Forecasting Carbon Dioxide Emissions

Brooks Piper

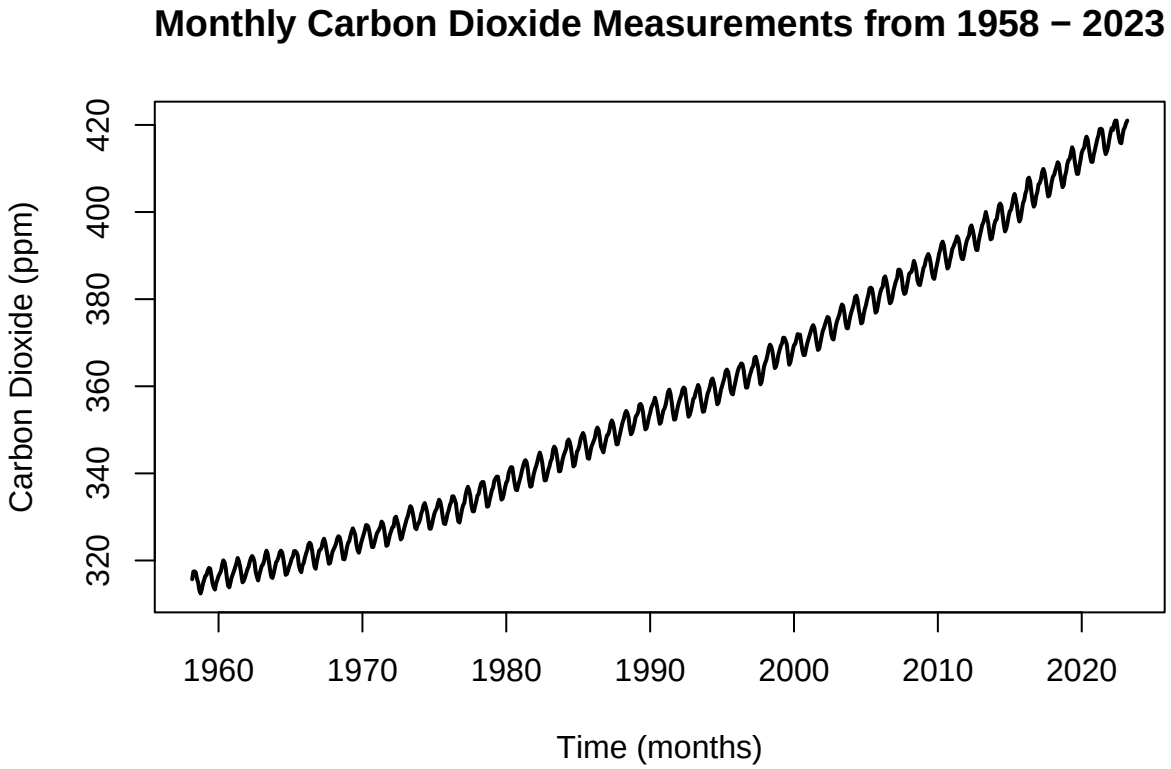
2025-03-03

Contents

Abstract

This project models Monthly Carbon Dioxide Levels at Mauna Loa for forecasting. Using seasonal and non-seasonal differencing, we fit a SARIMA model to data from 1958–2020. Our forecasts predict a 1.1% rise in CO_2 levels by 2023 and a 5.2% increase by 2030, highlighting the accelerating growth of atmospheric carbon dioxide and the urgency of addressing it.

Introduction



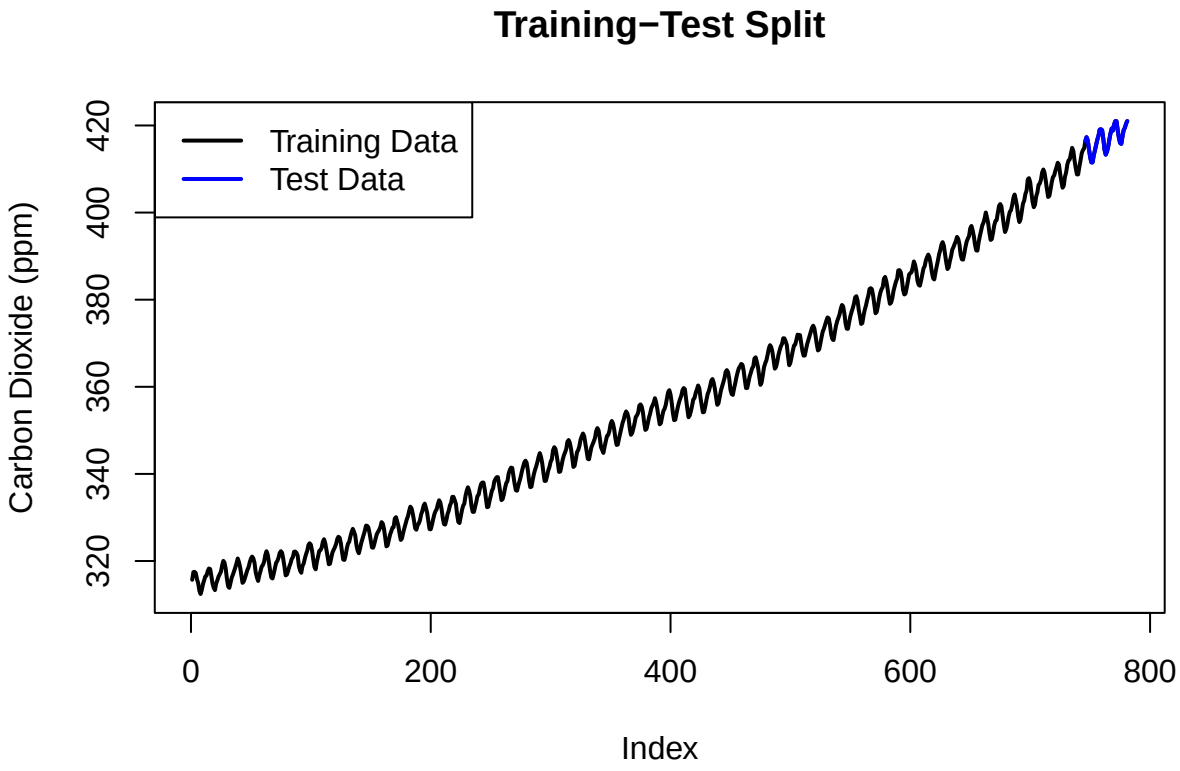
Greenhouse gases drive climate change, with carbon dioxide (CO_2) accounting for over 80% of U.S. emissions. As the primary contributor to rising temperatures, sea levels, and ecosystem disruptions—largely from fossil fuel combustion—its continued increase is inevitable (as seen in the chart above). However, understanding how CO_2 levels will grow is crucial. While human behavior is unpredictable, time series modeling allows us to analyze the trend-like and seasonal patterns of emissions.

