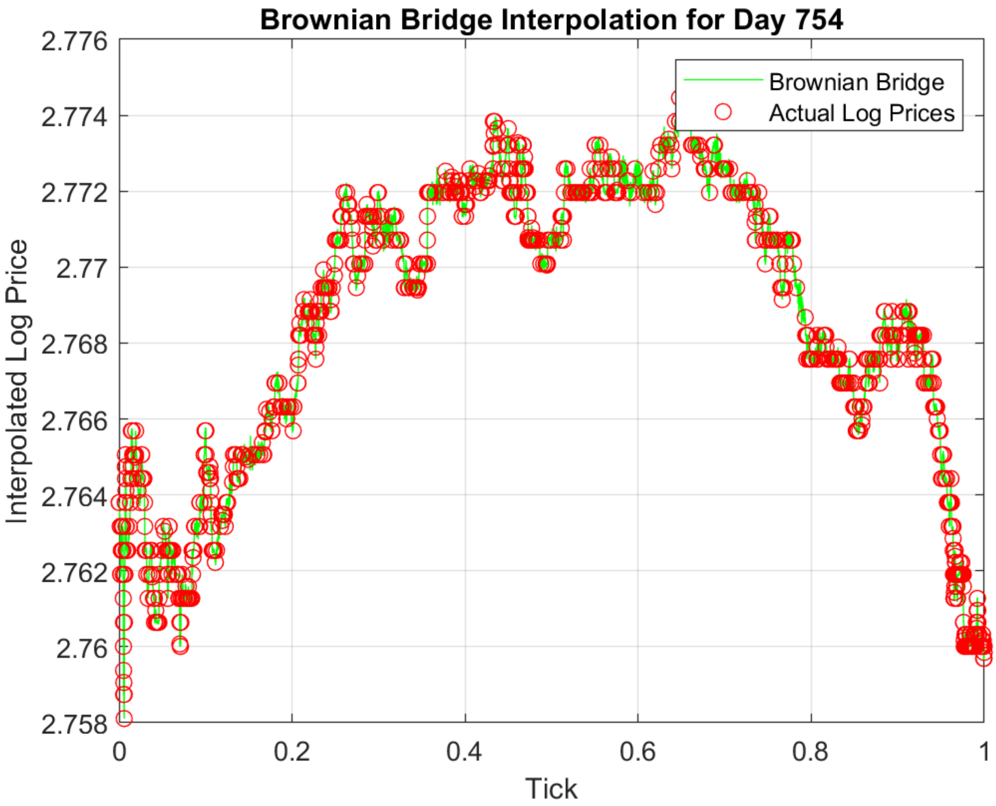

Figure 1.1: Illustration of the Brownian Bridge interpolation method. The path smoothly connects two observed prices at irregular time intervals, capturing both the expected mean and variance between observations.

Why Brownian Bridge?

The Brownian Bridge is particularly effective for high-frequency data where irregular intervals arise from:

- **Dark Pools:** Large trades are executed without immediate public disclosure, creating gaps in observable prices.
- **Fragmented Exchanges:** Trades occur across multiple platforms, with unsynchronized timing.
- **Algorithmic Trading:** Frequent cancellations and modifications introduce further variability.

By respecting the mean and variance structure between observations, the Brownian Bridge captures realistic price dynamics and provides a better foundation for calculating returns at regular intervals, essential for financial modeling on irregular time series.

1.1.2 Daily Splitting of High-Frequency Data

In high-frequency financial data analysis, splitting data by day is essential to exclude overnight jumps. These jumps reflect cumulative information from global economic events, news, and trading activities outside market hours. They introduce discontinuities in the price series, unrelated to the stochastic processes governing intraday trading dynamics. Ignoring these jumps can distort intraday volatility models and lead to inaccurate analysis.

