# STA 137 Forecasting

*Janice Luong*

*March 11, 2016*

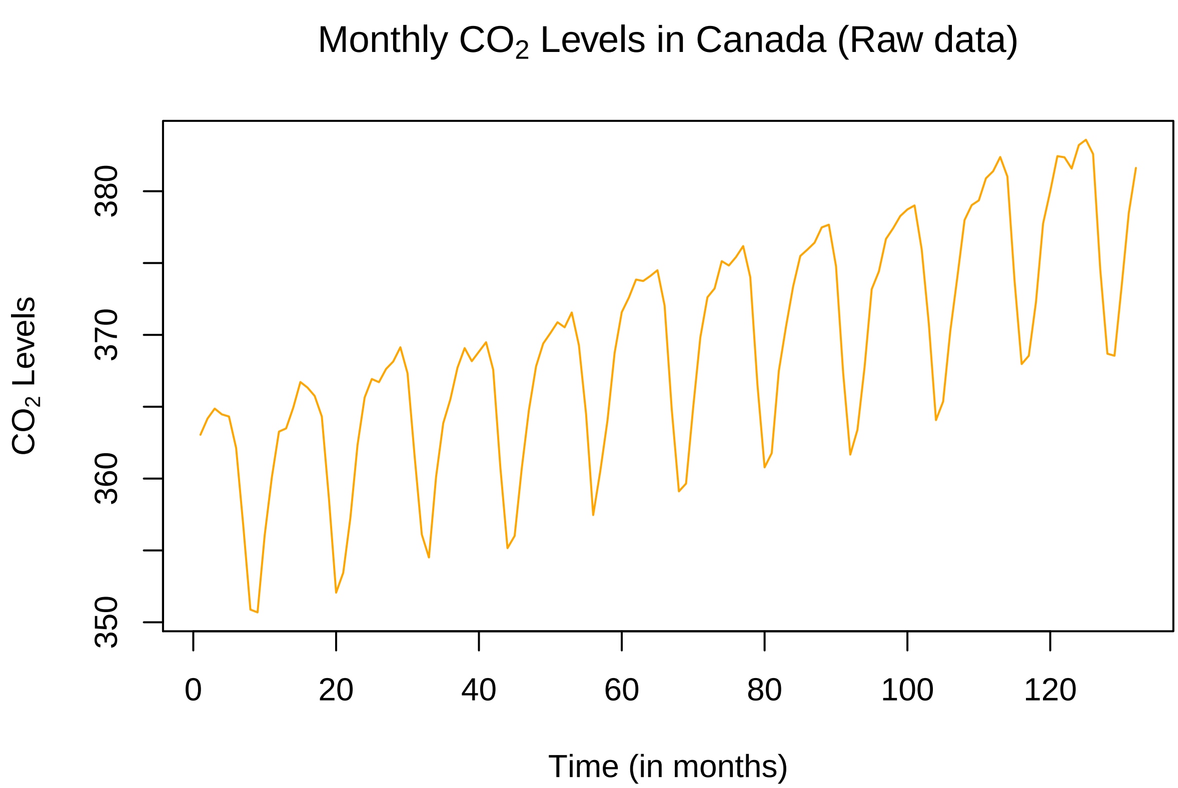
**Description of the Data**

The data set consists of monthly $CO_2$ levels at Alert, Northwest Territories, Canada. The data was collected from January 1994 through December 2004. With the given data, I will forecast the $CO_2$ levels for every month of 2005 (next 12 months) and provide interval forecasts for the next 12 months.

```
suppressMessages(library(forecast))
suppressMessages(library(tseries))

data("co2", package = "TSA")
x = as.vector(co2)
n = length(x)
t = 1:n
```

Here is the raw data.

```
plot(t,x, type="l",
     main = expression("Monthly "*CO[2]*" Levels in Canada (Raw data)"),
     xlab = "Time (in months)", ylab=expression(""*CO[2]*" Levels"), col = "orange")
```



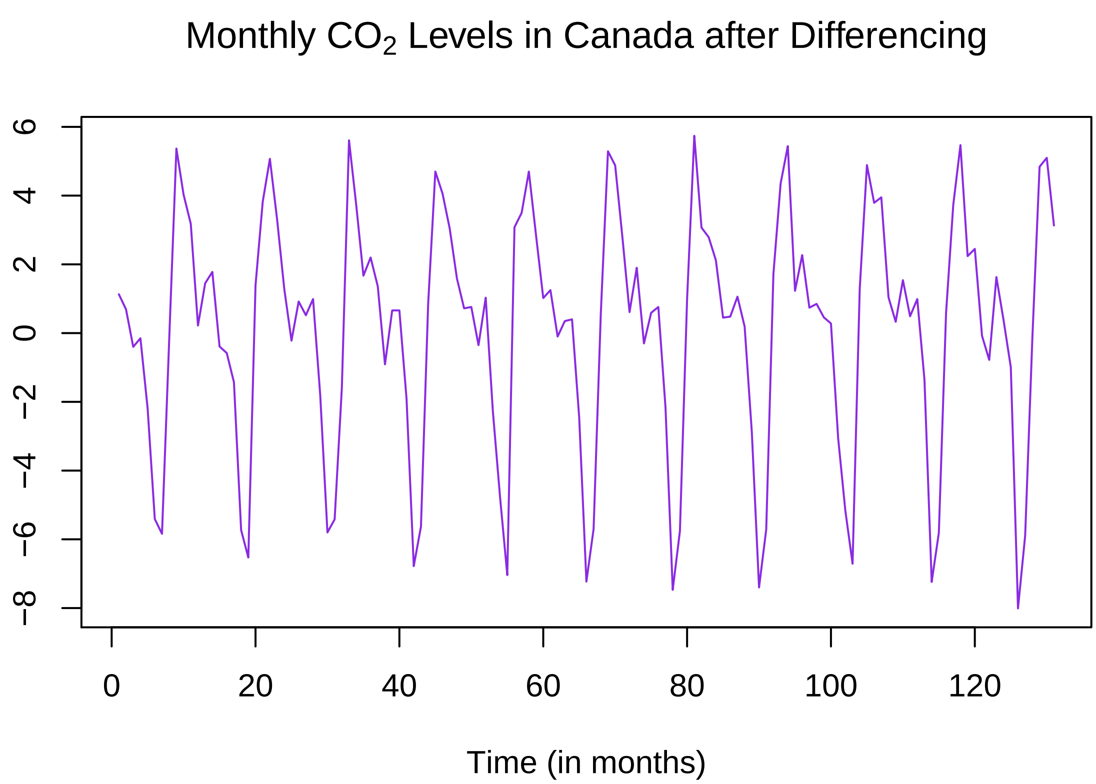
Through visual inspection of the data, I decided that a transformation was not necessary, and no outliers are present. I will need to remove the trend and seasonality so we can have white noise remaining.