In this project, we utilize the R coding language to examine the Monthly Carbon Dioxide Levels at Mauna Loa from 1958–2023. Using differencing to remove trend and seasonality, we fit multiple SARIMA models, conduct diagnostic checks, and forecast future CO_2 levels to better understand its trajectory.

Data Analysis

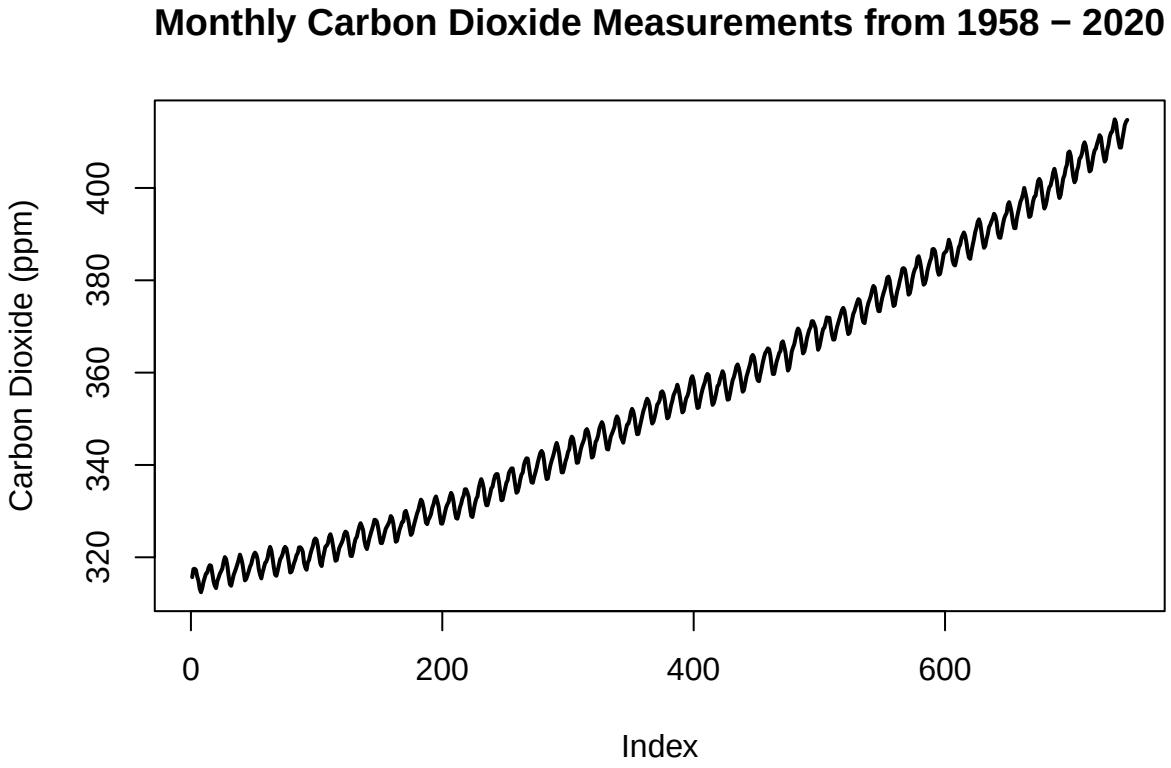
Training and Testing Set Split

As was previously mentioned, this data set spans from 1958 to 2023. We will create a cutoff at the beginning of 2020, reserving 745 months for training and 36 for testing. This split has been visualized below.

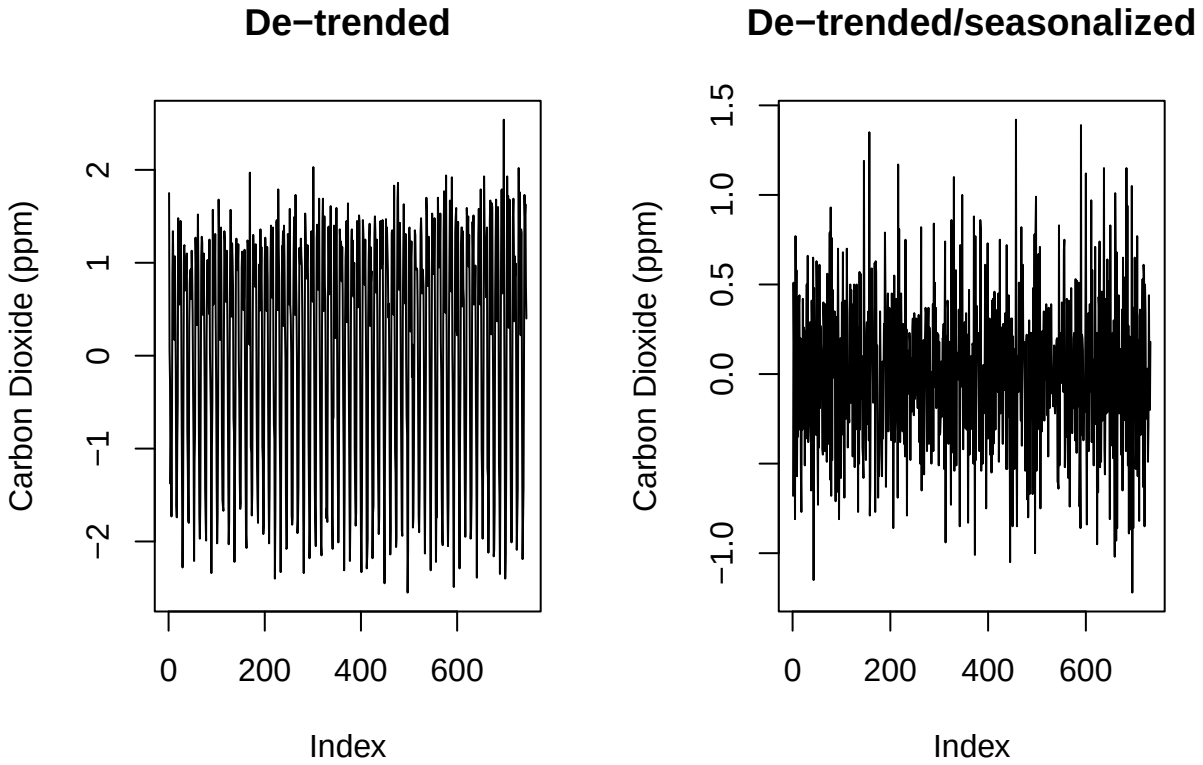


Achieving White Noise

The training data is visualized below.



We can identify several components of the time series that make it non-stationary. One, a linear trend is present. And two, there is seasonality. Fortunately, the variance appears stable throughout, indicating that there is no need for a stabilizing transformation. Thus, we will proceed to the differencing stage.



