Predicting CO_2 Concentrations with SARIMA

PSTAT 174 Final Project

June 6th, 2018

Group Phi:

Uma Kumar
Jacobo Pereira
Joey Cerquera
Khalid Nagib
Jeff Oliveria

Abstract

In this project we will present a time series analysis on CO_2 emissions of the volcano, Mauna Loa, gathered from the Mauna Loa Observatory in Hawaii. The objective of this analysis is to provide insight on future activity, to reduce the impact to both human and non-human inhabitants. The questions we addressed are whether the CO_2 emissions follow a seasonal trend, that is if the volcano has a certain periodicity to its activity. As well as if the activity could accurately be predicted.

By intuitively graphing the actual data and the acfs/pacfs, and transforming the data, we selected several models based on these prior findings. We then fit the models to the data, and compared the models with various criterions to further stratify and select the most effective model. The model we chose was a $SARIMA(1,1,1) \times (1,1,0)_{12}$, using AICC criteria, and this model was used to effectively forecast the future values in our test set.

1. Introduction

An estimated 200 billion metric tons of CO_2 is released and absorbed into the atmosphere each year by natural occurrence, while humans account for an additional 7 billion metric tons[1]. Many scientists believe that the addition of anthropogenic carbon dioxide is too much for nature to handle, so it is important to at least monitor the current levels, while being able to predict future levels can help with planning and economic policy. We tried several models, and ultimately with a SARIMA model. The data has a strong seasonal component, and models tested with no seasonal component all either had roots inside of the unit circle or performed poorly relatively to the SARIMA models during diagnostic testing. Our model was small, with only 3 parameters to estimate, as the larger models tested all proved to not be invertible.

The data set covers a span of 16 years, from 1965 through 1980. We reserved the final twelve observations as test data, and built the model with the first 180 observations. More information on the data can be found here. The fitted model did a very good job of predicting the test data, and we concluded that this type of data is an excellent candidate for predictive modeling using seasonal ARIMA modelling. All work with the data was done with R, using R-Studio, and the written portion/equations were coded with LaTeX.

1.1 Report Layout

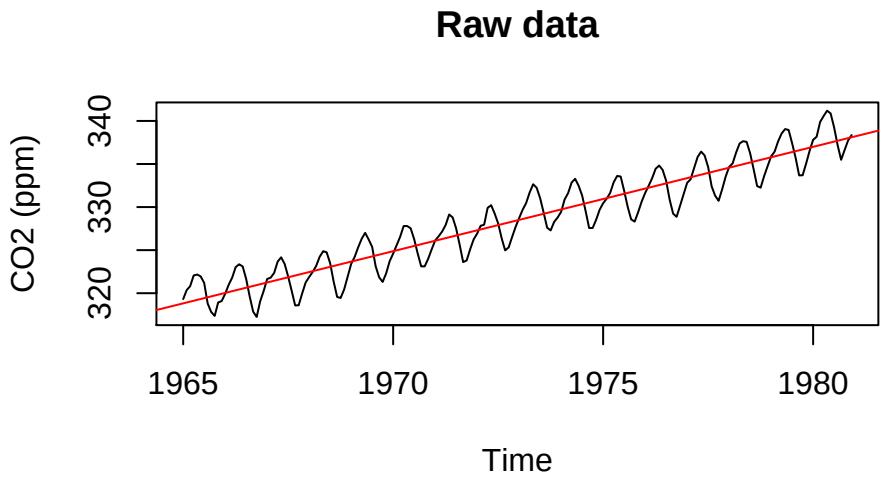
In **Section 2**, we perform exploratory data analysis, working with the subsetted training data and transforming/differencing in order to ensure stationarity before fitting the model. We tried two different models fit with two different methods, and performed diagnostic testing on both before moving on to forecasting with the better model. **Section 3** provides our conclusions and takeaways from writing this report, while **Section 4** talks about possible future studies. **Section 5** lists the references used to put together the report, while, finally, **Section 6** includes all of the code used to compile the report.

---

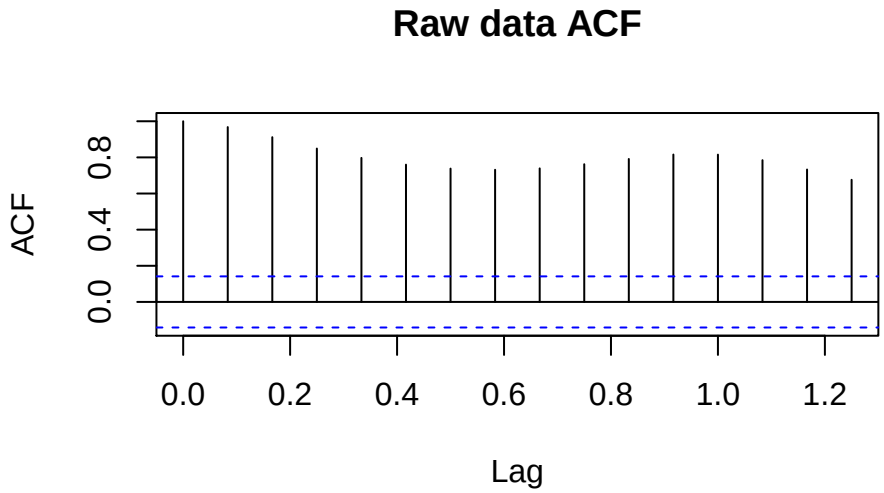
2. Data Analysis

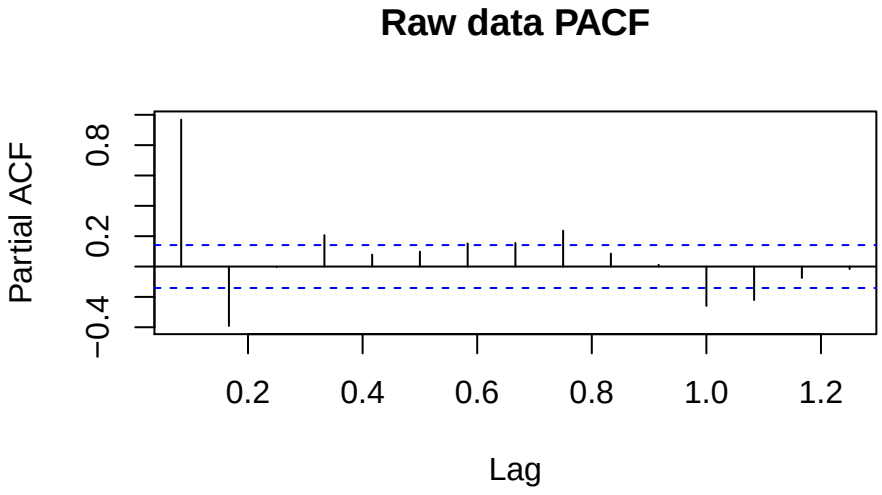
2.1 Plotting Original Data

Prior to beginning our exploratory data analysis, we split the raw data into a training set and a test set, reserving the last 12 observations for validation, as being able to go out a full year is good for planning purposes. After subsetting the data, we plotted the raw data in order to look at its characteristics and see where we needed to start.



When plotting the original data as a time series, it is apparent that there is trend and seasonality. The line fitted is the best fitting line through the data, and though there are fluctuations in variance, the data increases monotonically, so there are no sharp changes in behavior or variance. Some transformations of the data are required to stabilize the variance, and differencing appears to be needed to take care of the trend/seasonality in order to obtain a stationary series suitable for forecasting.



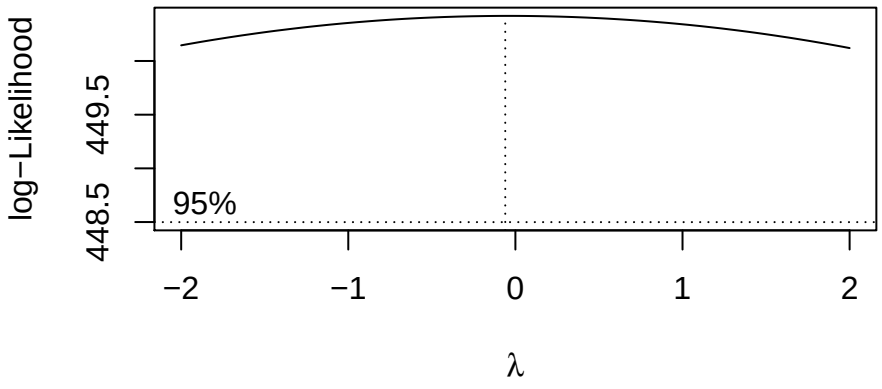


When plotting the sample ACF/PACF plots of the original data, it becomes clear that the ACF plot breaches over the confidence bound. On the other hand, the PACF plot has spikes

---

2.2 Data Transformations

We began transforming the training data by executing a box cox power transformation, to see what type of transformation would serve as a good starting point. Using R, we obtained $\lambda = -0.061$, which implies a log transform of the data, as it is close to 0. We also checked the square root transformation to see how it performed.




```
## [1] -0.06060606
```

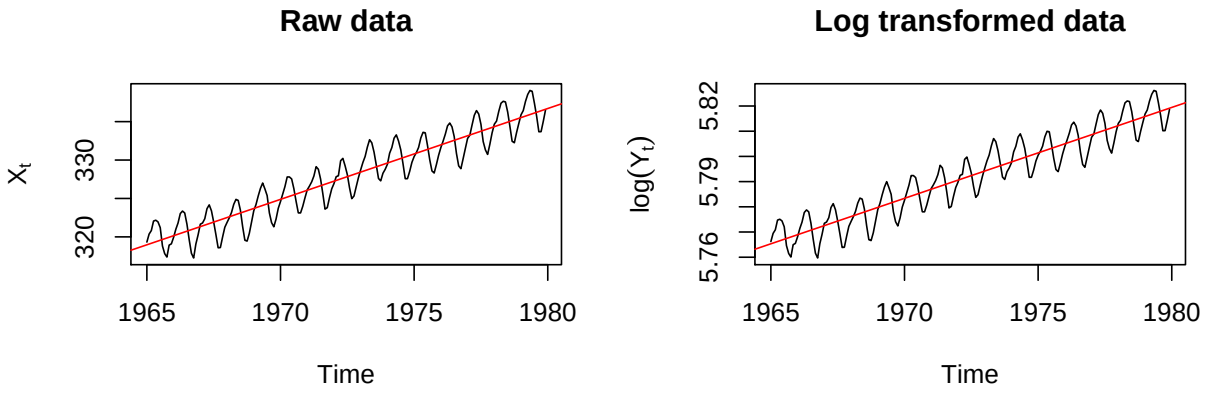
The variances resulting from the original data and the three data transformations are summarized in the table below:

Table 1: Data Transformation Variances

	variance
Raw Data	30.40229
Sqrt Transform	0.0231825131592236
Log Transform	0.000282872749098536
Box Cox	0.000140174870057968

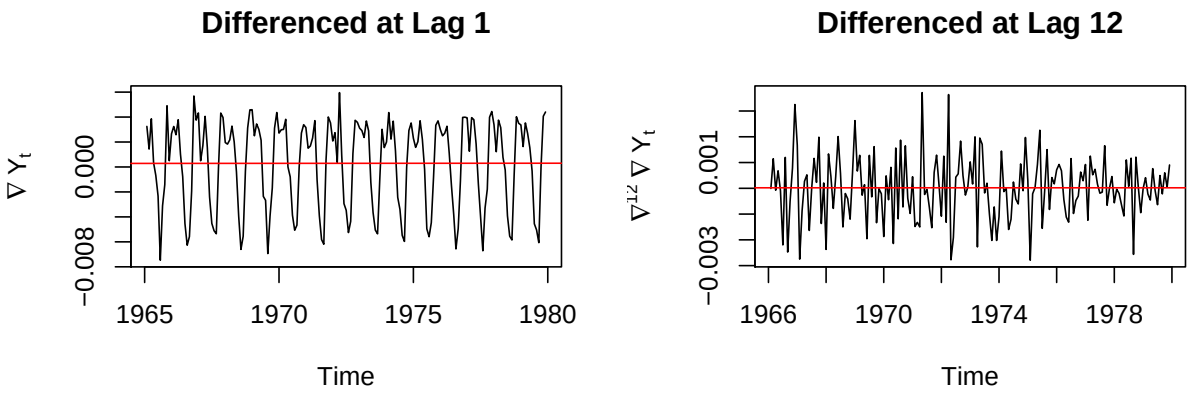
It is clear that both the box cox transformation with $\lambda = -0.061$ and $\lambda = 0$ (log transformation) did a good job of stabilizing the variance. While the box cox lowers variance the most, the log transform stabilizes the variance well and the trade-off of slightly less variance reduction is worth it since the log is easier to work with for forecasting. Thus, we will proceed with a log transform.

Graphing the transformed data shows that while the variance is stabilized, there is still a definite linear trend and seasonality. Next we will try differencing the data in order to make in stationary.



