Energy Consumption Forecast

Project

By – Aditya K Nagori

2nd May 2023

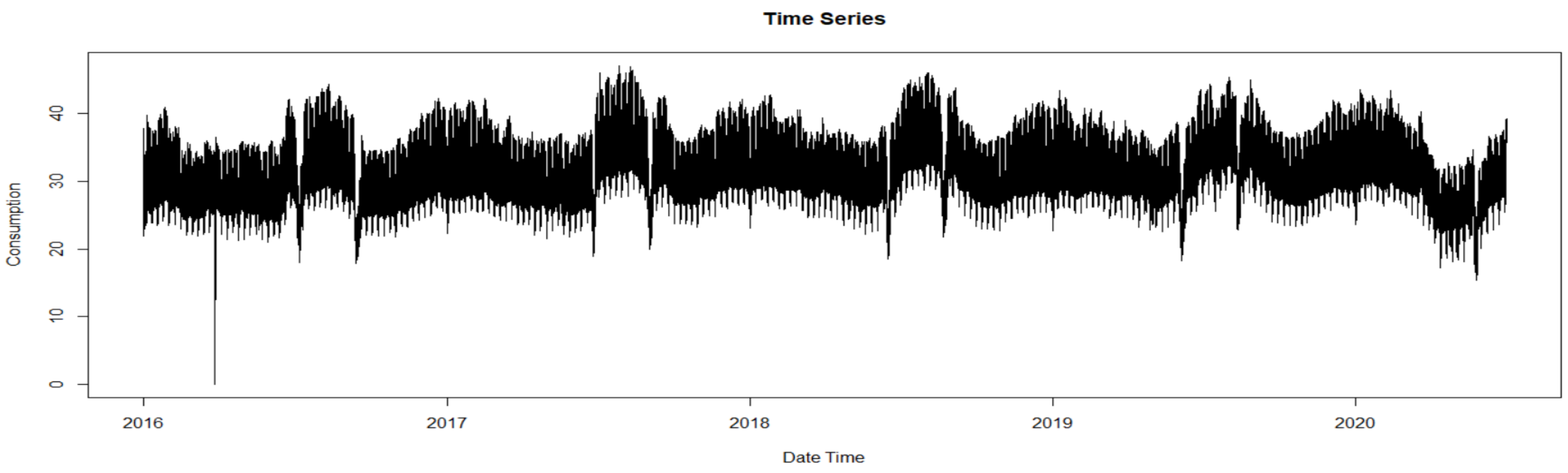
Problem Statement

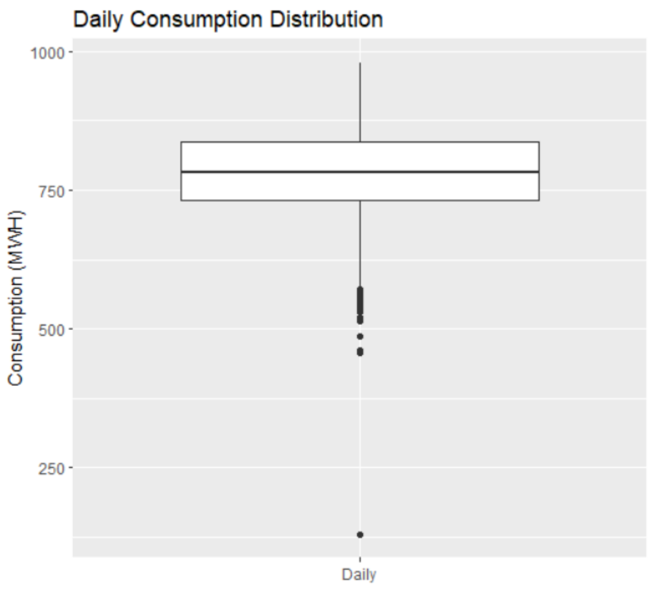
- To create a forecast of future demand in energy consumption of a power company to meet expected demand both monthly & daily.

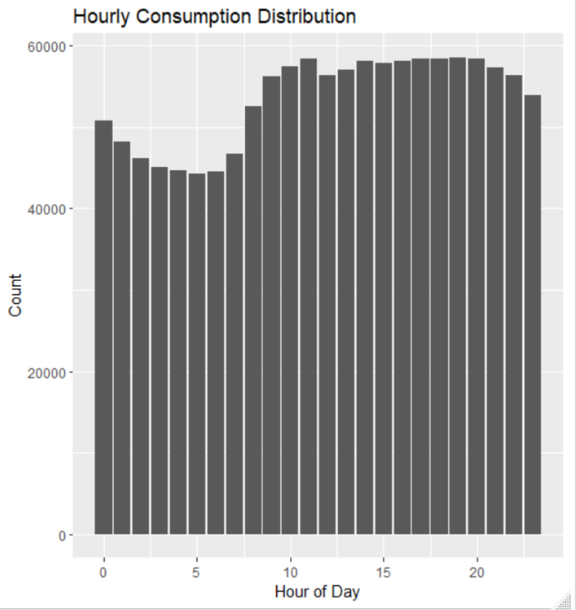
Understanding Data

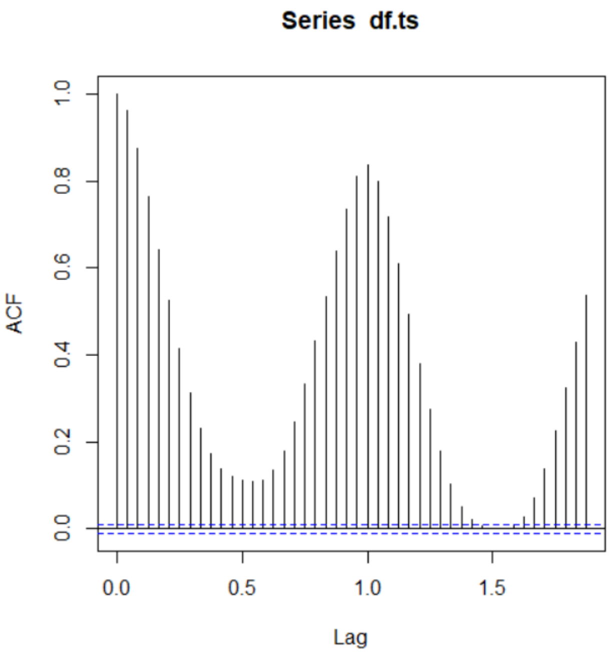
- The dataset has 39456 rows of observations and 3 variables with date, time, and consumption that show instances of consumption every hour of every day from 2015-12-31 00:00 till 2020-06-3 23:00. The date columns are converted into date time by merging them.

