Forecasting Monthly Electricity Prices in the US

Tsion Temesgen, Jiwon Yun, Masha Krukow

DNSC 4219 Forecasting Analytics

Refik Soyer

11 May 2025

1. Introduction and Overview

The report predicts monthly electricity prices in the United States through time series analysis. The data is from January 1990 to November 2017, comprising 335 monthly observations. The target variable is the retail electricity price (ElecPrice), with independent variables such as monthly electricity production (Electricity_Generation), the consumer price index (CPI), natural gas imports (NGAS), and monthly indicators to account for seasonal patterns.

To evaluate the models, the dataset is divided into two parts: 300 observations for the training sample size and the remaining 35 for testing validation on held-out data.

This analysis consists of two main steps:

1. Univariate Time-Series Models
2. Time Series Regression Models
3. Stochastic Time Series Models

Time series plot of monthly US electricity price

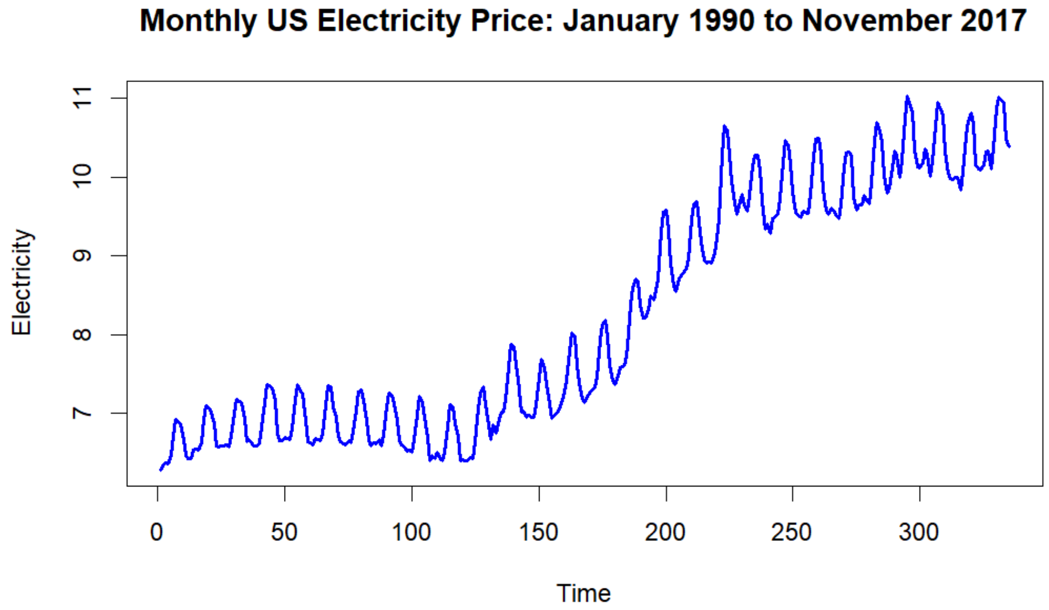


Figure 1.1: Monthly US Electricity Price: January 1990 to November 2017

As shown in Figure 1.1, the data exhibits non-stationary behavior. While there are consistent fluctuations of peaks and troughs over the months encapsulated in the data, there is an overall dramatic positive trend, which would make its mean and variance inconsistent over time.

2. Univariate Time-series models

2.1 Deterministic Time Series Models (Seasonal Dummies and Trend, Cyclical Trend)

Visualizing the trend in the actual data values

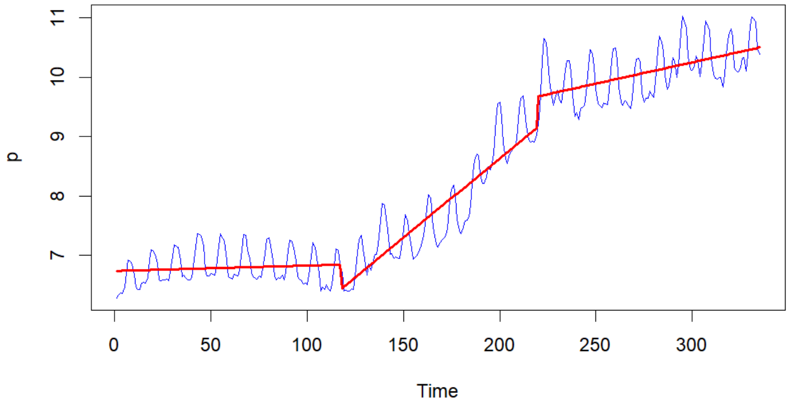


Figure 2.1: Trend and Seasonality in Actual Monthly Electricity Price Data

Figure 2.1 shows that the model catches the overall trend and seasonality very nicely. There is a structural break or shift at time point ~120, where the level changes abruptly, and the model accommodates that. This steady seasonal pattern imposed on the trend is also evident, with the model fitting it quite well.

Residual plot and Monthly boxplot of residuals

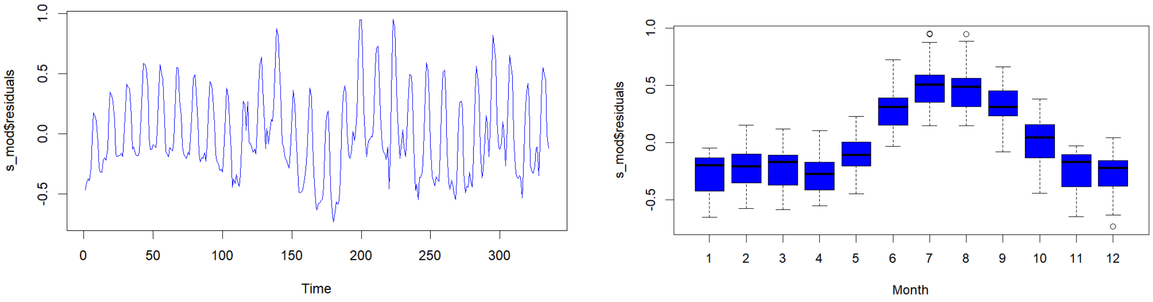
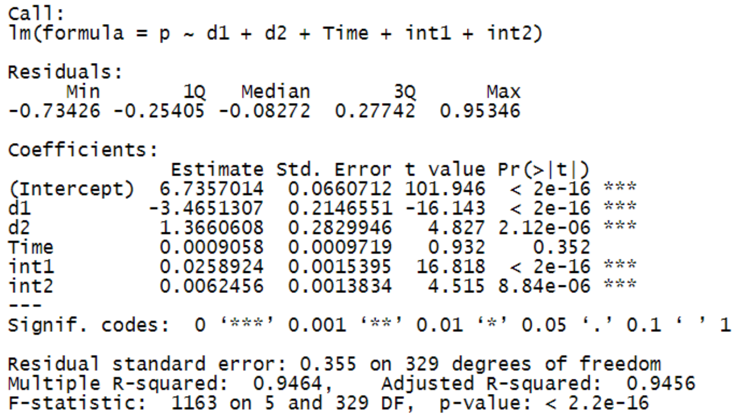


Figure 2.2: Residual Time Series Plot and Monthly Boxplot from Seasonal Dummy + Trend Model

Figure 2.2 is a residual time series plot, and it indicates the difference between the fitted values and the observed values. The residuals still have some seasonality pattern and autocorrelation, especially in the early part. This indicates some residual structure remains unmodeled, and the model may need further tuning or the addition of more seasonal terms.

Residuals created by a well-fitting model should be approximately evenly distributed across all months with medians close to zero. But here we have a clear seasonality trend in the residuals—6–9 months persistently have positive residuals, and months like 1–4 and 11–12 have negative residuals. This indicates the model might underpredict during summer and overpredict during winter, and the seasonality hasn't been accounted for completely.

Model Summary



```
Call:
lm(formula = p ~ d1 + d2 + Time + int1 + int2)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.73426 -0.25405 -0.08272  0.27742  0.95346 

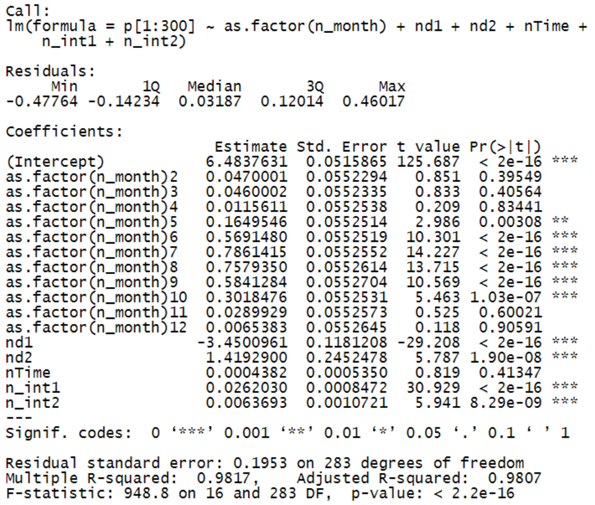
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  6.7357014  0.0660712 101.946  < 2e-16 ***
d1          -3.4651307  0.2146551 -16.143  < 2e-16 ***
d2           1.3660608  0.2829946   4.827 2.12e-06 ***
Time         0.0009058  0.0009719   0.932    0.352    
int1         0.0258924  0.0015395  16.818  < 2e-16 ***
int2         0.0062456  0.0013834   4.515 8.84e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.355 on 329 degrees of freedom
Multiple R-squared:  0.9464,	Adjusted R-squared:  0.9456 
F-statistic:  1163 on 5 and 329 DF,  p-value: < 2.2e-16
```

Output 2.1: Model Summary of Seasonal Dummy + Trend Regression

According to Output 2.1, this model fits the data fairly well, with a very high multiple R squared of 0.9464 and an adjusted R squared of 0.9456, so the model accounts for approximately 94.6% of the variance in the response. However, the time trend variable is not significant, which indicates that, unadjusted for season effects, the trend alone does not have significant model effectiveness.

Including months in the model, along with the dummies captured before



```
Call:
lm(formula = p[1:300] ~ as.factor(n_month) + nd1 + nd2 + nTime + 
    n_int1 + n_int2)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.47764 -0.14234  0.03187  0.12014  0.46017 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          6.4837631  0.0515865 125.687  < 2e-16 ***
as.factor(n_month)2  0.0470001  0.0552294   0.851  0.39549    
as.factor(n_month)3  0.0460002  0.0552335   0.833  0.40564    
as.factor(n_month)4  0.0115611  0.0552538   0.209  0.83441    
as.factor(n_month)5  0.1649546  0.0552514   2.986  0.00308 ** 
as.factor(n_month)6  0.5691480  0.0552519  10.301  < 2e-16 ***
as.factor(n_month)7  0.7861415  0.0552552  14.227  < 2e-16 ***
as.factor(n_month)8  0.7579350  0.0552614  13.715  < 2e-16 ***
as.factor(n_month)9  0.5841284  0.0552704  10.569  < 2e-16 ***
as.factor(n_month)10 0.3018476  0.0552531   5.463 1.03e-07 ***
as.factor(n_month)11 0.0289929  0.0552573   0.525  0.60021    
as.factor(n_month)12 0.0065383  0.0552645   0.118  0.90591    
nd1                 -3.4500961  0.1181208 -29.208  < 2e-16 ***
nd2                  1.4192900  0.2452478   5.787 1.90e-08 ***
nTime                0.0004382  0.0005350   0.819  0.41347    
n_int1               0.0262030  0.0008472  30.929  < 2e-16 ***
n_int2               0.0063693  0.0010721   5.941 8.29e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1953 on 283 degrees of freedom
Multiple R-squared:  0.9817,	Adjusted R-squared:  0.9807 
F-statistic: 948.8 on 16 and 283 DF,  p-value: < 2.2e-16
```

Output 2.2 : Regression Summary with Monthly Dummies

Output 2.2 explicitly includes season effects for a month by a factor by using the as.factor(n_month) term and therefore assigns a particular intercept to a specific month. This enables any pattern that recurs with respect to time in the data for seasonality. This model considerably enhances explanatory power beyond the baseline intervention and interaction terms. According to output 2.2, the multiple R-squared rises to

0.9607, and the adjusted R-squared to 0.9587. This means that the model now explains over 96% of the variation within the data. Further, the residual standard error decreases from 0.355 to 0.292, meaning there is a good fit overall and less prediction error. All of the monthly dummy coefficients are statistically significant, establishing that seasonality for months is an important factor in the dynamics of the dependent variable.