2.3 Differencing

We will start by differencing at lag 1 in order to remove the linear trend, and then recheck the shape of the data to see if any additional differencing will be beneficial. We will also pay attention to changes in the variances from differencing, as an increase in variance can be a sign of over differencing.




```
## [1] Differencing at lag 1 lowered the variance by 0.0002705
```

```
## [1] Differencing at lags 1 and 12 lowered the variance by 1.081e-05
```

After differencing at lag one, we have a horizontal mean, as desired, so the linear trend is now gone. We can see that the movements of the plot are not random, so there is still likely a component of seasonality. This can be helped by differencing a second time, this time at lag 12, since the data has a yearly cycle where it peaks in May and troughs in October.

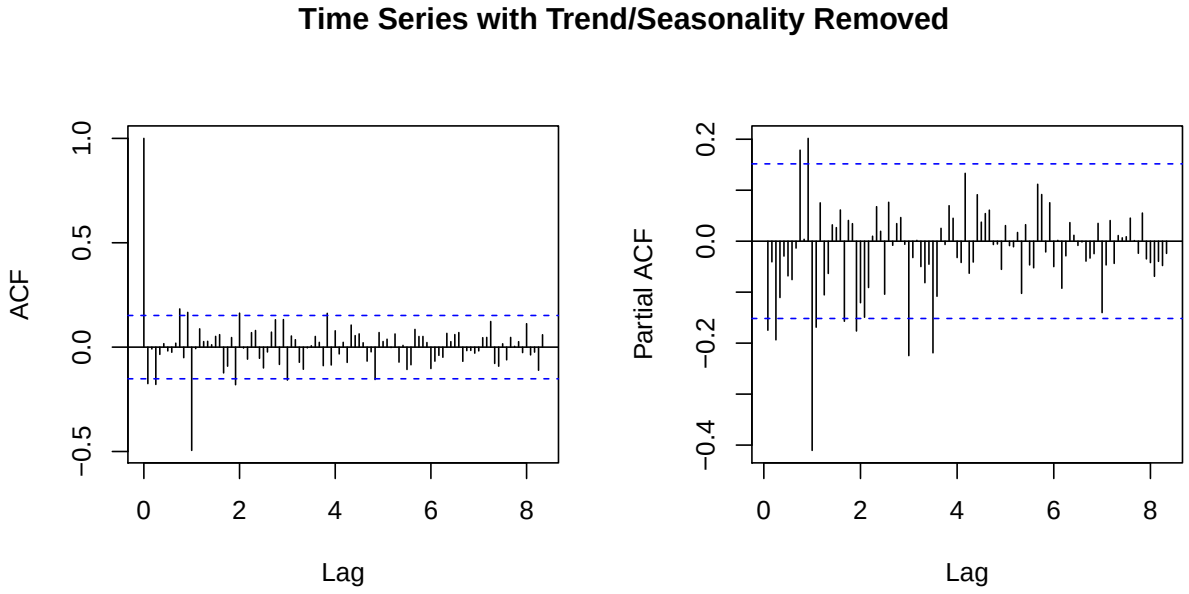
After the second round of differencing, the seasonality component now looks to be gone, and we have data that looks a lot like what you'd expect from Gaussian White Noise. Thus we now have stationary data that can be used for forecasting. Next, we will look at the ACF and PACF plots in order to identify components of suitable models.

2.4 Model Fitting

The ACF and PACF plots start as a good start point for model fitment. Looking at the shape and significant points on the plots tell a lot about what will be a good choice for the AR and MA components, as well as possible seasonal components required. We can also use iterative processes in R to determine the necessary components based on different informational tests, such as AIC, AICC, BIC, and Yule-Walker equations for example. We will also fit a second model using AICC.

2.4.1 ACF/PACF Model

We begin by plotting the ACF and PACF, using the transformed training data.



The PACF shows clear lag spikes up to just before lag 4, with nothing significant after that. That is indicative of an AR(3) model. The ACF has also has a spike around lag 4 before cutting off, which indicates a possible MA(4) model. Our data set is clearly seasonal, so seasonal components are likely needed. The ACF tapers off slowly, while the PACF has clusters of lag spikes each year, which indicates an AR(1). We differenced the data twice to get stationarity, at lags 1 (trend) and lag 12 (seasonality), so d = D = 1 makes sense. Thus, from the plots, the model we chose is:

$$SARIMA(3,1,4) \times (1,1,0)_{12}$$

This model has an estimated variance of 0.00000372, log likelihood of 915.83, and an AICC of -1812.5.

2.4.1.2 Diagnostic Testing and Estimation of First Model

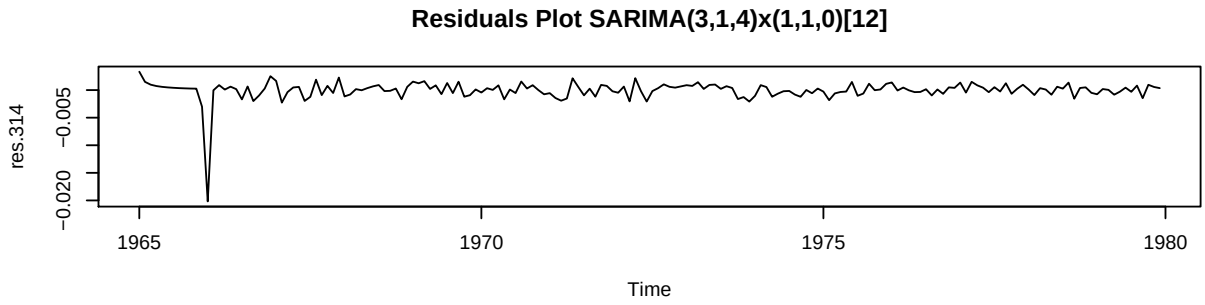
Next we'll proceed with diagnostic testing to make sure the model will be ok for forecasting. We're looking for residuals that approximate white noise, while also minimizing serial correlation. We will also check to ensure that the model is both causal and invertible by examining the roots of the characteristic polynomial.

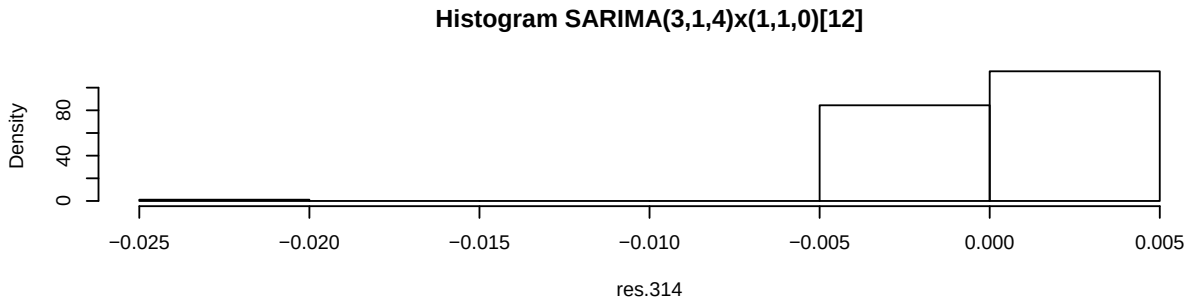
Table 2: SARIMA(3,1,4)x(1,1,0) Coefficients/Error

	COEFF	SE
ar1	0.530	0.541
ar2	-0.881	0.101
ar3	0.375	0.463
ma1	-0.743	0.544
ma2	1.042	0.237
ma3	-0.728	0.544
ma4	0.031	0.224
sar1	-0.491	0.069

The model coefficients are all significant at $\alpha = 0.05$. Next, we will start diagnostics with a look at the residuals. Here the coeffecients for $SARIMA(3,1,4) \times (1,1,0)$ were estimated with the MLE model and

displayed below side by side the standard error.





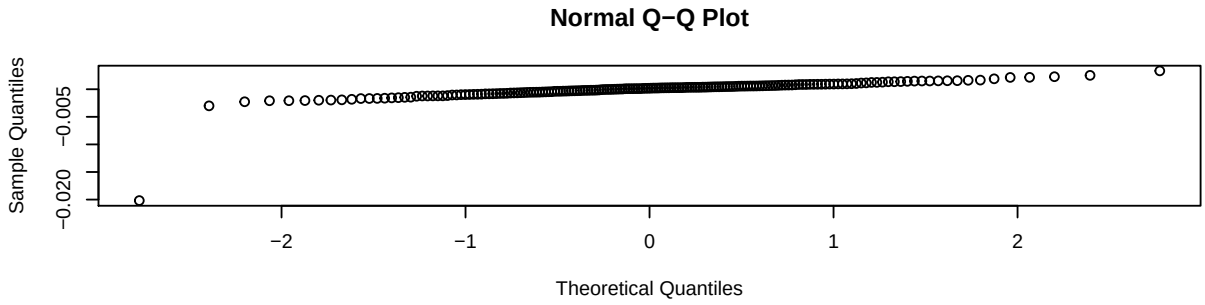
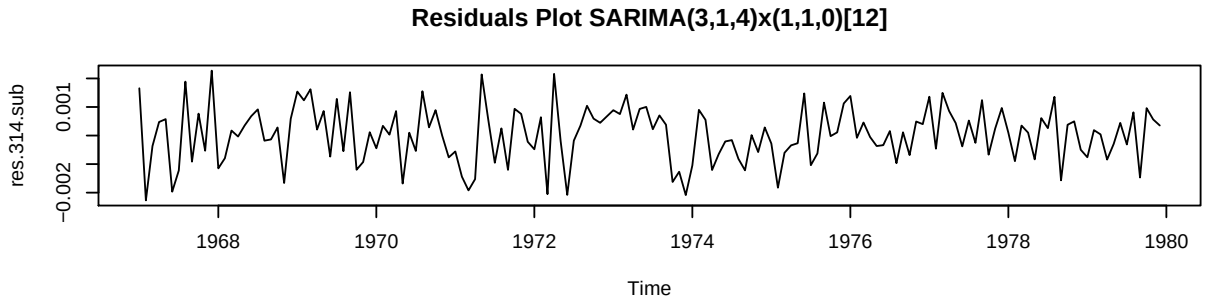
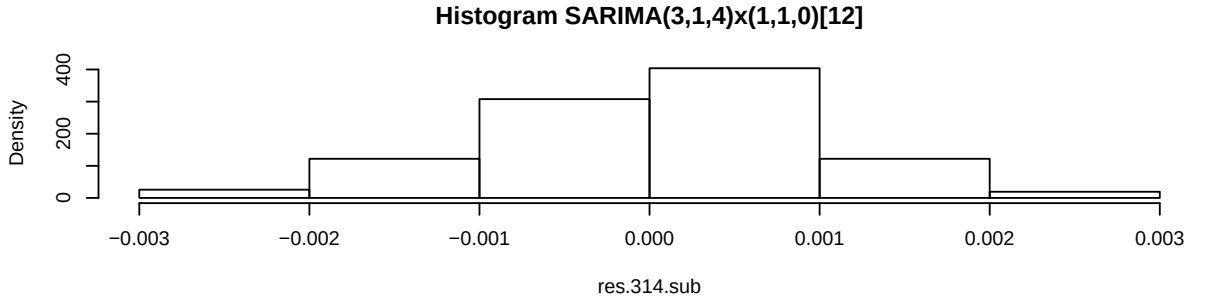


Table 3: Residual Tests for ARIMA(3,1,4)x(1,1,0)

Shapiro p	Ljung-Box p	Yule Walker Order
0	0.0696478	1

It is clear that these plots are not optimal, and the Shapiro-Wilk p-value was nearly 0. The Yule-Walker equations fitted an AR(1) model to the residuals instead of the desired AR(0). There is a large spike at the start of the data set caused by the differencing. We will proceed by subsetting the residuals and not include the first years worth of data, as it's obviously not contributing to establishing a proper pattern for the data.





