Energy Consumption Time-Series Analysis

Prateek Singh, Sumanth Reddy Thandra

2025-03-31

Data Pre-processing

Loading PJME hourly time-series data

```
setwd("F:/Studies/Master's Notes/Spring 2025/MA 5781 - Time Series Analysis/FP")
data <- read.csv("./data/PJME_hourly.csv", header = TRUE, sep = ",")
head(data)
```

```
##              Datetime PJME_MW
## 1 2002-12-31 01:00:00   26498
## 2 2002-12-31 02:00:00   25147
## 3 2002-12-31 03:00:00   24574
## 4 2002-12-31 04:00:00   24393
## 5 2002-12-31 05:00:00   24860
## 6 2002-12-31 06:00:00   26222
```

```
any(is.na(data$PJME_MW))
```

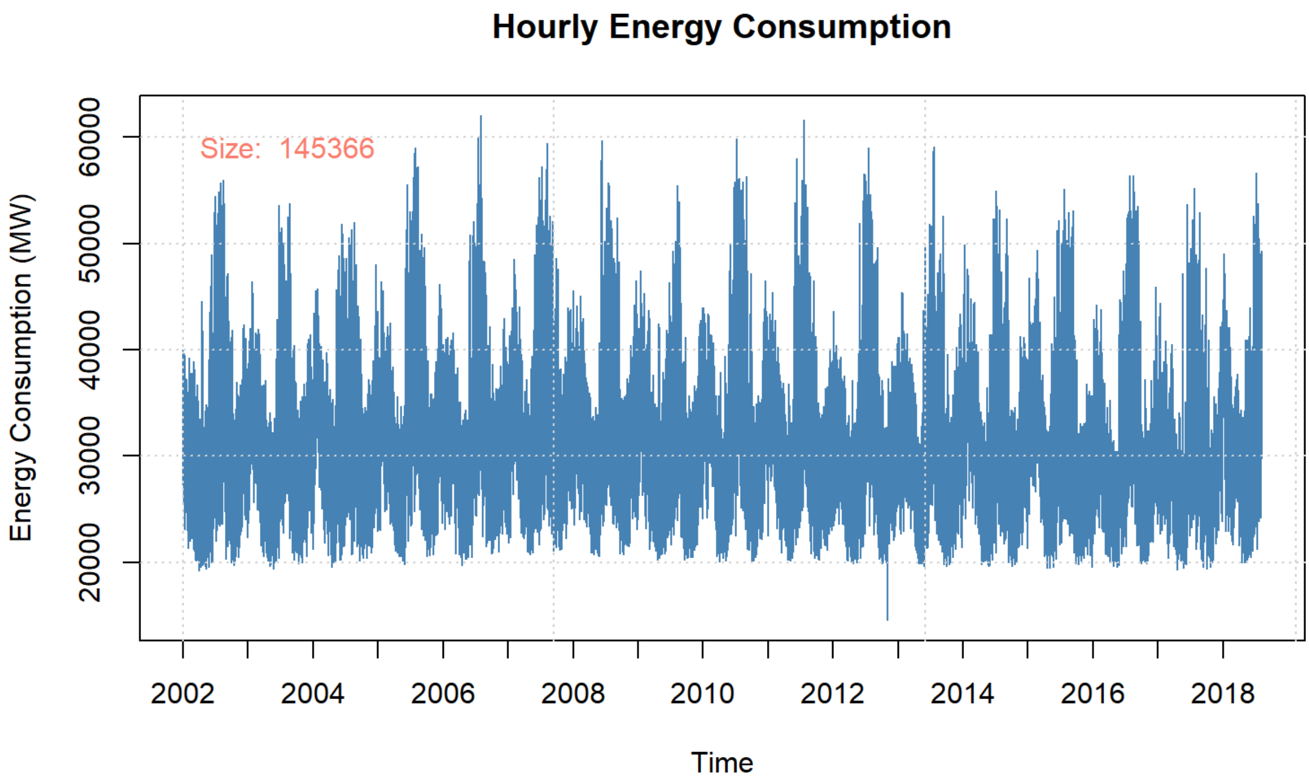
```
## [1] FALSE
```

```
data$Datetime <- as.POSIXct(data$Datetime, tz = "UTC", format = "%Y-%m-%d %H:%M:%S")
data <- data[order(data$Datetime), ]

ts.energy.hourly <- ts(data$PJME_MW)
summary(ts.energy.hourly)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14544   27573   31421   32080   35650   62009
```

Hourly Data and Stationarity Check

```
plot(ts.energy.hourly, main = "Hourly Energy Consumption", xlab = "Time",
     ylab = "Energy Consumption (MW)",col = "steelblue", lwd = 1, xaxt = "n")
axis(1, at = (2002:2018 - 2002) * 365 * 24, labels = 2002:2018, las = 1)
text(x = 0.95, y = max(ts.energy.hourly) * 0.95,
     labels = paste("Size: ", length(ts.energy.hourly)), col = "salmon", pos = 4)
grid()
```




```
adf.test(ts.energy.hourly); kpss.test(ts.energy.hourly); pp.test(ts.energy.hourly);
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts.energy.hourly
## Dickey-Fuller = -24.065, Lag order = 52, p-value = 0.01
## alternative hypothesis: stationary
```

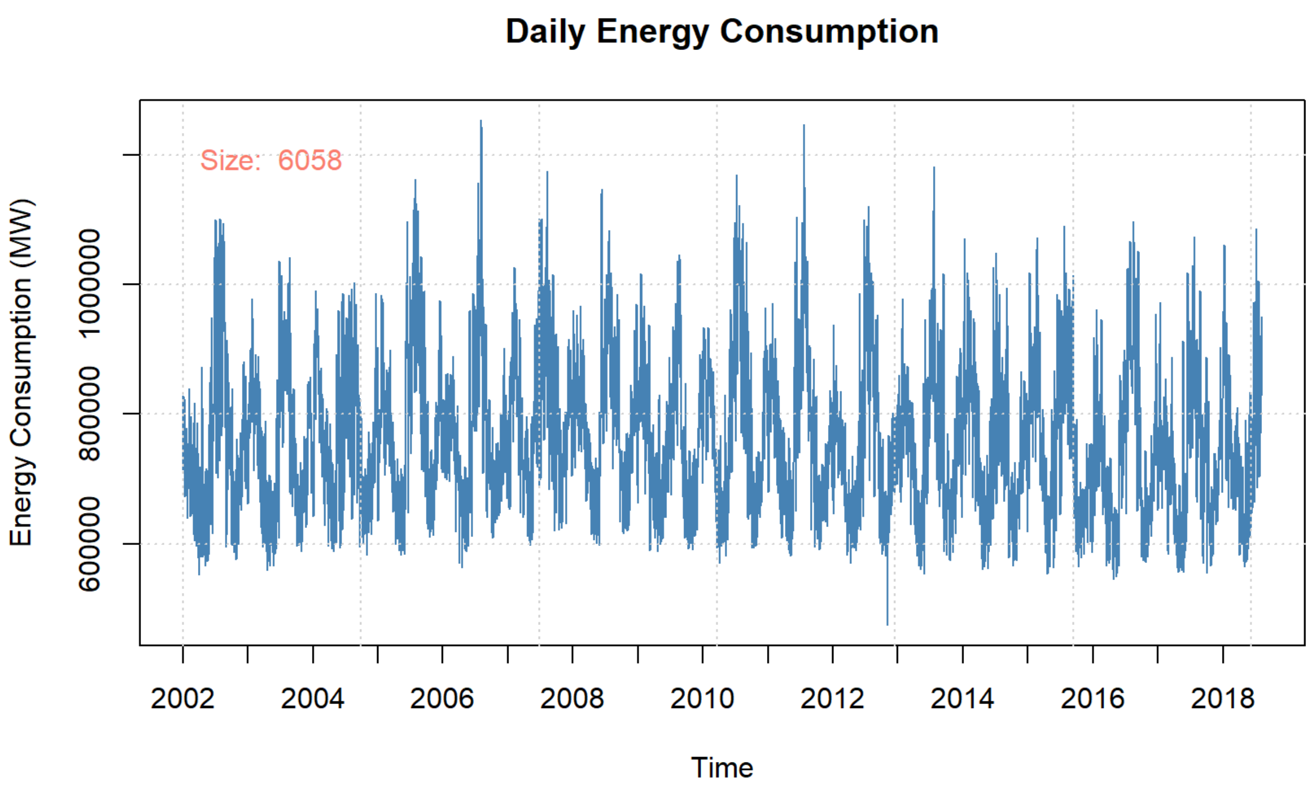
```
##
##  KPSS Test for Level Stationarity
##
## data:  ts.energy.hourly
## KPSS Level = 6.4861, Truncation lag parameter = 24, p-value = 0.01
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  ts.energy.hourly
## Dickey-Fuller Z(alpha) = -1433.6, Truncation lag parameter = 24,
## p-value = 0.01
## alternative hypothesis: stationary
```

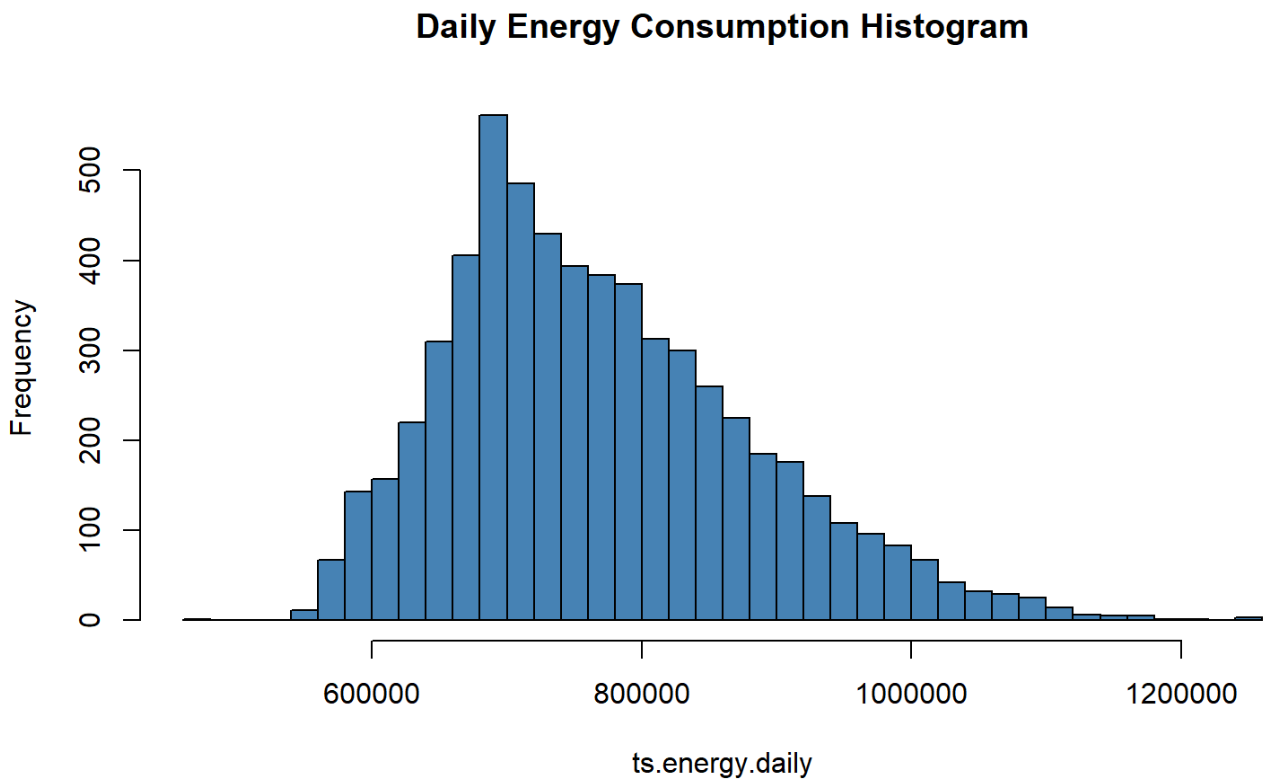
1. **ADF (lag = 52) and PP (lag = 24)** rejects the null → suggest the series is **stationary.**
2. **KPSS (lag = 24)** rejects the null of stationarity → suggests the series is **non-stationary.**
3. **Lag choices (24, 52)** suggest **daily/weekly patterns** are influencing stationarity

Daily Data and Stationarity Check

```
data$Date <- as.Date(data$Datetime)
data.daily <- aggregate(PJME_MW ~ Date, data = data, FUN = sum)
ts.energy.daily <- ts(data.daily[-nrow(data.daily), ]$PJME_MW)
plot(ts.energy.daily, main = "Daily Energy Consumption",xlab = "Time",
     ylab = "Energy Consumption (MW)", col = "steelblue", lwd = 1, xaxt = "n")
axis(1, at = (2002:2018 - 2002) * 365, labels = 2002:2018, las = 1)
text(x = 0.95, y = max(ts.energy.daily) * 0.95,
     labels = paste("Size: ", length(ts.energy.daily)), col = "salmon", pos = 4)
grid()
```




```
hist(ts.energy.daily, main="Daily Energy Consumption Histogram",
     breaks="FD", col="steelblue")
```




```
adf.test(ts.energy.daily); kpss.test(ts.energy.daily); pp.test(ts.energy.daily);
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts.energy.daily
## Dickey-Fuller = -7.656, Lag order = 18, p-value = 0.01
## alternative hypothesis: stationary
```

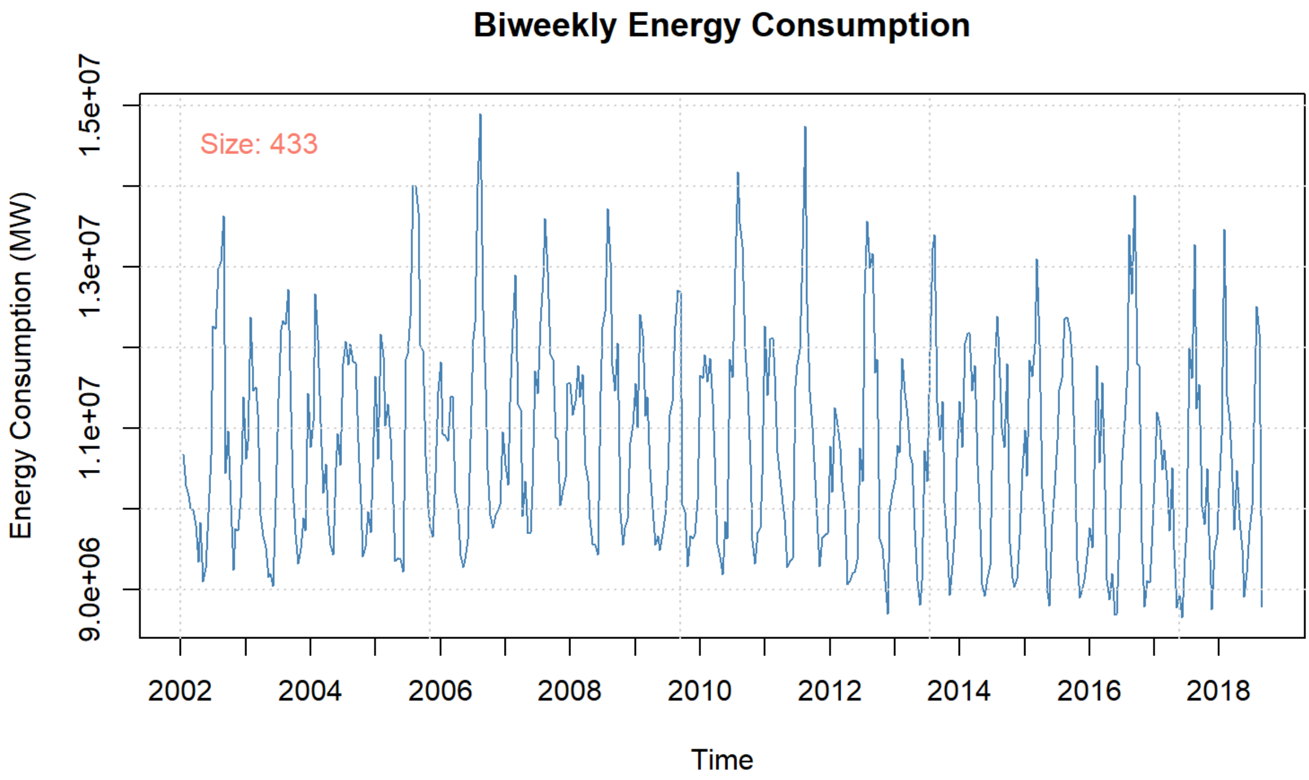
```
##
##  KPSS Test for Level Stationarity
##
## data:  ts.energy.daily
## KPSS Level = 0.83879, Truncation lag parameter = 11, p-value = 0.01
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  ts.energy.daily
## Dickey-Fuller Z(alpha) = -770.67, Truncation lag parameter = 11,
## p-value = 0.01
## alternative hypothesis: stationary
```

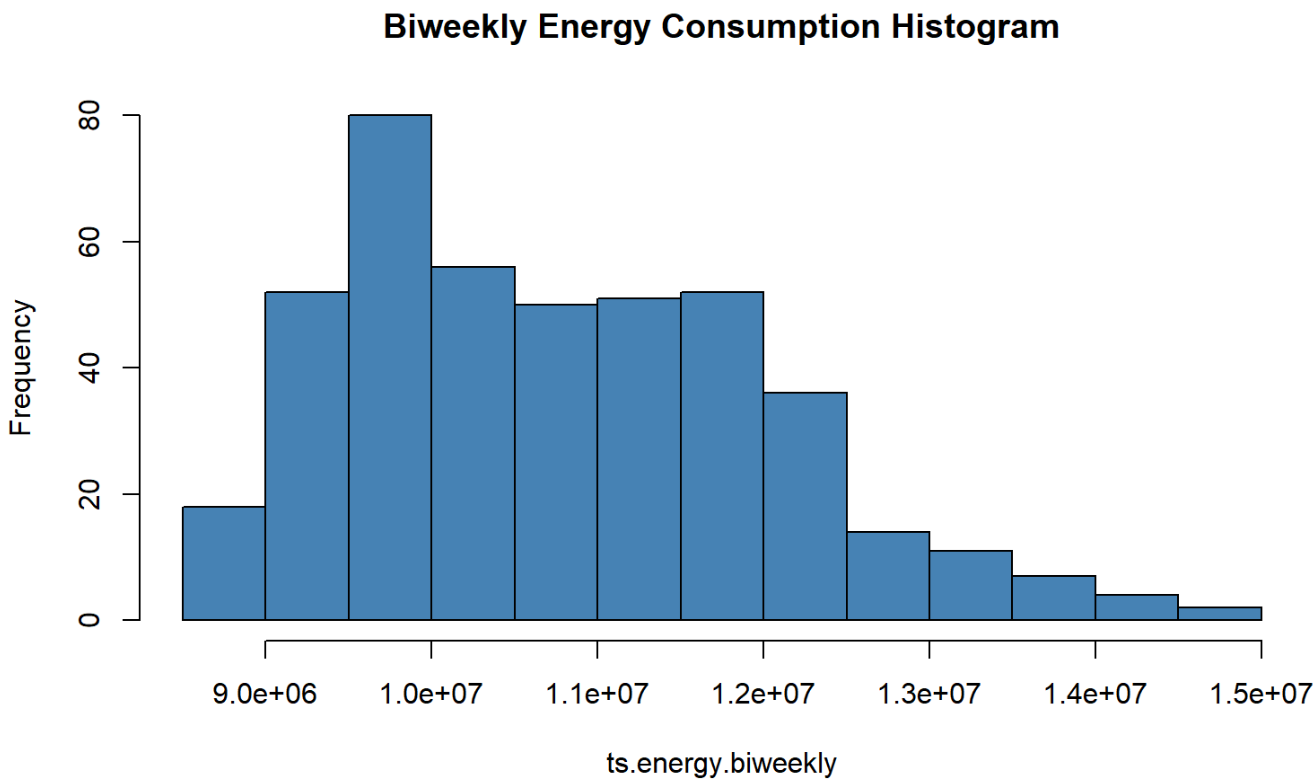
1. **ADF (lag = 18) and PP (lag = 1)** tests reject the null of non-stationarity → suggest the series is **stationary.**
2. **KPSS (lag = 11)** rejects the null of stationarity → suggests the series is **non-stationary.**
3. **Lag 11–18 (days)** suggests the tests are accounting for autocorrelation up to **~2.5 weeks.**