After the first difference, the trend was completely eliminated. However, seasonality was still present which necessitated an additional difference. We know that our data is measured monthly, with the seasonality

occurring in yearly cycles, thus indicating a difference at lag 12. After both of these operations, the trend and seasonality are no longer present in the time series, and it is visually akin to white noise. We can confirm these claims by intermittently calculating the variance at each step of the process.

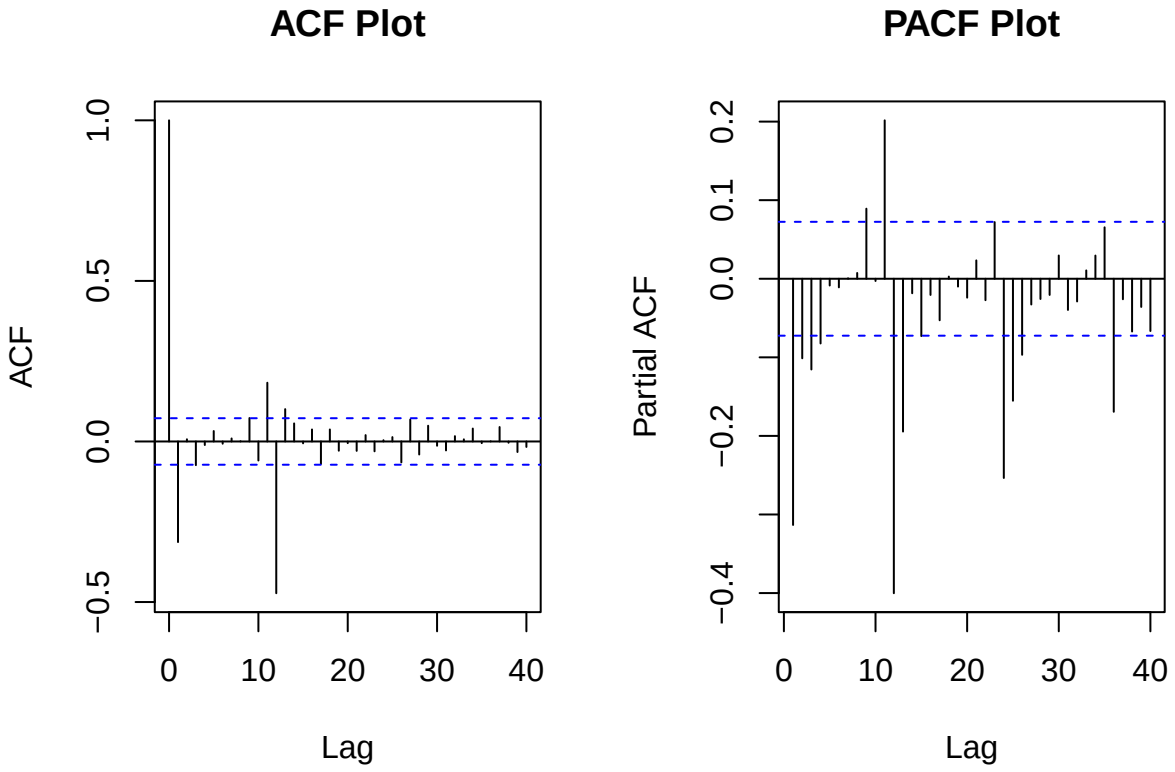
Table 1: Variance at several differencing steps

Training	817.7649926
De-trended	1.5135313
De-trended/seasonalized	0.1877926

The above table supports our previous intuition that the difference steps successfully reduced the variance, introducing a white noise process. Therefore, we will proceed to ACF and PACF analysis with our de-trended/seasonalized data.

ACF and PACF Analysis

In order to determine the presence and order of model components, we will analyze the patterns and structures of the ACF and PACF plots.



The ACF and PACF plots indicate a complex process with the presence of both non-seasonal and seasonal components, likely suggesting the necessity of modeling with SARIMA. Beginning with the ACF plot, we see four significant lags of interest: 1, 11, 12, and 13. As we previously discovered from the differencing operations, the process has a seasonal period of $s = 12$. Considering that we have a significant lag at $h = 1s = 12$, we know that we have a seasonal moving average process of order $Q = 1$. We can also notice that lags 1, 11, and 13 are also significant. The last two, which are equivalent $h = 1s \pm 1 = 11, 13$, in addition to the autocorrelation at lag 1, point towards a non-seasonal moving average process of order $q = 1$.

Moving on to the PACF plot, we again see several significant lags, namely 1, 11, 12, 13, and other. However, the pattern of exponential decay at the seasonal lags of $h = 1s = 2s = \cdots$ indicate that there is no seasonal autoregressive process, that is that $P = 0$. However, we do have a significant partial autocorrelation at lag 1,

indicating the presence of a non-seasonal moving average process of order $p = 1$. One may consider that the exponential decay exhibited by the PACF indicates a pure moving average process, so we will also select $p = 0$.

Finally, we previously performed a single non-seasonal difference at lag 1 and a single seasonal difference at lag 12 to achieve white noise, which indicates that we have $d = 1$, $D = 1$, and $s = 12$. Thus, we are left with two models to fit, summarized below.

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

Model Fitting

MLE Estimation

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

```
##
## Call:
## arima(x = train, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##          ar1      ma1     sma1
##       0.1936  -0.5517  -0.8615
## s.e.  0.0965   0.0827   0.0190
##
## sigma^2 estimated as 0.09593:  log likelihood = -188.91,  aic = 385.83
```

All coefficients are significant, indicating that we are left with a SARIMA$(1, 1, 1) \times (0, 1, 1)_{12}$ model.

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

```
##
## Call:
## arima(x = train, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##           ma1     sma1
##       -0.3816  -0.8611
## s.e.   0.0383   0.0192
##
## sigma^2 estimated as 0.09639:  log likelihood = -190.67,  aic = 387.34
```

All coefficients are significant and we are left with the same SARIMA$(0, 1, 1) \times (0, 1, 1)_{12}$ model as previous. Thus, we are left with two fitted models, represented algebraically below.

1. $(1 - 0.1936B)(1 - B)(1 - B^{12})X_t = (1 - 0.5517B)(1 - 0.8615B^{12})Z_t$
2. $(1 - B)(1 - B^{12})X_t = (1 - 0.3816B)(1 - 0.8611B^{12})Z_t$

Diagnostic Checking

Stationarity and Invertibility

For the first aspect of our diagnostic checking, we will ensure that the models are both stationary and invertible. Beginning with stationarity, we will need to check that the roots of the characteristic polynomials $\phi(z)$ and $\Phi(z)$ lie outside the unit circle. As for model 1, because our selected model only has one non-seasonal

autoregressive term, we can simplify this process to check if $\phi(z)$ satisfies $|\phi_1| < 1$. As for model 2, we have no autoregressive terms making it a pure moving average model, which implies that it is stationary.

