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

Brittany Dougall, Steve Hall, Prabhu Narsina, and Edward Salinas

Instructions (Please Read Carefully):

- Submit by the due date. **Late submissions will not be accepted**
- No page limit, but be reasonable
- Do not modify fontsize, margin or line-spacing settings
- One student from each group should submit the lab to their student github repo by the deadline
- Submit two files:

  1. A pdf file that details your answers. Include all R code used to produce the answers
  2. The R markdown (Rmd) file used to produce the pdf file

  The assignment will not be graded unless **both** files are submitted
- Name your files to include all group members names. For example, if the students' names are Stan Cartman and Kenny Kyle, name your files as follows:
  - `StanCartman_KennyKyle_Lab2.Rmd`
  - `StanCartman_KennyKyle_Lab2.pdf`
- Although it sounds obvious, please write your name on page 1 of your pdf and Rmd files
- All answers should include a detailed narrative; make sure that your audience can easily follow the logic of your analysis. All steps used in modelling must be clearly shown and explained; do not simply 'output dump' the results of code without explanation
- If you use libraries and functions for statistical modeling that we have not covered in this course, you must provide an explanation of why such libraries and functions are used and reference the library documentation
- For mathematical formulae, type them in your R markdown file. Do not e.g. write them on a piece of paper, snap a photo, and use the image file
- Incorrectly following submission instructions results in deduction of grades
- Students are expected to act with regard to UC Berkeley Academic Integrity.

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 attributed this pattern to varying rates of photosynthesis throughout the year, caused by differences 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. He soon observed a trend increase carbon dioxide levels in addition to the seasonal cycle, attributable to growth in global rates of fossil fuel combustion. Measurement of this trend at Mauna Loa has continued to the present.

The `co2` data set in R's `datasets` package (automatically loaded with base R) is a monthly time series of atmospheric carbon dioxide concentrations measured in ppm (parts per million) at the Mauna Loa Observatory from 1959 to 1997. The curve graphed by this data is known as the 'Keeling Curve'.

Part 1 (3 points)

Conduct a comprehensive Exploratory Data Analysis on the `co2` series. This should include (without being limited to) a thorough investigation of the trend, seasonal and irregular elements.

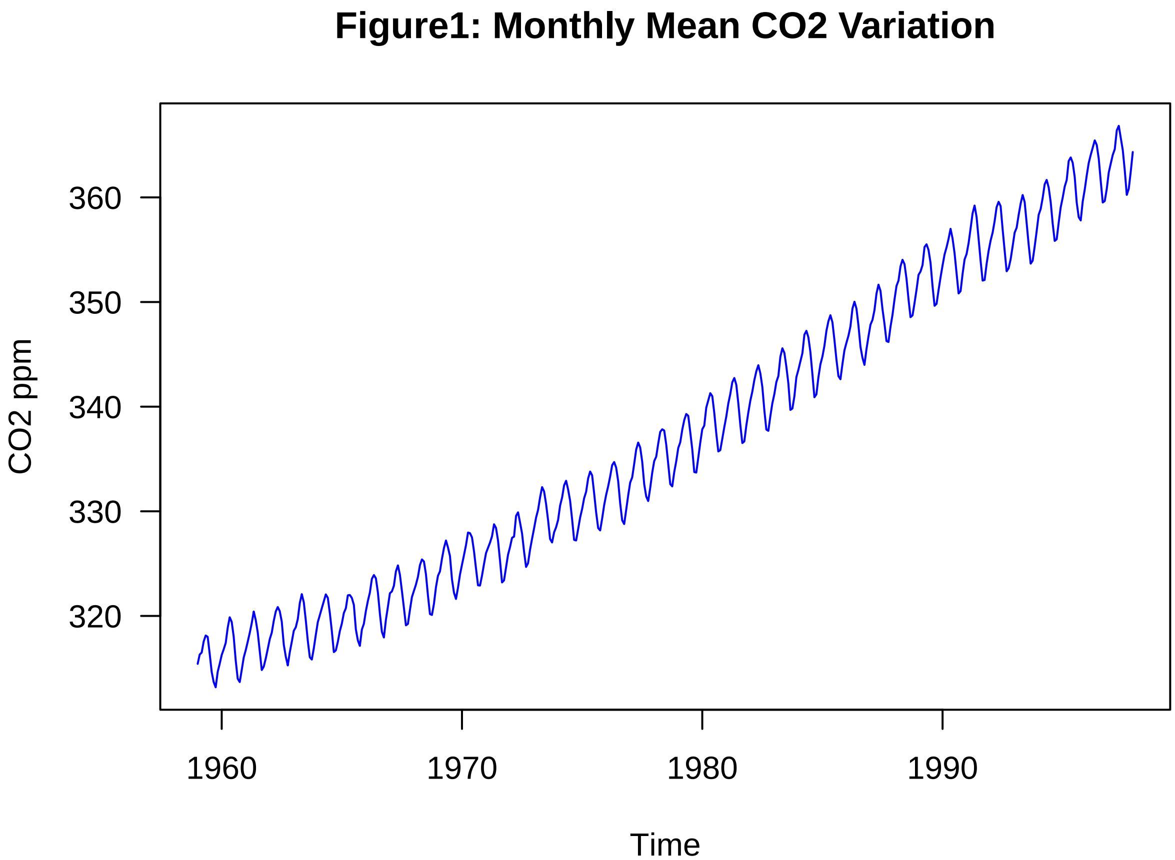
```
opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE,
    warning = FALSE, message = FALSE)

str(co2)
```

```
##  Time-Series [1:468] from 1959 to 1998: 315 316 316 318 318 ...
```

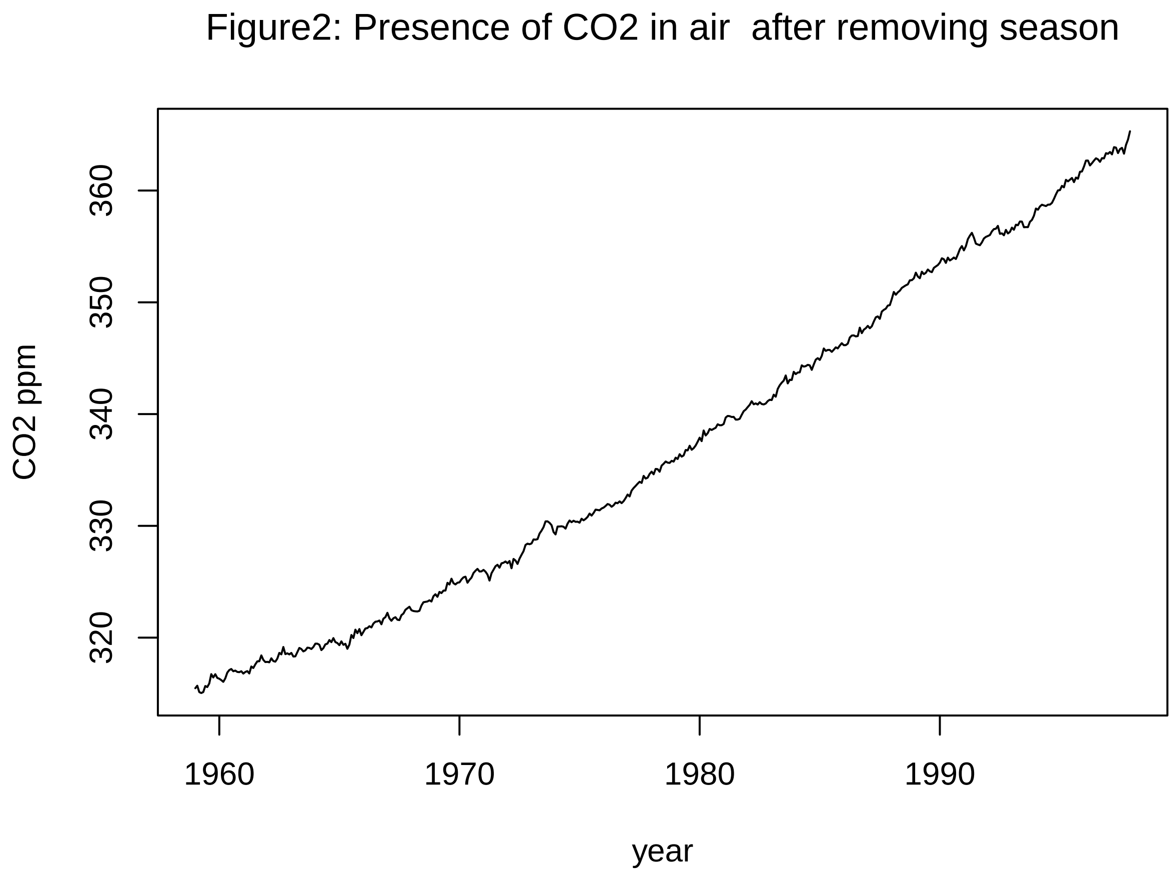
```
summary(co2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   313.2   323.5   335.2   337.1   350.3   366.8
```

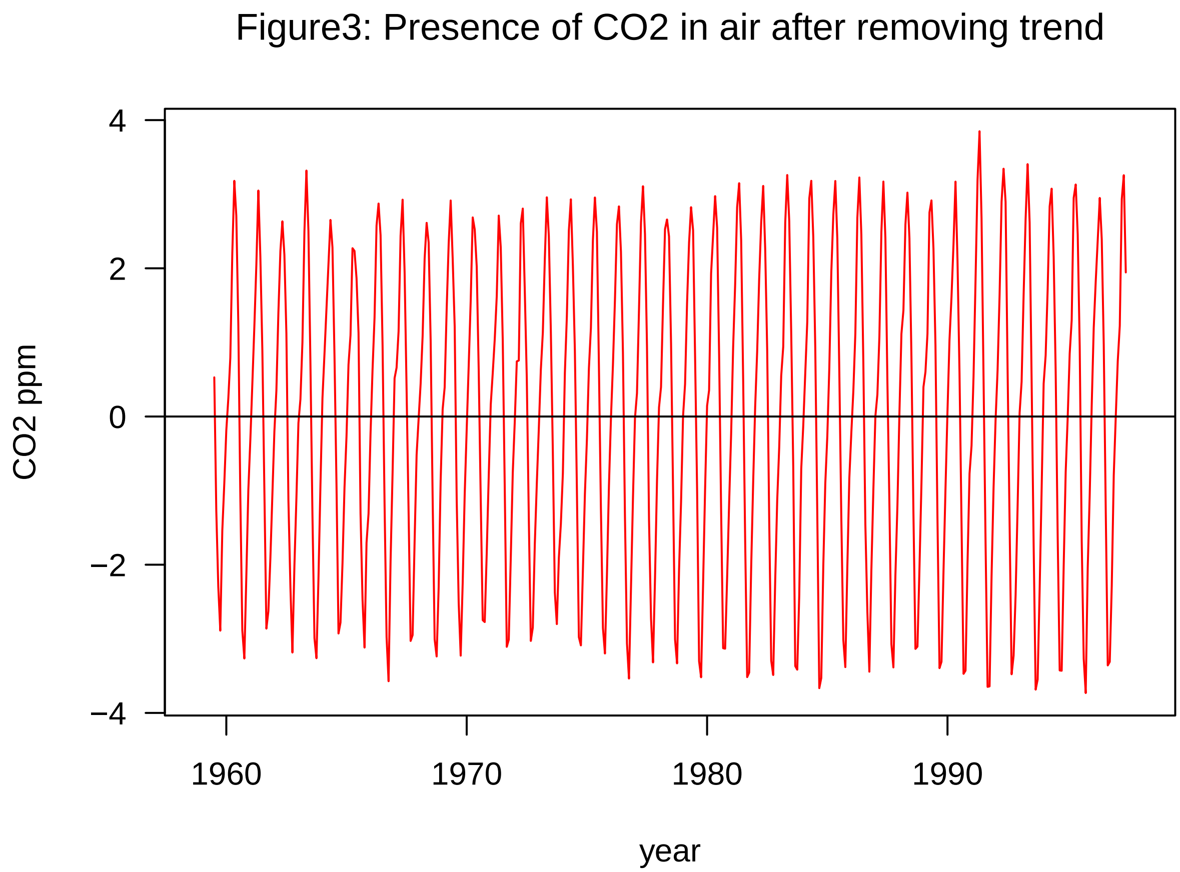
```
co2.decompose = decompose(co2)
co2.diff = diff(co2, 1)
co2.seasdiff = diff(co2, lag = 12)
co2.bothdiff = diff(co2.diff, lag = 12)

co2.deseasoned = co2 - co2.decompose$seasonal
co2.detrended = co2 - co2.decompose$trend

plot(co2, main = "Figure1: Monthly Mean CO2 Variation", ylab = expression("CO2 ppm"),
    col = "blue", las = 1)
```



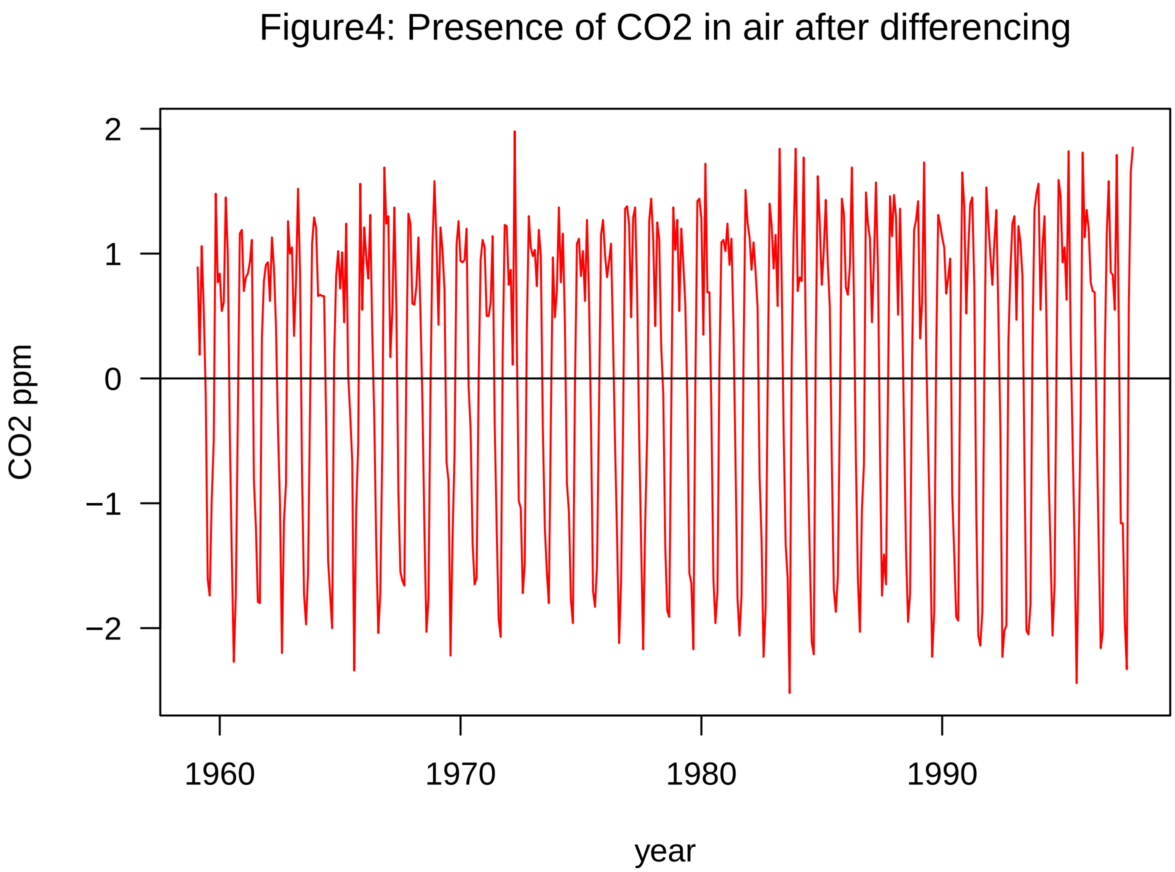

```
plot(co2.deseasoned, main = expression("Figure2: Presence of CO2 in air  after removing season'
    xlab = "year", ylab = "CO2 ppm")
```




```r
plot(co2.detrended, main = expression("Figure3: Presence of CO2 in air after removing trend"),
    xlab = "year", ylab = expression("CO2 ppm"), col = "red",
    las = 1)

abline(h = 0)
```

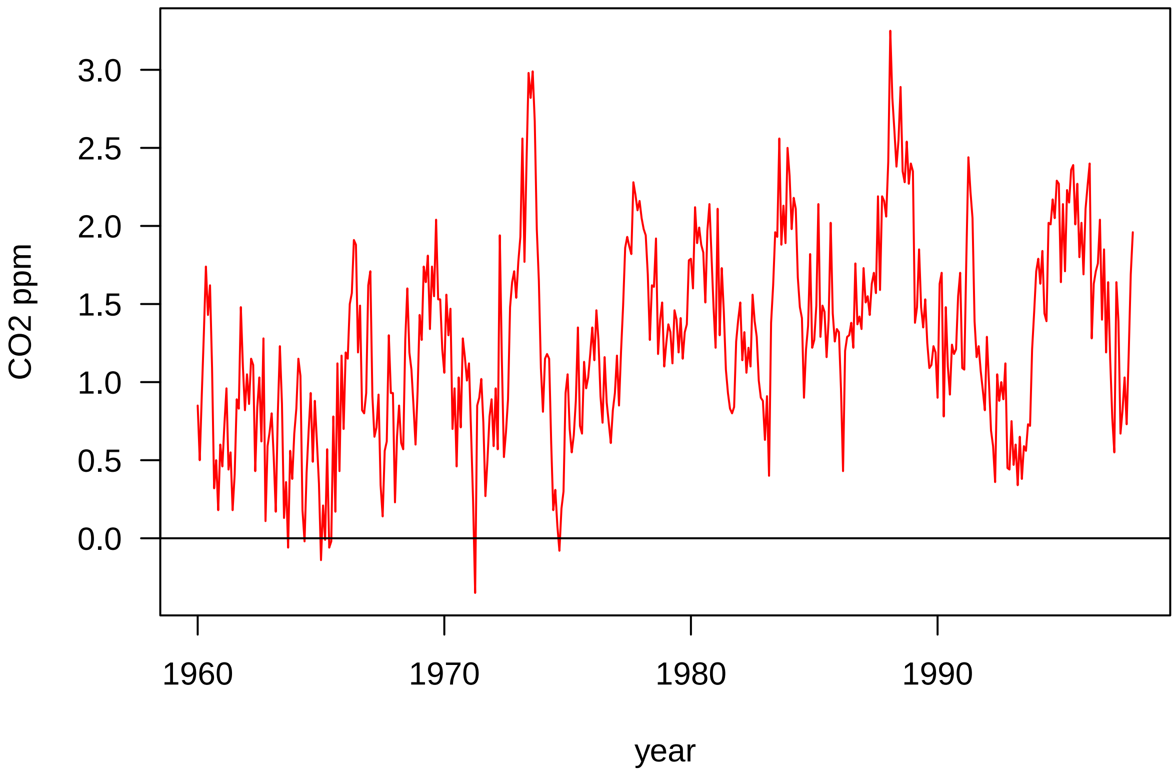



```
plot(co2.diff, main = expression("Figure4: Presence of CO2 in air after differencing"),
    xlab = "year", ylab = expression("CO2 ppm"), col = "red",
    las = 1)

abline(h = 0)
```

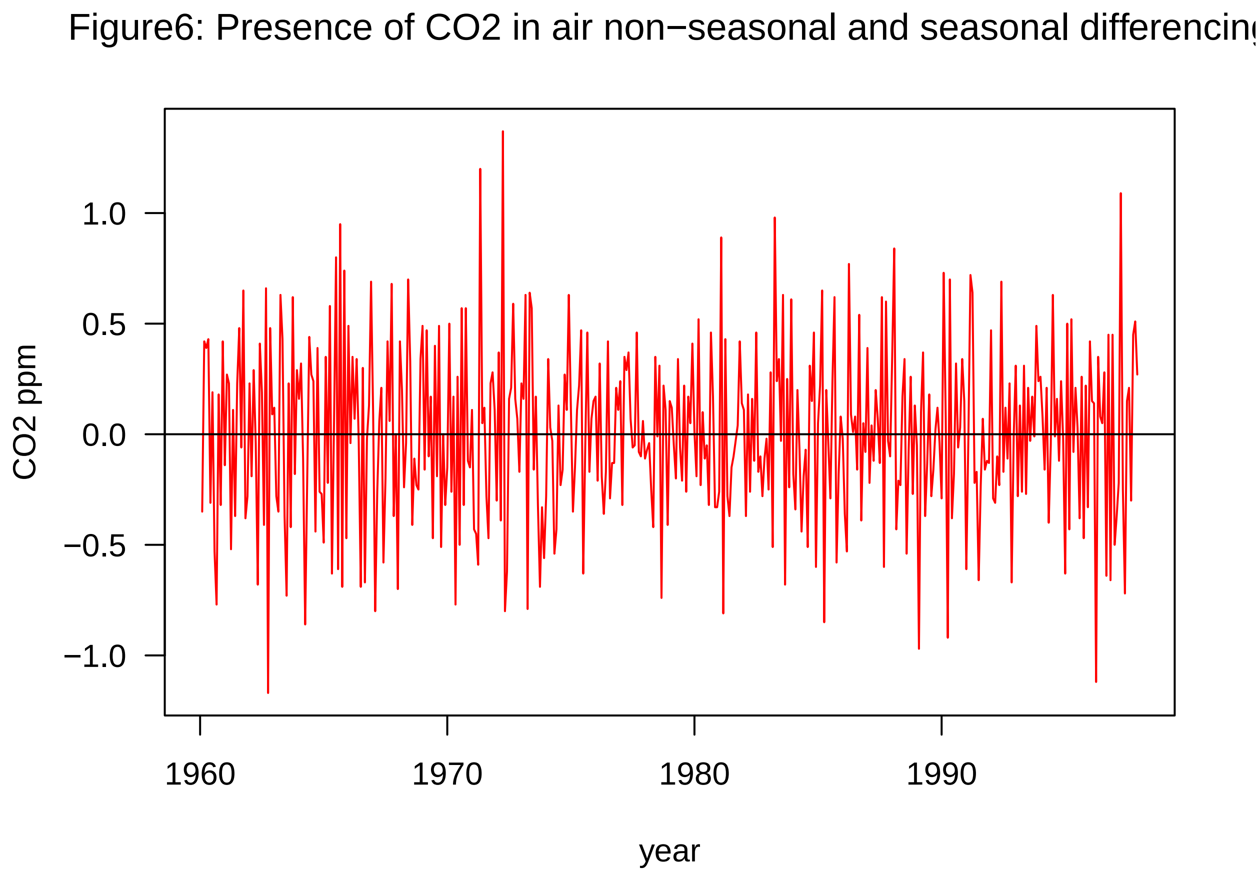



```
plot(co2.seasdiff, main = expression("Figure5: Presence of CO2 in air after seasonal differenci
     xlab = "year", ylab = expression("CO2 ppm"), col = "red",
     las = 1)
abline(h = 0)
```

