Bayesian Time Series Analysis of US Unemployment Rates

STAT 447C Final Project - Project Report | Rohan Joseph (67089839)

Due Date: April 19th 2024

Project GitHub URL: https://github.com/RohanUBC/STAT-447C-Final-Project

Introduction

The forecasting of US unemployment rates presents a complex and significant challenge for US policy makers. Accurate predictions of these rates enables the US government and businesses to make informed economic decisions, adapt to economic shifts, and implement strategies to foster stability and growth in the US economy. Furthermore, the unpredictability of labor markets, influenced by factors ranging from global economic conditions to domestic policy changes, necessitates the use of sophisticated methods not only provide a framework for forecasting but also allow for the quantification of uncertainty in those forecasts. Understanding and predicting unemployment rates is thus not only central to economic analysis but also to the well-being of the US economy.

The focus of this project is to model and forecast unemployment rates in the United States (US), utilizing a real-world dataset that has documented the monthly unemployment rate (from 01/1948 to 02/2024) in the US. The dataset that is being used is from the US Federal Reserve on monthly US unemployment rates.

URL: https://fred.stlouisfed.org/series/UNRATE

We will adopt a Bayesian time series analysis approach, utilizing a Bayesian Structural Time Series (BSTS) model to capture the dynamics of the US labor market based on the provided data. This involves uncovering and modeling historical trends and seasonal patterns within the data, providing probabilistic forecasts of future unemployment rates. We opt for BSTS due to its flexibility in modeling complex time series data with trend and seasonality components. BSTS allows for the estimation of latent states, such as trend and seasonal effects, and provides a framework for forecasting future states while quantifying uncertainty. Please see Appendix C on a general overview on the BSTS model.

This project faces challenges in applying Bayesian time series analysis with BSTS to economic data, particularly in modeling the seasonal patterns and long-term trends inherent in unemployment rates. The primary hurdle lies in selecting and fine-tuning parameters and models that can accurately reflect the dynamics of the US labor market. This includes determining appropriate prior distributions, specifying the structural components of the model, and validating the model's performance through diagnostics and posterior predictive checks. Additionally, handling the irregularity in seasonal variation presents a challenge, requiring careful consideration in the modeling approach to ensure accurate representation of the data.

In addition, we aim to also include a calibration of the uncertainty of the forecast, demanding a sophisticated approach in how the model is constructed and validated. This is done to ensure that the forecasts are not only precise, but are also accompanied by reliable uncertainty measures. Such measures are pivotal for economic planning and policy-making, providing a foundation for risk-aware decision-making. By accurately quantifying forecast uncertainty, this approach aims to assess potential risks effectively, thereby facilitating more informed and prudent economic decision making.

Literature Review

- Summarize key findings from existing literature on Bayesian forecasting and unemployment rate analysis.
- Highlight gaps your project aims to fill or how it builds upon previous work.
- Discussion of uncertainty quantification in economic forecasting.
- Calibration of probabilistic forecasts in existing literature.

[To be added]

Data Analysis

Splitting the Dataset into Training and Testing Sets

We will be using all the data up until 12/2020 as the training set, and have the remaining data (from 01/2021 to 02/2024) as the testing set (see Appendix B - Section 1). With 756 observations for training, the training set has a significant amount of historical data, spanning over 60 years, to learn from. This period should include multiple economic cycles, which can help in capturing the underlying economic dynamics. The test set has 158 observations, allowing the model to be validated against more recent data while also allowing to test the model's ability to predict under potentially different economic conditions (for example the COVID-19 pandemic).

Setting up the BSTS Model

As established in Appendix C, the general BSTS model consists of two sets of equations: the observational (response variable) and the state-space equations (how the parameters change over time). In a BSTS model, we would have to decide on whether an additive or multiplicative model was appropriate as the observational model, and also what specific components are relevant as part of the state-space of the BSTS model (trend, seasonality, etc.). In order to do this, we need to analyse the decomposition and autocorrelation (ACF) plots of a time series of the training set (see Appendix B). The decomposition and ACF plots of additive model are displayed in Figures 3 and 4, respectively, in Appendix B. Similarly, the decomposition and ACF plots of the multiplicative model are displayed in Figures 5 and 6, respectively, in Appendix B.

Key Definitions:

Decomposition of a Time Series: This involves separating a time series into its constituent components, such as trend, seasonality, and noise, to better understand its underlying patterns and behaviors.

Autocorrelation: This is the correlation between a time series and its own lagged values, revealing how past observations relate to present or future values within the same series.

Autocorrelation Function (ACF) Plot: This is a visual representation of the correlation between a time series and its lagged values at different lags, which helps to identify patterns of autocorrelation and inform the selection of appropriate models for time series analysis.

Analysis of the Decomposition and ACF Plots

Based on the additive and multiplicative decomposition plots and their respective autocorrelation (ACF) plots, we can establish the following:

ACF of Residuals: Both the additive and multiplicative decompositions shows significant autocorrelation at lag 1 in the residuals, suggesting that neither decomposition fully captures the autocorrelative structure of the series.

Decomposition of the Observed Time Series: The additive and multiplicative decomposition residuals do not show any clear pattern or trends, which would indicate that both models may be viable.

Seasonal Patterns: The seasonality in the additive plot is consistent and doesn't appear to change with the level of the series, indicating an additive nature. The multiplicative plot, while proportional, doesn't show a clear advantage over the additive model with regards to the residuals.

While neither model is perfectly viable, an additive model may be more useful because the seasonality doesn't seem to be increasing or decreasing in amplitude, suggesting a constant seasonality over time, and also because the residuals from the additive model do not show patterns that would indicate multiplicative effects.

