Forecasting U.S. New Plant & Equipment Expenditures (1964–1976)

Angel Alcantara

2025-03-18

Contents

1 Abstract

In this report, I used multiple time series techniques in order to get a good understanding behind the U.S.'s expenditures and why they decided to spend that much. First, I split my data into a training and testing data set in order to use the last 10 observations for testing. The data was seasonal and had a positive trend, therefore I decided to do a Box-Cox transformation to remove the variance, then I differenced it by lag 1

to remove the trend, and finally differenced it by lag 4 to remove the seasonality it had. The process went smoothly and I had no issues.

Then I checked the ACF and PACF in order to figure out what could be potential models. With these potential models, I checked them by using the arima function, estimated coefficients, and checking their roots. Once I decided on the model, I checked their residuals to make sure it was white noise, and then I did some forecasting with the new equation.

2 Introduction

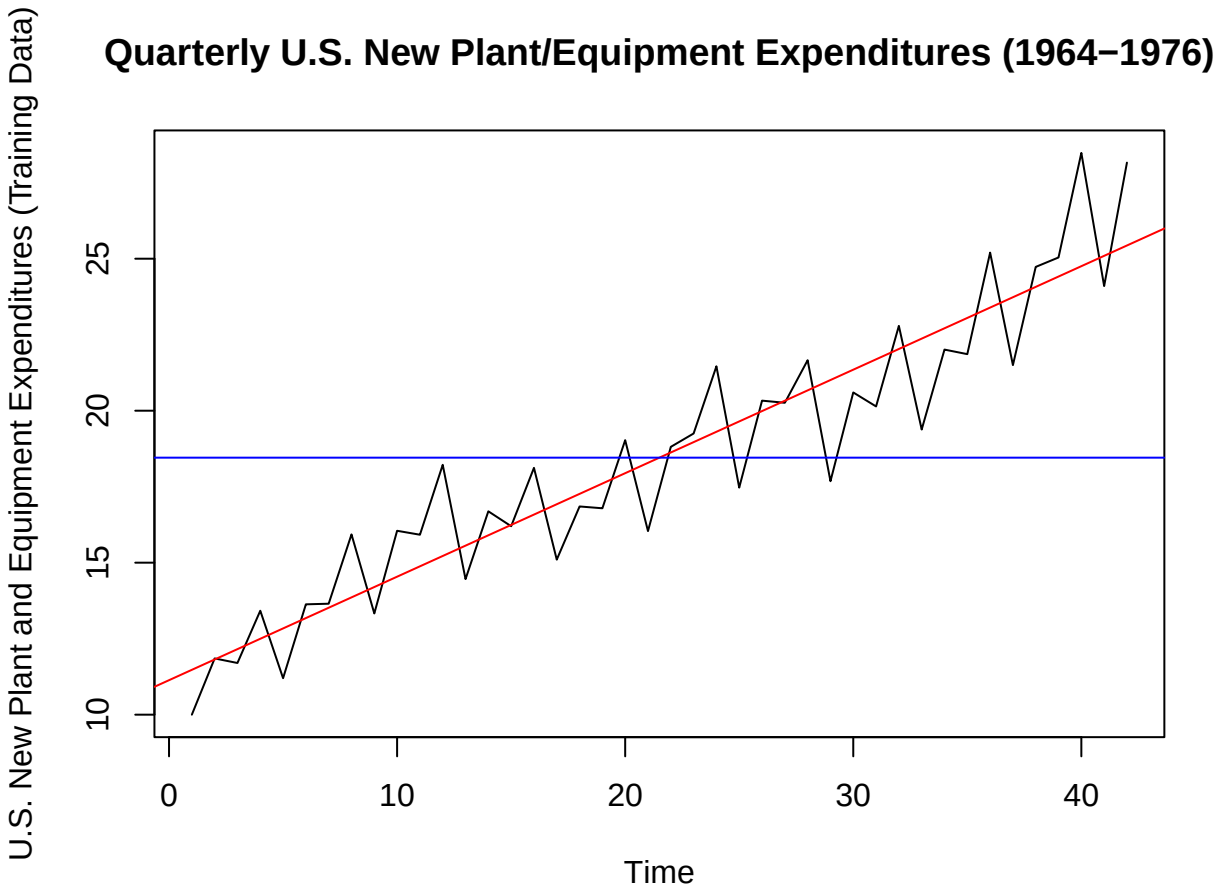
Can the U.S.'s expenditures be modeled in a way to predict how much they will spend in the future on new technology? The data set I decided to use was the "Quarterly U.S. new plant/equip. expenditures –64 – –76 billions" from the tsdl library in R. The data set has 52 observations that are divided quarterly from 1964 to 1976. I found this dataset interesting because it was about a topic that I have never looked into. I plan to forecast any future spending the U.S. might make on new equipment, especially with the constand advancement in technology. The techniques I plan to use are transformations, model fitting, and model comparison. I will do this by utilizing different functions in R, and as well as my own knowledge on time series. I acknowledge the use of RStudio and it's many available libraries with assisting me in this endevour.

3 Training/Testing Split

```r
# Dividing data to training and test sets
train_data = equip.ts[c(1:42)] # the rest of the observations
test_data = equip.ts[c(43:52)] # the last 10 observations

# Create a plot for the training set
plot.ts(as.numeric(train_data), cex.lab = 0.5,
        main = "Quarterly U.S. New Plant/Equipment Expenditures (1964-1976)",
        ylab = "U.S. New Plant and Equipment Expenditures (Training Data)")

## Generate trend and mean lines
len = length(as.numeric(train_data))# storing total number of observations in training set
fit_training <- lm(as.numeric(train_data) ~ as.numeric(1:len)) # fitted linear regression model estimat
abline(fit_training, col = "red") # trend line
abline(h=mean(as.numeric(train_data)), col = "blue") # mean of training set (horizontal line)
```



Main features of the graph:

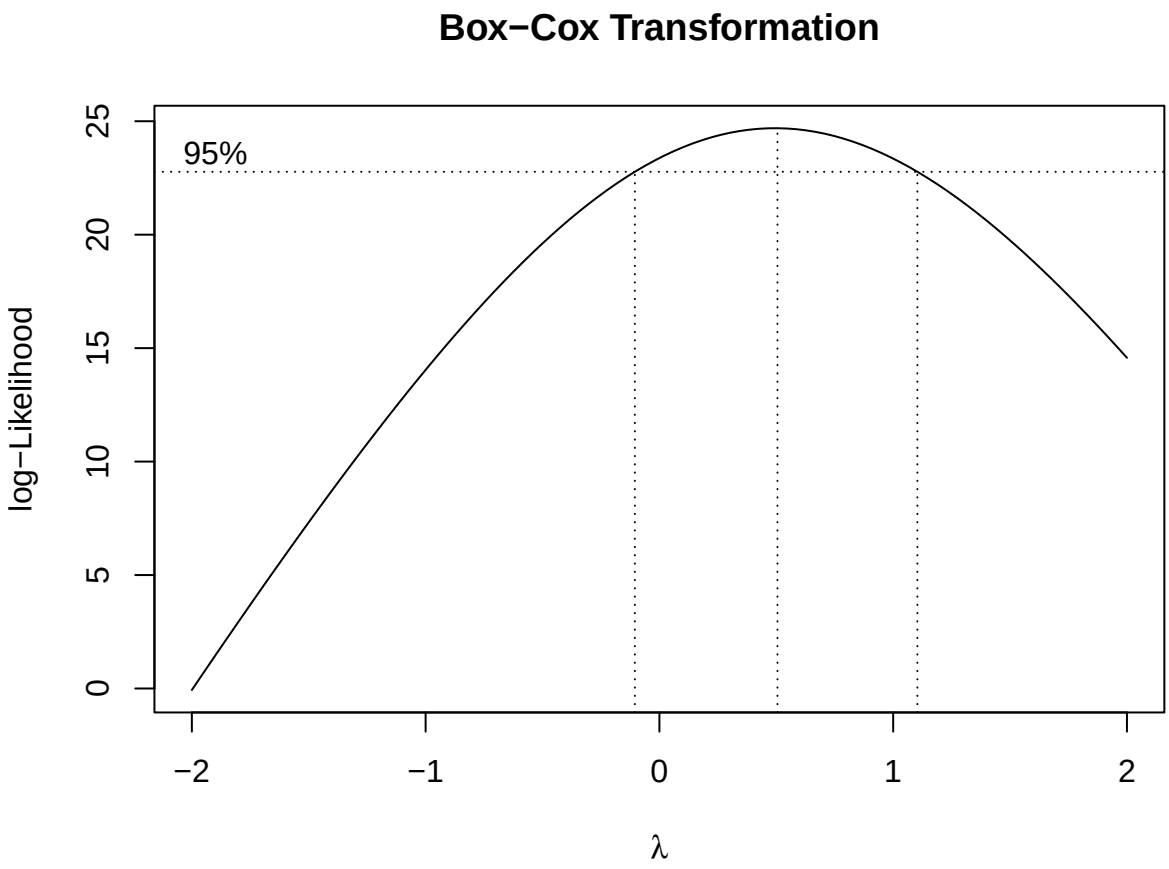
- **Trend:** There is a upward trend, and it is shown by the red line. This means that new plant and equipment expenditures by the U.S. have increased from 1964 to 1976.
- **Seasonality:** Seasonality is present, in the graph we can see in this order, one intial increase, then small drop, then small increase, and finally a significant, large drop. This pattern is noticeable every four quarters (1 year).
- **Any apparent sharp changes in behavior:** There are no apparent sharp changes in behavior, a very similar increase and decrease behavior can be seen as time moves forward. The pattern of gradual increases and decreases remains stable over time.
- **Variance:** Over time, the fluctuations seem to increase. Therefore, the variance is increasing over time.

4 Transformations

4.1 Stabilize the variance: Box-Cox Transformation

In the plot for our training data set, we can see that the variance of the data increases as time goes on. Therefore, the best approach is to first transform the data and stabilize the variance. The reasonable approach for this is to do a Box-Cox transformation.

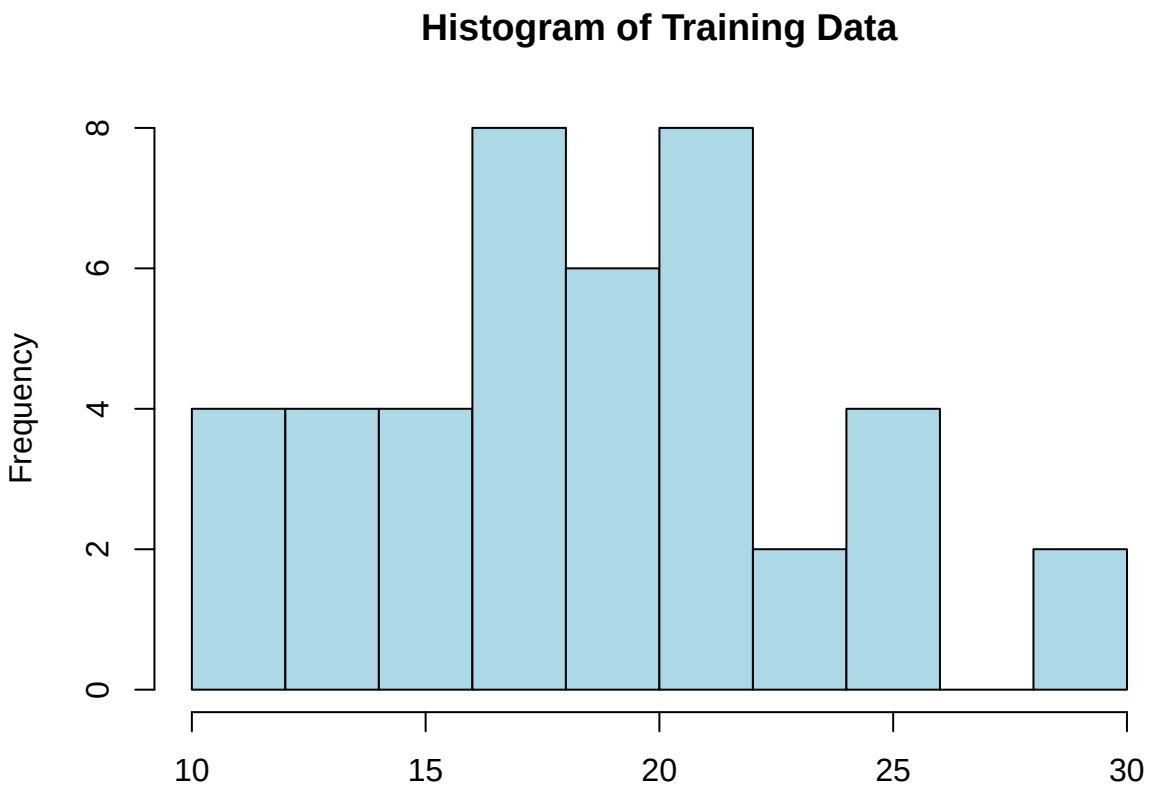
```r
# Box-Cox Transformation
bcTransform <- boxcox(as.numeric(train_data) ~ as.numeric(1:length(train_data)))
title(main = "Box-Cox Transformation")
```