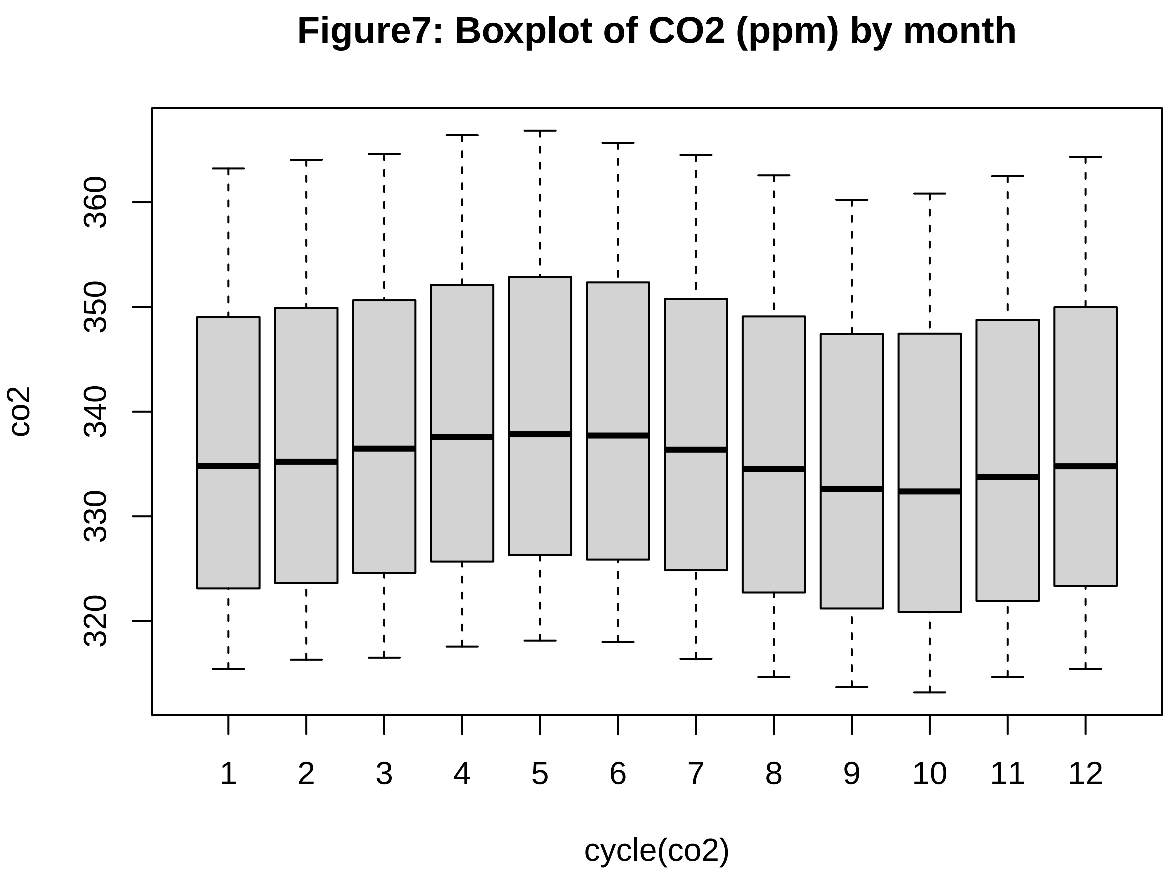
Figure5: Presence of CO2 in air after seasonal differencing




```
plot(co2.bothdiff, main = expression("Figure6: Presence of CO2 in air non-seasonal and seasonal
     xlab = "year", ylab = expression("CO2 ppm"), col = "red",
     las = 1)
abline(h = 0)
```

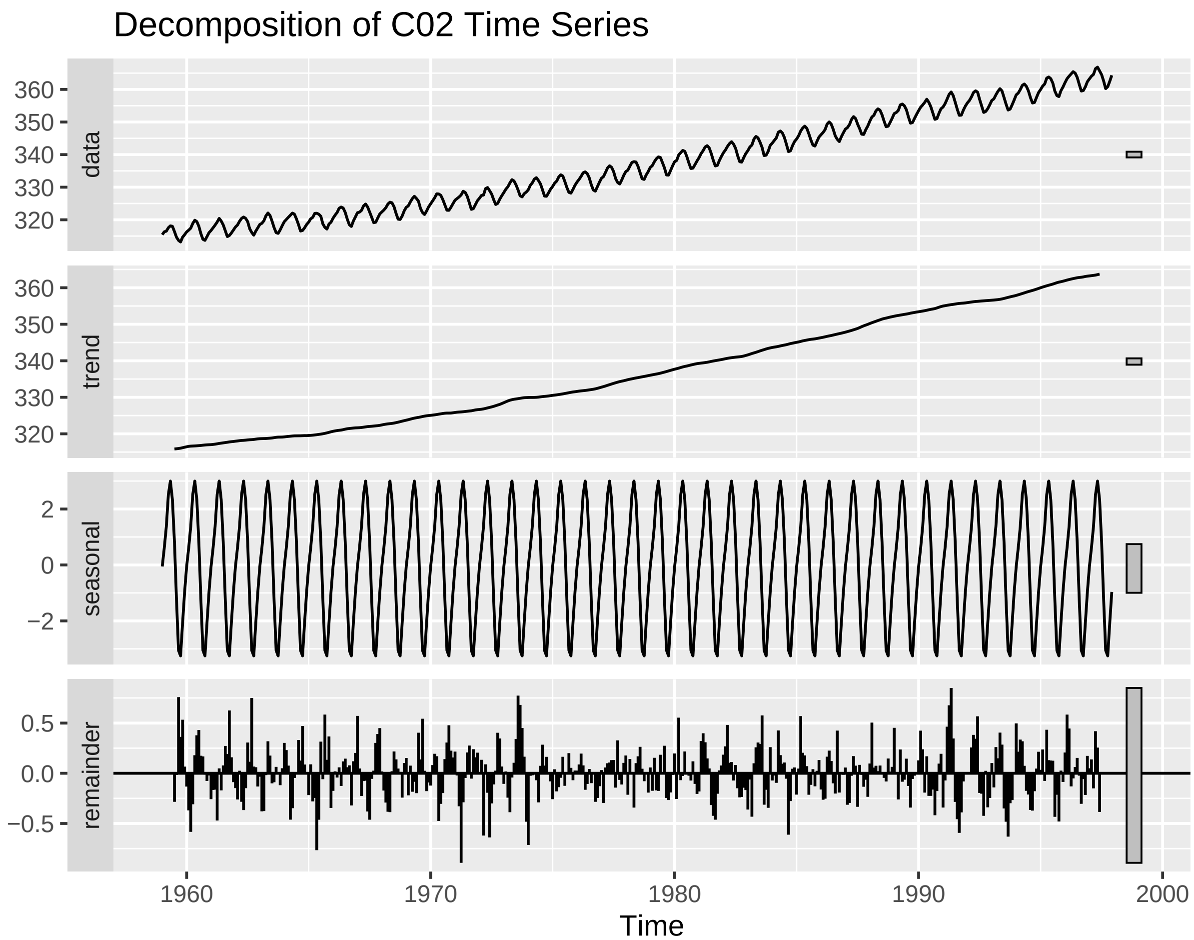



```
boxplot(co2 ~ cycle(co2), main = "Figure7: Boxplot of CO2 (ppm) by month")
```



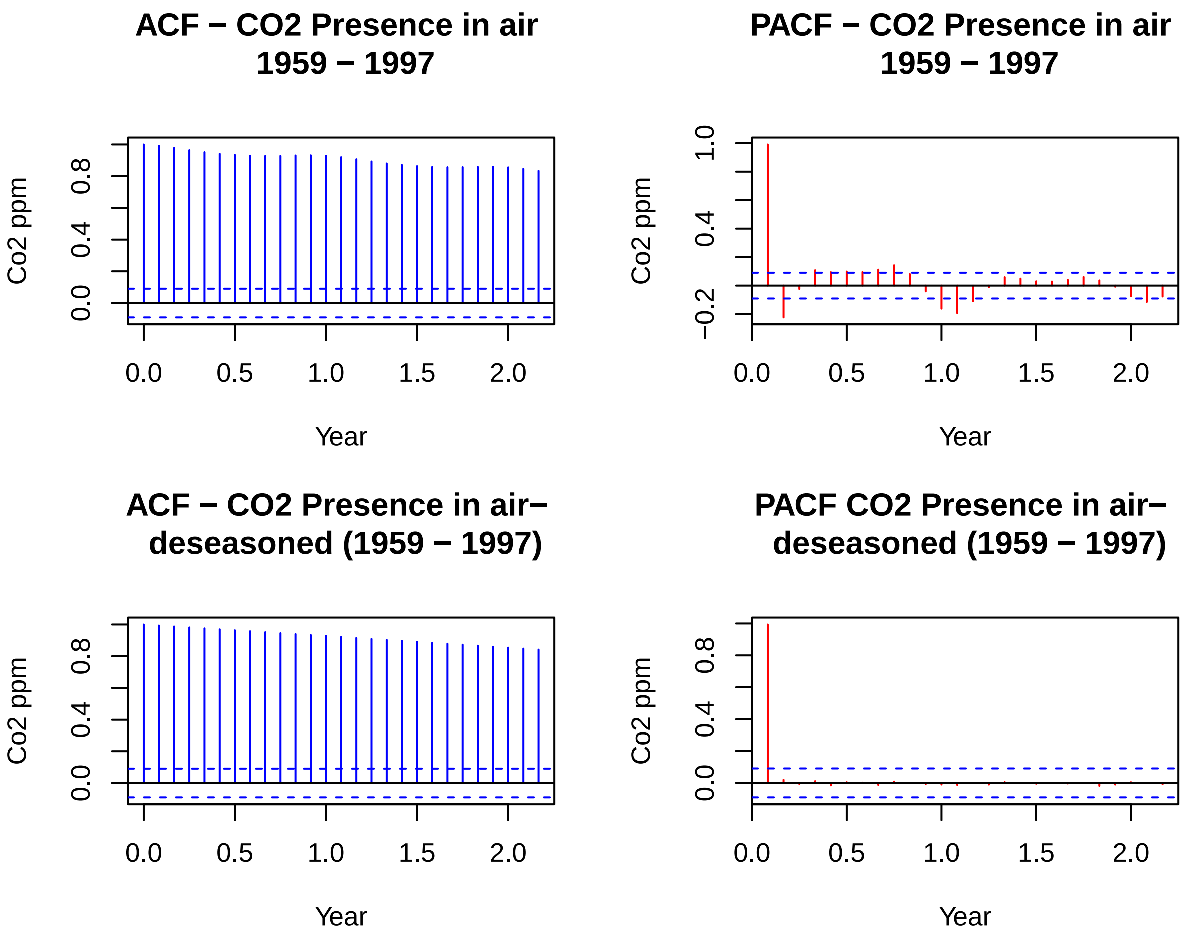
Data provided has CO2 presence in the air (parts per million) in monthly time series format from 1959 to 1998.
From Figure1: The time series plot of the mean of co2 presence in the air indicates a clear trend and seasonal effect. We also observe that the variance is constant over time, which suggests no need for transformation.
From Figure2: We see a clear upward trend in the mean of the presence of Co2 in the air.
From Figure3: Co2 presence in the air after removing the trend component from the time series indicates the persistent yearly seasonal effect. The de-trended series also appears to be mean and variance stationary.
From Figure4: Trend is abstracted by taking the first difference of the time series. It suggests we use ARIMA with integration/difference of 1.
From Figure5: Seasonality absent after applying difference of 12 lags for the season. We still see trends present.
From Figure6: Seasonality and trend are absent after one lag and 12 lags for the season. After seasonal and non-seasonal differencing, the series appears stationary with a relatively constant mean. From Figure7: Seasonality is apparent across months. For instance, notice the increase in CO2 from the first quarter to the end of the second quarter (month 6), and then the decline in CO2 until October.

```
autoplot(co2.decompose, main = "Decomposition of CO2 Time Series")
```



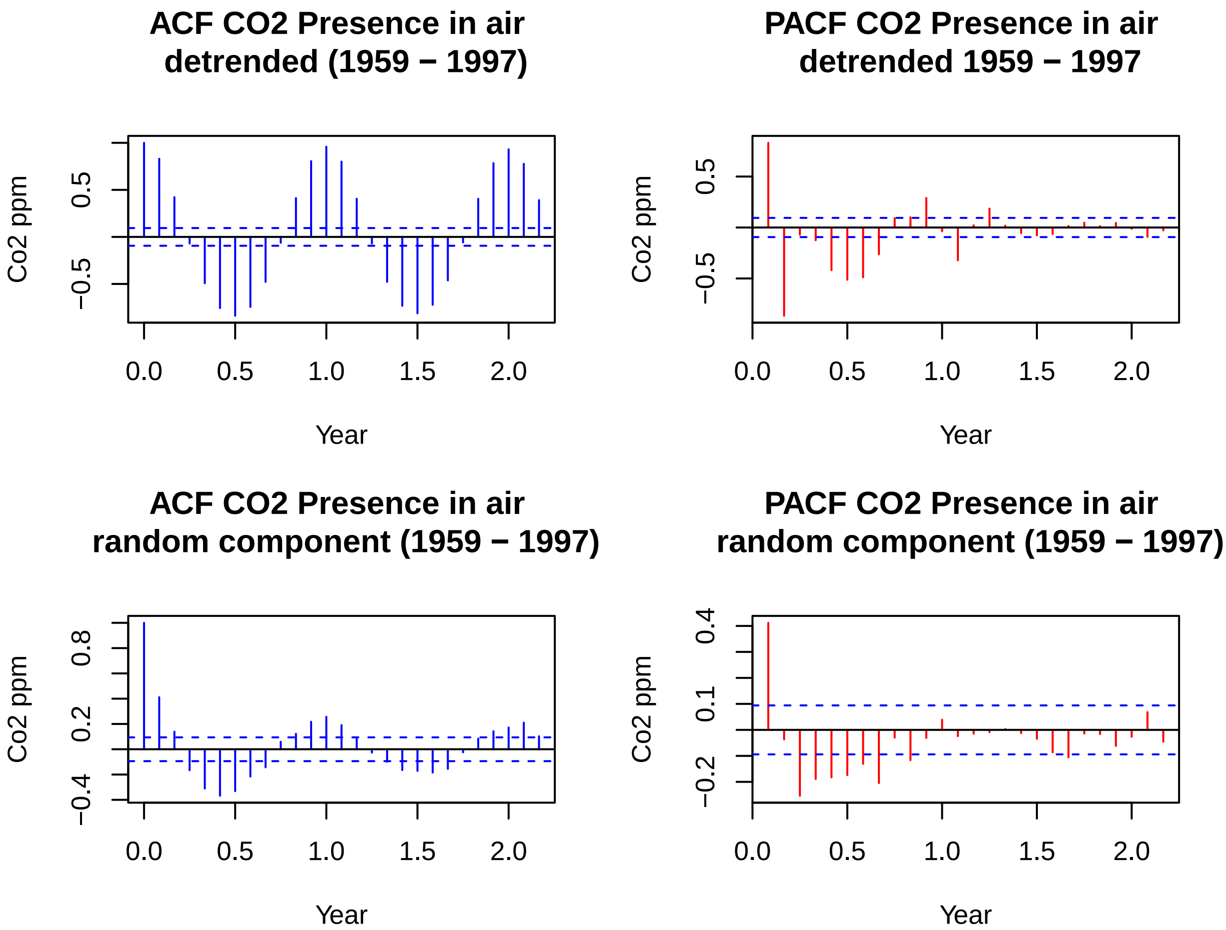

```
plot.acf.alldata = acf(co2, plot = FALSE)
plot.pacf.alldata = pacf(co2, plot = FALSE)

plot.acf.deseasoned = acf(co2.deseasoned, plot = FALSE)
plot.pacf.deseasoned = pacf(co2.deseasoned, plot = FALSE)

plot.acf.detrended = acf(window(co2.detrended, start = c(1960),
    end = c(1996)), plot = FALSE)
plot.pacf.detrended = pacf(window(co2.detrended, start = c(1960),
    end = c(1996)), plot = FALSE)

plot.acf.residual = acf(window(co2.decompose$random, start = c(1960),
    end = c(1996)), plot = FALSE)
plot.pacf.residual = pacf(window(co2.decompose$random, start = c(1960),
    end = c(1996)), plot = FALSE)

plot.acf.diff = acf(co2.diff, plot = FALSE)
plot.pacf.diff = pacf(co2.diff, plot = FALSE)

plot.acf.seasondiff = acf(co2.seasdiff, plot = FALSE)
plot.pacf.seasondiff = pacf(co2.seasdiff, plot = FALSE)
```

```r
plot.acf.bothdiff = acf(co2.bothdiff, plot = FALSE)
plot.pacf.bothdiff = pacf(co2.bothdiff, plot = FALSE)

par(mfrow = c(2, 2))
plot(plot.acf.alldata, main = "ACF - CO2 Presence in air \n 1959 - 1997",
    xlab = "Year", ylab = "Co2 ppm", col = "blue", cex.main = 0.5)
plot(plot.pacf.alldata, main = "PACF - CO2 Presence in air \n 1959 - 1997",
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)

plot(plot.acf.deseasoned, main = "ACF - CO2 Presence in air- \n deseasoned (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
plot(plot.pacf.deseasoned, main = "PACF CO2 Presence in air- \n deseasoned (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)
```

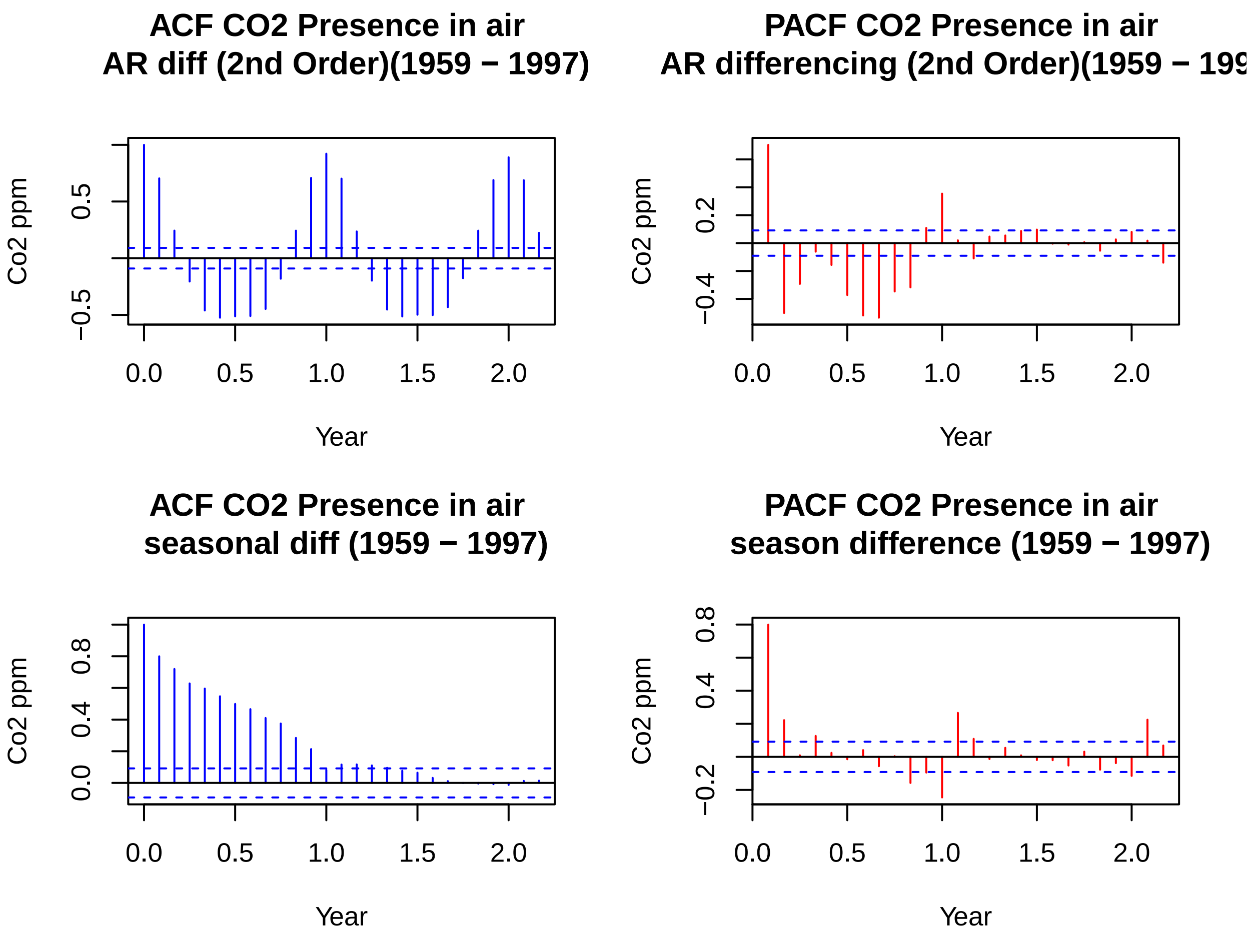



```r
plot(plot.acf.detrended, main = "ACF CO2 Presence in air \n detrended (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
plot(plot.pacf.detrended, main = "PACF CO2 Presence in air \n detrended 1959 - 1997",
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)

plot(plot.acf.residual, main = "ACF CO2 Presence in air \n random component (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
```

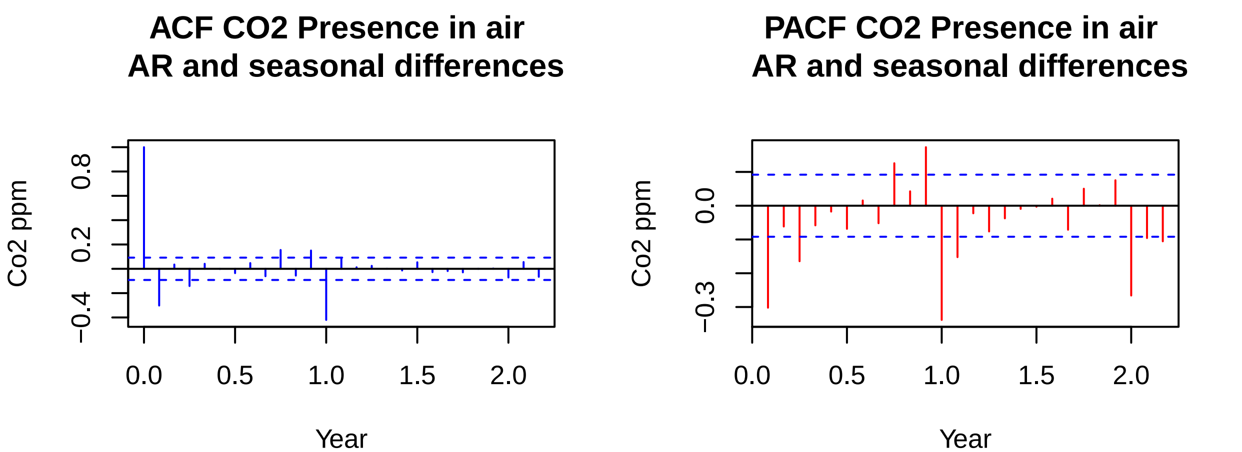
```
plot(plot.pacf.residual, main = "PACF CO2 Presence in air \n random component (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)
```




```
plot(plot.acf.diff, main = "ACF CO2 Presence in air \n AR diff (2nd Order)(1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
plot(plot.pacf.diff, main = "PACF CO2 Presence in air \n AR differencing (2nd Order)(1959 - 19
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)

plot(plot.acf.seasondiff, main = "ACF CO2 Presence in air \n seasonal diff (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
plot(plot.pacf.seasondiff, main = "PACF CO2 Presence in air \n season difference (1959 - 1997)
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)
```




```r
plot(plot.acf.bothdiff, main = "ACF CO2 Presence in air \n AR and seasonal differences",
     xlab = "Year", ylab = "Co2 ppm", col = "blue")
plot(plot.pacf.bothdiff, main = "PACF CO2 Presence in air \n AR and seasonal differences",
     xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)
```



Decomposition graph confirms the findings from EDA that trend and seasonality are present in the time series.

Above ACF and PACF graph shows for different adjustments of time series: 1) original series 2) de-seasoned 3) de-trended 4) random component of time series 5) One-period differenced for trend

6) One-period difference and seasonal difference time series. We noted a few observations below:
* PACF graph shows autocorrelation dying off after first lag after de-seasoned. This suggests to use only 1st order autoregressive model. This also suggests taking the first seasonal difference is important. * ACF graph shows clear seasonal effect after removing trend
* ACF graph after performing auto regressive (AR) and seasonal differences looks closer to white noise ACF graph. Significant correlations at a 1 year lag suggests the need for a MA term. * In the PACF with AR and seasonal differences plot, the significant negative correlations at 1 and 2 year lags suggest we should explore using a seasonal AR term in our model.

Part 2 (3 points)

Fit a linear time trend model to the `co2` series, and examine the characteristics of the residuals. Compare this to a higher-order polynomial 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 up to the present.

