Duke Energy Time Series Analysis

ST 534

Karthik Edupuganti, Chris Hill and Elizabeth Thompson

Introduction to the Data

For this project, we used Elizabeth's real Duke Energy data. This data was downloaded as an XML file from the Duke Energy website and then converted to a csv file. The data was originally hourly, but since we were most interested in energy usage by day, the data was cleaned to reflect daily energy usage.

The data (seen in *Figure 1*) covers from September 2022 to November 2023. Energy usage is measured in Kilowatt Hours (kWh) per day. Full SAS output for this data and an initial AR(2) model can be found in *Appendix A*.

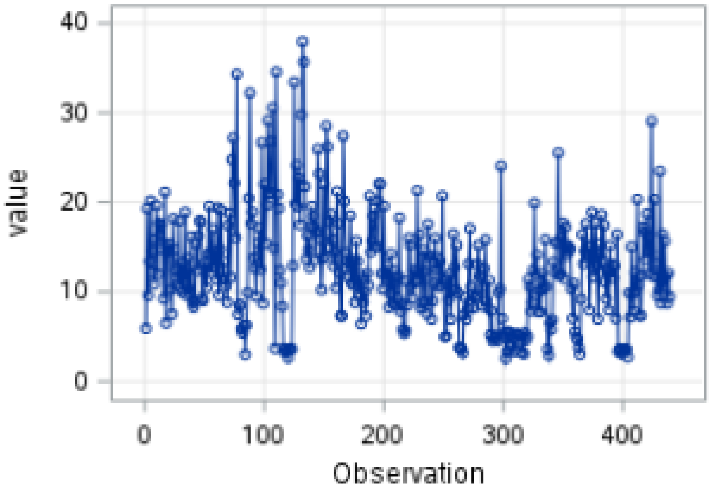


Figure 1: Time series plot of the Duke Energy data.

The Model

The chosen model is an ARMA(1, 1) that was fit on a square root transformation of the values based upon the recommendation of a Box-Cox analysis. All parameters were found to be significant using t-tests on the Conditional Least Squares Estimation. A Chi-Square test used for the Autocorrelation Check of Residuals also indicates that the residuals are uncorrelated (white noise) for this model. This model achieved the lowest scores of any model tested using both the Akaike Information Criterion (AIC) and the Schwarz Bayesian Information Criterion (SBC). These values were 917 and 929 respectively and can be seen in comparison with other models in *Model Comparisons*. The full SAS output for this model can be found in *Appendix B*.

Model formula for Transformed Data: $Z_t = 3.43029 + 0.82453Z_{t-1} + 0.38616a_{t-1} + a_t$

Forecast

We used our selected model to forecast beyond the end of the series. Because we had approximately a year of data, we chose to forecast 60 days into the future. The forecasted values along with the 80% and 95% prediction intervals are shown in *Figure 2*. These predicted values were generated using the square root of the data. Because of this, it was required that we revert the predictions back to the original scale by squaring them to coincide with the original data as shown.

We can see the rapid expansion of prediction intervals as we move further into the future, reflecting the increasing uncertainty associated with making predictions beyond the scope of historical data. We also see that the prediction for energy usage increases as the days progress before leveling off at approximately 12kWh. This is very near to our mean of 12.6kWh for the original dataset.

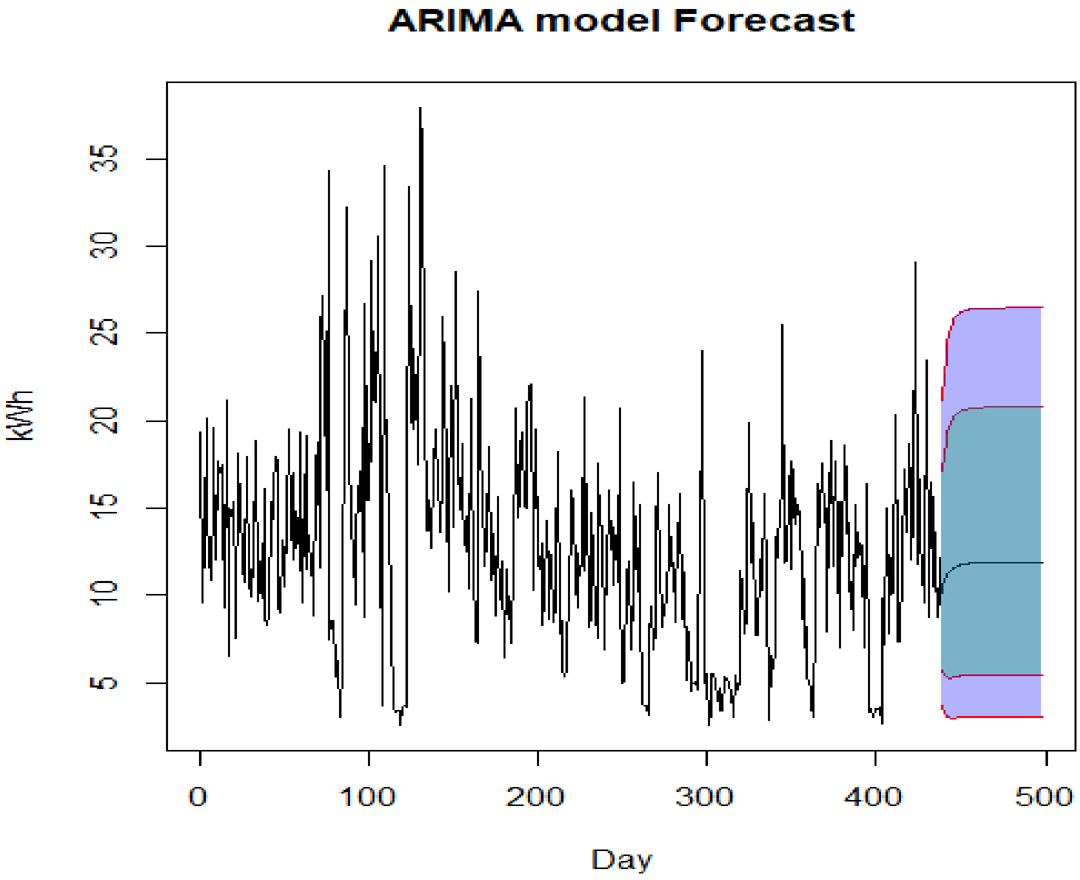
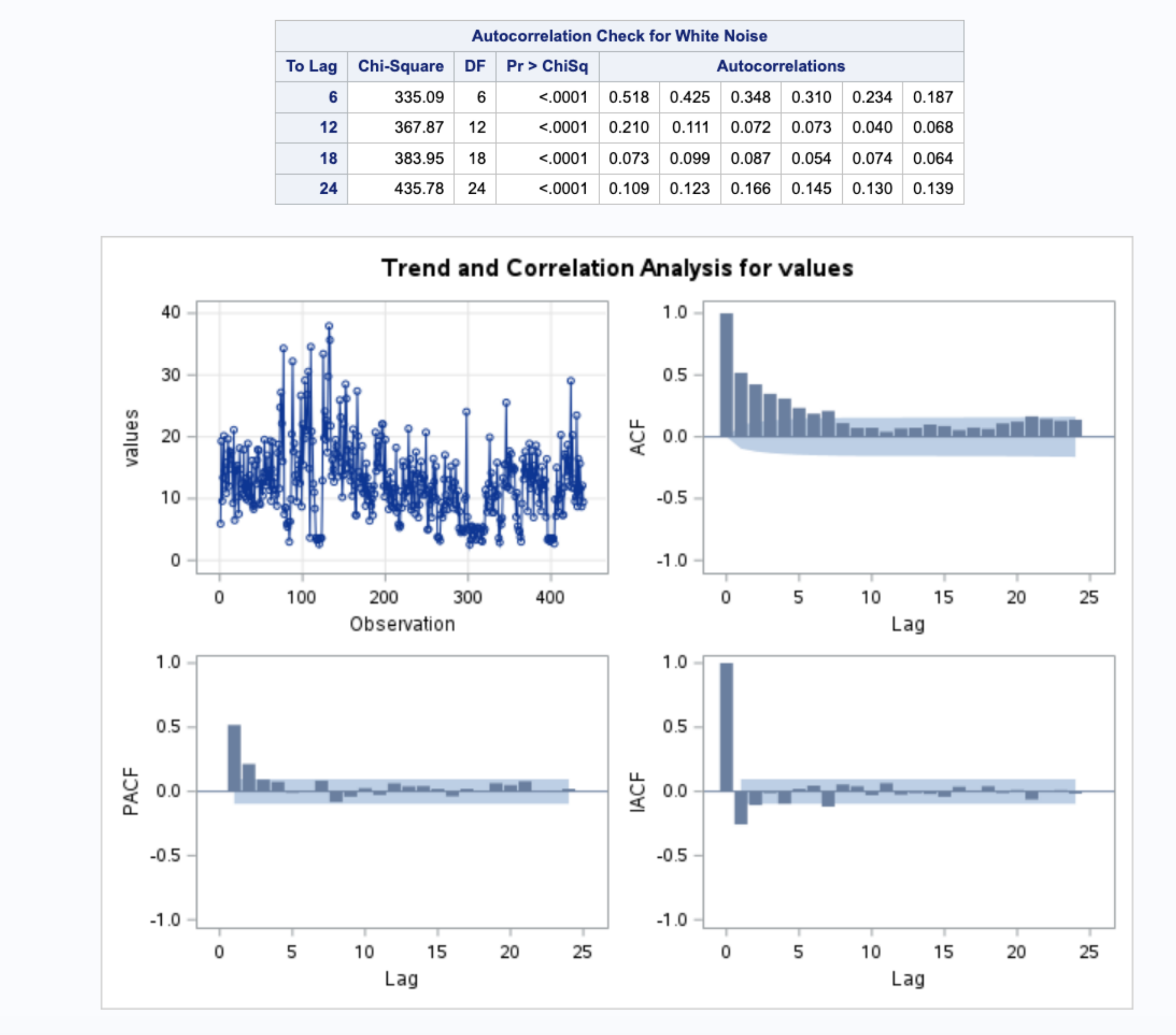