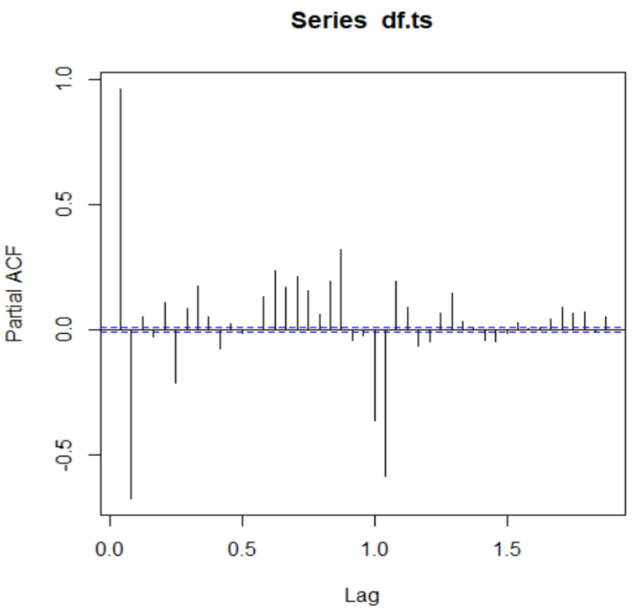
EDA Hourly

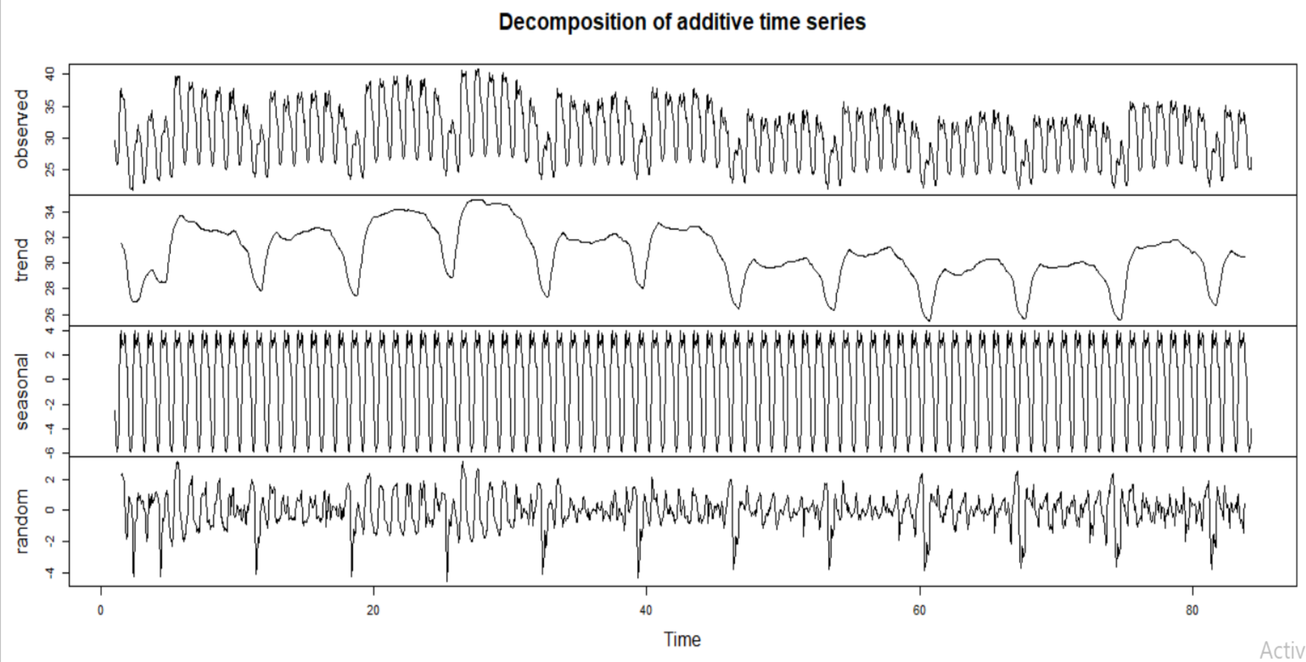

Time Series
Consumption
0
10
20
30
40
2016
2017
2018
2019
2020
Date Time

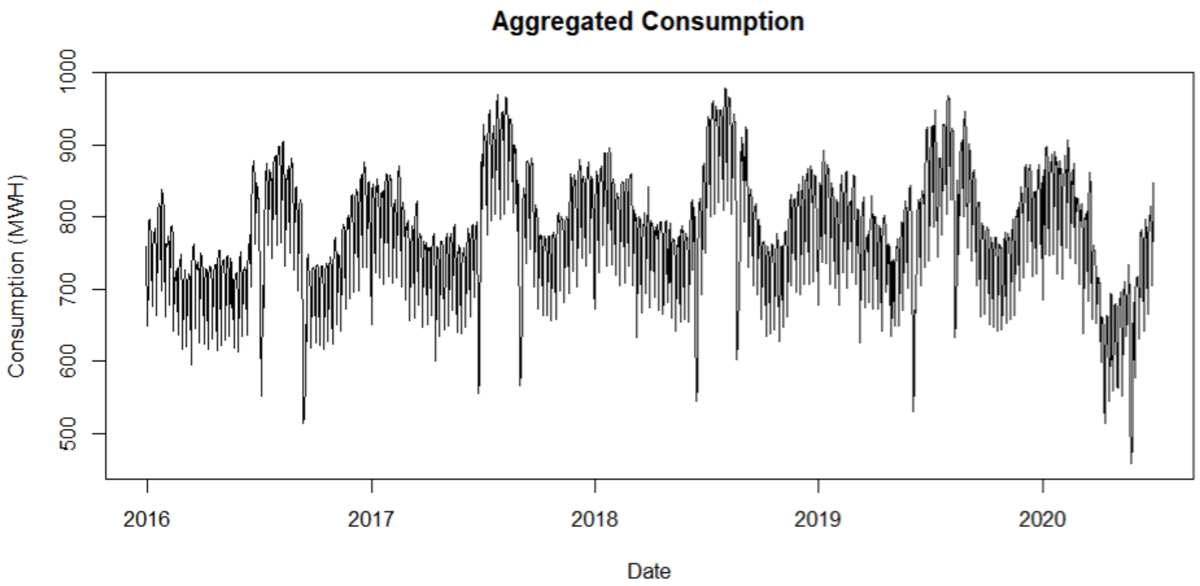

Daily Consumption Distribution
Consumption (MWH)
1000
750
500
250
Daily

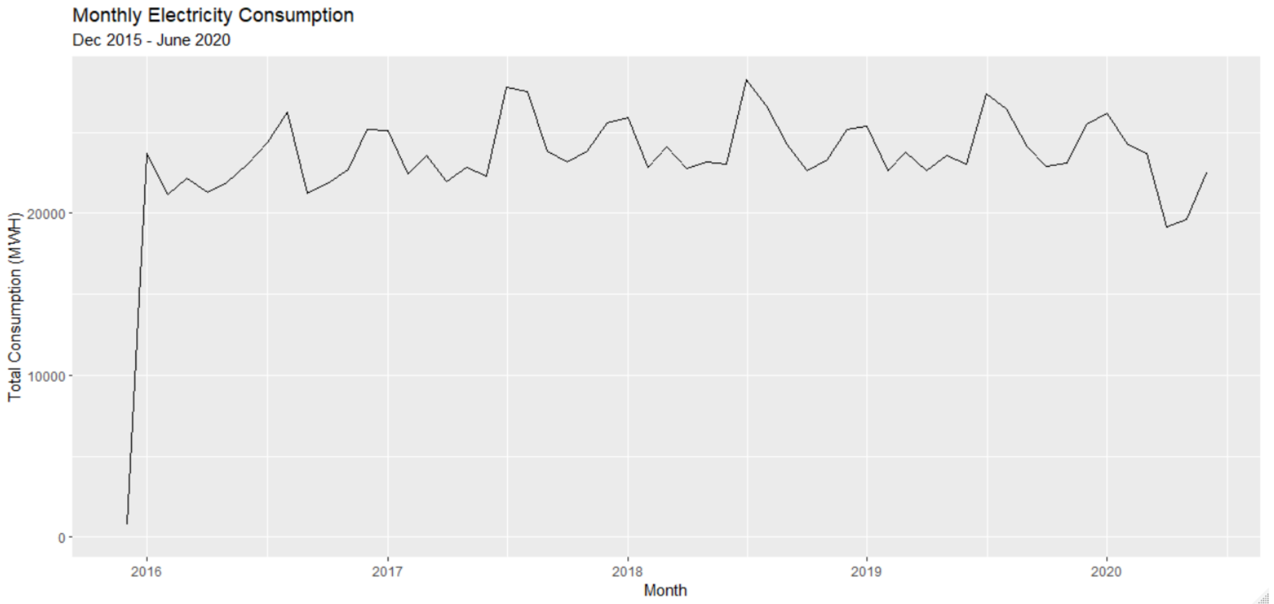

Hourly Consumption Distribution
Count
60000
40000
20000
