Atmospheric Carbon Dioxide Forecasting

Aidan Jackson, Sandip Panesar

Contents

Investigation and Forecasting of Carbon Dioxide in the Atmosphere

Summary

This exercise investigates concentration measurements of carbon dioxide (CO_2) in the Earth's atmosphere, and uses traditional time series methods to forecast their levels into the future.

This work was originally completed as part of the W271 Statistical Methods for Discrete Response, Time Series, and Panel Data course in the Master of Information and Data Science program at University of California, Berkeley.

Introduction - The Keeling Curve

The following introduction is reproduced with minor differences from the original academic exercise:

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 parts per million (ppm) at the Mauna Loa Observatory from 1959 to 1997. The curve graphed by this data is known as the 'Keeling Curve'.

```
plot(co2, ylab = expression("CO"[2]*" Concentration (ppm)"),
     xlab = expression("Time"),
     col = 'blue', las = 1)
title(main = expression("Figure 1. Keeling Curve - Monthly Mean CO"[2]*" Variation"))
```

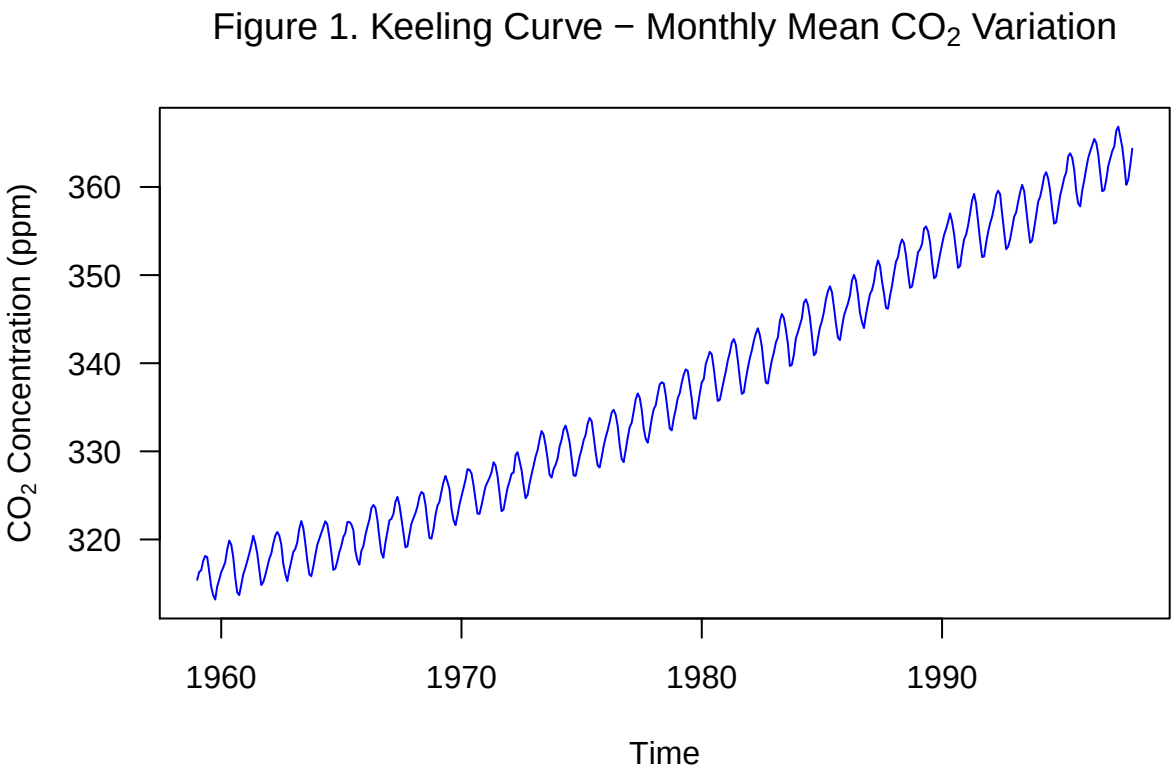


Figure 1, above, shows the Keeling Curve as generated by the built-in R dataset. In this exercise, this time series will be modeled and varying forecasts into the future will be produced.

Exploratory Data Analysis

```
library(forecast)
library(ggplot2)
library(dplyr)
library(lubridate)
library(zoo)
library(gridExtra)
```

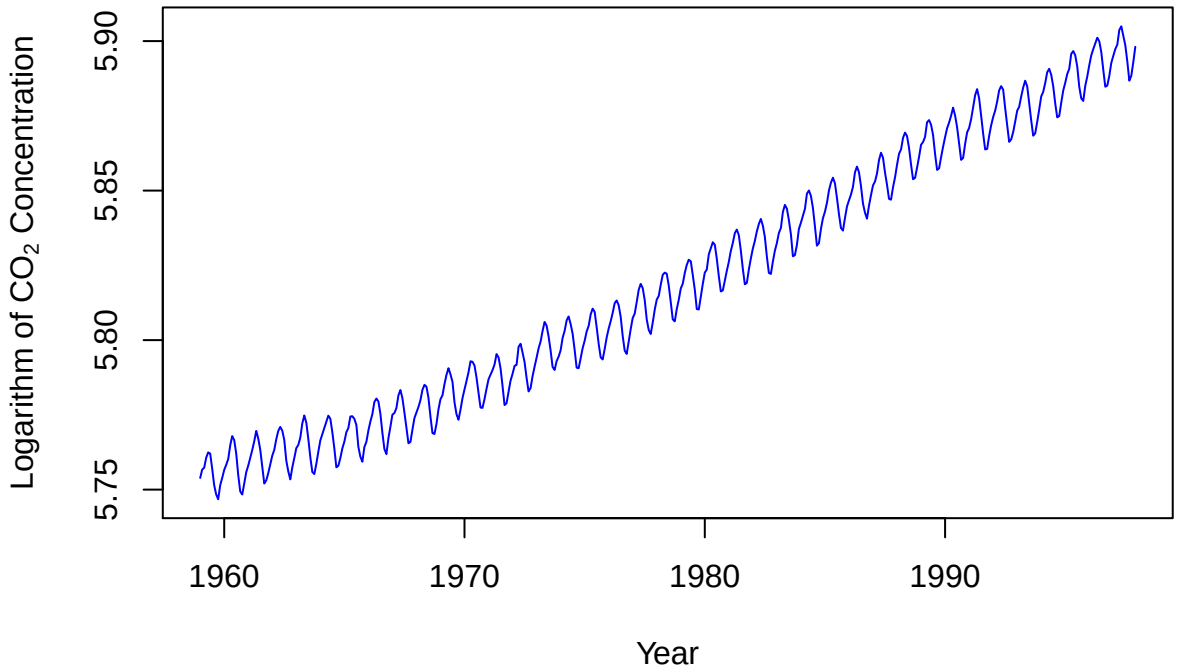
Unlike traditional statistics, time series modeling includes additional assumptions and unique model structures not found in problems where there is no time involved component. These must be investigated before the time series data is used, regardless of whether the goal is to produce a forecast or solely examine relationships within the data.

To begin, it can be found that there are 0 missing values. However, when inspecting the documentation of the dataset, it was noted that the months of February - April of 1964 were missing and filled in with linear interpolation. For the purposes of this exercise, this three measurements will be left as is. Overall, the measurements take place from Jan 1959 to Dec 1997, following the description that Keeling began in 1958.

From **Figure 1**, it is obvious that there is a positive trend that makes the series non-stationary in the mean. The mean of the series is about 337.1 ppm, which is clearly greater than the start of the series and lower than the end. It also appears that the variance could be increasing over time, with cycles being of greater amplitude later in the series than in earlier sections. Across the entire series, the standard deviation is about 15 ppm, which also appears greater than the amplitude of early cycles and closer in size to later cycles. This makes the series non-stationary in the variance as well. To address this increasing variance, the logarithm may be taken and examined.

```
d <- log(co2)

plot(d, xlab = "Year", ylab = expression("Logarithm of CO"[2]*" Concentration"), col = 'blue',
     main = "Figure 2. Logarithmic Time Series", type = "l")
```

Figure 2. Logarithmic Time Series



Shown in **Figure 2**, taking the logarithm appears to have better stabilized the variance over the course of the time series. Now the data can be considered stationary in the variance, but the positive time trend must still be examined. It appears as though the series could either have a first order or second order time trend, so both will be investigated.

```
d1 <- diff(d)
d2 <- diff(d1)

par(mfrow = c(1, 2), cex = 0.6)
plot(d1, xlab = "Year", ylab = "First Differenced Log Value",
     main = "Figure 3. First Differenced Time Series", type = "l")
abline(h = mean(d1), lty = 2, col = "red")
plot(d2, xlab = "Year", ylab = "Second Differenced Log Value",
     main = "Figure 4. Second Differenced Time Series", type = "l")
abline(h = mean(d2), lty = 2, col = "red")
```

Figure 3. First Differenced Time Series

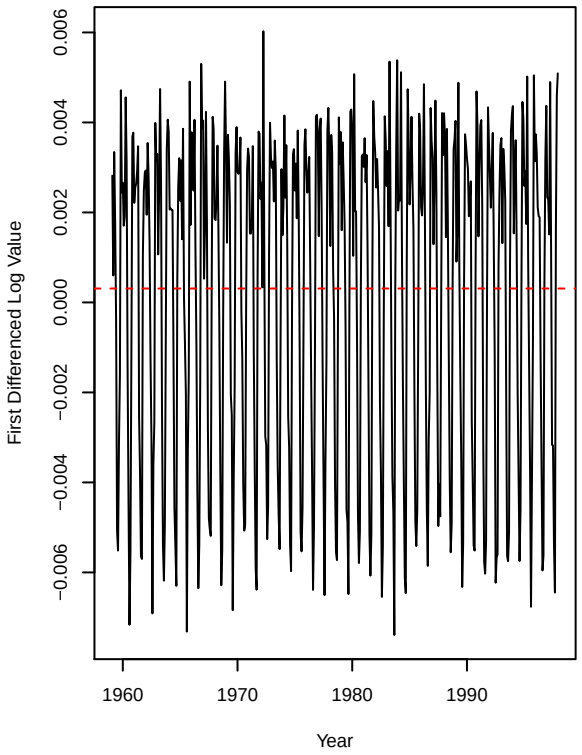
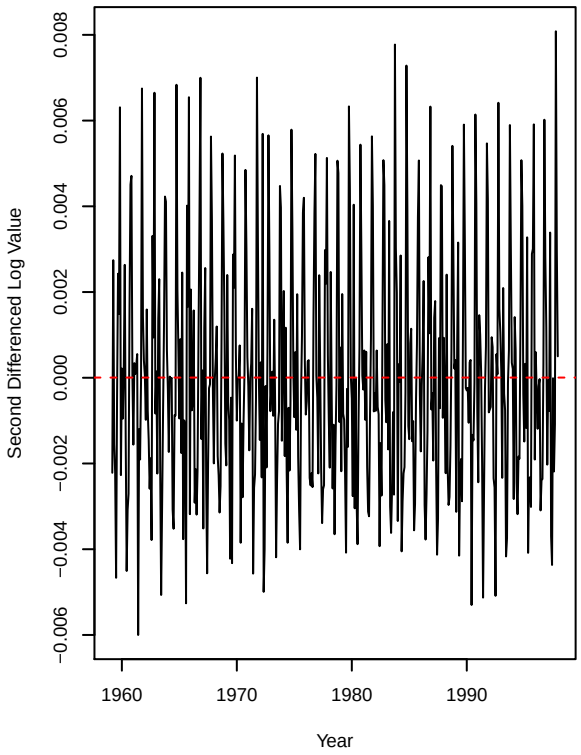


Figure 4. Second Differenced Time Series



Figures 3 and **4** show the first and second differenced times series after applying the previous logarithm transformation. It can be seen in **Figure 3** that the first difference does much to remove the positive time trend, but the majority of the data still appears to slightly fluctuate in the mean over time. For example, the second half of the data appears to have a slightly higher mean value than the first half. On the other hand, **Figure 4** shows that after taking the second difference, the data appears to completely be stationary in the mean. Therefore, any model with this data should also take the second difference to achieve this.

Next, the seasonality and other elements of the data will be investigated.

```
par(mfrow = c(1, 2), cex = 0.6)
plot(acf(d2, plot = FALSE), main = "")
title("Figure 5. Autocorrelation Plot")
plot(pacf(d2, plot = FALSE), main = "")
title("Figure 6. Partial Autocorrelation Plot")
```

Figure 5. Autocorrelation Plot

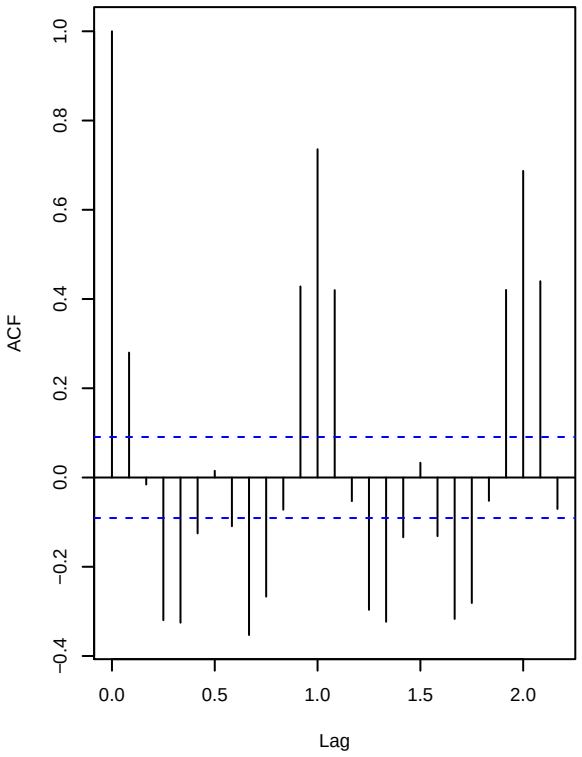
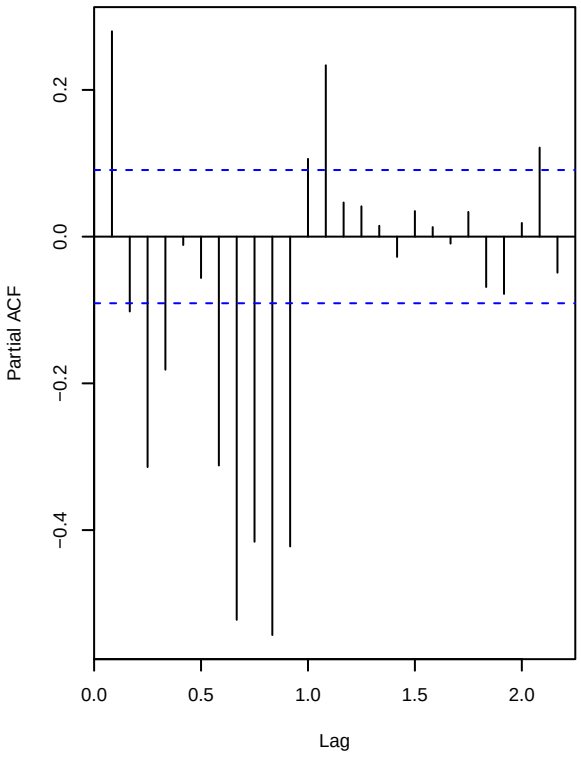


Figure 6. Partial Autocorrelation Plot



Figures 5 and **6** show the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the data after the logarithm transformation and second differencing. It can be seen there is a significant lag at $k = 1$ and $k = 2$, measured in years. In between these points (and also for the subsequent interval) appears to be a repetitive oscillatory pattern indicating that there is seasonality to the data. This fits with Keeling's original hypothesis that stated annual cyclic changes in the environment and vegetation would affect CO_2 concentrations in the atmosphere. Interestingly, the pattern in **Figure 5** assumes a shape where whole number lags are significantly positive but intermediate lags are significantly negative, and with decreasing magnitudes over later time-period shifts. Similar to **Figure 6**, these oscillations appears to be decreasing in magnitude over time. These repeating patterns may explain seasonality in the time series, rather than being indicative of the non-seasonal autoregressive (AR) or moving average (MA) components. Full investigation of AR/MA components of the data will be continued later when exploring ARIMA models.

