Statistical Methods for Discrete Response, Time Series, and Panel Data (W271): Lab 2

Nathan Martinez, Haile Bizunehe, Hannah George, and Meera Sharma

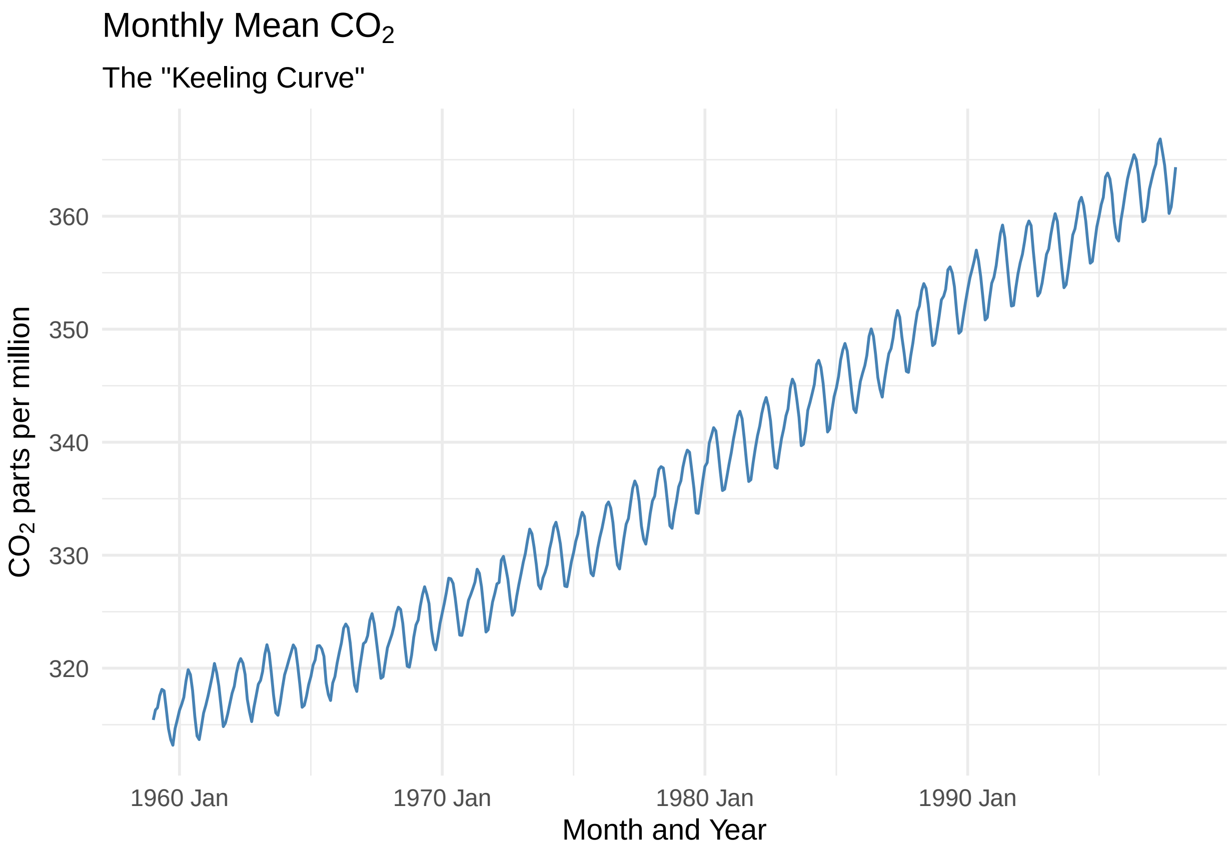
2022-11-01

Contents

1 The Keeling Curve

In the 1950s, the geochemist Charles David Keeling observed a seasonal pattern in the amount of carbon dioxide present in air samples collected over the course of several years. He was able to attribute this pattern to the variation in global rates of photosynthesis throughout the year, caused by the difference in land area and vegetation cover between the Earth's northern and southern hemispheres.

In 1958 Keeling began continuous monitoring of atmospheric carbon dioxide concentrations from the Mauna Loa Observatory in Hawaii and soon observed a trend increase carbon dioxide levels in addition to the seasonal cycle. He was able to attribute this trend increase to growth in global rates of fossil fuel combustion. This trend has continued to the present, and is known as the "Keeling Curve."



Monthly Mean CO_2
The "Keeling Curve"
CO_2 parts per million
360
350
340
330
320
1960 Jan
1970 Jan
1980 Jan
1990 Jan
Month and Year

2 Your Assignment

Your goal in this assignment is to produce a comprehensive analysis of the Mona Loa CO2 data that you will be read by an interested, supervising data scientist. Rather than this being a final report, you might think of this as being a contribution to your laboratory. You and your group have been initially charged with the task of investigating the trends of global CO2, and told that if you find "anything interesting" that the team may invest more resources into assessing the question.

Because this is the scenario that you are responding to:

1. Your writing needs to be clear, well-reasoned, and concise. Your peers will be reading this, and you have a reputation to maintain.
2. Decisions that you make for your analysis need also be clear and well-reasoned. While the main narrative of your deliverable might only present the modeling choices that you determine are the most appropriate, there might exist supporting materials that examine what the consequences of other choices would be. As a concrete example, if you determine that a series is an AR(1) process your main analysis might provide the results of the critical test that led you to that determination and the results of the rest of the analysis under AR(1) modeling choices. However, in an appendix or separate document that is linked in your main report, you might show what a MA model would have meant for your results instead.
3. Your code and repository are a part of the deliverable. If you were to make a clear argument that this is a question worth pursuing, but then when the team turned to continue the work they found a repository that was a jumble of coding idioms, version-ed or outdated files, and skeletons it would be a disappointment.

3 Report from the Point of View of 1997

For the first part of this task, suspend reality for a short period of time and conduct your analysis from the point of view of a data scientist doing their work in the early months of 1998. Do this by using data that is included in *every* R implementation, the `co2` dataset. This dataset is lazily loaded with every R instance, and is stored in an object called `co2`.

3.1 (3 points) Task 0a: Introduction

Introduce the question to your audience. Suppose that they *could* be interested in the question, but they don't have a deep background in the area. What is the question that you are addressing, why is it worth addressing, and what are you going to find at the completion of your analysis. Here are a few resource that you might use to start this motivation.

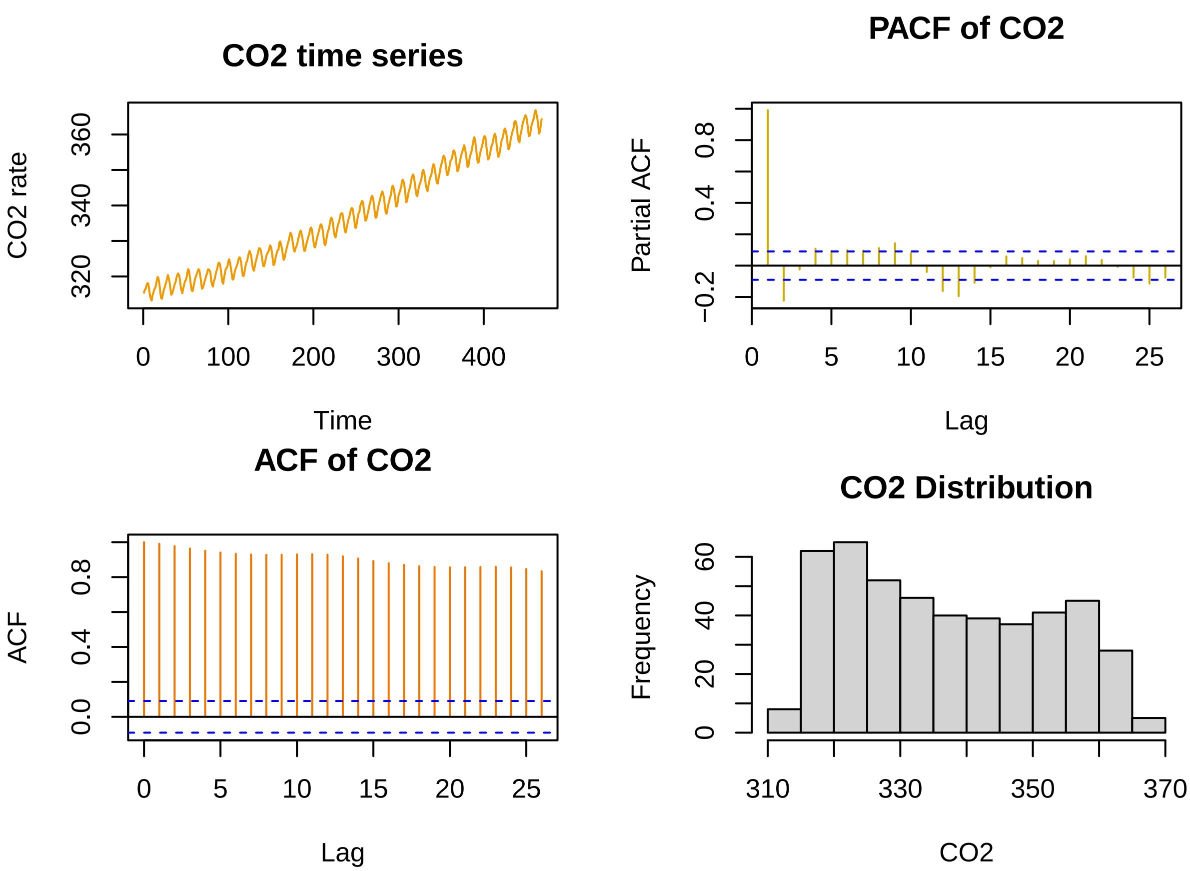
- Wikipedia
- First Publication
- Autobiography of Keeling

3.2 (3 points) Task 1a: CO2 data

Conduct a comprehensive Exploratory Data Analysis on the `co2` series. This should include (without being limited to) a description of how, where and why the data is generated, a thorough investigation of the trend, seasonal and irregular elements. Trends both in levels and growth rates should be discussed (consider expressing longer-run growth rates as annualized averages).

What you report in the deliverable should not be your own process of discovery, but rather a guided discussion that you have constructed so that your audience can come to an understanding as succinctly and successfully as possible. This means that figures should be thoughtfully constructed and what you learn from them should be discussed in text; to the extent that there is *any* raw output from your analysis, you should intend for people to read and interpret it, and you should write your own interpretation as well.

```r
co2 %>%
  as_tsibble() -> co2_ts

# Getting the overall CO2 values
value <- co2_ts$value

# Making four different plots for evaluation
par(mfrow=c(2,2), mar = c(4.1, 4.1, 3, 2))
plot(value, type = "l", col="orange2", main="CO2 time series",
     xlab="Time", ylab="CO2 rate")
pacf(value, col="gold3", main="PACF of CO2")
acf(value, col="darkorange2", main="ACF of CO2")
hist(value, main="CO2 Distribution",
     ylab = "Frequency", xlab="CO2")
```



, ,

3.3 (3 points) Task 2a: Linear time trend model

Fit a linear time trend model to the `co2` series, and examine the characteristics of the residuals. Compare this to a quadratic time trend model. Discuss whether a logarithmic transformation of the data would be appropriate. Fit a polynomial time trend model that incorporates seasonal dummy variables, and use this model to generate forecasts to the year 2020.

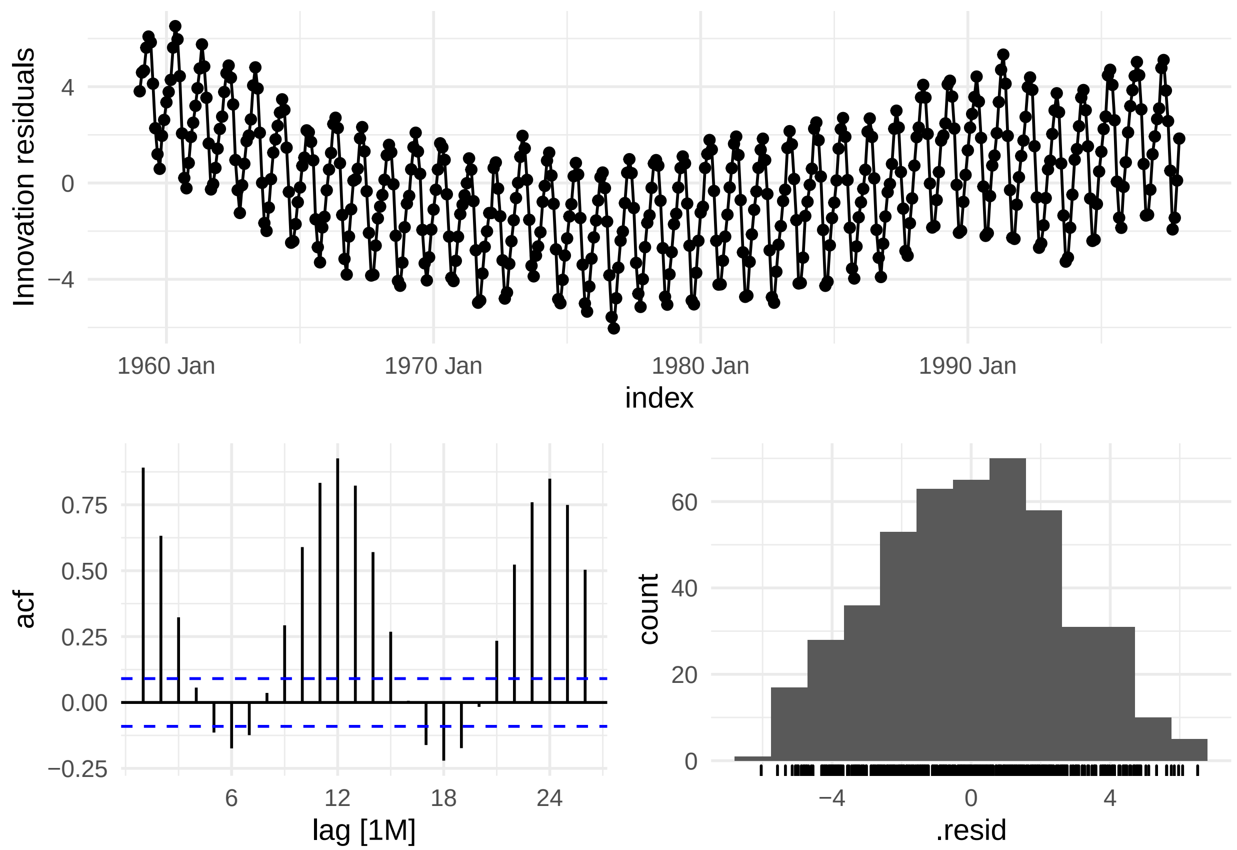
```r
as_tsibble(co2) -> co2_ts

co2_ts %>%
```

```
  model(TSLM(value ~ trend())) -> fit_co2_trend
report(fit_co2_trend)
```

```
## Series: value
## Model: TSLM
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -6.039885 -1.947575 -0.001671  1.911271  6.514852
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.115e+02  2.424e-01  1284.9   <2e-16 ***
## trend()     1.090e-01  8.958e-04   121.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.618 on 466 degrees of freedom
## Multiple R-squared: 0.9695,  Adjusted R-squared: 0.9694
## F-statistic: 1.479e+04 on 1 and 466 DF, p-value: < 2.22e-16
```

'There is an average upward trend of 0.11 CO2 level.'

```
fit_co2_trend %>%
  gg_tsresiduals()
```

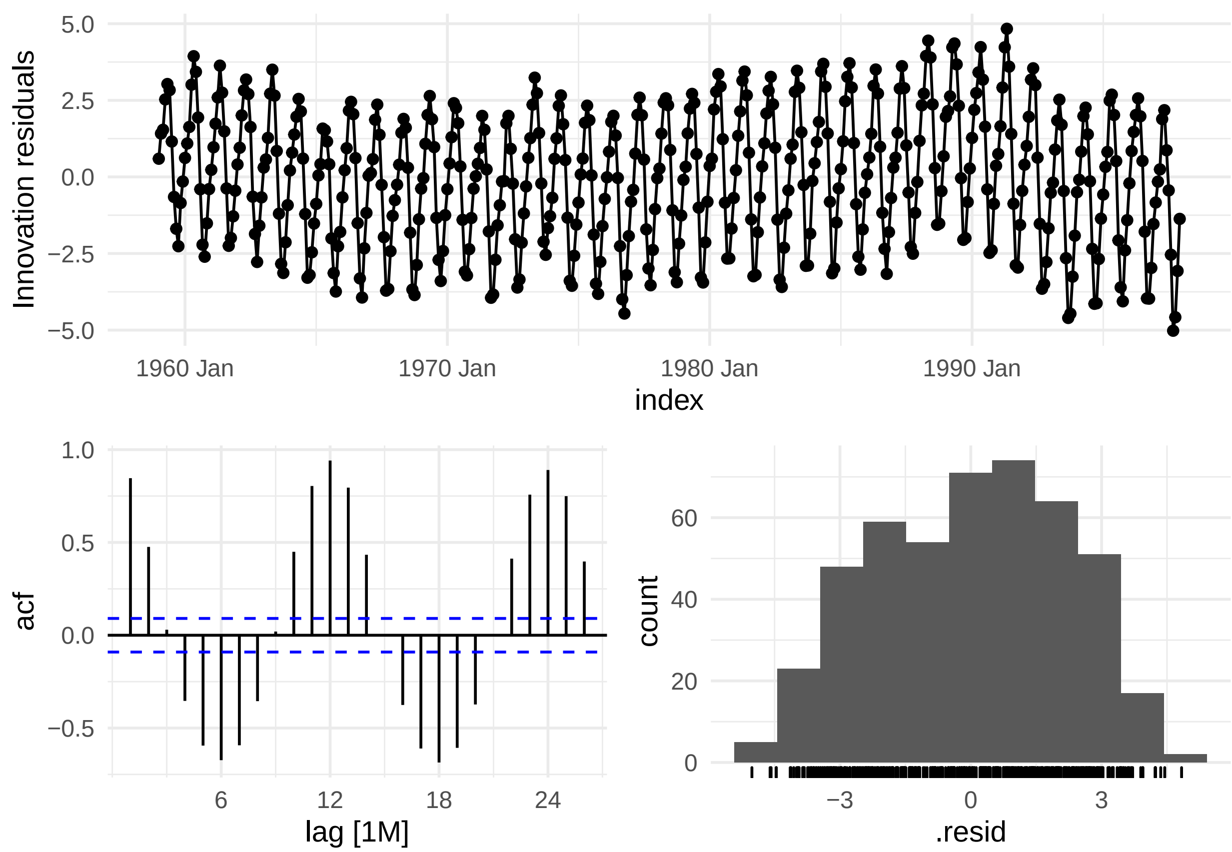


'Both in the scatter plot and the ACF, we see clearly that there is seasonality. There is also a curvy trend that we missed to be captured.'

```r
as_tsibble(co2) -> co2_ts

co2_ts %>%
  model(TSLM(value ~ trend()+ I(trend()^2))) -> fit_co2_quad_trend
report(fit_co2_quad_trend)
```

```
## Series: value
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0195 -1.7120  0.2144  1.7957  4.8345
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.148e+02  3.039e-01 1035.65   <2e-16 ***
## trend()      6.739e-02  2.993e-03   22.52   <2e-16 ***
## I(trend()^2) 8.862e-05  6.179e-06   14.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.182 on 465 degrees of freedom
## Multiple R-squared: 0.9788,  Adjusted R-squared: 0.9787
## F-statistic: 1.075e+04 on 2 and 465 DF, p-value: < 2.22e-16
```

```r
fit_co2_quad_trend %>%
  gg_tsresiduals()
```