Theoretical Justification

In theory, financial returns are assumed to follow a continuous-time stochastic process during trading hours, such as a Brownian motion or other diffusion process. However, prices often exhibit significant jumps at the beginning of each trading day. These jumps are exogenous to the within-day trading process and introduce non-stationary behavior into the series, which complicates intra-day volatility modeling. Modeling daily returns as a continuous series without separating overnight jumps would misrepresent the actual volatility within trading hours.

1.2 Statistical Analysis of Log Returns

To evaluate the statistical properties of the log returns, we computed descriptive statistics and applied normality and stationarity tests for each day.

1.2.1 Descriptive Statistics for Day 1 and Day 2

For Days 1 and 2, the descriptive statistics are as follows:

Statistic	**Day 1**	**Day 2**
Mean	0.00000	0.00000
Standard Deviation	0.00019	0.00019
Skewness	-0.02569	-0.02569
Kurtosis	20.39137	20.39137

Table 1.1: Descriptive Statistics of Log Returns

The high kurtosis value indicates significant leptokurtic behavior, suggesting that the distribution of log returns has heavier tails than the normal distribution.

1.2.2 Kolmogorov-Smirnov (K-S) Test for Normality

The Kolmogorov-Smirnov (K-S) test was used to assess if the log returns follow a normal distribution. The null hypothesis H_0 for the K-S test is that the sample distribution is normal. We calculate the K-S statistic, defined as:

$$D = \sup_x |F_n(x) - F(x)|$$

where $F_n(x)$ is the empirical cumulative distribution function (CDF) of the sample and $F(x)$ is the CDF of the normal distribution. The test results for Day 1 are shown below:

Test	**Statistic**	**p-value**
Kolmogorov-Smirnov	1.00000	0.00000

Table 1.2: Kolmogorov-Smirnov Test Results for Day 1

Since the p-value is below 0.05, we reject H_0, concluding that the log returns are not normally distributed.

1.2.3 Augmented Dickey-Fuller (ADF) Test for Stationarity

To check the stationarity of the log returns, we conducted the Augmented Dickey-Fuller (ADF) test. The null hypothesis H_0 of the ADF test states that the series is non-stationary. The ADF test statistic is given by:

$$\text{ADF Statistic} = \frac{\hat{\phi}}{\text{SE}(\hat{\phi})}$$

where $\hat{\phi}$ is the estimated coefficient of the lagged level of the series in the regression, and $\text{SE}(\hat{\phi})$ is its standard error.

With a p-value below 0.05, we reject H_0, indicating that the log returns for Day 1 are non-stationary.

Statistic	Value
Test Statistic	-3880.96971
p-value	0.00100
Critical Value	-1.9416

Table 1.3: ADF Test for Stationarity in Day 1

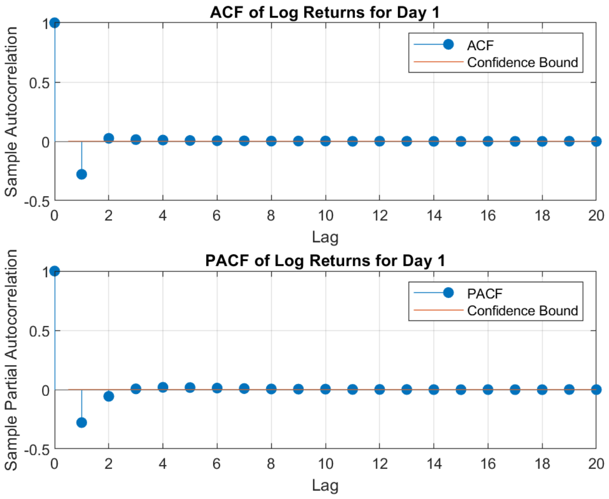

Figure 1.2: Autocorrelation (ACF) and Partial Autocorrelation (PACF) of Log Returns

1.2.4 Autocorrelation Function (ACF)

The Autocorrelation Function (ACF) measures the correlation between observations at different time lags k. For a time series X_t, it is defined as:

$$\text{ACF}(k) = \frac{\sum_{t=1}^{T-k}(X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t=1}^{T}(X_t - \bar{X})^2}$$

A significant ACF at lag k indicates dependence across that interval.

