# HuanHusted_TSA_Competition

Kelsey Husted

2023-03-31

## Time Series Competition

```
knitr::opts_chunk$set(echo = TRUE,tidy.opts=list(width.cutoff=80), tidy=FALSE)
```

##Load packages

```
library(readxl)
library(dplyr)
library(lubridate)
library(openxlsx)
library(ggplot2)
library(forecast)
library(Kendall)
library(tseries)
library(outliers)
library(tidyverse)
library(smooth)
library(zoo)
library(kableExtra)
```

## Data wrangling and processing from 2005-2009

Hourly data was transformed into daily data with aggregate functions and pipes (i.e., tidyverse).

```
#Import data
df <- read_xlsx("./Competition/Data/load.xlsx")

#Wrangle data from hourly to daily
#Wrangling date column 2005 to 2009
df_daily <- df %>%
  mutate( Date = ymd(date)) %>%
  filter(Date < '2010-01-01')

#removing no numeric columns so rowMeans() functino will work
df_dailyV2 <- df %>%
  mutate( Date = ymd(date)) %>%
  filter(Date < '2010-01-01') %>%
  select(3:26)

#Creating daily data
df_processed <- df_dailyV2 %>%
  mutate(rowMeans(df_dailyV2)) %>%
  rename(Daily_data = "rowMeans(df_dailyV2)") %>%
  select(25)

#Combining date and daily data
date <- df_daily[,2]
df_processed <- cbind(date, df_processed)

nobs = nrow(df_daily)
```

### Data wrangling and processing from 2005-2010

The data needs to be formatted to include 2010 as well since the objective is to forecast for 2011. Instead of making two separate datasets, I should use the window() function for future reference.

```
#Wrangle data from hourly to daily
#Wrangling date column 2005 to 2010
#removing no numeric columns so rowMeans() function will work
df_daily2010 <- df %>%
  mutate( Date = ymd(date)) %>%
  select(3:26)

#Creating daily data
df_processed2010 <- df_daily2010 %>%
  mutate(rowMeans(df_daily2010)) %>%
  rename(Daily_data = "rowMeans(df_daily2010)") %>%
  select(25)

#Combining data and daily data
date <- df[,2]
df_processed2010 <- cbind(date, df_processed2010)

nobs2010 = nrow(df_processed2010)
```

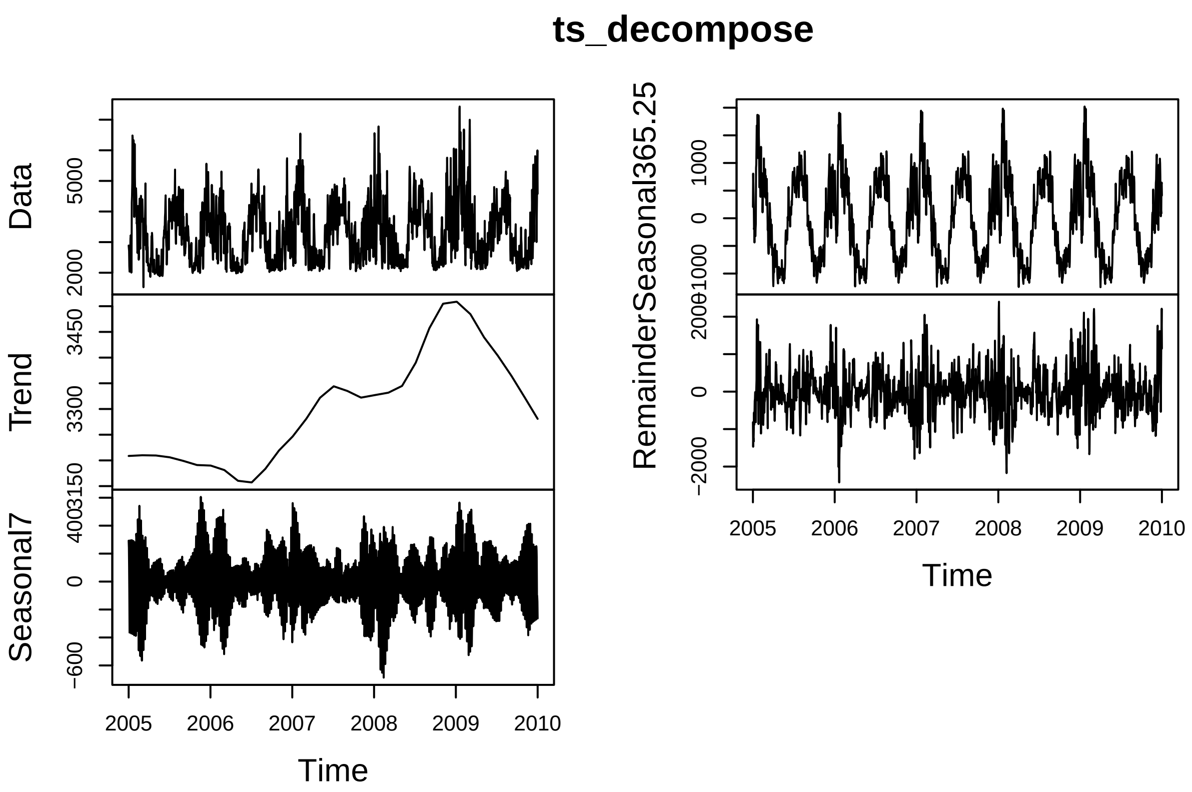
### Time series object transformation

```
#ts transformation 2005 to 2009
ts_daily <- msts(df_processed$Daily_data,
                 seasonal.periods=c(7,365.25),
                 start=c(2005, 01, 01))

#ts transformation 2005 to 2010
ts_daily2010 <- msts(df_processed2010$Daily_data,
                 seasonal.periods=c(7,365.25),
                 start=c(2005, 01, 01))
```

### Time series decomposition and plot

```
#Decompose time series
ts_decompose <- ts_daily %>%
  mstl()
plot(ts_decompose)
```