Biweekly Data and Stationarity Check

```
data.daily$Week <- as.integer((as.numeric(data.daily$Date - min(data.daily$Date)))/ 7) + 1
data.daily$Biweekly <- as.integer((as.numeric(data.daily$Date - min(data.daily$Date)))/14) +
1
data.biweekly <- aggregate(PJME_MW ~ Biweekly, data = data.daily, FUN = sum)
ts.energy.biweekly <- ts(data.biweekly$PJME_MW)

plot(ts.energy.biweekly, main = "Biweekly Energy Consumption",
     xlab = "Time", ylab = "Energy Consumption (MW)",
     col = "steelblue", lwd = 1, xaxt = "n")
axis(1, at = (2002:2018 - 2002) * 26, labels = 2002:2018, las = 1)
text(x = 0.95, y = max(ts.energy.biweekly) * 0.975,
     labels = paste("Size:", length(ts.energy.biweekly)), col = "salmon", pos = 4)
grid()
```




```
hist(ts.energy.biweekly, main="Biweekly Energy Consumption Histogram",
     breaks="FD", col="steelblue")
```




```
adf.test(ts.energy.biweekly); kpss.test(ts.energy.biweekly); pp.test(ts.energy.biweekly);
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts.energy.biweekly
## Dickey-Fuller = -9.51, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

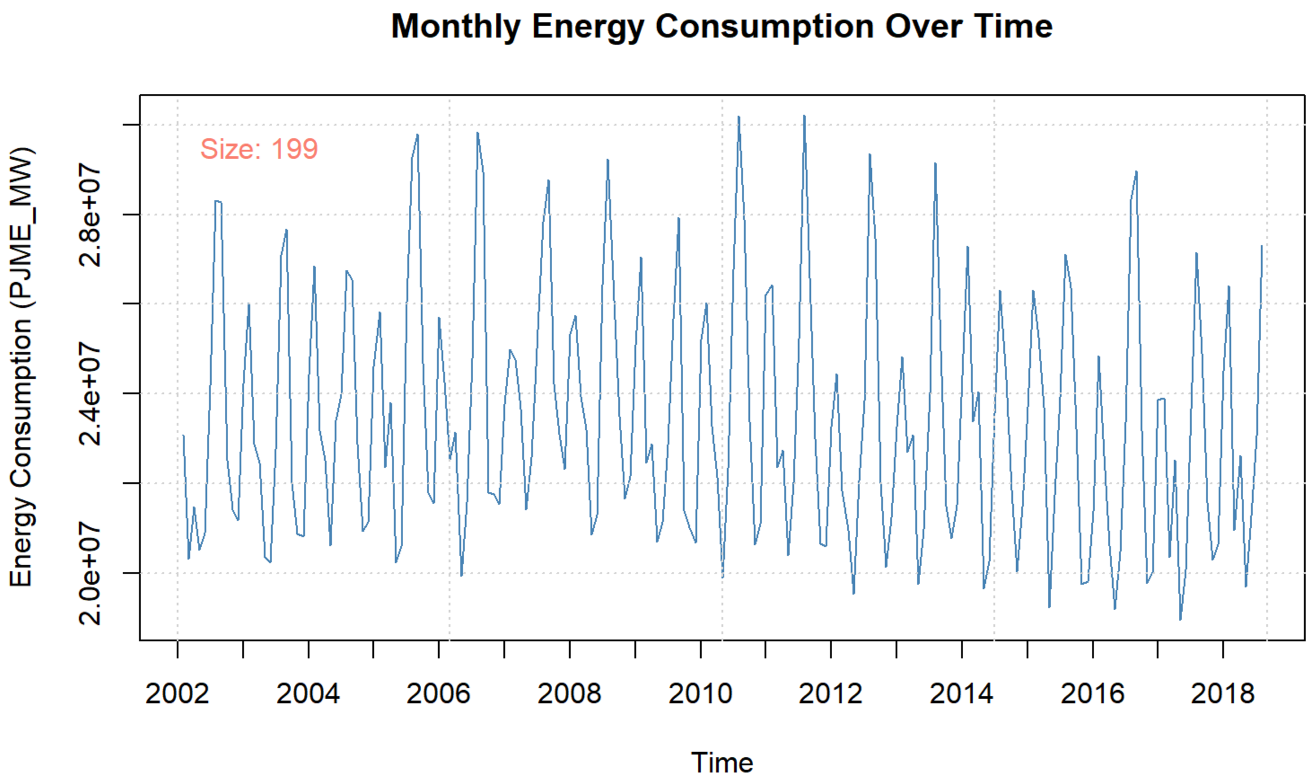
```
##
##  KPSS Test for Level Stationarity
##
## data:  ts.energy.biweekly
## KPSS Level = 0.29161, Truncation lag parameter = 5, p-value = 0.1
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  ts.energy.biweekly
## Dickey-Fuller Z(alpha) = -135.66, Truncation lag parameter = 5, p-value
## = 0.01
## alternative hypothesis: stationary
```

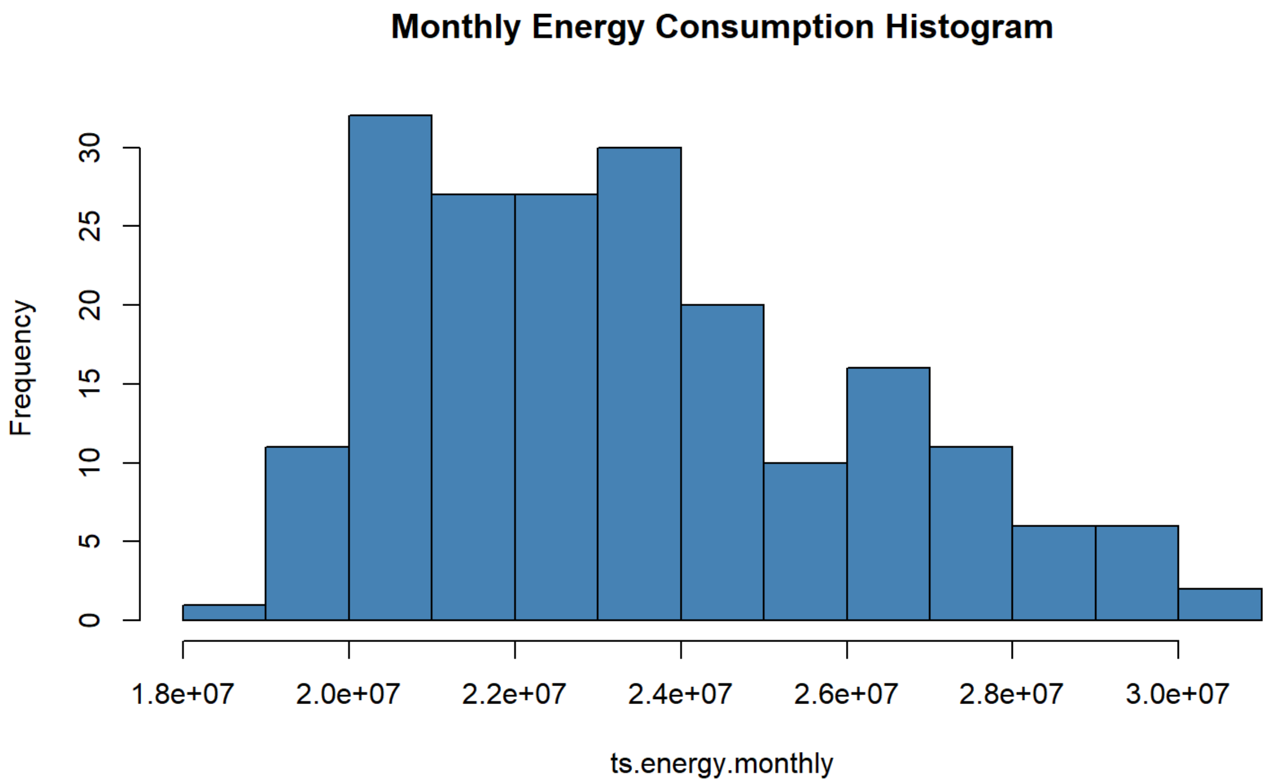
- **ADF & PP Tests (p = 0.01)** → Strong evidence against unit root (data is stationary).
- **KPSS Test (p = 0.1)** → Fails to reject level stationarity (supports stationarity)
- The decomposed plots, show strong seasonality.

Monthly Data and Stationarity Check

```
data$YearMonth <- format(data$Datetime, "%Y-%m")
data.monthly <- aggregate(PJME_MW ~ YearMonth, data, sum)
data.monthly <- data.monthly[-nrow(data.monthly), ]
ts.energy.monthly <- ts(data.monthly$PJME_MW)

plot(ts.energy.monthly, main = "Monthly Energy Consumption Over Time",
     ylab = "Energy Consumption (PJME_MW)", xlab = "Time",
     col = "steelblue", lwd = 1, xaxt = "n")
axis(1, at = (2002:2018 - 2002) * 12, labels = 2002:2018, las = 1)
text(x = 0.95, y = max(ts.energy.monthly) * 0.975,
     labels = paste("Size:", length(ts.energy.monthly)), col = "salmon", pos = 4)
grid()
```




```
hist(ts.energy.monthly, main="Monthly Energy Consumption Histogram",
     breaks = "FD", col="steelblue")
```




```
adf.test(ts.energy.monthly); kpss.test(ts.energy.monthly); pp.test(ts.energy.monthly);
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts.energy.monthly
## Dickey-Fuller = -5.9851, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  ts.energy.monthly
## KPSS Level = 0.4485, Truncation lag parameter = 4, p-value = 0.05625
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  ts.energy.monthly
## Dickey-Fuller Z(alpha) = -64.248, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

- **ADF & PP Tests (p = 0.01)** → Strong evidence against unit root (data is stationary).
- **KPSS Test (p = 0.5625)** → Fails to reject level stationarity (supports stationarity), albeit borderline

Checking for Stochastic Trend

```
adf.test(ts.energy.biweekly); kpss.test(ts.energy.biweekly); pp.test(ts.energy.biweekly);
```

```
## 
##  Augmented Dickey-Fuller Test
## 
## data:  ts.energy.biweekly
## Dickey-Fuller = -9.51, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
## 
##  KPSS Test for Level Stationarity
## 
## data:  ts.energy.biweekly
## KPSS Level = 0.29161, Truncation lag parameter = 5, p-value = 0.1
```

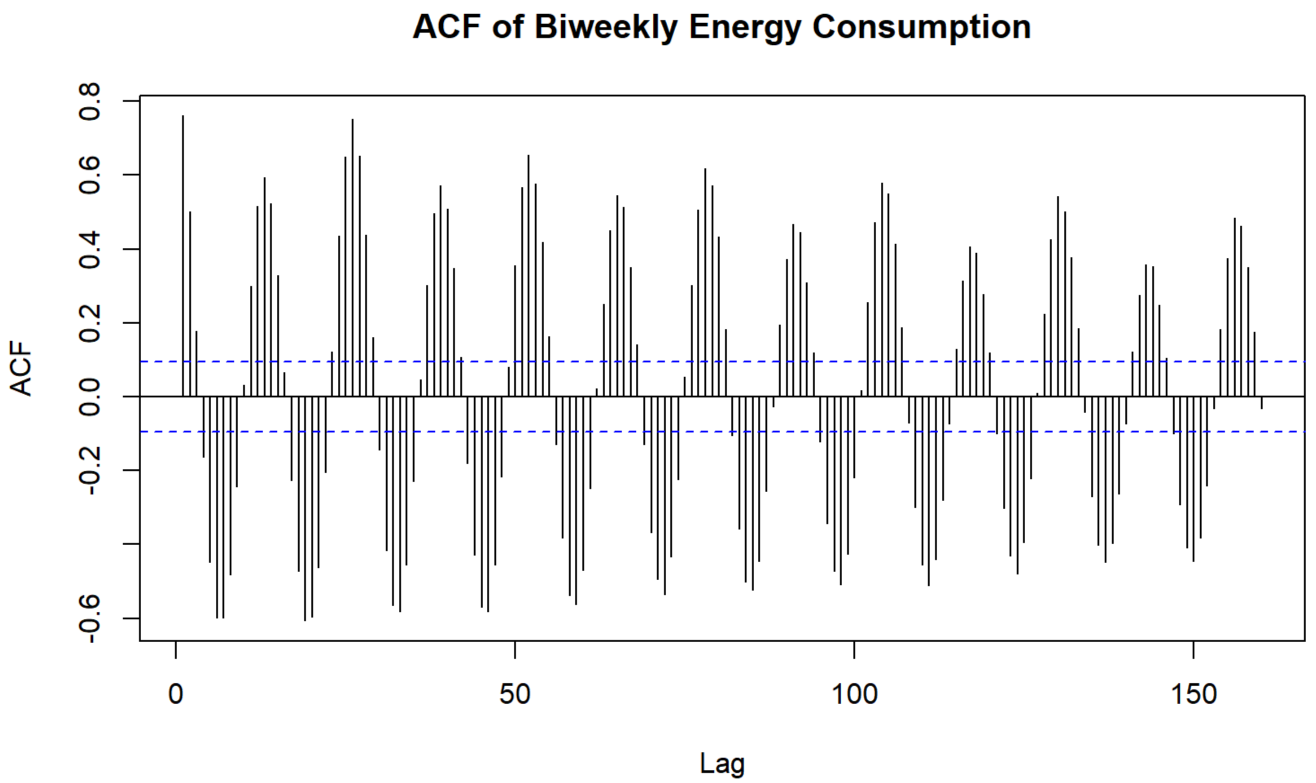
```
## 
##  Phillips-Perron Unit Root Test
## 
## data:  ts.energy.biweekly
## Dickey-Fuller Z(alpha) = -135.66, Truncation lag parameter = 5, p-value
## = 0.01
## alternative hypothesis: stationary
```

Auto-correlation Analysis and choosing d

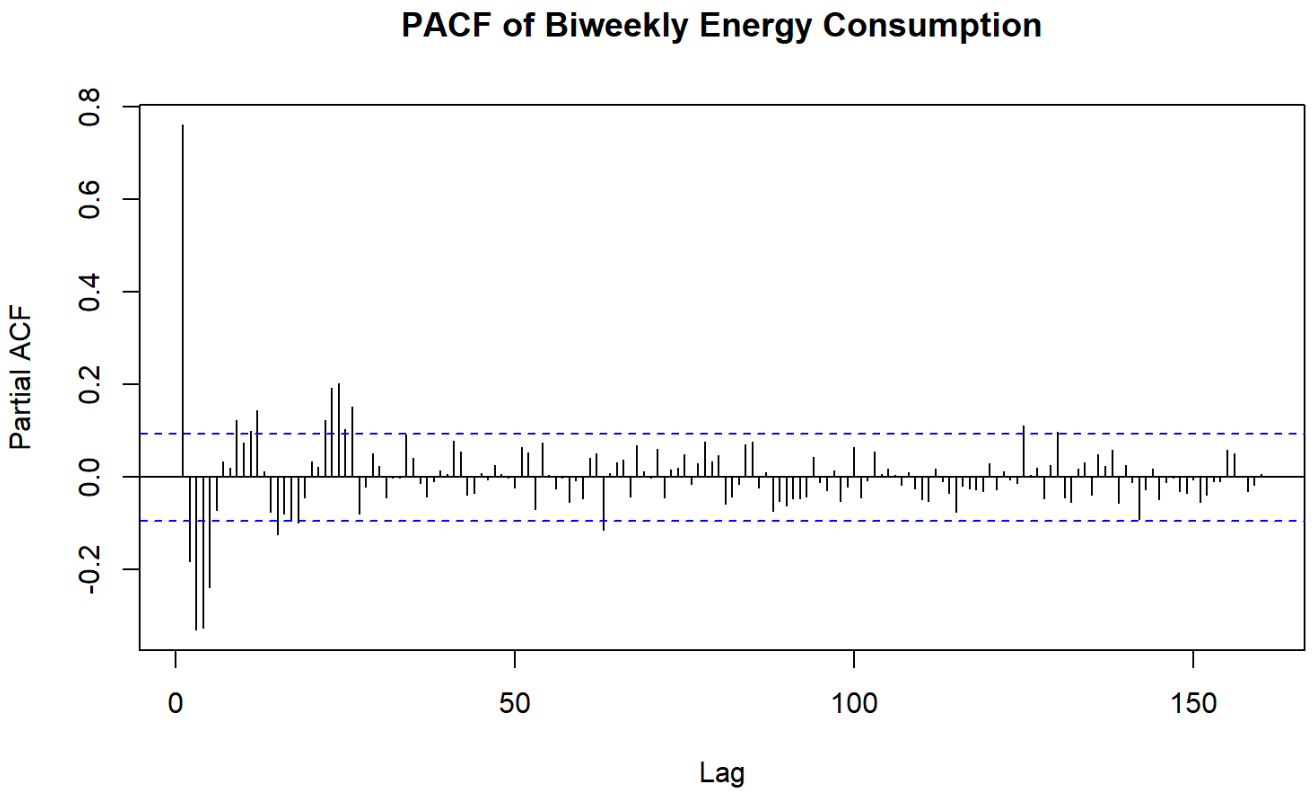
```
eacf(ts.energy.biweekly)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x o  x  x  x 
## 1 x x x x x x x x x x o  x  x  x 
## 2 x x o o o o o o o o o  o  o  o 
## 3 x x o o o o o o o o x  o  o  o 
## 4 x x x o o o o o o o o  o  o  o 
## 5 x x x x o o o o o o x  o  o  o 
## 6 x x o x x x o o o o x  o  o  o 
## 7 x x x x o x o o o o x  o  o  o
```

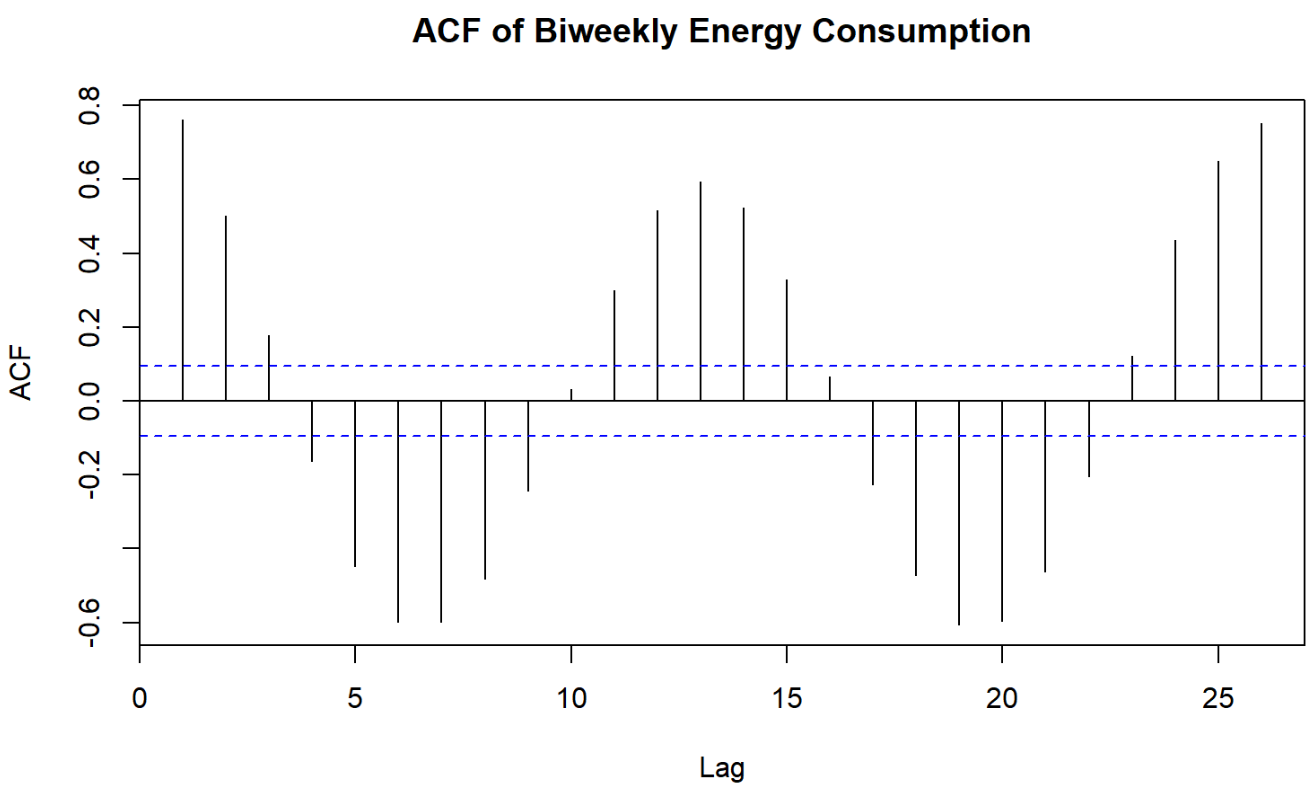
```
acf(ts.energy.biweekly, main = "ACF of Biweekly Energy Consumption", lag.max=160)
```



