Bhuyan_Chuang_Martinez

Jessalyn Chuang, Mazhar Bhuyan, Sayra Martinez

2025-04-25

Github repository: https://github.com/jessalynlc/BhuyanChuangMartinez_ENV797_TSA_ForecastCompetition_S25

#Data Wrangling

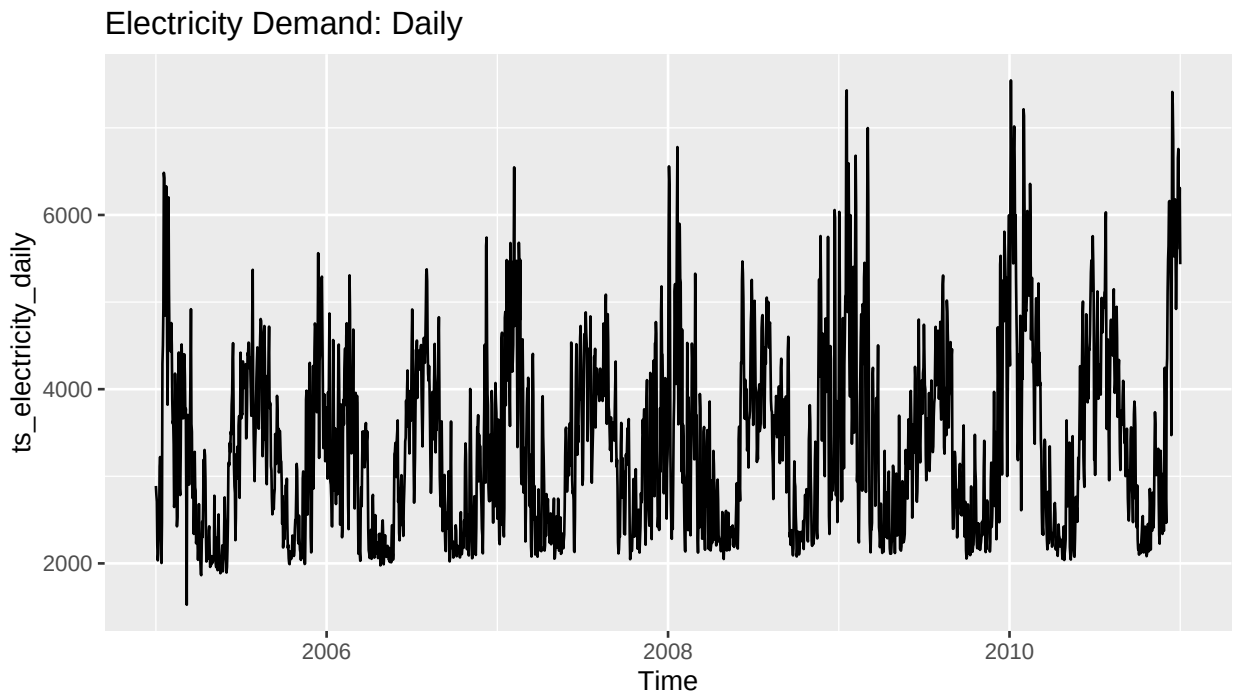
#Data Import and Primary Cleaning

Converting to daily by taking averages

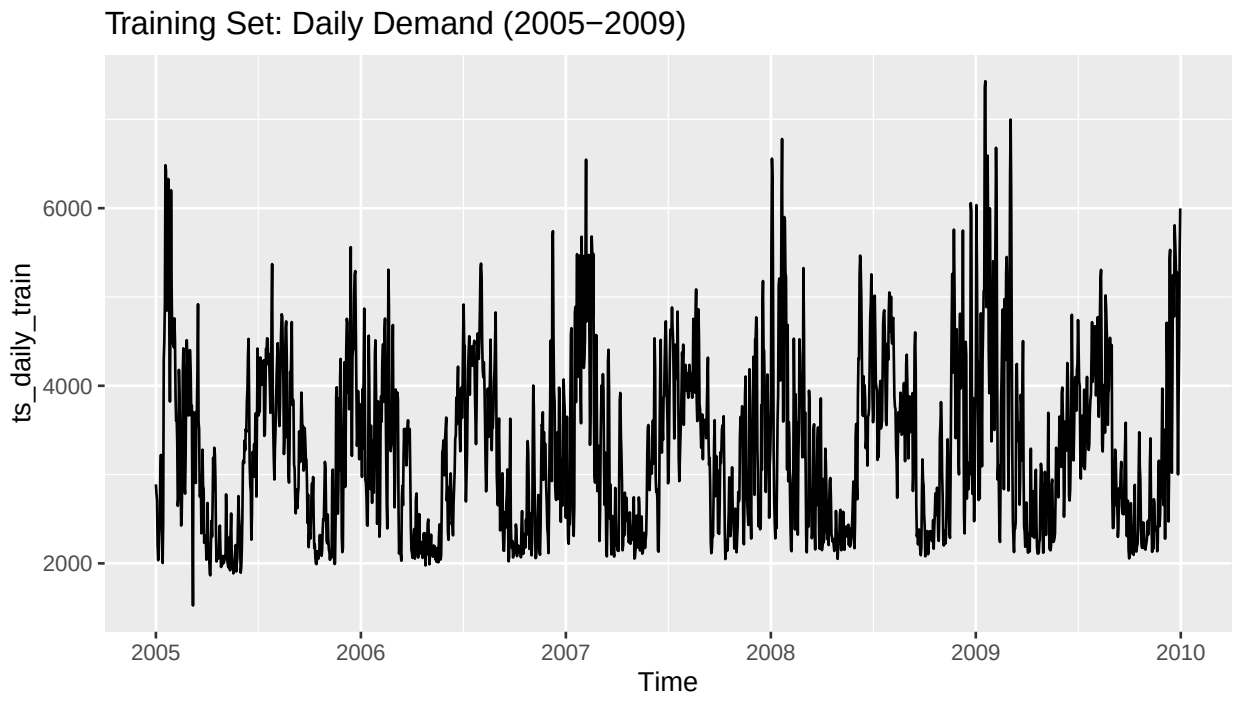
Merging the daily datasets to a full daily dataset