Assessing invertibility follows a similar process, but this time we will need to check that the roots of the characteristic polynomials $\theta(z)$ and $\Theta(z)$ lie outside the unit circle. However, both models 1 and 2 only have one seasonal and one non-seasonal moving average term, so as before, we can simplify this process to check if $\theta(z)$ and $\Theta(z)$ satisfy $|\theta_1| < 1$ and $|\Theta_1| < 1$.

Table 2: Model 1 Stationarity and Invertibility

	Estimated Coefficient	|Coef|<1
ϕ_1	0.1936	True
θ_1	-0.5517	True
Θ_1	-0.8615	True

Table 3: Model 2 Stationarity and Invertibility

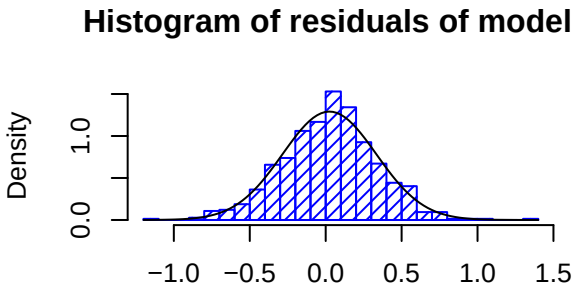
	Estimated Coefficient	|Coef|<1
θ_1	-0.3816	True
Θ_1	-0.8611	True

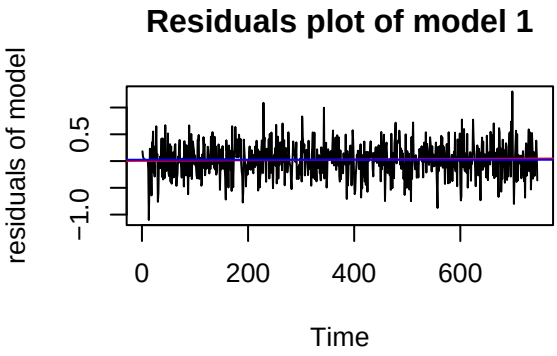
We can see that both models satisfy the criteria for being stationary and invertible. Thus, we can move on to residuals analysis.

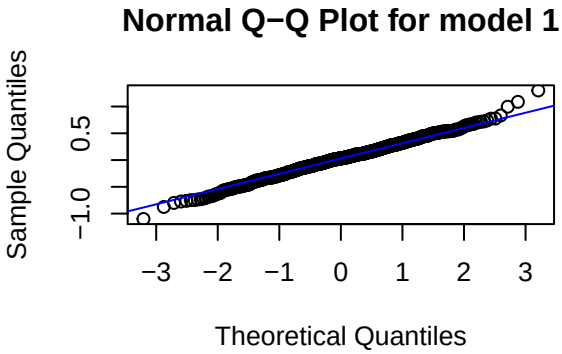
Residuals Analysis

For the final aspect of our diagnostic checking, we will confirm that the residuals of the models are white noise and normally distributed through visuals and several statistical tests.

Model 1 To visualize the distribution of the residuals for model 1, we produce the following plots.







Beginning with the histogram, the residuals appear to be normally distributed, with a symmetric bell-shaped density curve and a sample mean of 0.0247, almost being zero. Moving on to the time series format, the residuals are visually akin to white noise, lacking any trend or seasonality. Finally, the Q-Q plot has the majority of the quantiles on the Q-Q Line. Collectively, these analyses suggest that the residuals are white noise and normally distributed. Thus we will move on to performing a Shapiro-Wilk test and several Portmanteau tests.

Table 4: Shapiro-Wilk test for model 1

W	p-value
0.9963733	0.0855079

Table 5: Portmanteau tests for model 1

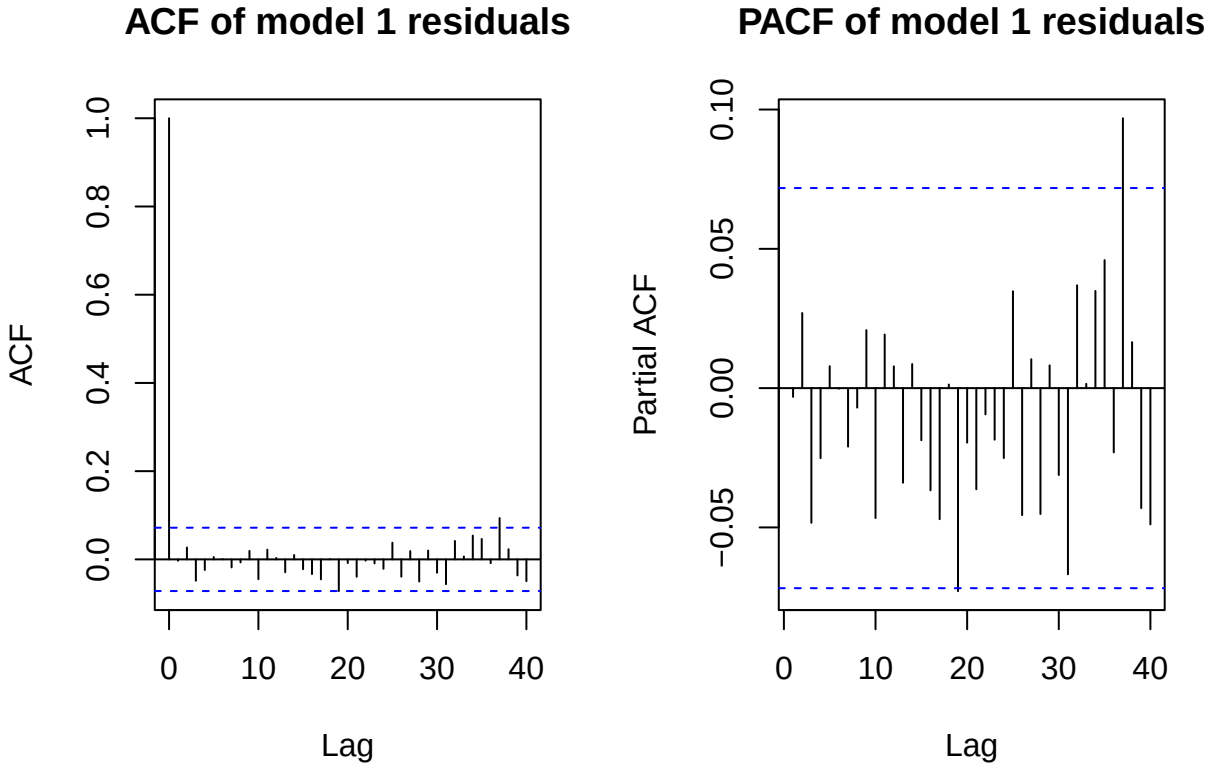
	χ^2	df	p-value
Box-Pierce	16.58818	24	0.8656865
Ljung-Box	16.98916	24	0.8491240
Mcleod-Li	33.11133	27	0.1934207

