STA137 Take Home Final

Chad Pickering A02

March 17, 2016

The following dataset consists of monthly CO_2 levels at Alert, Northwest Territories, Canada. The data was collected from January 1994 through December 2004, and is measured in parts per million.

The task is to forecast the CO_2 levels from January through December 2005. We will ultimately give point and interval forecasts along with the relevant plots showing the trend and seasonality. This report has a marked results and conclusion section at the end for convenience.

First, load in the relevant R packages and the "co2" dataset. Please ignore the unavoidable R output in the following one and a half pages resulting from loading in needed libraries.

```
library(astsa)
library(tseries)
library(forecast)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: timeDate
## This is forecast 6.2
##
##
## Attaching package: 'forecast'
##
## The following object is masked from 'package:astsa':
##
##     gas
```

```
require(TSA)
```

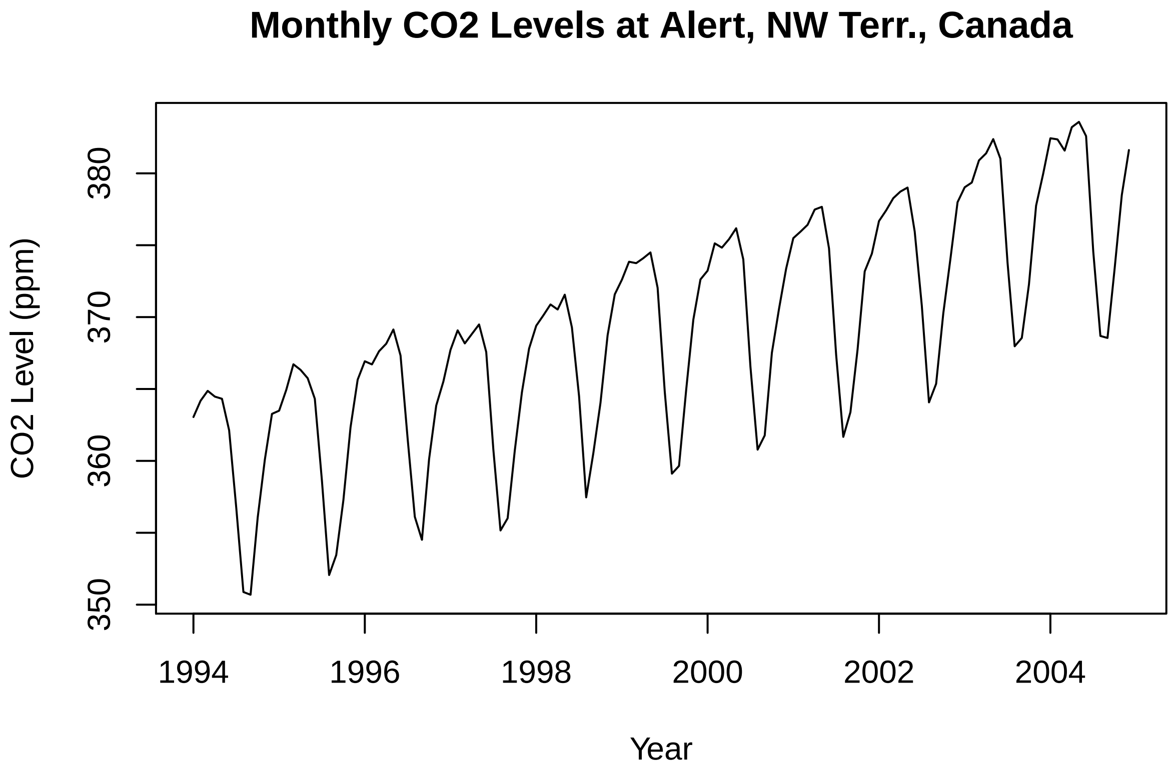
```
## Loading required package: TSA
## Loading required package: leaps
## Loading required package: locfit
## locfit 1.5-9.1    2013-03-22
## Loading required package: mgcv
## Loading required package: nlme
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:forecast':
##
##     getResponse
##
```

```
## This is mgcv 1.8-3. For overview type 'help("mgcv-package")'.
##
## Attaching package: 'TSA'
##
## The following objects are masked from 'package:forecast':
##
##     fitted.Arima, plot.Arima
##
## The following objects are masked from 'package:timeDate':
##
##     kurtosis, skewness
##
## The following objects are masked from 'package:stats':
##
##     acf, arima
##
## The following object is masked from 'package:utils':
##
##     tar
```

```
data(co2)

x <- as.vector(co2)
n <- length(x)
t <- 1:n
```

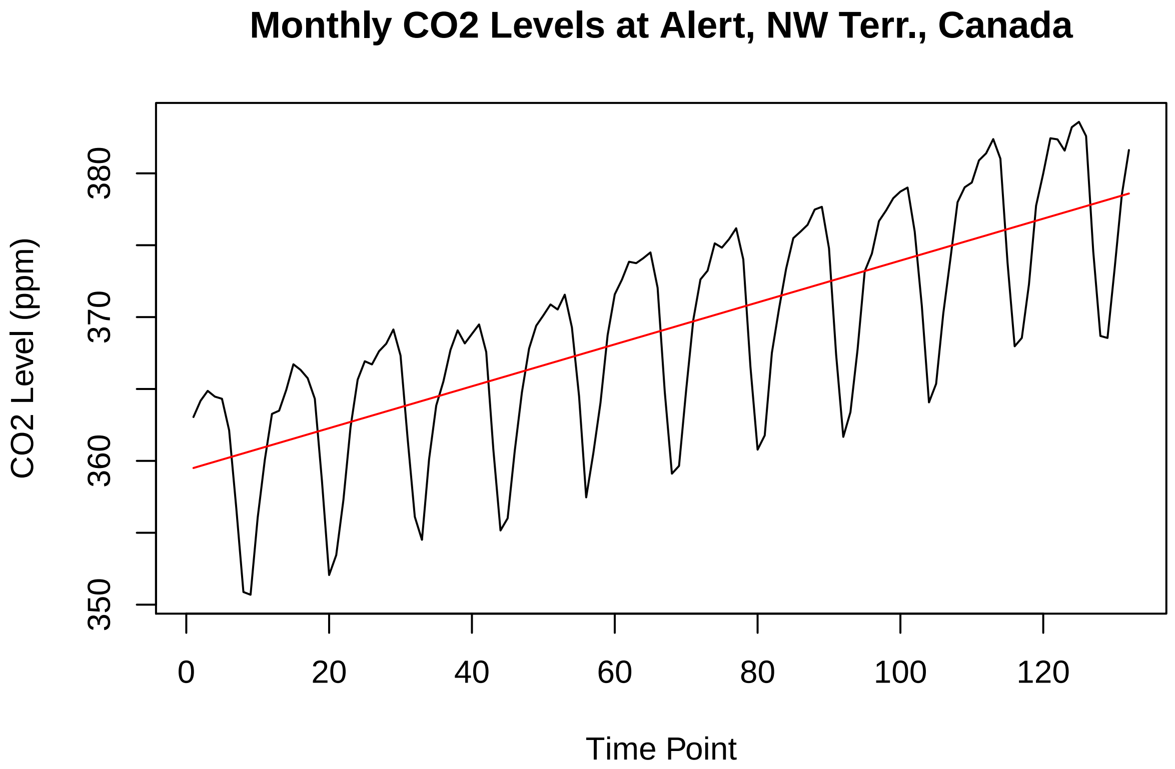
The following is the raw data of the CO_2 levels. We can see that there is a slow linear upwards trend and very regular seasonality evident. CO_2 levels seem to peak in the winter months and trough in the summer months.

```
ts.plot(co2, main = "Monthly CO2 Levels at Alert, NW Terr., Canada",
        xlab = "Year", ylab = "CO2 Level (ppm)")
```

