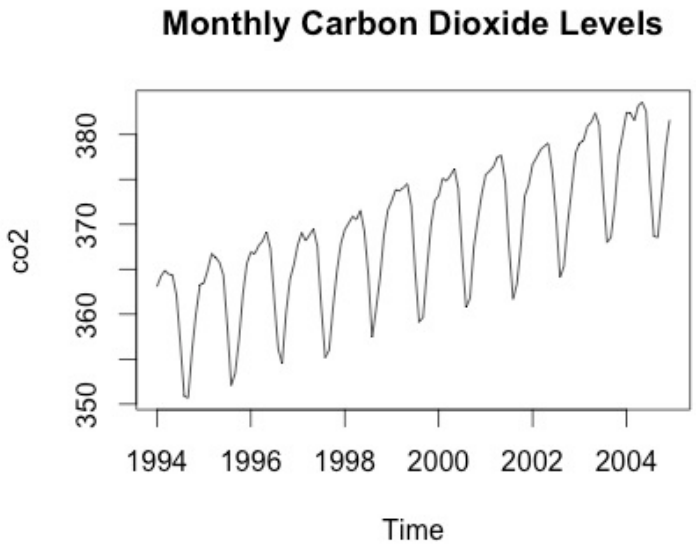
Matthew Lee
STA 137 A03
Professor Joshua Patrick

Carbon Dioxide Levels: Alert, Canada

INTRODUCTION

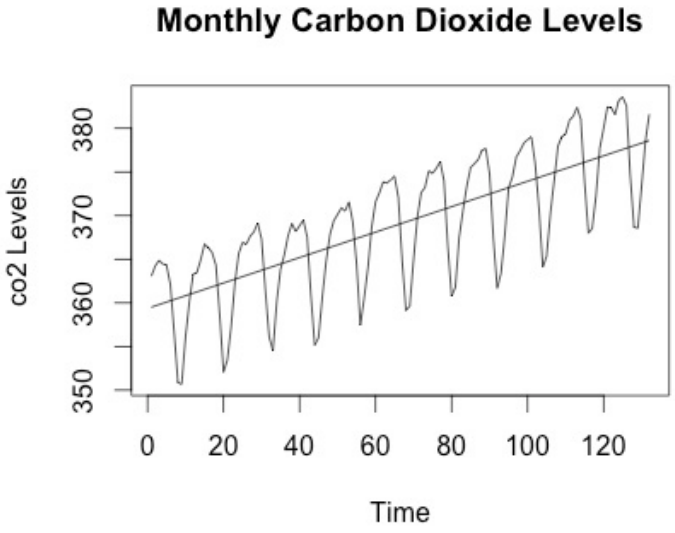
In order to do an accurate forecast of the next 12 months, it is important to understand the dataset. The dataset "co2" is a time series data that can be found in the TSA package. Carbon dioxide levels at Alert, Canada were taken monthly between January 1994 and December 2004 and measured in parts per million (ppm). From the time series plot, it appears that the months February-April have the highest levels of carbon dioxide, while August-September tend to have the lowest levels of carbon dioxide. The raw data also appears to be not be heteroskedastic, therefore a log transformation was not necessary. Another important aspect to note is that carbon dioxide levels have been increasing over time.

Figure 1.



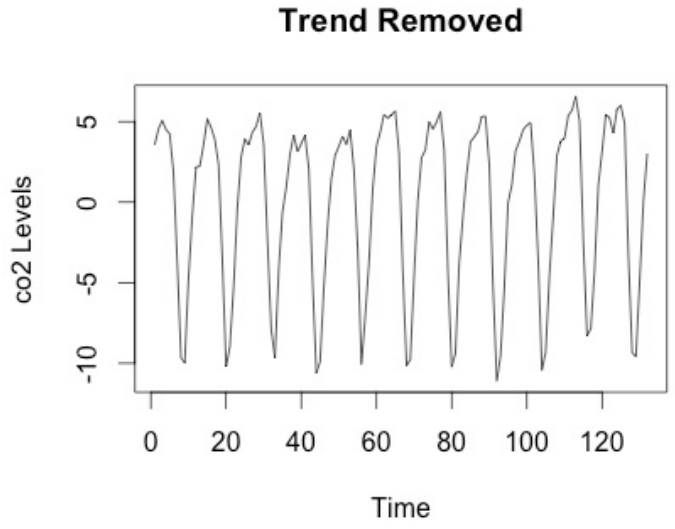
TIME SERIES MODELING

Figure 2.