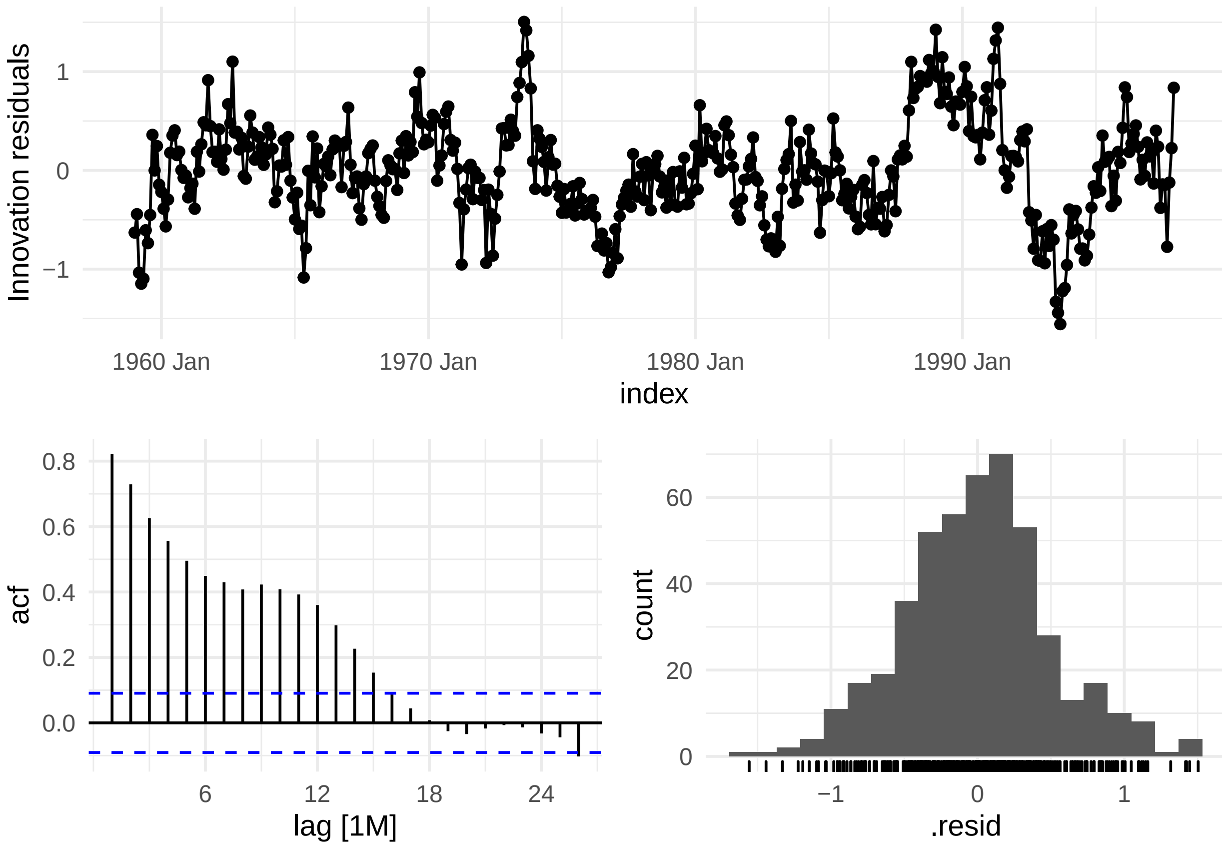
'The trend is better now but there is still an up and down trend that can be modeled using polynomial. We do not see a change on the variance so the log transformation won't be necessary.'

```
as_tsibble(co2) -> co2_ts

co2_ts %>%
  model(TSLM(value ~ trend() + I(trend()^2) + I(trend()^3) + season())) -> fit_co2_full_trend
report(fit_co2_full_trend)
```

```
## Series: value
## Model: TSLM
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.5573094 -0.3312054  0.0008042  0.2880086  1.5039635
##
## Coefficients:
##                  Estimate Std. Error  t value Pr(>|t|)
## (Intercept)     3.160e+02  1.210e-01 2611.629  < 2e-16 ***
## trend()         3.275e-02  1.740e-03   18.827  < 2e-16 ***
## I(trend()^2)    2.744e-04  8.614e-06   31.850  < 2e-16 ***
## I(trend()^3)   -2.640e-07  1.207e-08  -21.863  < 2e-16 ***
## season()year2   6.700e-01  1.145e-01    5.852 9.32e-09 ***
## season()year3   1.419e+00  1.145e-01   12.390  < 2e-16 ***
## season()year4   2.555e+00  1.145e-01   22.319  < 2e-16 ***
## season()year5   3.040e+00  1.145e-01   26.550  < 2e-16 ***
## season()year6   2.383e+00  1.145e-01   20.811  < 2e-16 ***
## season()year7   8.678e-01  1.145e-01    7.578 2.00e-13 ***
## season()year8  -1.194e+00  1.145e-01  -10.429  < 2e-16 ***
## season()year9  -3.013e+00  1.145e-01  -26.311  < 2e-16 ***
## season()year10 -3.191e+00  1.145e-01  -27.860  < 2e-16 ***
## season()year11 -1.996e+00  1.145e-01  -17.428  < 2e-16 ***
## season()year12 -8.738e-01  1.145e-01   -7.628 1.41e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5056 on 453 degrees of freedom
## Multiple R-squared: 0.9989,   Adjusted R-squared: 0.9989
## F-statistic: 2.92e+04 on 14 and 453 DF, p-value: < 2.22e-16
```

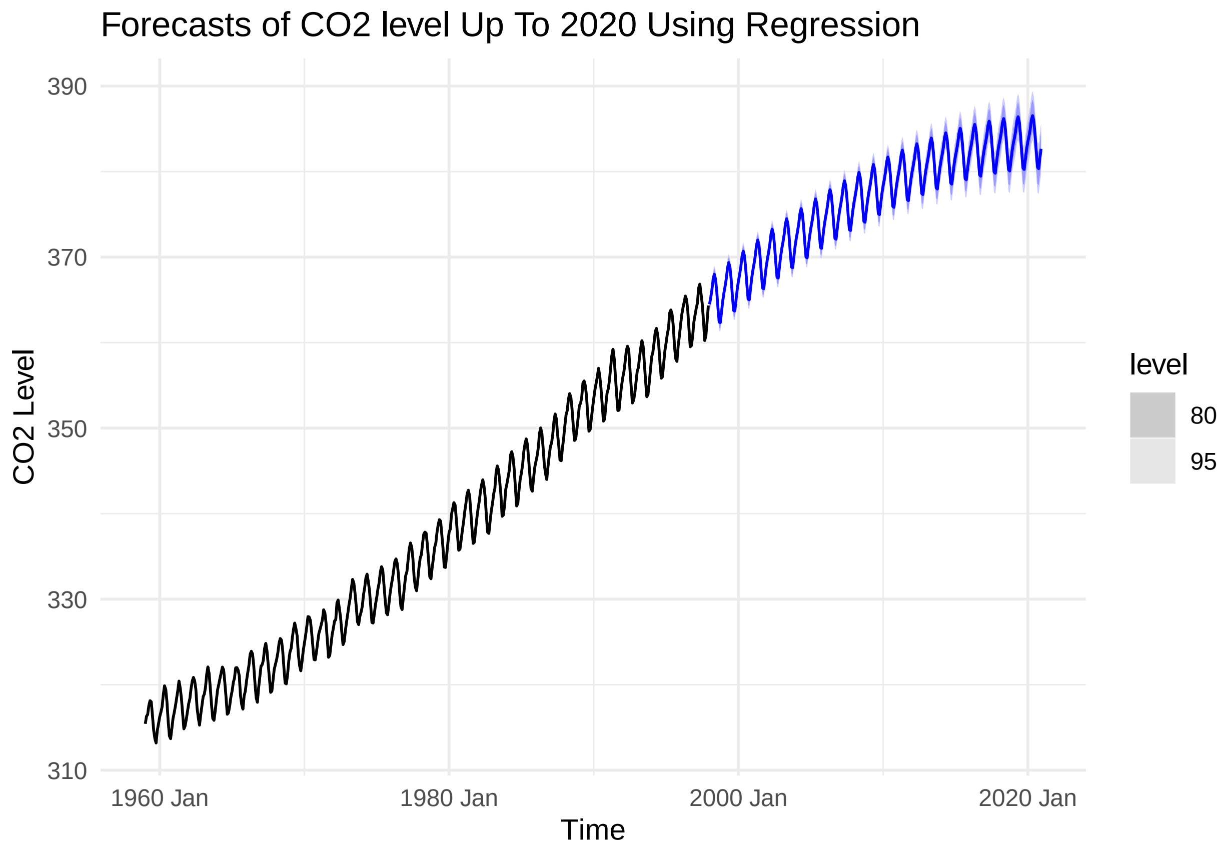
''

```
fit_co2_full_trend %>%
  gg_tsresiduals()
```



,,

```
fc_co2_2020 <- forecast(fit_co2_full_trend, h=276)

fc_co2_2020 %>%
  autoplot(co2_ts) +
  labs(
    title = "Forecasts of CO2 level Up To 2020 Using Regression",
    y = "CO2 Level", x = "Time"
  )
```



,,

3.4 (3 points) Task 3a: ARIMA times series model

Following all appropriate steps, choose an ARIMA model to fit to the series. Discuss the characteristics of your model and how you selected between alternative ARIMA specifications. Use your model (or models) to generate forecasts to the year 2022.

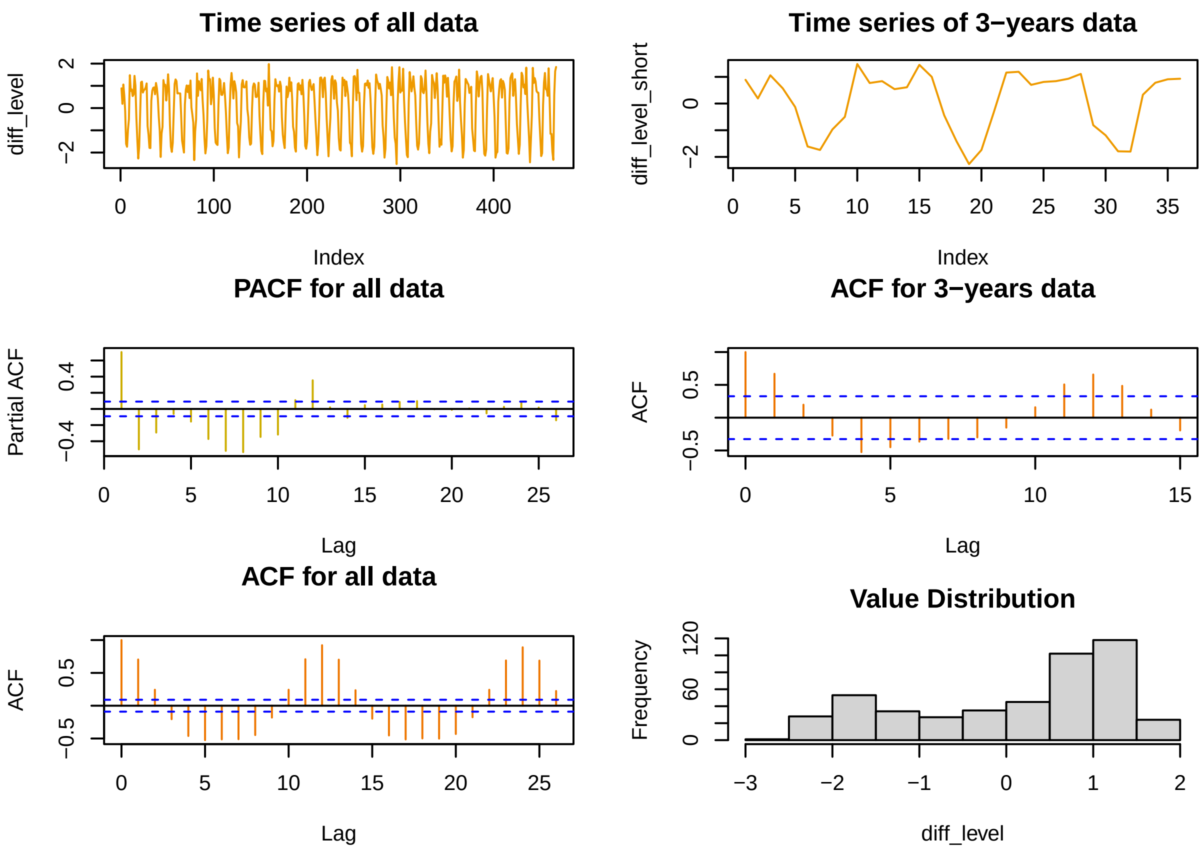
```
adf.test(co2_ts$value, alternative = "stationary", k = 12)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  co2_ts$value
## Dickey-Fuller = -2.1543, Lag order = 12, p-value = 0.5127
## alternative hypothesis: stationary
```

,,

```
# Getting the differenced values
diff_level <- diff(co2_ts$value, lag=1)
# Taking 3 years values to show zoom-in seasonal effect
diff_level_short <- diff_level[0:36]

# Making six different plots for evaluation
par(mfrow=c(3,2), mar = c(4.1, 4.1, 3, 2))
plot(diff_level, type = "l", col="orange2", main="Time series of all data")
plot(diff_level_short, type = "l", col="orange2",
     main="Time series of 3-years data")
```

```r
pacf(diff_level, col="gold3", main="PACF for all data")
acf(diff_level_short, col="darkorange2", main="ACF for 3-years data")
acf(diff_level, col="darkorange2", main="ACF for all data")
hist(diff_level, main="Value Distribution")
```



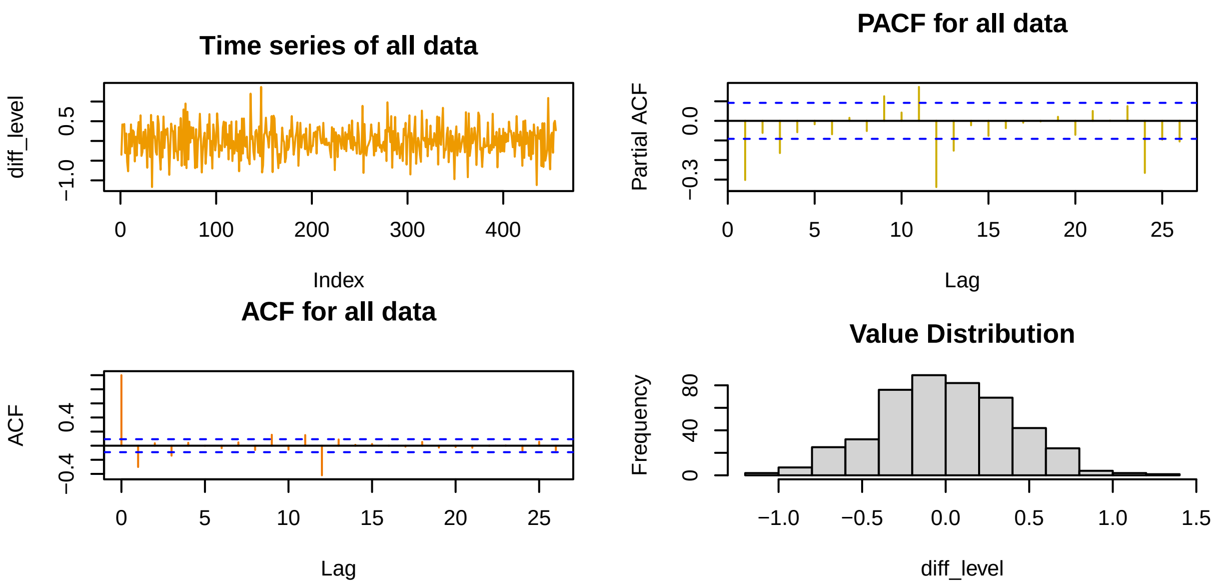
> ’’

```r
adf.test(diff(co2_ts$value),
         alternative = "stationary", k = 12)
```

```
## Warning in adf.test(diff(co2_ts$value), alternative = "stationary", k = 12): p-
## value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  diff(co2_ts$value)
## Dickey-Fuller = -5.778, Lag order = 12, p-value = 0.01
## alternative hypothesis: stationary
```

’’

```r
# Getting the differenced CO2 levels; first and seasonal differecing
diff_level <- diff(diff(co2_ts$value, lag=1), lag=12)

# Making four different plots for evaluation
par(mfrow=c(3,2), mar = c(4.1, 4.1, 3, 2))
plot(diff_level, type = "l", col="orange2", main="Time series of all data")
pacf(diff_level, col="gold3", main="PACF for all data")
acf(diff_level, col="darkorange2", main="ACF for all data")
hist(diff_level, main="Value Distribution")
```



,,

```
adf.test(diff(diff(co2_ts$value), lag=12),
         alternative = "stationary", k = 3)
```

```
## Warning in adf.test(diff(diff(co2_ts$value), lag = 12), alternative =
## "stationary", : p-value smaller than printed p-value
```