0
0
5
10
15
20
Hour of Day

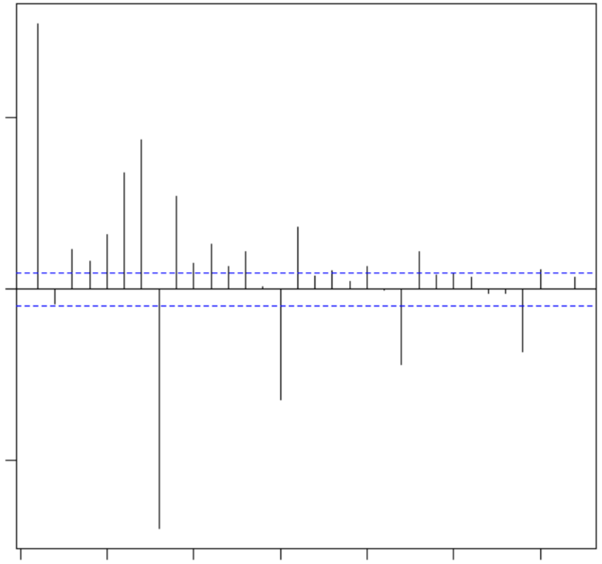

Series df.ts
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
Lag

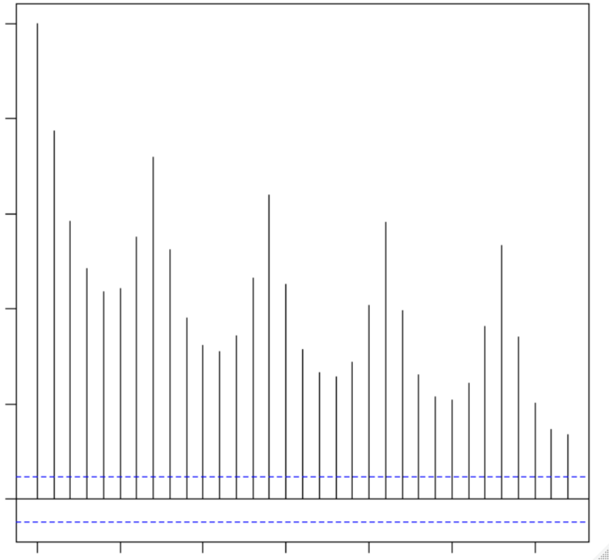

Series df.ts
Partial ACF
1.0
0.5
0.0
-0.5
0.0
0.5
1.0
1.5
Lag

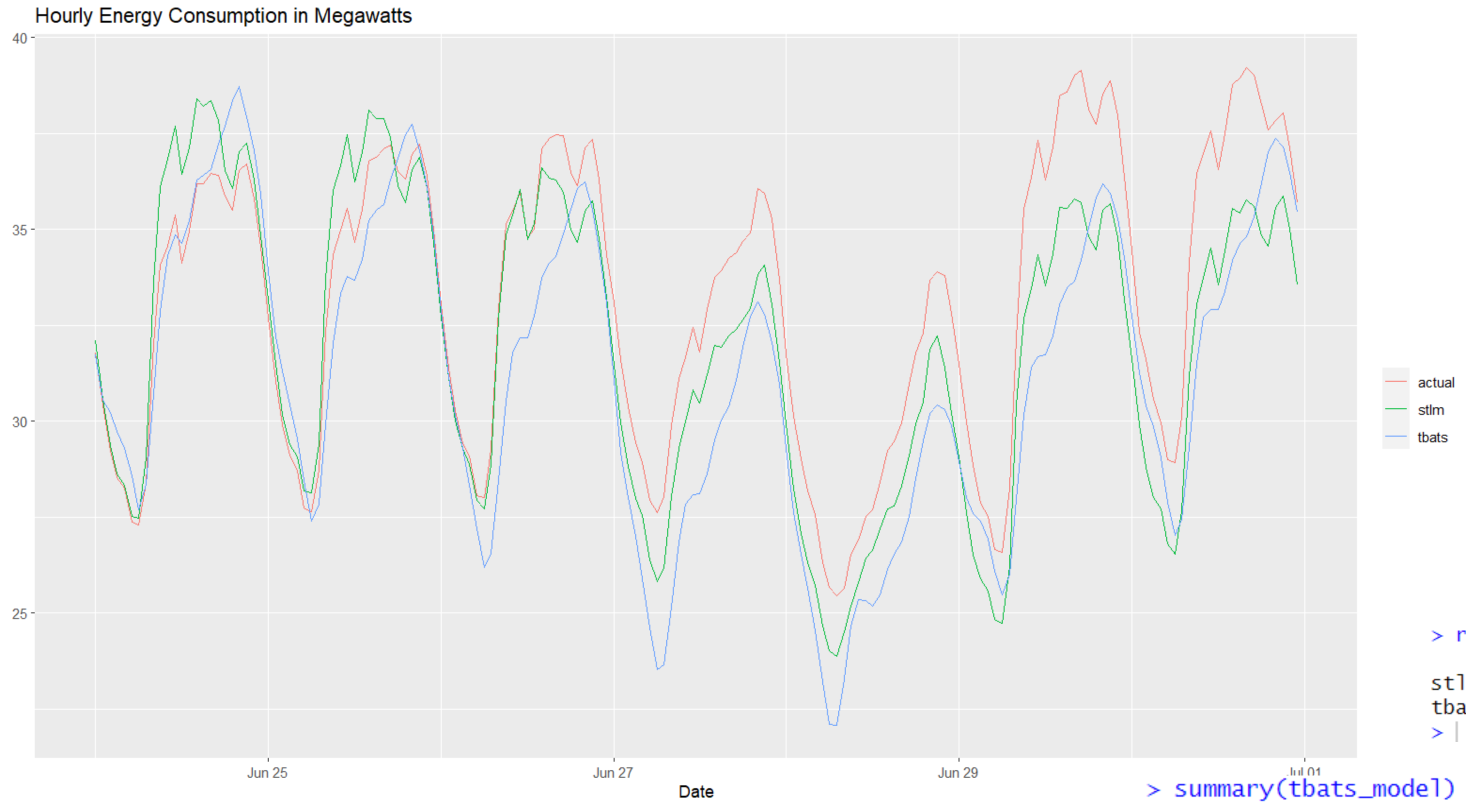

Decomposition of additive time series
observed
trend
seasonal
random
Time
0
20
40
60
80


