Forecasting Concentration of Carbon Dioxide (CO_2) in Atmosphere Using Seasonal ARIMA Model

1. Akshta (UIN: 652062814)
2. Laavanya Ganesh (UIN: 654324917)
3. Nabanita Ghosh (UIN: 664694644)
4. Pujitha Prasanth Hegde (UIN: 659660537)
5. Vineet Kumar Singh (UIN: 6544892895)

TABLE OF CONTENTS

1. ABSTRACT

As an ozone harming substance, carbon dioxide (CO2) is one of the main impetuses behind a worldwide temperature alteration. Since the start of the modern upset, barometrical CO2 levels have risen over 40% and give no sign of backing off. This venture uses Seasonal Auto-Regressive Integrated Moving Average (SARIMA) models to precisely conjecture environmental carbon dioxide fixations 10 months into what's to come. It is our expectation that these gauges might be valuable for envisioning other meteorological marvels, for example, cataclysmic climate occasions and worldwide temperature rises.

2. INTRODUCTION

A worldwide temperature alteration is a steady climatic change including an expansion in the normal temperature of the Earth& its seas. It is essentially an issue because of the nearness of overabundance measure of CO_2 in the climate, other than other nursery gases, for example, CH_4, water vapor, NO_2 and so forth which are discharged by copying petroleum products, intemperate horticulture, arrive clearing, & other human exercises. Regardless, the measure of CO_2 puts us at a higher hazard due to irreversible changes would keep on accumulating unabated in the climate. Investigate demonstrates the impacts of a worldwide temperature alteration to raise ocean levels which are brought on by liquefying of polar ice tops and furthermore the successive event of tempests and other climatic changes. Thus, it is basic to precisely anticipate the measure of CO_2 levels in the climate later.

The target of this venture is to precisely figure the centralization of environmental carbon dioxide 10 months into the future by building a model that best fits the occasional autoregressive coordinated moving normal model and after that anticipating the estimates for what's to come.

2.1 DATA DESCRIPTION

The dataset used to build the seasonal ARIMA model is taken from the National Oceanic & Atmospheric Administration website. Its name is Mauna Loa CO_2 monthly mean data. It contains the monthly mean CO_2 mole fraction determined from daily averages. The mole fraction of CO_2 is expressed in parts per million and is calculated by the number of molecules of CO_2 in per million molecules of dried air with water vapor removed.

2.2 SUMMARIZED ANALYSIS

The whole procedure of data preparation, cleaning, change, model building and predicting will be performed utilizing RStudio. The last 10 perceptions from the dataset will be kept aside while building the model. The autocorrelation function (ACF) and partial autocorrelation function (PACF) of the series will be analyzed to recognize reasonable models that meet the theory of regular ARIMA. Finally, the best model is chosen to foretell the convergence of air CO2 for the following 10 months. These last results will then be thought about against the 10 records, which were kept aside, to check the exactness of the anticipated information.

2.3 Results

SARIMA time series models provide a viable framework for forecasting atmospheric carbon dioxide concentrations. Model provides incredibly accurate forecasts of atmospheric CO2 concentrations 10 months into the future.

3. ANALYSIS

3.1 EXPLORATORY DATA ANALYSIS

We begin by loading the raw data into R and plotting the time series. We observed an upward trend and seasonality in the data.

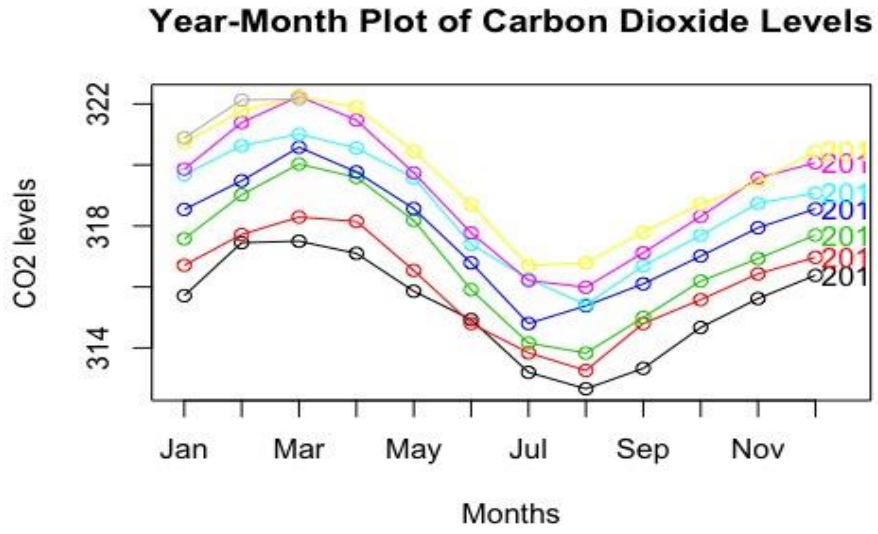

Figure 1: Seasonality plot representing the level of carbon-di-oxide

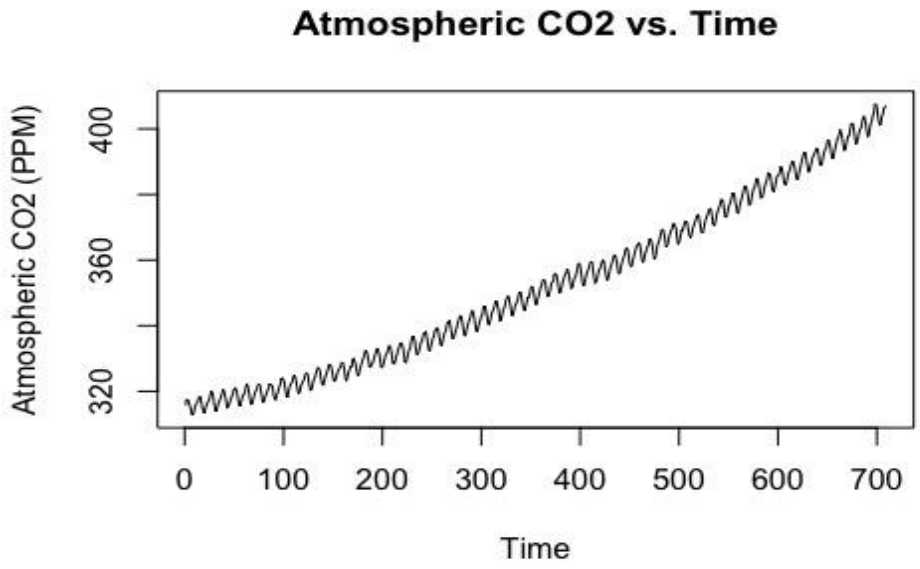

Figure 2: Plot representing an upward trend in the level of carbon-di-oxide

3.2 DATA TRANSFORMATIONS

Along with a strong positive trend, the data was found to be yearly seasonal with the variance increasing slightly over time. Therefore, we perform transformation to make it stationary. We impose a linear model on the time series to check the existence of the positive trend.

