Report from the Point of View of the Present

2024-03-11

```r
library(readr)
library(dplyr)
library(lubridate)
library(tsibble)
library(ggplot2)
if(!"fpp3"%in%rownames(installed.packages())) {install.packages("fpp3")}
library(fpp3)
if(!"grwat"%in%rownames(installed.packages())) {install.packages("grwat")}
library(grwat)
if(!"dplyr"%in%rownames(installed.packages())) {install.packages("dplyr")}
library(dplyr)
```

#(1 point) Task 0b: Introduction

In this introduction, you can assume that your reader will have just read your 1997 report. In this introduction, very briefly pose the question that you are evaluating, and describe what (if anything) has changed in the data generating process between 1997 and the present.

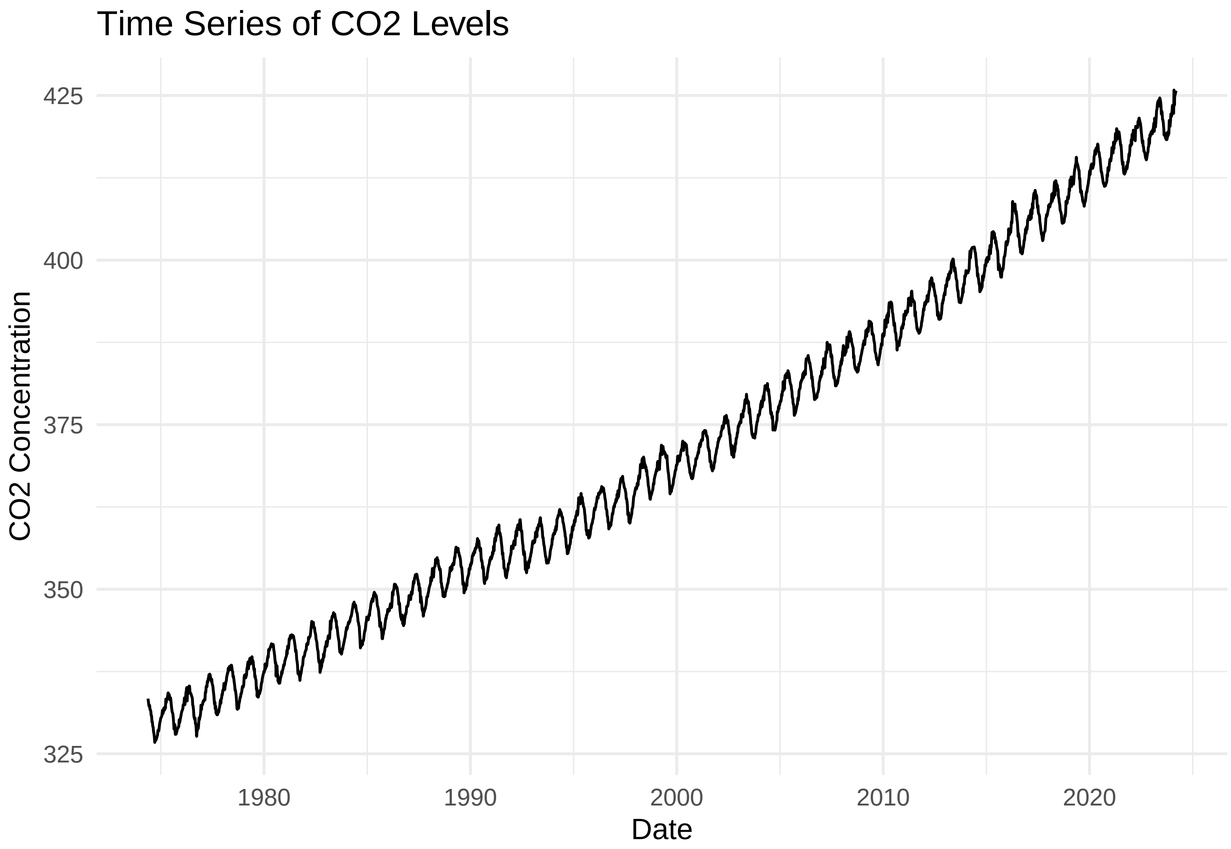
#(3 points) Task 1b: Create a modern data pipeline for Mona Loa CO2 data.

The most current data is provided by the United States' National Oceanic and Atmospheric Administration, on a data page [here]. Gather the most recent weekly data from this page. (A group that is interested in even more data management might choose to work with the hourly data.) Create a data pipeline that starts by reading from the appropriate URL, and ends by saving an object called co2_present that is a suitable time series object. Conduct the same EDA on this data. Describe how the Keeling Curve evolved from 1997 to the present, noting where the series seems to be following similar trends to the series that you "evaluated in 1997" and where the series seems to be following different trends. This EDA can use the same, or very similar tools and views as you provided in your 1997 report.

```r
weekly_co2_url <- "https://gml.noaa.gov/webdata/ccgg/trends/co2/co2_weekly_mlo.csv"

content <- read_lines(weekly_co2_url, skip_empty_rows = TRUE)

header_end_index <- max(grep("^#", content))

co2_present <- read_csv(weekly_co2_url, skip = header_end_index, show_col_types = FALSE)

co2_present <- co2_present %>%
  mutate(date = make_date(year, month, day))

co2_present <- as_tsibble(co2_present, index = date)

co2_present <- co2_present[co2_present$average != -999.99, ]

co2_present
```

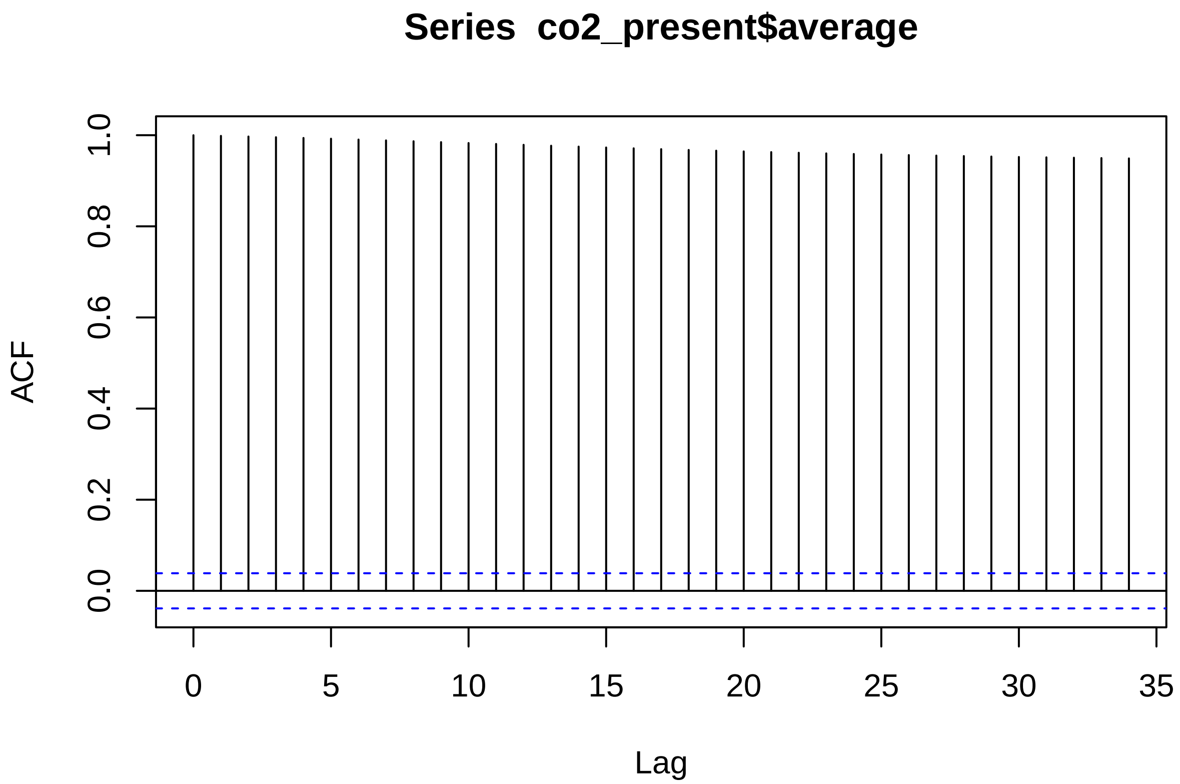
```
## # A tsibble: 2,582 x 10 [7D]
##     year month   day decimal average ndays ‘1 year ago‘ ‘10 years ago‘
##    <dbl> <dbl> <dbl>   <dbl>   <dbl> <dbl>        <dbl>          <dbl>
##  1  1974     5    19   1974.    333.     5       -1000.         -1000.
##  2  1974     5    26   1974.    333.     6       -1000.         -1000.
##  3  1974     6     2   1974.    332.     5       -1000.         -1000.
##  4  1974     6     9   1974.    332.     7       -1000.         -1000.
##  5  1974     6    16   1974.    332.     7       -1000.         -1000.
##  6  1974     6    23   1974.    332.     5       -1000.         -1000.
##  7  1974     6    30   1974.    332.     6       -1000.         -1000.
##  8  1974     7     7   1975.    331.     6       -1000.         -1000.
##  9  1974     7    14   1975.    331.     5       -1000.         -1000.
## 10  1974     7    21   1975.    331.     7       -1000.         -1000.
## # i 2,572 more rows
## # i 2 more variables: ‘increase since 1800‘ <dbl>, date <date>
```

```r
ggplot(co2_present, aes(x = date, y = average)) +
  geom_line() +
  theme_minimal() +
  labs(title = "Time Series of CO2 Levels", x = "Date", y = "CO2 Concentration")
```



