Production of Gas Time Series Data

Yosua Saputra

June 2021

Abstract:

The data set analyzed contains monthly production of Gas in Australia measured per million mega-joules from January 1956 - August 1995. The original data set was taken from the time series data library in R and for the purpose of this report I filtered the dates to April 1969 – April 1981. My goal for this project is to compare and contrast my 1-year forecast data to its original data, furthermore, examining how accurate my results are using several time series techniques.

To start, I extracted a train/test data from my data set where I set the last 12 observations of my data set as my test data and the remaining observations as my training data. Using Box-Cox and lambda value, plotting histograms to check normal distribution, studying the respective ACF and PACF, I am able to reduce variance, eliminate seasonality, and transform the non-stationary time series graph of the data. From the transformed, stationary time series plot, I further analyzed the updated ACF and PACF to help select possible parameters for the model estimation. Diagnostic procedures include Shapiro-Wilk's normality, Box-Pierce, Box-LJung, and McLeod test of our model's residuals. When the test passed, the model is then used for forecasting. My predicted results showed great similarity to the testing data and the original data analyzed.

Introduction / Interest:

I have always been intrigued by how our world is evolving. Interestingly enough, the world today relies on the use of natural gas to generate electricity, heat, and other industrial uses. Examples like the emerging of electric cars, increased population, or revolutionary technologies are of many reasons I found this data set of our past to be quite interesting and possibly important to take from. Dating back to around 50 years ago it is no surprise that gas usage was much lower in our everyday lives, which correlates to the production of Gas. The data I have chosen contains a time series of monthly production of gas in Australia around 50 years ago. In hindsight, production increases and continues to do so and my conclusions for this project will be how off my predictions are to the inevitable.

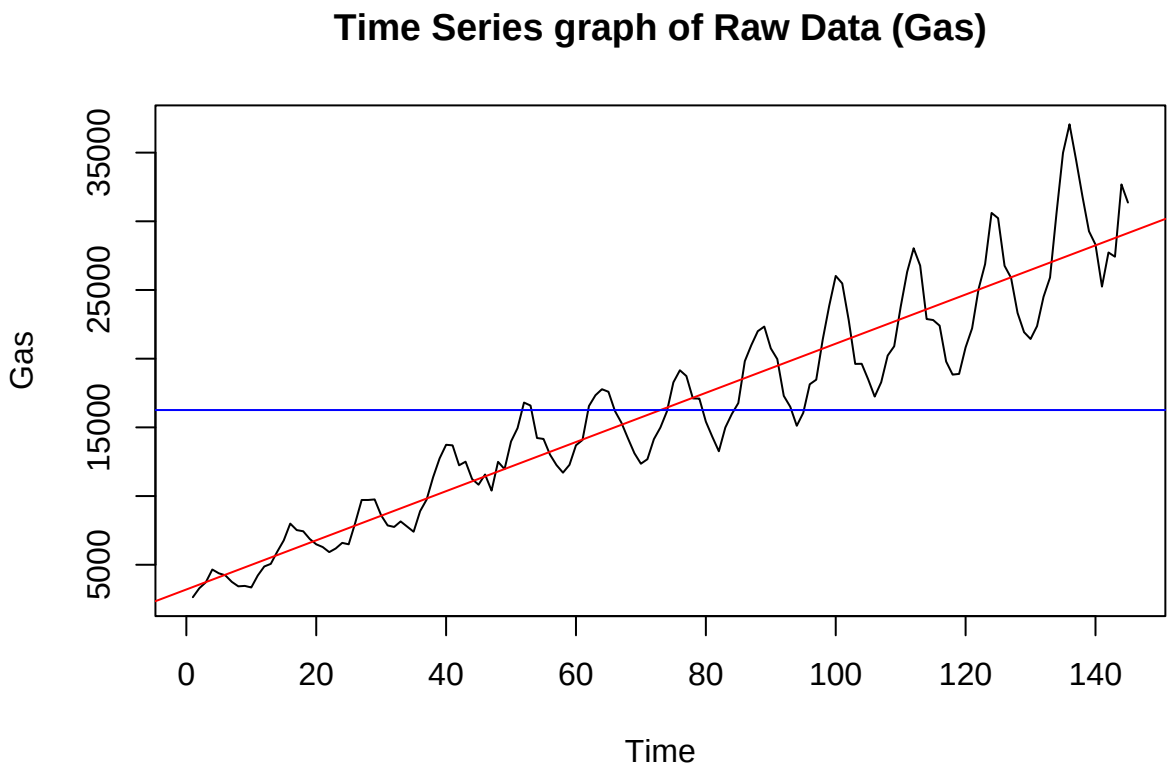
Source: This report is constructed using Rmarkdown

Data Processing:

To begin, I use the Time Series Data Library series #129 set as gas_data and will filter the data out to obtain my raw data set as Gas.

```
gas_data <- tsdl[[129]] #Monthly production of Gas in Australia: million mega-joules
#Jan 1956 - Aug 1995
Gas <- gas_data[c(160:304)] #Monthly production of Gas in Australia: million mega-joules.
#April 1969 - April 1981
```

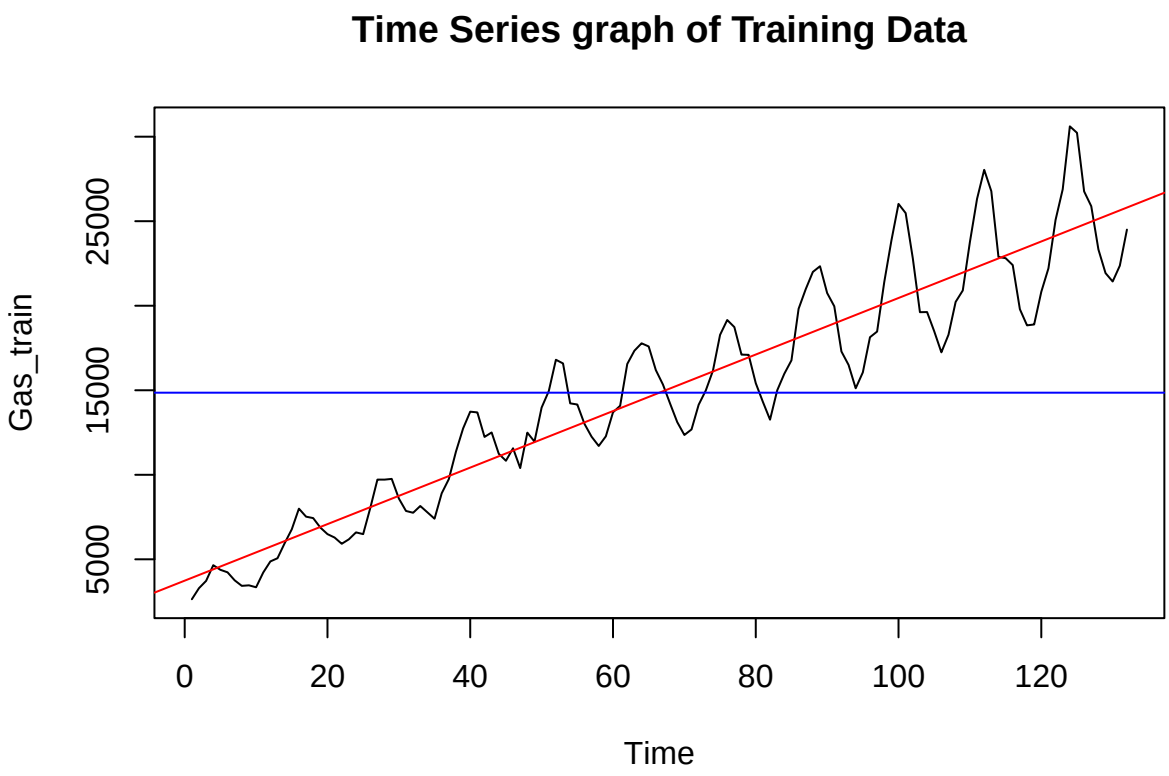
```r
ts.plot(Gas, main = "Time Series graph of Raw Data (Gas)")
nt <- length(Gas)
fit <- lm(Gas ~ as.numeric(1:nt))
abline(fit, col= "red")
m <- mean(Gas)
abline(h=m, col="blue")
```



I divide my raw data into two subsets, training & test, and will be using the training data for this project.

```r
Gas_train <- Gas[c(1:132)]  #training data (April 1969 - April 1980)
Gas_test <- Gas[c(133:144)] #testing data  (May 1980 - April 1981)

ts.plot(Gas_train, main = "Time Series graph of Training Data")
ts.plot(Gas_train, main = "Time Series graph of Training Data")
nt <- length(Gas_train)
fit <- lm(Gas_train ~ as.numeric(1:nt))
abline(fit, col= "red")
m <- mean(Gas_train)
abline(h=m, col="blue")
```



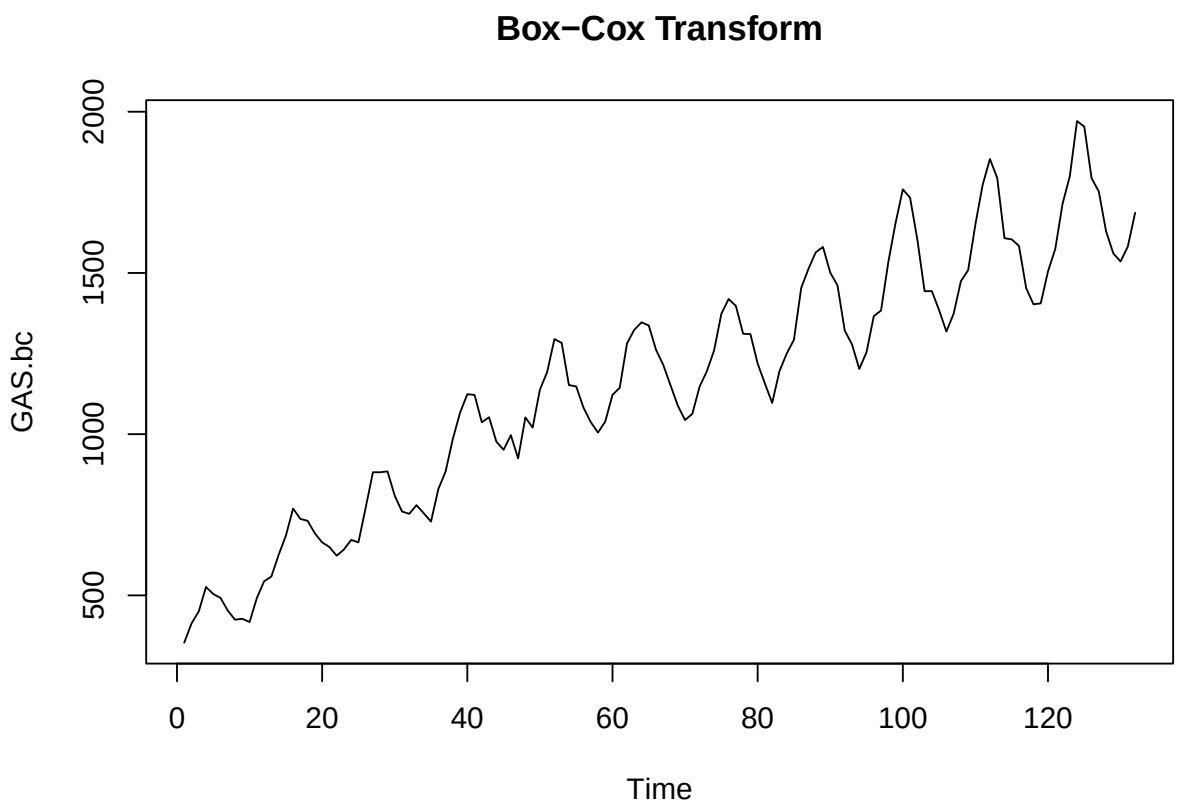
From here, our training data will be the data analyzed in this project. We can see conclude three things:

1. Data set is non stationary
2. Data shows a positive linear trend
3. Data has seasonality.

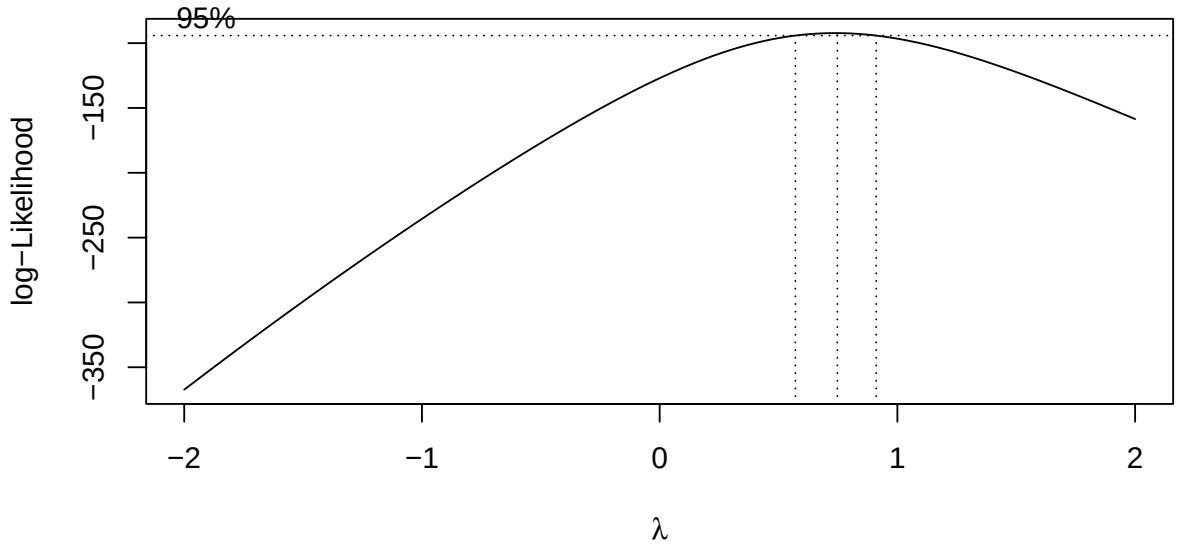
Also, the variance is unstable over time. In the start of the time series, variance was much smaller compared to the more recent time.