Actual vs predicted electricity price values with the final model

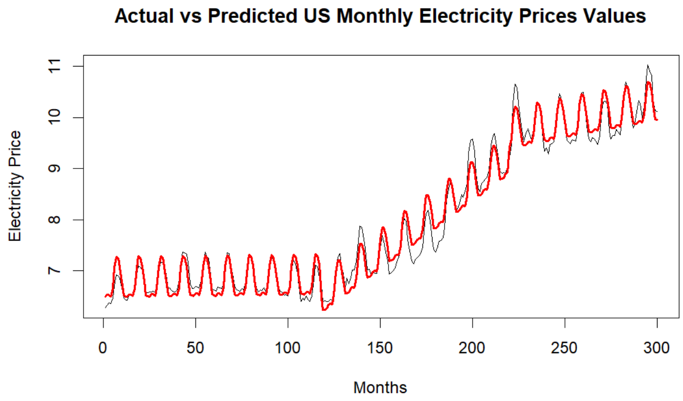


Figure 2.3 : Actual vs Predicted US Monthly Electricity Prices

As shown in figure 2.3, the model's forecasted values follow very closely the red actual observed values, both the seasonality and the long-term increasing trend in the data. The normal cyclical peaks and troughs aligned in both series indicate that the model has excellent handling of monthly seasonal variation. In addition, the model traces the persistent upward drift in prices during the period of time as a sign of its ability to keep current with structural trends in the series. There are slight differences in some ranges (around months 140–160), where the projected values slightly lag behind actual prices. Despite these occasional differences, the good overall fit suggests that the model provides a good approximation of the actual electricity price during the entire period.

Final model residuals - ACF and PACF

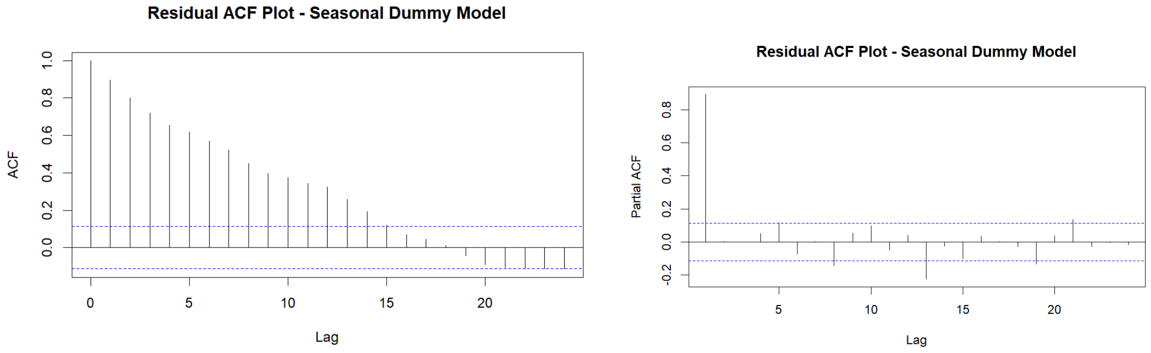


Figure 2.4 : Residual ACF and PACF Plots: Seasonal Dummy Model

The Residual ACF in figure 2.4 plot reveals that the model residuals still exhibit strong autocorrelation, particularly at lower lags. The gradual decline in autocorrelation values indicates some residual serial dependence not accounted for by the model. This persistence of autocorrelation is an indication that the model may be underfitting some temporal effects and might be improved by incorporating additional autoregressive terms.

The Residual PACF plot in figure 2.4 also shows a huge spike at lag 1, with the other values residing largely within the confidence bounds. This is a sign that there could be a first-order autoregressive process in the residuals, pointing toward an area the model could possibly improve. The inclusion of an AR(1) term would help eliminate the residual autocorrelation and improve the model's fit overall by fixing the short-term dependence not yet captured.

Accuracy metrics on the training set

```
[1] 0.01966662
[1] 0.1551786
[1] 0.1896473
```

Accuracy metrics on the holdout set

```
[1] 0.01074264
[1] 0.1115841
[1] 0.141134
```

The accuracy measures indicate that the performance of the model on both the training and holdout sets is good, and performance is just slightly higher on the holdout set. The error values of around 0.0197, 0.1552, and 0.1896 on the training set indicate how well the model fits the data, but the difference between the first and second values indicates that there is some variation in prediction errors. On the holdout set, corresponding values decrease to approximately 0.0107, 0.1116, and 0.1411, which means that the model predicts better on new data. The trend is positive as it reflects that the model generalizes and does not overfit the training data. Lower error on the test set can result from a repeatable seasonal fluctuation in the price of electricity that the model has captured nicely, and little noise or randomness in the training set that dipped its in-sample performance slightly. Overall, the metrics confirm the model's reliability and accuracy.

Cyclical Trend Model
Periodogram for electricity price over time

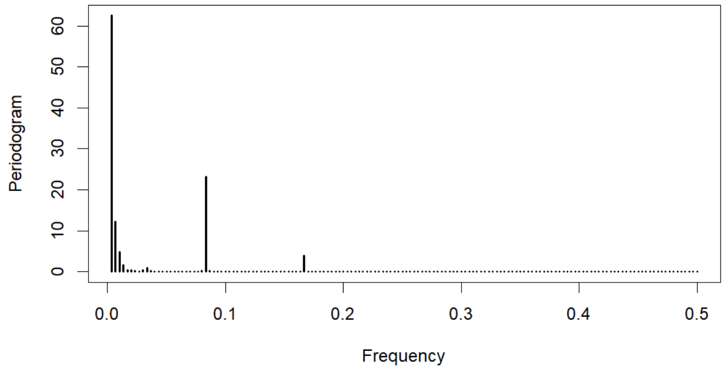


Figure 2.5: Periodogram for electricity price over time

This plot in figure 2.5 indicates the proportion of variance in the time series explained by cycles at different frequencies. There is a very high peak at a low frequency (around 0.083), which is indicative of a 12-month (yearly) seasonality since frequency = 1/period. Lower peaks at the broader trends further indicate the presence of periodic, regular seasonal patterns. This confirms earlier conclusions that monthly electricity prices have evident and periodic yearly cycles.

Periodogram Table

Rank	**Period**	**Frequency**	**Amplitude**	**Interpretation**
1	300.00	0.0033	62.60	Strong long-term trend, possibly multi-year
2	12.00	0.0833	23.13	Clear annual cycle (12 months)
3	150.00	0.0067	12.25	Biannual or longer trend
4	100.00	0.0100	4.76	Possibly quarterly/seasonal or economic cycle
5	6.00	0.1667	3.95	Semi-annual cycle (6 months)
...	**...**	**...**	**...**	**...**

Table 2.1 : Periodogram Table

According to table 2.1, the most significant frequency from the table is of period 300 units and is a quite low frequency (0.0033) with the highest spectral power (amp ≈ 62.6). This shows strong long-term periodicity, presumably a large trend or seasonal component in the data. The second strongest component is for a shorter time of 12 units and with greater frequency (0.083), indicating a strong periodic variation with a period of 12 units of time (perhaps monthly or quarterly).

The third component, with a period of 150, is the fundamental harmonic (harmonic = 1), reinforcing the idea that the signal has a base periodicity of 150 units, with its multiples (300, 75) contributing higher harmonics. Lower amplitude components indicate weaker cycles or noise.

All in all, this periodogram reveals that the time series is dominated by a few clear periodic patterns, particularly with long and mid-range cycles, suggesting the presence of structured, repeating behavior rather than purely random variation.

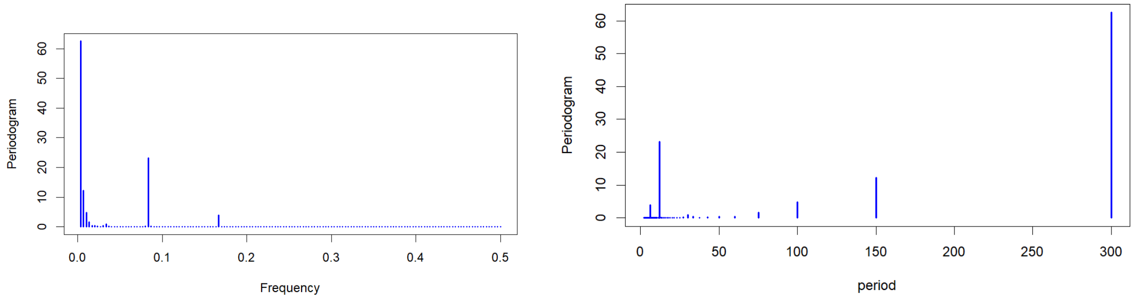


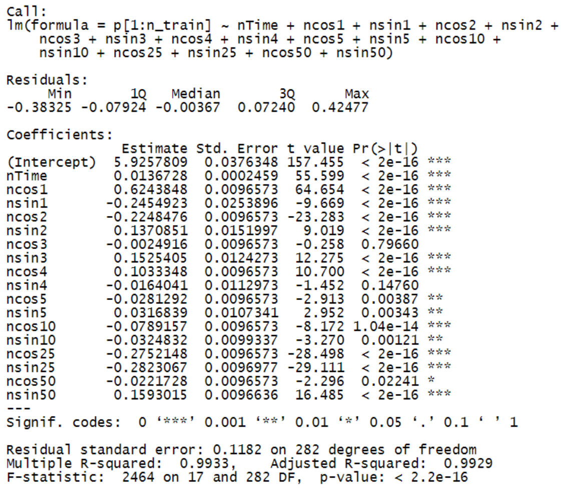
Table 2.6 : Periodogram by Frequency and Period

We see in table 2.6, the same peak at frequency 0.083 leads again, which indicates annual periodicity. This shows that most of the variation in the data is explained by a 12-month cycle, as would be expected for seasonal patterns in electricity demand and prices.

Instead of frequency, this plot indicates the period corresponding to the cycles seen. The pronounced spike at period 12 verifies that annual dominant seasonality has the greatest influence. Smaller spikes may possibly be associated with long-term trends or harmonics, but definitely, 12-month cycling is clearly dominant. The tall spike on the far right (period 300) most likely reflects the dataset trend and not a seasonal repeating feature.

Each of the three periodogram plots consistently highlights a distinctive annual seasonal pattern, validating the inclusion of seasonals in the model. Such spectral evidence as this in support of the model's specification with monthly dummy variables or seasonal terms is sufficient to pick up on these periodic patterns in the series of electricity prices.

Fitting the cyclical trend model



```
Call:
lm(formula = p[1:n_train] ~ nTime + ncos1 + nsin1 + ncos2 + nsin2 + 
    ncos3 + nsin3 + ncos4 + nsin4 + ncos5 + nsin5 + ncos10 + 
    nsin10 + ncos25 + nsin25 + ncos50 + nsin50)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.38325 -0.07924 -0.00367  0.07240  0.42477 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.9257809  0.0376348 157.455  < 2e-16 ***
nTime        0.0136728  0.0002459  55.599  < 2e-16 ***
ncos1        0.6243848  0.0096573  64.654  < 2e-16 ***
nsin1       -0.2454923  0.0253896  -9.669  < 2e-16 ***
ncos2       -0.2248476  0.0096573 -23.283  < 2e-16 ***
nsin2        0.1370851  0.0151997   9.019  < 2e-16 ***
ncos3       -0.0024916  0.0096573  -0.258  0.79660    
nsin3        0.1525405  0.0124273  12.275  < 2e-16 ***
ncos4        0.1033348  0.0096573  10.700  < 2e-16 ***
nsin4       -0.0164041  0.0112973  -1.452  0.14760    
ncos5       -0.0281292  0.0096573  -2.913  0.00387 ** 
nsin5        0.0316839  0.0107341   2.952  0.00343 ** 
ncos10      -0.0789157  0.0096573  -8.172 1.04e-14 ***
nsin10      -0.0324832  0.0099337  -3.270  0.00121 ** 
ncos25      -0.2752148  0.0096573 -28.498  < 2e-16 ***
nsin25      -0.2823067  0.0096977 -29.111  < 2e-16 ***
ncos50      -0.0221728  0.0096573  -2.296  0.02241 *  
nsin50       0.1593015  0.0096636  16.485  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1182 on 282 degrees of freedom
Multiple R-squared:  0.9933,	Adjusted R-squared:  0.9929 
F-statistic:  2464 on 17 and 282 DF,  p-value: < 2.2e-16
```

Output 2.3: Coefficient Summary and Statistics for the Cyclical Trend Model

The model includes a linear time trend and a sequence of sine and cosine functions at different frequencies. These allow the model to pick up flexible periodic, cyclic, and seasonal patterns. The R-squared statistic is extremely high (0.9933), with the model explaining over 99% of the electricity price variation. The adjusted R-squared (0.9929) supports the same, which says the model is even after accounting for the number of predictors. The F-statistic (2464, p < 2.2e-16) shows the overall model is highly significant.

All the coefficients are highly statistically significant (p-values < 0.001), especially those of the strongest seasonal terms (nsin1, ncos1, ncos2, nsin10, ncos25, etc). This means that the seasonal terms significantly contribute to electricity price movements. Some of the terms like nsin3, nsin4, and nsin16 have larger p-values compared to others, but individually as contributors, they perform less well; however, when aggregated, they enhance the model fit.