Linear Time Trend Model

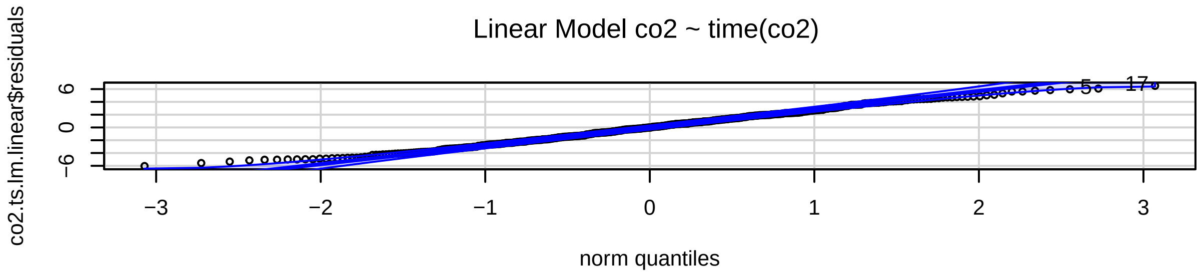
```
# First fit a linear time trend model
par(mfrow = c(3, 1))
co2.ts.lm.linear = lm(co2 ~ time(co2))
summary(co2.ts.lm.linear)
```

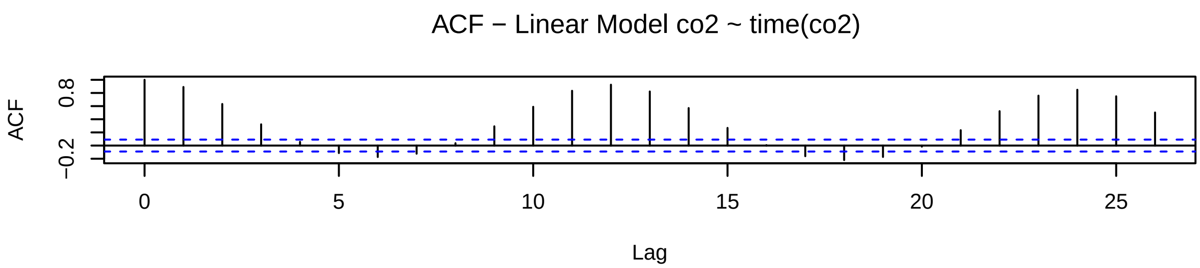
```
##
## Call:
## lm(formula = co2 ~ time(co2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0399 -1.9476 -0.0017  1.9113  6.5149
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.250e+03  2.127e+01  -105.8   <2e-16 ***
## time(co2)    1.308e+00  1.075e-02   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: < 2.2e-16
```

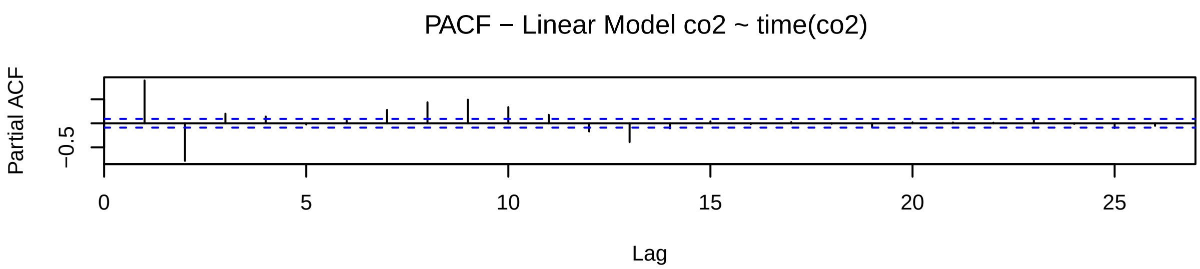
```
qqPlot(co2.ts.lm.linear$residuals, main = expression("Linear Model co2 ~ time(co2) "))
```

```
## [1] 17  5
```

```
plt.acf = acf(co2.ts.lm.linear$residuals, plot = FALSE)
plt.pacf = pacf(co2.ts.lm.linear$residuals, plot = FALSE)
plot(plt.acf, main = expression("ACF - Linear Model co2 ~ time(co2) "))
plot(plt.pacf, main = expression("PACF - Linear Model co2 ~ time(co2) "))
```








```
Box.test(co2.ts.lm.linear$residuals, type = "Ljung-Box")
```

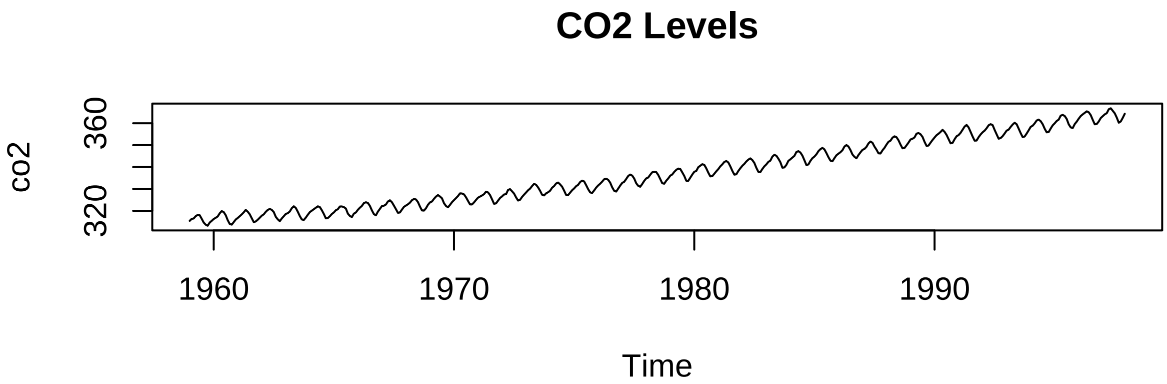
```
##
##  Box-Ljung test
##
## data:  co2.ts.lm.linear$residuals
## X-squared = 373.94, df = 1, p-value < 2.2e-16
```

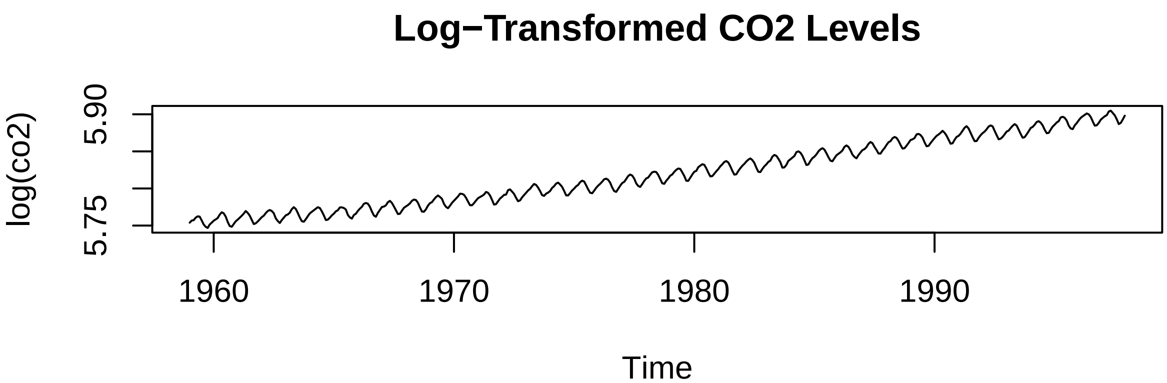
After fitting a time-trend model, we performed several checks to assess model fit. As seen above, the plot of the residuals against the normal distribution shows skewing in the tails, suggesting that the linear model residuals are not normally distributed.

The ACF and PACF plots show evidence of autocorrelation in the residuals. This suggests poor model fit and clustering of errors, which would underestimate standard errors of the coefficients. This latter finding is supported by the results of the Ljung-Box test, which has a small p-value (< 0.05), meaning that we can reject with 95% confidence the null hypothesis that the residuals are independently distributed.

Log Transformation of CO2 Levels

```
par(mfrow = c(2, 1))
plot(co2, main = "CO2 Levels")
plot(log(co2), main = "Log-Transformed CO2 Levels")
```





At first glance, the log-transformed series appears very similar to the raw series. Also, the raw monthly CO2 series does not appear to exhibit increasing variance through time, which suggests that a log-transformation is not necessary. We will continue to fit a log-transformed time trend model.

```
log.fit <- lm(log(co2) ~ time(co2) + I(time(co2)^2))
summary(log.fit)
```

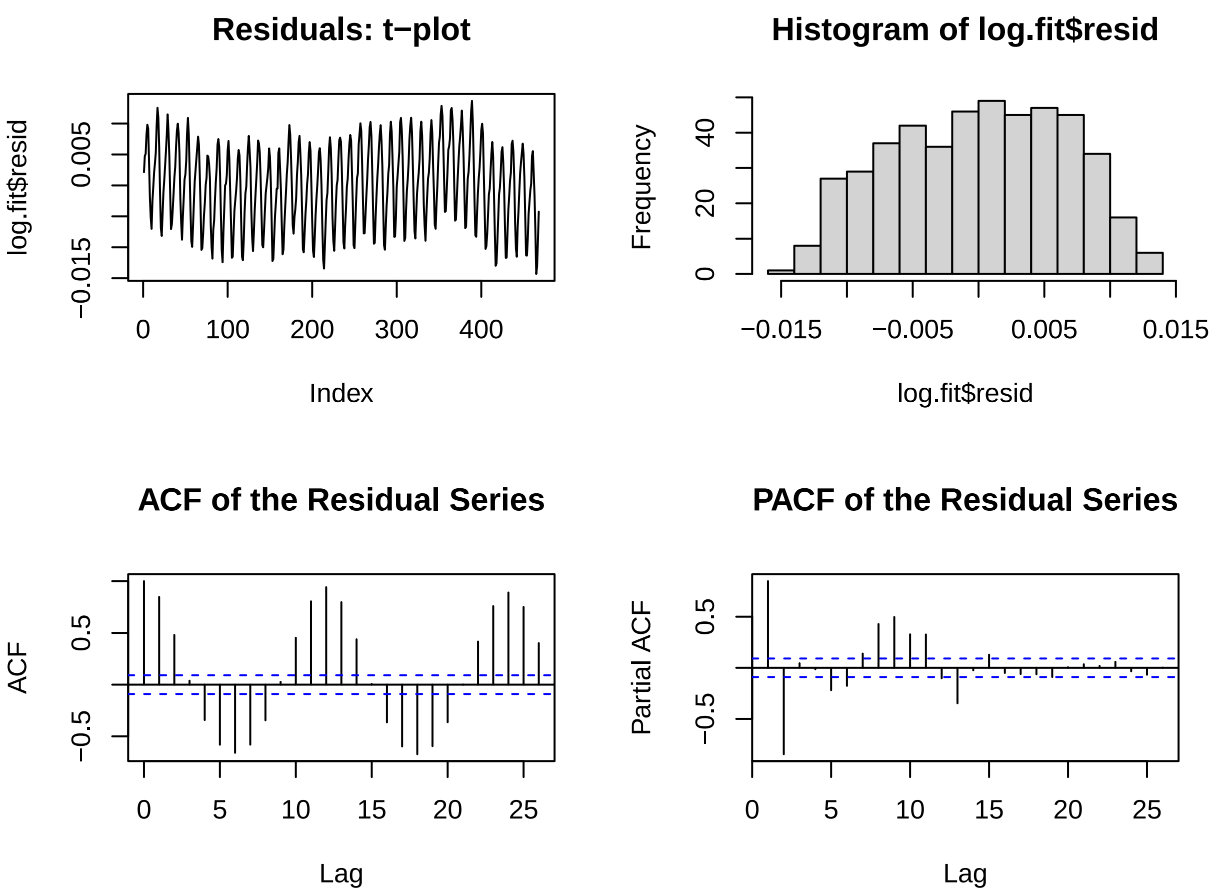
```
##
## Call:
## lm(formula = log(co2) ~ time(co2) + I(time(co2)^2))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0143052 -0.0050832  0.0005277  0.0052757  0.0136508
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.193e+02  1.036e+01   11.52   <2e-16 ***
## time(co2)      -1.186e-01  1.047e-02  -11.32   <2e-16 ***
## I(time(co2)^2)  3.094e-05  2.646e-06   11.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.00649 on 465 degrees of freedom
## Multiple R-squared:  0.9786, Adjusted R-squared:  0.9785
## F-statistic: 1.061e+04 on 2 and 465 DF,  p-value: < 2.2e-16
```

```
# Residual Diagnostics
summary(log.fit$resid)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0143052 -0.0050832  0.0005277  0.0000000  0.0052757  0.0136508
```

```
par(mfrow = c(2, 2))
plot(log.fit$resid, type = "l", main = "Residuals: t-plot")
hist(log.fit$resid)
acf(log.fit$resid, main = "ACF of the Residual Series")
pacf(log.fit$resid, main = "PACF of the Residual Series")
```




```
Box.test(residuals(log.fit), lag = 12, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  residuals(log.fit)
## X-squared = 1925.9, df = 12, p-value < 2.2e-16
```

The residuals are highly correlated and show evidence of seasonality in the ACF plot. The Ljung-

Box test supports the ACF plot and permits us to reject the null hypothesis that the series is independently distributed in favor of the alternative hypothesis that the series exhibits serial correlation. As mentioned earlier, since the variance appears constant through time, we will not log-transform the series going forward.

Seasonal Time-Trend Model

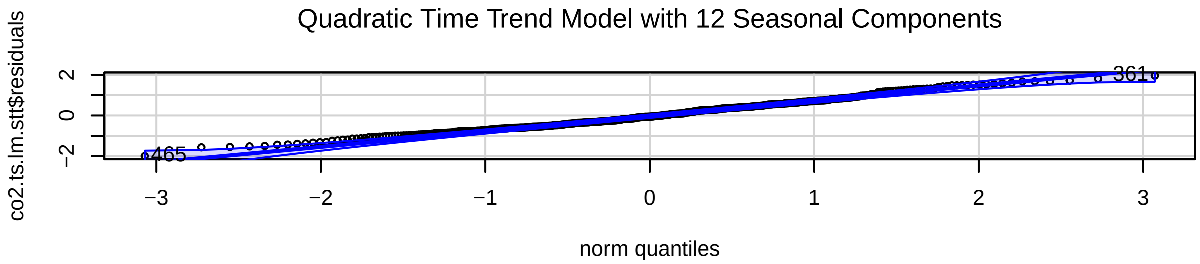
```
# Add seasonal dummy to data.frame
co2.df = data.frame(ppm = c(co2), time = c(time(co2)))
co2.df$season = as.factor(cycle(co2))

par(mfrow = c(3, 1))
co2.ts.lm.stt = lm(ppm ~ time + I(time^2) + season, data = co2.df)
summary(co2.ts.lm.stt)
```

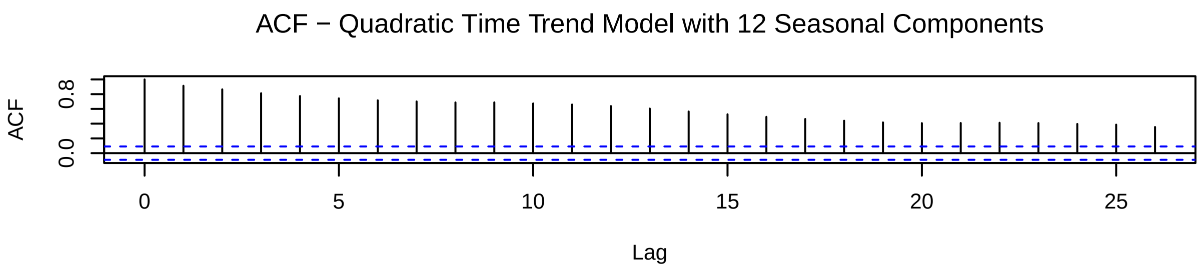
```
##
## Call:
## lm(formula = ppm ~ time + I(time^2) + season, data = co2.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.99478 -0.54468 -0.06017  0.47265  1.95480
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.771e+04  1.156e+03  41.289  < 2e-16 ***
## time        -4.920e+01  1.168e+00 -42.120  < 2e-16 ***
## I(time^2)    1.277e-02  2.952e-04  43.242  < 2e-16 ***
## season2      6.642e-01  1.640e-01   4.051 5.99e-05 ***
## season3      1.407e+00  1.640e-01   8.582  < 2e-16 ***
## season4      2.538e+00  1.640e-01  15.480  < 2e-16 ***
## season5      3.017e+00  1.640e-01  18.400  < 2e-16 ***
## season6      2.354e+00  1.640e-01  14.357  < 2e-16 ***
## season7      8.331e-01  1.640e-01   5.081 5.50e-07 ***
## season8     -1.235e+00  1.640e-01  -7.531 2.75e-13 ***
## season9     -3.059e+00  1.640e-01 -18.659  < 2e-16 ***
## season10    -3.243e+00  1.640e-01 -19.777  < 2e-16 ***
## season11    -2.054e+00  1.640e-01 -12.526  < 2e-16 ***
## season12    -9.374e-01  1.640e-01  -5.717 1.97e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.724 on 454 degrees of freedom
## Multiple R-squared:  0.9977, Adjusted R-squared:  0.9977
## F-statistic: 1.531e+04 on 13 and 454 DF,  p-value: < 2.2e-16
```

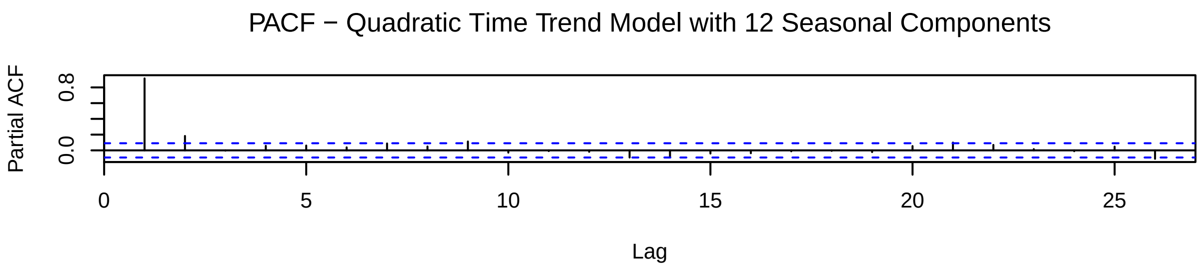
```
qqPlot(co2.ts.lm.stt$residuals, main = expression("Quadratic Time Trend Model with 12 Seasonal
```

```
## [1] 465 361
```

```
plt.acf = acf(co2.ts.lm.stt$residuals, plot = FALSE)
plt.pacf = pacf(co2.ts.lm.stt$residuals, plot = FALSE)
plot(plt.acf, main = expression("ACF - Quadratic Time Trend Model with 12 Seasonal Components")
plot(plt.pacf, main = expression("PACF - Quadratic Time Trend Model with 12 Seasonal Components
```








```
Box.test(co2.ts.lm.stt$residuals, type = "Ljung-Box")
```

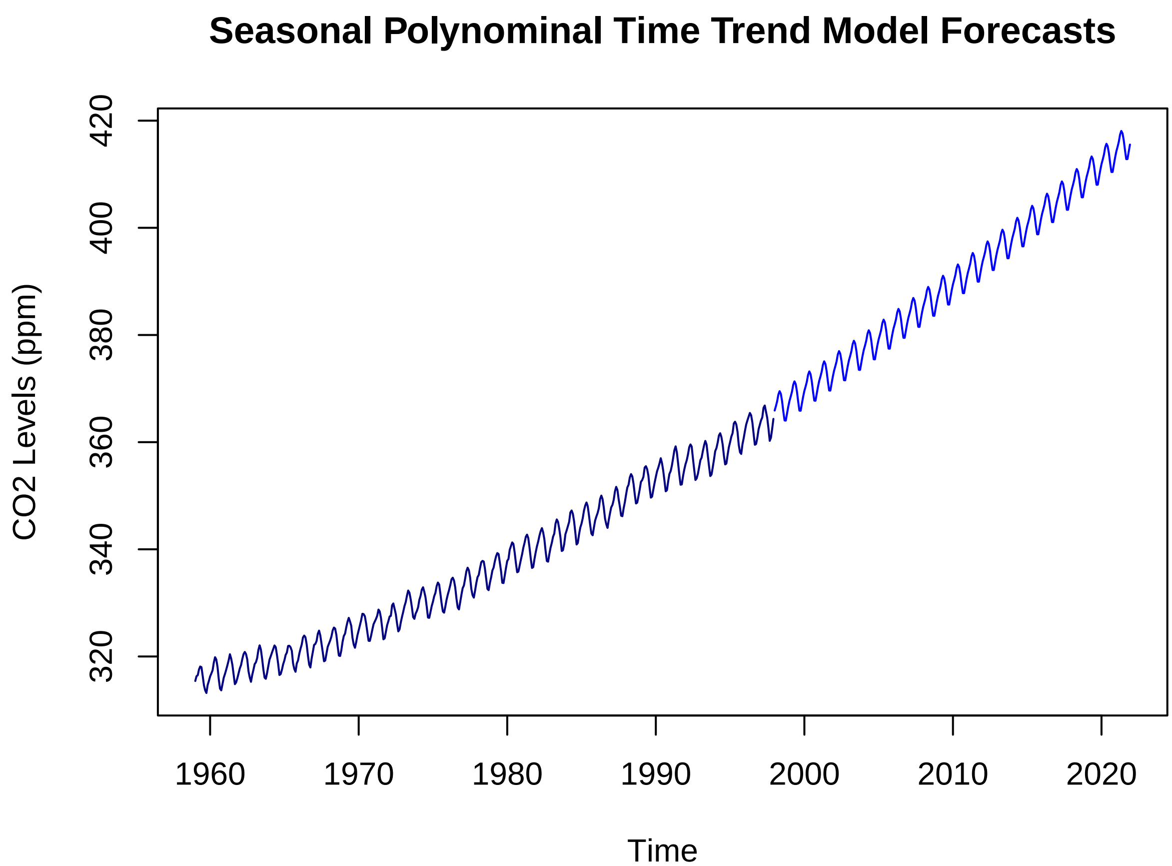
```
##
##  Box-Ljung test
##
## data:  co2.ts.lm.stt$residuals
## X-squared = 393.48, df = 1, p-value < 2.2e-16
```

Next, we fit a polynomial time trend model that incorporates seasonal dummy variables. Based upon residual plots, the quadratic model with time and seasonal dummy variables appears to be a better fit. The residual tails are closer to the quantiles of the normal distribution. However, the ACF plot of the residuals, like those of the linear time trend model, show a trend not captured by our model - the majority of autocorrelations are significant and there is a gradual decay in values over the lags. The PACF shows fewer significant autocorrelations. Again, we find that the model rejects the null hypothesis of the Ljung-Box test, indicating serial correlation in the residuals.

Despite these inadequacies, the model predictions in the short term do not appear unreasonable, as seen in our forecast plots below.

Seasonal Time-Trend Model Predictions

```
new.t = seq(1998, len = (2021 - 1997) * 12, by = 1/12)
new.season <- rep(1:12, (2021 - 1997))
new.dat <- data.frame(time = new.t, season = as.factor(new.season))
stt.preds <- ts(predict(co2.ts.lm.stt, new.dat), st = 1998, fr = 12)

ts.plot(co2, stt.preds, lty = 1, col = c("navy", "blue"), ylab = "CO2 Levels (ppm)",
    main = "Seasonal Polynominal Time Trend Model Forecasts")
```



Part 3 (4 points)

Following all appropriate steps, choose an ARIMA model to fit to this `co2` series. Discuss the characteristics of your model and how you selected between alternative ARIMA specifications. Use your model to generate forecasts to the present.

