# Forecasting Time Series with Holt-Winters and ARIMA Models: Imputing COVID-19 Impacts on Initial Claims Data

JAHNAVI GANGU

2024-09-09

```r
# Library Loading
library(tidyverse)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(fredr)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

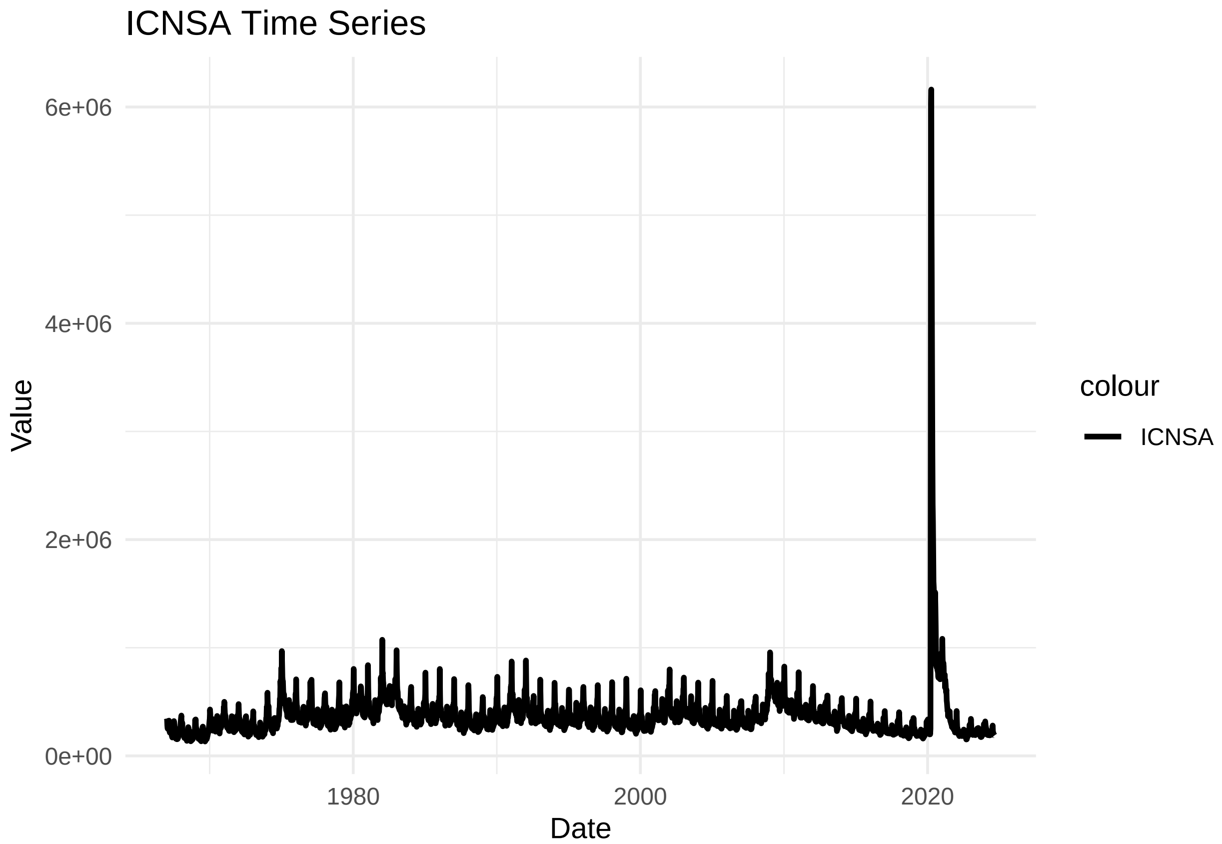
```r
library(splines)
library(urca)
library(dplyr)
library(readxl)
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
# Fetching ICNSA data from FRED
fredr_set_key("c387f7cbc3f36a5a52a03d391d17e253")
icnsa_data <- fredr(series_id = "ICNSA")
```

```r
# Convert 'date' column to Date format
icnsa_data$date <- as.Date(icnsa_data$date)
```

