Forecasting Carbon Dioxide Levels in Hawaii

Jenny Pon

6/5/2020

Executive Summary

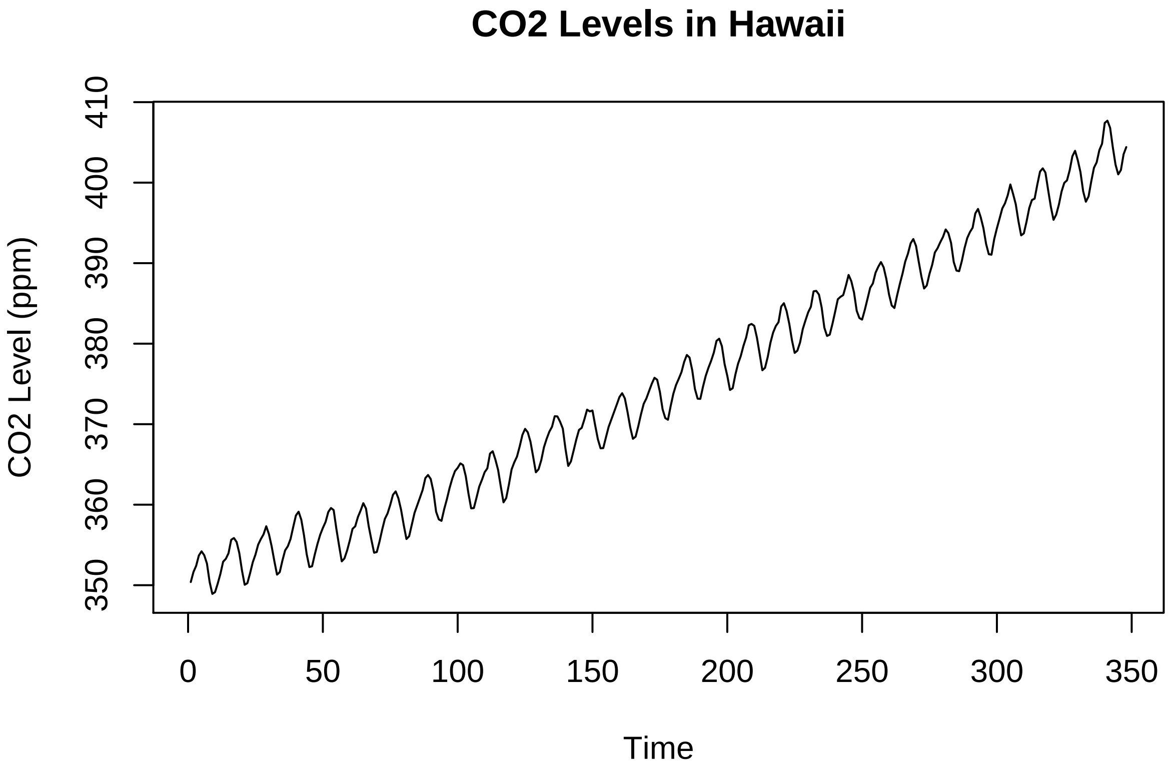
The dataset I am using is monthly historical values of CO_2 measured in Hawaii. Box-Jenkins methodology was used to build a model appropriate for this data through model formulation, estimation, evaluation, and forecasting. Two potential models were considered and compared. The selected model was then used to forecast future CO_2 levels. The forecasted values produced from the selected model were very accurate, as they aligned closely with our test data, and fell within the prediction interval.

Introduction

This time series data used in this report came from the CO2Hawaii dataset from the Stat2Data package in R. This dataset of historical values of CO_2 is measured in particles per million (ppm) from 1980 to 2017. The data from 1980 to 2016 was used to build a model, while the 2017 data was later used to test the accuracy of the selected model's forecasting ability. This data is important because powerful greenhouse gases like CO_2 are the most prominent contributers to climate change due to the greenhouse effect. I chose this dataset because I have an interest in environmental issues, and climate change is the defining environmental issue of our time.

In this project, I intend to forecast CO_2 values in Hawaii. Using Box-Jenkins methodology, two potential SARIMA models were identified and put through diagnostic checking. In diagnostic checking, neither model passed the Shapiro-Wilk normality test, but both passed all the other components of the procedure. Both models also had similar AICc values, with Model 1 having the lower value. Model 1 was ultimately chosen based on the AICc and the principle of parsimony. RStudio was the software used for statistical computing.