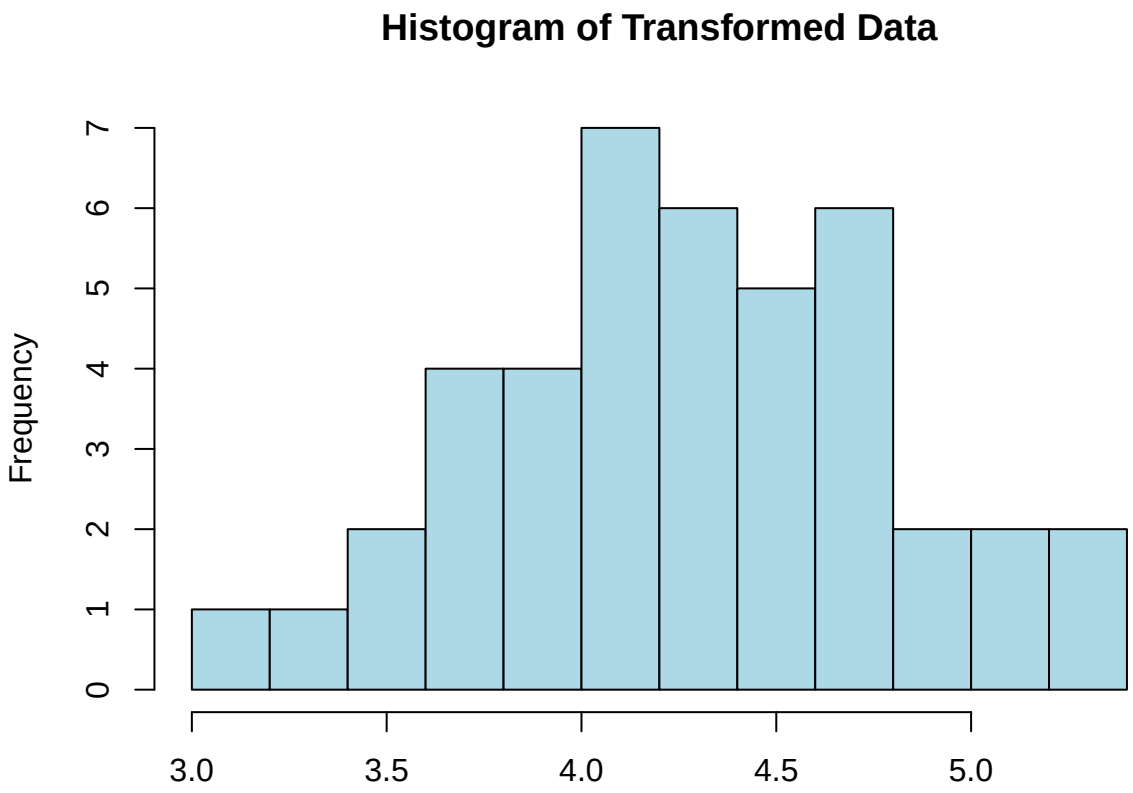


The above plot demonstrates the best λ to get the maximum log-likelihood, which is 0.5. This value will be used in order to do the Box-Cox transformation:

$$Y' = (Y^{\lambda} - 1)/\lambda$$

Where Y is the training data set and Y' is the transformed value

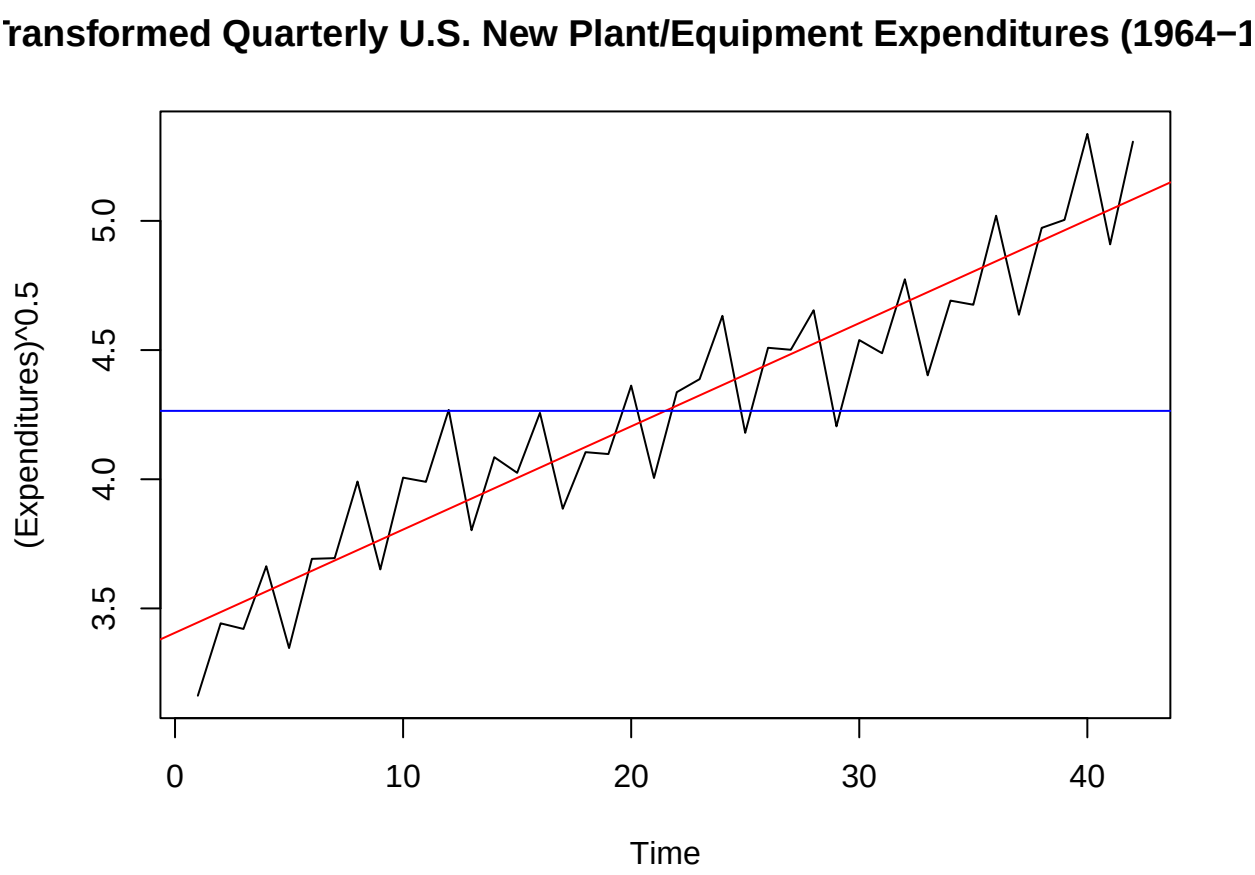

Histogram of Training Data
Frequency
0
2
4
6
8
10
15
20
25
30



The transformed data (after Box-Cox) looks more Gaussian (normal) compared to the original training data (before Box-Cox). Now, let's see how the training data looks after this transformation.

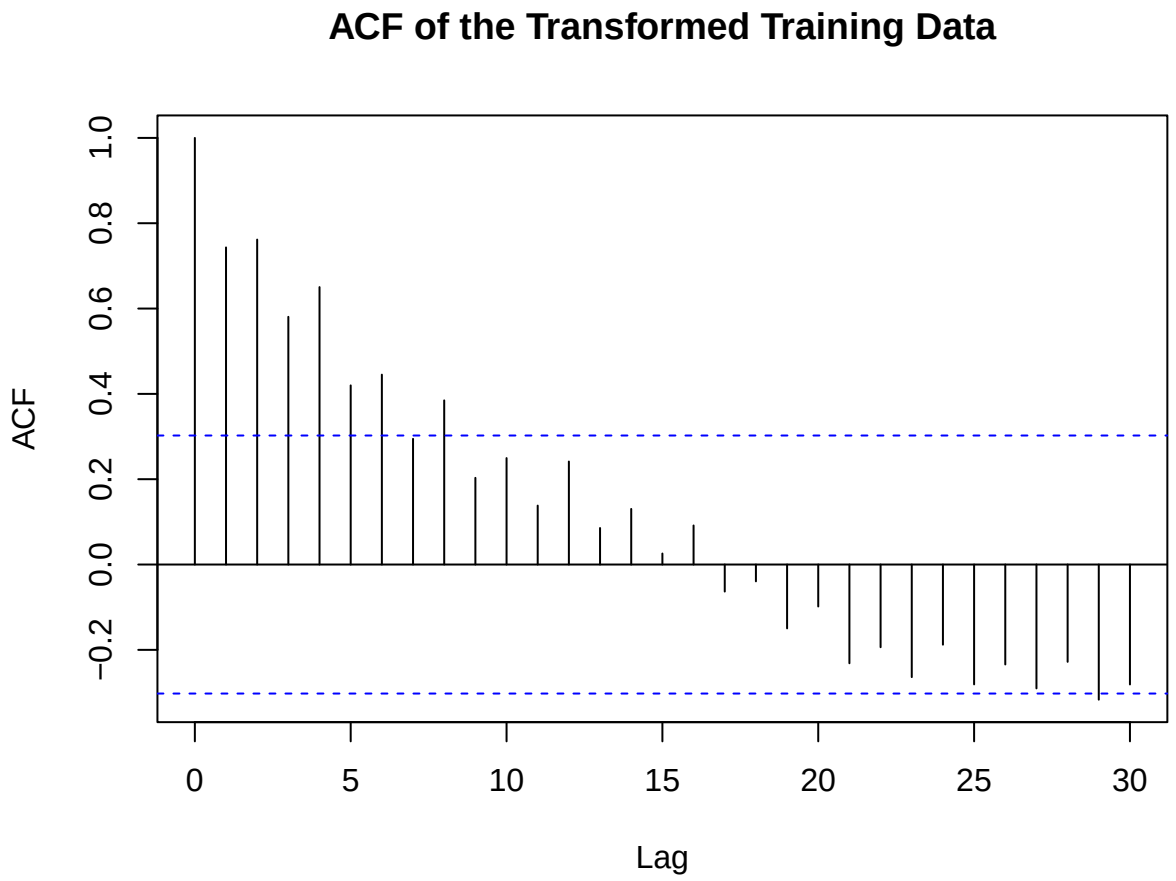
```
expend.bc_trans = as.numeric(expend.bc)
plot.ts(expend.bc_trans,
        main = "Transformed Quarterly U.S. New Plant/Equipment Expenditures (1964-1976) ",
        ylab = "(Expenditures)^0.5")

# To generate trend and mean lines
len = length(expend.bc_trans)
fit <- lm(expend.bc_trans ~ as.numeric(1:len))
abline(fit, col = "red")
abline(h = mean(expend.bc_trans), col = "blue")
```