I will remove the trend by taking the difference.

### Deterministic Components

Here is what the data looks like after taking the difference.

```
y = diff(x)

ts.plot(y, type = "l", col = "blueviolet", xlab = "Time (in months)", ylab = "",
        main = expression("Monthly "*CO[2]*" Levels in Canada after Differencing"))
```




I will use both the ADF and KPSS test to test to see if the residuals are stationary.

```
adf.test(y)
```

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

##
##  Augmented Dickey-Fuller Test
##
## data:  y
## Dickey-Fuller = -7.3263, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(y)
```

```
## Warning in kpss.test(y): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  y
## KPSS Level = 0.013899, Truncation lag parameter = 2, p-value = 0.1
```

*adf.test null and alternative hypthoese:*

$H_0$ = The residuals are not stationary

$H_a$ = The residuals are stationary

*kpss.test null and alternative hypthoese:*

$H_0$ = The residuals are stationary

$H_a$ = The residuals are not stationary

After removing the trend, based on the adf.test, our residuals are stationary because the p-value is smaller than 0.01. Since the the p-value is less than $\alpha = 0.05$, we reject $H_0$.

This can also be seen in the kpss.test. Our residuals are stationary because p-value is larger that 0.1. Since the p-value is greater than $\alpha = 0.05$, we failt to reject $H_0$.

I noticed there is also a seasonal component, so I will need to remove it to have white noise remaining. To remove the seasonal componment, I used sum of harmonics because the seasonality based on the plot above has a very apprent seasonality.

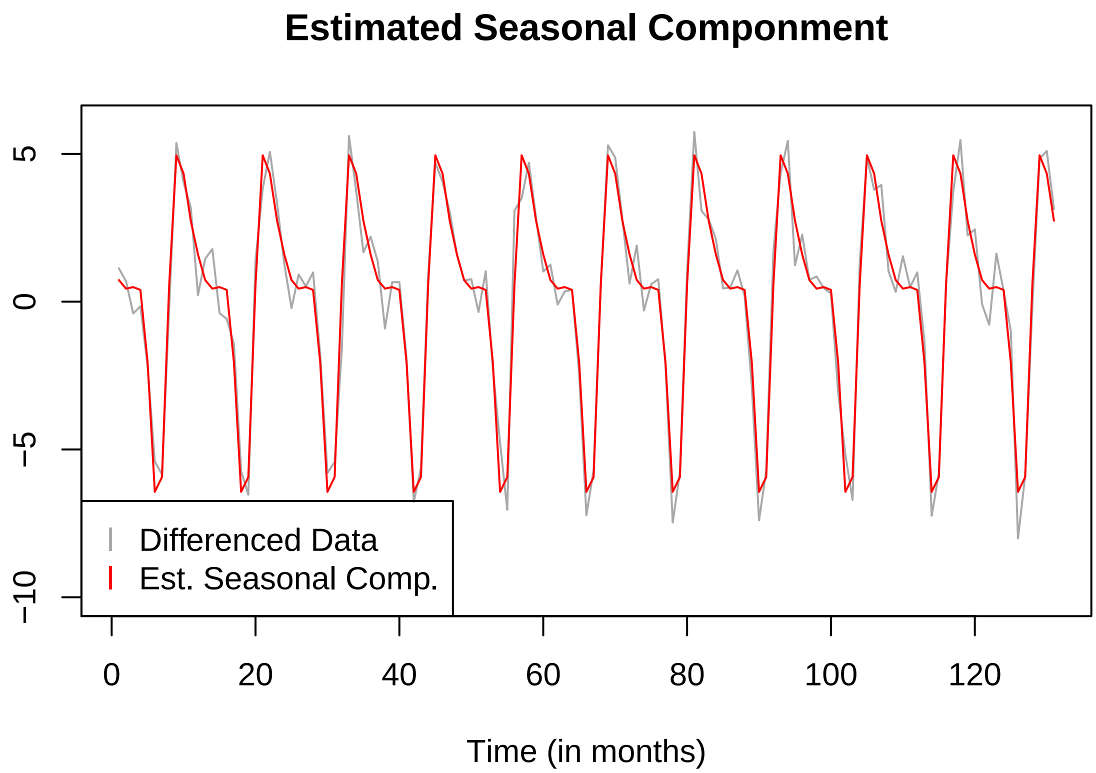
```r
#use t that is in the interval [0,1]
n = length(t)
t = 1:length(y)
t = (t) / n

# make matrix of the harmonics
d=12 #number of time points in each season
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, harm)
fit = lm(y~., data=dat)

