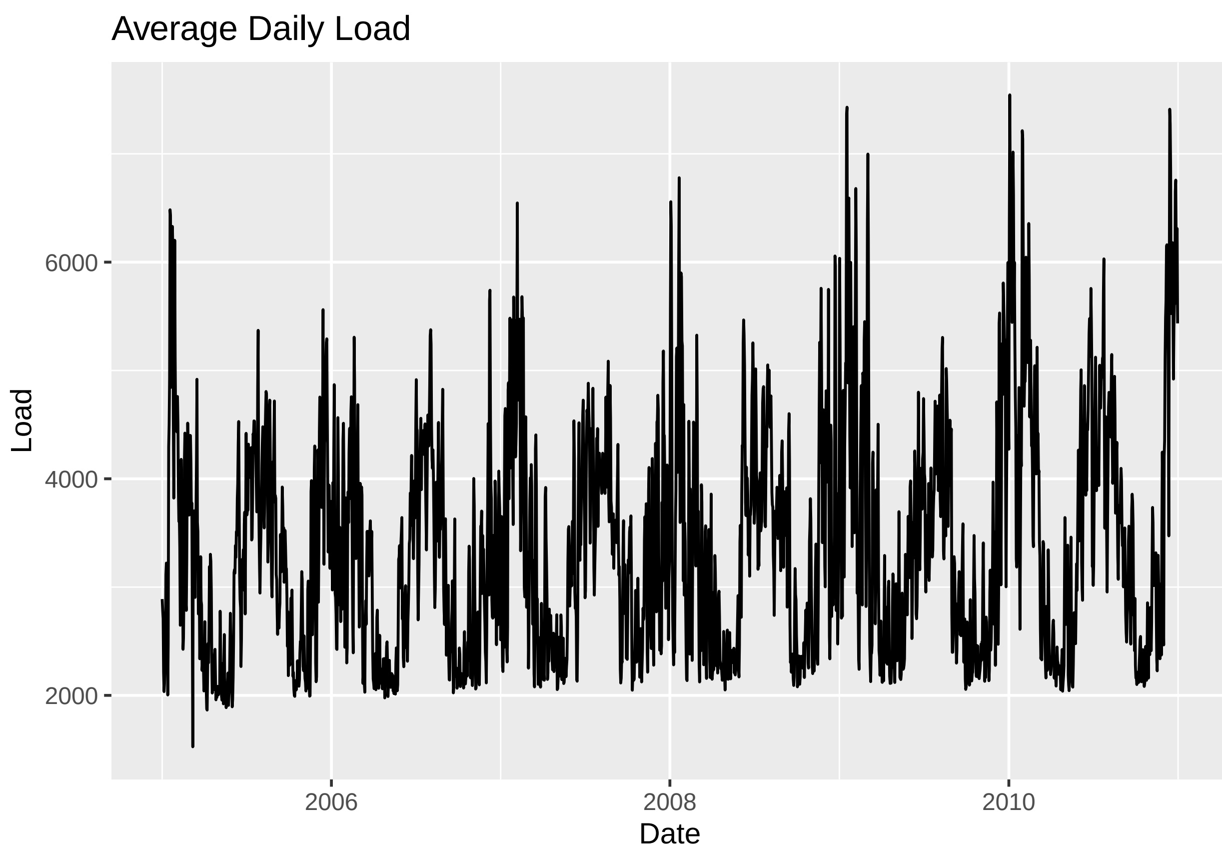
Sokolova, Biehl TSA Competition

Tatiana Sokolova, Kevin Biehl

4/13/2022

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

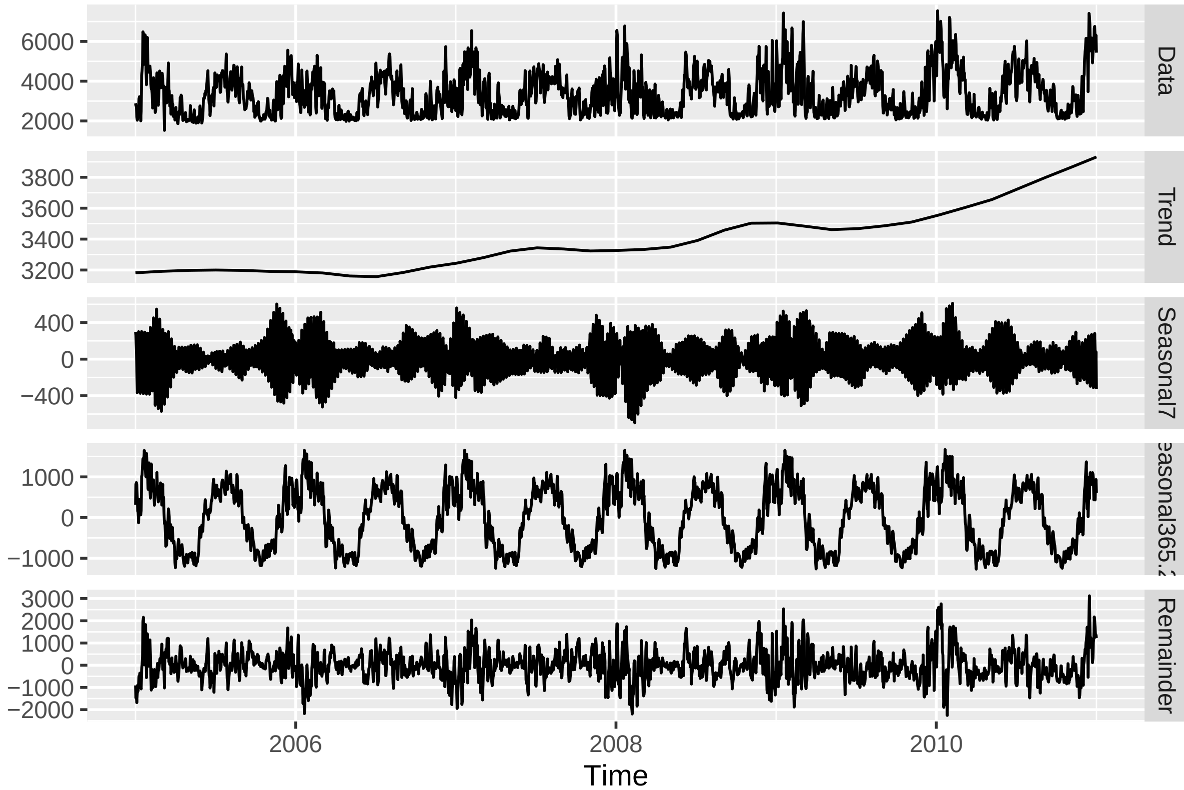
```r
#Importing data
load <- read_excel("./Data/load.xlsx")

#Aggregating from hourly to daily and omitting NAs from the calculation
DailyAvgLoad <- rowMeans(load[,3:26], na.rm=TRUE)

DailyAvgLoad <- data.frame(load$date,DailyAvgLoad)
colnames(DailyAvgLoad) <- c("Date","Load")
DailyAvgLoad$Date <- ymd(DailyAvgLoad$Date)

ggplot(DailyAvgLoad, aes(x=Date,y=Load)) +
  geom_line() +
  ggtitle("Average Daily Load")+
  ylab("Load")
```



