S&P 500 VOLATILITY ANALYSIS: BAYESIAN INFERENCE WITH ARCH, GARCH, AND STOCHASTIC VOLATILITY MODELS

December 2024

Contents

1 Introduction **1**

2 Data and Problem Description **1**

2.1 Dataset . . . 1

2.2 Analysis Problem . . . 1

3 ARCH: Autoregressive Conditional Heteroskedasticity Model of Volatility and GARCH: Generalised Autoregressive Conditional Heteroskedasticity Model **3**

3.1 Notations and Terminology . . . 3

3.2 ARCH Model . . . 3

3.2.1 Mean Square Error and ARCH/GARCH Models . . . 3

3.2.2 Prior Selection . . . 4

3.2.3 Code . . . 4

3.3 Results . . . 6

3.3.1 $\hat{R}$ (Gelman-Rubin statistic) Values, Convergence and Interpretation 6

3.3.2 ESS (Effective Sample Size) Analysis . . . 7

3.4 GARCH as an Extension of ARCH . . . 7

3.4.1 Prior Selection . . . 7

3.4.2 Code . . . 7

3.4.3 Results . . . 9

3.4.4 Posterior Predictive Checking ARCH and GARCH . . . 10

4 Stochastic Volatility Model **11**

4.1 Model Structure . . . 11

4.2 Running the Model . . . 11

4.2.1 Prior Selection . . . 11

4.2.2 Stan Code and MCMC Configurations . . . 12

4.3 Results . . . 15

4.3.1 $\hat{R}$ (Gelman-Rubin statistic) Values, Convergence and Interpretation 15

4.3.2 ESS (Effective Sample Size) Analysis . . . 15

4.3.3 Posterior Predictive Checking for Stochastic Volatility . . . 16

5 Prior Sensitivity Analysis **16**

5.1 Alternative 1: Priors Based on Financial Theory . . . 16

5.2 Alternative 2: Priors Informed by Empirical Studies . . . 17

5.3 Dummy Priors . . . 17

6 Discussions **18**

6.1 Model Comparison . . . 18

6.2 Problems and Possible Improvements . . . 19

7 Conclusion **20**

8 Self-Learning **20**

References **21**

A Appendix A: Additional Graphs **21**

1 Introduction

Understanding market volatility is a fundamental aspect of financial modeling. The S&P 500 index, a benchmark for U.S. financial markets, serves as a critical indicator of economic health and investor sentiment. This project aims to analyze the volatility of the S&P 500 by applying existing ARCH, GARCH and stochastic volatility models, complemented by Bayesian inference to enhance parameter estimation and uncertainty quantification.

The motivation for this project is that financial markets are inherently volatile, with periods of high uncertainty significantly impacting investment strategies and economic policies. Accurately analyzing volatility is essential for risk management, option pricing, and portfolio optimization. We employ ARCH, GARCH and stochastic volatility models, widely used in financial econometrics, which are effective for capturing volatility clustering, working by non-linearly modelling heteroscedasticity, which is used for heterogeneity of variance.

The initial challenge or problem for this project lies in determining whether relatively simple models, such as ARCH and GARCH, can adequately capture periods of high volatility and their underlying patterns in the S&P 500 data. To investigate this, we first applied these models to evaluate their performances. Once these models showed fairly good performances in exploring the data, we tested a more complex approach, the stochastic volatility model, to see if it is overly complicated in the context of this project or if it can make any additional improvements in terms of the inference.

This project utilizes the following models: ARCH, GARCH, and stochastic volatility. These models are applied to capture volatility clustering in the S&P 500 index data; Bayesian inference is used for parameter estimation, allowing us to incorporate prior knowledge and quantify the uncertainty in the model outputs. By applying these established techniques, we aim to provide insights into the dynamic behavior of market volatility.

Illustrative Figures. To provide an understanding of the data, Figure 1 shows the historical closing prices of the S&P 500 index. Additionally, Figure 2 illustrates the daily log returns to show periods of high and low volatility.

2 Data and Problem Description

2.1 Dataset

The source of information for the data is LSEG Eikon Data API. For retrieving data on the S&P index the ticker $.SP500Q$ is used which tracks the EOD closing price for the index.

2.2 Analysis Problem

Our analysis differs from existing work, such as “Volatility Clustering in the S&P500 Market Index: An ARCH/GARCH Approach" by Heng Sun & Bing Zhang [1] and “Pricing

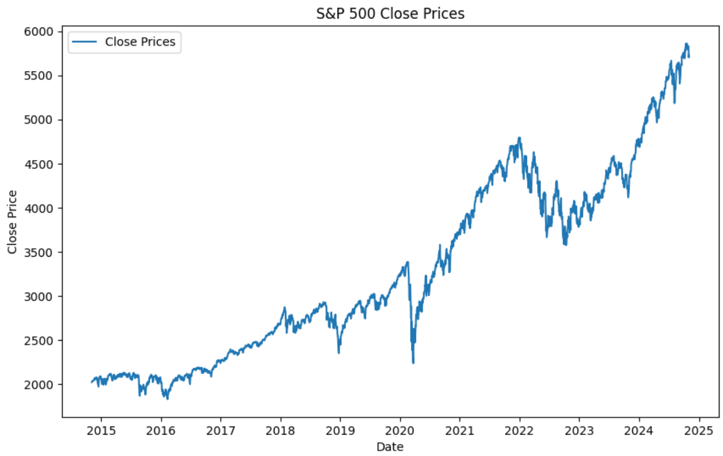


Figure 1: Historical closing prices of the S&P 500 index, showing the general upward trend and fluctuations.

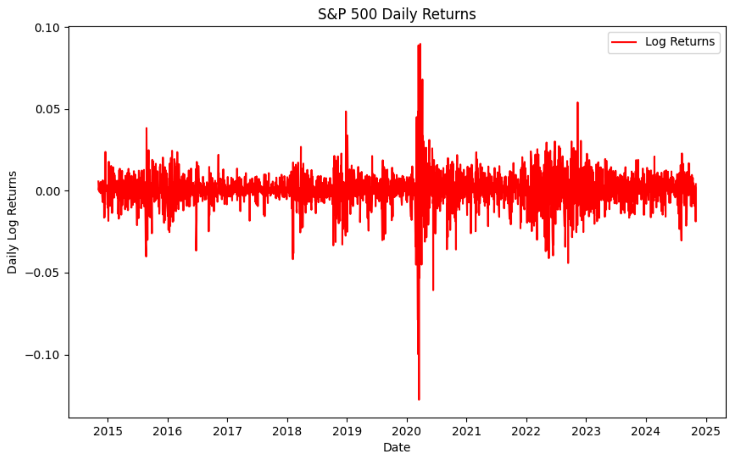


Figure 2: Daily log returns of the S&P 500 index, illustrating periods of high and low volatility over time.

S&P 500 Index Options under Stochastic Volatility with the Indirect Inference Method" by Jinghong Shu & Jin E. Zhang [2] in several key ways. First and foremost, unlike the existing analyses, we incorporate Bayesian inference in the workflow to estimate model parameters. This approach allows us to quantify parameter uncertainty and integrate prior information, thereby enhancing the robustness and interpretability of the results. Second, we are not limiting our analysis to either ARCH/GARCH models or to the stochastic volatility model. Instead, this work compares all three models to see how different approaches of using volatility to estimate stock market prices perform with the same dataset. Finally, this work focuses on the data from the last decade, meaning the dataset includes stock prices which are affected by events like the Covid-19 pandemic or the 2020 US elections. This is a specifically tough period to estimate due to the abundance of external events with a high effect on the stock market.