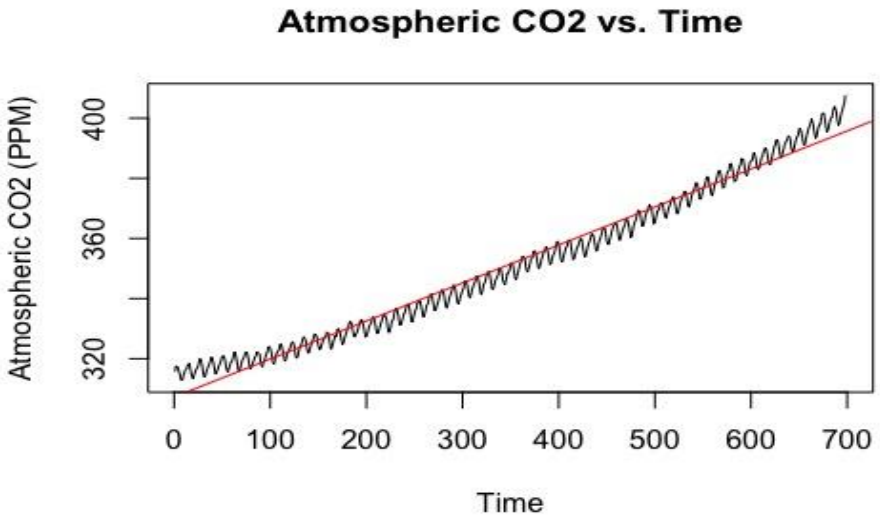

Figure 3: Plot representing an upward trend in the level of carbon-di-oxide with regression fit

Now we differentiate at lag 1 and find that unlike towards the end, the regression line almost perfectly fits the data indicating differencing at Lag=1 for trend removal. Differencing at Lag=12 is done to remove seasonality. To stabilize variance, we perform the Box-Cox Power transformation, which gives us λ=0.

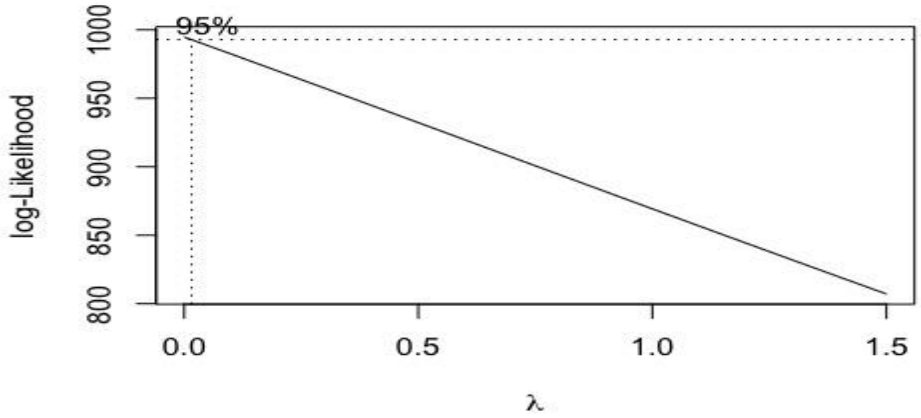

Figure 4: Plot representing the relationship between different learning rates and log-likelihood

Hence, we take the Log transformation and perform the time series fitted with the regression line. Model variance decreases by 0.005 thus, justifying the differencing. We now difference at lag = 12 to remove the seasonal component of the time series.

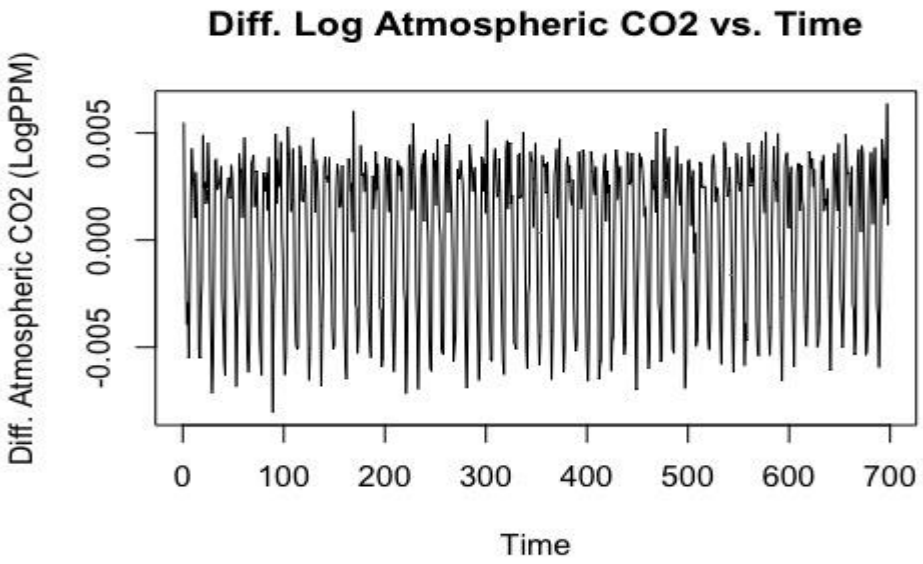

Figure 5: Plot representing the level of carbon-di-oxide after differentiating at lag 12