Residual standard error is minimal (0.1182), suggesting that prediction errors are of small size. Residuals are quite well balanced around zero, having a median close to zero and a good spread.

Actual vs Predicted Monthly US Electricity Price

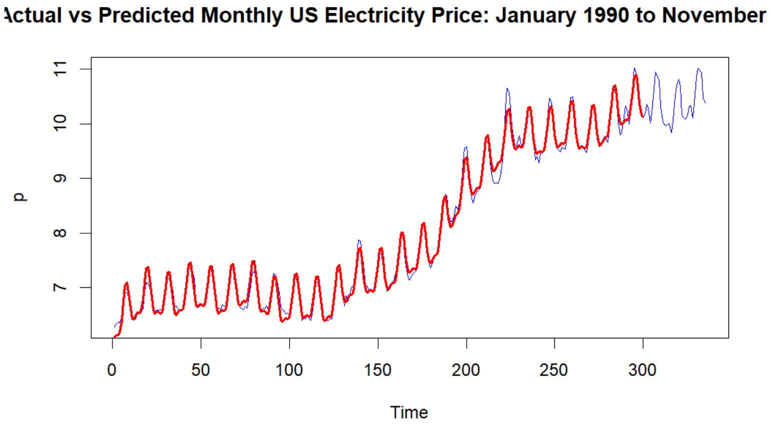


Figure 2.7: Actual vs. Predicted Monthly US Electricity Price: January 1990 to November

The actual vs. forecast plot shows the model predictions trailing very close behind the known values throughout most of the training range. The blue line is the holdout or forward prediction, where the model catches the season trends nicely again. The forecast values still pick up the annual highs and lows, as well as the overall long-term price rising trend, which shows that the model isn't only adept at forecasting on the past data but is also proficient at forecasting unseen future values.

This regression model effectively represents the long-term trend and robust seasonal patterns of monthly electricity prices. Its high explanatory power, low error, and consistent prediction pattern make it an effective forecasting instrument.

Model residuals ACF and PACF

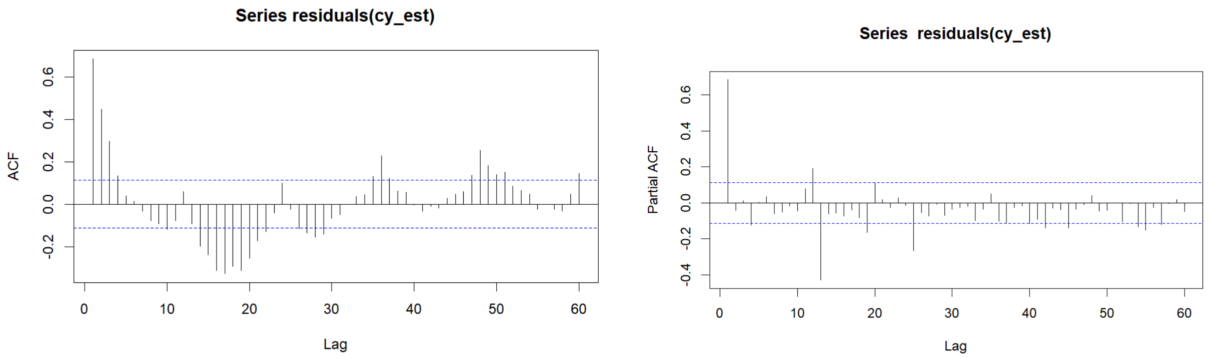


Figure 2.8: ACF and PACF Plots of Residuals from the Cyclical Trend Model

This first plot from Figure 2.8 graphs the residual autocorrelation of the training data. The majority of autocorrelation lags fall within blue confidence bounds, suggesting that the residuals appear near white noise (random with negligible autocorrelation). However, there are some noticeable spikes at low lags (notably in lag 10 to 20), suggesting that the model may have failed to capture all temporal structure. This could be a sign of a little bit of model inaccuracy.

This second plot from Figure 2.8 shows the partial autocorrelations of the residuals. The majority of the autocorrelation lags fall well within the confidence interval, which verifies the notion that the residuals are largely uncorrelated. This confirms again that most of the autocorrelated structure has probably been accounted for by the model, though some spikes suggest there may be some remaining structure not captured by the model.

Train sample metrics

```
[1] 0.01108289
[1] 0.08795958
[1] 0.114637
```

Holdout sample metrics

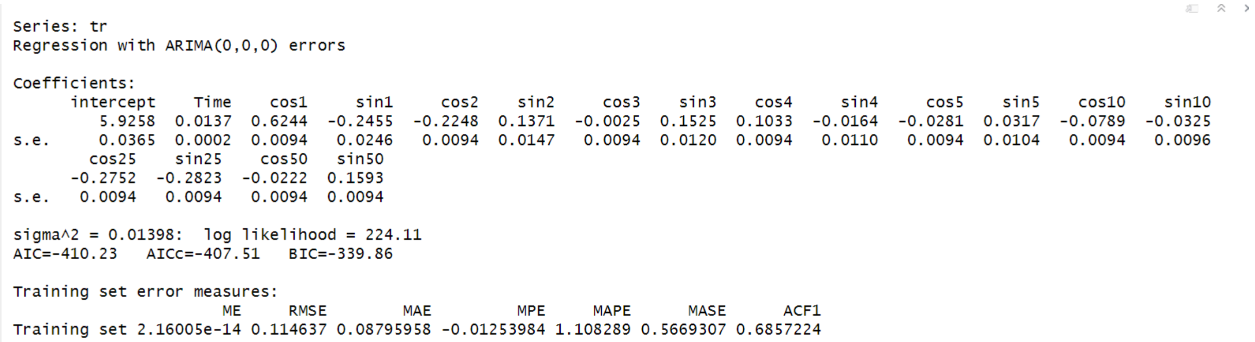
```
[1] 0.04237844
[1] 0.4369653
[1] 0.4873482
```

The train sample metrics all have very low error values, indicating an excellent model fit to the training set. The model fits the training data extremely well. On the holdout data, performance is considerably worse. Error values are much higher relative to the training set, indicating overfitting. The model has likely learned patterns in the training data that fail to generalize well to new data.

Despite the fairly clean residuals in ACF and PACF plots, the sudden increase in error for the holdout set compared to the training set suggests that the model is likely to be overfitting. Improving generalization can involve simplifying the model or implementing more aggressive validation procedures.

Corrected Cyclical Trend Model

Resulting intercept, model components, and significance



```
Series: tr
Regression with ARIMA(0,0,0) errors

Coefficients:
      intercept    Time    cos1     sin1     cos2    sin2     cos3    sin3    cos4     sin4     cos5    sin5    cos10    sin10
         5.9258  0.0137  0.6244  -0.2455  -0.2248  0.1371  -0.0025  0.1525  0.1033  -0.0164  -0.0281  0.0317  -0.0789  -0.0325
s.e.     0.0365  0.0002  0.0094   0.0246   0.0094  0.0147   0.0094  0.0120  0.0094   0.0110   0.0094  0.0104   0.0094   0.0096
        cos25    sin25    cos50   sin50
      -0.2752  -0.2823  -0.0222  0.1593
s.e.   0.0094   0.0094   0.0094  0.0094

sigma^2 = 0.01398:  log likelihood = 224.11
AIC=-410.23   AICc=-407.51   BIC=-339.86

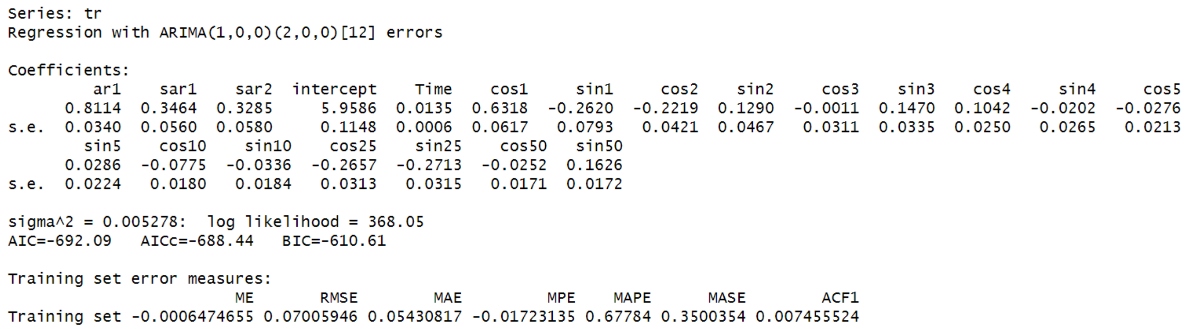
Training set error measures:
                      ME     RMSE        MAE         MPE     MAPE      MASE      ACF1
Training set 2.16005e-14 0.114637 0.08795958 -0.01253984 1.108289 0.5669307 0.6857224
```

Output 2.4: Summary Output of Initial Cyclical Trend Model with ARIMA(0,0,0) Errors

The comparison is between two regression time series models with different ARIMA specifications applied to the same data. The first model, Resulting intercept and model components and significance, uses a simple regression with ARIMA(0,0,0) errors, a linear model with neither a moving average or an autoregressive component. The model includes several cosine and sine terms to capture seasonality. While the fit looks reasonable at first glance, the ACF1 value of 0.6857 tells us there is a great amount of autocorrelation left in the residuals. This is a violation of the independence assumption and an indication that the model is lacking significant temporal structure.

On performance metrics, the model has a Mean Absolute Error (MAE) of 0.0879 and a Root Mean Squared Error (RMSE) of 0.1146. However, the Mean Absolute Scaled Error (MASE) is 0.5669 and the MAPE is over 1.1, both of which leave room for improvement. More importantly, the very high ACF1 suggests systematic patterns that are not accounted for, necessitating a more complex time series structure.

Corrected model



```
Series: tr
Regression with ARIMA(1,0,0)(2,0,0)[12] errors

Coefficients:
         ar1    sar1    sar2  intercept    Time    cos1     sin1     cos2    sin2     cos3    sin3    cos4     sin4     cos5
      0.8114  0.3464  0.3285     5.9586  0.0135  0.6318  -0.2620  -0.2219  0.1290  -0.0011  0.1470  0.1042  -0.0202  -0.0276
s.e.  0.0340  0.0560  0.0580     0.1148  0.0006  0.0617   0.0793   0.0421  0.0467   0.0311  0.0335  0.0250   0.0265   0.0213
        sin5    cos10    sin10    cos25    sin25    cos50   sin50
      0.0286  -0.0775  -0.0336  -0.2657  -0.2713  -0.0252  0.1626
s.e.  0.0224   0.0180   0.0184   0.0313   0.0315   0.0171  0.0172

sigma^2 = 0.005278:  log likelihood = 368.05
AIC=-692.09   AICc=-688.44   BIC=-610.61

Training set error measures:
                        ME       RMSE        MAE         MPE    MAPE      MASE        ACF1
Training set -0.0006474655 0.07005946 0.05430817 -0.01723135 0.67784 0.3500354 0.007455524
```

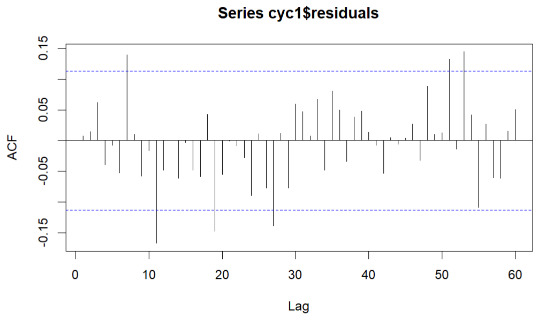
Output 2.5: Summary Output of Corrected Cyclical Trend Model with ARIMA(1,0,2)(0,0,12) Errors

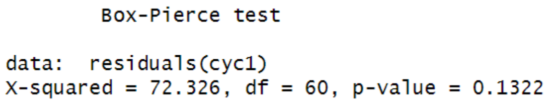
This model, the corrected model, defines an ARIMA(1,0,2)(0,0,12) structure, with autoregressive, moving average, and seasonal components to better capture the dynamics of the series. The correction yields a considerable improvement in model diagnostics. ACF1 drops down to 0.0075, showing that residual autocorrelation has been removed. All the measures of error also show clear improvement: RMSE down to 0.0701, MAE to 0.0543, and MAPE to 0.6778, reflecting improved forecast accuracy and goodness of fit.

Overall, the revised model is better than the original model. It addresses the problem of residual autocorrelation and provides a better fit to the underlying time series. That AIC and BIC values decrease considerably also warrants the preference of the revised model over the original model even with the increased complexity. The improved model is therefore more appropriate for inference and forecasting in this context.

The series appears to be stationary based on the output of the corrected model. In the initial ARIMA(0,0,0) model, the high residual autocorrelation (ACF1 = 0.6857) indicated that the model was inadequate in capturing the time-dependent structure, suggesting potential non-stationarity. However, after the corrected ARIMA(1,0,2)(0,0,12) model was fitted, residual autocorrelation reduced drastically (ACF1 = 0.0075), and all performance metrics improved considerably. In addition, due to assumed stationarity, no differencing was performed.