Aggregated Consumption
Consumption (MWH)
Date
2016
2017
2018
2019
2020


Monthly Electricity Consumption
Dec 2015 - June 2020
Total Consumption (MWH)
Month
2016
2017
2018
2019
2020





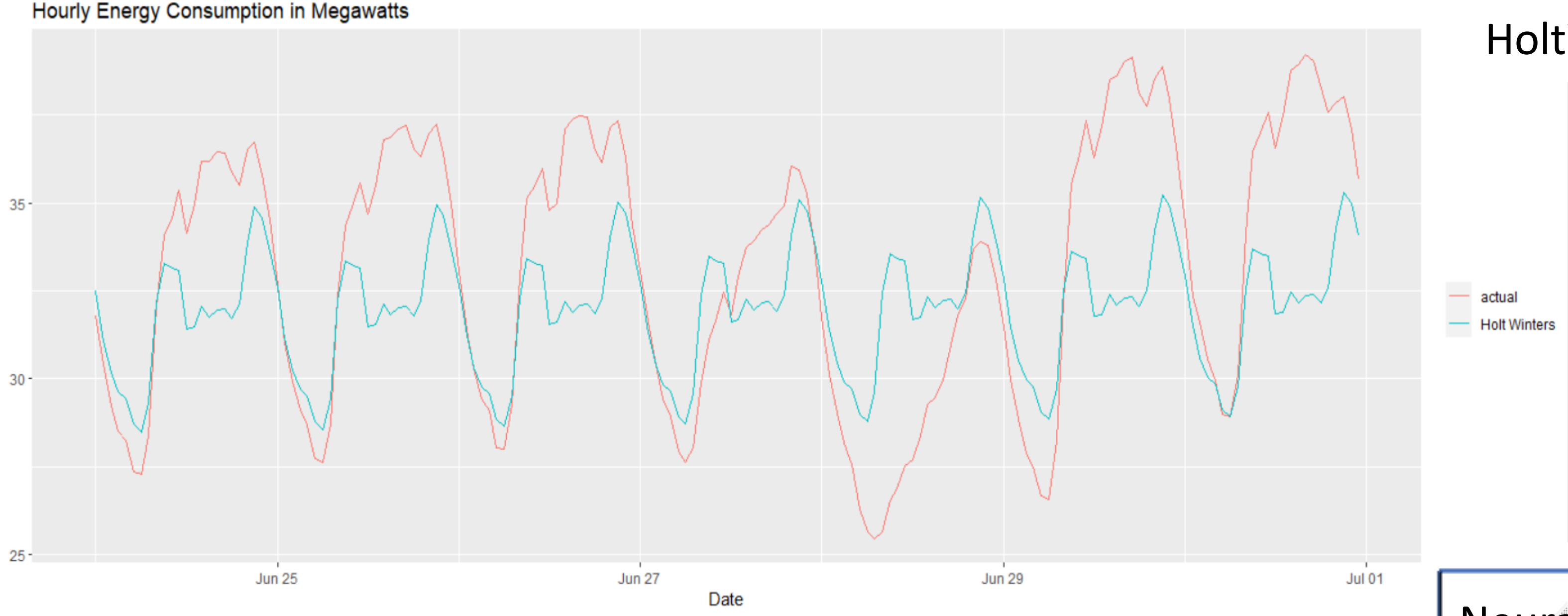


MULTIPLE SEASONALITY STLM (SEASONAL-TREND DECOMPOSITION USING LOESS) & TBATS(TRIGONOMETRIC SEASONALITY, BOX-COX TRANSFORMATION, ARMA ERRORS, TREND, & SEASONAL COMPONENTS) MODELS FOR HOURLY FORECASTING

```
> result
                  ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
stlm_model  1.140749 1.906523 1.639140 3.430968 4.885455 0.9810637  1.482507
tbats_model 1.950531 2.712279 2.268078 5.861349 6.833804 0.9327302  2.177417
>
```

```
> summary(stlm_model)

Forecast method: STL +  ETS(M,Ad,N)

Model Information:
ETS(M,Ad,N)

Call:
 ets(y = x, model = etsmodel, allow.multiplicative.trend = allow.multiplicative.trend)

  Smoothing parameters:
    alpha = 0.9998
    beta  = 0.6587
    phi   = 0.8

  Initial states:
    l = 3.3491
    b = 0.0043

  sigma:  0.0024

      AIC      AICc       BIC
-4158.535 -4158.493 -4124.882

Error measures:
                        ME      RMSE       MAE         MPE      MAPE       MASE        ACF1
Training set -0.001195875 0.1997284 0.1488458 -0.00173225 0.5580831 0.06893019 -0.04521683
```

```
> summary(tbats_model)

Forecast method: TBATS(1, {5,3}, 0.8, {<24,5>, <168,6>})

Model Information:
TBATS(1, {5,3}, 0.8, {<24,5>, <168,6>})

Call: tbats(y = ., use.box.cox = FALSE, use.trend = TRUE, use.damped.trend = TRUE)

Parameters
  Alpha: 0.3088479
  Beta: -0.05967678
  Damping Parameter: 0.800007
  Gamma-1 Values: -0.0009790418 0.0005416959
  Gamma-2 Values: 1.925414e-05 1.976599e-05
  AR coefficients: -0.606977 -0.090789 0.604626 0.249872 -0.259671
  MA coefficients: 1.758347 1.718667 0.8188

Sigma: 0.01981989
AIC: -376.2689

Error measures:
                       ME       RMSE        MAE          MPE      MAPE      MASE        ACF1
Training set 0.0001372653 0.01981989 0.01554381 0.0002114554 0.4749041 0.1843784 -0.03235324
```