At the $\alpha = 0.05$ significance level, we fail to reject all null hypotheses, suggesting that there is not statistically significant evidence that the residuals are not normally distributed nor not independent. Thus, we will proceed to fitting an AR(p) model to the residuals.

```
##
## Call:
## ar(x = res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
```

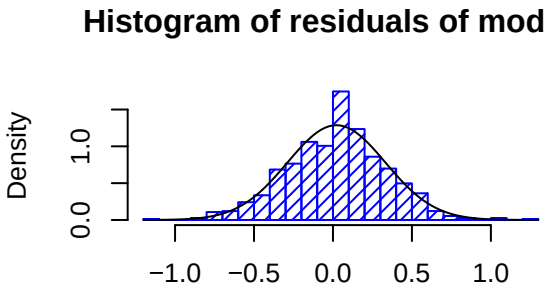
```
##
## Order selected 0  sigma^2 estimated as  0.0955
```

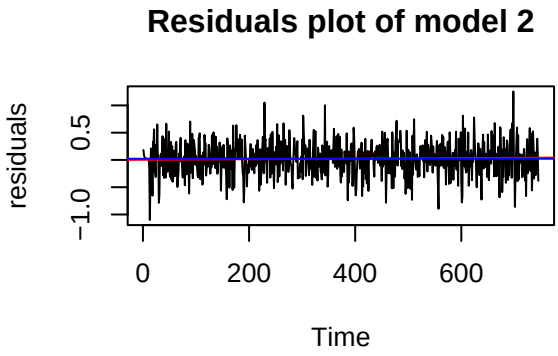
The fitted model is AR(0), indicating once again that residuals are white noise. Finally, we will create the ACF and PACF plots of the residuals.

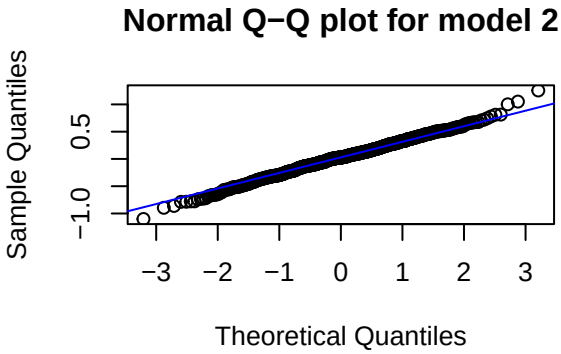


The ACF and PACF plots have no significant autocorrelations or partial-autocorrelations, aside from those at lag 37. However, due to the conservative nature of Bartlett's formula which calculates the error bounds and the relative proximity of said significant autocorrelations or partial-autocorrelations, these can be considered insignificant. Thus, we conclude that residuals for model 1 are white noise and normally distributed.

Model 2 Once again, we begin by visualizing the distribution of the residuals for model 2 with following plots.







Starting with the histogram, the residuals appear to be normally distributed, with a symmetric bell-shaped density curve and a sample mean of 0.022, almost being zero. As for the time series format, the residuals are characteristic of white noise, lacking any trend or seasonality. Finally, the Q-Q plot displays the majority of the quantiles on the Q-Q Line. Collectively, these observations suggest that the residuals are white noise and normally distributed. Thus we will move on to performing a Shapiro-Wilk test and several Portmanteau tests.

Table 6: Shapiro-Wilk test for model 2

W	p-value
0.9967182	0.1303926

Table 7: Portmanteau tests for model 2

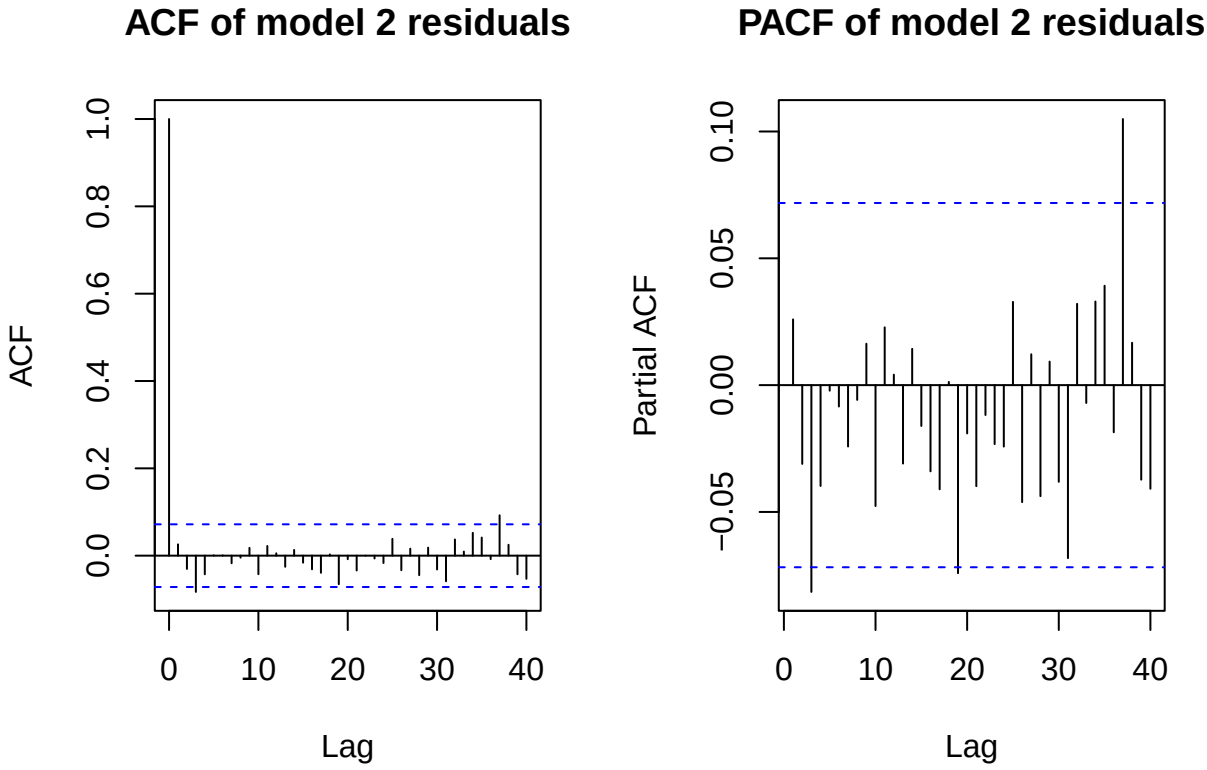
	χ^2	df	p-value
Box-Pierce	19.01955	25	0.7961823
Ljung-Box	19.38212	25	0.7784397
Mcleod-Li	32.76289	27	0.2050076

At the $\alpha = 0.05$ significance level, we fail to reject all null hypotheses, suggesting that there is not statistically significant evidence that the residuals are not normally distributed nor not independent. Thus, we will proceed to fitting an AR(p) model to the residuals.

```
##
## Call:
## ar(x = res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
## Coefficients:
```

```
##       1        2        3
##  0.0242  -0.0289  -0.0815
##
## Order selected 3  sigma^2 estimated as  0.09567
```

The fitted model is not AR(0), indicating that residuals are not white noise. Finally, we will create the ACF and PACF plots of the residuals.