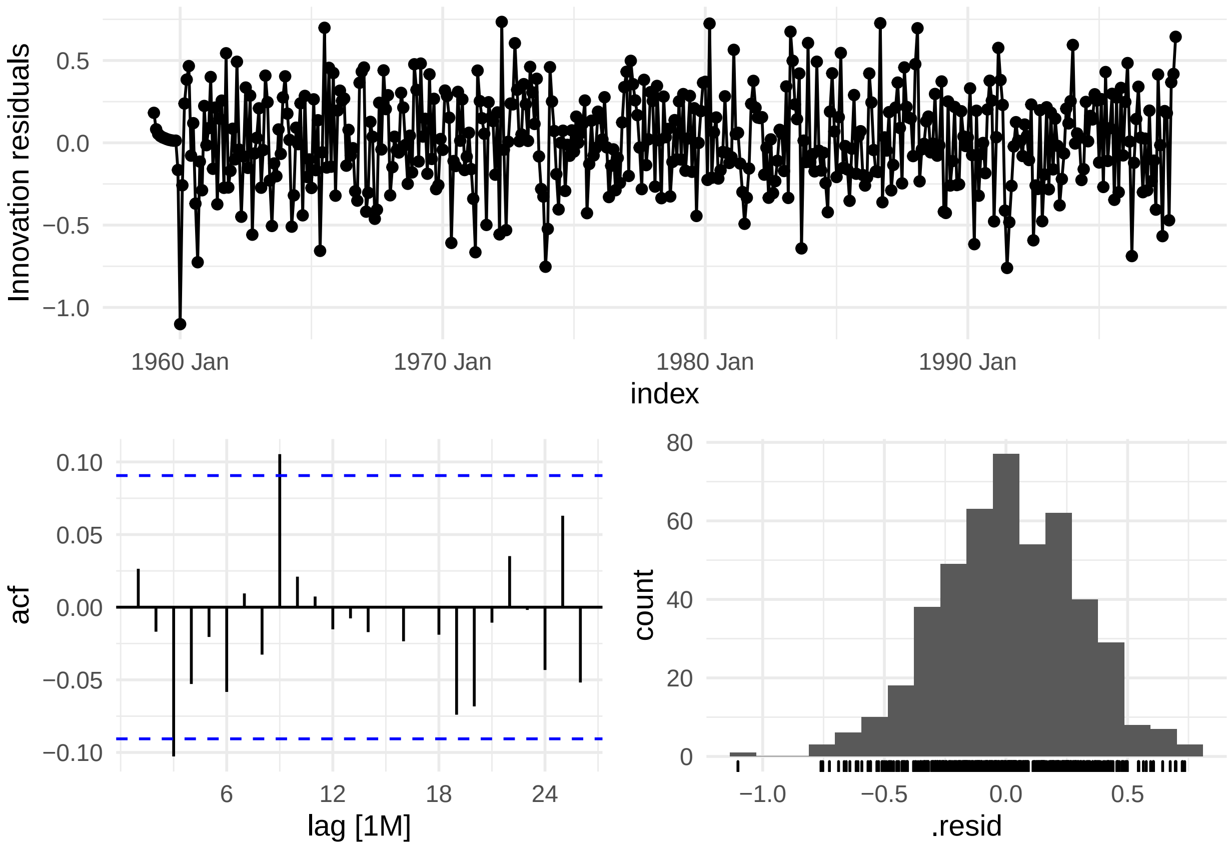
```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(diff(co2_ts$value), lag = 12)
## Dickey-Fuller = -13.69, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

,,

```
model.bic <- co2_ts %>%
  model(ARIMA(value ~ 0 + pdq(0:10, 1, 0:10) + PDQ(0:10, 1, 0:10),
                      ic="bic", stepwise=F, greedy=F))

model.bic %>%
  report()
```

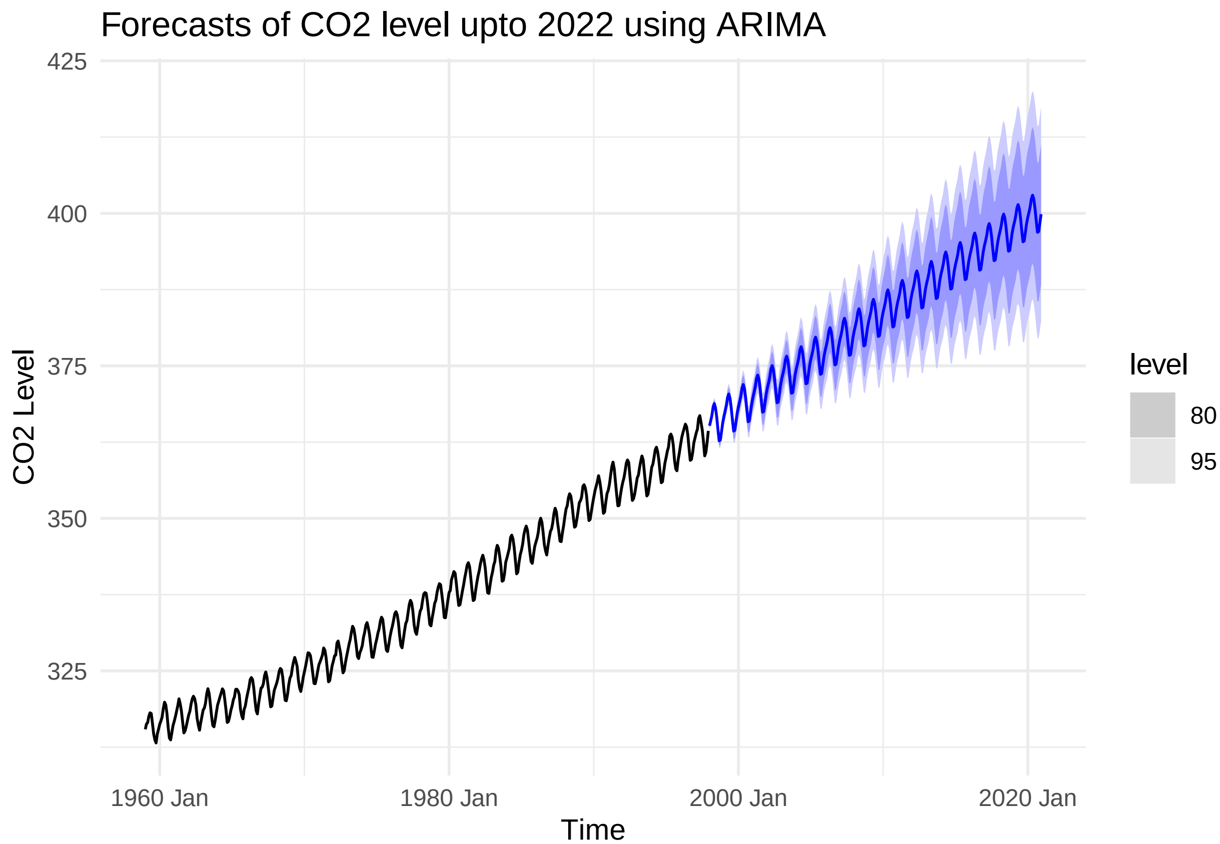
```
## Series: value
## Model: ARIMA(0,1,1)(1,1,2)[12]
##
## Coefficients:
##           ma1     sar1     sma1     sma2
##       -0.3482  -0.4986  -0.3155  -0.4641
## s.e.   0.0499   0.5282   0.5165   0.4367
##
## sigma^2 estimated as 0.08603:  log likelihood=-85.59
## AIC=181.18   AICc=181.32   BIC=201.78
```

```
model.bic %>%
  gg_tsresiduals()
```



,,

```
fc_co2_2022 <- forecast(model.bic, h=276)

fc_co2_2022 %>%
  autoplot(co2_ts) +
  labs(
    title = "Forecasts of CO2 level upto 2022 using ARIMA",
    y = "CO2 Level", x = "Time"
  )
```



, ,

3.5 (3 points) Task 4a: Forecast atmospheric CO2 growth

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 2100. How confident are you that these will be accurate predictions?

```
fc_100_years <- forecast(model.bic, h=1236) %>%
  hilo() %>%
  unpack_hilo(c(`80%`, `95%`))

fc_100_years %>%
  filter(.mean >= 420 & .mean < 421) -> fc_420

fc_100_years %>%
  filter(.mean >= 500 & .mean < 501) -> fc_500

# 420 PPM levels for the first and last time
fc_420_first <- head(fc_420, n =1)
fc_420_last  <- tail(fc_420, n =1)

fc_420_first_time  <- fc_420_first$index
fc_420_first_lower <- round(fc_420_first$`95%_lower`, 2)
fc_420_first_upper <- round(fc_420_first$`95%_upper`, 2)
fc_420_first_mean  <- round(fc_420_first$.mean, 2)
```

```
fc_420_last_time  <- fc_420_last$index
fc_420_last_lower <- round(fc_420_last$`95%_lower`, 2)
fc_420_last_upper <- round(fc_420_last$`95%_upper`, 2)
fc_420_last_mean  <- round(fc_420_last$.mean, 2)

# 500 PPM levels for the first and last time
fc_500_first <- head(fc_500, n =1)
fc_500_last  <- tail(fc_500, n =1)

fc_500_first_time  <- fc_500_first$index
fc_500_first_lower <- round(fc_500_first$`95%_lower`, 2)
fc_500_first_upper <- round(fc_500_first$`95%_upper`, 2)
fc_500_first_mean  <- round(fc_500_first$.mean, 2)

fc_500_last_time  <- fc_500_last$index
fc_500_last_lower <- round(fc_500_last$`95%_lower`, 2)
fc_500_last_upper <- round(fc_500_last$`95%_upper`, 2)
fc_500_last_mean  <- round(fc_500_last$.mean, 2)

# Atmospheric CO2 levels in the year 2100
fc_2100  <- tail(fc_100_years, n =1)

fc_2100_lower <- round(fc_2100$`95%_lower`, 2)
fc_2100_upper <- round(fc_2100$`95%_upper`, 2)
fc_2100_mean  <- round(fc_2100$.mean, 2)
```

> 'CO2 is expected to be at 420 ppm level for the first time on 2031 May with expected value of 420.06 with 392.3 - 447.82 95% confidence interval.'
> 'CO2 is expected to be at 420 ppm level for the last time on 2035 Oct with expected value of 420.3 with 387.79 - 452.8 95% confidence interval.'
> 'CO2 is expected to be at 500 ppm level for the first time on 2083 Apr with expected value of 500.21 with 403.22 - 597.2 95% confidence interval.'
> 'CO2 is expected to be at 500 ppm level for the last time on 2087 Sep with expected value of 500.89 with 396.82 - 604.96 95% confidence interval.'
> 'By the end of 2100 year, the CO2 is expected to be at 524.03 ppm level with 397.75 - 650.31 95% confidence interval.'

4 Report from the Point of View of the Present

One of the very interesting features of Keeling and colleagues' research is that they were able to evaluate, and re-evaluate the data as new series of measurements were released. This permitted the evaluation of previous models' performance and a much more difficult question: If their models' predictions were "off" was this the result of a failure of the model, or a change in the system?

4.1 (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.

4.2 (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.)

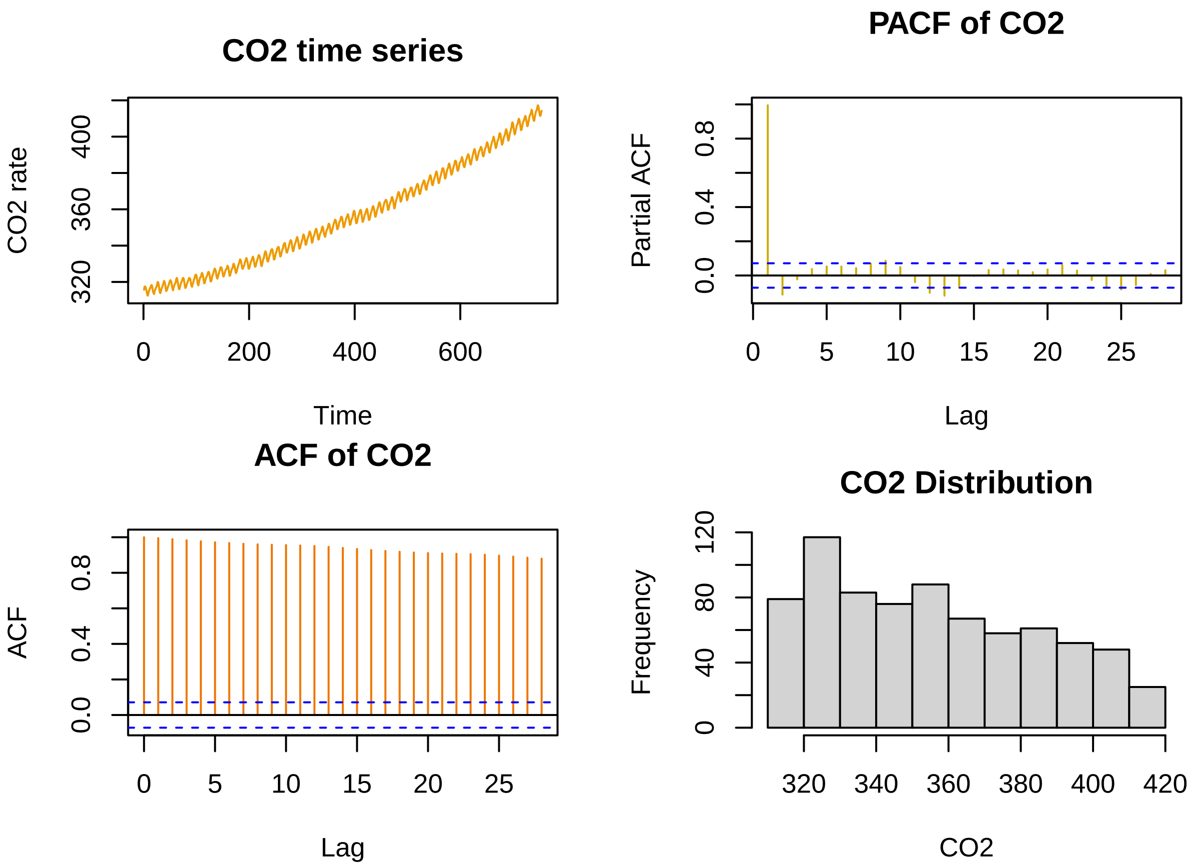
Create a data pipeline that starts by reading from the appropriate URL, and ends by saving an object called `co2_present` that is a suitable time series object.

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.

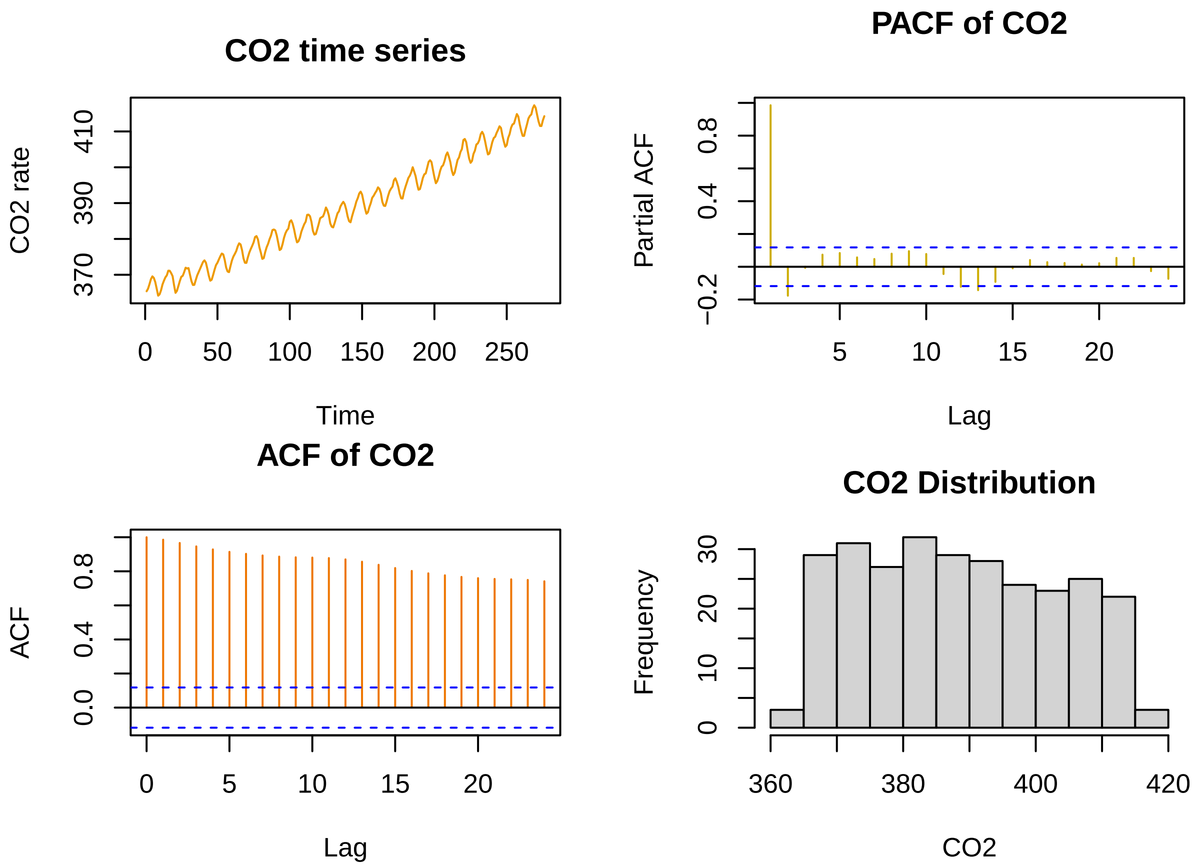
```
read.csv(
  url("https://gml.noaa.gov/webdata/ccgg/trends/co2/co2_mm_mlo.csv"),
  skip = 52) %>%
  mutate(time_index = make_datetime(year, month)) %>%
  mutate(time_index = yearmonth(time_index)) %>%
  filter(year < 2021) %>%
  as_tsibble(index = time_index) -> co2_present

head(co2_present)
```

```
## # A tsibble: 6 x 9 [1M]
##    year month decimal.date average deseasonalized ndays  sdev   unc time_index
##   <int> <int>        <dbl>   <dbl>          <dbl> <int> <dbl> <dbl>      <mth>
## 1  1958     3        1958.    316.           314.    -1 -9.99 -0.99   1958 Mar
## 2  1958     4        1958.    317.           315.    -1 -9.99 -0.99   1958 Apr
## 3  1958     5        1958.    318.           315.    -1 -9.99 -0.99   1958 May
## 4  1958     6        1958.    317.           315.    -1 -9.99 -0.99   1958 Jun
## 5  1958     7        1959.    316.           315.    -1 -9.99 -0.99   1958 Jul
## 6  1958     8        1959.    315.           316.    -1 -9.99 -0.99   1958 Aug
```