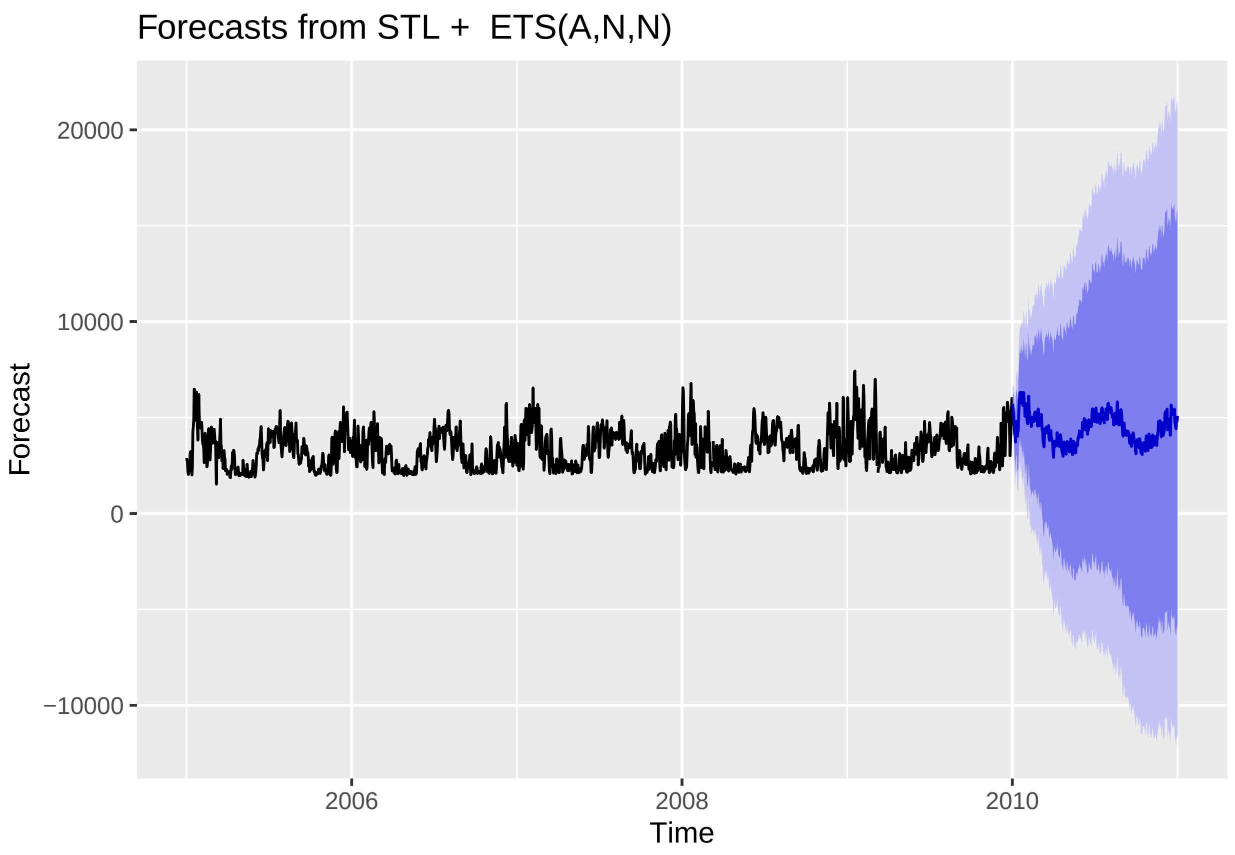


##

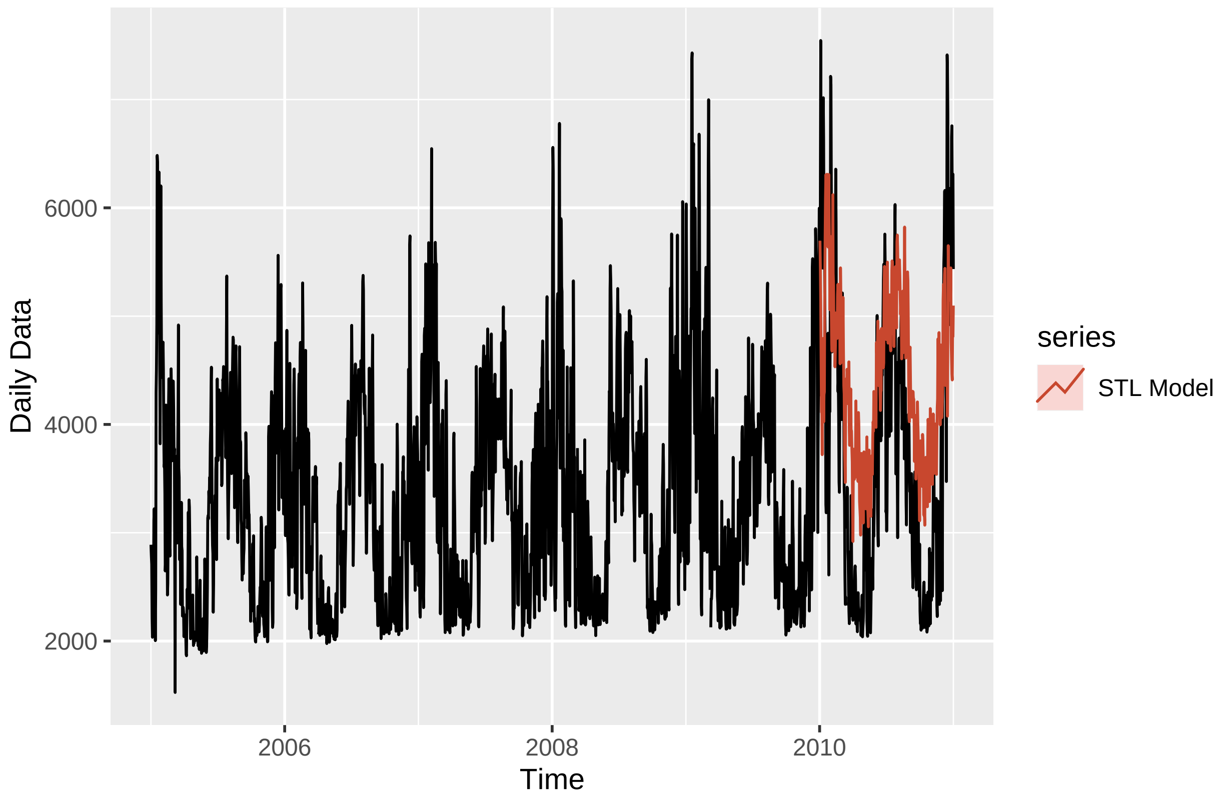
Model 1 STL + ETS: Forecast 2010

```
#Fit and forecast STL model
ETS_model <-  stlf(ts_daily,h=365)

#Plot foresting
autoplot(ETS_model) + ylab("Forecast")
```



```
#Plot model + observed data
autoplot(ts_daily2010) +
  autolayer(ETS_model, series="STL Model",PI=FALSE) +
  ylab("Daily Data")
```



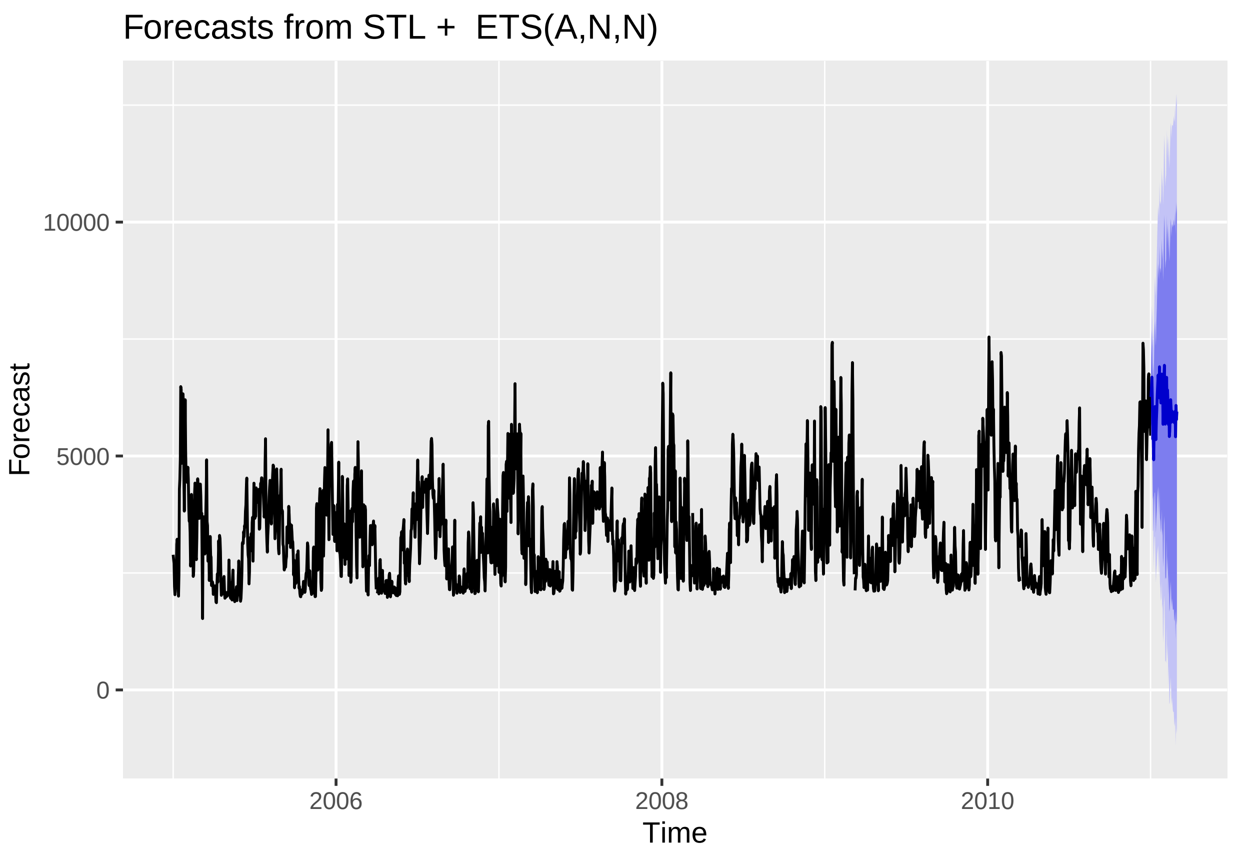
```
#Check accuracy of model
n_for <- 365
observed <- df_processed[(nobs-n_for+1):nobs, "Daily_data"]
ETS_scores <- accuracy(ETS_model$mean,observed)
print(ETS_scores)
```

```
##                   ME     RMSE      MAE       MPE     MAPE
## Test set -984.5201 1210.625 1079.332 -35.24753 36.80666
```