3 ARCH: Autoregressive Conditional Heteroskedasticity Model of Volatility and GARCH: Generalised Autoregressive Conditional Heteroskedasticity Model

3.1 Notations and Terminology

We define the volatility, σ_t as the standard deviation of returns for a time series. The term heteroskedasticity expresses the changing volatility σ_t over time for a predicted variable, in this case the volatility, which is treated as non-constant for a heteroskedasticity model.

3.2 ARCH Model

The ARCH model predicts the conditional variance of volatility, $Var(y_t|y_{t-1})$, of a time series based on past data. It assumes the volatility at a given time depending on previous deviations (errors) from the predicted values.

Mathematically, the general ARCH model is given as:

$$Var(y_t|y_{t-1}) = \sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2,$$

the degree q corresponds to the degree of influence that the preceding time periods have by modeling the squared returns of the previous q periods. For example, ARCH(1) relates the volatility of the squared returns to the previous period, ARCH(2) relates the volatility of the squared returns to the previous two periods, and so on. Here ε^2 refers to the squared error of previous predictions. $\alpha_0, 1$ are the learned parameters of the ARCH model. α_0 is the base volatility expected given any time $t, t-1$ where t is the current time period. α_1 controls how much the mean squared error from previous predictions, ε^2 affects the current volatility.

3.2.1 Mean Square Error and ARCH/GARCH Models

A model's fit to new data can be summarized by its deviation from the mean squared error. The deviation mean square of the predicted data from the actual data can be given as $\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \mathrm{E}\left(y_i \mid \theta\right)\right)^2$ where θ is the parameter dependence of the model. The weighted mean can be given as $\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \mathrm{E}\left(y_i \mid \theta\right)\right)^2 / \operatorname{var}\left(y_i \mid \theta\right)$. Mean squared errors are easy to compute; however, they are less appropriate for models that are not described well by a normal distribution, which is a trend in data represented by the ARCH model: they tend to have thicker tails or heads.

3.2.2 Prior Selection

By choice, having more informative priors improves MCMC sampling efficiency, leading to faster convergence and more accurate posterior distributions. Hence, by incorporating information from our knowledge domain, we come to better overall results.

For the parameters mentioned in Listing 1: μ the mean, α_0 the base volatility, and α_1 the learned ARCH parameter we consider informative priors based on previous S&P 500 analysis for the time period 2012-2020 [3]. Following this μ, the mean of log returns, is given as: $\mu \sim N(4.16e-4, 1.5 * 1.12e-3)$. Where $4.16e-4$ is the mean and $1.12e-3$ is the standard deviation of log returns in the dataset. The prior for α_0 is $N(0.1, 0.05)$ and α_1 is $Beta\ (2,5)$. We take these priors approximating values similar to those in [3] by setting the priors so that the mean of the distributions for the priors are similar to those specified in the literature. The multiplicative constant (1.5) allows for exploration while running the MCMC chains.

3.2.3 Code

Listing 1: ARCH(1) Model Implementation in Stan

```
data {
  int<lower=0> T;             // Number of observations
  vector[T] y;                // Returns data
  real prior_mean_mu;         // Informative prior mean for mu
  real<lower=0> prior_sd_mu; // Informative prior SD for mu
}
parameters {
  real mu;                    // Mean return
  real<lower=0> alpha0;       // Base volatility
  real<lower=0,upper=1> alpha1; // ARCH parameter
}
model {
  vector[T] h;  // Conditional variances - log volatilities

  // Priors
  mu ~ normal(prior_mean_mu, 1.5 * prior_sd_mu); // Informative prior for mean
  alpha0 ~ normal(0.1, 0.05);              // Base volatility prior/weakly informative prior
  alpha1 ~ beta(2, 5);                     // ARCH parameter prior/informative prior

  // ARCH(1) model
  h[1] = alpha0 / (1 - alpha1); // Stationary variance initialization
  for (t in 2:T) {
    h[t] = fmax(alpha0 + alpha1 * square(y[t-1] - mu), 1e-8); // Ensure positivity
  }

  // Likelihood
  for (t in 1:T) {
    y[t] ~ normal(mu, sqrt(h[t]));
  }
}
```

```
generated quantities {
  vector[T] volatility;
  vector[T] y_rep;  // Posterior predictive samples
  vector[T] log_lik; // Log-likelihood for LOO-CV

  // Compute volatility (conditional standard deviations)
  volatility[1] = sqrt(alpha0 / (1 - alpha1));
  for (t in 2:T) {
    volatility[t] = sqrt(fmax(alpha0 + alpha1 * square(y[t-1] - mu), 1e-8));
  }

  // Generate predictive samples
  for (t in 1:T) {
    y_rep[t] = normal_rng(mu, volatility[t]);
  }

  // Compute log-likelihood
  for (t in 1:T) {
    log_lik[t] = normal_lpdf(y[t] | mu, volatility[t]);
  }
}
```

Listing 2: R Code for MCMC of ARCH Model

```
# Calculate Mean and SD for Priors
log_return_mean <- mean(data$Log_Returns, na.rm = TRUE)
log_return_sd <- sd(data$Log_Returns, na.rm = TRUE)

stan_data <- list(
  T = nrow(data),
  y = data$Log_Returns,
  prior_mean_mu = log_return_mean,
  prior_sd_mu = log_return_sd
)

# Compile Stan Model
arch_model <- cmdstan_model(
  "../BDA_project/1.0.arch_original.stan",
  force_recompile = TRUE,
  quiet = FALSE
)

# Explanation of MCMC options
cat("MCMC Inference:\n")
cat("- The model was run with 4 chains, each with 1000 warmup iterations and 2000
    post-warmup iterations.\n")
cat("- A seed value (4911) was used for reproducibility.\n")

# Fit the model
fit <- arch_model$sample(
  data = stan_data,
  seed = 4911,
  chains = 4,
```

```
  iter_warmup = 1000,
  iter_sampling = 2000,
  refresh = 100
)
```

3.3 Results

Table 1: MCMC Analysis for ARCH Results and Convergence Diagnostics

Param	Mean	Med	SD	MAD	Q5	Q95	$\hat{R}$	ESS_bulk	ESS_tail
lp__	1.04e4	1.04e4	1.26	1.04	1.04e4	1.04e4	1.00	2627	3673
μ	9.62e-4	9.61e-4	1.71e-4	1.69e-4	6.83e-4	1.24e-3	1.00	8432	6049
α_0	6.44e-5	6.44e-5	2.60e-6	2.62e-6	6.03e-5	6.88e-5	1.00	2534	2461
α_1	0.506	0.506	0.0443	0.0445	0.433	0.579	1.00	2463	2474
vol[1]	1.15e-2	1.14e-2	4.61e-4	4.51e-4	1.07e-2	1.23e-2	1.00	2883	2850
vol[2]	8.70e-3	8.70e-3	1.43e-4	1.42e-4	8.47e-3	8.94e-3	1.00	3105	4035
vol[3]	8.27e-3	8.27e-3	1.54e-4	1.53e-4	8.02e-3	8.53e-3	1.00	2702	2840
vol[4]	8.04e-3	8.04e-3	1.61e-4	1.61e-4	7.78e-3	8.31e-3	1.00	2543	2470
vol[5]	8.17e-3	8.17e-3	1.57e-4	1.58e-4	7.92e-3	8.43e-3	1.00	2626	2674
vol[6]	8.03e-3	8.03e-3	1.62e-4	1.62e-4	7.77e-3	8.30e-3	1.00	2536	2462

Note: The table represents only the first 6 iterations where vol[n] represents volatility estimates for different time periods. ESS_b and ESS_t represent the bulk and tail effective sample sizes respectively. All $\hat{R}$ values indicate proper convergence of the MCMC chains. As the time series data had more than 2500 data points, it is practically impossible to include the whole diagnostics summary which is over 10000 rows.

