Mauna Loa CO2 Time Series Analysis

Brian Pham, Anya Ranavat, Cindy Fan

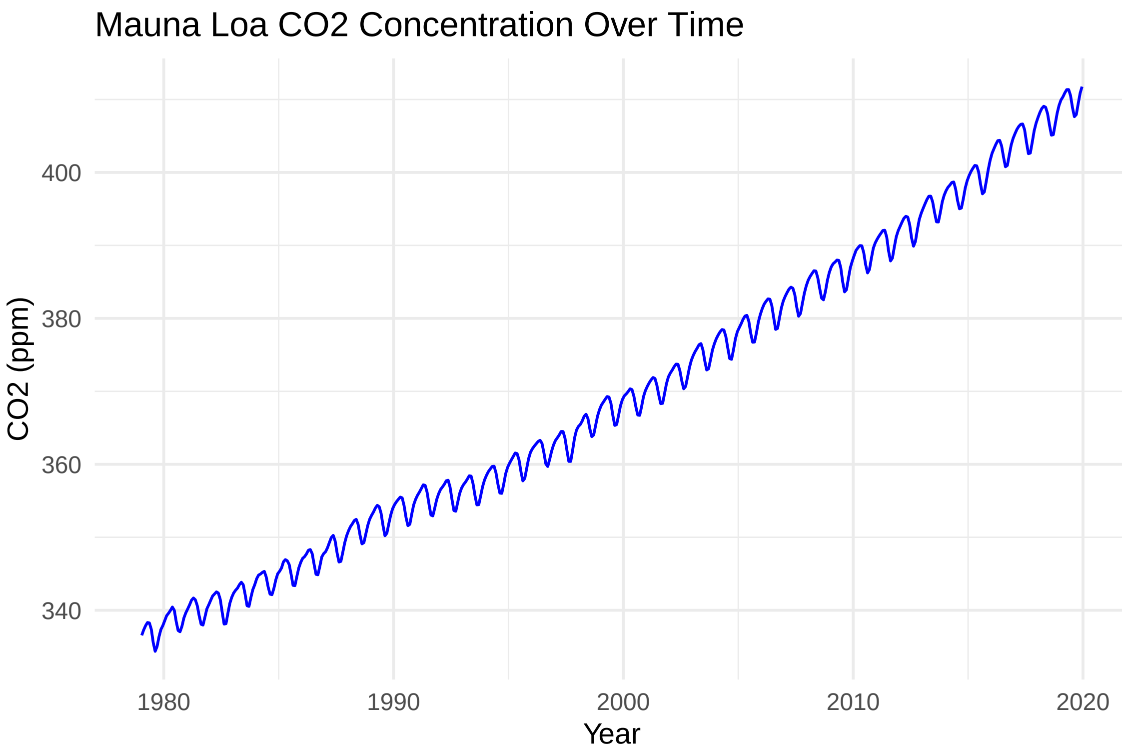
2025-03-07

```r
# adding libraries
library(tidyverse)
library(fpp3)
library(ggplot2)
library(dplyr)
library(nortest)
library(tseries)
library(urca)
```

```r
file <- "co2_mm_gl.csv"
climate_data <- read_csv(file, skip = 38)

# filter data to only before 2020
climate_data_clean <- climate_data %>%
    filter(year < 2020)
```