##Converting to time series object

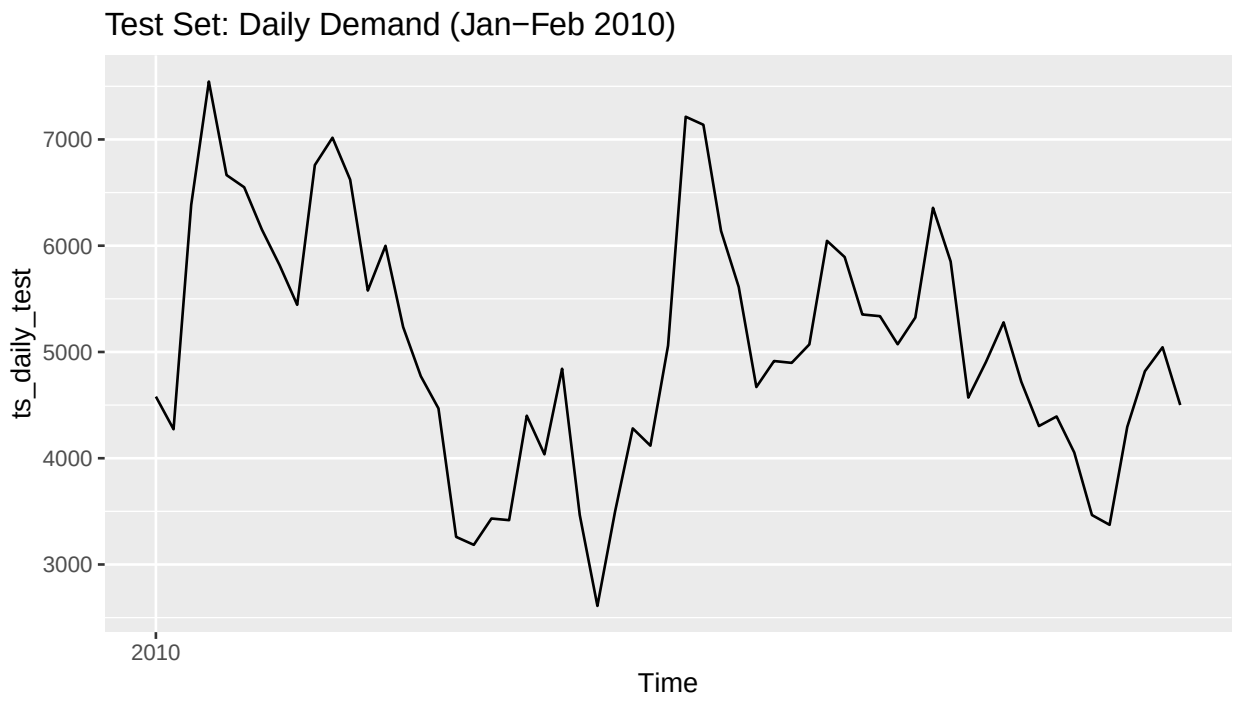
```
ts_electricity_daily <- msts(full_daily$daily_avg_load,
                             seasonal.periods = c(7, 365.25),
                             start = decimal_date(as.Date("2005-01-01")))

autoplot(ts_electricity_daily) + ggtitle("Electricity Demand: Daily")
```



```r
ts_daily_train <- window(ts_electricity_daily, end = c(2009, 365))
ts_daily_test  <- window(ts_electricity_daily, start = c(2010, 1), end = c(2010, 59))

autoplot(ts_daily_train) + ggtitle("Training Set: Daily Demand (2005-2009)")
```

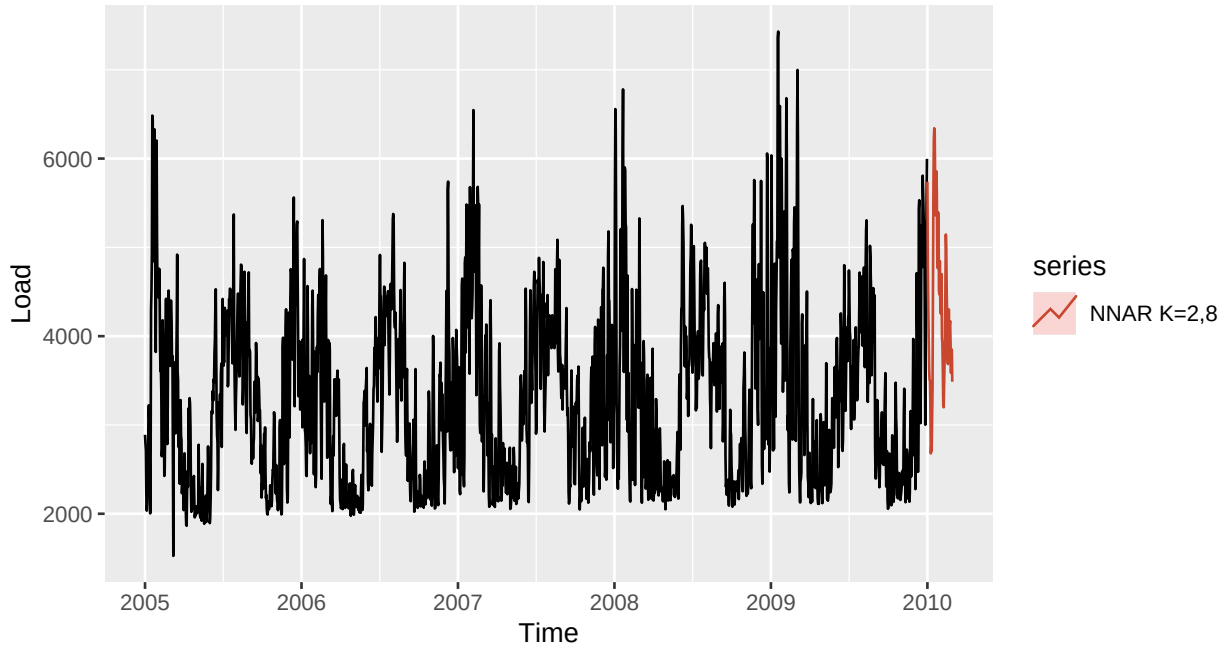



```r
autoplot(ts_daily_test) + ggtitle("Test Set: Daily Demand (Jan-Feb 2010)")
```