SARIMA Model Selection

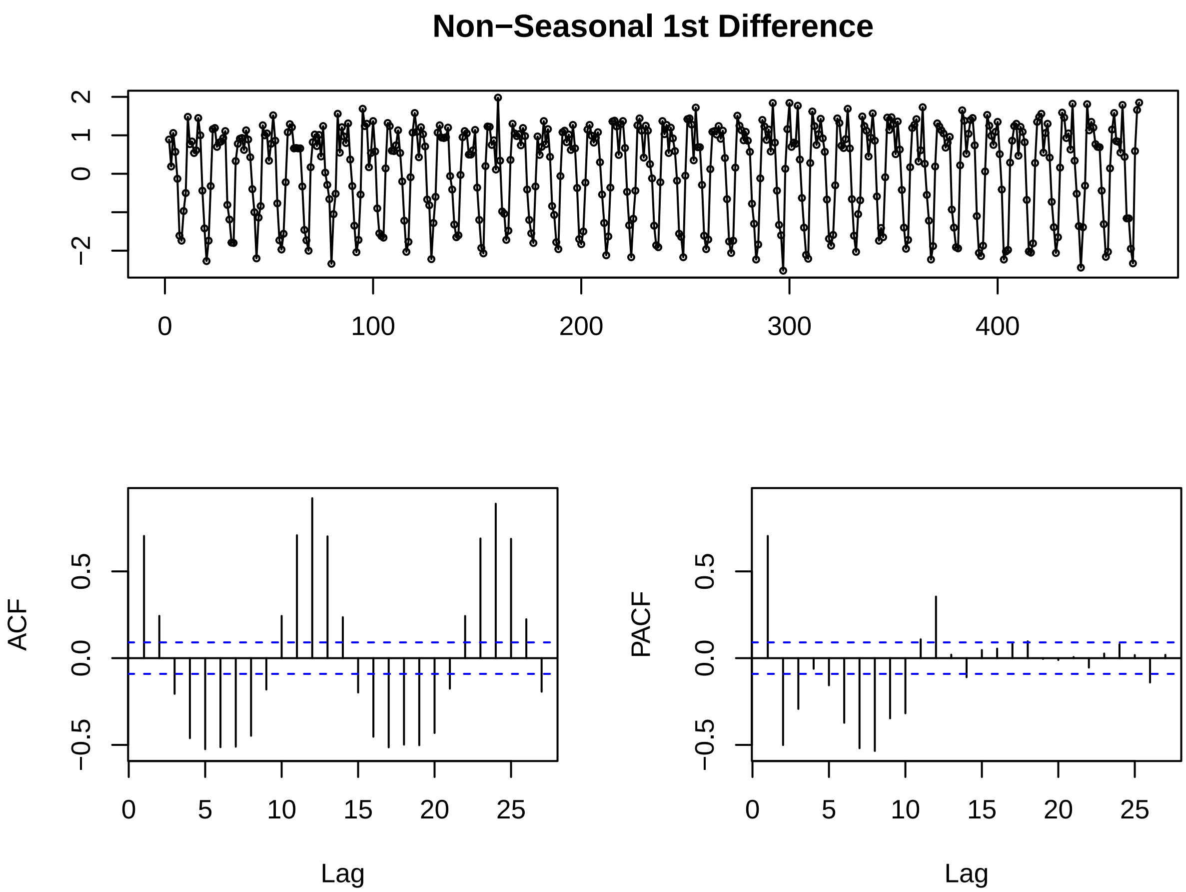
```
# Find the number of seasonal and non-seasonal differences
# needed for stationarity 1 non-seasonal difference and 0
# seasonal differences are required
unitroot_ndiffs(co2)
```

```
## ndiffs
##      1
```

```
unitroot_nsdiffs(co2)
```

```
## nsdiffs
##       0
```

```
# Plot the residuals, ACF, and PACF of the
# first-differenced series The PACF chart has fewer
# repeated significant spikes at seasonal lags than the ACF
# does so we'll use it for the seasonal part of the model
# in our initial estimate The PACF only a seasonal spike at
# a lag of 12 - (1,0,0) Since we used the PACF for the
# seasonal part, we'll estimate the non-seasonal with the
# ACF The first 2 autocorrelations in the ACF are
# significant, so we'll estimate an MA(2)
tsdisplay(difference(co2), main = "Non-Seasonal 1st Difference")
```




```
# Create an Arima model based upon our observations
co2.sarima = arima(co2, order = c(0, 1, 2), seas = list(order = c(1,
    0, 0), frequency(co2)), method = "CSS")

# Find the AIC of the Arima model, check the residuals, and
```

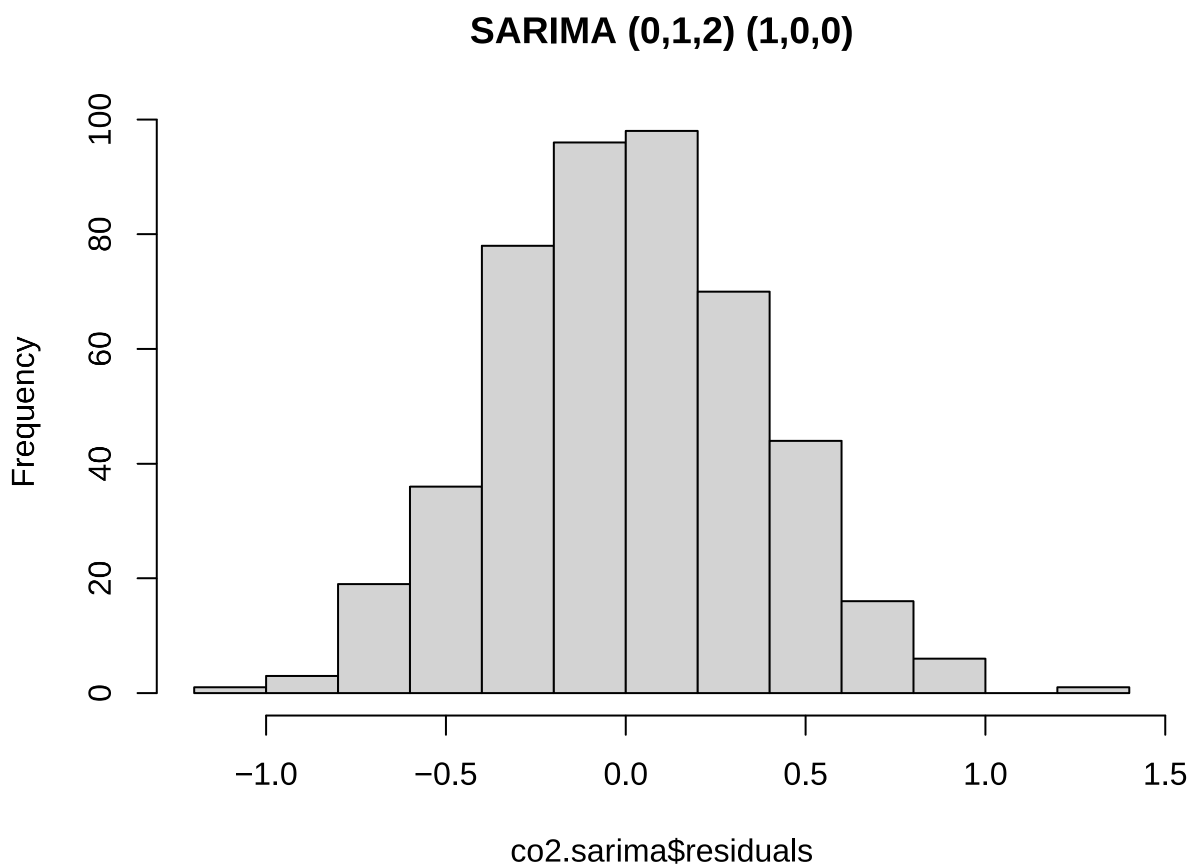
```r
# perform Ljung-Box
co2.sarima.aicc <- -2 * co2.sarima$loglik + log(length(co2) +
    1) * (length(co2.sarima$coef))
co2.sarima.aicc
```

```
## [1] 413.4629
```

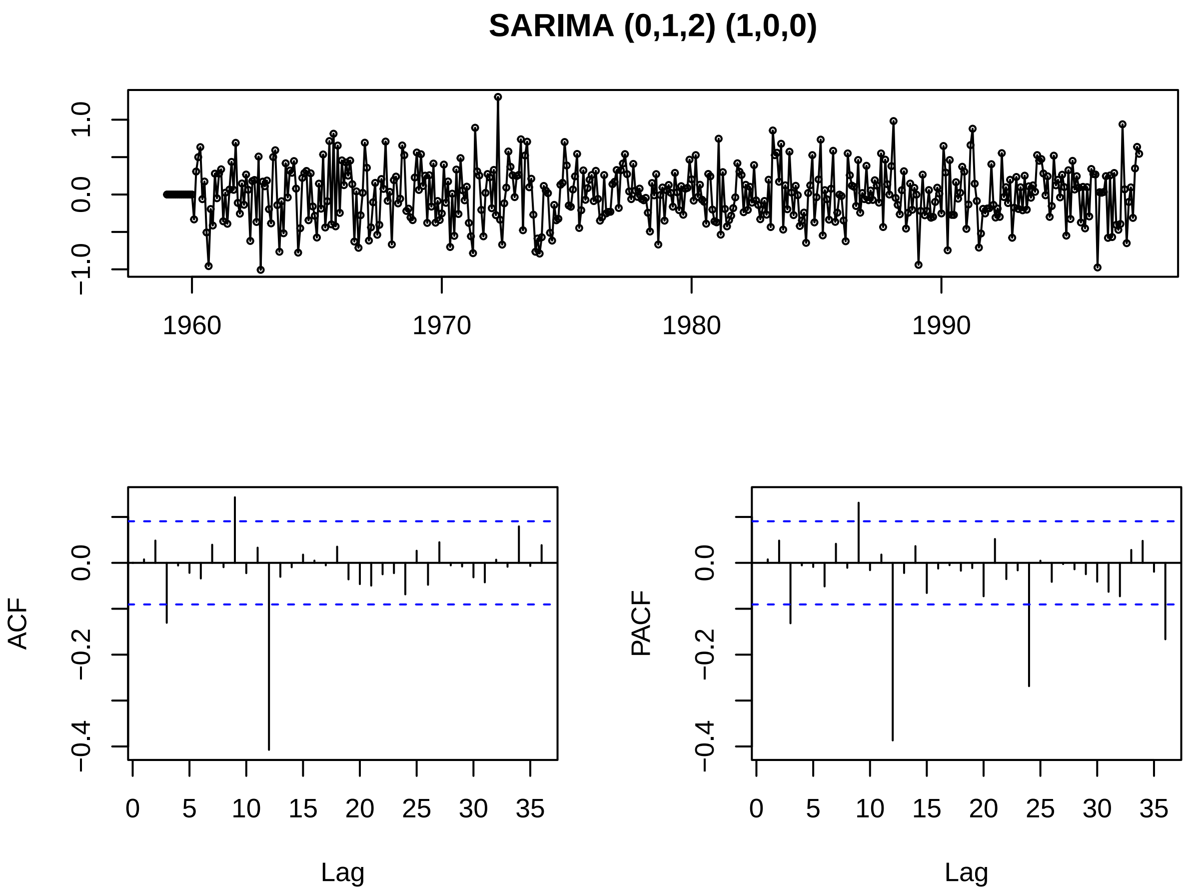
```r
# Look at the estimated coefficients
summary(co2.sarima)
```

```
##
## Call:
## arima(x = co2, order = c(0, 1, 2), seasonal = list(order = c(1, 0, 0), frequency(co2)),
##     method = "CSS")
##
## Coefficients:
##           ma1      ma2    sar1
##       -0.3501  -0.0577  0.9804
## s.e.   0.0462   0.0444  0.0108
##
## sigma^2 estimated as 0.1364:  part log likelihood = -197.51
##
## Training set error measures:
##                      ME      RMSE       MAE         MPE       MAPE      MASE
## Training set 0.00639654 0.3641826 0.2888305 0.001826364 0.08591893 0.2683615
##                     ACF1
## Training set 0.007648558
```

```r
# The histogram plot looks approximately normal
hist(co2.sarima$residuals, main = "SARIMA (0,1,2) (1,0,0)")
```



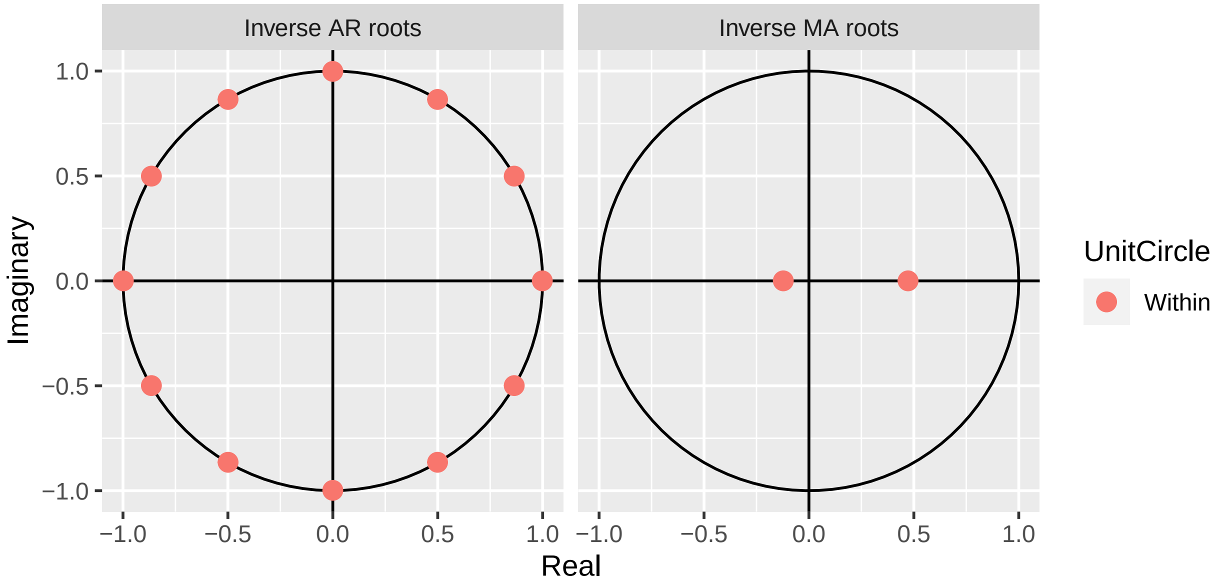

```r
# A time series plot of the residuals appears to have a
# constant mean The ACF and PACF plots show a few
# significant autocorrelations
tsdisplay(co2.sarima$residuals, main = "SARIMA (0,1,2) (1,0,0)")
```




```
# But the model fails to reject the null hypothesis,
# suggesting that the residuals are independently
# distributed
Box.test(co2.sarima$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  co2.sarima$residuals
## X-squared = 0.027554, df = 1, p-value = 0.8682
```

```
# Check the inverse unit roots for stationarity The inverse
# unit roots are near non-stationarity
autoplot(co2.sarima)
```



To create our initial model, we first ran unit root tests to check the number of seasonal and non-seasonal differences required for stationarity. These tests returned 1 non-seasonal difference and 0 seasonal differences required, so we used these values as our d and D to estimate our initial Arima model.

To obtain p, q, P, and Q, we took a first non-seasonal difference and plotted the ACF, PACF, and differenced values as a time series. The time series plot of the differenced values appeared relatively stationary. The ACF and PACF still showed evidence of autocorrelation. Since the PACF had fewer repeating seasonal lags, we used this plot to estimate the seasonal part of the Arima model. The PACF plot showed a significant autocorrelation at only the first seasonal lag, at 12, so we estimated $(1, 0, 0)$ for the seasonal part of the model. For the non-seasonal part of the Arima model, the ACF showed significant autocorrelation at lags 1 and 2, so we estimated a MA model of order 2, or $(0, 1, 2)$ for the non-seasonal component (with a difference of 1 since we took 1 non-seasonal difference).

The ACF and PACF plots of the residuals of this estimated model ($(0, 1, 2)(1, 0, 0)_{12}$) shows a few significant autocorrelations (notably at 1 year in the ACF and PACF and at 2 years in the PACF), although the majority of values fall within the confidence interval for white noise values.

The Ljung-Box test, however, shows a p-value > 0.05, so we fail to reject the null hypothesis that the residuals are independently distributed, suggesting that they are not serially correlated.

Since the ACF and PACF plots still showed a few strong autocorrelations and the plot of the inverse unit roots showed values near unity, we proceeded to iterate over model parameters to see if we could improve the AIC score and create a model with residuals that better approximated white noise.

Model Selection Algorithm

```
get.best.arima <- function(x.ts, maxord = c(1, 1, 1, 1, 1, 1)) {
    best.aic <- 1e+08
    df.results = data.frame()
    n <- length(x.ts)
```

```
    for (p in 0:maxord[1]) for (d in 0:maxord[2]) for (q in 0:maxord[3]) for (P in 0:maxord[4])
        tryCatch({
            fit <- arima(x.ts, order = c(p, d, q), seas = list(order = c(P,
                D, Q), frequency(x.ts)), method = "ML")

            npar <- length(fit$coef[fit$mask]) + 1
            nstar <- length(fit$residuals) - fit$arma[6] - fit$arma[7] *
                fit$arma[5]

            fit.aic <- fit$aic
            fit.bic <- fit.aic + npar * (log(nstar) - 2)
            fit.aicc <- fit.aic + 2 * npar * (nstar/(nstar -
                npar - 1) - 1)

            df <- data.frame(model = paste(p, d, q, P, D, Q),
                AICc = fit.aicc, AIC = fit.aic, BIC = fit.bic)
            df.results <- rbind(df.results, df)
        }, error = function(cond) {
            paste("[", p, ",", d, ",", q, "]", "[", P, ",", D,
                ",", Q, "]")
        })
    }
    df.results
}

arima.search <- get.best.arima(co2, maxord = c(2, 2, 2, 2, 2,
    2))
```

To find a parsimonious seasonal Arima model that better fit the time series, we looped over values in the range of 0 to 2 for the parameters p, q, P, and Q. We also chose the range of 0 to 2 for the number of seasonal and non-seasonal differences, since differencing beyond order 2 is rarely required.

For the best fit model, we chose to use the model with the lowest AICc, as seen in our table below (using AICc since it penalizes the model fit with increasing parameters and corrects for the bias in predictor selection introduced by AIC). As seen below, the best fitting model is $(0, 1, 1)(2, 1, 2)$.

```
best10.arima <- head(arima.search[with(arima.search, order(AICc)),
    ], n = 10)
row.names(best10.arima) <- NULL
kable(best10.arima, caption = "Top 10 Models.")
```

Table 1: Top 10 Models.

model	AICc	AIC	BIC
0 1 1 2 1 2	173.6886	173.5011	198.2229
0 1 2 2 1 2	174.2829	174.0323	202.8744
2 1 1 0 1 1	177.9614	177.8278	198.4293

model	AICc	AIC	BIC
1 1 1 0 1 1	178.1561	178.0672	194.5484
0 1 1 0 1 1	178.2089	178.1557	190.5166
1 0 1 2 1 2	178.6929	178.4429	207.3003
1 1 2 0 1 1	178.7607	178.6271	199.2286
0 1 2 0 1 1	179.1813	179.0924	195.5736
2 1 1 2 1 1	179.1879	178.9373	207.7794
2 1 2 0 1 1	179.2641	179.0766	203.7984

```
# Estimate an Arima model with the parameters of the model
# with the lowest AICc found from our parameter search
pdqPDQ <- as.list(unlist(strsplit(best10.arima[1, 1], "[[:space:]]")))
p <- strtoi(pdqPDQ[[1]])
d <- strtoi(pdqPDQ[[2]])
q <- strtoi(pdqPDQ[[3]])
P <- strtoi(pdqPDQ[[4]])
D <- strtoi(pdqPDQ[[5]])
Q <- strtoi(pdqPDQ[[6]])

# Estimate the model
co2.sarima.2 <- arima(co2, order = c(p, d, q), seasonal = list(order = c(P,
    D, Q)), method = "ML")
```

Our best sarima model can be expressed as below in the form backshift operator

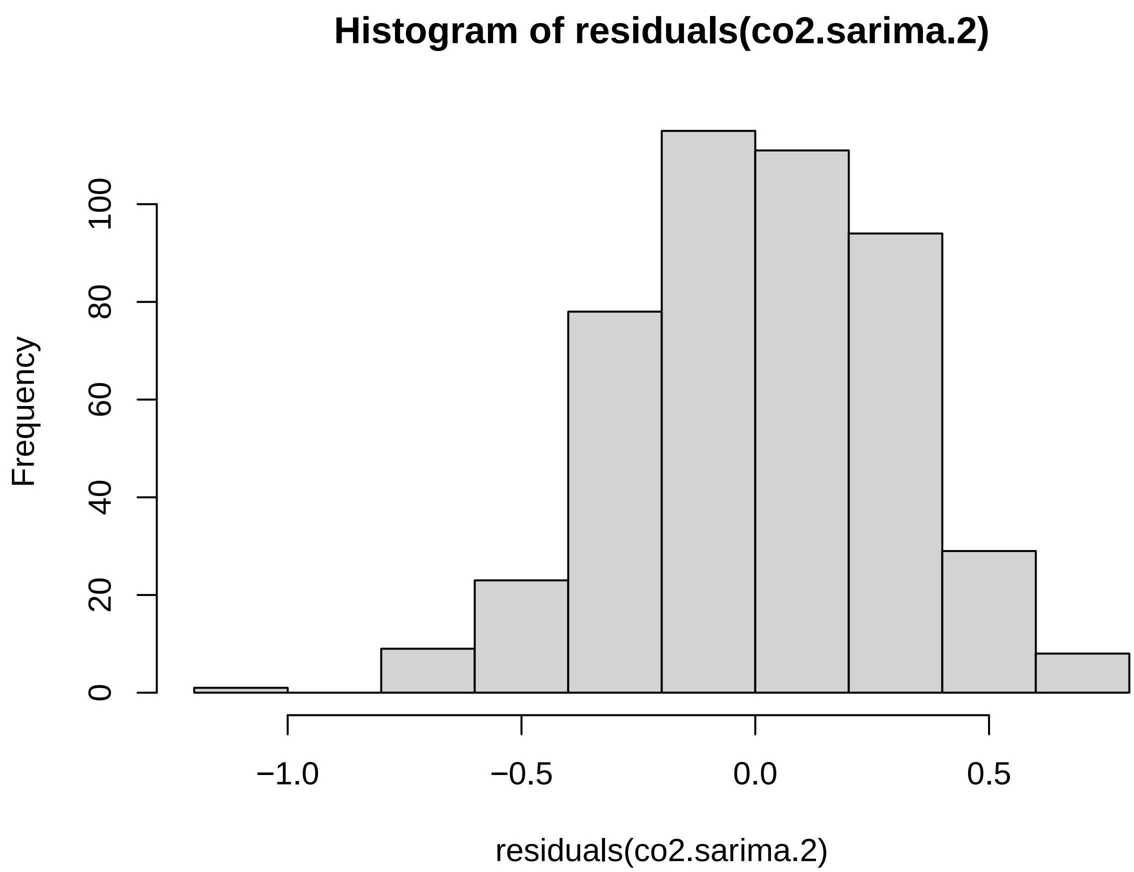
$$(1-\Phi_1 B^{12} - \Phi_2 B^{13})(1-B)(1-B^{12})x_t = (1+\theta_1 B)(1+\Theta_{12}B^{12}+\Theta_{13}B^{13})w_t$$

$(1-\Phi_1 B^{12} - \Phi_2 B^{13})$ represents seasonal auto regressive term, $(1+\theta_1 B)$ represents moving average term and $(\Theta_{12}B^{12}+\Theta_{13}B^{13})$ represents seasonal moving average of `arima` model. w_t represents white noise of the time series.