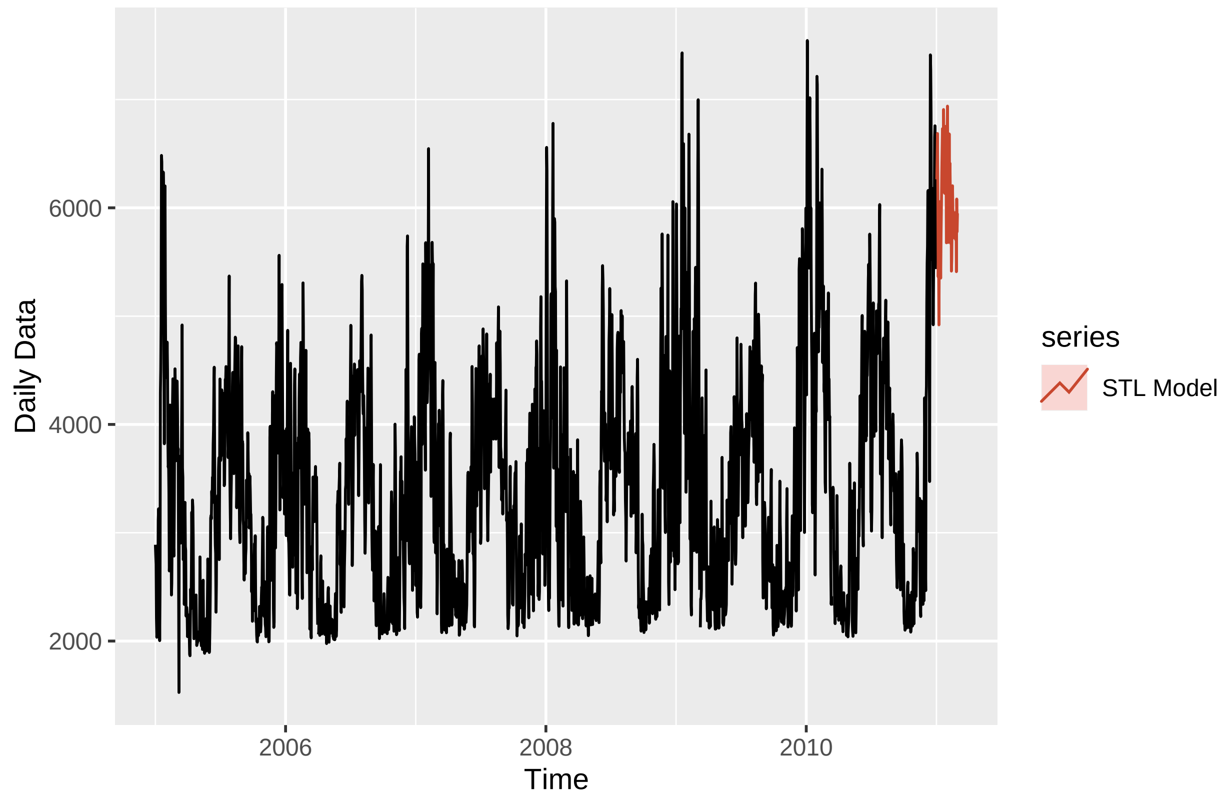
### Model 1 STL + ETS: Forecast 2011

```
#Fit and forecast STL model January 1st to February 28th 2011
ETS_model2011 <-  stlf(ts_daily2010,h=59)

#Plot foresting
autoplot(ETS_model2011) + ylab("Forecast")
```



```
#Plot model + observed data
autoplot(ts_daily2010) +
  autolayer(ETS_model2011, series="STL Model",PI=FALSE) +
  ylab("Daily Data")
```



```
#Check accuracy of model
n_for <- 59
observed <- df_processed[(nobs-n_for+1):nobs, "Daily_data"]
ETS_scores <- accuracy(ETS_model2011$mean,observed)
print(ETS_scores)
```

```
##                 ME     RMSE      MAE       MPE     MAPE
## Test set -2528.13 2846.706 2535.511 -90.47756 90.60064
```