Linear Time Trend Modeling with Keeling Data

Before more complicated time series methods, a simple linear time trend can be used as a baseline against which to compare future iterations. A linear time trend is a specific type of linear time series model where the only independent variable in the model is time. That is, the model would take the form of:

$$x_t = \alpha_0 + \alpha_1 * t + \alpha_2 * t^2 + ... + \alpha_n * t^n$$

for an n-th ordered polynomial with time series x_t and model parameters α as a linear time trend. Note that the *linear* description of the model comes from the linear relationship of model parameters, and not the relationship with time itself. In this report, if the relationship with time is

only of the first order, then the model will also be referred to as a Linear Model. If the relationship is of a higher order, then the model will be referred to as a Polynomial Model.

While in the EDA it was found that a logarithm and differencing operation was needed to make the time series stationary, linear time trends do not rely on any assumption of stationarity. Therefore, to start, a linear time trend of the first order will be fit to the raw data.

```
Time <- time(co2)
linear_model <- lm(co2 ~ Time)

plot(co2, xlab = "Year", ylab = expression("CO"[2]*" Concentration (ppm)"), main = "Figure 7. F
     type = "l")
lines(co2 + linear_model$resid, col = "blue")
legend('bottomright', legend=c("Original", "Linear Model"), col=c("black","blue"),
       lty=1)
```

Figure 7. Fitted Linear Model

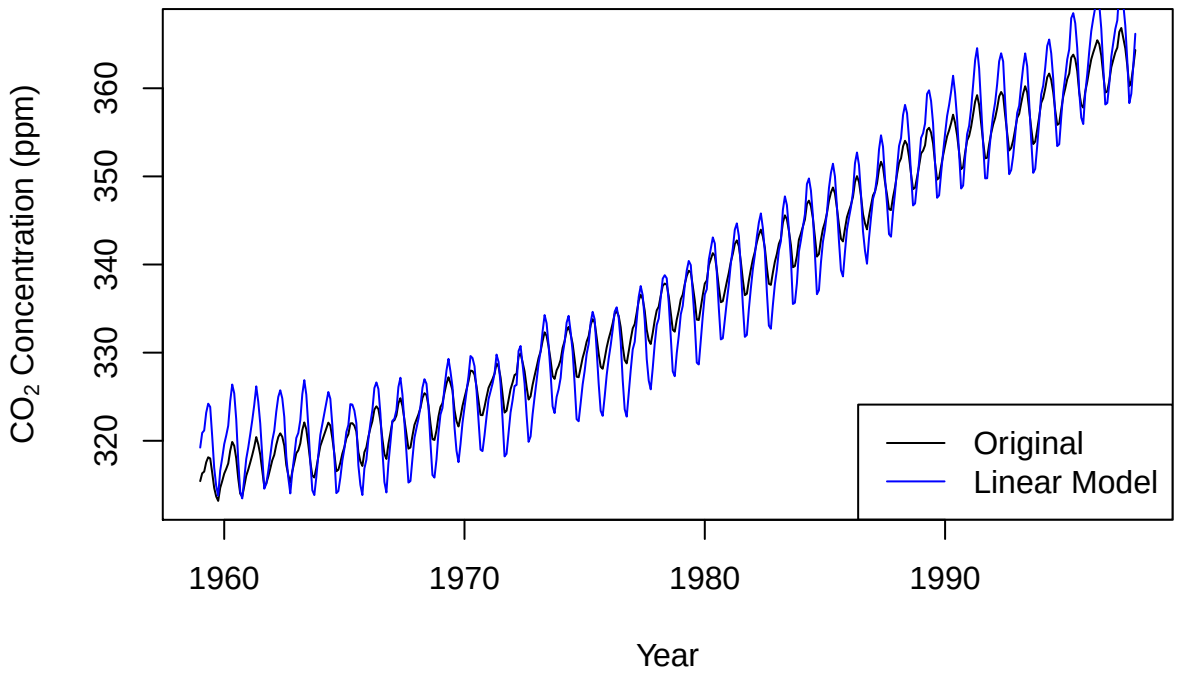
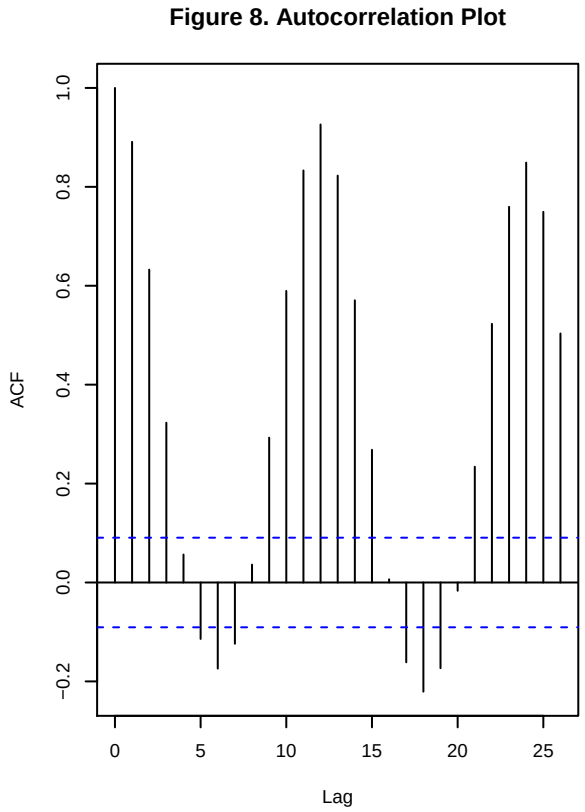


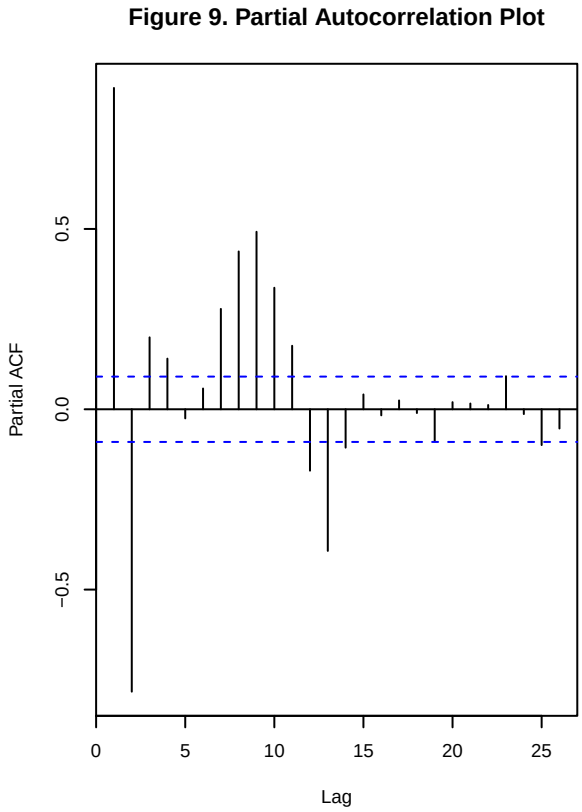
Figure 7 displays the fitted linear model of the first order overlaid on the original data. It can be seen that the linear model generally overestimates the amount of variance in the data, with cyclic amplitudes that are much greater than in the original time series. However, it is able to estimate the frequency of the cycles quite well, with an almost exact overlap between the crests and troughs of each series.

The correlation of the residuals produced by the model may be examined for an evaluation of the fit.

```
par(mfrow = c(1, 2), cex = 0.6)
plot(acf(linear_model$residuals, plot = FALSE), main = "")
```

```
title("Figure 8. Autocorrelation Plot")
plot(pacf(linear_model$residuals, plot = FALSE), main = "")
title("Figure 9. Partial Autocorrelation Plot")
```



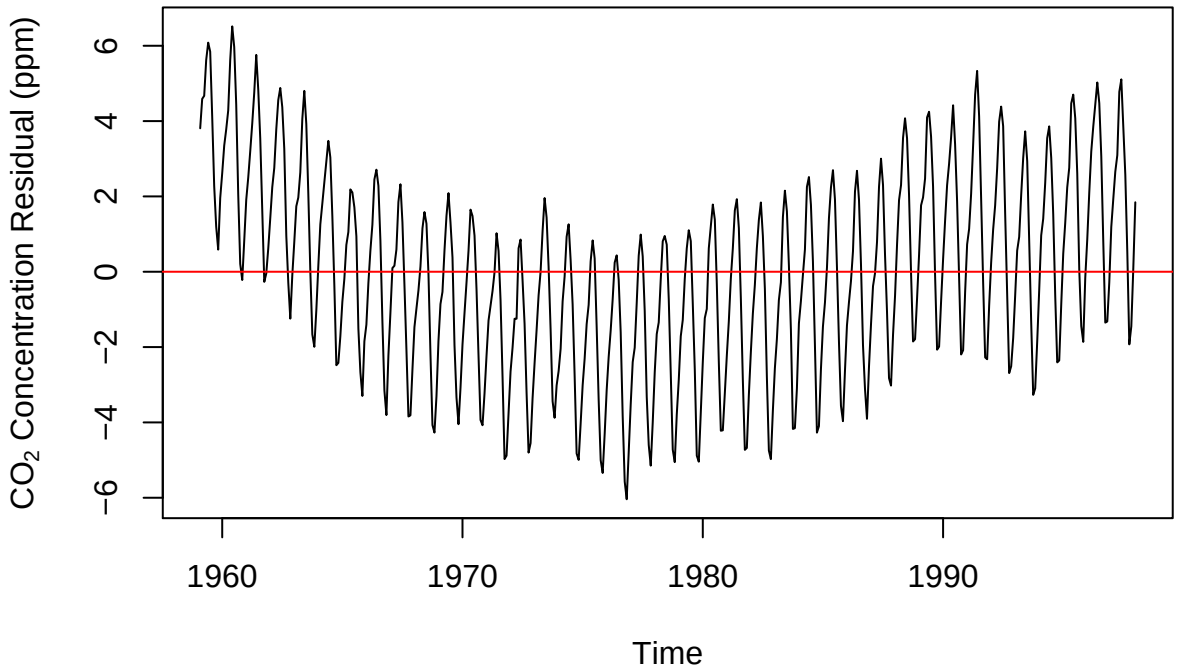


Figures 8 and **9** show the ACF and PACF of the model's residuals, with the lags shown in months. A slightly damped oscillatory pattern can be observed in both, similar to **Figures 5** and **6**. This indicates that the model does not account well for the seasonal variation which is present in the original data. Based on **Figure 8** and Keeling's hypothesis, these appear to have a period of 12 months.

Finally, the residuals can be viewed on their own over time.

```
plot(linear_model$residuals,
     xlab = "Time", ylab = expression("CO"[2]*" Concentration Residual (ppm)"), xaxt = "n",
     main = "Figure 10. Fitted Linear Model Residuals", type = "l")
axis(1, at=c(12,132,252,372), labels=c(1960, 1970, 1980, 1990))
abline(h = mean(linear_model$resid), col = "red")
```

Figure 10. Fitted Linear Model Residuals



Clearly, the residuals in **Figure 10** do not resemble a white noise series. At earlier and later decades, the residuals are consistently higher than the mean. In the middle of the time series, however, they are consistently less than the mean. This demonstrates that there are still components in the original time series that have not been accounted for by the linear model. It is also a reflection of what a first order linear model may achieve on non-stationary data, where the two sole model parameters only capture the average trend over the data. When this trend is not constant, the fit over the data is uneven.

For a better fit, a polynomial time trend can be estimated.

```
poly_model <- lm(co2 ~ Time + I(Time^2) + I(Time^3))

plot(co2, xlab = "Year", ylab = expression("CO"[2]*" Concentration (ppm)"), main = "Figure 11.
     type = "l")
lines(co2 + linear_model$resid, col = "blue", cex = 0.6)
lines(co2 + poly_model$resid, col = "red", cex = 0.6)
legend('bottomright',
       legend = c("Original", "Linear Model", "Polynomial Model"),
       col=c("black","blue","red"), lty=1)
```

Figure 11. Fitted Polynomial Model

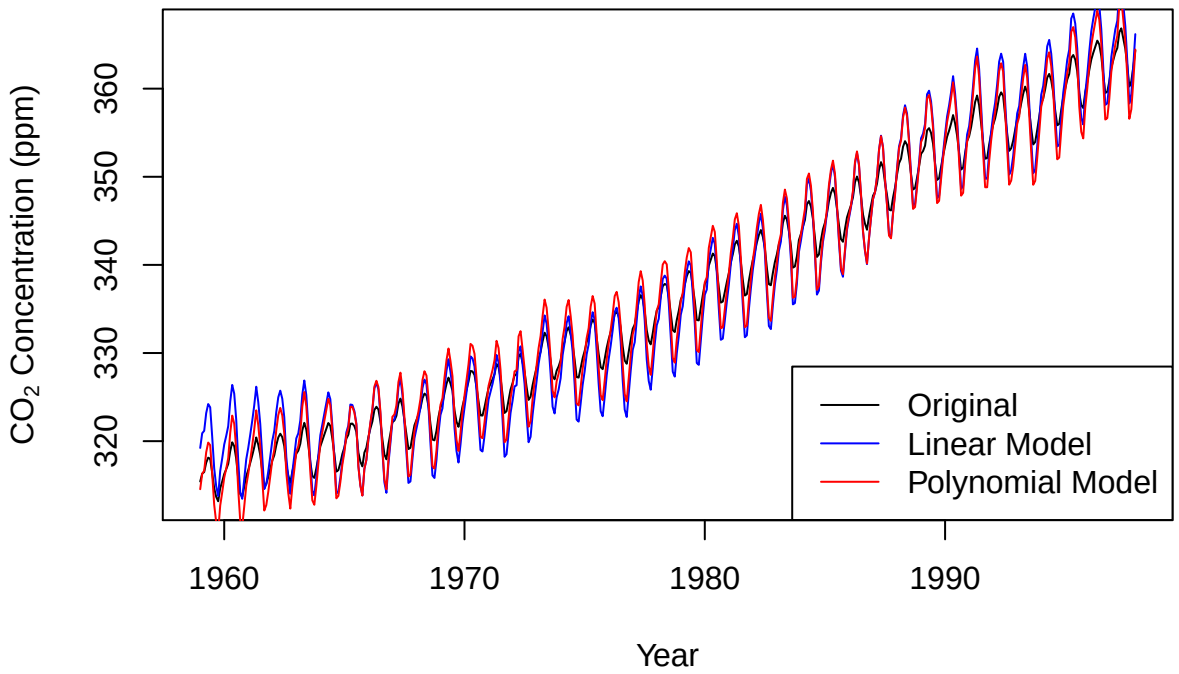
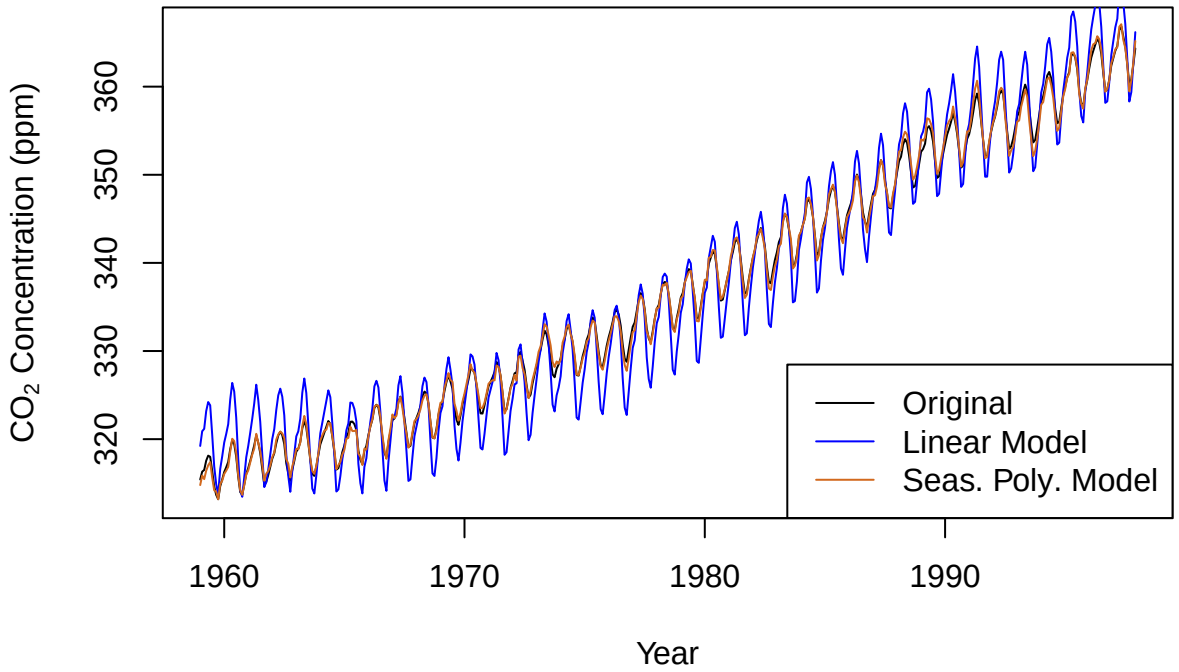