3.3.1 $\hat{R}$ (Gelman-Rubin statistic) Values, Convergence and Interpretation

The $\hat{R}$ values help determine if the MCMC chains converge correctly. When running MCMC sampling, we typically run multiple chains starting from different initial points. $\hat{R}$ compares two quantities: The average variance within each chain and the variance of the chain means. If the chains have converged properly, these two measures should be very similar. It follows from this $\hat{R}$ values close to 1.00 indicate well converging chains. Since all $\hat{R}$ values in the above fit are equal to 1.00 we can conclude the MCMC chains converge well.

We observe no divergent transitions in the model this is inline with the fact that the $\hat{R}$ values are consistently equal to 1.0. This means the model simulates the Hamiltonian dynamics well, zero divergences suggest that the sampler successfully explored the posterior distribution and the parameters used should be reliable including the step size. The tree depth is set by default to 10, the tree depth refers to the number of steps the sampler would take in either direction while building the No-U-Turn Sampler trajectory tree. The maximum tree depth of 10 allows us to explore the posterior space consistently and further changes are not required as the chain converges well.

3.3.2 ESS (Effective Sample Size) Analysis

The bulk ESS values range from 2300 to 9026. These values are consistently good as they are above 400 and even the lowest values are sufficently independent samples for reliable inference. We can see that the tail ESS values also remain consistently high, hence showing that the sampling efficiency for the tails is well accounted for.

3.4 GARCH as an Extension of ARCH

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model extends ARCH accounting for the persistence of volatility. While ARCH models the conditional variance based solely on past squared errors, GARCH incorporates both past squared errors and past conditional variances.

The GARCH(p,q) model can be expressed mathematically as:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2 \tag{1}$$

where p represents the order of the GARCH terms (σ_{t-j}^2), q represents the order of the ARCH terms (ε_{t-i}^2), $\alpha_0 > 0$ is the baseline volatility, $\alpha_i \geq 0$ are the ARCH parameters and $\beta_j \geq 0$ are the GARCH parameters.

The addition of the $\beta_j \sigma_{t-j}^2$ terms allows to model latent parameter of the dataset(the volatility directly), by modeling the influence of the preceding periods volatility. IT hence also can take into account volatility shocks that eventually decay and prevents the model from predicting explosive variance patterns, unlike ARCH.

3.4.1 Prior Selection

GARCH is an extension of the ARCH model, this validates the use of the same priors of the ARCH parameters including μ, α_0, α_1. For the GARCH parameter β_1 the prior $beta_1 \sim (5,2)$ is choosen as it is defined on the interval [0,1], which matches our requirement that $0 \leq \beta < 1$. It has a mean of 5/(5+2) $\approx$ 0.714 which is close to the value in the literature [3]. Additionally it is right skewed and less dense around 1 and 0.

3.4.2 Code

Since ARCH is closely related to GARCH only those parts that differentiate ARCH from GARCH are mentioned in this section.

Listing 3: Stan Code for the GARCH - Only parts that differ from ARCH

```
data {
    // Same as ARCH
}

parameters {
```

```
  /// ... rest parameters same as for ARCH
    real<lower=0, upper=1> beta1;     // Coefficient for lagged conditional
        variance

}

transformed parameters {
  vector[N] volatility;         // Conditional variances

  // Initialize the first conditional variance
  volatility[1] = alpha0 / (1.0 - alpha1 - beta1);  // Stationary assumption
  volatility[1] = sqrt(fmax(volatility[1], 1e-8));  // Ensure positivity

  // Calculate the conditional variances
  for (n in 2:N) {
    volatility[n] = alpha0 + alpha1 * square(y[n-1] - mu) + beta1 *
        square(volatility[n-1]);
    volatility[n] = sqrt(fmax(volatility[n], 1e-8));          // Ensure
        positivity
  }
}

model {

  mu ~ normal(prior_mean_mu, 1.5 * prior_sd_mu);                //
      Informative prior for mean
  alpha0 ~ normal(0.1, 0.05);        // Weakly informative prior
  alpha1 ~ beta(2, 5);               // Informative prior for ARCH term
  beta1 ~ beta(5, 2);                // Informative prior for GARCH term

  // Likelihood
  for (n in 1:N) {
    y[n] ~ normal(mu, volatility[n]);
    target += normal_lpdf(y[n] | mu, volatility[n]);
  }
}

///Rest is same as ARCH
```

```
#  ... Rest is same as ARCH
# Calculate Mean and SD for Priors
log_return_mean <- mean(data$Log_Returns, na.rm = TRUE)
log_return_sd <- sd(data$Log_Returns, na.rm = TRUE)

# Prepare Data for GARCH Model
stan_data <- list(
  N = nrow(data),
  y = data$Log_Returns,
  prior_mean_mu = log_return_mean,
  prior_sd_mu = log_return_sd
)
# ... repeated parts from ARCH
# Fit the model
```

```
fit <- garch_model$sample(
  data = stan_data,
  seed = 4911,
  chains = 4,
  iter_warmup = 1000,
  iter_sampling = 2000,
  refresh = 100
)
# ... Rest Remains same as ARCH
```

Listing 4: R Code for MCMC of GARCH

3.4.3 Results

Table 2: Convergence Diagnostics for GARCH

Param	**Mean**	**Med**	**SD**	**MAD**	**Q5**	**Q95**	$\hat{R}$	**ESS Bulk**	**ESS Tail**
lp__	1.90e+4	1.90e+4	1.42	1.19	1.90e+4	1.90e+4	1.00	2378.0	3412.0
μ	7.91e-4	7.92e-4	0.000104	0.000102	6.20e-4	9.58e-4	1.00	7738.0	5725.0
alpha0	4.04e-6	4.02e-6	0.000000436	0.000000434	3.34e-6	4.77e-6	1.00	1707.0	2249.0
alpha1	1.84e-1	1.83e-1	0.0138	0.0134	1.61e-1	2.07e-1	1.00	1614.0	1940.0
beta1	7.80e-1	7.81e-1	0.0144	0.0142	7.56e-1	8.03e-1	1.00	1454.0	1697.0
volatility[1]	1.07e-2	1.06e-2	0.000751	0.000697	9.58e-3	1.20e-2	1.00	4465.0	4288.0
volatility[2]	9.86e-3	9.78e-3	0.000628	0.000589	8.95e-3	1.10e-2	1.00	4937.0	4408.0
volatility[3]	9.03e-3	8.96e-3	0.000531	0.000496	8.27e-3	9.98e-3	1.00	5492.0	4892.0
volatility[4]	8.23e-3	8.17e-3	0.000454	0.000425	7.58e-3	9.03e-3	1.00	5942.0	4984.0
volatility[5]	7.61e-3	7.56e-3	0.000388	0.000368	7.05e-3	8.30e-3	1.00	6114.0	4715.0

Note: This table presents convergence diagnostics for the stochastic volatility model parameters. All $\hat{R}$ values near 1.0 indicate proper convergence of the MCMC chains, while the high ESS values suggest reliable posterior estimates.