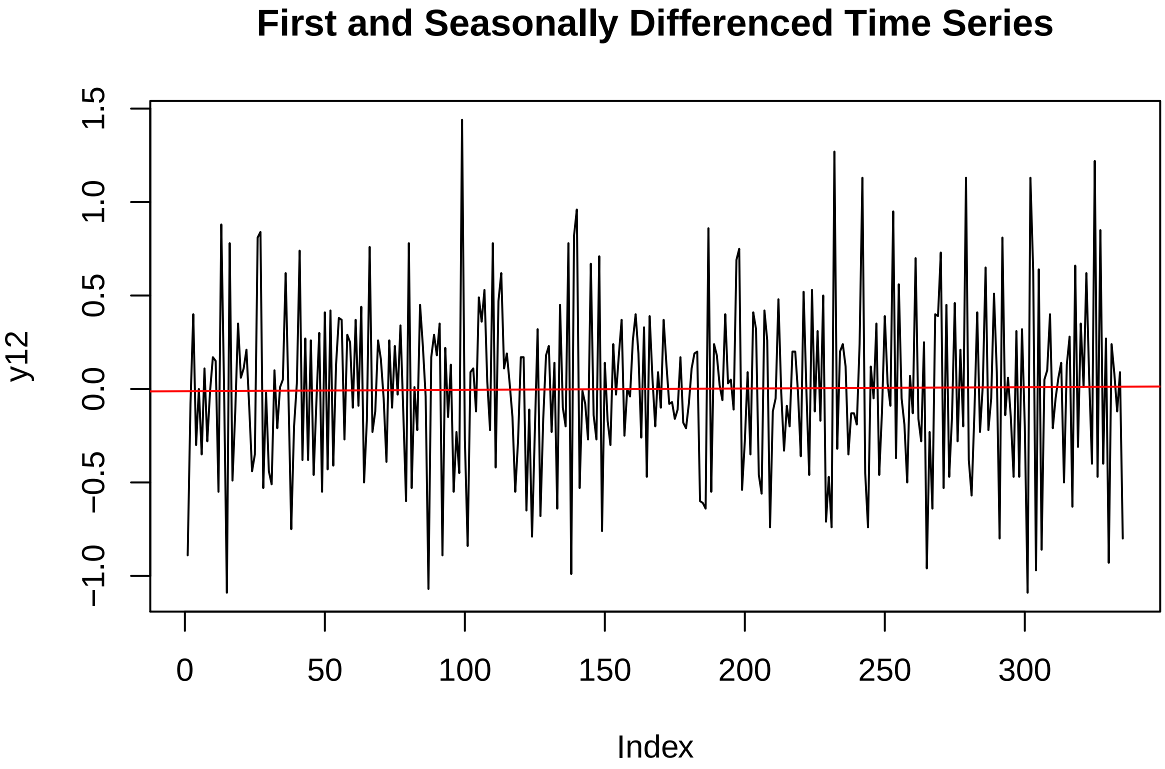
Section 1: Plot of Time Series



The plot of the CO_2 levels in Hawaii shows that the training data, X_t, is non-stationary because there is a trend and seasonality. There is a positive linear trend, as the data values steadily increase in an approximately straight line with time. There is seasonality because the data drops in value and then rises, at regular intervals. The data has a constant variance and contains no apparent sharp changes in behavior.

Section 2: Differencing

In order to make the data stationary, trend and seasonality must be eliminated. To eliminate trend, we difference at lag 1. Then, to eliminate seasonality, we difference at lag 12 because the data is monthly.



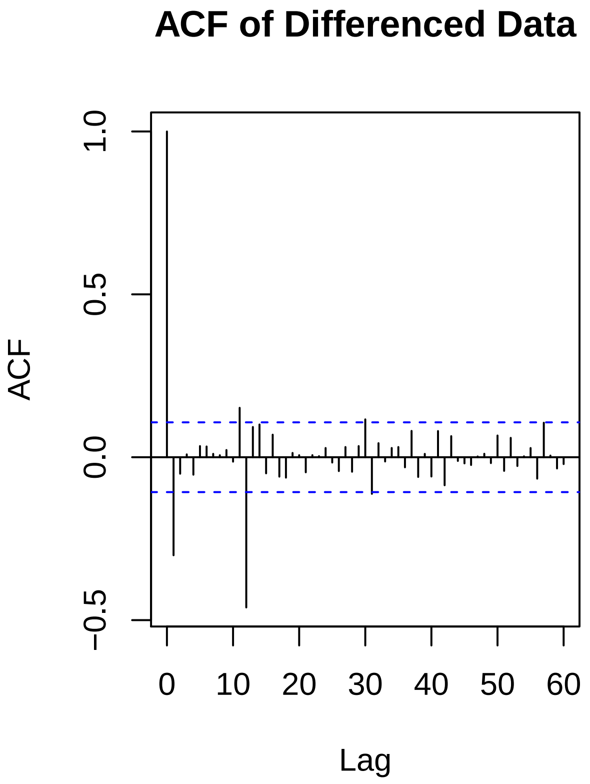
Variance value after each differencing

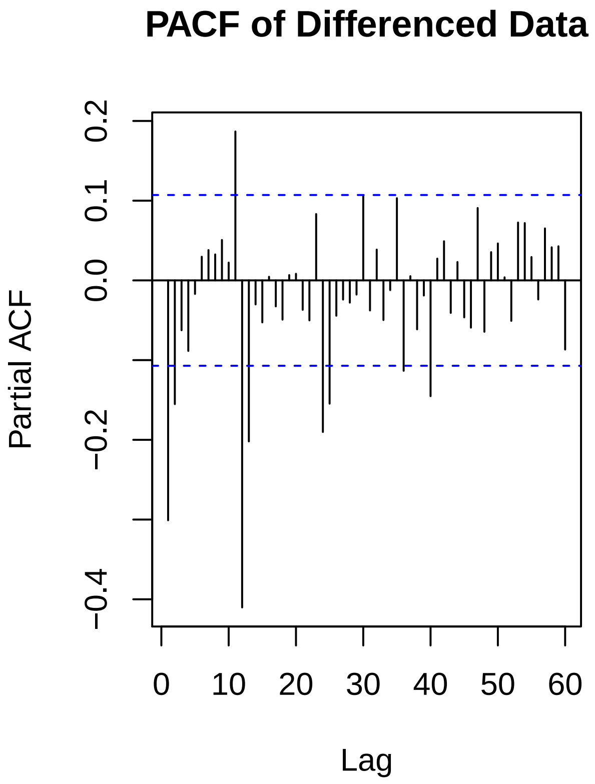
```
## [1] 1.62961
```

```
## [1] 0.200614
```

From the plot of the differenced data, we can see that trend was successfully eliminated, as the regression line is almost completely horizontal. We can also see that eliminating seasonality was necessary, as the variance decreased from 1.62961 to 0.200614.