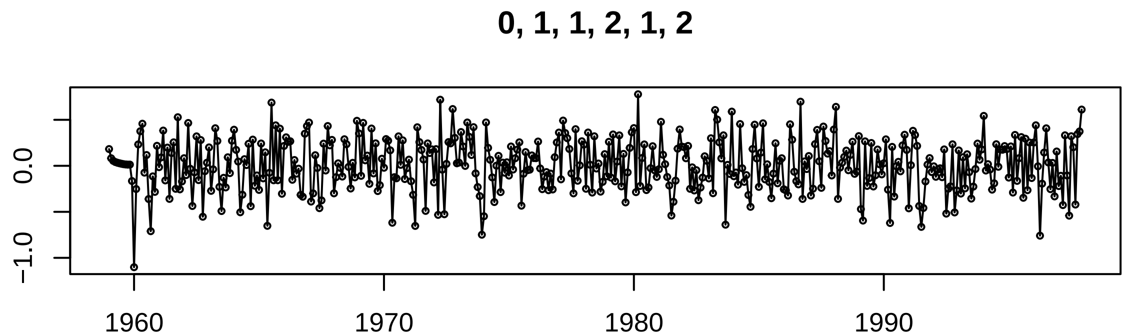
After solving for coefficients using R `arima` model, we get
$x_t = x_{t-1} +$ (0.9591506) $* x_{t-12}$ + + ($ -0.1266424) $* x_{t-13} + w_t +$ (-0.3520208) $* w_{t-1}$ + (-1.8151984) $* w_{t-12}$ + (0.8537521) $* w_{t-13}$

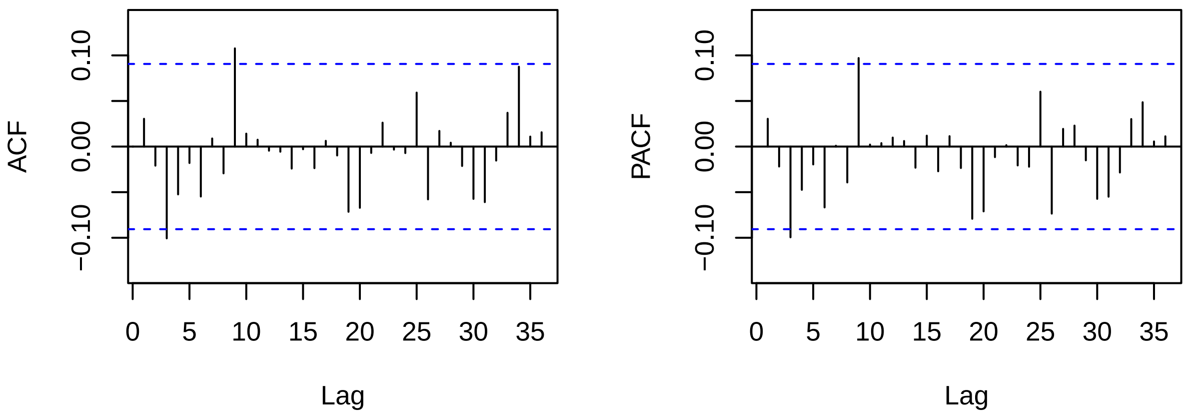
where x_{t-12} and x_{t-13}represents 12th & 13th lag of time series, which comes from seasonal part of arima model. The x_{t-1} is the results of first difference of time series i.e. $x_t^1 = x_t - x_{t-1}$
w_t is white noise from current time step, w_{t-1} is white noise from the previous time step, which is the result of AR moving average. w_{t-12} is the white noise from 12 steps before (seasonal) the current time step and w_{t-13} is the white noise from 13 steps before current time step. This is the result of seasonal moving average component of our model.

```
# Inspect the residual plots and find the estimated AICc
sarima2.aicc <- -2 * co2.sarima.2$loglik + (log(length(co2)) +
    1) * length(co2.sarima.2$coef)
hist(residuals(co2.sarima.2))
```




```
tsdisplay(co2.sarima.2$residuals, main = {
    toString(pdqPDQ)
})
```





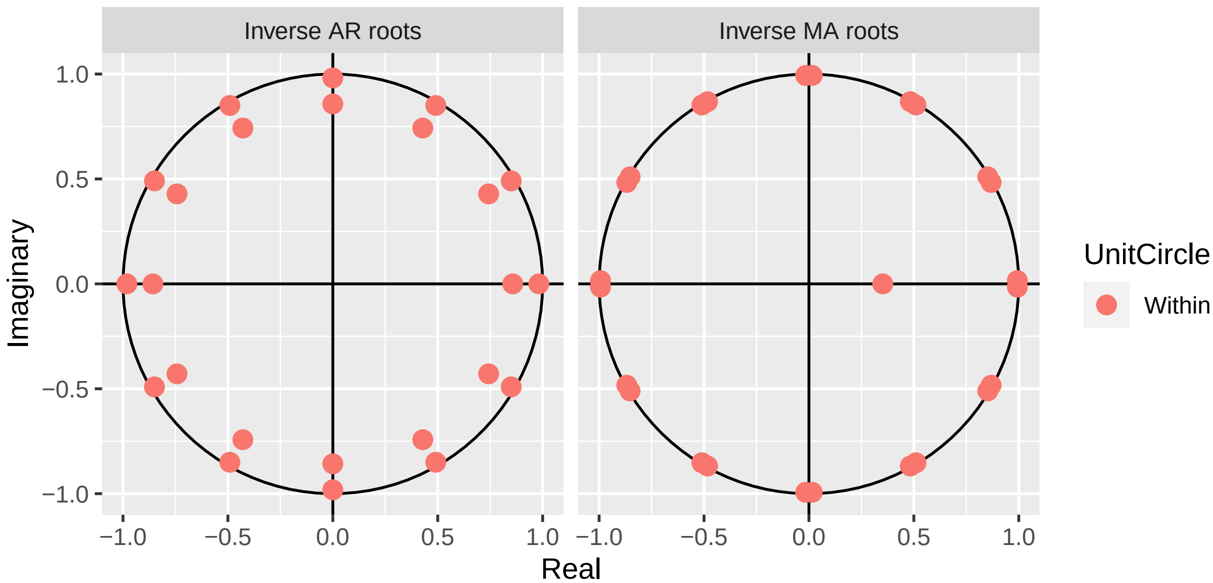

```
sarima2.aicc
```

```
## [1] 197.2434
```

```
Box.test(co2.sarima.2$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  co2.sarima.2$residuals
## X-squared = 0.43736, df = 1, p-value = 0.5084
```

```
autoplot(co2.sarima.2)
```



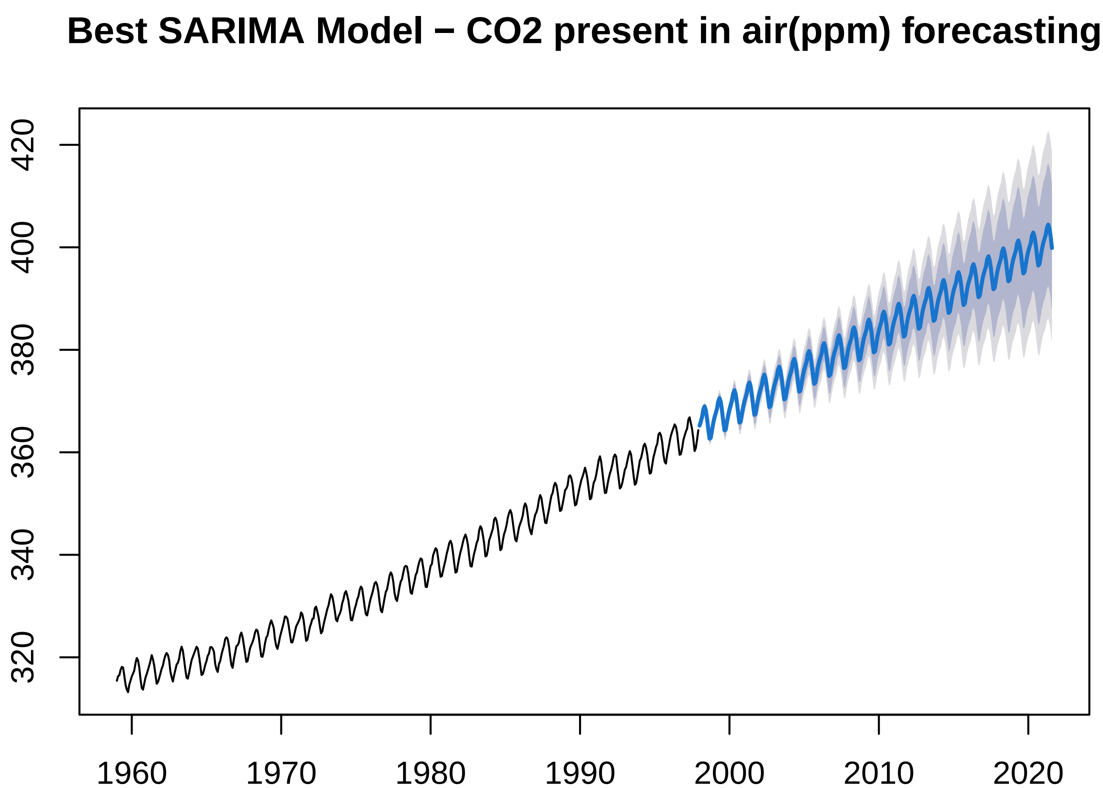
The AICc value is smaller than that of our initial model estimate, and the majority of ACF and PACF values fall within the 95% confidence interval bounds for white noise. In addition, the Ljung-Box test indicates that the data are independently distributed since we fail to reject the null hypothesis.

The histogram of the residuals shows them to be approximately normally distributed and the plot of the residuals as a time series resembles white noise. Moreover, as seen in the plots of the inverse unit roots, the absolute value of the inverse unit roots are less than unity, meaning that the residuals are stationary.

Since this model has a lower AICc than our initial base model estimate, the residuals resemble white noise, and we have not found significant evidence of residual autocorrelation, we will use this model for forecasting to the present day (as seen in the plot below).

Best Model Forecasts

```
co2.forecast <- forecast(co2.sarima.2, 284)
co2_forecast_ts <- co2.forecast[4]$mean
plot(co2.forecast, main = "Best SARIMA Model - CO2 present in air(ppm) forecasting")
```



Part 4 (5 points)

The file `co2_weekly_mlo.txt` contains weekly observations of atmospheric carbon dioxide concentrations measured at the Mauna Loa Observatory from 1974 to 2020, published by the National Oceanic and Atmospheric Administration (NOAA). Convert these data into a suitable time series object, conduct a thorough EDA on the data, addressing the problem of missing observations and comparing the Keeling Curve's development to your predictions from Parts 2 and 3. Use the weekly data to generate a month-average series from 1997 to the present and use this to generate accuracy metrics for the forecasts generated by your models from Parts 2 and 3.

```r
co2_weekly <- read.table("co2_weekly_mlo.txt", header = FALSE)
colnames(co2_weekly) <- c("year", "month", "day", "decimal",
    "ppm", "days", "1yr_ago", "10yrs_ago", "since1800")
summary(co2_weekly)
```

```
##       year          month             day           decimal
##  Min.   :1974   Min.   : 1.00   Min.   : 1.00   Min.   :1974
##  1st Qu.:1986   1st Qu.: 4.00   1st Qu.: 8.00   1st Qu.:1986
##  Median :1997   Median : 7.00   Median :16.00   Median :1998
##  Mean   :1997   Mean   : 6.52   Mean   :15.72   Mean   :1998
##  3rd Qu.:2009   3rd Qu.:10.00   3rd Qu.:23.00   3rd Qu.:2010
##  Max.   :2021   Max.   :12.00   Max.   :31.00   Max.   :2021
##       ppm                days           1yr_ago           10yrs_ago
##  Min.   :-1000.0   Min.   :0.000   Min.   :-1000.0   Min.   : -999.99
##  1st Qu.:  347.1   1st Qu.:5.000   1st Qu.:  345.6   1st Qu.:  331.48
##  Median :  365.2   Median :6.000   Median :  363.5   Median :  350.18
##  Mean   :  358.3   Mean   :5.871   Mean   :  328.4   Mean   :   59.61
```

```
##  3rd Qu.:  388.4   3rd Qu.:7.000   3rd Qu.:  386.2   3rd Qu.:  368.45
##  Max.   :  420.0   Max.   :7.000   Max.   :  417.8   Max.   :  395.23
##    since1800
##  Min.   : -999.99
##  1st Qu.:   66.95
##  Median :   84.55
##  Mean   :   80.38
##  3rd Qu.:  108.07
##  Max.   :  136.87
```

```
describe(co2_weekly)
```

```
## co2_weekly
##
##  9  Variables      2458  Observations
## --------------------------------------------------------------------------------
## year
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0       48        1     1997    15.71     1976     1979
##      .25      .50      .75      .90      .95
##     1986     1997     2009     2016     2019
##
## lowest : 1974 1975 1976 1977 1978, highest: 2017 2018 2019 2020 2021
## --------------------------------------------------------------------------------
## month
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0       12    0.993     6.52    3.965        1        2
##      .25      .50      .75      .90      .95
##        4        7       10       11       12
##
## lowest :  1  2  3  4  5, highest:  8  9 10 11 12
##
## Value          1     2     3     4     5     6     7     8     9    10    11
## Frequency    208   190   208   201   211   205   208   208   202   207   202
## Proportion 0.085 0.077 0.085 0.082 0.086 0.083 0.085 0.085 0.082 0.084 0.082
##
## Value         12
## Frequency    208
## Proportion 0.085
## --------------------------------------------------------------------------------
## day
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0       31    0.999    15.72    10.16        2        4
##      .25      .50      .75      .90      .95
##        8       16       23       28       29
##
## lowest :  1  2  3  4  5, highest: 27 28 29 30 31
## --------------------------------------------------------------------------------
```

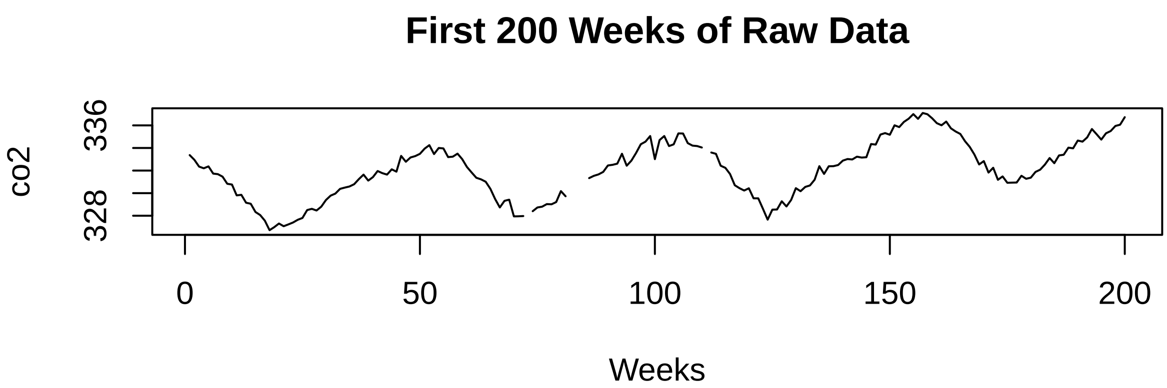
```
## decimal
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0     2458        1     1998    15.71     1977     1979
##      .25      .50      .75      .90      .95
##     1986     1998     2010     2017     2019
##
## lowest : 1974.380 1974.399 1974.418 1974.437 1974.456
## highest: 2021.390 2021.410 2021.429 2021.448 2021.467
## --------------------------------------------------------------------------------
## ppm
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0     2148        1    358.3    47.87    332.4    336.1
##      .25      .50      .75      .90      .95
##    347.1    365.2    388.4    404.6    410.6
##
## lowest : -999.99  326.72  326.99  327.07  327.23
## highest:  419.28  419.47  419.53  419.55  420.01
##
## Value      -1000   320   340   360   380   400   420
## Frequency     18    45   638   662   527   435   133
## Proportion 0.007 0.018 0.260 0.269 0.214 0.177 0.054
##
## For the frequency table, variable is rounded to the nearest 20
## --------------------------------------------------------------------------------
## days
##        n  missing distinct     Info     Mean      Gmd
##     2458        0        8    0.896    5.871    1.378
##
## lowest : 0 1 2 3 4, highest: 3 4 5 6 7
##
## Value          0     1     2     3     4     5     6     7
## Frequency     18    14    36   101   176   402   648  1063
## Proportion 0.007 0.006 0.015 0.041 0.072 0.164 0.264 0.432
## --------------------------------------------------------------------------------
## 1yr_ago
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0     2097        1    328.4    101.7    330.5    334.4
##      .25      .50      .75      .90      .95
##    345.6    363.5    386.2    402.0    408.2
##
## lowest : -999.99  326.73  326.84  326.98  327.21
## highest:  417.09  417.10  417.21  417.46  417.83
##
## Value      -1000   320   340   360   380   400   420
## Frequency     70    45   638   665   523   436    81
## Proportion 0.028 0.018 0.260 0.271 0.213 0.177 0.033
##
## For the frequency table, variable is rounded to the nearest 20
```

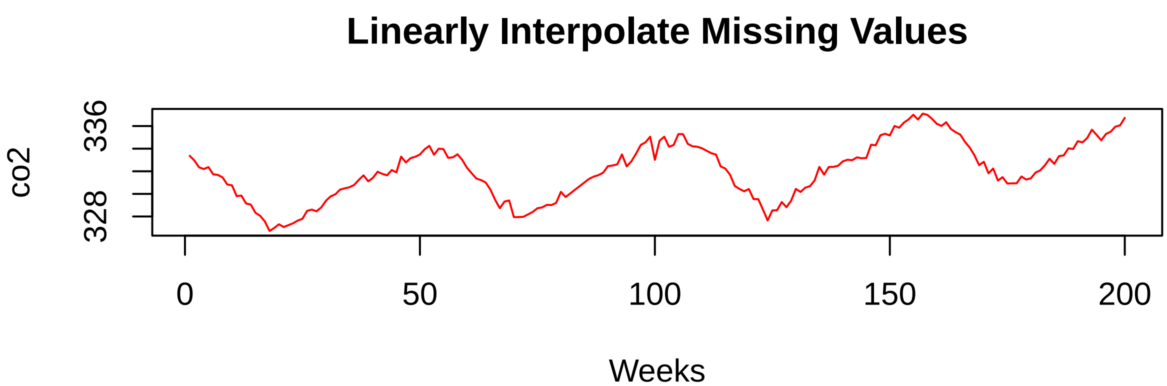
```
## ---------------------------------------------------------------------------------
## 10yrs_ago
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0     1644    0.989    59.61    479.1  -1000.0  -1000.0
##      .25      .50      .75      .90      .95
##    331.5    350.2    368.5    382.4    387.0
##
## lowest : -999.99  326.66  327.04  327.10  327.26
## highest:  394.08  394.15  394.43  395.13  395.23
##
## Value      -1000   330   340   350   360   370   380   390   400
## Frequency    541   196   328   343   339   286   248   175     2
## Proportion 0.220 0.080 0.133 0.140 0.138 0.116 0.101 0.071 0.001
##
## For the frequency table, variable is rounded to the nearest 10
## ---------------------------------------------------------------------------------
## since1800
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0     2086        1    80.38    43.66    52.11    55.81
##      .25      .50      .75      .90      .95
##    66.95    84.55   108.07   125.10   130.75
##
## lowest : -999.99   49.60   49.65   49.72   49.95
## highest:  136.49  136.61  136.64  136.74  136.87
##
## Value      -1000    50    60    70    80    90   100   110   120   130   140
## Frequency     18   194   326   325   371   270   260   245   200   216    33
## Proportion 0.007 0.079 0.133 0.132 0.151 0.110 0.106 0.100 0.081 0.088 0.013
##
## For the frequency table, variable is rounded to the nearest 10
## ---------------------------------------------------------------------------------
```

NOAA data provided in the file has 2458 weekly observations from 1974 to 2021 with 10 variables. Variable `ppm` tracks weekly co2 presence. We will be using `ppm` values for our analysis. It appears that NOAA uses -999 to represent missing values. For `ppm`, there are 18 observations missing.

Impute Missing Values Linearly

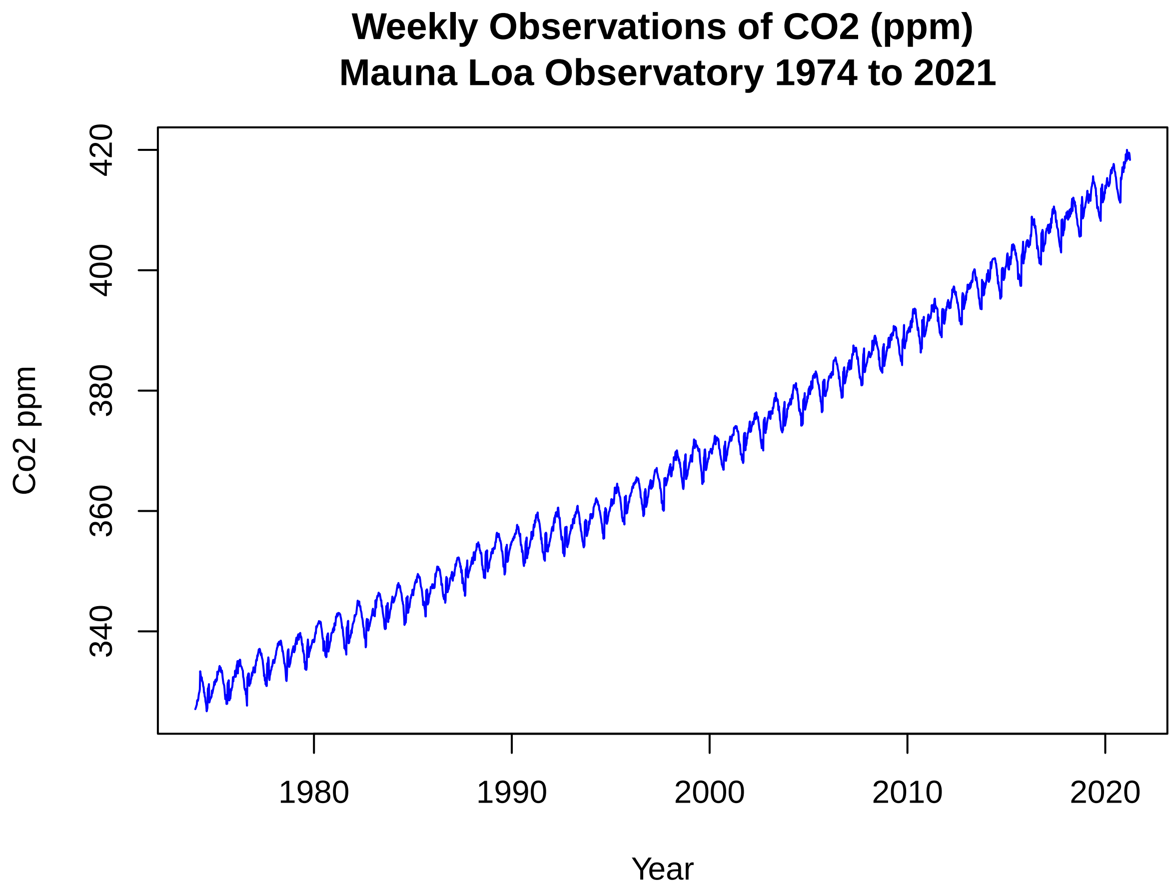
```
co2_weekly <- co2_weekly %>%
    mutate(ppm = ifelse(test = (ppm <= 0), NA, no = ppm))
co2_weekly2 <- data.frame(lapply(co2_weekly, function(X) approxfun(seq_along(X),
    X)(seq_along(X))))
par(mfrow = c(2, 1))
plot(co2_weekly$ppm[1:200], type = "l", xlab = "Weeks", ylab = "co2",
    main = "First 200 Weeks of Raw Data")

plot(co2_weekly2$ppm[1:200], type = "l", col = "red", xlab = "Weeks",
    ylab = "co2", main = "Linearly Interpolate Missing Values")
```





After careful observation of the data, most of the missing points are spread out across the data set (i.e. we do not need to impute 18 weeks in a row). As a result, we suggest it is reasonable to simply interpolate the missing values linearly. The plot above shows the first 200 weeks of the original data series with missing data and a new time series with missing values imputed.