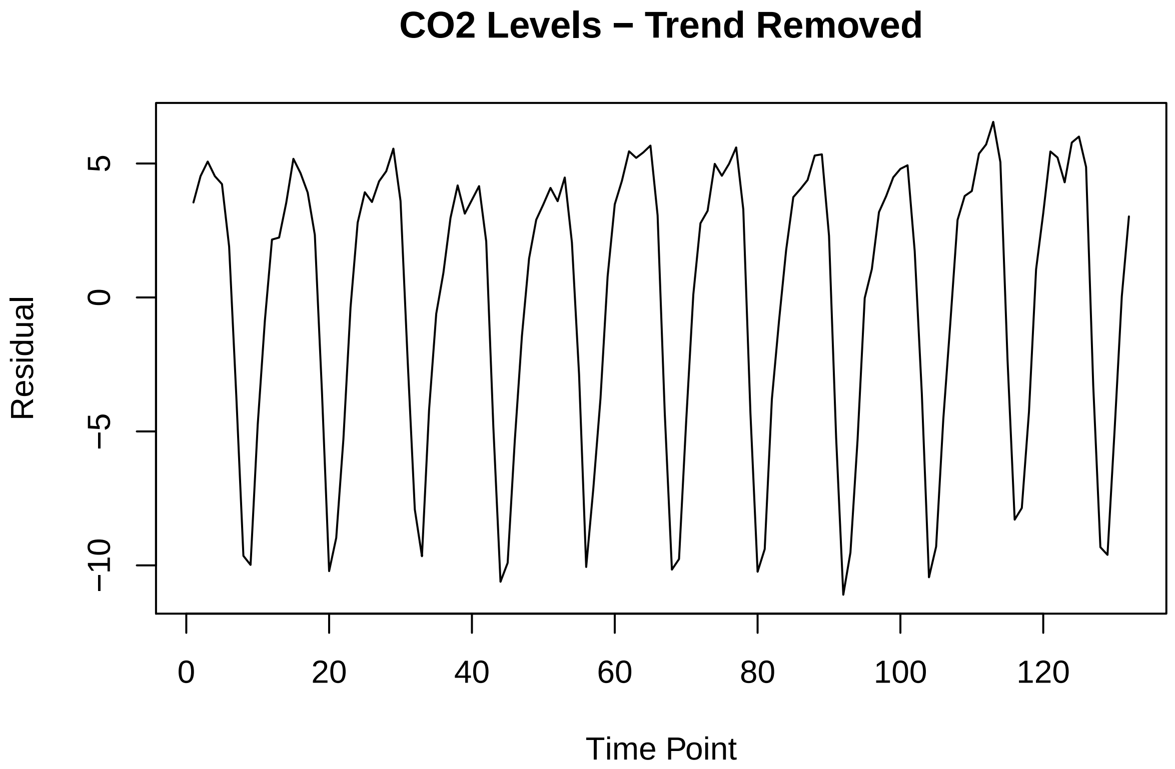


We will now fit the trend to the data and remove it. This trend is clearly linear, so there is no need for any polynomial fitting. A logarithmic transformation is not used either, as the variance does not significantly change through time. The x axis on all further plots that say "Time Point" represent the i^{th} data point observed, e.g. 4 represents the fourth data point, observed in April 1994.

```
# fit/remove the trend
trend.fit <- lm(x~t)
y.trend <- residuals(trend.fit)

ts.plot(x, main = "Monthly CO2 Levels at Alert, NW Terr., Canada",
        xlab = "Time Point", ylab = "CO2 Level (ppm)")
lines(fitted(trend.fit), col = "red")
```



```
ts.plot(resid(trend.fit), main = "CO2 Levels - Trend Removed",
        xlab = "Time Point", ylab = "Residual")
```



The seasonal component will now need to be removed. Visually, the data has a very clear seasonal component that looks like a sum of harmonics can easily remove it. A total of 12 sine and cosine functions are fit due to the fact that the period (d) is 12 months and our n.harm is d/2, or 6.