To check that the data is indeed stationary after differencing, the ACF and PACF plots of the differenced data are evaluated.





Since we detect no trend or seasonality in the ACF and PACF plots, we conclude that the data is stationary.

Section 3: Preliminary Identification of Models

Next, the ACF and PACF of the differenced data were analyzed for model identification. Referring back to the plots displayed in Section 2, the seasonal and nonseasonal terms for a $SARIMA(p, d, q)x(P, D, Q)_s$ model were determined.

For the seasonal part of the model, Q = 1 is a good choice because the ACF shows a large peak at 1s. P is 0 because the PACF tails off at lags that are multiples of 12. D = 1 because there was one seasonal differencing taken. s = 12 because the data has monthly seasonality.

For the non-seasonal part of the model, q = 1 is a good choice because the the ACF cuts off at lag 1. p = 0, 1, or 2 because the PACF either decays or cuts off at lag 2. d = 1 because one differencing was applied to remove trend.

Thus, we will examine two $SARIMA(p, d, q)x(P, D, Q)_s$ model candidates:

1. $SARIMA(0, 1, 1)x(0, 1, 1)_{12}$
2. $SARIMA(1, 1, 1)x(0, 1, 1)_{12}$

Section 4: Coefficient Estimation and Diagnostic Checking

Coefficient Estimation for Model 1

The formula for a SARIMA model is:

$$\phi(B)\Phi(B^s)(1 - B)^d(1 - B^s)^D X_t = \theta(B)\Theta(B^s)Z_t$$

To estimate the coefficients for Model 1, we use the arima() function in R by the method of maximum likelihood estimation.

```
##
## Call:
## arima(x = co2.train, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##           ma1     sma1
##       -0.3779  -0.8862
## s.e.   0.0574   0.0412
##
## sigma^2 estimated as 0.1034:  log likelihood = -104.55,  aic = 215.1
```

After using the arima() function, we see that the standard errors are such that all the coefficients are significant. Using these coefficients and the SARIMA formula, Model 1 is (1-B)(1-B^{12})X_t = (1-0.3779B)(1-0.8862B^{12})Z_t

Coefficient Estimation for Model 2

We now use the arima() function on Model 2 as well.

```
##
## Call:
## arima(x = co2.train, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##          ar1      ma1     sma1
##       0.2013  -0.5489  -0.8881
## s.e.  0.1225   0.1020   0.0401
##
## sigma^2 estimated as 0.1026:  log likelihood = -103.37,  aic = 214.74
```

From the output, we see that the standard errors are such that all the coefficients are significant. With these coefficients and the SARIMA formula, we can write the formula for Model 2: (1-0.2013B)(1-B)(1-B^{12})X_t = (1-0.5489B)(1-0.8881B^{12})Z_t

Check Stationarity and Invertibility of Model 1 and 2

Before we move onto diagonostic checking, we need to check that both models are stationary and invertible.

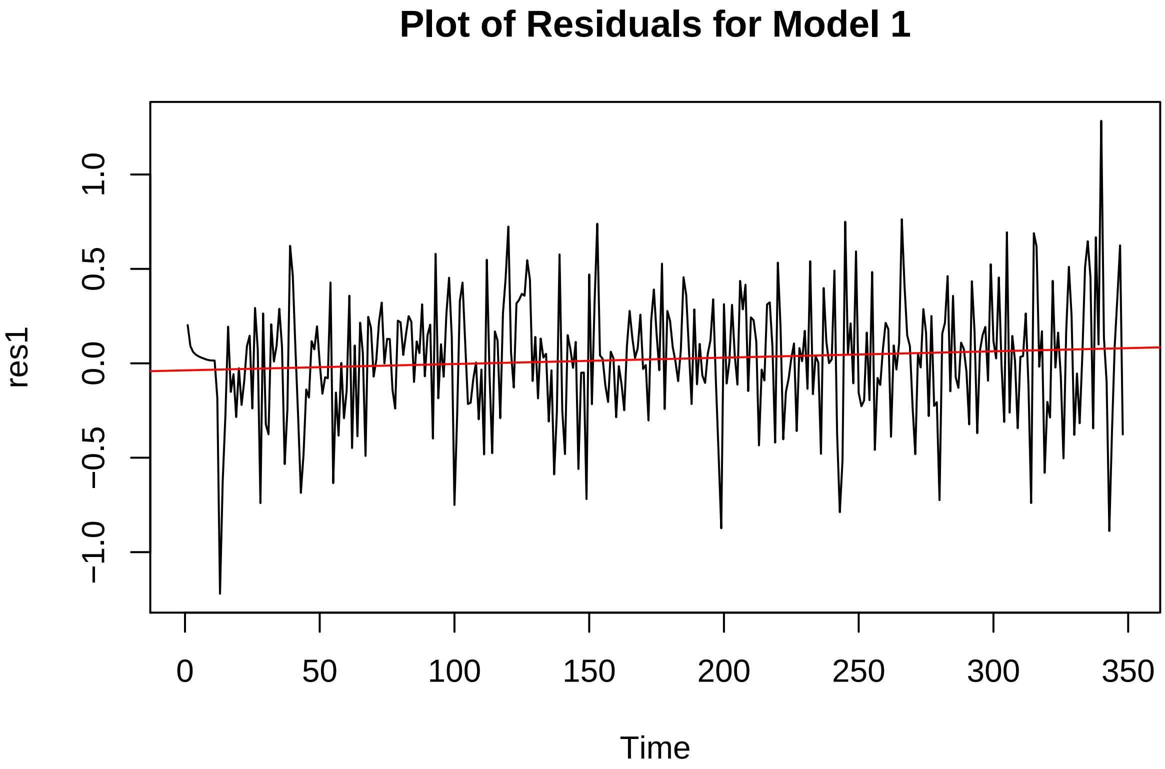
Our two models are:

1. Model 1: (1-B)(1-B^{12})X_t = (1-0.3779B)(1-0.8862B^{12})Z_t
2. Model 2: (1-0.2013B)(1-B)(1-B^{12})X_t = (1-0.5489B)(1-0.8881B^{12})Z_t

Both models are invertible because $|\theta_1| < 1$, and $|\Theta_1| < 1$ in both. Model 1 is stationary because it is a pure MA model. Model 2 is also stationary because $|\phi_1| < 1$.