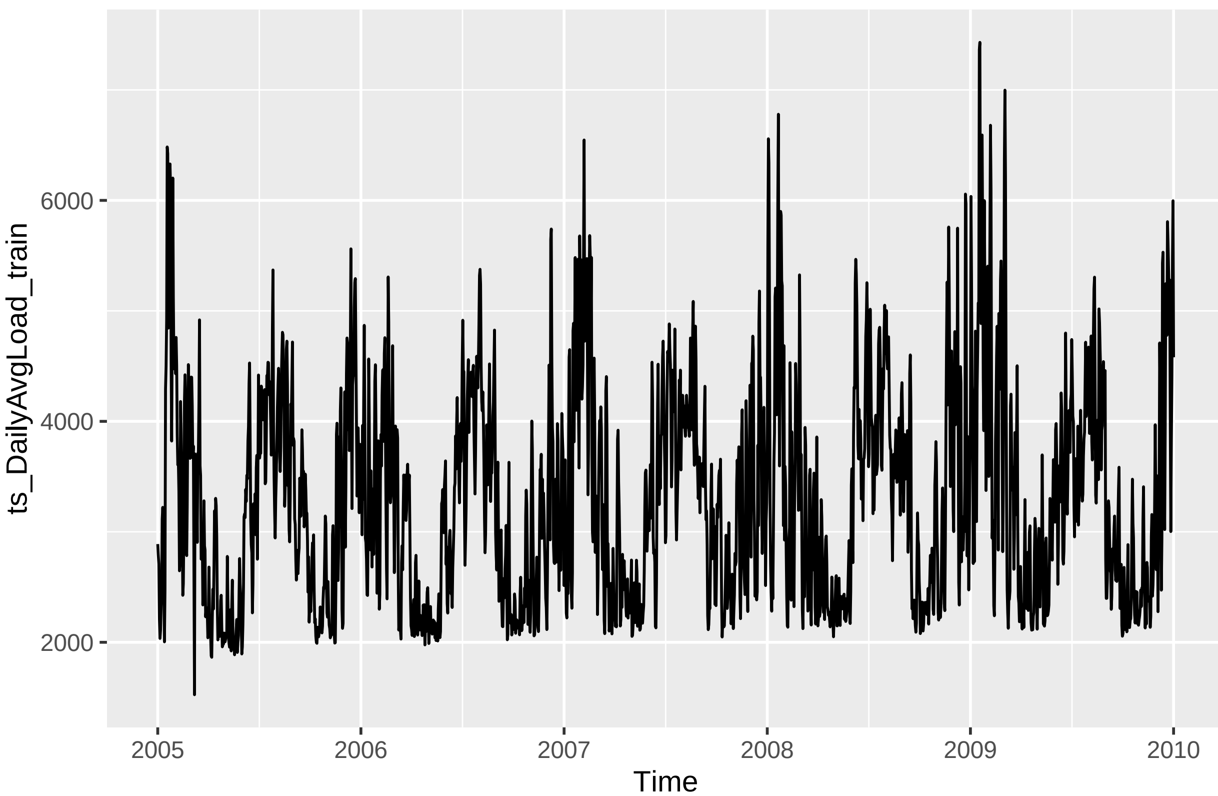

```
summary(DailyAvgLoad$Load)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1525    2453    3220    3382    4046    7545
```