Figure 11 overlays the fitted time trend model of the third order with the original linear model of the first order. Except for a few repeated locations, the linear and polynomial models mostly overlapped one another closely. This indicates that using higher orders of time trend alone will not improve the model while the seasonality is still not addressed.

The use of seasonal dummy variables in the model can attempt to address this.

```
poly_dummy_model <- tslm(formula = co2 ~ 0 + trend + I(trend^2) +
                           I(trend^3) + season)

plot(co2, xlab = "Year", ylab = expression("CO"[2]*" Concentration (ppm)"),
     main = "Figure 12. Fitted Seasonal Polynomial Model", type = "l")
lines(co2 + linear_model$resid, col = "blue", cex = 0.6)
lines(co2 + poly_dummy_model$resid, col = "chocolate", cex = 0.6)
legend('bottomright',
       legend = c("Original", "Linear Model", "Seas. Poly. Model"),
       col=c("black","blue","chocolate"), lty=1)
```

Figure 12. Fitted Seasonal Polynomial Model



Demonstrated in **Figure 12**, the use of seasonal dummy variables for each month results in a polynomial time trend model with a dramatic improvement. This is both compared to the original linear model, as well as the previous polynomial model without any seasonal components. Now, instead of greatly exaggerating the annual variance, the cycles appear to overlap much more closely with the original time series.

The residuals of this model will be examined similar to what was done for the linear model.

```
par(mfrow = c(1, 2), cex = 0.6)
plot(acf(poly_dummy_model$residuals, plot = FALSE), main = "")
title("Figure 13. Autocorrelation Plot")
plot(pacf(poly_dummy_model$residuals, plot = FALSE), main = "")
title("Figure 14. Partial Autocorrelation Plot")
```

Figure 13. Autocorrelation Plot

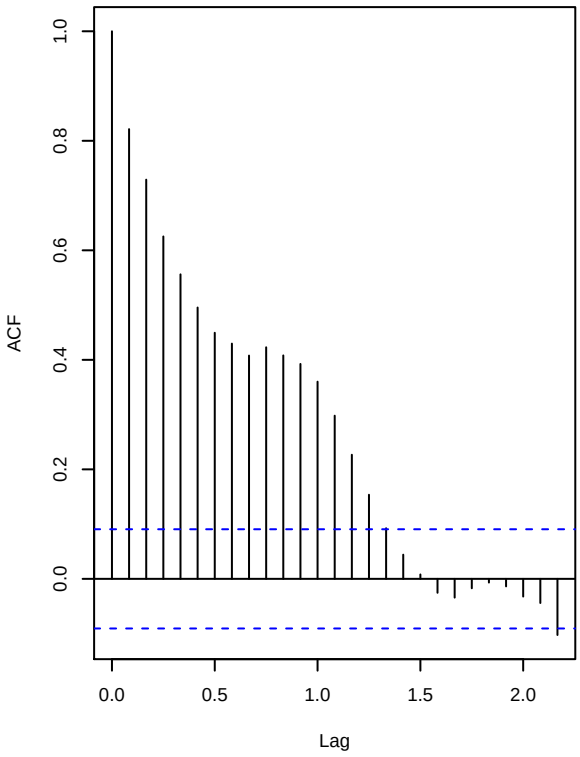
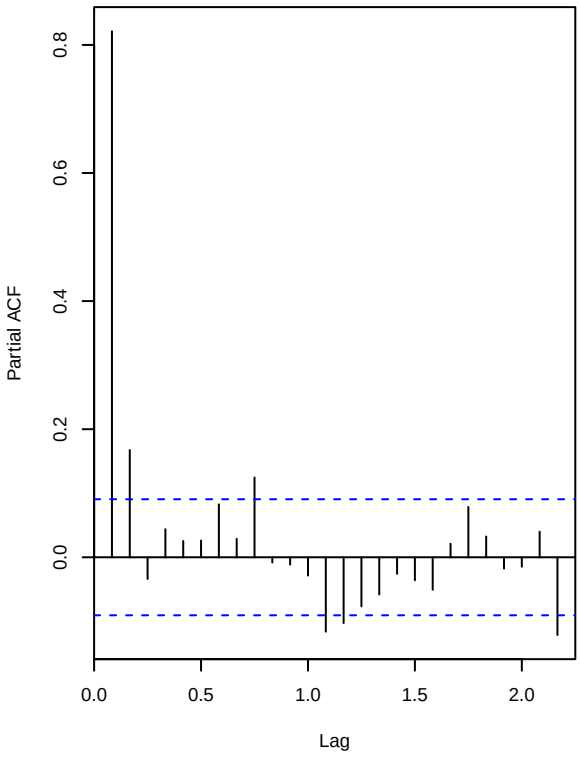


Figure 14. Partial Autocorrelation Plot



Figures 13 and **14** display the ACF and PACF of the residuals from the polynomial model with seasonal dummy variables. The cyclic components found with the linear model in **Figures 8** and **9** are no longer visible in these examples. **Figure 13** shows heavy autocorrelation until lag ~15, while **Figure 14** shows that the first lag still has a very significant partial autocorrelation associated with it.

The residuals appearance can also be directly examined as done previously.

```
plot(poly_dummy_model$residuals,
     xlab = "Time", ylab = expression("CO"[2]*" Concentration  Residual (ppm)"),
     xaxt = "n",
     main = "Figure 15. Seasonal Polynomial Model Residuals", type = "l")
axis(1, at=c(12,132,252,372), labels=c(1960, 1970, 1980, 1990))
abline(h = mean(poly_dummy_model$resid), col = "red")
```

Figure 15. Seasonal Polynomial Model Residuals

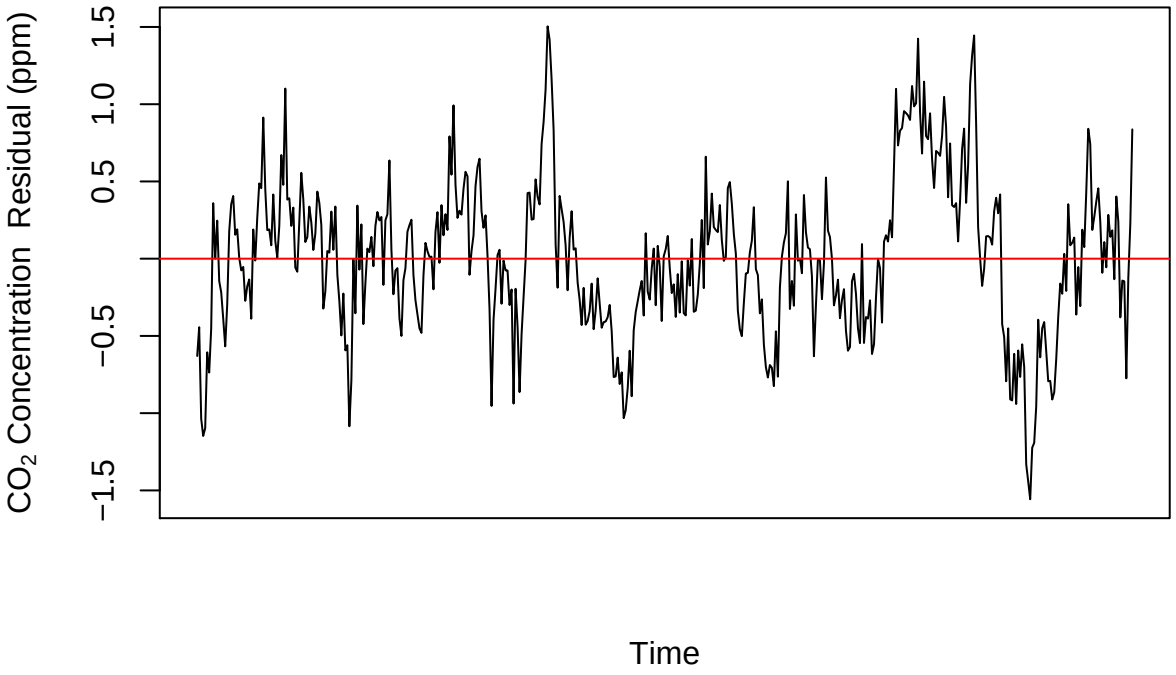


Figure 15 shows the residuals of the polynomial model with seasonal dummy variables. Though still fluctuating, the residuals now appear much closer to white noise compared to **Figure 10**. This is indicative of an improved fit. While there is still some cyclic pattern that can be observed, it is no longer clear nor consistent. These cycles still indicate that there may be components in the original time series not yet accounted for by the model, however.

Finally, this best performing polynomial model can also be used to forecast values to the present.

```
# make predictions 20 years from end of time series
predictions1 <- forecast(poly_dummy_model, h = 240)
pred.ts <- ts(predictions1$mean, start=c(1998,1), frequency=12)
ci.df <- data.frame(avg=predictions1$mean, upr=predictions1$upper[,2],
                    lwr=predictions1$lower[,2])

ggplot() +
  geom_line(data = co2, aes(x = time(co2), y = co2,
                            colour = "Original Values")) +
  geom_line(data=pred.ts, aes(x=time(pred.ts), y=pred.ts,
                              colour="Forecasted Values")) +
  geom_line(aes(x = 1959+(1:(39*12))/12,
                          y = co2 + poly_dummy_model$residuals,
                colour="Fitted Values")) +
  geom_ribbon(data=ci.df, aes(x=time(pred.ts),
                              ymin=lwr, ymax=upr), alpha=0.5) +
  labs(y = expression("CO"[2]*" Concentration (ppm)"),
       x = "Year", title = "Figure 16. Polynomial Model Forecasts",
       colour = "Time Series") +
```

```
theme(panel.background = element_blank(),
      axis.line = element_line(colour = "black"),
      legend.key=element_blank(),
      axis.text.x=element_text(colour="black"),
      axis.text.y=element_text(colour="black"))
```

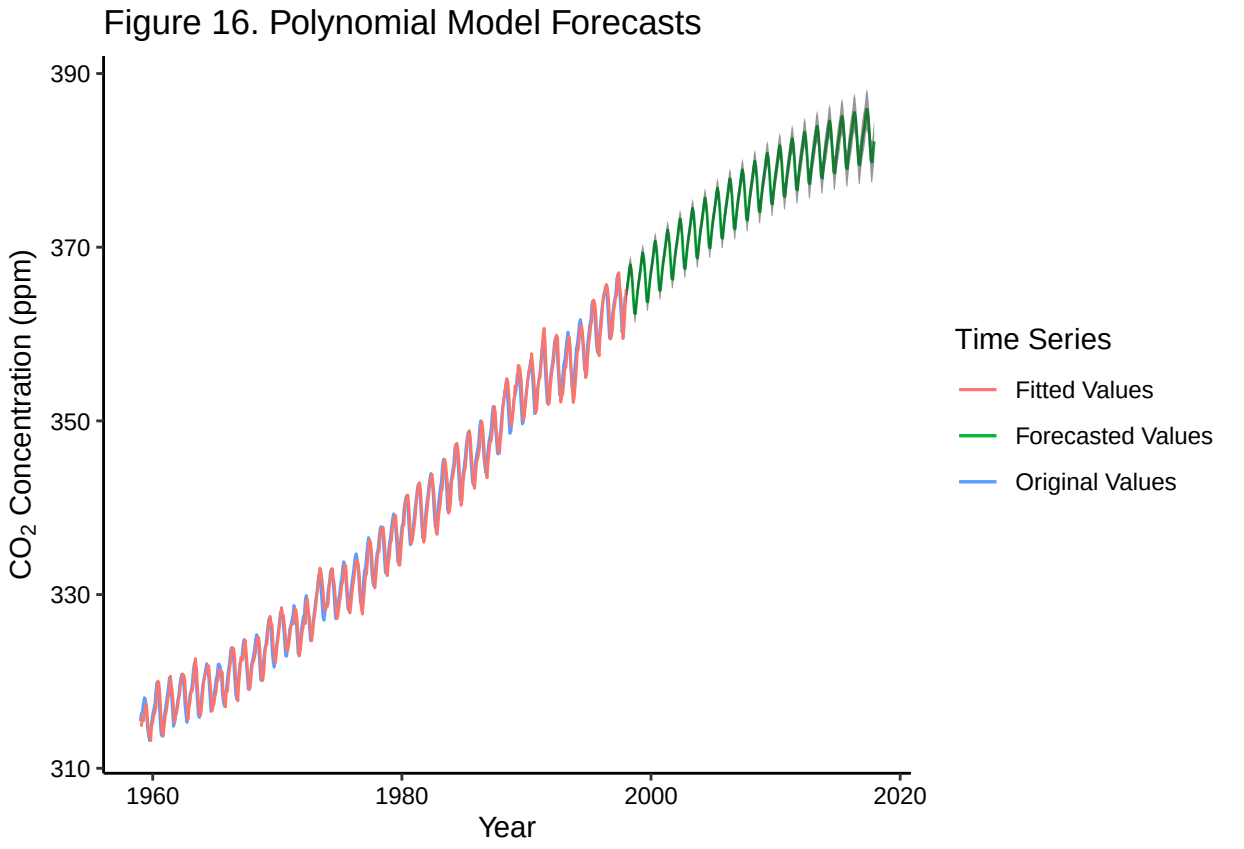


Figure 16 shows the original time series along with the polynomial trend and seasonal dummy variable fit. The forecast of the fitted model 20 years into the future is also shown. Finally, the confidence interval of the forecast is indicated by the transparent coloring around the values. It can be seen that these confidence intervals are quite tight to the forecast and do not project a large amount of uncertainty in the prediction. In general, uncertainty increases further into the future but not by an excessive amount.