Diagnostic Checking for Model 1

The first step to diagnostic checking is plotting the residuals. For a good fit, the residuals should resemble white noise.



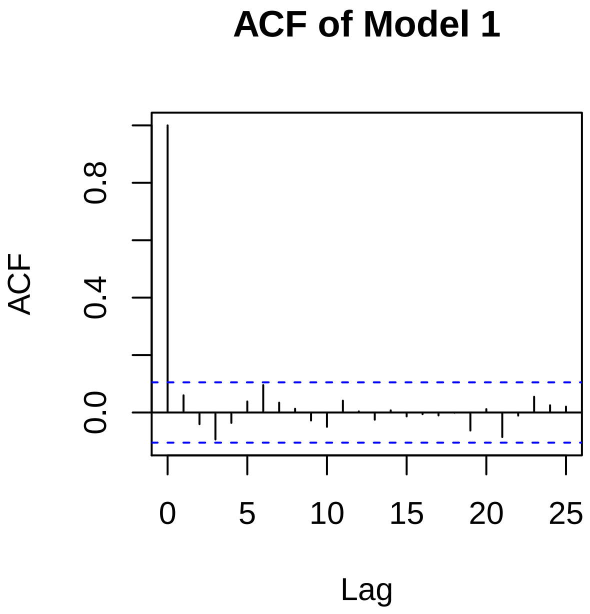
Sample mean of Model 1 Residuals

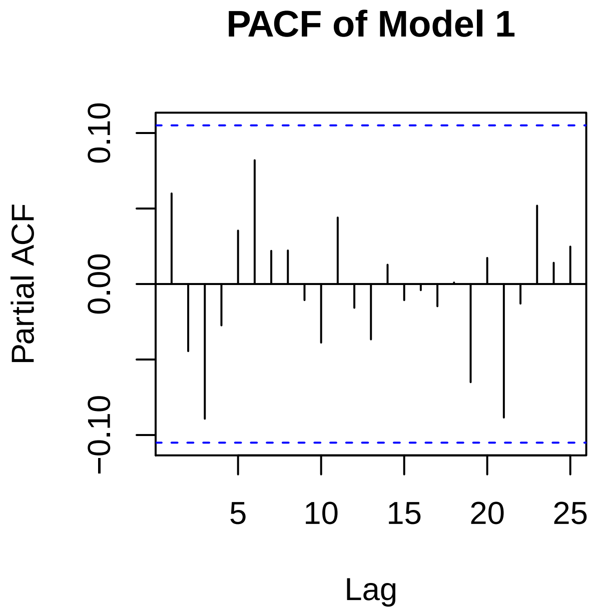
```
mean(res1)
```

```
## [1] 0.02137328
```

The plot of the residuals looks approximately like white noise. This is further demonstrated by the fact that the regression line is very close to a horizontal line, and the sample mean is approximately 0.

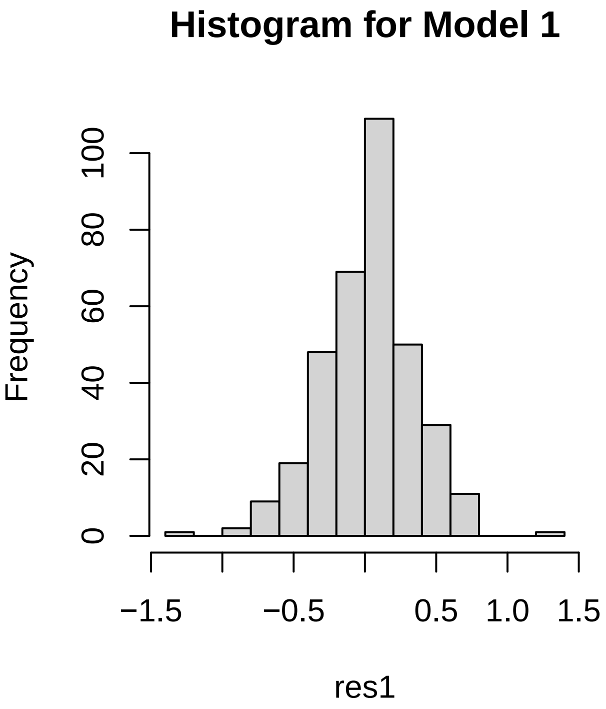
Next, to further verify that the residuals for Model 1 resemble white noise, we check the ACF and PACF plots.