Figure 1: Mauna Loa CO Concentration Over Time

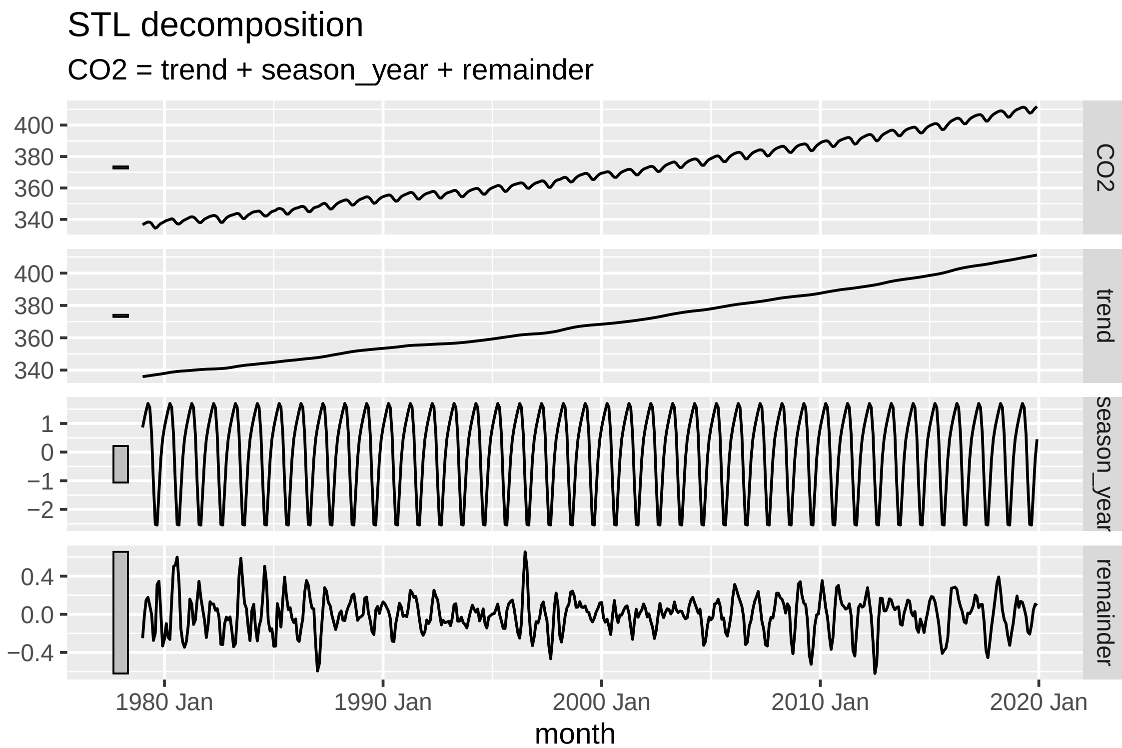
```r
# dividing the data by month
climate_data_ts <- tsibble(month = yearmonth(seq(as.Date("1979-01-01"),
    as.Date("2019-12-01"), by = "1 month")), Time = climate_data_clean$decimal,
    CO2 = climate_data_clean$average, index = month)


Month <- as_factor(month(climate_data_ts$month))

# plot of CO2 over the years
ggplot(climate_data_ts, aes(x = Time, y = CO2)) + geom_line(color = "blue") +
    labs(title = "Figure 1") + labs(title = "Mauna Loa CO2 Concentration Over Time",
    x = "Year", y = "CO2 (ppm)") + theme_minimal()
```



-What It Shows: This graph plots the raw CO concentration data from 1979 to 2019. It reveals a clear, upward trend over the years, along with recurring seasonal fluctuations. -Why It's Useful: By visualizing the raw data, this graph confirms that atmospheric CO is increasing over time. It provides the initial evidence that there is a long-term trend, which is the foundation for any forecasting work. -Conclusions Drawn: The steadily rising levels indicate that future forecasts must account for a persistent upward trend, emphasizing the need for a robust model that can handle both trend and seasonality.