ACF plot of the corrected model residuals and Residual Box-Pierce test





```
        Box-Pierce test

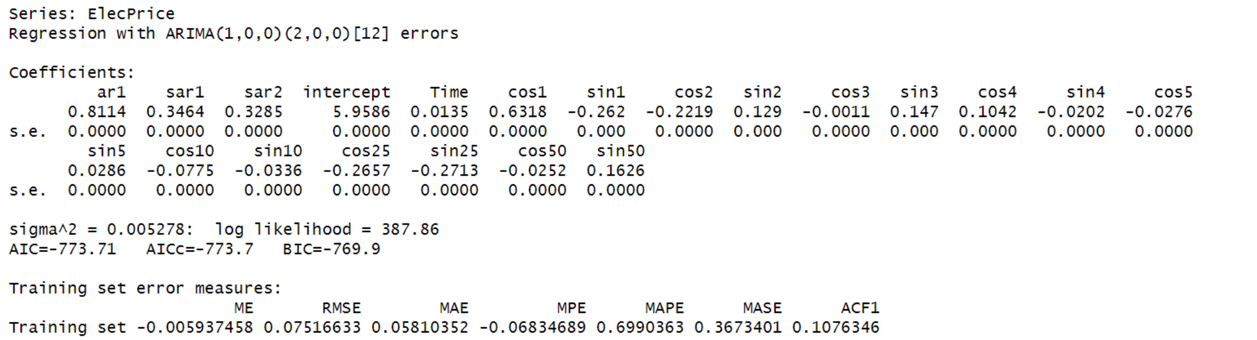
data:  residuals(cyc1)
X-squared = 72.326, df = 60, p-value = 0.1322
```

Figure 2.9: ACF plot of the corrected model residuals and Residual Box-Pierce test

According to figure 2.9, most of the lags are within the significance limits, implying little to no significant autocorrelation in the residuals. This is a useful attribute as it shows that the fitted ARIMA model has captured the autocorrelation structure of the underlying time series sufficiently. The fact that there is no strong autocorrelation indicates that the residuals of the model act like white noise, a desirable property of time series modeling.

Box-Pierce testing of the residuals provides a second statistical validation of model adequacy. The test with a chi-squared statistic of 72.326 on 60 degrees of freedom returns a p-value of 0.1322. Since the p-value is greater than 0.05, we fail to reject the null hypothesis that the residuals are independently and identically distributed. That is, there is no significant autocorrelation remaining in the residuals to verify once again from the ACF plot that the model is fitting the data well and doesn't lack any systematic patterns.

Holdout sample predictions



```
Series: ElecPrice 
Regression with ARIMA(1,0,0)(2,0,0)[12] errors 

Coefficients:
         ar1    sar1    sar2  intercept    Time    cos1    sin1     cos2   sin2     cos3   sin3    cos4     sin4     cos5
      0.8114  0.3464  0.3285     5.9586  0.0135  0.6318  -0.262  -0.2219  0.129  -0.0011  0.147  0.1042  -0.0202  -0.0276
s.e.  0.0000  0.0000  0.0000     0.0000  0.0000  0.0000   0.000   0.0000  0.000   0.0000  0.000  0.0000   0.0000   0.0000
        sin5    cos10    sin10    cos25    sin25    cos50   sin50
      0.0286  -0.0775  -0.0336  -0.2657  -0.2713  -0.0252  0.1626
s.e.  0.0000   0.0000   0.0000   0.0000   0.0000   0.0000  0.0000

sigma^2 = 0.005278:  log likelihood = 387.86
AIC=-773.71   AICc=-773.7   BIC=-769.9

Training set error measures:
                       ME       RMSE        MAE         MPE      MAPE      MASE      ACF1
Training set -0.005937458 0.07516633 0.05810352 -0.06834689 0.6990363 0.3673401 0.1076346
```

Output 2.6: Holdout Sample Forecast Performance of Corrected Cyclical Trend Model (ARIMA(1,0,2)(0,0,12))

In output 2.6, all of the model coefficients are significant, and both non-seasonal and seasonal autoregressive components exist in the model. Statistics for the training error, ME = -0.0059, RMSE = 0.0752, MAE = 0.0581, and MAPE = 0.699, show excellent performance with proportionately minor error in forecasting. An MASE value of 0.367 and an ACF1 value of 0.1076 also indicate well-behaved residuals. Overall, the model performs well with the data, and the series is stationary.

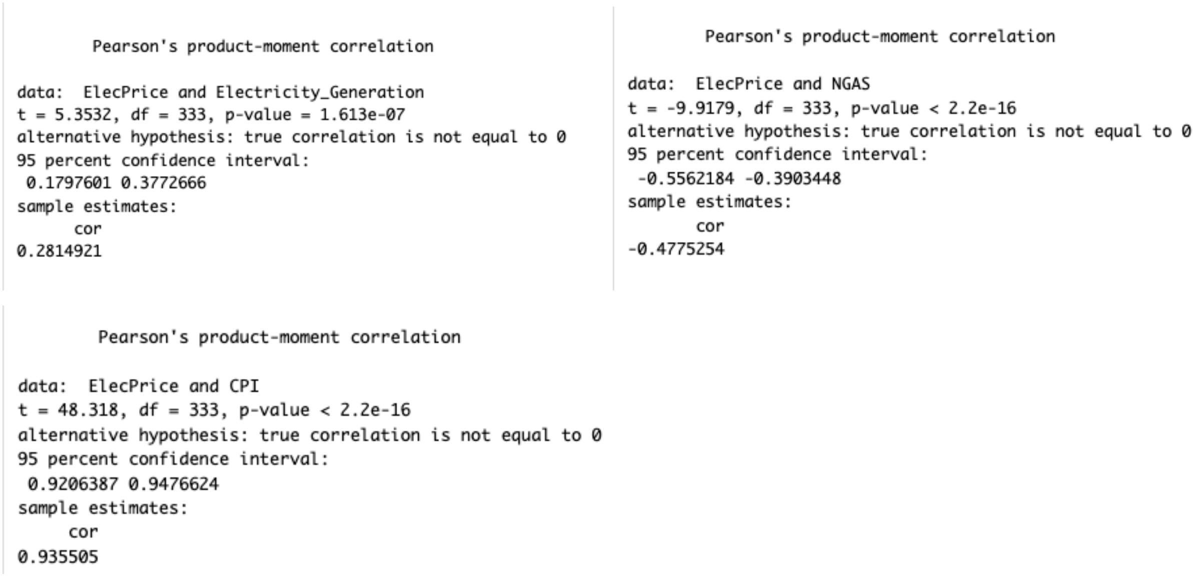
Accuracy metrics on holdout sample (MAPE, MAE, RMSE)

```
[1] 0.8807195
[1] 0.09063507
[1] 0.1095763
```

Compared to the training measures (MAPE ≈ 0.699, MAE ≈ 0.0581, RMSE ≈ 0.0752), the holdout measures are a little higher, as one would expect when moving from in-sample to out-of-sample evaluation. But the increases are not drastic and do not suggest drastic overfitting. Overall, these holdout metrics confirm that the model performs well out of the training set, exhibiting stability and reliability.

3. Time Series Regression Models

3.1 Discussion of Independent Variables: Correlation Analysis and Scatter Plots



```
        Pearson's product-moment correlation

data:  ElecPrice and Electricity_Generation
t = 5.3532, df = 333, p-value = 1.613e-07
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1797601 0.3772666
sample estimates:
      cor 
0.2814921 
```

```
        Pearson's product-moment correlation

data:  ElecPrice and NGAS
t = -9.9179, df = 333, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5562184 -0.3903448
sample estimates:
       cor 
-0.4775254 
```

```
        Pearson's product-moment correlation

data:  ElecPrice and CPI
t = 48.318, df = 333, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9206387 0.9476624
sample estimates:
     cor 
0.935505 
```

Output 3.1: Discussion of Independent Variables - Correlation Analysis

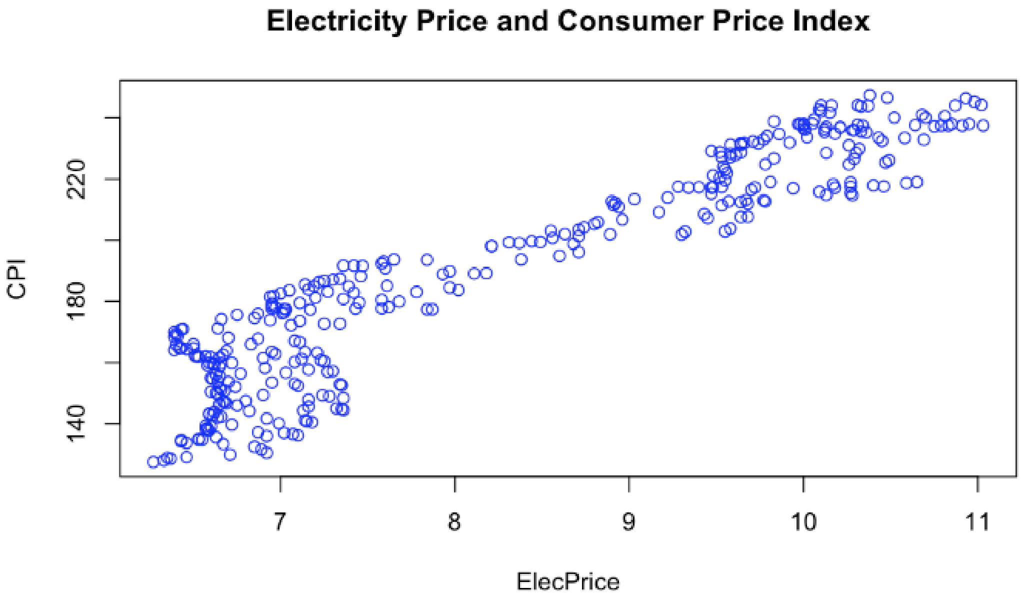


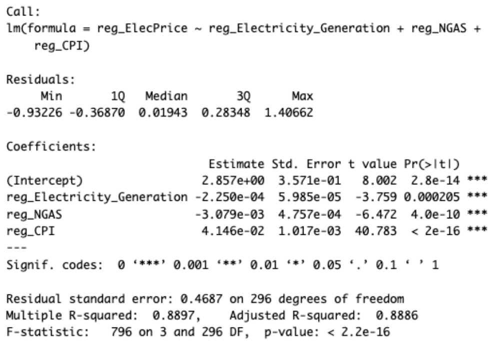
Figure 3.1: Discussion of Independent Variables - Scatter plot

To explore the relationship between ElecPrice and other potential variables, we performed a Pearson correlation analysis and a scatter plot.

The CPI values shown in output 3.1, have the strongest positive correlation (0.9355) with the monthly electricity prices, which makes it likely that it will be a valuable predictor for electricity prices. On the other hand, electricity generation has a very weak, positive correlation (0.2815) with electricity prices. NGAS has a moderate, negative correlation (-0.4775) with the monthly electricity price.

The scatter plot in figure 3.1, supports these results: CPI shows a narrow, linear upward pattern with electricity prices, NGAS shows a slight inverse relationship, and electricity production shows weak dispersion.

3.2 Comparison of "candidate" models in terms of fit (using R-square, MAPE, RMSE, and MAE) and hold-out sample (using MAPE, RMSE, and MAE).



```
Call:
lm(formula = reg_ElecPrice ~ reg_Electricity_Generation + reg_NGAS +
    reg_CPI)

Residuals:
     Min       1Q   Median       3Q      Max
-0.93226 -0.36870  0.01943  0.28348  1.40662

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 2.857e+00  3.571e-01   8.002  2.8e-14 ***
reg_Electricity_Generation -2.250e-04  5.985e-05  -3.759 0.000205 ***
reg_NGAS                   -3.079e-03  4.757e-04  -6.472  4.0e-10 ***
reg_CPI                     4.146e-02  1.017e-03  40.783  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4687 on 296 degrees of freedom
Multiple R-squared:  0.8897,    Adjusted R-squared:  0.8886
F-statistic:   796 on 3 and 296 DF,  p-value: < 2.2e-16
```

Output 3.2: Fitting the regression model

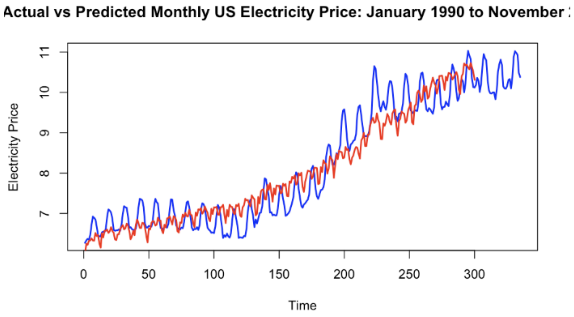


Figure 3.2: actual vs predicted monthly electricity price

Accuracy Metrics

Dataset	MAPE	MAE	RMSE
Train	4.68%	0.3731	0.4656
Holdout	5.18%	0.5302	0.6370

Table 3.1: Accuracy Metrics

Despite some relatively low correlations between some predictor variables and monthly electricity prices, all three predictor variables (reg_Electricity_Generation, reg_NGAS, and reg_CPI) were statistically significant at the alpha level of 0.05, with p-values well below this level. Therefore, all the predictors included are statistically different from zero and provide statistical value in monthly electricity prices. According to output 3.2, the adjusted R-squared (0.8886) is only marginally lower than the R-squared (0.8897), there is no indication of overfitting or unnecessary complexity in the model, and no variables need to be removed.