We have to stabilize its variance using transformation and remove seasonality by differencing. I checked Box-Cox transformation to see what it suggest for transformation:

```
#Box-Cox
t = 1:length(Gas_train)
bcTransform = boxcox(Gas_train ~ t,plotit = FALSE)
lambda = bcTransform$x[which(bcTransform$y ==max(bcTransform$y))]
GAS.bc = (1/lambda)*(Gas_train^lambda-1)
ts.plot(GAS.bc, main = "Box-Cox Transform")
```




```
bcTransform <- boxcox(Gas_train~ as.numeric(1:length(Gas_train)))
```



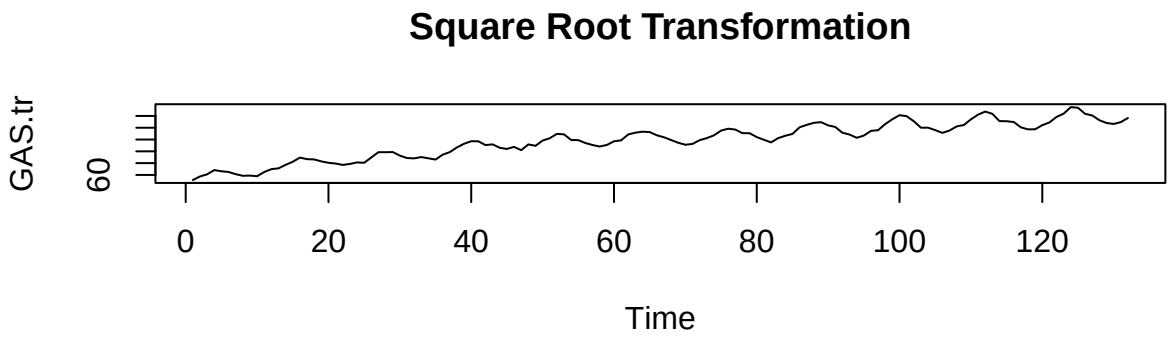
```
bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
```

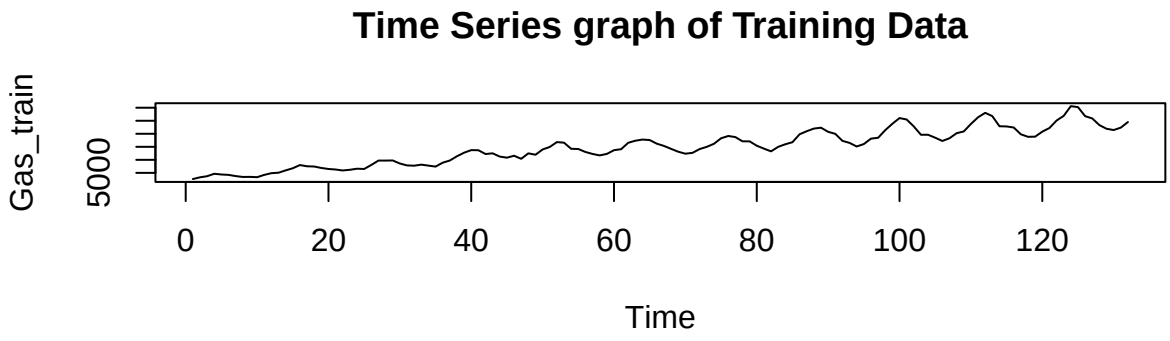
```
## [1] 0.7474747
```

Box-Cox suggest lambda value 0.74747, however I will use transformation of square roots because it is the nearest to lamba's confidence intervals.

```
#Transformation: square root
GAS.tr <- sqrt(Gas_train)

par(mfrow = c(2,1))
ts.plot(GAS.tr, main = "Square Root Transformation")
ts.plot(Gas_train, main = "Time Series graph of Training Data")
```

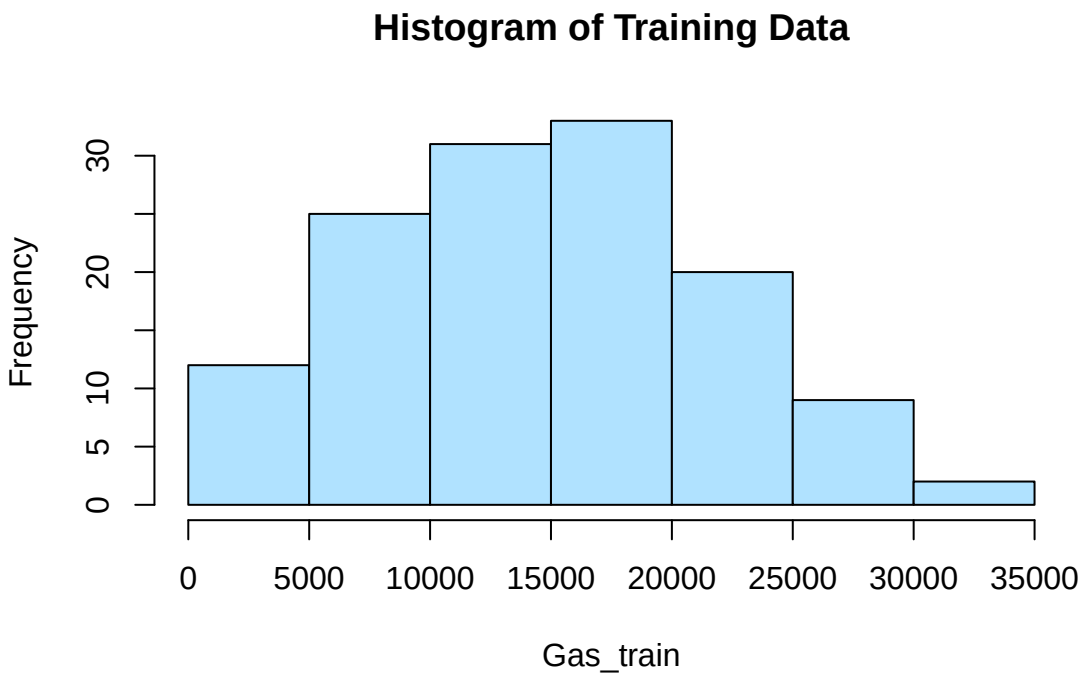




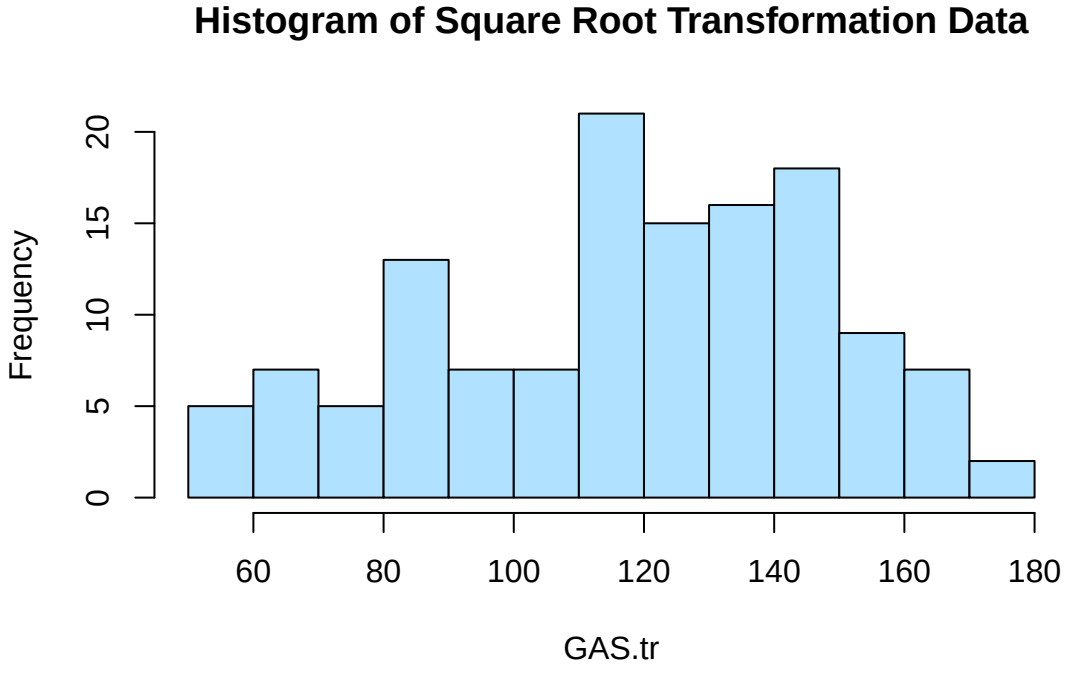
From comparing the two graphs we can see the variance is much more stable after transformation. In detail, the variance in the first half of the transformation graph does not look much different than its second half as oppose to the variance throughout the training data.

I now plot the histograms:

```
hist(Gas_train, col = "lightskyblue1", main = "Histogram of Training Data")
```




```
hist(GAS.tr, col = "lightskyblue1", main = "Histogram of Square Root Transformation Data")
```

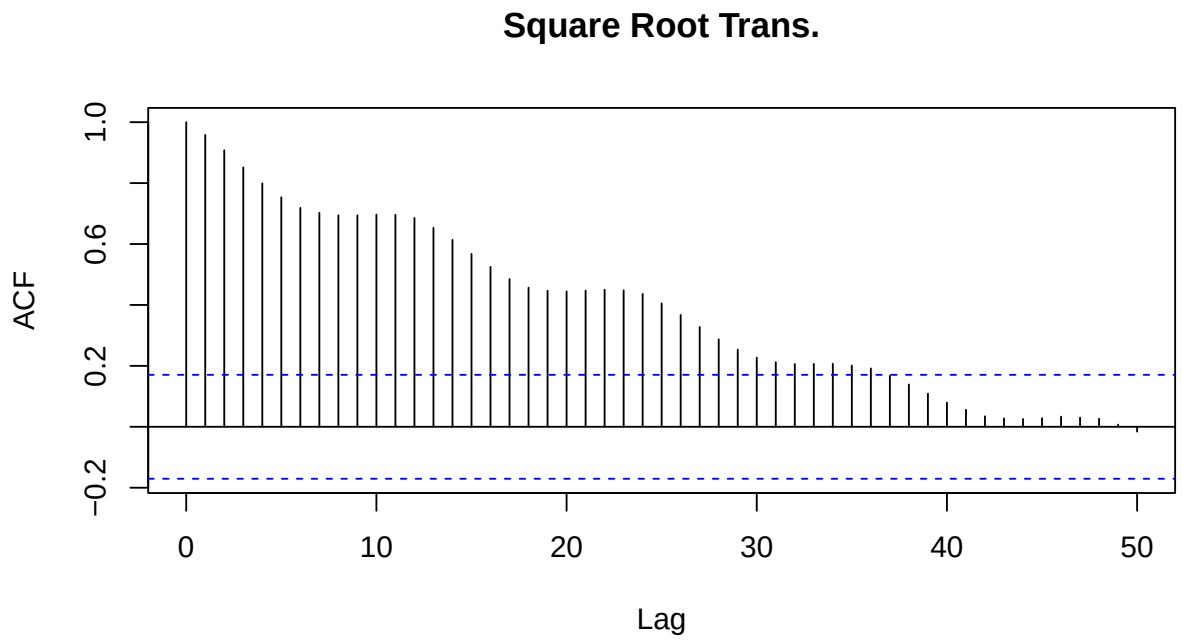


From the histograms it seems the training data is more normally distributed but also difficult to say both are normally distributed. However, because the variance is much more stable at square root transformation

I will proceed to use it and check its histogram after differencing.