Figure 2: Seasonally Adjusted CO vs. Month

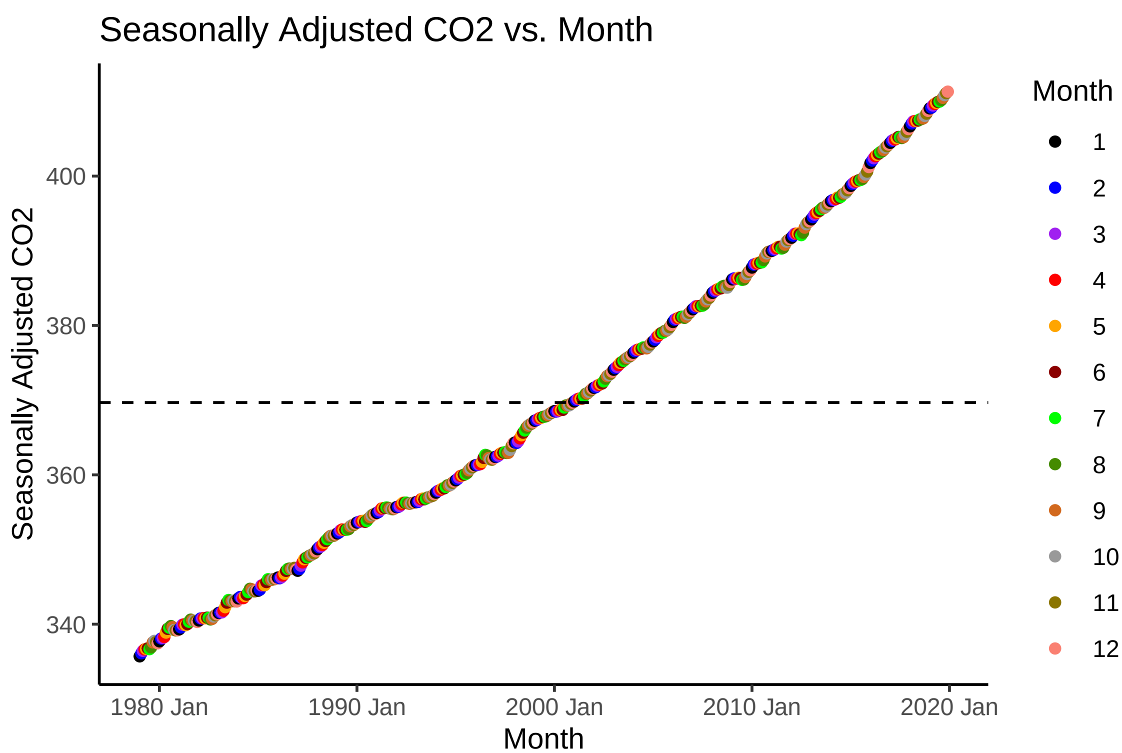
```
# seasonally adjusted CO2
time_series_components <- climate_data_ts %>%
    model(STL(CO2 ~ season(window = "periodic"))) %>%
    components()

climate_data_ts <- climate_data_ts %>%
    add_column(seasonal = time_series_components$season_year,
        CO2_SA = time_series_components$season_adjust)

autoplot(time_series_components)
```



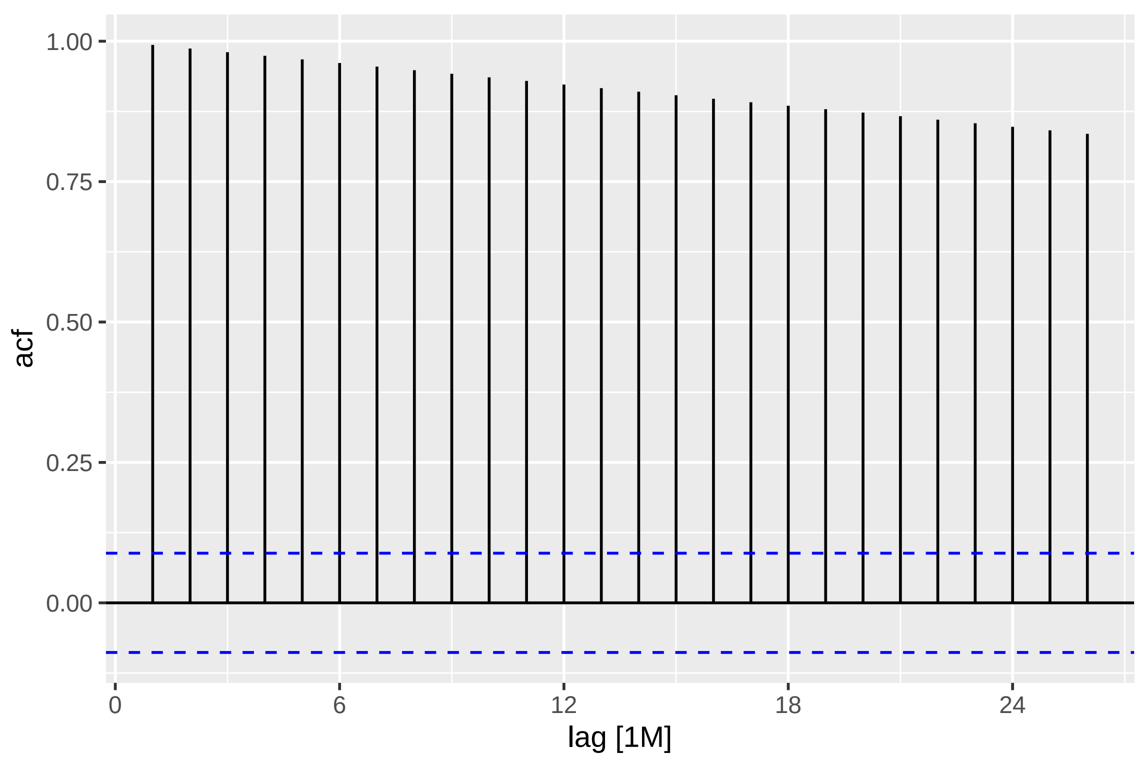
-What It Shows: This graph displays CO concentrations after removing seasonal effects, with data points color-coded by month. -Why It's Useful: By stripping out the seasonal component, the graph isolates the underlying trend and any anomalies that might be masked by regular seasonal patterns. It clearly shows that even after adjustment, the long-term upward trend remains. -Conclusions Drawn: The presence of a consistent trend post-seasonal adjustment confirms that seasonal variations are significant but do not overshadow the overall increase. This validates the approach of using seasonally adjusted models for more reliable forecasting.