The Box-Cox transformation has allowed the variance to start stabilizing over time. The only issue at the moment is that there is still a positive trend. Let's see what the ACF looks like and try to understadn our data a little better.

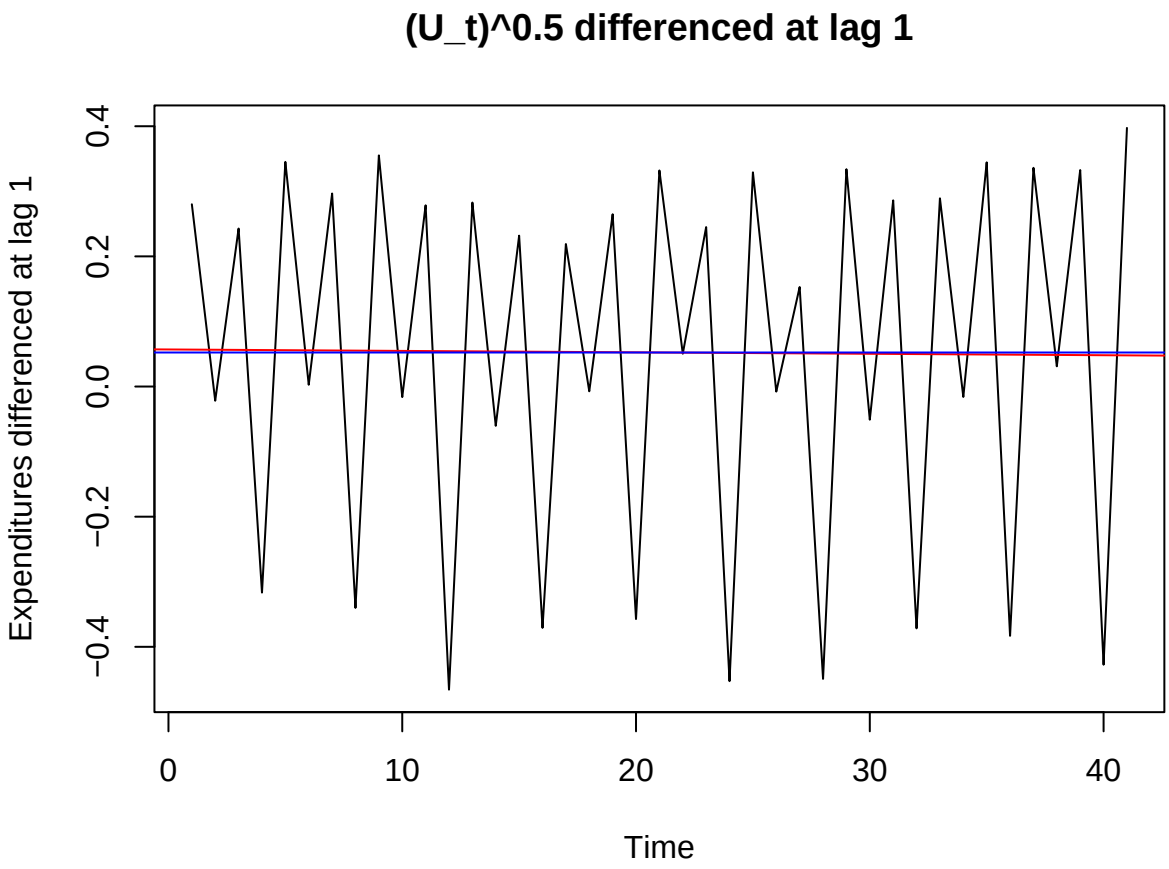
```
acf(expend.bc, lag.max = 30,
    main = "ACF of the Transformed Training Data")
```



From the Auto-Correlation Function (ACF) in the plot above, the ACF values decrease over time. Therefore, let's take a difference at lag 1 to eliminate the trend.

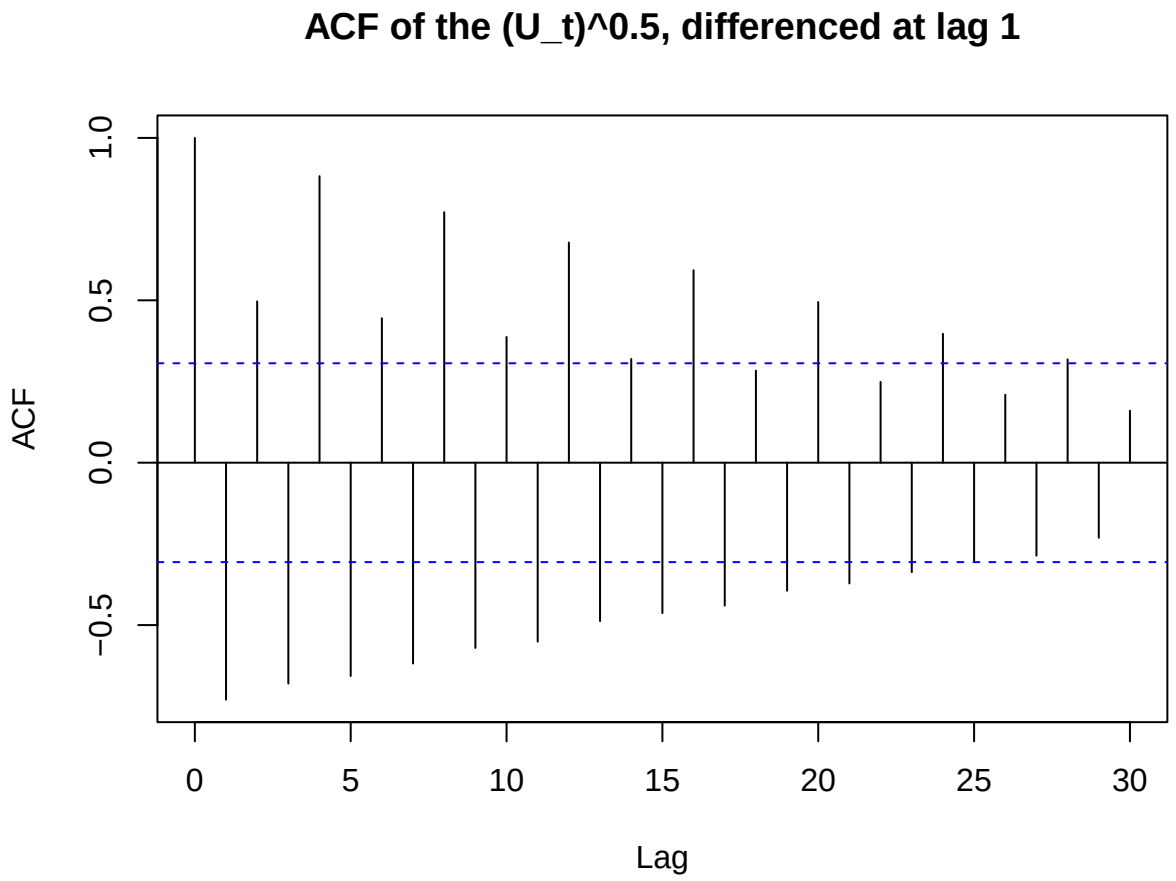
4.2 Removing the Trend

```
expend.lag_1 <- diff(expend.bc, lag = 1)
plot.ts(expend.lag_1,
        main = "(U_t)^0.5 differenced at lag 1",
        ylab = "Expenditures differenced at lag 1")
fit_1 <- lm(expend.lag_1 ~ as.numeric(1:length(expend.lag_1)))
abline(fit_1, col = "red")
abline(h = mean(expend.lag_1), col = "blue")
```