Figure 2: Time series forecast of energy usage 60 days out.

Model Selection and Procedure

After viewing an approximately constant mean and variance from the original daily dataset, several models were produced using the Autocorrelation (ACF) and Partial Autocorrelation Function (PACF) plots as seen in *Figure 3.* Looking at the initial output below from the PROC ARIMA, white noise checks show low p-values, there is a gradual decrease in ACF plot, and in the PACF plot there was a clear cutoff at lag 2. This suggests that an AR(2) model may be an appropriate starting point for this data.

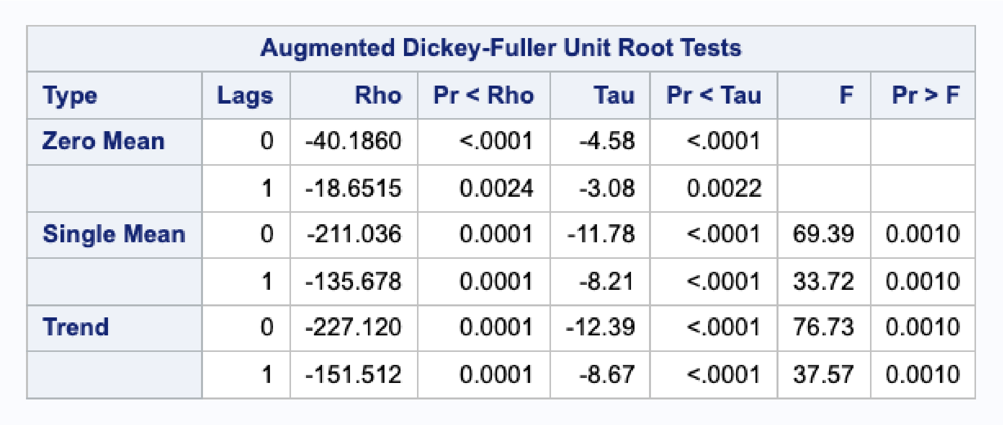


Autocorrelation Check for White Noise

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	335.09	6	<.0001	0.518	0.425	0.348	0.310	0.234	0.187
12	367.87	12	<.0001	0.210	0.111	0.072	0.073	0.040	0.068
18	383.95	18	<.0001	0.073	0.099	0.087	0.054	0.074	0.064
24	435.78	24	<.0001	0.109	0.123	0.166	0.145	0.130	0.139

Figure 3: Initial analysis for data, including plot, ACF, PACF and IACF.

Due to the gradual decay in ACF, we decided to perform an Augmented Dickey-Fuller test at lag 1 (see *Figure 4*). We rejected the null for both a linear trend and constant mean, so we did not take a difference.



Augmented Dickey-Fuller Unit Root Tests

Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Zero Mean	0	-40.1860	<.0001	-4.58	<.0001		
	1	-18.6515	0.0024	-3.08	0.0022		
Single Mean	0	-211.036	0.0001	-11.78	<.0001	69.39	0.0010
	1	-135.678	0.0001	-8.21	<.0001	33.72	0.0010
Trend	0	-227.120	0.0001	-12.39	<.0001	76.73	0.0010
	1	-151.512	0.0001	-8.67	<.0001	37.57	0.0010

Figure 4: Augmented Dickey-Fuller test at lag 1 for AR(2).

Looking further into the AR(2), the autocorrelation check shows low p-values for a couple of lags (using a significance level of 0.05), thus the test statistics reject the no-autocorrelation hypothesis. This means the residuals are not white noise and this model is not fully adequate to model this series. The Autocorrelation Check of Residuals for this can be seen in *Appendix A*.

Based on the information from the initial PROC ARIMA statement and AR(2) model, we decided to take a look at some more models. In particular, we looked at the ARMA(1,1), ARMA(1,2), ARMA(2,1), ARMA(2,2), and ARMA(2,7) models.

Employing further scrutiny towards the stationarity of the data, we next performed a Box-Cox analysis for more insight.

Box-Cox Transformation

The Box-Cox transformation can be used to identify an optimal transformation that stabilizes variance and improves the normality of the data. The analysis produced a lambda value of 0.5, suggesting the use of a square root transformation on the data (see *Figure 5*).