```
# Getting the overall CO2 values
value <- co2_present$average

# Making four different plots for evaluation
par(mfrow=c(2,2), mar = c(4.1, 4.1, 3, 2))
plot(value, type = "l", col="orange2", main="CO2 time series",
     xlab="Time", ylab="CO2 rate")
pacf(value, col="gold3", main="PACF of CO2")
acf(value, col="darkorange2", main="ACF of CO2")
hist(value, main="CO2 Distribution",
     ylab = "Frequency", xlab="CO2")
```



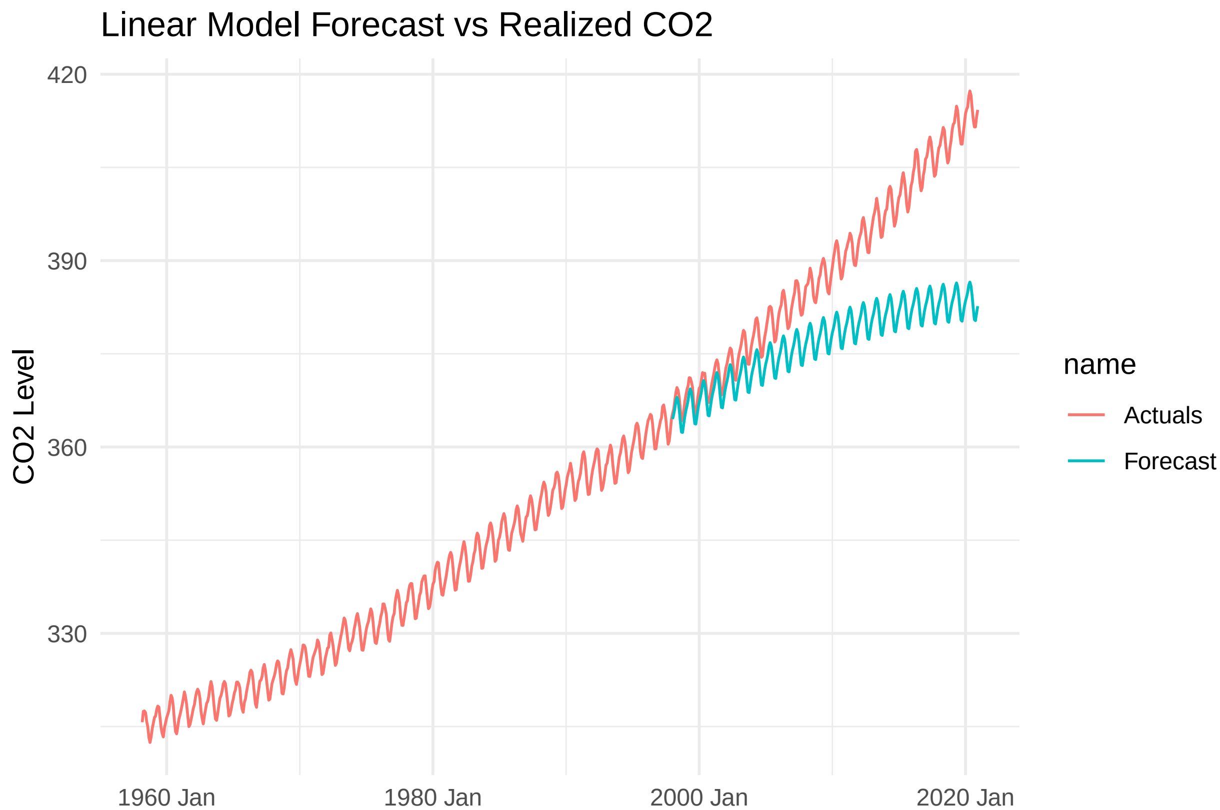

```
# Getting the CO2 values since 1997
value <- co2_present %>% filter(year > 1997)
value <- value$average

# Making four different plots for evaluation
par(mfrow=c(2,2), mar = c(4.1, 4.1, 3, 2))
plot(value, type = "l", col="orange2", main="CO2 time series",
     xlab="Time", ylab="CO2 rate")
pacf(value, col="gold3", main="PACF of CO2")
acf(value, col="darkorange2", main="ACF of CO2")
hist(value, main="CO2 Distribution",
     ylab = "Frequency", xlab="CO2")
```



4.3 (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.)

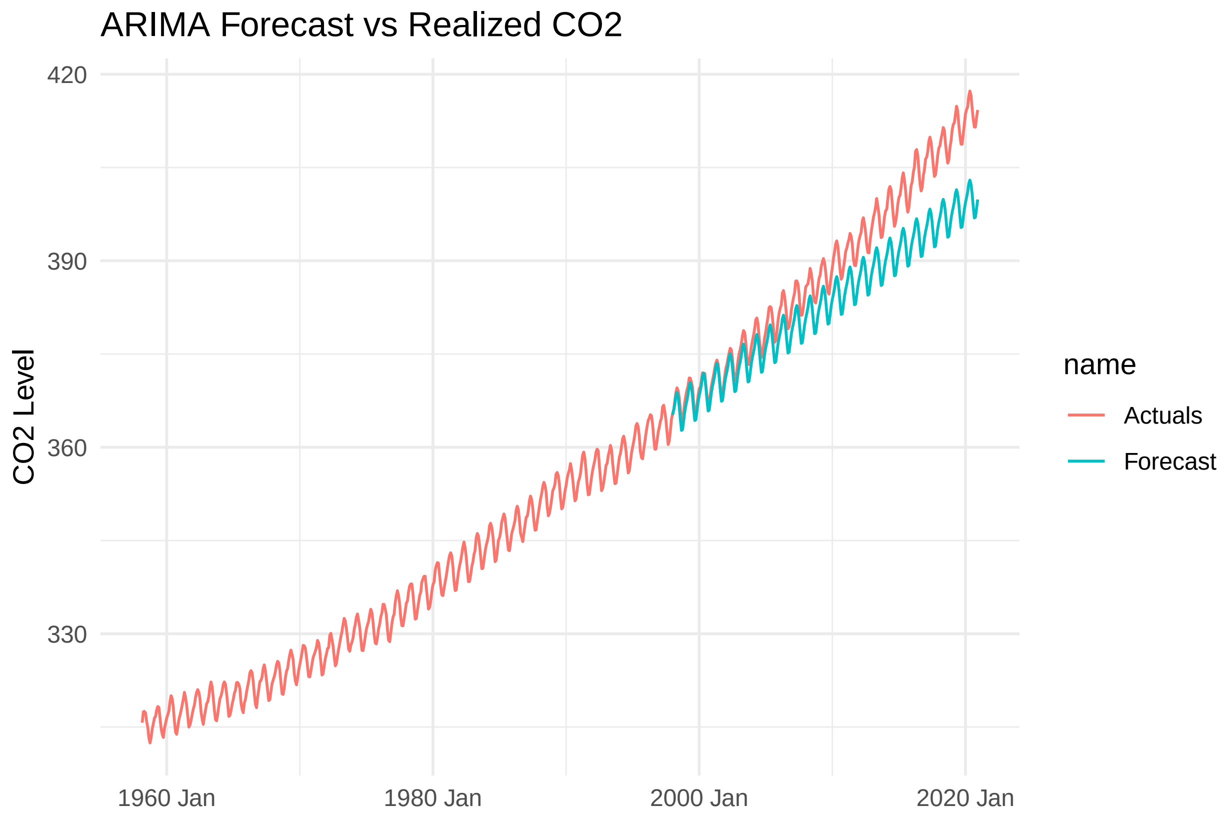
```
forecast <- append(rep(NA, length=nrow(co2_present)-nrow(fc_co2_2020)),
                   fc_co2_2020$.mean)
data.frame(time_index = co2_present$time_index, Actuals = co2_present$average,
           Forecast = forecast) %>%
  pivot_longer(cols=c('Actuals', 'Forecast')) -> lm_vs_actuals

lm_vs_actuals %>%
  ggplot(aes(x = time_index, y = value, color = name)) +
  geom_line() +
  labs(y = 'CO2 Level', x = '', title = 'Linear Model Forecast vs Realized CO2')
```



4.4 (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.

```
forecast <- append(rep(NA, length=nrow(co2_present)-nrow(fc_co2_2022)),
                   fc_co2_2022$.mean)
data.frame(time_index = co2_present$time_index, Actuals = co2_present$average,
           Forecast = forecast) %>%
  pivot_longer(cols=c('Actuals', 'Forecast')) -> arima_vs_actuals

arima_vs_actuals %>%
  ggplot(aes(x = time_index, y = value, color = name)) +
  geom_line() +
  labs(y = 'CO2 Level', x = '', title = 'ARIMA Forecast vs Realized CO2')
```


ARIMA Forecast vs Realized CO2
CO2 Level
420
390
360
330
1960 Jan
1980 Jan
2000 Jan
2020 Jan
name
Actuals
Forecast

4.5 (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.)

```
# Collecting both forecasts and actuals in the same dataframe.
lm_forecast <- append(rep(NA, length = nrow(co2_present) - nrow(fc_co2_2020)),
                      fc_co2_2020$.mean)
arima_forecast <- append(rep(NA, length = nrow(co2_present) - nrow(fc_co2_2022)),
                         fc_co2_2022$.mean)
data.frame(time_index = co2_present$time_index, actuals = co2_present$average,
           lm_forecast = lm_forecast, arima_forecast = arima_forecast) %>%
  filter(!is.na(lm_forecast), !is.na(arima_forecast)) %>%
  mutate(lm_error = lm_forecast - actuals,
         arima_error = arima_forecast - actuals) -> forecast_perf_df
```

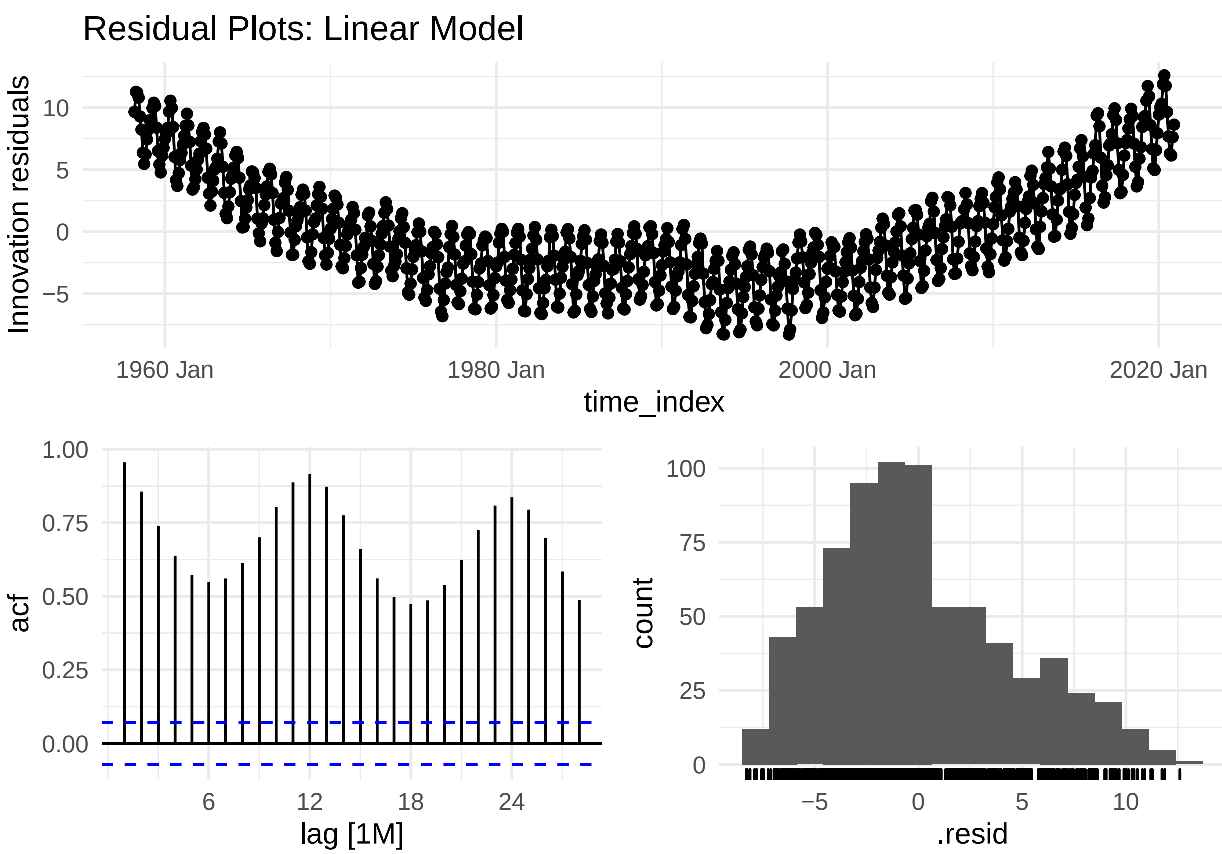
```
# Max actual data from 1958 to 2020
max_actual_row <- co2_present[which.max(co2_present$average),]
max_actual_date <- max_actual_row$time_index
max_actual_value <- max_actual_row$average

# Predicted values and date over the max actual
fc_100_years %>%
  filter(.mean >= max_actual_value) -> fc_417
fc_417_first <- head(fc_417, n =1)
fc_417_first_value <- fc_417_first$.mean
fc_417_first_time  <- fc_417_first$index

# Precited values on actual maximum date
fc_100_years %>%
  filter(index == max_actual_date) -> max_date_pred_values
pred_date_average <- round(max_date_pred_values$.mean, 2)
pred_date_lower <- round(max_date_pred_values$`95%_lower`, 2)
pred_date_upper <- round(max_date_pred_values$`95%_upper`, 2)

# Difference between predicted and actual on max date
pred_act_diff <- max_actual_value - pred_date_average
```

We originally predicted that CO2 would cross the 420 ppm threshold for the first time on 2031 May with an expected value of 420.06 with a 95% confidence interval between 392.3 - 447.82. The maximum average monthly value to date is 417.31 on 2020 May. In our predictions, the nearest date with a value at or over 417.31 is 2030 Apr. Therefore, our predicted average CO2 values were approximately 10-years behind the actuals. When comparing our predicted values in 2020 May to the actual data, we observe an average predicted value of 402.99 with a 95% confidence interval between 385.92 - 420.05. Although our predicted average value was lower than the actual by value by 14.32 ppm, the actual value of 417.31 is between our 95% confidence interval 385.92 - 420.05.

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.)

```r
# Build model
co2_present %>%
  model(TSLM(average ~ trend())) -> fit_co2_trend_present

# Print report
report(fit_co2_trend_present)
```

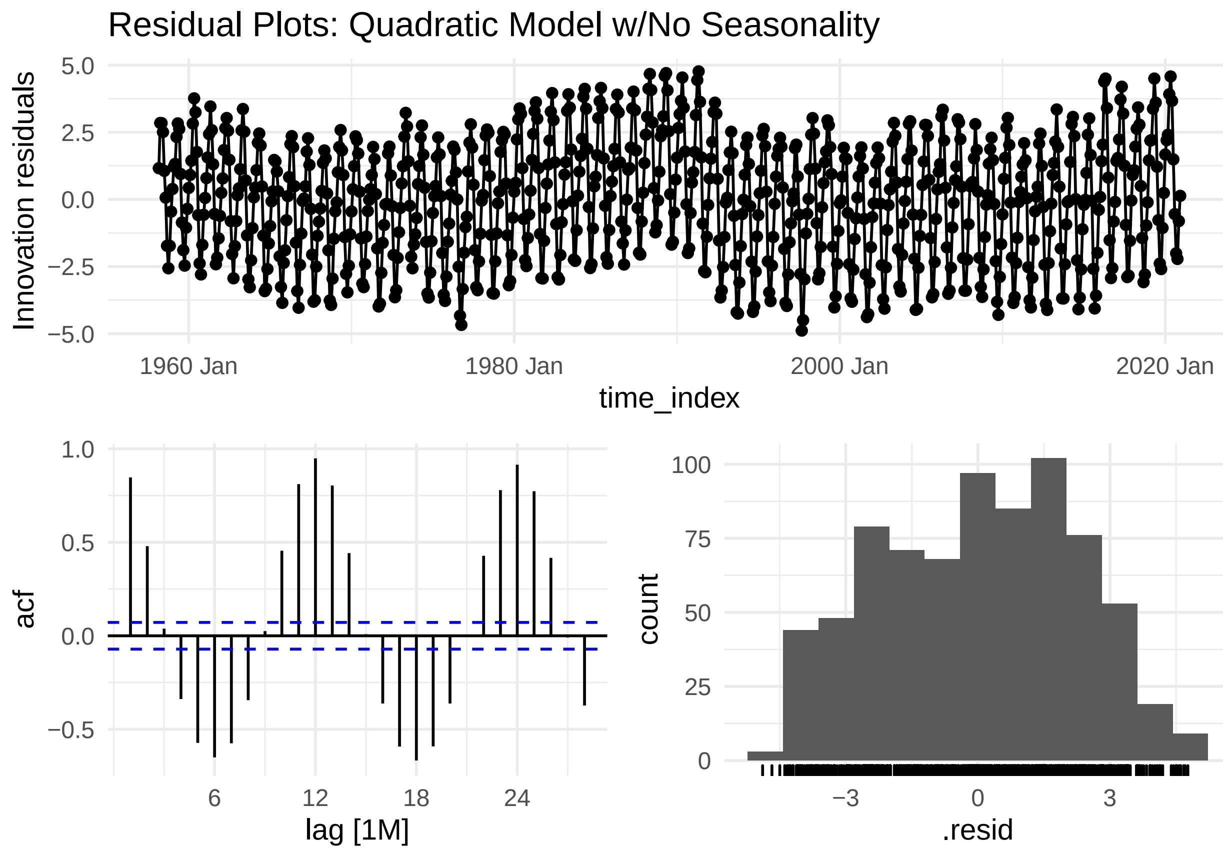
```
## Series: average
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2948 -3.1486 -0.6718  2.7815 12.6017
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.059e+02  3.223e-01   949.2   <2e-16 ***
## trend()     1.323e-01  7.396e-04   178.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.42 on 752 degrees of freedom
## Multiple R-squared: 0.977,   Adjusted R-squared: 0.977
## F-statistic: 3.198e+04 on 1 and 752 DF, p-value: < 2.22e-16
```



Similar to the previous linear model, the linear model exhibits a significant negative curve, meanding the linear prediction is lower than that actual values in the initial and last periods but that it has higher predicted values in the time period in-between the initial and last periods. Therefore, the mean of the residuals are not close to zero and the residual variance does not appear to be constant. The ACF shows a positive relationship between the lag periods, which becomes stronger as the month lag approaches 12. This demonstrates a strong correlation between time periods, which is due to the strong correlation between seasons and the overall positive trend in CO2 emissions. The histogram of the residuals is skewed to the right, indicating that the residuals are not normally distributed.