Trend Component

From the decomposition plots we can see that there is no long-term trend in the unemployment rates but that there are instances of increases and decreases in the unemployment rate, potentially influenced by other economic factors. For the BSTS model, the trend component would need to reflect long-term economic changes, without being affected by short-term changes. So, modelling the trend as a normally distributed value around its previous state, with a controlled variance, would allows it to evolve steadily.

Seasonal Component:

From the decomposition plots we can see a consistent pattern that repeats with a fixed frequency, which is indicates that there is the presence of seasonality in the data. For the BSTS model the seasonal component would need to account for predictable, repetitive patterns in unemployment rates. By constraining the seasonal effects to sum to zero over a year, the model would ensure that it captures the seasonality without influencing the overall trend of the unemployment rates. The sum would be normally distributed with a variance parameter allowing for some annual variation.

Autoregressive Component

Considering the presence of autocorrelation at lag 1 in the residuals of both models, it would be crucial to consider a model accounting for this. In a BSTS model, the autocorrelation can be modeled by adding an autoregressive (AR) component as part of the state space model, specifically an AR(1) process. In general, the autoregressive component X_t evolves over time as $X_t = \sum_{i=1}^{p} \phi_i \cdot X_{t-i} + \epsilon_t$, where ϕ_i are the autoregressive components for lags $i = 1, 2, ..., p$, p is the order of the autoregressive process, and ϵ_t is the white noise. However, since we are considering an AR(1) model (so $p = 1$), this simplifies to $X_t = \phi \cdot X_{t-1} + \epsilon_t$.

BSTS Model for US Unemployment Rates

BSTS Model Specification

Observation Model: The observed unemployment rate at time t, y_t, is modeled as:

$$y_t \sim \text{Normal}(m_t + s_t + X_t + Z_t, \sigma_y^2), \qquad \text{where } y_1 \sim \text{Normal}(m_1 + s_1, \sigma_y^2)$$

where m_t represents the trend component, s_t denotes the seasonal component, X_t denotes the autoregressive process, Z_t is the white noise, and σ_y^2 is the observation variance.

Local Level (Trend) Component: The trend component m_t evolves over time as:

$$m_t \sim \text{Normal}(m_{t-1}, \sigma_m^2), \qquad \text{where } m_1 \sim \text{Normal}(0, \sigma_m^2)$$

where m_{t-1} is the trend component at the previous time point, σ_m^2 is the variance of the trend component, and m_1 is the initial trend component.

Seasonal Component: The seasonal component s_t evolves over time as:

$$s_t \sim \text{Normal}\left(-\sum_{i=1}^{S-1} s_{t-i}, \sigma_s^2\right), \qquad \text{where } s_1, s_2, ..., s_{12} \sim \text{Normal}(0, \sigma_s^2)$$

where S is the seasonal period (in this case $S = 12$) for monthly, s_{t-i} is the seasonal component as the previous time point, σ_s^2 is the variance of the seasonal component, and $s_1, s_2, ..., s_{12}$ are the initial seasonal component values (associated with every month of a year).

Autoregressive Process:

The autoregressive process X_t evolves over time as:

$$X_t = \phi \cdot y_{t-1} + \epsilon_t$$

where is ϕ_i is the autoregressive component at lags 1 (since we assume an AR(1) process) is the unemployment rate at the previous time point., and ϵ_t is the white noise (this is implicitly modeled as part of the observation error).

Hyperparameters and Hyperpriors

Observation Noise Variance: $\sigma_y^2 \sim \text{Inverse-Gamma}(\alpha_y, \beta_y)$, where $\alpha_y = 2$ and $\beta_y = 0.04663646$.

Trend Variance: $\sigma_m^2 \sim \text{Inverse-Gamma}(\alpha_m, \beta_m)$, where $\alpha_m = 2$ and $\beta_m = 2.348358$.

Seasonal Variance: $\sigma_s^2 \sim \text{Inverse-Gamma}(\alpha_s, \beta_s)$, where $\alpha_m = 2$ and $\beta_m = 1.528651$.

Autoregressive Component: $\phi \sim \text{Normal}(0, 1)$

Stan Code - Please go to Appendix E to see the implementation of the above model in Stan. I, however, did not use this implementation as it was too computationally intensive, especially with regards to predictive testing.

`bsts` Code - Please go to Appendix F to see the implementation of the above model using the `bsts` library, which is an R library built for working with BSTS models.

Motivations of Priors

To justify the hyperpriors and set the hyperparameters, we analysed the historical unemployment rates to inform the choice of prior distributions for the observation noise variance, trend variance, and seasonal variance.

The α (shape) values are set to $\alpha = 2$ to ensure that the prior is relatively uninformative (weak prior belief), thus allowing the data to play a significant role in the posterior distribution.

The β (scale) values are informed the respective variance estimates, and for each component β = variance estimate $\times (\alpha - 1)$, centering the prior mean around the estimate while also allowing for some uncertainty.

Please see Appendix D to see the R code used to set the α and β values.

Observation Noise Variance σ_y^2: An Inverse-Gamma distribution is used to ensure positivity, with the α_y and β_y values informed by historical data to reflect the inherent unpredictability in unemployment rates.

Trend Variance σ_m^2: An Inverse-Gamma distribution is used to capture the variability in the long-term trend, where α_m and β_m adjusts for the observed trend stability.

Seasonal Variance σ_s^2: An Inverse-Gamma distribution is used to ensure positive, data-informed seasonal variations in unemployment rates, tailored by its α_s and β_s values.