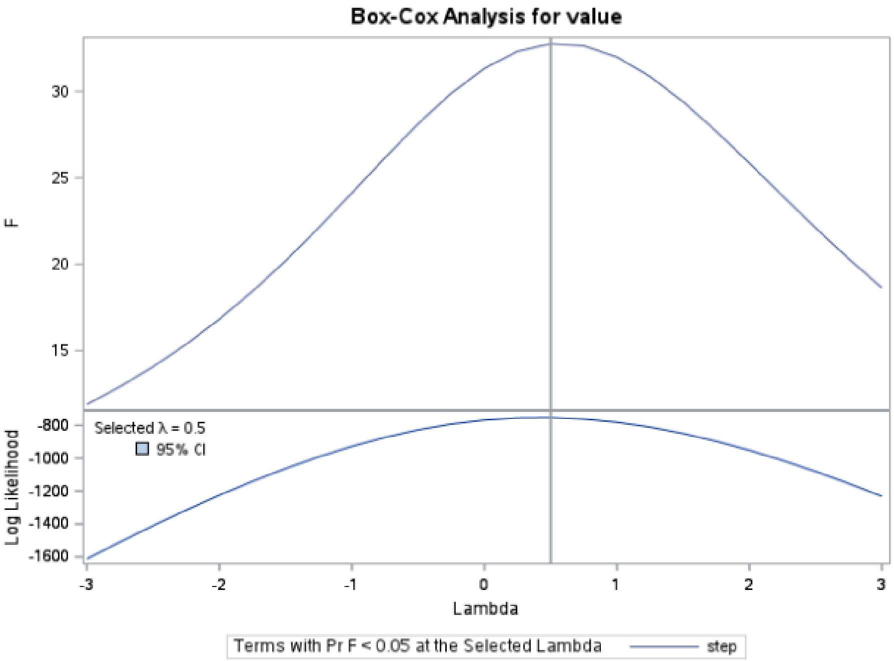


Figure 5: Box-Cox Analysis on un-transformed data.

This followed with the transformed dataset below in *Figure 6* with autocorrelation plots found in *Appendix C*. Though the visual appearance looks similar, this transformed data produced a significantly better model from what was fit using the unaltered data prior to this square root transformation.

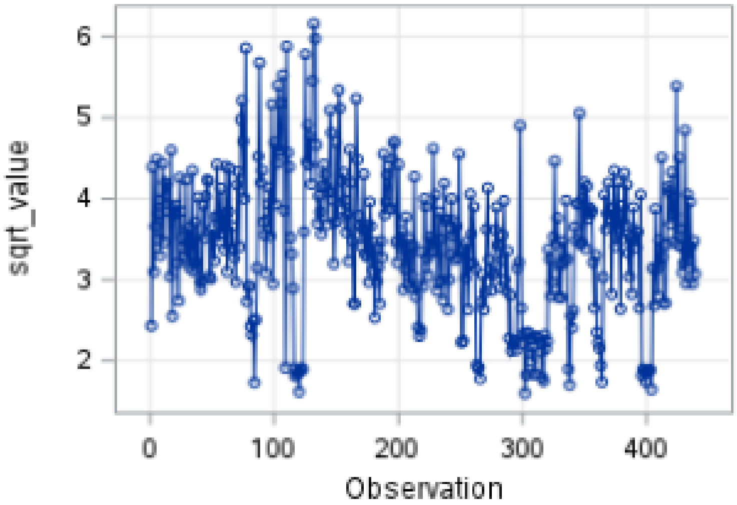


Figure 6: Time Series plot of the data transformed by taking a square root.

We also performed a Dickey-Fuller test for the transformed data, and once again taking a difference was rejected.

Transfer Function Model

Since we were working with energy data, we were interested in investigating if there was a relationship with temperature data. We gathered temperature data for Chapel Hill, N.C. from Weather Underground from September 2022 to November 2023.

When we plotted energy usage and temperature in *Figure 7*, we saw that energy usage appeared to increase around the same time temperature decreased. We chose to further investigate this relationship with a Transfer Function Model.

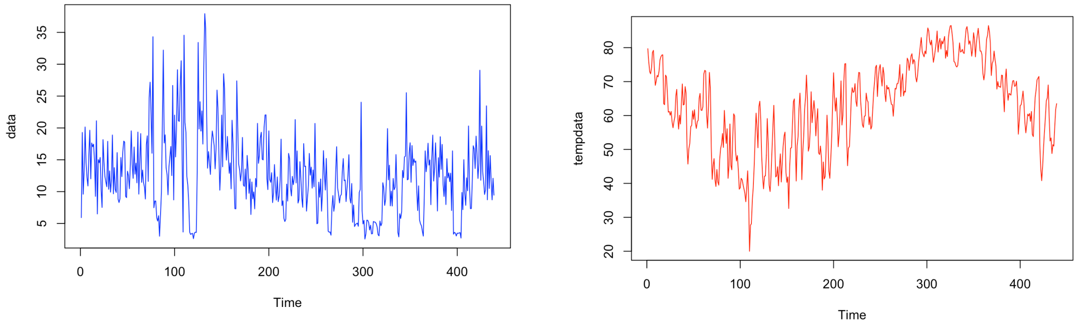


Figure 7: Time series plot of energy usage (blue) and average daily temperature (red). We can see that there may be some kind of relationship between energy usage and temperature.

We fit a Transfer Function model using the square root transformed energy data, and we also transformed the temperature data using square root. This model was fit in R using an ARMA(2,2) as the best model. The code and outputs from this analysis can be found in *Appendix D*. AIC was 296.65. However, we decided to not use this model as our final model since the cross-correlation function was not 0 at negative lags (see *Figure 8*). This indicates that there is feedback in our function. Since this was not included in our course, we decided to not use this model. In future analysis, it would be interesting to investigate this feedback model.

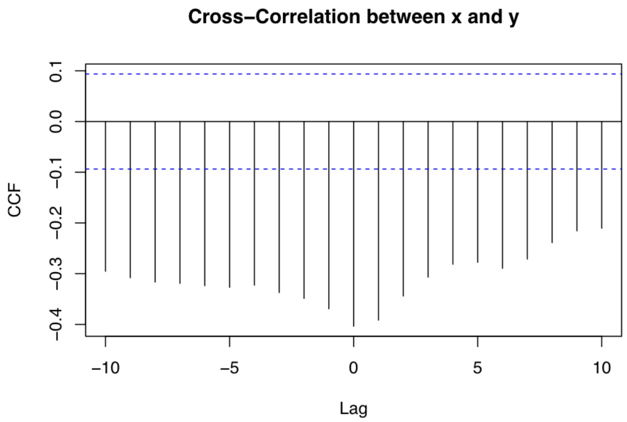


Figure 8: Cross Correlation Function between energy and temperature. We can see spikes at negative values, indicative of feedback in the model.

Model Comparisons

The models that follow are sorted on AIC and SBC, but also on hypothesis tests for the presence of autocorrelation and whether all the included parameters are significant in the model. Each test was evaluated using a significance of 0.05. If a model failed one of these hypothesis tests, it is highlighted in red below.

No Autocorrelation: The Chi-Square test statistics for the residuals series indicate whether the residuals are uncorrelated (white noise).
All Parameters Significant: In the context of a given model, all parameters used show evidence that they have a non-zero impact on the model and are considered significant.