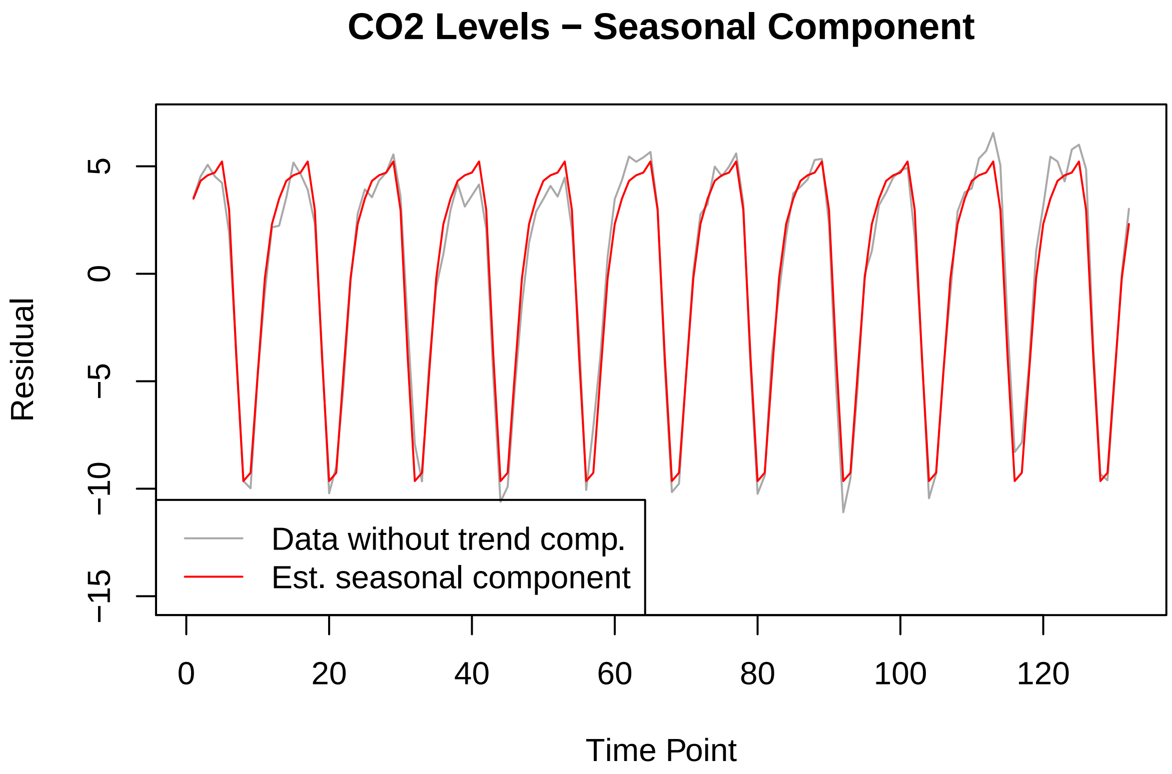
```
# remove the seasonal component
t <- (t) / n

d <- 12
n.harm <- 6 #set to [d/2]
harm <- matrix(nrow=length(t), ncol=2*n.harm)
for(i in 1:n.harm){
   harm[,i*2-1] = sin(n/d * i *2*pi*t)
   harm[,i*2] = cos(n/d * i *2*pi*t)
}
colnames(harm) <- paste0(c("sin", "cos"), rep(1:n.harm, each = 2))

# fit on all of the sines and cosines
dat <- data.frame(y.trend, harm)
fit <- lm(y.trend~., data=dat)
summary(fit)
```

To filter out the harmonic components that do not explain enough of the response in the full model, we use stepwise regression. This iterates from the full model and leaves us with a reduced model that contains only seven harmonic components; the AIC is reduced from -31.19 to -38.88. If any more harmonic components are removed or added back in, the AIC will increase.

```
# setup the full model and the model with only an intercept
full <- lm(y.trend~.,data=dat)
reduced <- lm(y.trend~1, data=dat)

# stepwise regression starting with the full model
fit.back <- step(full, scope = formula(reduced), direction = "both")
```

In the following plots, we can see the data without the trend component in grey, and the detrended, deseasonalized data in red. The sum of harmonics method has done a sufficient job, as the two plots are very similar.

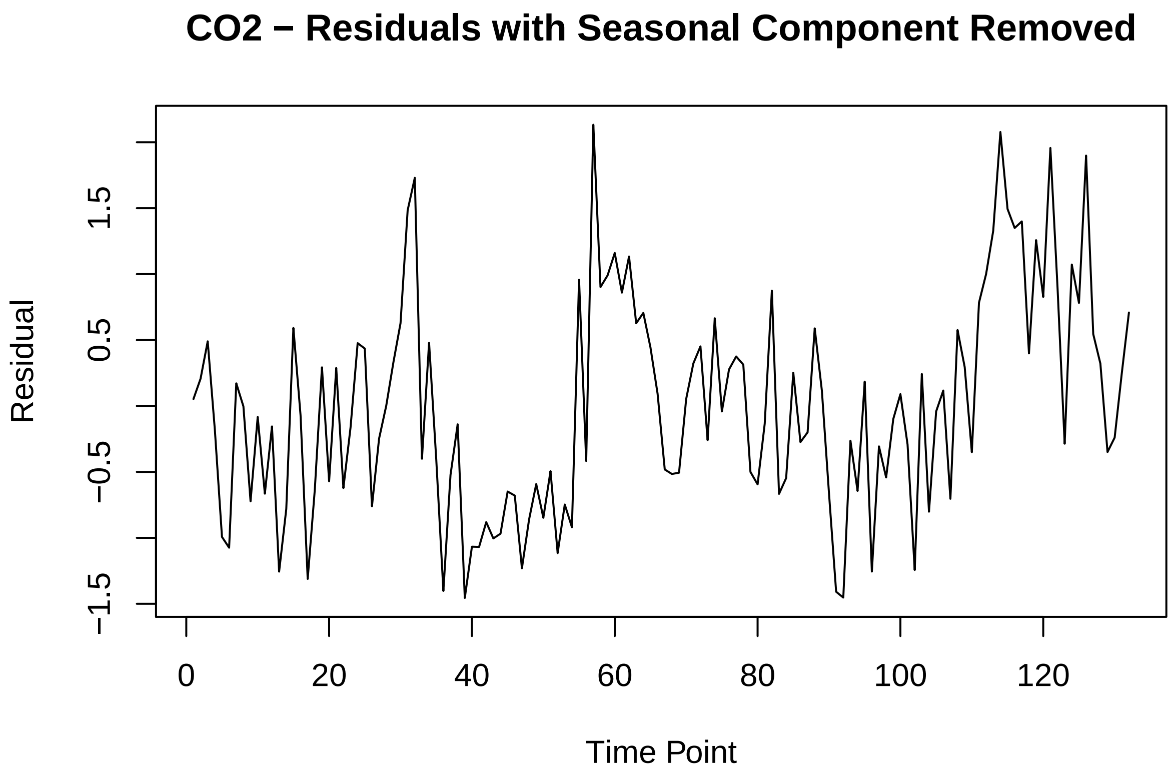
```
t <- 1:n

# plot the estimated seasonal components
plot(t, y.trend, type="l", col="darkgrey", ylim=c(-15,7),
     main = "CO2 Levels - Seasonal Component", xlab="Time Point", ylab="Residual")
lines(t, fitted(fit.back), col="red")
legend("bottomleft", lty = c(1,1),
       c("Data without trend comp.", "Est. seasonal component"), col = c("darkgrey","red"))
```



The following are the residuals after both previous components are removed. I checked if they are stationary with the Dickey-Fuller and KPSS tests; with Dickey-Fuller, the p-value is larger than 0.05, meaning that the residuals are not stationary, and the KPSS test agrees, giving a small p-value.