Initial Trend m_1 *and Seasonal Components* $s_1, s_2, ..., s_{12}$: Normal distributions with mean 0 and the trend/seasonal variance, respectively, are used for their flexibility in the initial values while constraining them within reasonable bounds set by their respective trend and seasonal variances.

Autoregressive Component ϕ: A Normal distribution with mean 0 and variance 1 indicates no initial bias towards trend direction or strength, allowing data to define unemployment rate persistence.

[To be added]

- *Model:* A Bayesian model is precisely described (e.g., using the … ~ … notation).
  - Description of BSTS for modeling unemployment rates.
- Implementation code in the appendix (e.g., using Stan)
  - R code for implementing the BSTS using Bayesian methods.
- *Motivation of prior choice:* If appropriate, several choices are compared or sensitivity analysis is performed.
- *Critical evaluation of the posterior approximation:* An appropriate combination of diagnostics, synthetic datasets and other validation strategies.
  - Techniques used for uncertainty quantification and calibration of forecasts.
  - Model diagnostics and validation strategies.
- *Methodological/Theoretical aspect:*
  - Assessment of the approach's robustness and creativity.
  - Discussion on the choice of BSTS and Bayesian calibration methods.
- *Calibration of uncertainty:*
  - Explanation of calibration tests for predictive intervals.
  - Presentation of calibration test results.
  - Analysis of how well the model's uncertainty measures align with observed data.

Discussion

- *Project Theme:* Assess the soundness and creativity of the approach within the context of Bayesian forecasting and time series analysis. Discuss the model's ability to handle the intricacies of unemployment data and its forecasting accuracy.
- *Summarizing results:* Summarize key findings and their implications for understanding and forecasting US unemployment rates.
  - Interpretation of the forecasting results.
- Implications of calibrated uncertainty for economic decision-making.
- Discuss the limitations of your study, such as data constraints, model assumptions, or potential biases.
- Suggest areas for future research or methodological improvements.

[To be added]

Conclusion

- Recap of the project's contributions to Bayesian forecasting and calibration.
- Reflect on the value of Bayesian methods in economic time series forecasting.
- Final thoughts on policy implications and the value of calibrated forecasting.

[To be added]

References

[To be added]

Appendix

Appendix A - Preliminary Data Analysis

Reading in the Dataset and Data Cleaning

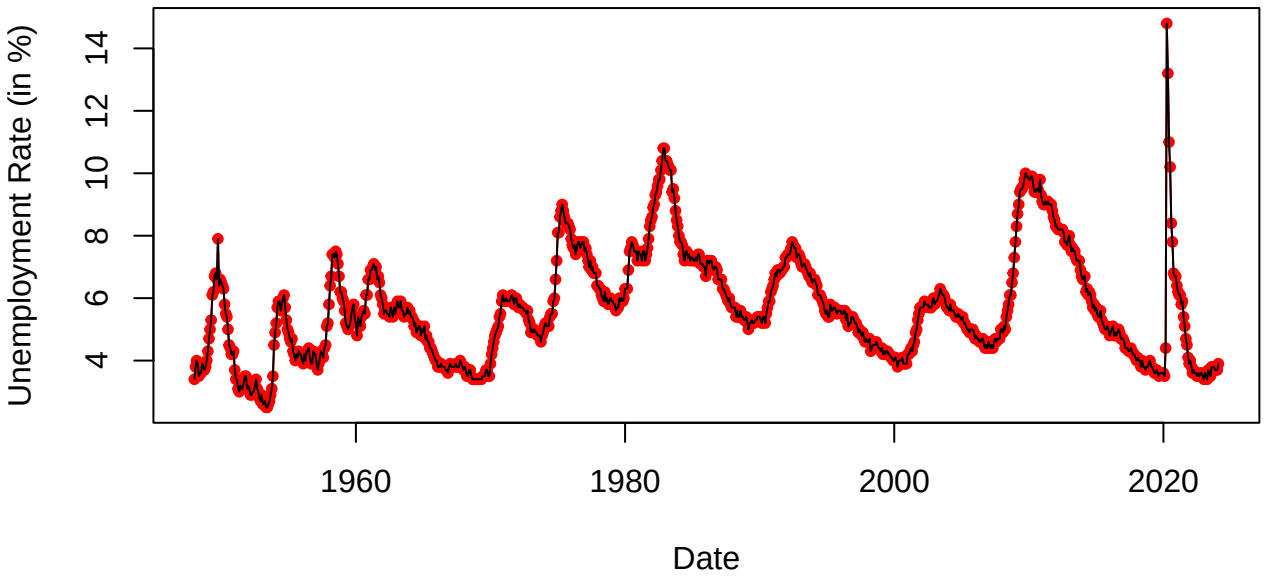
```
dat <- read_csv("UNRATE.csv")
dat$DATE <- as.Date(dat$DATE)
head(dat)
```

```
## # A tibble: 6 x 2
##   DATE       UNRATE
##   <date>      <dbl>
## 1 1948-01-01    3.4
## 2 1948-02-01    3.8
## 3 1948-03-01    4
## 4 1948-04-01    3.9
## 5 1948-05-01    3.5
## 6 1948-06-01    3.6
```

Historical US Unemployment Rates from 01/1948 to 02/2024

Figure 1: Historical US Unemployment Rate Over Time

```
plot(dat$DATE, dat$UNRATE,
     type = "o", pch = 20,
     col = "red",
     xlab = "Date",
     ylab = "Unemployment Rate (in %)")
lines(dat$DATE, dat$UNRATE, col = "black")
```