Transformed Data Models

P	Q	AIC	SBC	No Autocorrelation	All Parameters Significant
1	**1**	**917.086**	**929.339**	**TRUE**	**TRUE**
2	0	919.671	931.924	TRUE	TRUE
2	1	918.984	935.322	TRUE	FALSE

Original Data Models

P	Q	AIC	SBC	No Autocorrelation	All Parameters Significant
1	1	2680.442	2692.696	TRUE	TRUE
1	2	2682.424	2698.762	TRUE	FALSE
2	0	2685.986	2698.24	FALSE	TRUE
2	1	2682.424	2698.762	TRUE	FALSE
2	2	2684.424	2704.847	TRUE	FALSE
2	7	2686.358	2727.203	TRUE	FALSE

Taking a look at the different models, we already discussed that the AR(2) model is not suitable because the null hypothesis of uncorrelated data is rejected. Looking at the other discussed models which are ARMA(1,1), ARMA(1,2), ARMA(2,1), ARMA(2,2), and ARMA(2,7), they fail to reject the null hypothesis for white noise, thus they can be suitable models. Looking at the parameter significance levels for each model, for ARMA(1,2), ARMA(2,1), ARMA(2,2), and ARMA(2,7), at least one parameter in the model is not significant, thus we do not choose these models. The best model is the ARMA(1,1) model as it has no autocorrelation and the parameters are significant, along with having the lowest AIC and SBC.

Due to our doubts on stationarity of the data, we used the Box-Cox method to get a suitable transformation for the data. Testing the ARMA(1,1), ARMA(2,1), and AR(2), the models have no autocorrelation for all of them, but ARMA(2,1) does not have all significant parameters. The ARMA(1,1) and AR(2) both have significant parameters in the model, but out of the two, ARMA(1,1) has the lowest AIC and SBC. Comparing the ARMA(1,1) models from original data vs transformed data, it can be seen that the transformed ARMA(1,1) has a much lower AIC and SBC thus indicating it as the best model.

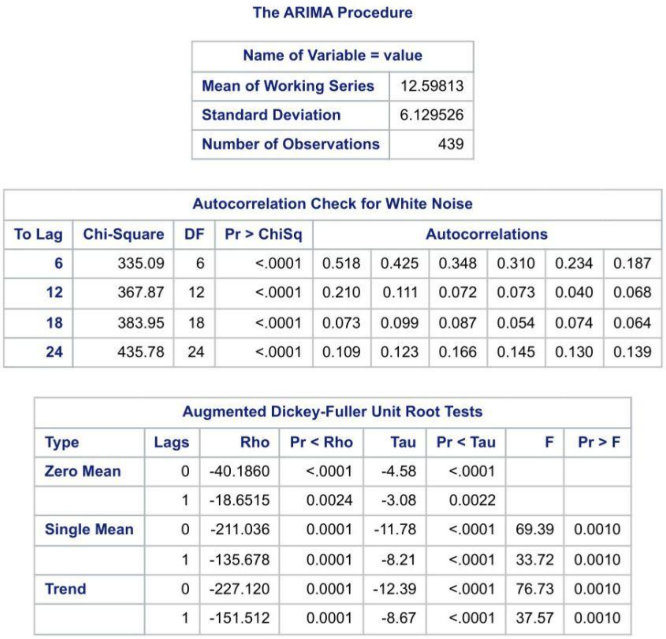
Conclusion

Based on our analysis, we decided to use the ARMA(1, 1) model with a square root transformation. We forecasted that energy usage will increase before leveling off just before the mean. This makes sense since temperatures are already decreasing considerably, leading to an increase in energy usage for heating.

Future modeling on this data could consider seasonal patterns if we have enough data for at least two years. It would also be interesting to further investigate the Transfer Function Model, since model performance significantly improved. We were surprised that feedback would show up in this model, since we would not expect temperature to be impacted by energy usage.

This data was downloaded from Elizabeth's Duke Energy portal. Recently, her apartment complex installed a Google Nest thermostat in order to increase energy efficiency. It would be interesting to perform analysis a year from now to assess whether the Google Nest does, in fact, reduce energy usage. If that is the case, we could perform an intervention analysis.

Appendix A - *Original Data with AR(2) model*



The ARIMA Procedure

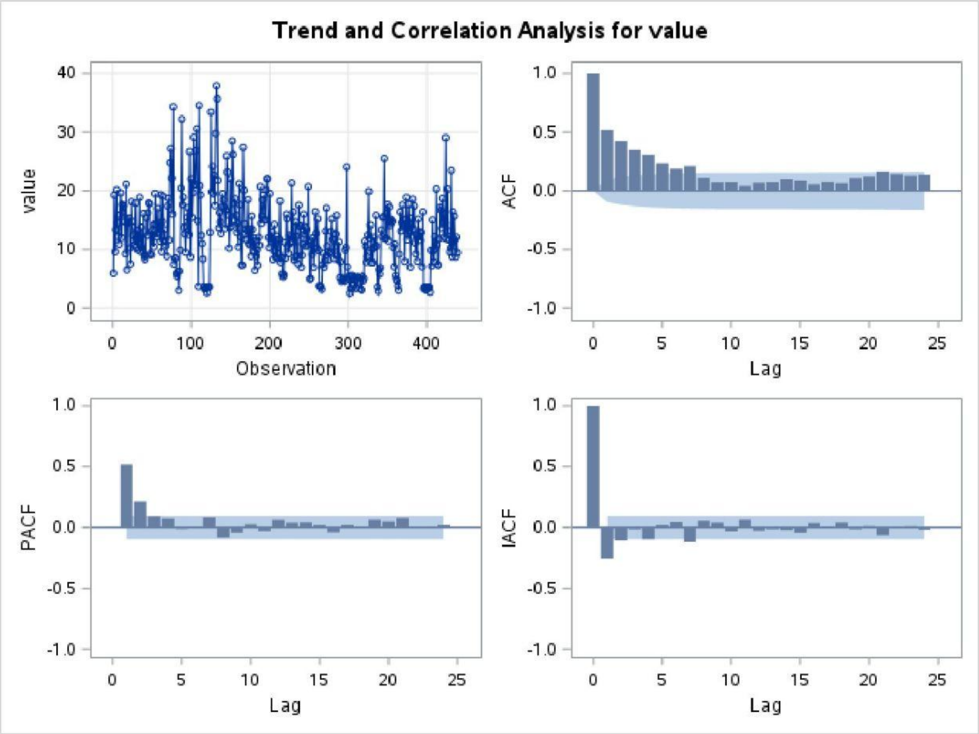
Name of Variable = value	
Mean of Working Series	12.59813
Standard Deviation	6.129526
Number of Observations	439