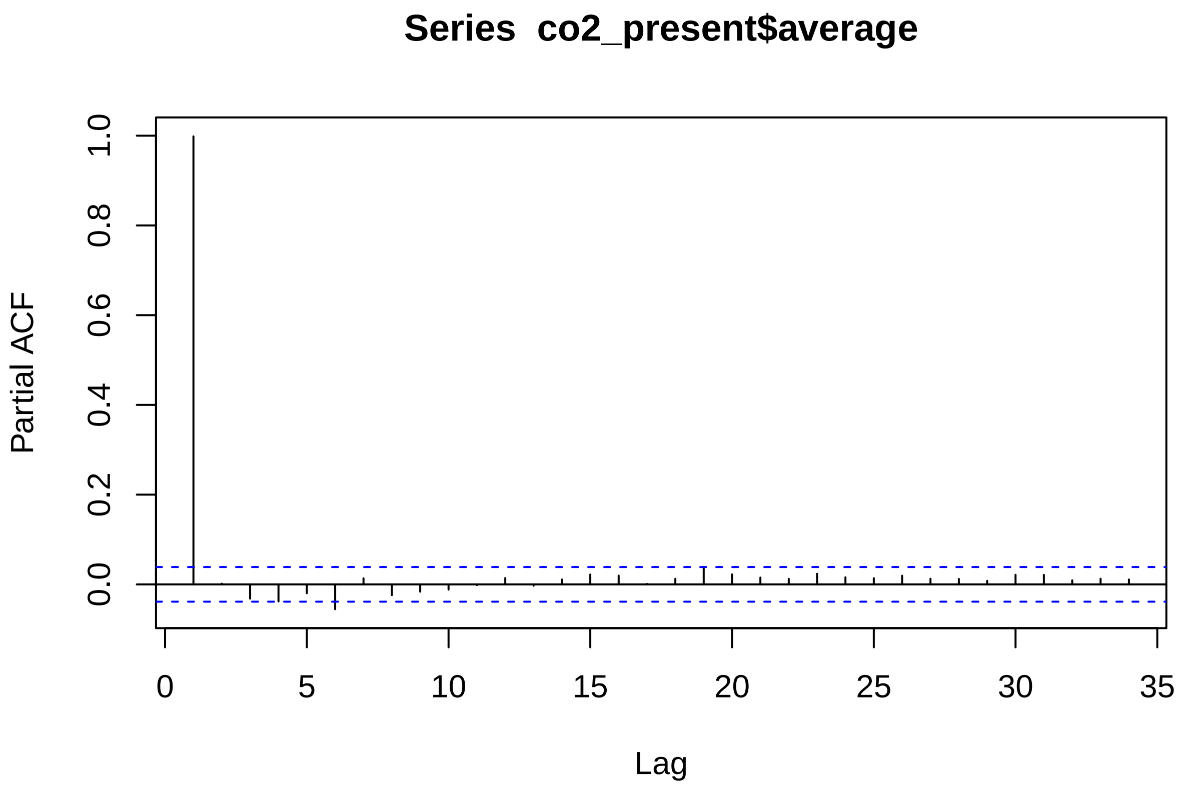
The plot shows a clear upward trend, indicating that CO2 levels have been increasing over the given time period. The pattern also shows seasonal fluctuations within each year, where the CO2 concentration peaks and then drops slightly, before rising again. The overall trend, however, is an increase in CO2 levels.

```
acf(co2_present$average, na.action = na.pass)
```



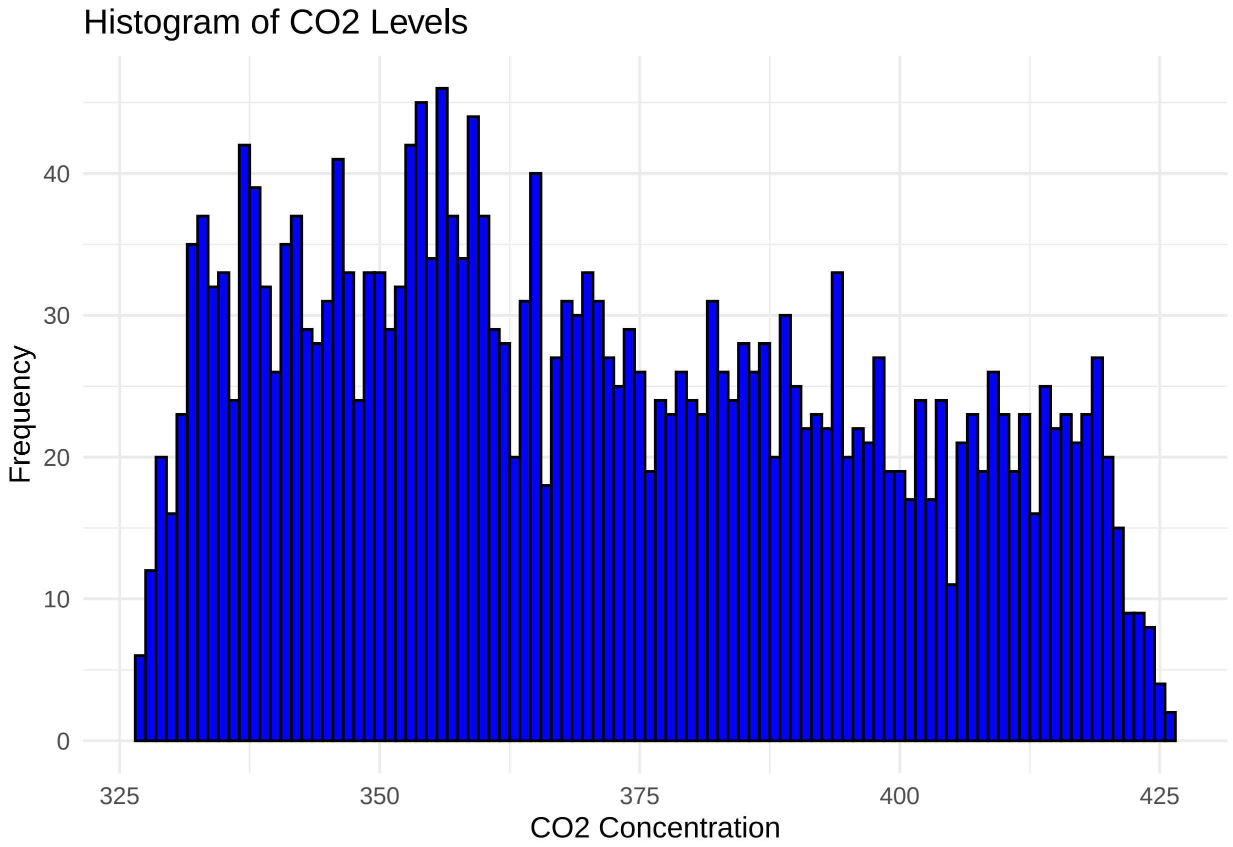
The plot shows a strong positive autocorrelation at all lags up to 35, and all are above the significance level, indicating a very persistent time series with a strong seasonal or cyclic pattern. This might suggest that the time series data of CO2 concentrations have a consistent pattern that repeats over time, with no significant decay in correlation as the lag increases.

```
pacf(co2_present$average, na.action = na.pass)
```



The plot indicates that there is no almost no significant partial autocorrelation in the data at lags greater than zero. This could suggest that a simple autoregressive model may not be a good fit for the data.

```r
ggplot(co2_present, aes(x = average)) +
  geom_histogram(binwidth = 1, fill = 'blue', color = 'black') +
  theme_minimal() +
  labs(title = "Histogram of CO2 Levels", x = "CO2 Concentration", y = "Frequency")
```



The plot shows multuple peaks suggesting more of a multimodal distribution. This could imply that there are multiple common CO2 levels within the data, possibly reflecting different environmental conditions or measurement periods. The data appears to be right-skewed.

