Forecasting Carbon Dioxide Emissions

Brooks Piper

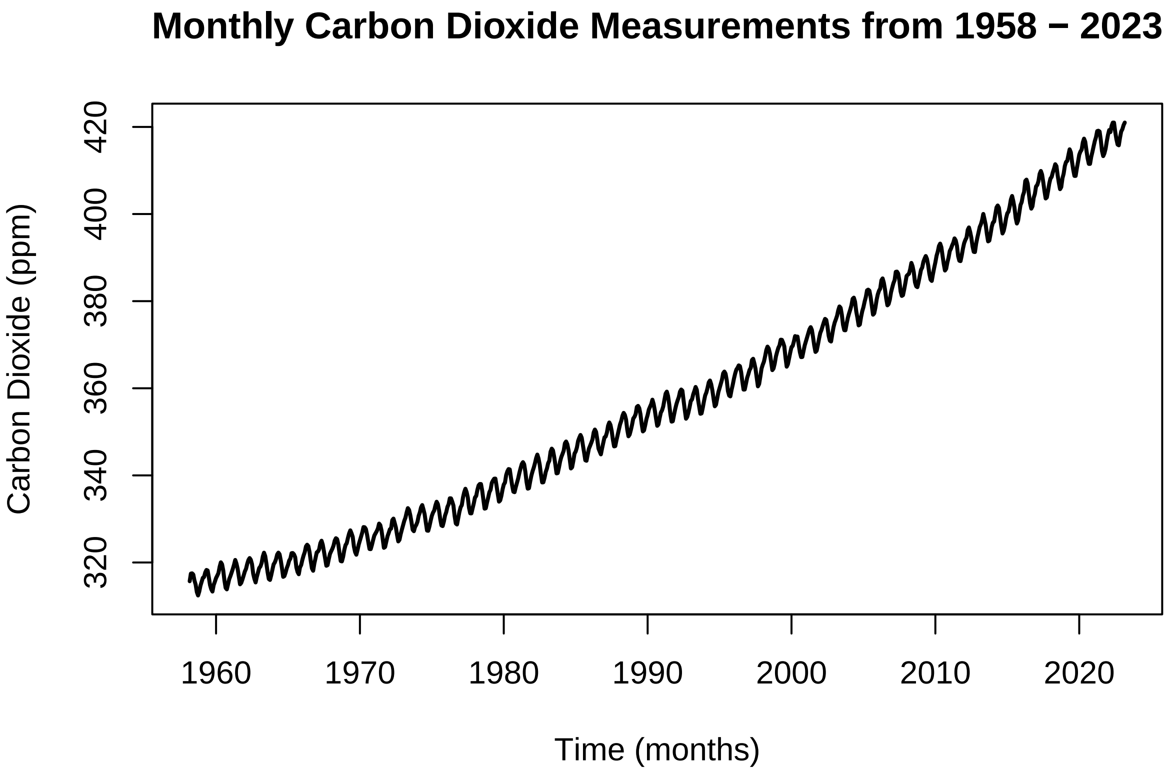
2025-03-03

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



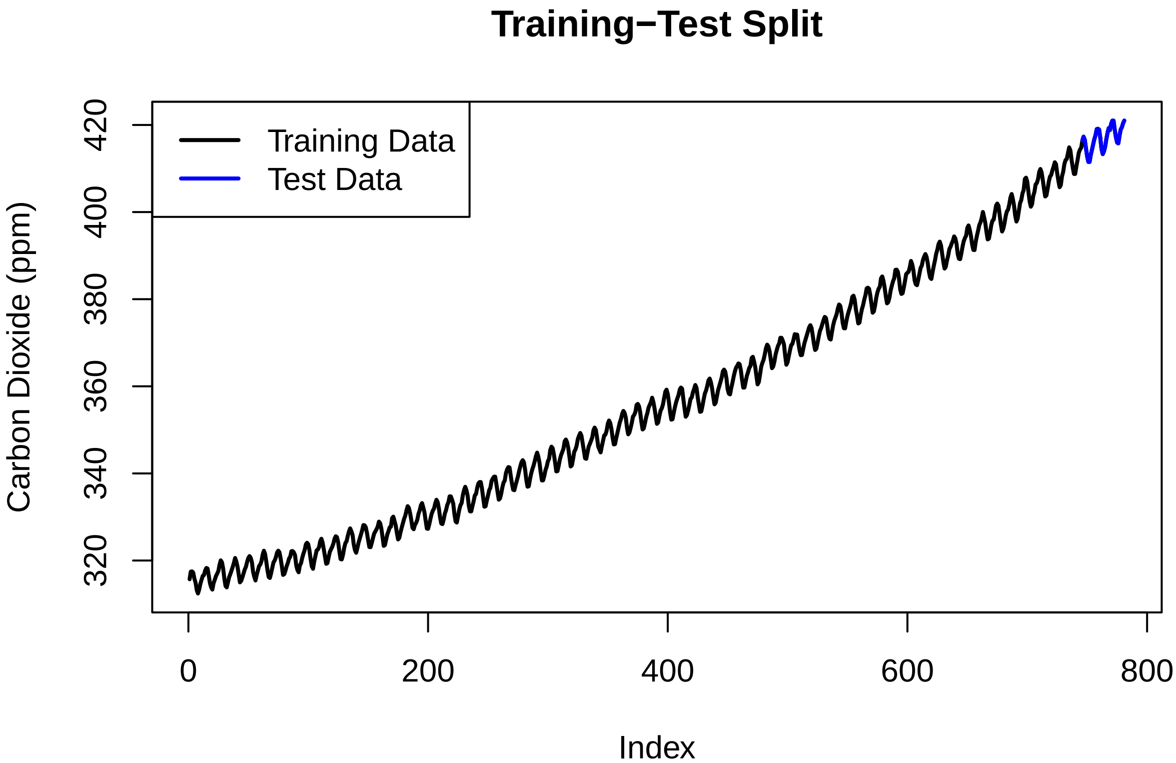
Greenhouse gases drive climate change, with carbon dioxide (CO_2) accounting for over 80% of U.S. emissions. As the primary contributor to rising temperatures, sea levels, and ecosystem disruptions—largely from fossil fuel combustion—its continued increase is inevitable (as seen in the chart above). 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