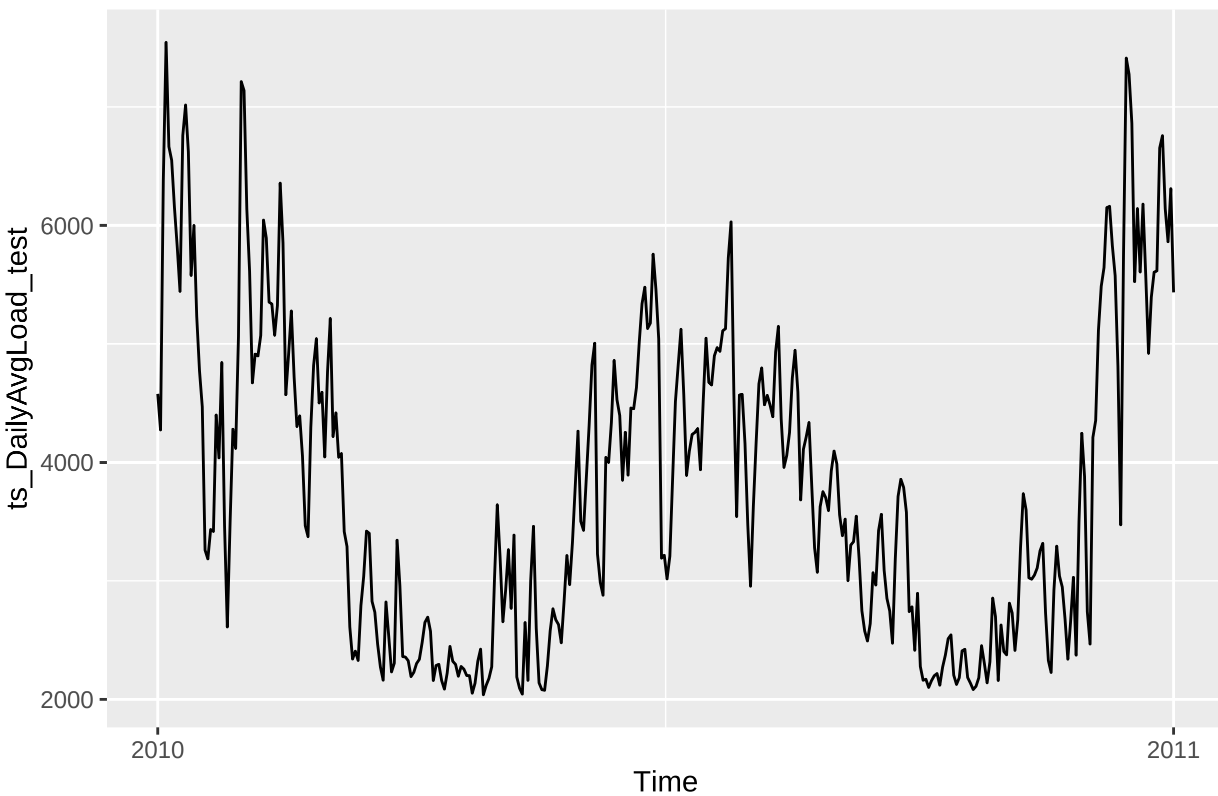
```
#Making time series
ts_DailyAvgLoad <- msts(DailyAvgLoad$Load,
                        seasonal.periods =c(7,365.25),
                        start=c(2005,1,1))

#Decomposing the series
ts_DailyAvgLoad %>% mstl() %>%
  autoplot()
```




```r
#subset for training
n_forecast = 365
ts_DailyAvgLoad_train <- subset(ts_DailyAvgLoad,
                                   end = length(ts_DailyAvgLoad)-n_forecast)

#subset for testing
ts_DailyAvgLoad_test <- subset(ts_DailyAvgLoad,
                                  start = length(ts_DailyAvgLoad)-n_forecast)

autoplot(ts_DailyAvgLoad_train)
```

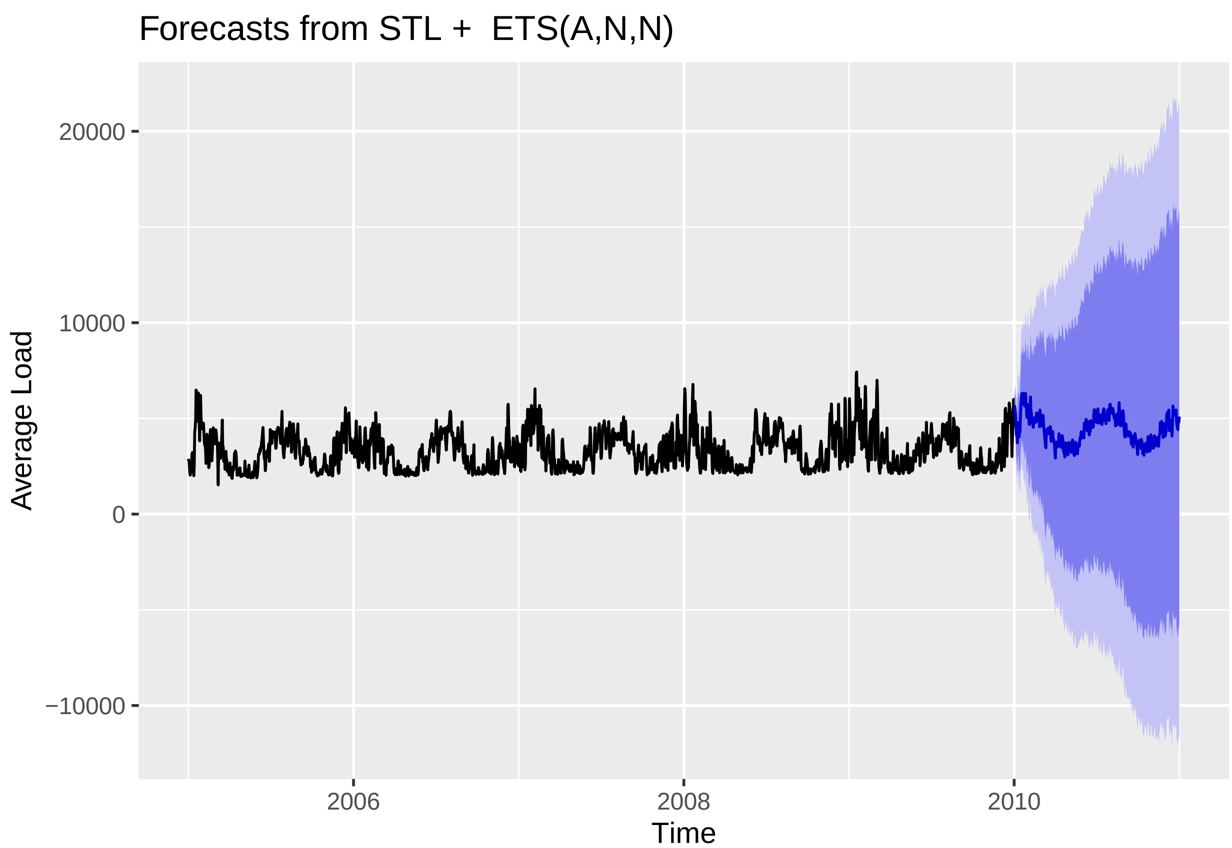



```
autoplot(ts_DailyAvgLoad_test)
```