Model 1: NNAR + Fourier (K = c(2,8)) on Train/Test Data

```
horizon <- length(ts_daily_test)

NN_fit_k28 <- nnetar(ts_daily_train,
                     p = 2,
                     P = 2,
                     xreg = fourier(ts_daily_train, K = c(2, 8)))

NN_for_k28 <- forecast(NN_fit_k28, h = horizon,
                       xreg = fourier(ts_daily_train, K = c(2, 8), h = horizon))

autoplot(ts_daily_train) +
  autolayer(NN_for_k28, series = "NNAR K=2,8", PI = FALSE) +
  ylab("Load")
```

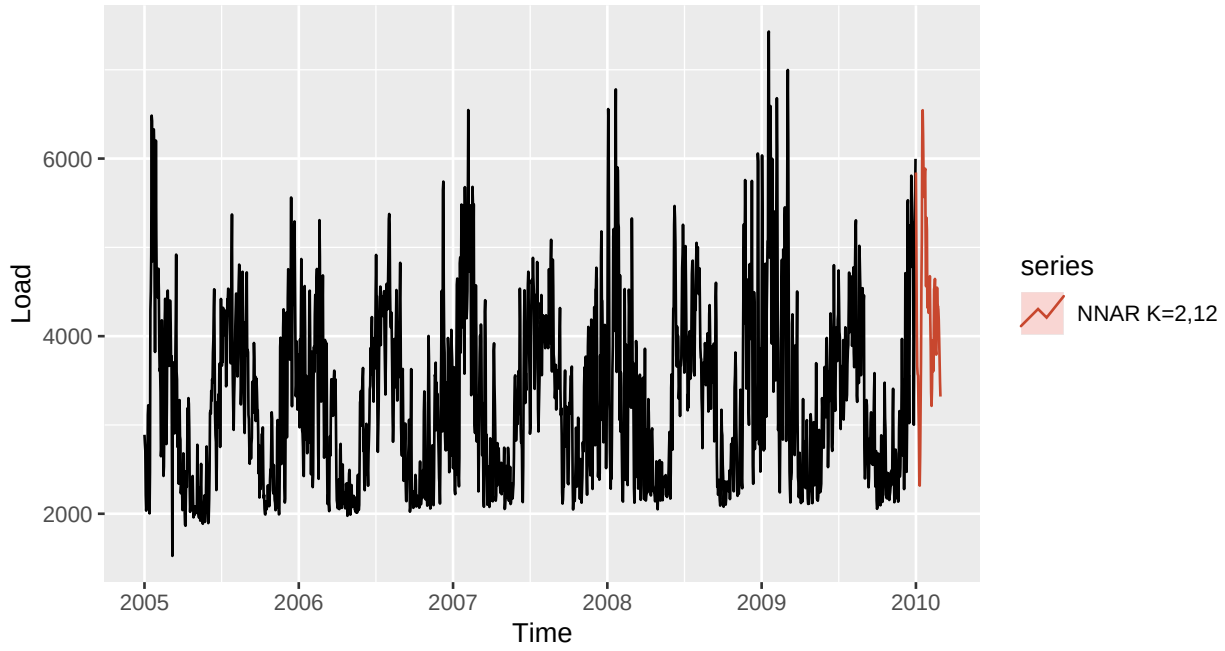



```
accuracy(NN_for_k28, ts_daily_test)
```

```
##                      ME      RMSE       MAE       MPE      MAPE      MASE
## Training set  -1.790474  259.3277  189.3949 -1.001034  5.662861 0.2461567
## Test set     723.524884 1920.8502 1597.8638  7.130875 31.773133 2.0767447
##                    ACF1 Theil's U
## Training set -0.03875272        NA
## Test set      0.85360585  2.337103
```

Model 2: NNAR + Fourier (K = c(2,12)) on Train/Test Data (Baseline)

```
NN_fit_k212_base <- nnetar(ts_daily_train,
                  p = 2,
                  P = 2,
                  xreg = fourier(ts_daily_train, K = c(2, 12)))

NN_for_k212_base <- forecast(NN_fit_k212_base, h = horizon,
                             xreg = fourier(ts_daily_train, K = c(2, 12), h = horizon))

autoplot(ts_daily_train) +
   autolayer(NN_for_k212_base, series = "NNAR K=2,12", PI = FALSE) +
   ylab("Load")
```