Plotting the ACF to check if there is seasonality:

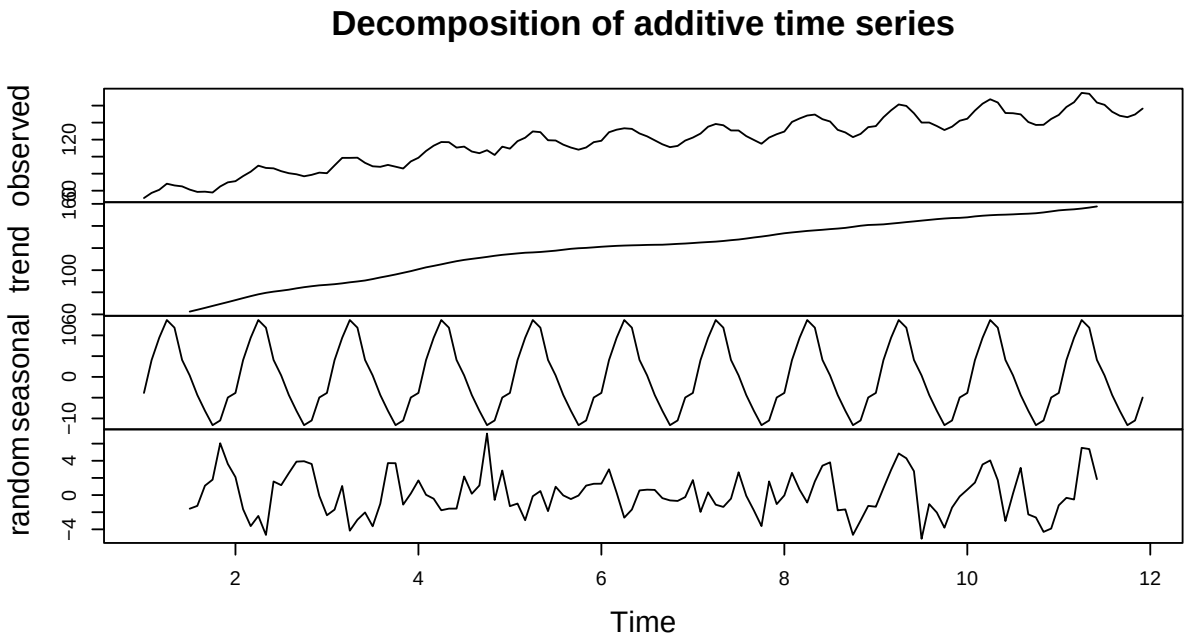
```
acf(GAS.tr, lag.max = 50, main = "Square Root Trans.")
```



The ACF for the data indicates there is seasonality. Therefore, I would have to difference my data to remove seasonality.

Decomposition of the model:

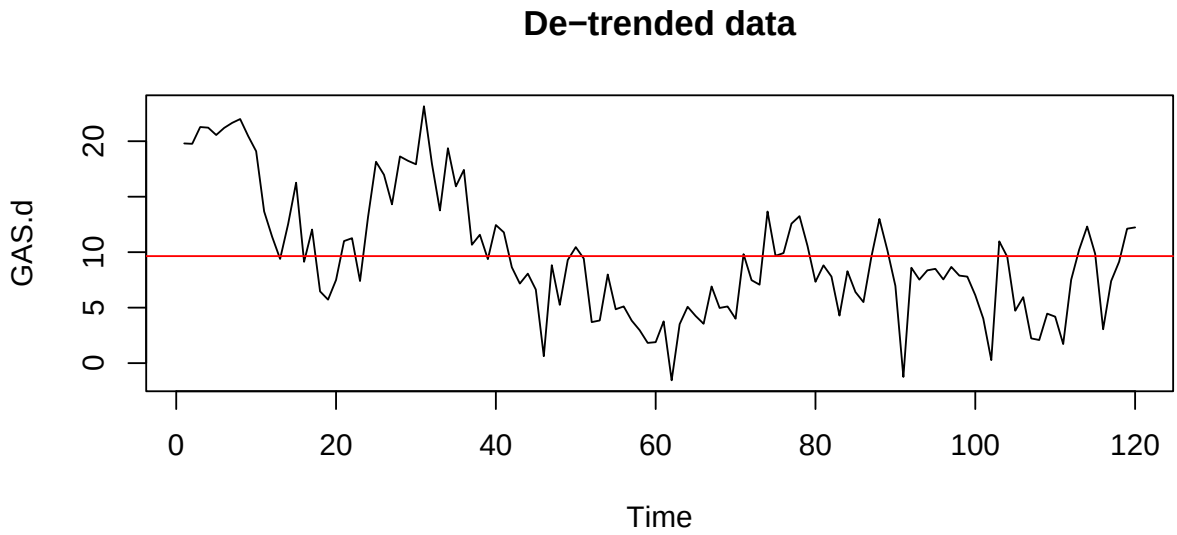
```
y <- ts(as.ts(GAS.tr), frequency = 12)
decomp <- decompose(y)
plot(decomp)
```



The decomposition of my training data shows that there is a trend and seasonality. Therefore, we can move over to transformation.

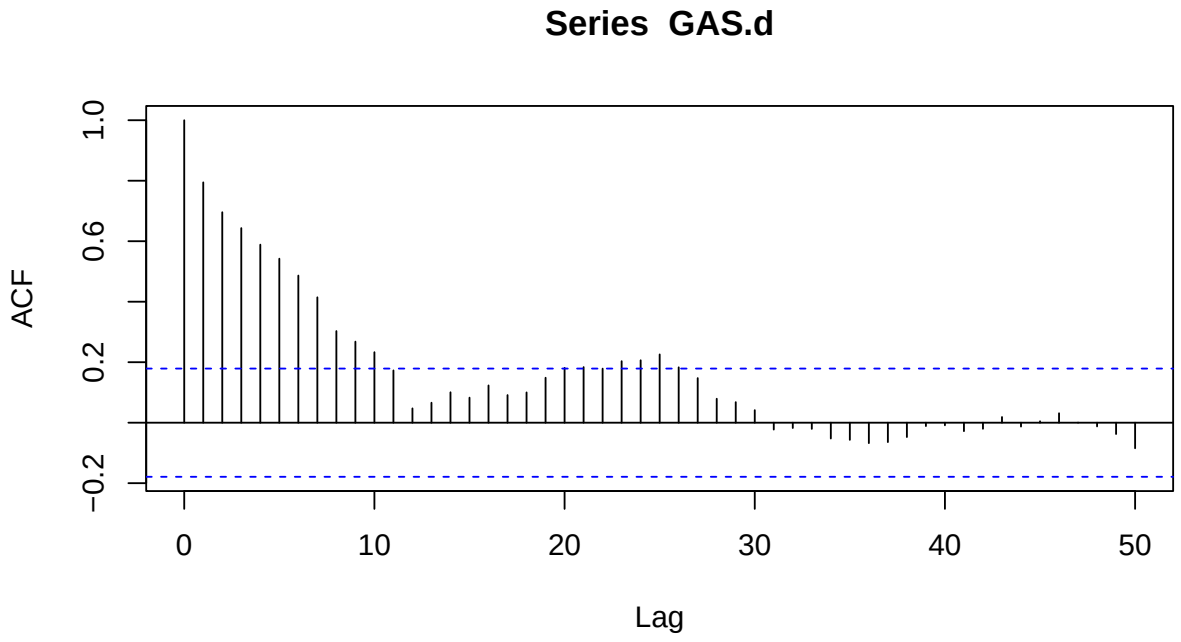
Differencing:

```
GAS.d <- diff(GAS.tr, lag =12)
ts.plot(GAS.d, main = "De-trended data")
abline(h= mean(GAS.d), col = 'red')
```

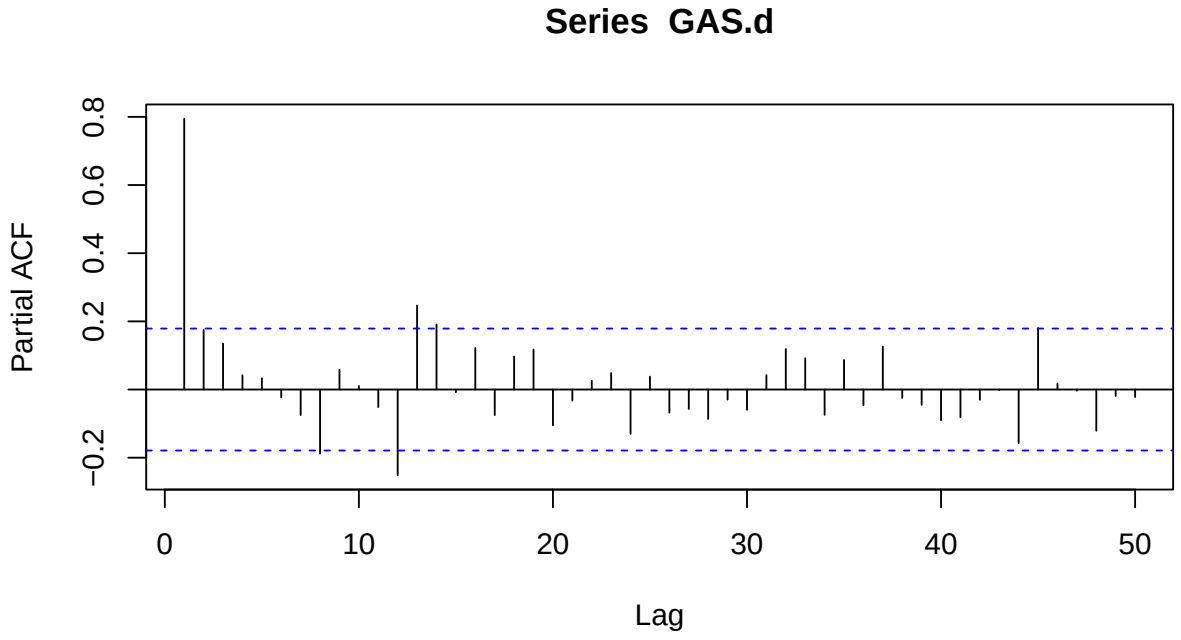


From diffencing at lag 12, we can see that seasonality is removed. Now I will check the ACF and PACF.

```r
acf(GAS.d, lag.max = 50)
```




```r
pacf(GAS.d, lag.max = 50)
```