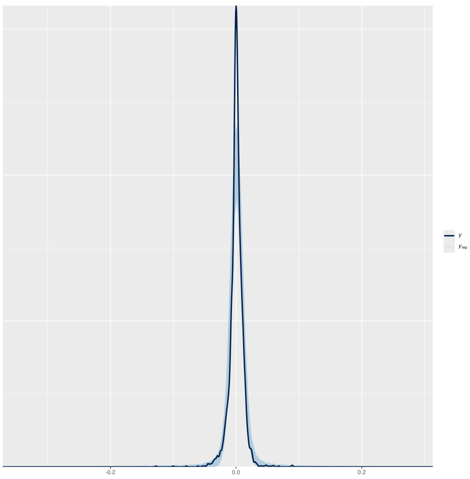
$\hat{R}$ (Gelman-Rubin statistic) Values, Convergence and Interpretation

We observe once again the value of $\hat{R}$ are close to 1.00 and for the above-mentioned results we conclude once again the parameters are reliable and the chains converge well. GARCH follows it from ARCH, where we had no transitions that deviate. All the chains converge in GARCH, the HMC dynamics are functioning as they should and the MCMC process has run correctly. There is no need to change the default tree depth and other parameters as well.

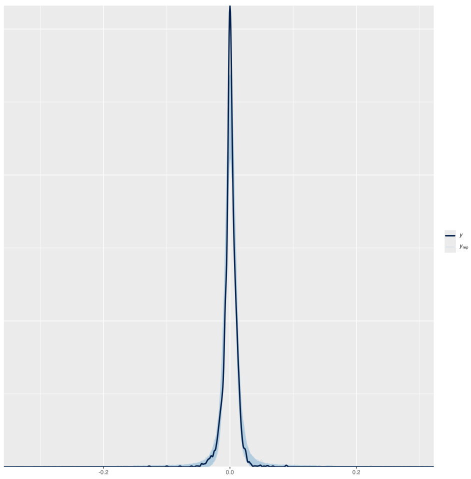
ESS Analysis

The ESS Bulk and ESS Tail for the log posterior of the GARCH model is 2378.0 and 3412.0 respectively as can be referred from the table above. These values suggest that sampling values in the tail are more reliable than in the bulk. Overall the ESS values for the volatility chains [1] to [7386] (not present in the table above) remain above 4000 suggesting the posterior relates well with the data. The global parameters (α_0 and α_1 and β_1) have lower ESS values than the observed parameters for which ESS values improve over the number of chains. In general, ESS values for the GARCH model suggest that there is no issue in the exploration of the distribution neither for the core parts nor for the extremities.

3.4.4 Posterior Predictive Checking ARCH and GARCH



(a) Posterior Predictive Check for ARCH



(b) Posterior Predictive Check for GARCH

Figure 3: Comparison of Posterior Predictive Checks for ARCH and GARCH Models

Figure 3 shows the posterior predictive check for the ARCH and GARCH volatility models. We plot the density of draws of daily log returns from the posterior distribution against the data. On the figure, the dark blue line represents the data while the light blue field represents the samples from the posterior distribution. As you can see, the distribution seems to explore the data very closely. While this might signify that the set of priors and stochastic volatility model structure used in this project can be suitable for the analysis goals, it can also mean that there is overfitting.

However, looking at the results of Leave-One-Out Cross-Validation at Table 4, shows that there isn't a risk of overfitting happening. Instead, both ARCH and GARCH performed quite well with high Expected Log Predictive Density and low Effective Number of Parameters. However, to distinguish between two it is possible to say that GARCH performed better. This is supported both by Figure 3 and the small difference in LOO-CV results. In Figure 3, the GARCH explored the peaks of the data better than ARCH while it also has better values in LOO-CV.

4 Stochastic Volatility Model

4.1 Model Structure

Like ARCH and GARCH models, the stochastic volatility model has volatility, σ_t, which is the standard deviation of returns as its focus point. However, the approach of stochastic volatility to the volatility term is completely different from the approaches of ARCH/GARCH models. Unlike their deterministic idea regarding the volatility and returns of the stock market, stochastic volatility assumes that the volatility is by nature stochastic and incorporated with random noise, meaning that it evolves randomly and cannot be directly predicted by using other external parameters. In stochastic volatility, the volatility is also latent, unlike ARCH and GARCH. This means that volatility is not directly calculated from the past data of returns but instead, it evolves in the stochastic process where each volatility is connected to past volatilities, as well as the randomness factor.

When it comes to the mathematical representation of the stochastic volatility model, there are three main variables used to approximate volatility, h, and daily log-returns of S&P 500, y: μ, mean log volatility, ϕ, the persistence of volatility, and σ, volatility of volatility. There are also other variables, such as ε, multiplicative error / white-noise shock, and δ, shock on volatility [4].

$$y_t = \varepsilon_t \exp(h_t/2) \tag{2}$$

$$h_{t+1} = \mu + \phi(h_t - \mu) + \delta_t \sigma \tag{3}$$

$$h_1 \sim normal\left(\mu, \frac{\sigma}{\sqrt{1-\phi^2}}\right) \tag{4}$$

$$h_t \sim normal(\mu + \phi(h_{t-1} - \mu), \sigma) \tag{5}$$

$$y_t \sim normal(0, \exp(h_t/2)) \tag{6}$$

As stated at Equation 5 and Equation 6, both volatility and daily log-returns modelled with normal distribution where μ, ϕ, and σ are used as parameters of the normal distributions.

4.2 Running the Model

4.2.1 Prior Selection

As it is with ARCH and GARCH models, the aim of the stochastic volatility model is to have informative or at least weakly informative priors when possible. The difference is, however, the existence of different parameters such as volatility of volatility. Estimating those numbers directly from the data is not straightforward, therefore, weakly informative priors are used. Those priors are, in this case, generally concluded by examining the graphs and combining it with the domain knowledge regarding the behaviours of S&P 500.

For the parameter μ, we used an informative prior of normal distribution centred around the standard deviations of the daily log-returns where the scale parameter is decided to be

0.025, as can be concluded from the Figure 2, where the majority of the movement regarding the daily log-returns happen in the range 0.025. For the parameter ϕ, we used a weakly informative prior $normal(0.5, 0.1)$, due to the domain knowledge regarding the tendency of the volatility of a day affecting the day after that. With this prior, the distribution will favour positive tendencies while still keeping it under balance in the sense that it won't let past volatility strongly skew future volatility. With a similar logic, it is concluded that volatility of volatility, σ, should be rather a small number which can be modelled by $cauchy(0, 0.1)$. This is especially logical since S&P 500 is not a company stock but it is a very big index, it is rather unlikely to have big shocks to the volatility. As it is financially intuitive, it is expected to keep a big index fund such as S&P 500 to stick to general trends as big shock movements in an index with a volume of a couple of trillion dollars would require movement of so much money.