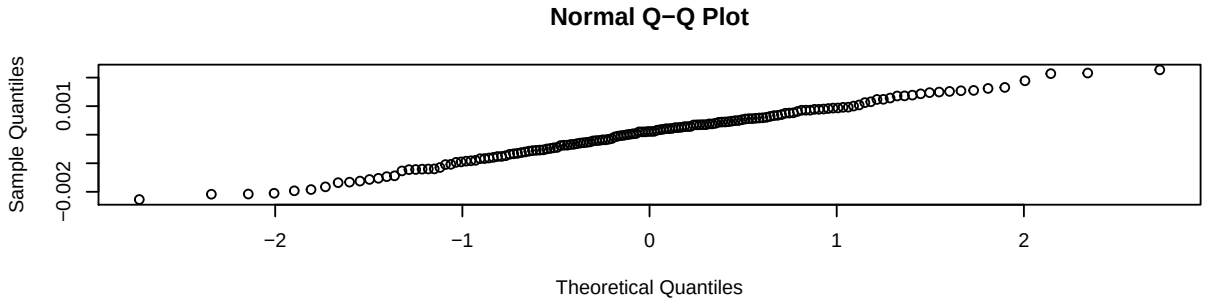
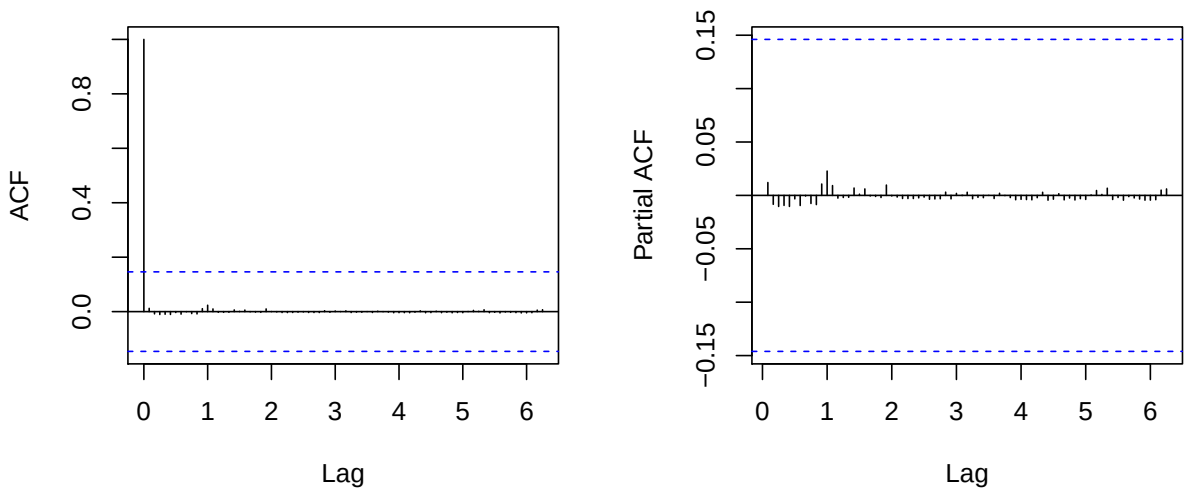


Table 4: Residual Tests for ARIMA(3,1,4)x(1,1,0) after Subset

Shapiro p	Ljung-Box p	Yule Walker Order
0.4558246	0.2696032	0

The results were much better this time, as the plots all looked approximately normal, and the p-values of the Shapiro-Wilks and Ljung-Box tests were all above 0.05, failing to reject normality of residuals. The residuals were also fitted to an AR(0) model by the Yule-Walker equation approximations, as desired. Next we'll check for homogeneity of variance using ACF and PACF plots of the residuals.



All residuals falling within confidence intervals implies residuals are homoscedastic as required. Last, we'll check the model for any unit roots to make sure it is causal and invertible, and suitable for forecasting.

```
## [1] AR Roots
```

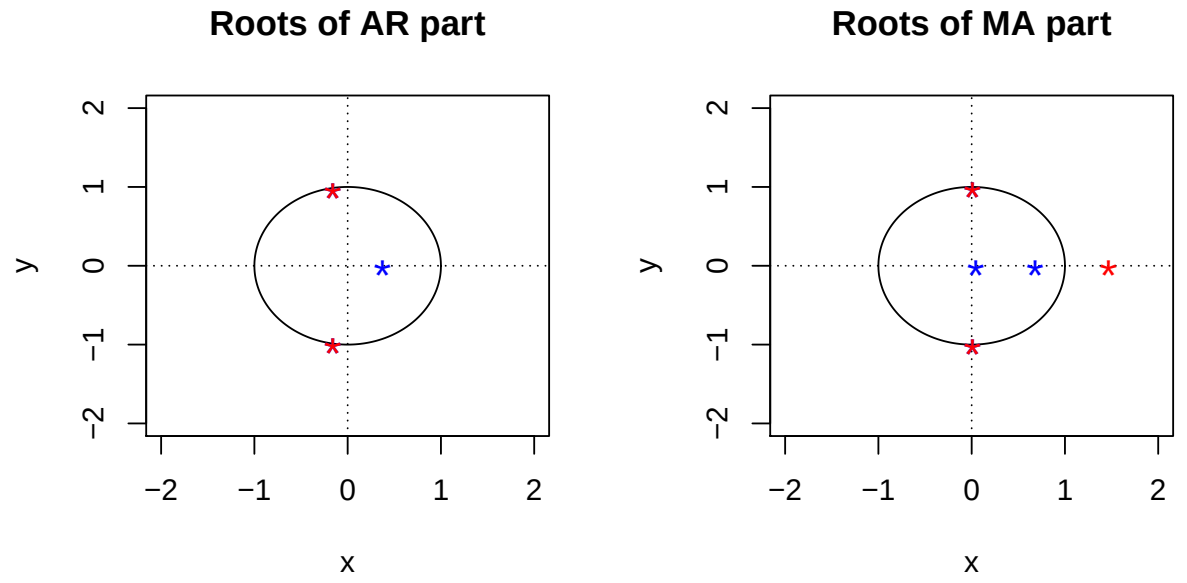
```
## [1] -0.1604777+0.9863521i -0.1604777-0.9863521i  2.6702887+0.0000000i
```

```
## [1] MA Roots
```

```
## [1]  0.007676+1.000058i  0.007676-1.000058i  1.465842+0.000000i
## [4] 22.002677-0.000000i
```

```
## [1] SAR Root
```

```
## [1] 2.03666+0i
```



This model produces complex conjugate unit roots for the AR and MA portions. $0.00788 \pm 1.000058i \approx 1.0000319$. One could argue that this is still out of the unit circle, however given the measurements contain variance the model may or may not be invertible, so we're discarding to err on the side of caution. Looking at the ACF/PACF, the ACF is almost 4 before it tails off, so next we'll try

$$SARIMA(4,1,4) \times (1,1,0)_{12}$$

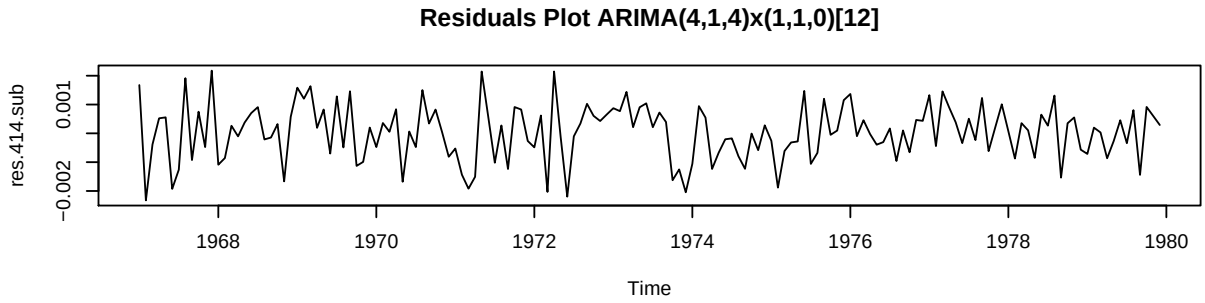
This model has an estimated variance of 0.00000347, log likelihood of 915.98, and an AICC of -1810.5

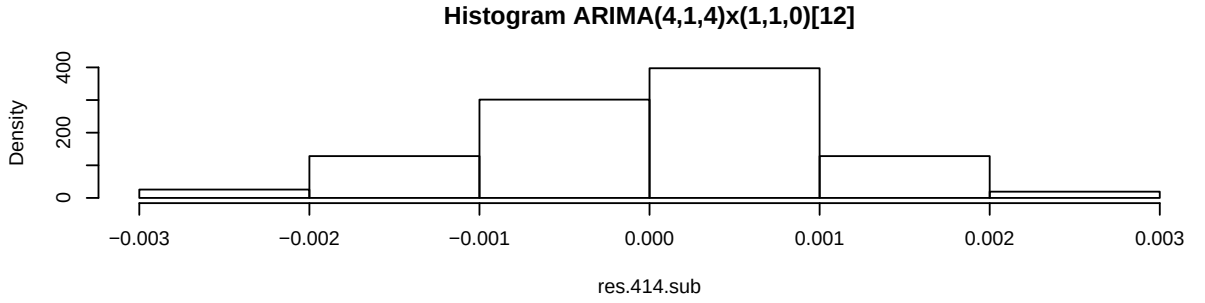
2.4.1.3 Diagnostic Testing and Estimation of Second Model

Table 5: SARIMA(4,1,4)x(1,1,0) Coefficients/Error

	Coefficients	STD ERROR
ar1	-0.0771904	0.5579813
ar2	-0.5404761	0.2635490
ar3	-0.1588742	0.4853026
ar4	0.2556404	0.2146788
ma1	-0.1405613	0.5528391
ma2	0.5932363	0.3446859
ma3	-0.1173574	0.5553547
ma4	-0.4086578	0.3362283
sar1	-0.4922910	0.0688930

Here is the esimation of the model coefficients for the $SARIMA(4,1,4) \times (1,1,0)$ model showcasing that all coeffecients are significant. Again, we will start diagnostics with a look at the residuals.





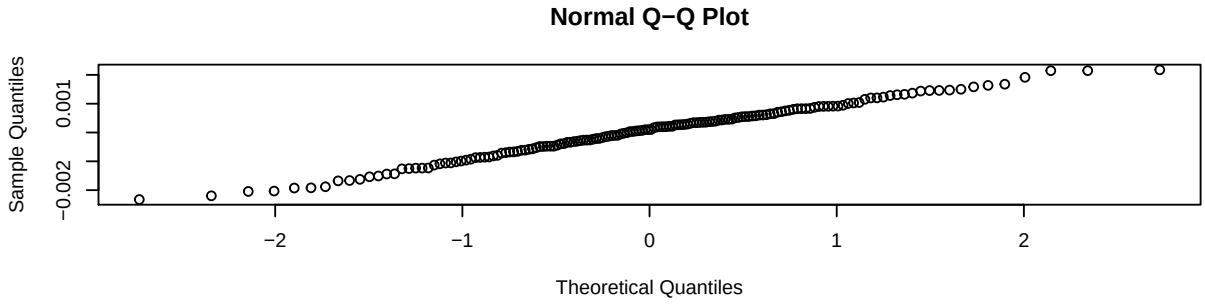
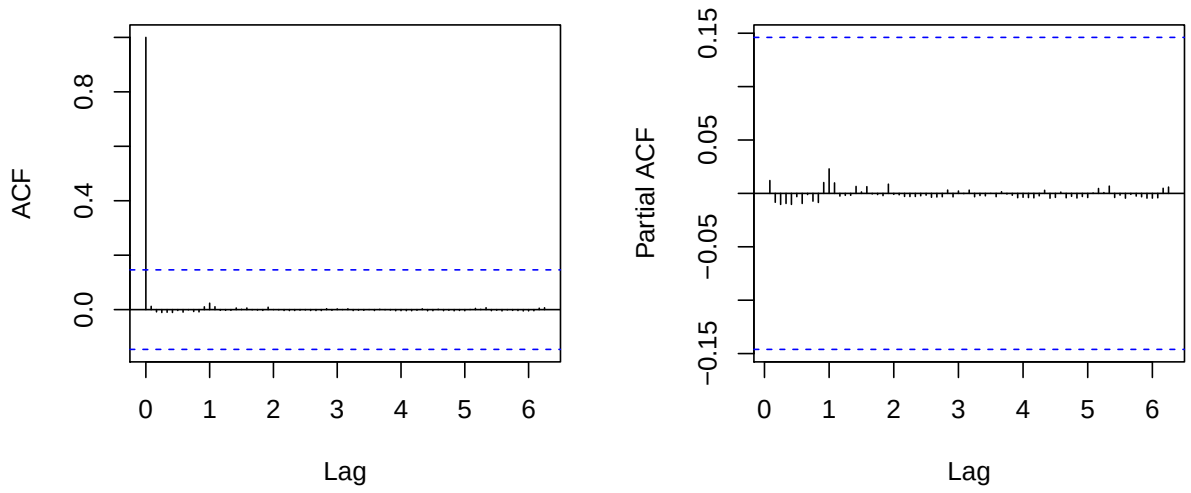


Table 6: Residual Tests for ARIMA(4,1,4)x(1,1,0) after Subset

Shapiro p	Ljung-Box p	Yule Walker Order
0.4171419	0.313532	0

Results are similar to the previous model. All of the tests were passed and residuals appear normal with no serial correlation.