#(1 point) Task 2b: Compare linear model forecasts against realized CO2 Descriptively compare realized atmospheric CO2 levels to those predicted by your forecast from a linear time model in 1997 (i.e. "Task 2a"). (You do not need to run any formal tests for this task.)

#(1 point) Task 3b: Compare ARIMA models forecasts against realized CO2 Descriptively compare realized atmospheric CO2 levels to those predicted by your forecast from the ARIMA model that you fitted in 1997 (i.e. "Task 3a"). Describe how the Keeling Curve evolved from 1997 to the present.

#(3 points) Task 4b: Evaluate the performance of 1997 linear and ARIMA models In 1997 you made predictions about the first time that CO2 would cross 420 ppm. How close were your models to the truth? After reflecting on your performance on this threshold-prediction task, continue to use the weekly data to generate a month-average series from 1997 to the present, and compare the overall forecasting performance of your models from Parts 2a and 3b over the entire period. (You should conduct formal tests for this task.)

```
co2_present <- read_csv(weekly_co2_url, skip = header_end_index, show_col_types = FALSE)

co2_present <- co2_present[co2_present$average != -999.99, ]

co2_monthly <- co2_present %>%
  group_by(year, month) %>%
  summarise(average = mean(average, na.rm = TRUE),
            .groups = "drop")

co2_monthly <- co2_monthly %>%
  mutate(Month = yearmonth(make_date(year, month, day = 1))) %>%
  select(-year, -month)

co2_monthly_present <- as_tsibble(co2_monthly, index = Month)

co2_monthly_present
```

```
## # A tsibble: 598 x 2 [1M]
##    average    Month
##      <dbl>    <mth>
##  1    333. 1974 May
##  2    332. 1974 Jun
##  3    331. 1974 Jul
##  4    329. 1974 Aug
##  5    327. 1974 Sep
##  6    327. 1974 Oct
##  7    328. 1974 Nov
##  8    330. 1974 Dec
##  9    331. 1975 Jan
## 10    332. 1975 Feb
## # i 588 more rows
```

```
scan_gaps(co2_monthly_present)
```

```
## # A tsibble: 1 x 1 [1M]
##      Month
##      <mth>
## 1 1975 Dec
```

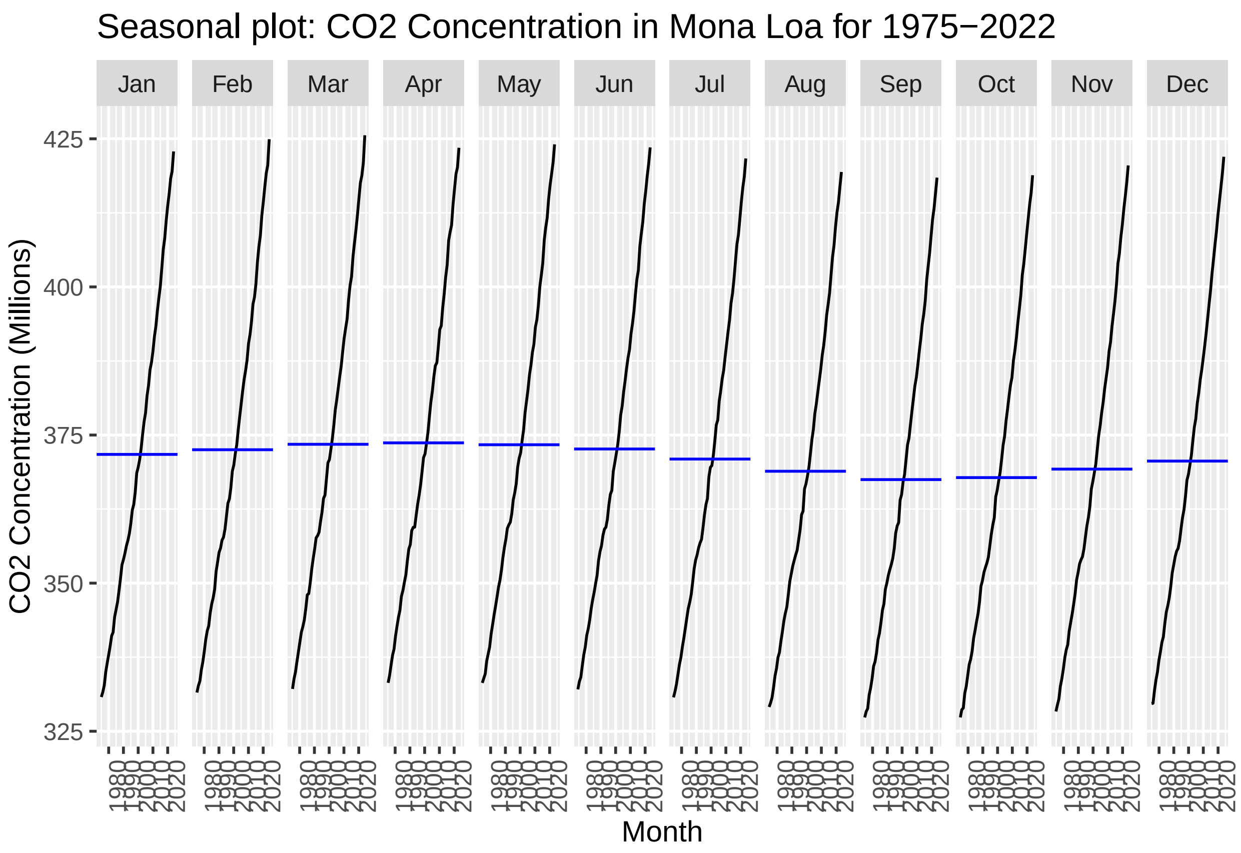
```
co2_monthly_present_gap_filled <- co2_monthly_present |>
  fill_gaps(average = 329.72)

scan_gaps(co2_monthly_present_gap_filled)
```

```
## # A tsibble: 0 x 1 [?]
## # i 1 variable: Month <mth>
```