4.2.2 Stan Code and MCMC Configurations

As seen in ARCH and GARCH, models in this project are coded directly in Stan and MCMC inferences are run by using cmdstanr.

```
data {
    int<lower=1> N;           // Number of observations
    vector[N] y;              // Observed log-returns
    real prior_mean_log_return; // Mean of our daily returns, we use this for
        meaningful sampling for daily log-returns
    real<lower=0> prior_sd_log_return; // Standard deviation of our daily
        returns (volatility), we use this for building informative prior
}

parameters {
    real mu;                  // Mean of the returns
    real<lower=-1,upper=1> phi;   // Persistence of volatility (AR1
        coefficient)
    real<lower=0> sigma_vol;      // Standard deviation of volatility process
        - white noise shock scale
    vector[N] h;              // Log-volatilities
}

transformed parameters {
    vector[N] volatility;         // Volatilities (exponential of
        log-volatilities)

    // Convert log-volatilities to volatilities
    volatility = exp(h / 2);
}

model {
    // Priors
    // INFORMATIVE PRIORS - CORRECT ONES
    // mu ~ normal(prior_sd_log_return, 0.05);   // Informative prior for mean
    // phi ~ normal(0.5, 0.1);;                     // Base volatility
        prior/weakly informative prior
    // sigma_vol ~ cauchy(0, 0.1);                 // Volatiltiy of
        volatility parameter prior/ informative prior
```

```
    //
        https://www.shs-conferences.org/articles/shsconf/pdf/2023/18/shsconf_fems2023_01077.pdf

    mu ~ normal(prior_sd_log_return, 0.025); // informative prior selection
        for mean log volatility
    phi ~ normal(0.5, 0.1);          // Prior for persistence coefficient
    sigma_vol ~ cauchy(0, 0.1);      // Half-cauchy prior for volatility of
        volatility

    mu ~ normal(prior_sd_log_return, 0.025); // informative prior selection
        for mean log volatility
    phi ~ normal(0.5, 0.1);          // Prior for persistence coefficient
    sigma_vol ~ cauchy(0, 0.1);      // Half-Normal prior for volatility of
        volatility

    // AR(1) process for log-volatilities
    h[1] ~ normal(mu, sigma_vol / sqrt(1 - phi * phi));

    for (t in 2:N) {
        h[t] ~ normal(mu + phi * (h[t-1] - mu), sigma_vol);
    }

    // Estimated returns on time t
    for (t in 1:N){
        y[t] ~ normal(prior_mean_log_return, exp(h[t] / 2));
    }
}

generated quantities {
    vector[N] y_rep;        // Posterior predictive samples
    vector[N] log_lik;      // Log-likelihood for LOO-CV

    // Generate predictive samples
    for (t in 1:N) {
        y_rep[t] = normal_rng(prior_mean_log_return, exp(h[t] / 2));
        log_lik[t] = normal_lpdf(y[t] | mu,  exp(h[t] / 2));
    }
}
```

Listing 5: Stochastic Volatility Model Implementation in Stan

```
# Calculate Mean and SD for Priors
log_return_mean <- mean(data$Log_Returns, na.rm = TRUE)
log_return_sd <- sd(data$Log_Returns, na.rm = TRUE)

# Prepare Data for Stochastic Volatility Model
stan_data <- list(
  N = nrow(data),
  y = data$Log_Returns,
  prior_mean_log_return = log_return_mean,
  prior_sd_log_return = log_return_sd
)

# Compile Stan Model
```

```
stochastic_volatility_model <- cmdstan_model("../BDA_project/
    stochastic_volatility_model.stan", force_recompile = TRUE,
    quiet = FALSE)

# Explanation of MCMC options
cat("MCMC Inference:\n")
cat("- The model was run with 4 chains, each with 1000 warmup
    iterations and 2000 post-warmup iterations.\n")
cat("- A seed value (4911) was used for reproducibility.\n")

# Fit the model
fit <- stochastic_volatility_model$sample(
  data = stan_data,
  seed = 4911,
  chains = 4,
  iter_warmup = 1000,
  iter_sampling = 2000,
  refresh = 100
)
```

Listing 6: R Code for MCMC of Stochastic Volatility Model

The exact model structure and MCMC parameters can be seen in Listing 5 and Listing 6, respectively. The reasoning behind these MCMC parameters is first their ease of use and second their reduced complexity. It is always possible to use more complex MCMC structures, however, with an already complicated model like stochastic volatility, it would require lots of computing power and time to run the model if the MCMC had more chains, higher tree depth, or more iterations.

4.3 Results

Table 3: Convergence Diagnostics for Stochastic Volatility

Param	Mean	Med	SD	MAD	Q5	Q95	$\hat{R}$	ESS Bulk	ESS Tail
lp__	13214.00	13209.25	192.89	196.59	12906.68	13540.51	1.06	65.61	240.87
μ	0.0113	0.0115	0.02	0.0247	-0.03	0.0512	1.00	9489.48	4971.08
ϕ	1	1	0.0003	0.0003	1	1	1.00	3190.45	3116.19
σ	0.2422	0.24	0.02	0.02	0.21	0.27	1.06	66.99	253.00
h[1]	-11.43	-11.46	0.55	0.55	-12.28	-10.49	1	8122.97	6604.24
h[2]	-11.49	-11.52	0.54	0.54	-12.34	-10.57	1.00	7952.55	6277.15
h[3]	-11.57	-11.6	0.54	0.54	-12.43	-10.65	1.00	7228.50	6668.71
h[4]	-11.61	-11.63	0.53	0.53	-12.46	-10.71	1.00	7044.94	6213.35
h[5]	-11.65	-11.68	0.53	0.53	-12.50	-10.74	1.00	6750.55	6476.77
h[6]	-11.66	-11.6	0.54	0.54	-12.53	-10.77	1.00	6449.24	6405.98

Note: The following table represents only the first 6 iterations where h[n] represents volatility estimates for different time periods. ESS represents the bulk effective sample size. All $\hat{R}$ values indicate proper convergence of the MCMC chains. As the time series data had more than 2500 data points, it is practically impossible to include the whole diagnostics summary which is over 10000 rows.

4.3.1 $\hat{R}$ (Gelman-Rubin statistic) Values, Convergence and Interpretation

The first thing to check while running an MCMC model is to check the convergence and chain mixing. In the case of stochastic volatility, all R-hat values are around 1.00. While the majority of it is already 1.00, there are a couple of parameters, such as volatility of volatility, σ, which had an R-hat value of 1.06. That being said, all values are smaller than 1.1, indicating proper chain mixing and convergence.

As shown in 6, the HMC structure used in this model has 1000 warm-up iterations and 2000 iterations, accompanied with 4 chains with a maximum tree depth of 10. Divergence diagnostics show that there is 0 divergent transition, showing that the HMC process was successful.

4.3.2 ESS (Effective Sample Size) Analysis

Examples of ESS Bulk and ESS Tail values are shown in the Table 3. There, it can be seen that the majority of the ESS Bulk values are extremely high, ranging between 3000 and 9500. These values are beyond acceptable and they signal that the central parts of the posterior are sampled in a very good manner. The only exceptions to this are parameters $lp\_\_$ and σ, which have ESS Bulk values equal to 65.61 and 66.99. These are not especially favourable numbers and it should be noted that these values suggest that in terms of the log posterior density and volatility of volatility, the core regions of the posterior are not explored as well as the other parameters. In ESS Tail, which is related to extremities of the posterior distribution, the situation is much better. All values are well above 200, showing that the model explored the tails of the posterior distribution well in general. Parameters $lp\_\_$ and σ have smaller ESS Tail values, similar to the situation at ESS Bulk, but this time even for those parameters the values are higher, meaning that there is less issue in capturing the tail behaviour.