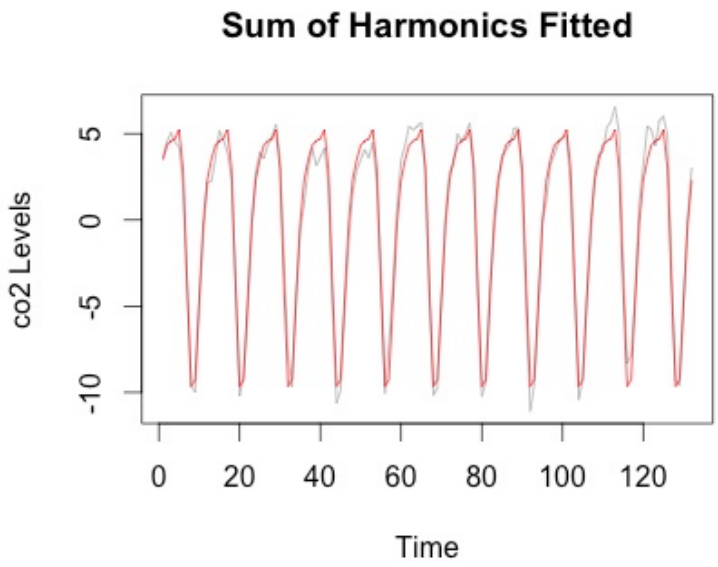
In order to accurately forecast the next 12 month's carbon dioxide levels, the data must be stationary. The data is clearly not stationary since there is both a trend and seasonality. From figure 2, a linear model of order 1 was used to remove the trend.

Figure 3.



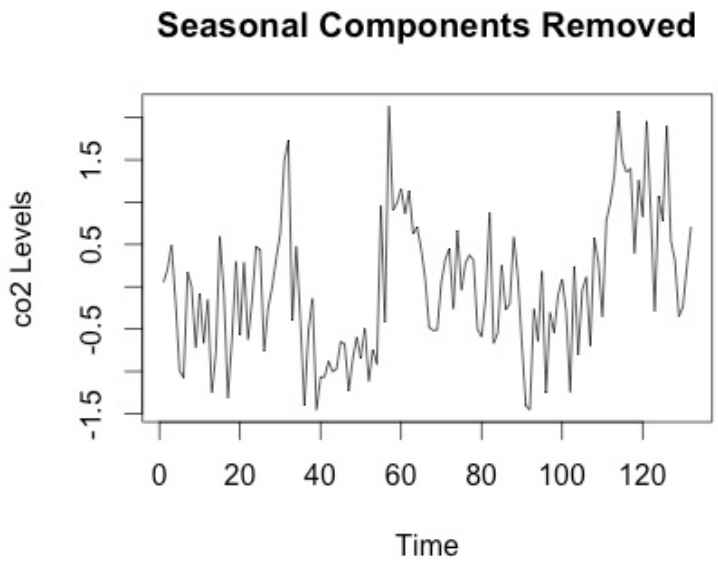
After removing the trend, the data begins to look more stationary. However, the plot is still not stationary since there are seasonal fluctuations where carbon dioxide levels are consistently higher and lower. To remove this, a sum of harmonics was fitted to the model above.

Figure 4.



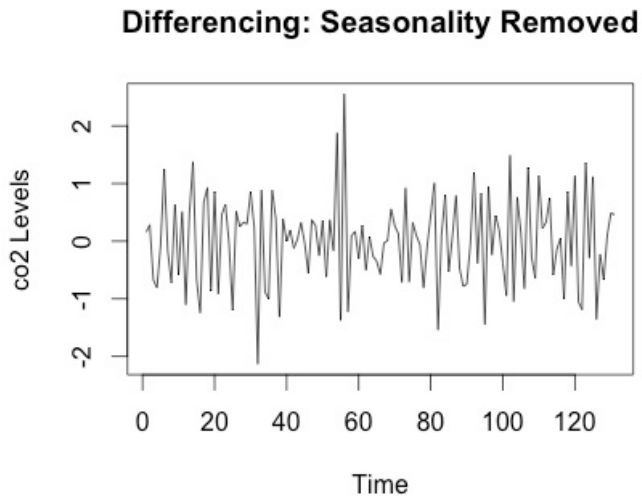
After the harmonics were fitted, the residuals were found. The new plot in figure 5 now looks like it could be stationary. In order to determine whether it was stationary or not, the Dickey-Fuller and KPSS test were used.

Figure 5.