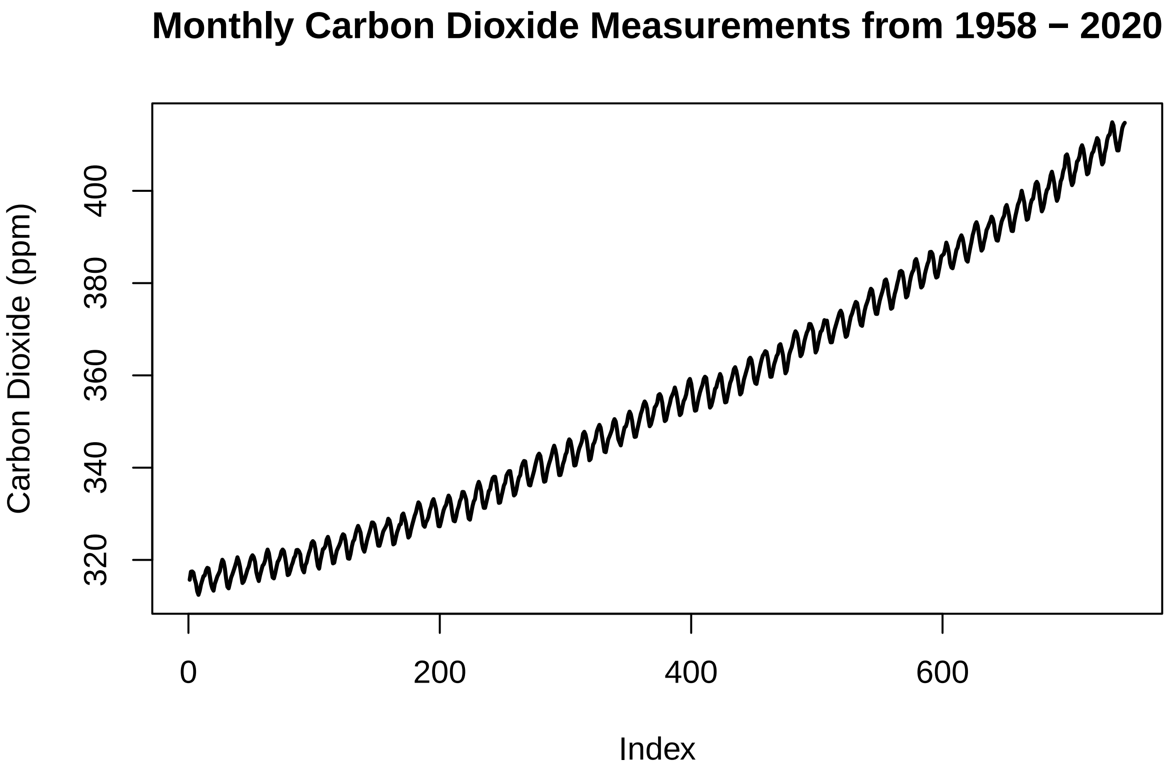
Training and Testing Set Split

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.