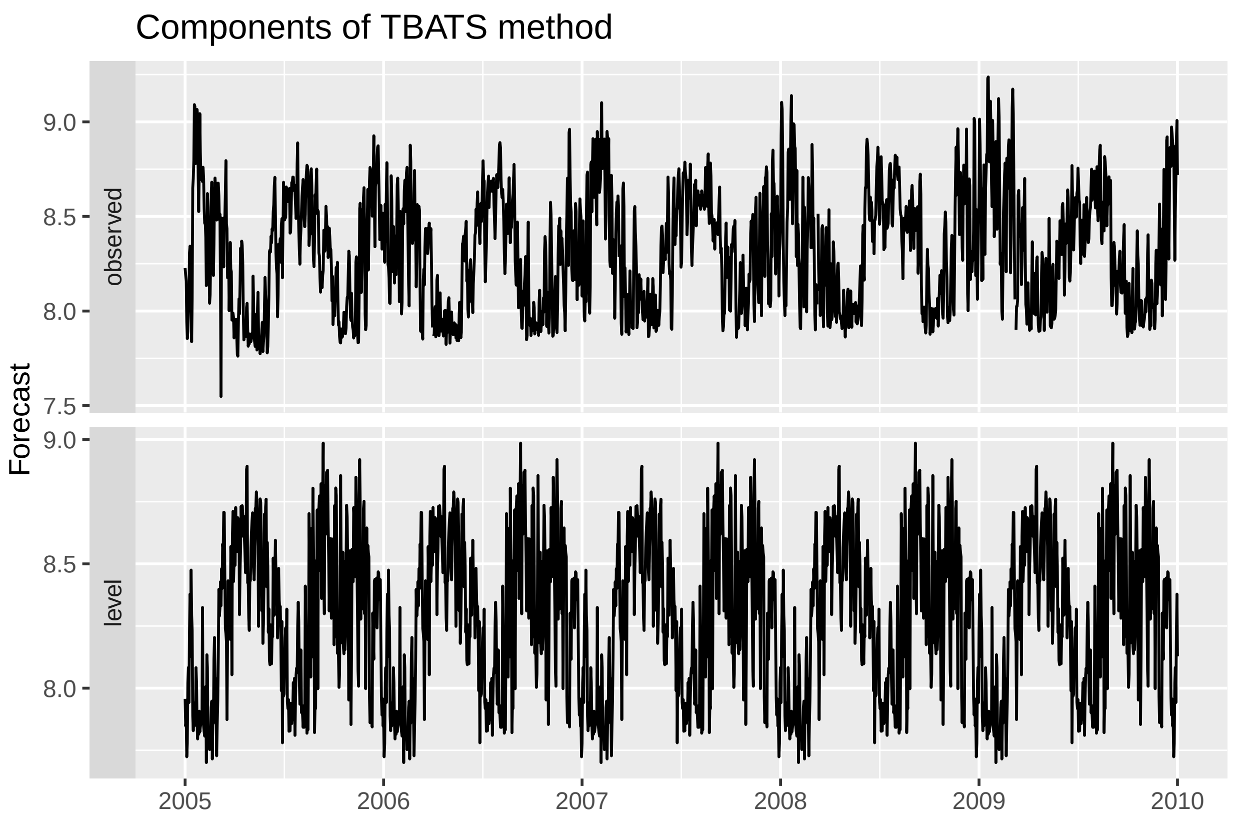
### Model 2 TBATS: Forecast 2010

## The model looks like a really bad fit visually and will not be used to forecast for 2011.

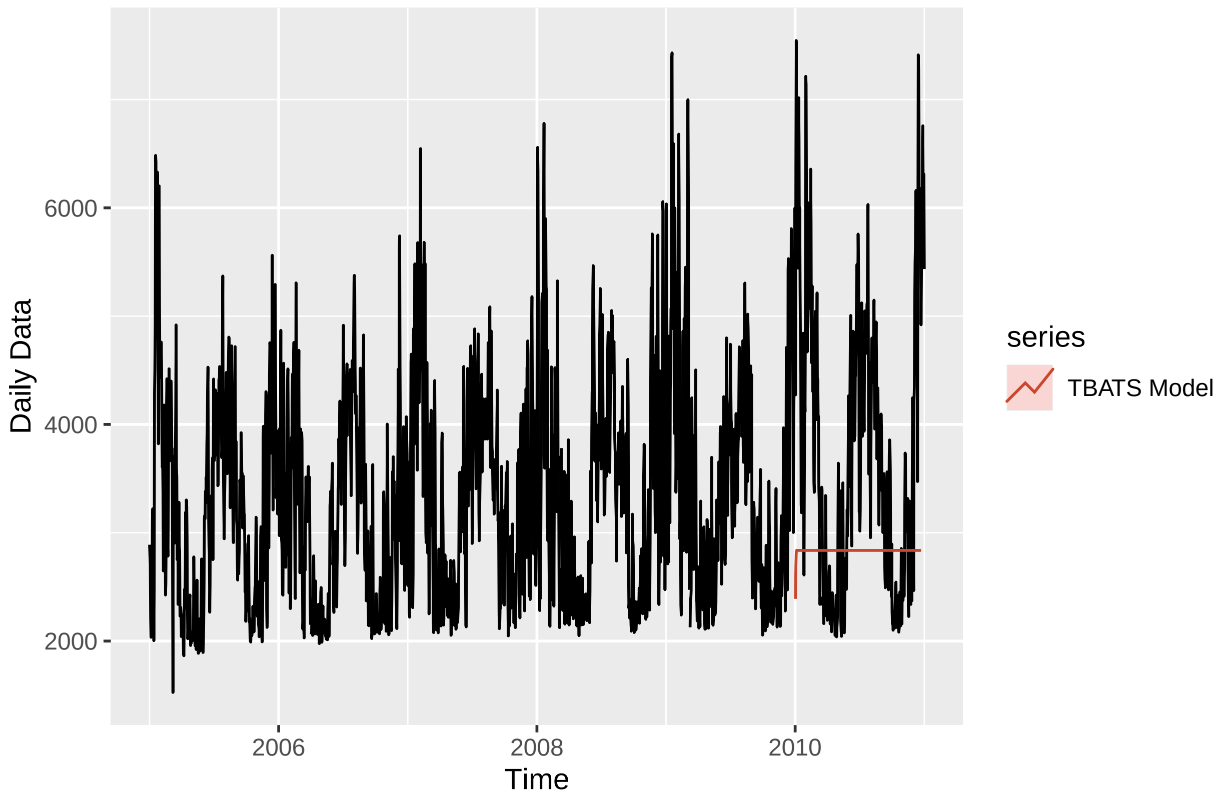
```
#Fit and forecast TBATS model
TBATS_model <-  tbats(ts_daily)

#forecast
TBATS_for <- forecast(TBATS_model,h=356)

#Plot foresting
autoplot(TBATS_model) + ylab("Forecast")
```



```
#Plot model + observed data
autoplot(ts_daily2010) +
  autolayer(TBATS_for, series="TBATS Model",PI=FALSE) +
  ylab("Daily Data")
```