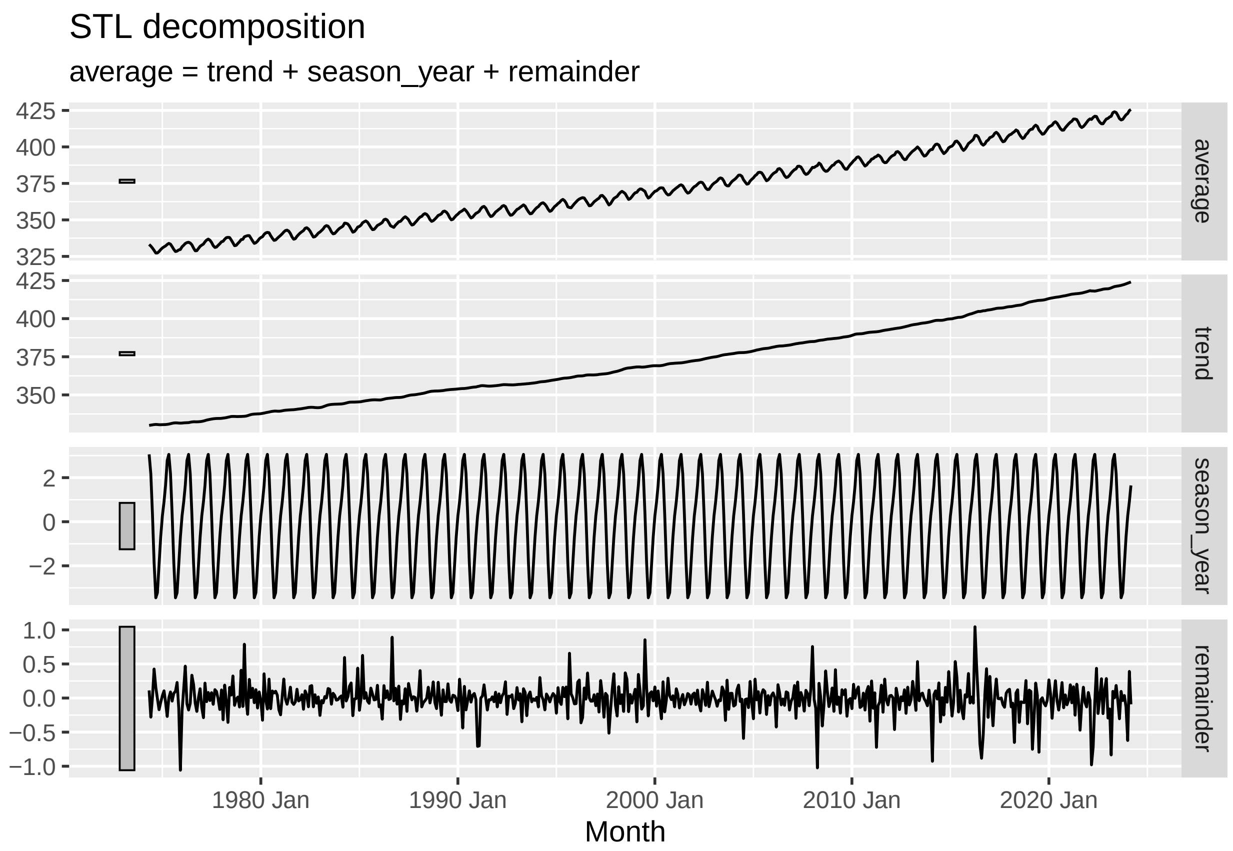
We identified a gap in the data series on December, 1975. Given that it was the only gap in the series and the closest data point observed is on November 30, 1975, we have decided to fill the gap with November 30, 1975 data point.

#(4 points) Task 5b: Train best models on present data Seasonally adjust the weekly NOAA data, and split both seasonally-adjusted (SA) and non-seasonally adjusted (NSA) series into training and test sets, using the last two years of observations as the test sets. For both SA and NSA series, fit ARIMA models using all appropriate steps. Measure and discuss how your models perform in-sample and (psuedo-) out-of-sample, comparing candidate models and explaining your choice. In addition, fit a polynomial time-trend model to the seasonally-adjusted series and compare its performance to that of your ARIMA model.

```
seasonal.plot <- co2_monthly_present_gap_filled |>
  gg_subseries(average) +
  labs(y = "CO2 Concentration (Millions)", x = "Month",
       title = "Seasonal plot: CO2 Concentration in Mona Loa for 1975-2022")

seasonal.plot
```

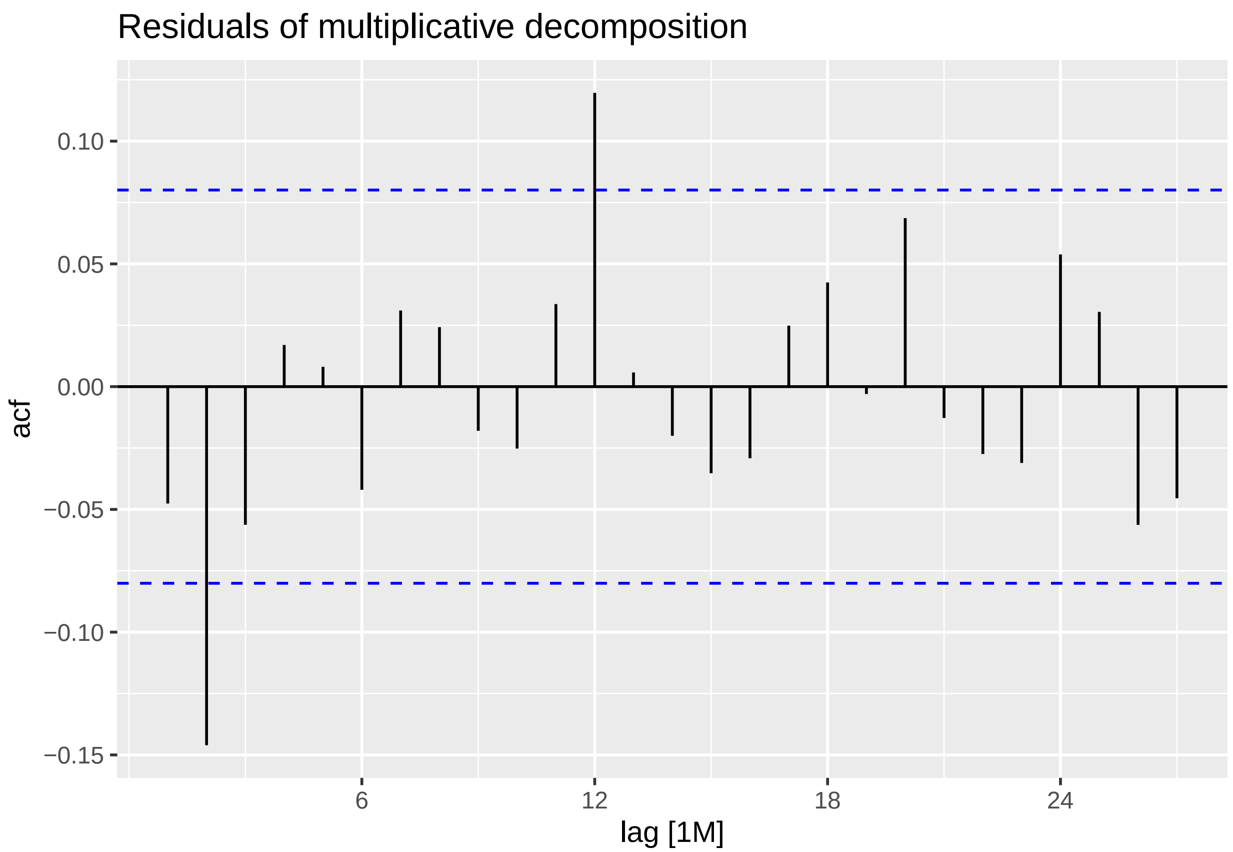


```
STL.model <- co2_monthly_present_gap_filled |>
  model(
    STL(average ~ trend(window = 6) + season(window = "periodic"),
    robust = TRUE))
STL.model |>
  components() |>
  autoplot()
```



As we have observed in our EDA section, we believe the CO2 levels have both overall upward trend and very possible seasonal trend. We used STL decomposition to decompose the data into 3 components: 1) upward trend using 6 months window, 2) seasonal trend which is observed as yearly trend, and 3) remainder, which the mean is observed close to zero, and fluctuates around reasonably bounded variance. We will proceed with further investigation to check for stationary on the remainder.

```
STL.model.resids <- components(STL.model) |>
ACF(remainder) |>
autoplot() + labs(title = "Residuals of multiplicative decomposition")

STL.model.resids
```



```
Box.test(components(STL.model)$remainder, lag = 10, type = "Ljung-Box")
```