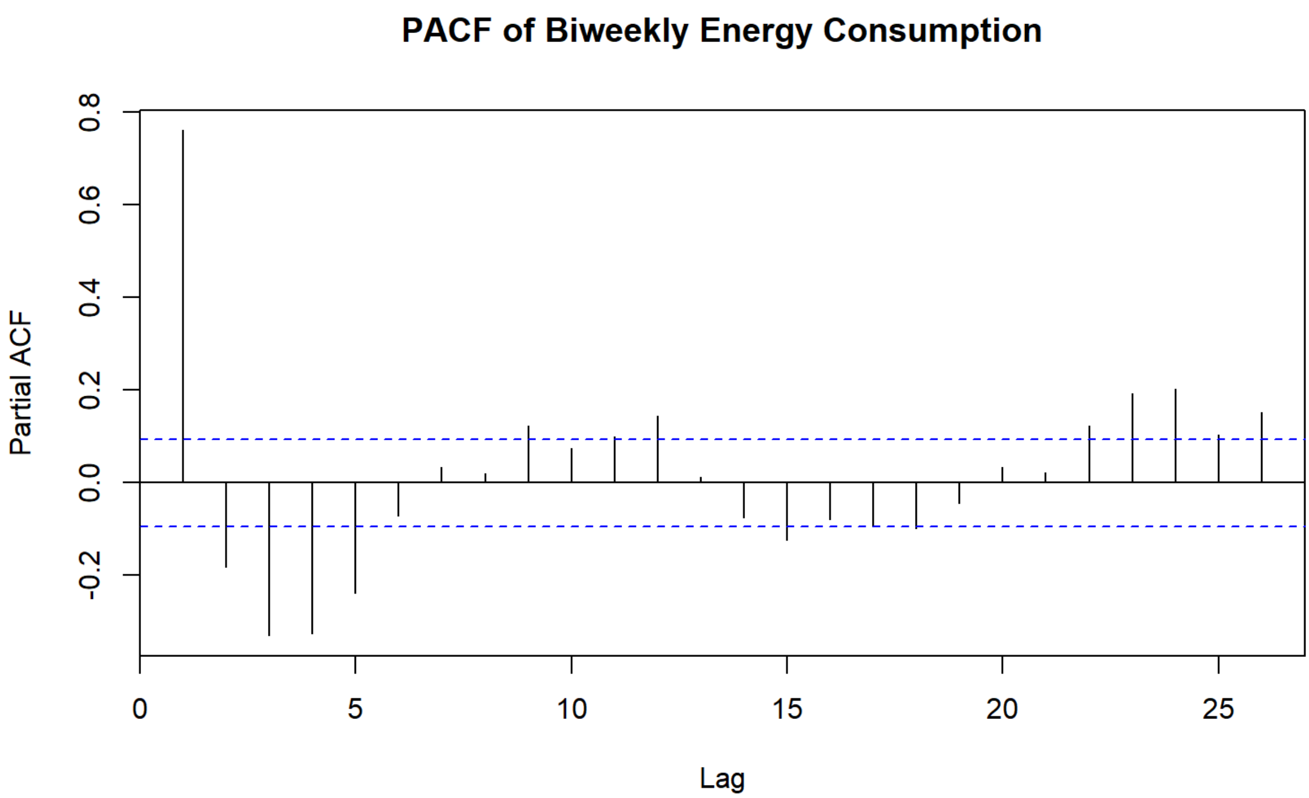

```
pacf(ts.energy.biweekly, main = "PACF of Biweekly Energy Consumption", lag.max=160)
```




```
acf(ts.energy.biweekly, main = "ACF of Biweekly Energy Consumption")
```




```
pacf(ts.energy.biweekly, main = "PACF of Biweekly Energy Consumption")
```



1. **For Non-Seasonal part**
    1. **ACF (decay with no sharp cutoff)** → MA(1), MA(2)
    2. **PACF (sharp cutoff at Lag 1)**→ AR(1)
    3. **From EACF** →
        1. **AR(2)**: Most lags become insignificant (o) after Lag 1 → **AR(2) may be adequate**.

        2. **MA(2)**: Lags beyond 2 become insignificant → **MA(2) likely sufficient**.
2. **For Seasonal part**
    1. **ACF (decays slowly with not sharp cutoff till lag 14)** → Differencing might be needed
    2. **PACF (Cuts off after lag 1) → AR(1) may be adequate**

Candidate modelling (Based on ACF, PACF, EACF)

ARMA models

1. ARIMA(1,0,0)
2. ARIMA(1,0,1)
3. ARIMA(1,0,2)
4. ARIMA(2,0,0)
5. ARIMA(2,0,1)
6. ARIMA(2,0,2)
7. Auto ARIMA (Selected using `auto.arima`)

```
# Dropping last aggregation (incompletete data)
ts.energy.biweekly <- ts.energy.biweekly[-length(ts.energy.biweekly)]

model.arima_1 <- arima(ts.energy.biweekly, order = c(1, 0, 0))
model.arima_2 <- arima(ts.energy.biweekly, order = c(1, 0, 1))
model.arima_3 <- arima(ts.energy.biweekly, order = c(1, 0, 2))
model.arima_4 <- arima(ts.energy.biweekly, order = c(2, 0, 0))
model.arima_5 <- arima(ts.energy.biweekly, order = c(2, 0, 1))
model.arima_6 <- arima(ts.energy.biweekly, order = c(2, 0, 2))
model.arima_ns_auto <- auto.arima(ts.energy.biweekly)

summary(model.arima_ns_auto)
```

```
## Series: ts.energy.biweekly
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2         mean
##       1.7124  -0.9274  -1.1413  0.4904  10778043.86
## s.e.  0.0225   0.0218   0.0611  0.0514     52544.37
##
## sigma^2 = 4.549e+11:  log likelihood = -6409.8
## AIC=12831.61   AICc=12831.81   BIC=12856.02
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE     MASE
## Training set -232.7162 670572.5 526771.1 -0.3685558 4.836914 0.807866
##                    ACF1
## Training set 0.05080482
```

```r
results <- data.frame(
  Model = c("ARMA(1,0,0)", "ARMA(1,0,1)", "ARMA(1,0,2)",
            "ARMA(2,0,0)", "ARMA(2,0,1)", "ARMA(2,0,2)", "Auto.ARIMA"),
  AIC = c(AIC(model.arima_1), AIC(model.arima_2), AIC(model.arima_3),
          AIC(model.arima_4), AIC(model.arima_5), AIC(model.arima_6),
          AIC(model.arima_ns_auto)),
  BIC = c(BIC(model.arima_1), BIC(model.arima_2), BIC(model.arima_3),
          BIC(model.arima_4), BIC(model.arima_5), BIC(model.arima_6),
          BIC(model.arima_ns_auto))
)
results[order(results$AIC), ]
```

```
##         Model      AIC      BIC
## 6 ARMA(2,0,2) 12831.61 12856.02
## 7  Auto.ARIMA 12831.61 12856.02
## 5 ARMA(2,0,1) 12890.38 12910.72
## 3 ARMA(1,0,2) 12955.42 12975.76
## 4 ARMA(2,0,0) 12968.22 12984.49
## 2 ARMA(1,0,1) 12976.12 12992.40
## 1 ARMA(1,0,0) 12983.49 12995.70
```

```r
results[order(results$BIC), ]
```

```
##         Model      AIC      BIC
## 6 ARMA(2,0,2) 12831.61 12856.02
## 7  Auto.ARIMA 12831.61 12856.02
## 5 ARMA(2,0,1) 12890.38 12910.72
## 3 ARMA(1,0,2) 12955.42 12975.76
## 4 ARMA(2,0,0) 12968.22 12984.49
## 2 ARMA(1,0,1) 12976.12 12992.40
## 1 ARMA(1,0,0) 12983.49 12995.70
```

Best Performing Non-Seasonal Components

1. ARMA(2,0,2)
2. ARMA(1,0,2)
3. ARMA(1,0,2)

SARIMA models

1. SARIMA(2,0,2)(0,1,1)
2. SARIMA(2,0,2)(1,1,1)
3. SARIMA(2,0,2)(1,1,2)
4. SARIMA(2,0,1)(0,1,1)
5. SARIMA(2,0,1)(1,1,1)
6. SARIMA(2,0,1)(1,1,2)
7. Auto SARIMA (Selected using `auto.arima`)

```r
ts.energy.biweekly <- ts(ts.energy.biweekly, start = c(2002, 1), frequency = 26)

model.sarima_1 <- arima(ts.energy.biweekly, order = c(2, 0, 2), seasonal = c(0, 1, 1))
model.sarima_2 <- arima(ts.energy.biweekly, order = c(2, 0, 2), seasonal = c(1, 1, 1))
model.sarima_3 <- arima(ts.energy.biweekly, order = c(2, 0, 2), seasonal = c(1, 1, 2))
model.sarima_4 <- arima(ts.energy.biweekly, order = c(2, 0, 1), seasonal = c(0, 1, 1))
model.sarima_5 <- arima(ts.energy.biweekly, order = c(2, 0, 1), seasonal = c(1, 1, 1))
model.sarima_6 <- arima(ts.energy.biweekly, order = c(2, 0, 1), seasonal = c(1, 1, 2))

model.sarima_auto <- auto.arima(ts.energy.biweekly)
summary(model.sarima_auto)
```

```
## Series: ts.energy.biweekly
## ARIMA(1,0,1)(2,1,0)[26]
##
## Coefficients:
##          ar1      ma1     sar1     sar2
##       0.5979  -0.2064  -0.5670  -0.3122
## s.e.  0.0955   0.1185   0.0512   0.0515
##
## sigma^2 = 4.068e+11:  log likelihood = -6006.07
## AIC=12022.14   AICc=12022.29   BIC=12042.17
##
## Training set error measures:
##                     ME     RMSE    MAE        MPE     MAPE      MASE       ACF1
## Training set -9641.886 615293.6 452939 -0.3225825 4.108352 0.7498197 0.00237244
```

```r
results_sarima <- data.frame(
  Model = c("SARIMA(2,0,2)(0,1,1)", "SARIMA(2,0,2)(1,1,1)", "SARIMA(2,0,2)(1,1,2)",
            "SARIMA(2,0,1)(0,1,1)", "SARIMA(2,0,1)(1,1,1)", "SARIMA(2,0,1)(1,1,2)",
            "Auto.ARIMA"),
  AIC = c(AIC(model.sarima_1), AIC(model.sarima_2), AIC(model.sarima_3),
          AIC(model.sarima_4), AIC(model.sarima_5), AIC(model.sarima_6),
          AIC(model.sarima_auto)),
  BIC = c(BIC(model.sarima_1), BIC(model.sarima_2), BIC(model.sarima_3),
          BIC(model.sarima_4), BIC(model.sarima_5), BIC(model.sarima_6),
          BIC(model.sarima_auto))
)
results_sarima[order(results_sarima$AIC), ]
```

```
##                  Model      AIC      BIC
## 2 SARIMA(2,0,2)(1,1,1) 11982.01 12010.05
## 3 SARIMA(2,0,2)(1,1,2) 11982.73 12014.79
## 4 SARIMA(2,0,1)(0,1,1) 11983.02 12003.05
## 5 SARIMA(2,0,1)(1,1,1) 11983.71 12007.75
## 6 SARIMA(2,0,1)(1,1,2) 11984.29 12012.33
## 1 SARIMA(2,0,2)(0,1,1) 11984.88 12008.92
## 7           Auto.ARIMA 12022.14 12042.17
```

```r
results_sarima[order(results_sarima$BIC), ]
```

```
##                     Model      AIC      BIC
## 4 SARIMA(2,0,1)(0,1,1) 11983.02 12003.05
## 5 SARIMA(2,0,1)(1,1,1) 11983.71 12007.75
## 1 SARIMA(2,0,2)(0,1,1) 11984.88 12008.92
## 2 SARIMA(2,0,2)(1,1,1) 11982.01 12010.05
## 6 SARIMA(2,0,1)(1,1,2) 11984.29 12012.33
## 3 SARIMA(2,0,2)(1,1,2) 11982.73 12014.79
## 7           Auto.ARIMA 12022.14 12042.17
```

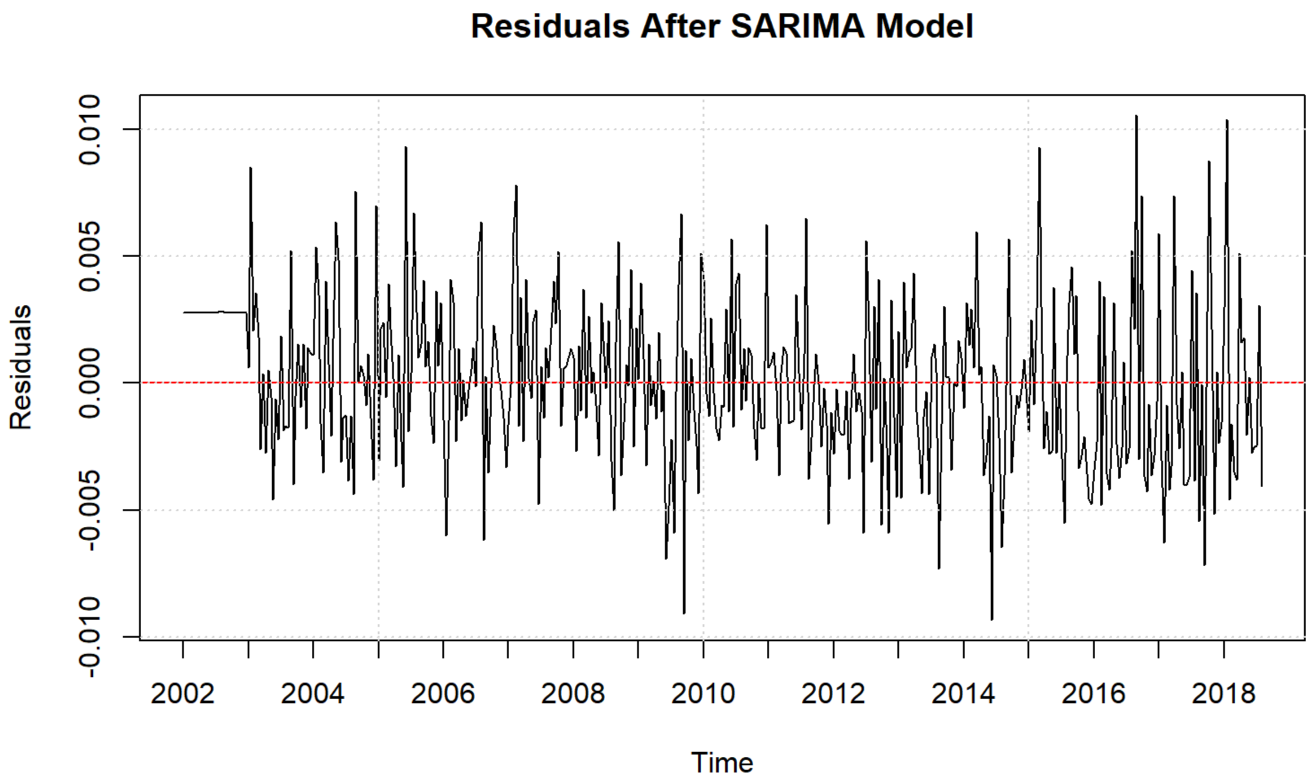
Best Performing Seasonal Models

1. SARIMA(2,0,1)[0,1,1]
2. SARIMA(2,0,2)[1,1,1]
3. SARIMA(2,0,1)(1,1,1)

Model Diagnostic: SARIMA(2,0,1)(0,1,1)[26]

Double log for transformation

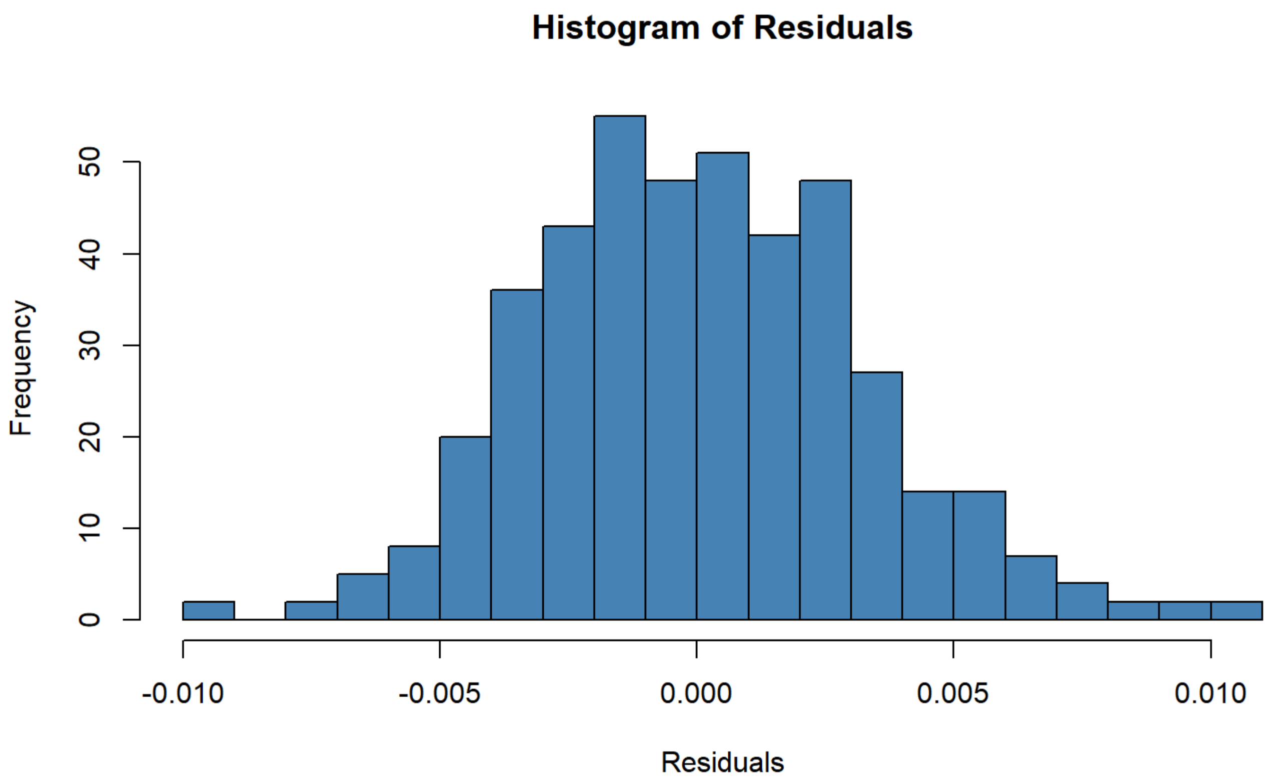
```
model.sarima <- arima(log(log(ts.energy.biweekly)),
                      order = c(2, 0, 1), seasonal = c(0, 1, 1))

residuals <- resid(model.sarima)
plot(residuals, main = "Residuals After SARIMA Model",
     ylab = "Residuals", xlab = "Time", xaxt="n")
axis(1, at = t <- time(ts.energy.biweekly)[d <- !duplicated(y <- floor(time(ts.energy.biweekl
y)))],
     labels = y[d], las = 1)
abline(h = 0, col = "red", lty = 1, lwd = 1)
grid()
```