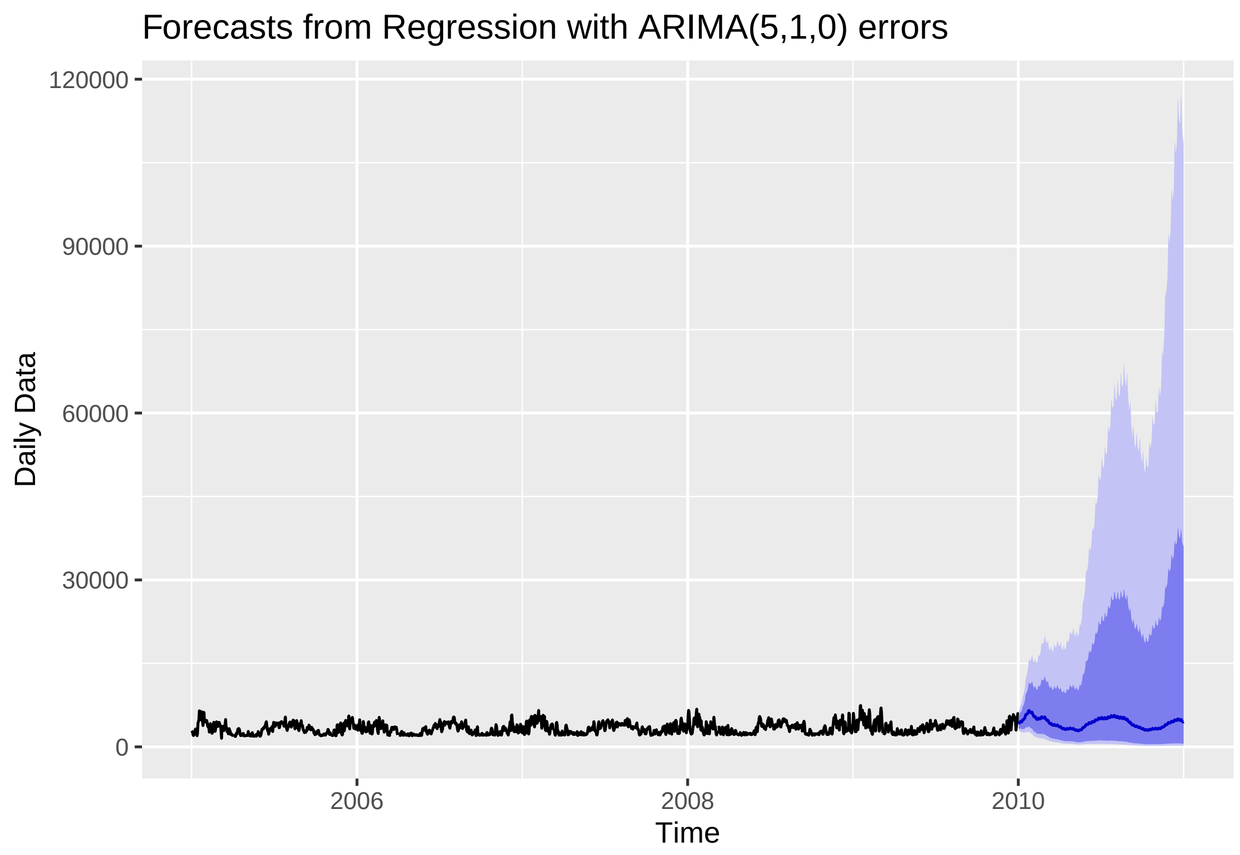


### Model 3 ARIMA + FOURIER terms: Forecast 2010

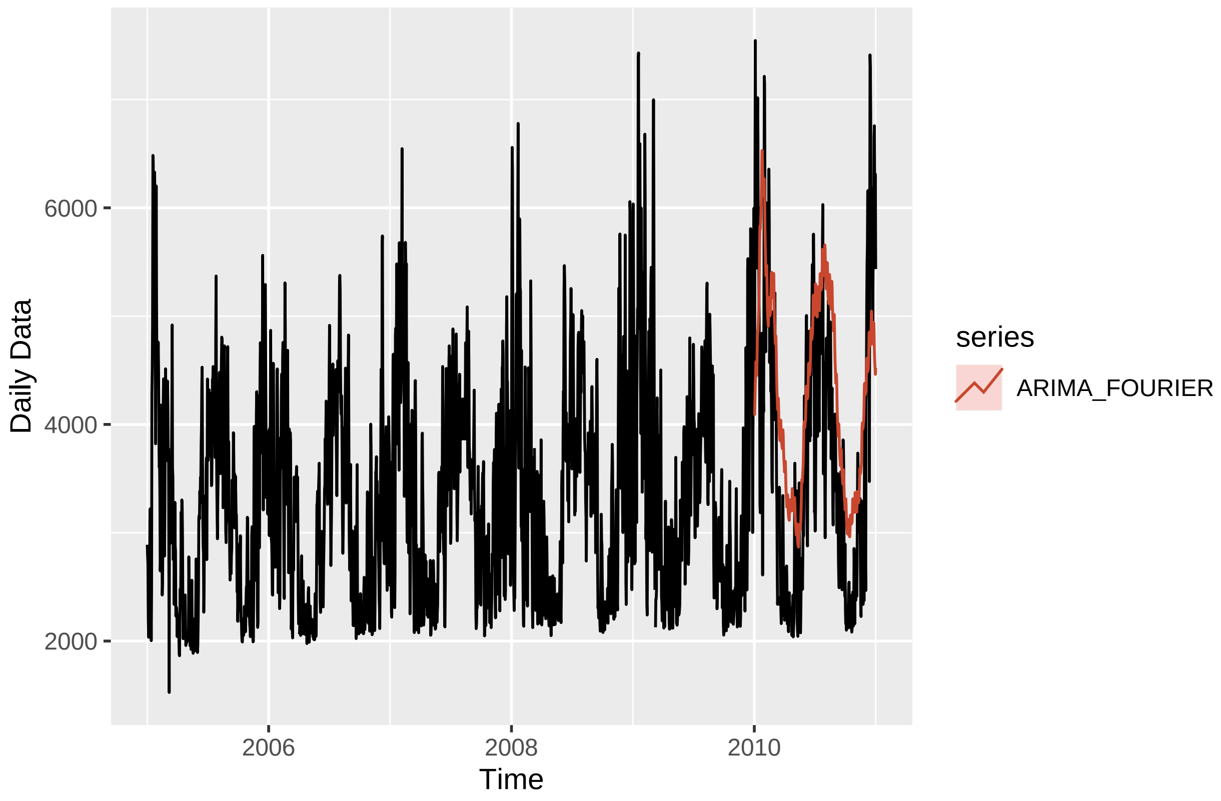
```
#Fit and forecast TBATS model
ARIMA_Four_model <- auto.arima(ts_daily,
                             seasonal=FALSE,
                             lambda=0,
                             xreg=fourier(ts_daily,
                                          K=c(2,12))
                             )

#Forecast
ARIMA_Four_for <- forecast(ARIMA_Four_model,
                           xreg=fourier(ts_daily,
                                        K=c(2,12),
                                        h=365),
                           h=365
                           )

#Plot foresting results
autoplot(ARIMA_Four_for) + ylab("Daily Data")
```



```
#Plot model + observed data
autoplot(ts_daily2010) +
  autolayer(ARIMA_Four_for, series="ARIMA_FOURIER",PI=FALSE) +
  ylab("Daily Data")
```



```r
#Check accuracy of model
n_for <- 365
observed <- df_processed[(nobs-n_for+1):nobs, "Daily_data"]
ARIMA_Four_scores <- accuracy(ARIMA_Four_for$mean,observed)
print(ARIMA_Four_scores)
```

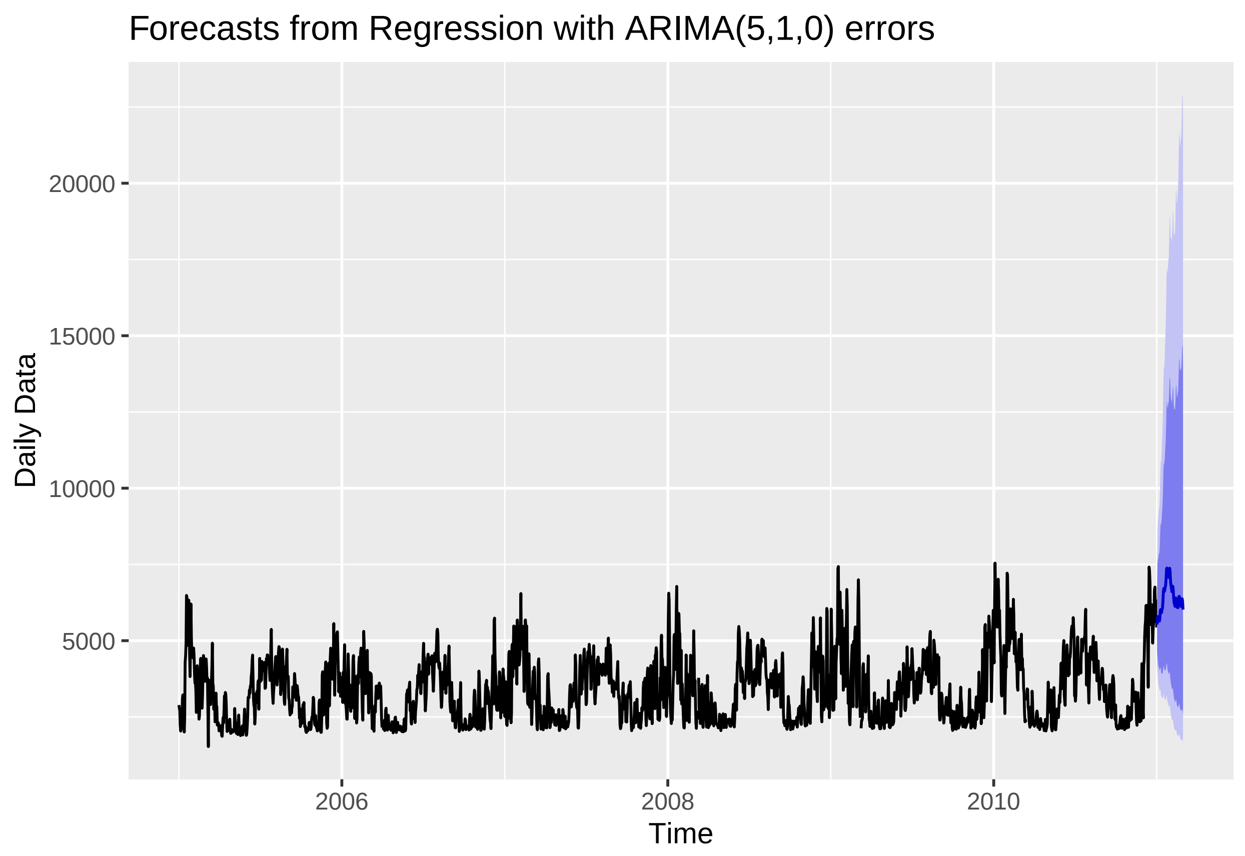
```
##                  ME     RMSE      MAE       MPE    MAPE
## Test set -857.4621 1162.004 994.7404 -30.51384 32.8675
```