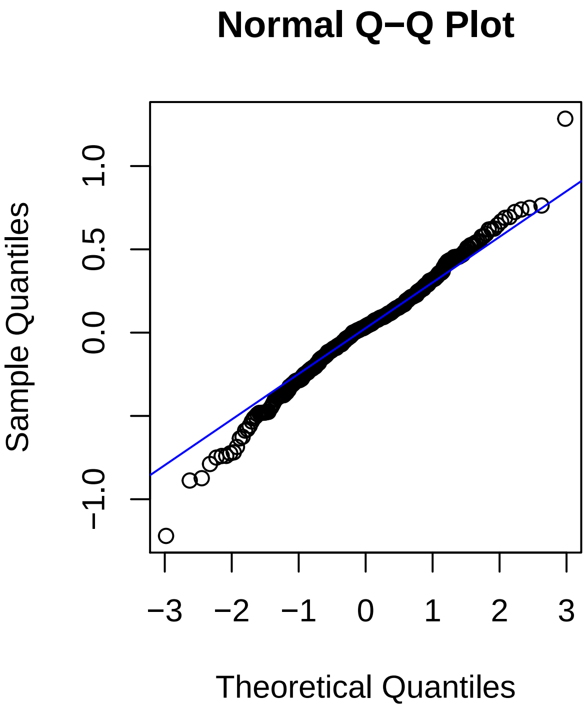




The ACF and PACF plot of the residuals look like white noise because values at all lags are within the confidence intervals.

Next, to check the normality assumption, we plot a histogram of the residuals, and plot a Normal Q-Q Plot.





The histogram of the residuals looks Gaussian, and the line generated in the Normal Q-Q Plot is approximately straight, with only slight deviation, indicating that the residuals are normally distributed.

To further check the normality assumption, we next conduct the Shapiro-Wilk normality test.

```
##
##  Shapiro-Wilk normality test
##
## data:  res1
## W = 0.9891, p-value = 0.01064
```

Our residuals do not pass the Shapiro-Wilk normality test, as the p-value is less than our α value, 0.05. We therefore conclude that heavy-tailed model would work better.

Next, we will conduct the Box-Pierce, Ljung-Box, and McLeod Li tests to test our white noise hypothesis.

```
##
##  Box-Pierce test
##
## data:  res1
## X-squared = 11.621, df = 14, p-value = 0.6367
```

```
##
##  Box-Ljung test
##
## data:  res1
## X-squared = 11.868, df = 14, p-value = 0.6169
```

```
##
##  Box-Ljung test
##
```

```
## data:  res1^2
## X-squared = 9.8903, df = 18, p-value = 0.9354
```

The data passes all three tests because the p-values are larger than 0.05. We fail to reject the white noise hypothesis.

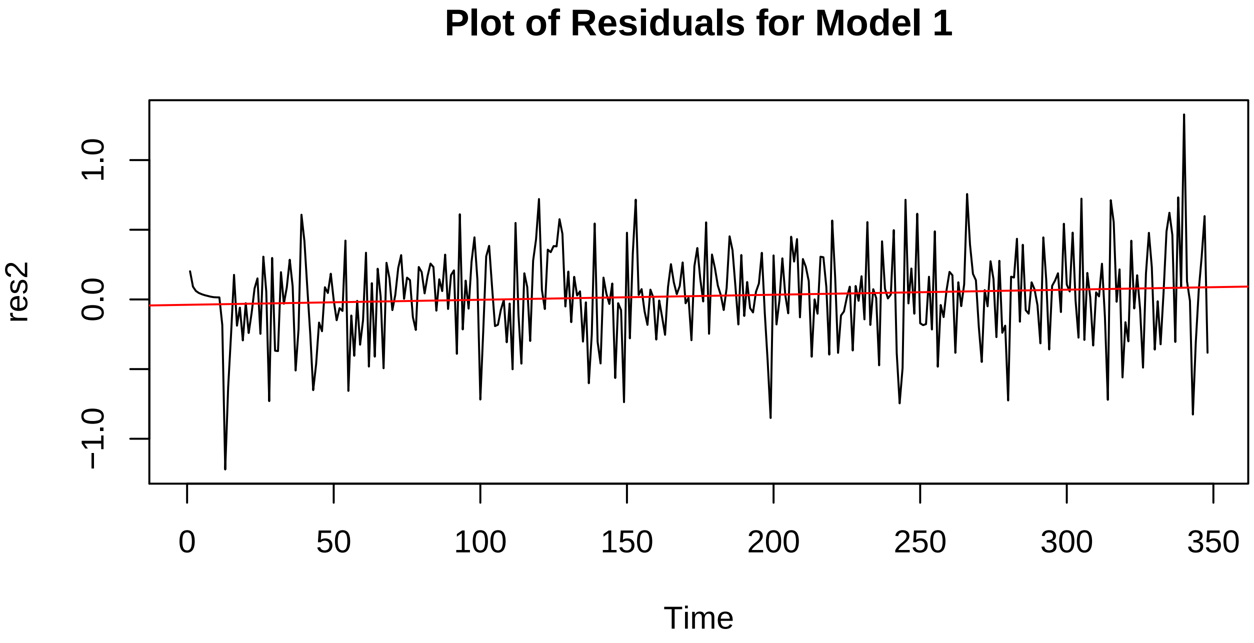
The last step of diagnostic checking is using Yule-Walker estimation to determine whether our residuals fit into an AR(0) model.

```
##
## Call:
## ar(x = res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.1039
```

The Yule-Walker method selected order 0, so the residuals were fitted to AR(0), or white noise. With this, Model 1 passed diagnostic checking.

Diagnostic Checking for Model 2

We repeat the diagnostic procedures for Model 2. First, we plot the residuals to see if they resemble Gaussian white noise, which would indicate that model 2 is a good fit.