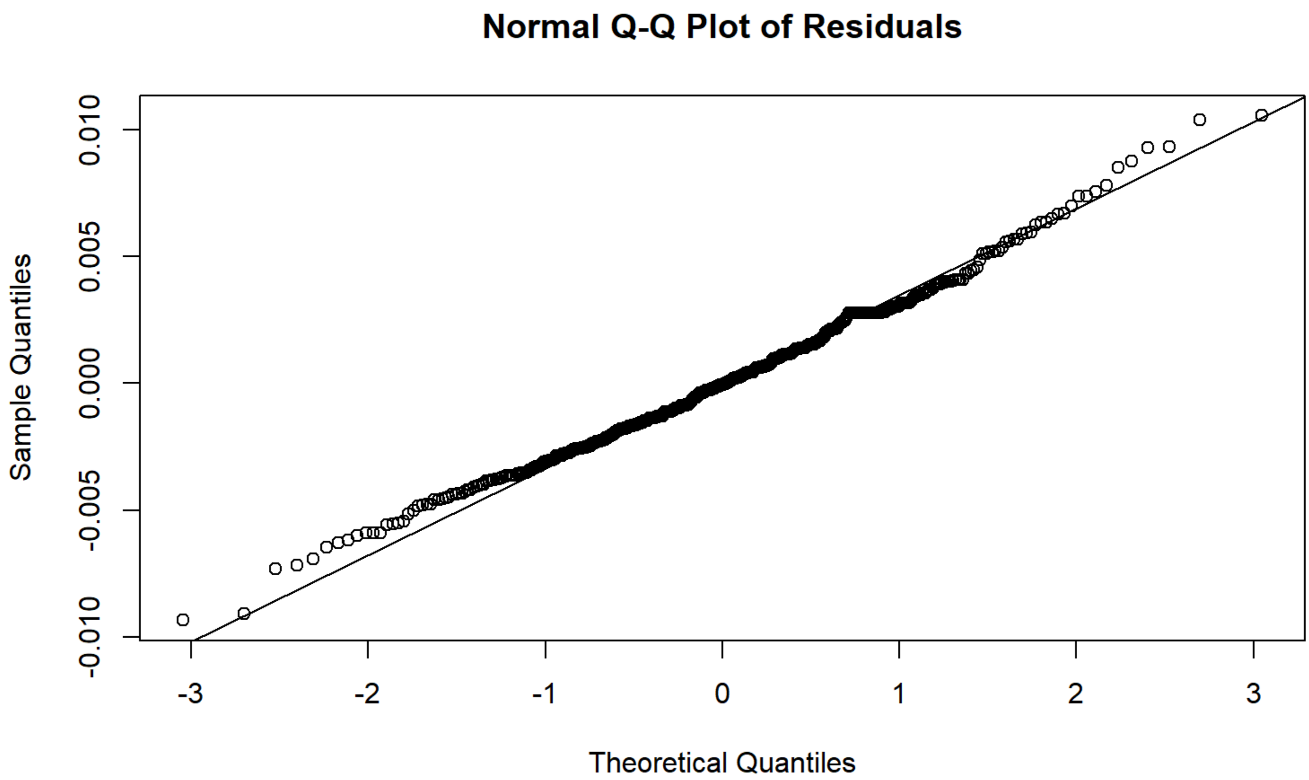
```
shapiro.test(residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.99304, p-value = 0.04306
```

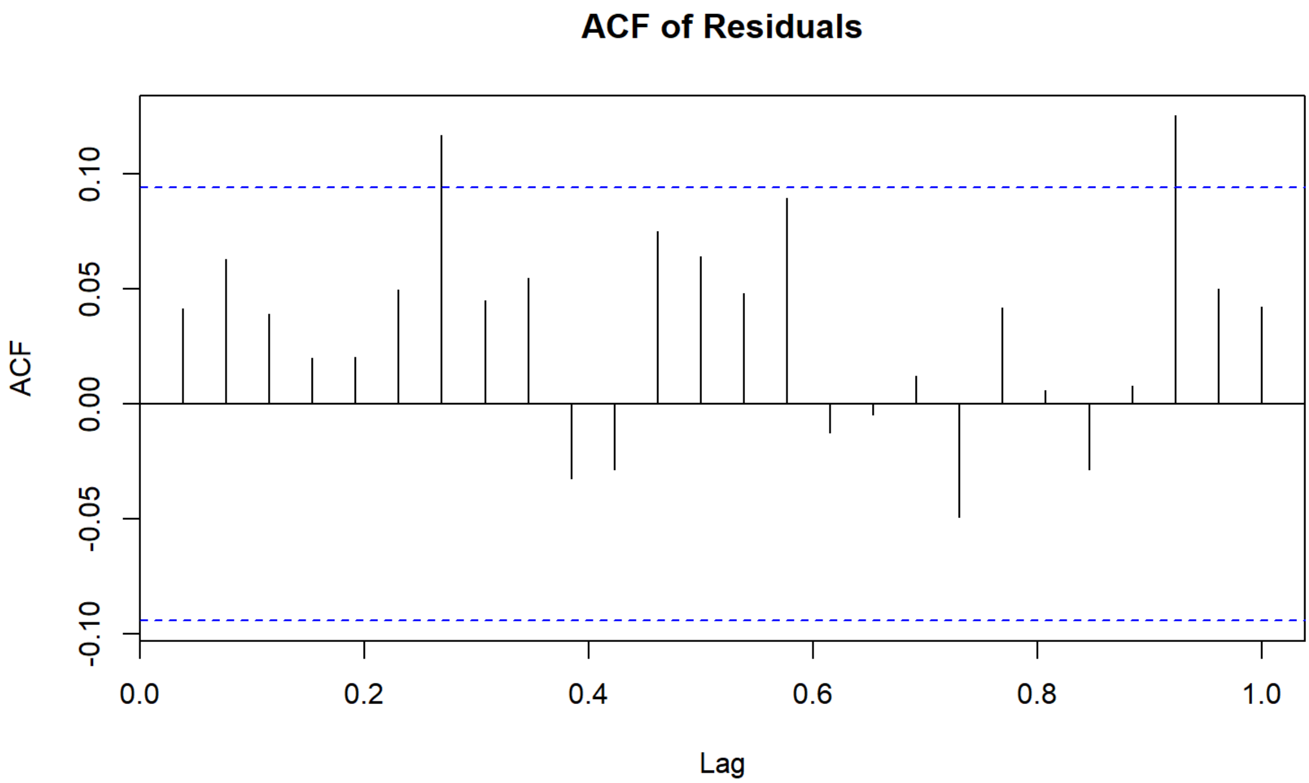
```
hist(residuals, main = "Histogram of Residuals",
     col="steelblue", xlab = "Residuals", breaks = "FD")
```



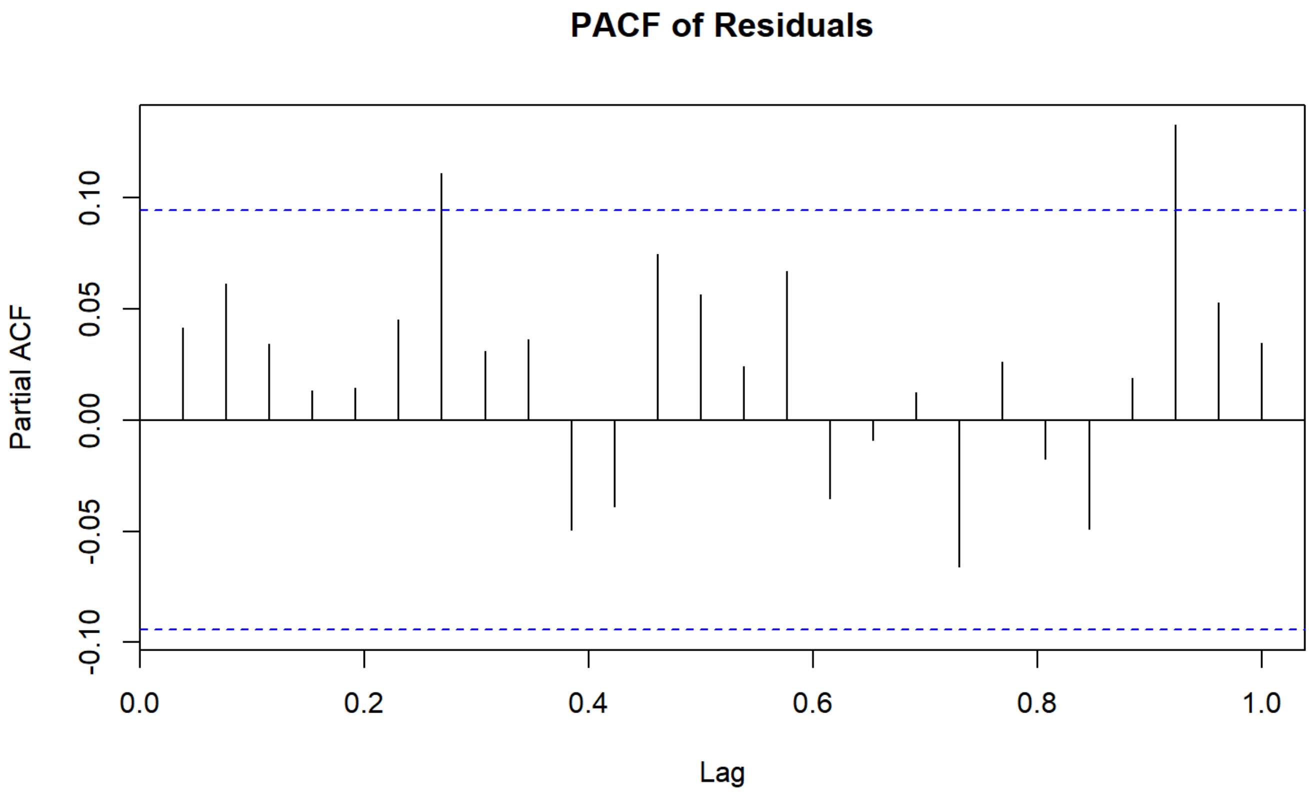

```
qqnorm(residuals, main = "Normal Q-Q Plot of Residuals"); qqline(residuals)
```




```
acf(residuals, main = "ACF of Residuals")
```



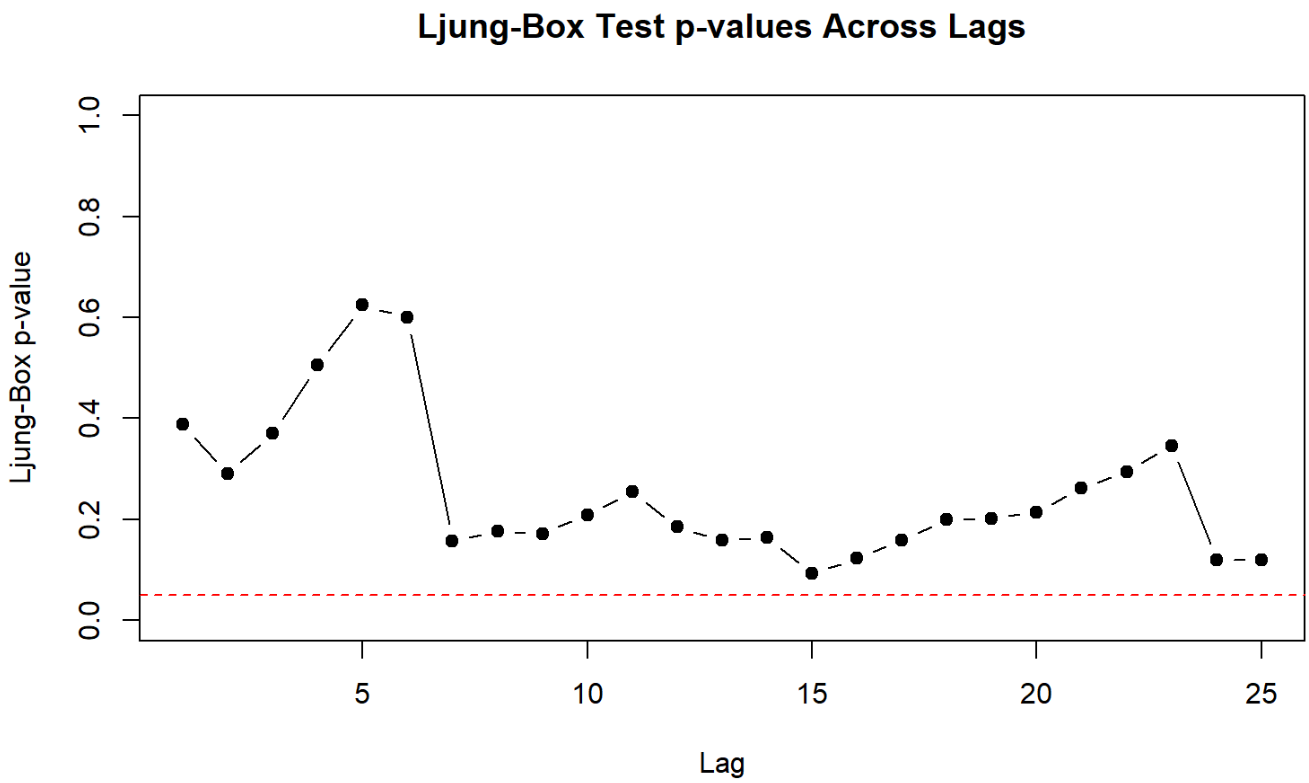

```
pacf(residuals, main = "PACF of Residuals");
```




```r
Box.test(residuals, lag = 20, type = "Box-Pierce")
```

```
##
##  Box-Pierce test
##
## data:  residuals
## X-squared = 23.981, df = 20, p-value = 0.2432
```

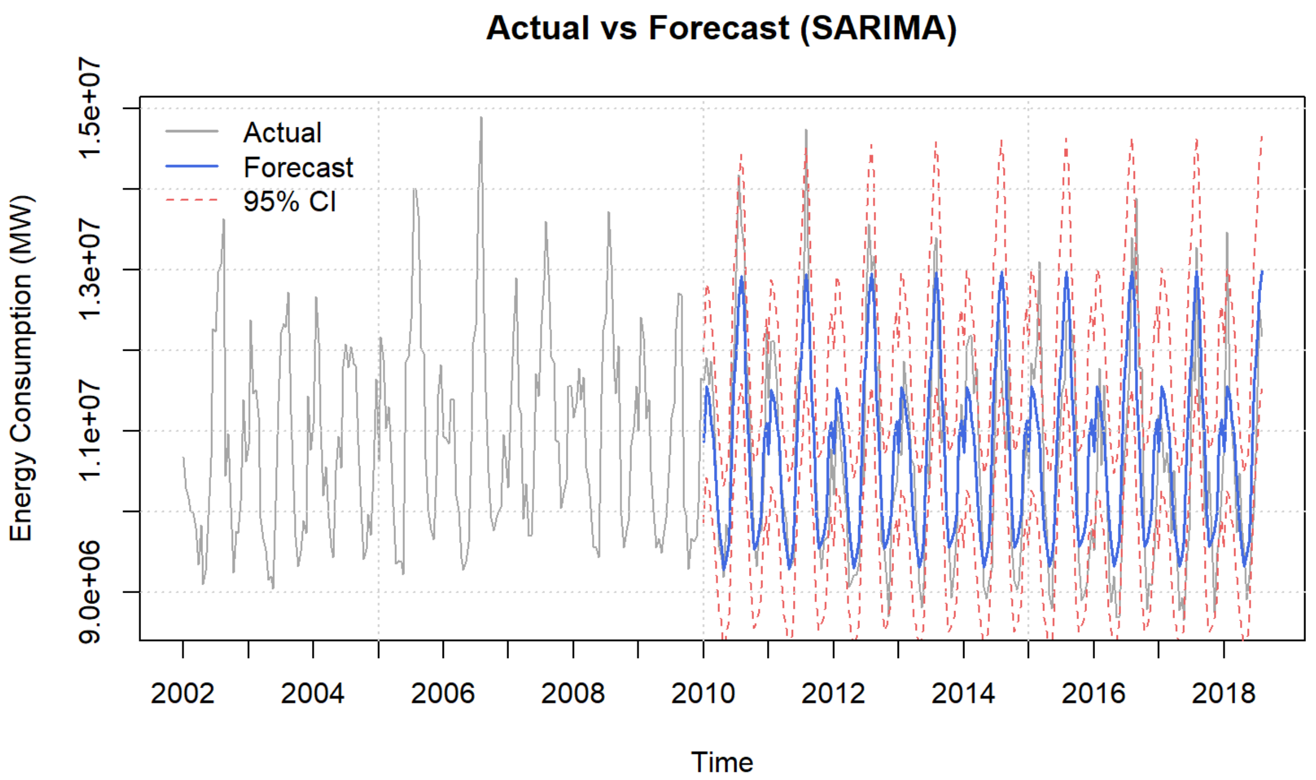
```r
plot(1:25,
     sapply(1:25, function(lag) Box.test(residuals, lag = lag, type = "Ljung-Box")$p.value),
     main = "Ljung-Box Test p-values Across Lags", xlab = "Lag", ylab = "Ljung-Box p-value",
     type = "b", pch = 19, ylim = c(0, 1))
abline(h = 0.05, col = "red", lty = 2)
```



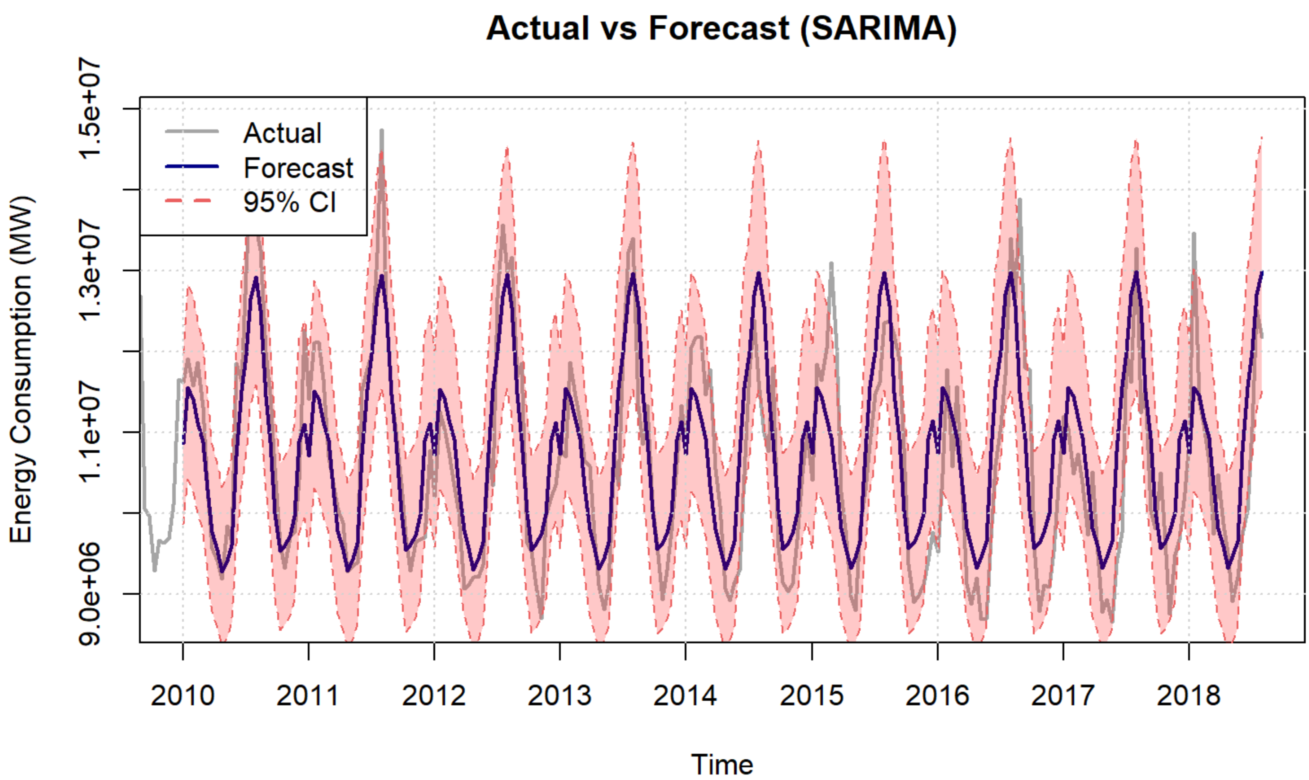
- **Residual Mean**: The residuals have a mean close to zero, indicating no significant bias in the model.
- **Homoscedasticity**: Residuals seem to have constant variance, but there are signs of increased fluctuations after 2010.
- **Normality**: The Shapiro-Wilk test shows that residuals are border-line normally distributed (p-value = 0.04306).
- **Independence**: ACF and PACF plots show no significant autocorrelation, and the Box-Ljung test confirms the independence of residuals (p-value = 0.2432).
- **Post-2010 Changes**: The model might not fully capture changes after 2010, suggesting the need for further decomposition and trend visualization.