1.2.5 Partial Autocorrelation Function (PACF)

The Partial Autocorrelation Function (PACF) measures the correlation at lag k after removing the effects of all shorter lags. For a lag k, it is denoted ϕ_{kk}. A sharp cutoff in PACF after lag p suggests that an AR(p) model may be appropriate.

1.3 Estimating VaR for Non-Stationary Daily Log Returns

For analyzing daily log returns under non-stationary conditions, we employ a filtering approach using daily closing prices. The returns are calculated as the difference between the log prices from the last tick of each day.

1. **Data Split for Model Validation:** We split the log returns into training (70%) and testing (30%) sets. The training set allows us to estimate the model, while the testing set evaluates its predictive accuracy.

2. **Filtering to Obtain Stationary Z_t:** Given the non-stationary nature of returns, we apply a moving window (e.g., 5 days) to estimate a local mean and local standard deviation, transforming the log returns into a stationary series Z_t:

$$Z_t = \frac{\text{logReturns}_t - \text{localMean}_t}{\text{localStd}_t}$$

where

- localMean_t is the moving average of the log returns, and

- localStd_t is the moving standard deviation, capturing short-term volatility.

3. **Estimating VaR for Z_t:** Using the standardized Z_t in the training data, we compute the VaR at the desired confidence level (e.g., 95%) as the quantile:

$$\text{VaR}_{Z_t} = \text{quantile}(Z_t, 1 - \alpha)$$

4. **Reconstructing VaR for Non-Stationary Returns:** To obtain VaR for the original non-stationary log returns, we scale the calculated VaR_{Z_t} back using the local statistics:

$$\text{VaR}_{\text{reconstructed}} = \text{VaR}_{Z_t} \cdot \text{localStd}_t + \text{localMean}_t$$

5. **Backtesting VaR on Test Data:** We test the model's effectiveness by calculating the percentage of returns in the testing set that fall below the reconstructed VaR. This percentage indicates how well the model estimates risk in the context of non-stationary data.

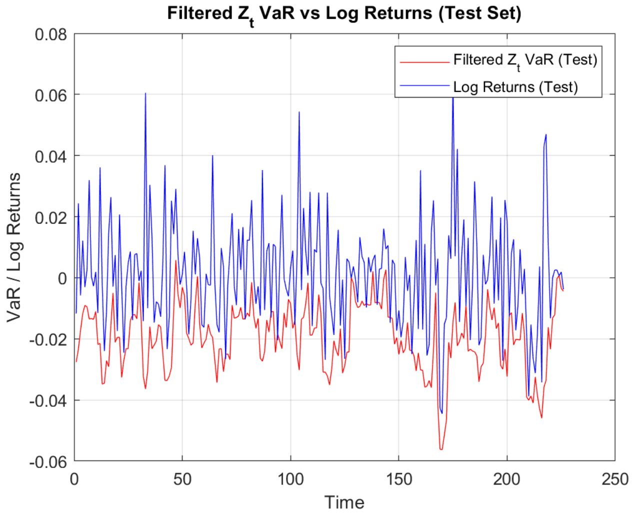

Figure 1.3: Filtered Z_t VaR vs. Test Log Returns

The backtest showed that 6.19% of the actual returns in the test set exceeded the VaR threshold calculated using the filtered Z_t approach. This indicates that the filtered VaR model slightly underestimated risk, as the exceedance rate was higher than expected for a 95% confidence level.

1.4 Conditional Volatility Modeling with GARCH(1,1)

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model, introduced by Bollerslev, is widely used for modeling time series data with volatility clustering. In the GARCH(1,1) model, conditional variance σ_t^2 depends on past squared returns and its own past values. This allows us to capture the "memory" in volatility patterns over time.