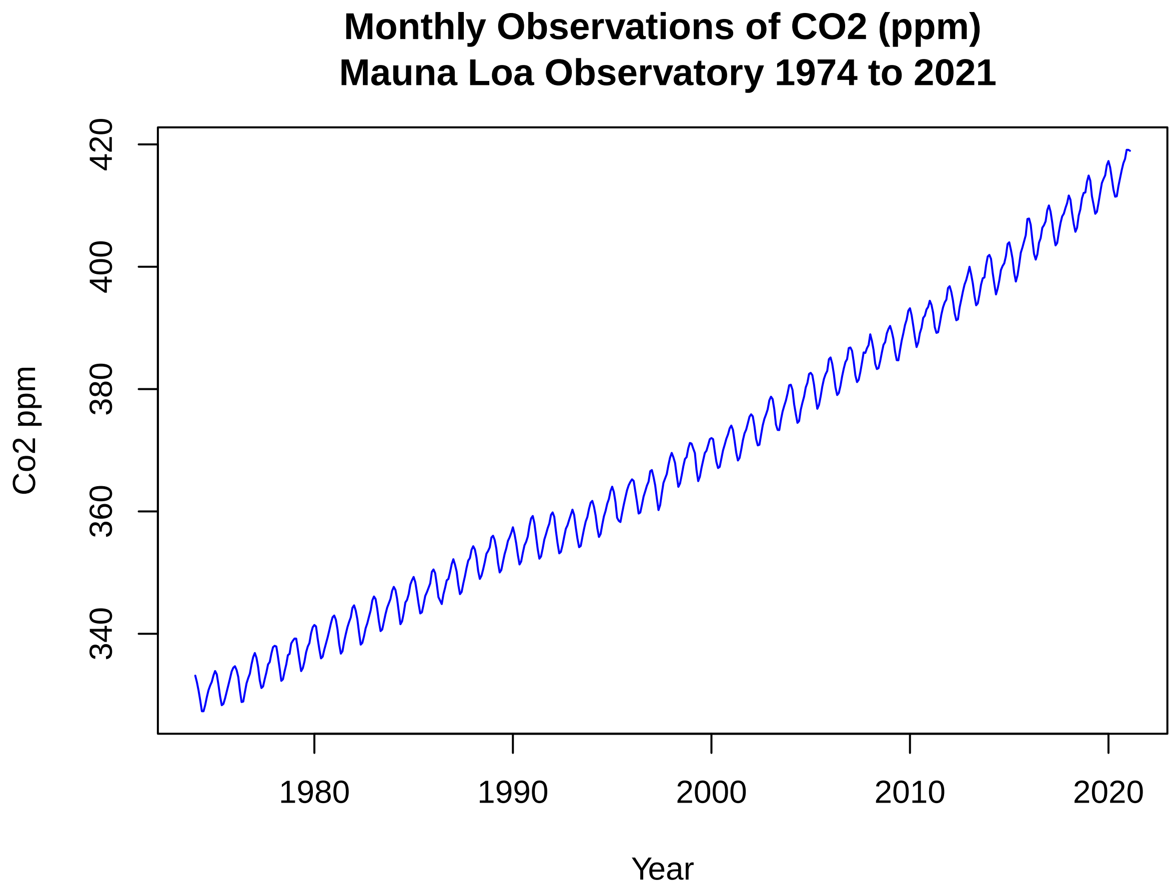
```
# Get monthly averages for replacement after imputing
# missing values linearly
co2_monthly <- co2_weekly2 %>%
    group_by(year, month) %>%
    summarise(ppm_month_avg = mean(ppm))

# join to add monthly averages
co2_merged <- merge(co2_weekly2, co2_monthly, by = c("year",
    "month"))

# Create weekly time series
co2_noaa_weekly_ts <- ts(co2_merged$ppm, start = c(1974), frequency = 52)

# Plot weekly time series
plot(co2_noaa_weekly_ts, main = "Weekly Observations of CO2 (ppm)\n Mauna Loa Observatory 1974
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
```




```r
# Calculate monthly averages as our forecast is only on
# monthly basis
co2_noaa_monthly_df <- co2_merged %>%
    group_by(year, month) %>%
    summarise(ppm_month_avg = mean(ppm))
summary(co2_noaa_monthly_df)
```

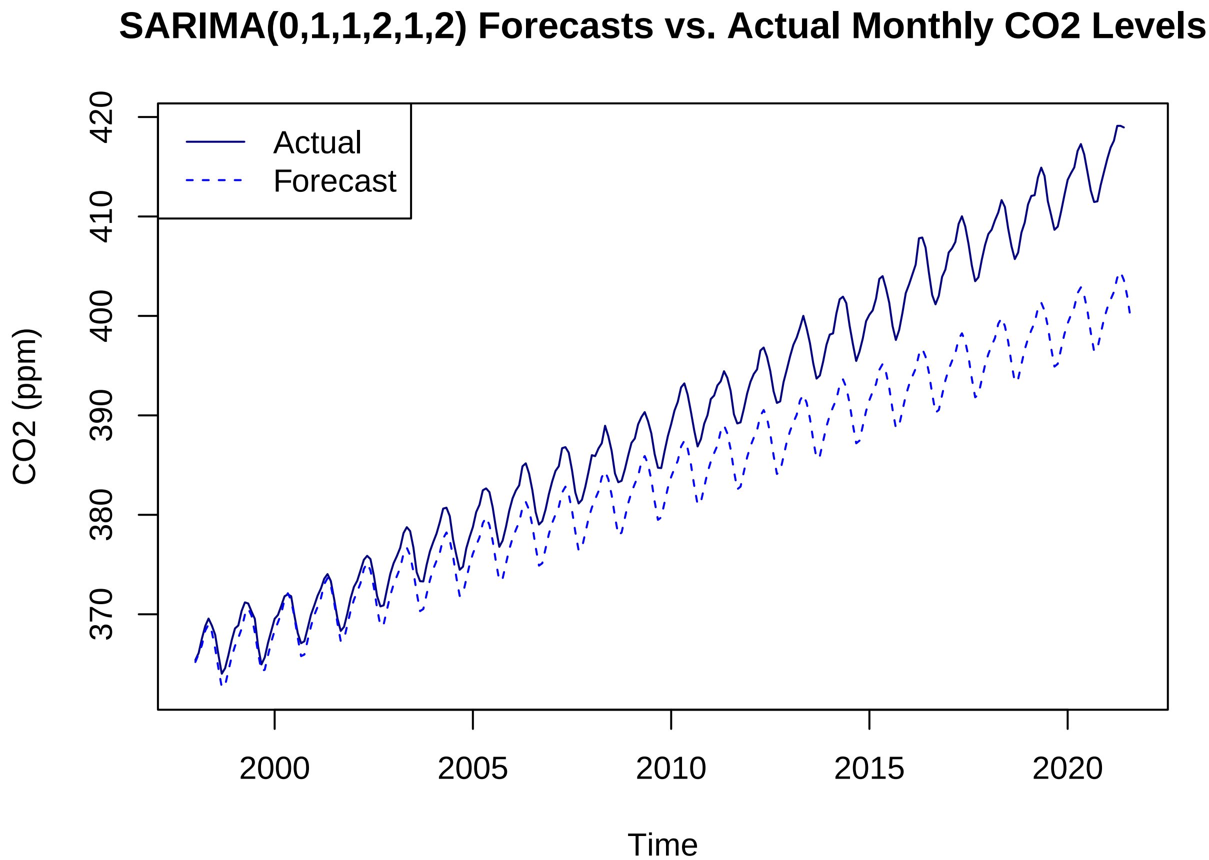
```
##       year          month        ppm_month_avg
##  Min.   :1974   Min.   : 1.000   Min.   :327.3
##  1st Qu.:1986   1st Qu.: 4.000   1st Qu.:347.2
##  Median :1997   Median : 6.000   Median :365.1
##  Mean   :1997   Mean   : 6.496   Mean   :368.2
##  3rd Qu.:2009   3rd Qu.: 9.000   3rd Qu.:388.1
##  Max.   :2021   Max.   :12.000   Max.   :419.1
```

```r
# Create monthly ts object (all observations)
co2_noaa_monthly_ts <- ts(co2_noaa_monthly_df$ppm_month_avg,
    start = c(1974), frequency = 12)

# Plot monthly time series
plot(co2_noaa_monthly_ts, main = "Monthly Observations of CO2 (ppm)\n Mauna Loa Observatory 19
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
```

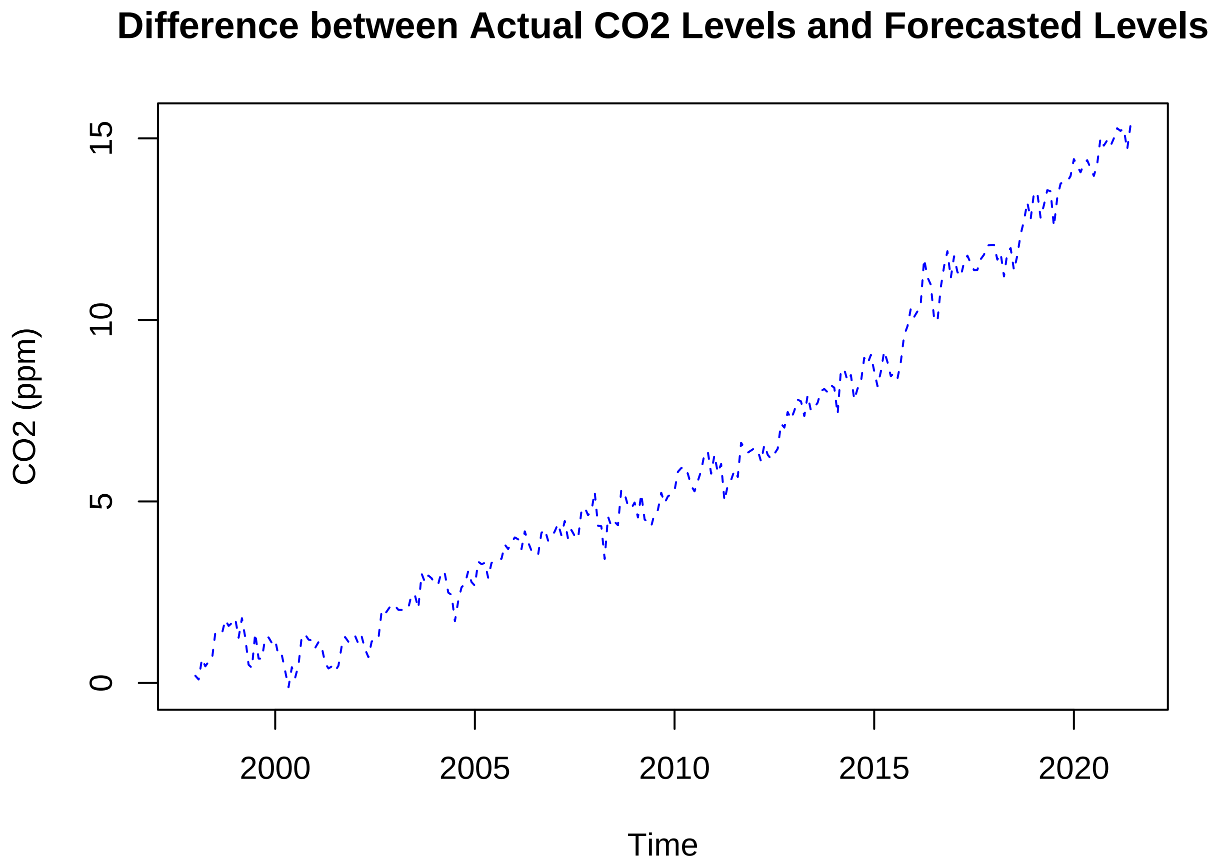


The monthly time series plotted above looks like a smoothed version of the weekly time series.

```
# transforming time series data to dataframe, so that we
# can join
co2_actuals_filtered <- co2_noaa_monthly_df %>%
    filter(year > 1997)

co2_actuals_ts <- ts(co2_actuals_filtered$ppm_month_avg, start = c(1998),
    frequency = 12)

ts.plot(co2_actuals_ts, co2_forecast_ts, lty = 1:2, col = c("navy",
    "blue"), ylab = "CO2 (ppm)", main = "SARIMA(0,1,1,2,1,2) Forecasts vs. Actual Monthly CO2 I
legend("topleft", legend = c("Actual", "Forecast"), col = c("navy",
    "blue"), lty = 1:2)
```




```
actuals_fore_diff <- co2_actuals_ts - co2_forecast_ts
ts.plot(actuals_fore_diff, lty = 2, col = c("blue"), ylab = "CO2 (ppm)",
    main = "Difference between Actual CO2 Levels and Forecasted Levels")
```



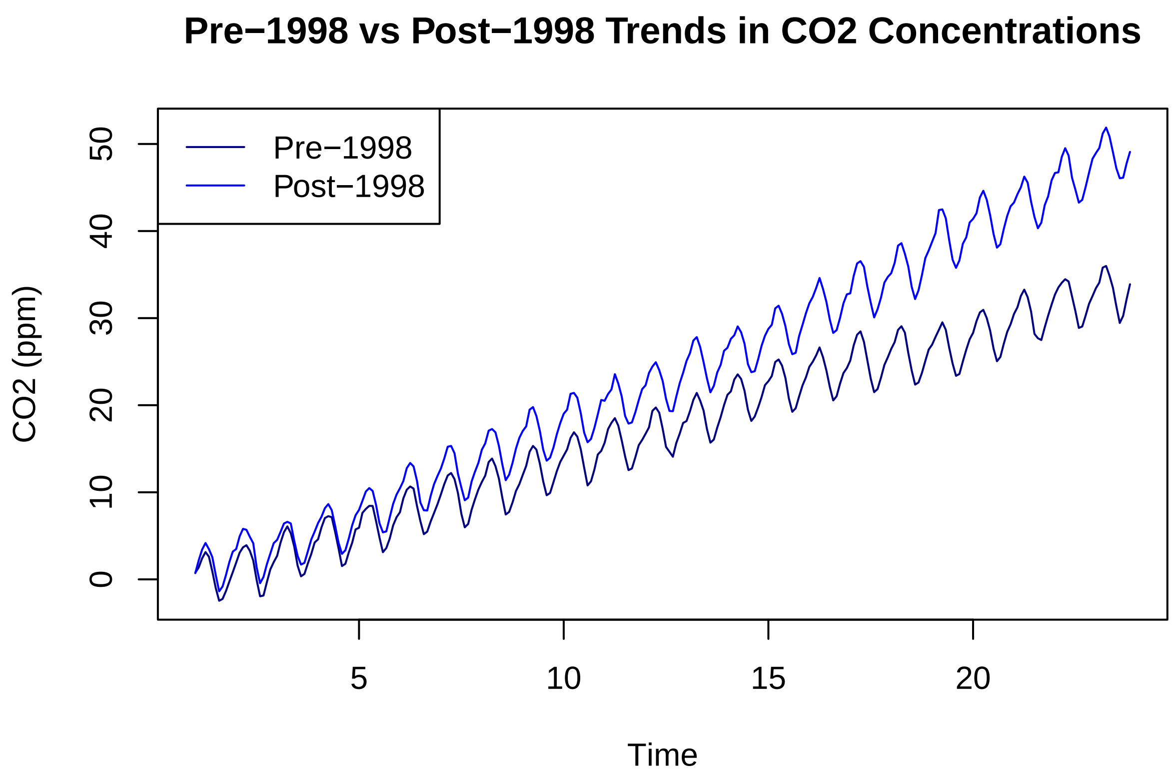
The difference between the actual measured CO2 levels from 1998 to present and our forecasts is stark. It is clear from the plot above that we underestimated the growth of the series over the subsequent 20+ years. Given that our best model's residuals were stationary and resembled to white noise, we would conclude that the forecast error was not necessarily due to a model misspecification, but rather a change in the underlying CO2 generating process. We hypothesize this could be due to the rapid growth of China's economy and other emerging market economies through the 2000s and 2010s[1]. This could be the subject of a deeper, causal understanding of what is driving the ever-increasing concentrations of atmospheric CO2.

To further illustrate this, we created cumulative CO2 (ppm) growth indices of the same number of months. In the plot below, notice that cumulative growth in ppm was greater post-1998 than pre-1998. This indicates that something exogenous changed the trajectory of atmospheric CO2 concentrations in the 276 months following our in-sample period, which would not be expected to be captured by our best ARIMA model.

```
cagr <- function(FV, PV, yrs = 4) {
    values <- ((FV/PV)^(1/yrs) - 1)
    return(values)
}

pre98.df <- subset(co2_noaa_monthly_df, (year < 1998 & year >
    1974))["ppm_month_avg"]
post98.df <- subset(co2_noaa_monthly_df, (year >= 1998))["ppm_month_avg"]
```

[1] https://climateactiontracker.org/countries/china/

```r
# cut to equal size in months
post98.df <- head(post98.df, n = nrow(pre98.df))

pre98.diff <- diff(as.matrix(pre98.df))
post98.diff <- diff(as.matrix(post98.df))

pre98.growth <- ts(cumsum(pre98.diff), frequency = 12)
post98.growth <- ts(cumsum(post98.diff), frequency = 12)

ts.plot(pre98.growth, post98.growth, lty = 1, col = c("navy",
    "blue"), ylab = "CO2 (ppm)", main = "Pre-1998 vs Post-1998 Trends in CO2 Concentrations")
legend("topleft", legend = c("Pre-1998", "Post-1998"), col = c("navy",
    "blue"), lty = 1)
```



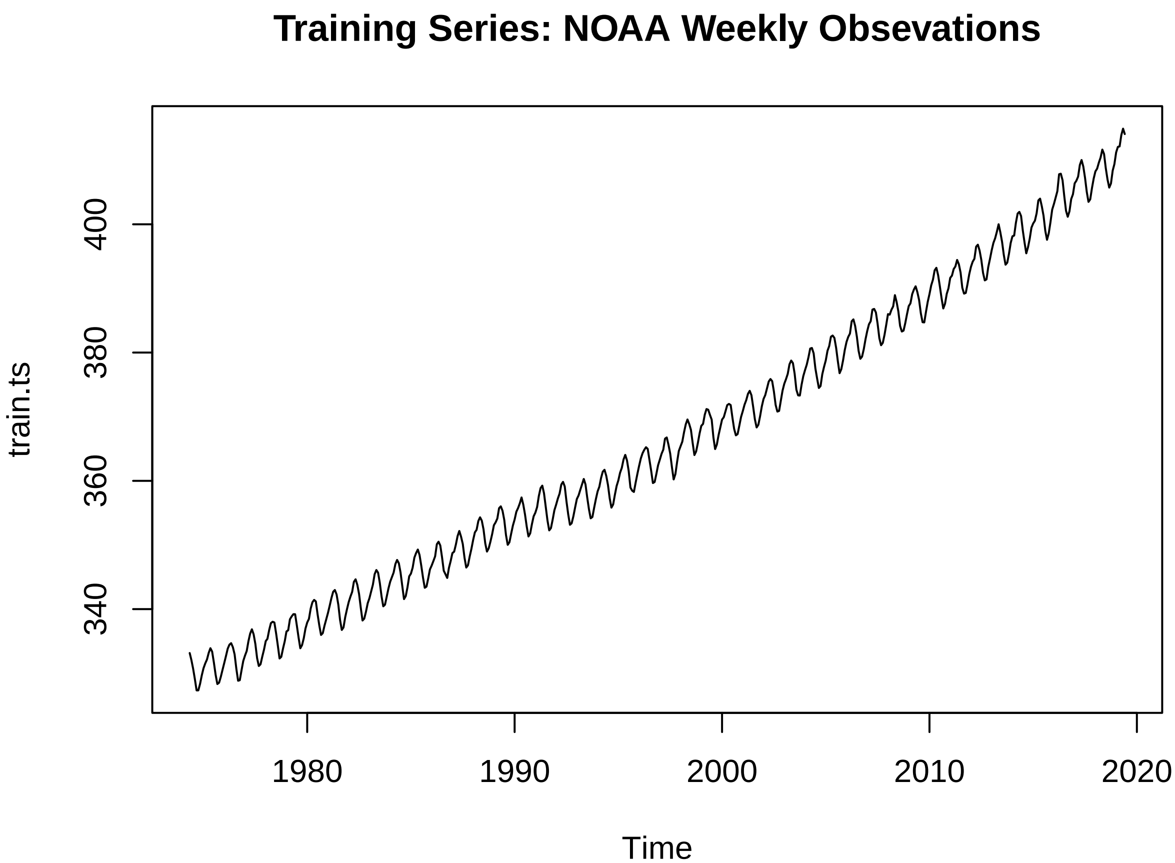
Part 5 (5 points)

Split the NOAA series into training and test sets, using the final two years of observations as the test set. Fit an ARIMA model to the series following all appropriate steps, including comparison of how candidate models perform both in-sample and (psuedo-) out-of-sample. Generate predictions for when atmospheric CO2 is expected to reach 450 parts per million, considering the prediction intervals as well as the point estimate. Generate a prediction for atmospheric CO2 levels in the year 2100. How confident are you that these will be accurate predictions?