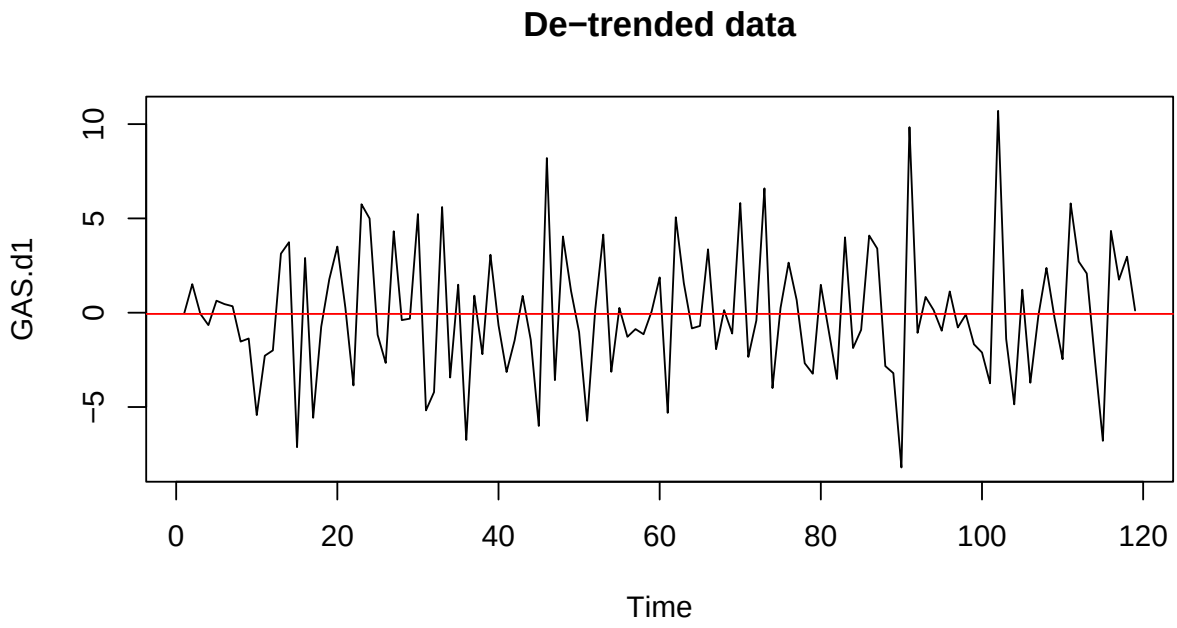


```r
paste("Variance:", var(GAS.d) )
```

```
## [1] "Variance: 31.6452960722791"
```

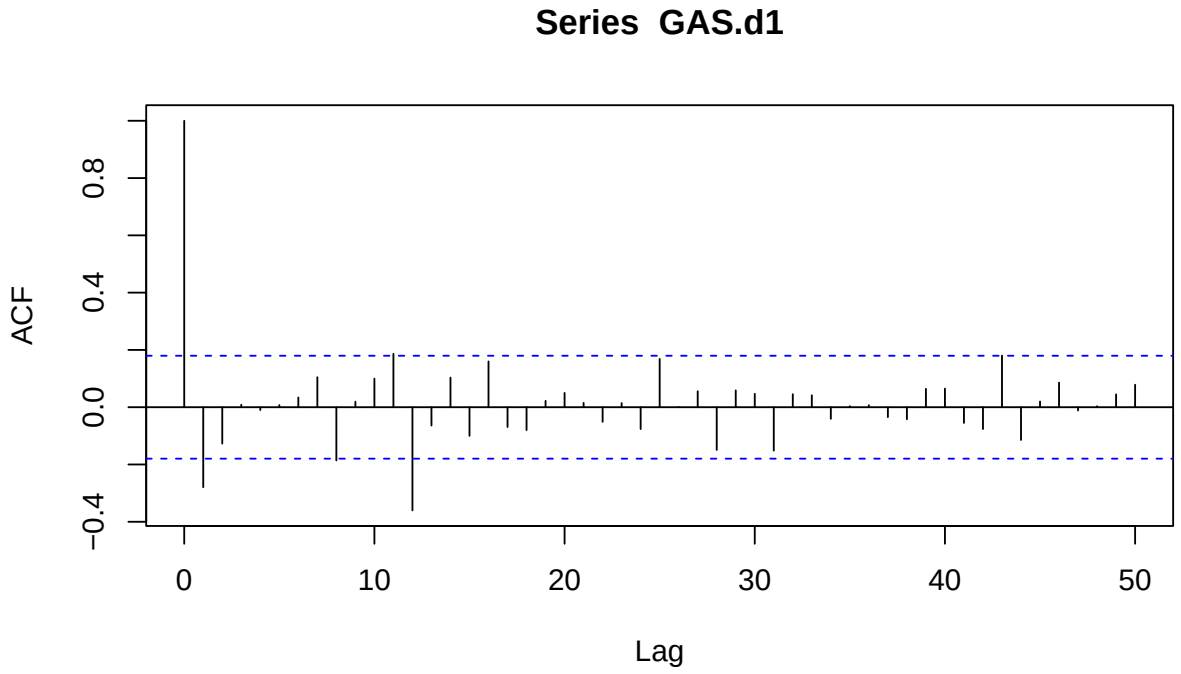
The plot still seems to be non-stationary and ACF also indicates we can difference again at lag = 1.

```r
GAS.d1 <- diff(GAS.d, lag = 1)
ts.plot(GAS.d1, main = "De-trended data")
abline(h= mean(GAS.d1), col = 'red')
```

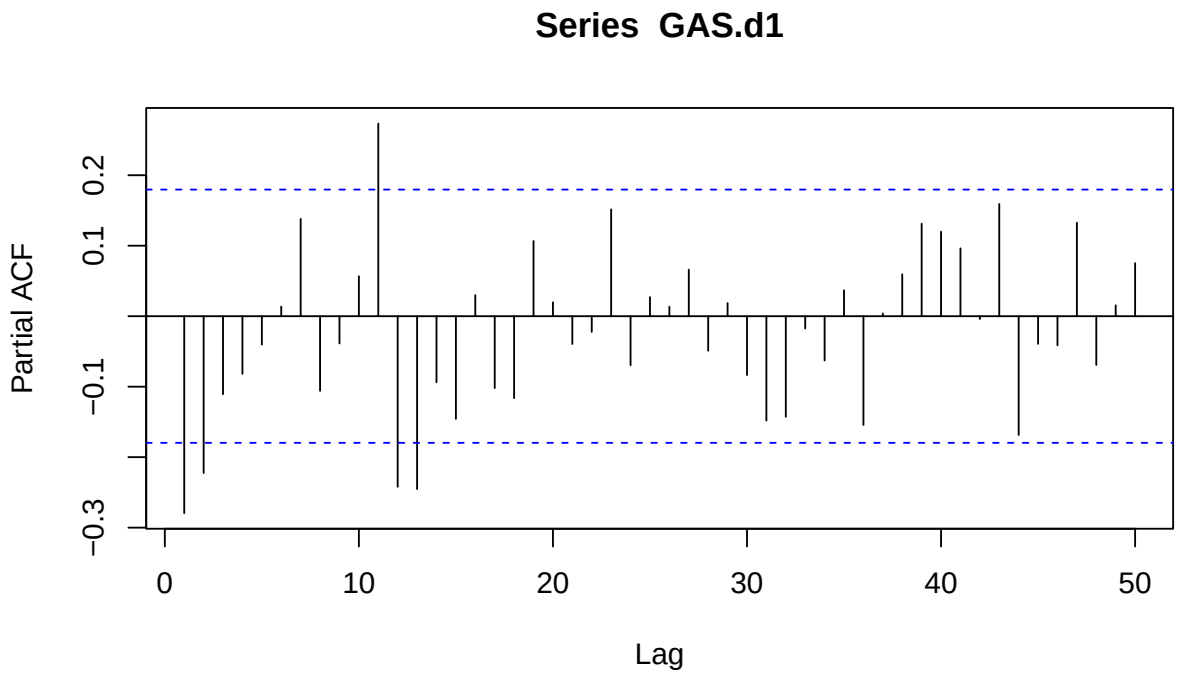


Here I see that there is no seasonality and plot looks stationary. Again I will check the ACFs and PACFs.

```r
acf(GAS.d1, lag.max = 50)
```




```
pacf(GAS.d1, lag.max = 50)
```



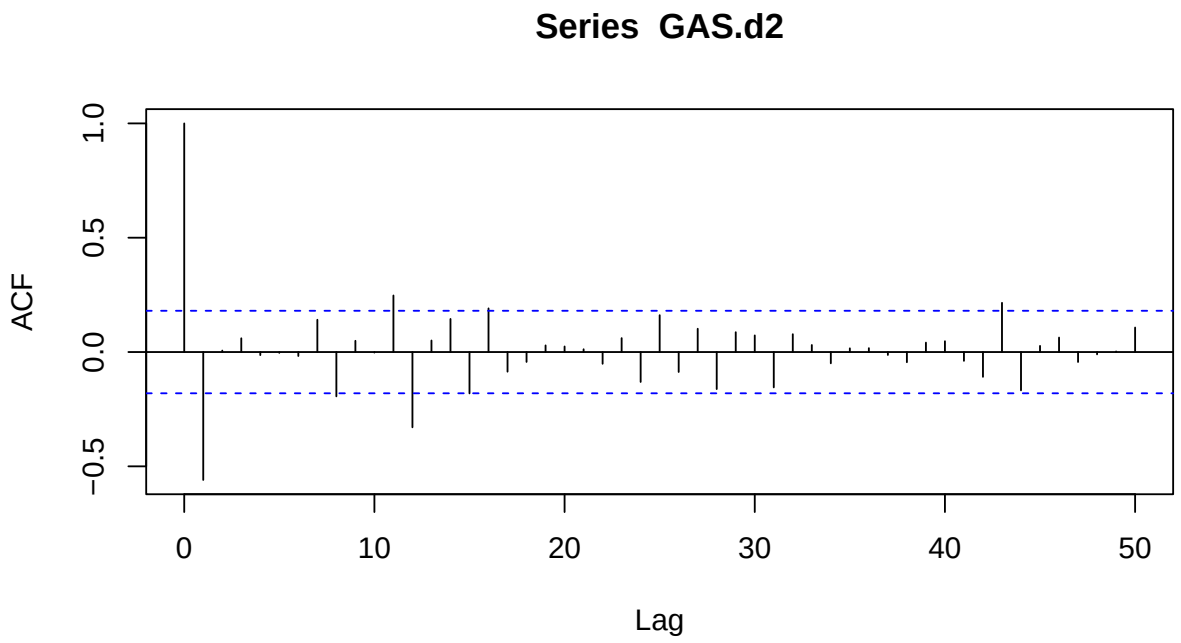

```
paste("Variance:", var(GAS.d1) )
```

```
## [1] "Variance: 12.1769412847815"
```

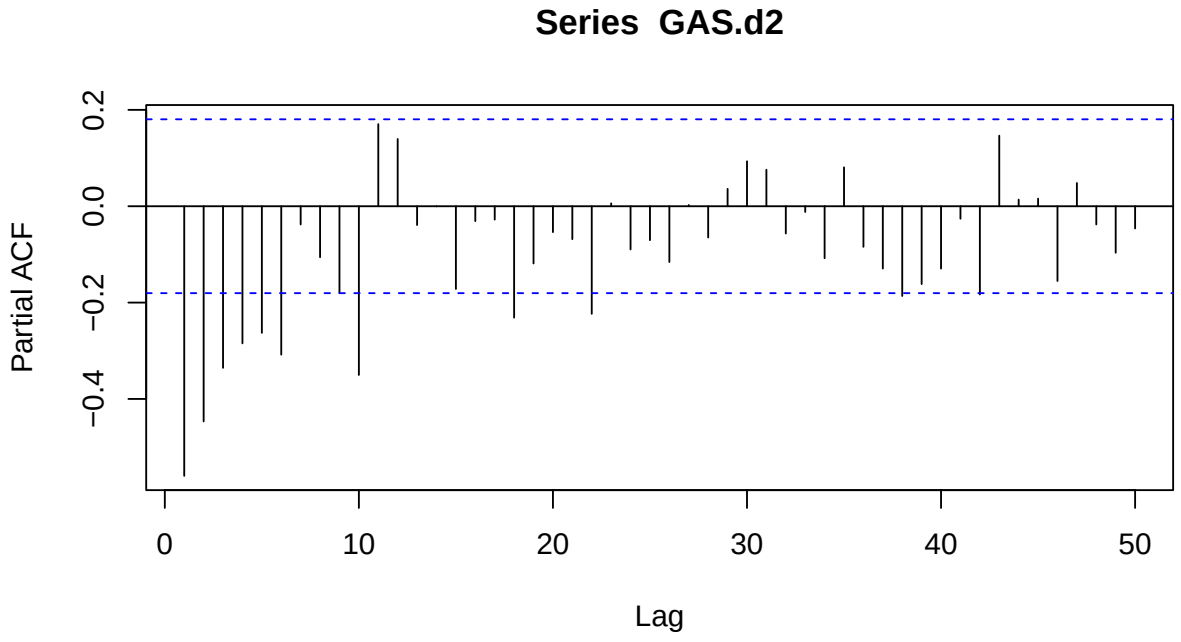
Notice that variance is lower after differncing again at lag = 1, from approximately 31.645 to 12.177.