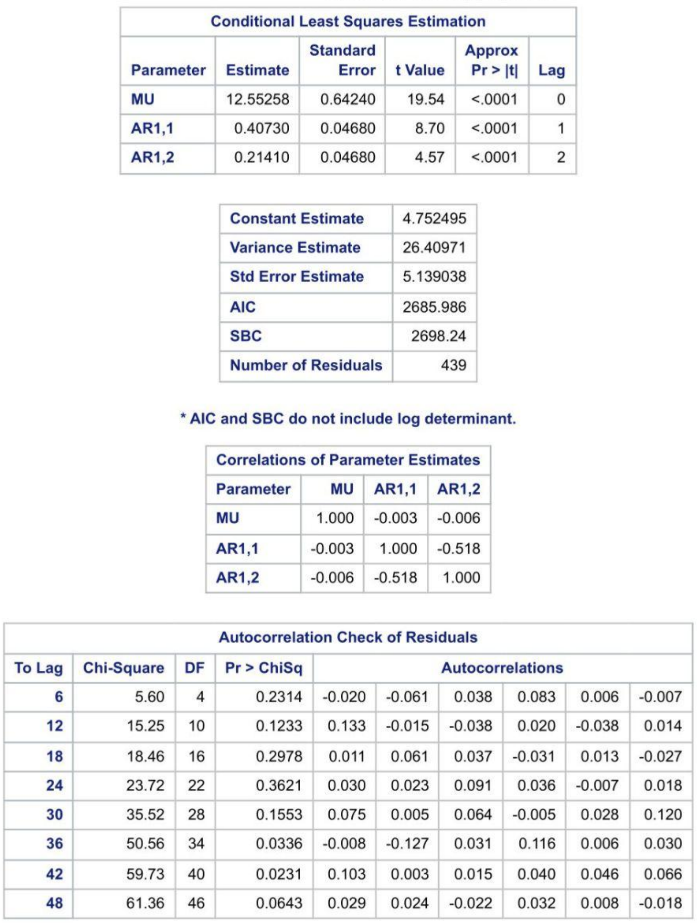
Autocorrelation Check for White Noise

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	335.09	6	<.0001	0.518	0.425	0.348	0.310	0.234	0.187
12	367.87	12	<.0001	0.210	0.111	0.072	0.073	0.040	0.068
18	383.95	18	<.0001	0.073	0.099	0.087	0.054	0.074	0.064
24	435.78	24	<.0001	0.109	0.123	0.166	0.145	0.130	0.139

Augmented Dickey-Fuller Unit Root Tests

Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Zero Mean	0	-40.1860	<.0001	-4.58	<.0001		
	1	-18.6515	0.0024	-3.08	0.0022		
Single Mean	0	-211.036	0.0001	-11.78	<.0001	69.39	0.0010
	1	-135.678	0.0001	-8.21	<.0001	33.72	0.0010
Trend	0	-227.120	0.0001	-12.39	<.0001	76.73	0.0010
	1	-151.512	0.0001	-8.67	<.0001	37.57	0.0010





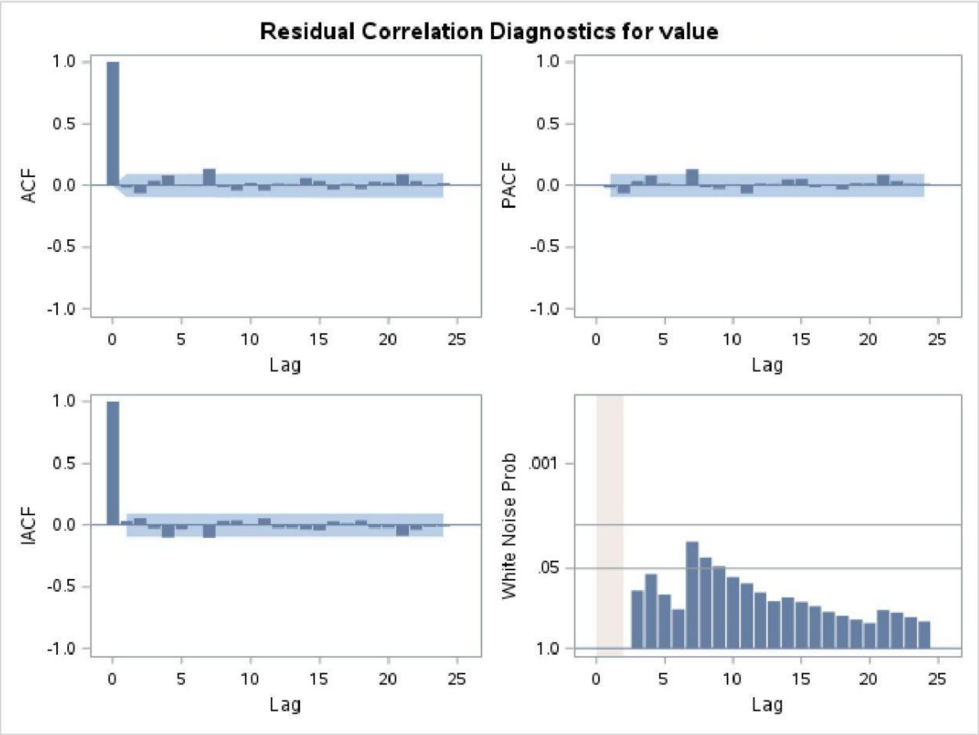
Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	12.55258	0.64240	19.54	<.0001	0
AR1,1	0.40730	0.04680	8.70	<.0001	1
AR1,2	0.21410	0.04680	4.57	<.0001	2

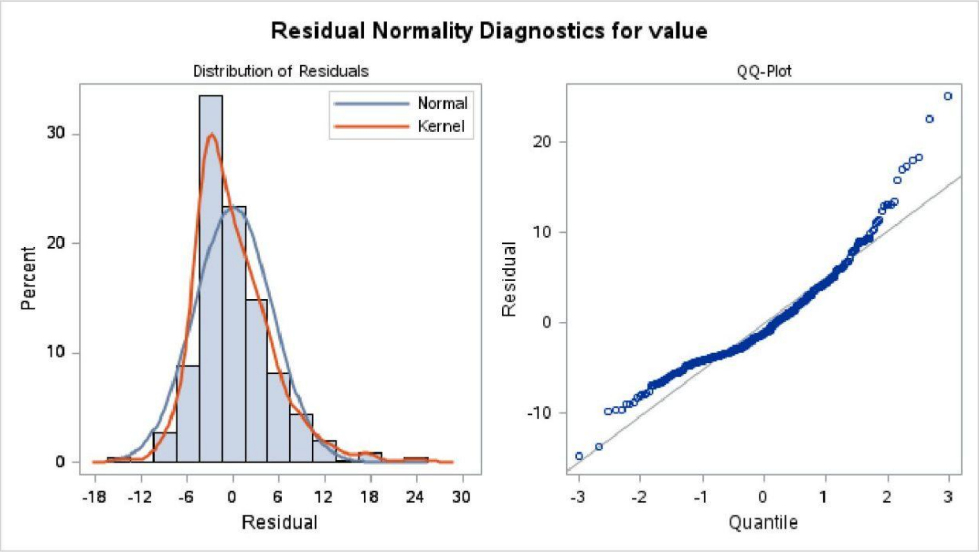
Constant Estimate	4.752495
Variance Estimate	26.40971
Std Error Estimate	5.139038
AIC	2685.986
SBC	2698.24
Number of Residuals	439

* AIC and SBC do not include log determinant.