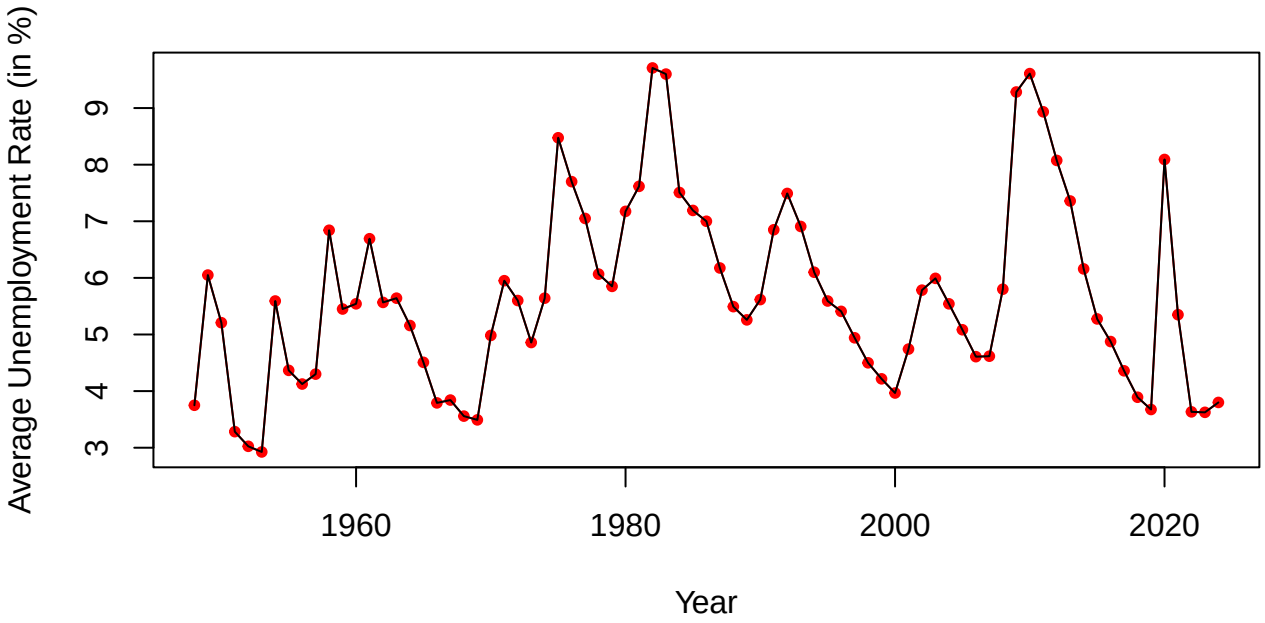
Average Yearly US Unemployment Rates from 01/1948 to 02/2024

Figure 2: Average Yearly Unemployment Rate in the US

```
avg_yearly_unrate <- dat |>
  mutate(YEAR = format(DATE, "%Y")) |>
  group_by(YEAR) |>
  summarise(Average_Unemployment_Rate = mean(UNRATE, na.rm = TRUE))

avg_yearly_unrate$YEAR <- as.numeric(as.character(avg_yearly_unrate$YEAR))

plot(avg_yearly_unrate$YEAR, avg_yearly_unrate$Average_Unemployment_Rate,
     type = "o", pch = 20,
     col = "red",
     xlab = "Year",
     ylab = "Average Unemployment Rate (in %)")
lines(avg_yearly_unrate$YEAR, avg_yearly_unrate$Average_Unemployment_Rate, col = "black")
```



Appendix B - Time Series Analysis

Section 1: Setting up the Training and Testing Sets

We will be using all the data up until 12/2020 as the training set, and have the remaining data (from 01/2021 to 02/2024) as the testing set.

```
start_date <- as.Date("1948-01-01")
train_end_date <- as.Date("2010-12-01")
test_start_date <- as.Date("2011-01-01")
end_date <- as.Date("2024-02-01")

dat_train <- subset(dat, DATE <= train_end_date)
dat_test <- subset(dat, DATE >= test_start_date)

nrow(dat_train)
```

```
## [1] 756
```

```
nrow(dat_test)
```

```
## [1] 158
```

With 756 observations for training, the training set still has a significant amount of historical data, spanning over 60 years, to learn from. This period should include multiple economic cycles, which can help in capturing the underlying economic dynamics. The test set has 158 observations, allowing your model to be validated against more recent data, covering a period of over 13 years

This split allows for the assessment of the model's performance across multiple seasons and any potential structural breaks that may have occurred at the end of 2020. This period is long enough to test the model's ability to predict under potentially different economic conditions (for example the COVID-19 pandemic).

Section 2: Creating a Time Series Object for the Training and Testing Sets

```
start_train <- c(1948, 1)
end_train <- c(2010, 12)
start_test <- c(2011, 1)
end_test <- c(2024, 2)

ts_unrate_train <-
  ts(dat_train$UNRATE, start = start_train, end = end_train, frequency = 12)

ts_unrate_test <-
  ts(dat_test$UNRATE, start = start_test, end = end_test, frequency = 12)
```

Section 3: Additive Decomposition of the Time Series

Figure 3: Additive Decomposition of the Time Series of US Unemployment Rates

```
additive_decompose_of_ts_train <- decompose(ts_unrate_train, type = "additive")
plot(additive_decompose_of_ts_train)
```