```r
# Build model
co2_present %>%
  model(TSLM(average ~ trend()+ I(trend()^2))) -> fit_co2_quad_trend_present

# Print model
report(fit_co2_quad_trend_present)
```

```
## Series: average
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8867 -1.8137  0.1124  1.7773  4.7687
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.145e+02  2.441e-01 1288.45   <2e-16 ***
## trend()      6.431e-02  1.493e-03   43.07   <2e-16 ***
## I(trend()^2) 8.999e-05  1.915e-06   47.00   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.228 on 751 degrees of freedom
## Multiple R-squared: 0.9942,  Adjusted R-squared: 0.9942
## F-statistic: 6.404e+04 on 2 and 751 DF, p-value: < 2.22e-16
```

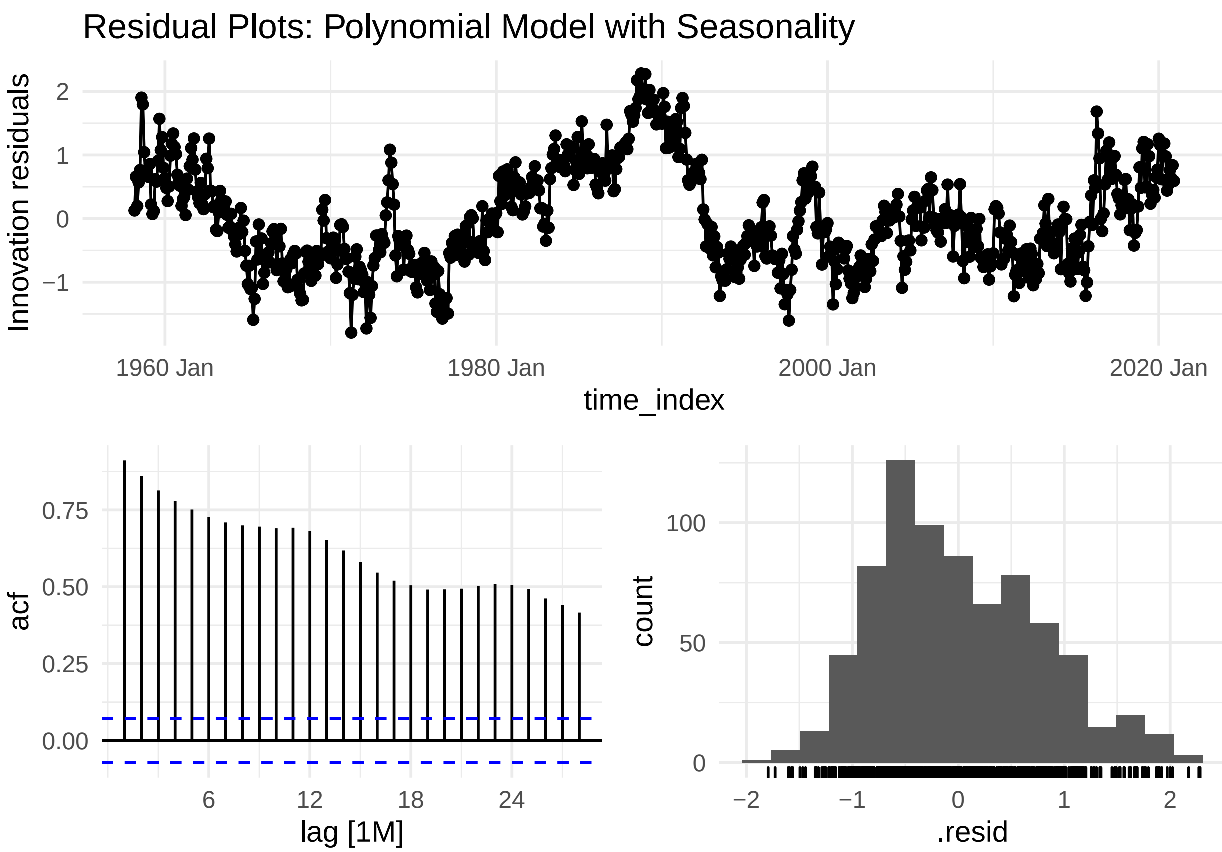


The residual plots for the polynomial model indicate a mean and a variance that is closer to zero. However, there are some clear trends in the residual plots, indicating that the residuals are not constant. The ACF shows a positive relationship between the lag periods that are 12 months apart, while exhibiting a negative correlation at a 6-month lag. These values are beyond the confidence thresholds, meaning the residual are strongly correlated. The histogram of the residusls has a distribution that is significantly wide, representing a cross between a bell-shaped curve and a uniform distriubtion.

```r
# Build model
co2_present %>%
  model(TSLM(average ~ trend() + I(trend()^2) + I(trend()^3) + season())) -> fit_co2_full_trend_present

# Print model
report(fit_co2_full_trend_present)
```

```
## Series: average
## Model: TSLM
##
## Residuals:
##      Min       1Q  Median      3Q     Max
## -1.7936 -0.5823 -0.1251  0.5784  2.2826
##
## Coefficients:
##                  Estimate Std. Error  t value Pr(>|t|)
## (Intercept)     3.141e+02  1.492e-01 2105.620  < 2e-16 ***
## trend()         7.030e-02  1.313e-03   53.551  < 2e-16 ***
## I(trend()^2)    7.040e-05  4.039e-06   17.430  < 2e-16 ***
## I(trend()^3)    1.735e-08  3.517e-09    4.934 9.96e-07 ***
## season()year2   6.277e-01  1.405e-01    4.469 9.11e-06 ***
```

```
## season()year3   1.361e+00  1.399e-01    9.727  < 2e-16 ***
## season()year4   2.508e+00  1.399e-01   17.927  < 2e-16 ***
## season()year5   2.960e+00  1.399e-01   21.152  < 2e-16 ***
## season()year6   2.247e+00  1.399e-01   16.060  < 2e-16 ***
## season()year7   6.013e-01  1.399e-01    4.298 1.96e-05 ***
## season()year8  -1.538e+00  1.399e-01  -10.995  < 2e-16 ***
## season()year9  -3.233e+00  1.399e-01  -23.103  < 2e-16 ***
## season()year10 -3.323e+00  1.399e-01  -23.749  < 2e-16 ***
## season()year11 -2.120e+00  1.399e-01  -15.150  < 2e-16 ***
## season()year12 -9.391e-01  1.399e-01   -6.712 3.83e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7821 on 739 degrees of freedom
## Multiple R-squared: 0.9993,  Adjusted R-squared: 0.9993
## F-statistic: 7.462e+04 on 14 and 739 DF, p-value: < 2.22e-16
```

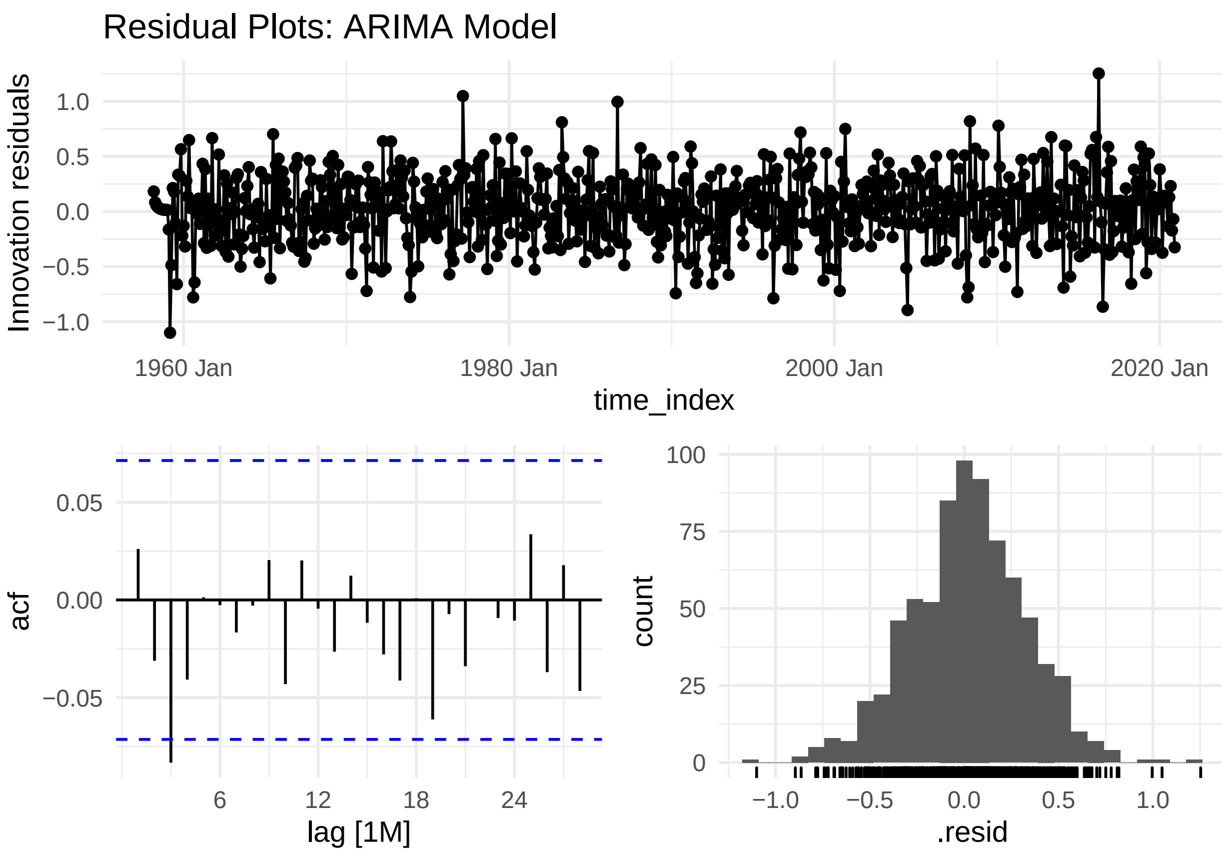


The time series plot of the residuals demonstrates residual values that may have a mean eqaul to zero but both the mean and the variance are not constant. This suggests that the the polynomial model with the seasonal adjustment might not be predicting the seasonal and/or positive trend of the underlying data. The ACF shows a strong correlation between the first period and the proceeding 24 - 30 months. The histogram of the residuals is skewed to the right, meaning the residuals are not normally distributed.

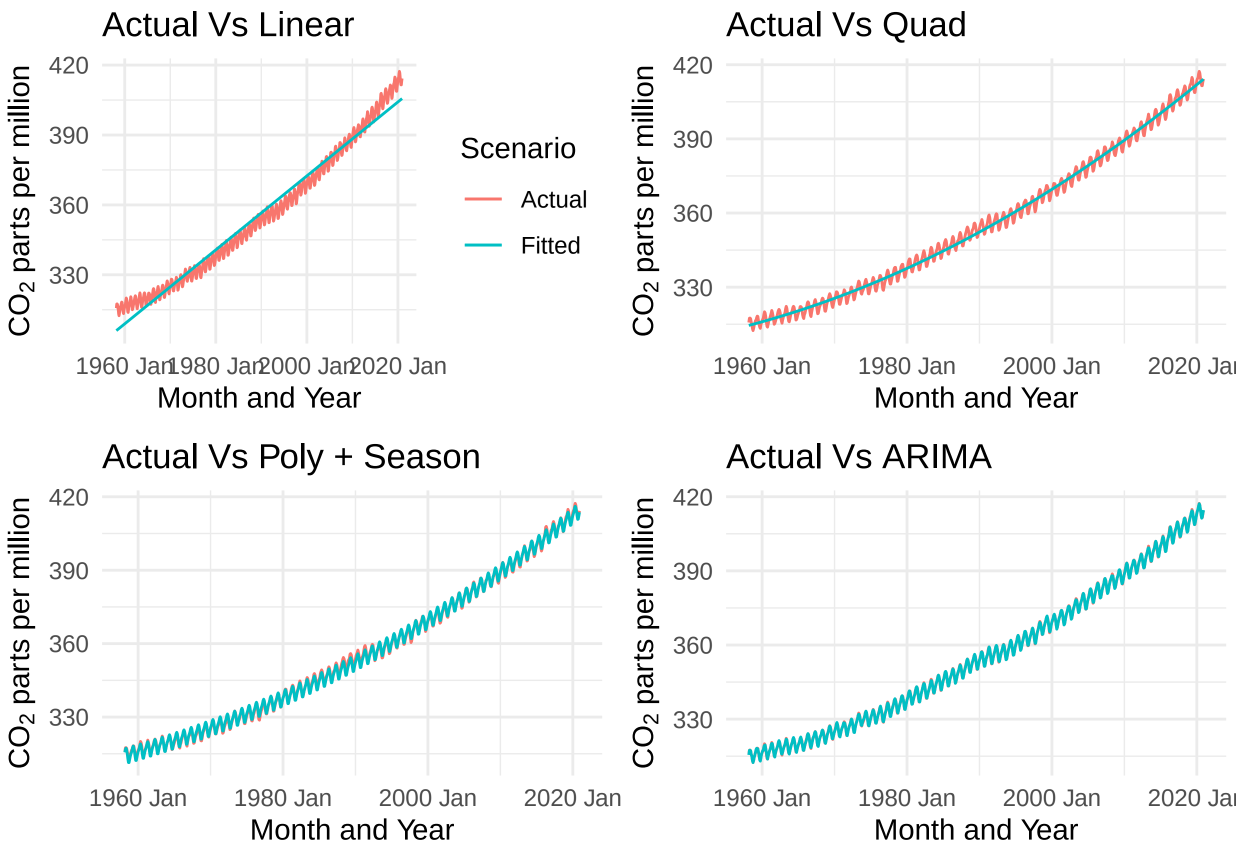
```
model.bic2 <- co2_present %>%
  model(ARIMA(average ~ 0 + pdq(0, 1, 1) + PDQ(1, 1, 2),
                       ic="bic"))
```

```
model.bic2 %>%
  report()
```

```
## Series: average
## Model: ARIMA(0,1,1)(1,1,2)[12]
##
## Coefficients:
##           ma1     sar1     sma1     sma2
##       -0.3816  -0.3091  -0.5396  -0.2793
## s.e.   0.0381   1.4942   1.4992   1.2918
##
## sigma^2 estimated as 0.09794:  log likelihood=-190.1
## AIC=390.2   AICc=390.29   BIC=413.24
```

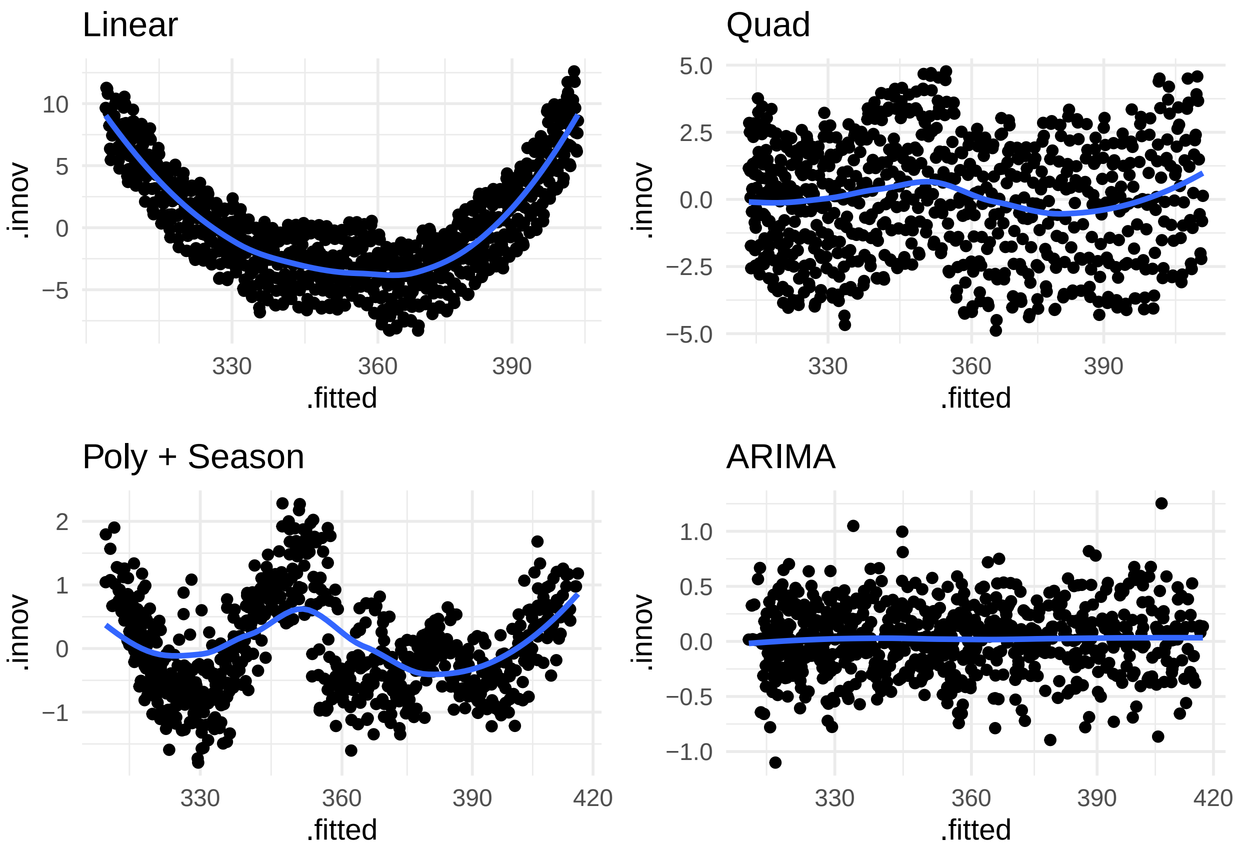


The original ARIMA model we had developed had the following hyper parameters (model.bic2): ARIMA(0,1,1)(1,1,2)[12]. However, when using the same code to identify the best model with the updated data from 1997 to 2020, the following hyper parameters were identified as the best fit: ARIMA(1,1,1)(2,1,1)[12]. This suggests that a different set of hyperparameters should be selected given the updated data. However, based on our research question we decided to use the same model as before in our diagnostic plots. The residual plots for the arima model indicate a mean and a variance that is is constant and closer to zero.The ACF shows what looks like a random trend and there is only one value at time lag 3 that extends beyond the threshold. If should be noted that no ACF values extend beyond this threshold using the new hyper parameters. The histogram of the residuals has a distribution that appears to be normally distributed.

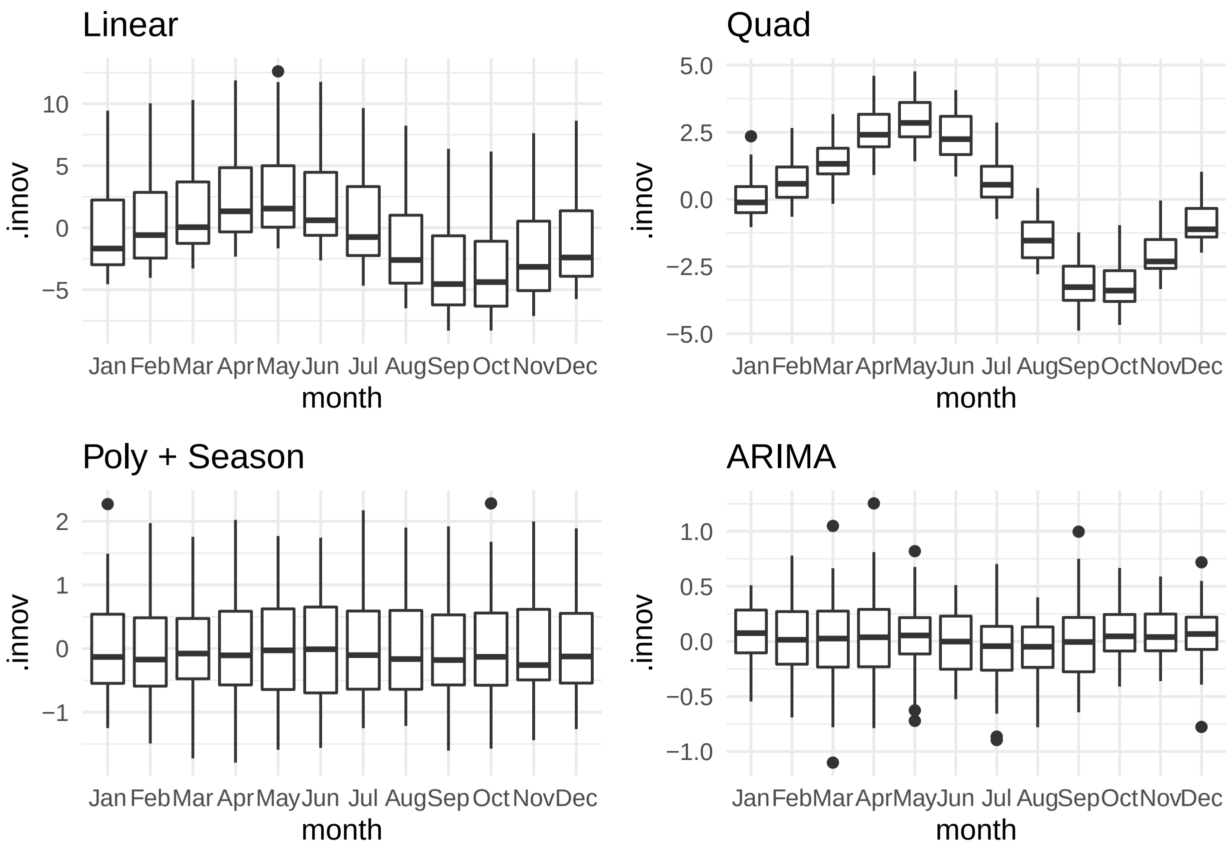


The graphs above shows the plots of the fitted models versus the actuals over the observation period. Both the linear and quadratic with no seasonal adjustment do not adjust for any seasonal variation. Although the polynomial model with seasonal variation appears to fit the data quite well, there are periods where the model under estimates or over estimates. The ARIMA model appears to have the most accurate predictions relative to the other models.

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The graphs above shows the plots of the residuals versus the fitted values. With the exception of the ARIMA model, all residual plots show a pattern, meaning there may be heteroscedasticity in the errors and the variance of the residuals may not be constant



The graphs above shows boxplots of the residuals by month. The linear and quadratic models show obvious deviations from mean of zero, while the quadratic plus seasonal model and the ARIMA model have residual values near zero.