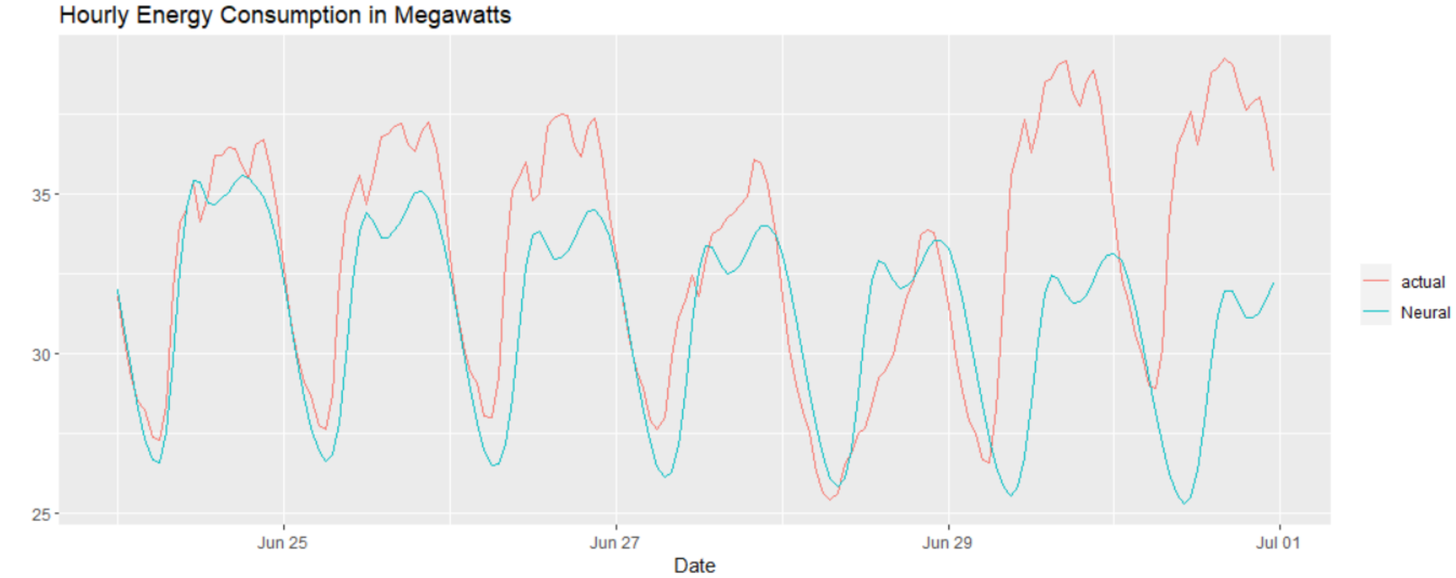


HoltWinter's MODEL FOR HOURLY FORECASTING

```
#Holtwinters
test_hw = ts(tail(df$`Consumption (MWH)`,24*7), frequency = 24)
train_hw = ts(head(df$`Consumption (MWH)`,39288), frequency = 24)
model_hw = HoltWinters(train_hw)
pred_hw = forecast(model_hw, 24*7)
ggplot() +
  geom_line(aes(x = (df$DateTime %>% tail(24*7)),
                y = (df$`Consumption (MWH)` %>% tail(24*7)), colour = "actual"))+
  geom_line(aes(x = (df$DateTime %>% tail(24*7)), y =(pred_hw$mean),
                colour = "Holt Winters "))+
  scale_y_continuous(labels = comma)+
  labs(
    title = "Hourly Energy Consumption in Megawatts",
    x = "Date",
    y = "",
    colour = ""
  )
p = as.data.frame(pred_hw$mean)
o=as.data.frame(tail(df$`Consumption (MWH)`,24*7))
mae(p$x,o$`tail(df$\`Consumption (MWH)\`, 24 * 7)`)
rmse(p$x,o$`tail(df$\`Consumption (MWH)\`, 24 * 7)`)
```

```
Forecast method: HoltWinters

Model Information:
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = train_hw)

Smoothing parameters:
 alpha: 0.8459884
 beta : 0.001127422
 gamma: 1
```

```
Error measures:
                      ME      RMSE       MAE  MPE MAPE      MASE      ACF1
Training set 0.006540962 0.6900649 0.4720893 -Inf  Inf 0.2613972 0.5591839
```

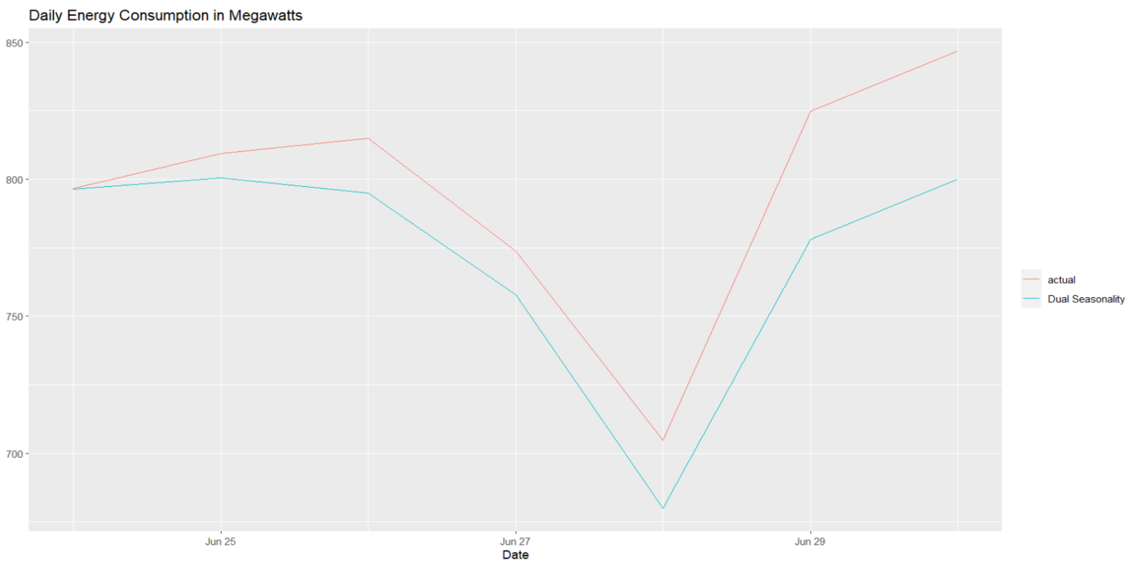


Neural network MODEL FOR HOURLY FORECASTING

```
#neuralnet
library(neuralnet)
model <- nnetar(train_hw, repeats = 20, p=11, P = 1, size =7)
summary(model$model[[1]])#weights first train
summary(model$model[[2]])#weights second train
pred_nn = forecast(model,24*7)


ggplot() +
  geom_line(aes(x = (df$DateTime %>% tail(24*7)),
                y = (df$`Consumption (MWH)` %>% tail(24*7)), colour = "actual"))+
  geom_line(aes(x = (df$DateTime %>% tail(24*7)), y =(pred_nn$mean),
                colour = "Neural"))+
  scale_y_continuous(labels = comma)+
  labs(
    title = "Hourly Energy Consumption in Megawatts",
    x = "Date",
    y = "",
    colour = ""
  )
```

```
> summary(pred_nn)

Forecast method: NNAR(11,1,7)[24]

Model Information:

Average of 20 networks, each of which is
a 12-7-1 network with 99 weights
options were - linear output units

Error measures:
                       ME      RMSE       MAE  MPE MAPE      MASE       ACF1
Training set 0.0006381286 0.6251858 0.4578825 -Inf  Inf 0.2535309 0.01061947
```