4.3.3 Posterior Predictive Checking for Stochastic Volatility

Figure 4 in the appendix is about the posterior predictive check for the stochastic volatility model. Here, we plotted the density of draws of daily log returns from the posterior distribution against the data. On the figure, the dark blue line represents the data while the light blue field represents the samples from the posterior distribution. As you can see, the distribution seems to explore the data very closely. While this might signify that the set of priors and stochastic volatility model structure used in this project can be suitable for the analysis goals, it can also mean that there is overfitting.

In fact, the results of Leave-One-Out Cross-Validation at Table 4, show that there is overfitting happening and despite the graph creating the illusion that this model and set of priors are suitable for the project, it in fact, is otherwise. Very low Expected Log Predictive Density and very high Effective Number of Parameters indicates that stochastic volatility model is overfitting the data, thus it has bad posterior predictive performance.

5 Prior Sensitivity Analysis

This section explores the impact of alternative priors on ARCH, GARCH, and stochastic volatility models to evaluate the robustness of Bayesian inference. The original priors were discussed in the previous section Here, we focus on testing alternative priors, including deliberately non-sensical ("dummy") priors, to highlight their effects on parameter estimates, model stability, and posterior predictions. For the stochastic volatility model, only dummy priors were tested to illustrate the counterintuitive result of visually reasonable predictive checks despite extreme and implausible prior choices, and it needs further investigation.

5.1 Alternative 1: Priors Based on Financial Theory

ARCH. The priors for the ARCH(1) model are grounded in financial theory and empirical observations. The mean return parameter μ is assigned a normal prior $\mu \sim \mathcal{N}(0, 0.2)$, reflecting moderate uncertainty around a small mean. The base volatility $\alpha_0 \sim \text{LogNormal}(-2, 0.5)$ ensures positivity and favors small values, while the ARCH parameter $\alpha_1 \sim \text{Beta}(2, 8)$ promotes stability with values near zero but allows larger values. Sensitivity analysis tested alternative priors, including a Student's t-distribution for μ, varied scales for α_0, and different Beta distributions for α_1. Results showed minimal changes in estimates of key quantities, confirming robustness to prior specifications. Together, these priors ensure a realistic and robust model.

GARCH. The priors for the GARCH(1,1) model extend these principles to accommodate complex volatility dynamics. The mean return $\mu \sim$ Student_t$(3,0,0.5)$ captures heavy-tailed behavior in financial returns. The base volatility $\alpha_0 \sim$ Normal$(0.2,0.1)$ reflects typical volatility levels (e.g., 10%–30%), and the GARCH parameter $\alpha_1 \sim$ Uniform$(0,1)$ ensures stationarity and flexibility. Sensitivity analysis explored alternative priors, including varying degrees of freedom for μ, scales for α_0, and ranges for α_1. Results indicated consistent estimates of key quantities, highlighting robustness to prior choices. These priors reflect empirical observations and theoretical constraints, ensuring the model captures the stylized facts of financial data.

5.2 Alternative 2: Priors Informed by Empirical Studies

ARCH. The priors for the ARCH(1) model are informed by empirical studies, aligning with observed behaviors in financial time series. The mean return μ follows a heavy-tailed prior $\mu \sim$ Student-t$(3,0,0.5)$, providing robustness to outliers and reflecting typical near-zero mean returns. The base volatility $\alpha_0 \sim \mathcal{N}(0.2,0.1)$ captures small but positive volatility with moderate uncertainty. The ARCH parameter $\alpha_1 \sim$ Uniform$(0,1)$ ensures flexibility and stationarity. These priors balance theoretical considerations and empirical evidence, allowing data to guide posterior estimates effectively.

GARCH. The priors for the GARCH(1,1) model are informed by empirical studies to handle more complex volatility dynamics. The mean return μ follows Student-t$(3,0,0.5)$, capturing heavy tails and reflecting typical near-zero returns. The base volatility $\alpha_0 \sim \mathcal{N}(0.2,0.1)$ represents small but positive volatility with moderate uncertainty. The ARCH parameter $\alpha_1 \sim$ Uniform$(0,1)$ and the GARCH parameter $\beta_1 \sim$ Uniform$(0,1)$ ensure flexibility while maintaining stationarity. These priors balance robustness and flexibility, aligning with observed financial data.

5.3 Dummy Priors

ARCH. The dummy priors were deliberately chosen to demonstrate the impact of inappropriate prior distributions in Bayesian modeling. These priors, such as $\mu \sim \mathcal{N}(100,1000)$, $\alpha_0 \sim \mathcal{N}(0,10)$, and $\alpha_1 \sim$ Uniform$(-1,2)$, are nonsensical for financial time series and lead to theoretical violations, instability, and convergence issues. Sensitivity analysis was conducted by replacing α_0 with a lognormal prior and restricting α_1 to a uniform prior within $[0,1]$. These adjustments restored theoretical validity, improved stability, and yielded more reliable inference. This analysis highlights how inappropriate priors can lead to unreliable results, underscoring the critical importance of thoughtful prior selection. See the graph in Appendix A for more details.

GARCH. The dummy priors for GARCH(1,1) model uses exaggerated priors, such as $\mu \sim$ Uniform$(-1000,1000)$, $\alpha_0 \sim \mathcal{N}(100,50)$, $\alpha_1 \sim$ Beta$(50,0.5)$, and $\beta_1 \sim$ LogNormal$(-5,0.5)$, to demonstrate the effects of extreme prior choices. Despite visually reasonable posterior predictive checks (see Figure 6 in the Appendix), sensitivity analysis revealed that narrowing priors to realistic ranges, such as $\mu \sim \mathcal{N}(0,1)$, $\alpha_0 \sim \mathcal{N}(0.2, 0.1)$, and uniform priors for α_1 and β_1 within $[0,1]$, improved interpretability and avoided implausible values. Power-scaling checks indicated that α_0 had the largest impact on posterior estimates, confirming its critical role in shaping volatility. This analysis underscores that visually reasonable predictions can mask unreliable parameter estimates caused by inappropriate priors, emphasizing the importance of realistic and informed prior selection.

Stochastic Volatility Model. The dummy priors for stochastic volatility uses exaggerated priors, including $\mu \sim \mathrm{Uniform}(-1000, 1000)$, $\phi \sim \mathrm{Uniform}(-10, 10)$, and $\sigma_{\mathrm{vol}} \sim \mathrm{Gamma}(0.01, 0.01)$, to explore their impact on Bayesian inference. Despite these priors, the posterior predictive check (see Figure 7 in the Appendix) appears reasonable. Sensitivity analysis tested more realistic priors: $\mu \sim \mathcal{N}(0,1)$ improved interpretability without affecting predictions; $\phi \sim \mathrm{Uniform}(-1,1)$, consistent with AR(1) bounds, eliminated instability; and $\sigma_{\mathrm{vol}} \sim \mathcal{N}(0.5, 0.2)$ enhanced interpretability without predictive changes. Power-scaling checks showed ϕ has the most impact on volatility dynamics, while μ and σ_{vol} influence parameter interpretability. While data dominance allowed reasonable predictions under extreme priors, such priors caused unreliable parameter estimates and theoretical issues. Adjusting priors restored stability and validity, highlighting the necessity of informed prior selection.