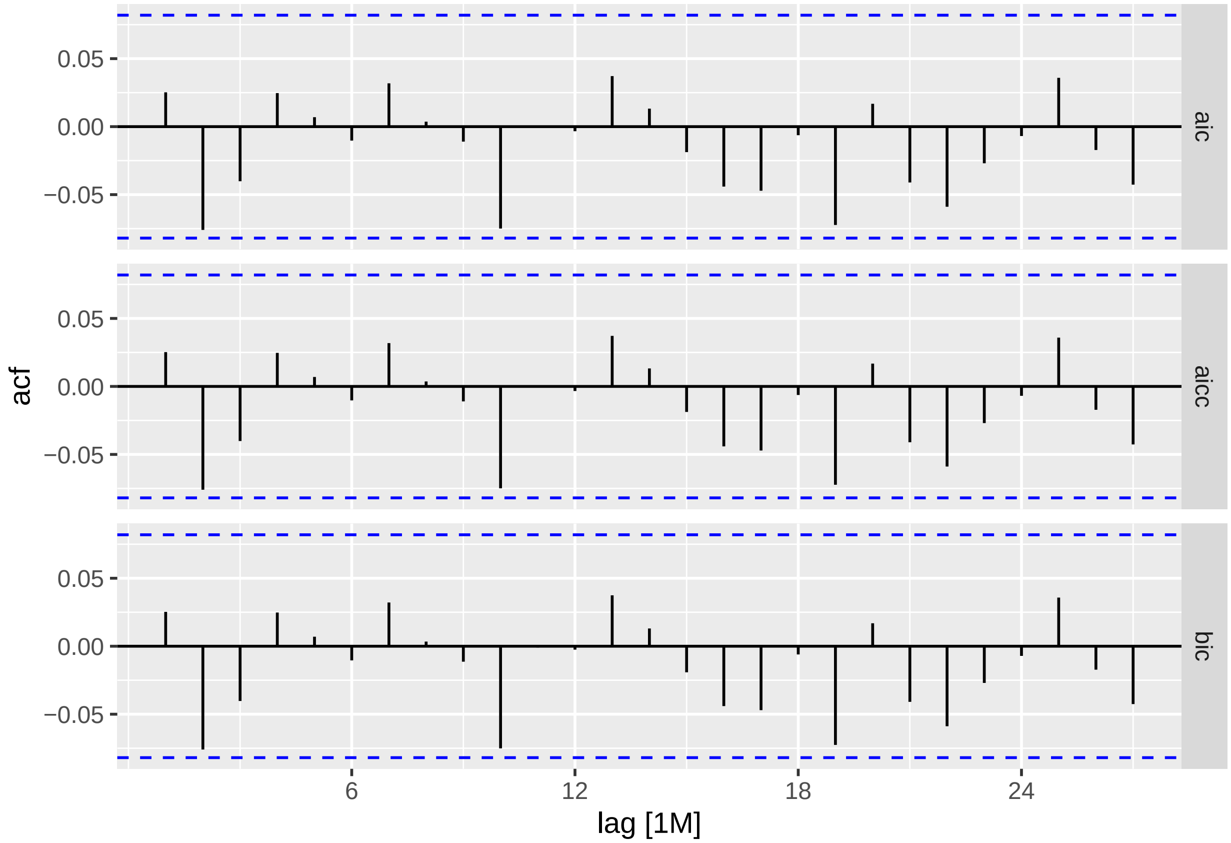
```
##
##  Box-Ljung test
##
## data:  components(STL.model)$remainder
## X-squared = 18.955, df = 10, p-value = 0.04084
```

From the residual plots and the Box-Ljung test above, it looks like although the residuals of the decomposition are stationary, they do not appear to be completely white noise. This means that while the decomposition method eliminates the deterministic components from this specific time series, there are some correlation remains in the data. As a result, we moved on to fit our data using ARIMA models that separately optimize for AIC, AICc and BIC.

```
co2_monthly_present_gap_filled.train <- co2_monthly_present_gap_filled  |>
  filter(year(Month) < 2022)

co2_monthly_present_gap_filled.test <- co2_monthly_present_gap_filled  |>
  filter(year(Month) >= 2022)

models <- co2_monthly_present_gap_filled.train |>
  model(aic = ARIMA(average ~ pdq(0:10,0:2,0:10)+PDQ(0:3,0:1,0:3, period=12), ic="aic", stepwise=F, gree
  aicc = ARIMA(average ~ pdq(0:10,0:2,0:10)+PDQ(0:3,0:1,0:3, period=12), ic="aicc", stepwise=F, greedy=
  bic = ARIMA(average ~ pdq(0:10,0:2,0:10)+PDQ(0:3,0:1,0:3, period=12), ic="bic", stepwise=F, greedy=F)
```

```
## Warning in sqrt(diag(best$var.coef)): NaNs produced
```

```
models |>
 augment() |>
 ACF(.resid)|>
 autoplot()
```



According to the above, the residuals appear to be close to white noise. There is no significant lags, but with some seasonal pattern that suggests we should run a statistical test on the residuals from the models to see if they are randomly distributed i.e. are white noise, which is what we want for a good model fit, or if they appear to have some serial correlation over time and violate the assumptions for a stationary time series fit.

```
models |>
  pivot_longer(everything(), names_to = "Model name", values_to = "SARIMA Model")
```

```
## # A mable: 3 x 2
## # Key:     Model name [3]
##   ‘Model name‘            ‘SARIMA Model‘
##   <chr>                          <model>
## 1 aic         <ARIMA(0,1,1)(2,1,3)[12]>
## 2 aicc        <ARIMA(0,1,1)(2,1,3)[12]>
## 3 bic         <ARIMA(0,1,1)(2,1,2)[12]>
```

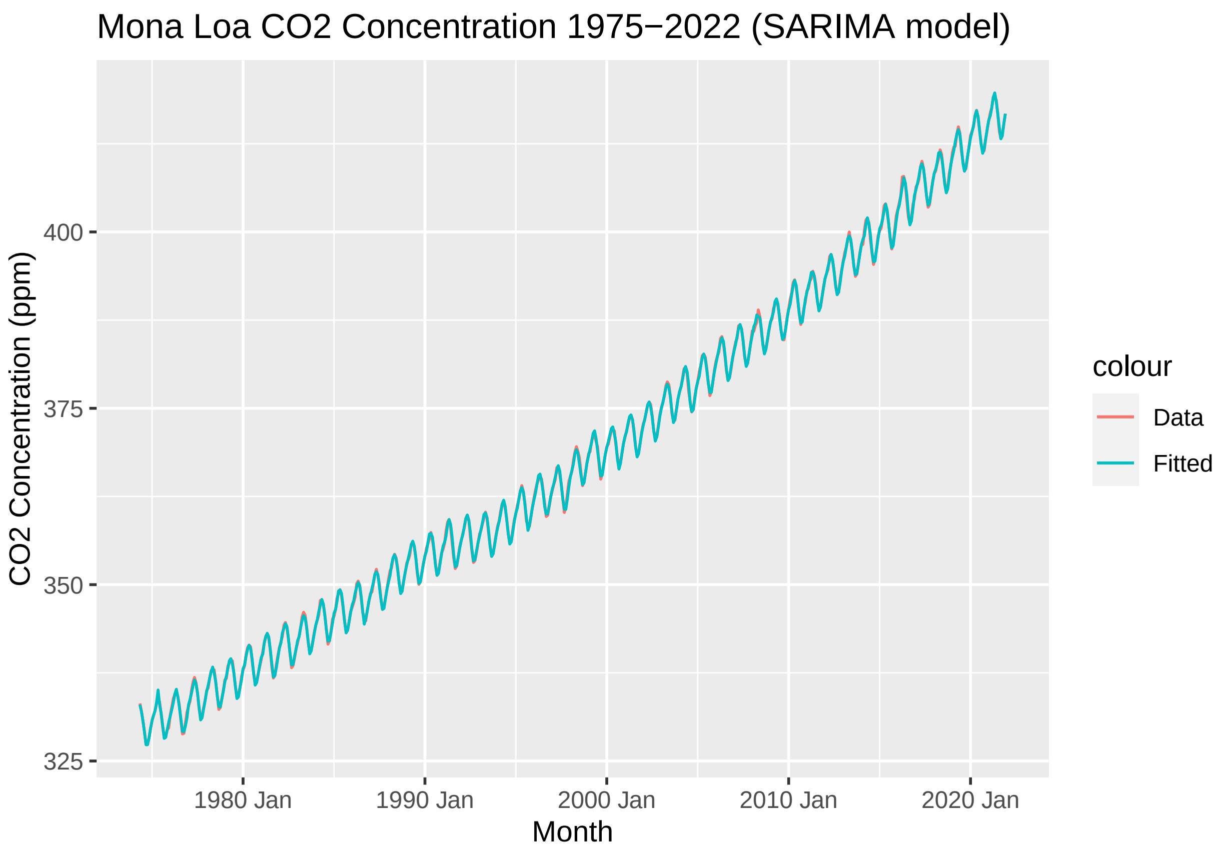
```
models |>
  report()
```

```
## Warning in report.mdl_df(models): Model reporting is only supported for
## individual models, so a glance will be shown. To see the report for a specific
## model, use ‘select()‘ and ‘filter()‘ to identify a single model.
```