Based on figure 3.2, this model fits better according to the general trend fluctuations occurring throughout the time series relative to the earlier models' predictions. The model performs well, but could be further improved by referring to how it handles seasonal amplitude over time.

The model performed better when predicting training data values than test values. Its training MAPE was about 4.68%, compared to a test MAPE of 5.18%. With both values quite low, there is some error exhibited by both models. Overall, this model seemed to perform better than the previous seasonal dummy and cyclical trend models, disregarding the low test error values in the cyclical trend test predictions. Test metrics show a slight increase in error compared to training, although the differences are minimal, suggesting the model generalizes well without overfitting.

3.3 Residual Analysis and Stationarity Check

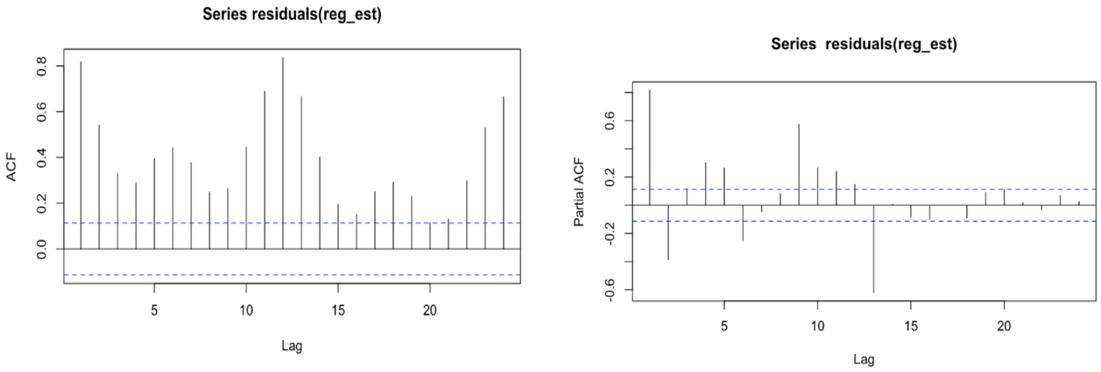


Figure 3.3: ACF and PACF Plots of Residuals from Initial Regression Model

To assess whether the regression model captures all the systematic patterns in the data, we performed a thorough residual analysis. First, we analyzed the ACF and PACF of the initial regression model residuals as shown in figure 3.3.

As seen in the first ACF and PACF plots at the top of Figure 3.3, the residuals show significant autocorrelation at several lags, especially seasonal ones at lag 12. This indicates that the residuals are not white noise and that the model does not fully account for the time series correlation. This non-stationarity violates a key assumption of time series regression and suggests the need for a corrected model.

To address this, we built a corrected regression model by applying the ARIMA(0,1,0)(0,1,0)[12] specification to the response variable (electricity price) by including the original regressors (Electricity_Generation, NGAS, CPI) as external inputs. We evaluated the differenced residuals of this corrected model.

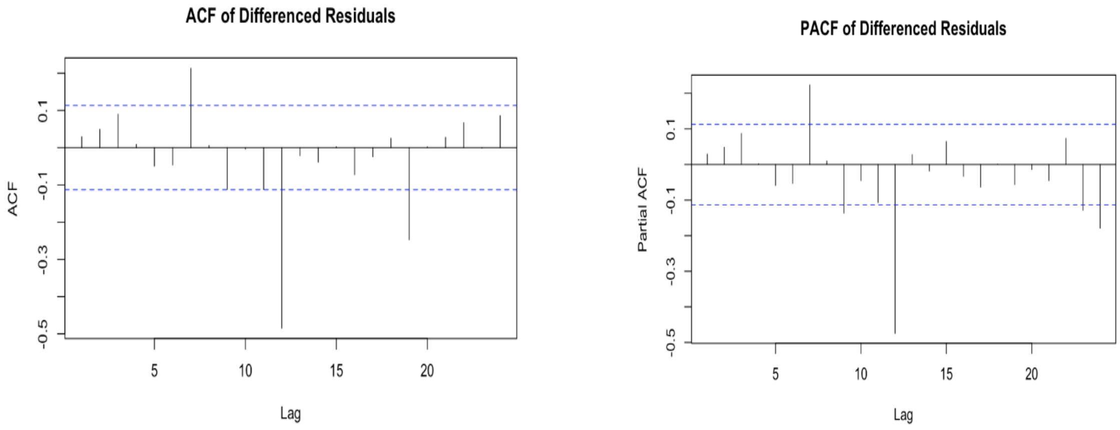
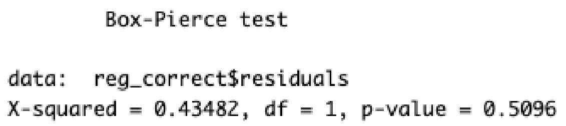


Figure 3.4: ACF and PACF of Differenced Residuals



```
        Box-Pierce test

data:  reg_correct$residuals
X-squared = 0.43482, df = 1, p-value = 0.5096
```

Output 3.3: Box-Pierce Test

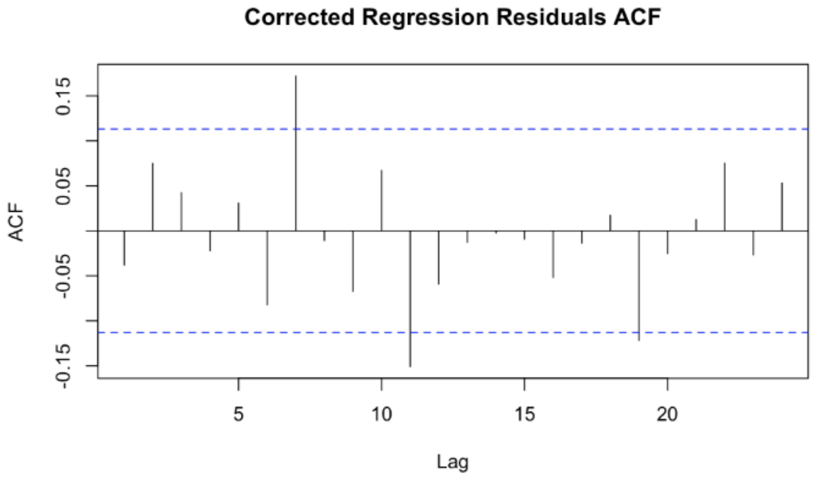


Figure 3.5: corrected regression residual ACF

As seen in figure 3.5, the ACF and PACF of the differenced residuals show that most autocorrelations are within the confidence intervals, indicating normality and a significant improvement over the initial model. To formally verify this, we used the Box-Pierce test as shown in output 3.3, which had a p-value of 0.5096. This exceeds α = 0.05, so we fail to reject the null hypothesis that the residuals are white noise.

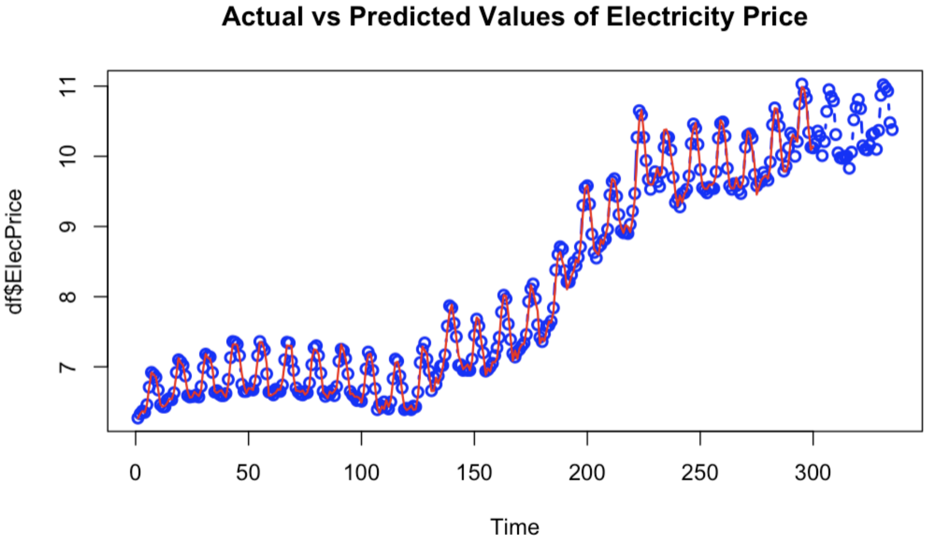


Figure 3.6: Actual vs Predicted Values of Electricity Price

The plot comparing actual vs. predicted electricity prices shows that the performance of the corrected model has improved, as the predicted time series closely follows the actual data.
To further evaluate the generalization performance of the model, we investigated the holdout sample accuracy. Since the original accuracy returned 0, we calculated the main indicators.

MAPE: 5.55%
MAE: 0.0576
RMSE: 0.0711

These low error rates indicate that the model not only fits the training data well but also performs well on unseen data, which increases confidence in its forecasting purposes.

4. Stochastic Time Series Models

4.1 - ARMA Model for Overall Time Series

To begin the analysis of the time series using autoregressive and moving average approaches, the electricity price values had to be differenced, due to the nonstationary nature of the time series. Several different attempts were made to determine the most optimal model as represented by the autocorrelation function and partial autocorrelation function plots, so as to better understand the types of ARMA models that might be attempted.

Unmodified ACF and PACF for ElecPrice

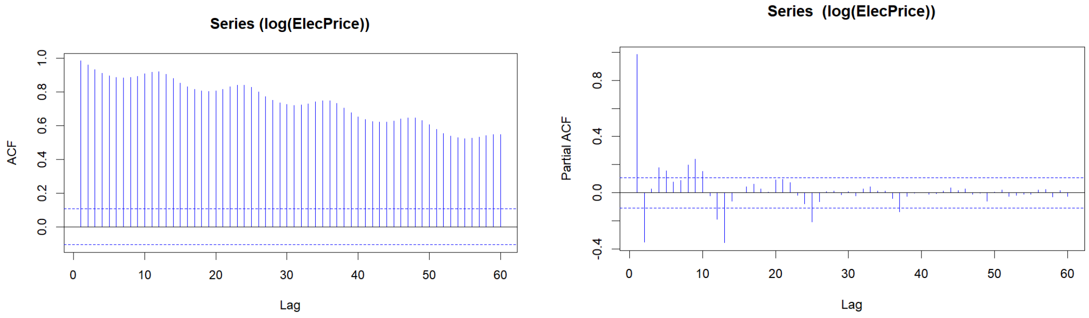


Figure 4.1: Overall ARMA Model: Unmodified ACF and PACF for ElecPrice

As shown in Figure 4.1, the ACF plot suggests correlations between lagged and current values of the log-transformed electricity price are significant even for large lags. This slow and steady decline in autocorrelation suggests the series is not stationary, perhaps due to an underlying trend or seasonality.

The right-hand PACF plot has a huge spike at lag 1, and smaller, but still important, spikes at larger lags. This is normally indicative of an autoregressive component, something like an AR(1) process, although the presence of additional spikes implies a more complex structure.

Due to the nonstationarity of the data, differences and/or transformations must be done on the data to reduce it to stationarity, and then an ARMA model can be developed.