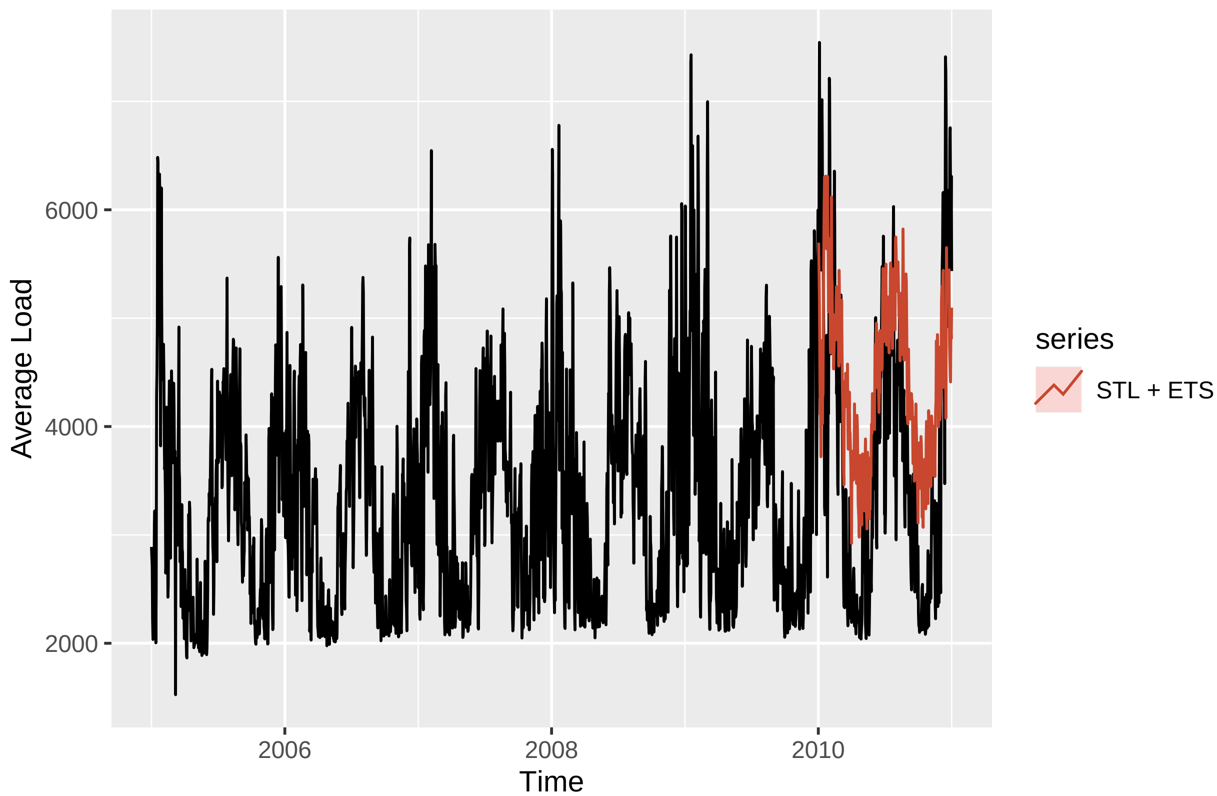


#Forecasting using STL+ETS (Model 1)

```
#Fit and forecast STL + ETS model to data
ETS_fit <- stlf(ts_DailyAvgLoad_train,h=365)

#Plot foresting results
autoplot(ETS_fit) + ylab("Average Load") #ANN = additive, no trend, non-seasonal
```