After taking the difference at lag 1, above is the plot of the difference data with time. It can be seen that the trend is gone because the mean is almost equal to 0 and the regression line (red line) is close to the mean line (blue line). Also, the variance of the data also decreased from 0.274 to 0.084.Therefore, differencing at lag 1 eliminates the trend. Let's see how the ACF has changed.

```
acf(expend.lag_1, lag.max = 30,
    main = "ACF of the (U_t)^0.5, differenced at lag 1")
```

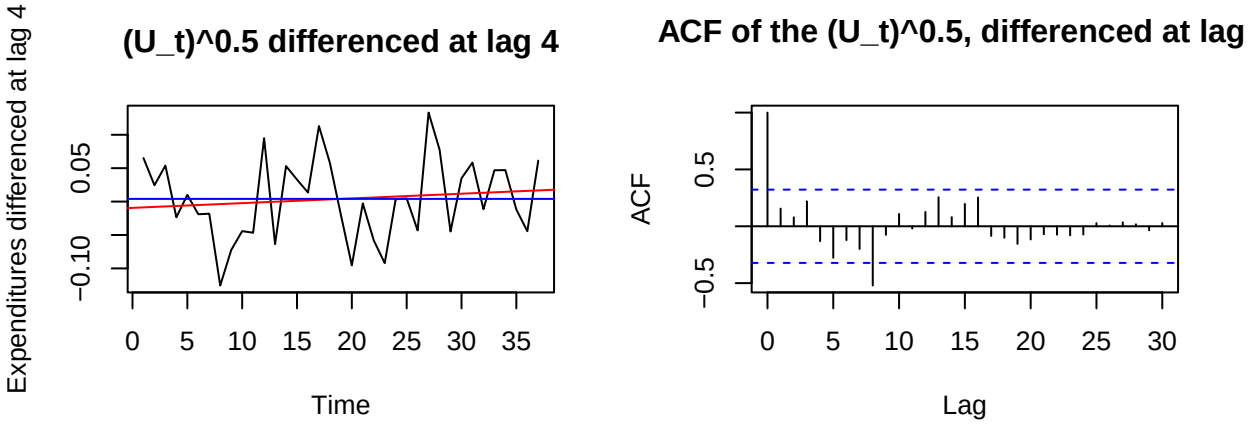


The ACF no longer shows a slow decay, which means that the trend is gone. The ACF also goes back and forth between positive and negative values in intervals, which demonstrates seasonality. Therefore, we differencing at lag 1 was a useful method.

4.3 Removing the Seasonality

```
par(mfrow=c(2,2))
plot.ts(expend.lag_4,
        main = "(U_t)^0.5 differenced at lag 4",
        ylab = "Expenditures differenced at lag 4")
fit_4 <- lm(expend.lag_4 ~ as.numeric(1:length(expend.lag_4)))
abline(fit_4, col = "red")
abline(h = mean(expend.lag_4), col = "blue")

# Plot ACF for series differenced at lag 4
acf(expend.lag_4, lag.max = 30,
    main = "ACF of the (U_t)^0.5, differenced at lag 4")
```



Time Series plot: The trend line (red line) is almost flat, and the mean does not change over time (blue line). The variance seems constant over time because the values are spread out randomly around zero.

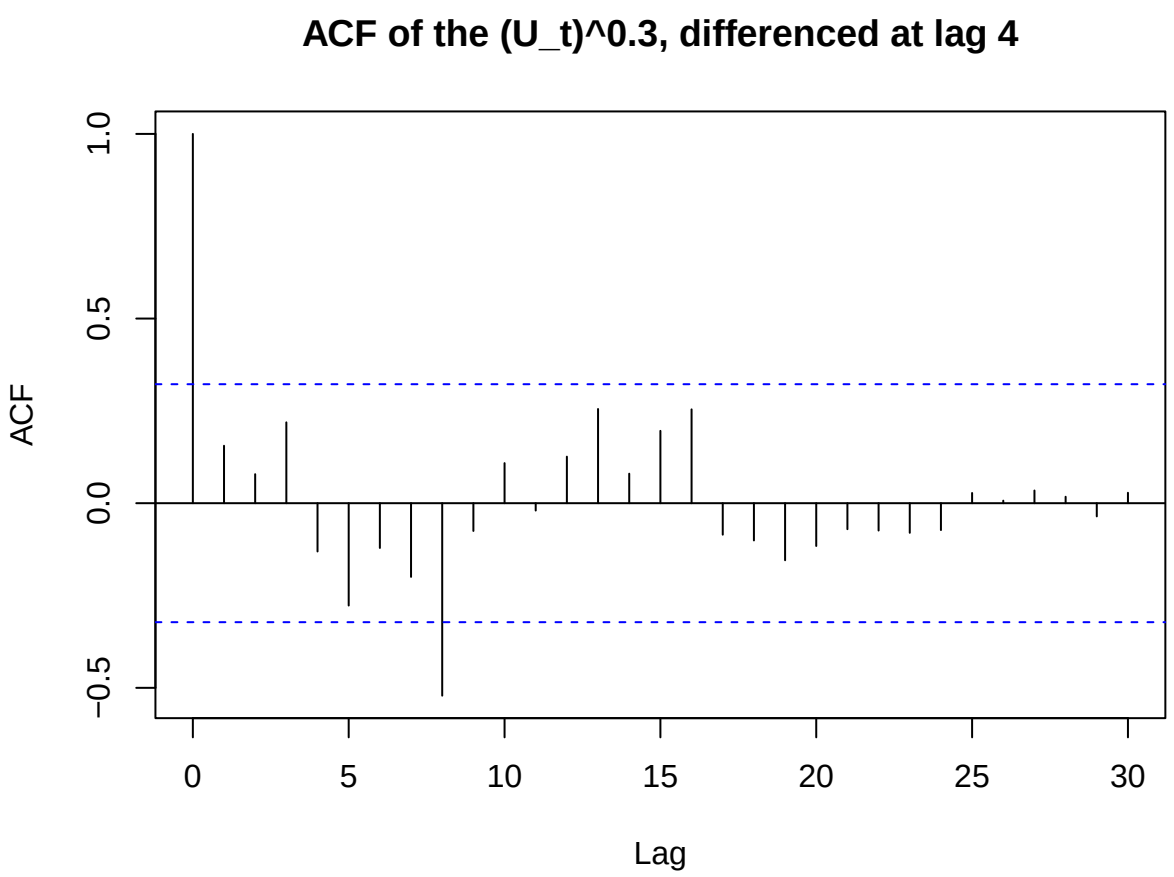
ACF: The high autocorrelation at lag 1 is nonexistent, so our trend is gone. The majority of the lags all have autocorrelations close to zero and inside the confidence interval (blue dashed line) beside lag 4. The ACF immediately drops to 0 from lag 0 to lag 1, which is a sign that the series is stationary.