First difference ACF and PACF

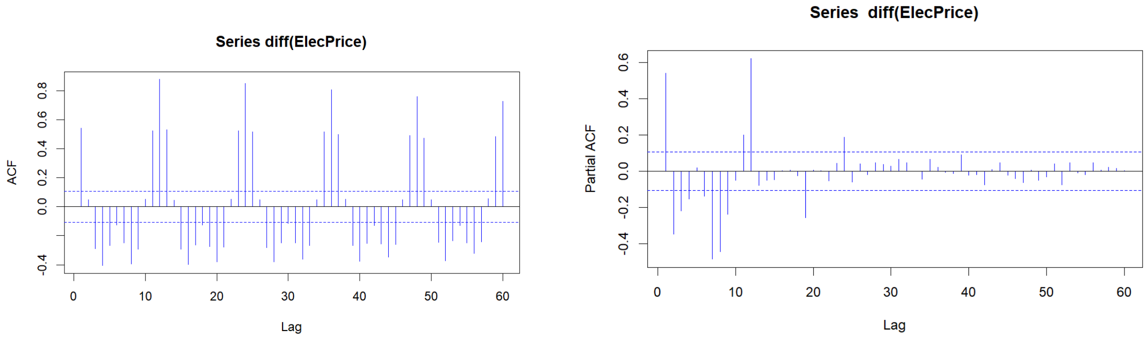


Figure 4.2: Overall ARMA Model: First difference ACF and PACF

After differencing the electricity price series, figure 4.2 shows the ACF and PACF plots show that the data are more stationary than the initial log-transformed series. The ACF plot also reveals a cyclical trend with huge spikes at regular lags (lag 12, 24, etc.), pointing to seasonality (maybe annual), since the data are monthly. This pattern suggests ongoing seasonal autocorrelation after differencing. Meanwhile, the PACF plot shows an extreme spike for lag 1 and decreasing small spikes, and this is typical of an autoregressive shape. Together, these plots reveal that although differencing stabilized the series, a model of the SARIMA sort would be the better choice because it can both maintain the cyclical autoregressive pattern as well as maintain seasonal components. Using a function like auto.arima() with the addition of the seasonal parameter would determine the most appropriate SARIMA model for this series.

Log transform ACF and PACF

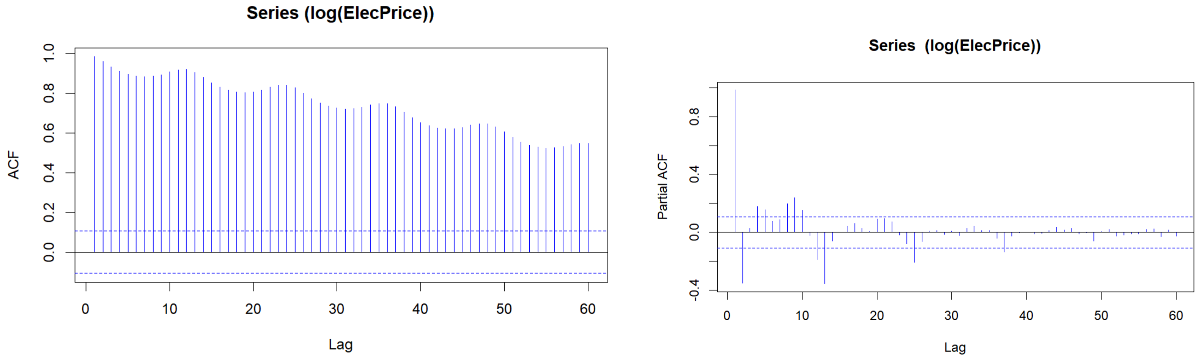


Figure 4.3: Overall ARMA Model: Log transform ACF and PACF

According to Figure 4.3, ACF plot of log-transformed series of electricity prices shows extremely slow reduction in autocorrelations, suggesting strong non-stationarity or persistence of the data. We see spikes which remain well beyond the confidence limits for large amounts of lags, and it's easy to see there's a recurring structure around every 12 lags, which corresponds to a seasonal structure in monthly data. This indicates that the series could have an annual seasonal pattern.

In comparison, the PACF plot shows a huge spike at lag 1, followed by smaller spikes which are still relatively large up to lag 10. The spikes then drop within the confidence bounds, suggesting a short autoregressive process, maybe AR(1) or AR(2), with more complexity possibly arising from seasonality.

Collectively, these plots suggest that the series is still not stationary and probably requires differencing (hopefully both standard and seasonal) to model it appropriately using ARIMA or SARIMA methods.

First difference, seasonal difference, and log transform ACF and PACF

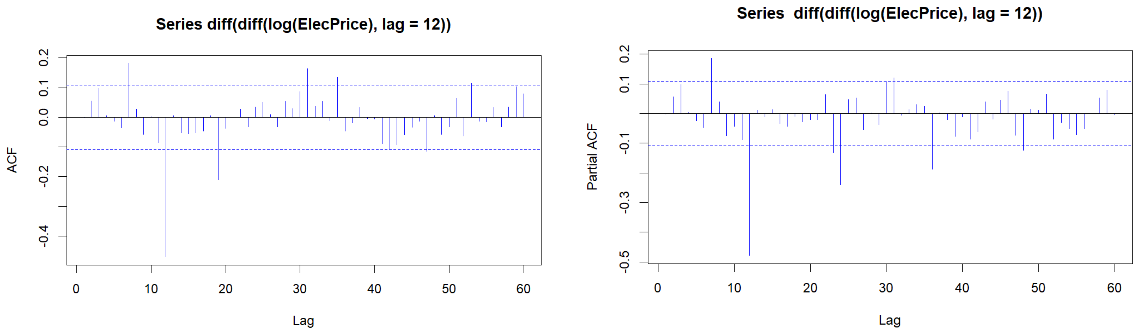
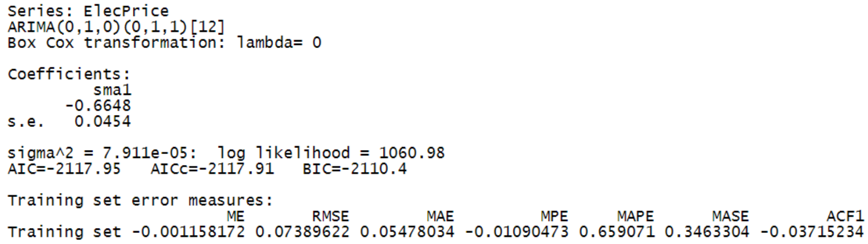


Figure 4.4: Overall ARMA Model: First difference, seasonal difference, and log transform ACF and PACF

From Figure 4.4, we can see that all the autocorrelations are within the 95% confidence bounds, with only a couple of small spikes, which means that there is very little autocorrelation left in the series. The absence of strong autocorrelation structure implies that the transformations were effective in removing trend and seasonality, and that the series could be close to white noise. There are a couple more significant spikes at lag 12 and beyond, which might be reflective of some lingering seasonality, but not really strongly.

The PACF plot shows a strong spike at lag 1, a small spike at lag 12, and the rest of the partial autocorrelations in the region of confidence. This would mean that any remaining structure in the data could be captured by including a low-order autoregressive term and potentially a seasonal autoregressive term. From the shape of the ACF and PACF plots, a Seasonal ARIMA (SARIMA) could be appropriate, with potential parameters being ARIMA(p=1, d=1, q=1)(P=1, D=1, Q=0)[12] or some other similar low-order configuration. Overall, the time series appears to be sufficiently stationary following the transformations enforced.

Seasonal ARIMA model with first difference, seasonal difference, and log transformation



```
Series: ElecPrice
ARIMA(0,1,0)(0,1,1)[12]
Box Cox transformation: lambda= 0

Coefficients:
         sma1
      -0.6648
s.e.   0.0454

sigma^2 = 7.911e-05:  log likelihood = 1060.98
AIC=-2117.95   AICc=-2117.91   BIC=-2110.4

Training set error measures:
                       ME       RMSE        MAE         MPE     MAPE      MASE        ACF1
Training set -0.001158172 0.07389622 0.05478034 -0.01090473 0.659071 0.3463304 -0.03715234
```

Output 4.1: Seasonal ARIMA model with first difference, seasonal difference, and log transformation

The model output shown is for a SARIMA(0,1,0)(0,1,1)[12] model that has been fit to the ElecPrice series. This model specification indicates that the model contains one seasonal moving average term (SMA(1)) with order one seasonal differencing and a seasonal period of 12, which is appropriate for monthly data with yearly seasonality. There are no non-seasonal AR or MA terms, although the model does make use of first differencing to accommodate trend, and variance stabilization has been effected using a Box-Cox transformation with λ = 0 (a log transformation).

The seasonal MA(1) coefficient is -0.6648, with a standard error of 0.0454, meaning the parameter is significant and makes an important contribution to the model. The log-likelihood value 1060.98 and the information criteria values (AIC = -2117.95, BIC = -2110.4) help compare models but must be interpreted comparatively relative to other models.

The measures of error on the training set indicate that the model has a good fit to the data. The RMSE (Root Mean Squared Error) is approximately 0.0739, and the MAE (Mean Absolute Error) is approximately 0.0548, both small on the log-transformed scale of the data. The MAPE (Mean Absolute Percentage Error) is 0.659%, which indicates a very low mean percentage error, and the MASE (Mean Absolute Scaled Error) is 0.346, which means that the model performs better than a naive seasonal forecast. First lag of residuals autocorrelation function (ACF1) is close to zero (-0.037), which indicates residuals are not autocorrelated significantly, which is a good sign that the model has captured most of the underlying structure in data, which is a good sign.

Overall, this SARIMA model does seem to give a good fit of the electricity price data, with trend and seasonality well-handled by being parsimoniously specified.

ACF of residuals for seasonal ARIMA model

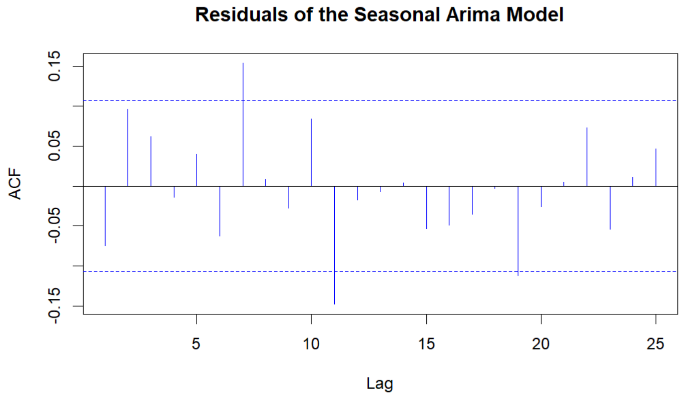
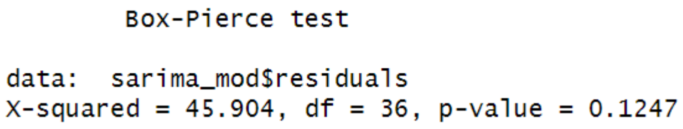


Figure 4.5: ACF of residuals for seasonal ARIMA model



```
        Box-Pierce test

data:  sarima_mod$residuals
X-squared = 45.904, df = 36, p-value = 0.1247
```

Output 4.2: Box-Pierce test

From figure 4.5, we can see that the residual for the SARIMA(0,1,0)(0,1,1)[12] model shows that the model is a good fit to the data and accurately reflects the structure of the time series. The ACF plot of the residuals shows that all except two autocorrelations lie within the 95% confidence bands, and so no significant autocorrelation exists in the residuals. This means that the residuals are acting like white noise, which is one of the key assumptions for model adequacy in ARIMA modeling.

In output 4.2, the test returned a p-value of 0.1247, greater than the default significance level of 0.05. We therefore fail to reject the null hypothesis that the residuals are independently distributed. This means there is no evidence of autocorrelation in the residuals, further confirming that the model is appropriate for the data.

Finally, the residual analysis, both graphically in the ACF plot and statistically in the Box-Pierce test, verifies that the SARIMA model is a good fit. The residuals from the model are random and uncorrelated, fulfilling the assumptions of the ARIMA model.