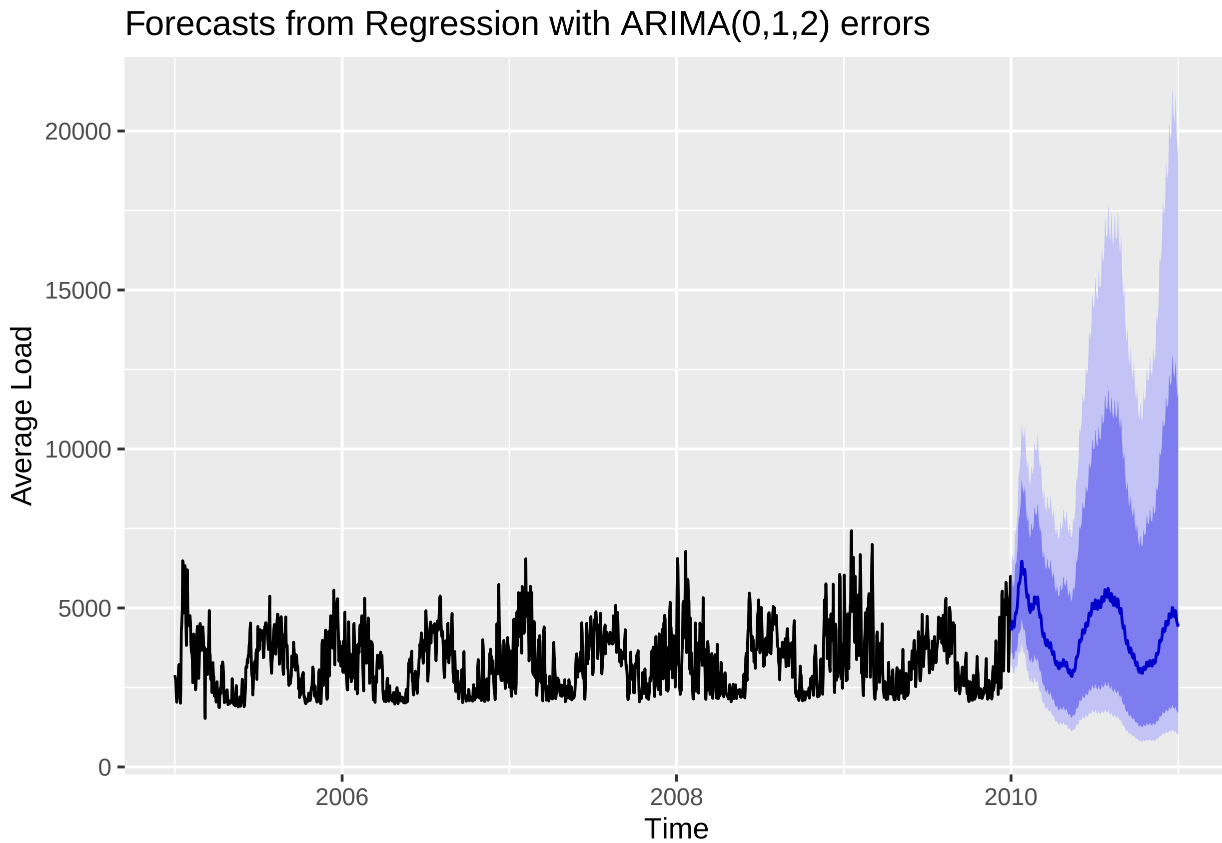



```r
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(ETS_fit, series="STL + ETS",PI=FALSE) +
  ylab("Average Load")
```