Prototype modelling: SARIMA(2,0,1)(0,1,1)[26]

```
train <- window(ts.energy.biweekly, end = time(ts.energy.biweekly)[208])
test <- window(ts.energy.biweekly, start = time(ts.energy.biweekly)[208 + 1])

model.final.sarima <- arima(log(log((train))), order = c(2, 0, 1), seasonal =  c(0, 1, 1))

forecast.raw <- predict(model.final.sarima, n.ahead = length(test))
forecast.pred <- exp(exp(forecast.raw$pred))
forecast.upper <- exp(exp(forecast.raw$pred + 1.96 * forecast.raw$se))
forecast.lower <- exp(exp(forecast.raw$pred - 1.96 * forecast.raw$se))
forecast.time <- as.Date(as.numeric(time(test)))

plot(ts.energy.biweekly, main = "Actual vs Forecast (SARIMA)",
     xlab = "Time", ylab = "Energy Consumption (MW)",
     col = "darkgray", lwd = 1, xaxt = "n")
lines(forecast.time, forecast.pred, col = "royalblue", lwd = 1.5)
lines(forecast.time, forecast.upper, col = "indianred2", lty = 2, lwd = 1)
lines(forecast.time, forecast.lower, col = "indianred2", lty = 2, lwd = 1)
axis(1, at = t <- time(ts.energy.biweekly)[d <- !duplicated(y <- floor(time(ts.energy.biweekl
y)))],
     labels = y[d], las = 1)
legend("topleft", legend = c("Actual", "Forecast", "95% CI"),
       col = c("darkgray", "royalblue", "indianred2"),
       lty = c(1, 1, 2), lwd = c(1.5, 1.5, 1), bty = "n")
grid()
```



```
plot(ts.energy.biweekly, main = "Actual vs Forecast (SARIMA)",
     xlab = "Time", ylab = "Energy Consumption (MW)",
     col = "darkgray", lwd = 2, xaxt = "n", xlim = range(time(test)))
lines(forecast.time, forecast.pred, col = "darkblue", lwd = 2)
polygon(c(time(test), rev(time(test))), c(forecast.upper, rev(forecast.lower)),
        col = rgb(1, 0, 0, 0.2), border = NA)
lines(forecast.time, forecast.upper, col = "indianred2", lty = 2)
lines(forecast.time, forecast.lower, col = "indianred2", lty = 2)
axis(1, at = t <- time(ts.energy.biweekly)[d <- !duplicated(y <- floor(time(ts.energy.biweekl
y)))],
     labels = y[d], las = 1)
legend("topleft", legend = c("Actual", "Forecast", "95% CI"),
       col = c("darkgray", "darkblue", "indianred2"), lty = c(1, 1, 2), lwd = 2)
grid()
```



Forecasting and Performance Metrics

```
actual_values <- as.numeric(test)
forecasted_values <- forecast.pred
residuals <- actual_values - forecasted_values

mae <- mean(abs(residuals))
rmse <- sqrt(mean(residuals^2))
mape <- mean(abs(residuals / actual_values)) * 100
r_squared <- 1 - (sum(residuals^2) / sum((actual_values - mean(actual_values))^2))

metrics <- data.frame(
  Metric = c("MAE", "RMSE", "MAPE (%)", "R-squared"),
  Value = round(c(mae, rmse, mape, r_squared), 3)
)
print(metrics, row.names = FALSE)
```

```
##     Metric      Value
##        MAE 585420.028
##       RMSE 739120.764
##   MAPE (%)      5.492
##  R-squared      0.681
```

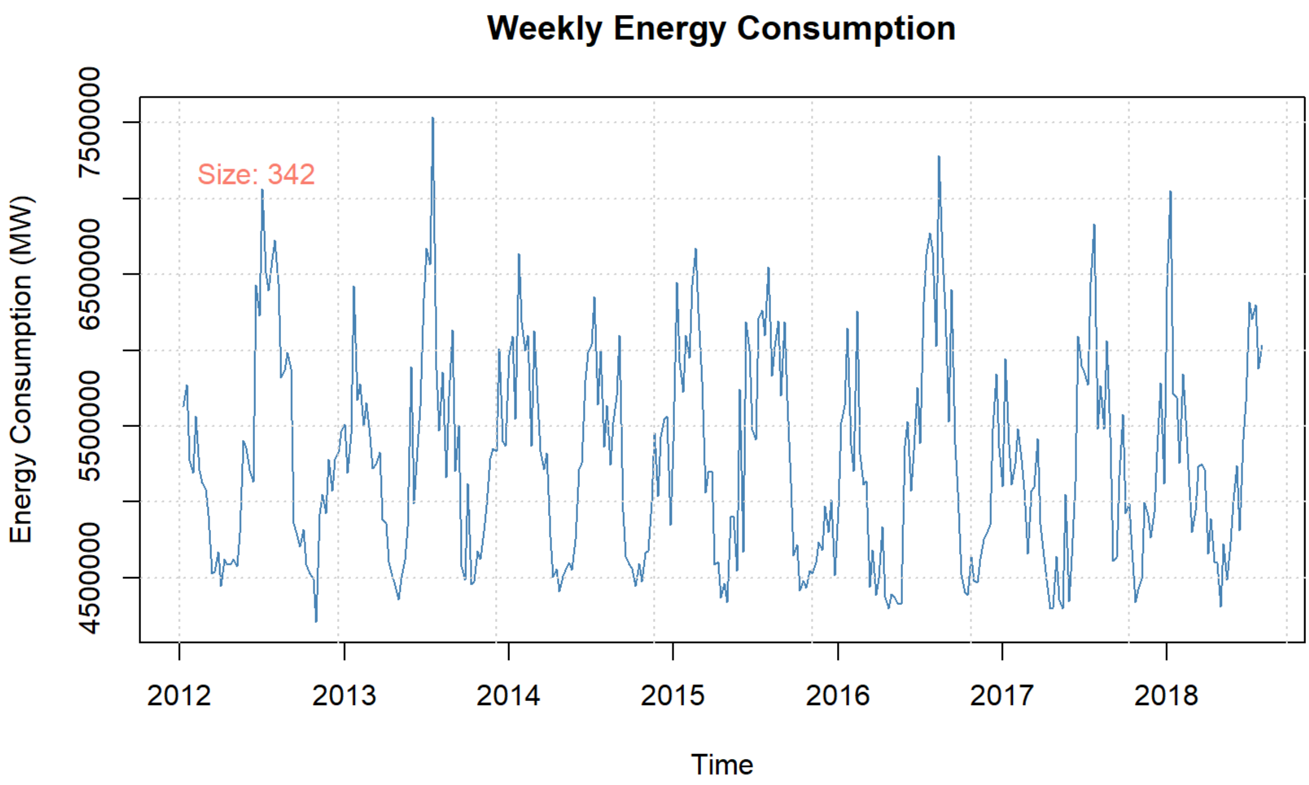
Splitting Time Series (Post 2011)

```
data.daily$Date <- as.POSIXct(data.daily$Date)
data.daily.post11 <- data.daily[data.daily$Date >= as.POSIXct("2012-01-01 00:00:00"), ]
data.daily.post11 <- data.daily.post11[-1, ]
data.daily.post11$Week <- data.daily.post11$Week - 522
data.daily.post11$Biweekly <- data.daily.post11$Biweekly - 261

# Aggregating and Pre-processing (Removing first and last observation)
data.post11.weekly <- aggregate(PJME_MW ~ Week, data = data.daily.post11, FUN = sum)
data.post11.weekly   <- data.post11.weekly[-c(1, nrow(data.post11.weekly)), ]
data.post11.weekly$Week <- data.post11.weekly$Week - 1

data.post11.biweekly <- aggregate(PJME_MW ~ Biweekly, data = data.daily.post11, FUN = sum)
data.post11.biweekly <- data.post11.biweekly[-c(1, nrow(data.post11.biweekly)), ]
data.post11.biweekly$Biweekly <- data.post11.biweekly$Biweekly - 1

ts.post11.weekly <- ts(data.post11.weekly$PJME_MW)
ts.post11.biweekly <- ts(data.post11.biweekly$PJME_MW)
```

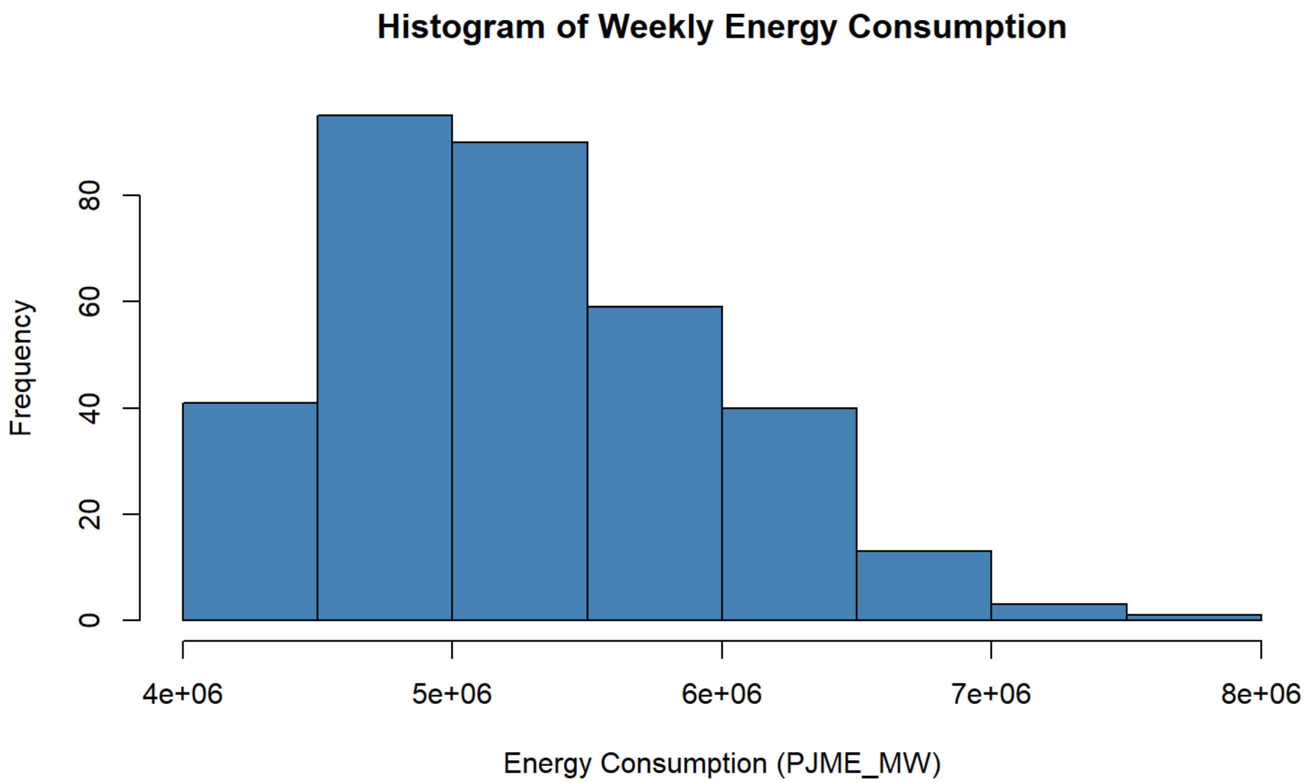
Data Pre-processing (Post 2011)

Plotting Weekly Data and Stationarity Check (Post 2011)

```
plot(ts.post11.weekly, main = "Weekly Energy Consumption",
     xlab = "Time", ylab = "Energy Consumption (MW)",
     col = "steelblue", lwd = 1, xaxt = "n")
text(x = 0,95, y = max(ts.post11.weekly) * 0.95,
     labels = paste("Size:", length(ts.post11.weekly)),
     col = "salmon", pos = 4)
axis(1, at = (2012:2018 - 2012) * 52, labels = 2012:2018, las = 1)
grid()
```




```
hist(ts.post11.weekly, main = "Histogram of Weekly Energy Consumption",
     xlab = "Energy Consumption (PJME_MW)", breaks = "FD", col = "steelblue")
```




```
adf.test(ts.post11.weekly); kpss.test(ts.post11.weekly); pp.test(ts.post11.weekly);
```

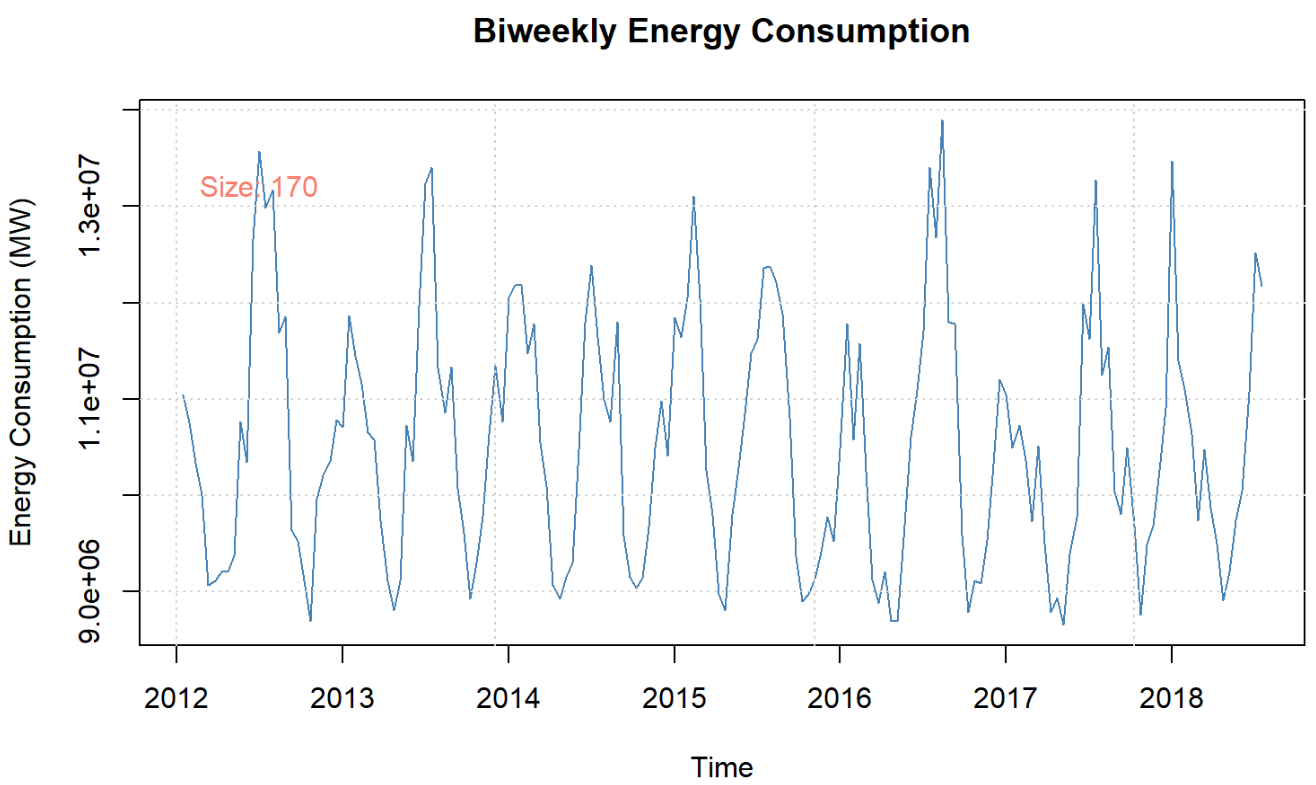
```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts.post11.weekly
## Dickey-Fuller = -7.8954, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  ts.post11.weekly
## KPSS Level = 0.049962, Truncation lag parameter = 5, p-value = 0.1
```

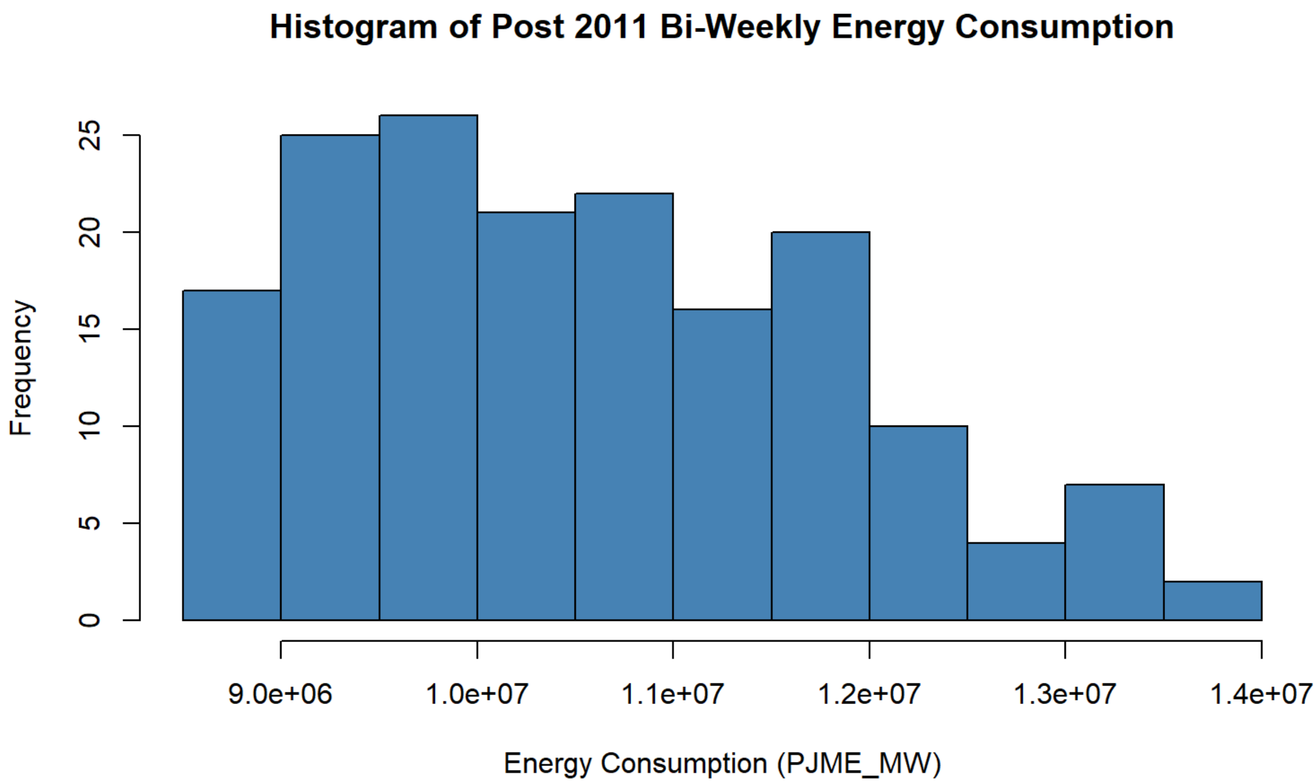
```
##
##  Phillips-Perron Unit Root Test
##
## data:  ts.post11.weekly
## Dickey-Fuller Z(alpha) = -85.679, Truncation lag parameter = 5, p-value
## = 0.01
## alternative hypothesis: stationary
```