### Model 3 ARIMA + FOURIER terms: Forecast 2011

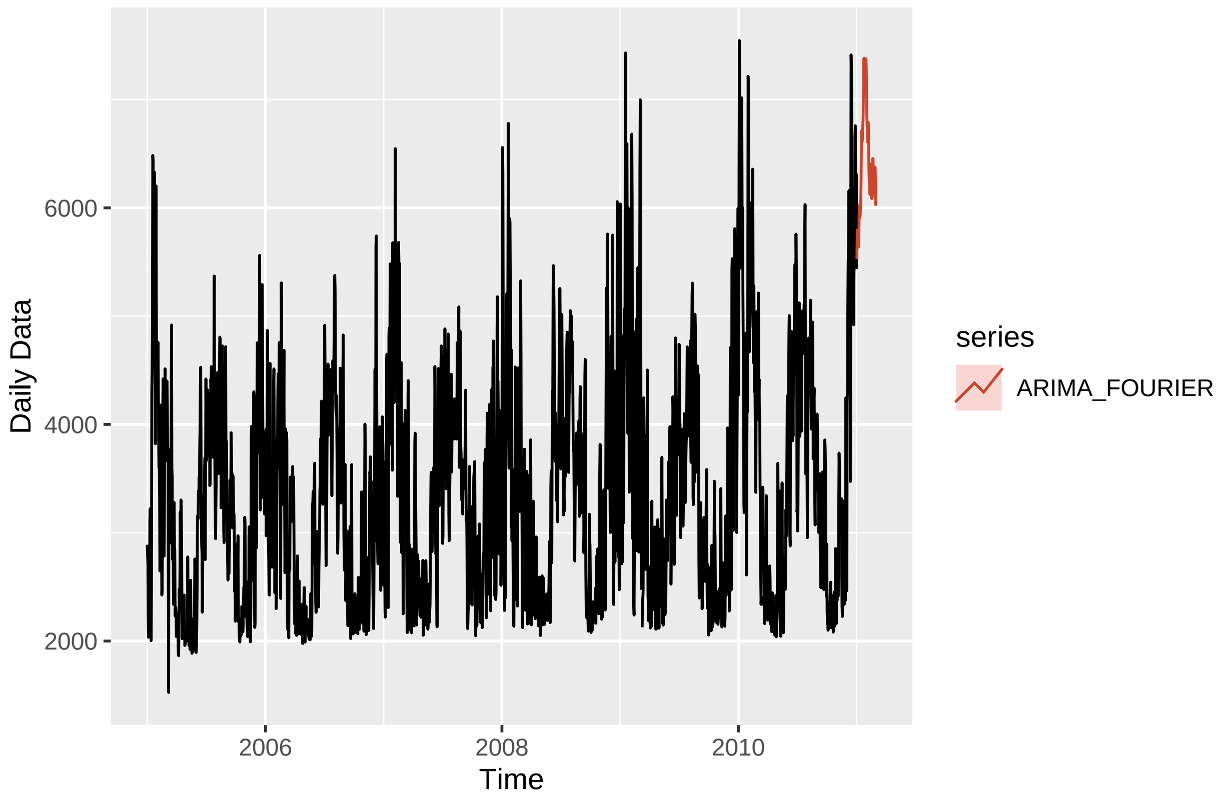
```r
#Fit and forecast TBATS model
ARIMA_Four_model2011 <- auto.arima(ts_daily2010,
                             seasonal=FALSE,
                             lambda=0,
                             xreg=fourier(ts_daily2010,
                                          K=c(2,12))
                             )

#Forecast
ARIMA_Four_for2011 <- forecast(ARIMA_Four_model2011,
                           xreg=fourier(ts_daily2010,
                                        K=c(2,12),
                                        h=59),
                           h=59
                           )

#Plot foresting results
autoplot(ARIMA_Four_for2011) + ylab("Daily Data")
```



```
#Plot model + observed data
autoplot(ts_daily2010) +
  autolayer(ARIMA_Four_for2011, series="ARIMA_FOURIER",PI=FALSE) +
  ylab("Daily Data")
```



```r
#Check accuracy of model
n_for <- 59
observed <- df_processed[(nobs-n_for+1):nobs, "Daily_data"]
ARIMA_Four_scores <- accuracy(ARIMA_Four_for$mean,observed)
print(ARIMA_Four_scores)
```

```
##                   ME     RMSE      MAE       MPE     MAPE
## Test set -1832.672 2258.961 1948.471 -67.73503 69.78746
```