Correlations of Parameter Estimates			
Parameter	MU	AR1,1	AR1,2
MU	1.000	-0.003	-0.006
AR1,1	-0.003	1.000	-0.518
AR1,2	-0.006	-0.518	1.000

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	5.60	4	0.2314	-0.020	-0.061	0.038	0.083	0.006	-0.007
12	15.25	10	0.1233	0.133	-0.015	-0.038	0.020	-0.038	0.014
18	18.46	16	0.2978	0.011	0.061	0.037	-0.031	0.013	-0.027
24	23.72	22	0.3621	0.030	0.023	0.091	0.036	-0.007	0.018
30	35.52	28	0.1553	0.075	0.005	0.064	-0.005	0.028	0.120
36	50.56	34	0.0336	-0.008	-0.127	0.031	0.116	0.006	0.030
42	59.73	40	0.0231	0.103	0.003	0.015	0.040	0.046	0.066
48	61.36	46	0.0643	0.029	0.024	-0.022	0.032	0.008	-0.018


Residual Correlation Diagnostics for value


Residual Normality Diagnostics for value

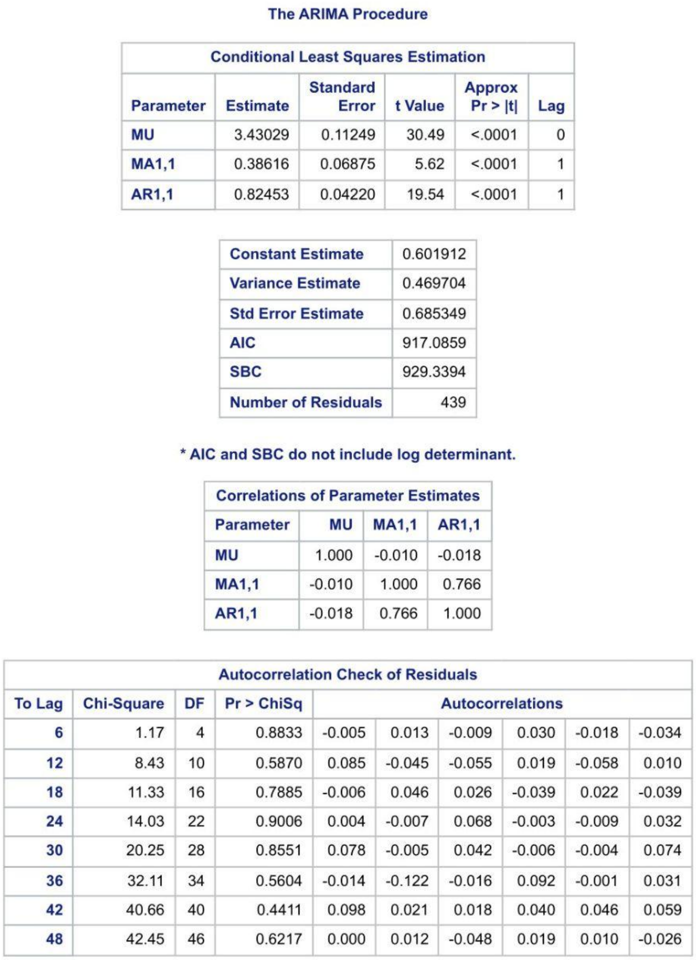
Distribution of Residuals

QQ-Plot

Model for variable value	
Estimated Mean	12.55258

Autoregressive Factors	
Factor 1:	1 - 0.4073 B**(1) - 0.2141 B**(2)

Appendix B - *Selected ARMA(1, 1) fit on a square root transformation*



The ARIMA Procedure

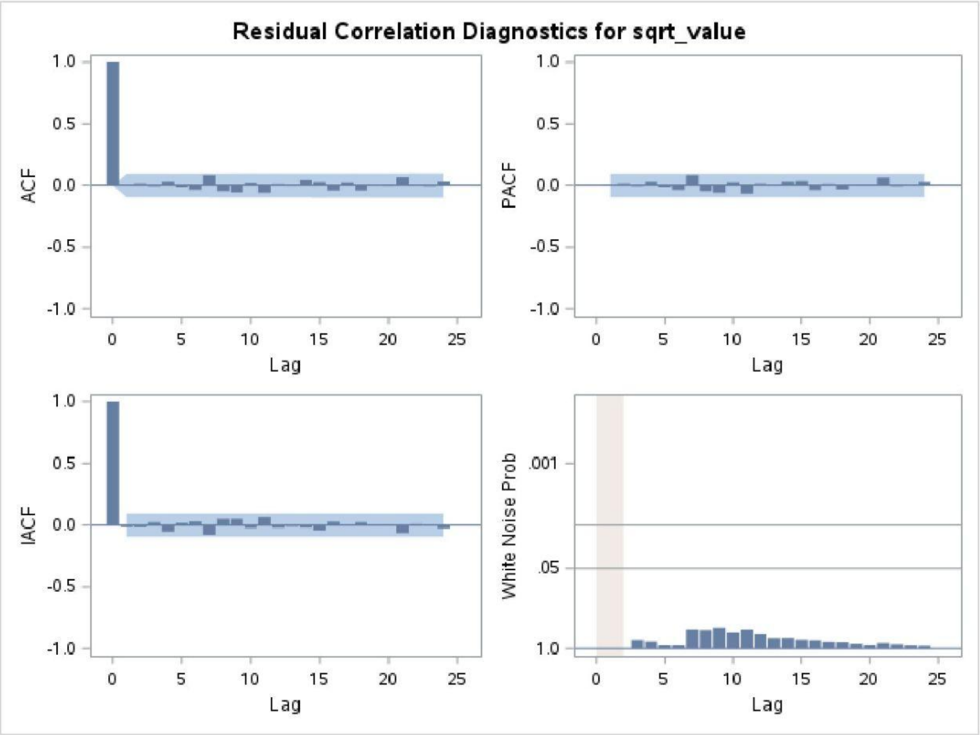
Conditional Least Squares Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	3.43029	0.11249	30.49	<.0001	0
MA1,1	0.38616	0.06875	5.62	<.0001	1
AR1,1	0.82453	0.04220	19.54	<.0001	1

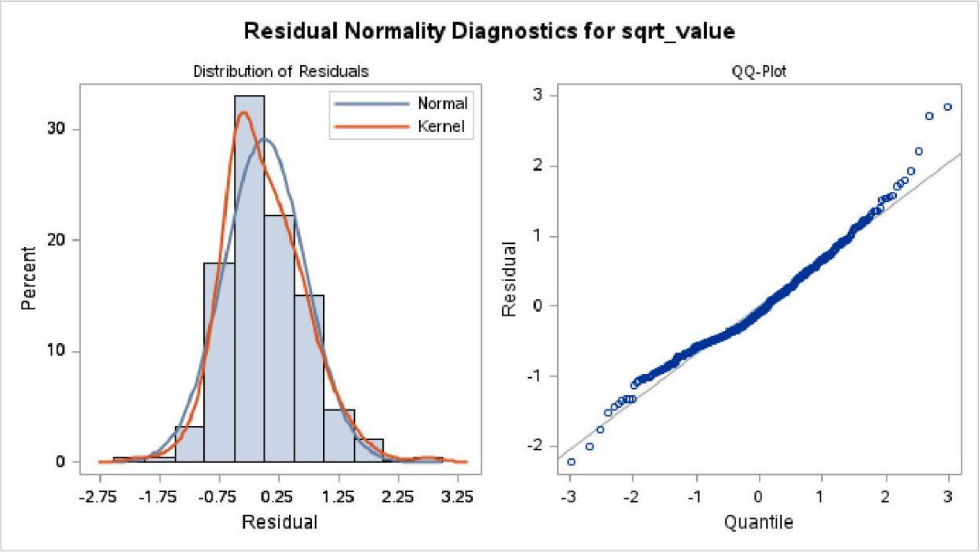
Constant Estimate	0.601912
Variance Estimate	0.469704
Std Error Estimate	0.685349
AIC	917.0859
SBC	929.3394
Number of Residuals	439

* AIC and SBC do not include log determinant.