I will difference once again by lag = 1 to see if variance will drop below 12.177

```
GAS.d2 <- diff(GAS.d1, 1)
acf(GAS.d2, lag.max = 50)
```




```
pacf(GAS.d2, lag.max = 50)
```



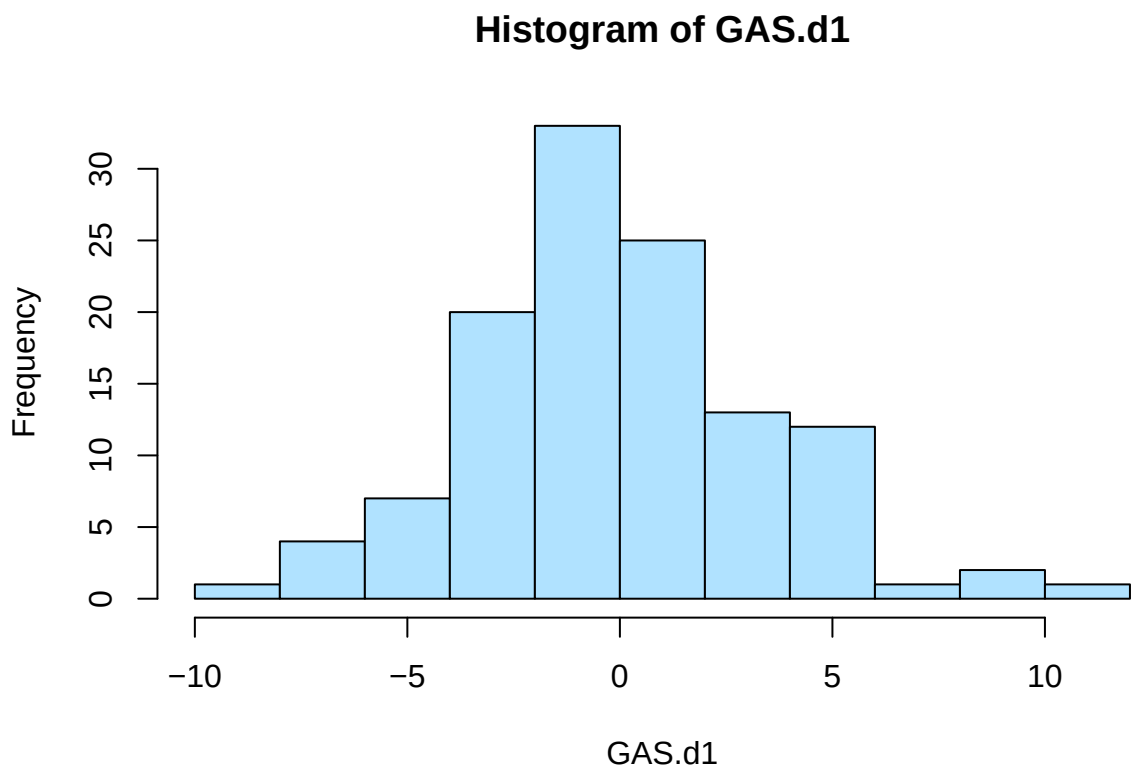
```
paste("Variance:", var(GAS.d2) )
```

```
## [1] "Variance: 31.4254330082356"
```

Variance increased to about 31.425 and the ACFs and PACFs is now harder to analyze the possible model parameters, indicating that we are over differencing. Therefore differencing once after differencing at lag = 12 is our best option for square transformation of the training data set.

The final data after differencing at lag 12 and lag 1 is called GAS.d1. Histogram shown below:

```
hist(GAS.d1, col = "lightskyblue1")
```



The histogram now looks to be more like a normal distribution so therefore I conclude that I will use the square root transformation after all.

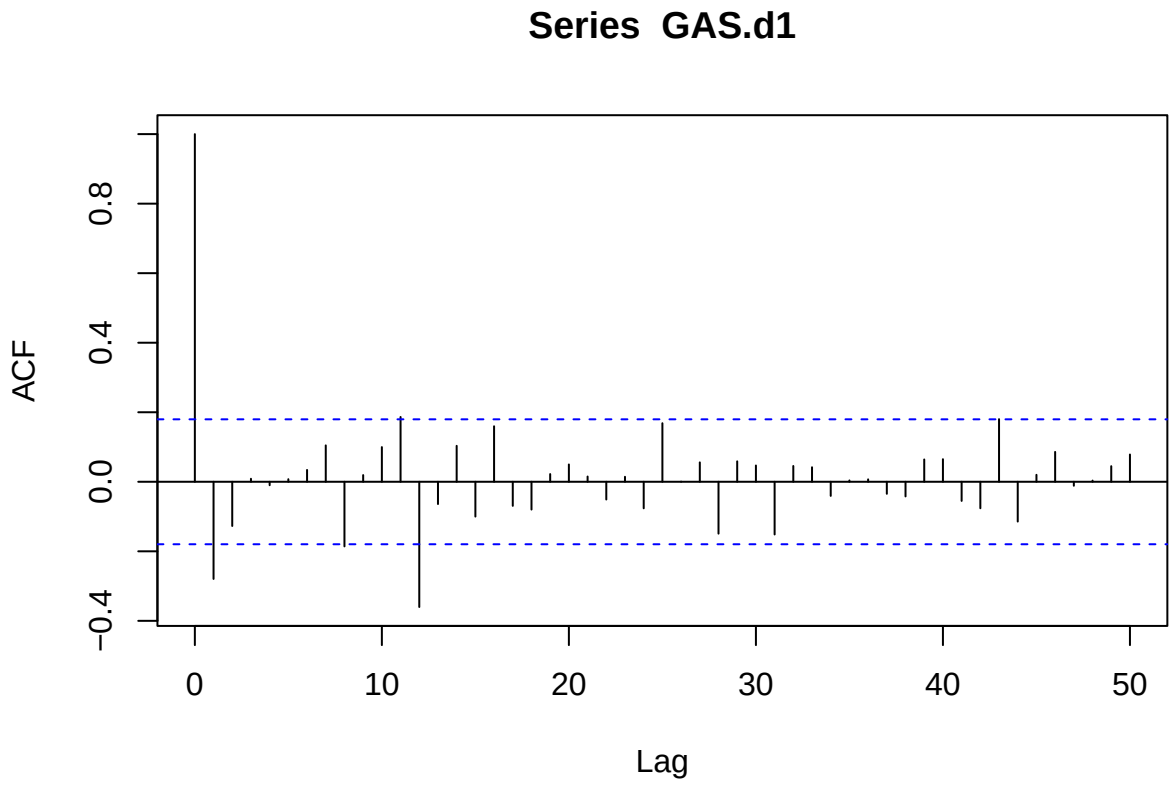
Model Identification / Estimation:

Since my data was seasonal and needed to be difference it by lag 12, I first acknowledge that I am dealing with a SARIMA model.

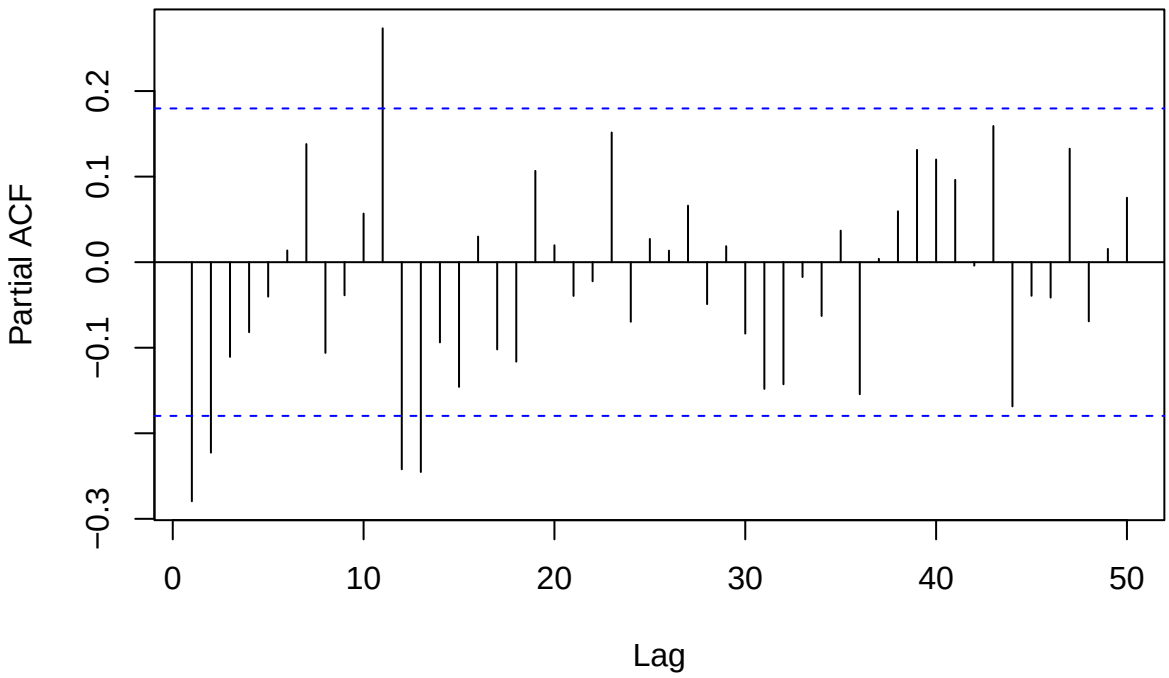
Because I difference once to remove seasonality and once more to minimize variance, d= 1 and D=1.

Now revisiting the ACF and PACF graphs to identify p,q,P,Q :

```r
acf(GAS.d1, lag.max = 50)
```




```r
pacf(GAS.d1, lag.max = 50)
```



By looking at the ACF model lags 1 and 12 are outside the C.I. Therefore, I conclude that q=1 and Q=1 above. I see that in the PACF, lags 1,2,11,12,and 13 are outside the C.I. Since P=1 and p can be 1 or 2, then ± 1 and ± 2 of lag 12 could be outside C.I which we see for lags 11 and 13. From the analysis of both my graphs I concluded that these are the suitable parameters for my SARIMA model:

p = 1 or 2	P = 1
d = 1	D = 1
q = 1	Q = 1

After considering and testing out several models I cut it down to three potential models based on lowest AICc value. The three models are:

1. $SARIMA(0,1,1)(0,1,1)_{s=12}$
2. $SARIMA(1,1,1)(0,1,1)_{s=12}$
3. $SARIMA(2,1,1)(0,1,1)_{s=12}$

To further shorten the model candidates, I apply principle of parsimony and eliminated the model with the most number of parameters out of the three. In this case it is Model 3. Therefore, I will now run the two models and check for its coefficients and compare AICc:

Model 1: $SARIMA(0,1,1)(0,1,1)_{s=12}$:

```
Model_1 <- arima(GAS.tr, order = c(0,1,1), seasonal = list(order = c(0,1,1), period = 12),
                 method = "ML")
Model_1
```

```
##
```

```
## Call:
## arima(x = GAS.tr, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##           ma1     sma1
##       -0.3905  -0.5688
## s.e.   0.1011   0.0926
##
## sigma^2 estimated as 8.014:  log likelihood = -295.11,  aic = 596.22
```

```
AICc(Model_1) #596.3149
```

```
## [1] 596.3149
```

Model 1: $SARIMA(1,1,1)(0,1,1)_{s=12}$:

```
Model_2 <- arima(GAS.tr, order = c(1,1,1), seasonal = list(order = c(0,1,1), period = 12),
                 method = "ML")
Model_2
```

```
##
## Call:
## arima(x = GAS.tr, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##          ar1      ma1     sma1
##       0.2408  -0.5984  -0.5654
## s.e.  0.2112   0.1726   0.0923
##
## sigma^2 estimated as 7.94:  log likelihood = -294.54,  aic = 597.08
```

```
AICc(Model_2)
```

```
## [1] 597.2674
```

For ar1, the confidence interval as shown $(0.2408 \pm 2 \cdot 0.2112) = (-0.1816,\ 0.6632)$ contains zero. Since zero is within the confidence interval, it is possible that ar1 could be zero as well. Therefore it is non significant and I test Model 2 for when ar1 is zero and see if the AICc is lower.

```
Model_2 <- arima(GAS.tr, order = c(1,1,1), seasonal = list(order = c(0,1,1), period = 12),
                 fixed = c(0,NA,NA), method = "ML")
```