Plotting Bi-Weekly Data and Stationarity Check (Post 2011)

```
plot(ts.post11.biweekly,
     main = "Biweekly Energy Consumption",
     xlab = "Time", ylab = "Energy Consumption (MW)",
     col = "steelblue", lwd = 1, xaxt = "n")
text(x = 0.95, y = max(ts.post11.biweekly) * 0.95,
     labels = paste("Size:", length(ts.post11.biweekly)),
     col = "salmon", pos = 4)
axis(1, at = (2012:2018 - 2012) * 26, labels = 2012:2018, las = 1)
grid()
```




```
hist(ts.post11.biweekly, main = "Histogram of Post 2011 Bi-Weekly Energy Consumption",
     xlab = "Energy Consumption (PJME_MW)", breaks = "FD", col = "steelblue")
```




```
adf.test(ts.post11.biweekly); kpss.test(ts.post11.biweekly); pp.test(ts.post11.biweekly);
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts.post11.biweekly
## Dickey-Fuller = -8.571, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  ts.post11.biweekly
## KPSS Level = 0.04428, Truncation lag parameter = 4, p-value = 0.1
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  ts.post11.biweekly
## Dickey-Fuller Z(alpha) = -58.024, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

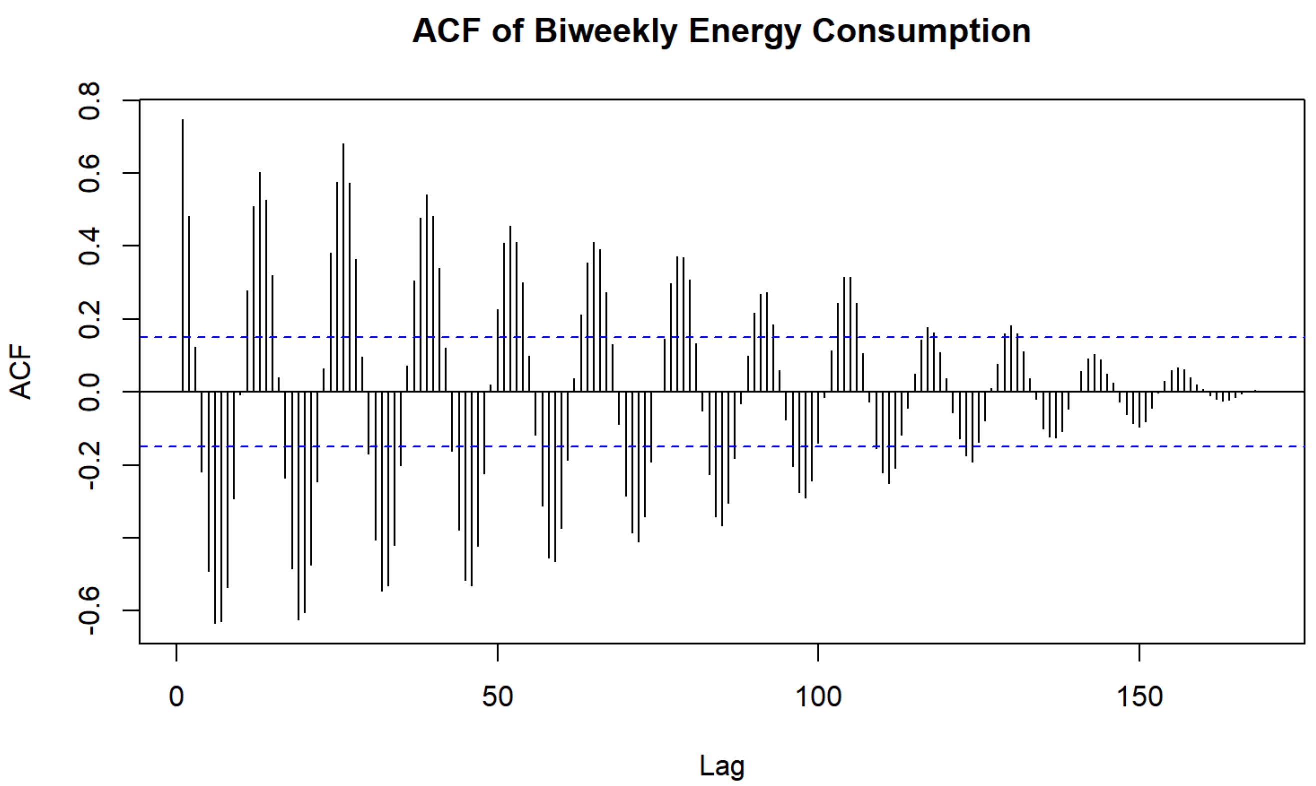
- **Biweekly data (Post-2011)** exhibited **stronger stationarity characteristics** than the weekly series, leading to the decision to continue with **biweekly stochastic trend analysis**
- However, **full-period biweekly data** (2002–2018) still displayed stronger stationarity overall

Auto-correlation Analysis and choosing d (Post 2011)

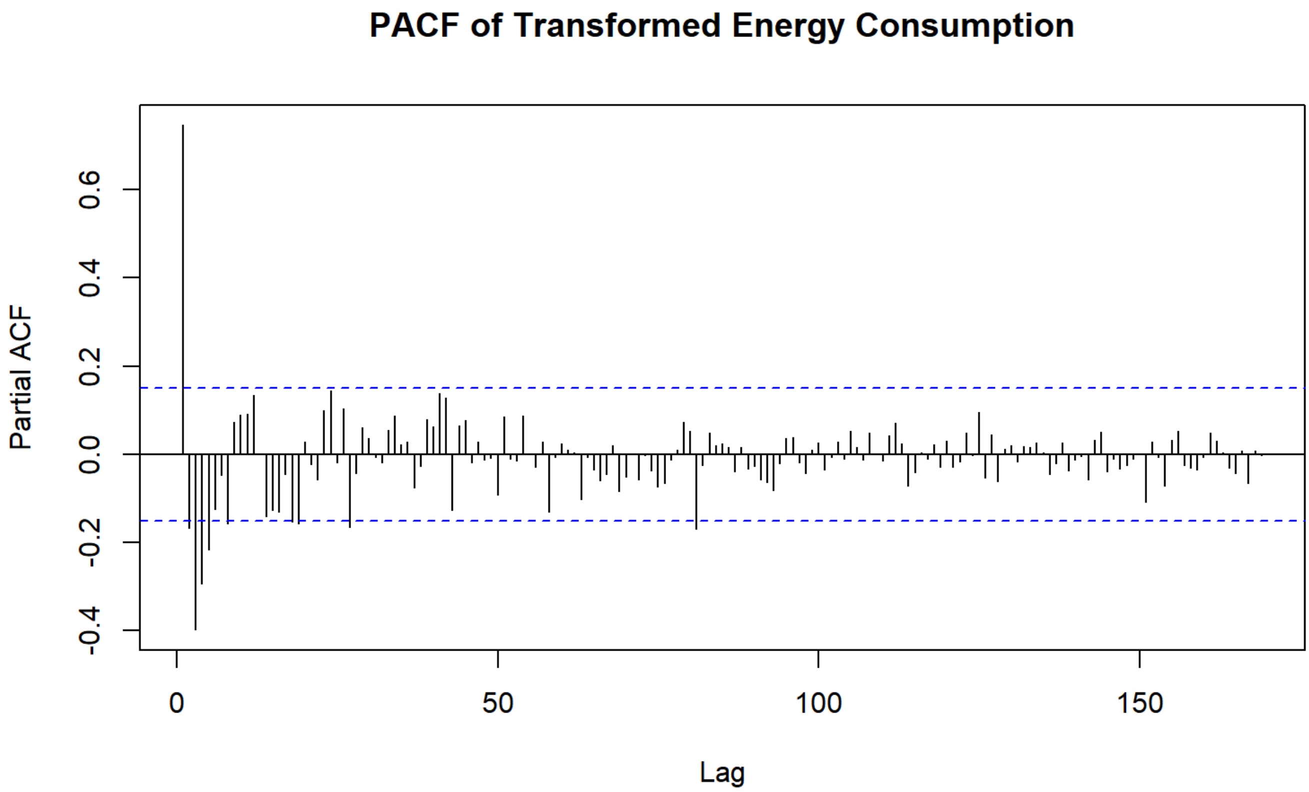
```
eacf(ts.post11.biweekly)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x o x x x x x x o x  x  x  x
## 1 x x o x x x x x x o x  x  x  x
## 2 x x o o o o o x o o o  o  o  o
## 3 x o o o o o o o o o o  o  o  o
## 4 x x o o o o o o o o o  o  o  o
## 5 x x o o o o x o o o o  o  o  o
## 6 x o x o o o x o o o o  o  o  o
## 7 x x x o x x x o o o o  o  o  o
```

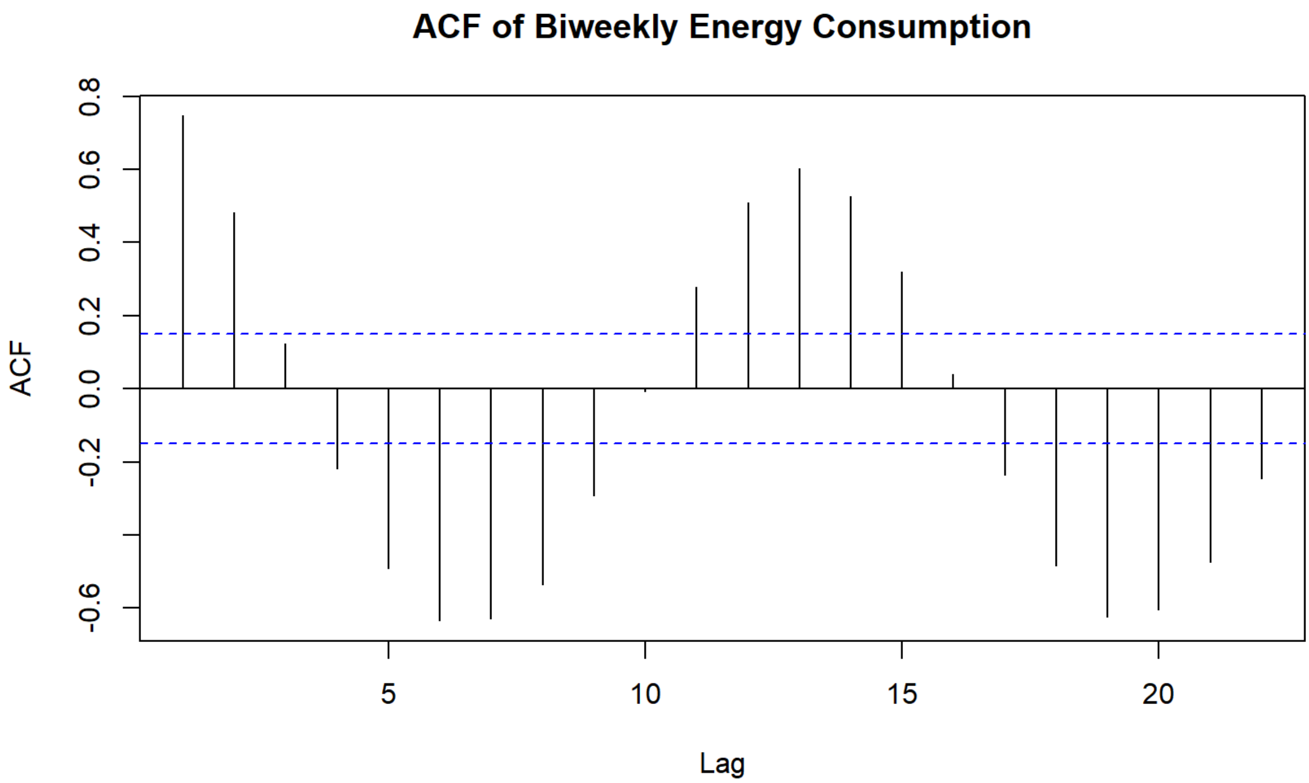
```
acf(ts.post11.biweekly, main = "ACF of Biweekly Energy Consumption", lag.max=260)
```



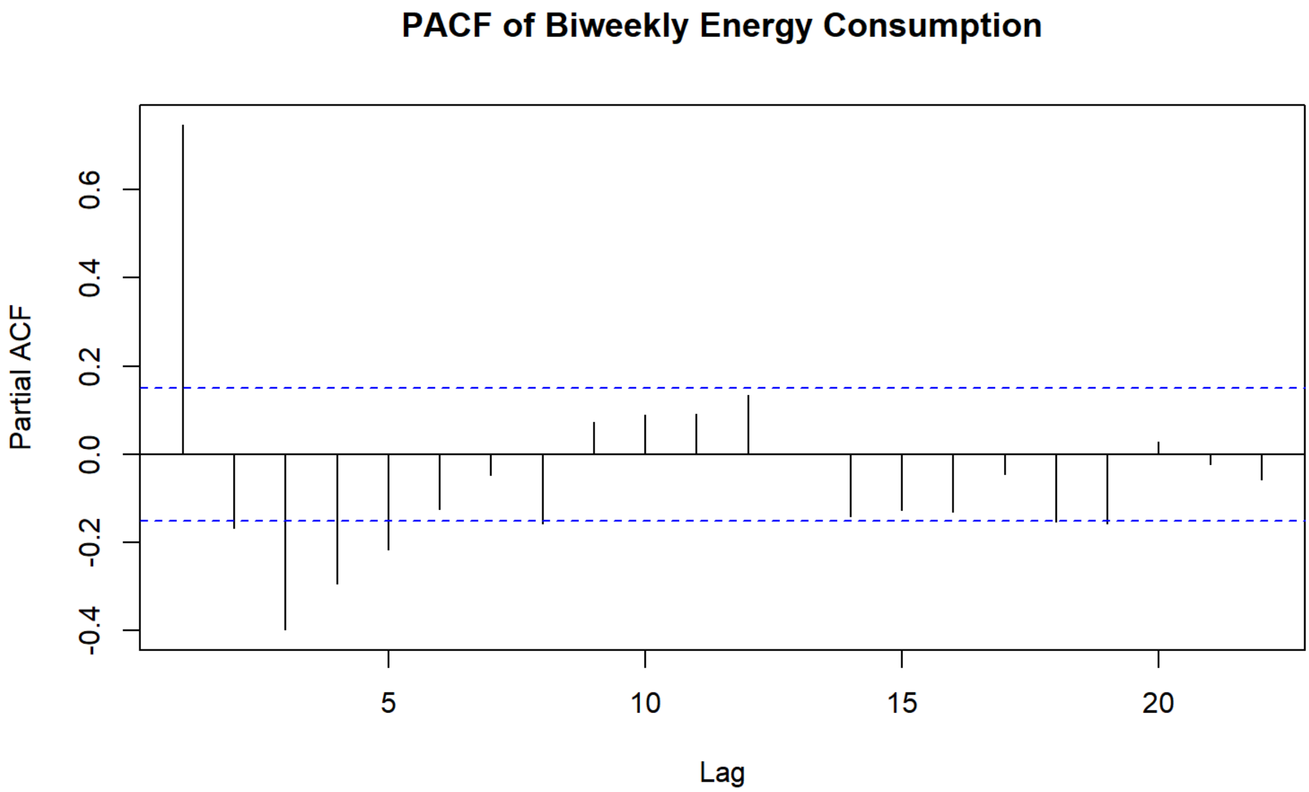

```
pacf(ts.post11.biweekly, main = "PACF of Transformed Energy Consumption", lag.max=260)
```




```
acf(ts.post11.biweekly, main = "ACF of Biweekly Energy Consumption")
```




```
pacf(ts.post11.biweekly, main = "PACF of Biweekly Energy Consumption")
```



1. **For Non-Seasonal part**
    - **ACF (decay with no sharp cutoff)** → MA(1), MA(2)
    - **PACF (sharp cutoff at Lag 1)**→ AR(1)
2. **From EACF** →
    - **AR(2)**: Most lags become insignificant (o) after Lag 1 → **AR(2) should be sufficient**.
    - **MA(2)**: Lags beyond 2 become insignificant → **MA(2) likely sufficient**.
3. **For Seasonal part**
    - **ACF (decays slowly with not sharp cutoff till lag 14)** → Differencing might be needed
    - **PACF (Cuts off after lag 1)** → **AR(1) may be adequate**

Candidate Modelling (Post 2011)

ARIMA models

1. ARIMA(0,0,1)
2. ARIMA(0,0,2)
3. ARIMA(1,0,0)
4. ARIMA(1,0,1)
5. ARIMA(1,0,2)
6. ARIMA(2,0,0)
7. ARIMA(2,0,1)
8. ARIMA(2,0,2)

```r
model.post.arima_1  <- arima(ts.post11.biweekly, order = c(0, 0, 1))
model.post.arima_2  <- arima(ts.post11.biweekly, order = c(0, 0, 2))
model.post.arima_3  <- arima(ts.post11.biweekly, order = c(1, 0, 0))
model.post.arima_4  <- arima(ts.post11.biweekly, order = c(1, 0, 1))
model.post.arima_5  <- arima(ts.post11.biweekly, order = c(1, 0, 2))
model.post.arima_6 <- arima(ts.post11.biweekly, order = c(2, 0, 0))
model.post.arima_7 <- arima(ts.post11.biweekly, order = c(2, 0, 1))
model.post.arima_8 <- arima(ts.post11.biweekly, order = c(2, 0, 2))
model.post.arima_ns_auto <- auto.arima(ts.post11.biweekly)
summary(model.arima_ns_auto)
```

```
## Series: ts.energy.biweekly
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     ma2         mean
##       1.7124  -0.9274  -1.1413  0.4904  10778043.86
## s.e.  0.0225   0.0218   0.0611  0.0514     52544.37
##
## sigma^2 = 4.549e+11:  log likelihood = -6409.8
## AIC=12831.61   AICc=12831.81   BIC=12856.02
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE     MASE
## Training set -232.7162 670572.5 526771.1 -0.3685558 4.836914 0.807866
##                    ACF1
## Training set 0.05080482
```

```r
results <- data.frame(
  Model = c("ARMA(0,0,1)", "ARMA(0,0,2)", "ARMA(1,0,0)", "ARMA(1,0,1)", "ARMA(1,0,2)",
            "ARMA(2,0,0)", "ARMA(2,0,1)", "ARMA(2,0,2)", "Auto.ARIMA"),
  AIC = c(AIC(model.post.arima_1), AIC(model.post.arima_2), AIC(model.post.arima_3),
          AIC(model.post.arima_4), AIC(model.post.arima_5), AIC(model.post.arima_6),
          AIC(model.post.arima_7), AIC(model.post.arima_8), AIC(model.post.arima_ns_auto)),
  BIC = c(BIC(model.post.arima_1), BIC(model.post.arima_2), BIC(model.post.arima_3),
          BIC(model.post.arima_4), BIC(model.post.arima_5), BIC(model.post.arima_6),
          BIC(model.post.arima_7), BIC(model.post.arima_8), BIC(model.post.arima_ns_auto)))
results[order(results$AIC), ]
```

```
##         Model      AIC      BIC
## 8 ARMA(2,0,2) 5064.824 5083.638
## 9  Auto.ARIMA 5064.824 5083.638
## 7 ARMA(2,0,1) 5085.821 5101.500
## 5 ARMA(1,0,2) 5114.198 5129.877
## 6 ARMA(2,0,0) 5124.838 5137.381
## 4 ARMA(1,0,1) 5127.187 5139.731
## 3 ARMA(1,0,0) 5127.421 5136.828
## 2 ARMA(0,0,2) 5135.915 5148.458
## 1 ARMA(0,0,1) 5179.251 5188.659
```

```r
results[order(results$BIC), ]
```

```
##         Model      AIC      BIC
## 8 ARMA(2,0,2) 5064.824 5083.638
## 9  Auto.ARIMA 5064.824 5083.638
## 7 ARMA(2,0,1) 5085.821 5101.500
## 5 ARMA(1,0,2) 5114.198 5129.877
## 3 ARMA(1,0,0) 5127.421 5136.828
## 6 ARMA(2,0,0) 5124.838 5137.381
## 4 ARMA(1,0,1) 5127.187 5139.731
## 2 ARMA(0,0,2) 5135.915 5148.458
## 1 ARMA(0,0,1) 5179.251 5188.659
```

Best Performing Seasonal Models (Post 2011)

1. ARMA(2,0,2)
2. ARMA(2,0,1)

SARIMA Models

1. SARIMA(2,0,2)(0,1,1)
2. SARIMA(2,0,2)(1,1,1)
3. SARIMA(2,0,2)(1,1,2)
4. SARIMA(2,0,1)(0,1,1)
5. SARIMA(2,0,1)(1,1,1)
6. SARIMA(2,0,1)(1,1,2)

```
ts.post11.biweekly <- ts(ts.post11.biweekly, start = c(2012, 1), frequency = 26)
model.post.sarima_1  <- arima(ts.post11.biweekly, order = c(2, 0, 2), seasonal = c(0, 1, 1))
model.post.sarima_2  <- arima(ts.post11.biweekly, order = c(2, 0, 2), seasonal = c(1, 1, 1))
model.post.sarima_3  <- arima(ts.post11.biweekly, order = c(2, 0, 2), seasonal = c(1, 1, 2))
model.post.sarima_4 <- arima(ts.post11.biweekly, order = c(2, 0, 1), seasonal = c(0, 1, 1))
model.post.sarima_5 <- arima(ts.post11.biweekly, order = c(2, 0, 1), seasonal = c(1, 1, 1))
model.post.sarima_6 <- arima(ts.post11.biweekly, order = c(2, 0, 1), seasonal = c(1, 1, 2))
model.post.sarima_auto <- auto.arima(ts.post11.biweekly)
summary(model.post.sarima_auto)
```

```
## Series: ts.post11.biweekly
## ARIMA(1,0,1)(0,1,1)[26]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.6190  -0.2516  -0.5986
## s.e.  0.1427   0.1727   0.1424
##
## sigma^2 = 4.679e+11:  log likelihood = -2143.42
## AIC=4294.83   AICc=4295.12   BIC=4306.71
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE    MAPE      MASE
## Training set -15535.82 622978.6 448054.3 -0.3752769 4.13975 0.7368919
##                      ACF1
## Training set -0.004517471
```

```r
results <- data.frame(
  Model = c("SARIMA(2,0,2)(1,0,2)", "SARIMA(2,0,2)(0,1,1)", "SARIMA(2,0,2)(1,1,1)",
            "SARIMA(2,0,1)(1,1,1)", "SARIMA(2,0,1)(1,1,2)", "SARIMA(0,1,2)(0,0,1)",
            "Auto.ARIMA"),
  AIC = c(AIC(model.post.sarima_1), AIC(model.post.sarima_2), AIC(model.post.sarima_3),
          AIC(model.post.sarima_4), AIC(model.post.sarima_5), AIC(model.post.sarima_6),
          AIC(model.post.sarima_auto)),
  BIC = c(BIC(model.post.sarima_1), BIC(model.post.sarima_2), BIC(model.post.sarima_3),
          BIC(model.post.sarima_4), BIC(model.post.sarima_5), BIC(model.post.sarima_6),
          BIC(model.post.sarima_auto))
)
results[order(results$AIC), ]
```