All residuals falling within confidence intervals implies residuals are homoscedastic as required. Again, we'll check the model for any unit roots to make sure it is causal and invertible, and suitable for forecasting.

```
## [1] AR Root
```

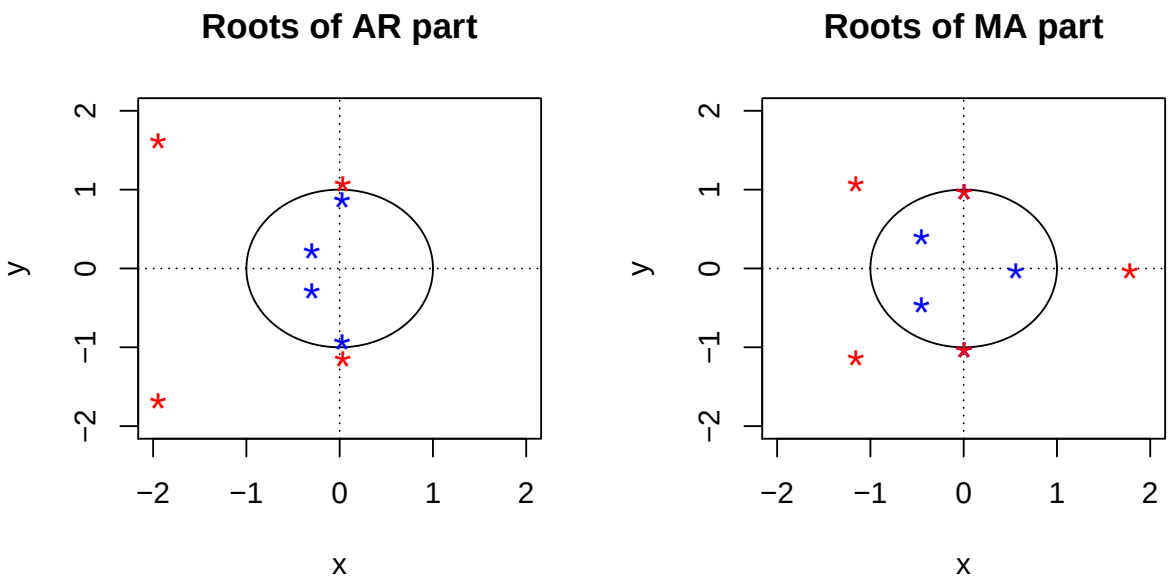
```
## [1]  0.034772+1.109277i  0.034772-1.109277i -1.946772-1.644733i
## [4] -1.946772+1.644733i
```

```
## [1] MA Root
```

```
## [1]  0.005550+1.000327i -1.157122+1.104930i  0.005550-1.000327i
## [4]  1.782595+0.000000i -1.157122-1.104930i
```

```
## [1] SAR Root
```

```
## [1] 2.03252+0i
```



Again this model produces an approximate unit root and is not invertible so not suitable for forecasting. We also tried a $SARIMA(4,1,3) \times (1,1,0)_{12}$ model, but it too had a unit root. Thus, we will proceed with using AICC criteria to choose our model with a nested for loop to see what R suggests for a model.

2.4.2 Model Fitment and Estimation with AICC

AICC is an extension of the Akaike information criterion (AIC), which corrects for our relatively small sample size. We used a nested for loop in R to generate the data frame below, which shows each process component from 0 to 4. their intersection is the corresponding AICC. We seek to minimize the function, so we're looking for the largest number in magnitude (since values are all negative). Subtracted off the AICC value of ARMA(0,0) to make it 0 and the other values more readable, as it doesn't change their values respective to one another.

Table 7: AICC's of ARMA(0,0) to ARMA(4,4)

	MA0	MA1	MA2	MA3	MA4
AR0	0.00000	-134.4231	-137.8159	-137.0964	-141.4775
AR1	-63.08896	-137.1871	-141.5365	-140.2893	-139.4529
AR2	-73.52154	-135.3127	-133.1580	-139.6300	-137.4736
AR3	-86.38389	-139.1852	-139.3785	-137.2316	-143.7957
AR4	-98.86893	-138.9024	-137.2331	-140.7165	-132.9546

We set the number of differences to d = 1 (differenced one time for trend), and the best choice according to AICC is an ARIMA(4,1,3) model, but AIC does not check for unit roots, and (4,1,3) was shown to not be invertible while trying different models for the ACF/PACF models. Thus, we'll select the smaller model with lowest AICC... ARIMA(1,1,2) and ARIMA(1,1,1), a simple model with a very low AIC compared to the other models of its size. We will look at both and see how they stand up to diagnostics.

For both $ARMA(1,1,2)$ and $ARMA(1,1,1)$, the coefficient estimations were also accomplished using MLE. As seen below the model coefficients are all significant. We next look at the residuals and commence diagnostics.

	COEFF	SE
AR(1)	0.4021972	0.0957459
MA(1)	0.5188575	0.0952483
MA(2)	0.3946682	0.0759370

	COEFF	SE
AR(1)	0.5677486	0.0728547
MA(1)	0.3394245	0.0688800

Table 8: ARMA(1,1,2) and ARMA(1,1,1)

2.4.2.2 Diagnostics of AICC Models

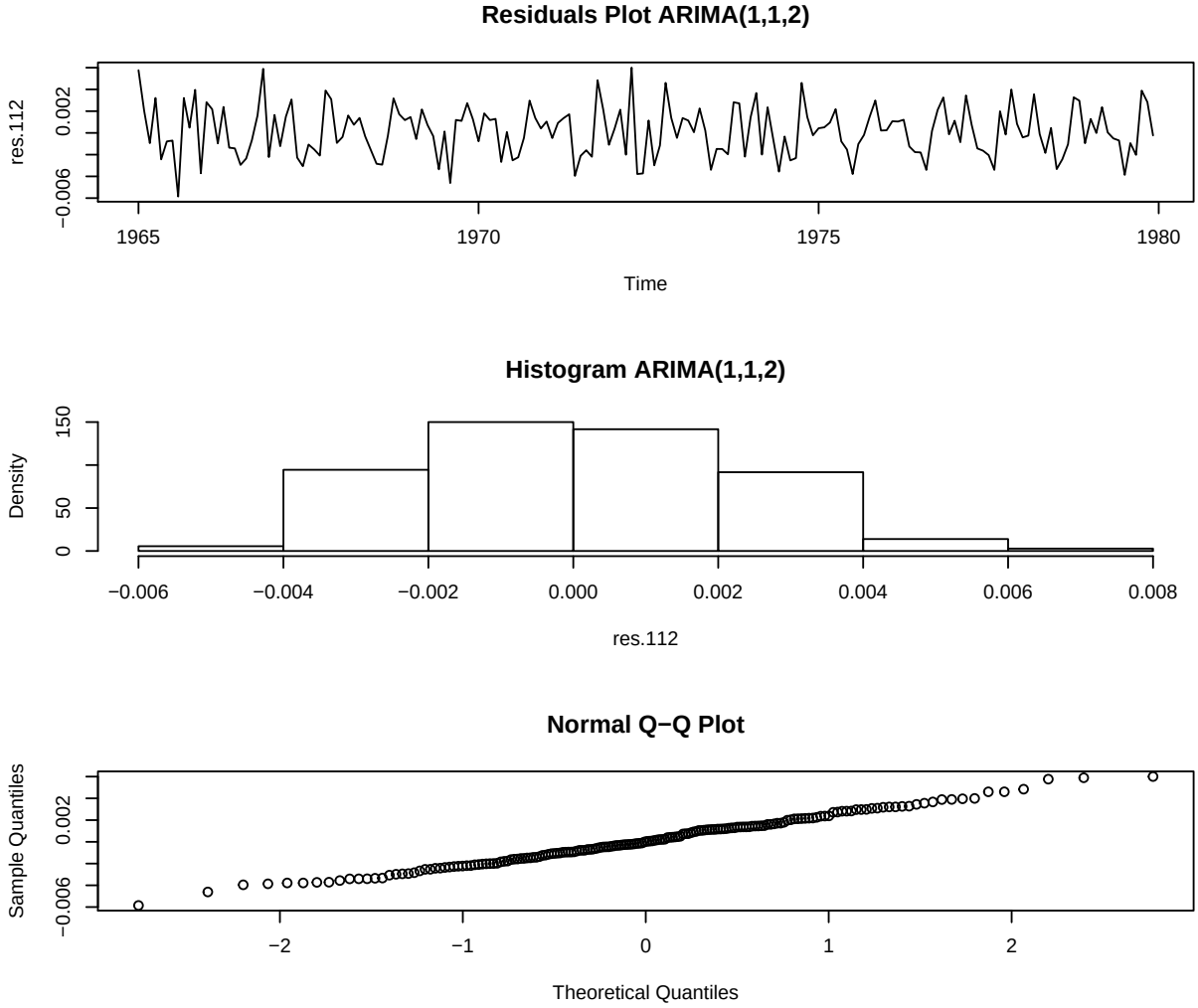


Table 9: Residual Tests for ARIMA(1,1,1)

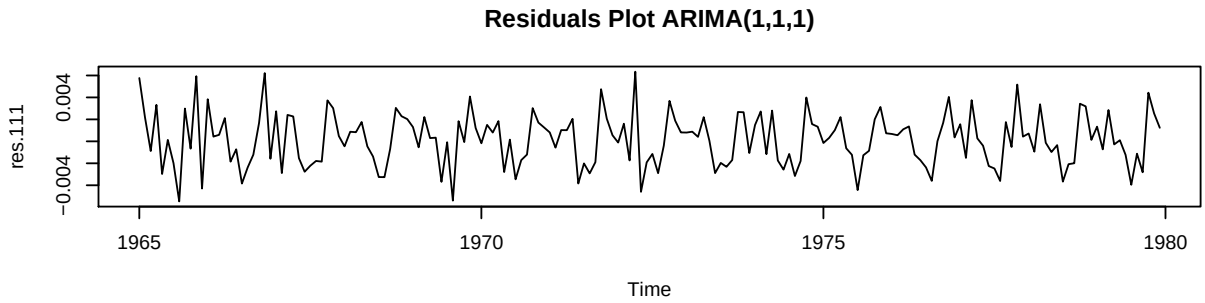
Shapiro p	Ljung-Box p	Yule Walker Order
0.5690319	0	14

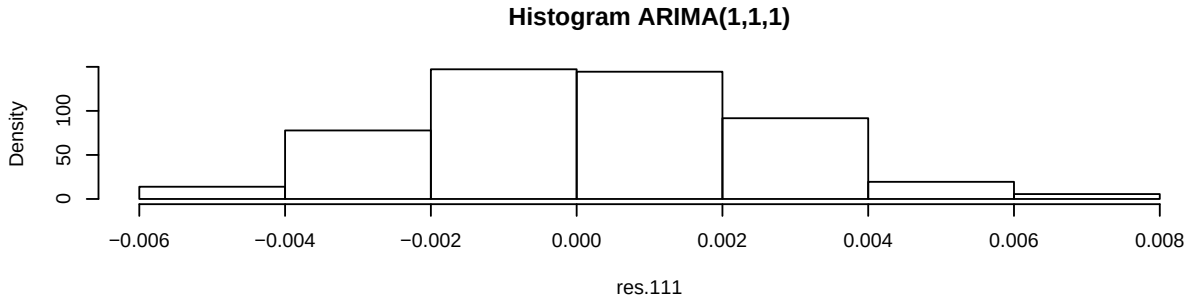
Although the residuals passed the normality tests, with a strong Shapiro-Wilks p-value and roughly normal looking plots, we see it failed the Box-Ljung test, implying serial correlation. There looks to be a seasonal component not explained by the model, judging from the plot of the residuals. To correct this, we'll add on the same seasonal component used on the ACF/PACF models above, for the same reasoning.

Thus, the model is now:

$$SARIMA(1,1,2) \times (1,1,0)_{12}$$

Now, we'll check the $ARIMA(1,1,1)$ model's residuals:





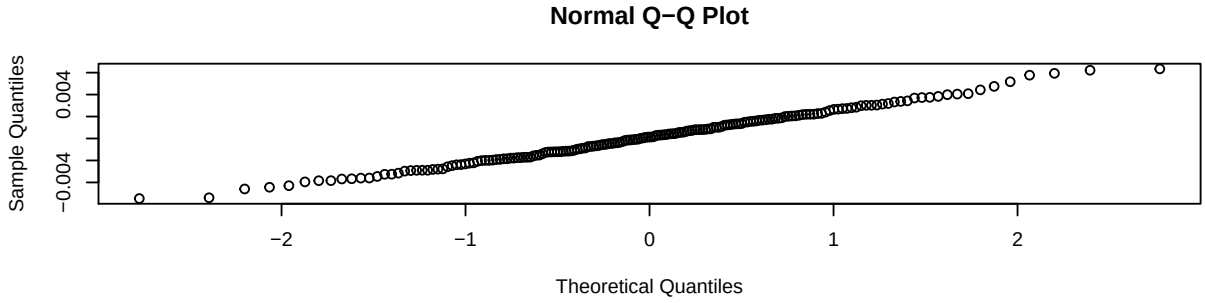


Table 10: Residual Tests for ARIMA(1,1,1)

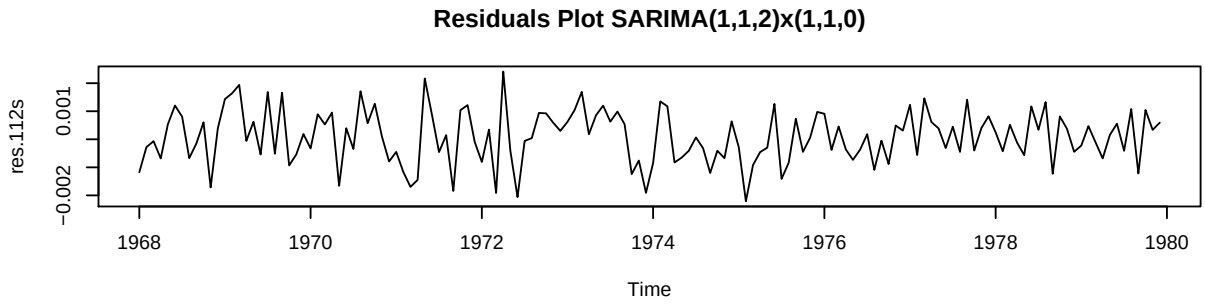
Shapiro p	Ljung-Box p	Yule Walker Order
0.711013	0	13

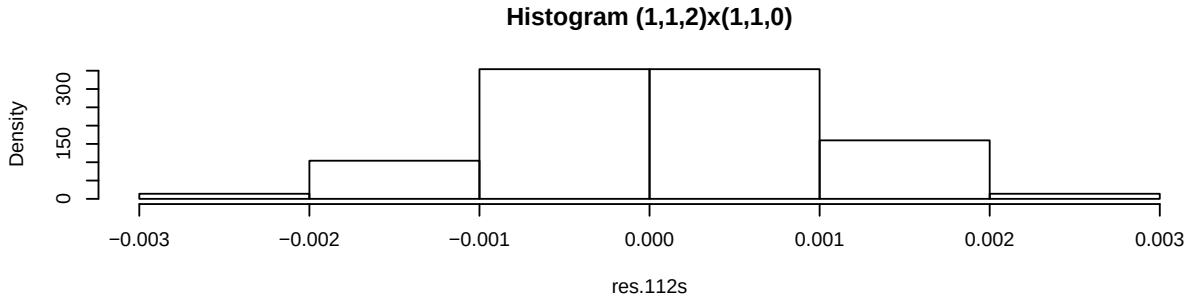
Results proved to be similar to the other model. . . fitting seasonal component to both and rerunning diagnostics.