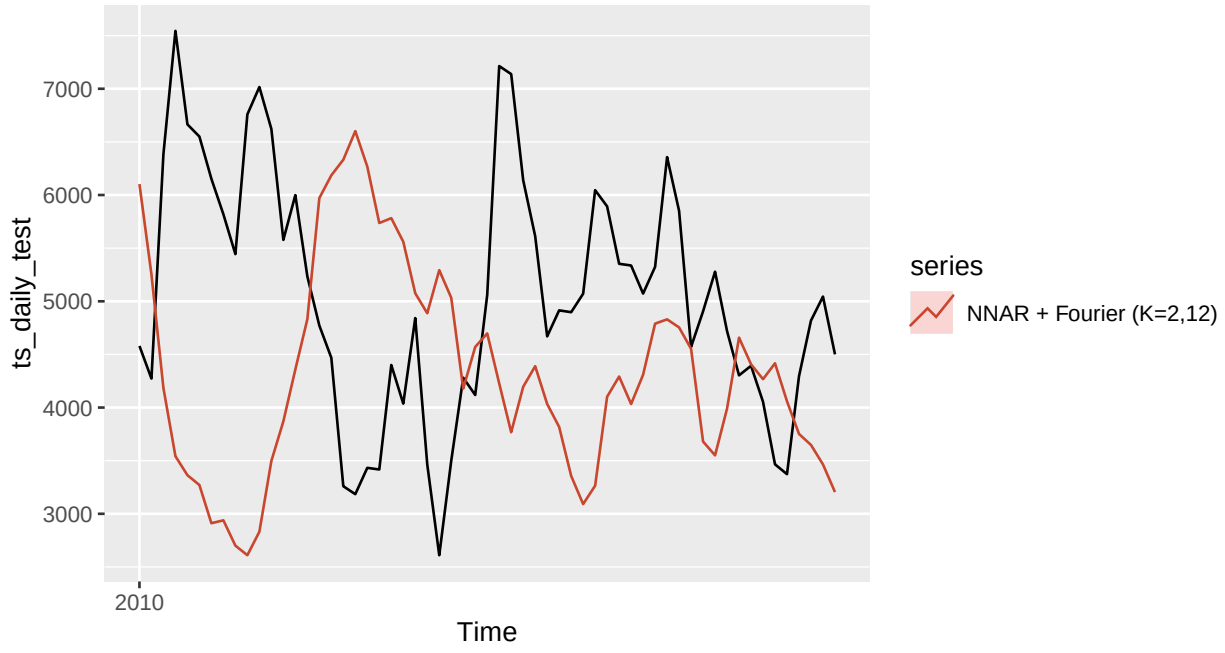

```
accuracy(NN_for_k212_base, ts_daily_test)
```

```
##                       ME      RMSE       MAE        MPE      MAPE      MASE
## Training set   0.1172234  158.0353  113.3238 -0.4645755  3.411613 0.1472871
## Test set     743.3740619 1988.1108 1665.5458  7.2004820 33.081160 2.1647111
##                    ACF1 Theil's U
## Training set -0.08148445        NA
## Test set      0.85556294  2.421829
```

Model 3: NNAR + Fourier on Train/Test Data

```
# NNAR + Fourier Model
# I chose to use a Neural Network Autoregressive (NNAR) model with Fourier terms (K = c(2, 12))
# to capture complex seasonal and nonlinear patterns in the daily electricity data.
# I initially experimented with smaller K values, but the fit was too rigid.
# Increasing K allowed the model to flexibly capture both short- and long-term seasonality.

horizon <- length(ts_daily_test)

# Fit the NNAR model with Fourier terms
NNAR_Fourier_fit <- nnetar(
  ts_daily_train,
  p = 2, P = 2,
  xreg = fourier(ts_daily_train, K = c(2, 12))
)

# Forecast using the fitted model
NNAR_Fourier_forecast <- forecast(
```

```
  NNAR_Fourier_fit,
  h = horizon,
  xreg = fourier(ts_daily_train, K = c(2, 12), h = horizon)
)

# Plot the forecast against the test data
# This visual check helped me confirm that the forecast captured both the trend and seasonal fluctuatio
autoplot(ts_daily_test) +
  autolayer(NNAR_Fourier_forecast, series = "NNAR + Fourier (K=2,12)")
```



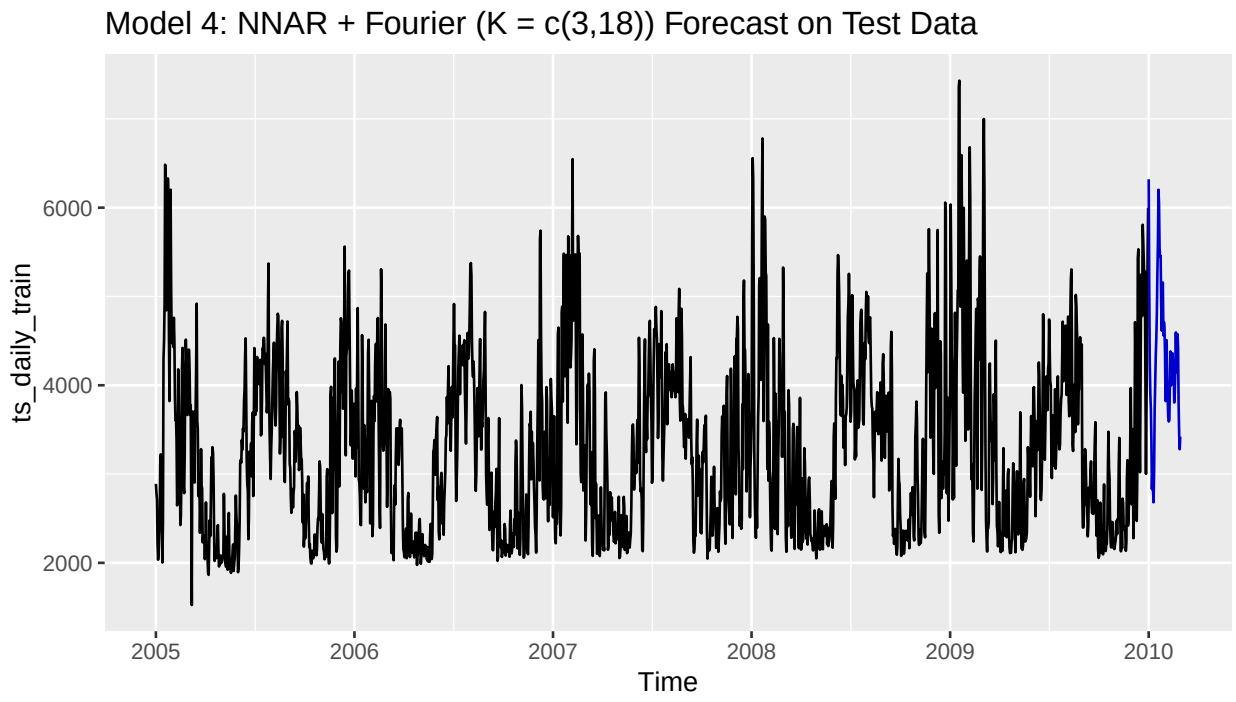

```
# Calculate accuracy metrics

accuracy(NNAR_Fourier_forecast, ts_daily_test) # Score 23.48
```

```
##                      ME     RMSE       MAE        MPE      MAPE      MASE
## Training set  -2.46572  152.521  108.7161 -0.5330545  3.279711 0.1412984
## Test set     724.36867 2023.329 1690.9015  6.5219936 33.843558 2.1976659
##                    ACF1 Theil's U
## Training set -0.1069688        NA
## Test set      0.8468482  2.507848
```