```
##                  Model      AIC      BIC
## 6 SARIMA(0,1,2)(0,0,1) 4288.874 4309.662
## 5 SARIMA(2,0,1)(1,1,2) 4290.744 4308.562
## 3 SARIMA(2,0,2)(1,1,1) 4290.867 4314.625
## 2 SARIMA(2,0,2)(0,1,1) 4291.909 4312.698
## 7           Auto.ARIMA 4294.834 4306.713
## 4 SARIMA(2,0,1)(1,1,1) 4296.010 4310.859
## 1 SARIMA(2,0,2)(1,0,2) 4296.876 4314.695
```

```r
results[order(results$BIC), ]
```

```
##                  Model      AIC      BIC
## 7           Auto.ARIMA 4294.834 4306.713
## 5 SARIMA(2,0,1)(1,1,2) 4290.744 4308.562
## 6 SARIMA(0,1,2)(0,0,1) 4288.874 4309.662
## 4 SARIMA(2,0,1)(1,1,1) 4296.010 4310.859
## 2 SARIMA(2,0,2)(0,1,1) 4291.909 4312.698
## 3 SARIMA(2,0,2)(1,1,1) 4290.867 4314.625
## 1 SARIMA(2,0,2)(1,0,2) 4296.876 4314.695
```

Best Performing Seasonal Model (Post 2011)

1. SARIMA(0,1,2)(0,0,1)
2. SARIMA(1,0,1)(0,1,1)
3. SARIMA(2,0,1)(1,1,2)

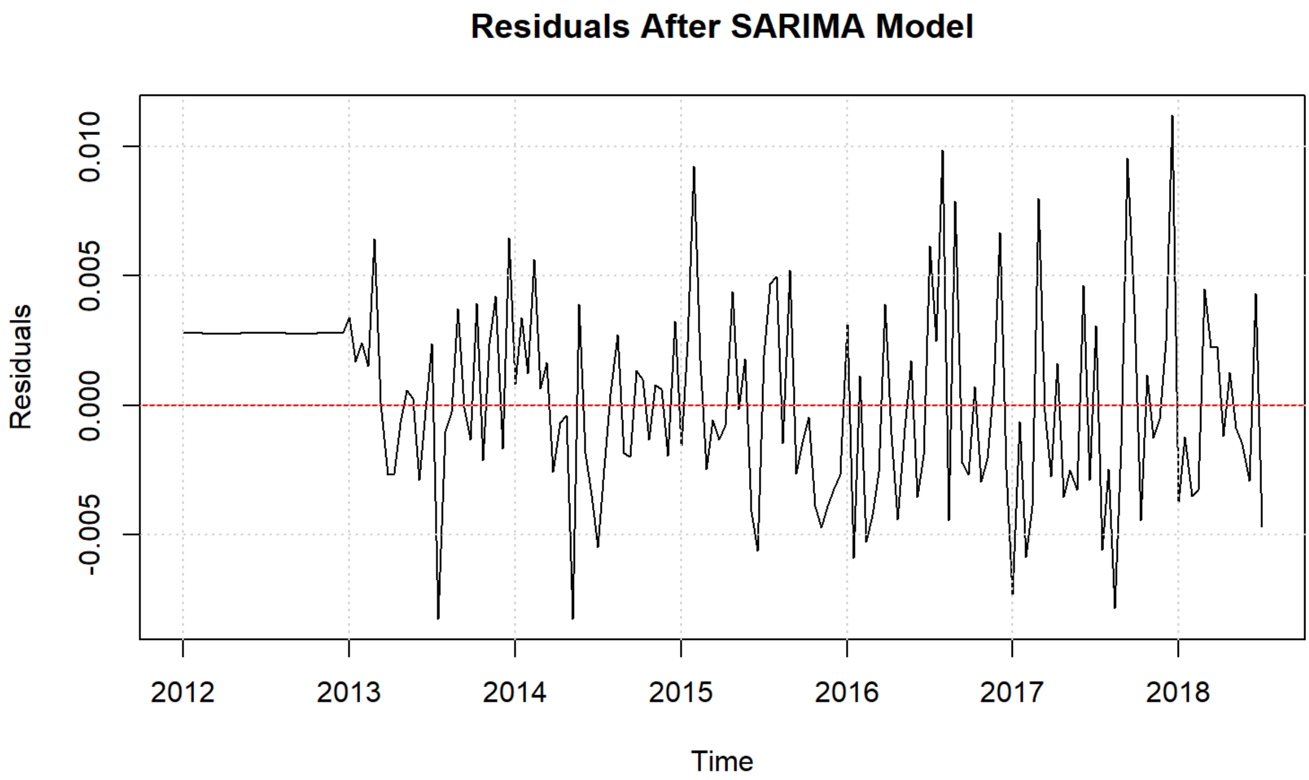
- **SARIMA(2,0,1)[0,1,1]** - Initial best candidate is **4th and 5th best choice now by BIC and AIC**.
- **Finalized model (SARIMA(1,0,1)(0,1,1)):** Previous suggested no differencing for non-seasonal part

Model Diagnostic: (Post 2011)

SARIMA(1,0,1)(0,1,1)[26]

Double Log transformation

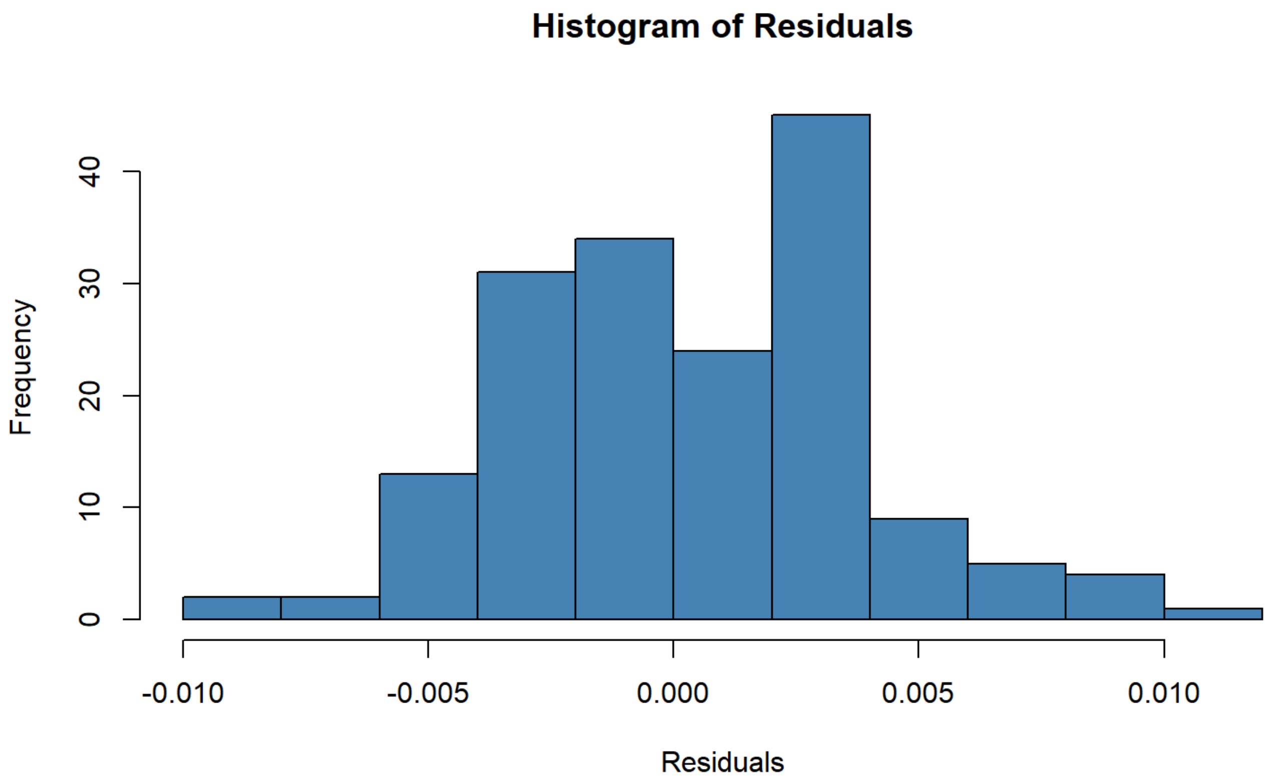
```
model.post.sarima <- arima(log(log(ts.post11.biweekly)),
                           order=c(1, 0, 1), seasonal = c(0, 1, 1))
residuals <- resid(model.post.sarima)

plot(residuals, main="Residuals After SARIMA Model",
     ylab="Residuals", xlab = "Time", xaxt = "n")
abline(h = 0, col = "red", lty = 1, lwd = 1)
axis(1, at = t <- time(ts.post11.biweekly)[d <- !duplicated(y <- floor(time(ts.post11.biweekl
y)))],
     labels = y[d], las = 1)
grid()
```



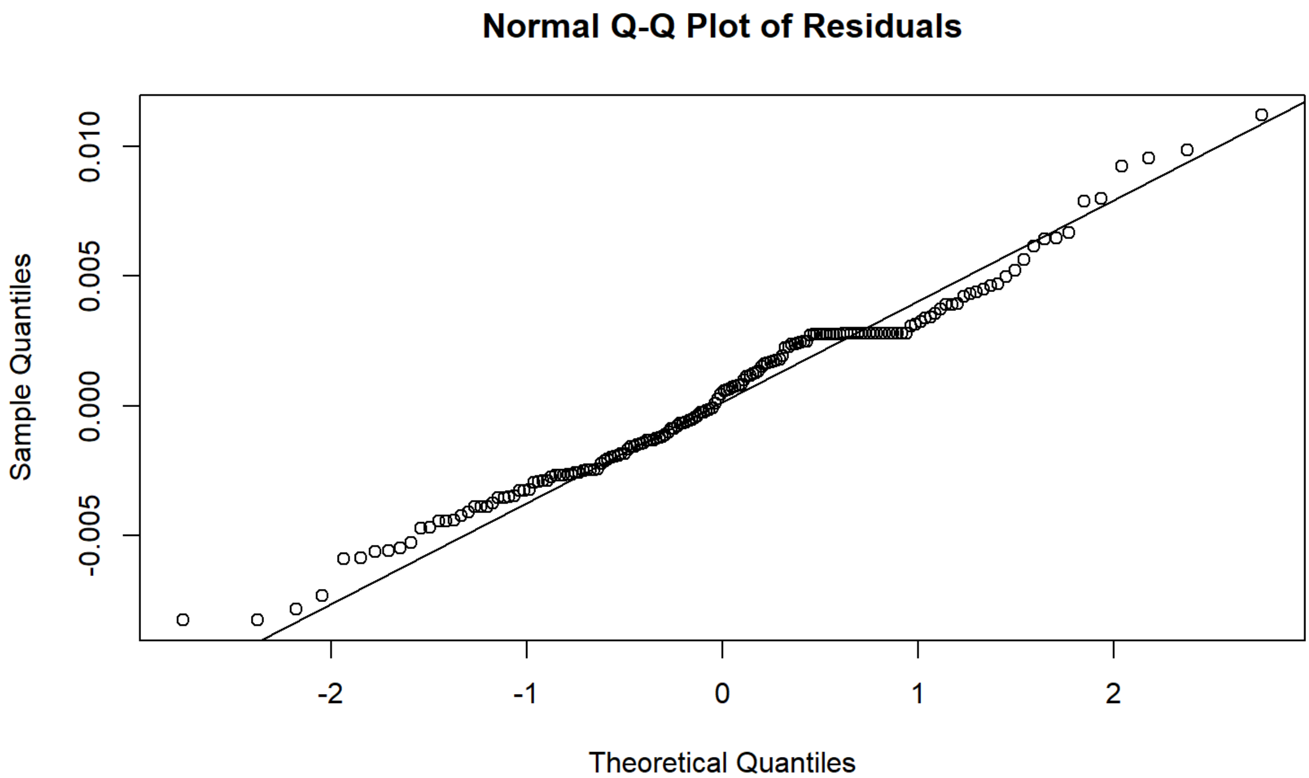

```
shapiro.test(residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98262, p-value = 0.03193
```

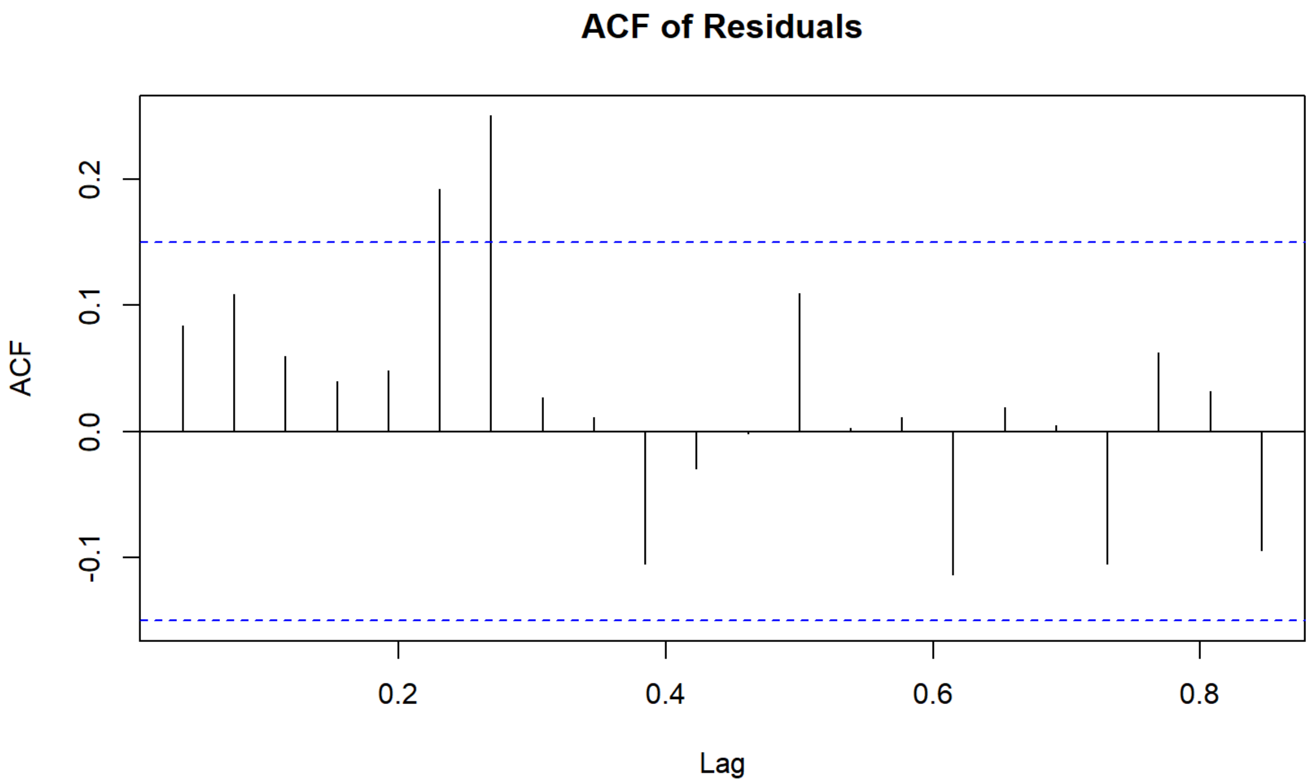
```
hist(residuals, main = "Histogram of Residuals", col="steelblue",
     xlab = "Residuals", breaks = "FD")
```



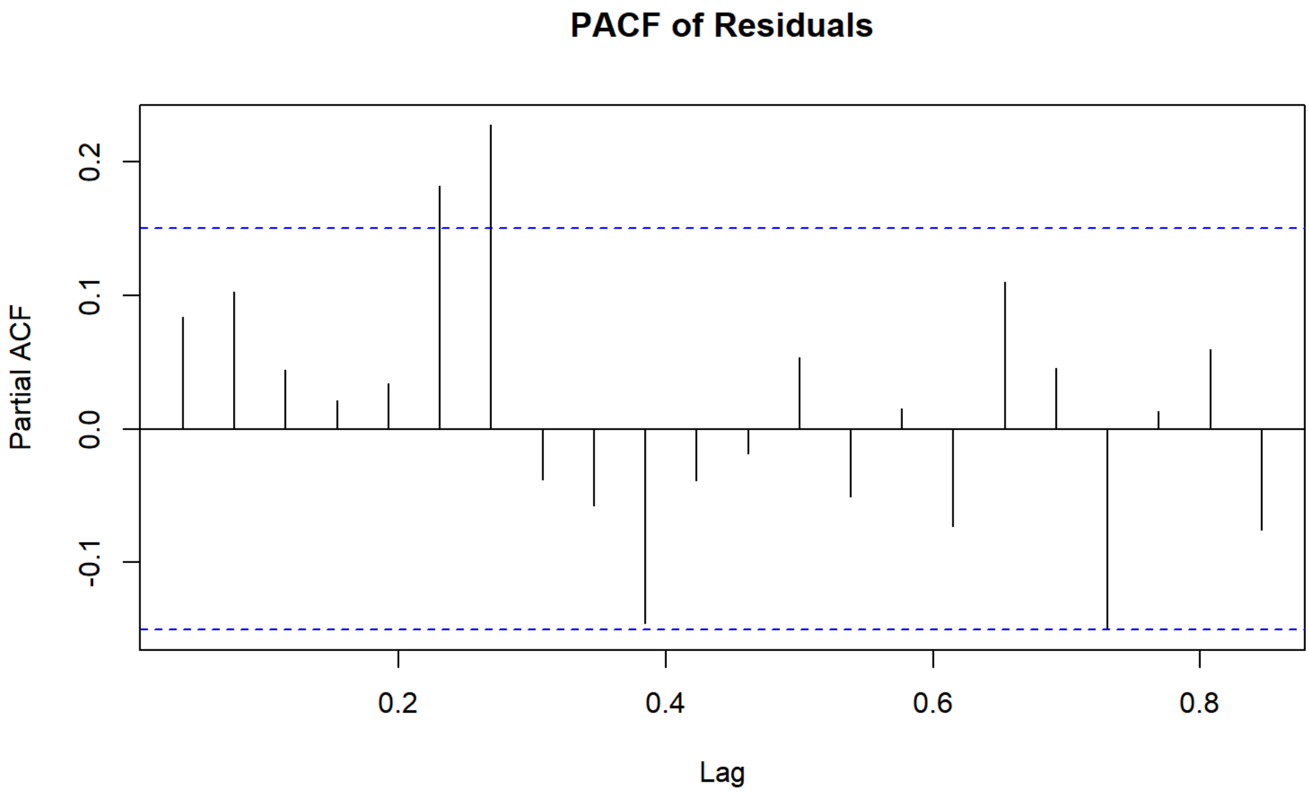

```
qqnorm(residuals, main = "Normal Q-Q Plot of Residuals"); qqline(residuals)
```




```
acf(residuals, main = "ACF of Residuals")
```



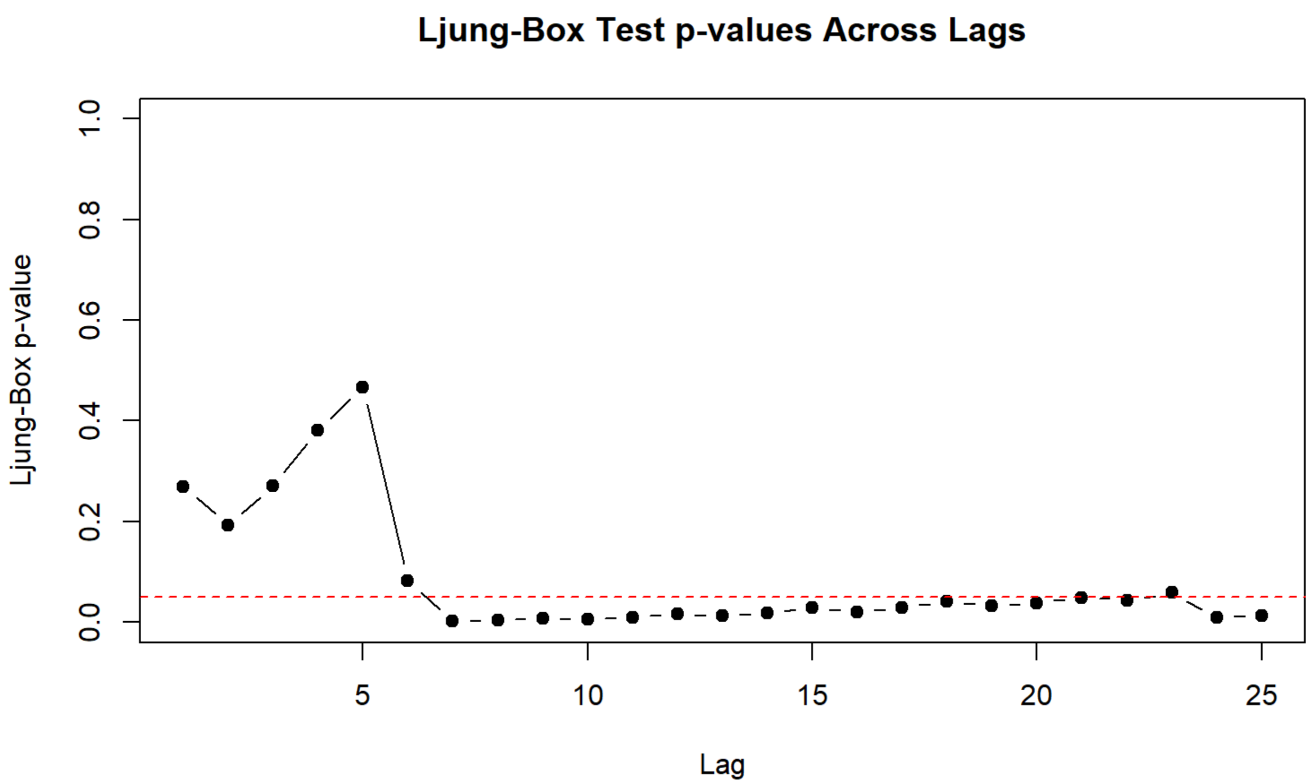