```
augment(fit_co2_trend_present) %>%
  features(.innov, ljung_box, dof = 14, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model                    lb_stat lb_pvalue
##   <chr>                       <dbl>     <dbl>
## 1 TSLM(average ~ trend())     9264.         0
```

```
augment(fit_co2_quad_trend_present) %>%
  features(.innov, ljung_box, dof = 14, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model                                   lb_stat lb_pvalue
##   <chr>                                      <dbl>     <dbl>
## 1 TSLM(average ~ trend() + I(trend()^2))     6081.         0
```

```
augment(fit_co2_full_trend_present) %>%
  features(.innov, ljung_box, dof = 14, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model                                                        lb_stat lb_pvalue
##   <chr>                                                           <dbl>     <dbl>
## 1 TSLM(average ~ trend() + I(trend()^2) + I(trend()^3) + seas~    7886.         0
```

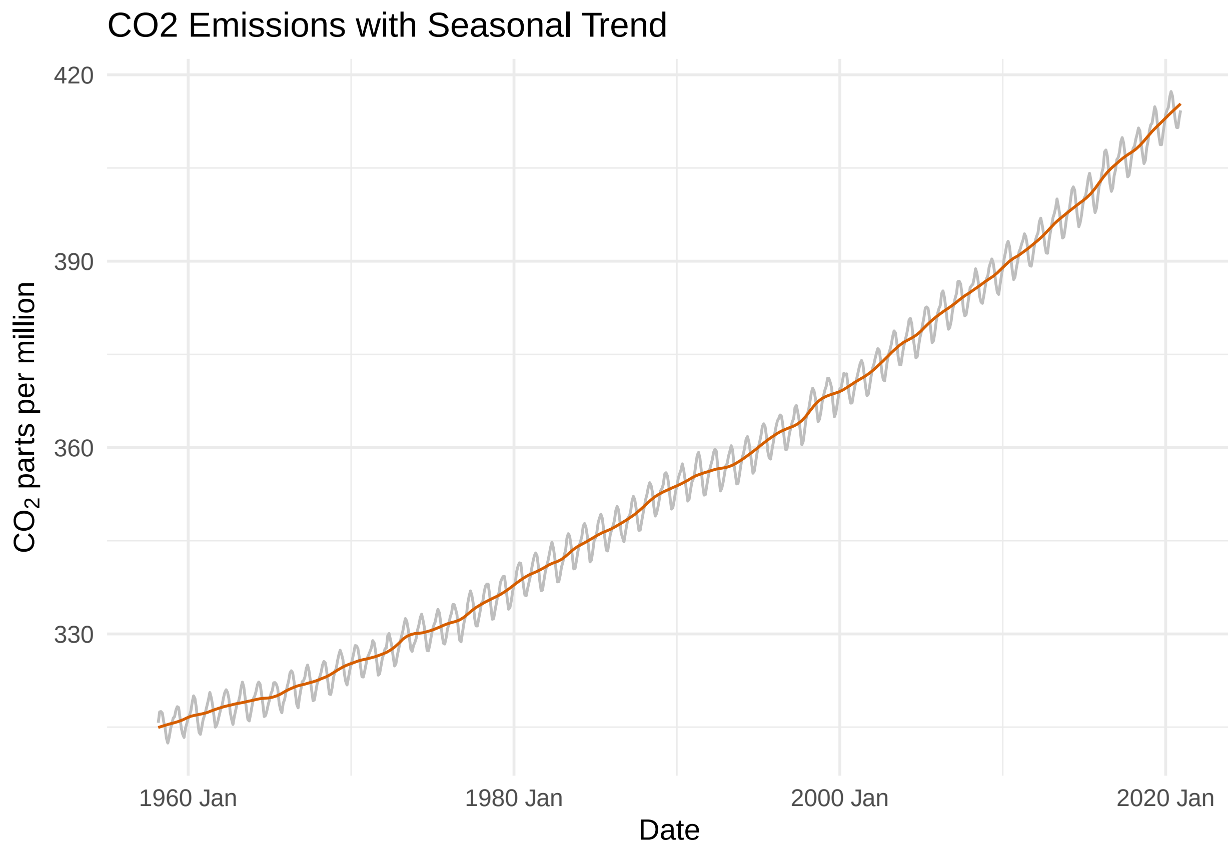
```
augment(model.bic2) %>%
  features(.innov, ljung_box, dof = 14, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model                                                           lb_stat lb_pvalue
##   <chr>                                                              <dbl>     <dbl>
## 1 "ARIMA(average ~ 0 + pdq(0, 1, 1) + PDQ(1, 1, 2), ic = \"bi~    16.7    0.0802
```

Finally, the p-value for all of the respective ljung_boxtest statistic is less than 0.05, except the ARIMA model. Therefore, we can conclude that the residual values are independent for the ARIMA model only.

(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.




```
# Create data frame with season values
df_season <- as_tsibble(data.frame(components(co2_present_season)))
```

```
## Using `time_index` as index variable.
```

```
# Test size
test.size <- 24 # Months
test.df <- co2_present %>%
  filter(year > 2018) %>%
  mutate(.mean = average) # used later for ggplot

# Train for non-seasonal (ns)
train_ns <- co2_present %>%
  slice(1:(n() - test.size))

# Train for seasonal (s)
train_s <-  df_season %>%
  slice(1:(n() - test.size))

head(train_s)
```

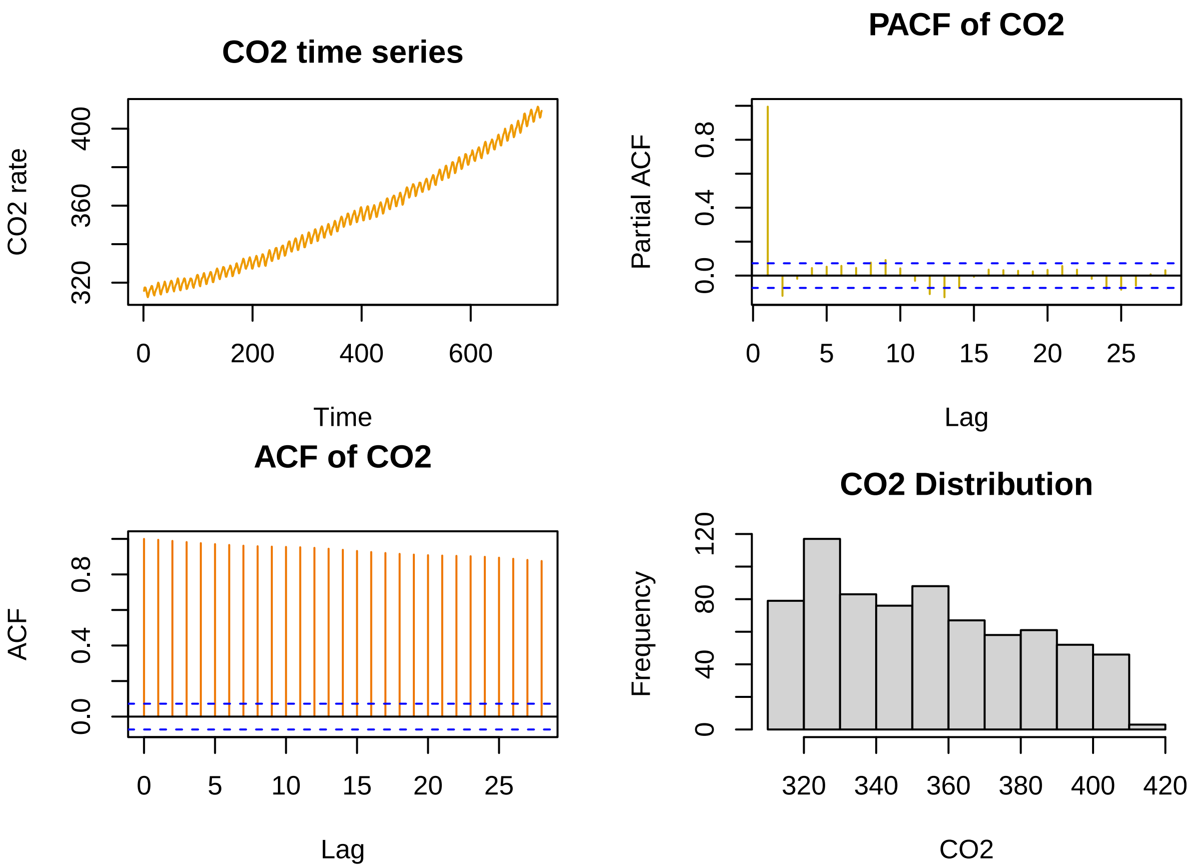
```
## # A tsibble: 6 x 7 [1M]
##   .model time_index average trend season_year remainder season_adjust
##   <chr>       <mth>   <dbl> <dbl>       <dbl>     <dbl>         <dbl>
## 1 stl      1958 Mar    316.  315.       1.03     -0.264          315.
## 2 stl      1958 Apr    317.  315.       2.22      0.217          315.
## 3 stl      1958 May    318.  315.       2.77     -0.347          315.
## 4 stl      1958 Jun    317.  315.       2.28     -0.189          315.
## 5 stl      1958 Jul    316.  315.       0.857    -0.217          315.
## 6 stl      1958 Aug    315.  315.      -1.02      0.667          316.
```

```
adf.test(train_ns$average, alternative = "stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train_ns$average
## Dickey-Fuller = -0.72341, Lag order = 8, p-value = 0.9684
## alternative hypothesis: stationary
```