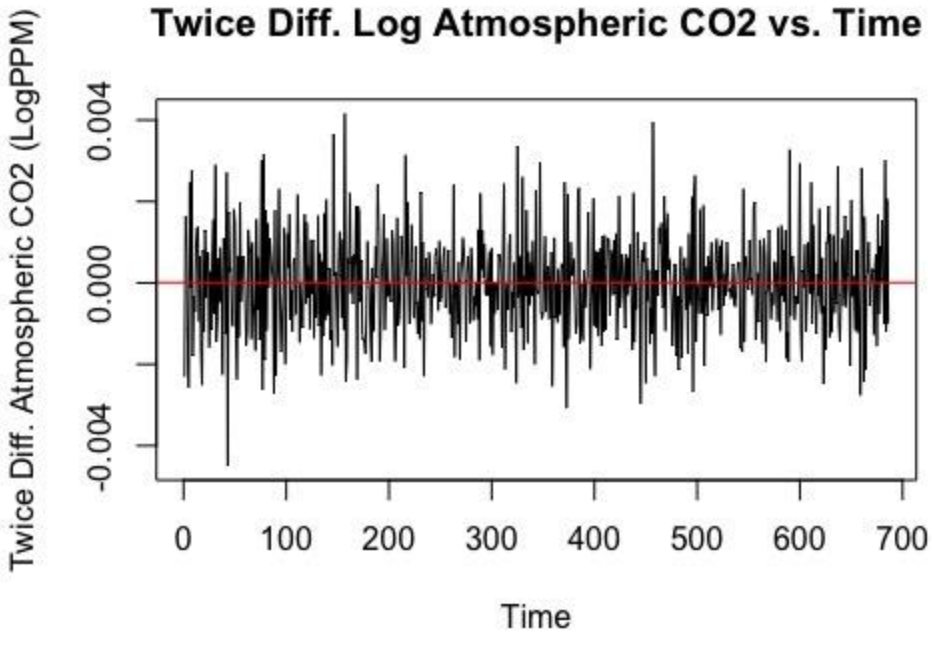

Figure 6: Plot representing the level of carbon-di-oxide after double differentiating

The variance decreases further. The mean of the time series is added in red. The model also appears to be stationary, as neither mean nor variance appears to be dependent on time. To confirm the stationarity of the series, we perform Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test in R under the null hypothesis that the series is stationary.

```
        KPSS Test for Level Stationarity

data:  Transform.log.1.12
KPSS Level = 0.018087, Truncation lag parameter = 6, p-value = 0.1

Warning message:
In kpss.test(Transform.log.1.12) : p-value greater than printed p-value
>
```

Figure 7: KPSS Test for Stationarity

From Figure 7, we observe that this test yields a p-value > 0.05, so we fail to reject the null hypothesis and conclude that the model is indeed stationary.

3.3 ACF and PACF Analysis

We, now analyze its ACF and PACF plots to identify the AR, MA, SAR, and SMA orders in the SARIMA model.

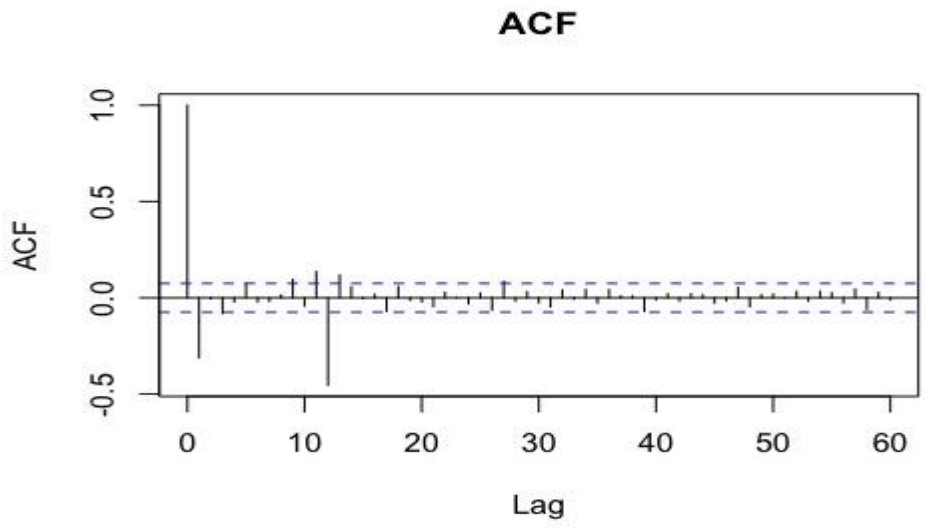

Figure 8: Auto-correlated function after the transformation

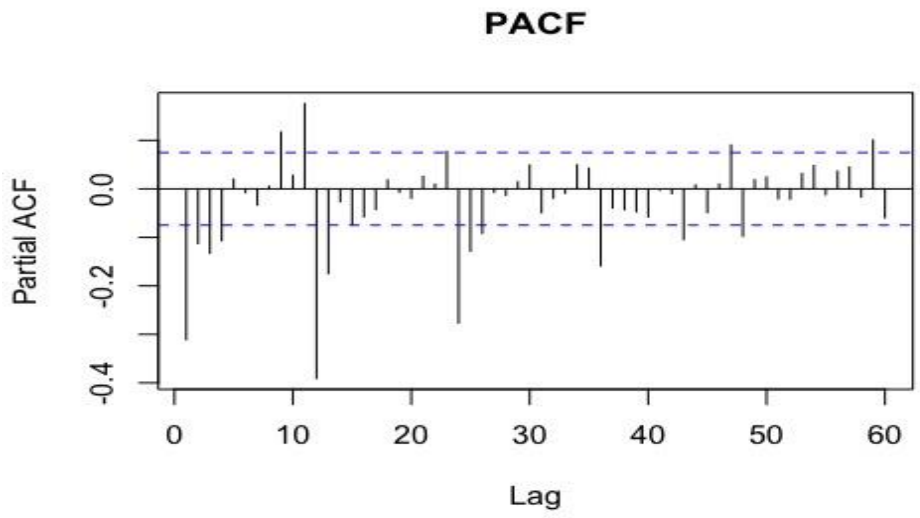

Figure 9: Partial Auto-correlated function after the transformation

We examine the seasonal components, at lags l = 12n, n ∈ N. We observe that the ACF cuts off after lag 12, while the PACF decays exponentially at multiples of 12 lag values. This leads us to consider that SAR = 0 and SMA = 1.

From ACF plot, it can be observed that the ACF cuts off at lag 1. Hence MA = 1. From the PACF plot, we observe that the PACF gradually decays over a period of time. Hence AR = 0. Therefore, we test the model for SARIM (0, 1, 1). Here d = 1 since we are differentiating at lag 1.

From ACF plot, it can also be observed that the ACF gradually decays over a period of time. Hence MA = 0. From the PACF plot, we observe that the PACF cuts off at lag 3. Hence AR = 3. Therefore we test the model SARIM (3, 1, 0). Here d = 1 since we are differentiating at lag 1.