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

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

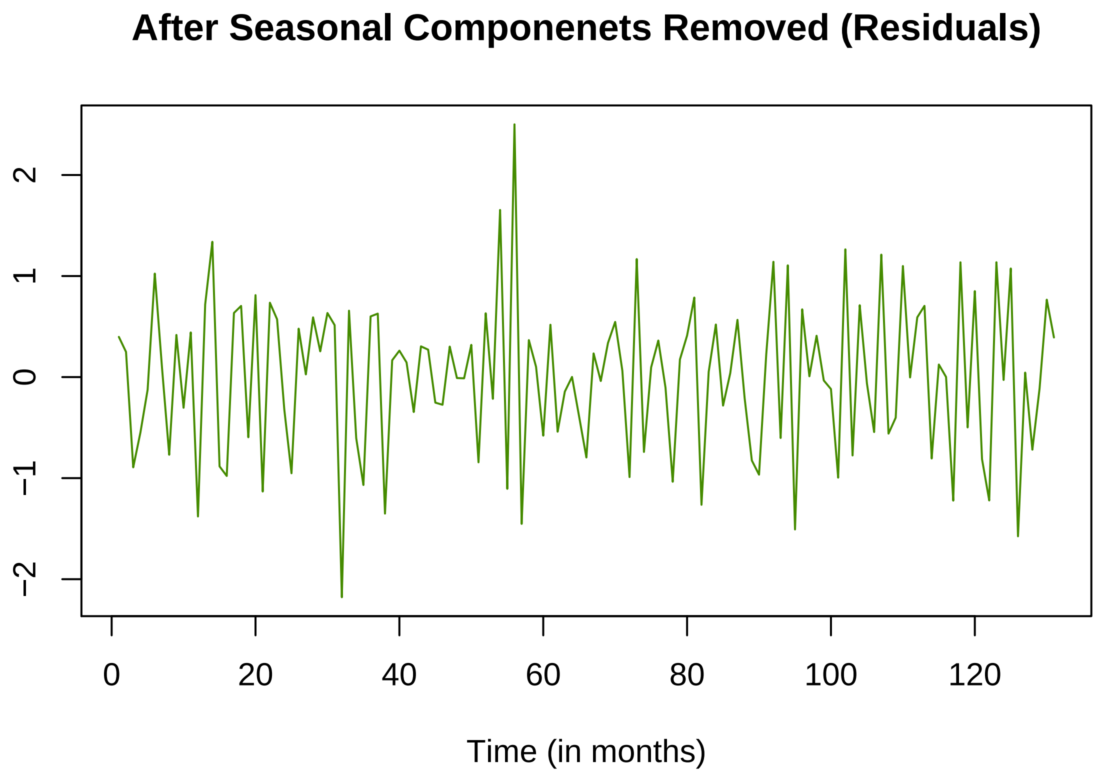
```r
summary(fit.back)
```

```
##
## Call:
## lm(formula = y ~ sin1 + cos1 + sin2 + cos2 + sin3 + cos3 + sin4,
##     data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.17869 -0.56947  0.04432  0.55641  2.50131
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.15280    0.06917   2.209  0.02902 *
## sin1        -1.17586    0.09743 -12.068  < 2e-16 ***
## cos1         3.31085    0.09821  33.712  < 2e-16 ***
## sin2        -1.38393    0.09743 -14.204  < 2e-16 ***
## cos2        -2.57024    0.09821 -26.171  < 2e-16 ***
## sin3         1.05379    0.09743  10.816  < 2e-16 ***
## cos3         0.70605    0.09821   7.189 5.53e-11 ***
## sin4        -0.31216    0.09743  -3.204  0.00173 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7915 on 123 degrees of freedom
## Multiple R-squared:  0.9507, Adjusted R-squared:  0.9479
## F-statistic: 339.2 on 7 and 123 DF,  p-value: < 2.2e-16
```

```r
#get back the original t so that we can plot over this range
t = 1:length(y)

#plot the estimated seasonal components
plot(t,y, type="l", col="darkgrey", ylab="", xlab = "Time (in months)",
     main = "Estimated Seasonal Componment", ylim = c(-10,6))
lines(t, fitted(fit.back), col="red")
legend("bottomleft", c("Differenced Data", "Est. Seasonal Comp."),
       pch = "l", col = c("darkgrey", "red"))
```




```r
#plot the residuals after seasonal component is removed
ts.plot(residuals(fit.back),
        main="After Seasonal Componenets Removed (Residuals)",
        ylab = "", xlab="Time (in months)", col = "chartreuse4")
```

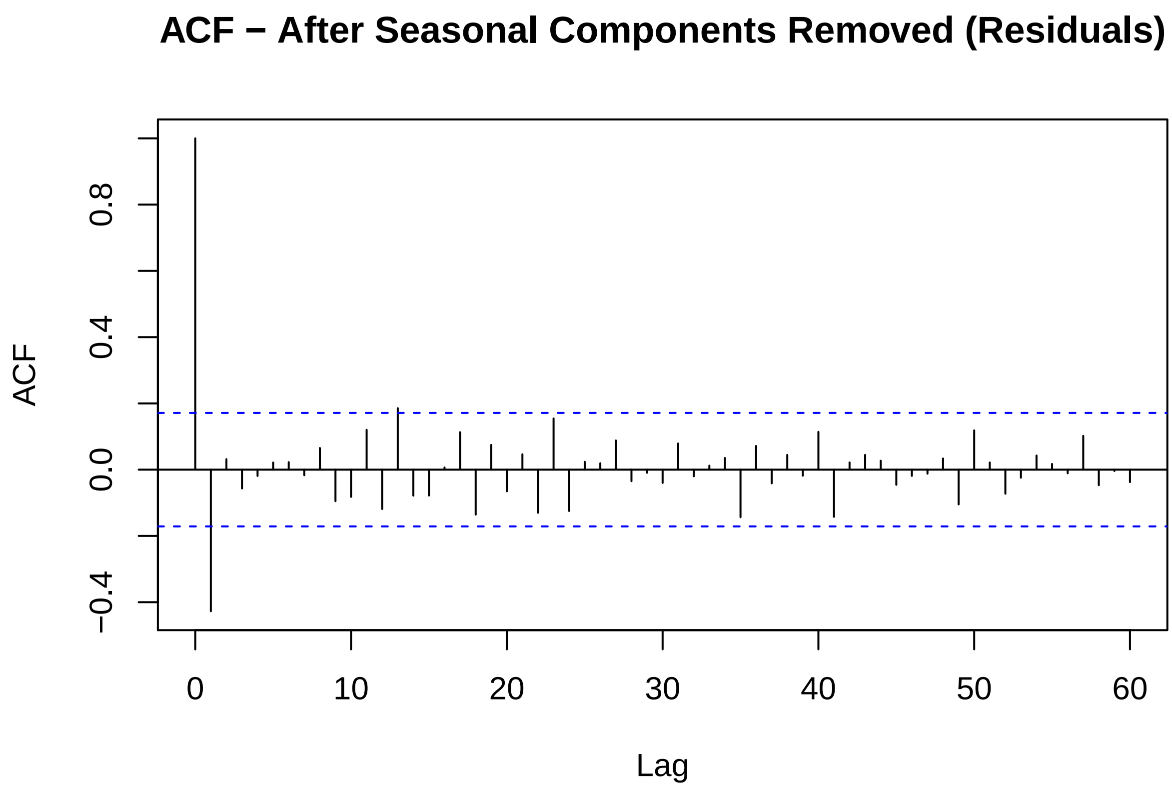



The residuals look mostly stationary because it is centered around mean 0.