```
pacf(residuals, main = "PACF of Residuals");
```




```
Box.test(residuals, lag = 20, type = "Box-Pierce")
```

```
##
##  Box-Pierce test
##
## data:  residuals
## X-squared = 30.54, df = 20, p-value = 0.06157
```

```
plot(1:25,
     sapply(1:25, function(lag) Box.test(residuals, lag = lag, type = "Ljung-Box")$p.value),
     main = "Ljung-Box Test p-values Across Lags", xlab = "Lag", ylab = "Ljung-Box p-value",
     type = "b", pch = 19, ylim = c(0, 1))
abline(h = 0.05, col = "red", lty = 2)
```

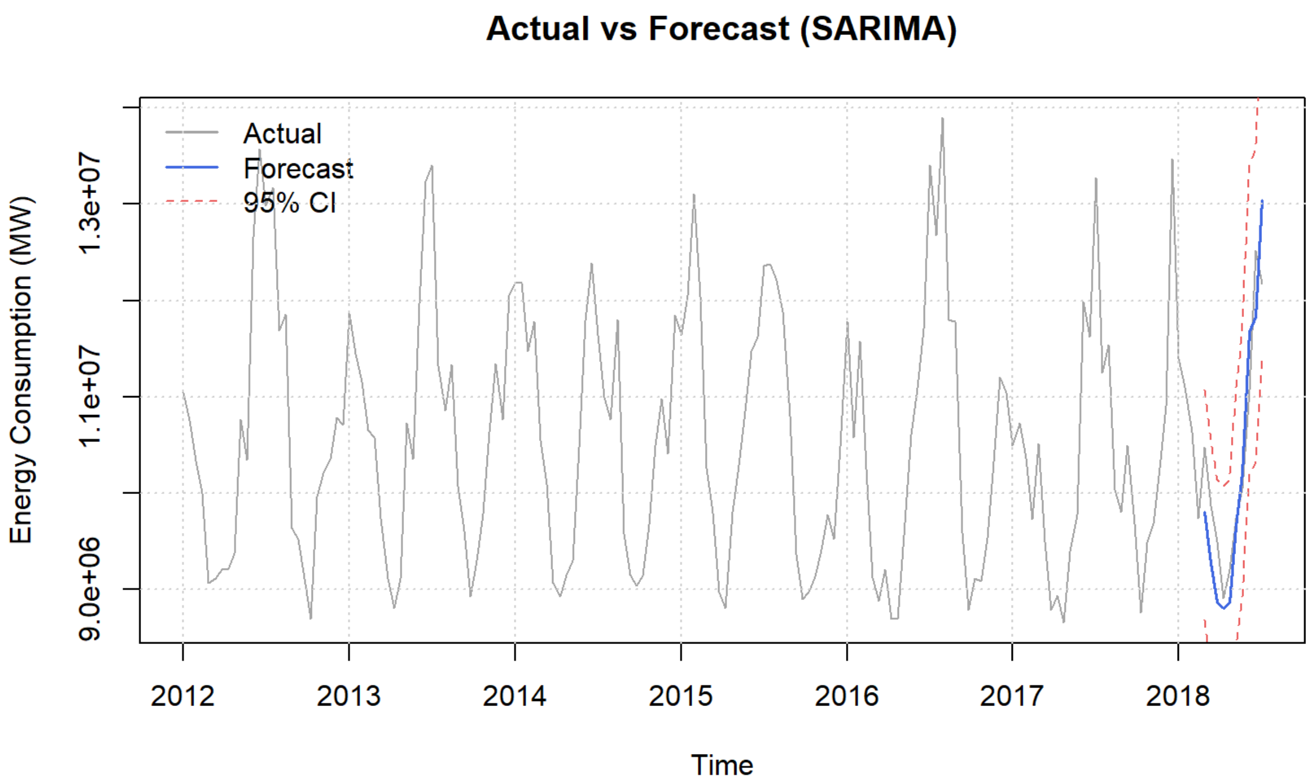


- **Residual Mean**: The residuals have a mean close to zero, indicating no significant bias in the model.
- **Homoscedasticity**: Residuals seem to have constant variance, but there are signs of increased fluctuations after 2010.
- **Normality**: The Shapiro-Wilk test shows that residuals are bordeline normally distributed (p-value = 0.2328).
- **Independence**: ACF and PACF plots show some correlation autocorrelation after lag 5, and the Box-Ljung test confirms the dependence of residuals (p-value = 6.083e-07).
- **Post-2010 Changes**: The model might not fully capture changes after 2010, suggesting the need for further decomposition and trend visualization.

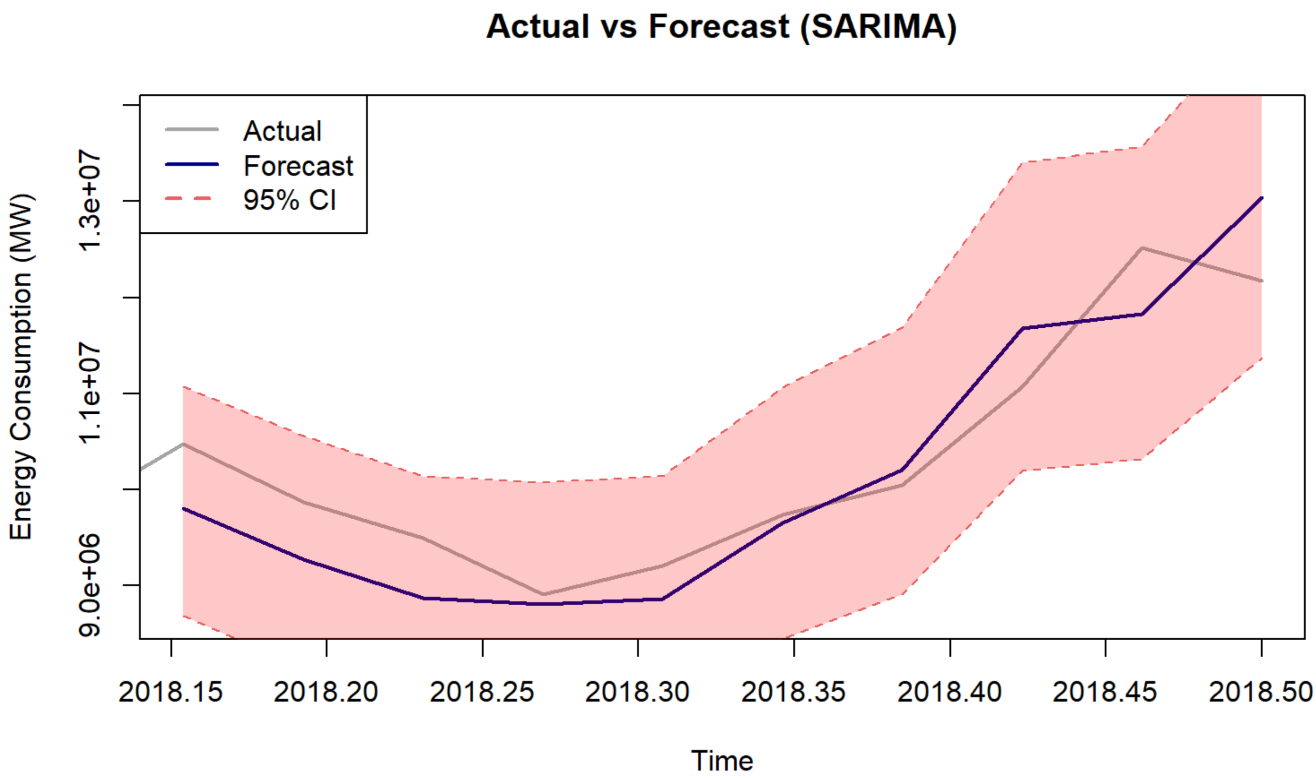
Prototype Modelling: (Post 2011)

SARIMA(1,0,1)(0,1,1)

```
train <- window(ts.post11.biweekly, end = time(ts.post11.biweekly)[160])
test <- window(ts.post11.biweekly, start = time(ts.post11.biweekly)[160 + 1])

model.post.final.sarima <- arima(log(log(train)), order = c(1, 0, 1), seasonal = c(0, 1, 1))

forecast.raw <- predict(model.post.final.sarima, n.ahead = length(test))
forecast.pred <- exp(exp(forecast.raw$pred))
forecast.upper <- exp(exp(forecast.raw$pred + 1.96 * forecast.raw$se))
forecast.lower <- exp(exp(forecast.raw$pred - 1.96 * forecast.raw$se))

plot(ts.post11.biweekly, main = "Actual vs Forecast (SARIMA)", xlab = "Time",
     ylab = "Energy Consumption (MW)",col = "darkgray", lwd = 1, xaxt = "n")
lines(forecast.pred, col = "royalblue", lwd = 1.5)
lines(forecast.upper, col = "indianred2", lty = 2, lwd = 1)
lines(forecast.lower, col = "indianred2", lty = 2, lwd = 1)
axis(1, at = t <- time(ts.post11.biweekly)[d <- !duplicated(y <- floor(time(ts.post11.biweekl
y)))],
     labels = y[d], las = 1)
legend("topleft", legend = c("Actual", "Forecast", "95% CI"),
col = c("darkgray", "royalblue", "indianred2"),
       lty = c(1, 1, 2), lwd = c(1.5, 1.5, 1), bty = "n")
grid()
```



```
plot(ts.post11.biweekly, main = "Actual vs Forecast (SARIMA)",
     ylab = "Energy Consumption (MW)", xlab = "Time",
     col = "darkgray", lwd = 2, xlim = range(time(test)))
lines(forecast.pred, col = "darkblue", lwd = 2)
polygon(c(time(test), rev(time(test))), c(forecast.upper, rev(forecast.lower)),
        col = rgb(1, 0, 0, 0.2), border = NA)
lines(forecast.upper, col = "indianred2", lty = 2)
lines(forecast.lower, col = "indianred2", lty = 2)
legend("topleft", legend = c("Actual", "Forecast", "95% CI"),
       col = c("darkgray", "darkblue", "indianred2"), lty = c(1, 1, 2), lwd = 2)
```



Forecasting and Performance Metrics (Post 2011)

```
res <- as.numeric(test) - forecast.pred
mape <- mean(abs(res / as.numeric(test))) * 100
r_squared <- 1 - (sum(res^2) / sum((as.numeric(test) - mean(as.numeric(test)))^2))
metrics <- data.frame(
 Metric = c("MAE", "RMSE", "MAPE (%)", "R-squared"),
 Value = round(c(mean(abs(res)), sqrt(mean(res^2)), mape, r_squared), 3))
print(metrics, row.names = FALSE)
```

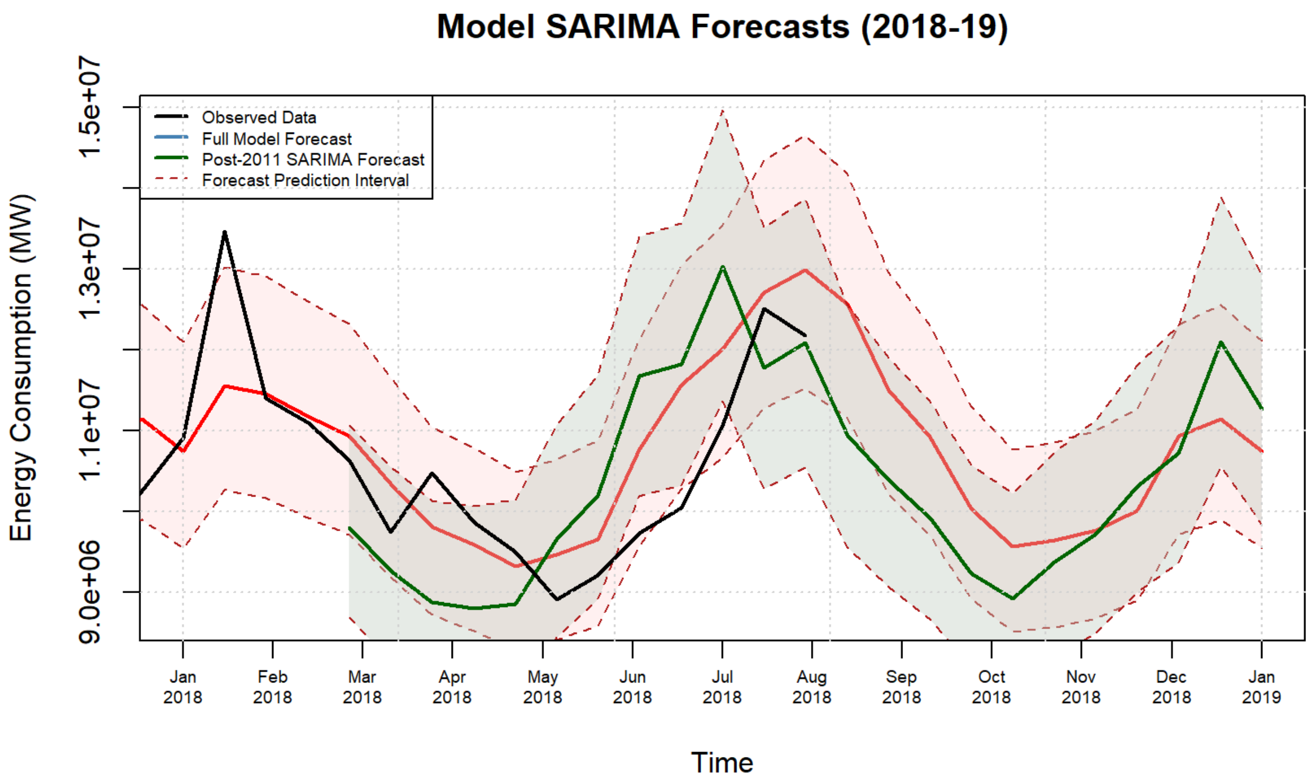
```
##     Metric      Value
##        MAE 474926.190
##       RMSE 544114.244
##   MAPE (%)      4.451
##  R-squared      0.778
```

Visualizing Both Forecasting results

- **Data Full** - SARIMA(2,0,1)(0,1,1)
- **Post 2011** - SARIMA(1,0,1)(1,1,1)

```
for.full.raw <- predict(model.final.sarima, n.ahead = 235)
for.full.pred <- exp(exp(for.full.raw$pred))
for.full.upper <- exp(exp(for.full.raw$pred + 1.96 * for.full.raw$se))
for.full.lower <- exp(exp(for.full.raw$pred - 1.96 * for.full.raw$se))

for.post.raw <- predict(model.post.final.sarima, n.ahead = 23)
for.post.pred <- exp(exp(for.post.raw$pred))
for.post.upper <- exp(exp(for.post.raw$pred + 1.96 * for.post.raw$se))
for.post.lower <- exp(exp(for.post.raw$pred - 1.96 * for.post.raw$se))
```

```
plot(ts.energy.biweekly, main = "Model SARIMA Forecasts (2018-19)",
     xlab = "Time", ylab = "Energy Consumption (MW)",
     col = "black", lwd = 2, xaxt = "n", xlim = c(2018, 2019))
polygon(c(time(for.full.upper), rev(time(for.full.lower))),
        c(for.full.upper, rev(for.full.lower)),
        col = adjustcolor("mistyrose", alpha.f = 0.4), border = NA)
lines(for.full.pred, col = "red", lwd = 2)
lines(for.full.upper, col = "firebrick", lty = 2)
lines(for.full.lower, col = "firebrick", lty = 2)
polygon(c(time(for.post.upper), rev(time(for.post.lower))),
        c(for.post.upper, rev(for.post.lower)),
        col = adjustcolor("honeydew3", alpha.f = 0.4), border = NA)
lines(for.post.pred, col = "darkgreen", lwd = 2)
lines(for.post.upper, col = "firebrick", lty = 2)
lines(for.post.lower, col = "firebrick", lty = 2)
lines(ts.energy.biweekly, col = "black", lwd = 2)
months_seq <- seq(from = as.Date("2018-01-01"), to = as.Date("2019-12-01"), by = "1 month")
month_ticks <- as.numeric(format(months_seq, "%Y")) + (as.numeric(format(months_seq, "%m")) -
1) / 12
axis(1, at = month_ticks, labels = format(months_seq, "%b\n%Y"), cex.axis = 0.6, las = 1)
legend("topleft", legend = c("Observed Data", "Full Model Forecast",
                             "Post-2011 SARIMA Forecast", "Forecast Prediction Interval"),
       col = c("black", "steelblue", "darkgreen", "firebrick"),
       lty = c(1, 1, 1, 2), lwd = c(2, 2, 2, 1), bg = "white", cex = 0.6)
grid()
```



Forecasting till May 2023 (Covid-19 Era)

```
n.periods <- 115
for.full.raw <- predict(model.final.sarima, n.ahead = 235 + n.periods)
for.full.pred <- exp(exp(for.full.raw$pred))
for.full.upper <- exp(exp(for.full.raw$pred + 1.96 * for.full.raw$se))
for.full.lower <- exp(exp(for.full.raw$pred - 1.96 * for.full.raw$se))

for.post.raw <- predict(model.post.final.sarima, n.ahead = 23 + n.periods)
for.post.pred <- exp(exp(for.post.raw$pred))
for.post.upper <- exp(exp(for.post.raw$pred + 1.96 * for.post.raw$se))
for.post.lower <- exp(exp(for.post.raw$pred - 1.96 * for.post.raw$se))

com.pred <- 0.75 * tail(for.full.pred, n.periods) + 0.25 * tail(for.post.pred, n.periods)
com.upper <- 0.75 * tail(for.full.upper, n.periods) + 0.25 * tail(for.post.upper, n.periods)
com.lower <- 0.75 * tail(for.full.lower, n.periods) + 0.25 * tail(for.post.lower, n.periods)
```

```r
plot(ts.energy.biweekly, main = "Model SARIMA Forecasts (Covid-19 Period)",
     xlab = "Time", ylab = "Energy Consumption (MW)",
     col = "black", lwd = 2, xaxt = "n", xlim = c(2019, 2023 + 5/12))
polygon(c(time(for.full.upper), rev(time(for.full.lower))),
        c(for.full.upper, rev(for.full.lower)),
        col = adjustcolor("mistyrose", alpha.f = 0.4), border = NA)
lines(com.pred, col = "red", lwd = 2)
lines(com.upper, col = "firebrick", lty = 2)
lines(com.lower, col = "firebrick", lty = 2)
polygon(c(time(for.post.upper), rev(time(for.post.lower))),
        c(for.post.upper, rev(for.post.lower)),
        col = adjustcolor("honeydew3", alpha.f = 0.4), border = NA)
lines(for.post.pred, col = "darkgreen", lwd = 2)
lines(for.post.upper, col = "firebrick", lty = 2)
lines(for.post.lower, col = "firebrick", lty = 2)
lines(ts.energy.biweekly, col = "black", lwd = 2)
months_seq <- seq(from = as.Date("2018-01-01"), to = as.Date("2023-05-01"), by = "1 month")
month_ticks <- as.numeric(format(months_seq, "%Y")) + (as.numeric(format(months_seq, "%m"))-
1)/12
axis(1, at = month_ticks, labels = format(months_seq, "%b\n%Y"), cex.axis = 0.6, las = 1)
legend("topleft", legend = c("Observed Data", "Full Model Forecast",
                             "Post-2011 SARIMA Forecast", "Forecast Prediction Interval"),
       col = c("black", "red", "darkgreen", "firebrick"),
       lty = c(1, 1, 1, 2), lwd = c(2, 2, 2, 1), bg = "white", cex = 0.6)
grid()
```