Running auto.arima() later will difference the residuals automatically, so when the final model is determined, the residuals will be stationary.

```
# plot the residuals after seasonal component is removed
ts.plot(residuals(fit.back),
        main="CO2 - Residuals with Seasonal Component Removed",
        ylab="Residual", xlab="Time Point")
```



```
# not stationary, check with adf.test - correct
adf.test(residuals(fit.back))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  residuals(fit.back)
## Dickey-Fuller = -2.7921, Lag order = 5, p-value = 0.2471
## alternative hypothesis: stationary
```

```
# check with kpss.test - not stationary
kpss.test(residuals(fit.back))
```

```
## Warning in kpss.test(residuals(fit.back)): p-value smaller than printed p-
## value
```

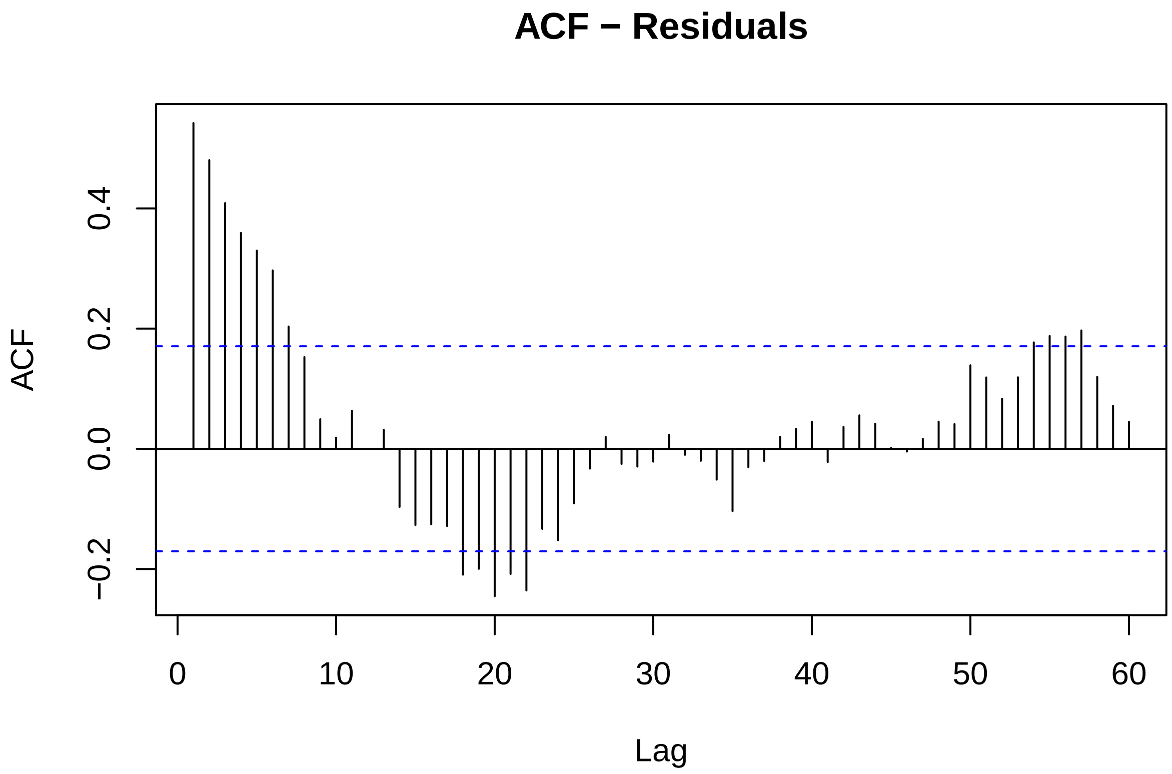
```
##
##  KPSS Test for Level Stationarity
##
```

```
## data:  residuals(fit.back)
## KPSS Level = 0.7543, Truncation lag parameter = 2, p-value = 0.01
```

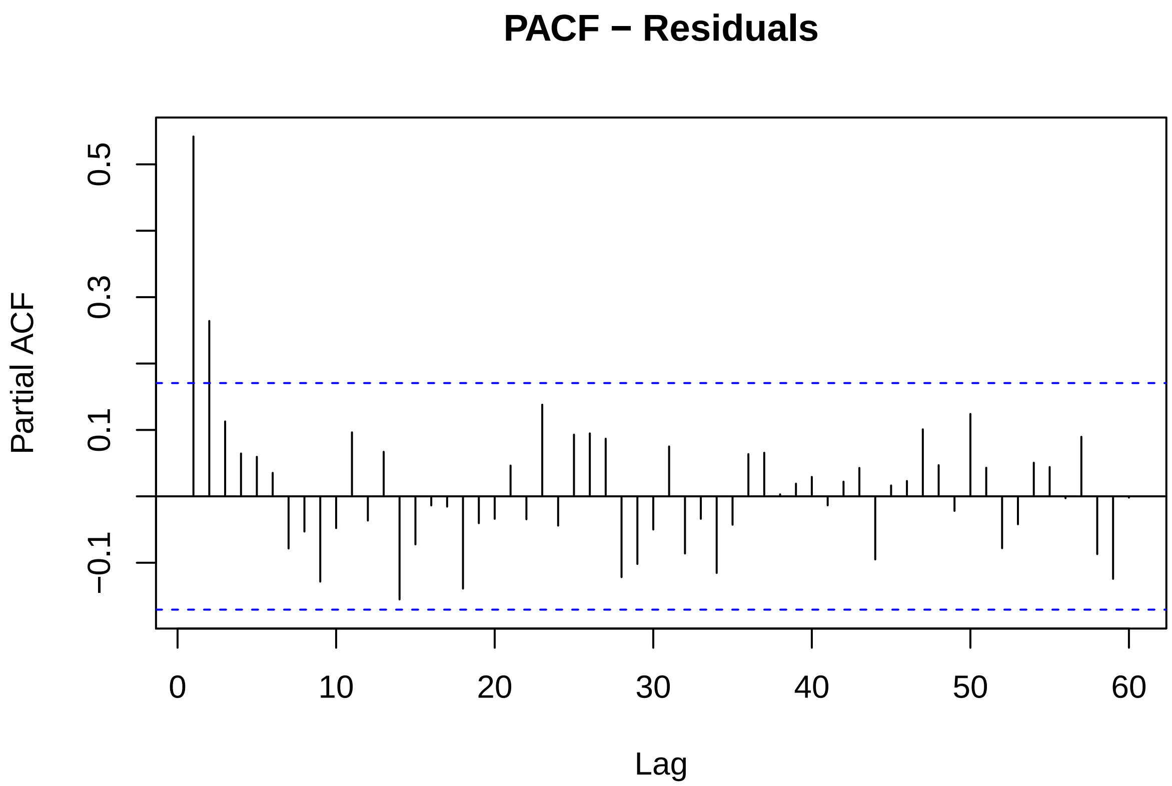
Now plot the autocovariance and partial autocovariance functions of the residuals to observe any dependence structure and see if a model can be pre-determined. In the ACF and PACF, the lags exhibit a great amount of dependence structure as can be seen by the slow tapering to 0 through oscillations. This indicates an ARMA(p,q) model.