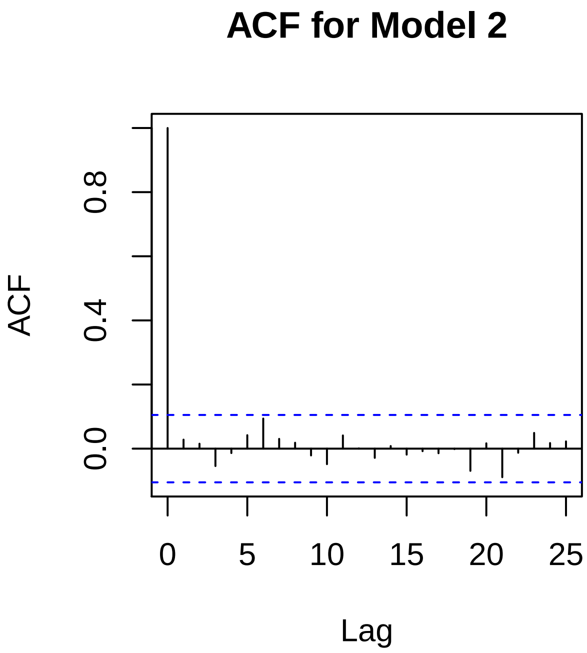
Sample Mean of Model 2 Residuals

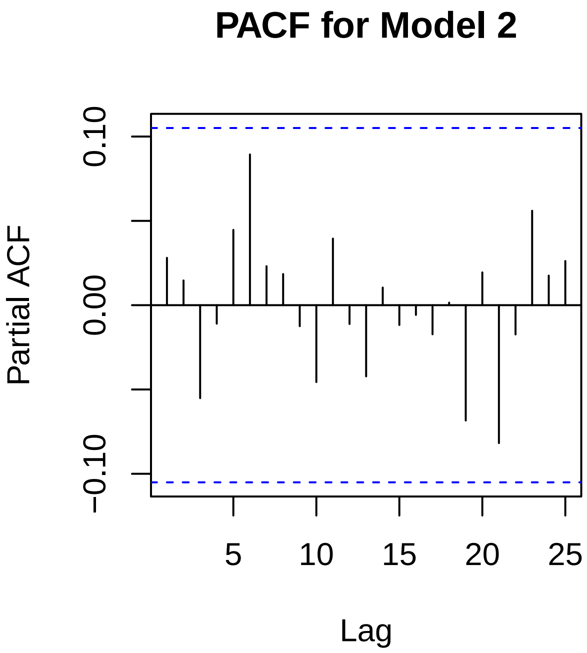
```
mean(res2)
```

```
## [1] 0.02438623
```

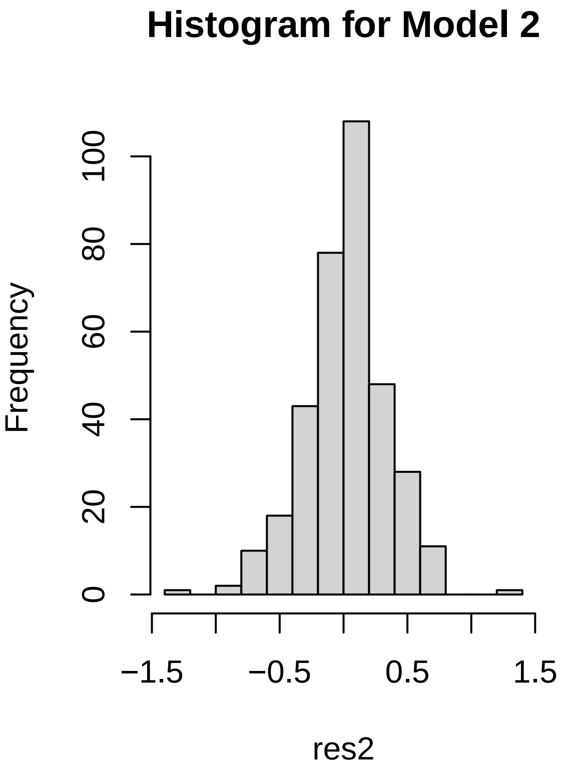
The plot of the residuals looks approximately like white noise. The regression line has a slight trend, but it is not very noticable, and the sample mean of the residuals is approximately zero.

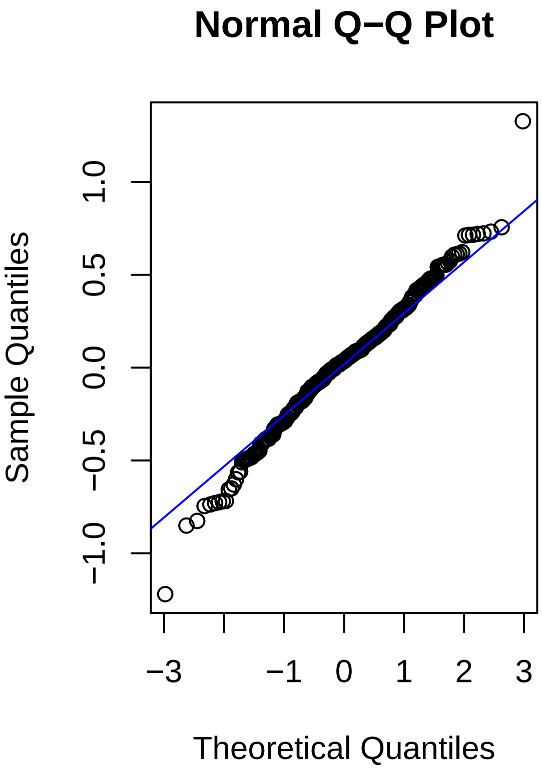
Next, we plot the ACF and PACF of the residuals to check if they look like white noise.