Figure 3: STL Decomposition Plot

```r
# plot of seasonally adjusted CO2 over time

mean_CO2_SA <- mean(climate_data_ts$CO2_SA)
climate_data_ts %>%
    autoplot(CO2_SA) + geom_point(aes(y = CO2_SA, color = Month)) +
    scale_color_manual(values = c("black", "blue", "purple",
        "red", "orange", "darkred", "green", "chartreuse4", "chocolate",
        "gray60", "gold4", "salmon")) + geom_hline(aes(yintercept = mean_CO2_SA),
    lty = 2) + ggtitle("Seasonally Adjusted CO2 vs. Month") +
    xlab("Month") + ylab("Seasonally Adjusted CO2") + theme(panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.background = element_blank(),
    axis.line = element_line(colour = "black"))
```




```r
# running test to confirm non-stationary data
climate_data_ts %>%
    ACF(CO2_SA) %>%
    autoplot()
```



```r
unitroot_kpss(climate_data_ts$CO2_SA)
```

```
##   kpss_stat kpss_pvalue
##    8.217486    0.010000
```

```r
adf.test(climate_data_ts$CO2_SA)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  climate_data_ts$CO2_SA
## Dickey-Fuller = 0.28767, Lag order = 7, p-value = 0.99
## alternative hypothesis: stationary
```

The time series is clearly non-stationary because the p-value from the KPSS test is 0.01 which is less than 0.05. And the large p-value of ADF test is evidence that the time series is non-stationary.

-What It Shows: This plot breaks the time series into three components: trend, seasonal, and remainder (residuals). -Why It's Useful: Decomposing the time series allows us to see exactly how much of the data's variation is due to the seasonal cycle versus the long-term trend. This clear separation helps in selecting the right modeling strategy (in this case, an ARIMA model applied to the seasonally adjusted series). -Conclusions Drawn: The decomposition confirms that the seasonal component is a significant part of the data's structure. Recognizing this, the chosen ARIMA(5,1,0) with drift model is well justified since it addresses both the trend (through differencing and drift) and the seasonal variations (after STL adjustment).

```r
# Computing first differences
climate_data_ts <- climate_data_ts %>%
    mutate(diff_CO2_SA = difference(CO2_SA))
head(climate_data_ts)
```