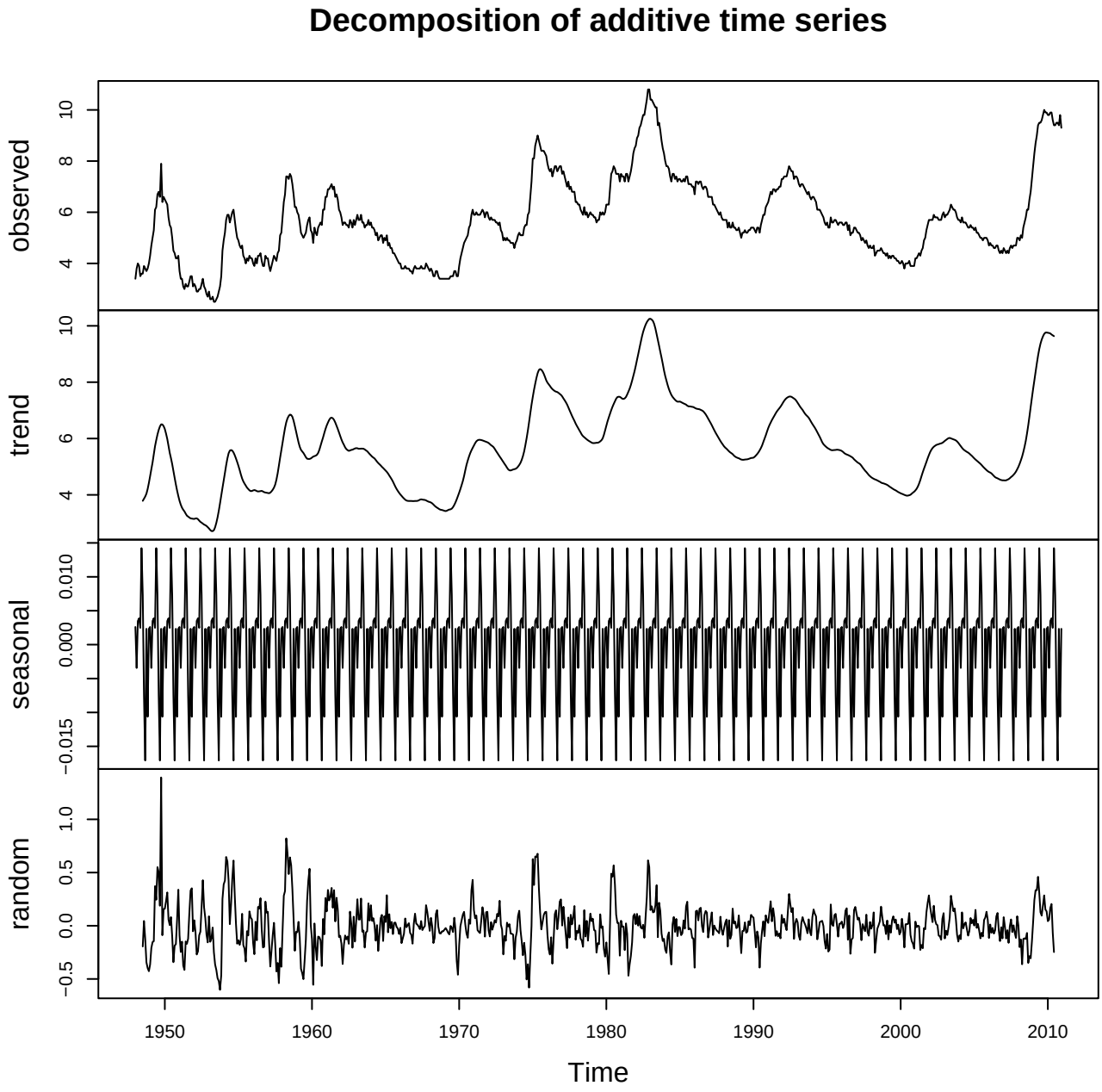
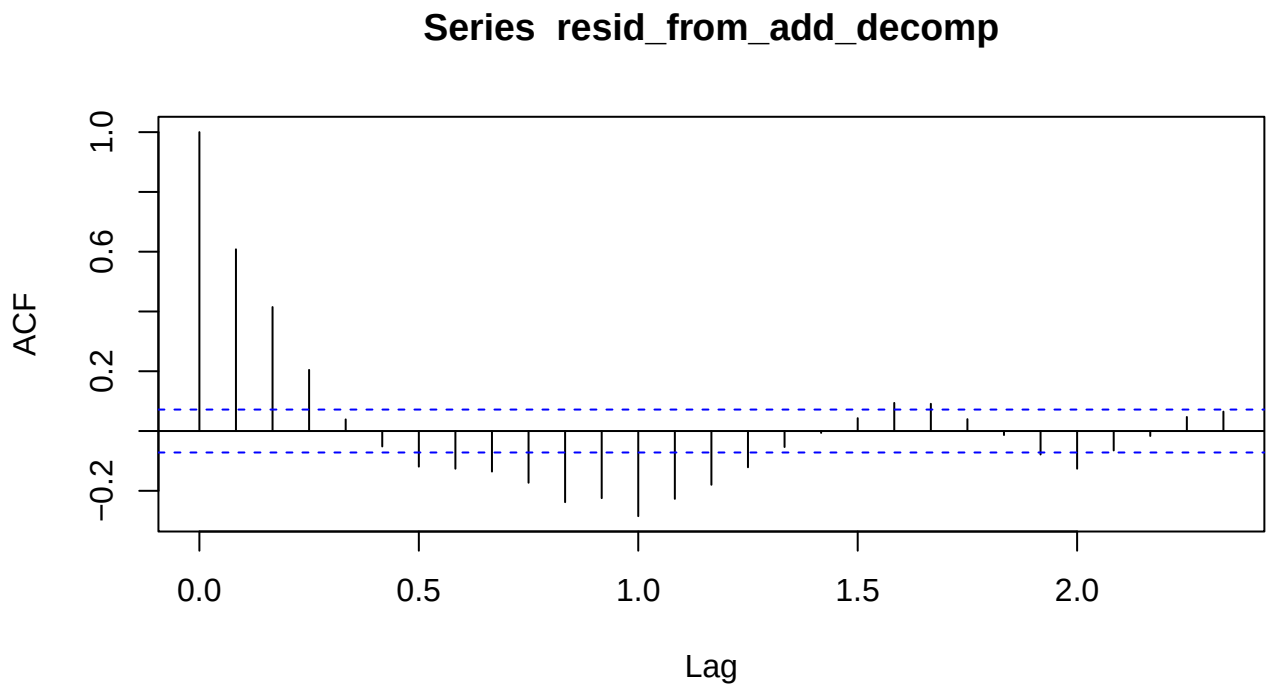