Fitting a model with the auto.arima() function will difference automatically, so, in theory, there should be no dependence structure remaining after that.

```
# plot the acf and pacf of the residuals
y <- residuals(fit.back)

acf(y, lag.max=60, main = "ACF - Residuals")
```



```
pacf(y, lag.max=60, main = "PACF - Residuals")
```

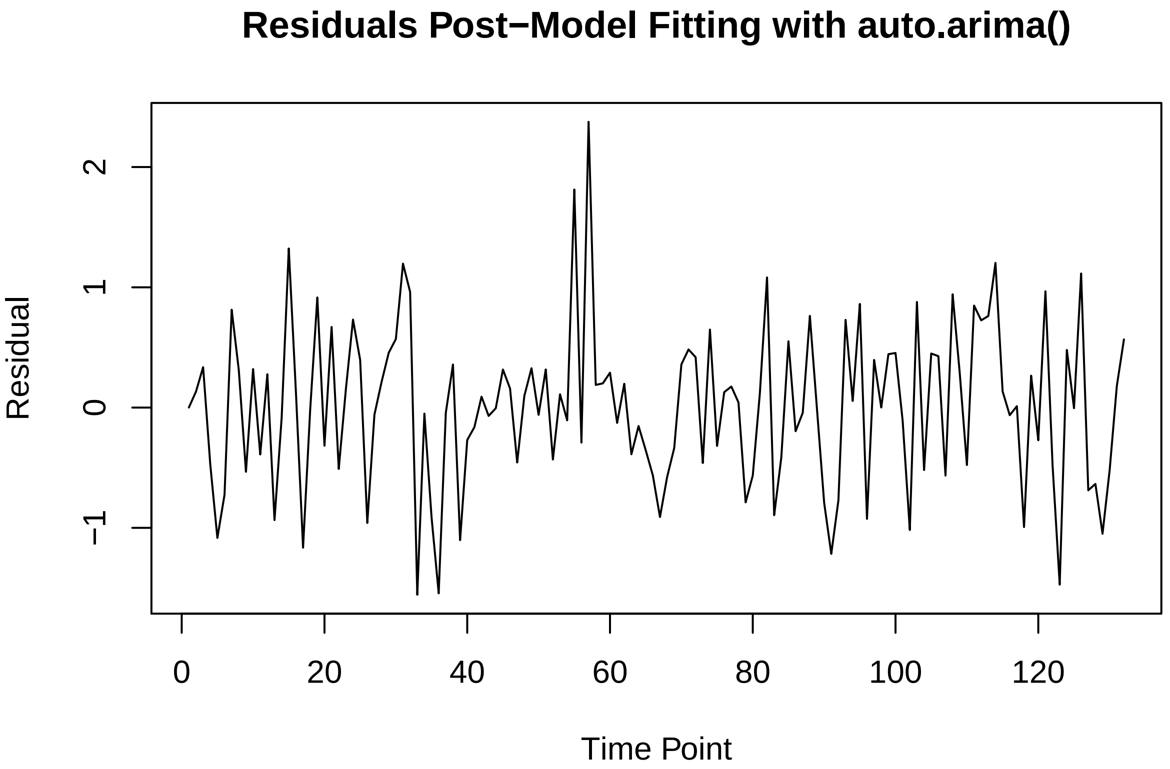


Here, the auto.arima() function fits an ARIMA(0,1,1) model. Both the ACF and PACF have no significant lags, as expected. The Dickey-Fuller test shows that these residuals are stationary because the p-value is less than 0.01. The Ljung-Box test is run, and with a p-value of 0.721, we can conclude that there is no dependence structure in these residuals; white noise remains. All results here agree.