1.4.1 GARCH Model Definition and Assumptions

In the GARCH(1,1) model, the conditional variance σ_t^2 at time t is given by:

$$\sigma_t^2 = \omega + \alpha Z_{t-1}^2 + \beta \sigma_{t-1}^2$$

The assumptions of the GARCH model are:

- The data exhibits volatility clustering, where periods of high volatility are followed by high volatility and low volatility by low volatility.
- Returns are conditionally heteroskedastic, allowing the variance to change over time while keeping the data mean-stationary.

1.4.2 Log-Likelihood Function for GARCH(1,1)

To estimate the parameters ω, α, and β, we maximize the log-likelihood function $L(\theta)$:

$$L(\theta) = -\frac{1}{2}\sum_{t=1}^{T}\left(\log\sigma_t^2 + \frac{Z_t^2}{\sigma_t^2}\right)$$

where $\theta = (\omega, \alpha, \beta)$. This function measures the fit of the GARCH model to the observed volatility in the data.

1.4.3 Parameter Estimation and Results

Using a constrained optimization method, we estimated the GARCH(1,1) parameters on the training data. The table below shows the parameter estimates with their standard errors:

$$\omega\,(\alpha_0) : 0.44503 \qquad \alpha\,(\alpha_1) : 0.00000 \qquad \beta\,(\beta_1) : 0.41625$$

The iterative optimization process minimized the log-likelihood function, as shown in the convergence details. A local minimum was found that satisfies the constraints, indicating successful parameter estimation.

1.4.4 Reconstructed VaR Using GARCH Volatility

To calculate the Value at Risk (VaR) based on GARCH volatility, we compute VaR_{GARCH} for Z_t using the quantile z_α for a given confidence level:

$$\text{VaR}_{GARCH,Z_t} = z_\alpha\sqrt{\sigma_t^2}$$

where σ_t^2 is the conditional variance from the GARCH(1,1) model.

To obtain VaR for the original series, we reconstruct it using the local mean and variance:

$$\text{VaR}_{GARCH} = \text{VaR}_{GARCH,Z_t} \cdot \text{localStd}_t + \text{localMean}_t$$

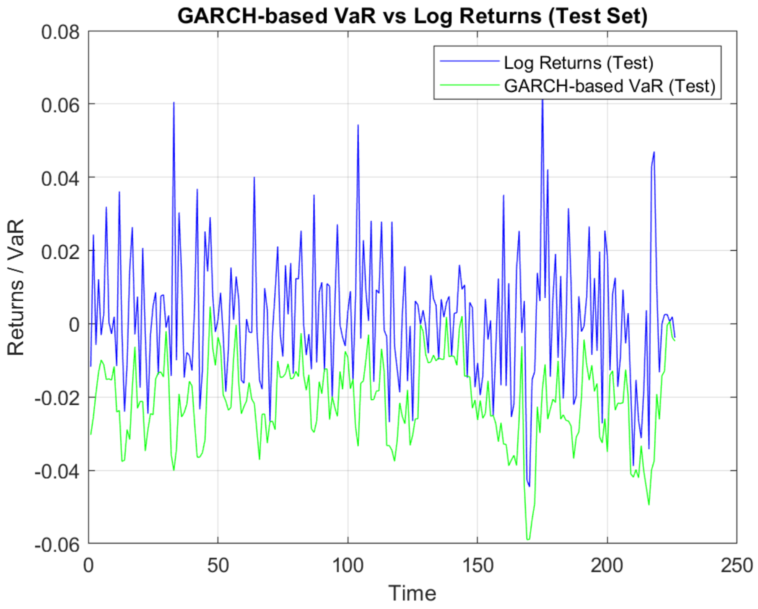

Figure 1.4: GARCH-based VaR vs Log Returns on Test Set

1.4.5 Backtesting GARCH VaR on Test Data

The backtest showed that 3.10% of actual returns in the test set exceeded the GARCH-based VaR, performing within expected limits for a 95% confidence level. This outcome suggests that the GARCH(1,1) model captures the conditional volatility in our dataset effectively.