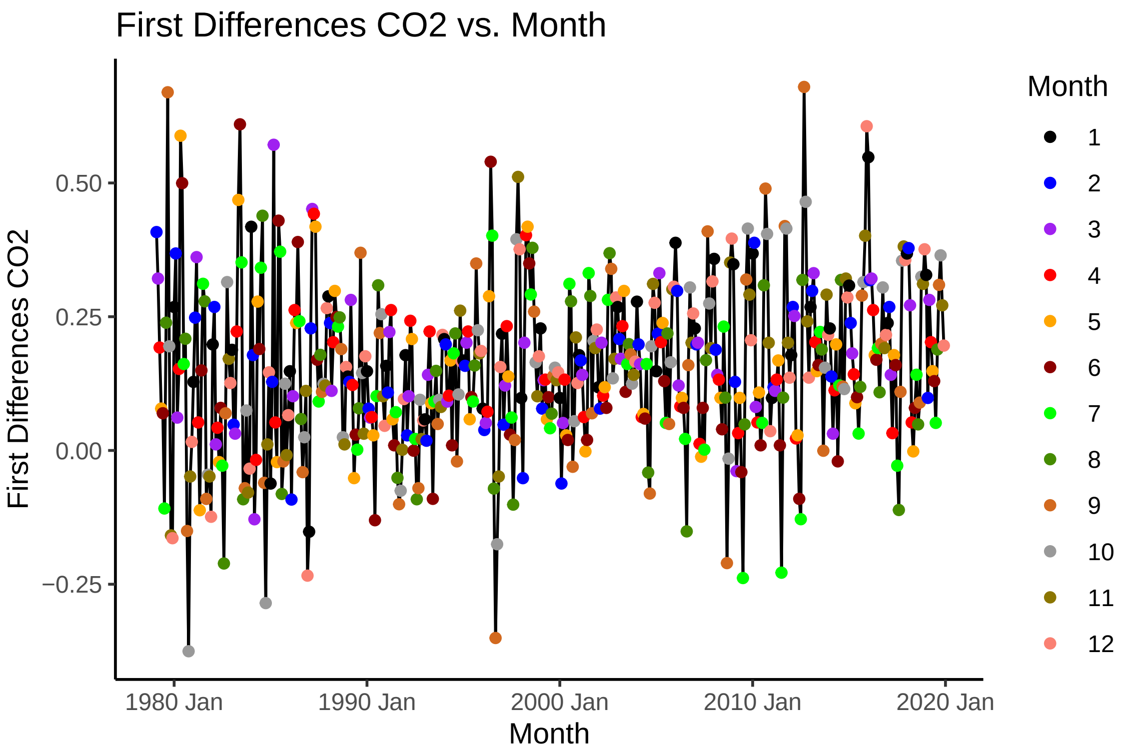
```
## # A tsibble: 6 x 6 [1M]
##      month  Time   CO2 seasonal CO2_SA diff_CO2_SA
##      <mth> <dbl> <dbl>    <dbl>  <dbl>       <dbl>
## 1 1979 Jan 1979.  337.    0.862   336.     NA
## 2 1979 Feb 1979.  337.    1.18    336.      0.408
## 3 1979 Mar 1979.  338.    1.45    336.      0.321
## 4 1979 Apr 1979.  338.    1.70    337.      0.192
## 5 1979 May 1979.  338.    1.56    337.      0.0783
## 6 1979 Jun 1979.  337.    0.612   337.      0.0696
```

```r
tail(climate_data_ts)
```