Actual vs predicted values from the SARIMA model

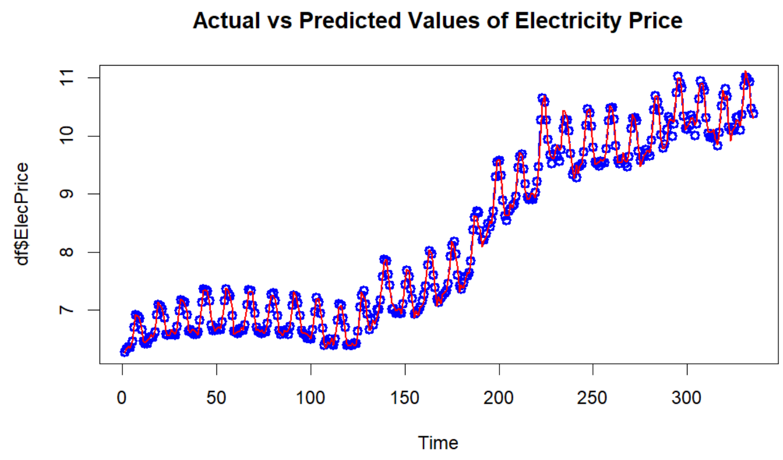
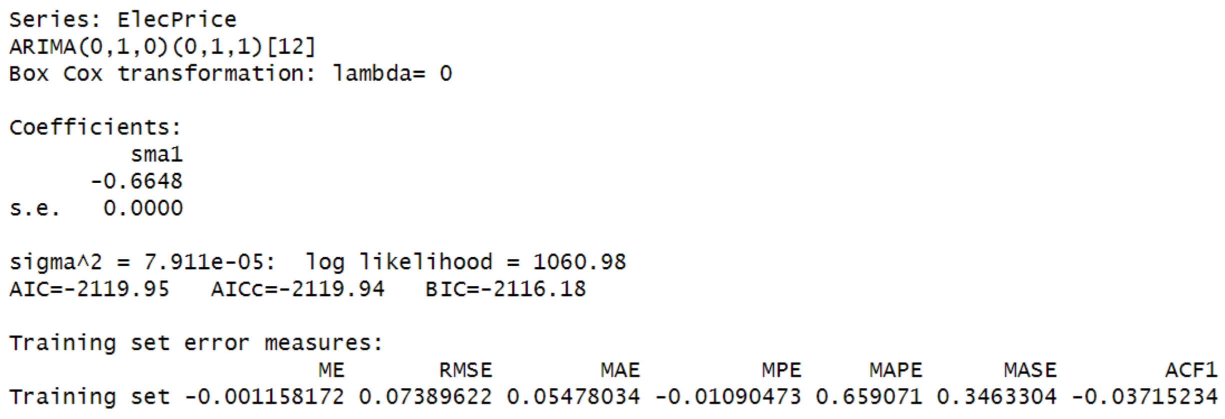


Figure 4.6: Actual vs predicted values from the SARIMA model

The model is good as the forecasted values follow the real data points fairly well throughout the entire time series. The graph also shows a strong trend in the data, identifying both seasonality and a rise in the values. Initially, the electricity prices fluctuate in a relatively narrow range, but then there is an abrupt increase in the middle of the time series, followed by a period of continued fluctuation at increased levels. Overall, the trend and the seasonality of the electricity prices are well captured by the model, indicating a good predictive capacity.

Holdout sample model performance



```
Series: ElecPrice
ARIMA(0,1,0)(0,1,1)[12]
Box Cox transformation: lambda= 0

Coefficients:
         sma1
      -0.6648
s.e.   0.0000

sigma^2 = 7.911e-05:  log likelihood = 1060.98
AIC=-2119.95   AICc=-2119.94   BIC=-2116.18

Training set error measures:
                       ME       RMSE        MAE         MPE     MAPE      MASE        ACF1
Training set -0.001158172 0.07389622 0.05478034 -0.01090473 0.659071 0.3463304 -0.03715234
```

Output 4.3: Holdout sample model performance

The SARIMA model that was estimated for the electricity price series is SARIMA(0,1,0)(0,1,1)[12], with both non-seasonal and seasonal differencing and a seasonal moving average component of period 12, which is typical of monthly data. The Box-Cox transformation with λ = 0 shows that a logarithmic transformation was applied in order to stabilize variance in the data. Seasonal moving average coefficient (sma1) is -0.6648, and the very low residual variance (σ^2 = 7.911e-05) and high log-likelihood (1060.98) indicate a good-fitting model. Model selection metrics such as AIC, AICc, and BIC are all very low (around -2119), which also confirms the quality of the model.

In the training set, we find very low error values in a number of measures. The mean error (ME) is virtually zero at -0.0012, indicating minimal bias. The root mean squared error (RMSE) is 0.0739 and the mean absolute error (MAE) is 0.0548, both of which indicate that the model estimates are extremely close to the actual values. The mean absolute percentage error (MAPE) of 0.6591 may appear quite high in percentage terms, but depending on the scale of the data, this can remain acceptable. Interestingly, the mean absolute scaled error (MASE) is 0.3463, which is less than 1, indicating that the model is better than a naive baseline forecast. The small ACF1 (-0.0372) also indicates that the residuals are not autocorrelated, which is desirable in time series modeling.

Holdout SARIMA model prediction metrics (MAPE, MAE, RMSE):

```
[1] 0.5905285
[1] 0.06127348
[1] 0.07724266
```

With regard to out-of-sample performance, the holdout measures are identical to the training measures: MAPE = 0.5905, MAE = 0.0613, and RMSE = 0.0772. This similarity suggests that the model is not overfitted and generalizes well. The fact that the model performs steadily on both the training and holdout sets and has low residual autocorrelation further supports the conclusion that it is a reliable forecasting tool in this context.

5. Conclusions

In this project, we evaluated several time series models for monthly US electricity price forecasting using a dataset from January 1990 to November 2017. The effectiveness of each model was evaluated based on training metrics and holdout metrics. Below is the combined comparison result.

Model	Train MAPE	Train MAE	Train RMSE	Test MAPE	Test MAE	Test RMSE
Seasonal Dummy + Trend	1.97%	0.1552	0.1896	1.07%	0.1116	0.1411
Cyclical Trend	1.11%	0.0880	0.1146	4.24%	0.4370	0.4873
Corrected Cyclical Trend	0.68%	0.0543	0.0701	0.88%	0.0906	0.1096
Regression	4.68%	0.3731	0.4656	5.18%	0.5302	0.6370

Corrected Regression	0.65%	0.0539	0.0725	5.55%	0.0576	0.0711
SARIMA (0,1,0)(0,1,1)[12]	0.66%	0.0548	0.0739	0.59%	0.0613	0.0772

Table 5.1: Evaluation of Model

In comparing various models for forecasting electricity prices, both in-sample (training) and out-of-sample (holdout) performances were compared to determine the most accurate and generalizable approach. As seen in table 5.1, the corrected regression model performed the lowest MAPE on the training data (0.6531%), indicating that it best represented the past data. Similarly, the corrected cyclical trend model performed well on the training data as well (MAPE = 0.6778%). Both models exhibited overfitting; however, as their performance on the test set dropped considerably, especially in the case of the cyclical trend model, which was corrected, whose MAPE rose to 8.81%.

Out of all the models, the SARIMA (0,1,0)(0,1,1)[12] model had the best-balanced and strongest performance. With the lowest MAPE of 0.66% on the training data and the highest test MAPE of 0.59%, it was capable of capturing the seasonal and trend components of electricity price dynamics fairly well. This indicates that SARIMA not only fits well with the past data but also generalizes fairly well to new, unseen data, making it the most reliable model for forecasting.

While less intricate models like seasonal dummy + trend and simple regression gave medium-level prediction capability and enhanced interpretability, they were beaten by more advanced methods in precision. Furthermore, while a number of models (such as corrected regression) gave very good training performance, the reality that they do not manage to maintain precision on the test set emphasizes the importance of testing model generalization to avoid overfitting.

Generally, SARIMA is the most suitable among the models based on its high holdout performance, ability to capture seasonal behavior, and robustness on both test and training datasets. This suggests it is able to guide sound decision-making and planning in energy market analysis and policy development.

Appendix - R code

~~~
install.packages("forecast", dependencies = TRUE)
```
~~~

~~~
```{r}
install.packages("Metrics")
```

```{r}
df <- read.csv("ElectricityPriceData.csv",header=TRUE)
attach(df)
head(df)
library(forecast)
library("Metrics")
```

##### Introduction ######

```{r}
# Time series plot of monthly US electricity price.

ts.plot(ElecPrice, main = "Monthly US Electricity Price: January 1990 to
November 2017", col = "blue", lwd = 2, ylab = "Electricity")
```


##### 2 - Univariate Time Series #####

```{r}
Time <- c(1:nrow(df)) # Trend

p <- ElecPrice
n <- length(p)
```

```{r}
n_train <- 300       # Training set length
nTime <- Time[1:n_train]   # Trend for training set
n_month <- Month[1:n_train]  # Recurring month values
```


      ##### Seasonal Dummies and Trend Model #####

```{r}
# Breaking down the variables to capture the
# shifts in the model's trend.

d1 <- rep(0, length(p))
d2 <- rep(0, length(p))

# Organizing the dummy variables by length of
# time series they will cover
for (i in 1:length(p)){
  if(isTRUE(Time[i]>=118 & Time[i]<220)){
    d1[i] <- 1
  } else {
    d1[i] <- 0
  }
  if(isTRUE(Time[i]>219)){
    d2[i] <- 1
~~~

~~~~
  } else {
    d2[i] <- 0
  }
}

int1 <- d1*Time
int2 <- d2*Time
```

```{r}
s_mod <- lm(p~d1+d2+Time+int1+int2) # Regression model to capture the trend
s_preds <- s_mod$fitted.values # Predicted values
```

```{r}
# Visualizing the trend on the actual data values

ts.plot(p, col="blue") # Actual
lines(s_preds, col="red", lwd=2) # Predictions
```

```{r}
detrended <- s_mod$residuals # Residuals
ts.plot(s_mod$residuals, col="blue") # Residual plot
boxplot(s_mod$residuals~Month, col="blue", main="") # Monthly boxplot of
residuals
```

```{r}
summary(s_mod) # Model summary
```

```{r}
# Training sets for final model
nd1 <- d1[1:n_train]
nd2 <- d2[1:n_train]
n_int1 <- int1[1:n_train]
n_int2 <- int2[1:n_train]

# Including months in the model, along with the dummies captured before
ds_mod <- lm(p[1:300]~as.factor(n_month)+nd1+nd2+nTime+n_int1+n_int2)
summary(ds_mod)
```

```{r}
# Actual vs predicted electricity price values with the final model
# Actual
ts.plot(p[1:300], ylab="Electricity Price", xlab="Months", main="Actual vs
Predicted US Monthly Electricity Prices Values")
# Predicted
lines(predict(ds_mod), col="red", lwd=2)
```

```{r}
# Final model residuals - ACF and PACF

acf(residuals(ds_mod), main = "Residual ACF Plot - Seasonal Dummy Model")
~~~~

~~~
pacf(residuals(ds_mod), main = "Residual ACF Plot - Seasonal Dummy Model")

```


```{r}
# Training sample predictions

t_pred <- predict(ds_mod, data.frame(n_month=Month[1:n_train],
nd1=d1[1:n_train], nd2=d2[1:n_train], nTime=Time[1:n_train],
n_int1=int1[1:n_train], n_int2=int2[1:n_train]), interval="prediction")

# Holdout sample predictions

ds_pred <- predict(ds_mod, data.frame(n_month=Month[(n_train+1):335],
nd1=d1[(n_train+1):335], nd2=d2[(n_train+1):335], nTime=Time[(n_train+1):335],
n_int1=int1[(n_train+1):335], n_int2=int2[(n_train+1):335]),
interval="prediction")
```

```{r}
# Accuracy metrics on the training set

mape(p[1:n_train], t_pred[,1])
mae(p[1:n_train], t_pred[,1])
rmse(p[1:n_train], t_pred[,1])

```

```{r}
# Accuracy metrics on the holdout set

mape(p[(n_train+1):335], ds_pred[,1])
mae(p[(n_train+1):335], ds_pred[,1])
rmse(p[(n_train+1):335], ds_pred[,1])
```




      ##### Cyclical Trend Model #####

```{r}
install.packages("TSA")
```

```{r}
library("TSA")


```


```{r}
# Periodogram for electricity price over time

mod0 <- lm(p~Time)
~~~

~~~
pgram <- periodogram(mod0$residuals[1:length(nTime)])
```

```{r}
period <- 1/pgram$freq # Period values
freq <- pgram$freq # Frequency
amp <- pgram$spec # Amplitude
harmonic <- c(1/(nTime/2)) # Harmonic number

gram <- cbind(period, harmonic, amp, freq) # Periodogram values

gram <- gram[order(-amp),] # Ordering the greatest amplitudes first
gram
```

```{r}
periodogram(mod0$residuals[1:length(nTime)], col = "blue")   # Frequency x-axis
plot(period, pgram$spec, type="h", col="blue", ylab="Periodogram", lwd=2)    #
Period x-axis
```

```{r}
# Trigonometric function terms based on periodogram values

cos1 <- cos(2*pi*(1/n_train)*Time)
sin1 <- sin(2*pi*(1/n_train)*Time)

cos2 <- cos(2*pi*(2/n_train)*Time)
sin2 <- sin(2*pi*(2/n_train)*Time)

cos3 <- cos(2*pi*(3/n_train)*Time)
sin3 <- sin(2*pi*(3/n_train)*Time)

cos4 <- cos(2*pi*(4/n_train)*Time)
sin4 <- sin(2*pi*(4/n_train)*Time)

cos5 <- cos(2*pi*(5/n_train)*Time)
sin5 <- sin(2*pi*(5/n_train)*Time)

cos10 <- cos(2*pi*(10/n_train)*Time)
sin10 <- sin(2*pi*(10/n_train)*Time)

cos25 <- cos(2*pi*(25/n_train)*Time)
sin25 <- sin(2*pi*(25/n_train)*Time)

cos50 <- cos(2*pi*(50/n_train)*Time)
sin50 <- sin(2*pi*(50/n_train)*Time)
```

```{r}
# Training sample for each term

ncos1 <- cos1[1:n_train]
nsin1 <- sin1[1:n_train]

ncos2 <- cos2[1:n_train]
nsin2 <- sin2[1:n_train]
~~~

~~~
ncos3 <- cos3[1:n_train]
nsin3 <- sin3[1:n_train]

ncos4 <- cos4[1:n_train]
nsin4 <- sin4[1:n_train]

ncos5 <- cos5[1:n_train]
nsin5 <- sin5[1:n_train]

ncos10 <- cos10[1:n_train]
nsin10 <- sin10[1:n_train]

ncos25 <- cos25[1:n_train]
nsin25 <- sin25[1:n_train]

ncos50 <- cos50[1:n_train]
nsin50 <- sin50[1:n_train]