First, we'll check the $SARIMA(1,1,2) \times (1,1,0)_{12}$ model:

```
## Series: y.tr
## ARIMA(1,1,1)(1,1,0)[12]
##
## Coefficients:
##          ar1      ma1     sar1
##       0.5165  -0.7180  -0.5114
## s.e.  0.2217   0.1832   0.0666
##
```

```
## sigma^2 estimated as 3.695e-06:  log likelihood=911.41
## AIC=-1814.82   AICc=-1814.57   BIC=-1802.35
```





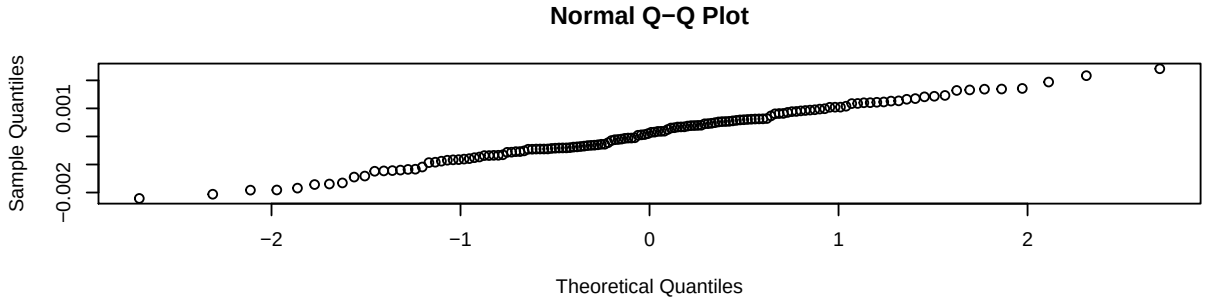
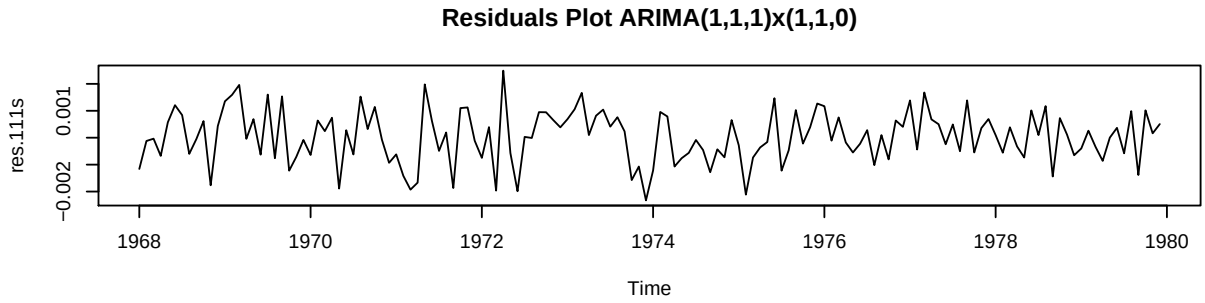
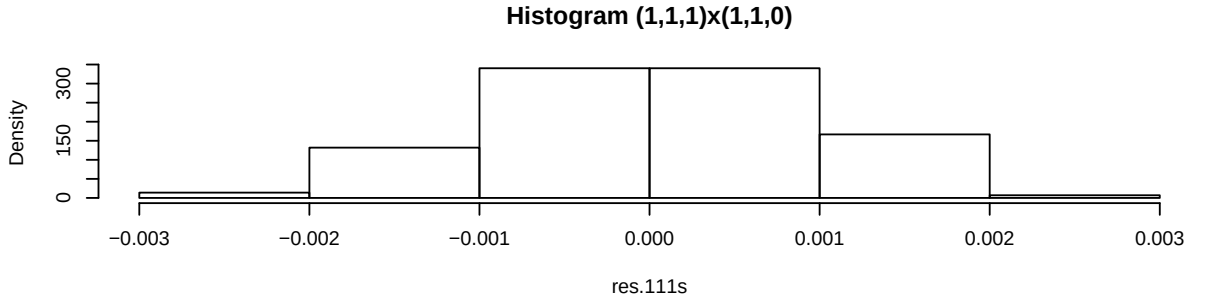


Table 11: Residual Tests for ARIMA(1,1,2)x(1,1,0)

Shapiro p	Ljung-Box p	Yule Walker Order
0.5717504	0.4355177	0

Adding the seasonal component cleared up the previous issues, and the model is now passing all diagnostic tests for the residuals. Next we'll test the other smaller model with the addition of the seasonal component:





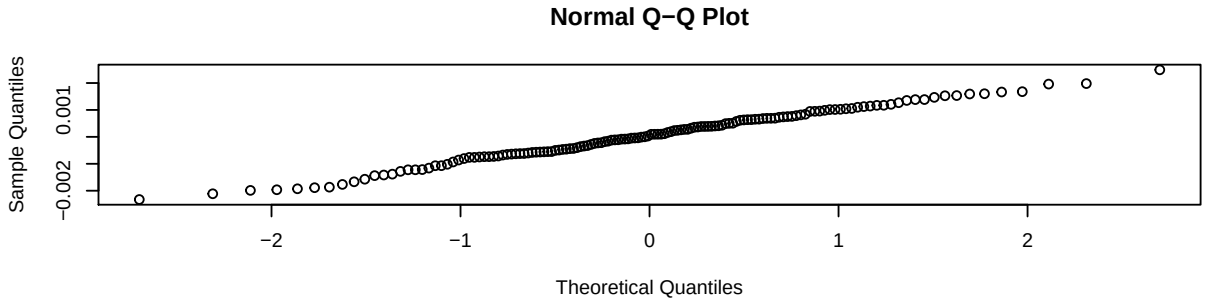
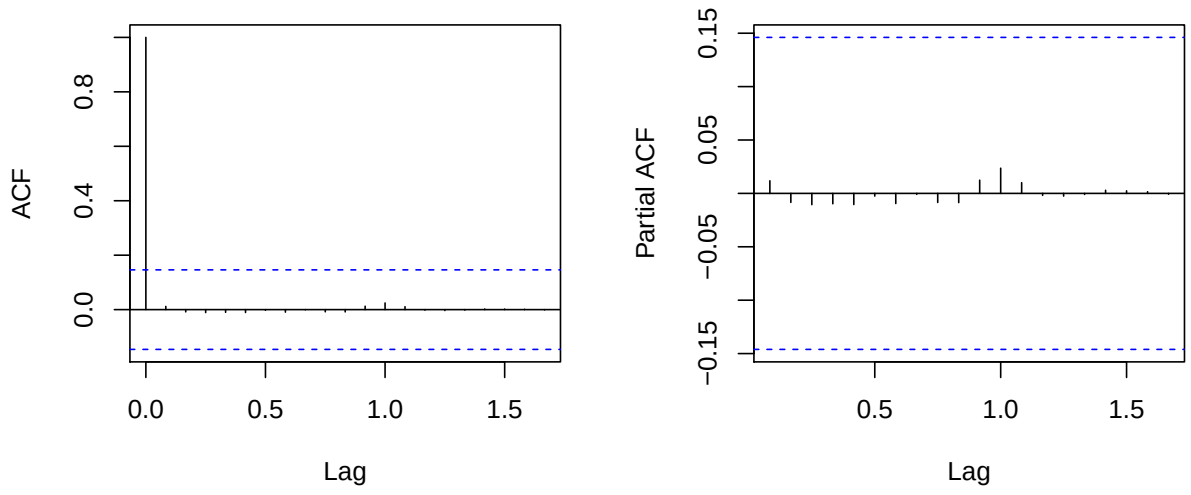


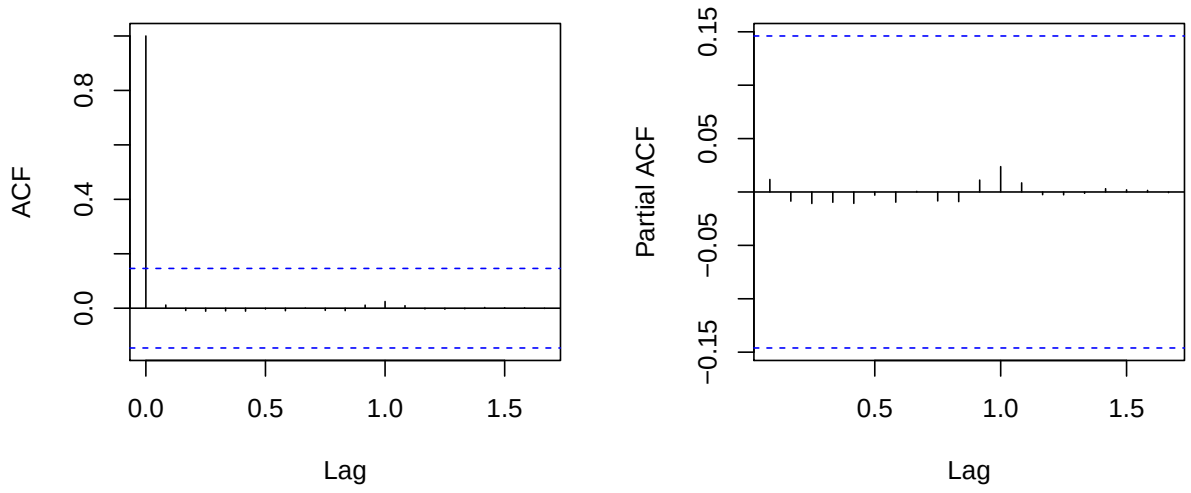
Table 12: Residual Tests for ARIMA(1,1,1)x(1,1,0)

Shapiro p	Ljung-Box p	Yule Walker Order
0.5721905	0.4467208	0

This model is now passing residual diagnostics as well. We're now ready to move on to checking the ACF/PACF of Squared residuals and then checking the roots of the respective characteristic equations.



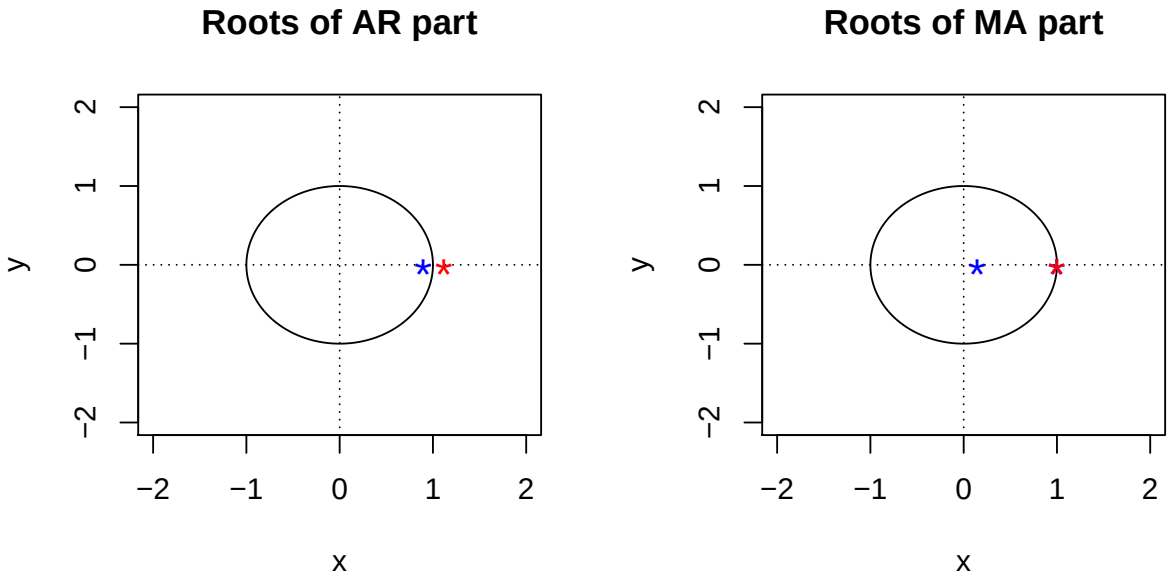
ACF and PACF of squared residuals have all values within confidence interval, which implies homoscedasticity.