Correlations of Parameter Estimates			
Parameter	MU	MA1,1	AR1,1
MU	1.000	-0.010	-0.018
MA1,1	-0.010	1.000	0.766
AR1,1	-0.018	0.766	1.000

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	1.17	4	0.8833	-0.005	0.013	-0.009	0.030	-0.018	-0.034
12	8.43	10	0.5870	0.085	-0.045	-0.055	0.019	-0.058	0.010
18	11.33	16	0.7885	-0.006	0.046	0.026	-0.039	0.022	-0.039
24	14.03	22	0.9006	0.004	-0.007	0.068	-0.003	-0.009	0.032
30	20.25	28	0.8551	0.078	-0.005	0.042	-0.006	-0.004	0.074
36	32.11	34	0.5604	-0.014	-0.122	-0.016	0.092	-0.001	0.031
42	40.66	40	0.4411	0.098	0.021	0.018	0.040	0.046	0.059
48	42.45	46	0.6217	0.000	0.012	-0.048	0.019	0.010	-0.026


Residual Correlation Diagnostics for sqrt_value


Residual Normality Diagnostics for sqrt_value

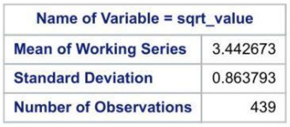
Model for variable sqrt_value	
Estimated Mean	3.430285

Autoregressive Factors	
Factor 1:	1 - 0.82453 B**(1)

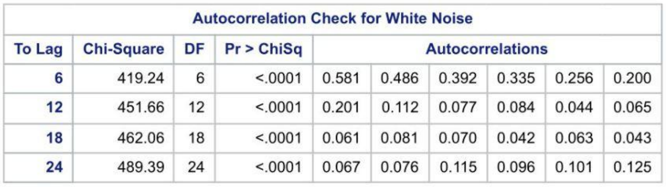
Moving Average Factors	
Factor 1:	1 - 0.38616 B**(1)

Appendix C - *Data after square root transformation*

The ARIMA Procedure

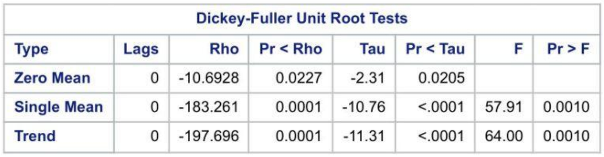


Name of Variable = sqrt_value	
Mean of Working Series	3.442673
Standard Deviation	0.863793
Number of Observations	439



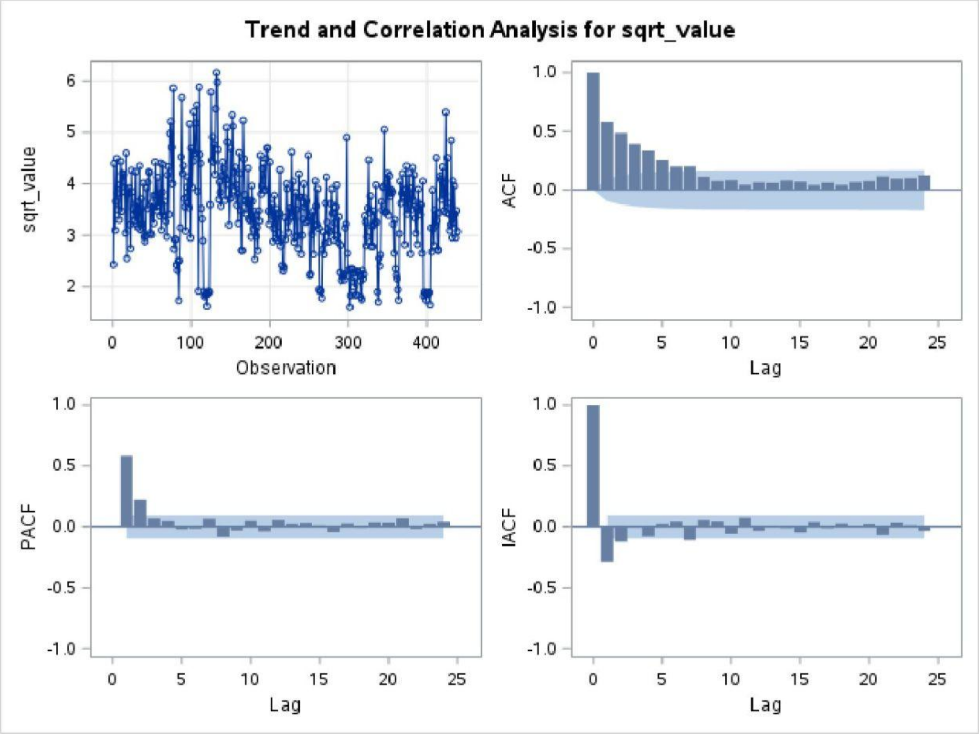
Autocorrelation Check for White Noise

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	419.24	6	<.0001	0.581	0.486	0.392	0.335	0.256	0.200
12	451.66	12	<.0001	0.201	0.112	0.077	0.084	0.044	0.065
18	462.06	18	<.0001	0.061	0.081	0.070	0.042	0.063	0.043
24	489.39	24	<.0001	0.067	0.076	0.115	0.096	0.101	0.125



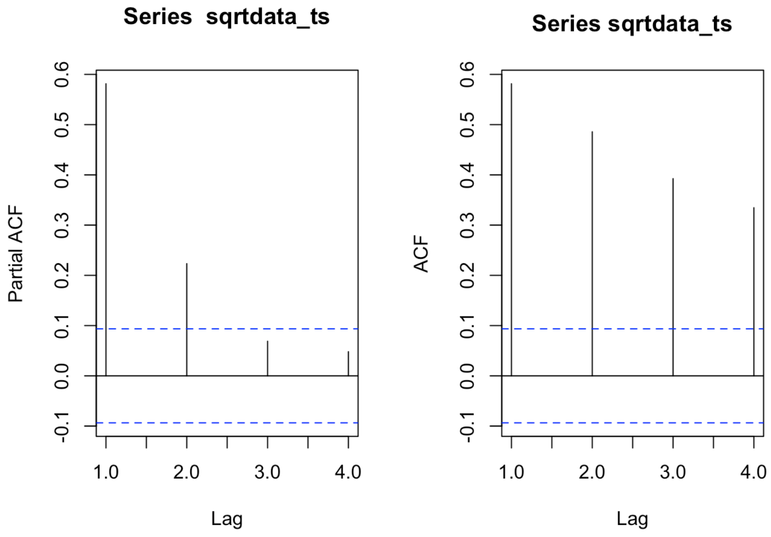
Dickey-Fuller Unit Root Tests

Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau	F	Pr > F
Zero Mean	0	-10.6928	0.0227	-2.31	0.0205		
Single Mean	0	-183.261	0.0001	-10.76	<.0001	57.91	0.0010
Trend	0	-197.696	0.0001	-11.31	<.0001	64.00	0.0010