6 Discussions

6.1 Model Comparison

One of the common ways to compare different models is by using Leave-One-Out Cross-Validation. In this approach, a part of the data is left out intentionally to see how the models perform with incomplete data. Comparing the LOO-CV results is a valid way to compare models' performances.

Table 4: LOO-CV Results for ARCH Model

Parameter	**Estimate**	**Standard Error**
Expected Log Predictive Density (ARCH)	8056.4	61.6
Effective Number of Parameters (ARCH)	6.9	1.1
Leave-One-Out Information Criterion (ARCH)	-16112.7	123.2
Expected Log Predictive Density (GARCH)	8359.6	56.6
Effective Number of Parameters (GARCH)	3.7	0.7
Leave-One-Out Information Criterion (GARCH)	-16719.1	113.2
Expected Log Predictive Density (SV)	-419549.5	11467.2
Effective Number of Parameters (SV)	425998.9	11469.8
Leave-One-Out Information Criterion (SV)	839098.9	22934.3

In LOO-CV, higher Expected Log Predictive Density indicates a better fit to the data thus it is a desired quantity while Effective Numbers of Parameters are about the model complexity thus higher value might mean overfitting. Leave-One-Out Information Criterion, on the other hand, is an estimation of the Expected Log Predictive Density for the new data where lower values indicate a better predictive capacity.

Looking at Table 4, the best-performing model in the metrics of LOO-CV is GARCH. That is because it has the highest Expected Log Predictive Density as well as the lowest Leave-One-Out Information Criterion, implying that it explored the data best and has the highest predictive capacity. More impressively, this happened while it has the lowest Effective Number of Parameters, meaning that it is the least complex model in terms of number of parameters used.

On the other hand, these values indicate something very interesting. Stochastic Volatility, which has the alternative approach to the problem, is performing the worst. It is interesting because it is generally considered to be a superior model to ARCH and GARCH. However, LOO-CV results show that it doesn't have a best fit in the case of some data left out. This creates a thought of overfitting, which is verified by the Effective Number of Parameters. Stochastic volatility has the by far highest value for this measurement, implying that the model is overly complex. This is an undesired situation signalling the unsuitability of the stochastic volatility model to the problem of this project.

6.2 Problems and Possible Improvements

As discussed in the Model Comparison section, the GARCH model is the best suited for the given problem and dataset. This is especially critical to realise that more complicated approaches to a probabilistic problem are not necessarily the best ones.

In this case, stochastic volatility represented the most complex model selected, however, it didn't perform the best, in fact, it performed the worst. It overfitted the data and when some of the data was unavailable to it, the performance got suddenly worse as LOO-CV results showed clearly. This possibly is because of its over-reliance on the data as well as its nature. Unlike ARCH/GARCH models' deterministic approach to volatility, the stochastic volatility model assumes volatility is a latent parameter meaning that it is hidden in the process and stochastically decided. There is also some relation to the nature of the data. S&P 500 is rather a stable index, since the volume of the index is very high thus it is not prone to sudden changes. Therefore, it can actually be very well analysed by using simpler models as the relation is simple and more complex models might over estimate the relationships in the data.

In our project, our biggest problem was the unsuitability of stochastic volatility in the context of this project. Implementing the model and running it over the dataset showed us that it is complicated beyond the scope of this project and indeed, not needed.

That being said, there is always room for improvement. Stochastic volatility uses a normal distribution to estimate daily log-returns, it is possible to use another distribution instead. For example, student's t-distribution could be an option here as it is more prone to overfitting. That possibly would make the stochastic volatility model more usable in the case of this project.

7 Conclusion

The overall workflow of this project work showed that in a big index like S&P 500, there is no need for complicated models and when the data can be modelled by simpler models using more complicated ones does more harm than good.

Although it is more complicated than ARCH/GARCH, stochastic volatility in this project is overfitting the data and, thus is unsuitable. It's assumption of latent stochastic relationship in volatility doesn't perform well while a more simple approach of GARCH assuming a deterministic relation regarding the volatility and daily log-returns performs very well. In this project, GARCH model performed the best in all metrics, did not overfit the data and at the same time, kept low computational complexity thus a lower need for computing power.

The models and priors should be selected considering the nature of the data as the data is the main ground for applying those techniques and distributions. Even though Bayesian inference runs without any issue in a technical sense, the accuracy and precision of results are another aspect and should be carefully examined while taking any decision.

8 Self-Learning

Working as a group is not an easy task. There is a need for high coordination but at the same time, good division of tasks. In our project, we missed especially the latter part, creating imbalances in the workloads of group members. This further leads to differences in their understanding of the project work and their engagement with it. A careful planning stage is critical for a smooth and successful project and we learned the importance of it first-hand while working on the project.

References

[1] B. Z. Heng Sun, "Volatility clustering in the S&P 500 market index: An arch/garch approach," M.S. thesis, California State Polytechnic University, Pomona, Pomona, 2017.

[2] J. Shu and J. E. Zhang, "Pricing S&P 500 index options under stochastic volatility with the indirect inference method," *Journal of Derivatives Accounting*, vol. 01, no. 02, pp. 171–186, Sep. 2004, ISSN: 0219-8681. DOI: 10.1142/s021986810400021x. [Online]. Available: http://dx.doi.org/10.1142/S021986810400021X.

[3] X. Chen, "Comparing various garch-type models in the estimation and forecasts of volatility of S&P 500 returns during global finance crisis of 2008 and covid-19 financial crisis," *SHS Web of Conferences*, vol. 169, M. Yin, P. Wang, and T. Kuang, Eds., p. 01 077, 2023, ISSN: 2261-2424. DOI: 10.1051/shsconf/202316901077. [Online]. Available: https://www.shs-conferences.org/10.1051/shsconf/202316901077.

[4] A. Vehtari and B. Ward, *Time-series models - stochastic volatility*, 27AD. [Online]. Available: https://mc-stan.org/docs/stan-users-guide/time-series.html#stochastic-volatility-models.