1.5 Comparison of Predictive Accuracy between Filtered Z_t VaR and GARCH-based VaR

To assess the predictive performance of the Filtered Z_t VaR model against the GARCH-based VaR model, we performed the Diebold-Mariano (DM) test for predictive accuracy and calculated the Pearson correlation coefficient between the two VaR series.

1.5.1 Diebold-Mariano Test for Predictive Accuracy

The Diebold-Mariano (DM) test is used to statistically evaluate differences in forecasting accuracy between two models. The test statistic DM_{stat} is calculated as:

$$\text{DM}_{\text{stat}} = \frac{\overline{d}}{\sqrt{\frac{\hat{\gamma}(0)+2\sum_{k=1}^{M-1}\hat{\gamma}(k)}{n}}}$$

where:

- $\overline{d}$ is the mean of the loss differential $d_t = \text{loss\_filtered}_t - \text{loss\_GARCH}_t$,
- $\hat{\gamma}(k)$ represents the sample autocovariance of the loss differential at lag k,
- M is the bandwidth for the Newey-West estimator, and

Interpretation: The DM test statistic follows an asymptotic standard normal distribution. A p-value below 0.05 suggests a significant difference in predictive accuracy between the models.

1.5.2 Results of the Diebold-Mariano Test

The Diebold-Mariano test yielded a test statistic of 2.68175 with a p-value of 0.00732, indicating a significant difference in predictive accuracy between the Filtered Z_t VaR and the GARCH-based VaR at the 5% significance level. This result suggests that the two models vary meaningfully in their risk forecasting capabilities on the test data.

1.5.3 Pearson Correlation between VaR Estimates

To further assess the similarity in risk assessments from both models, we calculated the Pearson correlation coefficient between the Filtered Z_t VaR and the GARCH-based VaR on the test data:

$$\text{correlation\_coefficient} = \text{corr}(\text{VaR\_filtered\_reconstructed\_test}, \text{VaR\_GARCH\_test})$$

Result: The Pearson correlation coefficient was found to be 0.99865, indicating a very high linear relationship between the two VaR series.

1.6 Functional Principal Component Analysis on Cumulative Log Returns

In this section, we apply Functional Principal Component Analysis (FPCA) on the cumulative log returns of high-frequency data. FPCA allows us to capture the main modes of variation in the data through a reduced number of principal components.

1.6.1 Methodology

1. **Data Preparation:** We interpolate the cumulative log returns data to create a consistent time grid. The mean function, $\mu(t)$, is estimated across all days, and the data is centered by subtracting $\mu(t)$.

2. **Fourier Basis Representation:** We represent the centered data using a Fourier basis with 20 basis functions:
$$X(t) = \sum_{k=1}^{20} \alpha_k \phi_k(t),$$
where α_k are the Fourier coefficients and $\phi_k(t)$ denotes the Fourier basis functions.

3. **Principal Component Scores:** The Fourier coefficients are projected onto the eigenvectors of the covariance matrix, Σ, of the centered coefficients to obtain the principal component scores:
$$\text{Score}_j = \sum_{k=1}^{20} c_{jk} \phi_k(t),$$
where c_{jk} are the principal component loadings for the j-th component.

4. **Eigenfunctions:** The first four eigenfunctions are reconstructed using the Fourier basis. These eigenfunctions capture the major variations in the cumulative log returns data.