TBATS(TRIGONOMETRIC SEASONALITY, BOX-COX TRANSFORMATION, ARMA ERRORS, TREND, & SEASONAL COMPONENTS) MODEL FOR DAILY FORECASTING




```
#Duol Sesonlity
fit <- tbats(df.train.ts, seasonal.periods = c(365, 7))
# Make forecasts
h <- length(df.valid.ts)
fc <- forecast(fit, h = h)
ggplot() +
  geom_line(aes(x = (agg_data$Date %>% tail(7)),
                y = (agg_data$`Consumption (MWH)` %>% tail(7))
                , colour = "actual"))+
  geom_line(aes(x = (agg_data$Date %>% tail(7)),
                y = fc$mean,    colour = "tbats"))+
  scale_y_continuous(labels = comma)+
  labs(
    title = "Daily Energy Consumption in Megawatts",
    x = "Date",
    y = "",
    colour = ""
  )
```

```
Forecast method: TBATS(1, {1,5}, -, {<7,3>, <365,3>})

Model Information:
TBATS(1, {1,5}, -, {<7,3>, <365,3>})

Call: tbats(y = df.train.ts, seasonal.periods = c(365, 7))

Parameters
  Alpha: 0.01974141
  Gamma-1 Values: -7.914104e-05 2.743492e-05
  Gamma-2 Values: -0.0001207815 2.138013e-05
  AR coefficients: 0.415948
  MA coefficients: 0.675938 0.461709 0.360958 0.205915 0.101218
```

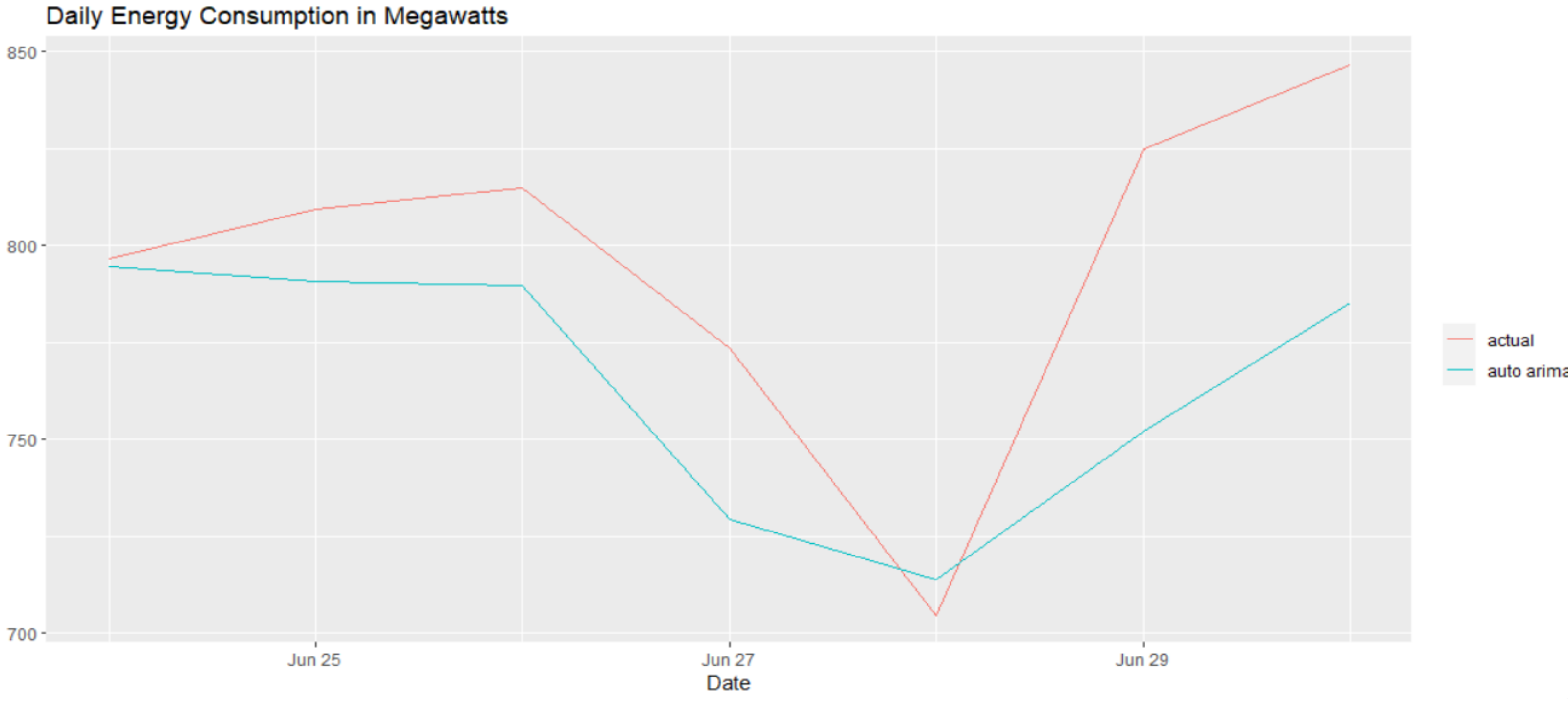
```
Sigma: 22.75495
AIC: 22405.35

Error measures:
                    ME     RMSE      MAE         MPE     MAPE      MASE
Training set 0.1925264 22.75495 12.66482 -0.06817274 1.717797 0.2036847
                     ACF1
Training set -0.001785213
```