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

```
Model_2
```

```
##
## Call:
## arima(x = GAS.tr, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),
##     fixed = c(0, NA, NA), method = "ML")
##
## Coefficients:
##       ar1      ma1     sma1
##         0  -0.3905  -0.5688
## s.e.    0   0.1011   0.0926
##
## sigma^2 estimated as 8.014:  log likelihood = -295.11,  aic = 596.22
```

```
AICc(Model_2) #596.4093
```

```
## [1] 596.4093
```

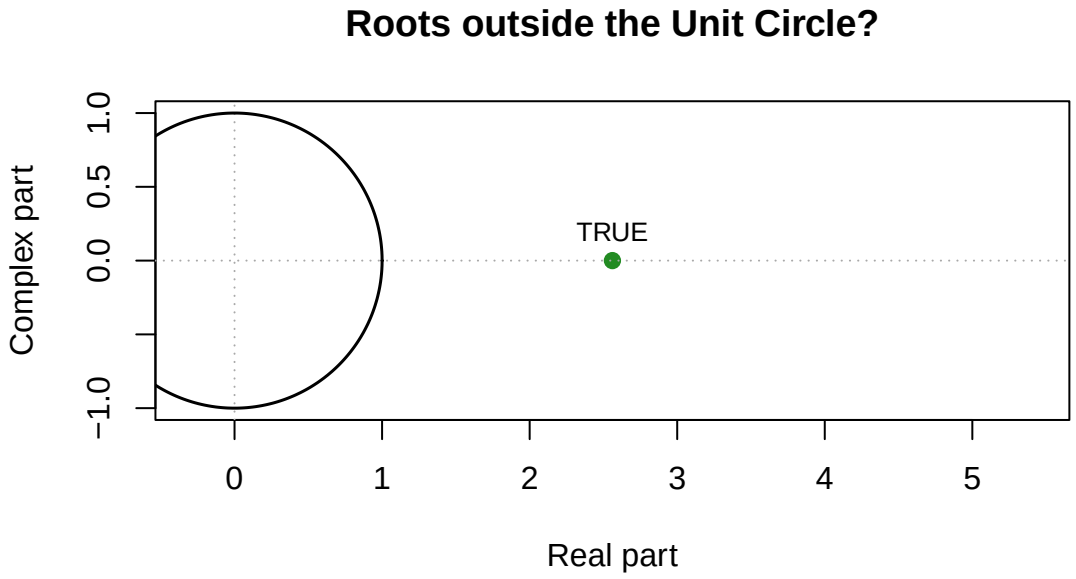
The AICc did get lower when ar1 = 0 but compared to Model 1 it is still just slightly greater (596.4093 > 596.2674). Additionally, Model 2 has more parameters than Model 1. Therefore I will use Model 1:

Model 1 equation: $\nabla_1\nabla_{12}\text{GAS.tr} = (1 - 0.3905_{(0.1001)}B)(1 - 0.5688_{(0.0926)}B^{12})Z_t$ $\hat{\sigma}^2 = 8.014$

Before proceeding to diagnostic checking I will check if the model is stationary and invertible. The Model_1 is stationary because it is pure moving average. To check for invertibility I need to check if all roots lie outside the unit circle.

```
#SARIMA(0,1,1)(0,1,1) s=12
uc.check(pol_ = c(1, -0.3905), plot_output = TRUE)
```

```
##       real complex outside
## 1 2.560819       0    TRUE
## *Results are rounded to 6 digits.
```

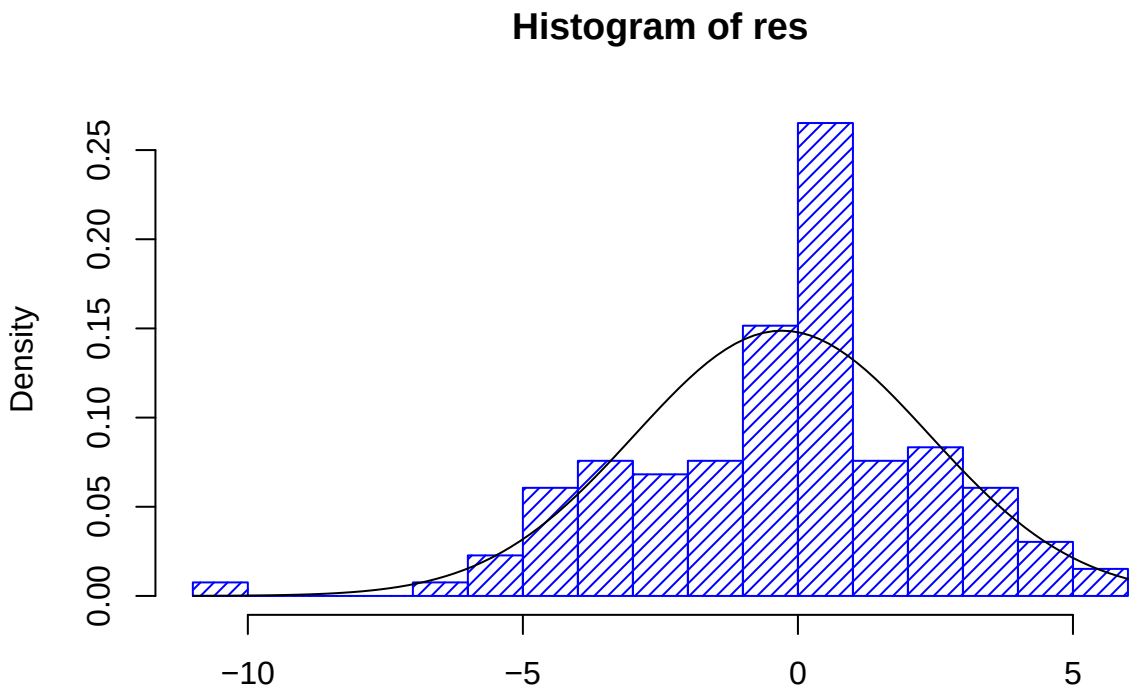


The model is invertible since it is a MA process and all roots are outside the unit circle. Therefore, we can proceed to diagnostic checking.

Diagnostic Checking:

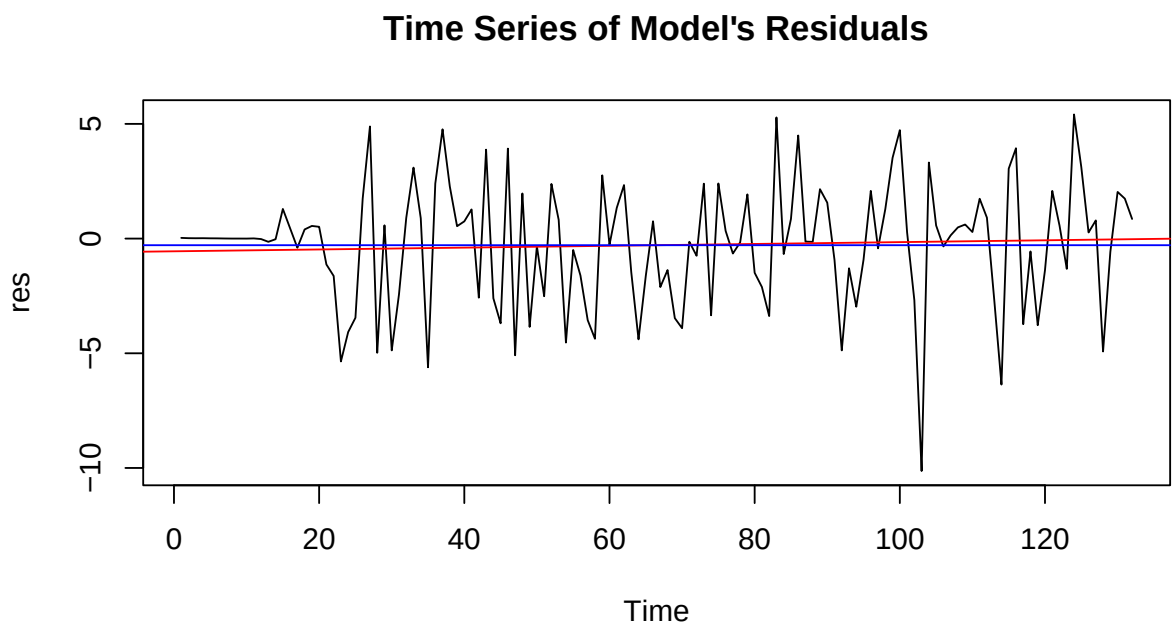
We do a diagnostic check to see whether the residuals have properties resembling White Noise:

```
#Diagnostic checking of SARIMA:
res <- residuals(Model_1)
hist(res, density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
```



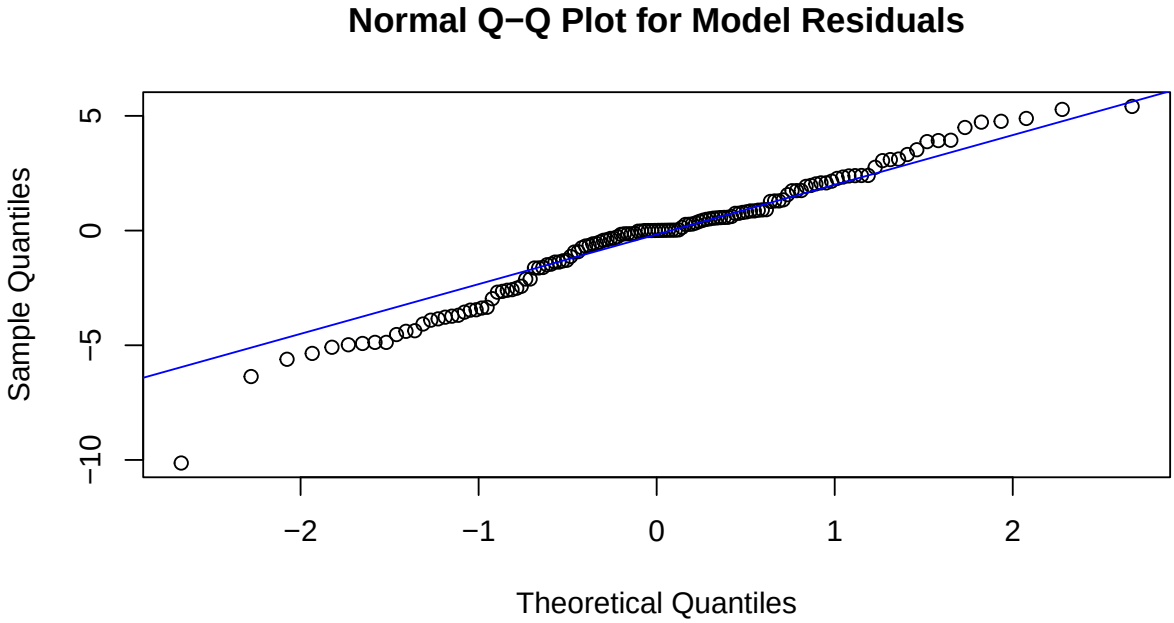
The histogram of the residuals does not look quite like a normal distribution. What we see is a heavy tail because there is some probability in the distribution taking on a significantly larger value relative to the mean 0, in this case -10.

```
plot.ts(res, main = "Time Series of Model's Residuals")
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
```



After plotting the residuals I do not see any trends, visible change of variance, or seasonality and sample mean looks to be almost zero.