The ACF and PACF plot of the residuals look like white noise because values at all lags are within the confidence intervals. Next, to check the normality assumption, we plot a histogram of the residuals and a Normal Q-Q Plot.





The histogram of the residuals looks Gaussian, and the line Normal Q-Q Plot is approximately straight, with slight deviation, indicating that the residuals are normally distributed.

To further check the normality assumption of our Model 2 residuals, we next conduct the Shapiro-Wilk normality test.

```
##
##  Shapiro-Wilk normality test
##
## data:  res2
## W = 0.98883, p-value = 0.009083
```

Our residuals do not pass the Shapiro-Wilk normality test, as the p-value is less than our α value, 0.05. We therefore conclude that heavy-tailed model would work better.

Next, we will test our white noise hypothesis of our residuals using the Box-Pierce, Ljung-Box, and McLeod Li tests.

```
##
##  Box-Pierce test
##
## data:  res2
## X-squared = 7.6639, df = 14, p-value = 0.9061
```

```
##
##  Box-Ljung test
##
## data:  res2
## X-squared = 7.862, df = 14, p-value = 0.8964
```

```
##
##  Box-Ljung test
##
## data:  res2^2
## X-squared = 8.3482, df = 18, p-value = 0.973
```

All three tests are passed because the p-values are all larger than 0.05. Therefore, we fail to reject the white noise hypothesis.

Lastly, we use Yule-Walker estimation to determine whether our residuals fit into an AR(0) model.

```
##
## Call:
## ar(x = res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.103
```

The Yule-Walker method selected order 0, so the residuals were fitted to AR(0), or white noise. With this, Model 2 also passed all the tests in diagnostic checking except the Shapiro-Wilk normality test.

AICc for Model 1 and Model 2

Next, we use the AICc() function to calculate the AICc (Akaike Information Criterion, Corrected for Bias) to compare the fit of Model 1 and 2.

AICc for Model 1

```
## [1] 215.1346
```

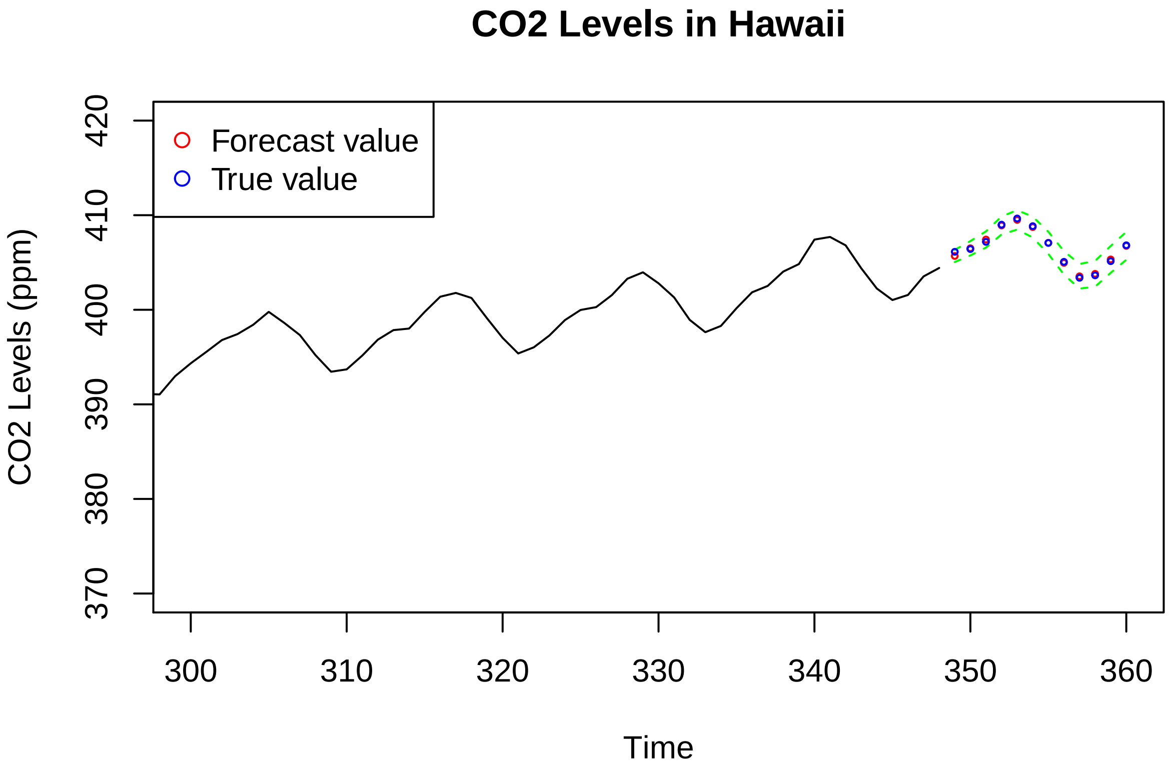
AICc for Model 2

```
## [1] 216.8515
```

The AICc for Model 1 (215.1346) is lower than the AICc for Model 2 (216.8515). Also taking into consideration the fact that Model 1 has fewer coefficients than Model 2 does, we choose Model 1, (1-B)(1-B^{12})X_t = (1-0.3779B)(1-0.8862B^{12})Z_t, as the final model based on the principle of parsimony and the AICc.

Forecasting

Next, we use our final model, (1-B)(1-B^{12})X_t = (1-0.3779B)(1-0.8862B^{12})Z_t, to forecast the next 12 months. We graph our prediction interval (green lines) and compare our forecasted values (red dots) to our test data (blue dots).



From the plot of our forecasted data, we can see that the 12 forecasted values align very closely with the true test values, and they lie within the prediction intervals. This indicates that our model is very accurate for forecasting data.