The Dickey-Fuller test had a p-value of 0.2471, which is too large to reject, therefore the data is still non-stationary. After conducting the KPSS test, the p-value was 0.01 and the same conclusion of non-stationary was made. The next step was to use differencing to stabilize the mean and hopefully remove all of the remaining trend and seasonality.

Figure 6.

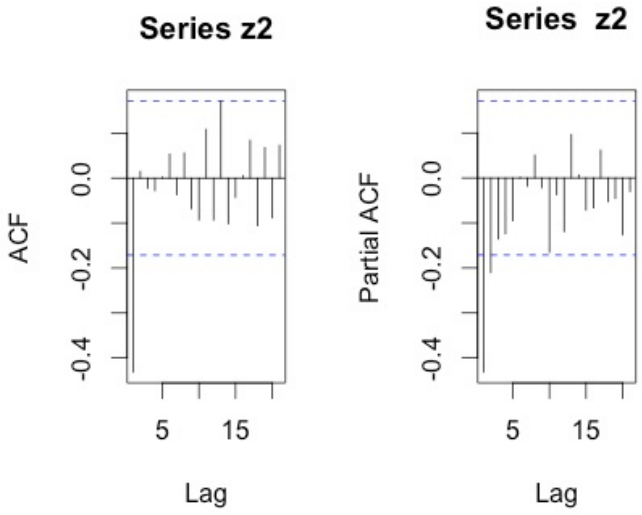


Because the sum of harmonics already removed the seasonality component, a difference of order 1 was used. Now the Dickey-Fuller test and KPSS test both conclude that the data is stationary. Now that the data is stationary, all that is left before forecasting is to fit an appropriate ARIMA model to determine whether or not residuals are white noise.

MODEL SELECTION

The final step is to try to fit an ARMA model onto the plot and see if the residuals left over are going to be white noise and normally distributed. The ACF and PACF plots did not clearly indicate what type of model to fit, so the function "auto.arima" was used.

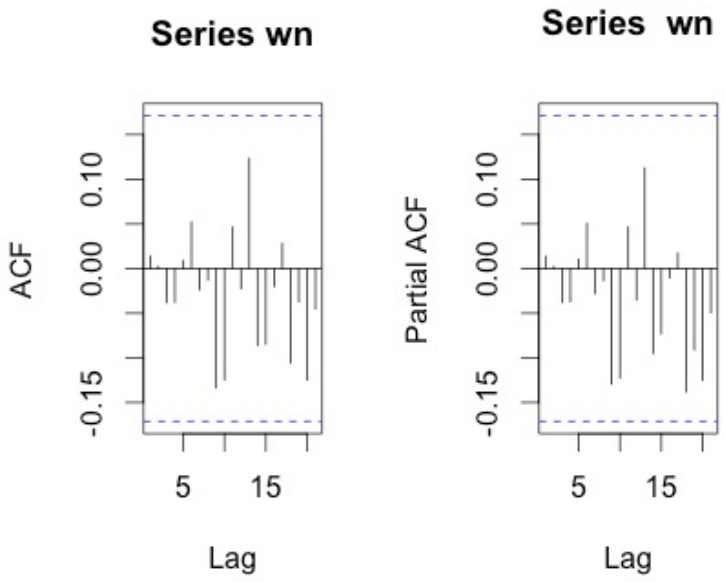
Figure 7.



Auto.arima chose an MA(1) function that would best fit the time series plot from figure 6. It is important to remember that this function chooses the lowest AIC, and tends to select an underfitted model. The best fitted model can be expressed as $X_t = Z_t - 0.5973Z_{t-1}$.

After the MA(1) model was fitted to the data, the residuals were found, and the ACF and PACF were plotted. The ACF and PACF plots from figure 8 do not indicate any significant values at any of the lags, so white noise is assumed. To further test this, the Ljung-Box test was conducted on the residuals, resulting in a p-value of 0.7282. Since the p-value is large, the null hypothesis is not rejected and the residuals are indeed concluded to be white noise.