We consider the following two models:

- SARIMA (0, 1, 1) x $(0, 1, 1)_{12}$
  – AICc = −7695.52.12 BIC = −7681.96.57
  – $r_{12}rX_t = (1 - 3.9B)\ (1 - 0.89B^{12})\ Z_t$

- SARIMA (3, 1, 0) x $(0, 1, 1)_{12}$
  – AICc = −7691.88 BIC = −7669.32
  – $(1 + 0.36B + 0.15B^2 + 0.1B^3\ )r_{12}rX_t = (1 - 0.89B^{12})\ Z_t$

3.4 MODEL 1 DIAGNOSTICS

For model 1, we get the following equation

$$- r_{12} r X_t = (1 - 3.9B)(1 - 0.89B^{12}) Z_t$$

We perform diagnosis for the SARIMA (0, 1, 1) x $(0, 1, 1)_{12}$ model. We begin by plotting the residuals of this model

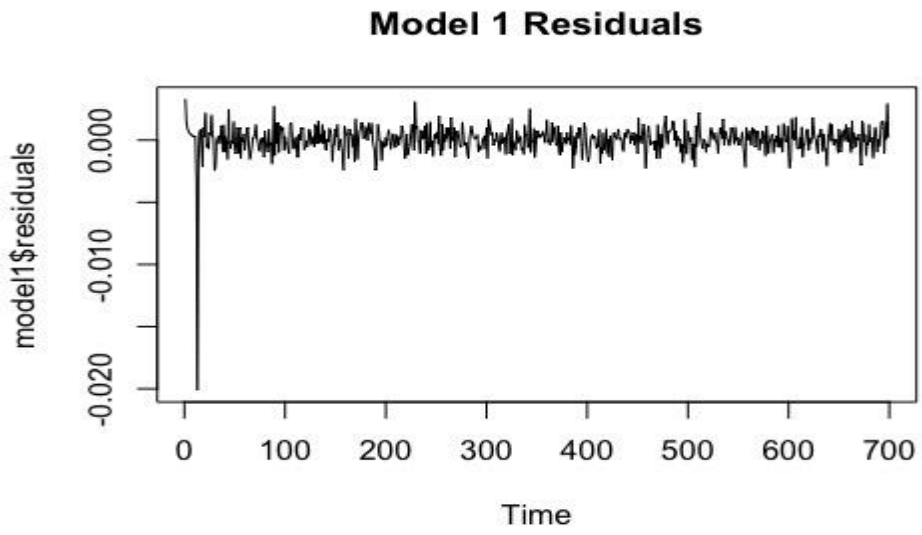

Figure 10: Residuals plot of model 1

The residuals appear to resemble white noise. The residuals fail the Shapiro-Wilks normality test with the outlier, put pass the test when it is removed.

```
### Model 1 Diagnostics
> plot (model1$residuals, main="Model 1 Residuals")
> shapiro.test(model1$residuals)

Shapiro-Wilk normality test
data:  model1$residuals
W = 0.68877, p-value < 2.2e-16
> shapiro.test (model1$residuals [14:709])

Shapiro-Wilk normality test
data:  model1$residuals [14:709]
W = 0.99606, p-value = 0.08314
```

```
> hist(model1$residuals, xlim = c(-0.005,0.005), main = "Model 1 Residuals", breaks = 50)
> qqnorm(model1$residuals)
> qqline(model1$residuals)
```

We now plot the density and the QQ-Plot of the residuals.