ARIMA MODEL FOR DAILY FORECASTING

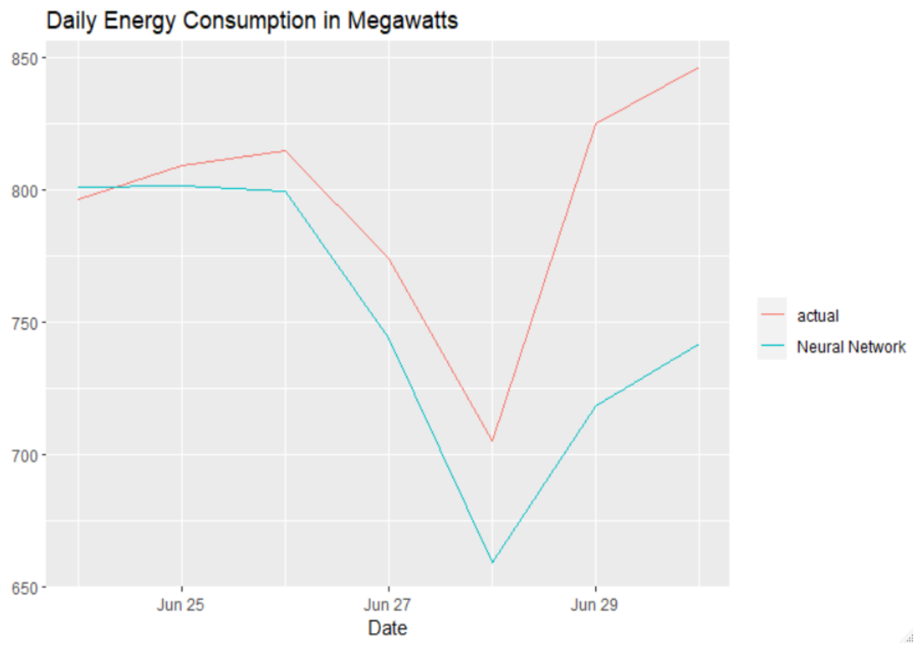
```
#auto arima
arima_model <- auto.arima(df.train.ts)
summary(arima_model)
forecast_values <- forecast(arima_model, h = length(df.valid.ts))
# Combine the actual and forecasted values into a single data frame
plot_data <- data.frame(actual = test_data, predicted = forecast_values$mean)

# Create a line plot
ggplot() +
  geom_line(aes(x = (agg_data$Date %>% tail(7)),
                y = (agg_data$`Consumption (MWH)` %>% tail(7))
                , colour = "actual"))+
  geom_line(aes(x = (agg_data$Date %>% tail(7)),
                y = forecast_values$mean,    colour = "auto arima"))+
  scale_y_continuous(labels = comma)+
  labs(
    title = "Daily Energy Consumption in Megawatts",
    x = "Date",
    y = "",
    colour = ""
  )
```

```
> summary(arima_model)
Series: df.train.ts
ARIMA(5,1,2)

Coefficients:
         ar1      ar2      ar3      ar4      ar5      ma1     ma2
      0.0879  -0.6857  -0.2788  -0.2765  -0.5349  -0.4197  0.6331
s.e.  0.0370   0.0221   0.0268   0.0209   0.0260   0.0478  0.0197

sigma^2 = 1721:  log likelihood = -8414.29
AIC=16844.58   AICc=16844.67   BIC=16887.78

Training set error measures:
                     ME     RMSE      MAE        MPE     MAPE      MASE        ACF1
Training set 0.02261221 41.38865 29.68606 -0.2523777 3.982399 0.4774328 -0.05679296
```

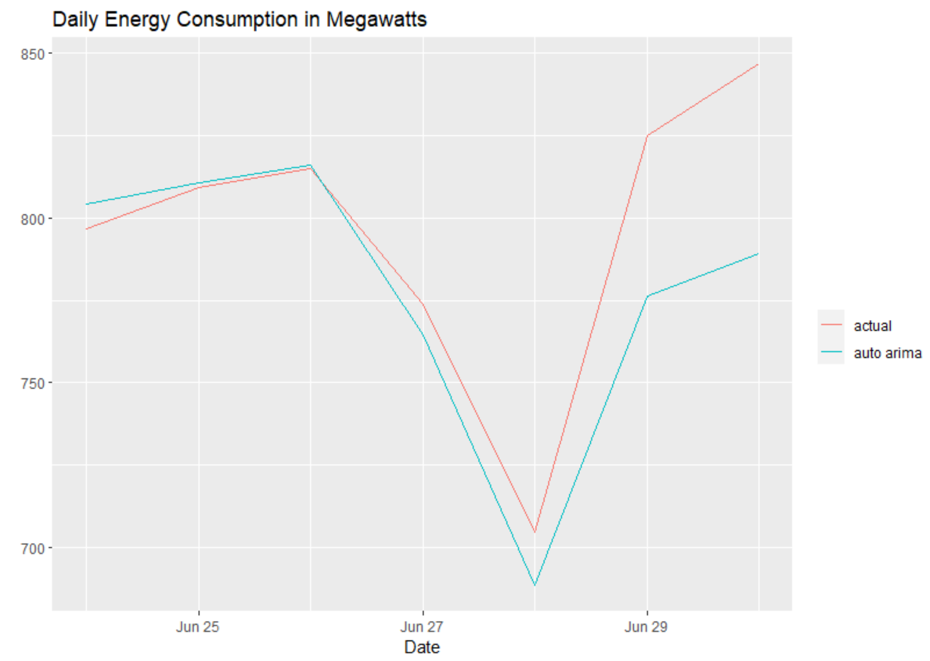
```
> plot_data
    actual predicted
1 796.4970  794.5751
2 809.2957  790.8831
3 814.9564  789.7461
4 773.7083  729.4855
5 704.7139  714.0783
6 824.8187  752.3205
7 846.6656  785.3386
```



NEURAL NETWORK MODEL FOR DAILY FORECASTING



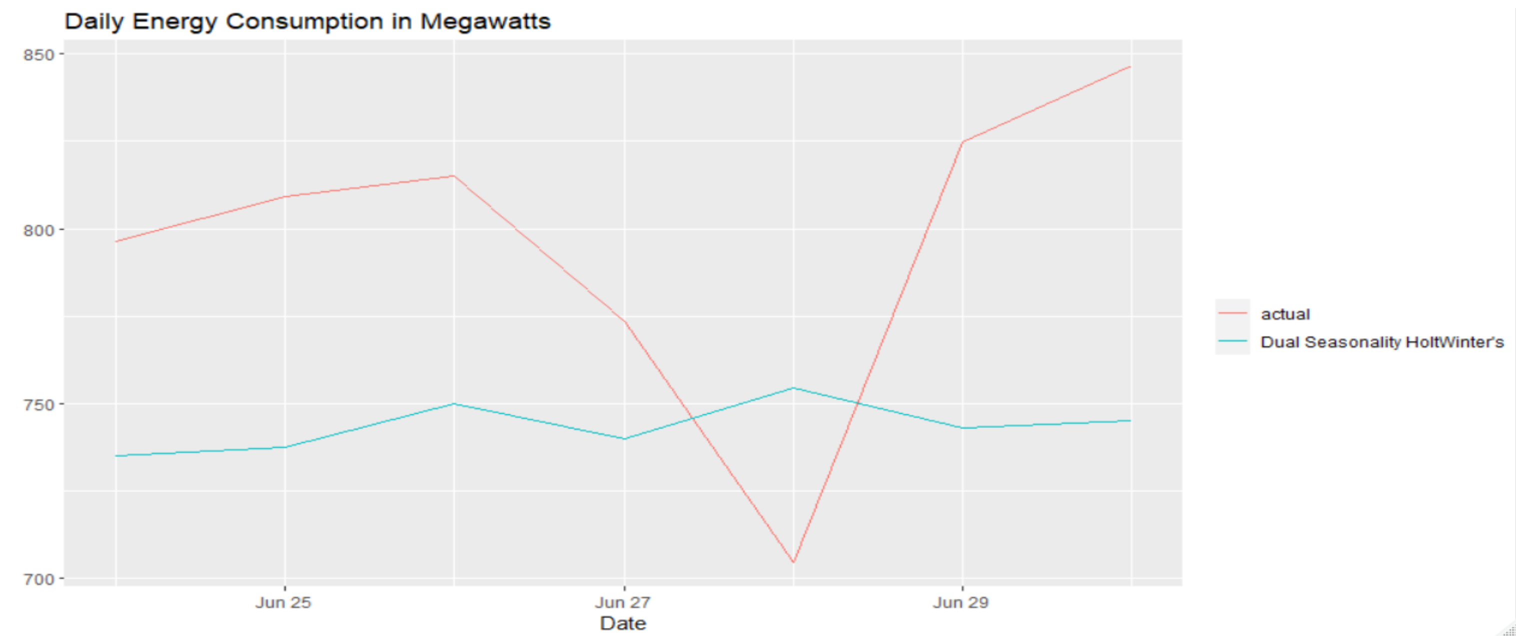

```
> summary(nn_pred)

Forecast method: NNAR(32,1,20)[365]

Model Information:

Average of 20 networks, each of which is
a 33-20-1 network with 701 weights
options were - linear output units

Error measures:
                     ME     RMSE      MAE         MPE      MAPE      MASE
Training set 0.01539986 11.37774 6.811858 -0.03775881 0.8957066 0.1095532
                   ACF1
Training set 0.05303482
```