In addition, the forecast generally predicts CO_2 concentrations to continue to increase, but at a decreasing rate. By the year 2020, it almost appears as if the forecast expects them to level off. However, knowing that CO_2 concentrations have continued to increase at about the same rate, this is inaccurate. Because of this, the model predicts a CO_2 concentration of about ~382.2 ppm by the year 2020 when in reality the measured value is about ~420 ppm. Later in the report the forecasts will be compared with a full set of recorded values during this time.

SARIMA Modeling with Keeling Data

While a polynomial time trend with seasonal effects fit the data well in **Figure 12**, other methods may better capture the relationships within the data. These include the use of autoregressive

(AR), moving average (MA), and/or integrated (I) terms. When used in tandem, this results in the ARIMA model that is one of the most common approaches in time series modeling. Without an I component, an ARMA model would take the form of:

$$x_t = \alpha_1 * x_{t-1} + \alpha_2 * x_{t-2} + ... + \alpha_p * x_{t-p} + \beta_1 * w_{t-1} + \beta_2 * w_{t-2} + ... + \beta_q * w_{t-q}$$

where x_t is the time series being modeled, p is the order of the AR process in the model, q is the order of the MA process in the model, and w_t is white noise. It can be seen that, given the name, the AR components use the previous values of the time series in order to forecast future values. Similarly, the MA components use the white noise residual of previous model forecasts to true values to adjust future forecasts. The use of the backwards shift operator $\boldsymbol{B}$, common in math which involves time, can more succinctly express the above formula as:

$$\theta_p(\boldsymbol{B})x_t = \phi_q(\boldsymbol{B})w_t$$

With a differencing of order d, the short form can be extended to

$$\theta_p(\boldsymbol{B})(1-\boldsymbol{B})^d x_t = \phi_q(\boldsymbol{B})w_t$$

Finally, if seasonality is included into the model, a set of identical but separate model terms may be included to capture the seasonal variation while the original set captures the non-seasonal variation. The seasonal components are indicated by the capital letter version of the same previous terms, resulting in a final seasonal ARIMA (SARIMA) model of the form

$$\Theta_P(\boldsymbol{B}^s)\theta_p(\boldsymbol{B})(1-\boldsymbol{B}^s)^D(1-\boldsymbol{B})^d x_t = \Phi_Q(\boldsymbol{B}^s)\phi_q(\boldsymbol{B})w_t$$

where the seasonal and non-seasonal terms need not be of the same order, e.g. $p = P$ or $p \neq P$.

Unlike the linear time trend, stationarity is needed for any SARIMA or ARIMA model. This will now be explored and compared to the previous linear time trend model.

As shown in the EDA, a second differenced time series is a better fit to the data than one with no differencing or only one difference. This would result in a value of $d = 2$ for the ARIMA model and potentially $D = 2$ for the seasonal component. Likewise, the ACF and PACF of the **Figures 5** and **6** can also be used to estimate the seasonality of the model. In both figures it can clearly be seen that there is an annual seasonal pattern, which should be included to estimate a SARIMA model. With this determined, the time series can be split into seasonal and non-seasonal components to investigate the potentially different AR/MA terms of each.

```
d_non_seas <- diff(d2, lag = 12)
d_seas <- d2 - d_non_seas

par(mfrow = c(2, 2), cex = 0.6)
plot(acf(d_non_seas, plot = FALSE), main = "")
title("Figure 17. Non-Seasonal ACF")
plot(pacf(d_non_seas, plot = FALSE), main = "")

title("Figure 18. Non-Seasonal PACF")
plot(acf(d_seas, plot = FALSE), main = "")
title("Figure 19. Seasonal ACF")
plot(pacf(d_seas, plot = FALSE), main = "")
title("Figure 20. Seasonal PACF")
```

Figure 17. Non−Seasonal ACF

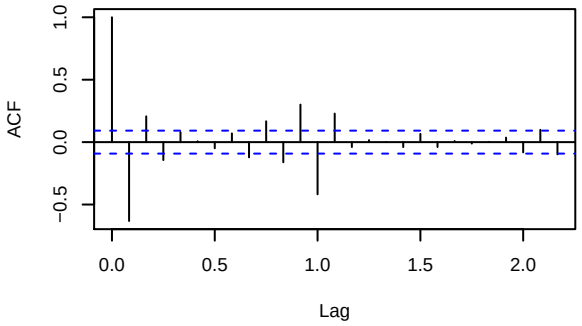


Figure 18. Non−Seasonal PACF

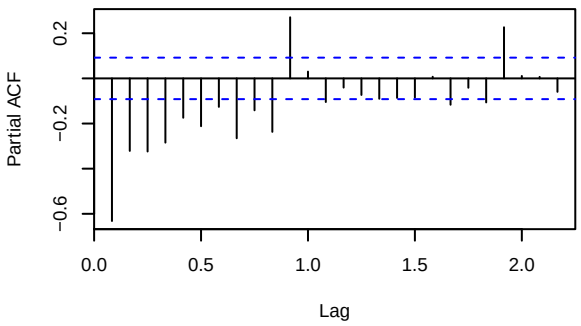


Figure 19. Seasonal ACF

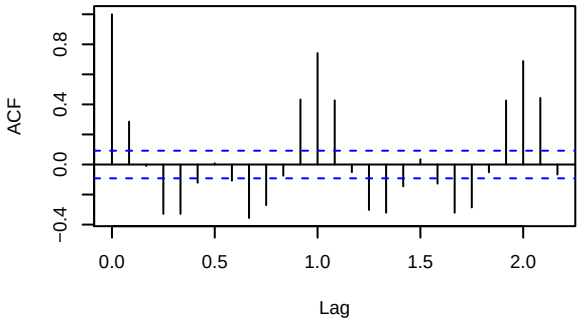
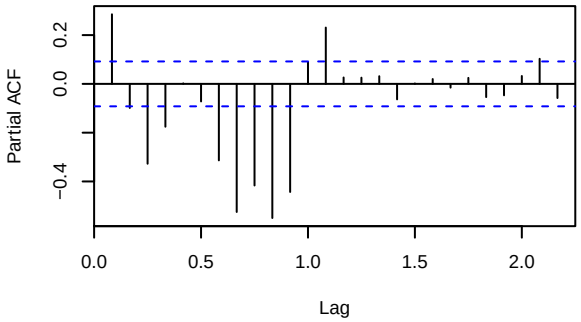


Figure 20. Seasonal PACF



Figures 17-18 show the ACF and PACF of the non-seasonal time series component while **Figures 19-20** show the same information for the seasonal component. It can be seen that the majority of the cyclic pattern is removed from the non-seasonal component once the lag difference was subtracted, but that some still remains with smaller magnitude. For the non-seasonal component, the single significant lag at *k=1* in both **Figures 17** and **18** suggests $p = q = 1$. There are more significant lags in **Figure 18**, but upon comparison with **Figure 20** it appears these could be related to seasonality which was imperfectly removed. For the seasonal component of the time series, a similar observation can be made of **Figure 19** and **20** that suggests $P = Q = 1$. Likewise, the significant lag at $k = 3$ in **Figure 20** suggests that P could be higher order, but this may still be a product of the seasonality that is imperfectly isolated.

Although these values may be what are found when manually examining the time series, the use of an information criterion, such as the Akaike Information Criterion (AIC), to differentiate one potential model from others can also be used. This will be investigated for SARIMA models which vary from the above specifications by one or two orders for each of the components. To simplify the search, and because the order of the terms appeared similar above, the seasonal and non-seasonal components will be assumed to be of the same order (i.e. $p = P$, $d = D$, and $q = Q$).

```
# placeholders
aic <- 0
arima_model <- NA

for (p in 1:3) {
  for (q in 0:1) {
    for (d_arima in 1:2) {
```

```
      model <- try(arima(d[], order = c(p,d_arima,q),
                         seasonal = list(order = c(p, d_arima, q), period = 12)),
                   silent = TRUE)
      if (length(model) > 1) {
        if (model$aic < aic) {
          d_model <- d_arima
          aic <- model$aic
          arima_model <- model
        }
      }
    }
  }
}

cat("Order of Differencing:", d_model, "\n")
```

```
## Order of Differencing: 1
```

```
print(arima_model$coef)
```

```
##         ar1         ar2         ma1        sar1        sar2        sma1
##  0.35919671  0.09848122 -0.71127156  0.02266567 -0.08062029 -0.90213416
```

Shown above, the SARIMA model with the best AIC had an order of $p = P = 2$, $d = D = 1$, and $q = Q = 1$. Notably, not performing a second differencing was found to fit better with the data. Having determined the order of each of the components, the final SARIMA model can be created on its own.

```
arima_model <- arima(d, order = c(2, 1, 1),
                     seasonal = list(order = c(2,1,1)))

plot(d, xlab = "Year",
     ylab = expression("CO"[2]*" Concentration (ppm), Transformed"),
     main = "Figure 21. Fitted SARIMA Model", type = "l")
lines(d + arima_model$resid, col = "red", cex = 0.6)
legend('bottomright',
       legend = c("Original", "SARIMA Model"),
       col=c("black","red"), lty=1)
```

Figure 21. Fitted SARIMA Model

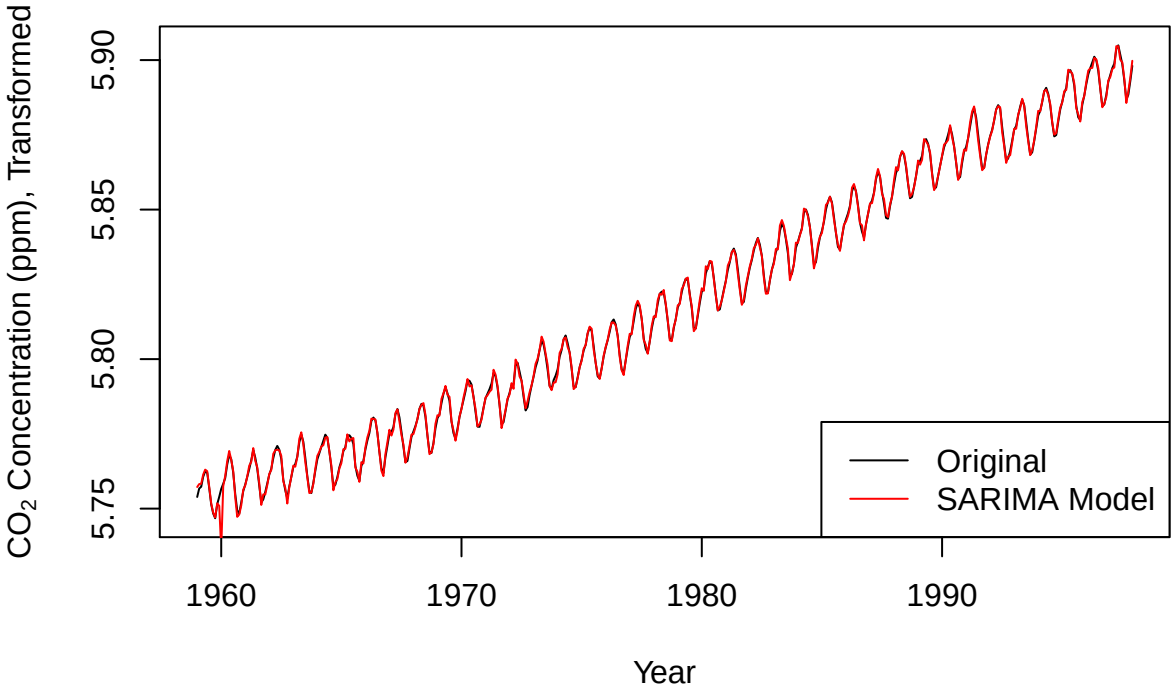


Figure 21 overlays the AIC-estimated SARIMA model with the original data. It can be seen that except for a small amount of jitter early in the time series, the model produces a very good fit.

The residuals may also be examined to investigate whether there is any remaining pattern not captured by the model.

```
plot(arima_model$residuals,
     xlab = "Time", ylab = "Model Residual", xaxt = "n",
     main = "Figure 22. SARIMA Model Residuals", type = "l")
axis(1, at=c(12,132,252,372), labels=c(1960, 1970, 1980, 1990))
abline(h = mean(arima_model$resid), col = "red")
```

Figure 22. SARIMA Model Residuals

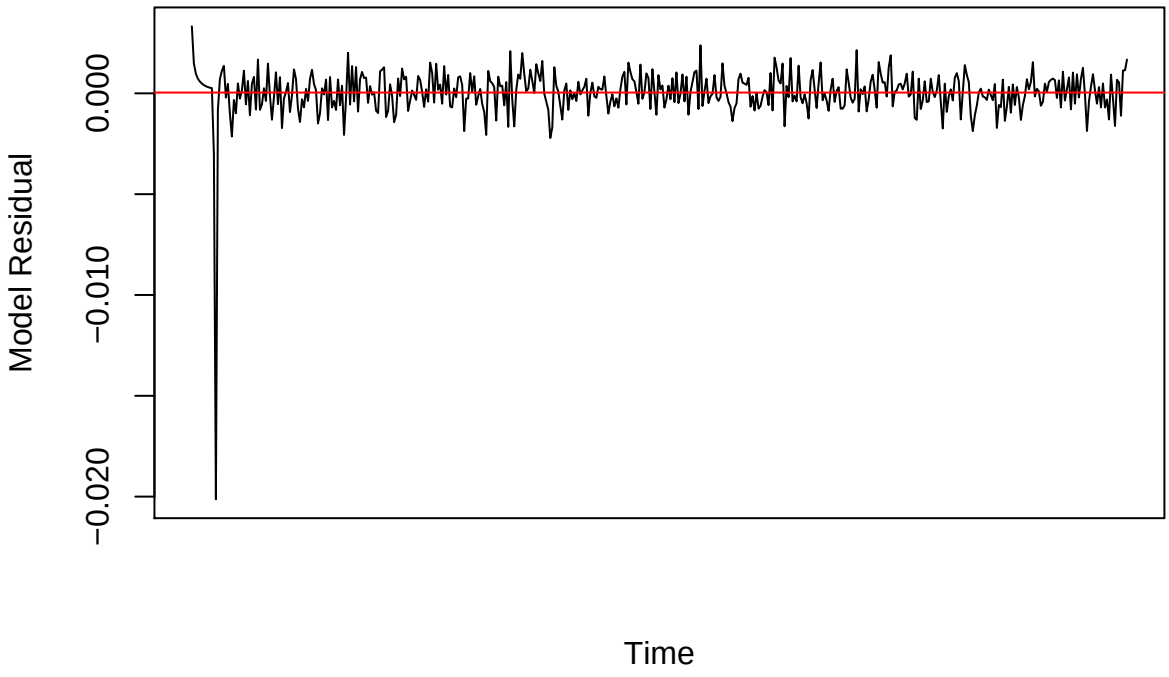


Figure 22 displays the residuals over time for the model, along with their mean value. There is a large fluctuation in the beginning of the model, corresponding to the error that is also seen in **Figure 21**. However, the rest of the series appears to resemble white noise indicating that the model specification is a good fit.