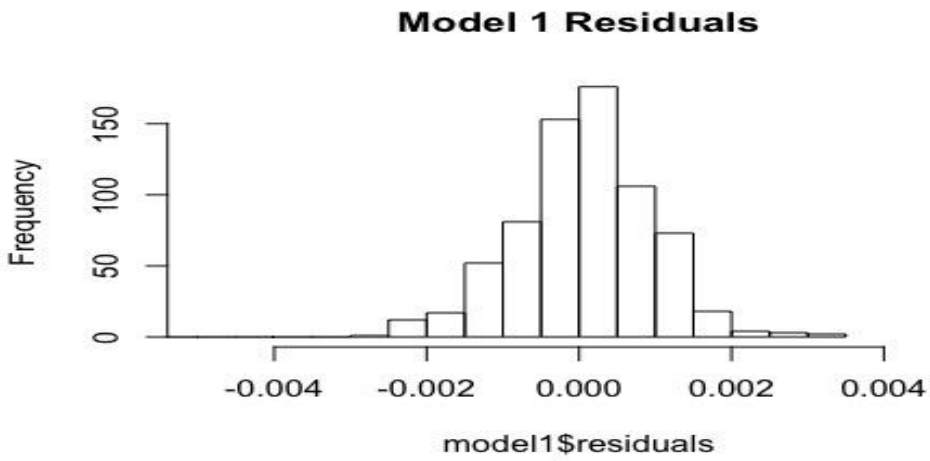

Figure 11: Residuals histogram plot of model 1

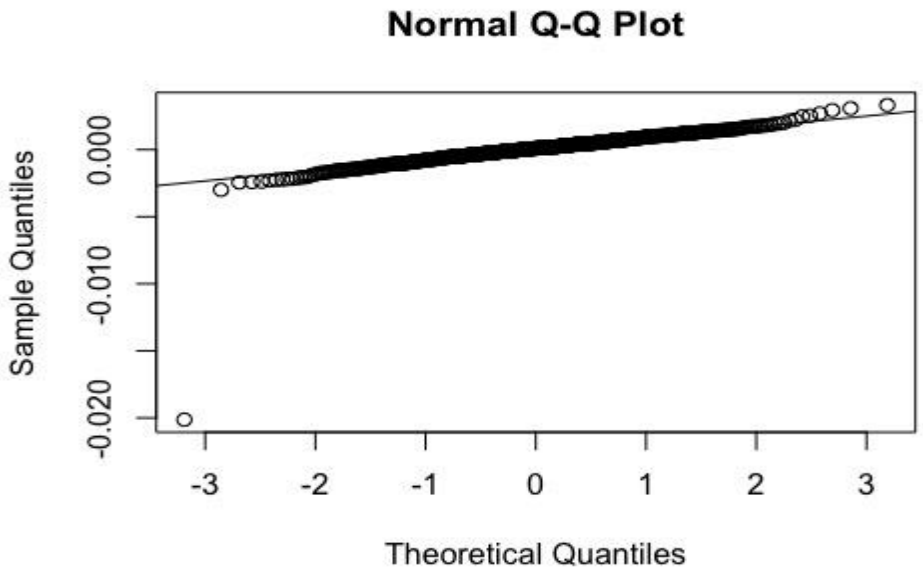

Figure 12: Q-Q plot of model 1

The residuals appear to be almost normally distributed as per the graph. We now perform the Box-Pierce, Ljung-Box, and McLeod-Li tests on the residuals.

```
## [1] "Box-Pierce"
[1] "Box-Pierce"
> Box.test(model1$residuals, lag = 26, type = "Box-Pierce", fitdf=2)

Box-Pierce test
data:  model1$residuals
X-squared = 30.215, df = 24, p-value = 0.1777

> paste("Ljung-Box")
[1] "Ljung-Box"
> Box.test(model1$residuals, lag = 26, type = "Ljung-Box", fitdf=2)

Box-Ljung test
data:  model1$residuals
X-squared = 30.672, df = 24, p-value = 0.1635

> paste("McLeod-Li")
[1] "McLeod-Li"
> Box.test((model1$residuals)^2, lag=26-2, type="Ljung-Box")

Box-Ljung test
data:  (model1$residuals)^2
X-squared = 1.1536, df = 24, p-value = 1
```

This model passes all adequacy tests at an α value of 0.05 significance level. We now plot the ACF and PACF of the residuals to ensure they resemble white noise.

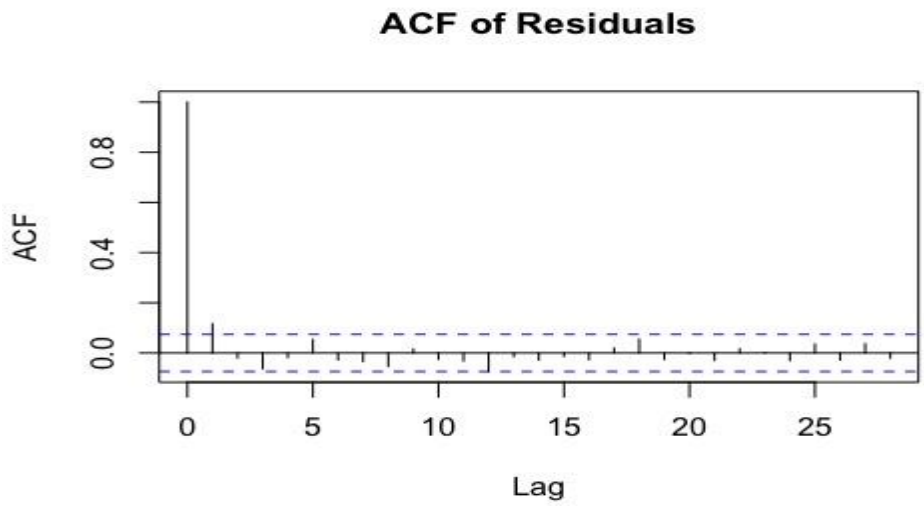

Figure 13: Auto-correlated function of model 1

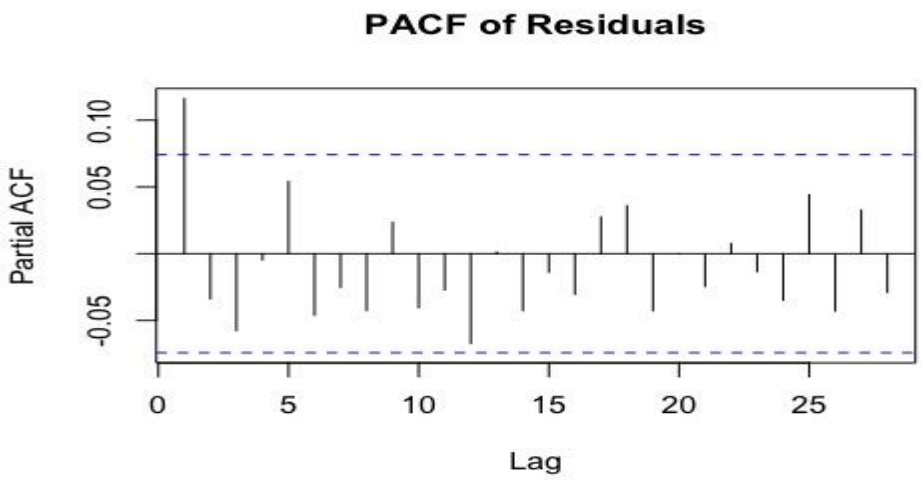

Figure 14: Partial Auto-correlated function of model 1

The ACFs and PACFs extend slightly beyond the confidence intervals at lag 12, but resembles white noise. Based on the results of this test, we consider this model adequate for forecasting.

3.5 MODEL 2 DIAGNOSTICS

For model 1, we get the following equation

$$-(1 + 0.36B + 0.15B2 + 0.1B3)\ r12rXt = (1 - 0.89B12)\ Z_t$$

We now check the accuracy of the second model. We begin by plotting the model residuals.

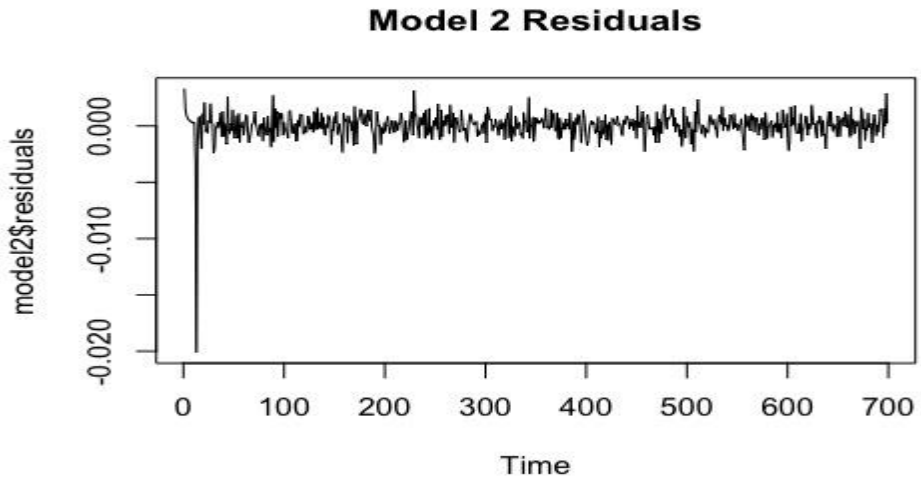

Figure 15: Residuals plot of model 2

Like Model 1, we have an outlier at t = 13. Besides this single observation, the residuals resemble white noise. The model fails Shapiro-Wilks normality test for the unmodified residuals, but indicates normality with the outlier removed.

```
##
> plot (model2$residuals, main="Model 2 Residuals")
> shapiro.test(model2$residuals)

Shapiro-Wilk normality test
data:  model2$residuals
W = 0.68876, p-value < 2.2e-16

> shapiro.test (model2$residuals [14:709])

Shapiro-Wilk normality test
data:  model2$residuals [14:709]
W = 0.99619, p-value = 0.09628
```

We now plot the density of the residuals and construct a QQ-Plot.