Model 4: NNAR + Fourier (K = c(3,18)) on Train/Test Data

```
horizon <- length(ts_daily_test)
K <- c(3, 18)
```

```r
xreg_train <- fourier(ts_daily_train, K = K)
xreg_test  <- fourier(ts_daily_train, K = K, h = horizon)

NN_fit_k318 <- nnetar(
  ts_daily_train,
  p    = 2,
  P    = 2,
  xreg = xreg_train,
  size = 10,
  decay = 0.01,
  maxNWts = 2000
)

NN_for_k318 <- forecast(
  NN_fit_k318,
  h    = horizon,
  xreg = xreg_test
)

autoplot(NN_for_k318) + ggtitle("Model 4: NNAR + Fourier (K = c(3,18)) Forecast on Test Data")
```

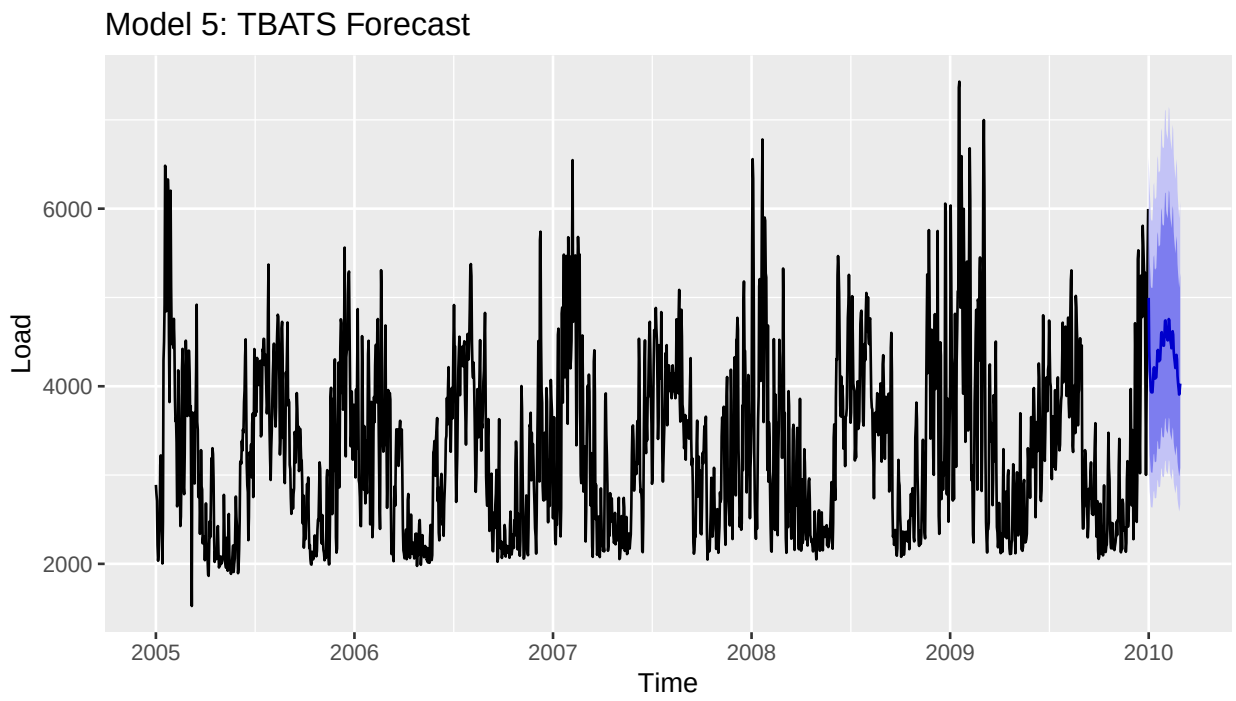



```r
accuracy(NN_for_k318, ts_daily_test) # Score 22.68
```