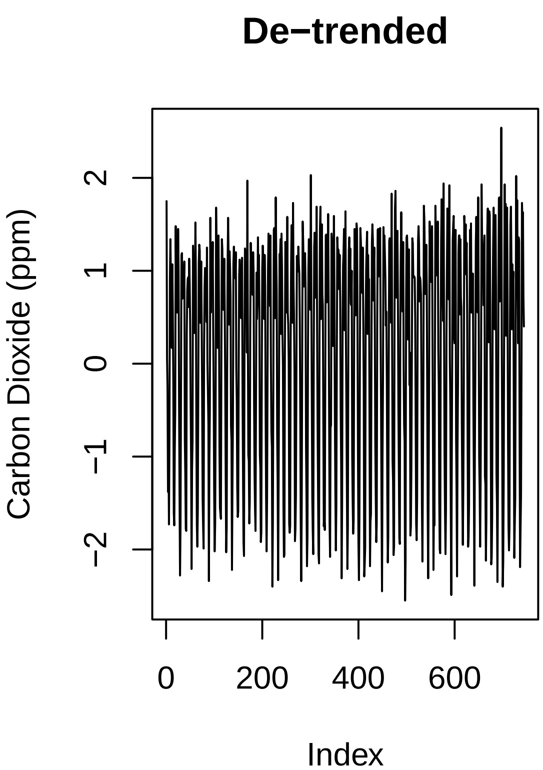


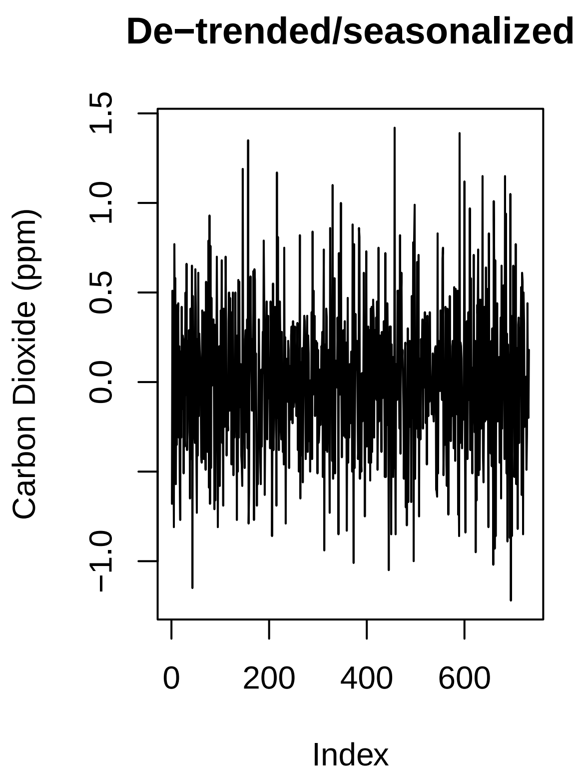
Achieving White Noise

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.

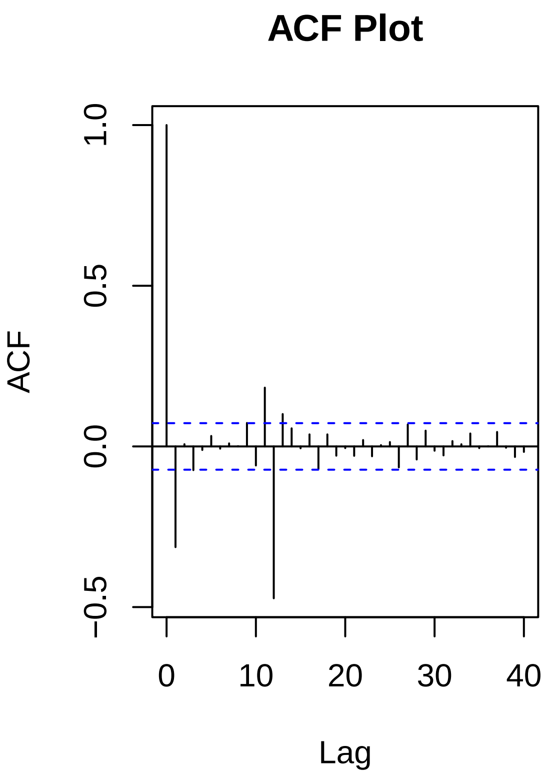
Table 1: Variance at several differencing steps