Figure 4: ACF of the residuals from the Additive Decomposition

```
resid_from_add_decomp <- additive_decompose_of_ts_train$random
resid_from_add_decomp <- na.omit(resid_from_add_decomp)
acf(resid_from_add_decomp)
```



Additive Decomposition and ACF Plots Analysis:

Additive Decomposition Plot:

- The observed time series displays cyclical fluctuations with varying amplitude.
- The trend component shows long-term movement in the data, with periods of increases and decreases.
- The seasonal component shows consistent seasonality within each year.
- The residuals (random) component does not show any obvious pattern or changing variance over time.

Additive Residuals ACF Plot:

- The ACF of residuals from the additive decomposition shows a significant spike at lag 1, indicating a potential autocorrelation at that lag.
- The spikes at all subsequent lags are within the confidence interval, suggesting that there is no significant autocorrelation beyond lag 1.

Section 4: Multiplicative Decomposition of the Time Series

Figure 5: Multiplicative Decomposition of the Time Series of US Unemployment Rates

```
multiplicative_decompose_of_ts_train <- decompose(ts_unrate_train, type = "multiplicative")
plot(multiplicative_decompose_of_ts_train)
```

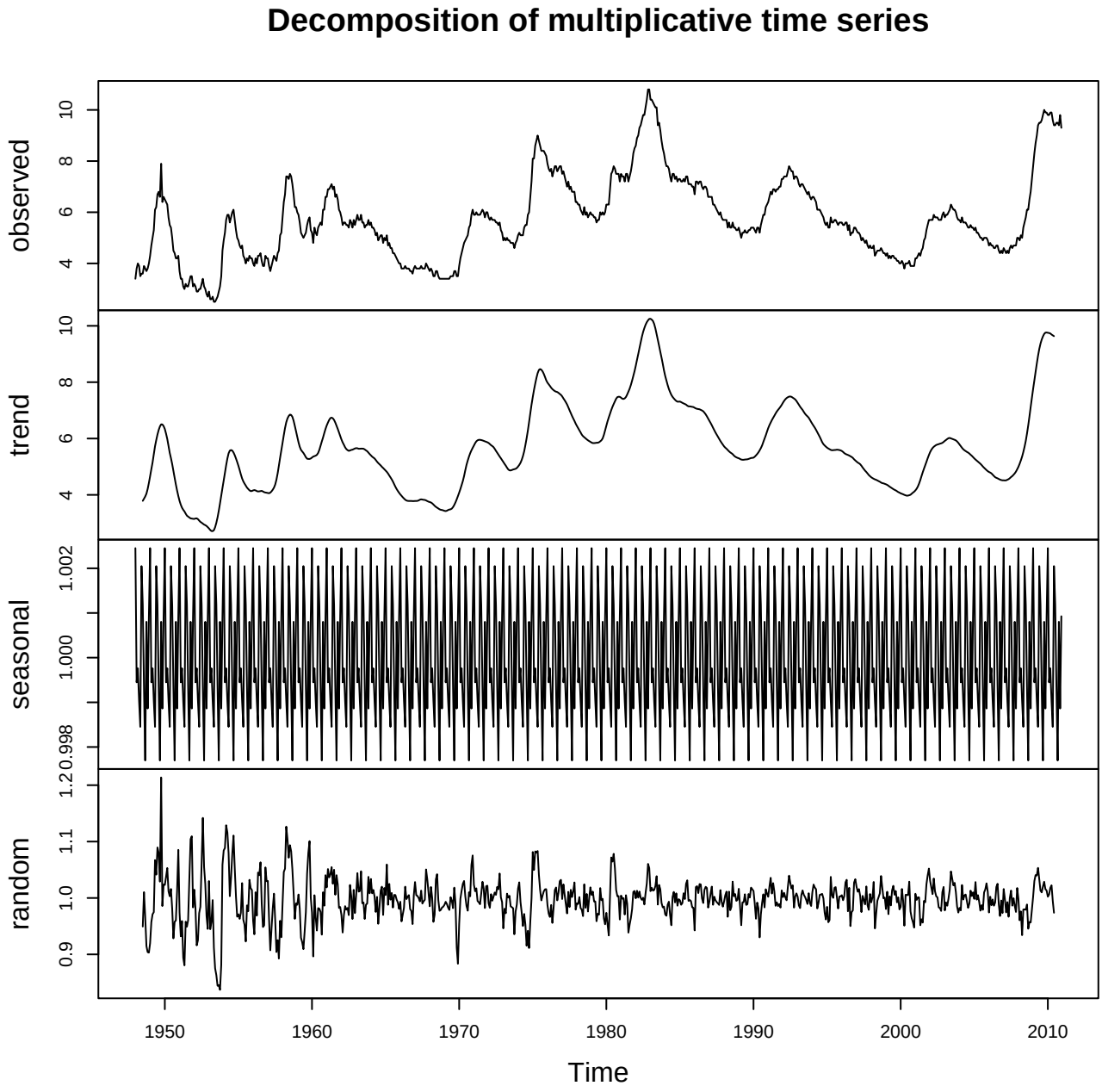
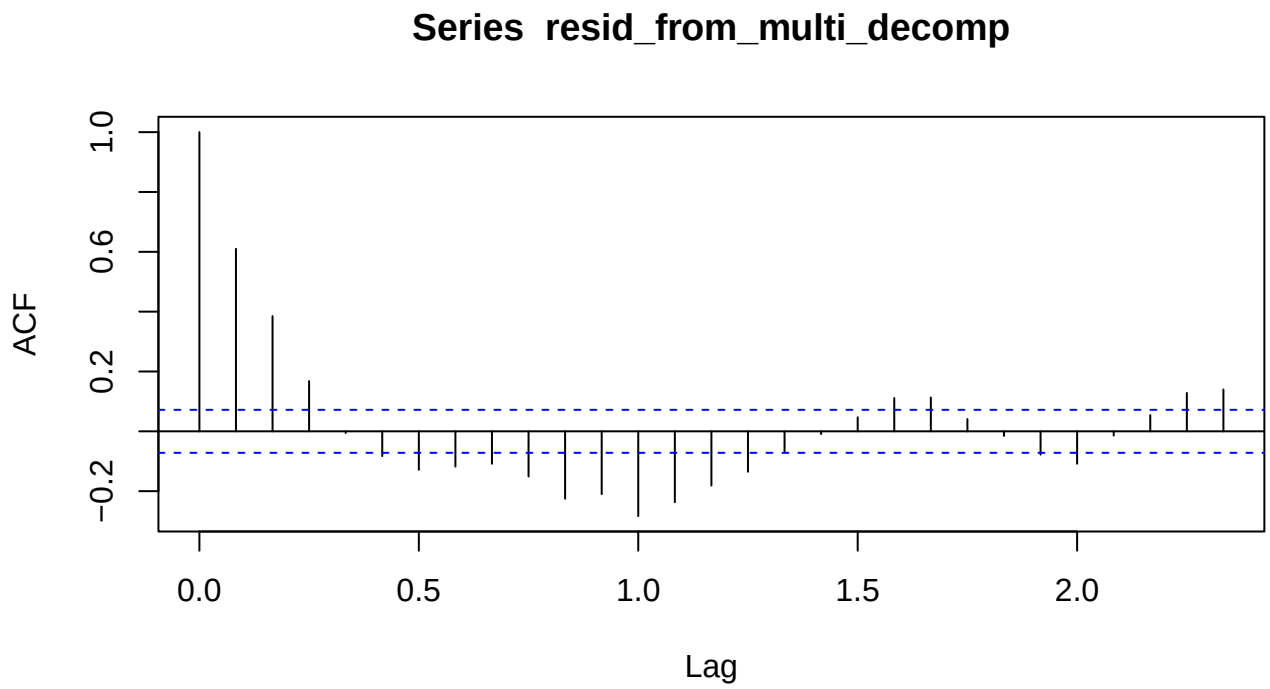


Figure 6: ACF of the residuals from the Multiplicative Decomposition

```
resid_from_multi_decomp <- multiplicative_decompose_of_ts_train$random
resid_from_multi_decomp <- na.omit(resid_from_multi_decomp)
acf(resid_from_multi_decomp)
```



Multiplicative Decomposition and ACF Plots Analysis:

Multiplicative Decomposition Plot:

- The observed time series displays cyclical fluctuations with varying amplitude.
- The trend component shows long-term movement in the data, with periods of increases and decreases.