Therefore, because of these observations, the series is now stationary.

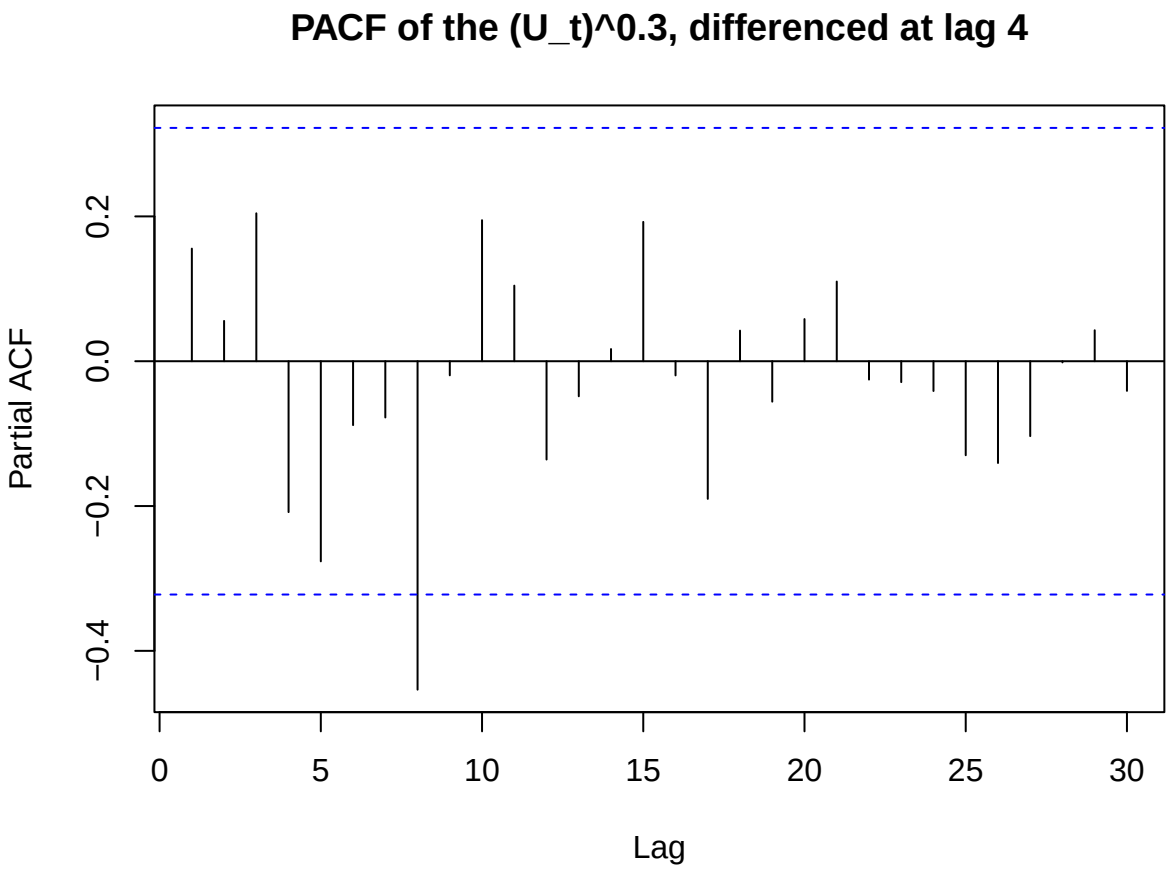
5 Model Identification

To preliminarily identify our model, we will plot and analyze the ACF and PACF. Since our data is seasonal (quarterly), we will look at potential SARIMA models.

```
acf(expend.lag_4, lag.max=30, main="")
title("ACF of the (U_t)^0.3, differenced at lag 4")
```




```
pacf(expend.lag_4, lag.max=30, main="")
title("PACF of the (U_t)^0.3, differenced at lag 4")
```



Identifying suitable **p**, **q**, **P**, **Q**:

- p (Non-seasonal MA term): 0 because the PACF has a sharp cutoff after lag 0 and there are no significant values at any other lags
- P (Seasonal AR term): 1 because there is a spike at lag 8 in PACF
- q (Non-seasonal MA term): 0 because the ACF cuts off quickly after lag 0 and there are no significant values at any other lags
- Q (Seasonal MA term): 1 because there is a significant spike at lag 8 in ACF

With these values, we can start comparing models and picking th best one for our training data.

6 Model Fitting

We will be comparing three models:

Model A: $SARIMA(0,1,0)(1,1,1)_4$

Model B: $SARIMA(0,1,0)(1,1,0)_4$

Model C: $SARIMA(0,1,0)(0,1,1)_4$

Using the arima function in R, we are going to estimate the coefficients of each model.

6.1 Estimated Coefficients

Estimated coefficients were calculated using arima function in R

6.1.1 Model A:

The estimated coefficient of the Seasonal AR(1) (SAR(1)) is 0.2497. This is an insignificant coefficient because of how small and close it is to zero. The estimated coefficient of the Seasonal MA(1) (SMA(1)) is -0.9998, which is extremely close to -1, and indicated there is a root that is outside of the unit circle, and this wouldn't be ideal.

6.1.2 Model B

The estimated coefficient of the Seasonal AR(1) (SAR(1)) is -0.1326, and it's standard error is greater, therefore it is likely 0 is in the confidence interval of the estimated coefficient. Therefore, this coefficient is insignificant.

6.1.3 Model C:

The estimated coefficient for the Seasonal MA(1) (SMA(1)) is -0.8898, and it is greater than two times the standard error, so it is considered significant.

The next part in deciding what our model will be is to check the AICc values of each model.

6.2 AICc

6.2.1 Model A: -105.9986

6.2.2 Model B: -99.92032

6.2.3 Model C: -106.712

We can see that Model C: $SARIMA(0,1,0)(0,1,1)_4$ is the best model according to AICc because it has the lowest one. There is still one more thing we need to go over, and that is checking each model's stationarity/inveritbility.

6.3 Stationary/Invertibility

Calculated roots using polyroot function

Let's find the roots of each model's characteristic functions.

6.3.1 Model A:

$$(1 - 0.2497B^4)(1 - B)X_t = (1 + 0.9998B^4)Z_t$$

and $X_t = U_t^{1/2}$ where U_t was the original data.

- SAR part: the root is 4, and this is greater than 1, therefore the seasonal AR(1) is stationary, and it is already invertible by definition.
- SMA part: the root is -1.0002, it is technically invertible, but this is not an ideal situation. It is already stationary by definition.

6.3.2 Model B:

$$(1 + 0.1326B^4)(1 - B)X_t = Z_t$$

and $X_t = U_t^{1/2}$ where U_t was the original data.