The ACF and PACF plots have no significant autocorrelations or partial-autocorrelations, aside from those at lags 3 and 37. However, due to the conservative nature of Bartlett's formula which calculates the error bounds and the relative proximity of said significant autocorrelations or partial-autocorrelations, these can be considered insignificant. While all other tests and plots suggested that the residuals for model 2 are white noise and normally distributed, the AR(3) fit for the residuals suggests otherwise.

Model Comparison

We can compare models 1 and 2 through their calculated AIC values, summarized below.

Table 8: AIC values for models 1 and 2

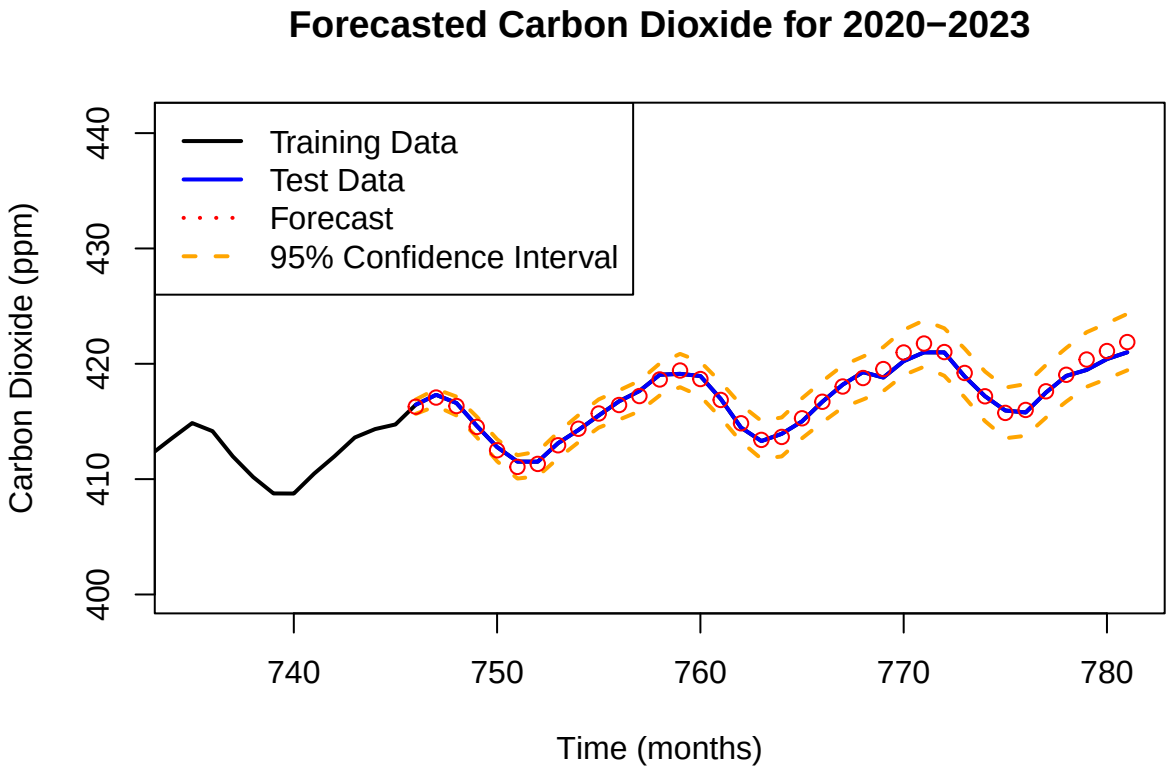
Model 1	385.8260
Model 2	387.3435

The above table indicates that model 1 marginally outperforms model 2. One may consider the principle of parsimony, which suggests that picking a simpler model is always optimal. Upon observing the near-equivalent AIC values, this holds. However, as we found in the diagnostic checking, we could not confidently conclude that the residuals of model 2 were normally distributed and white noise, so we will select model 1 for forecasting. This model is algebraically represented below.

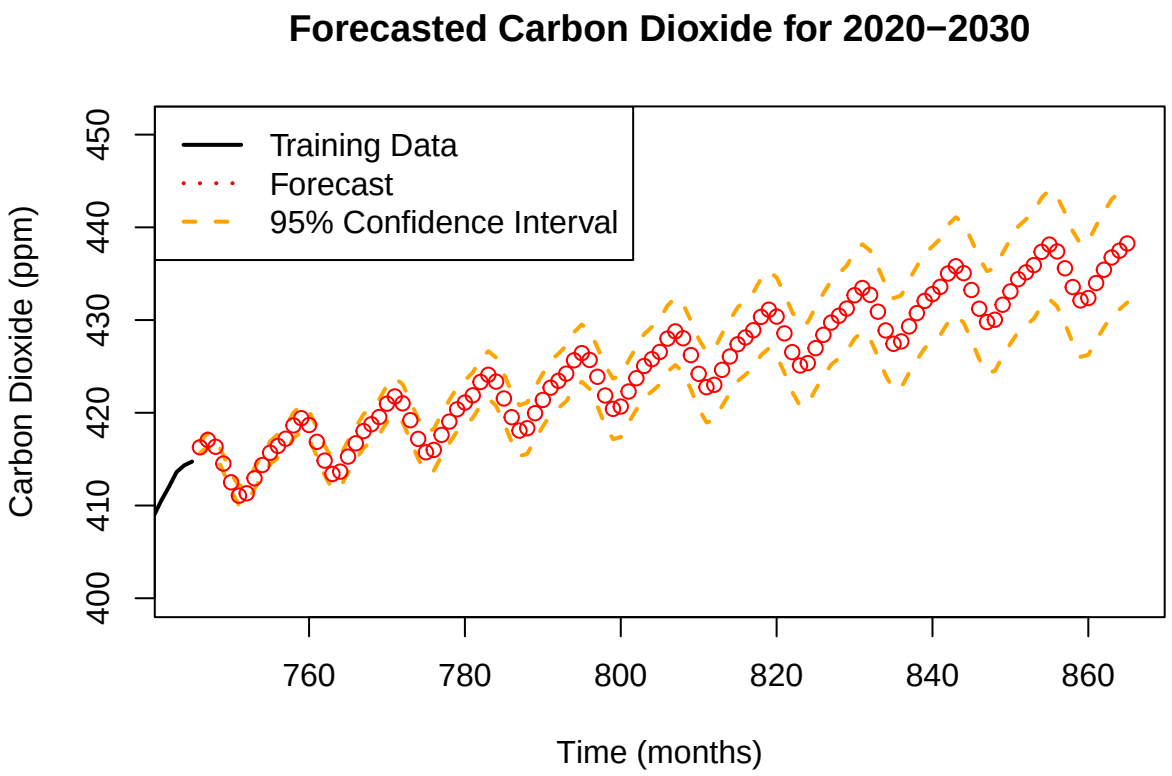
$$(1 - 0.1936B)(1 - B)(1 - B^{12})X_t = (1 - 0.5517B)(1 - 0.8615B^{12})Z_t$$

Forecasting

To begin our forecasting, we will predict the carbon dioxide (ppm) from 2020 to 2023, utilizing the test set for validation.