Finally, the model can be used to forecast values to 2017.

```
# make predictions 20 years from end of time series
predictions2 <- forecast(arima_model, h = 240)
pred2.ts <- ts(predictions2$mean, start=c(1998,1), frequency=12)
ci.df2 <- data.frame(avg=predictions2$mean, upr=predictions2$upper[,2],
                     lwr=predictions2$lower[,2])

ggplot() +
  geom_line(data=pred2.ts, aes(x=time(pred2.ts), y=pred2.ts,
                               colour="Forecasted Values")) +
  geom_line(aes(x = 1959+(1:(39*12))/12,
                          y = d + arima_model$residuals,
                colour="Fitted Values")) +
  geom_ribbon(data=ci.df2, aes(x=time(pred2.ts),
                               ymin=lwr, ymax=upr), alpha=0.5) +
  labs(y = expression("CO"[2]*" Concentration (ppm), Transformed"),
       x = "Year", title = "Figure 23. SARIMA Model Forecast",
       colour = "Time Series") +
  theme(panel.background = element_blank(),
        axis.line = element_line(colour = "black"),
        legend.key=element_blank(),
```

```
        axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black"))
```

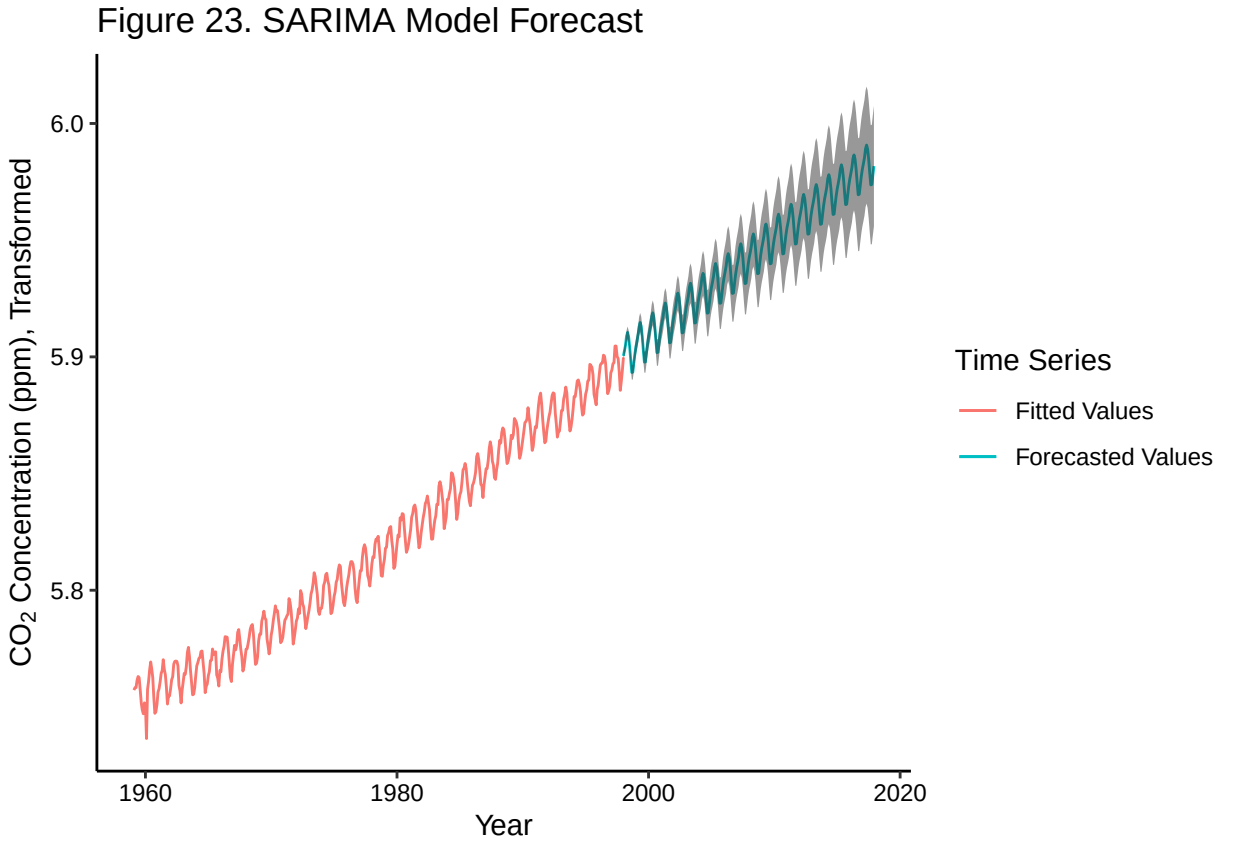


Figure 23 shows the forecast of the SARIMA model to the near present along with the 95% confidence intervals. As expected, the further into the future the model is used for forecasting, the wider the bounds become on the forecast. At the most recent point the model predicts a CO_2 concentration of about ~396 ppm, which is greater than the polynomial model and closer to the true recorded value of ~420 ppm. This is likely because the polynomial model of order 3 increases at a decreasing rate over time, whereas the SARIMA model continues closer to the observed rate in the time series. With this, the SARIMA model appears to be a better fit to the time series than any of the time trend models even though there is greater uncertainty in its predictions.

Forecast Evaluation with NOAA Data

In addition to Keeling's original observations, many other scientific groups across the world have studied the Earth's atmosphere. One such in the US is the National Oceanic and Atmospheric Administration (NOAA), whom continued Keeling's original observations at the same site. The file `co2_weekly_mlo.txt` contains weekly observations of atmospheric CO_2 concentrations measured at the same Mauna Loa Observatory from 1974 to 2020, published by NOAA. With this new dataset containing observations over the forecast period, at the same location as the original data, it can be used as a source of true values against which to compare the different forecasts.

```r
# Read table skipping all the irrelevant text
w <- read.table("co2_weekly_mlo.txt", skip=47)

# Rename columns
colnames(w) <- c("yr", "mon", "day", "decimal", "ppm",
                 "no.days", "1yr", "10yr", "since.1800")

# Merge
w <- w %>%
  mutate(date = make_date(yr, mon, day))
```

In addition to the measured concentrations of CO_2 in ppm, there is additional information provided by NOAA. For use as true values against which to compare a forecast, however, only the concentrations are needed.

It appears there are 18 missing weekly measurements of CO_2 concentrations, which need to be addressed. Potential options include dropping these dates or imputing the values with the surrounding data in time. This may be less desirable if there are serial missing values rather than isolated points which are missing.

```r
# First drop irrelevant columns
w2 <- w[c('date','ppm')]

subset(w2, w2$ppm==-999.99)
```

```
##            date     ppm
## 73   1975-10-05 -999.99
## 82   1975-12-07 -999.99
## 83   1975-12-14 -999.99
## 84   1975-12-21 -999.99
## 85   1975-12-28 -999.99
## 111  1976-06-27 -999.99
## 410  1982-03-21 -999.99
## 413  1982-04-11 -999.99
## 414  1982-04-18 -999.99
## 482  1983-08-07 -999.99
## 516  1984-04-01 -999.99
## 517  1984-04-08 -999.99
## 518  1984-04-15 -999.99
## 519  1984-04-22 -999.99
## 1640 2005-10-16 -999.99
## 1781 2008-06-29 -999.99
## 1782 2008-07-06 -999.99
## 1783 2008-07-13 -999.99
```

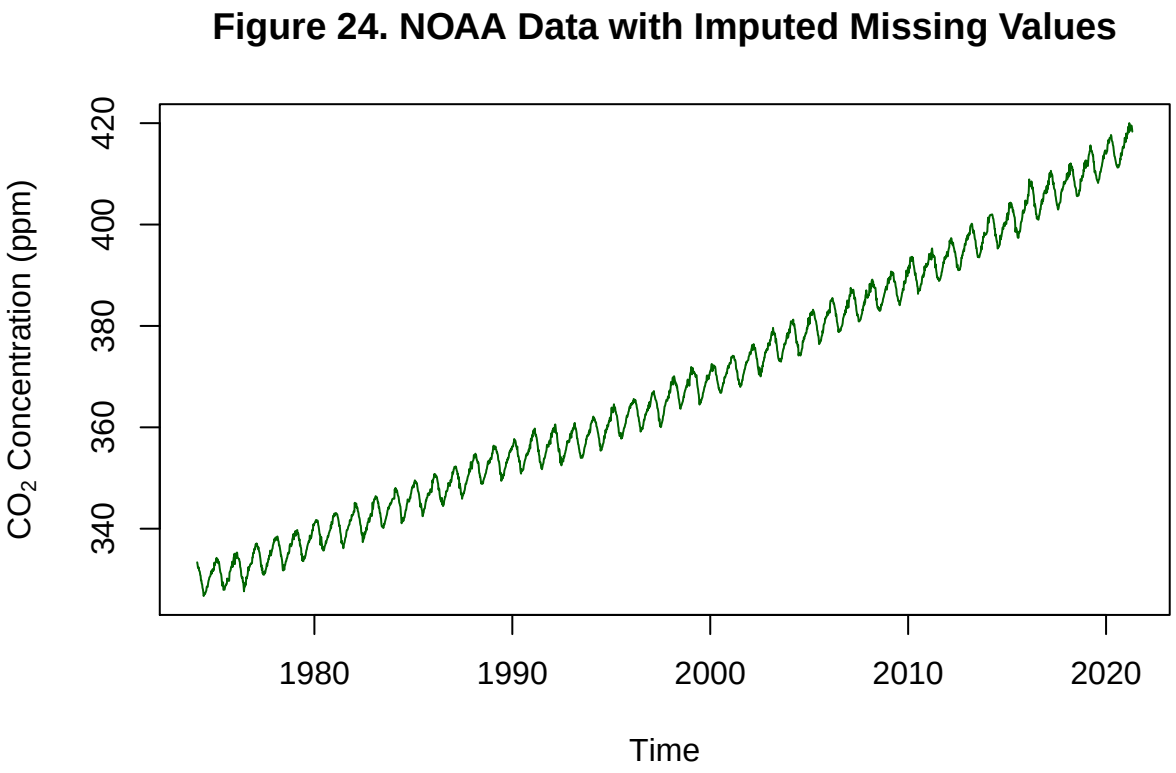
It appears that there are a mixture of time periods containing missing values. For example, in 1975 there are no measurements for the entire month of December, while in 1982 there are two

weeks worth of missing values for April. Similarly, in 1984 there are almost no data for April. For simplicity, the missing values may be replaced with the last available value. This will mean that for some consecutive missing points all values will be identical to the previously recorded point. For example, the four weeks in December 1975 will be identical to the last measurement from November 1975. Linear interpolation may also be used, but for simplicity this was not chosen.

```
# recursively replace consecutive series of missing values
for (i in 1:(length(w2[,2])-1)) {
   if (w2[,2][i+1]==-999.99) {w2[,2][i+1]<- w2[,2][i]}
 }
```

With the missing data filled in, it can then be visualized to compare with the Keeling Curve of **Figure 1**.

```
ts1 <- ts(w2$ppm, start=c(1974, 5), frequency=52)

plot(ts1, main="Figure 24. NOAA Data with Imputed Missing Values",
     ylab = expression("CO"[2]*" Concentration (ppm)"), col="darkgreen")
```



Based on **Figures 24** and **1**, the NOAA carbon dioxide measurements closely resemble that of the Keeling curve. In **Figure 24** particularly, the imputation has not produced any obvious or drastic deviations from the overall seasonality and increasing trend. Unlike the original Keeling series, it does not appear that the variance of the trend is increasing. This can be double checked by decomposing the time series into constituent parts.

```
decomp.ts1 <- decompose(ts1, type="additive")
plot(decomp.ts1$seasonal,
     main="Figure 25. Seasonal Component of the NOAA Data",
     ylab="Deviation")
```

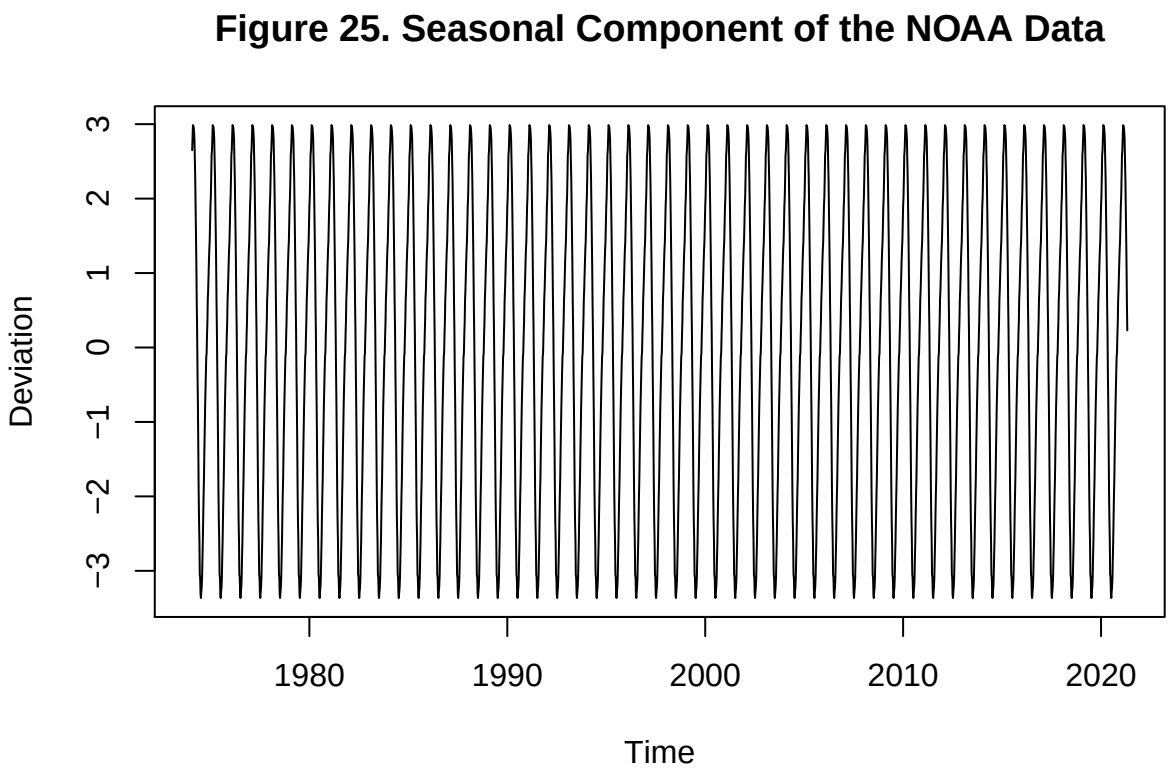


Figure 25 demonstrates that there is no variation in the seasonal component of the NOAA data, unlike that which was observed in Keeling's data set. This would make a logarithm transformation of this data unnecessary as it is already stationary in the variance. Next, the increasing mean must still be addressed, which can be done by differencing the values.

```
ts1.d1 <- diff(ts1)
ts1.d2 <- diff(ts1.d1)

par(mfrow = c(1, 2), cex = 0.6)
plot(ts1.d1, xlab = "Year", ylab = "First Differenced Value",
     main = "Figure 26. First Differenced Real Time Series", type = "l")
abline(h = mean(d1), lty = 2, col = "red")
plot(ts1.d2, xlab = "Year", ylab = "Second Differenced Value",
     main = "Figure 27. Second Differenced Real Time Series", type = "l")
abline(h = mean(d2), lty = 2, col = "red")
```