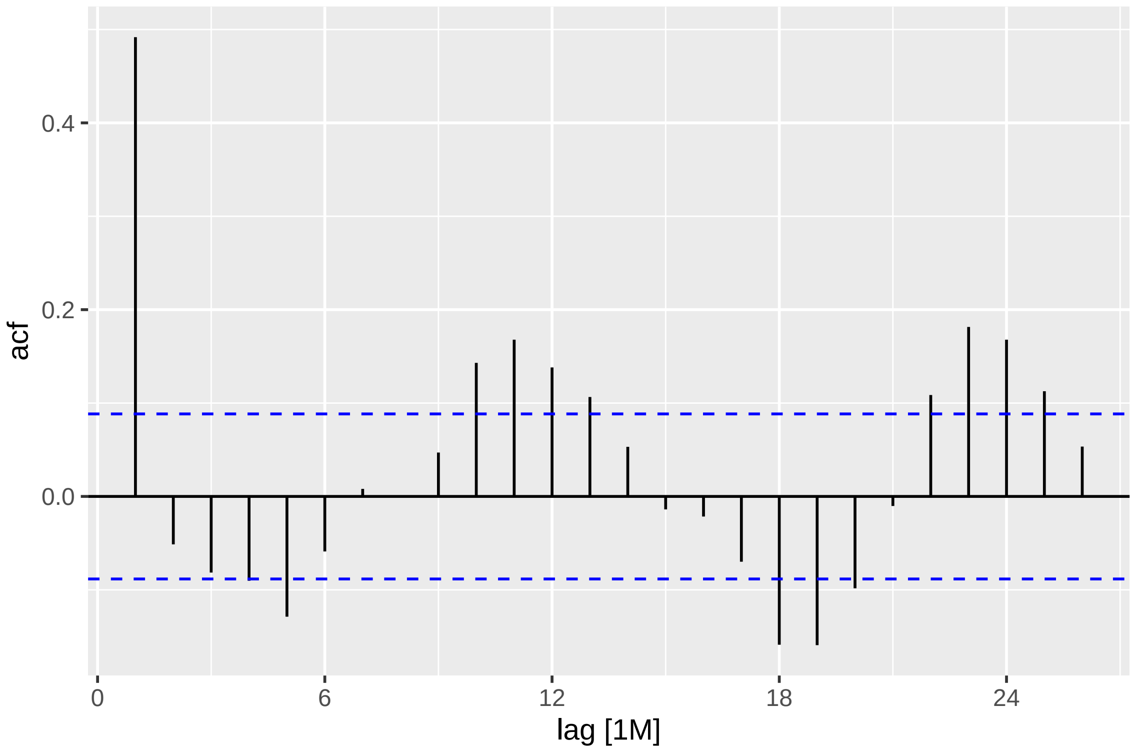
```
## # A tsibble: 6 x 6 [1M]
##      month  Time   CO2 seasonal CO2_SA diff_CO2_SA
##      <mth> <dbl> <dbl>    <dbl>  <dbl>       <dbl>
## 1 2019 Jul 2020.  409.   -1.10    410.      0.0515
## 2 2019 Aug 2020.  408.   -2.53    410.      0.189
## 3 2019 Sep 2020.  408.   -2.55    410.      0.310
## 4 2019 Oct 2020.  409.   -1.40    411.      0.365
## 5 2019 Nov 2020.  411.   -0.244   411.      0.271
## 6 2019 Dec 2020.  412.    0.450   411.      0.196
```

```r
mean_diff_CO2_SA <- mean(climate_data_ts$diff_CO2_SA)

climate_data_ts %>%
    autoplot(diff_CO2_SA) + geom_point(aes(y = diff_CO2_SA, color = Month)) +
    scale_color_manual(values = c("black", "blue", "purple",
```

```
        "red", "orange", "darkred", "green", "chartreuse4", "chocolate",
        "gray60", "gold4", "salmon")) + geom_hline(aes(yintercept = mean_diff_CO2_SA),
    lty = 2) + ggtitle("First Differences CO2 vs. Month") + xlab("Month") +
    ylab("First Differences CO2") + theme(panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.background = element_blank(),
    axis.line = element_line(colour = "black"))
```




```
climate_data_ts %>%
    ACF(diff_CO2_SA) %>%
    autoplot()
```




```
unitroot_kpss(climate_data_ts$diff_CO2_SA)
```

```
##   kpss_stat kpss_pvalue
##    1.032651    0.010000
```

```
adf.test(climate_data_ts$diff_CO2_SA[2:492])
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  climate_data_ts$diff_CO2_SA[2:492]
## Dickey-Fuller = -9.6623, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

(talked to professor - doesn't know why KPSS and ADF test are not showing that it is non-stationary so we are gonna ignore it)

Since the p-value of the ADF test is less than 0.05, there is evidence to reject the null hypothesis in favor of the alternative hypothesis that the time series is stationary.