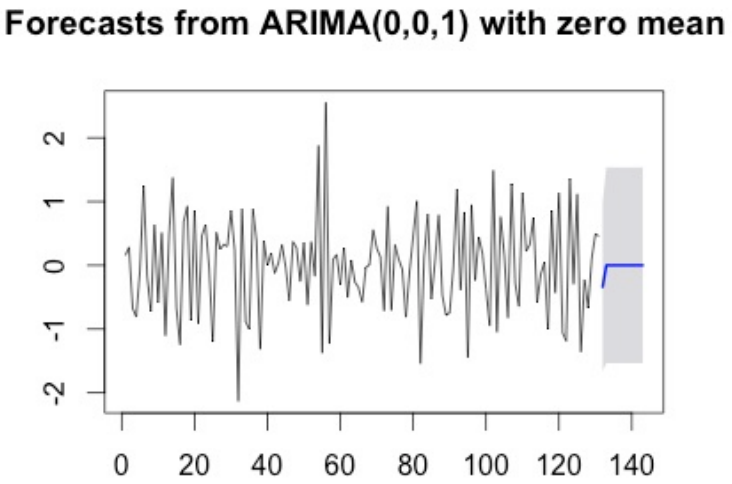
Figure 8.



Before forecasting can be done, it is important to distinguish whether or not the white noise residuals are normally distributed or not. Therefore, the Shapiro-Wilk test was performed, resulting in a p-value of 0.3749. This p-value is larger than 0.05 so the conclusion was that the residuals are indeed normally distributed.

FORECASTING

Figure 9.



The first step for forecasting was to plot the noise forecast and then work backwards to fit the model again. In order to undo the difference, the "diffinv" function was used. The next step was to forecast the seasonal and trend components for the 133-144 data points. In figure 10 below, the next 12 months of carbon dioxide levels are predicted, as well as the lower and upper bounds with a 95% confidence level.

Figure 10.

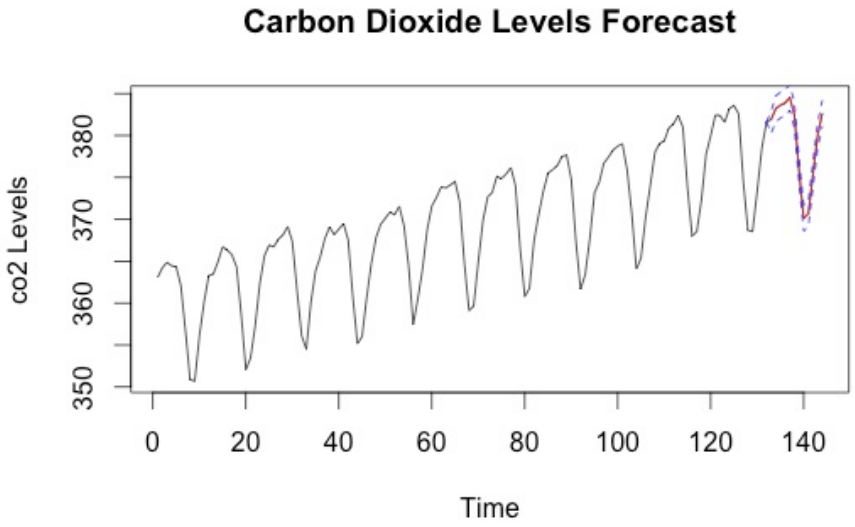


Figure 11.

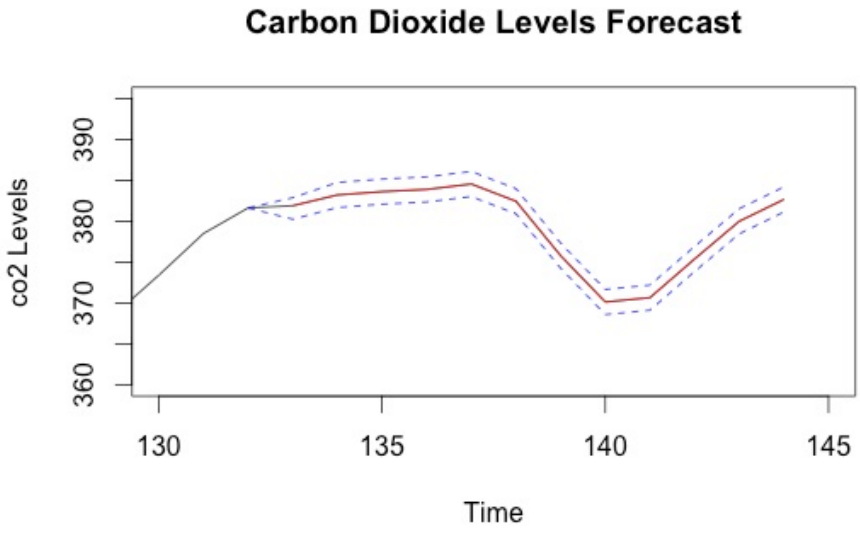


Figure 11 is simply a zoomed in plot of the forecast from January 2005-December 2005. Below in table 1 are the actual forecasted carbon dioxide levels and a 95% interval forecast for the next 12 months. The results are similar to what one would expect, with February-April having the highest levels of carbon dioxide, and August-September having the lowest levels. The people of Alert, Canada should expect an increase in carbon dioxide levels in 2005, similar to what has been seen over the last 10 years.