DOUBLE SEASONALITY HOLTWINTER'S MODEL FOR DAILY FORECASTING

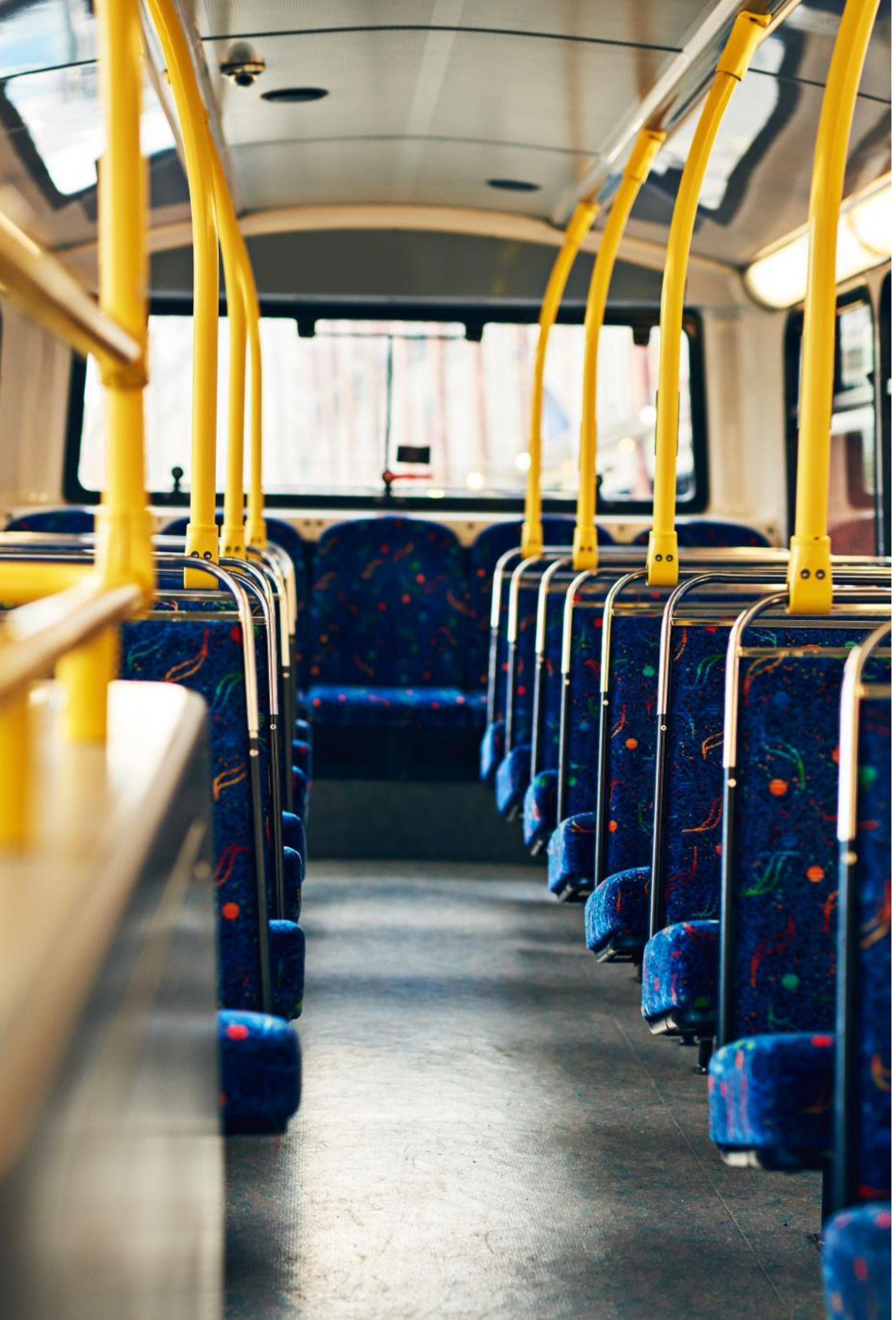
```r
# Split the data into training and testing sets
train_data <- df.train.ts
test_data <- df.valid.ts
# Fit a dual seasonality model to the training data
model_ds <- dshw(train_data, biasadj = TRUE, period1 = 30, period2 = 60)
summary(model_ds)
# Forecast the test data
predictions <- forecast(model_ds, 7)
x1=as.data.frame(predictions$fitted)
df_data <- as.data.frame(train_data)
names(df_data) <- "x"
df3 <- merge(df_data, x1, sort = FALSE, all=TRUE )
ggplot() +
  geom_line(aes(x = (agg_data$Date %>% tail(7)),
                y = (agg_data$`Consumption (MWH)` %>% tail(7))
                , colour = "actual"))+
  geom_line(aes(x = (agg_data$Date %>% tail(7)),
                y = head(predictions$mean, 7),   colour = "Neural Network"))+
  scale_y_continuous(labels = comma)+
  labs(
    title = "Daily Energy Consumption in Megawatts",
    x = "Date",
    y = "",
    colour = ""
  )
```

```
> accuracy(model_ds)
                   ME     RMSE      MAE        MPE     MAPE      MASE      ACF1
Training set 5.510798 66.72409 50.70614 0.06211221 6.770746 0.4974401 0.4904791
>
```



Major Findings

- Energy consumption from the forecast tells us that consumption always goes up after mid-year i.e. requirement is always high in the months of July, August, and September.
- The data we have is almost normal, it is not right or left-skewed.
- It is also seen that the 10 AM energy usage is lost the same for the rest of the hours.
- High autocorrelation at the lags indicates that data is nonstationary and needs to be addressed.
- Every year’s data pattern indicates the presence of level, and seasonality in the data there is no constant trend. But there is dual seasonality.
- The forecasted data indicates that the demand continues the same pattern and numbers every year.



Recommendations

- The production should be increased when the months of July, August, and September are near.
- The production should be decreased according to the graph where it is indicating low usage.
- The performance of the resources is to be tracked to make sure Energy production is well equipped for the changing demand.
- Develop a contingency plan to support sudden spikes in the consumption of energy.