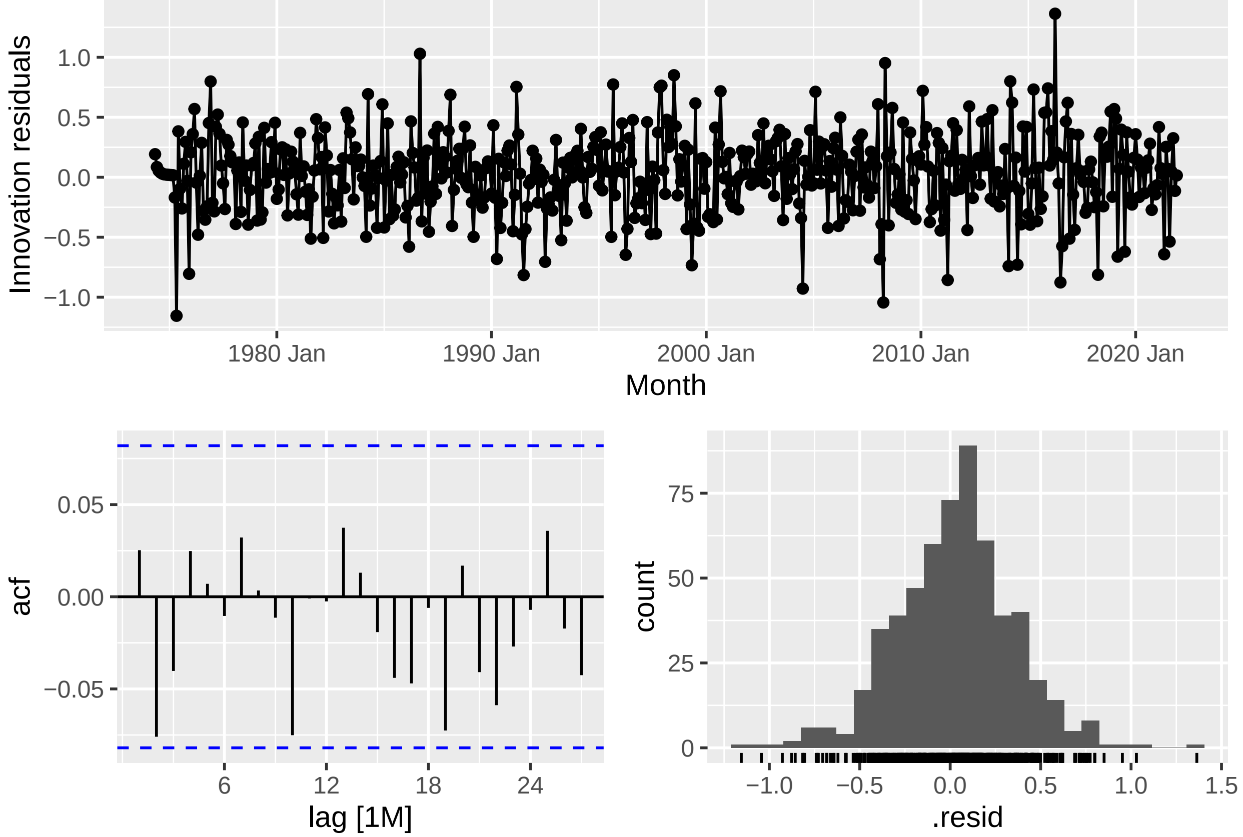
```
## # A tibble: 3 x 8
##   .model sigma2 log_lik   AIC  AICc   BIC ar_roots    ma_roots
##   <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl> <list>      <list>
## 1 aic     0.109   -172.  358.  358.  388. <cpl [24]> <cpl [37]>
## 2 aicc    0.109   -172.  358.  358.  388. <cpl [24]> <cpl [37]>
## 3 bic     0.109   -172.  356.  356.  382. <cpl [24]> <cpl [25]>
```

As depicted above, there is no major difference in the underlying SARIMA model, all elected SARIMA(0,1,1)(2,1,3)[12] amongst the aic, aicc, and bic models. We have decided to use BIC model since it produced the minimum IC scores across the three metrics.

```
SARIMA.model <- models |>
  select(.model=bic)
SARIMA.model |>
  augment() |>
  ggplot(aes(x = Month)) +
  geom_line(aes(y = average, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted")) +
  labs(x = "Month", y = "CO2 Concentration (ppm)",
       title = "Mona Loa CO2 Concentration 1975-2022 (SARIMA model)")
```



```
SARIMA.model |>
  gg_tsresiduals()
```

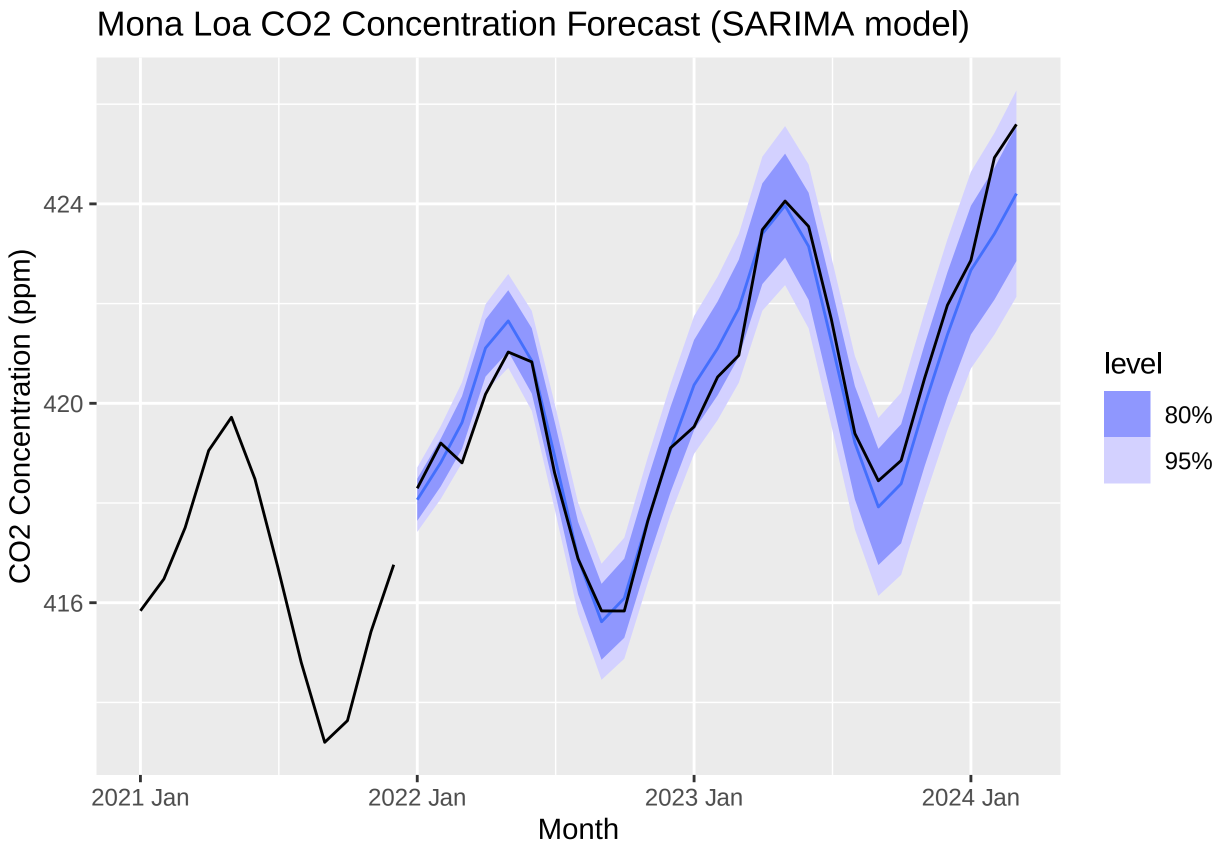


```
models |>
  augment() |>
  filter(.model=="bic") |>
  select(.resid) %>%
  as.ts() %>%
  Box.test(., lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  .
## X-squared = 9.0508, df = 10, p-value = 0.5273
```