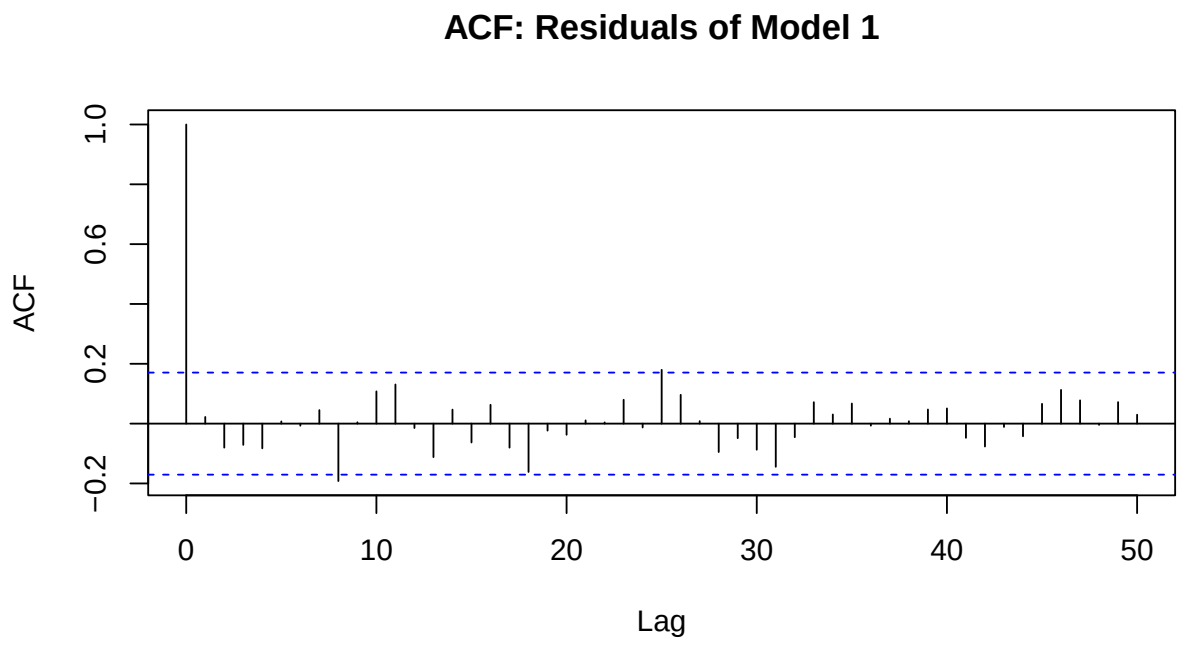
```
qqnorm(res, main= "Normal Q-Q Plot for Model Residuals")
qqline(res,col="blue")
```



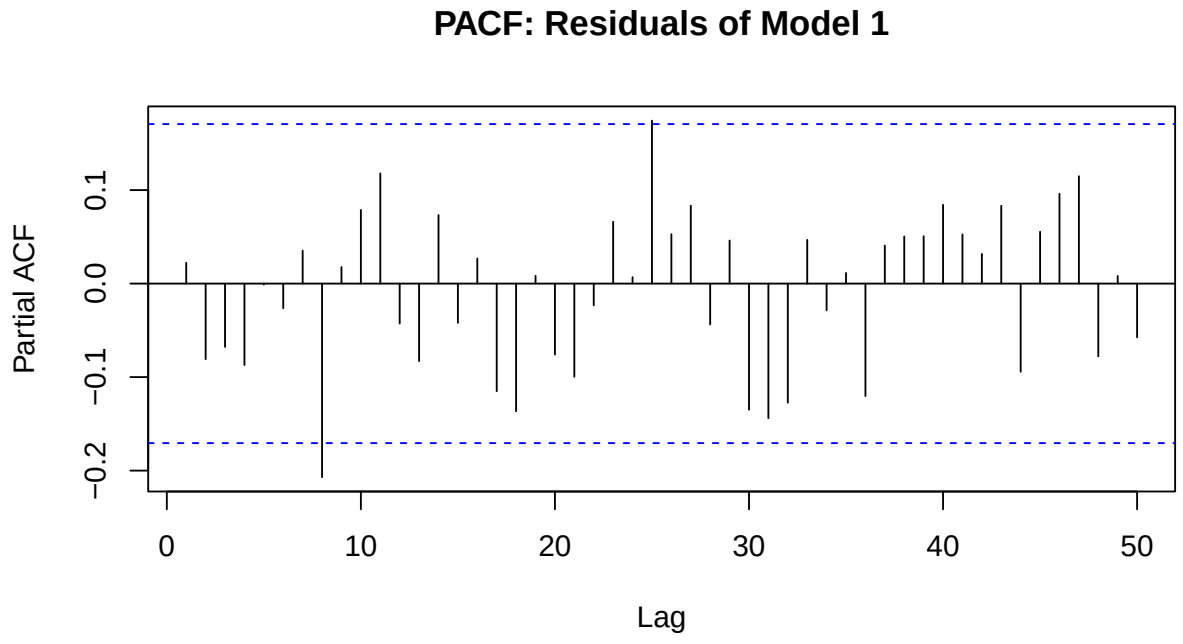
The Q-Q plot seems to have some deviations between ± 2, especially outside of ± 1.

Next I plot the ACF and PACF of the residuals:

```r
acf(res, lag.max=50, main = "ACF: Residuals of Model 1")
```




```r
pacf(res, lag.max=50, main = "PACF: Residuals of Model 1")
```



In the ACF graph lag=8 seem to be just outside the C.I and same goes for lag=8 in the PACF. Additionally lag=25 in PACF is just barely outside the C.I as well. Since 50 lags were tested, I would say it looks good enough for sample ACFs and will assume them to be white noise.

Checking Shapiro-Wilk's test, Box-Pierce test, Ljung-Box test, and McLeod test:

```
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.97482, p-value = 0.01483
```

```
Box.test(res, lag = 11, type = c("Ljung-Box"), fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 12.335, df = 9, p-value = 0.1951
```

```
Box.test(res, lag = 11, type = c("Box-Pierce"), fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 11.432, df = 9, p-value = 0.2472
```

```
Box.test((res)^2, lag = 11, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  (res)^2
## X-squared = 15.913, df = 11, p-value = 0.1444
```

Results show that all test have passed except for Shapiro-Wilk normality test as shown that its p-value is less than 0.05. This tells me that my data is Non-Gaussian. Looking back at my Q-Q plot I mentioned earlier that it deviates from the line which is why the test did not pass. I learned not all data can be transformed to make residuals Gaussian and for this particular case I would need to utilize Non-Gaussian methods instead.

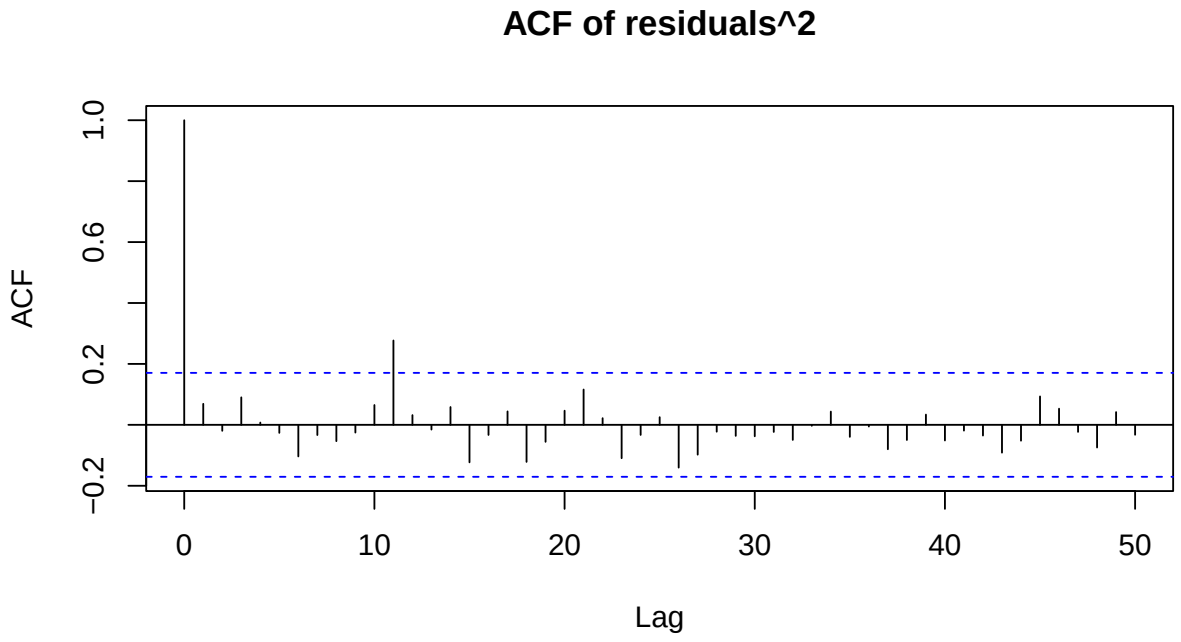
However, what I can take from the other tests is that the residuals from my my SARIMA model have no auto-correlation.
Note: Model 2 has same coefficients as Model 1 and after trying diagnostic checking for Model 2, tests outcomes did not change.

Lastly, I plug my residuals for Model 1 to Yule-Walker method and check ACF for residual squared:

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

```
##
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  7.196
```

```r
acf(res^2, lag.max=50, main = "ACF of residuals^2")
```




```r
#ALL models are order selected 0
```

The ACF of residuals squared shows lag=11 is outside C.I so it does not seem to be white noise. The fitted residuals of Model 1 into Yule-Walker ran in R automatically chose orders "Order selected 0". Therefore, with our diagnostic results we are ready to move to forecasting.

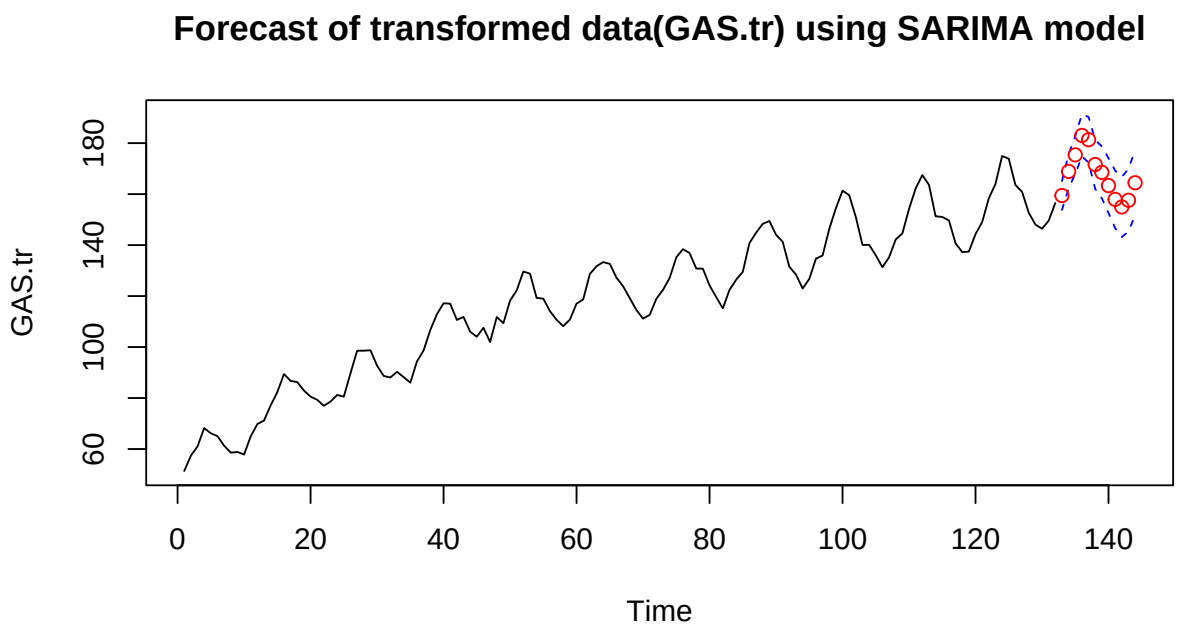
Forecasting:

I printed out my predicted future observations using the analysis from Model 1 and compared it to how similar it is to the raw data.