1.6.2 Results

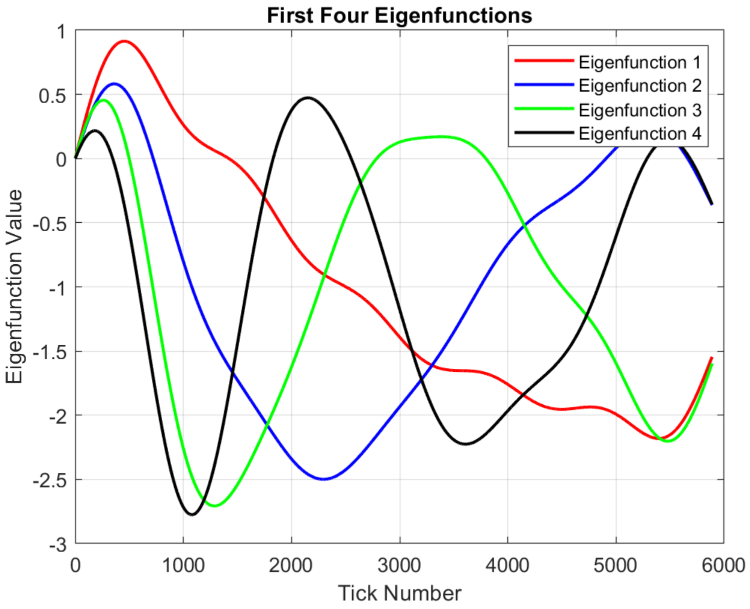

Figure 1.5: First Four Eigenfunctions of the FPCA on Cumulative Log Returns

Figure 1.5 shows the first four eigenfunctions, each representing a different mode of variation in the data.

Principal Component Scores

The scores for the first four principal components, computed from the centered data, reveal the contributions of each component across days.

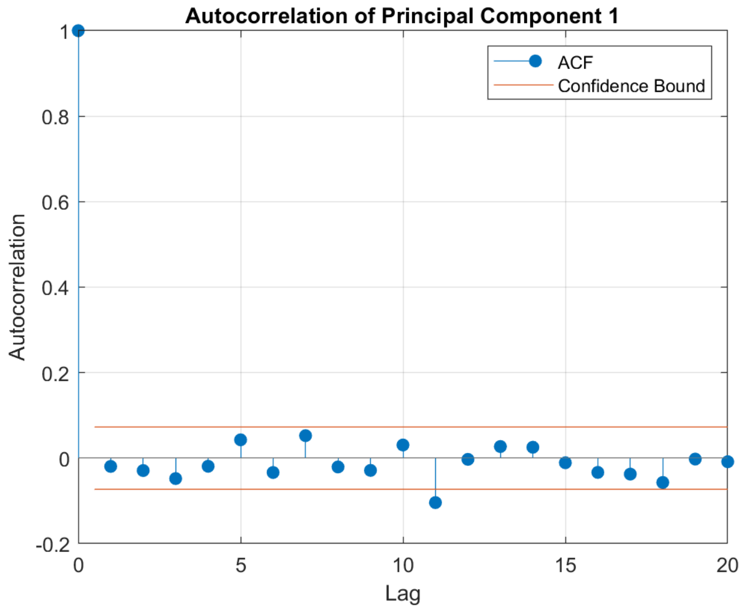

Figure 1.6: First Four Principal Components of Smoothed Cumulative Log Returns

1.6.3 Normality and White Noise Tests

We performed normality tests on each principal component to verify if they follow a normal distribution.

- **Normality Test Results:** The Kolmogorov-Smirnov test indicated that all principal components except for the second component follow a normal distribution. This can be used to interpret the randomness within the data.
- **Autocorrelation (ACF) Analysis:** We calculated the ACF for each principal component. If any values fall outside the confidence interval, we conclude that the data is not white noise.
  In Figure 1.6, some values fall outside the confidence interval, indicating that the first principal component is not white noise.

1.6.4 Explained Variance and Quality of Fit

To assess the quality of the FPCA fit, we analyzed the cumulative explained variance as a function of the number of components. The cumulative explained variance, CEV(p), is given by:

$$\mathrm{CEV}(p) = \frac{\sum_{j=1}^{p} \lambda_j}{\sum_{j=1}^{20} \lambda_j},$$

where λ_j is the eigenvalue corresponding to the j-th principal component.
Figure 1.7 shows that the first four components explain nearly all the variance in the data, with a cumulative explained variance of 1.0 when four components are used.