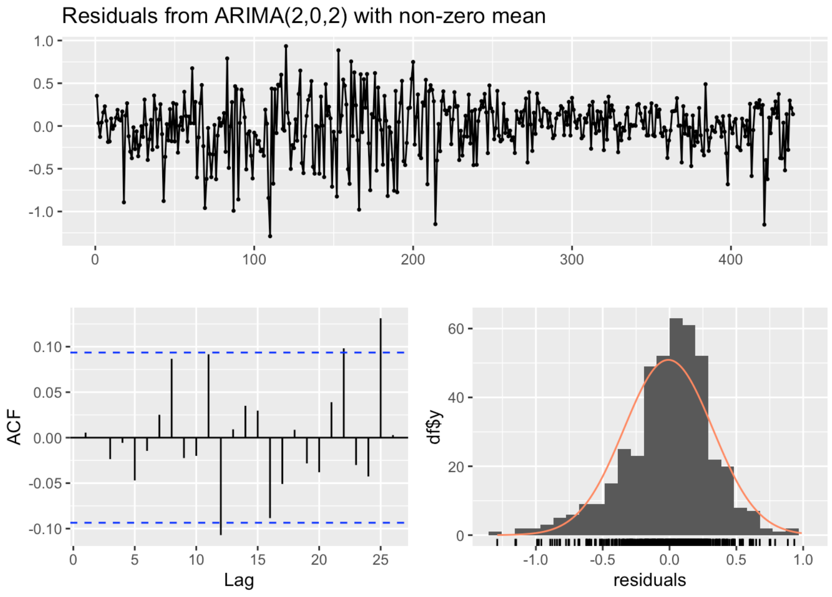


Appendix D - *Transfer Function Model Analysis*

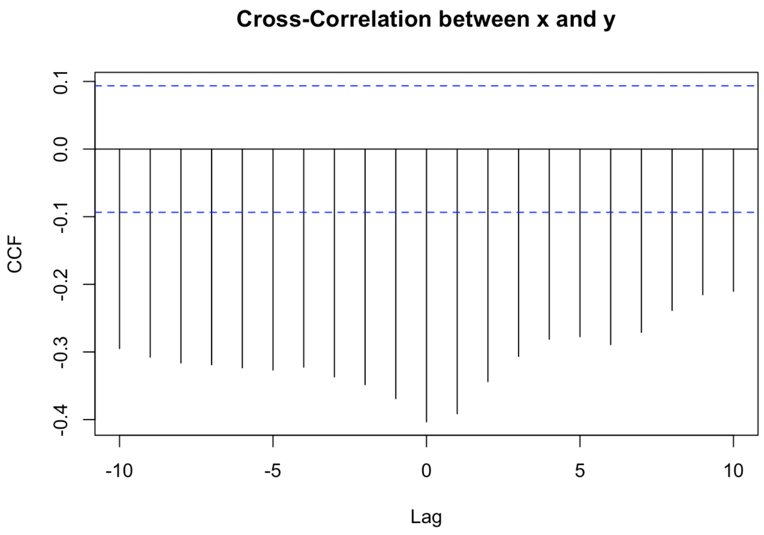
```
par(mfrow = c(1,2))
pacf(sqrtdata_ts, sqrttemp_ts)
acf(sqrtdata_ts, sqrttemp_ts)
```



```
tfmod = arimax(sqrttemp_ts, order=c(2,0,2), xreg=sqrtdata_ts, xtransf=sqrtdata_ts, trans
fer=list(c(1,0)))

# Check model diagnostics
checkresiduals(tfmod)
```



```r
tempdata_ts = ts(tempdata)
sqrtdata_ts = ts(sqrtdata)

# Compute and plot cross-correlation
ccf_result = ccf(sqrtdata_ts, tempdata_ts, lag.max=10, plot = FALSE)

plot(ccf_result, main="Cross-Correlation between x and y", ylab="CCF")
```



Appendix E - *Forecast Source Code*

```
# Load necessary libraries
library(forecast)

# Read the CSV file into a variable (change the file path accordingly)
file_path <- "C:/Users/User/Desktop/GradSchool/ST534/Project/originalData.csv"
data <- read.csv(file_path)

# Convert data to a numeric vector and time series
ts_data_numeric <- as.numeric(unlist(data))
ts_data_original_scale <- ts(ts_data_numeric)

# Square root transformation
transformed_data <- sqrt(ts_data_original_scale)

# Fit ARMA(1,1) model
arma_model <- Arima(transformed_data, order = c(1, 0, 1))

# Forecast ahead
forecasted_values <- forecast(arma_model, h = 60)

#plot(forecasted_values, main = "ARIMA model Forecast", xlab = "Day", ylab = "Sqrt(kWh)")

# Forecasted values in the original scale
forecasted_mean_original_scale <- forecasted_values$mean^2
forecasted_80Upper_original_scale <- forecasted_values$upper[,1]^2
forecasted_95Upper_original_scale <- forecasted_values$upper[,2]^2
forecasted_80Lower_original_scale <- forecasted_values$lower[,1]^2
forecasted_95Lower_original_scale <- forecasted_values$lower[,2]^2

# Extend the time index for the forecasted values
total_length<-length(ts_data_original_scale) + length(forecasted_mean_original_scale)

time_index_forecast <- 1:total_length

# We need the CI data to be the same length as the rest
combined_CI_upper_95 <- c(rep(NA, length(ts_data_original_scale)),
forecasted_95Upper_original_scale)
combined_CI_lower_95 <- c(rep(NA, length(ts_data_original_scale)),
forecasted_95Lower_original_scale)
```

```
combined_CI_upper_80 <- c(rep(NA, length(ts_data_original_scale)),
forecasted_80Upper_original_scale)
combined_CI_lower_80 <- c(rep(NA, length(ts_data_original_scale)),
forecasted_80Lower_original_scale)

# Plot the original data and the forecasts
plot(c(ts_data_original_scale, forecasted_mean_original_scale), main = "ARIMA model
Forecast", xlab = "Day", ylab = "kWh", type="l")

# Add confidence intervals to the plot
lines(forecasted_80Upper_original_scale, col = "red", lty = 1)
lines(forecasted_95Upper_original_scale, col = "red", lty = 1)
lines(forecasted_80Lower_original_scale, col = "red", lty = 1)
lines(forecasted_95Lower_original_scale, col = "red", lty = 1)

# Adding the 80% CI bounds to the plot and shading the area in between
polygon(c(time_index_forecast, rev(time_index_forecast)), c(combined_CI_upper_80,
rev(combined_CI_lower_80)), col = rgb(0, 1, 0, 0.3), border = NA)

# Adding the 95% CI bounds to the plot and shading the area in between
polygon(c(time_index_forecast, rev(time_index_forecast)), c(combined_CI_upper_95,
rev(combined_CI_lower_95)), col = rgb(0, 0, 1, 0.3), border = NA)
```