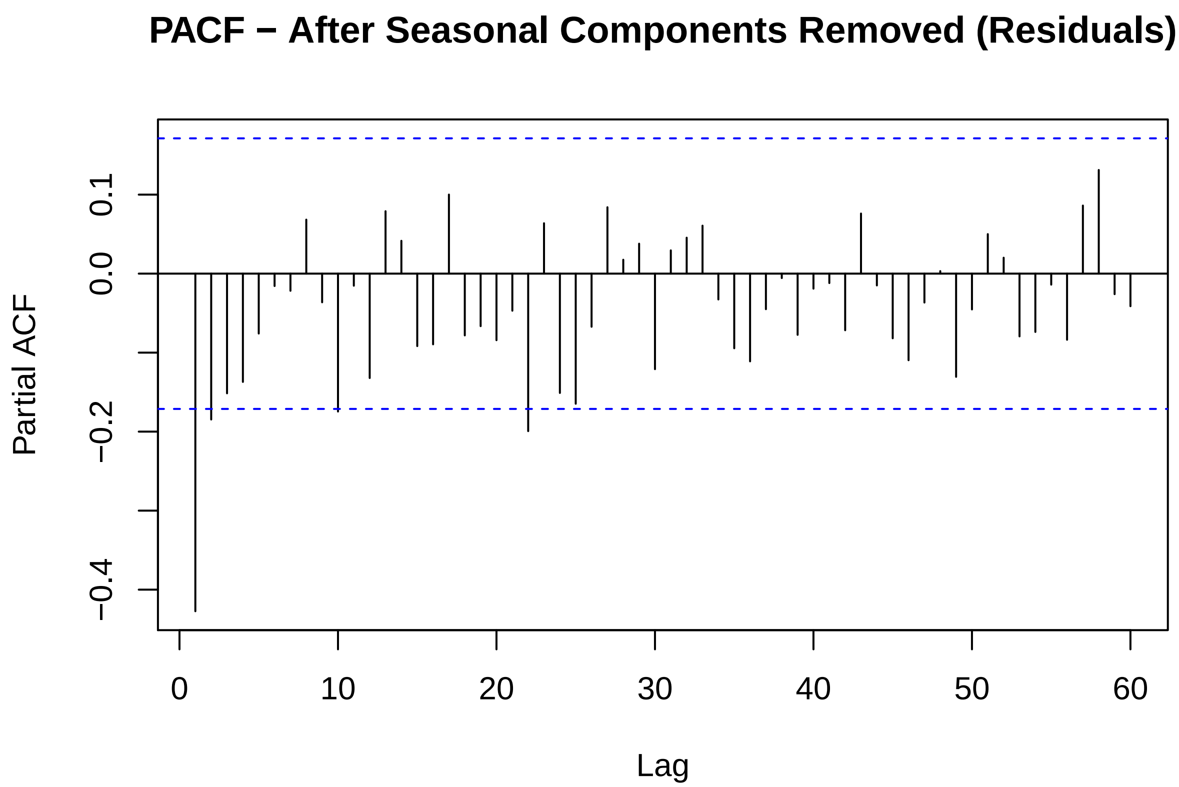
### Time Series Model

I will now take a look at the ACF and PACF of the residuals to decide the best fit model.

```
acf(resid(fit.back), lag.max = 60,
    main = "ACF - After Seasonal Components Removed (Residuals)")
```




```r
pacf(resid(fit.back), lag.max = 60,
     main = "PACF - After Seasonal Components Removed (Residuals)")
```




According to ACF and PACF we should fit a non-seasonal MA(1) model because the PACF trails off and the ACF drops off after lag 1.

I will now use the Hydmen-Khandaker (H-K) algorithm to determine the best fit model for our data. The H-K algorithm will give us the model that has the smallest AIC; therefore, it is more reliable than looking at the ACF and PACF (above).

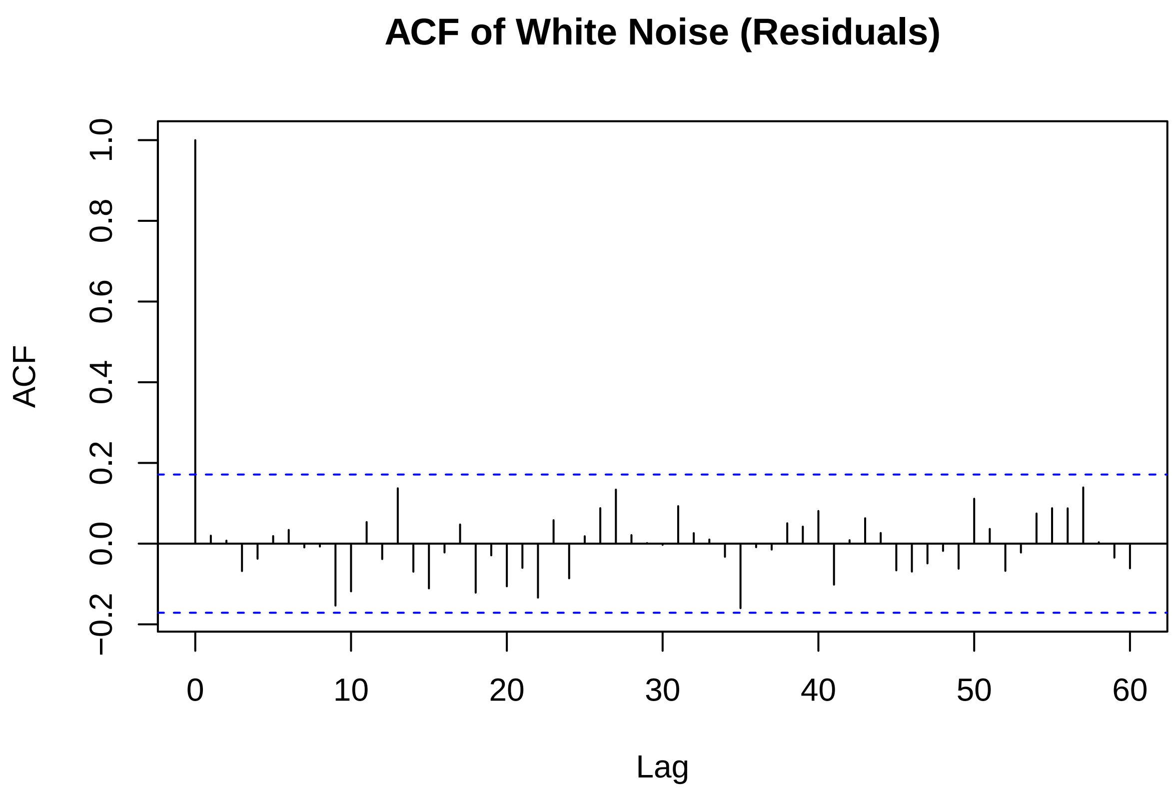
```
# Use H-K algorithm to determine best model
arma.fit = auto.arima(resid(fit.back), allowmean = F, step = F)
arma.fit
```

```
## Series: resid(fit.back)
## ARIMA(0,0,1) with zero mean
##
## Coefficients:
##           ma1
##       -0.5906
## s.e.   0.0768
##
## sigma^2 estimated as 0.4442:  log likelihood=-132.95
## AIC=269.9   AICc=269.99   BIC=275.65
```