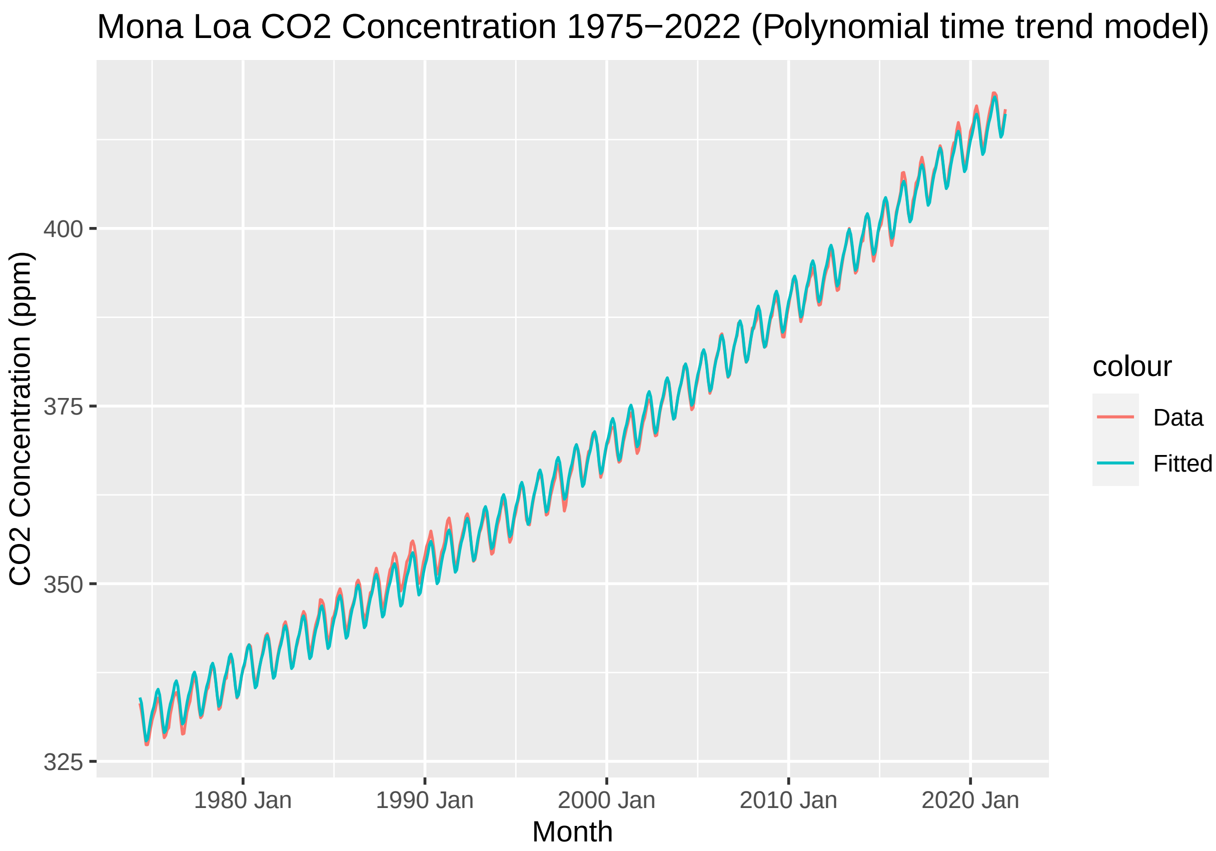
From the residual plots and the Box-Ljung test above, it looks like although the residuals of the SARIMA model are stationary and appear to be white noise. As a result, we decided to use the SARIMA model to forecast using our test data.

```
SARIMA.forecast <- SARIMA.model |>
  forecast(co2_monthly_present_gap_filled.test)

SARIMA.forecast  |>
  autoplot() +
  autolayer(co2_monthly_present_gap_filled.test) +
  geom_line(data=SARIMA.model |> augment() |>
  filter(year(Month) > 2020), aes(Month, .fitted)) +
  labs(title = "Mona Loa CO2 Concentration Forecast (SARIMA model)", x = "Month", y = "CO2 Concentratio
## Plot variable not specified, automatically selected `.vars = average`
```



```
sarima.accuracy <- accuracy(SARIMA.forecast, co2_monthly_present_gap_filled.test)
```

```
polynomial_time_trend.model <- co2_monthly_present_gap_filled.train |>
 model(trend_model = TSLM(average ~ trend() + I(trend()^2) + season()))

polynomial_time_trend.model |>
  augment() |>
  ggplot(aes(x = Month)) +
  geom_line(aes(y = average, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted")) +
  labs(x = "Month", y = "CO2 Concentration (ppm)",
       title = "Mona Loa CO2 Concentration 1975-2022 (Polynomial time trend model)")
```

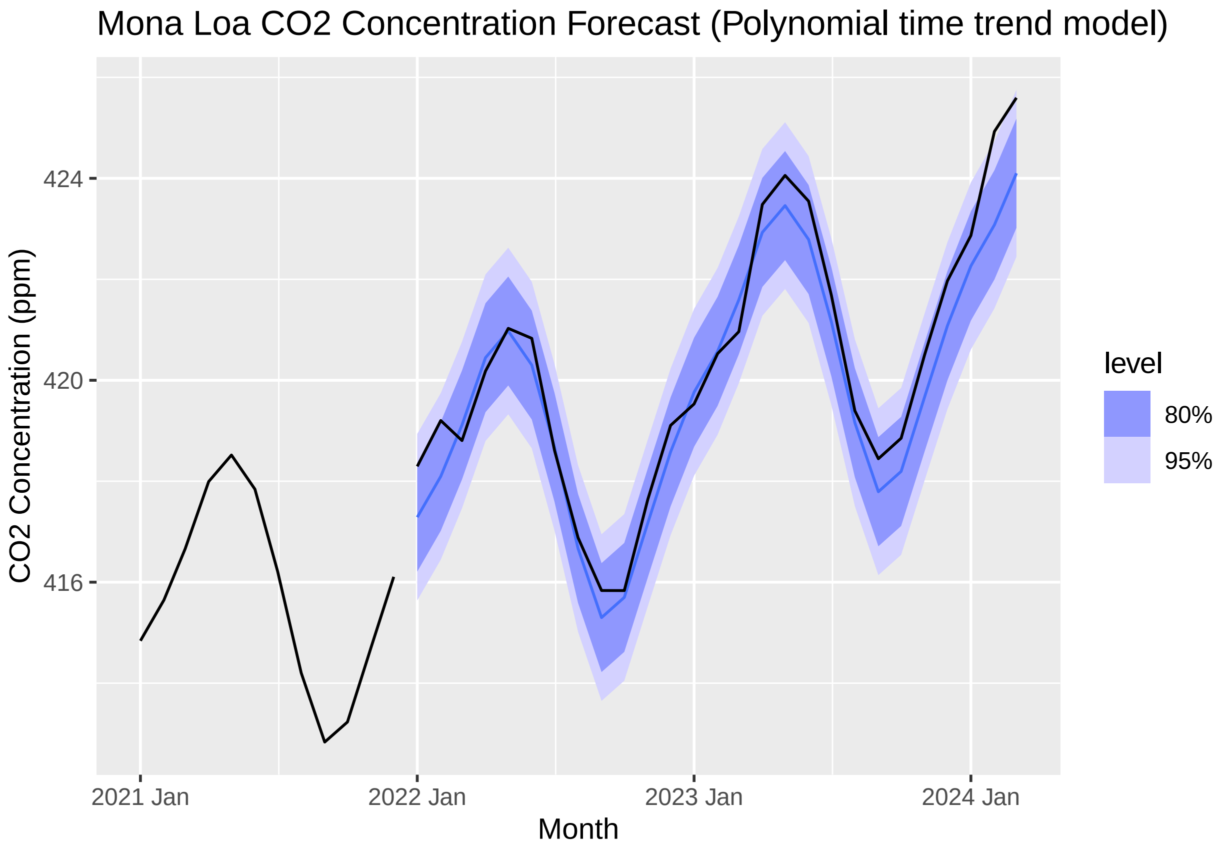


```
polynomial_time_trend.model |>
report()
```

```
## Series: average
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2809 -0.6462 -0.1327  0.6417  2.3169
##
## Coefficients:
##                  Estimate Std. Error  t value Pr(>|t|)
## (Intercept)     3.310e+02  1.560e-01 2122.506  < 2e-16 ***
## trend()         9.600e-02  8.391e-04  114.406  < 2e-16 ***
## I(trend()^2)    9.521e-05  1.418e-06   67.141  < 2e-16 ***
## season()year2   6.036e-01  1.706e-01    3.539 0.000435 ***
## season()year3   1.416e+00  1.706e-01    8.301 7.84e-16 ***
## season()year4   2.544e+00  1.706e-01   14.913  < 2e-16 ***
## season()year5   2.867e+00  1.697e-01   16.894  < 2e-16 ***
## season()year6   1.986e+00  1.697e-01   11.703  < 2e-16 ***
## season()year7   1.488e-01  1.697e-01    0.877 0.380904
## season()year8  -2.060e+00  1.697e-01  -12.138  < 2e-16 ***
## season()year9  -3.636e+00  1.697e-01  -21.424  < 2e-16 ***
## season()year10 -3.442e+00  1.697e-01  -20.284  < 2e-16 ***
## season()year11 -2.174e+00  1.697e-01  -12.814  < 2e-16 ***
## season()year12 -9.759e-01  1.697e-01   -5.751 1.46e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8269 on 558 degrees of freedom
## Multiple R-squared: 0.9989,  Adjusted R-squared: 0.9989
## F-statistic: 4.037e+04 on 13 and 558 DF, p-value: < 2.22e-16
```

```
poly.forecast <- polynomial_time_trend.model |>
  forecast(co2_monthly_present_gap_filled.test)

poly.forecast  |>
  autoplot() +
  autolayer(co2_monthly_present_gap_filled.test) +
  geom_line(data=polynomial_time_trend.model |> augment() |>
  filter(year(Month) > 2020), aes(Month, .fitted)) +
  labs(title = "Mona Loa CO2 Concentration Forecast (Polynomial time trend model)", x = "Month", y = "C(
```

```
## Plot variable not specified, automatically selected `.vars = average`
```



```
poly.accuracy <- accuracy(poly.forecast, co2_monthly_present_gap_filled.test)
```