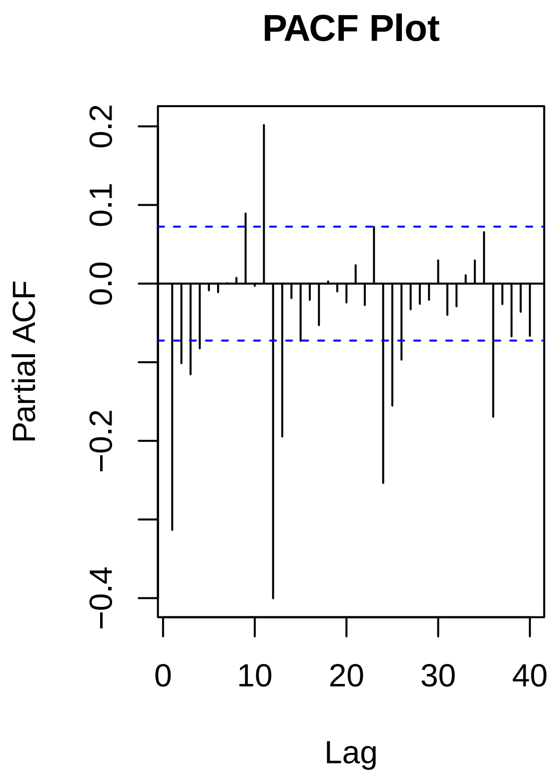
Training	817.7649926
De-trended	1.5135313
De-trended/seasonalized	0.1877926

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 three significant lags of interest: 1, 11, and 12. As for the first two, these suggest a non-seasonal moving average process of order $q = 1$ and $q = 11$. As for the third, we have a significant lag at $h = 1s = 12$ which suggests a seasonal moving average process of order $Q = 1$.

Moving on to the PACF plot, we again see several significant lags, namely 1 and 11. Both suggest a non-seasonal autoregressive process of order $p = 1$ or $p = 11$. One could argue that there is also a seasonal autoregressive process, however, the seasonal lags $h = 1s = 2s = \cdots$ are exponentially decaying towards insignificance. Thus, we will 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 four final models, summarized below.

1. SARIMA$(1, 1, 1) \times (0, 1, 1)_{12}$
2. SARIMA$(11, 1, 1) \times (0, 1, 1)_{12}$
3. SARIMA$(1, 1, 11) \times (0, 1, 1)_{12}$
4. SARIMA$(11, 1, 11) \times (0, 1, 1)_{12}$

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$(11, 1, 1) \times (0, 1, 1)_{12}$

```
##
## Call:
## arima(x = train, order = c(11, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6      ar7     ar8
##       -0.1771  -0.0989  -0.1055  -0.0631  -0.0177  -0.0041  -0.0196  0.0003
## s.e.   3.4747   1.2505   0.5723   0.4729   0.3072   0.1234   0.0545  0.0902
##          ar9     ar10    ar11      ma1     sma1
##       0.0337  -0.0339  0.0182  -0.1825  -0.8614
## s.e.  0.0441   0.1028  0.1520   3.4692   0.0224
##
## sigma^2 estimated as 0.09528:  log likelihood = -186.5,  aic = 401.01
```

All coefficients aside from the seasonal moving average component are insignificant and will be removed.

```
## Warning in arima(train, order = c(11, 1, 1), seasonal = list(order = c(0, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```
##
## Call:
## arima(x = train, order = c(11, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),
##     fixed = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, NA), method = "ML")
##
## Coefficients:
```

```
##       ar1  ar2  ar3  ar4  ar5  ar6  ar7  ar8  ar9  ar10  ar11  ma1     sma1
##         0    0    0    0    0    0    0    0    0     0     0    0  -1.1130
## s.e.    0    0    0    0    0    0    0    0    0     0     0    0   0.0216
##
## sigma^2 estimated as 0.08691:  log likelihood = -232.82,  aic = 469.63
```

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

SARIMA$(1,1,11)\times(0,1,1)_{12}$

```
##
## Call:
## arima(x = train, order = c(1, 1, 11), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1      ma1      ma2      ma3      ma4     ma5     ma6      ma7
##       -0.0724  -0.2862  -0.0594  -0.0696  -0.0110  0.0174  0.0084  -0.0130
## s.e.   0.7046   0.7018   0.2540   0.0459   0.0634  0.0392  0.0413   0.0383
##          ma8     ma9     ma10    ma11     sma1
##       0.0153  0.0286  -0.0437  0.0351  -0.8648
## s.e.  0.0422  0.0410   0.0439  0.0478   0.0207
##
## sigma^2 estimated as 0.09528:  log likelihood = -186.5,  aic = 401
```

Similar to before, all coefficients aside from the seasonal moving average component are insignificant and will be removed.

```
## Warning in arima(train, order = c(1, 1, 11), seasonal = list(order = c(0, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```
##
## Call:
## arima(x = train, order = c(1, 1, 11), seasonal = list(order = c(0, 1, 1), period = 12),
##     fixed = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, NA), method = "ML")
##
## Coefficients:
##       ar1  ma1  ma2  ma3  ma4  ma5  ma6  ma7  ma8  ma9  ma10  ma11     sma1
##         0    0    0    0    0    0    0    0    0    0     0     0  -1.1130
## s.e.    0    0    0    0    0    0    0    0    0    0     0     0   0.0216
##
## sigma^2 estimated as 0.08691:  log likelihood = -232.82,  aic = 469.63
```

All coefficients are significant and we are left with the same SARIMA$(0,1,0)\times(0,1,1)_{12}$ model as previous.