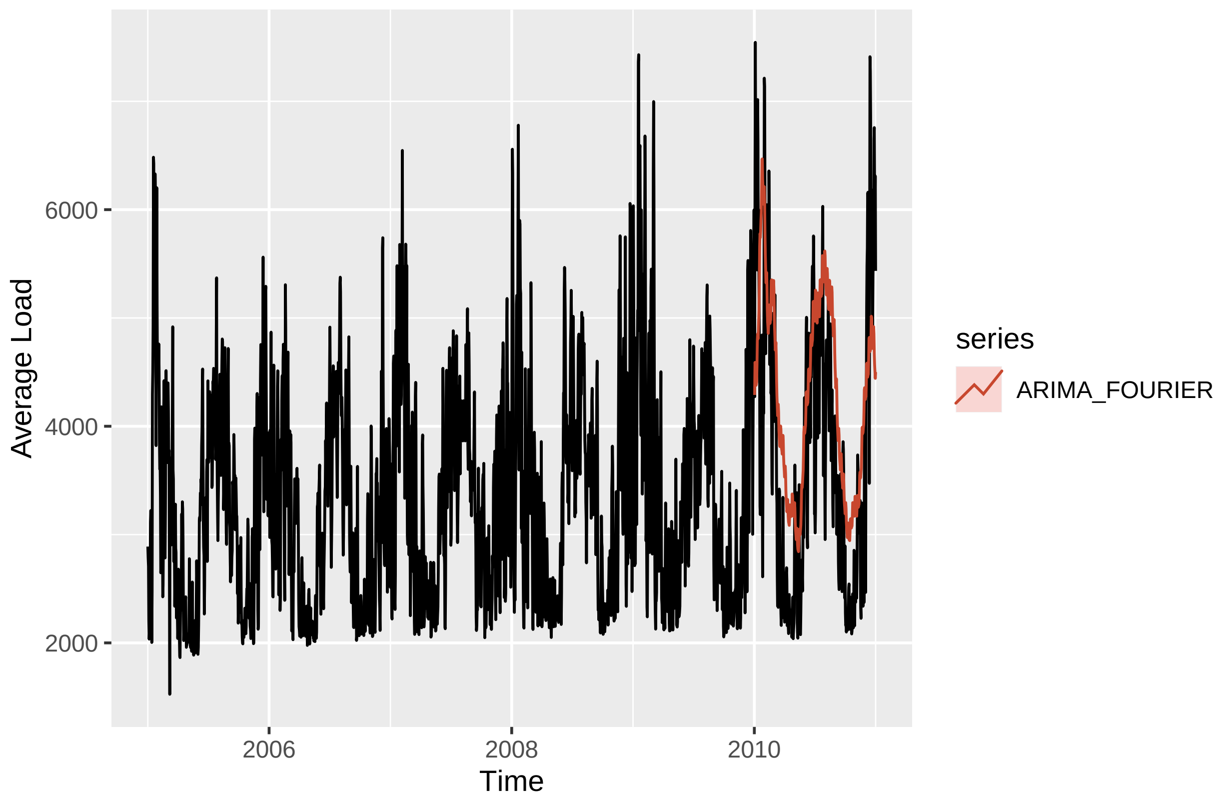


#Forecasting using ARIMA + Fourier (Model 2)

```
ARIMA_Four_fit <- auto.arima(ts_DailyAvgLoad_train,
                             seasonal=FALSE, #P,D,Q = 0
                             lambda=0,
                             xreg=fourier(ts_DailyAvgLoad_train,
                                          K=c(2,12))
                             )

#Forecast with ARIMA fit
ARIMA_Four_for <- forecast::forecast(ARIMA_Four_fit,
                           xreg=fourier(ts_DailyAvgLoad_train,
                                        K=c(2,12),
                                        h=365), #generates fourier terms 365 step ahead of time
                           h=365
                           )

#Plot foresting results
autoplot(ARIMA_Four_for) + ylab("Average Load")
```

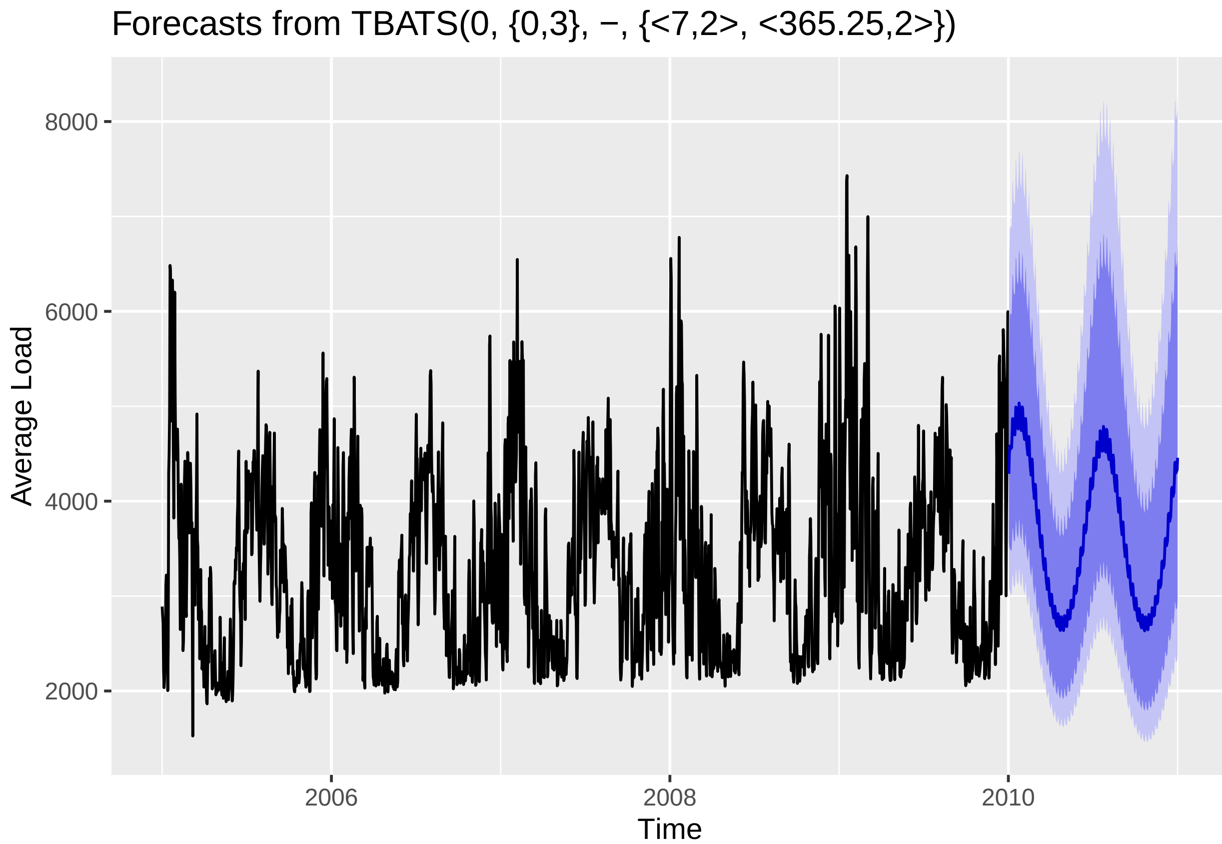



```r
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(ARIMA_Four_for, series="ARIMA_FOURIER",PI=FALSE) +
  ylab("Average Load")
```