Figure 26. First Differenced Real Time Series

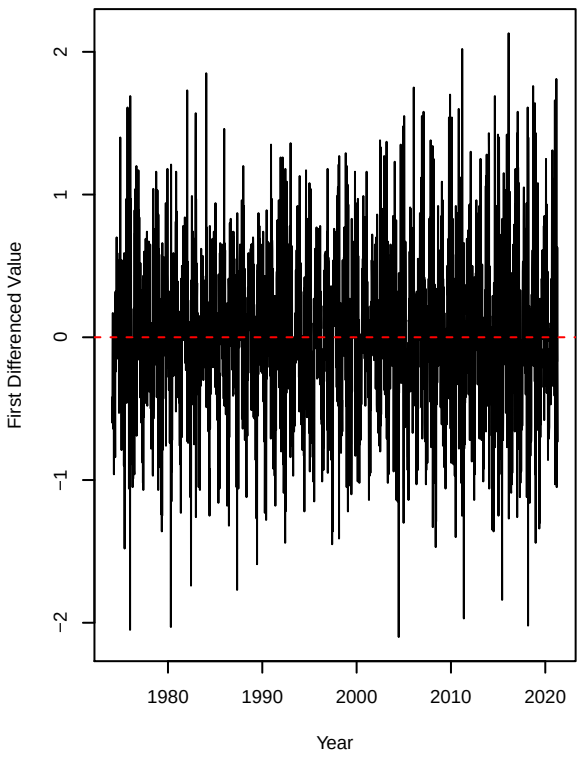
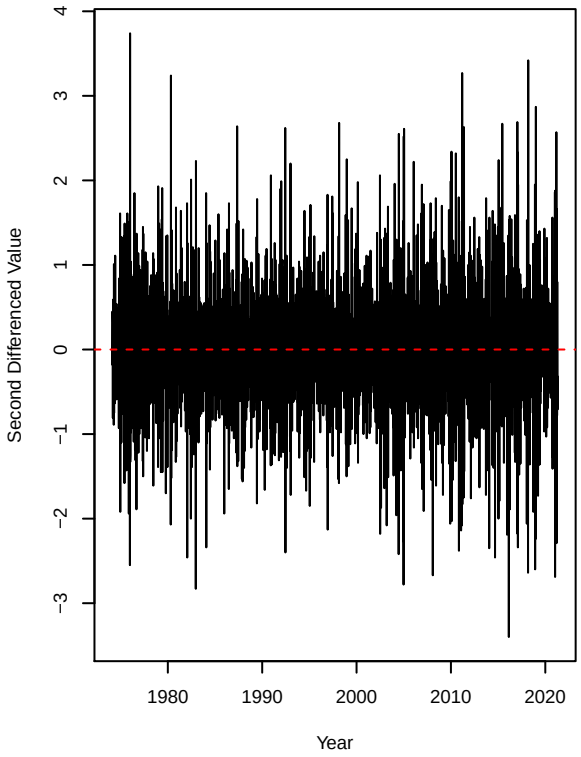


Figure 27. Second Differenced Real Time Series



Figures 26 and **27** display the time series with first and second order differencing respectively. Similar to Keeling's original series, it can be seen that though the differencing has removed the main increasing trend, there is still a degree of fluctuation about the mean. Compared to the original series, the increased granularity of data at the weekly level accentuates this fluctuation. Nevertheless, just like with Keeling's series, a second-order differencing operation not only removes the increasing trend but the data also becomes completely stationary around the mean. Next, a deeper look into the seasonality of the data can be taken using the ACF and PACF.

```
par(mfrow = c(1, 2), cex = 0.6)
plot(acf(ts1.d2, plot = FALSE), main = "")
title("Figure 28. Autocorrelation Plot")
plot(pacf(ts1.d2, plot = FALSE), main = "")
title("Figure 29. Partial Autocorrelation Plot")
```

Figure 28. Autocorrelation Plot

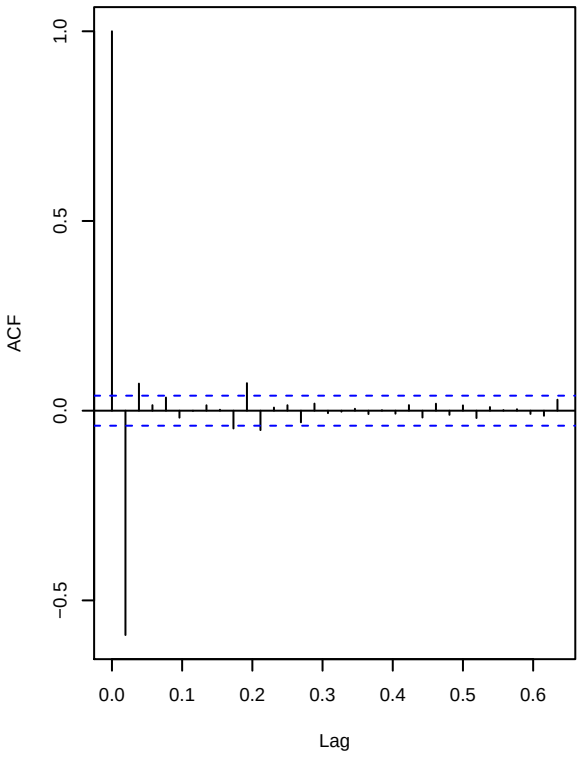
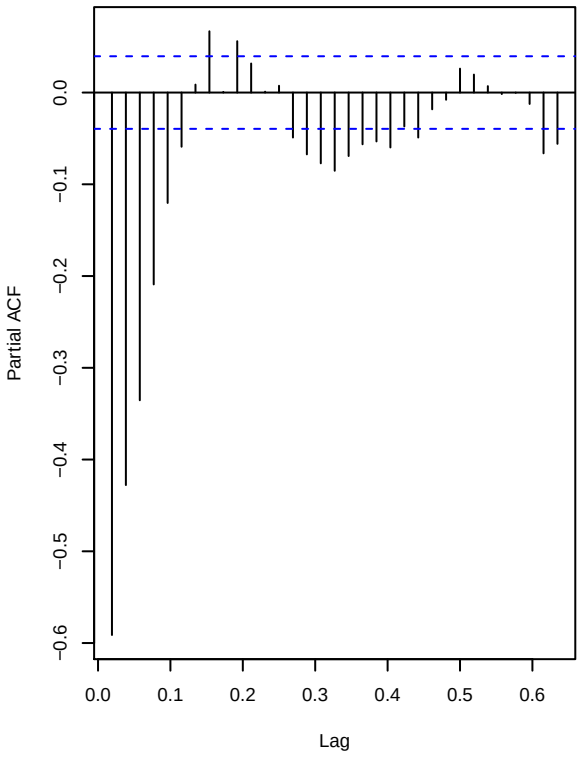


Figure 29. Partial Autocorrelation Plot

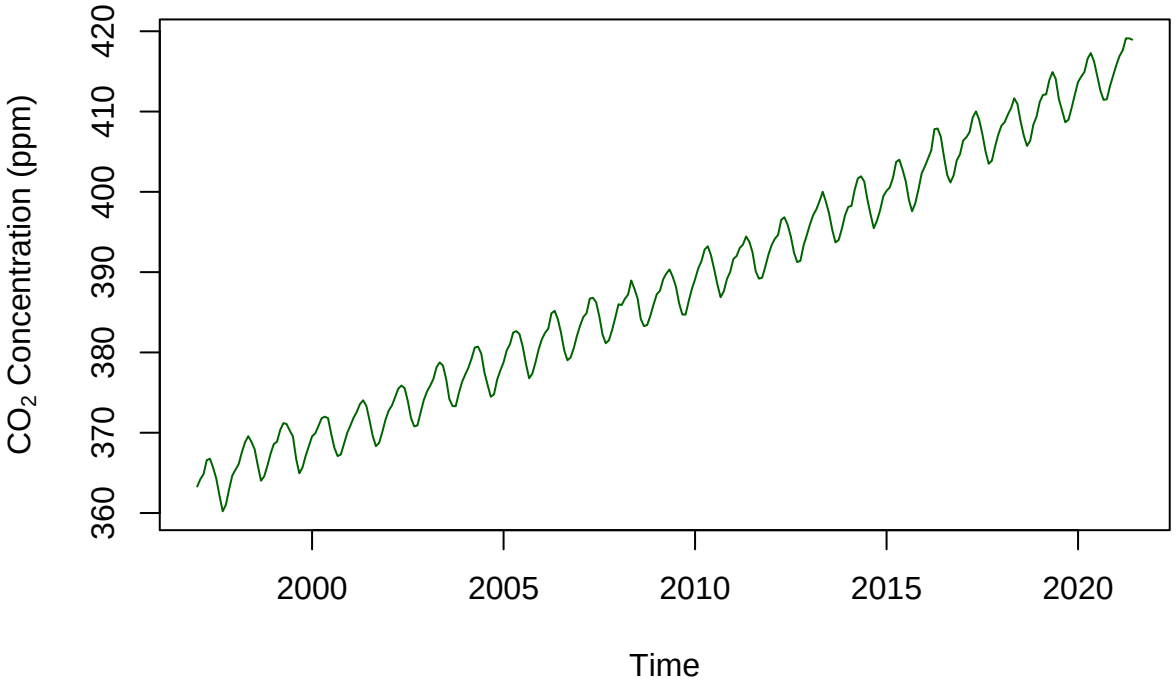


Figures 28 and **29** show the ACF and PACF of the NOAA data after second differencing. Unlike the ACF for Keeling's data, the ACF for this dataset shows a substantial and highly significant negative lag at $k = 1$. Thereafter the lags drastically decrease, and it appears only the third and eighth lags are weakly significant. Like the Keeling series, the PACF has an oscillatory pattern that appears to be decreasing in magnitude over time. These all indicate a seasonal element may be present with the data.

For a more direct comparison with Keeling's data and the previous forecasted values, aggregation can be performed to take the weekly (or approximately weekly) format into a monthly one.

```
fmt <- "%Y-%m-%d"
w3 <- aggregate(w2["ppm"], list(Date = as.yearmon(w2$date, fmt)), mean)
ts2 <- ts(subset(w3, w3$Date >= "Jan 1997")$ppm, start=c(1997,1), frequency=12)

plot(ts2, main="Figure 30. Aggregated Monthly NOAA Data From 1997",
     ylab = expression("CO"[2]*" Concentration (ppm)"), col="darkgreen")
```

Figure 30. Aggregated Monthly NOAA Data From 1997

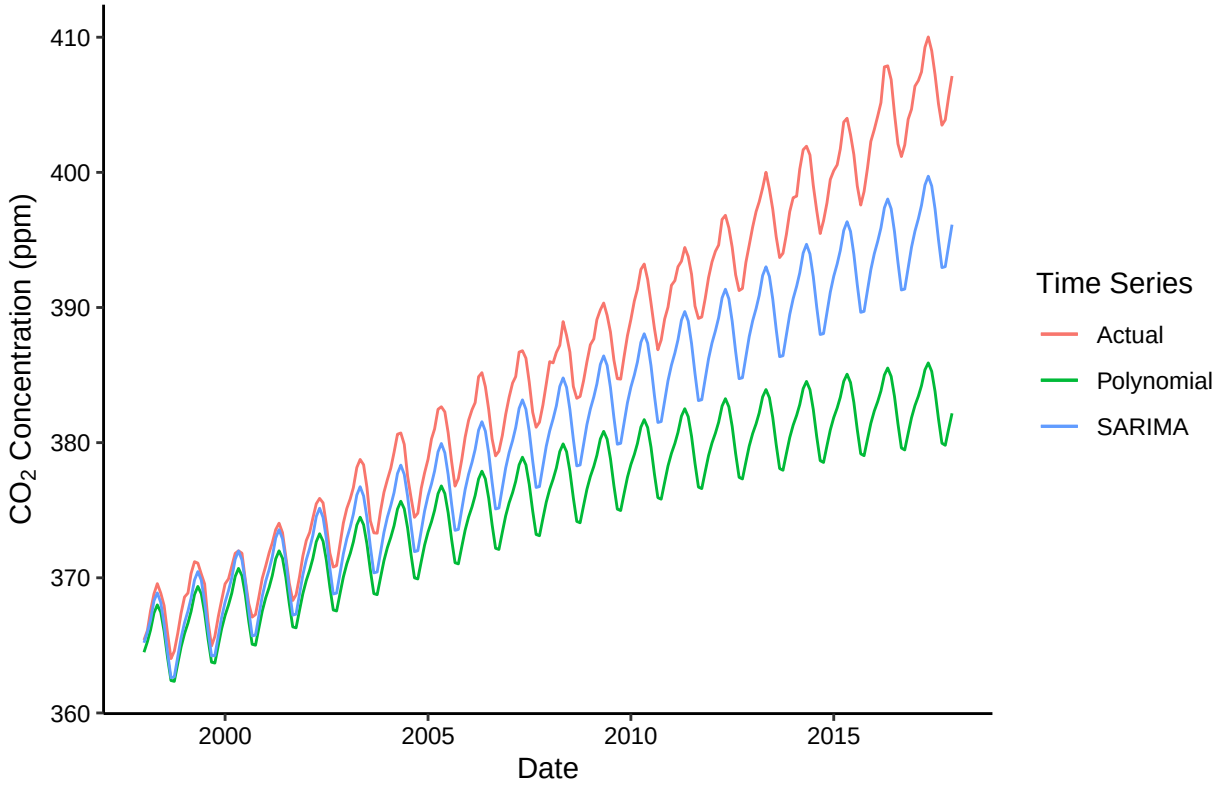


It can be seen from **Figure 30** that the monthly aggregated time series generally resembles the original weekly series shown in **Figure 24**. Thus, this aggregation method does not significantly alter the data but puts it at the same grain as the Keeling dataset.

In order to compare the previous forecasts with the NOAA data, they may be visualized together.

```
ts3 <- ts(subset(w3, w3$Date >= "Jan 1998" & w3$Date <= "Dec 2017")$ppm,
          start=c(1998,1), frequency=12)
p.ts <- ts(predictions1$mean, start=c(1998,1), frequency=12)
p.ts2 <- ts(exp(predictions2$mean), start=c(1998,1), frequency=12)
months <- as.Date(subset(w3$Date, w3$Date >= "Jan 1998" &
                           w3$Date <= "Dec 2017"))

ggplot() +
  geom_line(data=ts3, aes(x=months, y=ts3, color="Actual")) +
  geom_line(data=p.ts, aes(x=months, y=p.ts, color="Polynomial")) +
  geom_line(data=p.ts2, aes(x=months, y=p.ts2, color="SARIMA")) +
  ggtitle(expression("Figure 31. Forecasted vs NOAA CO"[2]*" Concentrations")) +
  labs(x="Date", y = expression("CO"[2]*" Concentration (ppm)"),
       colour = "Time Series") +
  theme(panel.background = element_blank(),
        axis.line = element_line(colour = "black"),
        legend.key=element_blank(),
        axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black"))
```

Figure 31. Forecasted vs NOAA CO_2 Concentrations



Shown in **Figure 31**, the SARIMA model generally predicts CO_2 levels very accurately, including seasonal variation, for the first six years until ~2003. Thereafter, the model begins to underestimate the true CO_2 levels, though it appears the cyclical variation remains well represented by the model. In comparison, though the polynomial can also provide a reasonable short-term forecast until ~2003, thereafter it begins to severely underestimate CO_2 levels even relative to the SARIMA model. In order to quantify the performance of each, metrics such as mean absolute error (MAE) and root mean squared error (RMSE) can be used, among others.

```
poly.acc <- accuracy(p.ts, ts3)
sarima.acc <- accuracy(p.ts2, ts3)
poly.acc
```

```
##                 ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 9.723005 11.84884 9.723005 2.471724 2.471724 0.9838427  8.809834
```

```
sarima.acc
```

```
##                 ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 4.477818 5.330831 4.477818 1.140596 1.140596 0.9743537  3.973992
```

Here, it can be seen that the SARIMA model quantitatively outperforms the polynomial model, with a MAE of about 4.48 compared to about 9.72 for the polynomial model and an RMSE of about 5.33 compared to about 11.85 for the polynomial model. Because of this and the visual

evidence in **Figure 31**, the SARIMA model approximates the recorded NOAA data more closely. Nevertheless, if forecasting continued further, the SARIMA and actual values will further bifurcate both with the visual gap as well as in the error metrics.