- The seasonal component shows consistent seasonality within each year.
- The residuals (random) component does not show any obvious pattern or changing variance over time.

Multiplicative Residuals ACF Plot:

- The ACF of residuals from the additive decomposition shows a significant spike at lag 1, indicating a potential autocorrelation at that lag.
- The spikes at all subsequent lags are within the confidence interval, suggesting that there is no significant autocorrelation beyond lag 1.

The multiplicative decomposition and ACF plots seems to be fairly similar.

Appendix C - General Overview on the Bayesian Structural Time Series (BSTS) Model

The Bayesian Structural Time Series (BSTS) model is a statistical method that serves several applications such as feature selection, time series forecasting, and causal inference analysis. It operates on time series data to ascertain underlying patterns and forecast future data points.

BSTS models are composed of three primary elements:

1. *Kalman Filter:* A method for decomposing time series into components like trend and seasonality, allowing state variables to be modeled dynamically over time.
2. *Spike-and-Slab Prior:* A technique for feature selection that identifies which predictors in a regression are most informative.
3. *Bayesian Model Averaging (BMA):* A process where multiple models are averaged together to produce predictions or infer parameters, accounting for model uncertainty. In the BSTS framework, BMA is utilized extensively to generate samples from repeated Markov Chain Monte Carlo (MCMC) simulations into final model outputs, providing a comprehensive prediction that encompasses model variability.

A general BSTS model consists of twos set of equations,

1. *Observational equation:* Response variable as a function of predictors and/or latent variables.
2. *State-space equations:* How the parameters evolve over the time.