Conclusion

The goal of this project was to use the historical values of CO_2 in Hawaii to build a model, via Box-Jenkins methodology, to forecast future CO_2 levels. This goal was achieved. We evaluated two potential models. The final model chosen was (1-B)(1-B^{12})X_t = (1-0.3779B)(1-0.8862B^{12})Z_t. Using this model, the forecasted data was very accurate.

References

Brockwell,P. and Davis, R. 1996, *Introduction to Time Series and Forecasting.* 2nd Edition; New York;

Springer-Verlag New York, Inc.

Appendix

```
# create time series
co2 <- ts(CO2Hawaii[,3], start=c(1988,1),frequency=12)

# training data
co2.train <- co2[c(1:348)]

# test data
co2.test <- co2[c(349:360)]

# plot time series
plot(co2.train, type="l",main="CO2 Levels in Hawaii", xlab="Time")

# difference at lag 1 to remove trend
y1 <- diff(co2.train,1)

# difference at lag 12 to remove seasonality
y12 <- diff(y1,12)

# plot differenced data
plot(y12, main="First and Seasonally Differenced Time Series", type="l")
fit <- lm(y12 ~ as.numeric(1:length(y12)))
abline(fit, col="red")

# compare variances
var(y1);var(y12)

# plot acf and pacf of differenced data
par(mfrow = c(1,2))
acf(y12, main="ACF of Differenced Data",lag.max=60)
pacf(y12, main="PACF of Differenced Data", lag.max=60)

# estimate model 1 coefficients
arima(co2.train, order=c(0,1,1), seasonal=list(order=c(0,1,1), period=12),
      method="ML")

# estimate model 2 coefficients
arima(co2.train, order=c(1,1,1), seasonal=list(order=c(0,1,1), period=12),
      method="ML")

# plot model 1 residuals
fit1 <- arima(co2.train, order=c(0,1,1), seasonal=list(order=c(0,1,1),
                                                       period=12), method="ML")
res1 <- residuals(fit1)
plot(res1,main="Plot of Residuals for Model 1")
fitres <- lm(res1 ~ as.numeric(1:length(res1)))
abline(fitres, col="red")

# sample mean of model 1 residuals
mean(res1)
```

```r
# acf and pacf of model 1 residuals
acf(residuals(fit1),main="Autocorrelation")
pacf(residuals(fit1), main="Partial Autocorrelation")

# Histogram of model 1 residuals
hist(res1,main="Histogram for Model 1")

# Q-Q Plot
qqnorm(res1)
qqline(res1,col="blue")

# Shapiro-Wilk test on model 1 residuals
shapiro.test(res1)

# Box-Pierce test on model 1 residuals
Box.test(res1, lag=18, type=c("Box-Pierce"),fitdf=4)
# Ljung-Box
Box.test(res1, lag=18, type=c("Ljung-Box"), fitdf=4)
# McLeod Li
Box.test(res1^2, lag=18, type=c("Ljung-Box"), fitdf=0)

# Yule-Walker Estimation for Model 1
ar(res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# plot the residuals for model 2
fit2 <- arima(co2.train, order=c(1,1,1), seasonal=list(order=c(0,1,1),
              period=12), method="ML")
res2 <- residuals(fit2)
fitres <- lm(res2 ~ as.numeric(1:length(res2)))
abline(fitres, col="red")

plot(res2,main="Plot of Model 2 Residuals")

# Sample mean of the residuals
mean(res2)

# acf and pacf of model 2 residuals
acf(residuals(fit2),main="ACF of Model 2 Residuals")
pacf(residuals(fit2), main="PACF of Model 2 Residuals")

# histogram of model 2 residuals
hist(res2,main="Histogram for Model 2")

# Q-Q plot of model 2 residuals
qqnorm(res2)
qqline(res2,col="blue")

# Shapiro-Wilk test on model 2 residuals
shapiro.test(res2)

# Box-Pierce test on model 2 residuals
Box.test(res2, lag=18, type=c("Box-Pierce"),fitdf=4)
# Ljung-Box test
```

```
Box.test(res2, lag=18, type=c("Ljung-Box"), fitdf=4)
# McLeod Li test
Box.test(res2^2, lag=18, type=c("Ljung-Box"), fitdf=0)

# Yule-Walker estimation on model 2 residuals
ar(res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# AICc values for model 1 and model 2
AICc(arima(co2.train, order=c(0,1,1), seasonal=list(order=c(0,1,1), period=12),
           method="ML"))
AICc(arima(co2.train, order=c(1,1,1), seasonal=list(order=c(0,1,1), period=12),
           method="ML"))

# predict 12 observations
pred.tr <- predict(fit1, n.ahead=12)

# upper bound of prediction interval
U.tr <- pred.tr$pred+2*pred.tr$se

# lower bound of prediction interval
L.tr <- pred.tr$pred-2*pred.tr$se

# plot training data
ts.plot(co2.train, xlim=c(300,length(co2.train)+12), ylim = c(370,420),
        main="CO2 Levels in Hawaii", xlab ="Time", ylab="CO2 Levels (ppm)")

# plot confidence intervals
lines(U.tr, col="green", lty="dashed")
lines(L.tr, col="green", lty="dashed")

# graph forecasted data
points((length(co2.train)+1):(length(co2.train)+12), pred.tr$pred, col="red",cex=0.4)
points((length(co2.train)+1):(length(co2.train)+12), co2.test, col="blue",
       cex=0.4)
legend("topleft", pch=1, col=c("red", "blue"),
       legend=c("Forecast value", "True value"))
```