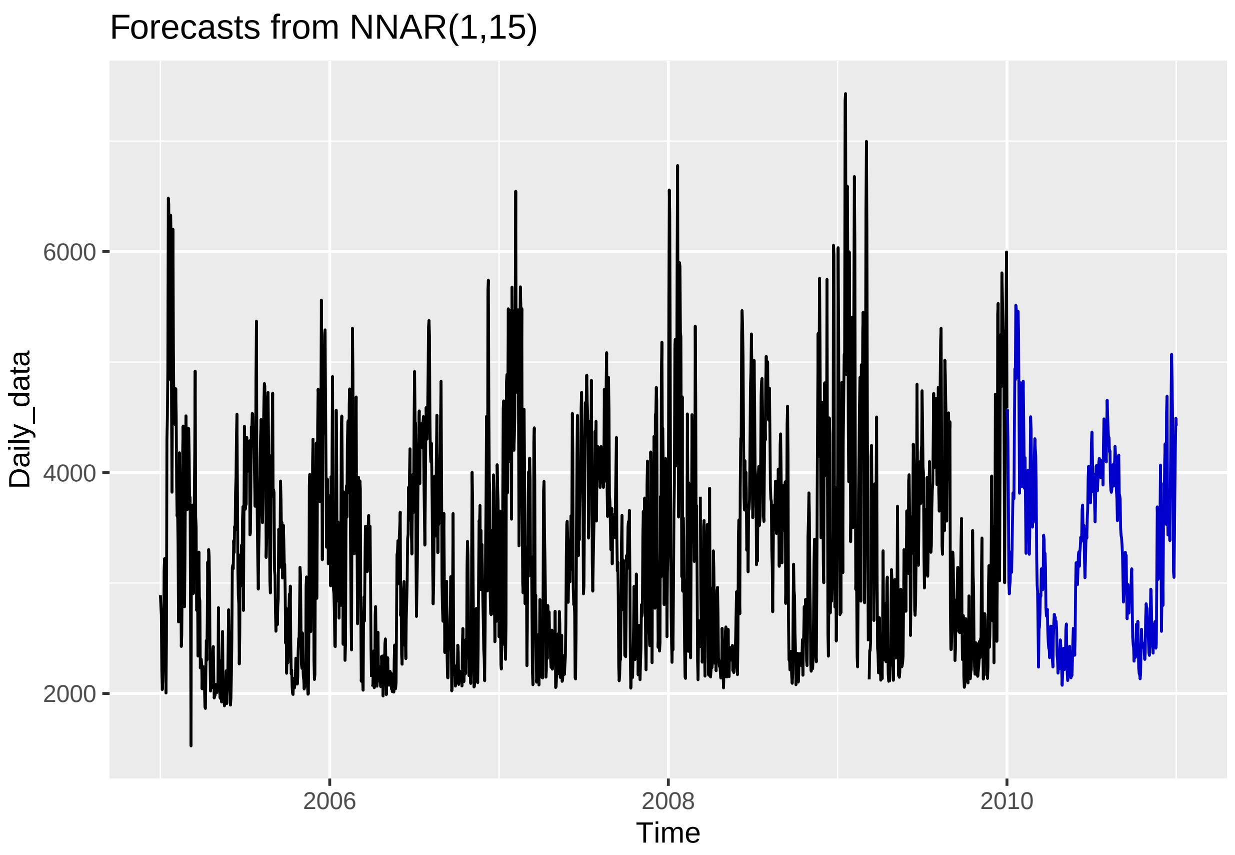
```r
#print(ARIMA_Four_for2011$mean)
```

### Model 4 Neural Network Time Series: Forecasts 2010

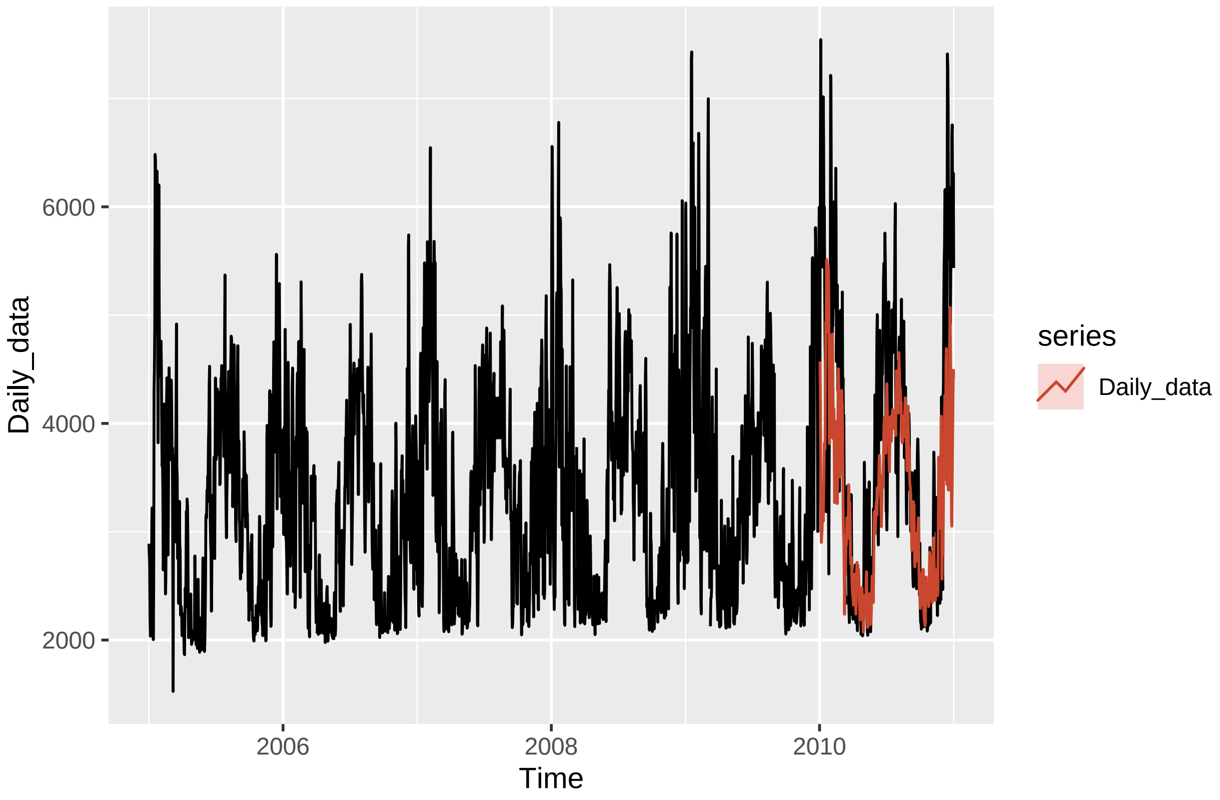
```r
#NN_fit <- nnetar(ts_act_power_daily_train,p=1,P=1)
NN_model <- nnetar(ts_daily,decay=0.5, maxit=150, p=1,P=0,xreg=fourier(ts_daily, K=c(2,12)))

#NN_for <- forecast(NN_fit, h=365)
NN_for <- forecast(NN_model, h=365,xreg=fourier(ts_daily,
                                          K=c(2,12),h=365))

#Plot foresting results
autoplot(NN_for) +
  ylab("Daily_data")
```



```
#Plot model + observed data
autoplot(ts_daily2010) +
  autolayer(NN_for, series="Daily_data",PI=FALSE)+
  ylab("Daily_data")
```



```r
#Checking error variables to decide which model fits the data the best
n_for <- 365
observed <- df_processed[(nobs-n_for+1):nobs, "Daily_data"]
NN_scores1 <- accuracy(NN_for$mean,observed)
print(NN_scores1)
```

```
##                 ME     RMSE      MAE       MPE     MAPE
## Test set 132.4837 725.6423 538.1543 0.2831779 14.99085
```