BSTS models are usually implemented using the `bsts` package in R, but can also be implemented using `rstan`. Stan's advanced MCMC algorithms enhance the flexibility and scalability of BSTS models, making it a powerful tool for time series analysis.

References

"Bayesian Structural Time Series." *SAP HANA Predictive Analysis Library (PAL)*, help.sap.com/docs/SAP_HANA_PLATFORM/2cfbc5cf2bc14f028cfbe2a2bba60a50/b9972576368640da9831d73a9d749c3b.html. Accessed 8 Apr. 2024.

Radtke, Tim. "Minimize Regret." *Minimize Regret - Rediscovering Bayesian Structural Time Series*, minimizeregret.com/post/2020/06/07/rediscovering-bayesian-structural-time-series/. Accessed 9 Apr. 2024.

Yabe, Taka. "Pystan - Causal Inference Using Bayesian Structural Time Series." *Takahiro Yabe*, 21 Feb. 2021, www.takayabe.net/post/pystan-bsts. Accessed 9 Apr. 2024.

Appendix D - Estimation of Hyperparameters for BSTS Model

Before we can specify the BSTS model for the US Unemployment rates, we need to estimate the hyperparameters that will inform the prior distributions. This involves analyzing historical unemployment rate data to inform the choice of prior distributions for various model components, specifically the observation noise variance, trend variance, seasonal variance, and white noise variance. Note, that we are assuming that the white noise variance is a fraction of the observation noise.

```
obs_noise_variance_estimate <- var(diff(ts_unrate_train))

trend_model <- lm(ts_unrate_train ~ time(ts_unrate_train))
trend_variance_estimate <- var(resid(trend_model))

seasonal_differences <- diff(ts_unrate_train, lag = 12)
seasonal_variance_estimate <- var(seasonal_differences)

alpha <- 2

beta_obs_noise <- obs_noise_variance_estimate * (alpha - 1)
beta_trend <- trend_variance_estimate * (alpha - 1)
beta_seasonal <- seasonal_variance_estimate * (alpha - 1)

cat("Inverse-Gamma alpha:", alpha, "\n")
```

```
## Inverse-Gamma alpha: 2
```

```
cat("Inverse-Gamma beta for Observation Noise:", beta_obs_noise, "\n")
```

```
## Inverse-Gamma beta for Observation Noise: 0.04663646
```

```
cat("Inverse-Gamma beta for Trend:", beta_trend, "\n")
```

```
## Inverse-Gamma beta for Trend: 2.348358
```

```
cat("Inverse-Gamma beta for Seasonal:", beta_seasonal, "\n")
```

```
## Inverse-Gamma beta for Seasonal: 1.528651
```

Appendix E - BSTS Model in Stan

```
us_unemployment_rates_bsts_model <- "
data {
  int<lower=1> T;                       // Number of time points
  vector[T] y;                          // Observed unemployment rates
}

parameters {
  vector[T] mt;                         // Trend component
  vector[T] st;                         // Seasonal component
  real<lower=0> sigma_y;                // Std dev of observation noise
  real<lower=0> sigma_m;                // Std dev of trend component
  real<lower=0> sigma_s;                // Std dev of seasonal component
  real<lower=-1, upper=1> phi;          // Autoregressive coefficient
}

model {
  // Hyperpriors
  sigma_y ~ inv_gamma(2, 0.04663646);
  sigma_m ~ inv_gamma(2, 2.348358);
  sigma_s ~ inv_gamma(2, 1.528651);
  phi ~ normal(0, 1);                   // Autoregressive prior

  // Prior for the initial trend component
  mt[1] ~ normal(0, sigma_m);

  // Trend component model
  for (t in 2:T) {
    mt[t] ~ normal(mt[t-1], sigma_m);
  }

  // Priors for the initial seasonal component values
  for (s in 1:12) {
    st[s] ~ normal(0, sigma_s);
  }

  // Seasonal component model
  for (t in 13:T) {
    st[t] ~ normal(-sum(st[(t-11):(t-1)]) / 11, sigma_s);
  }

  // Initial observation
  y[1] ~ normal(mt[1] + st[1], sigma_y);

  // Observation model with AR(1) process
  for (t in 2:T) {
    y[t] ~ normal(mt[t] + st[t] + phi * y[t-1], sigma_y);
  }
}"
```

Appendix F - BSTS Model Implementation using the `bsts` Library

```
# # Model specification
# state_space <- list()
#
# # Adding a local level (trend) component
# state_space <- AddLocalLinearTrend(state_space, ts_unrate_train)
#
# # Adding a seasonal component with a 12-month cycle
# state_space <- AddSeasonal(state_space, y = ts_unrate_train, nseasons = 12)
#
# # Adding an AR(1) component
# state_space <- AddAr(state_space, y = ts_unrate_train, lags = 1)
#
# # Fit the BSTS model
# model <- bsts(ts_unrate_train, state.specification = state_space, niter = 10000)
#
# # Predicting future values
# prediction <- predict(model, horizon = length(ts_unrate_test), burn = SuggestBurn(0.1, model))
#
# # Plotting the forecast with credible intervals
# plot(prediction, plot.original = 12)
#
# # Calculating MAE and RMSE
# predicted_means <- as.numeric(prediction$mean)
# actuals <- as.numeric(ts_unrate_test)
#
# # MAE
# mae <- mean(abs(predicted_means - actuals))
# cat("Mean Absolute Error (MAE):", mae, "\n")
#
# # RMSE
# rmse <- sqrt(mean((predicted_means - actuals)^2))
# cat("Root Mean Squared Error (RMSE):", rmse, "\n")
```

Appendix [] - []

[To be added]