- SAR part: the root is about -7.54, which has an absolute value greater than 1, therefore the seasonal AR(1) is stationary, and it is already invertible by definition.

6.3.3 Model C:

$$(1 - B)(1 - B^4)X_t = (1 + 0.8898B^4)Z_t$$

and $X_t = U_t^{1/2}$ where U_t was the original data.

- SMA part: the root is about -1,1, which has an absolute value greater than 1, therefore the seasonal MA(1) is invertible, and it is already stationary by defintion.

Therefore, all three models pass the stationarity/invertibility check. According to AICc, we choose Model C: $SARIMA(0, 1, 0)(0, 1, 1)_4$

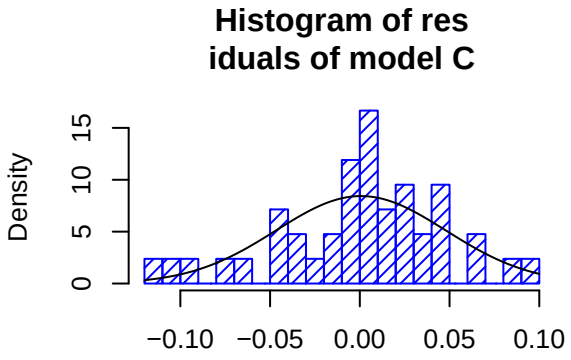
$$(1 - B)(1 - B^4)X_t = (1 + 0.8898B^4)Z_t$$

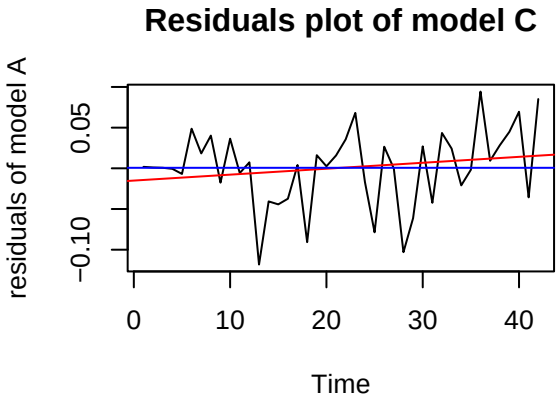
and $X_t = U_t^{1/2}$ where U_t was the original data.

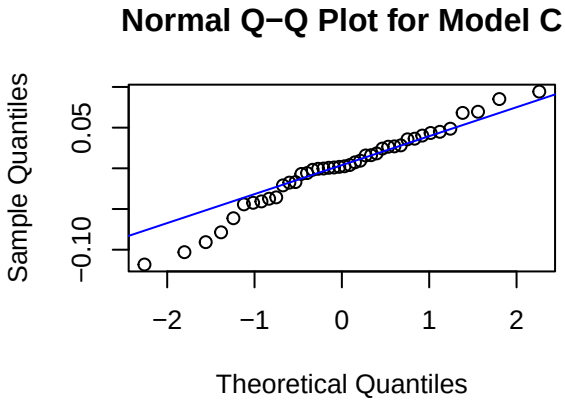
6.4 Diagnostics Checking

To check if the residuals of Model C follow White Noise distribution, we perform several diagnostic tools in this section. We first check normality assumptions and get the plots below.

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







We can observe that it roughly follow a normal distribution from the histogram and q-q plot. Also, there is no trend or obvious seasonality from the time series plot of the residuals.

Then, we also check for several independence assumptions by Portmanteau Statistics.

6.4.1 Shapiro Test for Normality

The p-value (0.3299) is greater than 0.05, so the residuals are normal.

6.4.2 Box-Pierce Test

The p-value (0.3449) is greater than 0.05, so the residuals are normal.

6.4.3 Box-Ljung Test

The p-value (0.2536) is greater than 0.05, so the residuals are normal.

6.4.4 McLeod-Li Test

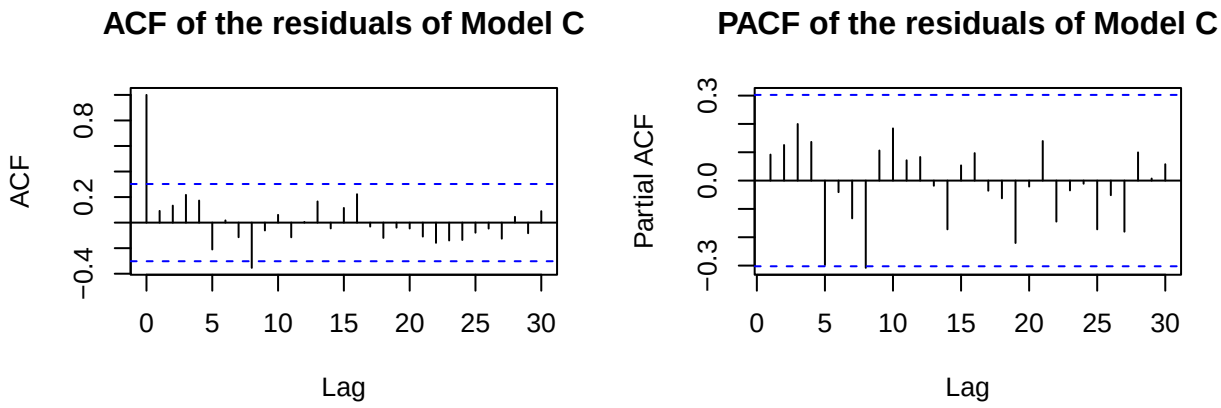
The p-value (0.7736) is greater than 0.05, so the residuals are normal.

We can see that the p-values are larger than 0.05 for all the tests, therefore model B passes all the test.

Next, we will see if the residuals are White Noise by using the ar() function. If it recommends order 0 result for the residuals, then the residuals are White Noise.

The order selected was 0, therefore the residuals are White Noise.Finally, we will make an ACF and PACF of the residuals.

```
par(mfrow = c(2,2))
acf(res_c, lag.max=30,main="")
title("ACF of the residuals of Model C")
pacf(res_c, lag.max=30,main="")
title("PACF of the residuals of Model C")
```