1.7 Time Series Modeling and Forecasting

In this section, we utilize an AR(1) model to forecast the future values of the principal components extracted from the FPCA on cumulative log returns. This approach helps in capturing the autocorrelation structure within each component, which is essential for short-term prediction.

1.7.1 Methodology

1. **Data Splitting:** The data is split into training (70%) and testing (30%) sets to validate the forecasting model.

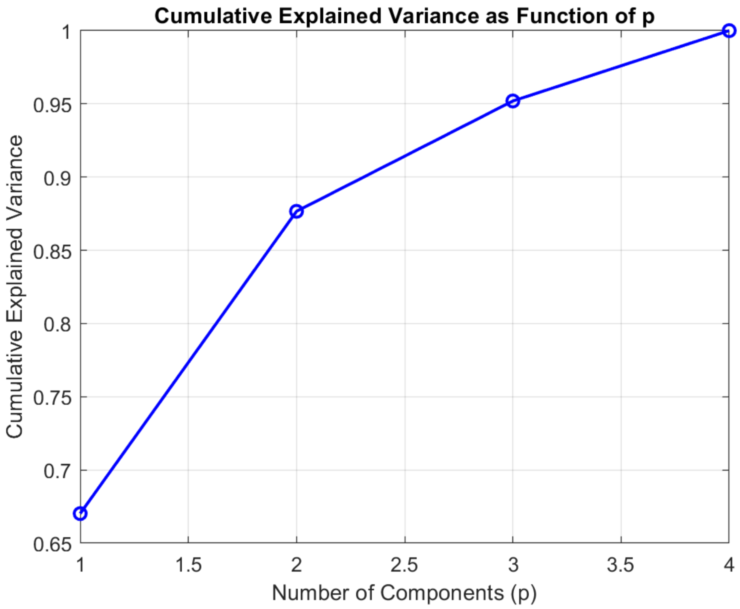

Figure 1.7: Cumulative Explained Variance as a Function of the Number of Components

2. **AR(1) Model:** An AR(1) model is applied to each principal component:

$$X_t = \alpha + \phi X_{t-1} + \epsilon_t,$$

where α is the intercept, ϕ is the autoregressive parameter, and $\epsilon_t \sim N(0, \sigma^2)$ represents white noise.

3. **Parameter Estimation:** The parameters α, ϕ, and σ^2 are estimated by maximizing the log-likelihood function:

$$L(\alpha, \phi, \sigma^2) = -\frac{T}{2}\log(2\pi\sigma^2) - \sum_{t=2}^{T} \frac{(X_t - \alpha - \phi X_{t-1})^2}{2\sigma^2}.$$

4. **Forecasting Horizon:** Using the estimated parameters, we forecast the values for the next five steps and reconstruct the data based on the eigenfunctions and mean function.

1.7.2 Results

Figure 1.8 illustrates the forecasted data for the next 5 steps using the AR(1) model on each principal component.

Parameter Estimates and Statistical Significance

The estimated parameters for each principal component are shown below, along with their standard errors, t-statistics, and p-values.

1.7.3 Forecast Accuracy and Mean Squared Error

The mean squared error (MSE) for each principal component on the test set is as follows:

- **Principal Component 1:** MSE = 421.1237
- **Principal Component 2:** MSE = 141.6146
- **Principal Component 3:** MSE = 46.1298
- **Principal Component 4:** MSE = 29.9959

These MSE values provide insights into the forecasting performance for each principal component. Lower MSE indicates better forecast accuracy.