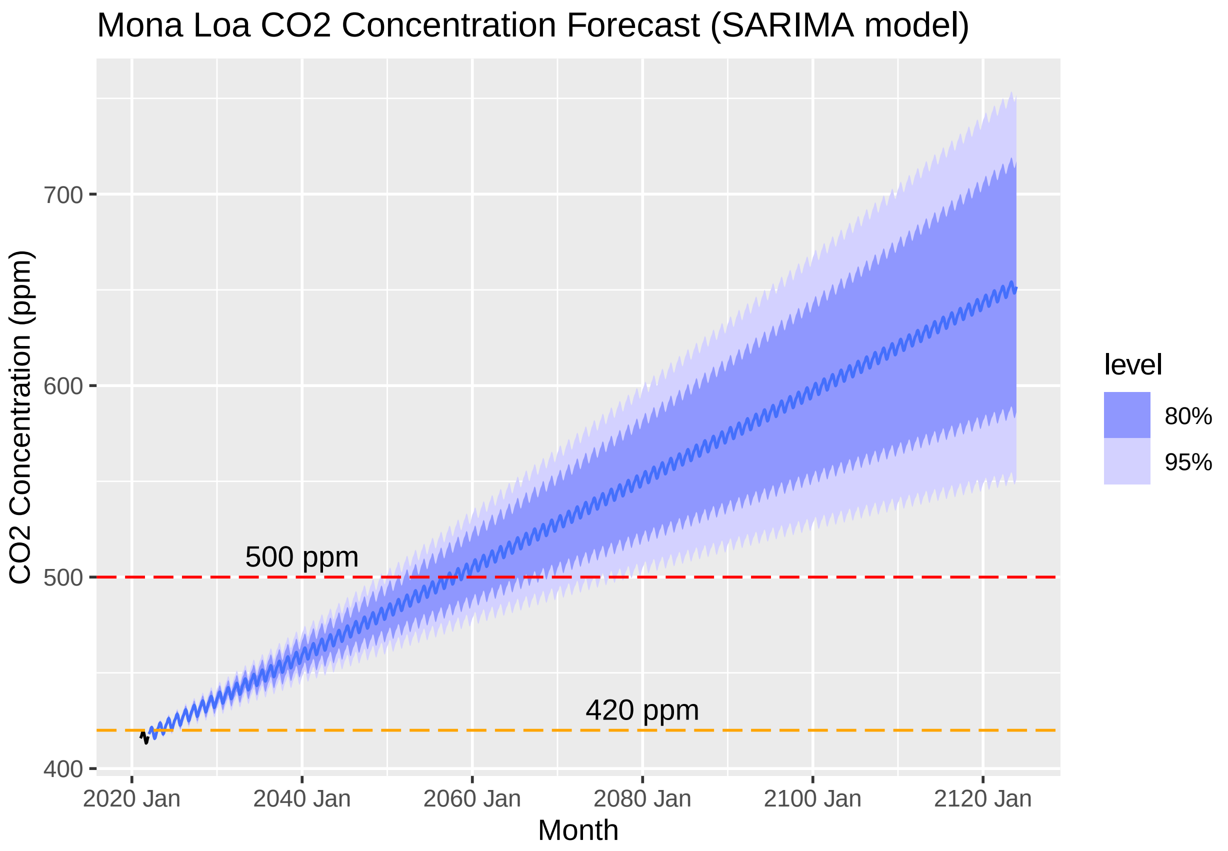
```
accuracy_table <- rbind(sarima.accuracy, poly.accuracy)
accuracy_table
```

```
## # A tibble: 2 x 10
##   .model      .type     ME  RMSE   MAE    MPE  MAPE  MASE RMSSE  ACF1
##   <chr>       <chr>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 .model      Test  0.0718 0.611 0.469 0.0167 0.111   NaN   NaN 0.574
## 2 trend_model Test  0.470  0.718 0.584 0.111  0.139   NaN   NaN 0.437
```

From the above accuracy table, it is pretty evident that SARIMA model has outperformed polynomial time-trend model in terms of minimize the forecast error in the training dataset.

#(3 points) Task Part 6b: How bad could it get? With the non-seasonally adjusted data series, generate predictions for when atmospheric CO2 is expected to be at 420 ppm and 500 ppm levels for the first and final times (consider prediction intervals as well as point estimates in your answer). Generate a prediction for atmospheric CO2 levels in the year 2122. How confident are you that these will be accurate predictions?

```
forecase.size = (2122-2020)*12
SARIMA.forecast.2122 <- SARIMA.model |>
  forecast(h=forecase.size)

SARIMA.forecast.2122  |>
  autoplot() +
  geom_line(data=SARIMA.model |> augment() |>
  filter(year(Month) > 2020), aes(Month, .fitted)) +
  geom_hline(yintercept=500, linetype='longdash', col = 'red')+
  annotate("text", x = yearmonth("2040-01"), y = 500, label = "500 ppm", vjust = -0.5) +
  geom_hline(yintercept=420, linetype='longdash', col = 'orange')+
  annotate("text", x = yearmonth("2080-01"), y = 420, label = "420 ppm", vjust = -0.5) +
  labs(title = "Mona Loa CO2 Concentration Forecast (SARIMA model)", x = "Month", y = "CO2 Concentratio
```



We can see that using our SARIMA model the forecasts also have a trend and seasonal movement and fluctuations increase overtime. But these forecasts will get very inaccurate as we move beyond at best 5 year forecasts of the model, as the confidence intervals of the model prediction begin to open up very widely due to the inherent model restriction and the unpredictability of the future. We can be pretty confident to say that we will hit 420 ppm CO2 level in 2023-2025, less confident about 500 ppm CO2 level as the model suggest it can be as early as 2045 or as late as 2060 or beyond.