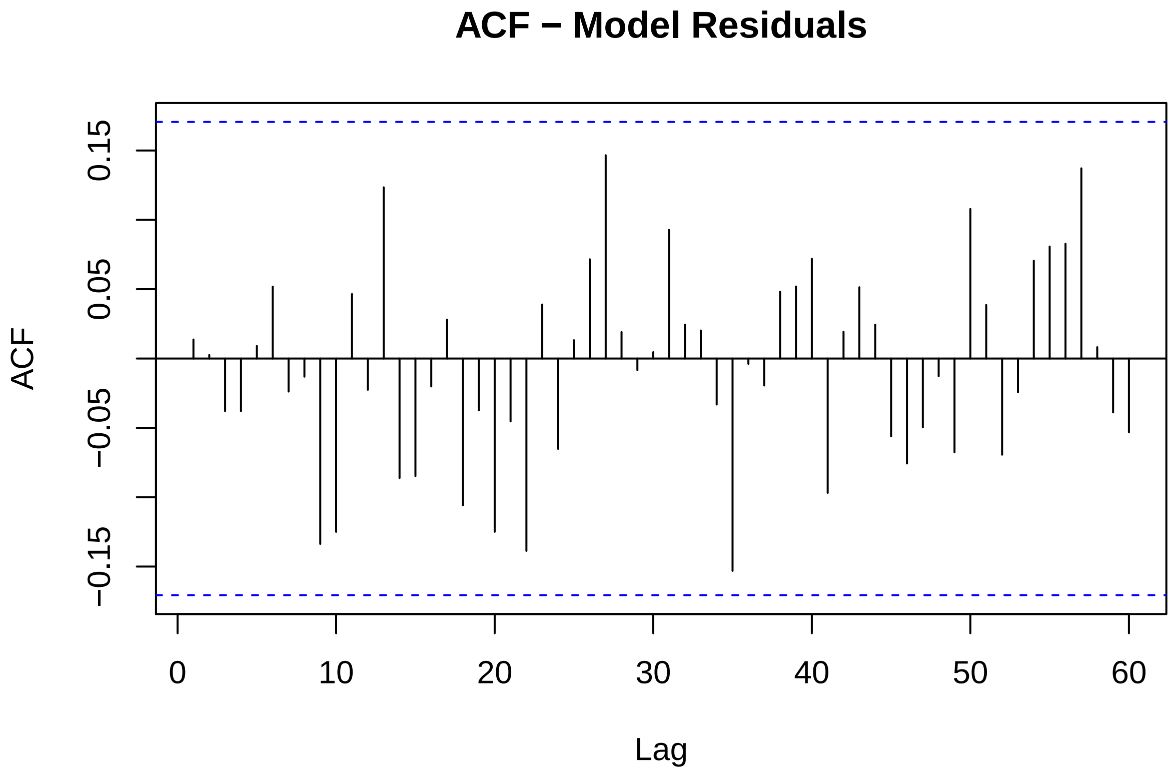
```
fit.y <- auto.arima(y, allowmean=FALSE, trace=FALSE, stepwise=FALSE)
fit.y
```

```
## Series: y
## ARIMA(0,1,1)
##
## Coefficients:
##           ma1
##       -0.5973
## s.e.   0.0751
##
## sigma^2 estimated as 0.4519:  log likelihood=-134.08
## AIC=272.16   AICc=272.25   BIC=277.91
```

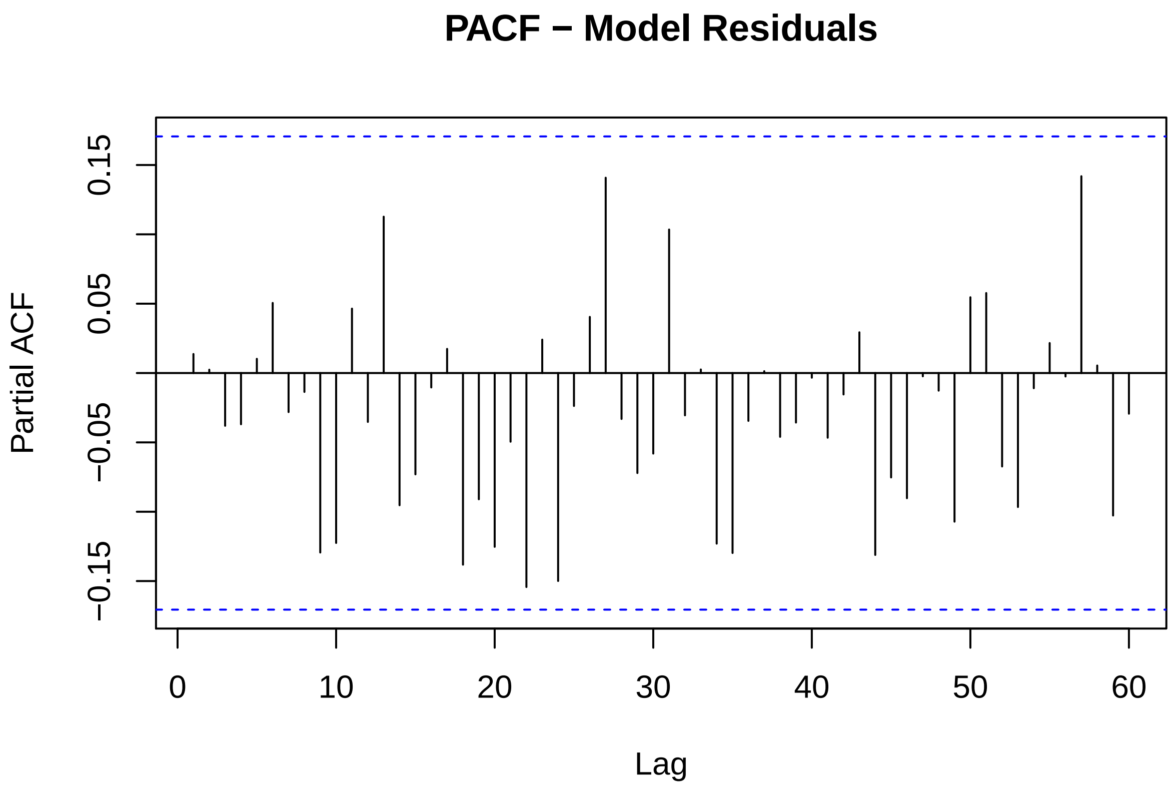
```
z <- resid(fit.y)

ts.plot(z, main = "Residuals Post-Model Fitting with auto.arima()",
        xlab = "Time Point", ylab = "Residual")
```



```r
acf(z, lag.max=60, main = "ACF - Model Residuals")
```



```
pacf(z, lag.max=60, main = "PACF - Model Residuals")
```



```
adf.test(z)
```

```
## Warning in adf.test(z): p-value smaller than printed p-value
```