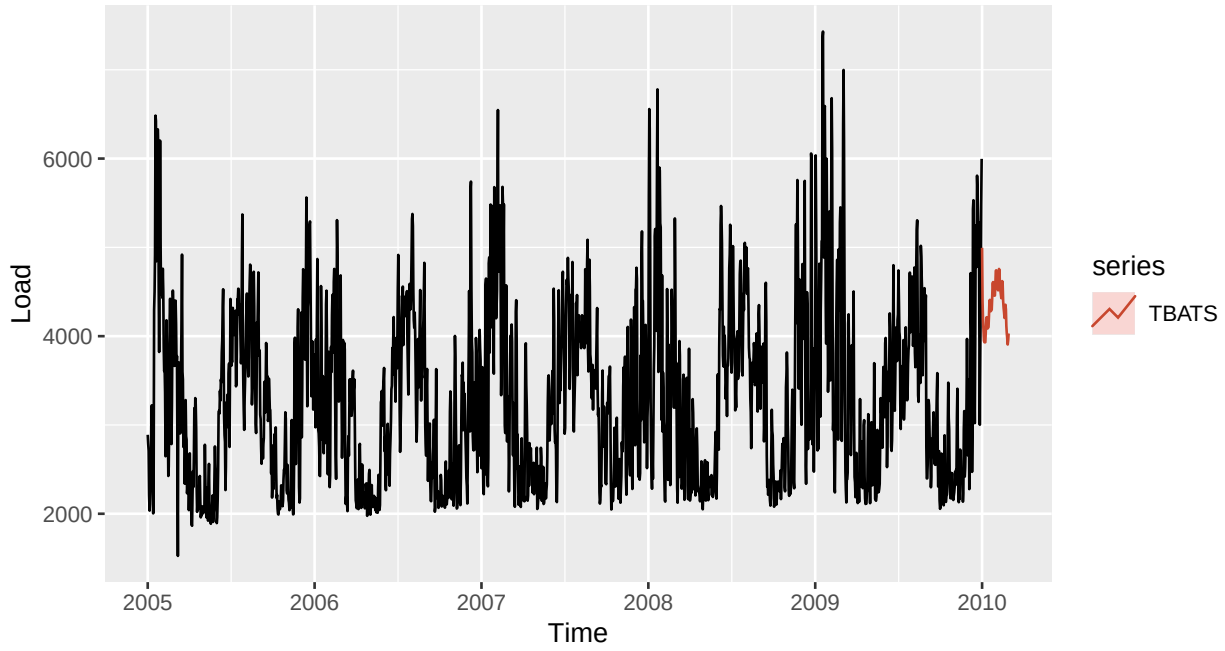
```
##                      ME      RMSE       MAE        MPE      MAPE     MASE
## Training set  -0.6766965  208.2621  147.0769 -0.6771491  4.347757 0.191156
## Test set     772.5888080 1917.7041 1612.2576  7.9162739 32.015832 2.095452
##                    ACF1 Theil's U
## Training set -0.06162805        NA
## Test set      0.83299805  2.349537
```

Model 5: TBATS Model

```
# Fit TBATS model (training data assumed to be ts_daily_train)
TBATS_fit <- tbats(ts_daily_train)

TBATS_for <- forecast(TBATS_fit, h = horizon)

# Forecast plot
autoplot(TBATS_for) +
  ggtitle("Model 5: TBATS Forecast") +
  ylab("Load")
```




```
# Overlay plot
autoplot(ts_daily_train) +
  autolayer(TBATS_for, series = "TBATS", PI = FALSE) +
  ylab("Load")
```




```
# Accuracy on test set
accuracy(TBATS_for, ts_daily_test)
```

```
##                     ME      RMSE       MAE        MPE     MAPE      MASE
## Training set  43.90081  501.3385  363.7827 -0.6194292 10.78758 0.4728086
## Test set     711.90657 1376.0978 1083.3750  9.2346813 20.45227 1.4080633
##                    ACF1 Theil's U
## Training set 0.03361668        NA
## Test set     0.79537200  1.553679
```

Model Comparison: NNAR + Fourier + TBATS

```
# Compare performance across all six models

comparison <- data.frame(
  Model = c(
    "Model 1: NNAR + Fourier (K = c(2,8))",
    "Model 2: NNAR + Fourier (K = c(2,12)) Baseline",
    "Model 3: NNAR + Fourier (K = c(2,12)) on Test",
    "Model 4: NNAR + Fourier (K = c(3,18))",
    "Model 5: TBATS"
  ),
  RMSE = c(
    accuracy(NN_for_k28, ts_daily_test)[2, "RMSE"],
    accuracy(NN_for_k212_base, ts_daily_test)[2, "RMSE"],
    accuracy(NNAR_Fourier_forecast, ts_daily_test)[2, "RMSE"],
    accuracy(NN_for_k318, ts_daily_test)[2, "RMSE"],
    accuracy(TBATS_for, ts_daily_test)[2, "RMSE"]
```

```
  ),
  MAE = c(
    accuracy(NN_for_k28, ts_daily_test)[2, "MAE"],
    accuracy(NN_for_k212_base, ts_daily_test)[2, "MAE"],
    accuracy(NNAR_Fourier_forecast, ts_daily_test)[2, "MAE"],
    accuracy(NN_for_k318, ts_daily_test)[2, "MAE"],
    accuracy(TBATS_for, ts_daily_test)[2, "MAE"]
  ),
  MAPE = c(
    accuracy(NN_for_k28, ts_daily_test)[2, "MAPE"],
    accuracy(NN_for_k212_base, ts_daily_test)[2, "MAPE"],
    accuracy(NNAR_Fourier_forecast, ts_daily_test)[2, "MAPE"],
    accuracy(NN_for_k318, ts_daily_test)[2, "MAPE"],
    accuracy(TBATS_for, ts_daily_test)[2, "MAPE"]
  )
)

# Nicely formatted table
kable(
  comparison,
  caption = "Performance Comparison of All 5 Models (Train/Test Evaluation)",
  digits = 3
)
```

Table 1: Performance Comparison of All 5 Models (Train/Test Evaluation)

Model	RMSE	MAE	MAPE
Model 1: NNAR + Fourier (K = c(2,8))	1920.850	1597.864	31.773
Model 2: NNAR + Fourier (K = c(2,12)) Baseline	1988.111	1665.546	33.081
Model 3: NNAR + Fourier (K = c(2,12)) on Test	2023.329	1690.902	33.844
Model 4: NNAR + Fourier (K = c(3,18))	1917.704	1612.258	32.016
Model 5: TBATS	1376.098	1083.375	20.452