ACF and PACF of squared residuals have all values within confidence interval, which implies homoscedasticity, so both models pass the test. Next, we'll check roots to establish causality and invertibility.

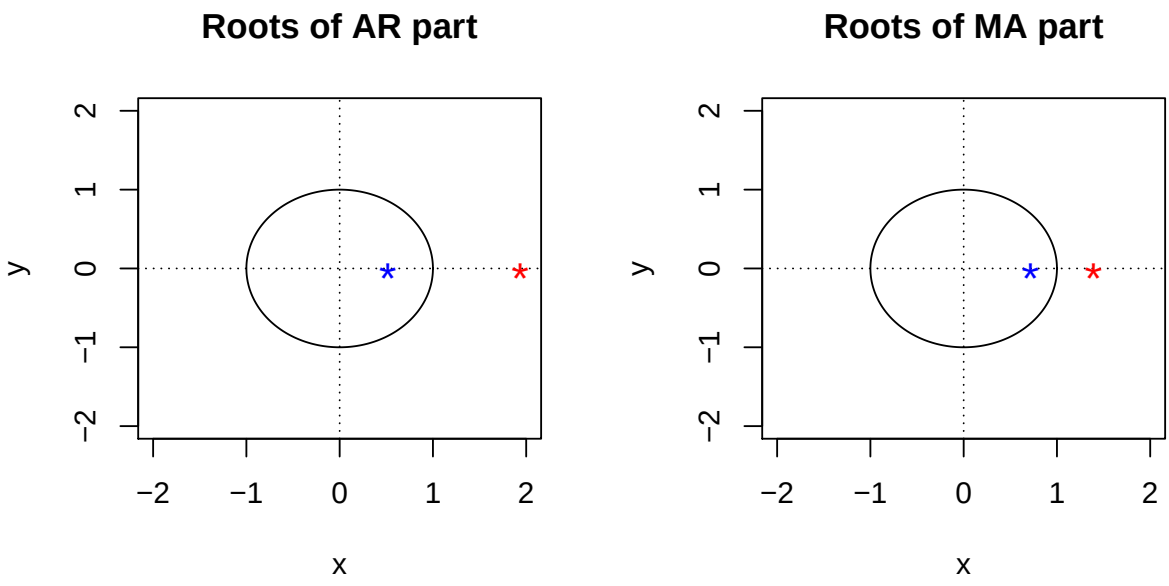
```
## [1] AR Root
## [1] 1.115822+0i
## [1] MA Root
## [1] 1.000000-0i 6.939625+0i
## [1] SAR Root
```

```
## [1] 1.953125+0i
```



The $SARIMA(1,1,2) \times (1,1,0)_{12}$ model has a unit root for MA(1), which can be a sign of over differencing. We'll check the other model and proceed from there.

```
## [1] AR Root
## [1] 1.934236+0i
## [1] MA Root
## [1] 1.392758+0i
## [1] SAR Root
## [1] 1.956947+0i
```



This model has an AR root of 1.93, an MA root of 1.39, and a SAR root of 1.96, which all lie outside of the unit circle, and thus our model is causal and invertible and suitable for forecasting. Thus, our final model going forward will be: $SARIMA(1,1,1) \times (1,1,0)_{12}$, which was also the smallest model we tested.
Thus, our final model is:

$$(1 - 0.517B)\nabla^{12}\nabla X_t = (1 - 0.718B - 0.511B^{12})\hat{\sigma}^2 = 0.00000370$$

The models used in our analysis are summarized in the table below. Our chosen model performed very well in all diagnostic testing, and had very good AICC and BIC numbers as well. The variances were uniform across all models.

	Variance	Log Likelihood	AICC	BIC	Shapiro	Box-Ljung
(3,1,4)x(1,1,0)	0.00000372	915.8255	-1812.504	-1785.589	0.456	0.270
(4,1,4)x(1,1,0)	0.00000375	915.9817	-1810.553	-1780.783	0.417	0.314
(1,1,2)x(1,1,0)	0.00000368	913.1316	-1815.891	-1800.673	0.572	0.436
(1,1,1)x(1,1,0)	0.0000037	911.4109	-1814.575	-1802.350	0.572	0.447

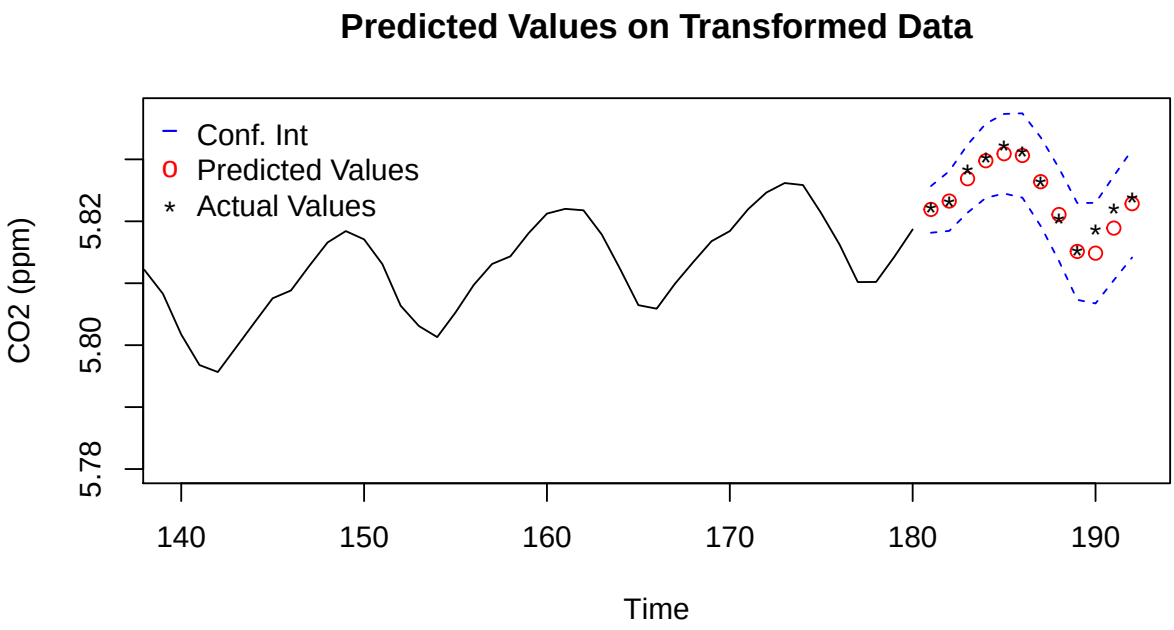
Next, we proceed to forecasting.

2.5 Forecasting

We set aside 12 months of observations which weren't touched for the model fitting/diagnostics, and we will now see how well our model is able to predict those actual results.

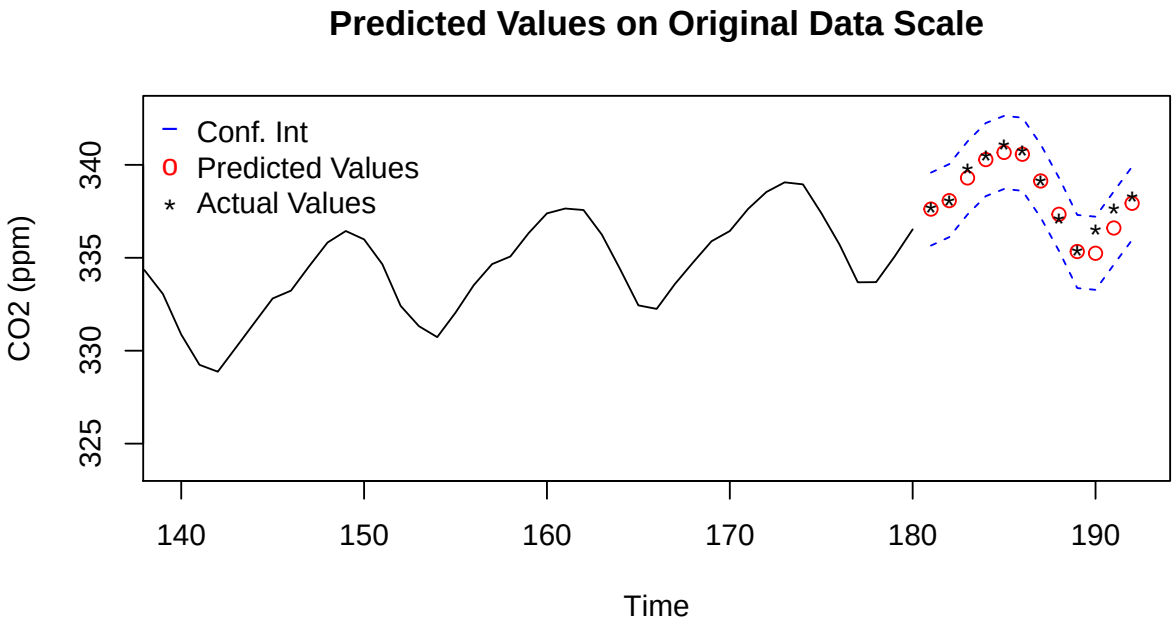
2.5.1 Forecasting Transformed Data

We start with plotting the transformed data, and adding in our predicted values.



The predicted values (red circles) do a very good job of approximating the actual values (black '*'). The model extended to the test data very well and does a good job of predicting the proper values. Next, we'll rescale the data back to its original form so the predictions make more intuitive sense.

2.5.2 Forecasting after undoing the Transformation



This plot is the exact same as the last, aside from rescaling the Y-axis.

3. Conclusion

This project aimed to investigate the monthly amount of CO_2 emitted from the volcano Mauna Loa. After initial data exploratory analysis we decided that this data was nonstationary and transformation and differencing should be used. After trying multiple transformations which included square root, log, and box cox we settled on using box cox because it was the easiest to work with. Based on ACF and PACF plots we decided on a model SARIMA (4,1,4) x (1,1,0) and ran diagnostic checks on this model. This model passed all the diagnostic checks and a had an acceptable AIC value, but it, like the next model chosen, SARIMA (4,1,3) x (1,1,0), had a unit root for a portion of its MA process and was thus not invertible which means it may not have been suitable for forecasting. Thus, we moved on to iterative processes to find our model.

We ran a for loop to see which model gave us the lowest AICC and it ended up being ARIMA(1,1,2). However when running diagnostic checks on this model, it had roots on the unit circle so we choose the next best model which was ARIMA (1,1,1). This model did not pass the Ljung Box test or serial correlation part of diagnostic checking, so we added a seasonal component to this model. The model ended up being a SARIMA (1,1,1) x (1,1,0), which passed all the diagnostic checks so we were able to move onto foresting.

Based on the model we choose, it predicted the values very accurately as seen by the graph. It is within the confidence interval bounds and follows the trend very closely as seen in the original plotting of the time series. We also included the actual values which are shown as asterisks and our predicted values which are red circles. Most of the predicted values are at the same spot if not very close to what the actual values were. From this we can say that we successfully fitted a model, that will help us predict the next year of CO_2 emitted from Mauna Loa.

4. Future Study

The results of the project were quite accurate given the unique data set. To improve the study in the future, more complex forms of prediction would be useful to employ. A weighted use of previous values, seasonality, and trend could yield a better prediction. In this data set, data is only collected for 16 years and it could be considered a really small data set in predicting future values as it might be part a cycle caused by a recent event such as a volcano eruption with hazardous particles affecting the air quality or recent human activity. The forecasting done was based solely on the previous values in the data set. It did not take into consideration any external factors that might affect the Carbon Dioxide emission levels. That could include but is not limited to any activity in the Mauna Loa or any other occurrence caused by erroneous human activity that would increase levels of Carbon Dioxide. There also is cause for concern as there could be potential errors in the measurement technique used at the observatory. The technology used in measuring the Carbon Dioxide levels is over 50 years old and there is a possibility for more accurate instruments available in the later years that could produce different results than those produced in this data set.