```r
# Plot original time series
ggplot() +
  geom_line(data = icnsa_data, aes(x = date, y = value, color = "ICNSA"), linewidth = 1) +
  labs(title = "ICNSA Time Series", x = "Date", y = "Value") +
  scale_color_manual(values = c("ICNSA" = "black")) +
  theme_minimal()
```



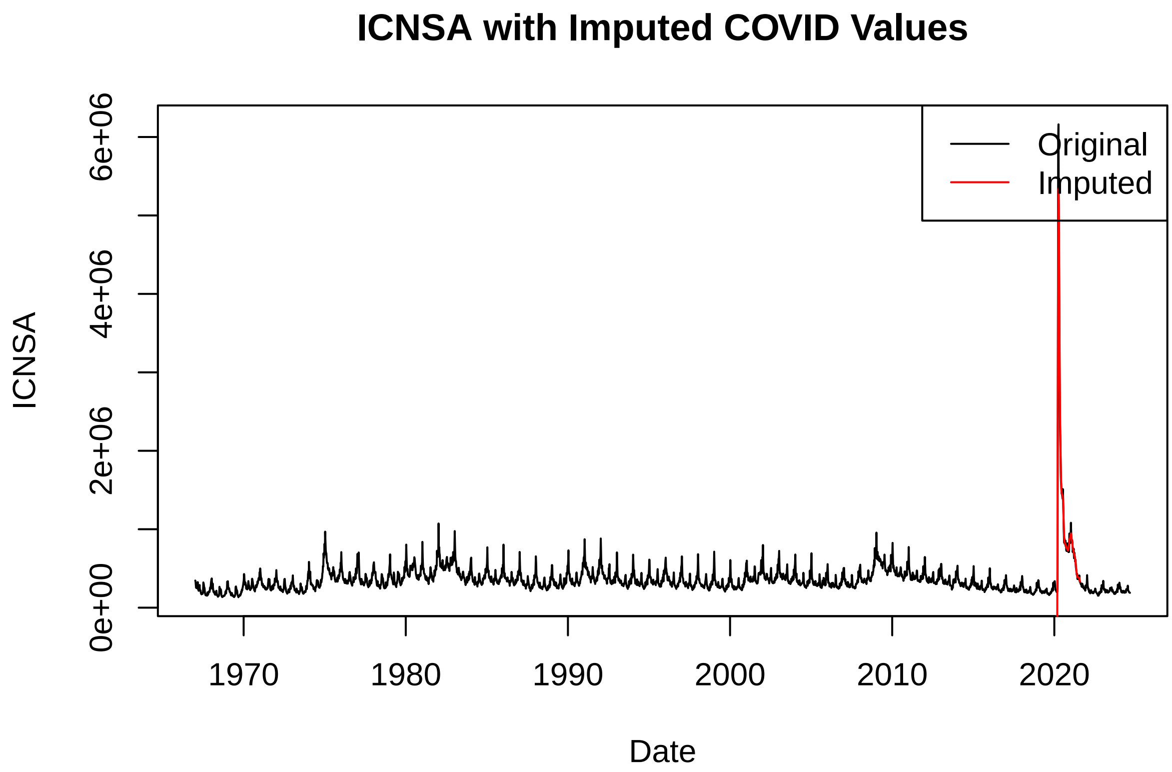

```r
# Set start and end dates for COVID period
start_date <- as.Date("2020-03-01")
end_date <- as.Date("2021-07-31")
```

```r
# Creating cubic spline model to impute values for the COVID period
covid_period <- icnsa_data %>%
  filter(date >= start_date & date <= end_date)
```

```r
# Fit cubic spline with automatic lambda using cross-validation
spline_fit <- smooth.spline(x = as.numeric(covid_period$date), y = covid_period$value, cv = TRUE)
```

```r
# Predict values for the COVID period
imputed_values <- predict(spline_fit, x = as.numeric(covid_period$date))
```

```r
# Plot original data and imputed values
plot(icnsa_data$date, icnsa_data$value, type = "l", xlab = "Date", ylab = "ICNSA", main = "ICNSA with In
lines(covid_period$date, imputed_values$y, col = "red")
legend("topright", legend = c("Original", "Imputed"), col = c("black", "red"), lty = 1)
```




```r
# Converting 'value' column to numeric
icnsa_data$value <- as.numeric(icnsa_data$value)
```

```r
# Combining original and imputed data
imputed_data <- data.frame(date = covid_period$date, value = imputed_values$y)
combined_data <- bind_rows(icnsa_data %>% filter(date < start_date),
                           imputed_data,
                           icnsa_data %>% filter(date > end_date))
```

```r
# Converting to time series
ts_data <- ts(combined_data$value, start = c(year(min(combined_data$date)), month(min(combined_data$dat

# Fitting multiplicative Holt-Winters model
hw_multiplicative <- HoltWinters(ts_data, seasonal = "multiplicative")
forecast_multiplicative <- forecast(hw_multiplicative, h = 1)

# Fitting additive Holt-Winters model
hw_additive <- HoltWinters(ts_data, seasonal = "additive")
forecast_additive <- forecast(hw_additive, h = 1)

# Accuracy comparison between the two models
accuracy_multiplicative <- accuracy(forecast_multiplicative)
accuracy_additive <- accuracy(forecast_additive)