Retraining and Forecasting on Full Data

Model 1: NNAR + Fourier (K = c(2,8))

```
# Use horizon for forecasting
horizon <- length(ts_daily_test)

# Fit the NNAR model with Fourier terms using the full training dataset
NN_fit_k28_full <- nnetar(ts_electricity_daily, p = 2, P = 2, xreg = fourier(ts_electricity_daily, K = c

# Forecast for the desired horizon
NN_for_k28_full <- forecast(NN_fit_k28_full, h = horizon, xreg = fourier(ts_electricity_daily, K = c(2,

# Create a data frame for the forecast results
forecast_results_k28 <- data.frame(
```

```
  date = seq(as.Date("2011-01-01"), by = "day", length.out = horizon),
  load = NN_for_k28_full$mean
)

# View the forecast results
head(forecast_results_k28)
```

```
##         date     load
## 1 2011-01-01 5553.398
## 2 2011-01-02 5550.989
## 3 2011-01-03 4986.128
## 4 2011-01-04 4916.766
## 5 2011-01-05 5135.657
## 6 2011-01-06 5460.052
```

Model 2: NNAR + Fourier (K = c(2,12))

```
# Use horizon for forecasting
horizon <- length(ts_daily_test)

# Fit the NNAR model with Fourier terms using the full training dataset
NN_fit_k212_base_full <- nnetar(ts_electricity_daily, p = 2, P = 2, xreg = fourier(ts_electricity_daily

# Forecast for the desired horizon
NN_for_k212_base_full <- forecast(NN_fit_k212_base_full, h = horizon, xreg = fourier(ts_electricity_dai

# Create a data frame for the forecast results
forecast_results_k212_base <- data.frame(
  date = seq(as.Date("2011-01-01"), by = "day", length.out = horizon),
  load = NN_for_k212_base_full$mean
)

# View the forecast results
head(forecast_results_k212_base)
```

```
##         date     load
## 1 2011-01-01 5390.780
## 2 2011-01-02 5742.879
## 3 2011-01-03 5531.183
## 4 2011-01-04 5492.843
## 5 2011-01-05 5380.042
## 6 2011-01-06 5440.070
```

Model 3: NNAR + Fourier

```
# Fit the NNAR model with Fourier terms using the full training dataset
NNAR_Fourier_full_fit <- nnetar(ts_electricity_daily, p = 2, P = 2, xreg = fourier(ts_electricity_daily

# Forecast for the desired horizon (January 1 to February 28, 2011)
```

```r
horizon_full <- 59  # Number of days to forecast
NNAR_Fourier_full_forecast <- forecast(NNAR_Fourier_full_fit, h = horizon_full, xreg = fourier(ts_electi

# Create a data frame for the forecast results
forecast_results <- data.frame(
  date = seq(as.Date("2011-01-01"), by = "day", length.out = horizon_full),
  load = NNAR_Fourier_full_forecast$mean
)
```

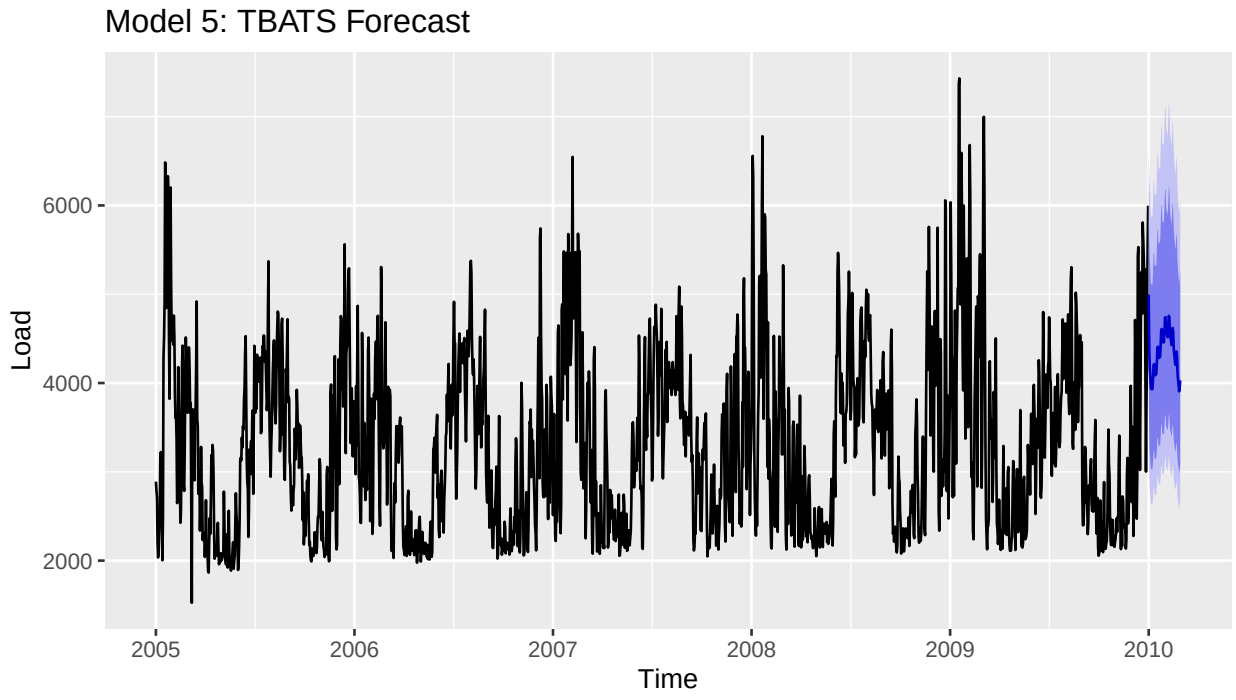
Model 4: NNAR + Fourier (K = c(3,18))

```r
# Set K for Fourier
K4 <- c(3, 18)
# Training
ts_full <- window(ts_electricity_daily, end = c(2010, 365))

# Create Fourier regressors
xreg_full_4 <- fourier(ts_full, K = K4)
xreg_fc_4   <- fourier(ts_full, K = K4, h = horizon)

# Fit the NNAR model with extended size and decay
fit_nnar_k318 <- nnetar(
  ts_full,
  p = 2,
  P = 2,
  xreg = xreg_full_4,
  size = 10,
  decay = 0.01,
  maxNWts = 2000
)

# Forecast
fc_nnar_k318 <- forecast(
  fit_nnar_k318,
  h = horizon,
  xreg = xreg_fc_4
)

# Create forecast dataframe
final_nnar_k318_df <- data.frame(
  date = seq(as.Date("2011-01-01"), by = "day", length.out = horizon_full),
  load = as.numeric(fc_nnar_k318$mean)
)
```