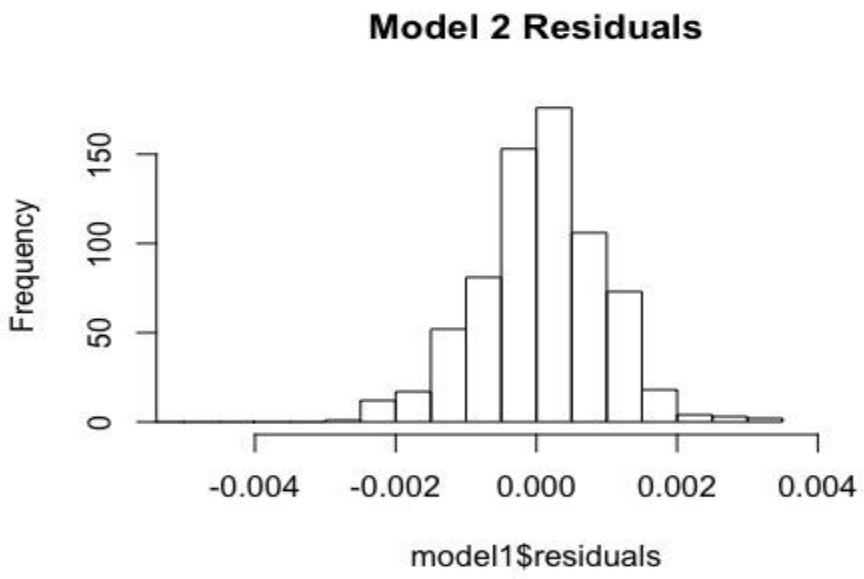

Figure 16: Residuals histogram plot of model 2

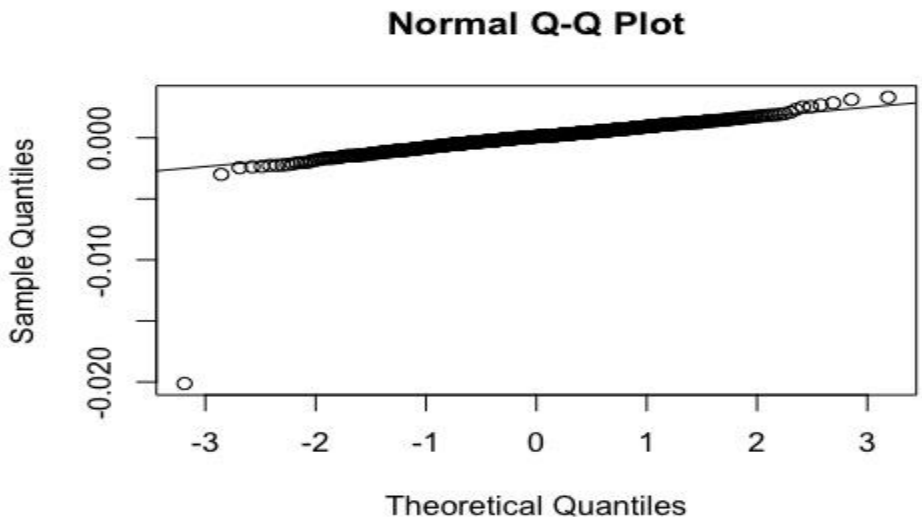

Figure 17: Q-Q plot of model 2

The histogram and QQ-Plot also show that the residuals are distributed normally. We now perform the Box-Pierce, Ljung-Box, and McLeod-Li tests on the residuals.

[1] "Box-Pierce"
> Box.test(model2$residuals, lag = 26, type = "Box-Pierce", fitdf=5)

Box-Pierce test
data: model2$residuals
X-squared = 25.838, df = 21, p-value = 0.2127

> paste("Ljung-Box")
[1] "Ljung-Box"
> Box.test (model2$residuals, lag = 26, type = "Ljung-Box", fitdf=5)

Box-Ljung test
data: model2$residuals
X-squared = 26.269, df = 21, p-value = 0.1964

> paste("McLeod-Li")
[1] "McLeod-Li"
> Box.test ((model2$residuals)^2, lag=26-4, type="Ljung-Box")

Box-Ljung test
data: (model2$residuals)^2
X-squared = 1.1053, df = 22, p-value = 1

The model passes all accuracy tests at an α value of 0.05 significance level. Finally, we check that the ACF and PACF of the residuals and observe it to resemble white noise.