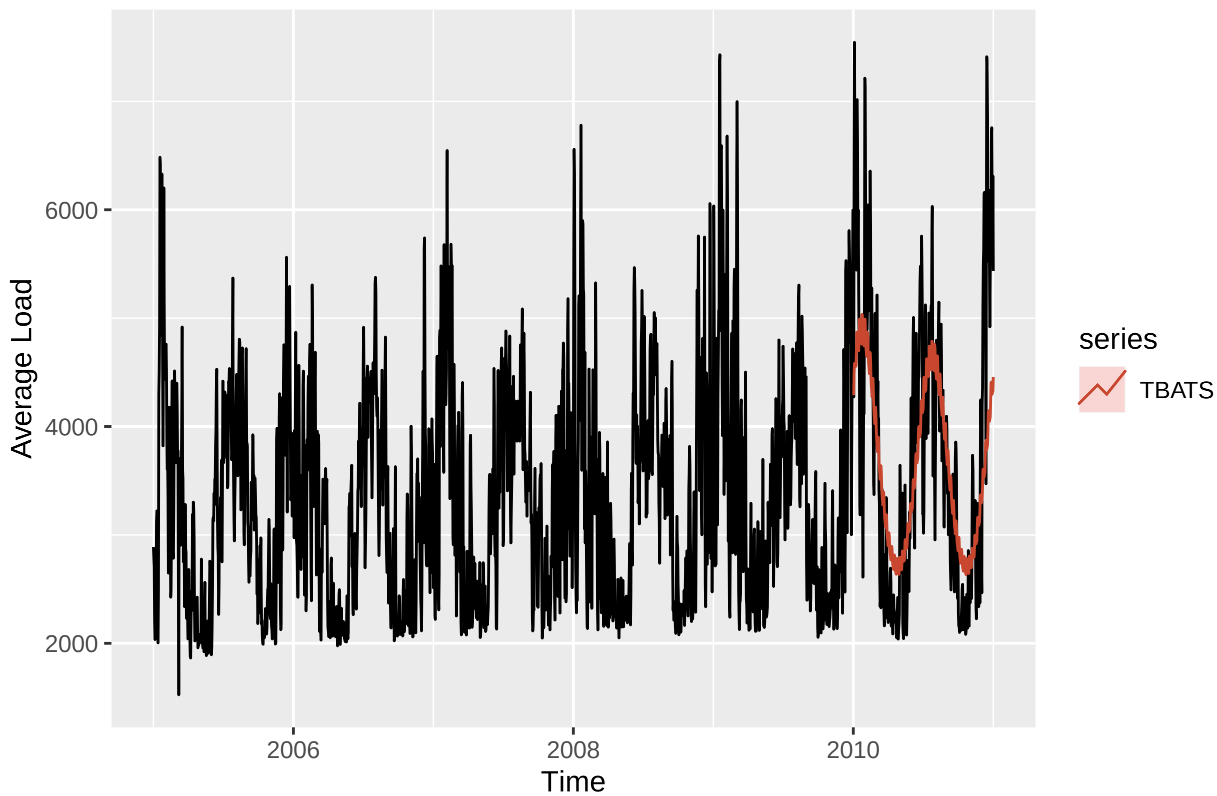


#Forecasting using TBATS (Model 3)

```
TBATS_fit <- tbats(ts_DailyAvgLoad_train)

TBATS_for <- forecast::forecast(TBATS_fit, h=365)

#Plot foresting results
autoplot(TBATS_for) +
  ylab("Average Load")
```

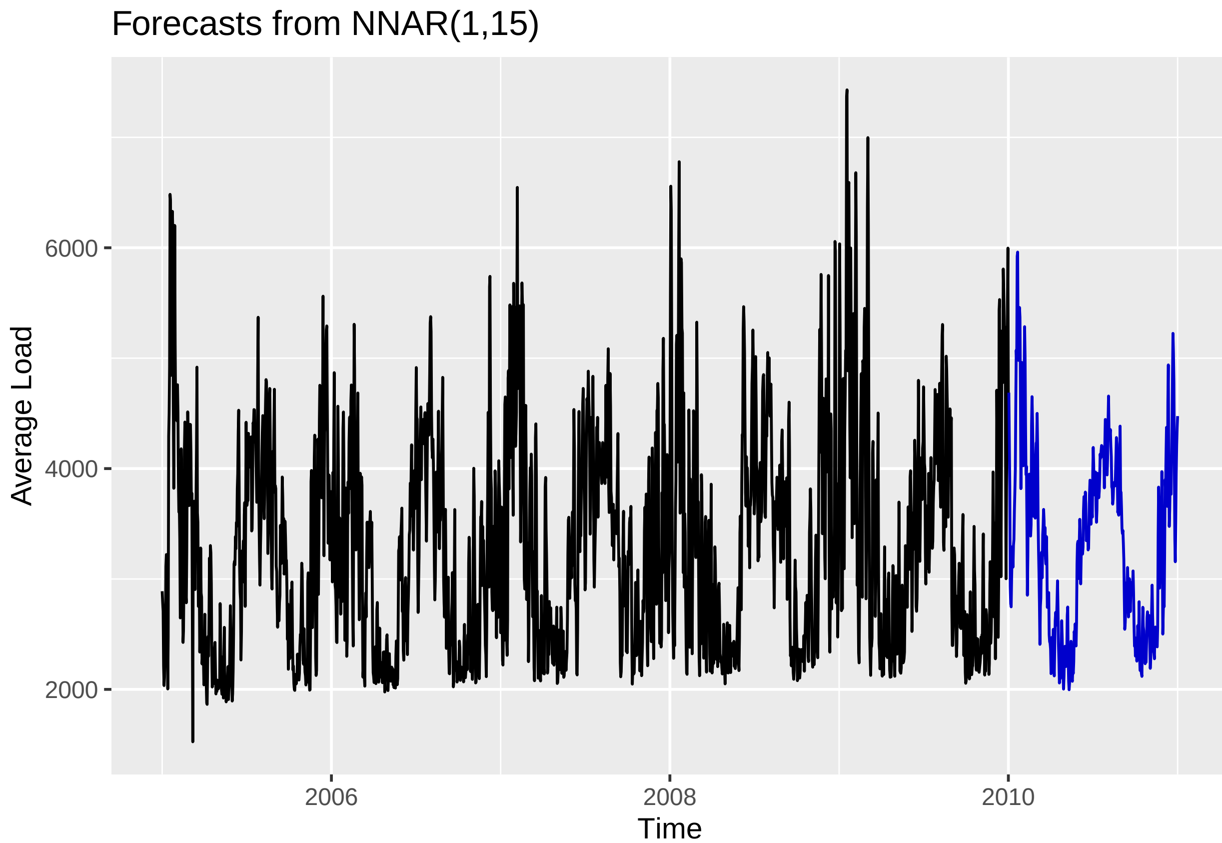



```
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(TBATS_for, series="TBATS",PI=FALSE)+
  ylab("Average Load")
```