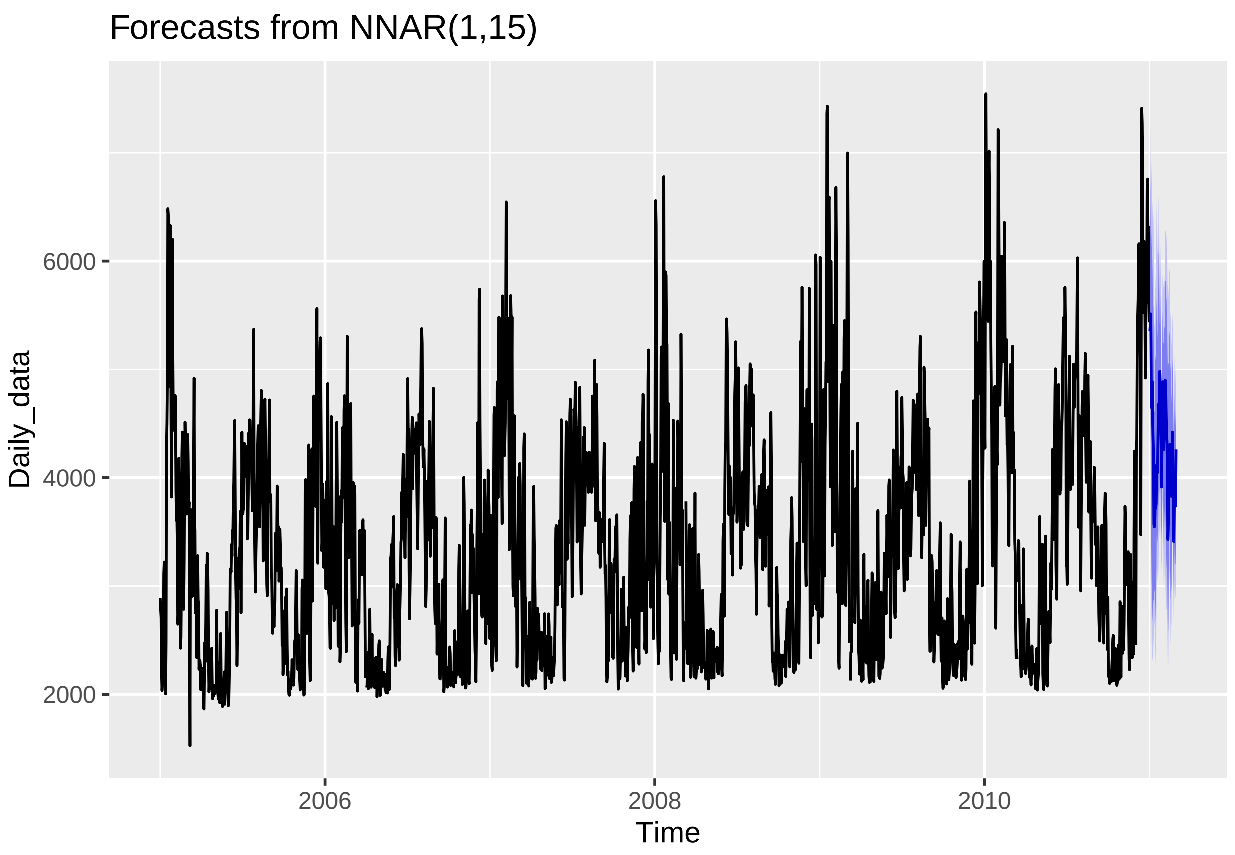
## Model 4 Neural Network Time Series: Forecasts 2011

Note: Based on the error variables calculated with the accuracy() function, the Neural Network model seems to fit the data the best.

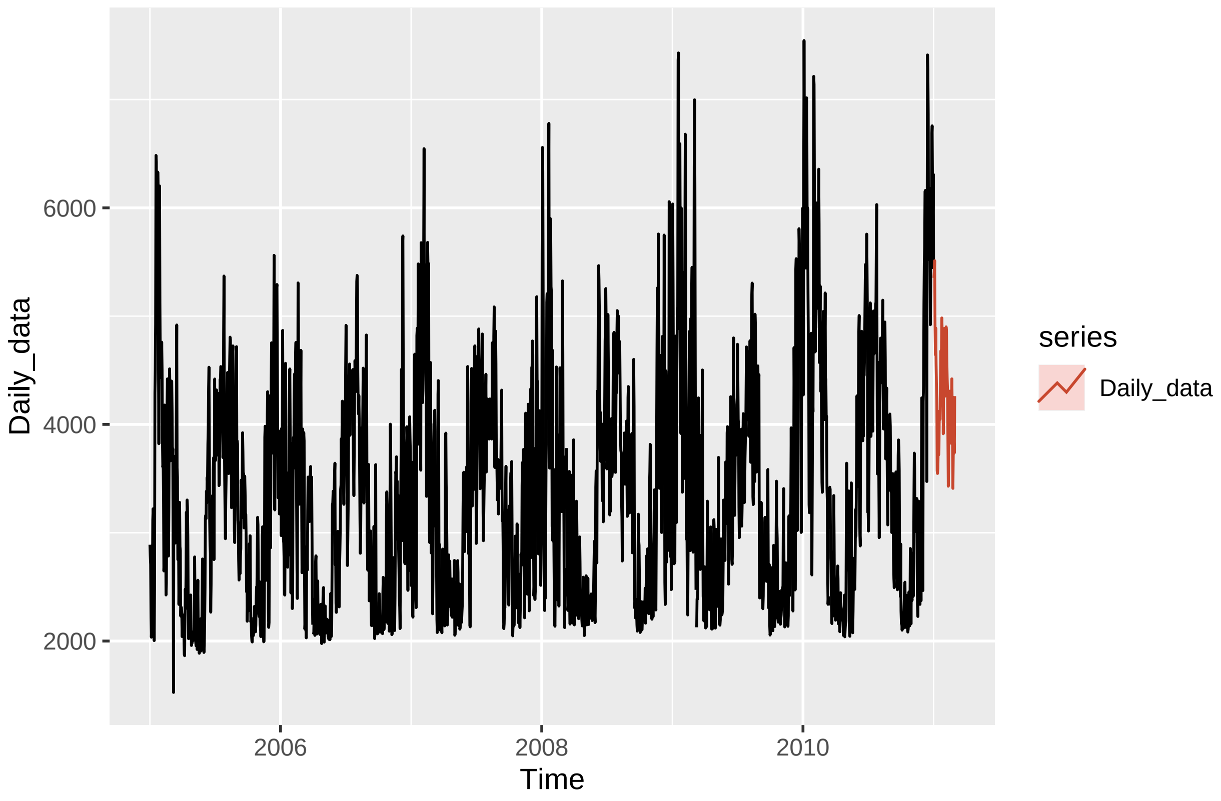
```r
#NN_fit <- nnetar(ts_act_power_daily_train,p=1,P=1)
NN_model2010 <- nnetar(ts_daily2010,lambda = 0.5,p=1,P=0,xreg=fourier(ts_daily2010, K=c(2,12)))

#NN_for <- forecast(NN_fit, h=365)
NN_for2010 <- forecast(NN_model2010,PI=TRUE, h=59,xreg=fourier(ts_daily,
                                             K=c(2,12),h=59))

#Plot foresting results
autoplot(NN_for2010) +
  ylab("Daily_data")
```



```
#Plot model + observed data
autoplot(ts_daily2010) +
  autolayer(NN_for2010, series="Daily_data",PI=FALSE)+
  ylab("Daily_data")
```



```
n_for <- 59
observed <- df_processed[(nobs-n_for+1):nobs, "Daily_data"]
NN_scores1 <- accuracy(NN_for2010$mean,observed)
print(NN_scores1)
```

```
##                 ME    RMSE     MAE       MPE     MAPE
## Test set -783.4217 1559.52 1366.36 -35.62615 46.43443
```

```
print(NN_for2010$mean)
```

```
## Multi-Seasonal Time Series:
## Start: 2011 2
## Seasonal Periods: 7 365.25
## Data:
##  [1] 5353.322 5414.404 5512.305 4905.975 4645.073 4888.908 4557.303 4373.035
##  [9] 4236.501 3572.071 3546.786 3596.116 3804.270 3719.135 3764.665 4114.617
## [17] 4123.748 4047.657 4352.350 4677.749 4666.738 4680.017 4983.692 4860.866
## [25] 4580.942 4314.108 3913.692 4348.505 4793.128 4886.330 4425.797 4260.492
## [33] 4572.972 4670.933 4901.104 4861.384 4594.116 4422.188 4318.501 3889.098
## [41] 3429.671 3593.582 4075.036 4146.549 4307.565 4157.280 4030.105 3895.836
## [49] 3826.355 4210.435 4421.229 4183.051 3776.125 3409.297 3730.085 4004.851
## [57] 3824.923 3736.642 4261.250
```