SARIMA$(11,1,11)\times(0,1,1)_{12}$

```
## Warning in arima(train, order = c(11, 1, 11), seasonal = list(order = c(0, :
## possible convergence problem: optim gave code = 1
```

```
##
## Call:
## arima(x = train, order = c(11, 1, 11), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##          ar1      ar2      ar3      ar4     ar5     ar6     ar7      ar8
##       0.0469  -0.6108  -0.0886  -0.0457  0.0496  0.1848  0.5422  -0.3499
## s.e.     NaN   0.8389      NaN   0.0901  0.1968  0.2026  0.2531   0.2800
##          ar9     ar10    ar11      ma1     ma2      ma3      ma4      ma5
##       0.5681  -0.2242  0.0730  -0.3936  0.5935  -0.2015  -0.0092  -0.0831
## s.e.  0.3935   0.5919  0.1028      NaN  0.8449   0.3701   0.2425   0.1952
##           ma6      ma7     ma8      ma9    ma10     ma11     sma1
##       -0.1938  -0.4789  0.5849  -0.6778  0.4364  -0.1330  -0.8542
## s.e.   0.1316   0.1430  0.1148   0.2891  0.7931   0.2633   0.0229
##
## sigma^2 estimated as 0.09273:  log likelihood = -179.55,  aic = 407.11
```

Once again, as before, all coefficients aside from the seasonal moving average component are insignificant and will be removed.

```
## Warning in arima(train, order = c(11, 1, 11), seasonal = list(order = c(0, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```
##
## Call:
## arima(x = train, order = c(11, 1, 11), seasonal = list(order = c(0, 1, 1), period = 12),
##     fixed = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
##         0, NA), method = "ML")
##
## Coefficients:
##       ar1  ar2  ar3  ar4  ar5  ar6  ar7  ar8  ar9  ar10  ar11  ma1  ma2  ma3
##         0    0    0    0    0    0    0    0    0     0     0    0    0    0
## s.e.    0    0    0    0    0    0    0    0    0     0     0    0    0    0
##       ma4  ma5  ma6  ma7  ma8  ma9  ma10  ma11     sma1
##         0    0    0    0    0    0     0     0  -1.1130
## s.e.    0    0    0    0    0    0     0     0   0.0216
##
## sigma^2 estimated as 0.08691:  log likelihood = -232.82,  aic = 469.63
```

Once coefficients are significant and we are left with the same $\text{SARIMA}(0,1,0) \times (0,1,1)_{12}$ model as with the previous two iterations. This concludes the model fitting.

Model Comparison

After the fitting stage, the models either converged to $\text{SARIMA}(1,1,1) \times (0,1,1)_{12}$ or $\text{SARIMA}(0,1,0) \times (0,1,1)_{12}$. For loss of generality, we will assign the latter to be model 2 despite models 3 and 4 identically converging. Only having models 1 and 2 to decide between, we can compare their calculated AIC values, summarized below.

Table 2: AIC Values for Models 1 and 2

Model 1	385.8260
Model 2	469.6331

The above table indicates that model 1 far outperforms model 2. One may consider the principle of parsimony, which suggests that picking a simpler model is always optimal, yet the stark difference in model fit supersedes that concept. Thus, we will select model 1 which can be algebraically represented as follows.

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

Diagnostic Checking

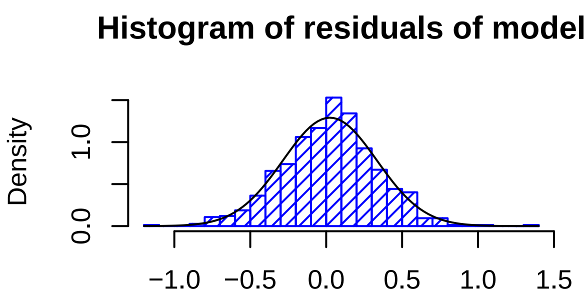
Stationarity and Invertibility

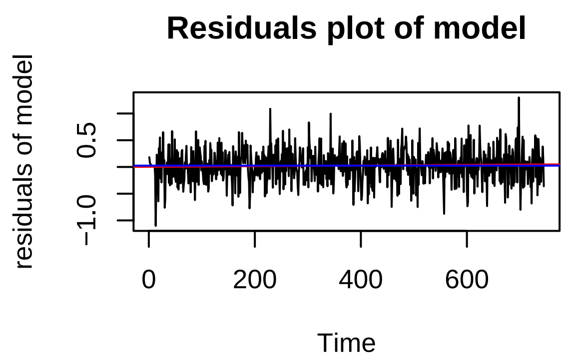
For the first aspect of our diagnostic checking, we will ensure that the model is both stationary and invertible. Beginning with stationarity, we will need to check that the roots of the characteristic polynomials $\phi(z)$ and $\Phi(z)$ lie outside the unit circle. However, because our selected model only has one non-seasonal autoregressive term, we can simplify this process to check if $\phi(z)$ satisfies $|\phi_1| < 1$. We have that $|\phi_1| = 0.1936 < 1$, which indicates that the model is stationary.