```
train_ns %>% features(average, unitroot_kpss)
```

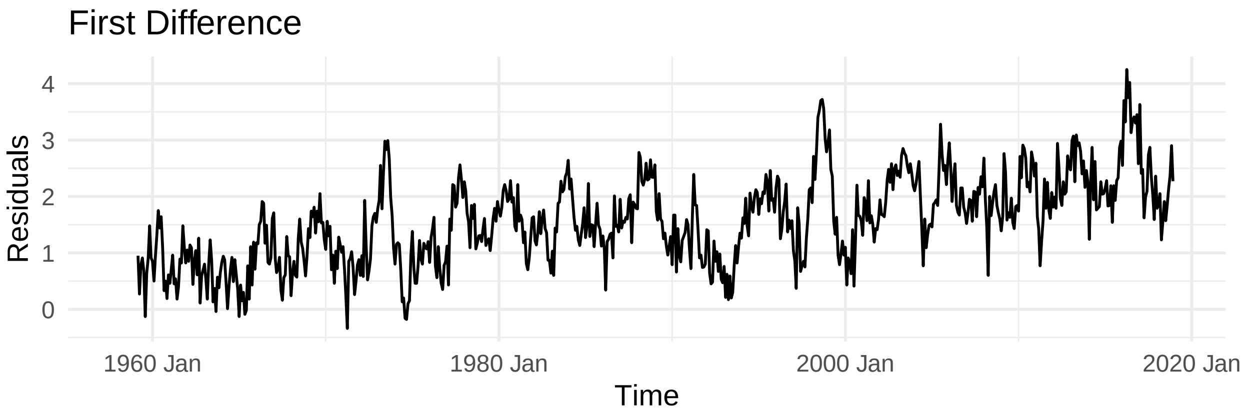
```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1      10.4        0.01
```

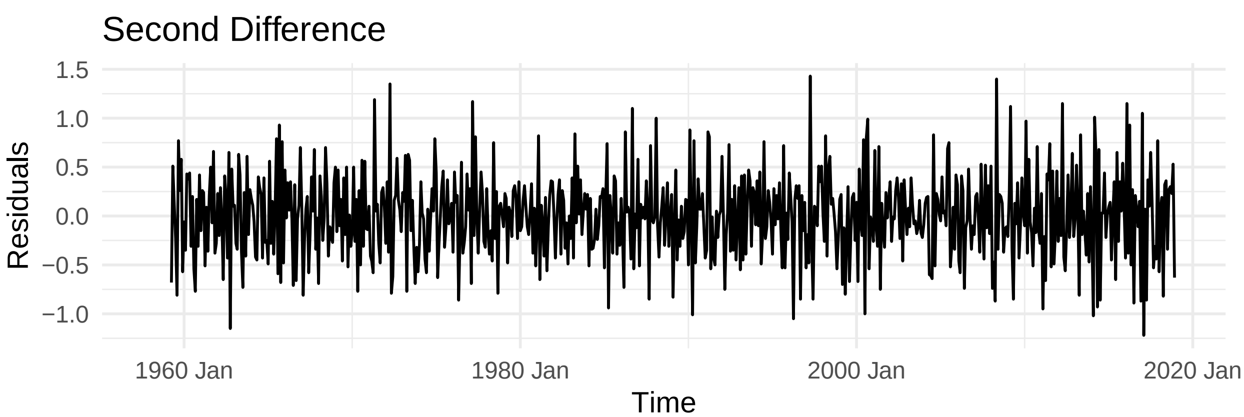


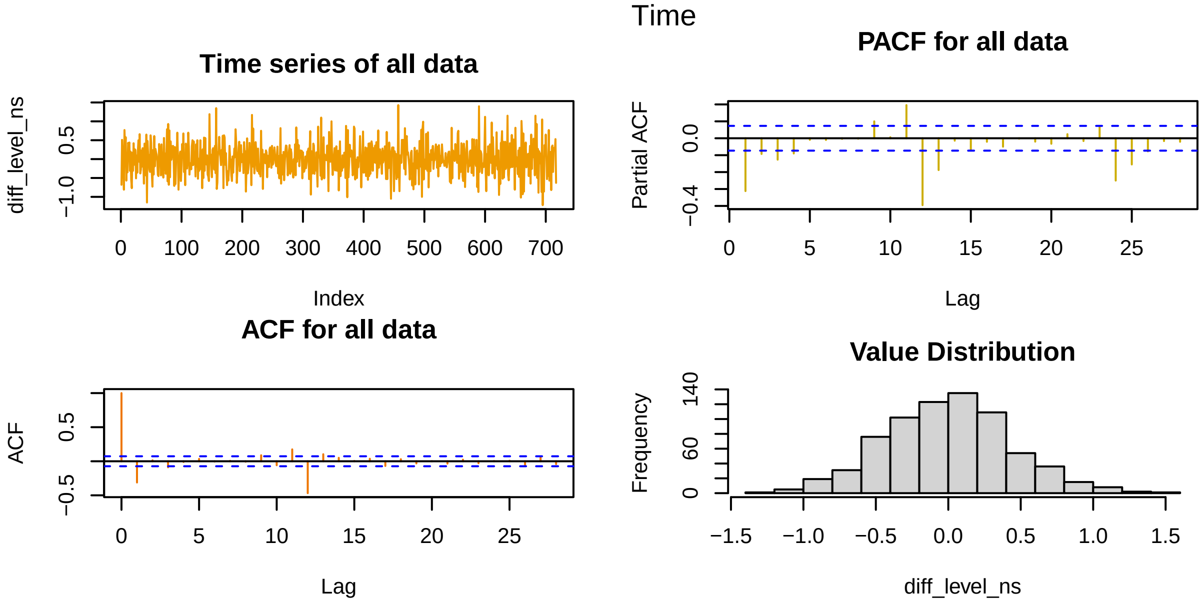
The ADF test statistic and p-value come out to be equal to -0.72 and 0.96, respectively. Since the p-value is greater than 0.05, we would fail to reject the null hypothesis that the time series is non-stationary. Similarly, the p-value is reported as 0.01 and the test statistic (10.36) is bigger than the 1% critical value, and the p-value is less than 0.01, indicating that the null hypothesis is rejected and the data is not stationary. The plots show a clear seasonal trend in the time series plot. The PACF shows strong correlation at time lag 1 and then a slightly negative correlation around time 12. The ACF shows strong correlation between the all periods between 0 and 30, which is most likely indicative of the the positive trend in CO_2 emissions.

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 13 row(s) containing missing values (geom_path).
```







```r
ns_diff_data <- train_ns %>%
  mutate(second_diff = difference(difference(average, 12)), 1) %>%
  filter(!is.na(second_diff))



adf.test(ns_diff_data$second_diff, alternative = "stationary")
```

```
## Warning in adf.test(ns_diff_data$second_diff, alternative = "stationary"): p-
## value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ns_diff_data$second_diff
## Dickey-Fuller = -8.8912, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

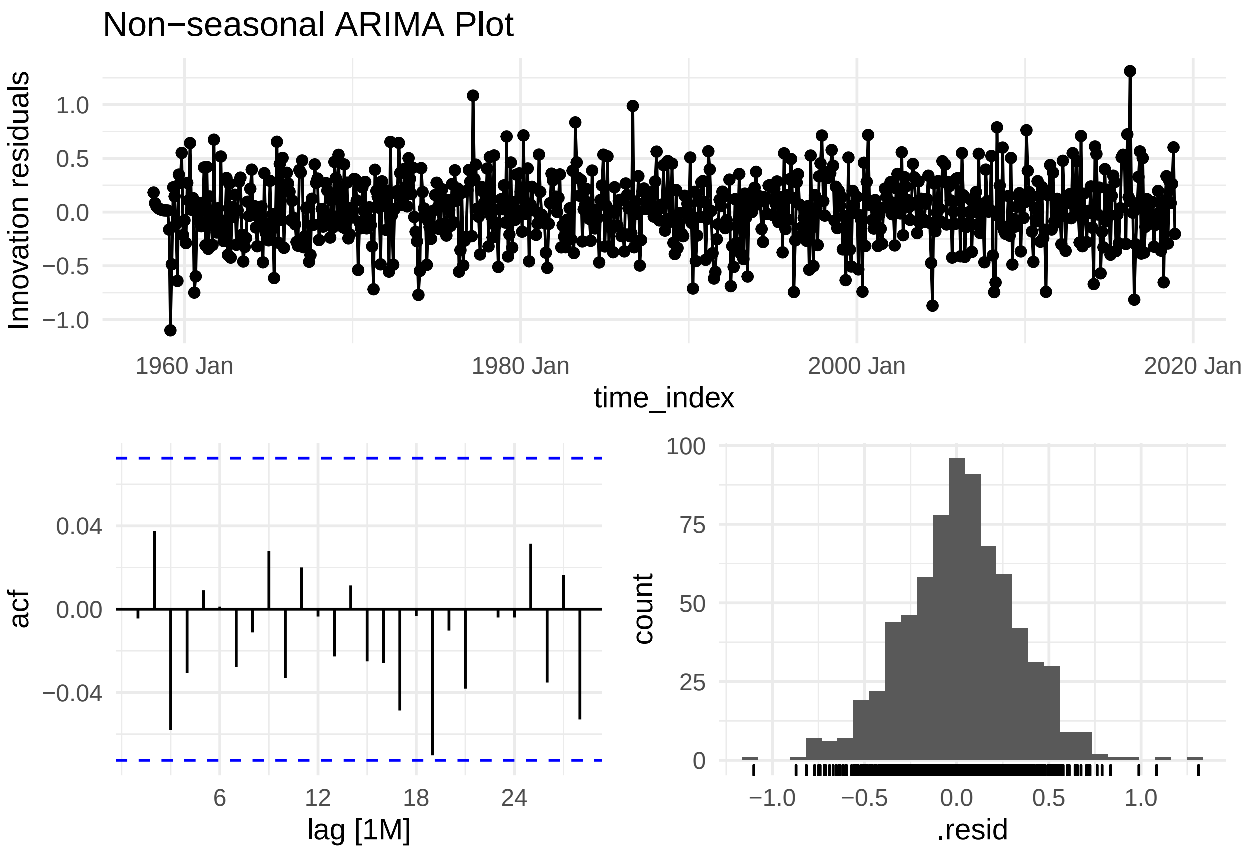
```r
train_ns %>%
  mutate(second_diff = difference(difference(average, 12)), 1) %>%
  features(second_diff, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1   0.00708         0.1
```

The ADF test statistic and p-value come out to be equal to -8.8912 and 0.01, respectively. Since the p-value is less than 0.05, we would reject the null hypothesis that the time series is non-stationary. Similarly, the p-value is reported as 0.1 and the test statistic (0.007) is less than the 1% critical value, and the p-value is greater than 0.01, indicating that the null hypothesis is not rejected and the data is stationary. Although the plots show that residual time series plot is similar to white noise and the residual histogram appears to normally distriubted, the PACF and the ACF has values beyond the threshold values around the 12-month mark. We applied a log transformation but this did not improve the residual plots. Because we know we should use first differencing, and 1 seasonal difference, we will set these hyper parameters and then select a model by the lowest aic

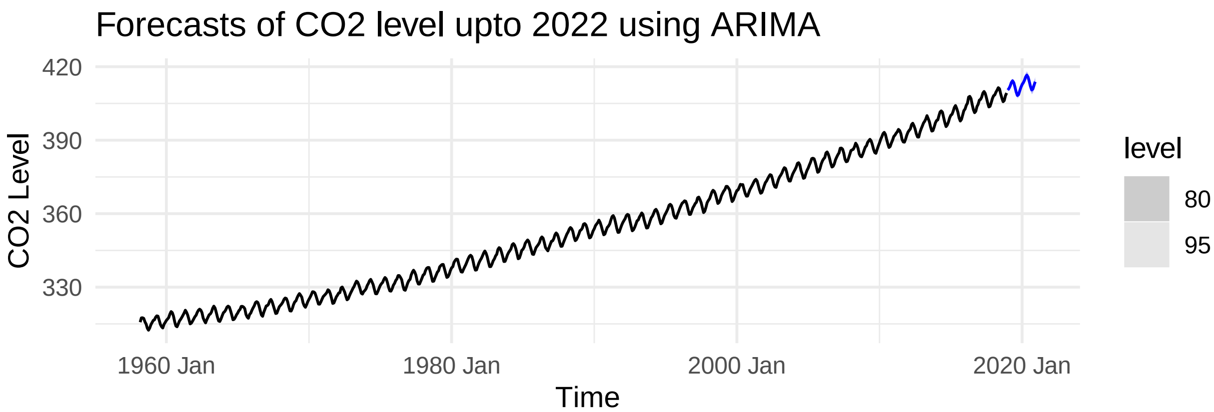
```r
model.ns <- train_ns %>%
  model(ARIMA(average ~ 0 + pdq(0:10, 1, 0:10) + PDQ(0:10, 1, 0:10),
                          ic="bic", stepwise=F, greedy=F))
model.ns %>%
  report()
```

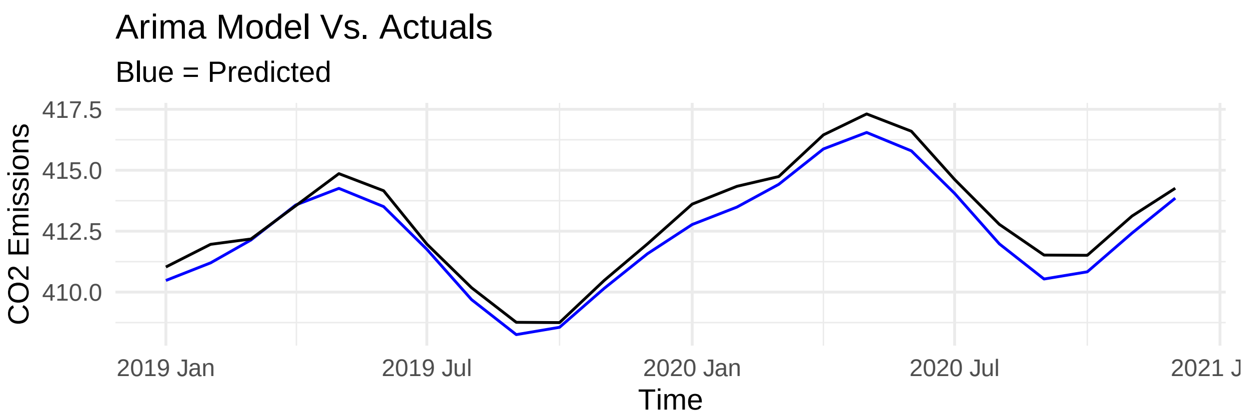
```
## Series: average
## Model: ARIMA(1,1,1)(2,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sar1     sar2     sma1
##       0.1961  -0.5508  -0.0052  -0.0252  -0.8596
## s.e.  0.1019   0.0880   0.0443   0.0427   0.0244
##
## sigma^2 estimated as 0.0981:  log likelihood=-184.29
## AIC=380.57   AICc=380.69   BIC=408.02
```




```
augment(model.ns) %>%
  features(.innov, ljung_box, dof = 14, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model                                                       lb_stat lb_pvalue
##   <chr>                                                          <dbl>     <dbl>
## 1 "ARIMA(average ~ 0 + pdq(0:10, 1, 0:10) + PDQ(0:10, 1, 0:10~    14.8     0.140
```

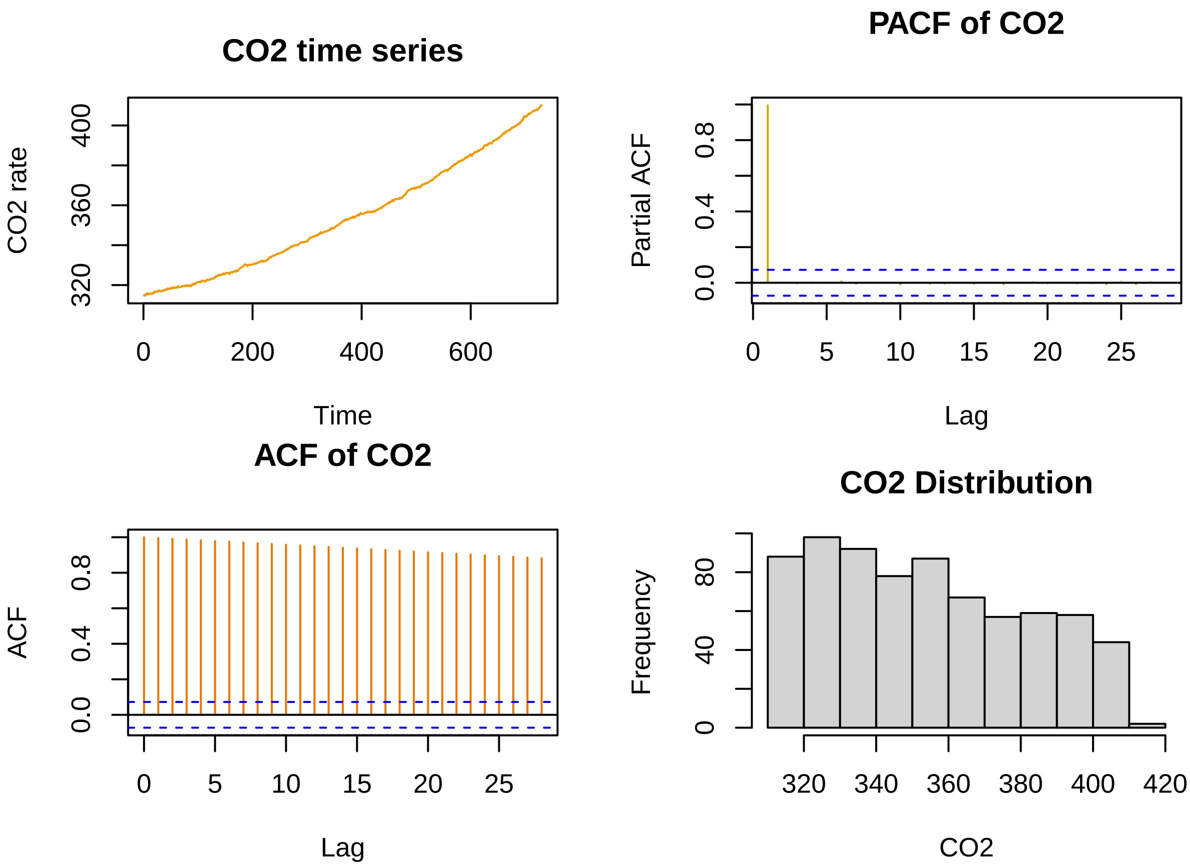




The residual plots for the ARIMA model appear to have constant variance and a mean near zero. The residuals pass the ljung_box test and all residual values are within the threshold values for ACF. The histogram also appears to be normally distributed. The forecast plots seem to track nicely with the actuals for the 24 month period.

```
adf.test(train_s$season_adjust, alternative = "stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train_s$season_adjust
## Dickey-Fuller = -0.33025, Lag order = 8, p-value = 0.9895
## alternative hypothesis: stationary
```

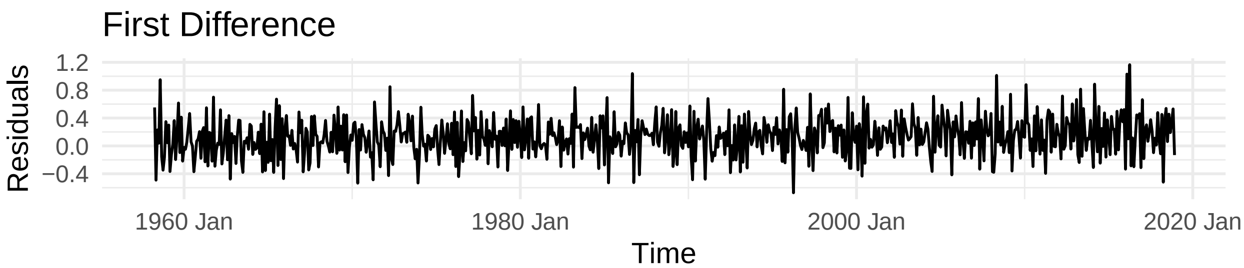


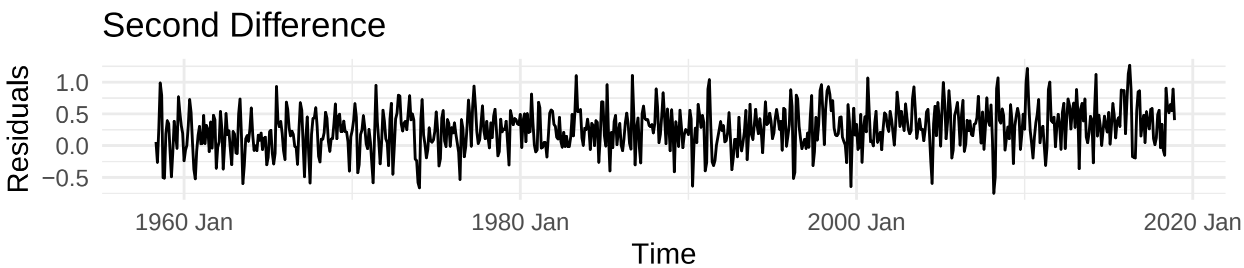
The ADF test statistic and p-value come out to be equal to -0.33 and 0.99, respectively. Since the p-value is greater than 0.05, we would fail to reject the null hypothesis that the time series is non-stationary. . The plots show a positive trend in the time series plot. The PACF shows strong correlation at time lag 1 and then no correlation thereafter. The ACF shows strong correlation between the all periods between 0 and 30, which is most likely indicative of the the positive trend in CO2 emissions. The histogram is positively skewed to the right.

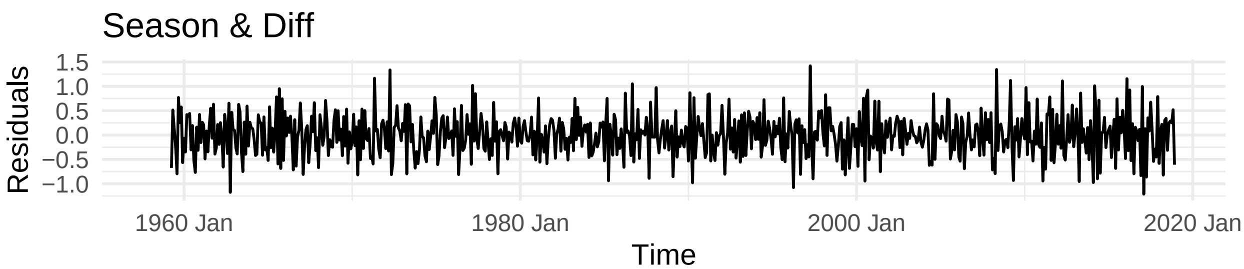
```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 13 row(s) containing missing values (geom_path).
```

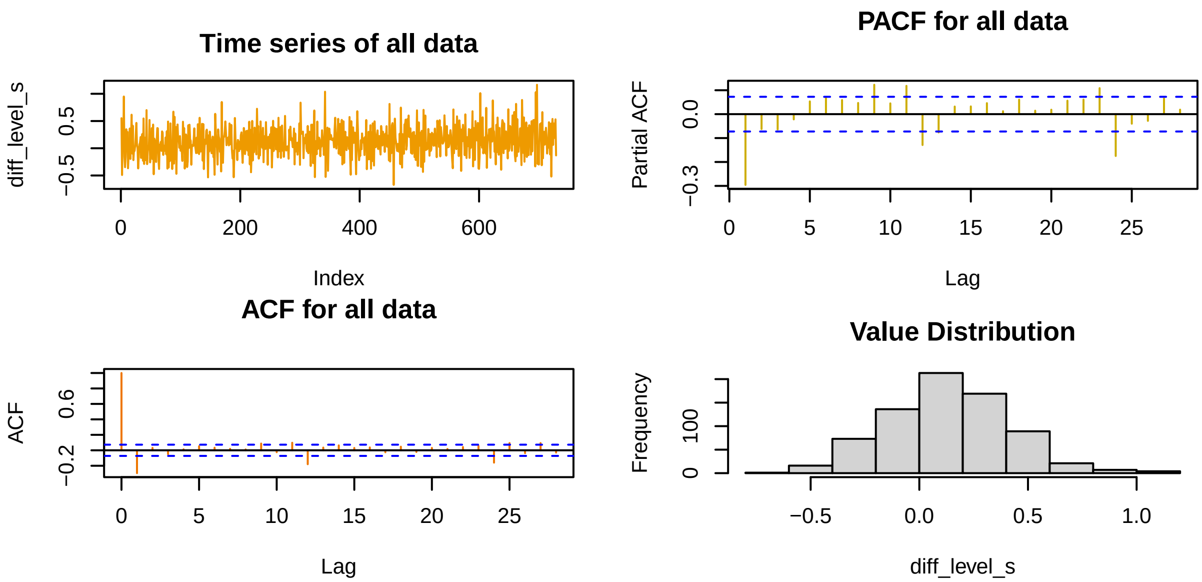