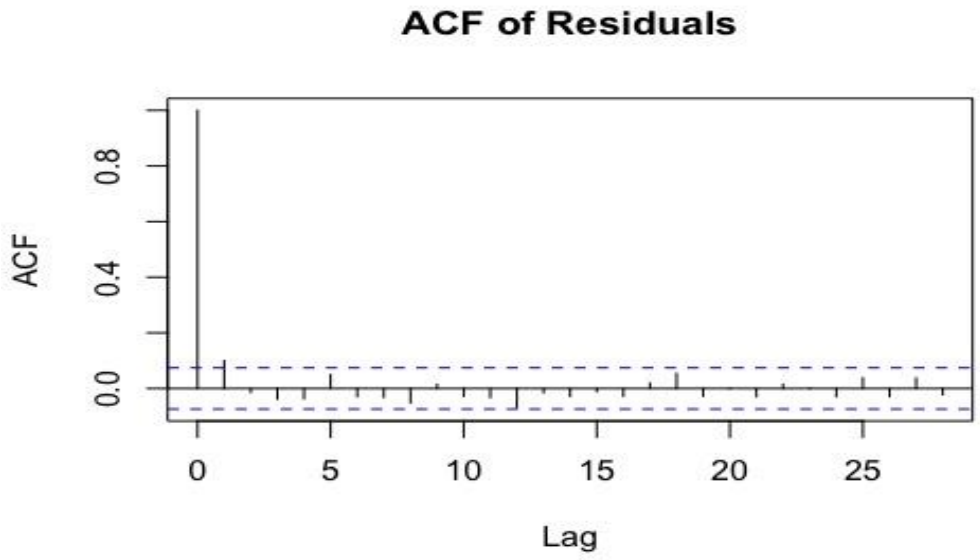

Figure 18: Auto-correlated function of model 2

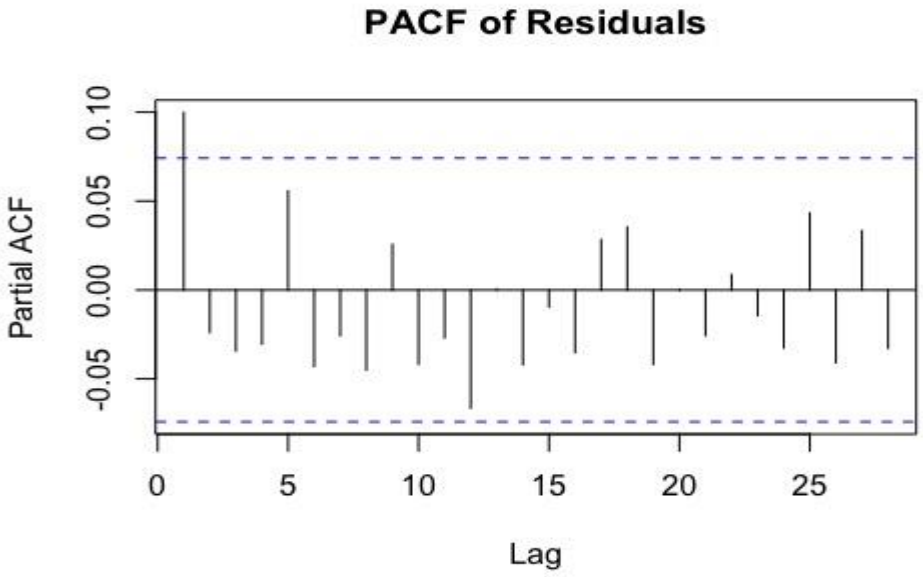

Figure 19: Partial Auto-correlated function of model 2

Like before, there are significant spikes at lag 12 in both plots, but considered acceptable. Like Model 1, Model 2 is accepted for forecasting.

3.6 CHOOSING THE BEST MODEL

From the histogram of Residuals of Model 1 and Model 2, we observe that both the models pass the residuals tests and are normally distributed. The AIC and BIC values of Model 1 and Model 2 are as follows:

Model 1:	**Model 2:**
AIC = -7695.55	AIC = -7691.97
BIC = -7681.96	BIC = -7669.32

The lower the AIC value, the better the model. By comparing the AIC values of both models, we find that model 1 has a lower AIC value and is thus selected as the best model.

3.7 FORECASTING

Figure 20 depicts the following:

- Forecasted observations - in red
- Actual observations - as asterisks.
- 95% CI of the forecasts - blue dotted lines
- Forecasts lie very close to the realized observations

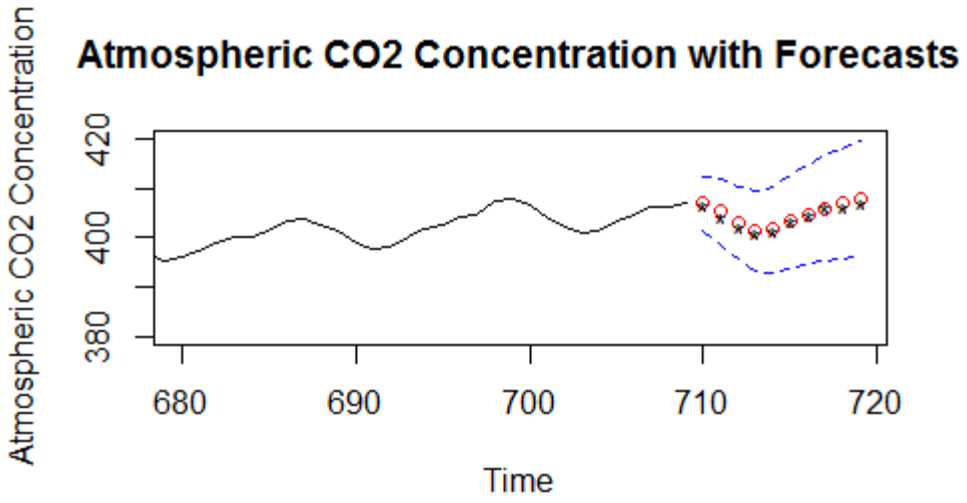

Figure 20: Plot representing the forecast of next 10 months into future

As observed, the forecasts lie very close to the actual observations. Hence, we conclude that the model provides good accuracy in forecasting the concentration level of CO_2.

4. CONCLUSION

- SARIMA time series models provide a viable framework for forecasting atmospheric carbon dioxide concentrations.
- Models meet the assumptions of the Box-Jenkins methodology for time series forecasting.
- This model is given by: $\nabla 12 \nabla Xt = (1 - 3.9B)(1 - 0.89B12)\, Z_t$
- Model provides incredibly accurate forecasts of atmospheric CO2 concentrations 10 months into the future.