---

5. References

1. https://www.esrl.noaa.gov/gmd/ccgg/trends/
2. https://www.otexts.org/fpp
3. https://robjhyndman.com/categories/time-series/
4. http://datamarket.com/data/list/?q=provider:tsdl)
5. https://www.google.com

---

6. Appendix

```
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(eval = FALSE)
knitr::opts_chunk$set(digits = 4)
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
```

```
# install.packages('forecast') install.packages('qpcR')
library(stats)
library(forecast)
library(qpcR)
library(MASS)
library(readr)
library(knitr)

co2 <- read_csv("https://docs.google.com/spreadsheets/d/e/2PACX-1vQulcZcSOK9l5iNKU_MsBvdDlgF0qdm9rJfqA3
    col_types = cols(month = col_skip()))  ##Skipped 'Months' column as it's superflous...months added

mauna_loa <- ts(co2, frequency = 12, start = c(1965, 1))  ##Set up the data in time series format with
```

```
plot.roots <- function(ar.roots = NULL, ma.roots = NULL, size = 2,
    angles = FALSE, special = NULL, sqecial = NULL, my.pch = 1,
    first.col = "blue", second.col = "red", main = NULL) {
    xylims <- c(-size, size)
    omegas <- seq(0, 2 * pi, pi/500)
    temp <- exp(complex(real = rep(0, length(omegas)), imag = omegas))
    plot(Re(temp), Im(temp), typ = "l", xlab = "x", ylab = "y",
        xlim = xylims, ylim = xylims, main = main)
    abline(v = 0, lty = "dotted")
    abline(h = 0, lty = "dotted")
    if (!is.null(ar.roots)) {
        points(Re(1/ar.roots), Im(1/ar.roots), col = first.col,
            pch = my.pch)
        points(Re(ar.roots), Im(ar.roots), col = second.col,
            pch = my.pch)
    }
    if (!is.null(ma.roots)) {
        points(Re(1/ma.roots), Im(1/ma.roots), pch = "*", cex = 1.5,
            col = first.col)
        points(Re(ma.roots), Im(ma.roots), pch = "*", cex = 1.5,
            col = second.col)
    }
    if (angles) {
        if (!is.null(ar.roots)) {
            abline(a = 0, b = Im(ar.roots[1])/Re(ar.roots[1]),
                lty = "dotted")
            abline(a = 0, b = Im(ar.roots[2])/Re(ar.roots[2]),
                lty = "dotted")
        }
```

```
        if (!is.null(ma.roots)) {
            sapply(1:length(ma.roots), function(j) abline(a = 0,
                b = Im(ma.roots[j])/Re(ma.roots[j]), lty = "dotted"))
        }
    }
    if (!is.null(special)) {
        lines(Re(special), Im(special), lwd = 2)
    }
    if (!is.null(sqecial)) {
        lines(Re(sqecial), Im(sqecial), lwd = 2)
    }
}
```

```
spec.arma <- function(ar = 0, ma = 0, var.noise = 1, n.freq = 500,
    ...) {
    # check causality
    ar.poly <- c(1, -ar)
    z.ar <- polyroot(ar.poly)
    if (any(abs(z.ar) <= 1))
        cat("WARNING: Model Not Causal", "\n")
    # check invertibility
    ma.poly <- c(1, ma)
    z.ma <- polyroot(ma.poly)
    if (any(abs(z.ma) <= 1))
        cat("WARNING: Model Not Invertible", "\n")
    if (any(abs(z.ma) <= 1) || any(abs(z.ar) <= 1))
        stop("Try Again")
    #
    ar.order <- length(ar)
    ma.order <- length(ma)
    # check (near) parameter redundancy [i.e. are any roots
    # (approximately) equal]
    for (i in 1:ar.order) {
        if ((ar == 0 & ar.order == 1) || (ma == 0 & ma.order ==
            1))
            break
        if (any(abs(z.ar[i] - z.ma[1:ma.order]) < 0.001)) {
            cat("WARNING: Parameter Redundancy", "\n")
            break
        }
    }
    #
    freq <- seq.int(0, 0.5, length.out = n.freq)
    cs.ar <- outer(freq, 1:ar.order, function(x, y) cos(2 * pi *
        x * y)) %*% ar
    sn.ar <- outer(freq, 1:ar.order, function(x, y) sin(2 * pi *
        x * y)) %*% ar
    cs.ma <- outer(freq, 1:ma.order, function(x, y) cos(2 * pi *
        x * y)) %*% -ma
    sn.ma <- outer(freq, 1:ma.order, function(x, y) sin(2 * pi *
        x * y)) %*% -ma
    spec <- var.noise * ((1 - cs.ma)^2 + sn.ma^2)/((1 - cs.ar)^2 +
```

```
        sn.ar^2)
    spg.out <- list(freq = freq, spec = spec)
    class(spg.out) <- "spec"
    plot(spg.out, ci = 0, main = "", ...)
    return(invisible(spg.out))
}
```

```
# Set up training data/test data. Saved last year (12
# observations) for validation of model.

# Training Data = first 15 years of data
mauna.train <- window(mauna_loa, start = c(1965, 1), end = c(1979,
    12))

# Validation data = last year (12 observations) of data
mauna.test <- window(mauna_loa, start = c(1980, 1))
```

```
ts.plot(mauna_loa, main = "Raw data", ylab = "CO2 (ppm)")
abline(reg = lm(mauna_loa ~ time(mauna_loa)), col = 2)
```

```
acf(mauna_loa, main = "Raw data ACF", lag.max = 15)
pacf(mauna_loa, main = "Raw data PACF", lag.max = 15)
```

```
bc = MASS::boxcox(lm(mauna.train ~ time(mauna.train)), plotit = TRUE)
lambda = bc$x[which(bc$y == max(bc$y))]
mauna.train.bc = (1/lambda) * (mauna.train^lambda - 1)
lambda
```

```
mauna.train.log <- log(mauna.train)
```

```
mauna.train.sqrt <- sqrt(mauna.train)
```

```
var.raw <- var(mauna.train)
var.bc <- var(mauna.train.bc)
var.log <- var(mauna.train.log)
var.sqrt <- var(mauna.train.sqrt)

variance <- c(format(var.raw, scientific = 0), var.sqrt, var.log,
    var.bc)
variance <- as.data.frame(variance, row.names = c("Raw Data",
    "Sqrt Transform", "Log Transform", "Box Cox"))
variance
```

```
kable(variance, format = "pandoc", caption = "Data Transformation Variances")
```

```
op <- par(mfrow = c(1, 2))
ts.plot(mauna.train, main = "Raw data", ylab = expression(X[t]))
abline(reg = lm(mauna.train ~ time(mauna.train)), col = 2)

ts.plot(mauna.train.log, main = "Log transformed data", ylab = expression(log(Y[t])))
abline(reg = lm(mauna.train.log ~ time(mauna.train.log)), col = 2)
```

```
y <- mauna.train
y.tr <- mauna.train.log
```

```
# Difference at lag = 1 to remove trend component
y1 = diff(y.tr, 1)
var.y1 <- var(y1)

# Plot differenced at Lag 1
op <- par(mfrow = c(1, 2))
plot(y1, main = "Differenced at Lag 1", ylab = expression(nabla ~
    Y[t]))
abline(reg = lm(y1 ~ time(y1)), col = 2)

# Difference at lags 1 and 12
y.diff = diff(y1, 12)
var.diff <- var(y.diff)

# Plot differenced at Lag 12
plot(y.diff, main = "Differenced at Lag 12", ylab = expression(nabla^{
    12
} ~ nabla ~ Y[t]))
abline(reg = lm(y.diff ~ time(y.diff)), col = 2)

print(noquote(paste("Differencing at lag 1 lowered the variance by",
    format(var.log - var.y1, digits = 4))))
print(noquote(paste("Differencing at lags 1 and 12 lowered the variance by",
    format(var.y1 - var.diff, digits = 4))))
```

```
op = par(mfrow = c(1, 2))
acf(y.diff, lag.max = 100, main = "")
pacf(y.diff, lag.max = 100, main = "")
title("Time Series with Trend/Seasonality Removed", line = -1,
    outer = TRUE)
```

```
fit_arima314 <- Arima(y.tr, order = c(3, 1, 4), seasonal = list(order = c(1,
    1, 0), period = 12))
res.314 <- fit_arima314$residuals
fit_arima314
```

```r
# Generate variables to make a data frame to neatly display a
# table in output.
fit.314.coef <- round(fit_arima314$coef, digits = 3)
fit.314.se <- round(sqrt(diag(vcov(fit_arima314))), digits = 3)
fit.314.df <- data.frame(fit.314.coef, fit.314.se)
colnames(fit.314.df) <- c("COEFF", "SE")
```

```r
# Table of coefficients and std error
kable(fit.314.df, caption = "SARIMA(3,1,4)x(1,1,0) Coefficients/Error")
```

```r
op = par(mfrow = c(3, 1))
plot(res.314, main = "Residuals Plot SARIMA(3,1,4)x(1,1,0)[12]")
hist(res.314, main = "Histogram SARIMA(3,1,4)x(1,1,0)[12]", breaks = 6,
    freq = 0)
qqnorm(res.314)

st.314 <- shapiro.test(res.314)
st314.p <- st.314$p.value

box.314 <- Box.test(res.314, lag = 18, type = c("Ljung-Box"),
    fitdf = 8)
box314.p <- box.314$p.value

ar.314 <- ar(res.314, aic = TRUE, order.max = NULL, method = c("yule-walker"))
ar314.order <- ar.314$order

tests.314 <- data.frame(st314.p, box314.p, ar314.order)
colnames(tests.314) <- c("Shapiro p", "Ljung-Box p", "Yule Walker Order")
kable(tests.314, caption = "Residual Tests for ARIMA(3,1,4)x(1,1,0)")
```

```r
res.314.sub <- window(res.314, start = c(1967, 1), end = c(1979,
    12))

# Code for plots
op = par(mfrow = c(3, 1))
plot(res.314.sub, main = "Residuals Plot SARIMA(3,1,4)x(1,1,0)[12]")
hist(res.314.sub, main = "Histogram SARIMA(3,1,4)x(1,1,0)[12]",
    breaks = 6, freq = 0)
qqnorm(res.314.sub)

# Code for Residuals Testing
st.314.sub <- shapiro.test(res.314.sub)
st314.p.sub <- st.314.sub$p.value

box.314.sub <- Box.test(res.314.sub, lag = 18, type = c("Ljung-Box"),
    fitdf = 8)
box314.p.sub <- box.314.sub$p.value

ar.314.sub <- ar(res.314.sub, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```r
ar314.order.sub <- ar.314.sub$order

tests.314.sub <- data.frame(st314.p.sub, box314.p.sub, ar314.order.sub)
colnames(tests.314.sub) <- c("Shapiro p", "Ljung-Box p", "Yule Walker Order")
kable(tests.314.sub, caption = "Residual Tests for ARIMA(3,1,4)x(1,1,0) after Subset")
```

```r
op = par(mfrow = c(1, 2))
acf(res.314^2, lag.max = 75, main = "")
pacf(res.314^2, lag.max = 75, main = "")
title("SARIMA(3,1,4)x(1,1,0) Squared Residuals", line = -1, outer = TRUE)
```

```r
print(noquote(paste("AR Roots")))
# roots of AR part 1 - 0.531x + 0.881x^2 - 0.375x^3 = 0
polyroot(c(1, -0.0531, 0.881, -0.375))

print(noquote(paste("MA Roots")))
# roots of MA part 1 - 0.743x + 1.042x^2 - 0.728x^3 +
# 0.031x^4 = 0
polyroot(c(1, -0.743, 1.042, -0.728, 0.031))

print(noquote(paste("SAR Root")))
polyroot(c(1, -0.491))
```

```r
op = par(mfrow = c(1, 2))
plot.roots(NULL, polyroot(c(1, -0.0531, 0.881, -0.375)), main = "Roots of AR part")
plot.roots(NULL, polyroot(c(1, -0.743, 1.042, -0.728, 0.031)),
    main = "Roots of MA part")
```

```r
fit_arima414 <- Arima(y.tr, order = c(4, 1, 4), seasonal = list(order = c(1,
    1, 0), period = 12))
res.414 <- fit_arima414$residuals
res.414.sub <- window(res.414, start = c(1967, 1), end = c(1979,
    12))
fit_arima414
```

```r
# Generate variables to make a data frame to neatly display a
# table in output.
fit.414.coef <- fit_arima414$coef
fit.414.se <- sqrt(diag(vcov(fit_arima414)))
fit.414.df <- data.frame(fit.414.coef, fit.414.se)
colnames(fit.414.df) <- c("Coefficients", "STD ERROR")
```

```r
# Table of coefficients and std error
kable(fit.414.df, caption = "SARIMA(4,1,4)x(1,1,0) Coefficients/Error")
```

```r
res.414.sub <- window(res.414, start = c(1967, 1), end = c(1979,
    12))

# Code for plots
op = par(mfrow = c(3, 1))
plot(res.414.sub, main = "Residuals Plot ARIMA(4,1,4)x(1,1,0)[12]")
hist(res.414.sub, main = "Histogram ARIMA(4,1,4)x(1,1,0)[12]",
    breaks = 6, freq = 0)
qqnorm(res.414.sub)

# Code for Residuals Testing
st.414.sub <- shapiro.test(res.414.sub)
st414.p.sub <- st.414.sub$p.value

box.414.sub <- Box.test(res.414.sub, lag = 18, type = c("Ljung-Box"),