We can see that the model performed exceptionally well on the test set, with all of the forecasts within the 95% confidence interval, and almost all directly on the test line with the remainder closely adjacent. Now, what do these forecasts actually mean for the projected increase in CO_2 emission? The recorded amount of carbon dioxide at the beginning of 2020 was 416.45 ppm and increased to 421 ppm by the end of 2023. That accounts for a 4.55 ppm increase or a 1.0926% change over the course of those three years. Does that sound like a lot? Now, we will forecast the CO_2 for the next ten years to get a sense of the long term changes to these figures.



The measured amount of carbon dioxide is projected to continue its increase, reaching 438.2617358 ppm by the end of 2030. In comparison to the value at the start of 2020, this jump corresponds to a 5.2375% increase.

Conclusion

This project aimed to develop a suitable model for forecasting atmospheric carbon dioxide levels. After applying differencing to remove trend and seasonality, ACF and PACF analysis, MLE-based model fitting, and diagnostic checking led us to select a SARIMA$(1,1,1)\times(0,1,1)_{12}$ model:

$$(1-0.1936B)(1-B)(1-B^{12})X_t = (1-0.5517B)(1-0.8615B^{12})Z_t$$

The model successfully captured the underlying patterns in the data, yielding highly accurate forecasts. More critically, it highlighted the accelerating rise in emissions, emphasizing the urgency of addressing climate change. These tools not only quantify the problem but serve as a wake-up call—the time for action is now.

References

National Oceanic and Atmospheric Administration. (n.d.). Carbon dioxide trends at Mauna Loa Observatory. Retrieved from https://gml.noaa.gov/ccgg/trends/

Stoffer, D. S. (2025). astsa: Applied Statistical Time Series Analysis (Version 2.2) [R package].