5. REFERENCES

- Warrick, Joby. 2014. The Washington Post - CO2 levels in atmosphere rising at dramatically faster rate, U.N. report warns. Link.
- Brockwell, PJ., Davis, RA. 2016. Introduction to Time Series and Forecasting.
- Hyndman, Rob J. 2012. Constants and ARIMA models in R. Link.
- Shumway, Robert H., Stoffer, David S. Time Series Analysis and Its Applications with R Examples. 3rd Edition. 2010.
- The National Oceanic and Atmospheric Administration
- The R Project for Statistical Computing

TECHNICAL APPENDIX

The R code used to create the analysis is listed below.

```
library(stats)
install.packages("dplyr")
install.packages("tseries")
install.packages("forecast")

# Import Data
ts <- read.table("CO2_data.txt", skip = 2, header = F, sep = "")
colnames <- c("Year","Month","DecDate","Mean_Co2_Mole_Fraction","Interpolated_Values","Seasonal_Trend","Days")
colnames(ts) <- colnames
View(ts)

library(dplyr)
ts_monthly <- ts %>% group_by(Year,Month) %>%summarize(average = mean(Interpolated_Values))

# Convert to time series data
co2 <- ts(ts_monthly[,3], start=c(2010,1), end=c(2017,3), freq=12)

library(forecast)
par(mfrow=c(1,2))
dev.off()
monthplot(co2, xlab = "Months", ylab = "CO2 levels", main = "Year-Month Plot of Carbon Dioxide Levels") # by month
dev.off()
seasonplot(co2, year.labels=T, col=seq(1:11),xlab = "Months", ylab = "CO2 levels", main = "Year-Month Plot of Carbon Dioxide Levels" ) # by season

# Remove exraneous featurs
ts_V1 <- ts$Interpolated_Values

# Plot original series
plot.ts(ts_V1, xlab= "Time", ylab = "Atmospheric CO2 (PPM)", main = "Atmospheric CO2 vs. Time")

# Copy series to new variable
orig <- ts_V1

# Reserve last 10 observations
Subsequent10 <- orig[700:709]
```

```
# Convert vector to time series object
ts_V2 <- as.ts(ts_V1, start=c(1958,3), frequency=12)
ts_V2 <- ts_V1[1:699]

# Plot with regression line
RegLine <- lm(ts_V2 ~ as.numeric(1:length(ts_V2)))
plot.ts(ts_V2, xlab="Time", ylab = "Atmospheric CO2 (PPM)",
      main = "Atmospheric CO2 vs. Time")
abline(RegLine, col = "red")

# Find lambda for Box-Cox transformation
library(MASS)
BoxCoxTransform <- boxcox(ts_V2~as.numeric(1:length(ts_V2)), lambda = seq(0,1.5,1/10))
Transformed <- BoxCoxTransform$x[which.max(BoxCoxTransform$y)]

# Transform data, plot with regression line
Transform.log<- log(ts_V2)
RegLine2 <- lm(Transform.log~as.numeric(1:length(Transform.log)))
plot.ts(Transform.log, xlab="Time", ylab = "Atmospheric CO2 (LogPPM)", main = "Log Atmospheric CO2 vs. Time")
abline(RegLine2, col = "red")

# Difference at lag1, compare variance
Transform.log.1 <- diff(Transform.log,1)
plot.ts(Transform.log.1, xlab="Time", ylab = "Diff. Atmospheric CO2 (LogPPM)", main = "Diff. Log Atmospheric CO2 vs. Time")
var.Transform.log <- var(Transform.log, na.rm = T)
var.Transform.log.1 <- var(Transform.log.1, na.rm =T)

# Difference at lag12, compare variance
Transform.log.1.12 <- diff(Transform.log.1,12)
plot.ts(Transform.log.1.12, xlab="Time", ylab = "Twice Diff. Atmospheric CO2 (LogPPM)", main = "Twice Diff. Log Atmospheric CO2 vs. Time")
abline(h=mean(Transform.log.1.12, na.rm = T), col = "red")
var.Transform.log.1.12 <- var(Transform.log.1.12, na.rm =T)

# KPSS Test for stationarity
library(tseries)
kpss.test(Transform.log.1.12)
```

```
# Plot ACF, PACF
acf <- acf(Transform.log.1.12, type = "correlation", plot = T, na.action = na.pass, lag.max=12*5, main ="ACF")
pacf <- acf(Transform.log.1.12, type = "partial", plot = T, na.action = na.pass, lag.max=12*5, main = "PACF")

# Build two appropriate models
library(forecast)
model1 <- Arima(Transform.log, order = c(0,1,1), seasonal = list(order = c(0,1,1), period = 12))
model1

model2 <- Arima(Transform.log, order = c(3,1,0), seasonal = list(order = c(0,1,1), period = 12))
model2

# Model 1 Diagnostics
plot(model1$residuals, main="Model 1 Residuals")
shapiro.test(model1$residuals)
shapiro.test(model1$residuals[14:709])
hist(model1$residuals, xlim = c(-0.005,0.005), main = "Model 1 Residuals", breaks = 50)
qqnorm(model1$residuals)
qqline(model1$residuals)
paste("Box-Pierce")
Box.test(model1$residuals, lag = 26, type = "Box-Pierce", fitdf=2)

paste("Ljung-Box")
Box.test(model1$residuals, lag = 26, type = "Ljung-Box", fitdf=2)
paste("McLeod-Li")
Box.test((model1$residuals)^2, lag=26-2, type="Ljung-Box")
acf(model1$residuals, na.action=na.pass, main = "ACF of Residuals")
pacf(model1$residuals, na.action = na.pass, main = "PACF of Residuals")

# Model 2 Diagnostics
plot(model2$residuals, main="Model 2 Residuals")
shapiro.test(model2$residuals)
shapiro.test(model2$residuals[14:709])
hist(model1$residuals, xlim = c(-0.005,0.005), main = "Model 2 Residuals", breaks = 50)
qqnorm(model2$residuals)
qqline(model2$residuals)

paste("Box-Pierce")
Box.test(model2$residuals, lag = 26, type = "Box-Pierce", fitdf=5)
paste("Ljung-Box")
Box.test(model2$residuals, lag = 26, type = "Ljung-Box", fitdf=5)
paste("McLeod-Li")
```

```
Box.test((model2$residuals)^2, lag=26-4, type="Ljung-Box")
acf(model2$residuals, na.action=na.pass, main = "ACF of Residuals")
pacf(model2$residuals, na.action = na.pass, main = "PACF of Residuals")

# Forecasting with Model 2
predicted <- predict(model2, n.ahead = 10)
predicted.orig <- exp(predicted$pred)
predicted.se <- exp(predicted$pred) * predicted$pred * predicted$se
plot.ts(orig, xlim = c(680,length(orig)+10), ylim = c(380,420), ylab = "Atmospheric CO2 Concentration", main =
"Atmospheric CO2 Concentration with Forecasts")
points((length(orig)+1):(length(orig)+10),predicted.orig, col="red")
points((length(orig)+1):(length(orig)+10),Subsequent10, pch = "*")
lines((length(orig)+1):(length(orig)+10),predicted.orig+1.96*predicted.se,lty=2, col="blue")
lines((length(orig)+1):(length(orig)+10),predicted.orig-1.96*predicted.se,lty=2, col="blue")
```