    fitdf = 8)
box414.p.sub <- box.414.sub$p.value

ar.414.sub <- ar(res.414.sub, aic = TRUE, order.max = NULL, method = c("yule-walker"))
ar414.order.sub <- ar.414.sub$order

tests.414.sub <- data.frame(st414.p.sub, box414.p.sub, ar414.order.sub)
colnames(tests.414.sub) <- c("Shapiro p", "Ljung-Box p", "Yule Walker Order")
kable(tests.414.sub, caption = "Residual Tests for ARIMA(4,1,4)x(1,1,0) after Subset")
```

```r
op = par(mfrow = c(1, 2))
acf(res.414^2, lag.max = 75, main = "")
pacf(res.414^2, lag.max = 75, main = "")
title("ARIMA(4,1,4)x(1,1,0) Squared Residuals", line = -1, outer = TRUE)
```

```r
print(noquote(paste("AR Root")))
# roots of AR part 1 + 0.077x + 0.540x^2 + 0.159x^3 -
# 0.256x^4= 0
polyroot(c(1, 0.543, 0.932, 0.478, 0.125))

print(noquote(paste("MA Root")))
# roots of MA part 1 - 0.141x + 0.593x^2 - 0.117x^3 -
# 0.409x^4 = 0
polyroot(c(1, 0.332, 0.879, 0.125, -0.114, -0.219))

print(noquote(paste("SAR Root")))
polyroot(c(1, -0.492))
```

```r
op = par(mfrow = c(1, 2))
plot.roots(NULL, polyroot(c(1, 0.543, 0.932, 0.478, 0.125)),
    main = "Roots of AR part")
plot.roots(NULL, polyroot(c(1, 0.332, 0.879, 0.125, -0.114, -0.219)),
    main = "Roots of MA part")
```

```r
mylist <- list()  #create an empty list

for (i in 0:4) {
    vec <- numeric()  #preallocate a numeric vector
    for (j in 0:4) {
        temp <- AICc(Arima(y.diff, order = c(i, 1, j), method = "ML")) +
            1608.1505086  #MAKING ARMA(0,0) = 0 for clarity
        vec[j + 1] <- temp
    }
    mylist[[i + 1]] <- vec  #put all vectors in the list
}

AIC.df <- do.call("rbind", mylist)  #combine all vectors into a matrix
rownames(AIC.df) <- c("AR0", "AR1", "AR2", "AR3", "AR4")
colnames(AIC.df) <- c("MA0", "MA1", "MA2", "MA3", "MA4")

AIC.df
```

```r
kable(AIC.df[, ], format = "pandoc", caption = "AICC's of ARMA(0,0) to ARMA(4,4)")
```

```r
fit_arima112 <- Arima(y.tr, order = c(1, 1, 2))
res.112 <- fit_arima112$residuals
fit_arima112
```

```r
fit_arima111 <- Arima(y.tr, order = c(1, 1, 1))
res.111 <- fit_arima111$residuals
fit_arima111
```

```r
fit.112.coef <- fit_arima112$coef
fit.112.se <- sqrt(diag(vcov(fit_arima112)))
fit.112.df <- data.frame(fit.112.coef, fit.112.se)
colnames(fit.112.df) <- c("COEFF", "SE")
rownames(fit.112.df) <- c("AR(1)", "MA(1)", "MA(2)")

fit.111.coef <- fit_arima111$coef
fit.111.se <- sqrt(diag(vcov(fit_arima111)))
fit.111.df <- data.frame(fit.111.coef, fit.111.se)
colnames(fit.111.df) <- c("COEFF", "SE")
rownames(fit.111.df) <- c("AR(1)", "MA(1)")

t1 <- fit.112.df
t2 <- fit.111.df
```

```r
t1 <- kable(fit.112.df, format = "latex")
t2 <- kable(fit.111.df, format = "latex")
cat(c("\\begin{table}[h] \\centering ", t1, "\\hspace{1cm} \\centering ",
    t2, "\\caption{ARMA(1,1,2) and ARMA(1,1,1)} \\end{table}"))
```

```
op = par(mfrow = c(3, 1))
plot(res.112, main = "Residuals Plot ARIMA(1,1,2)")
hist(res.112, main = "Histogram ARIMA(1,1,2)", breaks = 6, freq = 0)
qqnorm(res.112)

st.112 <- shapiro.test(res.112)
st112.p <- st.112$p.value

box.112 <- Box.test(res.112, lag = 18, type = c("Ljung-Box"),
    fitdf = 8)
box112.p <- box.112$p.value

ar.112 <- ar(res.112, aic = TRUE, order.max = NULL, method = c("yule-walker"))
ar112.order <- ar.112$order

tests.112 <- data.frame(st112.p, box112.p, ar112.order)
colnames(tests.112) <- c("Shapiro p", "Ljung-Box p", "Yule Walker Order")
kable(tests.112, caption = "Residual Tests for ARIMA(1,1,1)")
```

```
op = par(mfrow = c(3, 1))
plot(res.111, main = "Residuals Plot ARIMA(1,1,1)")
hist(res.111, main = "Histogram ARIMA(1,1,1)", breaks = 6, freq = 0)
qqnorm(res.111)

st.111 <- shapiro.test(res.111)
st111.p <- st.111$p.value

box.111 <- Box.test(res.111, lag = 18, type = c("Ljung-Box"),
    fitdf = 8)
box111.p <- box.111$p.value

ar.111 <- ar(res.111, aic = TRUE, order.max = NULL, method = c("yule-walker"))
ar111.order <- ar.111$order

tests.111 <- data.frame(st111.p, box111.p, ar111.order)
colnames(tests.111) <- c("Shapiro p", "Ljung-Box p", "Yule Walker Order")
kable(tests.111, caption = "Residual Tests for ARIMA(1,1,1)")
```

```
fit.arima112.s <- Arima(y.tr, order = c(1, 1, 2), seasonal = list(order = c(1,
    1, 0), period = 12))
res.112 <- fit.arima112.s$residuals
res.112s <- window(res.112, start = c(1968, 1), end = c(1979,
    12))
fit.arima112.s
```

```
fit.arima111.s <- Arima(y.tr, order = c(1, 1, 1), seasonal = list(order = c(1,
    1, 0), period = 12))
res.111 <- fit.arima111.s$residuals
res.111s <- window(res.111, start = c(1968, 1), end = c(1979,
```

```
    12))
fit.arima111.s
```

```
op = par(mfrow = c(3, 1))
plot(res.112s, main = "Residuals Plot SARIMA(1,1,2)x(1,1,0)")
hist(res.112s, main = "Histogram (1,1,2)x(1,1,0)", breaks = 6,
    freq = 0)
qqnorm(res.112s)

st.112 <- shapiro.test(res.112s)
st112.p <- st.112$p.value

box.112 <- Box.test(res.112s, lag = 18, type = c("Ljung-Box"),
    fitdf = 3)
box112.p <- box.112$p.value

ar.112 <- ar(res.112s, aic = TRUE, order.max = NULL, method = c("yule-walker"))
ar112.order <- ar.112$order

tests.112 <- data.frame(st112.p, box112.p, ar112.order)
colnames(tests.112) <- c("Shapiro p", "Ljung-Box p", "Yule Walker Order")
kable(tests.112, caption = "Residual Tests for ARIMA(1,1,2)x(1,1,0)")
```

```
op = par(mfrow = c(3, 1))
plot(res.111s, main = "Residuals Plot ARIMA(1,1,1)x(1,1,0)")
hist(res.111s, main = "Histogram (1,1,1)x(1,1,0)", breaks = 6,
    freq = 0)
qqnorm(res.111s)

st.111 <- shapiro.test(res.111s)
st111.p <- st.111$p.value

box.111 <- Box.test(res.111s, lag = 18, type = c("Ljung-Box"),
    fitdf = 3)
box111.p <- box.111$p.value

ar.111 <- ar(res.111s, aic = TRUE, order.max = NULL, method = c("yule-walker"))
ar111.order <- ar.111$order

tests.111 <- data.frame(st111.p, box111.p, ar111.order)
colnames(tests.111) <- c("Shapiro p", "Ljung-Box p", "Yule Walker Order")
kable(tests.111, caption = "Residual Tests for ARIMA(1,1,1)x(1,1,0)")
```

```
op = par(mfrow = c(1, 2))
acf(res.112^2, lag.max = 20, main = "")
pacf(res.112^2, lag.max = 20, main = "")
title("ARIMA(1,1,2)x(1,1,0) ACF/PACF of Residuals", line = -1,
    outer = TRUE)
```

```r
op = par(mfrow = c(1, 2))
acf(res.111^2, lag.max = 20, main = "")
pacf(res.111^2, lag.max = 20, main = "")
title("ARIMA(1,1,1)x(1,1,0) ACF/PACF of Residuals", line = -1,
    outer = TRUE)
```

```r
print(noquote(paste("AR Root")))
polyroot(c(1, -0.8962))
print(noquote(paste("MA Root")))
polyroot(c(1, -1.1441, 0.1441))
print(noquote(paste("SAR Root")))
polyroot(c(1, -0.512))
```

```r
op = par(mfrow = c(1, 2))
plot.roots(NULL, polyroot(c(1, -0.8962)), main = "Roots of AR part")
plot.roots(NULL, polyroot(c(1, -1.1441, 0.1441)), main = "Roots of MA part")
```

```r
print(noquote(paste("AR Root")))
polyroot(c(1, -0.517))
print(noquote(paste("MA Root")))
polyroot(c(1, -0.718))
print(noquote(paste("SAR Root")))
polyroot(c(1, -0.511))
```

```r
op = par(mfrow = c(1, 2))
plot.roots(NULL, polyroot(c(1, -0.517)), main = "Roots of AR part")
plot.roots(NULL, polyroot(c(1, -0.718)), main = "Roots of MA part")
```

```r
variance <- c(format(round(fit_arima314$sigma2, digits = 8),
    scientific = F), format(round(fit_arima414$sigma2, digits = 8),
    scientific = F), format(round(fit.arima112.s$sigma2, digits = 8),
    scientific = F), format(round(fit.arima111.s$sigma2, digits = 8),
    scientific = F))

log_like <- c(fit_arima314$loglik, fit_arima414$loglik, fit.arima112.s$loglik,
    fit.arima111.s$loglik)


model.AICC <- c(fit_arima314$aicc, fit_arima414$aicc, fit.arima112.s$aicc,
    fit.arima111.s$aicc)


model.bic <- c(fit_arima314$bic, fit_arima414$bic, fit.arima112.s$bic,
    fit.arima111.s$bic)

shapiro <- c(round(st314.p.sub, digits = 3), round(st414.p.sub,
    digits = 3), round(st112.p, digits = 3), round(st111.p, digits = 3))
```

```r
BL <- c(round(box314.p.sub, digits = 3), round(box414.p.sub,
    digits = 3), round(box112.p, digits = 3), round(box111.p,
    digits = 3))

model.summary <- data.frame(variance, log_like, model.AICC, model.bic,
    shapiro, BL)
row.names(model.summary) <- c("(3,1,4)x(1,1,0)", "(4,1,4)x(1,1,0)",
    "(1,1,2)x(1,1,0)", "(1,1,1)x(1,1,0)")
colnames(model.summary) <- c("Variance", "Log Likelihood", "AICC",
    "BIC", "Shapiro", "Box-Ljung")

kable(model.summary, "markdown", caption = "Model Summary")
```

```r
# Casting data to numeric vectors to make handling the
# forecast data transformations easier
mauna.train.cast <- as.numeric(mauna.train)
mauna.cast.log <- log(mauna.train.cast)
mauna.test.cast <- as.numeric(mauna.test)
```

```r
pred.tr <- predict(fit.arima111.s, n.ahead = 12)
Upper.tr = pred.tr$pred + 2 * pred.tr$se  # upper bound for the C.I. for transformed data
Lower.tr = pred.tr$pred - 2 * pred.tr$se  # lower bound

# Forecasting Transformed Data
ts.plot(mauna.cast.log, xlim = c(140, length(mauna.cast.log) +
    12), ylim = c(5.78, max(Upper.tr)), main = "Predicted Values on Transformed Data",
    ylab = "CO2 (ppm)")  #plot y.tr and forecast
lines((length(mauna.cast.log) + 1):(length(mauna.cast.log) +
    12), pred.tr$pred + 1.96 * pred.tr$se, lty = 2, col = "blue")
lines((length(mauna.cast.log) + 1):(length(mauna.cast.log) +
    12), pred.tr$pred - 1.96 * pred.tr$se, lty = 2, col = "blue")
points((length(mauna.cast.log) + 1):(length(mauna.cast.log) +
    12), pred.tr$pred, col = "red")
points((length(mauna.cast.log) + 1):(length(mauna.cast.log) +
    12), log(mauna.test.cast), pch = "*")
legend("topleft", bty = "n", col = c("blue", "red", "black"),
    c("Conf. Int", "Predicted Values", "Actual Values"), pch = c("-",
        "o", "*"))
```

```r
pred.y1 <- predict(fit.arima111.s, n.ahead = 12)
pred.y <- exp(pred.y1$pred)
se.y <- exp(pred.y1$se)  #Undoing the log transform so predicted points are on proper scale
Upper.y = exp(Upper.tr)
Lower.y = exp(Lower.tr)


# Going back to Original Data Scale
ts.plot(mauna.train.cast, xlim = c(140, length(mauna.train.cast) +
    12), ylim = c(exp(5.78), max(Upper.y)), ylab = "CO2 (ppm)",
```

```
    main = "Predicted Values on Original Data Scale")
lines((length(mauna.train.cast) + 1):(length(mauna.train.cast) +
    12), pred.y + 1.96 * se.y, lty = 2, col = "blue")
lines((length(mauna.train.cast) + 1):(length(mauna.train.cast) +
    12), pred.y - 1.96 * se.y, lty = 2, col = "blue")
points((length(mauna.train.cast) + 1):(length(mauna.train.cast) +
    12), pred.y, col = "red")
points((length(mauna.train.cast) + 1):(length(mauna.train.cast) +
    12), mauna.test.cast, pch = "*")
legend("topleft", bty = "n", col = c("blue", "red", "black"),
    c("Conf. Int", "Predicted Values", "Actual Values"), pch = c("-",
        "o", "*"))
```