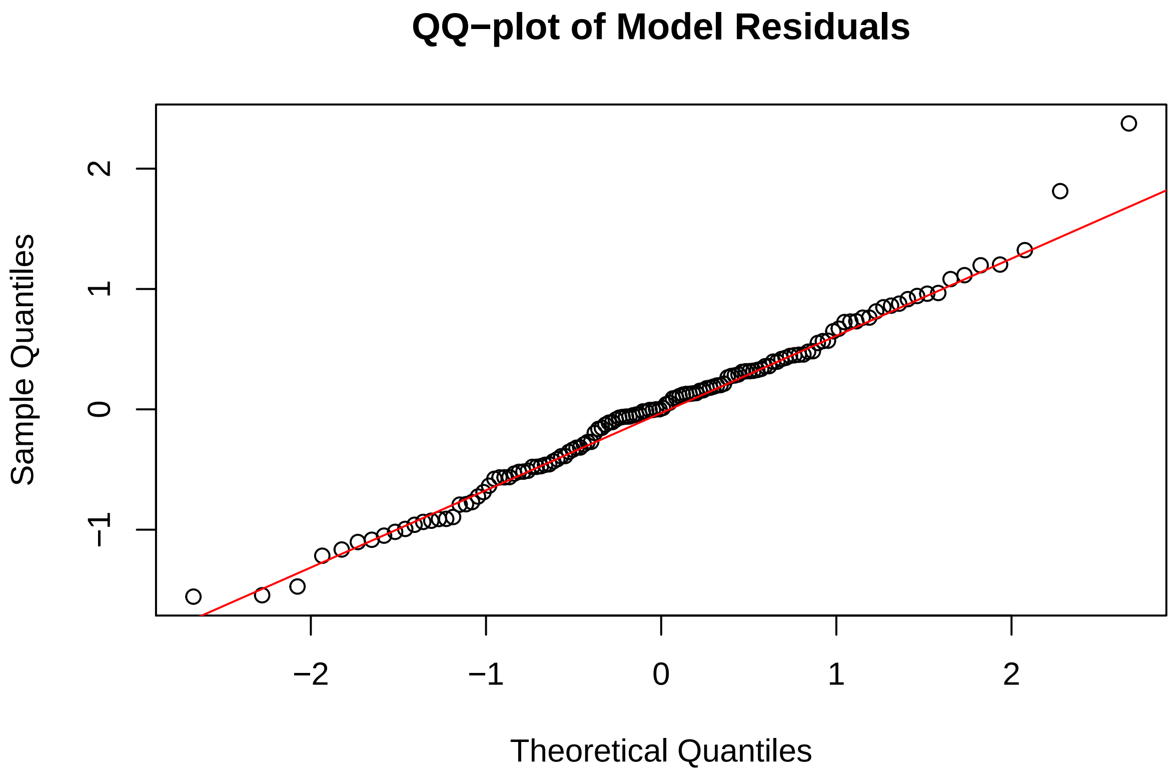
```
##
##  Augmented Dickey-Fuller Test
##
## data:  z
## Dickey-Fuller = -4.3785, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
Box.test(z,type="Ljung-Box", lag = floor(min(2*d, n/5)))
```

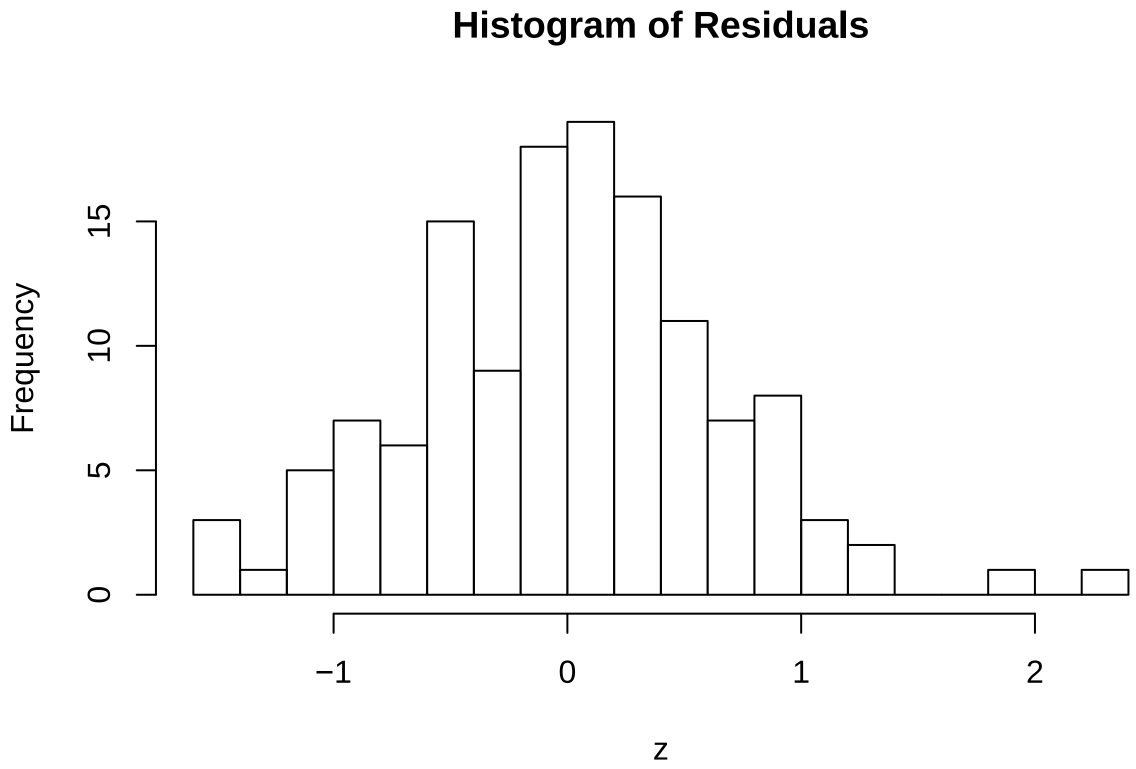
```
##
##  Box-Ljung test
##
## data:  z
## X-squared = 19.5691, df = 24, p-value = 0.721
```

In order to forecast the residuals in the next step, we need to check to see if normality can be assumed with a QQ-plot and a histogram. The QQ-plot has minimal deviations or indications of skewness or thicker tails at the ends, and the histogram looks roughly normal. Running the Shapiro-Wilk test gives a p-value of 0.3524, meaning we can fail to reject the null hypothesis; there is evidence to suggest that the residuals are normal. We can conclude that the upcoming forecast intervals are probably reliable.

```r
qqnorm(z, main = "QQ-plot of Model Residuals")
qqline(z, col = "red")
```



```r
hist(z, breaks = 20, main = "Histogram of Residuals")
```



```
shapiro.test(z)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  z
## W = 0.9887, p-value = 0.3524
```