A Appendix A: Additional Graphs

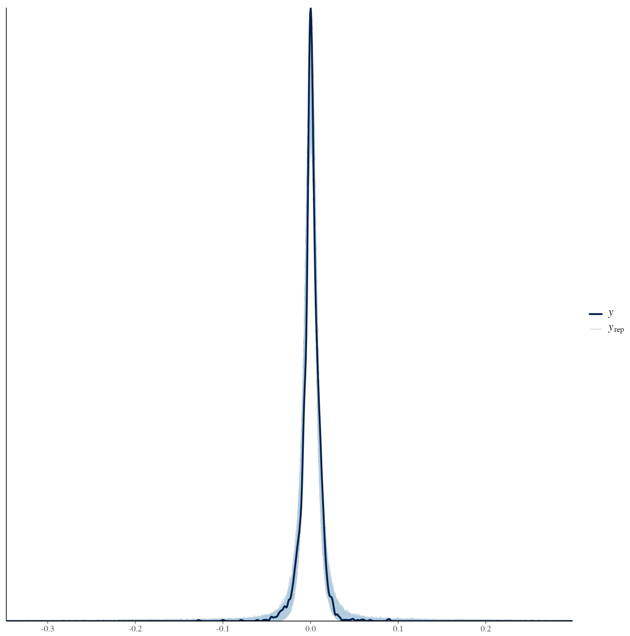


Figure 4: Posterior Predictive Check for Stochastic Volatility

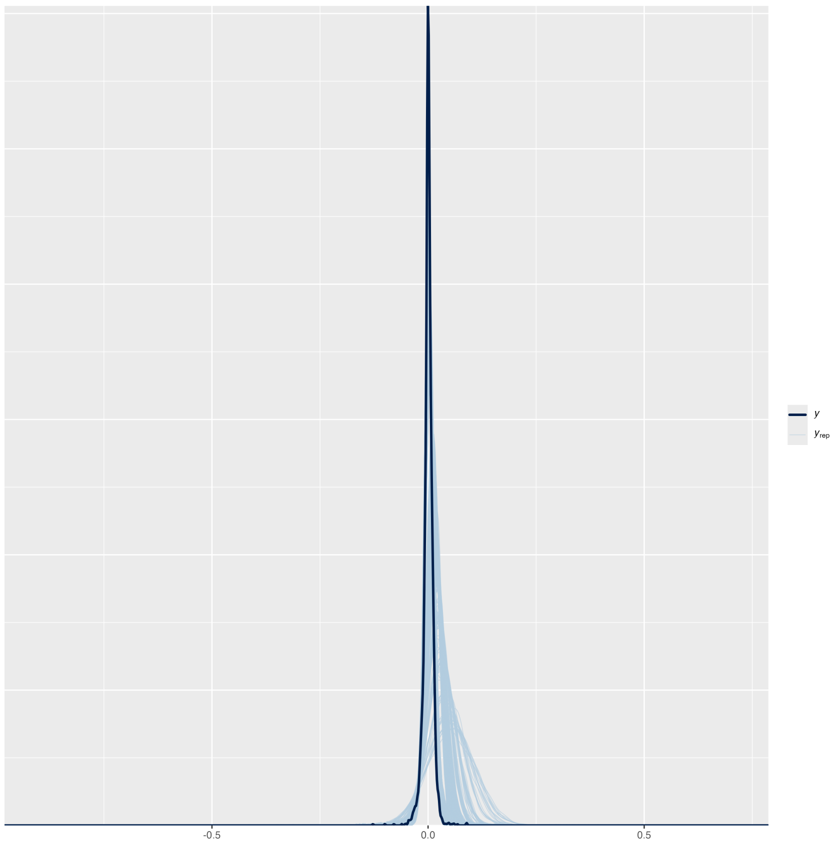


Figure 5: Posterior Predictive Check with Dummy Priors for ARCH model. The plot shows the density overlay of observed data y and replicated data y_{rep}. Despite the nonsensical priors, the fit appears skewed but somewhat reasonable, highlighting the dominance of the likelihood.

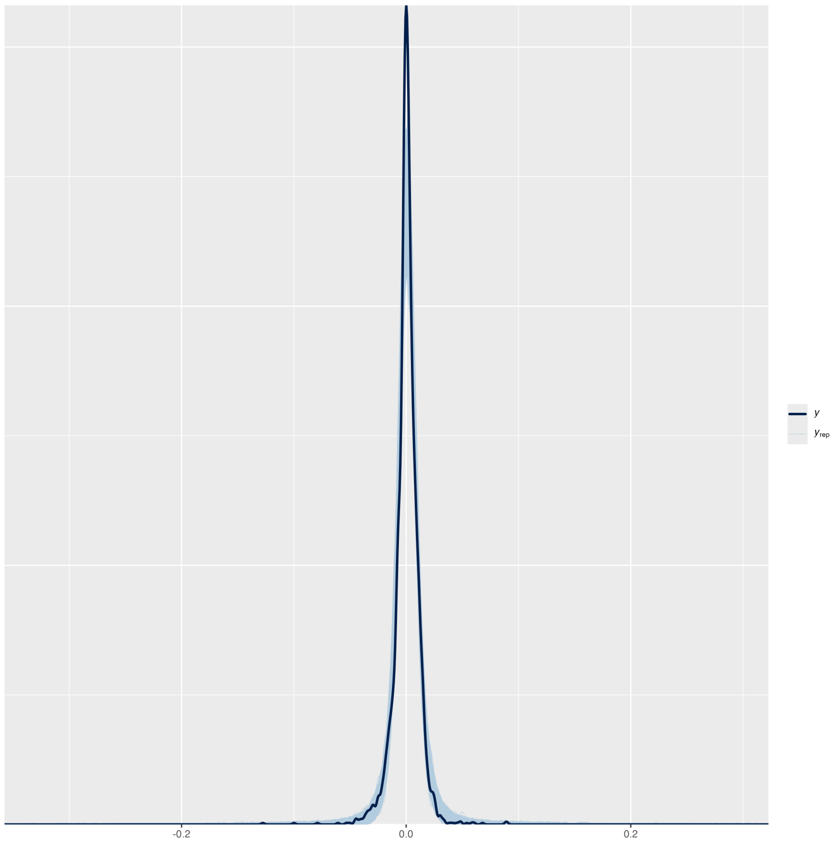


Figure 6: Posterior predictive check for the dummy GARCH(1,1) model with exaggerated priors.

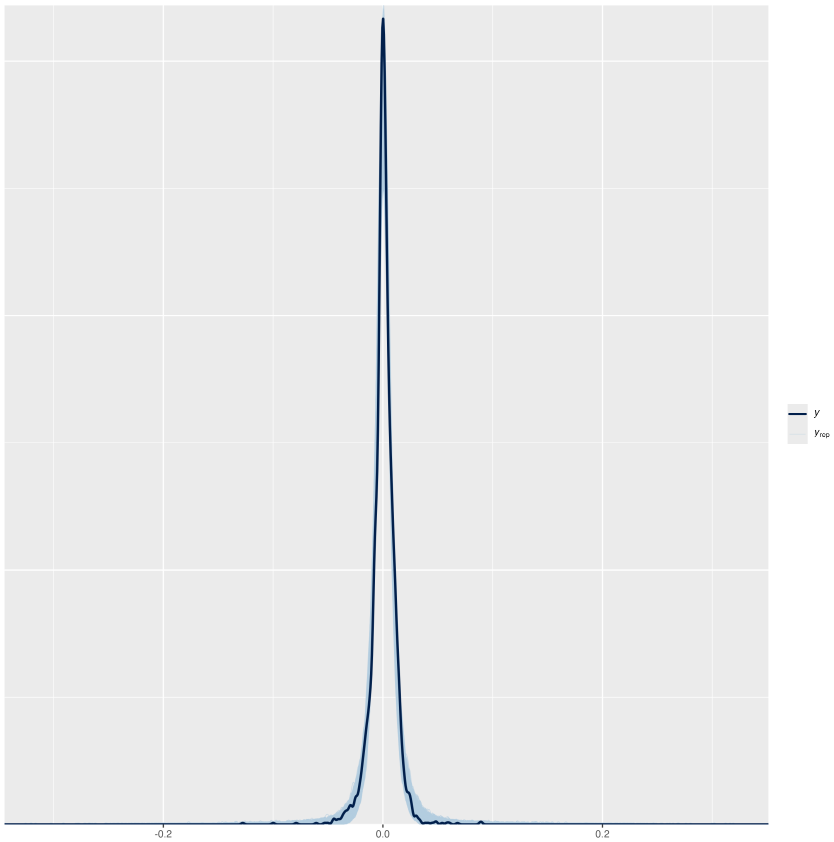


Figure 7: Posterior predictive check for the stochastic volatility model with exaggerated priors. Despite the extreme priors, the predictive performance appears reasonable.