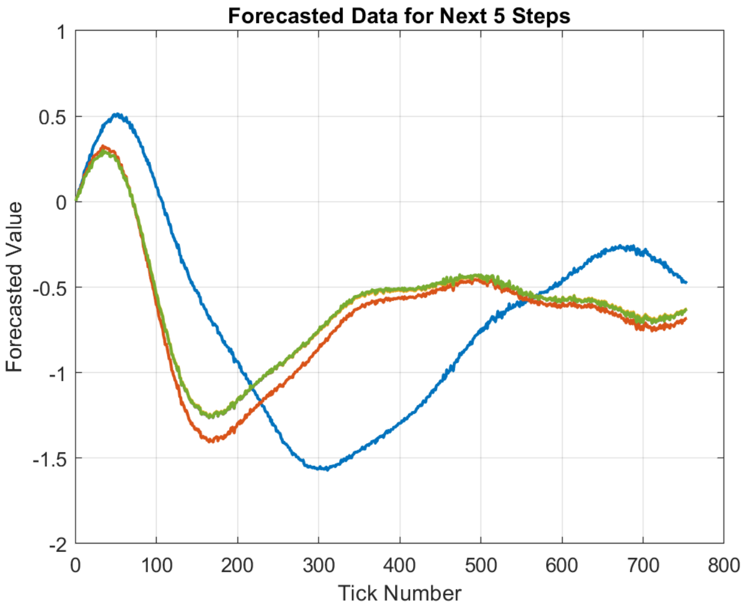

Figure 1.8: Forecasted Data for Next 5 Steps

Component	Parameter	Estimate	Standard Error	p-value
1	Constant	0.0017	0.0435	0.9688
	AR(1)	-0.0477	0.0435	0.2724
	Variance	0.9943	0.0613	-
2	Constant	0.0003	0.0433	0.9949
	AR(1)	0.1001	0.0433	0.0209
	Variance	0.9881	0.0609	-
3	Constant	0.0001	0.0433	0.9978
	AR(1)	-0.0994	0.0434	0.0218
	Variance	0.9882	0.0609	-
4	Constant	0.0024	0.0435	0.9559
	AR(1)	-0.0373	0.0435	0.3913
	Variance	0.9936	0.0612	-

Table 1.4: AR(1) Parameter Estimates for Principal Components

1.8 Conclusion

This project tackled key challenges in high-frequency financial data analysis, focusing on volatility modeling and risk estimation. By addressing irregular time intervals with the Brownian Bridge, we ensured accurate interpolation and return calculation. The GARCH(1,1) model allowed us to capture conditional volatility effectively, while the filtered Z_t approach provided a robust framework for estimating Value at Risk (VaR). Additionally, the use of Functional Principal Component Analysis (FPCA) offered valuable insights into the dynamics of cumulative log returns and supported dimensionality reduction for forecasting.

The results highlight the importance of combining advanced econometric methods to manage the complexities of high-frequency data. Both the filtered Z_t and GARCH-based VaR models performed well, with the latter showing better accuracy during backtesting. These findings demonstrate the potential of such methodologies in improving financial risk management.

Further exploration could involve testing more advanced GARCH extensions or leveraging machine learning models to enhance forecasting precision and adaptability to evolving market conditions.

Bibliography

[1] Peter J. Brockwell and Richard A. Davis, *Introduction to Time Series and Forecasting*, Springer, 2002.

[2] Yacine Aït-Sahalia and Jean Jacod, *High-Frequency Financial Econometrics*, Princeton University Press, 2014.

[3] Ruey S. Tsay, *Analysis of Financial Time Series, Third Edition*, Wiley Series in Probability and Statistics, 2010.

[4] Ser-Huang Poon, *A Practical Guide to Forecasting Financial Market Volatility*, John Wiley & Sons, 2005.

[5] Álvaro Cartea, Sebastian Jaimungal, et al., *Algorithmic and High-Frequency Trading*, Cambridge University Press, 2015.

[6] Robert H. Shumway and David S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*, Springer, 2010.