Now, we will forecast the next 12 data points after December 2004 using the ARIMA(0,1,1) model derived from the auto.arima() output. The noise is forecasted from the auto.arima() model and then the mean of that is added to the 12 fitted values of the seasonal component. We then predict the 12 trend components and add those to the seasonal and noise forecasts to arrive at the final forecast for 2005.

```
# forecast the noise
fc <- forecast(fit.y, h=12, level=.95)

# forecast the seasonal component and noise
season.fc <- fit.back$fitted.values[1:12]+fc$mean

# forecast the trend
trend.fc <- predict(trend.fit, newdata=data.frame(t=133:144))

# add the seasonal and noise forecasts
x.hat <- season.fc+trend.fc
```

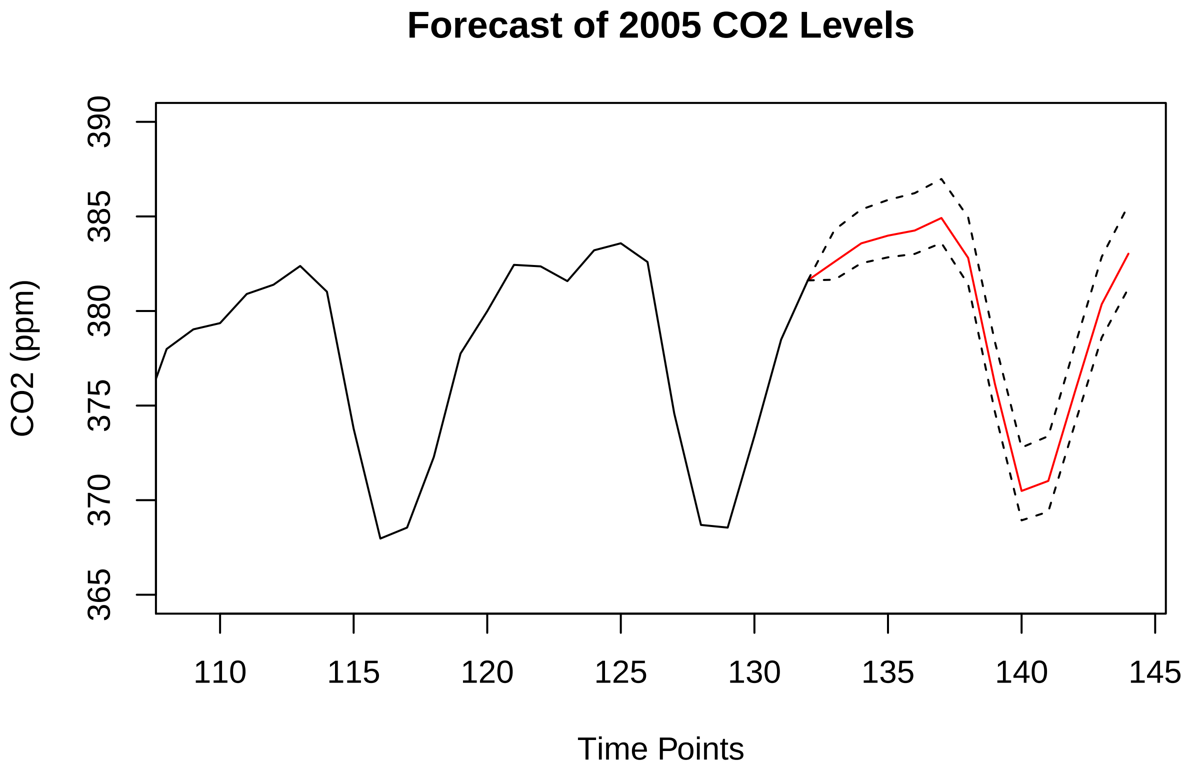
Results/Conclusion: The following is a table of the point forecasts and 95% forecast interval bounds for each month in 2005.

```
point_f <- data.frame(x.hat)
lower <- data.frame(x.hat+fc$lower)
upper <- data.frame(x.hat+fc$upper)

month <- data.frame(seq(1, 12, 1))
year <- data.frame(rep(2005, 12))

interval <- cbind(month, year, point_f, lower, upper)
colnames(interval) <- c("Month", "Year", "Point Forecast",
                        "95% Lower Bound", "95% Upper Bound")
interval
```

```
##    Month Year Point Forecast 95% Lower Bound 95% Upper Bound
## 1      1 2005       382.6022        381.6547        384.2899
## 2      2 2005       383.5780        382.5277        385.3686
## 3      3 2005       383.9846        382.8384        385.8710
## 4      4 2005       384.2560        383.0197        386.2325
## 5      5 2005       384.9177        383.5960        386.9796
## 6      6 2005       382.8096        381.4067        384.9527
## 7      7 2005       376.1533        374.6727        378.3741
## 8      8 2005       370.4874        368.9323        372.7828
## 9      9 2005       371.0177        369.3909        373.3848
## 10    10 2005       375.7485        374.0523        378.1848
## 11    11 2005       380.3597        378.5965        382.8631
## 12    12 2005       383.0299        381.2018        385.5983
```

The following is the zoomed in plot of the observed CO2 levels (black solid) and forecasted levels (red) for 2005. The black dashed lines are the 95% forecast interval bounds.

```
plot(t, co2, xlim = c(109,144), ylim = c(365,390), type="l",
     main="Forecast of 2005 CO2 Levels", ylab="CO2 (ppm)", xlab="Time Points")
lines(132:144, c(co2[132], x.hat), col="red")

# add the forecast intervals
lines(132:144, c(co2[132], x.hat+fc$lower), col="black", lty=2)
lines(132:144, c(co2[132], x.hat+fc$upper), col="black", lty=2)

# the actual values
lines(132:144, co2[132:144], col="blue")
```



In conclusion, the point estimates and 95% forecast intervals for each month in 2005 show that the CO_2 levels' trend should rise in a linear fashion and the well-defined seasonality should continue. When fitting a model to this data without removing the trend or seasonality first, the AIC is slightly lower than that of the model I fit after removing trend and seasonality; however, when the residuals of the model were inputted into the Shapiro-Wilk test, the p-value was 0.1652, so those forecast values would not have been reliable. Therefore, I use a more self-constructed model and use the process shown here - it generates normal residuals with which to forecast. These forecasts can be extended beyond that of 2005, which may play a further role in deciding whether or not the Canadian government and/or a coalition of major countries should intervene and take more aggressive action in preventing this steady increase in CO_2 levels in our atmosphere.