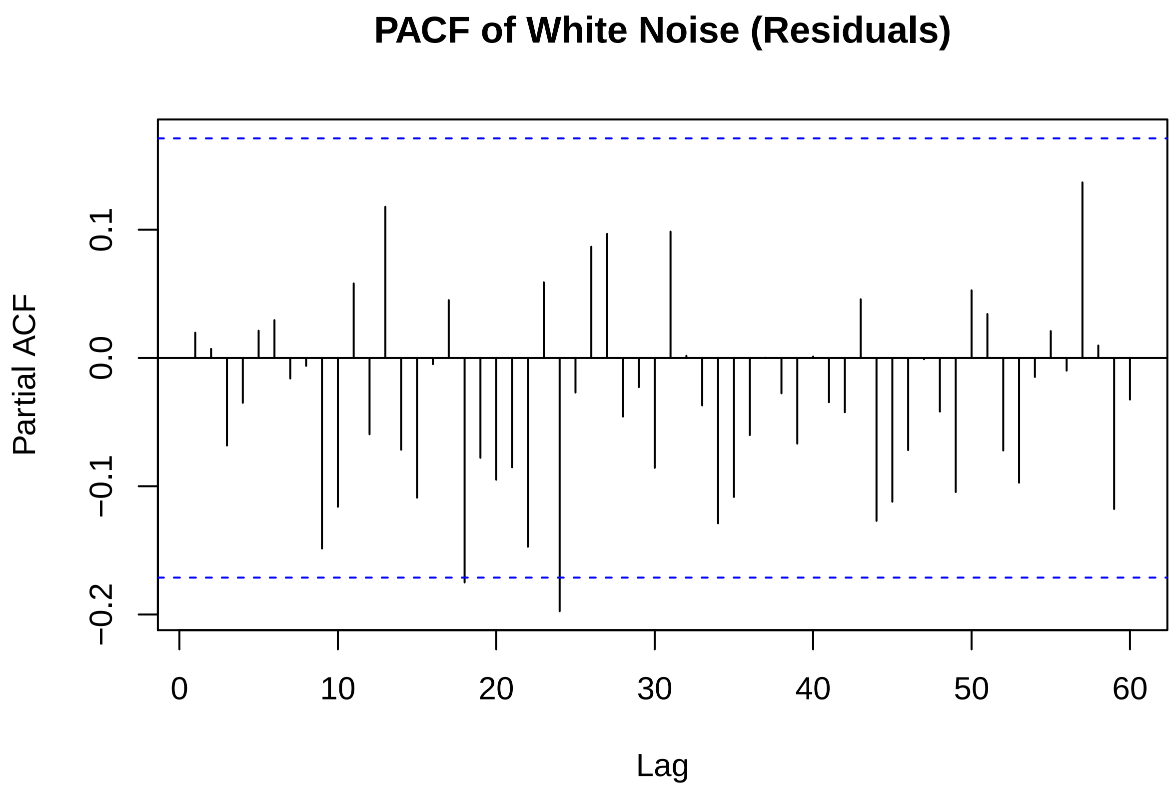
Based on auto.arima (H-K algorithm), our best model is non-seasonal MA(1), which is the same model I found based on looking at the ACF and PACF.

Next, I will check to see if the residuals are white noise. If the residuals are white noise, then the residuals are independent.

```r
# examine the residuals of the arma fit
wn = resid(arma.fit)

acf(wn, lag.max = 60, main = "ACF of White Noise (Residuals)")
```




```r
pacf(wn, lag.max = 60, main = "PACF of White Noise (Residuals)")
```




```r
# since there were some significant correlations in the plots,
# test to see if there is enough to reject independence
Box.test(wn, type="Ljung-Box", lag = min(2*d, floor(n/5)))
```

```
##
##  Box-Ljung test
##
## data:  wn
## X-squared = 22.236, df = 24, p-value = 0.5652
```

*ACF/PACF null and alternative hypothese test:*

$H_0$: residuals are independent (no dependence structure remaining)

$H_a$: residuals are dependent (dependence structure remaining)

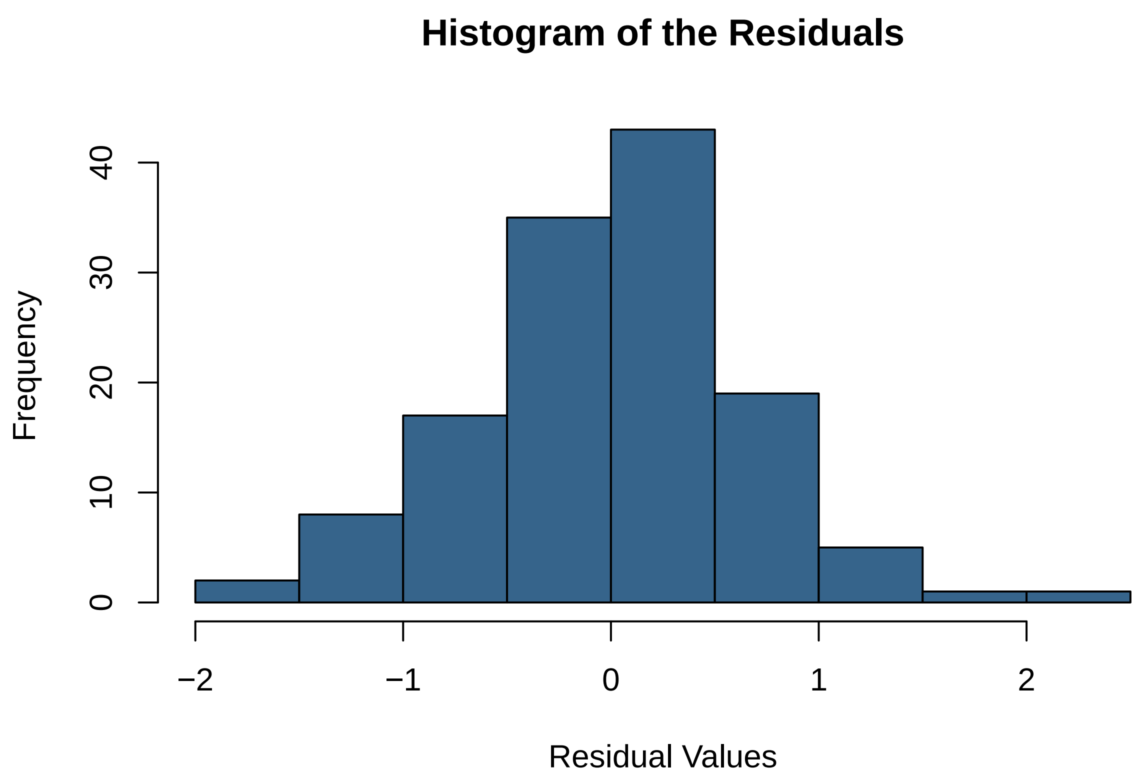
According to the ACF and PACF, there are still some dependence structure left in the residuals (some significant correlations in the plots). This means our residuals may not be white noise remaining only. However, aftering checking the Ljung-Box test, I get a p-value of 0.5652, which is greater than $\alpha = 0.05$; therefore, we fail to reject $H_0$. Since we fail to reject $H_0$, then the residuals are independent (no dependence structure remaining).