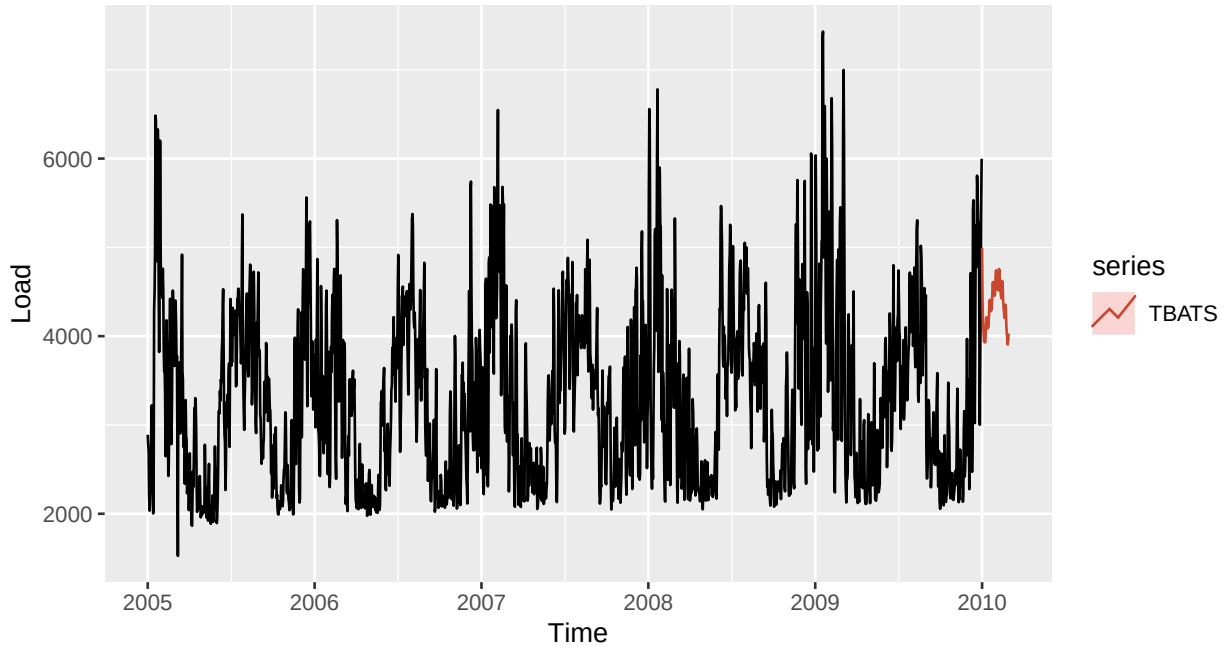
Model 5: TBATS

```r
# Use horizon for forecasting
horizon <- length(ts_daily_test)

# Fit TBATS model (training data assumed to be ts_daily_train)
```

```r
TBATS_fit_full <- tbats(ts_daily_train)

# Forecast for the desired horizon
TBATS_for_full <- forecast(TBATS_fit_full, h = horizon)

# Forecast plot
autoplot(TBATS_for_full) +
  ggtitle("Model 5: TBATS Forecast") +
  ylab("Load")
```




```r
# Overlay plot
autoplot(ts_daily_train) +
  autolayer(TBATS_for_full, series = "TBATS", PI = FALSE) +
  ylab("Load")
```




```
# Accuracy on test set
accuracy(TBATS_for_full, ts_daily_test)
```

```
##                      ME      RMSE       MAE        MPE     MAPE      MASE
## Training set  43.90081  501.3385  363.7827 -0.6194292 10.78758 0.4728086
## Test set     711.90657 1376.0978 1083.3750  9.2346813 20.45227 1.4080633
##                    ACF1 Theil's U
## Training set 0.03361668        NA
## Test set     0.79537200  1.553679
```

```
# Create a data frame for the forecast results
forecast_results_tbats <- data.frame(
  date = seq(as.Date("2011-01-01"), by = "day", length.out = horizon),
  load = as.numeric(TBATS_for_full$mean)
)
```

Exporting Dataset

```
write.csv(forecast_results_k28, file = here::here("Submission Files", "NNAR_Fourier_K28.csv"), row.names
write.csv(forecast_results_k212_base, file = here::here("Submission Files", "forecast_results_k212_base
write.csv(forecast_results, file = here::here("Submission Files", "NNAR_Fourier.csv"), row.names = FALSE
write.csv(final_nnar_k318_df,file = here::here("Submission Files", "submission_nnar_k318.csv"), row.name
write.csv(forecast_results_tbats,file = here::here("Submission Files", "forecast_results_tbats.csv"), r
write.csv(forecast_results_tbats, "submission_arima_nnar_hybrid.csv", row.names = FALSE)
```