```

```{r}
tr <- p[1:n_train] # Training set values

cy_var <- cbind(Time, cos1, sin1, cos2, sin2, cos3, sin3, cos4, sin4, cos5,
sin5, cos10, sin10, cos25, sin25, cos50, sin50)

xtr <- cy_var[1:n_train,]
```

```{r}
# Fitting the cyclical trend model

cy_est <-
lm(p[1:n_train]~nTime+ncos1+nsin1+ncos2+nsin2+ncos3+nsin3+ncos4+nsin4+ncos5+nsi
n5+ncos10+nsin10+ncos25+nsin25+ncos50+nsin50)
summary(cy_est)
```

```{r}
# Training sample predictions

cy_p <- predict(cy_est, data.frame(nTime=nTime[1:n_train],
ncos1=ncos1[1:n_train], nsin1=nsin1[1:n_train], ncos2=ncos2[1:n_train],
nsin2=nsin2[1:n_train], ncos3=ncos3[1:n_train], nsin3=nsin3[1:n_train],
ncos4=ncos4[1:n_train], nsin4=nsin4[1:n_train], ncos5=ncos5[1:n_train],
nsin5=nsin5[1:n_train], ncos10=ncos10[1:n_train], nsin10=nsin10[1:n_train],
ncos25=ncos25[1:n_train], nsin25=nsin25[1:n_train], ncos50=ncos50[1:n_train],
nsin50=nsin50[1:n_train]), interval="prediction")


# Holdout sample predictions

cy_p_hold <- predict(cy_est, data.frame(nTime=Time[(n_train+1):335],
ncos1=cos1[(n_train+1):335], nsin1=sin1[(n_train+1):335],
ncos2=cos2[(n_train+1):335], nsin2=sin2[(n_train+1):335],
ncos3=cos3[(n_train+1):335], nsin3=sin3[(n_train+1):335],
~~~

~~~
ncos4=cos4[(n_train+1):335], nsin4=sin4[(n_train+1):335],
ncos5=cos5[(n_train+1):335], nsin5=sin5[(n_train+1):335],
ncos10=cos10[(n_train+1):335], nsin10=sin10[(n_train+1):335],
ncos25=cos25[(n_train+1):335], nsin25=sin25[(n_train+1):335],
ncos50=cos50[(n_train+1):335], nsin50=sin50[(n_train+1):335]),
interval="prediction")
```

```{r}
# Plot of the actual vs predicted monthly electricity prices on the
# first 300 rows of data.

plot.ts(p, main= "Actual vs Predicted Monthly US Electricity Price: January
1990 to November 2017", col = "blue")
lines(predict(cy_est), col = "red", lwd=2)
```

```{r}
# Model residuals ACF and PACF

acf(residuals(cy_est), lag=60)
pacf(residuals(cy_est), lag=60)
```

```{r}
# Train sample metrics

mape(p[1:n_train], cy_p[,1])
mae(p[1:n_train], cy_p[,1])
rmse(p[1:n_train], cy_p[,1])
```

```{r}
# Holdout sample metrics

mape(p[(n_train+1):335], cy_p_hold[,1])
mae(p[(n_train+1):335], cy_p_hold[,1])
rmse(p[(n_train+1):335], cy_p_hold[,1])
```


          ##### Corrected Cyclical Trend Model #####

```{r}
# Resulting intercept and model components and signficance

cyc0 <- Arima(tr, order=c(0,0,0), xreg=xtr)
summary(cyc0)
```

```{r}
# Corrected model

cyc1 <- Arima(tr, order=c(1,0,0), seasonal=list(order=c(2,0,0), period=12),
xreg=xtr)
summary(cyc1)
```
~~~

~~~
```{r}
# ACF plot of the corrected model residuals

acf(cyc1$residuals, lag=60)

# Resdual Box-Pierce test

Box.test(residuals(cyc1), lag=60)
```



```{r}
# Holdout sample predictions

hold_cy <- ElecPrice[301:335]
x_h <- cbind(Time, cos1, sin1, cos2, sin2, cos3, sin3, cos4, sin4, cos5, sin5,
cos10, sin10, cos25, sin25, cos50, sin50)

preds_cy <- Arima(ElecPrice, model = cyc1, xreg=x_h)
summary(preds_cy)
```

```{r}
# Accuracy metrics on holdout sample

holdpreds_cy <- fitted(preds_cy)
accuracy(hold_cy, holdpreds_cy[301:335])
```

********If the above returns 0, just use this:
```{r}
# Holdout MAPE (in %)

cy_mape <- mean(abs(hold_cy - holdpreds_cy[301:335])/hold_cy)
print(cy_mape*100)

# Holdout MAE

cy_mae <- mean(abs(hold_cy - holdpreds_cy[301:335]))
print(cy_mae)

# Holdout RMSE

cy_rmse <- sqrt(mean((hold_cy - holdpreds_cy[301:335])**2))
print(cy_rmse)

```



##### 3 - Regression Model #####

```{r}
# Correlation tests between ElecPrice and each of the
~~~

~~~
# other variables

cor.test(ElecPrice, Electricity_Generation)
cor.test(ElecPrice, NGAS)
cor.test(ElecPrice, CPI)
```

```{r}
# Scatterplot for electricity price and electricity generation

plot(ElecPrice, Electricity_Generation, col="blue", main="Electricity Price and
Electricity Generation")


# Scatterplot for electricity price and natural gas imports

plot(ElecPrice, NGAS, col="blue", main="Electricity Price and Natural Gas
Imports")


# Scatterplot for electricity price and consumer price index

plot(ElecPrice, CPI, col="blue", main="Electricity Price and Consumer Price
Index")
```

```{r}
# Subsets of each variable to be used as the training set for the
# regression model.

reg_ElecPrice <- ElecPrice[1:300]
reg_Electricity_Generation <- Electricity_Generation[1:300]
reg_NGAS <- NGAS[1:300]
reg_CPI <- CPI[1:300]
```

```{r}
# Fitting the regression model

reg_est <- lm(reg_ElecPrice~reg_Electricity_Generation+reg_NGAS+reg_CPI)
summary(reg_est)
```

```{r}
# Actual versus predicted values for electricity price: training sample

ts.plot(ElecPrice, lwd=2, main="Actual vs Predicted Monthly US Electricity
Price: January 1990 to November 2017", ylab= "Electricity Price", col="blue")
lines(predict(reg_est), col="red", lwd=2)
```

```{r}
# ACF and PACF for model residuals.

acf(residuals(reg_est))
pacf(residuals(reg_est))
```
~~~

~~~
```{r}
# Training sample predictions
reg_p <- predict(reg_est,
data.frame(reg_Electricity_Generation=reg_Electricity_Generation[1:n_train],
reg_NGAS=reg_NGAS[1:n_train], reg_CPI=reg_CPI[1:n_train]),
interval="prediction")

# Holdout sample predictions
hold_reg <- predict(reg_est,
data.frame(reg_Electricity_Generation=Electricity_Generation[(n_train+1):335],
reg_NGAS=NGAS[(n_train+1):335], reg_CPI=CPI[(n_train+1):335]),
interval="prediction")
```

```{r}
# Train sample metrics

mape(p[1:n_train], reg_p[,1])
mae(p[1:n_train], reg_p[,1])
rmse(p[1:n_train], reg_p[,1])

```

```{r}
# Holdout sample metrics

mape(p[(n_train+1):335], hold_reg[,1])
mae(p[(n_train+1):335], hold_reg[,1])
rmse(p[(n_train+1):335], hold_reg[,1])
```


      ##### Corrected Regression Model #####

```{r}
# Differenced regression model
x <- cbind(reg_Electricity_Generation, reg_NGAS, reg_CPI)
diff_mod <- arima((reg_ElecPrice), order=c(0,1,0),
seasonal=list(order=c(0,1,0), period = 12), xreg=x)
summary(diff_mod)
```


```{r}
# ACF and PACF of differenced residuals:

par(mfrow = c(1,1))
acf(diff_mod$residuals, main="ACF of Differenced Residuals")
pacf(diff_mod$residuals, main="PACF of Differenced Residuals")
```


```{r}
# Corrected model:
~~~

~~~
reg_correct <- arima((reg_ElecPrice), order=c(0,1,0),
seasonal=list(order=c(0,1,1), period = 12), xreg=x)
summary(reg_correct) # Metrics and coefficients
```


```{r}
# ACF of residuals for corrected regression model

par(mfrow = c(1,1))
acf(reg_correct$residuals, main="Corrected Regression Residuals ACF")
Box.test(reg_correct$residuals)
```

```{r}
# Actual vs predicted values from corrected regression model

reg_preds <- fitted(reg_correct)
plot.ts(df$ElecPrice, col="blue", type="b", lwd=2, main="Actual vs Predicted
Values of Electricity Price")
lines(reg_preds, col="red", lwd=1.5)
```

```{r}
# Holdout sample accuracy:

hold_reg <- ElecPrice[301:335]
x_hold <- cbind(Electricity_Generation, NGAS, CPI)

pred_reg <- Arima(ElecPrice, model=reg_correct, xreg=x_hold)
summary(pred_reg)
```

```{r}
# Accuracy for holdout sample

holdfit_reg <- fitted(pred_reg)
accuracy(hold_reg, holdfit_reg[301:335])
```

```{r}
# Holdout MAPE (in %)

reg_mape <- mean(abs(hold_reg - holdfit_reg[301:335])/hold_reg)
print(reg_mape*100)

# Holdout MAE

reg_mae <- mean(abs(hold_reg - holdfit_reg[301:335]))
print(reg_mae)

# Holdout RMSE

reg_rmse <- sqrt(mean((hold_reg - holdfit_reg[301:335])**2))
print(reg_rmse)

```
~~~

~~~
##### 4 - Stochastic Time Series #####
  ##### Overall ARMA Model #####

```{r}
# Unmodified ACF and PACF for ElecPrice

acf(ElecPrice, col="blue", lag=60)
pacf(ElecPrice, col="blue", lag=60)
```


```{r}
# First difference ACF and PACF

par(mfrow=c(1,1))
acf(diff(ElecPrice), col="blue", lag=60)
pacf(diff(ElecPrice), col="blue", lag=60)
```

```{r}
# Log transform ACF and PACF

par(mfrow=c(1,1))
acf((log(ElecPrice)), col="blue", lag=60)
pacf((log(ElecPrice)), col="blue", lag=60)
```

```{r}
# First difference, seasonal difference, and log transform ACF and PACF

par(mfrow=c(1,1))
acf(diff(diff(log(ElecPrice), lag=12)), col="blue", lag=60)
pacf(diff(diff(log(ElecPrice), lag=12)), col="blue", lag=60)
```

```{r}
# Seasonal ARIMA model with first difference, seasonal difference, and log
transformation

sarima_mod <- Arima(ElecPrice, order = c(0, 1, 0),
seasonal=list(order=c(0,1,1), period=12),lambda=0)
summary(sarima_mod)
```


```{r}
# ACF of residuals for seasonal ARIMA model

par(mfrow=c(1,1))
acf(sarima_mod$residuals,col="blue", main="Residuals of the Seasonal Arima
Model")
Box.test(sarima_mod$residuals,lag=36) # Box-Pierce test for white noise
```
~~~

~~~
```{r}
# Actual vs predicted values from SARIMA model

sarima_preds <- fitted(sarima_mod)
plot.ts(df$ElecPrice, col="blue", type="b", lwd=2, main="Actual vs Predicted
Values of Electricity Price")
lines(sarima_preds, col="red", lwd=1.5)
```

```{r}
# Holdout sample model performance

hold <- ElecPrice[301:335]

preds <- Arima(ElecPrice, model = sarima_mod)
summary(preds)
```

```{r}
# Holdout SARIMA model prediction metrics

holdpreds <- fitted(preds)
accuracy(hold, holdpreds[301:335])
```

```{r}
# Holdout MAPE (in %)

h_mape <- mean(abs(hold - holdpreds[301:335])/hold)
print(h_mape*100)

# Holdout MAE

h_mae <- mean(abs(hold - holdpreds[301:335]))
print(h_mae)

# Holdout RMSE

h_rmse <- sqrt(mean((hold_cy - holdpreds[301:335])**2))
print(h_rmse)
~~~