I may be getting a Type I error, where I am detecting an effect that is not present. In this case, I am seeing that the residuals have some dependence structure left, but the Ljung-Box tests tells me that not enough significant correlation to reject independence.

**Forecasting**

Now I will check to see if the residuals are normal. If the residuals are normal, then our forecast intervals will be reliable.

```
# look at the qqplot an histogram of the residuals to see if
# normality can be assumed
hist(wn, col = "steelblue4", main = "Histogram of the Residuals",
     xlab = "Residual Values")
```




```
# a small p-value reject the hypothesis that the residuals are normal
shapiro.test(wn)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  wn
## W = 0.98866, p-value = 0.3583
```

*Shapiro Walk Test null and alternative hypothese:*

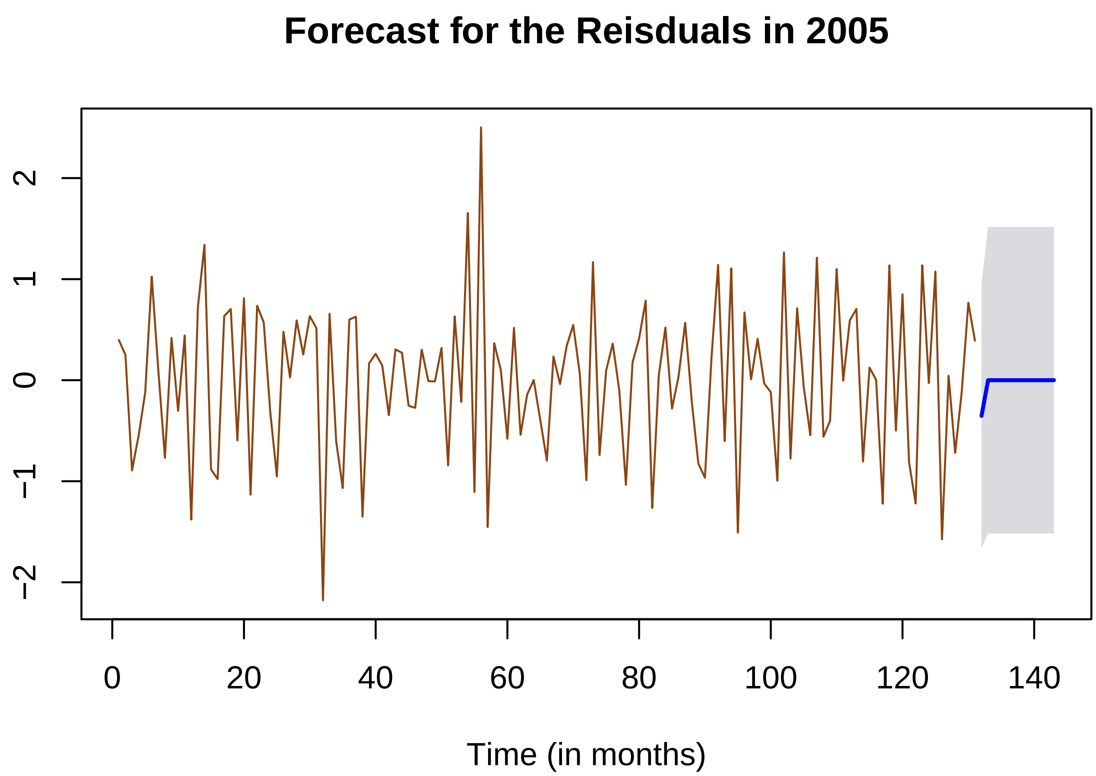
$H_0$: the residuals are normal

$H_a$: the residuals are not normal

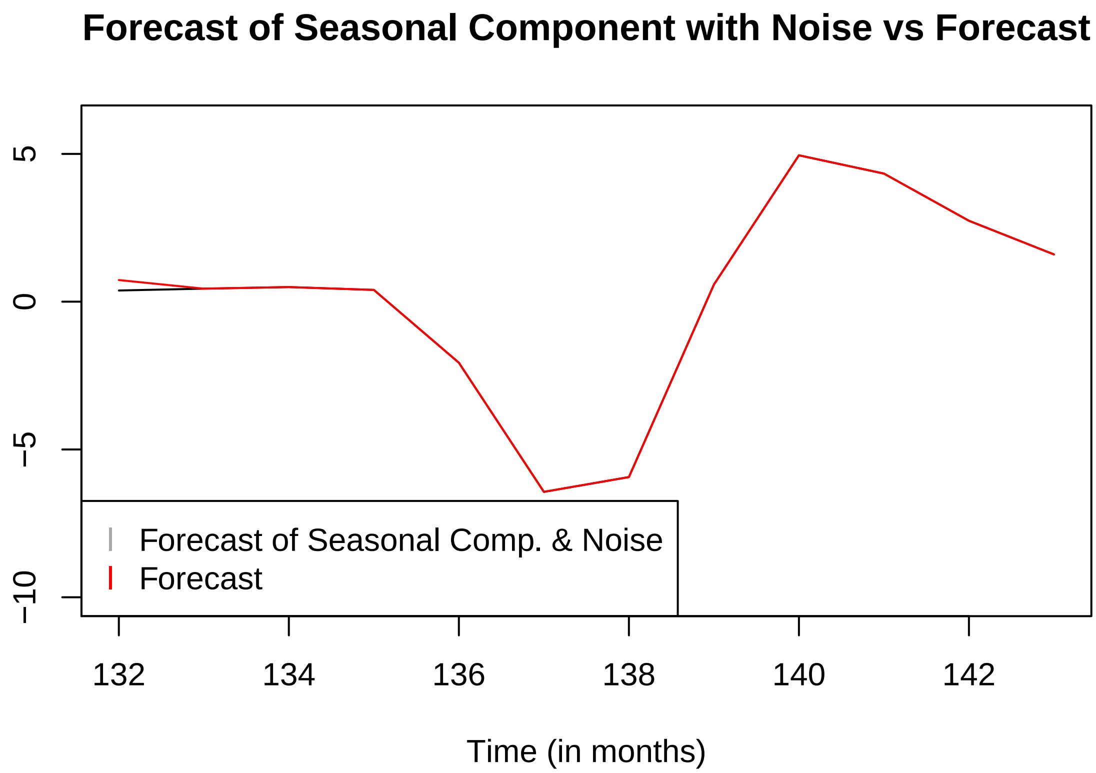
Since we have a p-value of 0.3583, it is greater than $\alpha = 0.05$, which means we fail to reject $H_0$. This means that our residuals are normal. This can also be seen in the histogram because hisogram shows us that residuals appear to be normally distributed. Since we are not rejecting normality here, it is safe to say that the forecasts intervals are reliable.