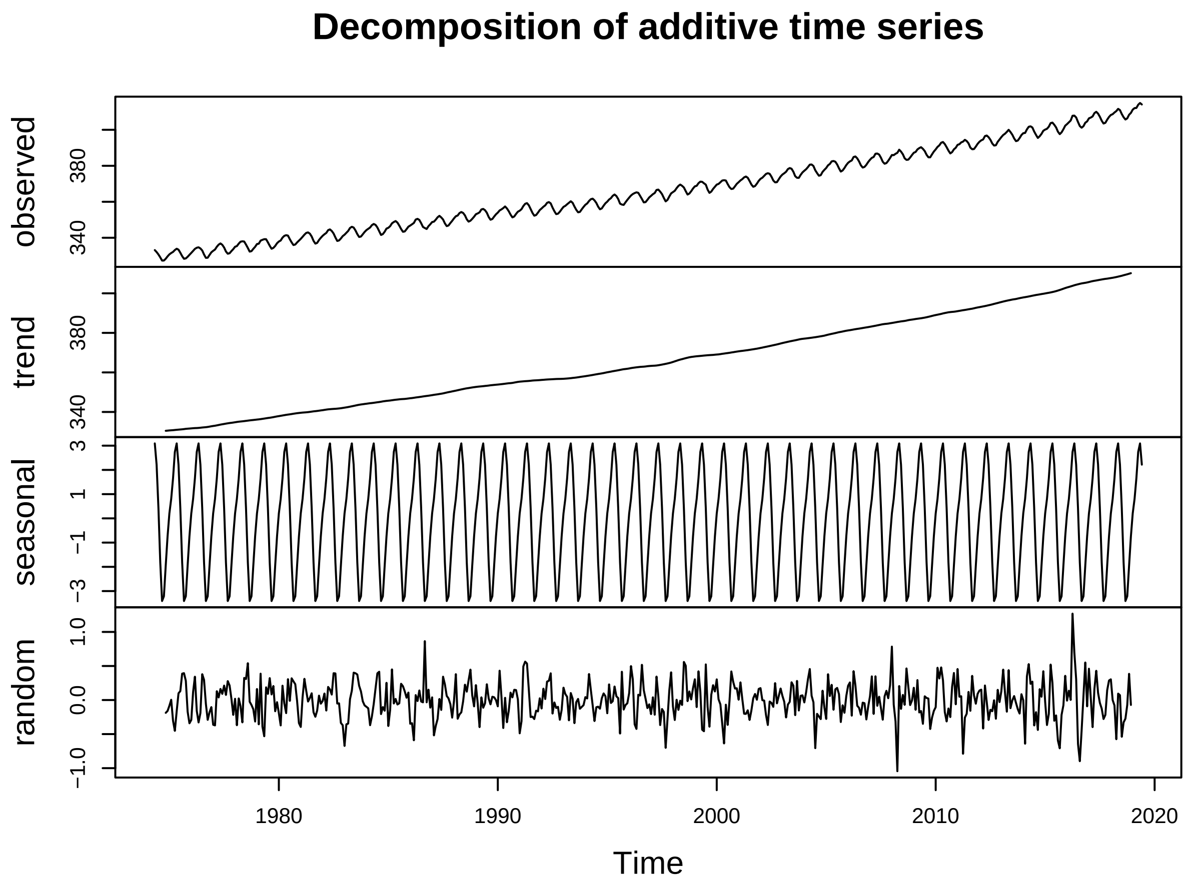
```r
# Extract the relevant time periods for training and
# testing
train.df <- subset(co2_noaa_monthly_df, (year < 2019 & month >=
    1) | (year == 2019 & month <= 6))
```

```r
test.df <- subset(co2_noaa_monthly_df, (year == 2019 & month >
    6) | (year > 2019 & month >= 1))

# Create a time series out of the training and testing
# observations Since it's a monthly series, use a frequency
# of 12
train.ts <- ts(train.df$ppm_month_avg, start = c(1974, 5), end = c(2019,
    6), frequency = 12)
test.ts <- ts(test.df$ppm_month_avg, start = c(2019, 7), frequency = 12)
```

```r
# The time series exhibits an upward trend and seasonality
plot(train.ts, main = "Training Series: NOAA Weekly Obsevations")
```




```r
# Use additive decomposition, since the magnitude of the
# seasonal fluctuation doesn't appear to vary by time
# series level
plot(decompose(train.ts))
```




```
# Find the number of seasonal and non-seasonal differences
# for stationarity 1 seasonal and non-seaonsal differences
# are required
nsdiffs(train.ts)
```

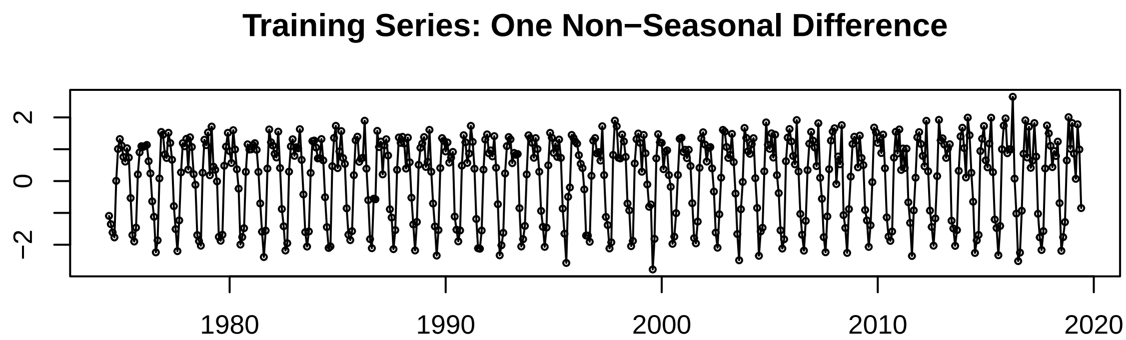
```
## [1] 1
```

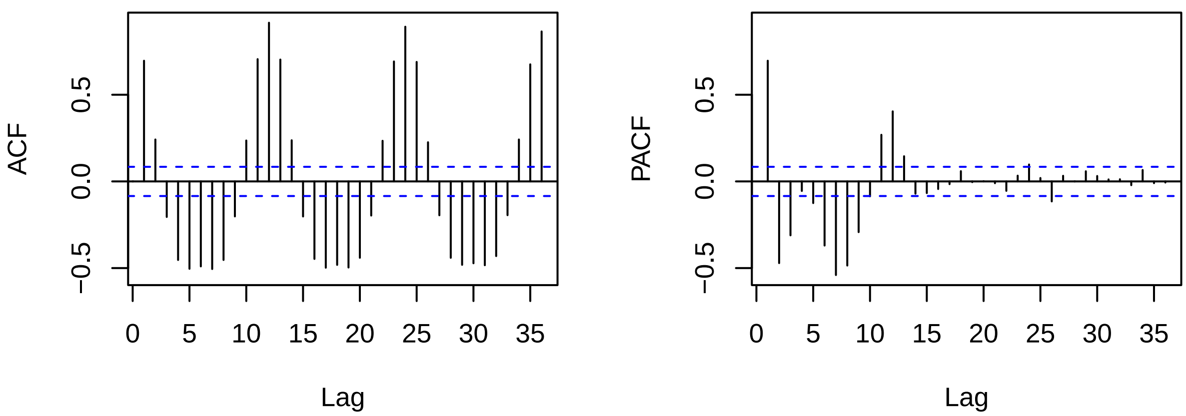
```
ndiffs(train.ts)
```

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

The number of non-seasonal and seasonal differences as indicated by the unit root test is 1.

```
# Plot the ACF and PACF plots after 1 difference, and check
# for stationarity
tsdisplay(diff(train.ts), main = "Training Series: One Non-Seasonal Difference")
```





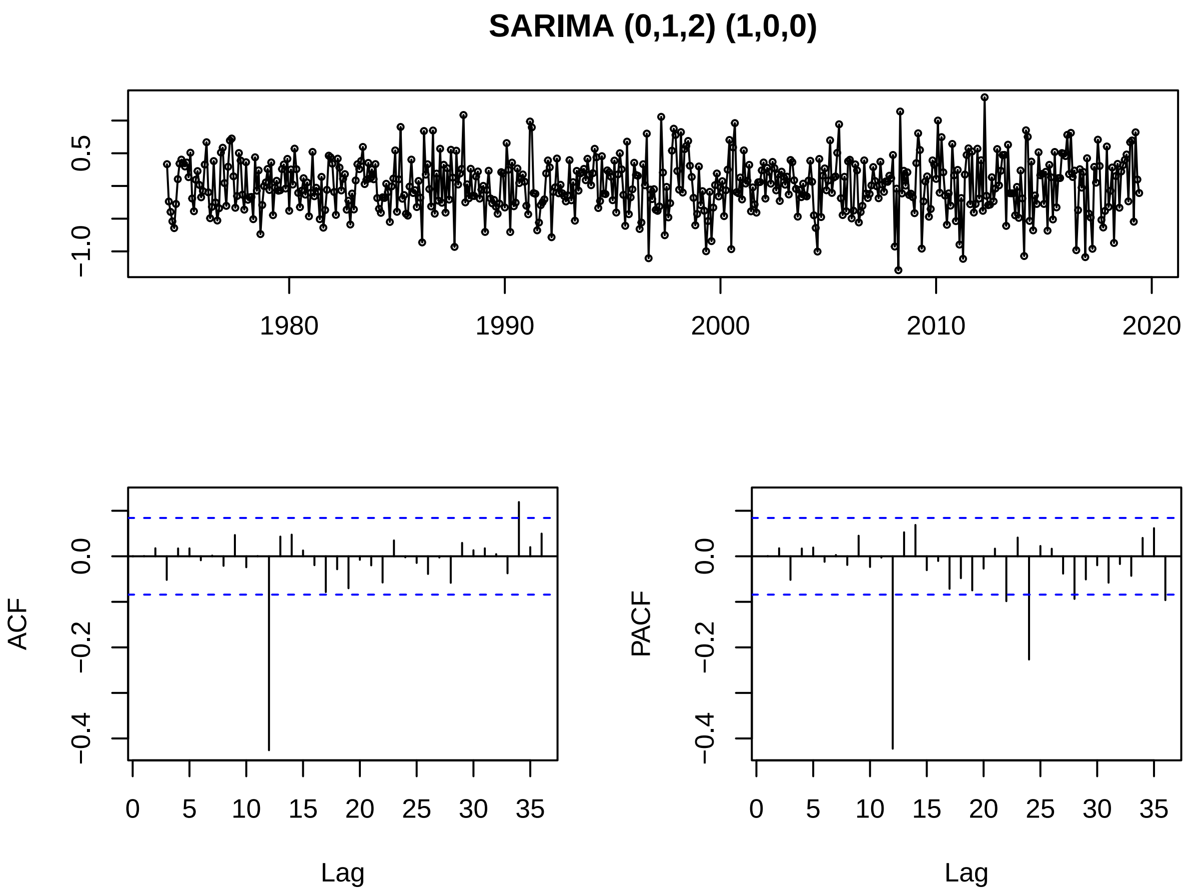
After taking one non-seasonal difference of the weekly NOAA series, we noticed there were fewer repeated seasonal lags in the PACF (only 1 at 12), so we'll use the PACF to estimate the seasonal part of the model (1,0,0). Furthermore, there are 2 significant autocorrelations at lags 1 and 2 in the ACF, so we'll estimate (0,1,2).

```
# There are fewer repeated seasonal lags in the PACF (only
# 1 at 12) so we'll use the PACF to estimate the seasonal
# part of the model (1,0,0) There are 2 significant
# autocorrelations at lags 1 and 2 in the ACF, so we'll
# estimate (0,1,2)
arima.mod1 <- arima(train.ts, order = c(0, 1, 2), seas = list(order = c(1,
    0, 0), frequency(train.ts)), method = "ML")
# Look at the estimated coefficients
summary(arima.mod1)
```

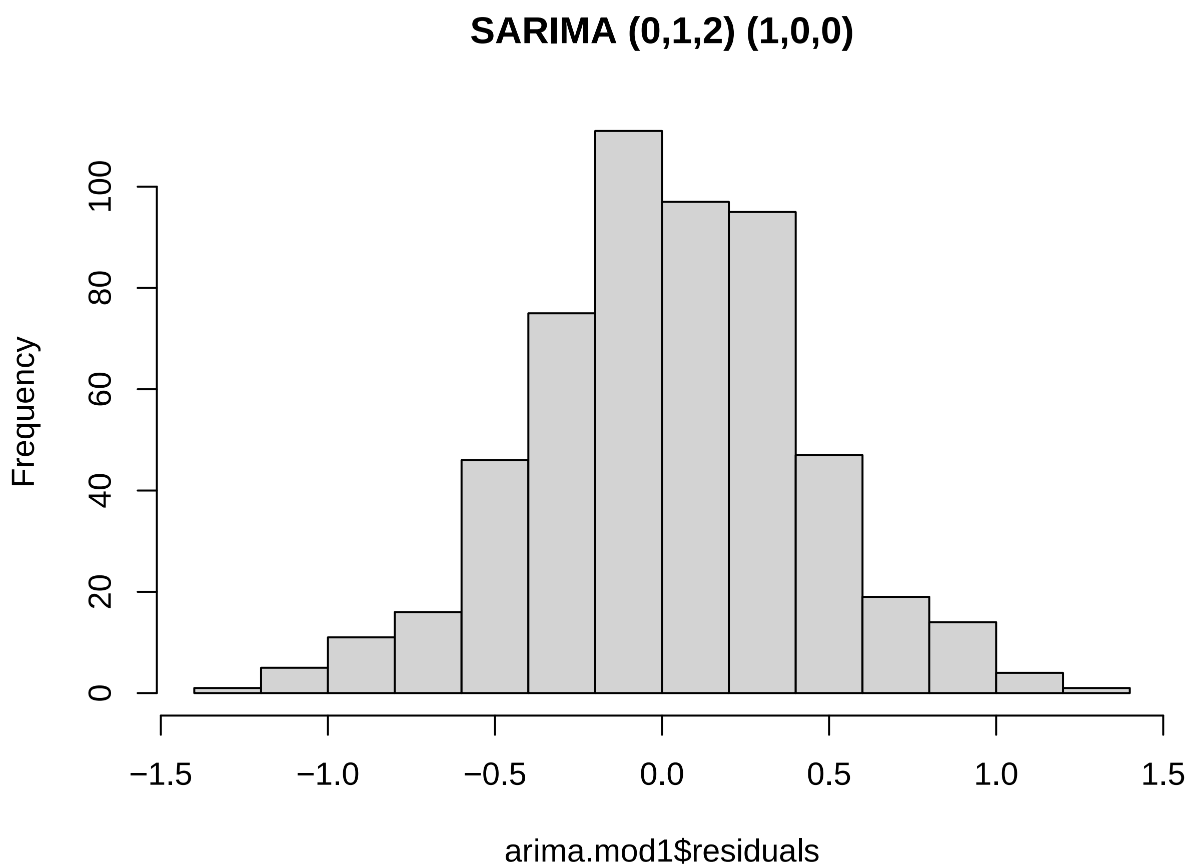
```
##
## Call:
## arima(x = train.ts, order = c(0, 1, 2), seasonal = list(order = c(1, 0, 0),
##     frequency(train.ts)), method = "ML")
##
## Coefficients:
##           ma1      ma2    sar1
##       -0.4017  -0.0340  0.9721
```

```
## s.e.    0.0436   0.0414  0.0079
##
## sigma^2 estimated as 0.1693:  log likelihood = -304.69,  aic = 617.39
##
## Training set error measures:
##                        ME      RMSE       MAE         MPE       MAPE      MASE
## Training set 0.01526793 0.4113246 0.3262361 0.004029418 0.08876883 0.2869071
##                      ACF1
## Training set 0.0005232591
```

```
# The histogram plot looks approximately normal

# A time series plot of the residuals appears to have a
# constant mean The ACF and PACF plots still have a few
# significant autocorrelations
tsdisplay(arima.mod1$residuals, main = "SARIMA (0,1,2) (1,0,0)")
```



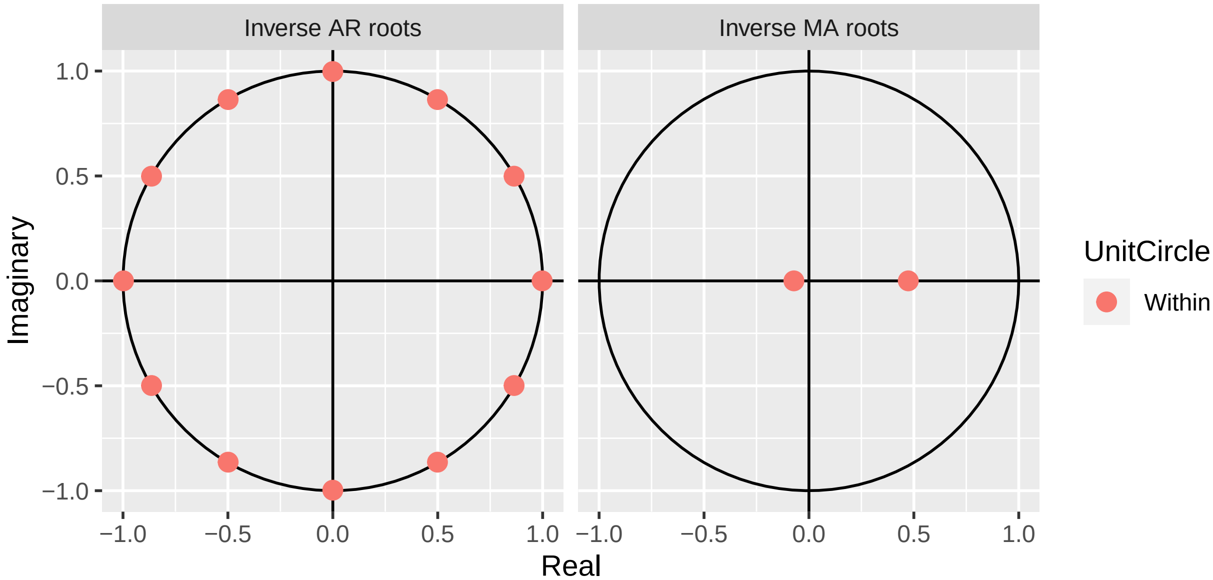

```
hist(arima.mod1$residuals, main = "SARIMA (0,1,2) (1,0,0)")
```




```r
# However, the model passes the Ljung-Box test
Box.test(arima.mod1$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  arima.mod1$residuals
## X-squared = 0.00014922, df = 1, p-value = 0.9903
```

```r
# Check the inverse unit roots for stationarity The inverse
# unit roots are near non-stationarity
autoplot(arima.mod1)
```



The ACF and PACF plots of the residuals of our initial model both show significant correlations at lag 12. Additionally, the PACF plot shows a significant correlation at lag 24. This suggests that we should explore additional seasonal parameters. Similar to the finding the optimal monthly model, we will search for the best seasonal Arima parameters using our `get.best.arima` function.

```
# Minimize AICc
best.mod <- get.best.arima(train.ts, maxord = c(2, 2, 2, 2, 2,
    2))
best10.mods <- head(best.mod[with(best.mod, order(AICc)), ],
    n = 10)
row.names(best10.mods) <- NULL
kable(best10.mods, caption = "Top 10 Models.")
```

Table 2: Top 10 Models.

model	AICc	AIC	BIC
1 1 1 0 1 1	318.4800	318.4037	335.4876
0 1 1 0 1 1	318.5723	318.5266	331.3395
0 1 2 0 1 1	318.5922	318.5159	335.5998
0 1 1 1 1 1	319.5161	319.4397	336.5237
0 1 1 0 1 2	319.5281	319.4517	336.5357
1 1 1 0 1 2	319.5488	319.4341	340.7891
1 1 1 1 1 1	319.5532	319.4384	340.7934
0 1 2 0 1 2	319.6574	319.5427	340.8976
0 1 2 1 1 1	319.6613	319.5466	340.9015
2 1 1 0 1 1	320.5178	320.4031	341.7581

```
pdqPDQ.2 <- as.list(unlist(strsplit(best10.mods[1, 1], "[[:space:]]")))
p.2 <- strtoi(pdqPDQ.2[[1]])
d.2 <- strtoi(pdqPDQ.2[[2]])
```

```r
q.2 <- strtoi(pdqPDQ.2[[3]])
P.2 <- strtoi(pdqPDQ.2[[4]])
D.2 <- strtoi(pdqPDQ.2[[5]])
Q.2 <- strtoi(pdqPDQ.2[[6]])

# Estimate the model
co2.sarima.3 <- arima(train.ts, order = c(p.2, d.2, q.2), seasonal = list(order = c(P.2,
    D.2, Q.2)), method = "ML")
```

Our best sarima model can be expressed as below in the form backshift operator

$$(1 - \phi_1 B)(1 - B)(1 - B^{12})x_t = (1 + \theta_1 B)(1 + \Theta_{12} B^{12})w_t$$

$(1-\phi_1 B)$ represents auto regressive term, $(1+\theta_1 B)$ represents moving average term and $(1+\Theta_{12} B^{12})$ represents seasonal moving average of `arima` model. w_t represents white noise of the time series.

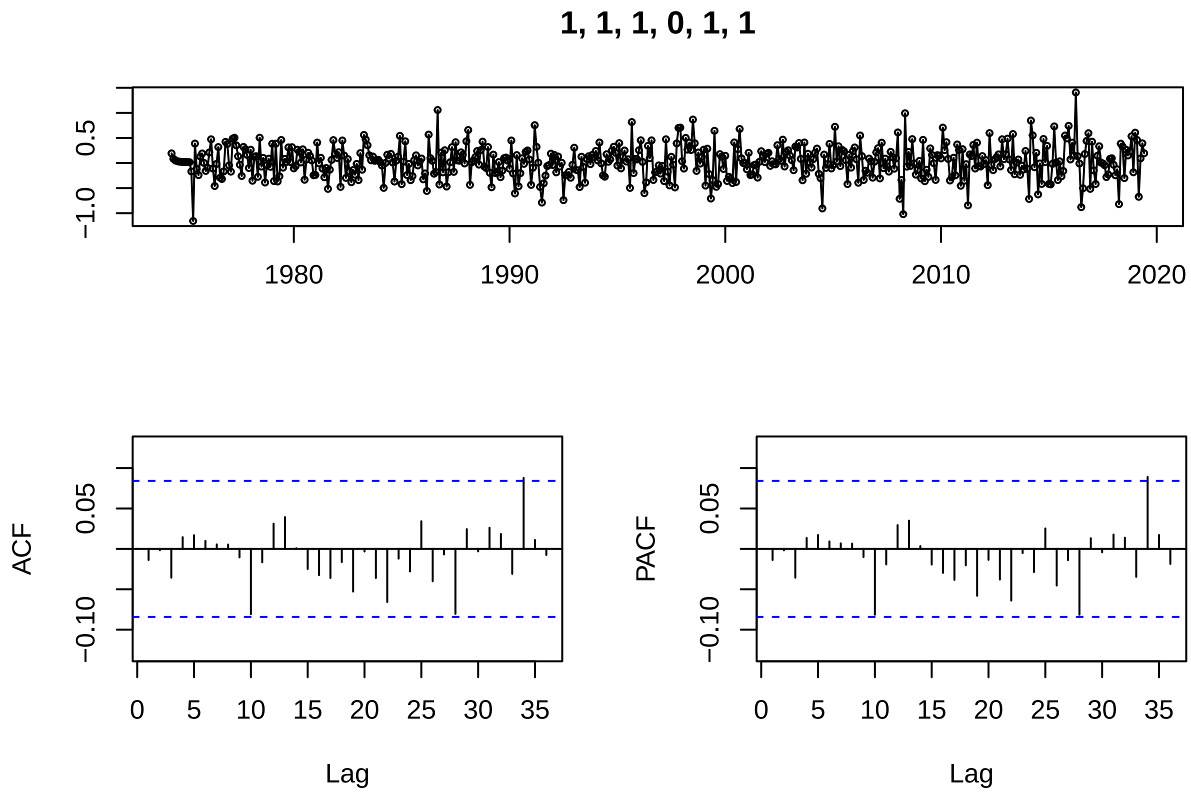
After solving for coefficients using R `arima` model, we get
$x_t = x_{t-1} +$ (-0.5480189) $* x_{t-1} + w_t +$ (-0.5480189) $* w_{t-1} +$ (-0.8691625) $* w_{t-12}$

where x_{t-1} is the results of first difference of time series i.e. $x_t^1 = x_t - x_{t-1}$ and second x_{t-1} is the result of auto regressive term.
w_t is the white noise from the current time step, w_{t-1} is white noise from the previous time step, which is the result of the AR moving average. w_{t-12} is the white noise from 12 steps before (i.e. seasonal) current time step. This is the result of the seasonal moving average component of our model.

```r
# Inspect the residual plots and find the estimated AICc
Box.test(co2.sarima.3$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  co2.sarima.3$residuals
## X-squared = 0.10304, df = 1, p-value = 0.7482
```

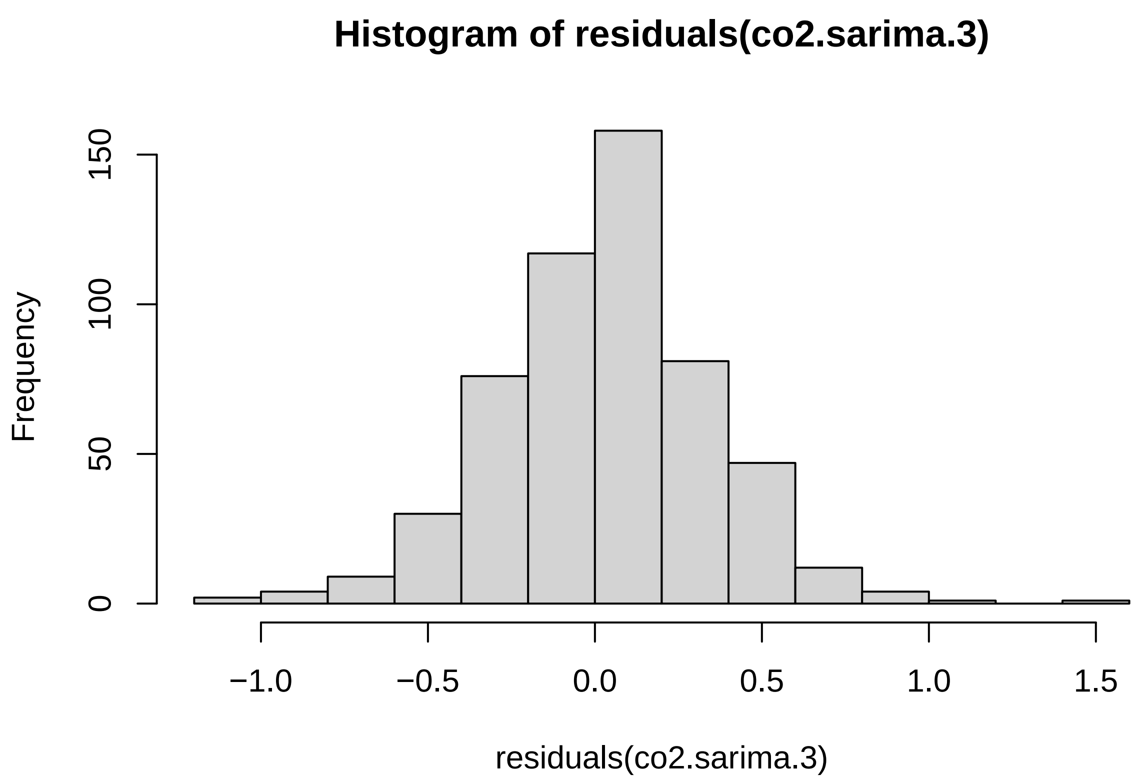
```r
sarima3.aicc <- -2 * co2.sarima.3$loglik + (log(length(co2)) +
    1) * length(co2.sarima.3$coef)
tsdisplay(co2.sarima.3$residuals, main = {
    toString(pdqPDQ.2)
})
```