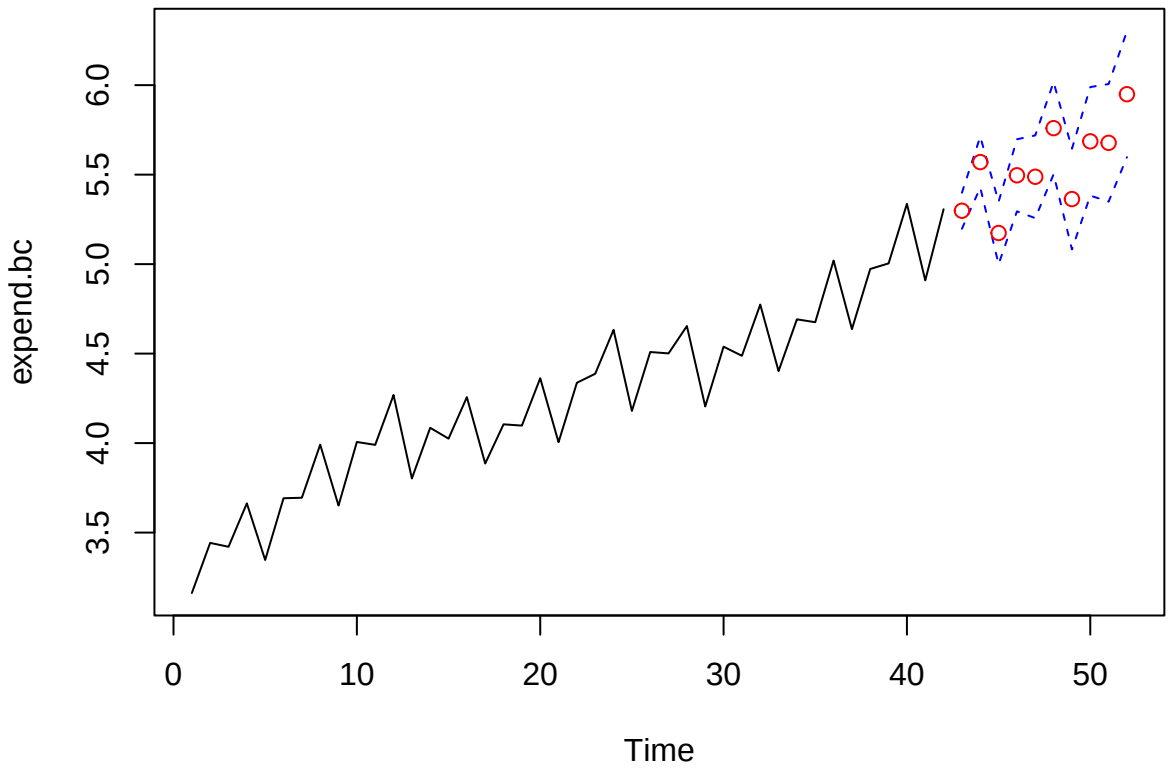
ACF and PACF values at all lags are within the confidence interval. PACF at lag 5 is outside but could be considered a borderline case. So the residuals can be seen as White Noise.

Therefore, model C passes all diagnostics checking and is ready to use for forecasting.

7 Forecasting

In forecasting section, we predict the new plant/equipment expenditures from halfway of 1974 to 1976 (quarterly) based on our model and then compare it with the true values.

```
pred.tr <- predict(fit_c, n.ahead = 10)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(expend.bc, xlim=c(1,length(expend.bc)+10), ylim = c(min(expend.bc),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(expend.bc)+1):(length(expend.bc)+10), pred.tr$pred, col="red")
```



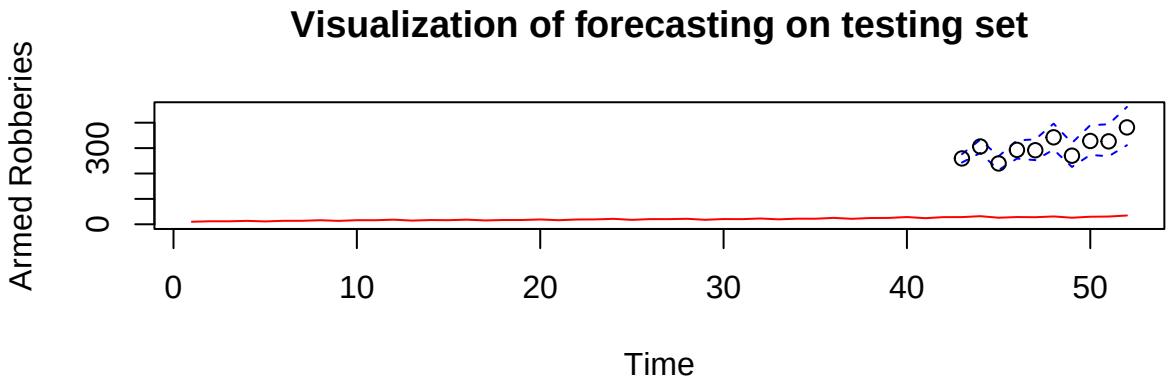
The above figure is the forecast on the transformed data. The true values are within the confidence interval of the forecasting. If we want to compare with the true values of the last 10 quarters (3 months/quarter), we need to convert the forecasting values back to the scale before box-cox transformation.

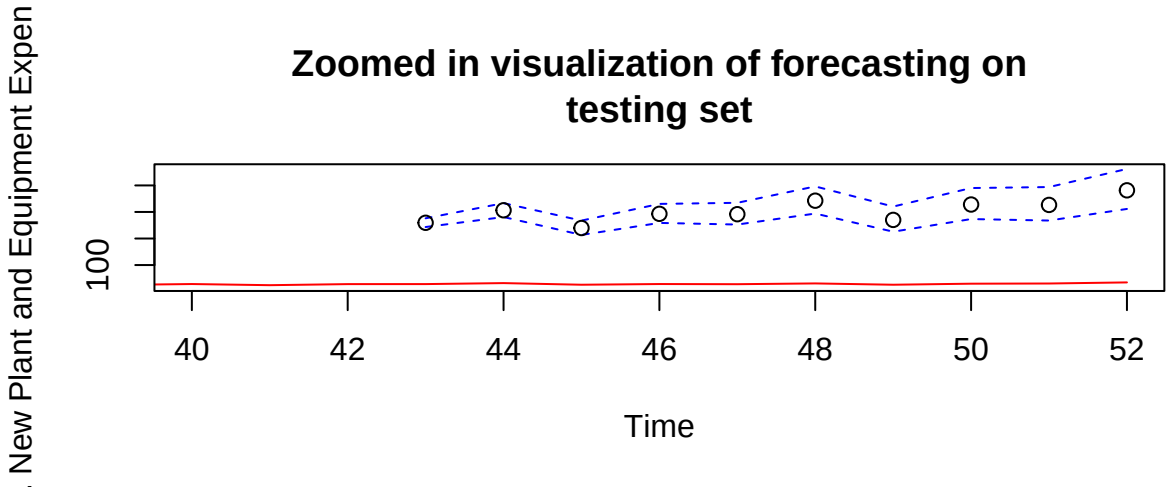
This part shows how to convert the data back to the scale before box-cox transformation and compare the true values with predicted values.

```
pred.orig <- (pred.tr$pred)^(1/0.3)
U= (U.tr)^(1/0.3)
L= (L.tr)^(1/0.3)
par(mfrow=c(2,1))
ts.plot(as.numeric(equip.ts), ylim = c(0,max(U)),col="red",
ylab="Armed Robberies",main="Visualization of forecasting on testing set")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
```

```
points((length(train_data)+1):(length(train_data)+10), pred.orig, col="black")


ts.plot(as.numeric(equip.ts), xlim = c(40,length(train_data)+10), ylim = c(20,max
(U)),col="red",ylab="U.S. New Plant and Equipment Expenditures",main="Zoomed in visualization of foreca
testing set")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(train_data)+1):(length(train_data)+10), pred.orig, col="black")
```





The true values are within the confidence interval of the forecasting. If we see the zoomed version from Lower part, the forecasting results are very close to the true values. Therefore, our model performs well in forecasting for future U.S. new plant/equipment expenditures.

My goal in this project was to be able to determine future spending for the U.S., and I believe I have been able to reach that goal. Although, there are some flaws, I'm satisfied with the progress I made with my model.

$$(1-B)(1-B^4)X_t = (1+0.8898B^4)Z_t$$

8 References

Bartoń K (2024). *MuMIn: Multi-Model Inference*. R package version 1.48.4, https://CRAN.R-project.org/package=MuMIn.

Hyndman R, Yang Y (2025). *tsdl: Time Series Data Library*. R package version 0.1.0, "Quarterly U.S. new plant/equip. expenditures –64 – –76 billions" commit 56e091544cb81e573ee6db20c6f9cd39c70e6243, https://github.com/FinYang/tsdl.

Hyndman R, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O'Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F (2024). *forecast: Forecasting functions for time series and linear models*. R package version 8.23.0, https://pkg.robjhyndman.com/forecast/.

Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0