Next, I will forecast the next 12 months (2005) of noise at confidence level of 95%.

```
# forecast the next year of noise
noise.f = forecast(arma.fit, 12, level = 0.95)

plot(noise.f, col = "saddlebrown", main= "Forecast for the Reisduals in 2005",
     ylab = "", xlab = "Time (in months)")
```

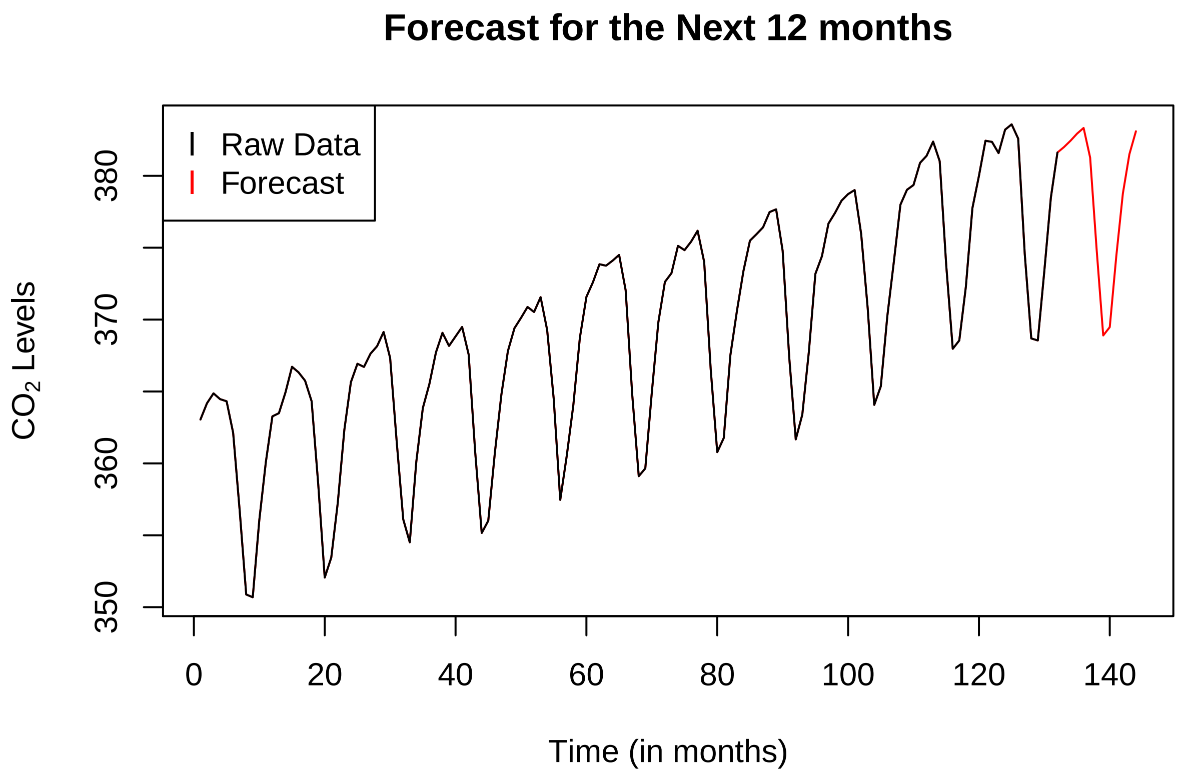



```
# forecast the seasonal component with the noise
# Since the seasonal component just repeats for every year
# the forecast is just the estimated seasonal components for 1,...,12
season.f = fitted(fit.back)[1:12]

plot(season.f+noise.f$mean, xlab = "Time (in months)", ylab = "",
     ylim = c(-10,6),
     main = "Forecast of Seasonal Component with Noise vs Forecast")
lines(132:143, season.f, col="red")
legend("bottomleft", c("Forecast of Seasonal Comp. & Noise", "Forecast"),
       pch = "l", col = c("darkgrey", "red"))
```

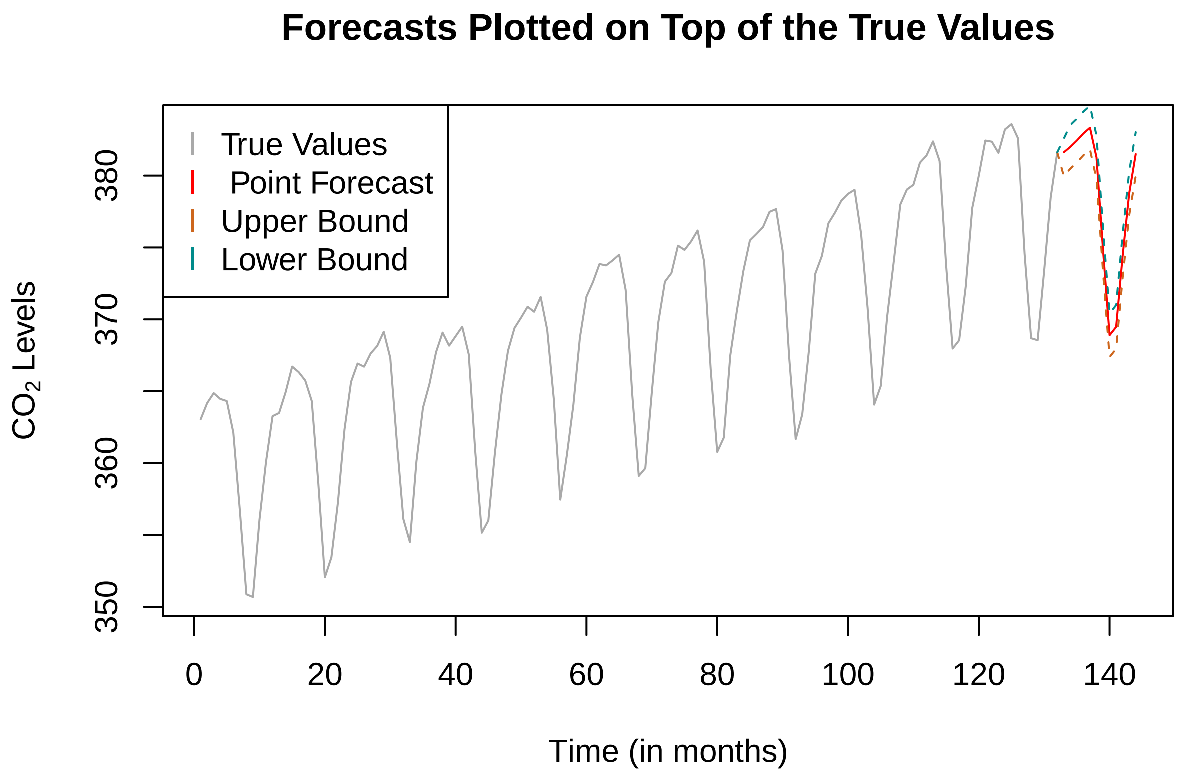



I will undo the differencing, that I did earlier to remove the trend, because I want to forecast the original data and not the data that was differenced.