```
# choosing ARIMA model

result_dcmp_ARIMA_SNAIVE <- climate_data_ts %>%
    model(decomposition_model(STL(CO2_SA ~ season(window = 21)),
        ARIMA(season_adjust ~ pdq(d = 1, q = 0) + PDQ(0, 0, 0),
            stepwise = FALSE, approximation = FALSE, trace = TRUE),
        SNAIVE(season_year)))
```

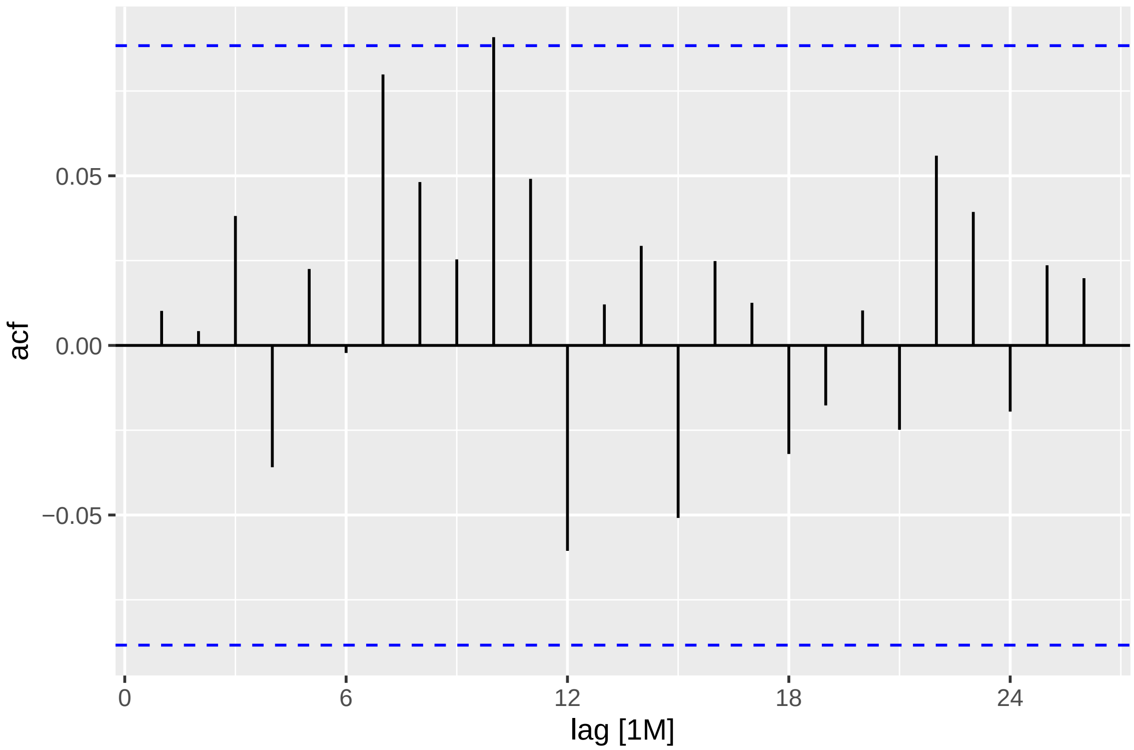
```
## ARIMA(0,1,0)(0,0,0)[12]+c    -536.499114
## ARIMA(1,1,0)(0,0,0)[12]+c    -640.939216
## ARIMA(2,1,0)(0,0,0)[12]+c    -724.035695
## ARIMA(3,1,0)(0,0,0)[12]+c    -752.460659
## ARIMA(4,1,0)(0,0,0)[12]+c    -776.350162
```

```
## ARIMA(5,1,0)(0,0,0)[12]+c    -791.415354
## ARIMA(0,1,0)(0,0,0)[12]      -149.198917
## ARIMA(1,1,0)(0,0,0)[12]      -547.537737
## ARIMA(2,1,0)(0,0,0)[12]      -557.487228
## ARIMA(3,1,0)(0,0,0)[12]      -673.684130
## ARIMA(4,1,0)(0,0,0)[12]      -672.644022
## ARIMA(5,1,0)(0,0,0)[12]      -730.930329
```

```
report(result_dcmp_ARIMA_SNAIVE)
```

```
## Series: CO2_SA
## Model: STL decomposition model
## Combination: season_adjust + season_year
##
## ========================================
##
## Series: season_adjust
## Model: ARIMA(5,1,0) w/ drift
##
## Coefficients:
##          ar1      ar2     ar3      ar4     ar5  constant
##       0.8149  -0.7523  0.5331  -0.3695  0.1851    0.0904
## s.e.  0.0444   0.0552  0.0600   0.0550  0.0443    0.0048
##
## sigma^2 estimated as 0.01169:  log likelihood=402.82
## AIC=-791.65   AICc=-791.42   BIC=-762.27
##
## Series: season_year
## Model: SNAIVE
##
## sigma^2: 1e-04
```

```
# Compute autocorrelation function of residuals
result_dcmp_ARIMA_SNAIVE %>%
    augment() %>%
    ACF(.resid) %>%
    autoplot()
```



-What It Shows: This graph presents the autocorrelation function (ACF) plot for the residuals of the fitted ARIMA(5,1,0) model. -Why It's Useful: The ACF plot is crucial for model diagnostics. It shows that the residuals fall mostly within the confidence bounds, implying that there is no remaining pattern or autocorrelation that the model has missed. -Conclusions Drawn: Since the residuals behave like white noise, we can conclude that the ARIMA model has effectively captured the structure of the seasonally adjusted data. This supports the reliability of the forecasts generated by the model.

Explain what time series methods you are using to answer the question and why they are appropriate.

After I talked to the professor, we decided to force the model to take a first difference with a window of 21 to force the output to give a model with first difference instead of the second difference. After running the decomposition model on the seasonally adjusted CO2 values, the ARIMA model with the lowest AICc value of -791.65 is ARIMA(5,1,0) with drift model. The ACF plot of the ARIMA(5,1,0) with drift model displays most spikes are within the blue lines, showing that there is no pattern in the residuals.