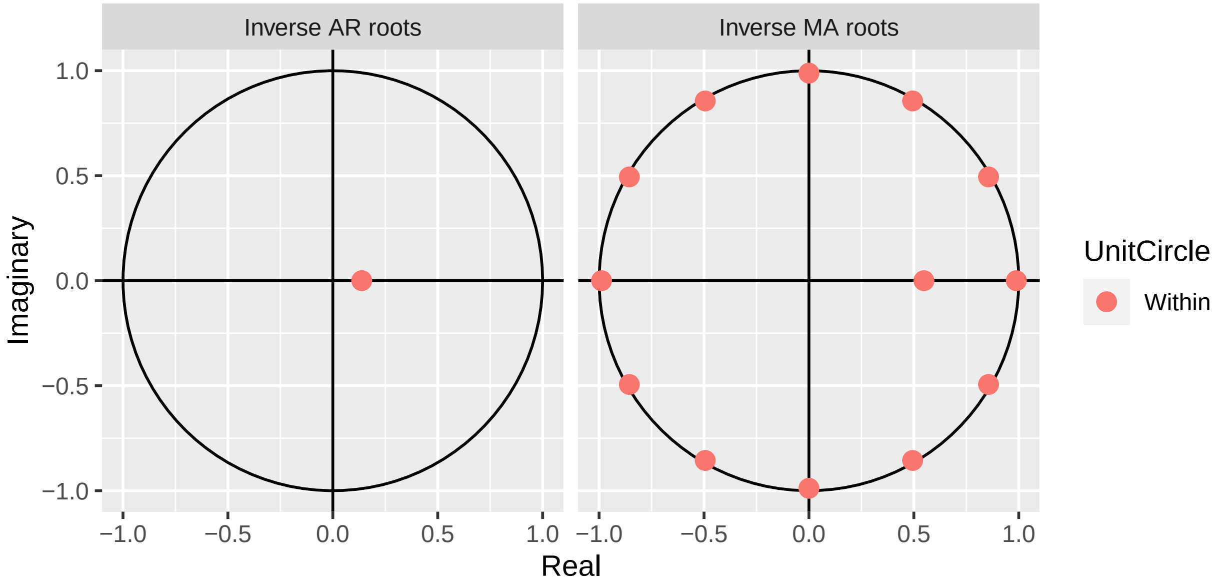

```r
sarima3.aicc
```

```
## [1] 331.8491
```

```r
hist(residuals(co2.sarima.3))
```



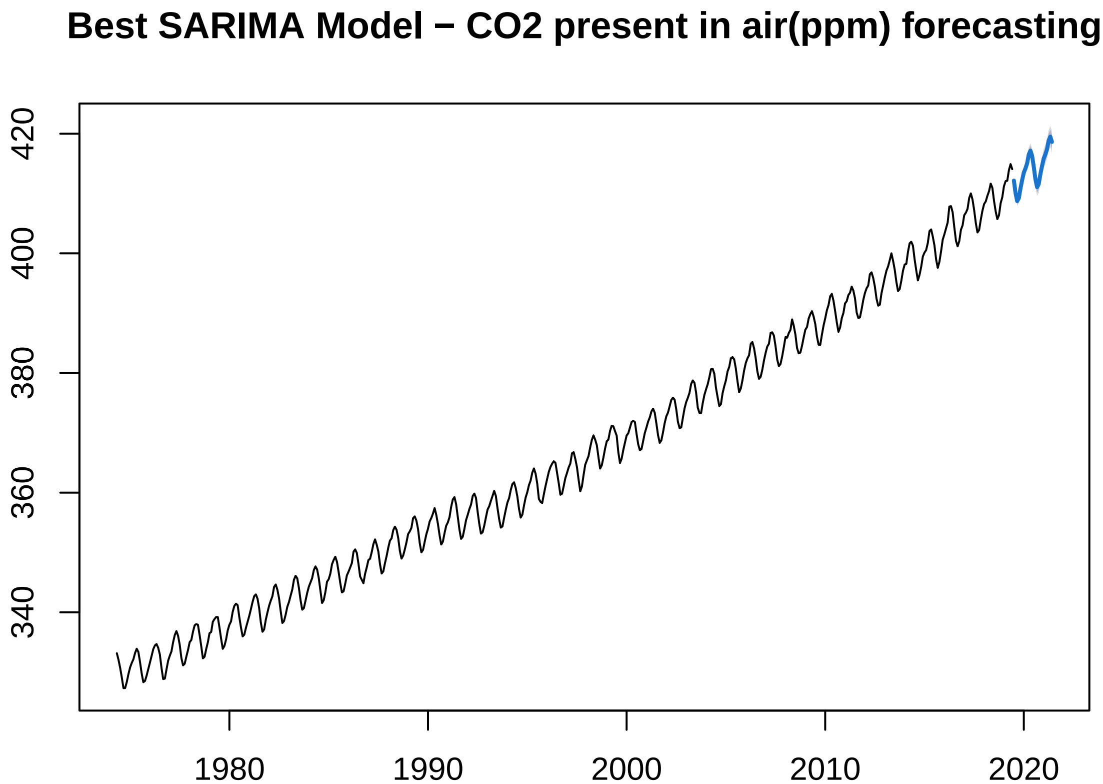
```
autoplot(co2.sarima.3)
```



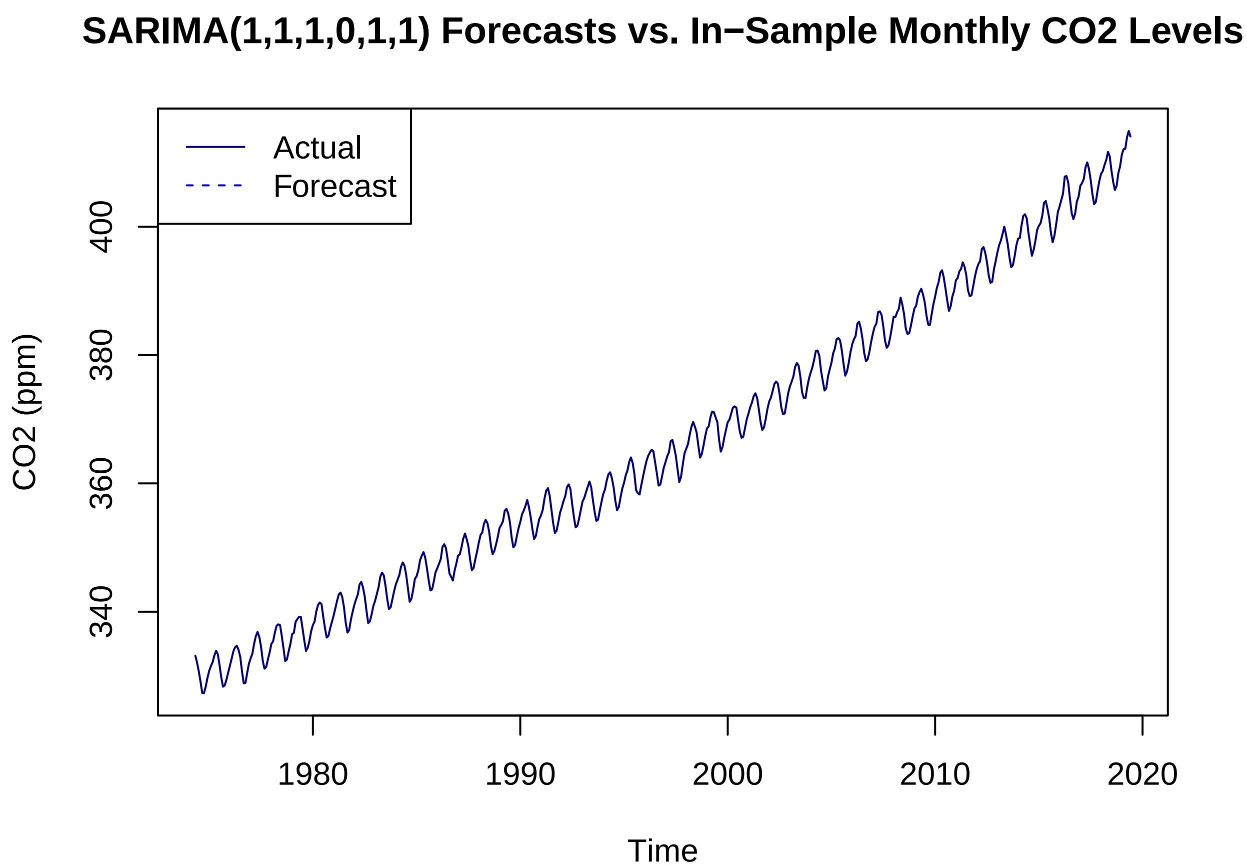
The ACF and PACF plots of the best model do not show any significant autocorrelations, indicating that the residual series could be similar to white noise. Additionally, the high p-value of the Ljung-Box test suggests that we cannot reject the null hypothesis that the series is independently distributed. Lastly, the roots are all positioned within the unit circle. As such, we will use this model to predict CO2 concentrations over the next 80 years.

Best Model Forecasts

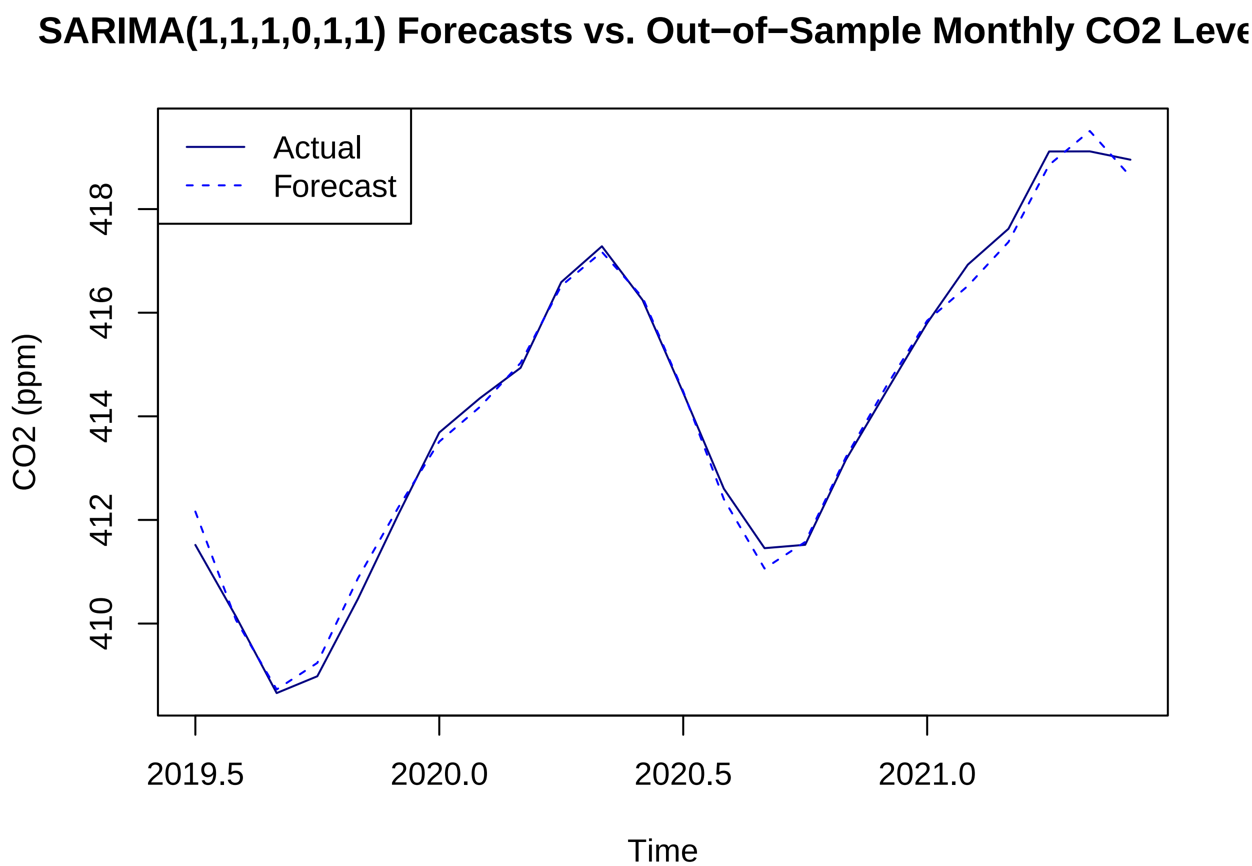
```
co2.forecast.24mo <- forecast(co2.sarima.3, 24)  # 24 month forecast
co2_forecast_ts24mo <- co2.forecast.24mo[4]$mean
plot(co2.forecast.24mo, main = "Best SARIMA Model - CO2 present in air(ppm) forecasting")
```



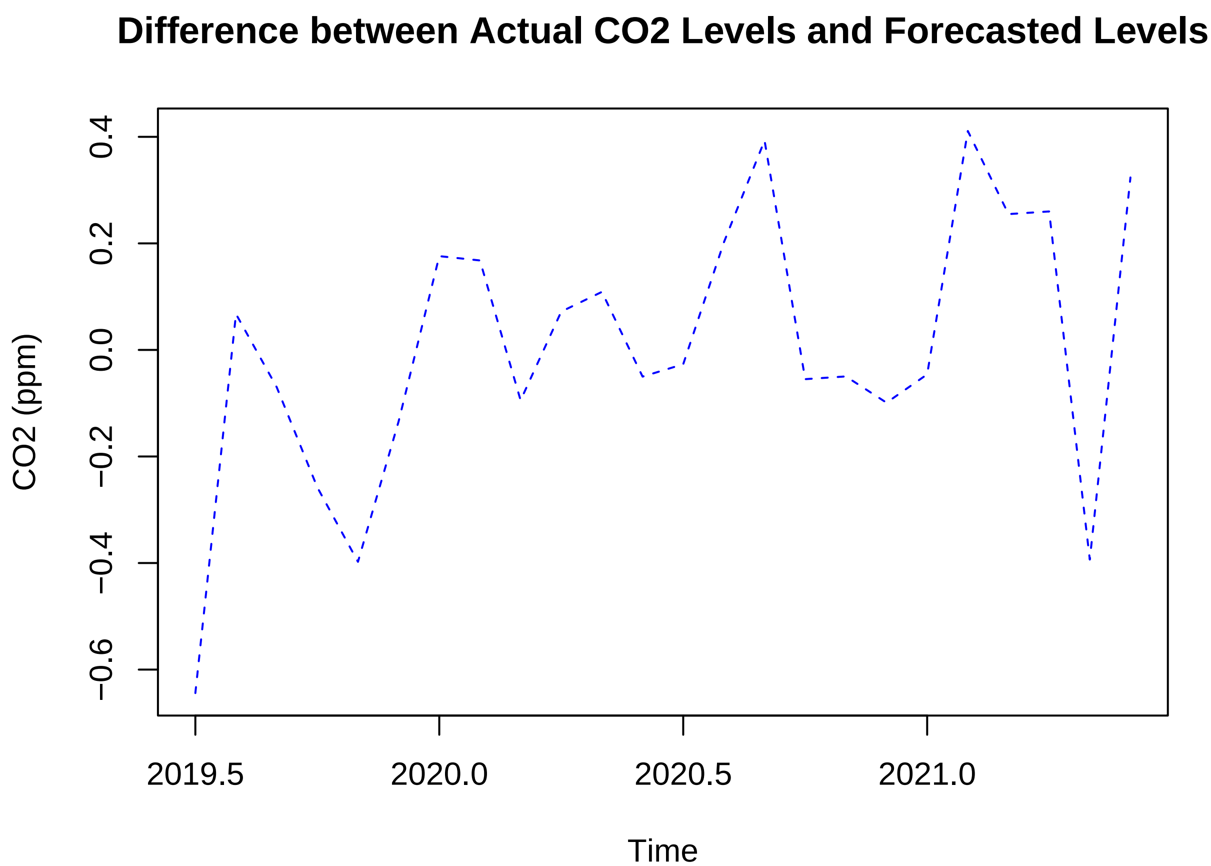

```r
# Plot forecast vs. test data
ts.plot(train.ts, co2.sarima.3$fit, lty = 1:2, col = c("navy",
    "blue"), ylab = "CO2 (ppm)", main = "SARIMA(1,1,1,0,1,1) Forecasts vs. In-Sample Monthly C
legend("topleft", legend = c("Actual", "Forecast"), col = c("navy",
    "blue"), lty = 1:2)
```




```
# Plot forecast vs. test data
ts.plot(test.ts, co2_forecast_ts24mo, lty = 1:2, col = c("navy",
    "blue"), ylab = "CO2 (ppm)", main = "SARIMA(1,1,1,0,1,1) Forecasts vs. Out-of-Sample Monthl
legend("topleft", legend = c("Actual", "Forecast"), col = c("navy",
    "blue"), lty = 1:2)
```



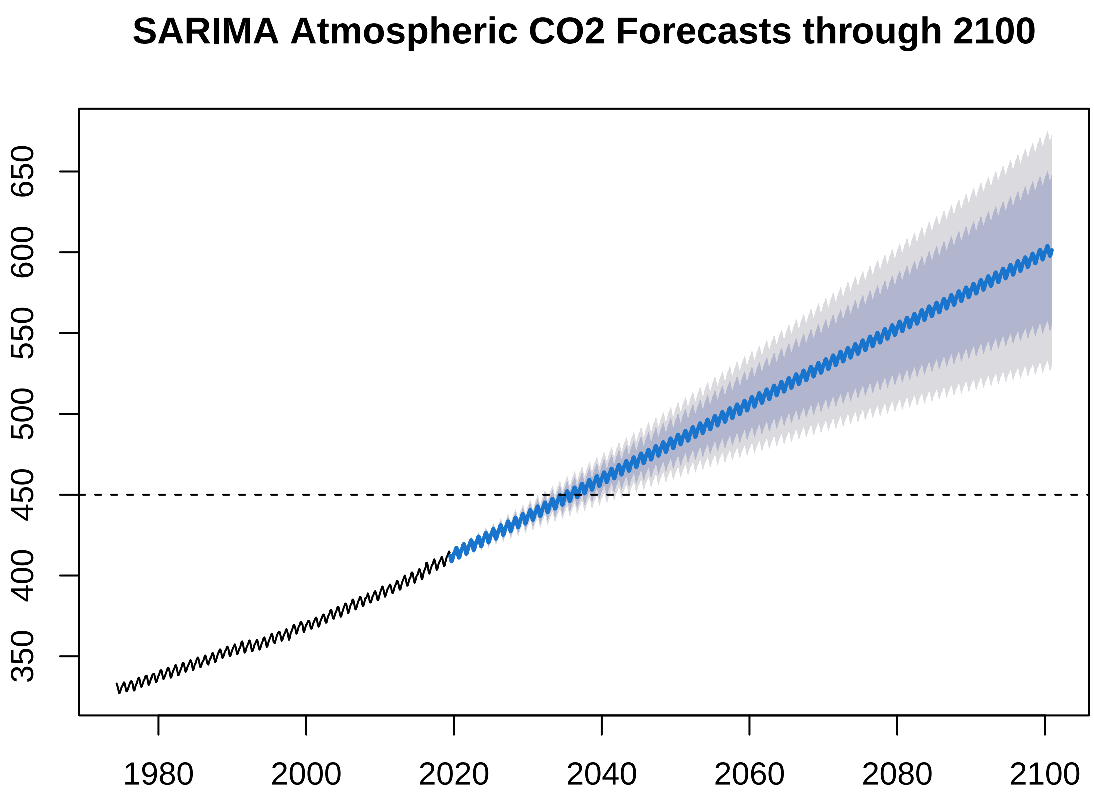

```
actuals_fore_diff2 <- test.ts - co2_forecast_ts24mo
ts.plot(actuals_fore_diff2, lty = 2, col = c("blue"), ylab = "CO2 (ppm)",
    main = "Difference between Actual CO2 Levels and Forecasted Levels")
```



The out-of-sample predictions followed the actual series closely. Given that this is only a 24-month forecast, this does not seem unreasonable. The observed difference is also fairly stationary, unlike our earlier predictions, which clearly did not capture the increasing growth of CO2 levels of the 2000-2019 period.

Generate predictions for when atmospheric CO2 is expected to reach 450 parts per million, considering the prediction intervals as well as the point estimate. Generate a prediction for atmospheric CO2 levels in the year 2100. How confident are you that these will be accurate predictions?

```
mo_to_forecast <- (2100 - 2019) * 12 + 6
co2.forecast.2100 <- forecast(co2.sarima.3, mo_to_forecast)  # 24 month forecast
co2_forecast_ts2100 <- co2.forecast.2100[4]$mean
lower.bound.2100 <- co2.forecast.2100$lower[, 2]  # 95% confidence
upper.bound.2100 <- co2.forecast.2100$upper[, 2]
plot(co2.forecast.2100, main = "SARIMA Atmospheric CO2 Forecasts through 2100")
abline(h = 450, lty = 2)
```




```
co2_forecast_ts2100 <- co2.forecast.2100[4]$mean
```

We forecast that atmospheric CO2 would exceed 600 ppm by December 2100, and that the level of CO2 will be between 529 and 673 ppm with 95% confidence. This is nearly 43% higher than current CO2 levels! It is extremely important to note, however, that these forecasts assume the current CO2 generating process will be the same over the next 80 years as it has been over the past 45 years. This seems highly unlikely as there is clear evidence that carbon emissions impact the environment and climate negatively, and there is significant momentum among most of the world's governments to make a coordinated effort to combat climate change[2]. Ultimately, we hope that our forecasts are only accurate in the very immediate term, and over time, the growth of CO2 concentrations decelerates, and CO2 levels eventually decline.

```
#################################### get upper confidence
#################################### interval data
upper_data = co2.forecast.2100$upper
# at the 95% confidence level
upper_data_95 = upper_data[, "95%"]
# find the first interval covering 450ppm
first_95_int_w450 = which(upper_data_95 >= 450)[1]
time_upper_95 = as.yearmon(time(co2.forecast.2100$upper))[first_95_int_w450]

#################################### get point estimate
#################################### data
pt_data = co2.forecast.2100[4]$mean
# and where it hits at least 450ppm
```

[2]https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement

```r
pt_first_idx_w450 = which(pt_data >= 450)[1]
time_pt_first_pt = as.yearmon(time(pt_data))[pt_first_idx_w450]

#################################### get lower confidence
#################################### interval data
lower_data = co2.forecast.2100$lower
# at the 95% confidence level
lower_data_95 = upper_data[, "95%"]
# find the first interval covering 450ppm
first_95_int_w450 = which(lower_data_95 >= 450)[1]
time_lower_95 = as.yearmon(time(co2.forecast.2100$upper))[first_95_int_w450]

print(paste("Based on inspecting the forecast results, the first time the upper-confidence inte
    time_upper_95, ".", sep = ""))
```

```
## [1] "Based on inspecting the forecast results, the first time the upper-confidence interval
```

```r
print(paste("Furthermore, based on inspecting the forecast results, the point estimate predicti
    time_pt_first_pt, ".", sep = ""))
```

```
## [1] "Furthermore, based on inspecting the forecast results, the point estimate prediction re
```

```r
print(paste("Lastly, the first time the lower-confidence interval (at the 95% level) reaches 45
    time_upper_95, ".", sep = ""))
```

```
## [1] "Lastly, the first time the lower-confidence interval (at the 95% level) reaches 450 is
```

Finally, upon inspecting the forecast's confidence intervals' upper bounds, we observe that the first time that it includes 450ppm is in March 2032 ; additionally, the first time that the main prediction itself meets or exceeds 450ppm is 3 years later in March 2035.