1-Year Forecast of Transformed Data:

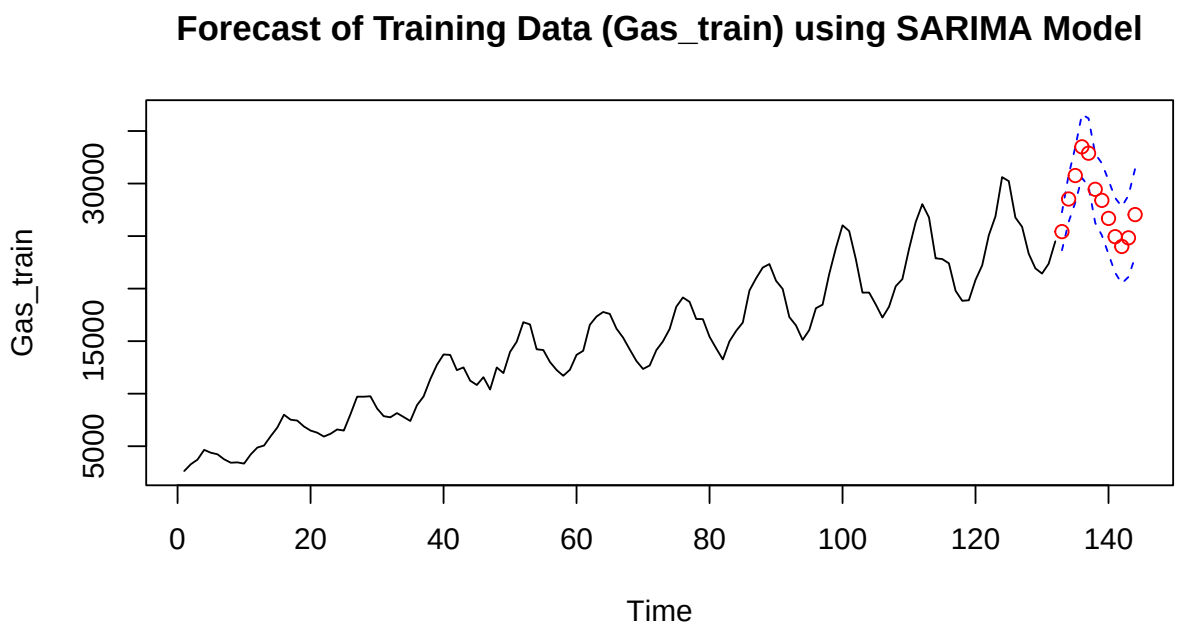
```r
fit.A <- arima(GAS.tr, order = c(0,1,1), seasonal = list(order = c(0,1,1), period = 12),
               method = "ML")

#Graph with 12 observations forecast on transformed data:
pred.tr <- predict(fit.A, n.ahead = 12)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(GAS.tr, xlim=c(1,length(GAS.tr)+12), ylim = c(min(GAS.tr),max(U.tr)),
        main = "Forecast of transformed data(GAS.tr) using SARIMA model")
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(GAS.tr)+1):(length(GAS.tr)+12), pred.tr$pred, col="red")
```



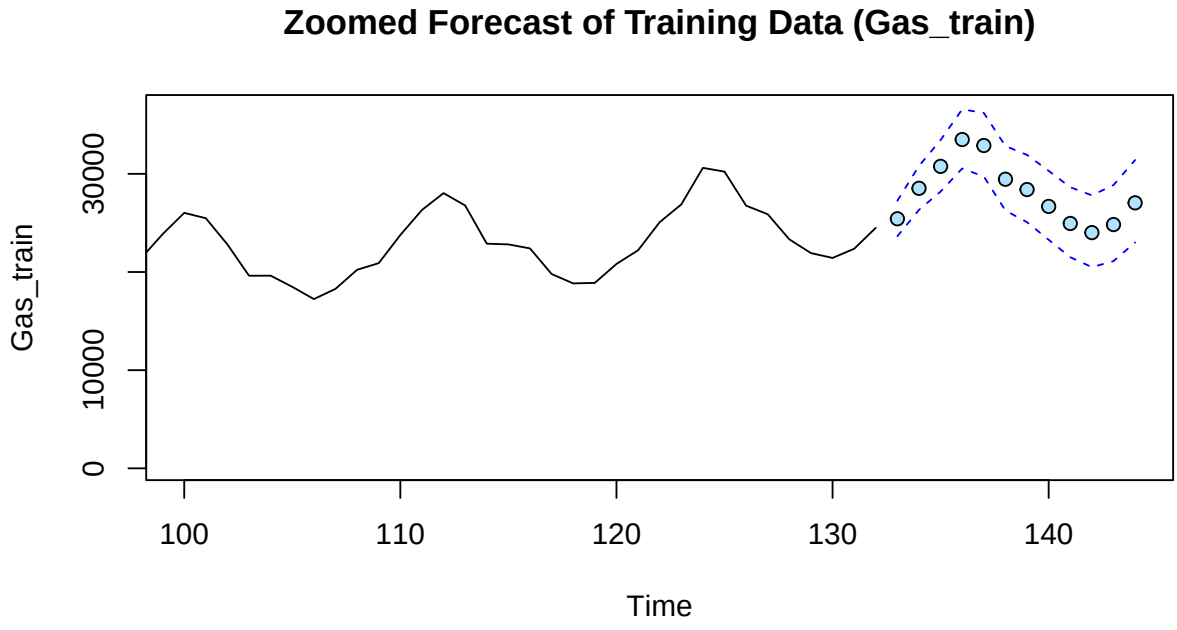
1-Year Forecast of Training Data:

```
#Produce graph with forecast on training data:
pred.orig <- (pred.tr$pred)^2
U= (U.tr)^2
L= (L.tr)^2
ts.plot(Gas_train, xlim=c(1,length(Gas_train)+12), ylim = c(min(Gas),max(U)),
        main = "Forecast of Training Data (Gas_train) using SARIMA Model")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(Gas_train)+1):(length(Gas_train)+12), pred.orig, col="red")
```



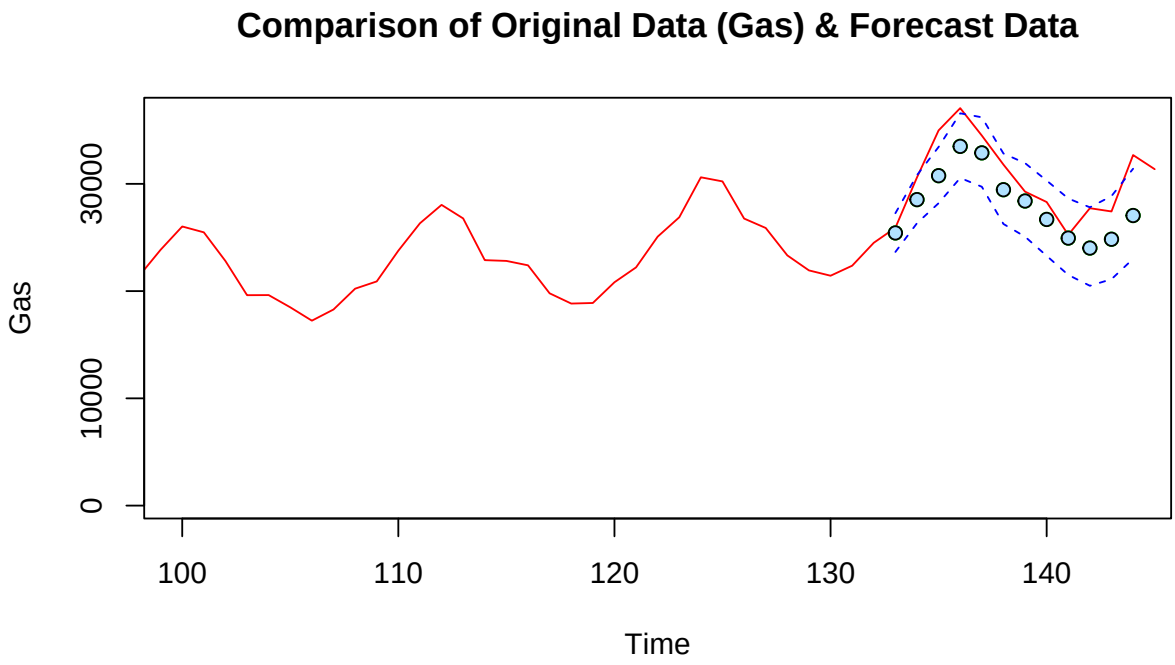
Zoom of previous graph starting from entry 100:

```
ts.plot(Gas_train, xlim = c(100,length(Gas_train)+12), ylim = c(250,max(U)),
        main = "Zoomed Forecast of Training Data (Gas_train)")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(Gas_train)+1):(length(Gas_train)+12), pred.orig, col="black",
       cex = 1, pch = 21, bg = "lightskyblue1")
```



Zoomed graph of 1-Year Forecasts & Original(Raw) Data:

```
#To plot zoomed forecasts and true values:
ts.plot(Gas, xlim = c(100,length(Gas_train)+12), ylim = c(250,max(U)), col="red",
        main = "Comparison of Original Data (Gas) & Forecast Data")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(Gas_train)+1):(length(Gas_train)+12), pred.orig, col="green")
points((length(Gas_train)+1):(length(Gas_train)+12), pred.orig, col="black",
       pch = 21, bg = "lightskyblue1")
```

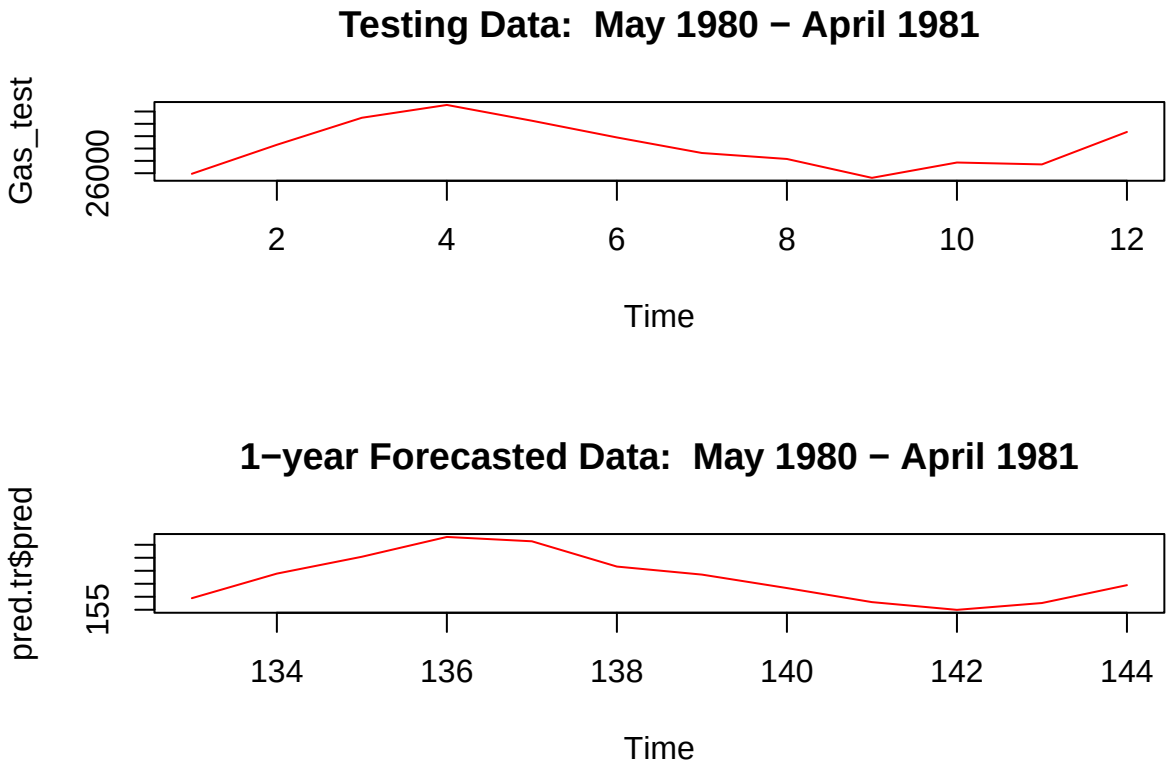


Quick analysis shows that my raw data and 1-year forecast data resemble a similar trend. However, I noticed that the raw data is not completely within the prediction interval. This is because going back to diagnostic checking, my model contains some deviation in the Q-Q plot of residuals and also display heavy-tail distribution when plotting the histogram of its residuals. Resolving this would require heavy-tail testing methods. Overall, forecasting data of 1-year ahead does not look bad at all.

Conclusion:

The goal for this project was to compare and contrast my 1-year forecast data to its original data and examine how accurate or as to how far my results differ. What resulted was my forecast data having its prediction interval slightly off the raw data I was trying to predict. The model I chose to forecast the data was $SARIMA(0,1,1)(0,1,1)_{s=12}$ written by its algebraic form as $\nabla_1\nabla_{12}\text{GAS.tr} = (1-0.3905_{(0.1001)}B)(1-0.5688_{(0.0926)}B^{12})Z_t$, $\hat{\sigma}^2 = 8.014$. Overall this model when used for forecasting data did not deviate to far from the raw data as shown below. Given that statistical methods cannot control such events like a huge recession or a global pandemic etc., data is almost never perfectly accurate to predict. I would say my prediction analysis turned out quite well.

```r
par(mfrow=c(2,1))
ts.plot(Gas_test, main = "Testing Data:  May 1980 - April 1981",
        col = "red")
ts.plot(pred.tr$pred, main = "1-year Forecasted Data:  May 1980 - April 1981",
        col = "red")
```



References:

Hyndman, R.J. “Time Series Data Library”, https://datamarket.com/data/list/?q=provider:tsdl.