Table 1.

Month	***Carbon Dioxide Levels***	***95% Interval Forecast***
January	381.89 ppm	(380.24, 382.87)
February	383.21 ppm	(381.67, 384.74)
March	383.61 ppm	(382.08, 385.15)
April	383.89 ppm	(382.35, 385.42)
May	384.55 ppm	(383.01, 386.08)
June	382.44 ppm	(380.9, 383.97)
July	375.78 ppm	(374.25, 377.32)
August	370.12 ppm	(368.58, 371.65)
September	370.65 ppm	(369.11, 372.18)
October	375.38 ppm	(373.84, 376.91)
November	379.99 ppm	(378.45, 381.52)
December	382.66 ppm	(381.13, 384.19)

APPENDIX

```
library(tseries)
library(forecast)
require(TSA)

par(mfrow=c(1,1))
plot(co2,main="Monthly Carbon Dioxide Levels")
x=as.vector(co2)
t=as.vector(time(x))
n=length(x)
kpss.test(co2)
plot(t,x,type="l",ylab="co2 Levels",xlab="Time", main="Monthly Carbon Dioxide Levels")

trend.fit=lm(x~t)
lines(t, fitted(trend.fit))

y=residuals(trend.fit)
plot(t,y,type="l",ylab="co2 Levels", main="Trend Removed", xlab="Time")

n=length(t)
t=1:length(y)
t=(t)/n
d=12
n.harm=6
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))
dat = data.frame(y, harm)
fit = lm(y~., data=dat)
summary(fit)
full = lm(y~.,data=dat)
reduced = lm(y~1, data=dat)
fit.back = step( full, scope = formula(reduced),trace=F, direction = "both")
fit.back
t = as.vector(time(x))

plot(t,y, type="l", col="darkgrey", ylab="co2 Levels",xlab="Time", main="Sum of Harmonics
Fitted")
lines(t, fitted(fit.back), col="red")
```

```
ts.plot(residuals(fit.back), ylab="co2 Levels",main="Seasonal Components Removed",
xlab="Time")

z1=residuals(fit.back)
par(mfrow=c(1,2))
acf(z1)
pacf(z1)
adf.test(z1)
kpss.test(z1)
z2=diff(z1)
ts.plot(z2, ylab="co2 Levels",main="Differencing: Seasonality Removed", xlab="Time"))

par(mfrow=c(1,2))
acf(z2)
pacf(z2)
adf.test(z2)
kpss.test(z2)

arma.fit=auto.arima(z2)
arma.fit
wn = resid(arma.fit)
acf(wn, na.action = na.pass)
pacf(wn, na.action = na.pass)

Box.test(wn, type="Ljung-Box",lag = min(2*d, floor(n/5)) )
shapiro.test(wn)
shapiro.test(z2)
par(mfrow=c(1,1))
noise.f = forecast(arma.fit, 12,level=.95)
plot(noise.f)

y.fc = c(z2,noise.f$mean)
fc.all = diffinv(y.fc, difference=1, xi =  x[1] )
fc = fc.al[133:144]
season.f = fitted(fit.back)[1:12]
plot(season.f+noise.f$mean)
lines(132:143, season.f, col="blue")
t.f = 133:144
t.f2 = t.f^2
trend.f = predict(trend.fit,newdata = data.frame(t=t.f, t2 = t.f2))
x.f = trend.f + season.f + noise.f$mean
plot(x.f)
plot(1:144, c(x,x.f), type="l", col="black",main="Carbon Dioxide Levels
Forecast",xlab="Time",ylab="co2 Levels")
lines(133:144, (x.f), col="red")
lines(132:144, c(x[132], x.f+noise.f$lower), col="blue", lty=2)
```

```
lines(132:144, c(x[132], x.f+noise.f$upper), col="blue", lty=2)

plot(1:144, c(x,x.f), type="l", col="black",main="Carbon Dioxide Levels
Forecast",xlim=c(130,145),ylim=c(360,395),xlab="Time",ylab="co2 Levels")
lines(133:144, (x.f), col="red")
lines(132:144, c(x[132], x.f+noise.f$lower), col="blue", lty=2)
lines(132:144, c(x[132], x.f+noise.f$upper), col="blue", lty=2)
```