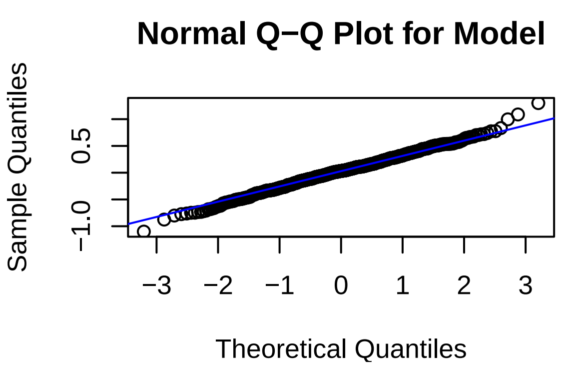
Assessing invertibility follows a similar process, but this time we will need to check that the roots of the characteristic polynomials $\theta(z)$ and $\Theta(z)$ lie outside the unit circle. Once again, our selected model only has one seasonal and one non-seasonal moving average term, so we can simplify this process to check if $\theta(z)$ and $\Theta(z)$ satisfy $|\theta_1| < 1$ and $|\Theta_1| < 1$. We have that $|\theta_1| = 0.5517 < 1$ and $|\Theta_1| = 0.8615 < 1$, which indicates that the model is also invertible.

Residuals Analysis

For the final aspect of our diagnostic checking, we will confirm that the residuals of the model are white noise and normally distributed through visuals and several statistical tests. Beginning with the former, we produce the following plots.







Beginning with the histogram, the residuals appear to be normally distributed, with a symmetric bell-shaped density curve. 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 3: Shapiro-Wilk Test

W	p-value
0.9963733	0.0855079

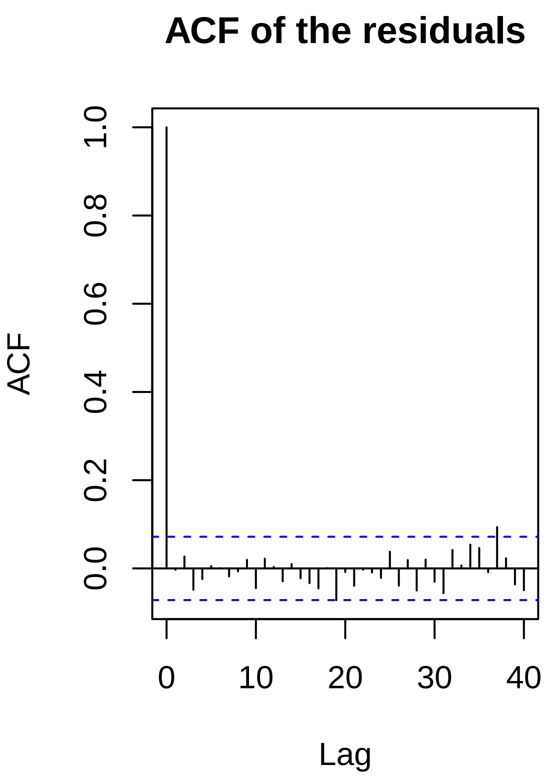
Table 4: Portmanteau Tests

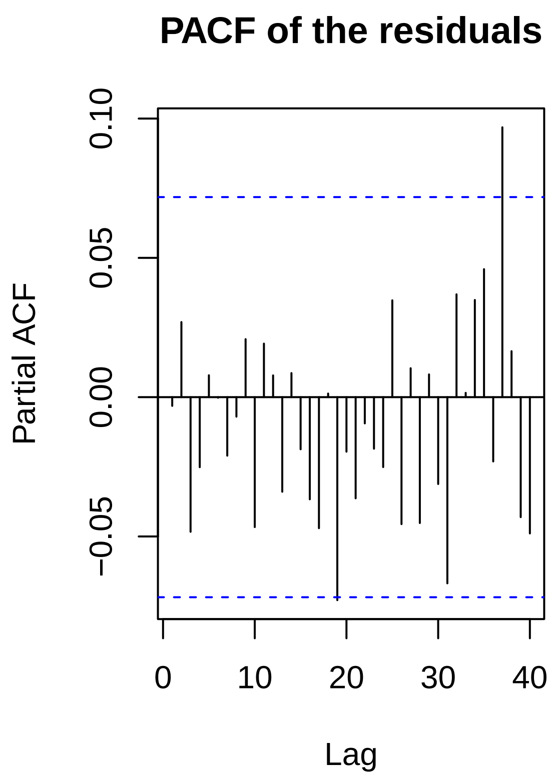
	χ^2	df	p-value
Box-Pierce	16.58818	24	0.8656865
Ljung-Box	16.98916	24	0.8491240
Mcleod-Li	33.11133	27	0.1934207

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 = res, 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.