SARIMA Modeling with NOAA Data

Previously, both types of models generally underestimated the concentrations of CO_2 in forecasts over the 30 years from the 1990s to ~2020. However, these models were based on the Keeling dataset, with only ~40 years of total data to begin with. In this case, the 30 year forecasting period was over half the size of the original training period. Such long forecasts, especially when the length of the forecast is comparable to the training period length, are inadvisable and may not yield significant information.

In certain use cases, including atmospheric or planetary science, long term forecasting may still be important and preferable. However, for understanding and comparing model types, the short term forecasts can be useful when first deciding on which models or characteristics may be important for one which is longer term. Following this notion, a SARIMA model will be fit and evaluated only on a two-year forecast before a long term projection is created.

To begin, the NOAA data will be re-partitioned into the training and testing sets and evaluated for transformations.

```
train <- subset(w3, w3$Date <= "Jun 2019")
test <- subset(w3, w3$Date > "Jun 2019")

train.ts <- ts(train$ppm, start=c(1974, 5), frequency=12)
test.ts <- ts(test$ppm, start=c(2019, 6), frequency=12)
```

```
d_non_seas <- diff(train.ts, lag = 12)
d_seas <- train.ts - d_non_seas

par(mfrow = c(2, 2), cex = 0.6)
plot(acf(d_non_seas, plot = FALSE), main = "")
title("Figure 32. Non-Seasonal ACF")
plot(pacf(d_non_seas, plot = FALSE), main = "")
title("Figure 33. Non-Seasonal PACF")
plot(acf(d_seas, plot = FALSE), main = "")
title("Figure 34. Seasonal ACF")
plot(pacf(d_seas, plot = FALSE), main = "")
title("Figure 35. Seasonal PACF")
```

Figure 32. Non−Seasonal ACF

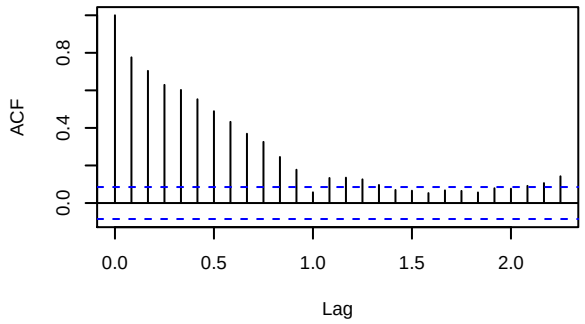


Figure 33. Non−Seasonal PACF

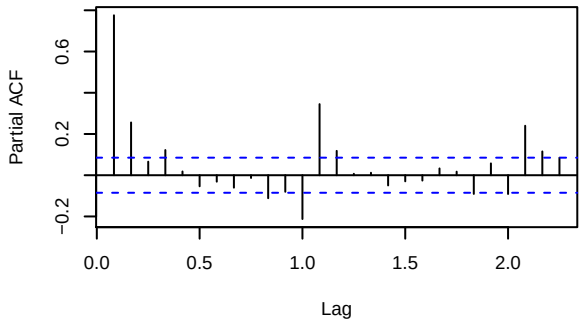


Figure 34. Seasonal ACF

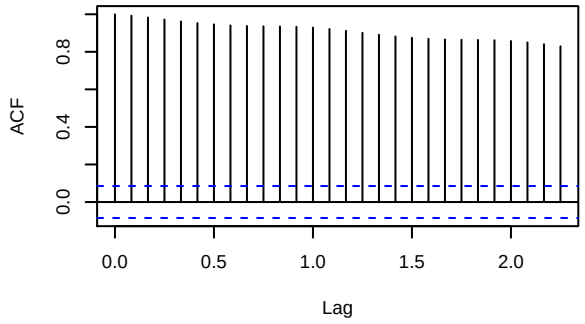
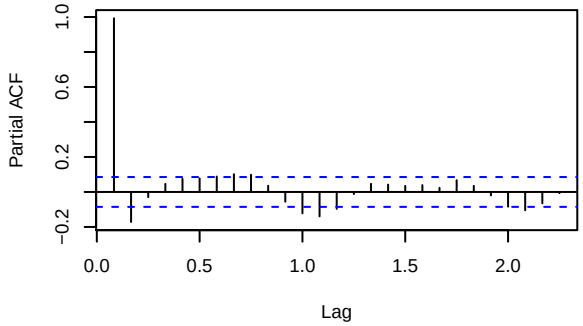


Figure 35. Seasonal PACF



Based upon the above non-seasonal (**Figures 32** and **33**) and seasonal (**Figures 34** and **35**) ACF and PACF plots, there is an element of seasonality in the data as the non-seasonal ACF (**Figure 32**) still shows a seasonal pattern. Similarly, the non-seasonal PACF (**Figure 33**) shows a significant spike at the 12th lag, with a slightly smaller (yet still significant) lag at the 24th. It can then be observed that after differencing and accounting for seasonality, the seasonal PACF (**Figure 35**) has a significant spike at the first, second, eleventh and twelfth lags. This indicates that an AR(1) or AR(2) model should be chosen, since only the first two immediate periods are significant before the seasonal pattern begins. Moreover in **Figure 34**, though every lag appears to be highly significant, the autocorrelations greater than the second lag might be caused by propagation of first lag's autocorrelation.

Based upon this, an autoregressive model of this data should be of at least the first order, with a differencing term of one as well. Nevertheless, due to the high number of significant lags in **Figure 34**, a moving average parameter may also be important to incorporate into the model. In order to find the best possible model, combinations of orders for p, d, q, and their seasonal equivalents can be created and compared with one another. Instead of making simultaneous forecasts, potentially causing an issue with the number of comparisons being made, they can be compared based on the AIC as done previously.

```
params <- list()
aics <- list()

for (p in 0:2) {
  for (q in 0:2) {
    for (d in 1:2) {
```

```
      for (P in 0:2) {
        for (Q in 0:2) {
          for (D in 1:2) {
            model <- try(arima(train.ts, order=c(p,q,d),
                               seasonal=list(order=c(P,D,Q),
                                             period=12), method="ML"),
                         silent=T)
            params <- append(params, paste(p,d,q,P,D,Q))
            default <- NULL
            aic <- try(default <- model$aic, silent=T)
            aics <- append(aics, aic)
              }
            }
          }
        }
      }
    }
```

With the different combinations of models created, the one which minimizes the AIC may be selected.

```
aics2 <- as.numeric(unlist(aics))
min <- ifelse(!all(is.na(aics2)), min(aics2, na.rm=T), NA)
index <- which(aics2==min)
```

Based on AIC, model 147 produced the best fitting model, with an AIC of about 342.3. The model had SARIMA parameters $p = 1$, $d = 1$, and $q = 1$, with seasonal parameters of $P = 0$, $D = 1$ and $Q = 1$, and an annual seasonality as used before. This model can be re-built outside of the prior loop for further use.

```
f.md <- arima(train.ts, order=c(1,1,1), seasonal=list(order=c(0,1,1), period=12))

plot(train.ts, xlab = "Year", ylab = expression("CO"[2]*" Concentration (ppm)"),
     main = "Figure 36. Fitted Best Performing SARIMA Model", type = "l")
lines(train.ts + f.md$resid, col = "orange", cex = 0.5)
legend('bottomright',
       legend = c("Original", "SARIMA Model"),
       col=c("black","orange"), lty=1)
```

Figure 36. Fitted Best Performing SARIMA Model

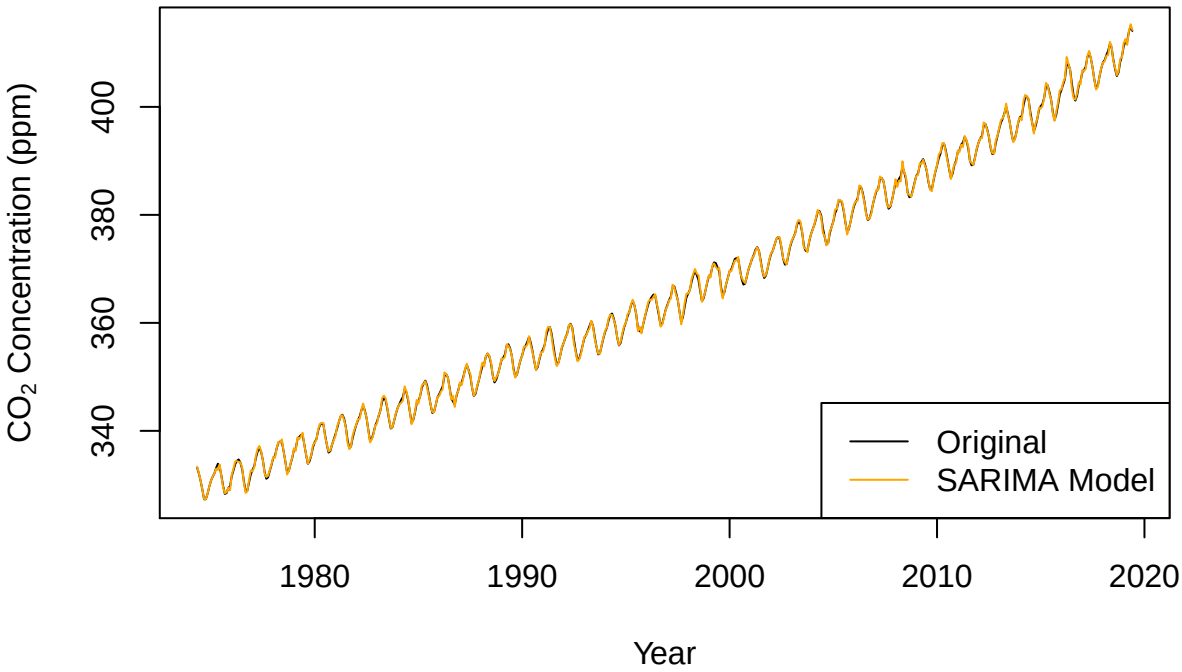


Figure 36 shows the best fit SARIMA model on the dataset, which matches the original data almost identically. This is similar to the fit of the original SARIMA model to Keeling's data in **Figure 21**.

The residuals of the model can also be examined.

```
par(mfrow = c(1, 2), cex = 0.6)
plot(f.md$residuals,
     xlab = "Time", ylab = "Residual Value", xaxt = "n",
     main = "Figure 37. Best Performing SARIMA Model Residuals", type = "l")
axis(1, at=c(12,132,252,372), labels=c(1960, 1970, 1980, 1990))
abline(h = mean(f.md$resid), col = "red")
plot(hist(f.md$residuals, plot = FALSE), xlab = "Residual Value",
     main="Figure 38. Histogram of Residuals")
```

Figure 37. Best Performing SARIMA Model Residual

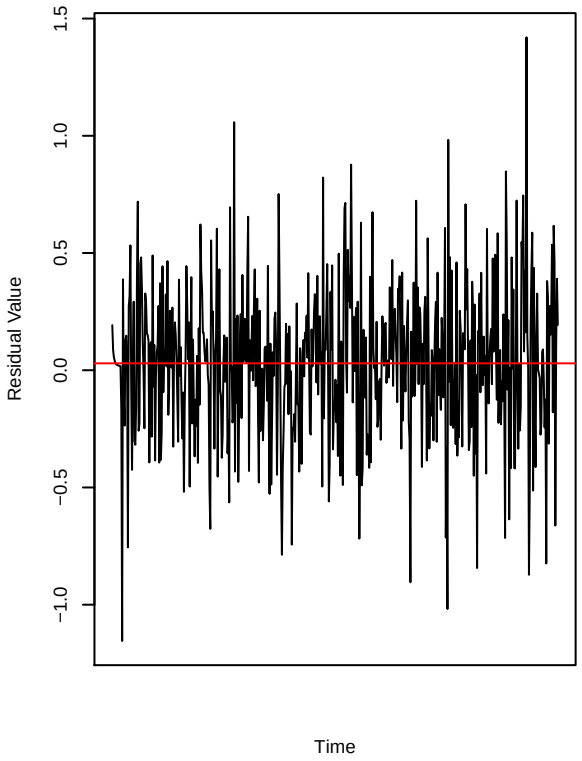
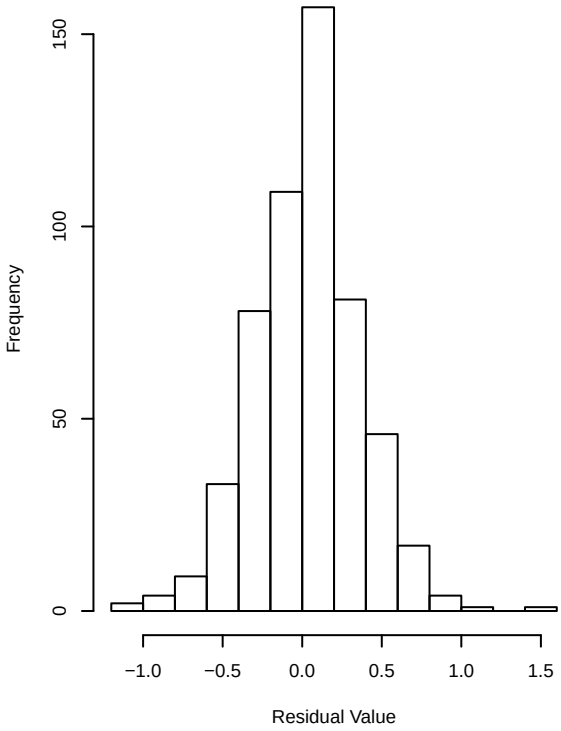


Figure 38. Histogram of Residuals



Figures 37 and **38** show the residuals of the model over time along with their distribution. It can be seen that there is no obvious pattern remaining in **Figure 37**, suggesting the model accounts for the majority of the components in the underlying data. Likewise the distribution appears normal, also suggesting there is no remaining trend in the data not accounted for by the applied transformations and modeling.

Following this visual inspection, the in-sample model performance metrics can be examined.