# Print accuracy for both models
cat("Accuracy for Multiplicative Model:\n")
```

```
## Accuracy for Multiplicative Model:
```

```r
print(accuracy_multiplicative)
```

```
##                    ME     RMSE     MAE        MPE     MAPE      MASE      ACF1
## Training set 551.8738 85044.87 35659.9 -0.5734456 8.468546 0.3784939 0.4648245
```

```r
cat("Accuracy for Additive Model:\n")
```

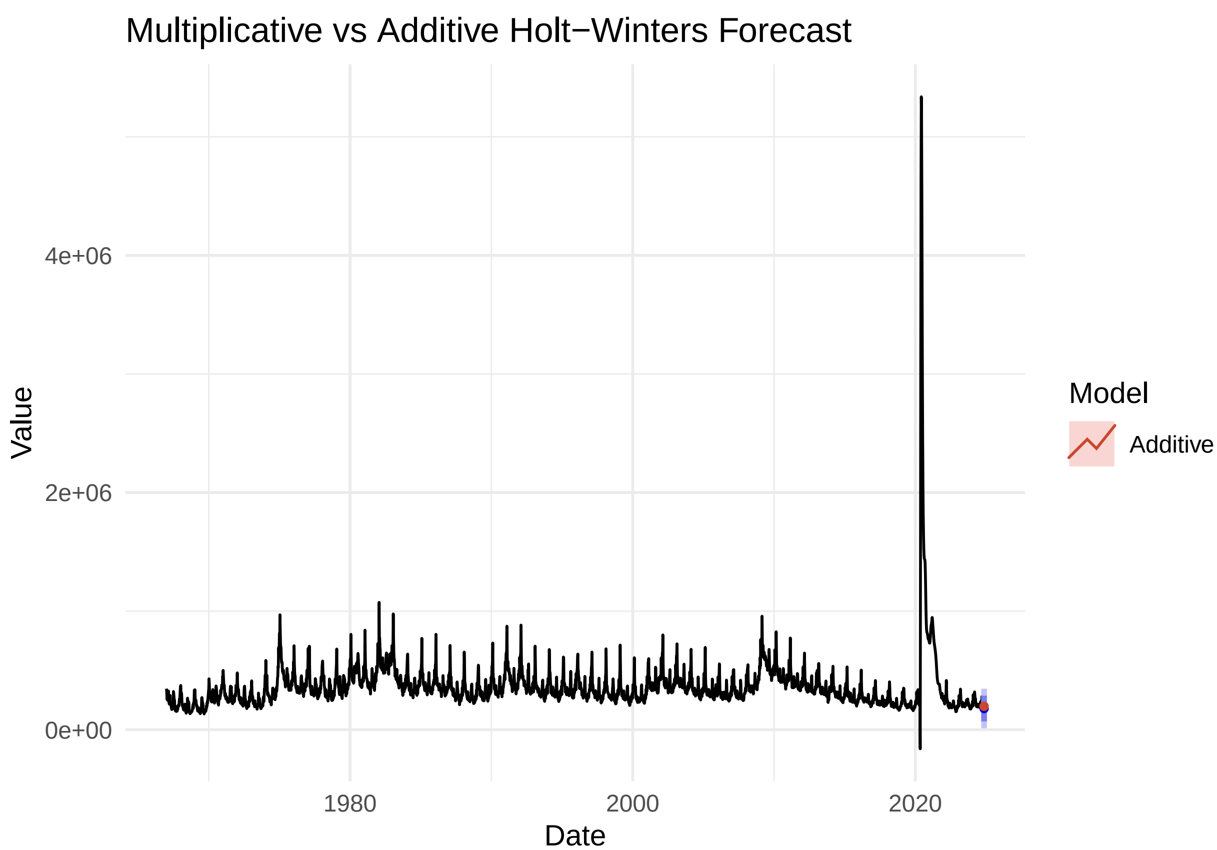
```
## Accuracy for Additive Model:
```

```r
print(accuracy_additive)
```

```
##                    ME     RMSE      MAE        MPE     MAPE      MASE      ACF1
## Training set 593.4192 85617.23 45044.44 -0.8152771 11.83285 0.4781012 0.2819634
```

```r
# Point forecasts for both models
point_forecast_multiplicative <- forecast_multiplicative$mean[[1]]
point_forecast_additive <- forecast_additive$mean[[1]]
```

```r
# Plot both forecasts on the same plot for comparison
autoplot(forecast_multiplicative) +
  autolayer(forecast_additive, series = "Additive", PI = FALSE) +
  labs(title = "Multiplicative vs Additive Holt-Winters Forecast", x = "Date", y = "Value") +
  guides(colour = guide_legend(title = "Model")) +
  theme_minimal()
```