Appendix

```r
# Loading the Cardox dataset from the astsa package
library(astsa)
data(cardox)

# Plotting the entire time series
plot(cardox,
     main = "Monthly Carbon Dioxide Measurements from 1958 - 2023",
     xlab = "Time (months)",
     ylab = "Carbon Dioxide (ppm)",
     lwd = 2)
```

```r
# Training test split
forecast_length <- 36 # three years from 2020-2023 (beginning of)
cutoff <- c(1:(length(cardox) - forecast_length))
train <- cardox[cutoff]
test <- cardox[-cutoff]
```

```r
# Plotting the training test split
t <- 1:length(cardox)
t_1 <- 1:length(train)
t_2 <- (length(train)+1):length(cardox)

plot(t, cardox,
     main = "Training-Test Split",
     xlab = "Index",
     ylab = "Carbon Dioxide (ppm)",
     lwd = 2,
     type = "l")
lines(t_2, test,
      lwd = 2,
      type = "l",
      col = "blue")
legend("topleft", legend = c("Training Data", "Test Data"),
       col = c("black", "blue"), lty = 1, lwd = 2)
```

```r
# Plotting the training data
plot(t_1, train,
     main = "Monthly Carbon Dioxide Measurements from 1958 - 2020",
     xlab = "Index",
     ylab = "Carbon Dioxide (ppm)",
     lwd = 2,
     type = "l")
```

```r
# Plotting the differenced data and calculating the variance at each step
op = par(mfrow = c(1,2))
train_d1 <- diff(train, 1)
plot(train_d1,
     main = "De-trended",
     ylab = "Carbon Dioxide (ppm)",
     type = "l")
d1_var <- var(train_d1)
train_d1_12 <- diff(train_d1, 12)
plot(train_d1_12,
```

```
     main = "De-trended/seasonalized",
     ylab = "Carbon Dioxide (ppm)",
     type = "l")
d1_12_var <- var(train_d1_12)
```

```
# Aggregating the calculated variances

# Use of knitr for the kable() function to create tables
library(knitr)

train_var <- var(train)

table <- matrix(
  c(train_var, d1_var, d1_12_var),
  nrow = 3,
  ncol = 1
)

rownames(table) <- c("Training", "De-trended", "De-trended/seasonalized")
kable(table, caption = "Variance at several differencing steps")
```

```
# Creating ACF and PACF plots to choose p, q, P, and Q
op = par(mfrow = c(1,2))
acf(train_d1_12, lag.max = 40, main="")
title("ACF Plot")
pacf(train_d1_12, lag.max = 40, main="")
title("PACF Plot")
```

```
# Fitting model 1 with MLE estimation
mod1 <- arima(train, order=c(1,1,1),
               seasonal = list(order = c(0,1,1),
                               period = 12),
              method = "ML")
mod1
```

```
# Fitting model 2 with MLE estimation
mod2 <- arima(train, order=c(0,1,1),
               seasonal = list(order = c(0,1,1),
                               period = 12),
              method = "ML")
mod2
```

```
# Stationarity and invertibility checking for model 1

# Use of kableExtra allows Latex to render inside of kable table
library(kableExtra)
table <- matrix(
  c(0.1936, -0.5517, -0.8615,
    "True", "True", "True"),
  nrow = 3, ncol = 2, byrow = FALSE
)

rownames(table) <- c("$\\phi_1$", "$\\theta_1$", "$\\Theta_1$")
colnames(table) = c("Estimated Coefficient", "|Coef|<1")
kable(table, caption = "Model 1 Stationarity and Invertibility",
```

```
      escape = FALSE)
```

```
# Stationarity and invertibility checking for model 2
table <- matrix(
  c(-0.3816, -0.8611,
    "True", "True"),
  nrow = 2, ncol = 2, byrow = FALSE
)

rownames(table) <- c("$\\theta_1$", "$\\Theta_1$")
colnames(table) = c("Estimated Coefficient", "|Coef|<1")
kable(table, caption = "Model 2 Stationarity and Invertibility",
      escape = FALSE)
```

```
# Plotting the residuals for model 1
res1 = residuals(mod1)
par(mfrow=c(2,2))
hist(res1,density=20,breaks=20,
     col="blue",
     xlab="",
     prob=TRUE,
     main="Histogram of residuals of model 1")
m <- mean(res1)
std <- sqrt(var(res1))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res1,ylab= "residuals of model",main="Residuals plot of model 1")
fitt <- lm(res1~ as.numeric(1:length(res1)))
abline(fitt, col="red")
abline(h=mean(res1), col="blue")
qqnorm(res1,main= "Normal Q-Q Plot for model 1")
qqline(res1,col="blue")
```

```
# Performing Shapiro-Wilk test and Portmanteau tests for model 1 residuals
shapiro <- shapiro.test(res1)

table <- matrix(
  c(shapiro$statistic, shapiro$p.value),
  nrow = 1,
  ncol = 2,
  byrow = TRUE
)

colnames(table) = c("W", "p-value")
kable(table, caption = "Shapiro-Wilk test for model 1")

h <- round(sqrt(length(train)))
box_pierce <- Box.test(res1, lag = h, type = c("Box-Pierce"), fitdf = 3)
ljung_box <- Box.test(res1, lag = h, type = c("Ljung-Box"), fitdf = 3)
mcleod_li <- Box.test(res1^2, lag = h, type = c("Ljung-Box"), fitdf = 0)

table <- matrix(
  c(box_pierce$statistic, box_pierce$parameter, box_pierce$p.value,
    ljung_box$statistic, ljung_box$parameter, ljung_box$p.value,
    mcleod_li$statistic, mcleod_li$parameter, mcleod_li$p.value),
```

```
  nrow = 3,
  ncol = 3,
  byrow = TRUE
)

rownames(table) <- c("Box-Pierce", "Ljung-Box", "Mcleod-Li")
colnames(table) = c("$\\chi^2$", "df", "p-value")
kable(table, caption = "Portmanteau tests for model 1",
      escape = FALSE)
```

```
# Fitting AR(p) model to the model 1 residuals
ar(res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
# Creating ACF and PACF plots for the model 1 residuals
par(mfrow=c(1,2))
acf(res1, lag.max=40,main="")
title("ACF of model 1 residuals")
pacf(res1, lag.max=40,main="")
title("PACF of model 1 residuals")
```

```
# Plotting the residuals for model 2
res2 = residuals(mod2)
par(mfrow=c(2,2))
hist(res2,density=20,breaks=20,
     col="blue",
     xlab="",
     prob=TRUE,
     main="Histogram of residuals of model 2")
m <- mean(res2)
std <- sqrt(var(res2))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res2,ylab = "residuals", main="Residuals plot of model 2")
fitt <- lm(res2~ as.numeric(1:length(res2)))
abline(fitt, col="red")
abline(h=mean(res2), col="blue")
qqnorm(res2,main= "Normal Q-Q plot for model 2")
qqline(res2,col="blue")
```

```
# Performing Shapiro-Wilk test and Portmanteau tests for model 2 residuals
shapiro <- shapiro.test(res2)

table <- matrix(
  c(shapiro$statistic, shapiro$p.value),
  nrow = 1,
  ncol = 2,
  byrow = TRUE
)

colnames(table) = c("W", "p-value")
kable(table, caption = "Shapiro-Wilk test for model 2")

library(kableExtra)
h <- round(sqrt(length(train)))
box_pierce <- Box.test(res2, lag = h, type = c("Box-Pierce"), fitdf = 2)
ljung_box <- Box.test(res2, lag = h, type = c("Ljung-Box"), fitdf = 2)
```

```r
mcleod_li <- Box.test(res2^2, lag = h, type = c("Ljung-Box"), fitdf = 0)

table <- matrix(
  c(box_pierce$statistic, box_pierce$parameter, box_pierce$p.value,
    ljung_box$statistic, ljung_box$parameter, ljung_box$p.value,
    mcleod_li$statistic, mcleod_li$parameter, mcleod_li$p.value),
  nrow = 3,
  ncol = 3,
  byrow = TRUE
)

rownames(table) <- c("Box-Pierce", "Ljung-Box", "Mcleod-Li")
colnames(table) = c("$\\chi^2$", "df", "p-value")
kable(table, caption = "Portmanteau tests for model 2",
      escape = FALSE)
```

```r
# Fitting AR(p) model to the model 2 residuals
ar(res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```r
# Creating ACF and PACF plots for the model 2 residuals
par(mfrow=c(1,2))
acf(res2, lag.max=40,main="")
title("ACF of model 2 residuals")
pacf(res2, lag.max=40,main="")
title("PACF of model 2 residuals")
```

```r
# Comparing the AIC for models 1 and 2
table <- matrix(
  c(mod1$aic, mod2$aic),
  nrow = 2,
  ncol = 1
)

rownames(table) <- c("Model 1", "Model 2")
kable(table, caption = "AIC values for models 1 and 2")
```

```r
# Forecasting on the test set
library(forecast)
forecast_length <- 36
pred.tr <- predict(mod1, n.ahead = forecast_length)
U.tr = pred.tr$pred + 2*pred.tr$se
L.tr = pred.tr$pred- 2*pred.tr$se

ts.plot(as.numeric(cardox),
        xlim = c(length(train)-10,length(train)+forecast_length),
        ylim = c(400,max(cardox) + 20),
        lwd = 2, col="black",
        main = "Forecasted Carbon Dioxide for 2020-2023",
        xlab = "Time (months)",
        ylab="Carbon Dioxide (ppm)")
lines((length(train)+1):length(cardox), test, lwd = 2, col="blue")
lines(U.tr, lwd = 2, col="orange", lty="dashed")
lines(L.tr, lwd = 2, col="orange", lty="dashed")
points((length(cardox)-forecast_length+1):length(cardox), pred.tr$pred, col="red")
legend("topleft",
```

```
       legend = c("Training Data",
                  "Test Data",
                  "Forecast",
                  "95% Confidence Interval"),
       col = c("black", "blue", "red", "orange"),
       lty = c(1, 1, 3, 2), lwd = 2)
```

```
# Forecasting beyond the dataset
library(forecast)
forecast_length <- 120
pred.tr <- predict(mod1, n.ahead = forecast_length)
U.tr = pred.tr$pred + 2*pred.tr$se
L.tr = pred.tr$pred- 2*pred.tr$se

ts.plot(as.numeric(train),
        xlim = c(length(train),length(train)+forecast_length),
        ylim = c(400,max(cardox) + 30),
        lwd = 2, col="black",
        main = "Forecasted Carbon Dioxide for 2020-2030",
        xlab = "Time (months)",
        ylab="Carbon Dioxide (ppm)")
lines(U.tr, lwd = 2, col="orange", lty="dashed")
lines(L.tr, lwd = 2, col="orange", lty="dashed")
points((length(train)+1):(length(train)+forecast_length), pred.tr$pred, col="red")
legend("topleft",
       legend = c("Training Data",
                  "Forecast",
                  "95% Confidence Interval"),
       col = c("black", "red", "orange"),
       lty = c(1, 3, 2), lwd = 2)
```