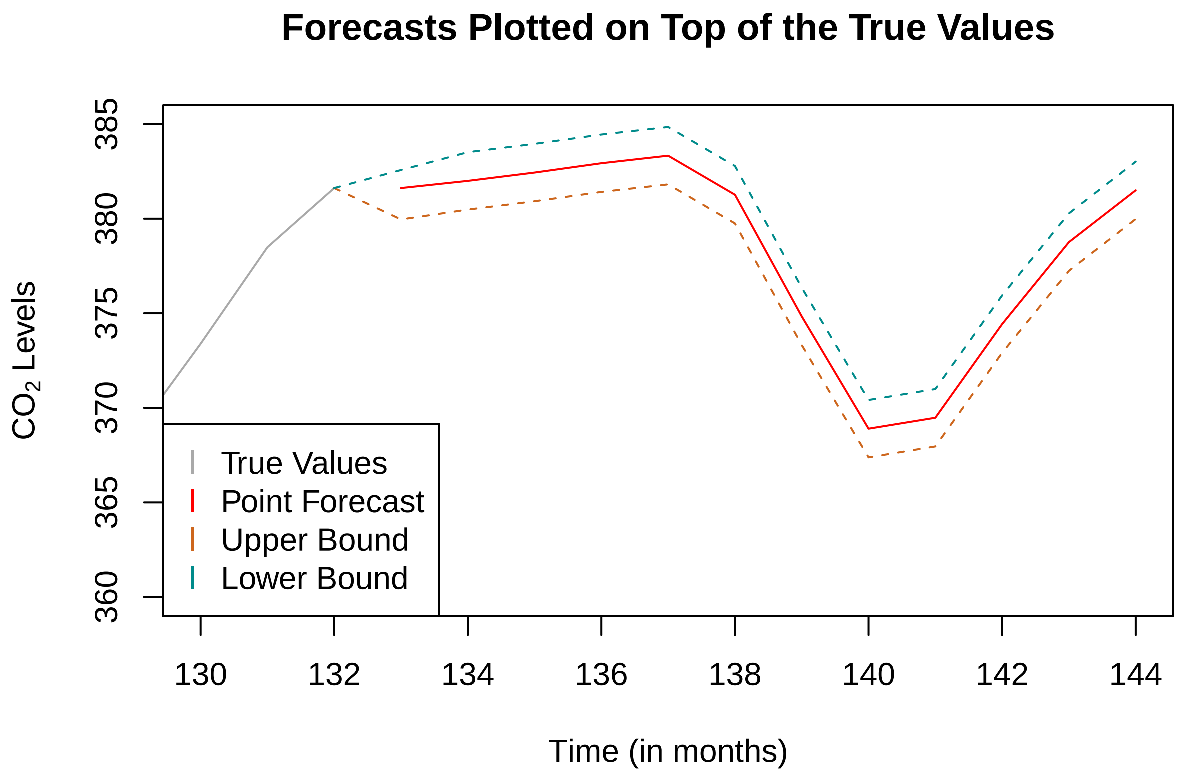
```
#now undo the difference
#first combine the differenced series with the forecast
y.fc = c(y,season.f+noise.f$mean)
fc.all = diffinv(y.fc, difference=1, xi =  x[1])
fc = fc.all[132:143]

ts.plot(fc.all, col="red", xlab = "Time (in months)",
        ylab = expression(""*CO[2]*" Levels"), main = "Forecast for the Next 12 months")
lines(x)
legend("topleft", c("Raw Data", "Forecast"), pch = "l", col = c("black", "red"))
```




```
#plot the forecasts on top of the true values
x = as.vector(co2)
plot(1:132, x, type="l", col="darkgrey", xlim = c(1, 144),
     main = "Forecasts Plotted on Top of the True Values",
     xlab = "Time (in months)", ylab = expression(""*CO[2]*" Levels"))
lines(133:144, fc, col="red")
lines(132:144, c(co2[132], fc + noise.f$lower), col = "chocolate3", lty = 2)
lines(132:144, c(co2[132], fc + noise.f$upper), col = "cyan4", lty = 2)
legend("topleft", c("True Values", " Point Forecast", "Upper Bound", "Lower Bound"),
       pch = "l", col = c("darkgrey", "red", "chocolate3", "cyan4"))
```




```
#note that we are translated over one due to taking one difference

#closer view
x = as.vector(co2)
plot(1:132, x, type="l", col="darkgrey", xlim = c(130,144), ylim = c(360, 385),
     main = "Forecasts Plotted on Top of the True Values",
     xlab = "Time (in months)", ylab = expression(""*CO[2]*" Levels"))
lines(133:144, fc, col="red")
lines(132:144, c(co2[132], fc + noise.f$lower), col = "chocolate3", lty = 2)
lines(132:144, c(co2[132], fc + noise.f$upper), col = "cyan4", lty = 2)
legend("bottomleft", c("True Values", "Point Forecast", "Upper Bound", "Lower Bound"),
       pch = "l", col = c("darkgrey", "red", "chocolate3", "cyan4"))
```




```r
year2005 = rep(2005, 12)
intervalForecast = data.frame(year2005, fc + noise.f$lower, fc, fc + noise.f$upper)
colnames(intervalForecast) = c("Year", "Lower Bound",
                               "Point Forecast", "Upper Bound")
rownames(intervalForecast) = c("January", "Febuary", "March", "April", "May",
                               "June", "July", "August", "September",
                              "October", "November", "December")
```

The interval forcast for the $CO_2$ levels for every month of 2005 at confidence level 95% are:

```r
intervalForecast
```

```
##           Year Lower Bound Point Forecast Upper Bound
## January   2005    379.9603       381.6200    382.5729
## Febuary   2005    380.4814       381.9986    383.5157
## March     2005    380.9222       382.4393    383.9565
## April     2005    381.4156       382.9327    384.4499
## May       2005    381.8140       383.3311    384.8483
## June      2005    379.7491       381.2662    382.7834
## July      2005    373.3147       374.8319    376.3490
## August    2005    367.3804       368.8976    370.4147
## September 2005    367.9591       369.4763    370.9934
## October   2005    372.9118       374.4290    375.9461
## November  2005    377.2456       378.7627    380.2799
## December  2005    379.9836       381.5007    383.0179
```

For each interval forecast, I am 95% confident that the forecast intervals of each month fall between the month's respective lower and upper bounds.