```r
# Print point forecasts
cat("Point forecast using multiplicative Holt-Winters model:", point_forecast_multiplicative, "\n")
```

```
## Point forecast using multiplicative Holt-Winters model: 178612.2
```

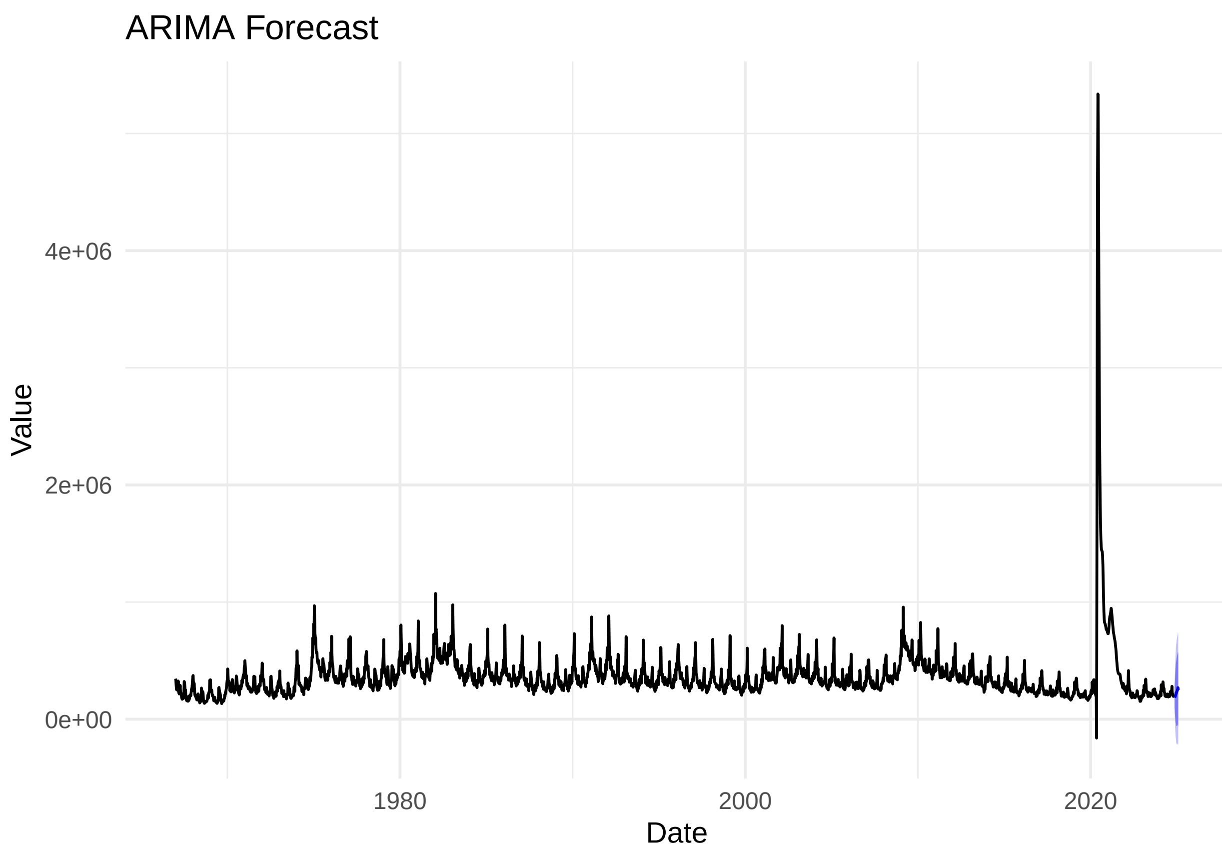
```r
cat("Point forecast using additive Holt-Winters model:", point_forecast_additive, "\n")
```

```
## Point forecast using additive Holt-Winters model: 197219.3
```

```r
# Fit ARIMA model on the time series data
arima_model <- auto.arima(ts_data)

# Forecast using the ARIMA model
arima_forecast <- forecast(arima_model, h = 12)

# Plot the ARIMA forecast
autoplot(arima_forecast) +
  labs(title = "ARIMA Forecast", x = "Date", y = "Value") +
  theme_minimal()
```




```r
# Calculate accuracy metrics for ARIMA model
arima_accuracy <- accuracy(arima_forecast)
cat("ARIMA Model Accuracy:\n")
```

```
## ARIMA Model Accuracy:
```

```r
print(arima_accuracy)
```

```
##                     ME     RMSE      MAE        MPE     MAPE     MASE
## Training set -80.70336 61920.95 30494.94 -0.9582173 8.099736 0.323673
##                       ACF1
## Training set -0.000973471
```