```r
s_diff_data <- train_s %>%
  mutate(season2_diff1 = difference(season_adjust, 1)) %>%
  filter(!is.na(season2_diff1))



adf.test(s_diff_data$season2_diff1, alternative = "stationary")
```

```
## Warning in adf.test(s_diff_data$season2_diff1, alternative = "stationary"): p-
## value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  s_diff_data$season2_diff1
## Dickey-Fuller = -8.8273, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```



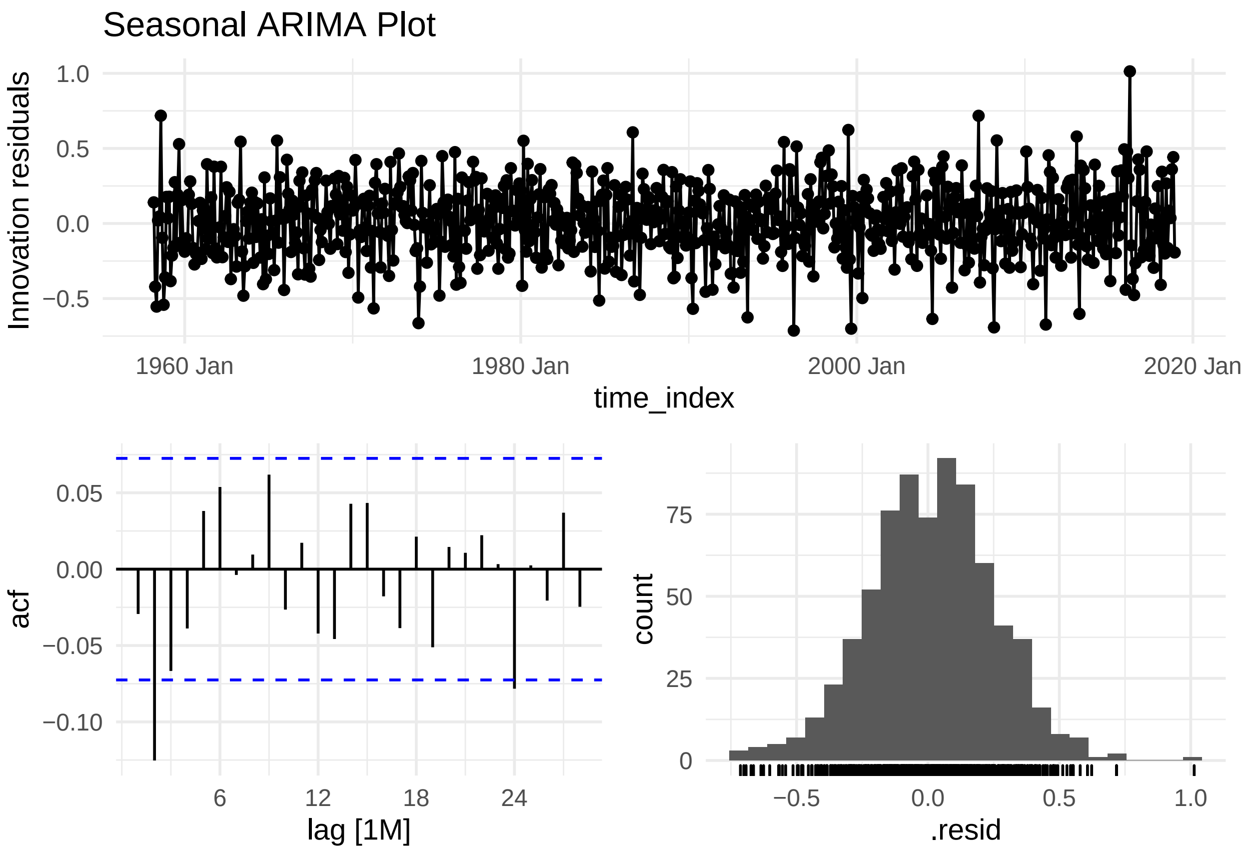
The ADF test statistic and p-value come out to be equal to -8.82 and 0.01, respectively. Since the p-value is less than 0.05, we could reject the null hypothesis that the time series is non-stationary. The time series plot appears to have white noise at a difference of 1. The PACF demonstrates some outlier values at a time lag of 1 and around the 12 the and 24 month lag. Similarly, the ACF has high values at these same lags. The histogram of the residuals appears to be normally distributed. Due to these results, we decided to use a first difference of 1 or 2, a seasonal difference of 0, and we let the ARIMA function choose the remaining hyper parameters.

```
model.s <- train_s %>%
  model(ARIMA(season_adjust ~ 0 + pdq(0:10, 1:2, 0:10) + PDQ(0:10, 0, 0:10),
                         ic="bic", stepwise=F, greedy=F))
```

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

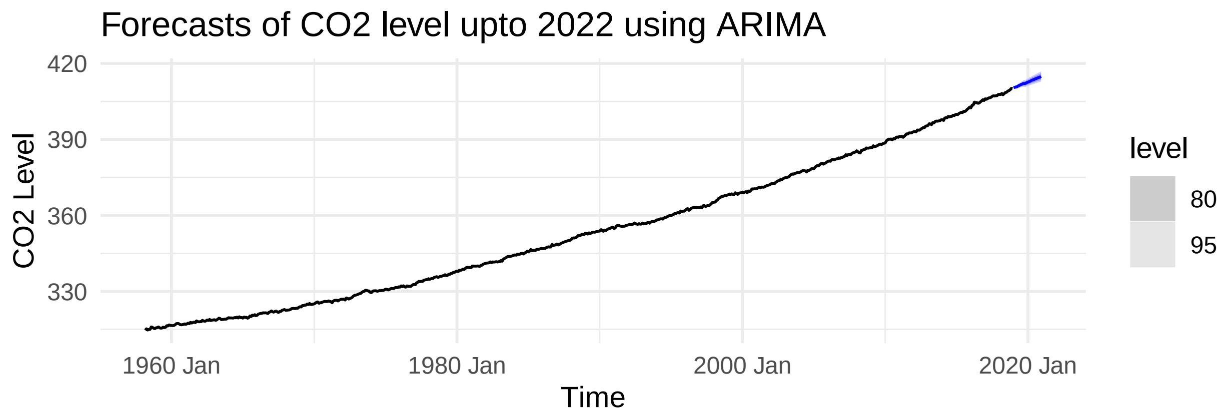
```
model.s %>%
  report()
```

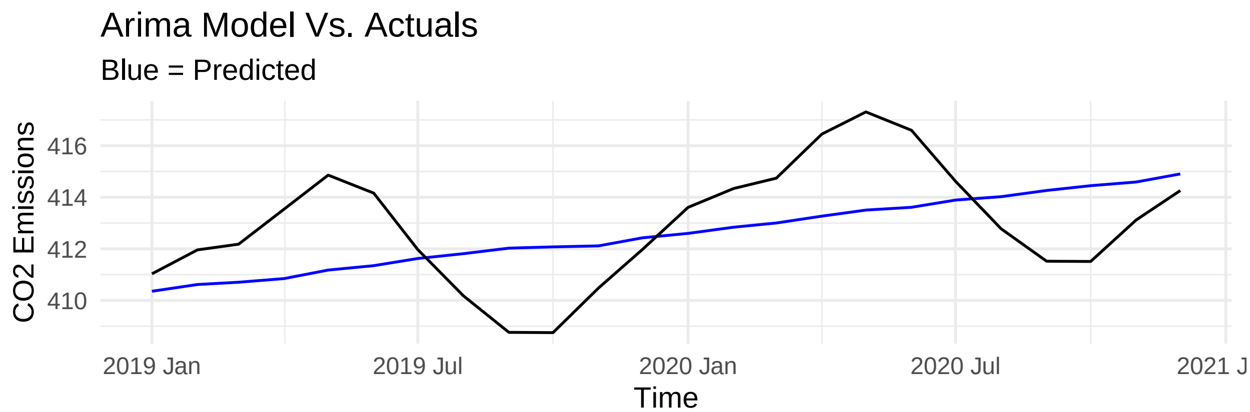
```
## Series: season_adjust
## Model: ARIMA(1,2,1)(4,0,0)[12]
##
## Coefficients:
##           ar1      ma1     sar1     sar2     sar3     sar4
##       -0.3249  -0.9694  -0.3854  -0.3802  -0.3001  -0.2129
## s.e.      NaN   0.0078   0.0037      NaN      NaN      NaN
##
## sigma^2 estimated as 0.05821:  log likelihood=1.5
## AIC=11   AICc=11.16   BIC=43.14
```




```
augment(model.s) %>%
  features(.innov, ljung_box, dof = 14, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model                                                      lb_stat lb_pvalue
##   <chr>                                                         <dbl>     <dbl>
## 1 "ARIMA(season_adjust ~ 0 + pdq(0:10, 1:2, 0:10) + PDQ(0:10,~     38.0 0.0000385
```



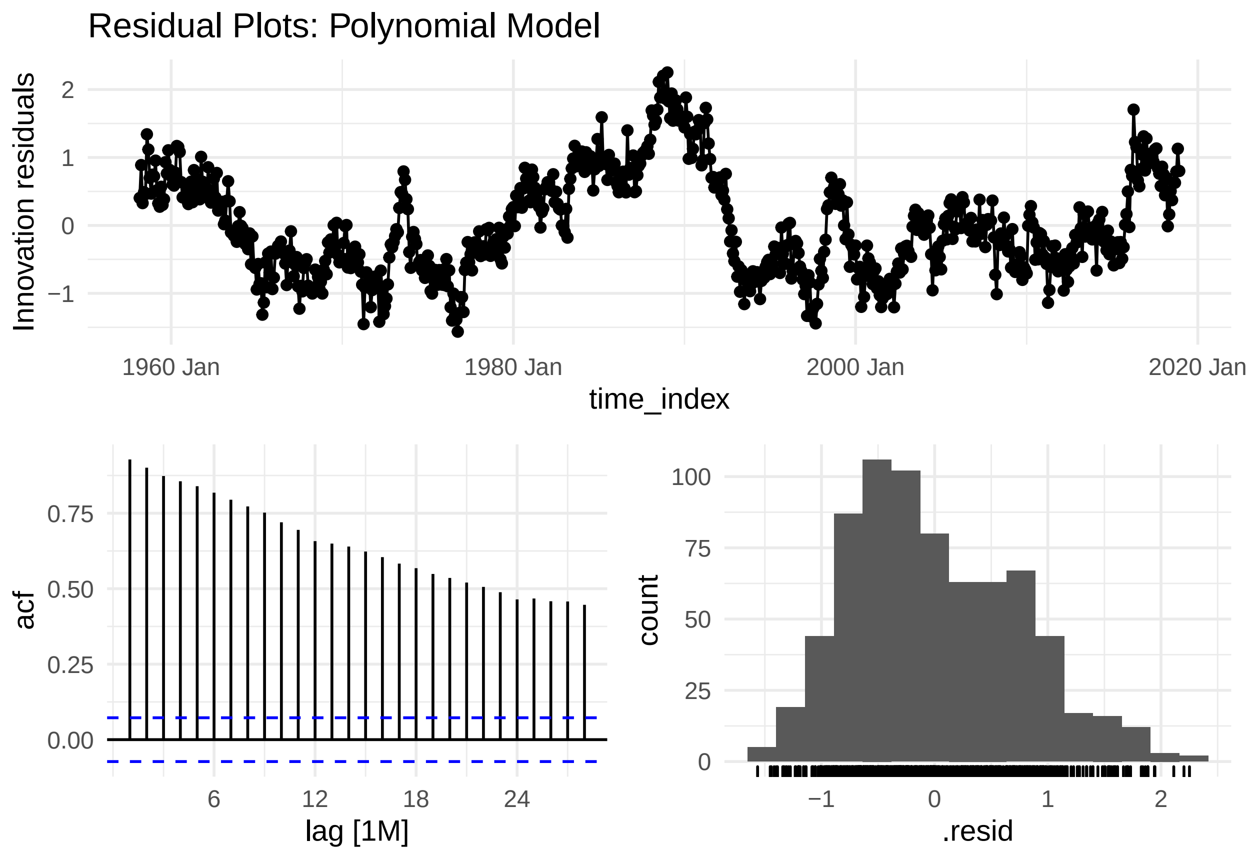


The ARIMA hyper parameters that were selected include the following: ARIMA(1,2,1)(4,0,0)[12]. Using these values, the residual plots appear to meet the conditin of constant variance and a mean near zero. The test results also past the ljung_box statistical test. The ACF had some extreme values at the 2nd and 24th time lag, which seems a bit strange. The forecast estimates appear strange when juxtaposed next to the monthly data inclusive of the seasonal trends but that is because the underlying seasonality has been removed.

```
# Build model
train_s %>%
  model(TSLM(season_adjust ~ trend() + I(trend()^2) + I(trend()^3))) -> fit_poly

# Print model
report(fit_poly)
```

```
## Series: season_adjust
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5643 -0.5742 -0.1199  0.5283  2.2516
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  3.142e+02  1.103e-01 2849.568   <2e-16 ***
## trend()      6.772e-02  1.305e-03   51.875   <2e-16 ***
## I(trend()^2) 8.074e-05  4.148e-06   19.463   <2e-16 ***
## I(trend()^3) 6.554e-09  3.730e-09    1.757   0.0793 .
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.741 on 726 degrees of freedom
## Multiple R-squared: 0.9993,  Adjusted R-squared: 0.9993
## F-statistic: 3.375e+05 on 3 and 726 DF, p-value: < 2.22e-16
```



The residual diagnostic plots do no produce promising results. The residual time series does not have a constant variance or mean, and it is not clear that the mean is near zero. All values within the ACF exceed the threshold values and the histogram of the residuals appears to be skewed to the right.

As a result, we have decided to go with the ARIMA model because it adjusts for the underlying seasonality, which we may need to adjust going forward, all of its residual values are are within the acceptable bands of the ACF, and it is easier to explain the model results to a non-technical audience. The ARIMA model includes the following features: Model: ARIMA(1,1,1)(2,1,1)[12]

4.6 (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?

```
model.bic3 <- co2_present %>%
  model(ARIMA(average ~ 0 + pdq(1, 1, 1) + PDQ(2, 1, 1),
                     ic="bic"))



model.bic3 %>%
  report()
```

```
## Series: average
## Model: ARIMA(1,1,1)(2,1,1)[12]
##
## Coefficients:
##          ar1      ma1    sar1     sar2     sma1
##       0.1954  -0.5544  0.0057  -0.0219  -0.8583
## s.e.  0.0954   0.0818  0.0432   0.0417   0.0236
##
## sigma^2 estimated as 0.09756:  log likelihood=-188.22
## AIC=388.44   AICc=388.56   BIC=416.09
```

```
# Prediction from 2021 to 2121
fc2_100 <- forecast(model.bic3, h=1224) %>%
  hilo() %>%
  unpack_hilo(c(`80%`, `95%`))

fc2_100 %>%
  filter(.mean >= 420 & .mean < 421) -> fc2_420

# 420 PPM levels for the first and last time
fc2_420_first <- head(fc2_420, n =1)
fc2_420_last  <- tail(fc2_420, n =1)

# Get values for 420 ppm predictions
fc2_420_first_time  <- fc2_420_first$time_index
fc2_420_first_lower <- round(fc2_420_first$`95%_lower`, 2)
fc2_420_first_upper <- round(fc2_420_first$`95%_upper`, 2)
fc2_420_first_mean  <- round(fc2_420_first$.mean, 2)

fc2_420_last_time  <- fc2_420_last$time_index
fc2_420_last_lower <- round(fc2_420_last$`95%_lower`, 2)
fc2_420_last_upper <- round(fc2_420_last$`95%_upper`, 2)
fc2_420_last_mean  <- round(fc2_420_last$.mean, 2)
```

Based on the findings from our model, we predict that CO2 will cross the 420 ppm threshold for the first time on 2023 Jan with an expected value of 420.42 with a 95% confidence interval between 418.48 - 422.36. Our model also predicts that the last time CO2 will be between the 420 and 421 ppm threshold will be on 2024 Oct with an expected value of 420.77 with a 95% confidence interval between 417.92 - 423.62.

```
fc2_100 %>%
  filter(.mean >= 500 & .mean < 501) -> fc2_500
```

```r
# 420 PPM levels for the first and last time
fc2_500_first <- head(fc2_500, n =1)
fc2_500_last  <- tail(fc2_500, n =1)

# Get values for 420 ppm predictions
fc2_500_first_time  <- fc2_500_first$time_index
fc2_500_first_lower <- round(fc2_500_first$`95%_lower`, 2)
fc2_500_first_upper <- round(fc2_500_first$`95%_upper`, 2)
fc2_500_first_mean  <- round(fc2_500_first$.mean, 2)

fc2_500_last_time  <- fc2_420_last$time_index
fc2_500_last_lower <- round(fc2_500_last$`95%_lower`, 2)
fc2_500_last_upper <- round(fc2_500_last$`95%_upper`, 2)
fc2_500_last_mean  <- round(fc2_500_last$.mean, 2)
```

Based on the findings from our model, we predict that CO2 will cross the 500 ppm threshold for the first time on 2056 Apr with an expected value of 500.96 with a 95% confidence interval between 475.25 - 526.67. Our model also predicts that the last time CO2 will be between the 500 and 501 ppm threshold will be on 2024 Oct with an expected value of 500.68 with a 95% confidence interval between 472.63 - 528.72.

```r
# Atmospheric CO2 levels in the year 2122
fc_2122  <- tail(fc2_100, n =1)

# Carbon values in 2122
fc_2122_lower <- round(fc_2122$`95%_lower`, 2)
fc_2122_upper <- round(fc_2122$`95%_upper`, 2)
fc_2122_mean  <- round(fc_2122$.mean, 2)
```

'By the end of 2122, our model predicts that CO2 will be 654.1 ppm with a 95% confidence interval between 546.98 - 761.22. Although we are not very confident in the point estimate for the average CO2 emissions at the end of 2022, we are fairly confident (95%!) that average CO2 emissions will be between our confidence interval of 546.98 - 761.22.