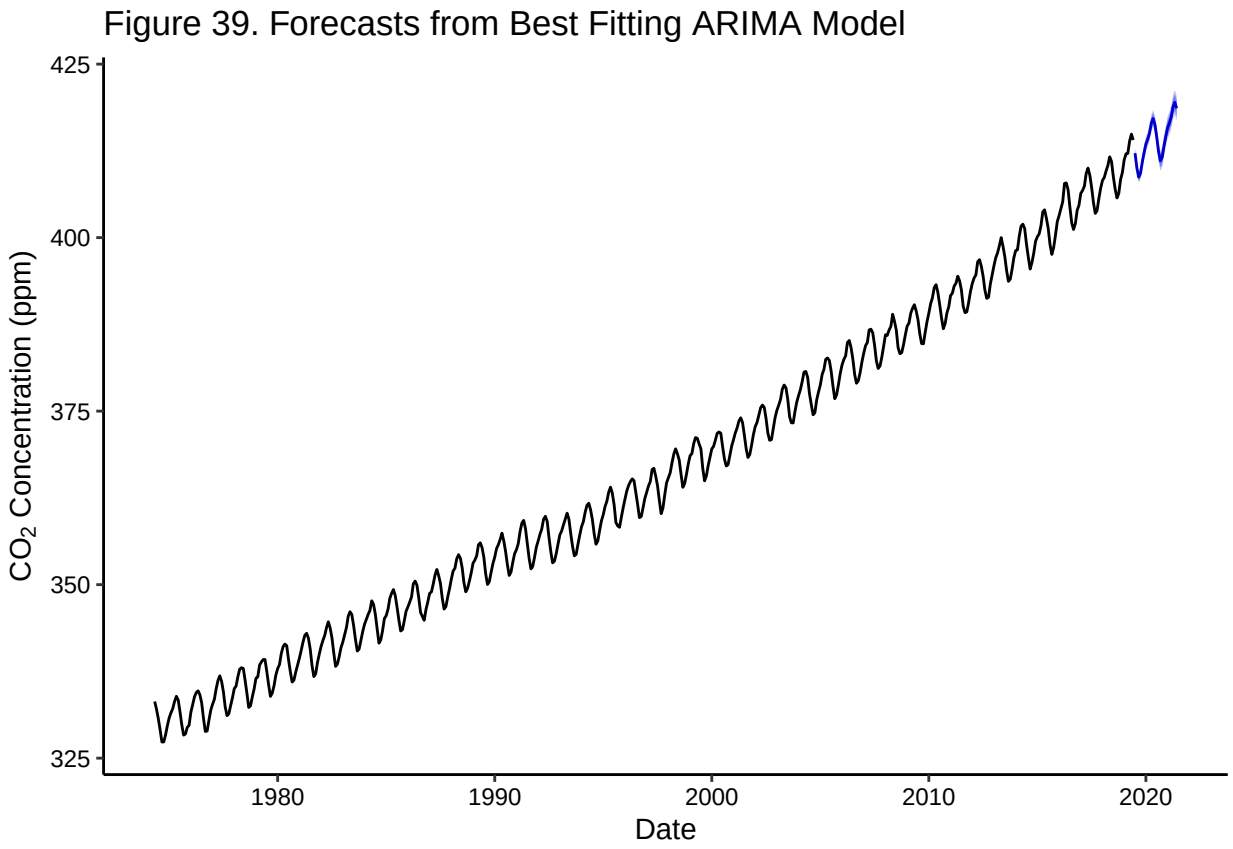
```
fitteds <- train.ts + f.md$residuals
isa <- accuracy(fitteds, train.ts)
isa
```

```
##                      ME    RMSE       MAE          MPE       MAPE        ACF1
## Test set -0.02942438 0.32666 0.2508338 -0.007799298 0.06834831 -0.01355779
##          Theil's U
## Test set 0.2543037
```

Based upon **Figure 36** the best performing model has a fit that is almost identical to the training set values. This is evidenced by the fact that it is not easy to visually differentiate the fitted values from the real values, and there are only small locations where there is not an exact overlap. In the performance metrics shown above, the in-sample performance achieves a MAE of about 0.2508 and a RMSE of about 0.3267, which reinforce the visual findings. While reassuring that the model has fit the training data, it is also important that this does not extend to the point of overfitting where any forecast would not be valuable. However, with only three terms in the model, this is not likely.

Next, this model may be used to forecast for the next two years, which corresponds to the testing set. These predictions and true values can be visualized together.

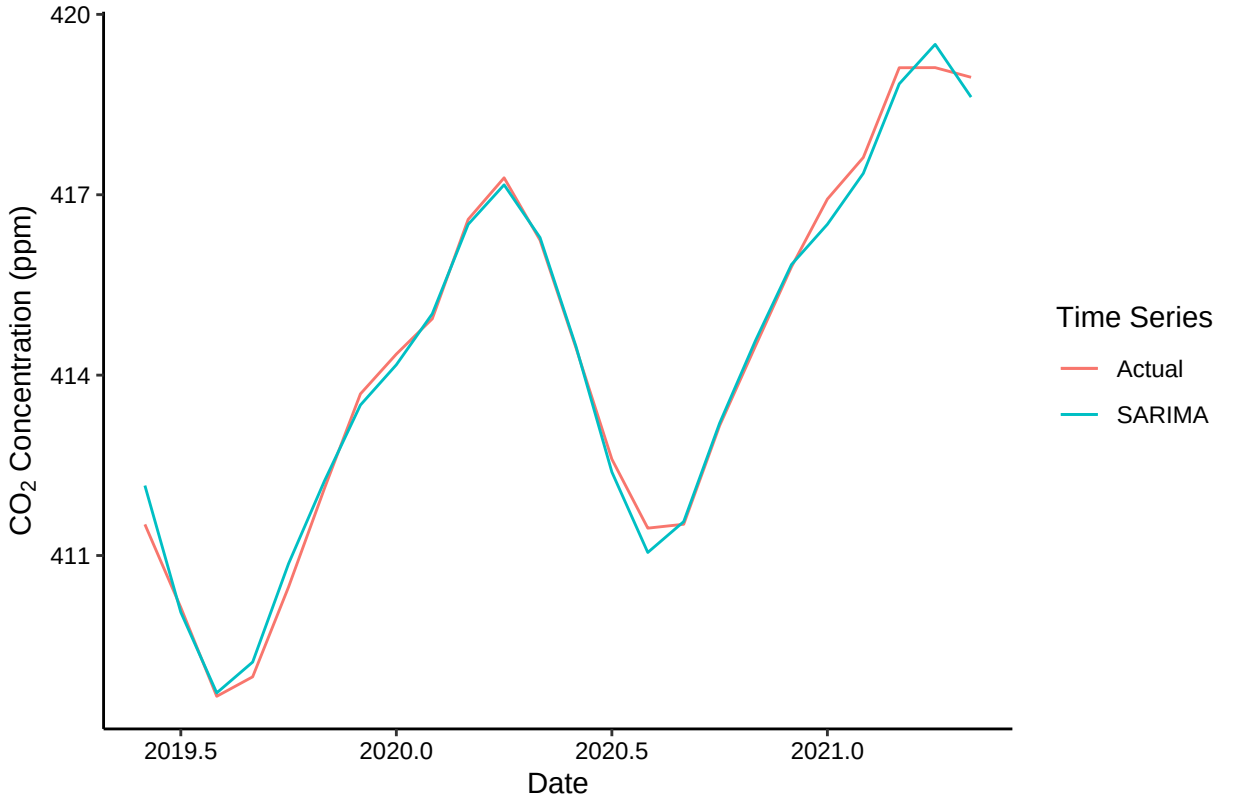
```
f1 <- forecast(f.md, h=24)
f.ts <- ts(f1$mean, start=c(2019,7), frequency=12)

autoplot(f1) +
  ggtitle("Figure 39. Forecasts from Best Fitting ARIMA Model") +
  labs(y=expression("CO"[2]*" Concentration (ppm)"), x="Date",
       colour = "Time Series") +
  theme(panel.background = element_blank(),
        axis.line = element_line(colour = "black"),
        legend.key=element_blank(),
        axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black"))
```




```
ggplot() +
  geom_line(data=test.ts, aes(x=time(test.ts), y=test.ts, color="Actual")) +
  geom_line(data=f.ts, aes(x=time(test.ts), y=f.ts, color="SARIMA")) +
  labs(x="Date", y=expression("CO"[2]*" Concentration (ppm)"),
       colour = "Time Series") +
  ggtitle("Figure 40. Comparison of Predicted Vs. Actual Values") +
  theme(panel.background = element_blank(),
```

```
        axis.line = element_line(colour = "black"),
        legend.key=element_blank(),
        axis.text.x=element_text(colour="black"),
        axis.text.y=element_text(colour="black"))
```

Figure 40. Comparison of Predicted Vs. Actual Values



In **Figure 39** the two year prediction of the SARIMA (1,1,1)(0,1,1)[12] model can be seen as a continuation of the previous trend. Moreover, the confidence intervals are extremely narrow to the point of being almost indistinguishable from the mean values. This might be due to over-fitting of the model during training which incorrectly improves its confidence, or the fact that it is a relatively short-term forecast. However, upon closer inspection in **Figure 40**, the CO_2 forecasts almost completely overlap the true values. This implies that the model and its transformations were well specified for the forecast to so closely match. Specifically, for June 2020, the short term forecast of the model is about ~416 ppm, matching the value NOAA later recorded to the nearest whole number.

With the forecast made, the out of sample error metrics may be calculated.

```
accf1 <- accuracy(f.ts, test.ts)
accf1
```

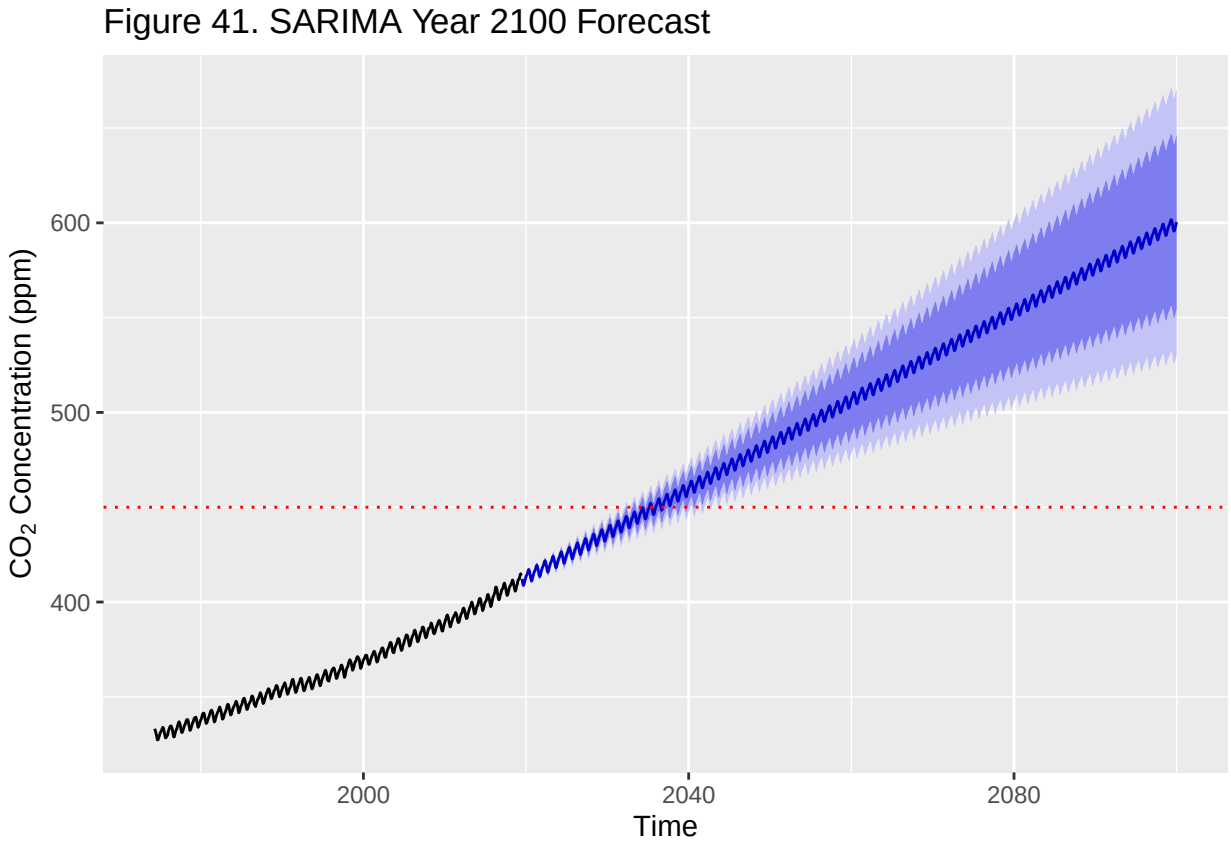
```
##                    ME     RMSE      MAE         MPE      MAPE      ACF1 Theil's U
## Test set 0.3233109 1.253458 1.152814 0.07729329 0.2784139 0.6536864 0.9856333
```

Now, on the testing dataset, the MAE is about 1.1528 while the RMSE is about 1.2535. These values are very low given the range of the outcome variable in the hundreds of ppm. Together with

the visual inspection conducted previously, it may be concluded that the SARIMA model fits these CO_2 concentrations very well. While this is a positive initial result, it should be remembered that this is a relatively short-term forecast at 24 months, with one value predicted per month. Longer term forecasts should still be taken with a grain of salt even though the results show promise on the shorter term, as any errors will accumulate and lead to greater divergence between the true and predicted values.

While any long term predictions should be interpreted with less certainty, sometimes they are more important or interesting to investigate than the short term. For example, some scientists have noted that the most recent time in Earth's history where CO_2 concentrations were this high is estimated to be the Pliocene era[1], raising questions for how this atmospheric change could lead to other changes among the Earth's processes.

```
f2 <- forecast(f.md, h=967)
f2df <- data.frame(f2)

autoplot(f2) +
  geom_hline(yintercept = 450, linetype="dotted", color='red') +
  ggtitle("Figure 41. SARIMA Year 2100 Forecast") +
  labs(x="Time", y=expression("CO"[2]*" Concentration (ppm)"))
```



According to the final model and depicted by the dotted red line in **Figure 41**, the CO_2 concentration is expected to first potentially reach or exceed 450 ppm in March 2035. The 95% confidence interval for this month is between 440.8005 and 459.3223 ppm. This level is notable as it is about the maximum allowable CO_2 concentration in the RCP2.6 modeling scenario[2], which is considered

the mildest put forth by researchers. In this scenario, peak CO_2 concentrations in the atmosphere should occur between ~2040-2050, which would result in an increase of an estimated 2°C in global mean temperature. With the model forecast in this project putting this concentration passed in 2035, it is likely that the Earth will experience a more greatly changing climate than this scenario describes.

Figure 41 also shows a ~1000 month forecast from June 2019, approximately until the year 2100. The trend the model predicts is approximately linear at such a long timescale. Nevertheless, the uncertainty in the predictions also increases the further a prediction is made into the future, shown by the widening 95% confidence intervals. When considering this, 450 ppm is first reached by the upper 95% confidence interval in approximately ~2032 and could be as late as ~2040. Thus, while it may be predicted that in January 2100 the point estimate of CO_2 levels will be about 600.4, the 95% confidence interval is extremely wide and between about 530.1 and 670.7 ppm. Consequently this indicates that the model should not be used to make predictions so far ahead into the future with certainty, and only used as a general estimate of where concentrations are headed.

Citations

[1] https://climate.nasa.gov/climate_resources/7/graphic-carbon-dioxide-hits-new-high/

[2] van Vuuren, D.P., Stehfest, E., den Elzen, M.G.J. et al. RCP2.6: exploring the possibility to keep global mean temperature increase below 2°C. Climatic Change 109, 95 (2011). https://doi.org/10.1007/s10584-011-0152-3