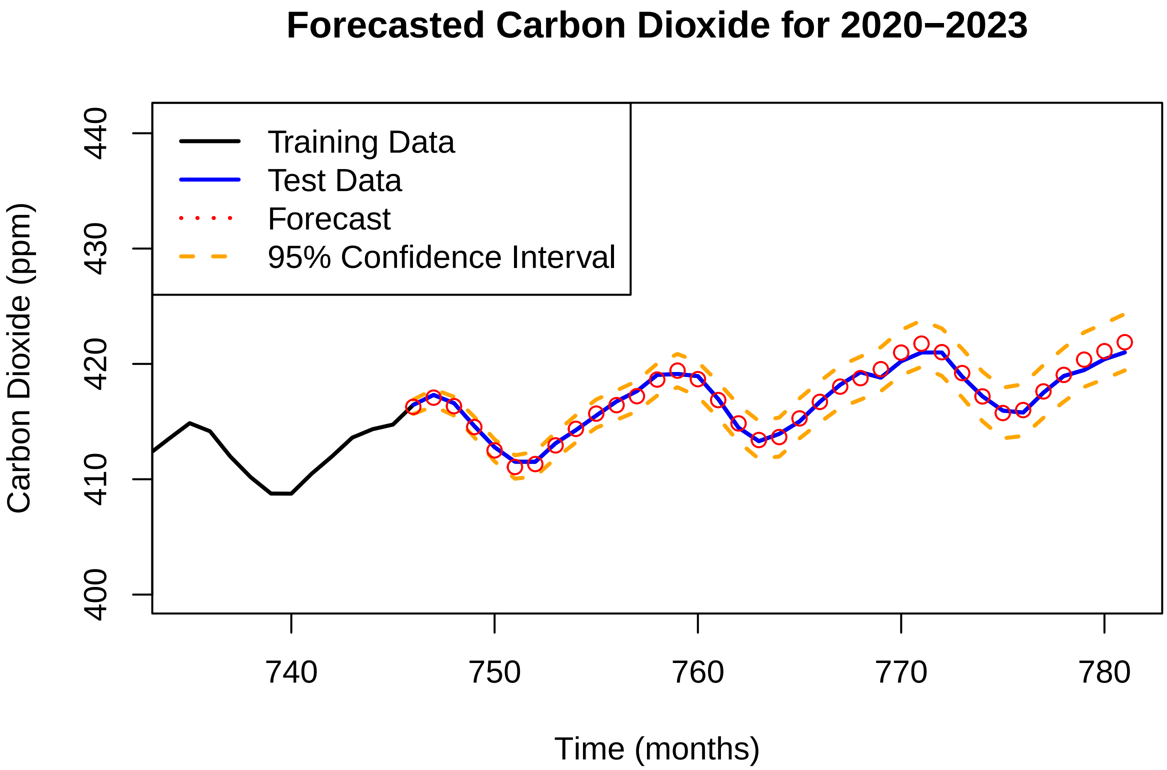




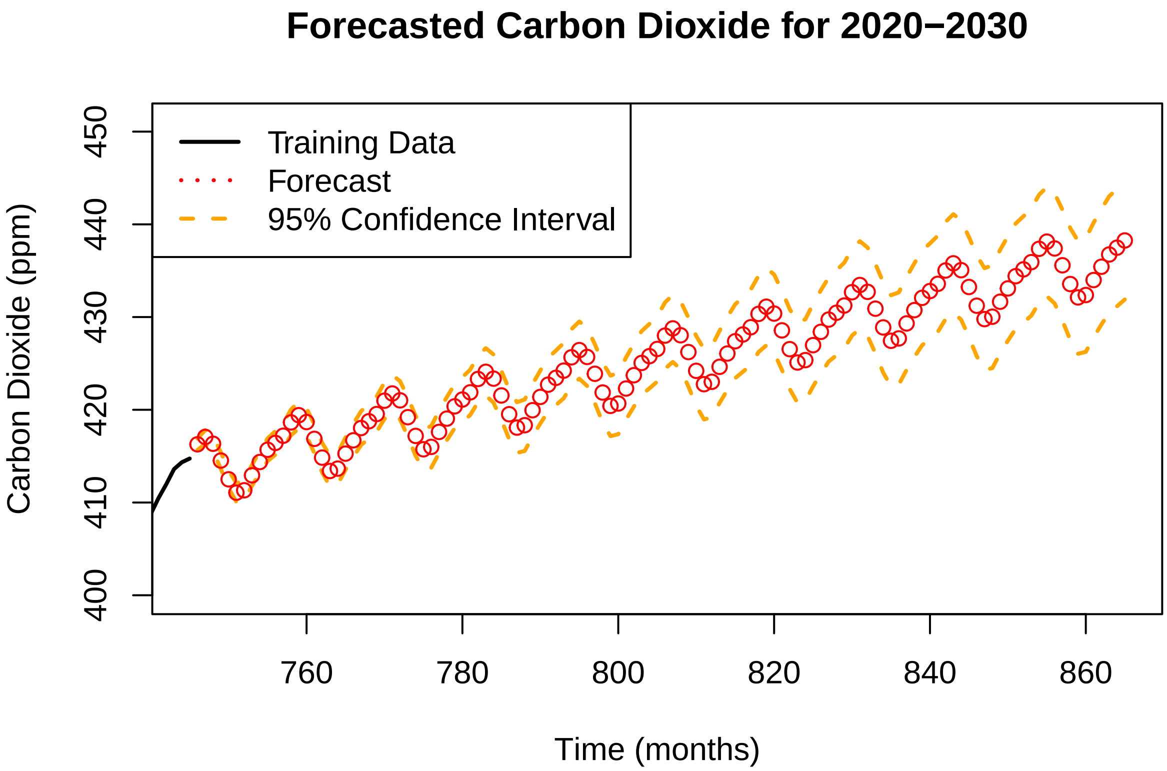
The ACF and PACF plots have no significant autocorrelations or partial-autocorrelations, aside from those at lag 37. However, due to the conservative nature of Bartlett's formula which calculates the error bounds and the relative proximity of said significant autocorrelations or partial-autocorrelations,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

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, along with MLE-based model fitting, 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
# Downloading the Mauna Loa Cardox Dataset from the astsa package
library(astsa)
data(cardox)

# Plotting the 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
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
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,
```

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

```r
# Aggregating the variance at each step of differencing
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")
```

```r
# Plotting the ACF and PACF for the stationary data
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")
```

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

```r
# Fitting model 2
mod2 <- arima(train, order=c(11,1,1),
               seasonal = list(order = c(0,1,1),
                               period = 12),
              method = "ML")
mod2
```

```r
# Adjusting model 2 to remove insignificant coefficients
mod2 <- arima(train, order=c(11,1,1),
               seasonal = list(order = c(0,1,1),
                               period = 12),
              method = "ML",
              fixed = c(0,0,0,0,0,0,0,0,0,0,0,0,NA))
mod2
```

```r
# Fitting model 3
mod3 <- arima(train, order=c(1,1,11),
               seasonal = list(order = c(0,1,1),
                               period = 12),
              method = "ML")
mod3
```

```r
# Adjusting model 3 to remove insignificant coefficients
mod3 <- arima(train, order=c(1,1,11),
               seasonal = list(order = c(0,1,1),
                               period = 12),
              method = "ML",
              fixed = c(0,0,0,0,0,0,0,0,0,0,0,0,NA))
mod3
```

```r
# Fitting model 4
mod4 <- arima(train, order=c(11,1,11),
               seasonal = list(order = c(0,1,1),
                               period = 12),
              method = "ML")
mod4
```

```r
# Adjusting model 4 to remove insignificant coefficients
mod4 <- arima(train, order=c(11,1,11),
               seasonal = list(order = c(0,1,1),
                               period = 12),
              method = "ML",
              fixed = c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,NA))
mod4
```

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

```r
# Performing Shapiro-Wilk test and Portmanteau tests
shapiro <- shapiro.test(res)

table <- matrix(
  c(shapiro$statistic, shapiro$p.value),
```

```r
  nrow = 1,
  ncol = 2,
  byrow = TRUE
)

colnames(table) = c("W", "p-value")
kable(table, caption = "Shapiro-Wilk Test")

# Using kableExtra to allow Latex to render inside of kable
library(kableExtra)
h <- round(sqrt(length(train)))
box_pierce <- Box.test(res, lag = h, type = c("Box-Pierce"), fitdf = 3)
ljung_box <- Box.test(res, lag = h, type = c("Ljung-Box"), fitdf = 3)
mcleod_li <- Box.test(res^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",
      escape = FALSE)
```

```r
# Fitting an autoregressive model to residuals
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```r
# Plotting the ACF and PACF of residuals
par(mfrow=c(1,2))
acf(res, lag.max=40,main="")
title("ACF of the residuals")
pacf(res, lag.max=40,main="")
title("PACF of the residuals")
```

```r
# Forecasting on the test data
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")
```

```r
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)
```

```r
# 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)
```