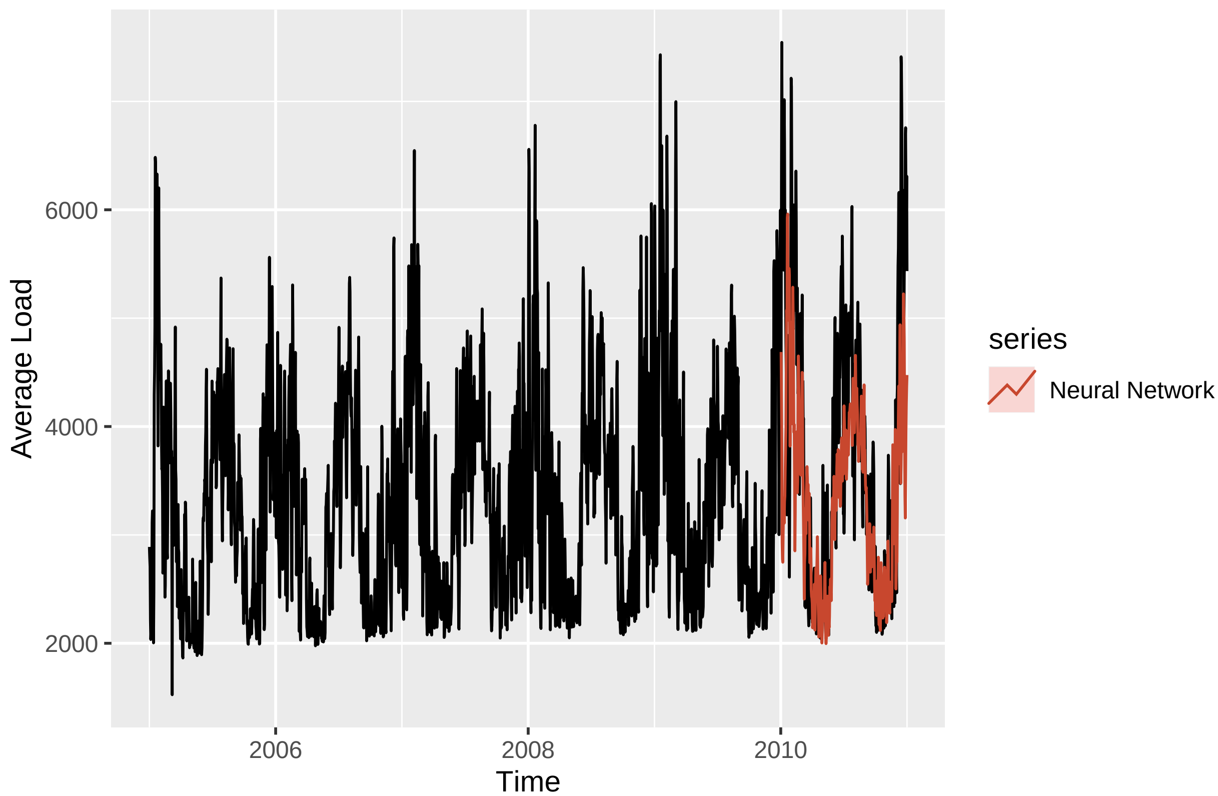


#Forecasting using Neural Network Time Series, p=1,P=0 (Model 4)

```
NN_fit <- nnetar(ts_DailyAvgLoad_train,p=1,P=0,xreg=fourier(ts_DailyAvgLoad_train, K=c(2,12)))

NN_for <- forecast::forecast(NN_fit, h=365,xreg=fourier(ts_DailyAvgLoad_train,
                                          K=c(2,12),h=365))

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




```r
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(NN_for, series="Neural Network",PI=FALSE)+
  ylab("Average Load")
```



Checking accuracy of the models

```
#Model 1: STL + ETS
ETS_scores <- accuracy(ETS_fit$mean,ts_DailyAvgLoad_test)

#Model 2: ARIMA + Fourier
ARIMA_scores <- accuracy(ARIMA_Four_for$mean,ts_DailyAvgLoad_test)

# Model 3:  TBATS
TBATS_scores <- accuracy(TBATS_for$mean,ts_DailyAvgLoad_test)

# Model 4:  Neural Network
NN_scores <- accuracy(NN_for$mean,ts_DailyAvgLoad_test)
```

Comparing Performance metrics

```
#creating data frame
scores <- as.data.frame(
  rbind(ETS_scores, ARIMA_scores, TBATS_scores, NN_scores)
  )
```

```r
row.names(scores) <- c("STL+ETS", "ARIMA+Fourier","TBATS","NN")

#choosing model with lowest RMSE
best_model_index_RMSE <- which.min(scores[,"RMSE"])
#choosing model with lowest MAPE
best_model_index_MAPE <- which.min(scores[,"MAPE"])

#printing results
cat("The best model by RMSE is:", row.names(scores[best_model_index_RMSE,]))
```

```
## The best model by RMSE is: TBATS
```

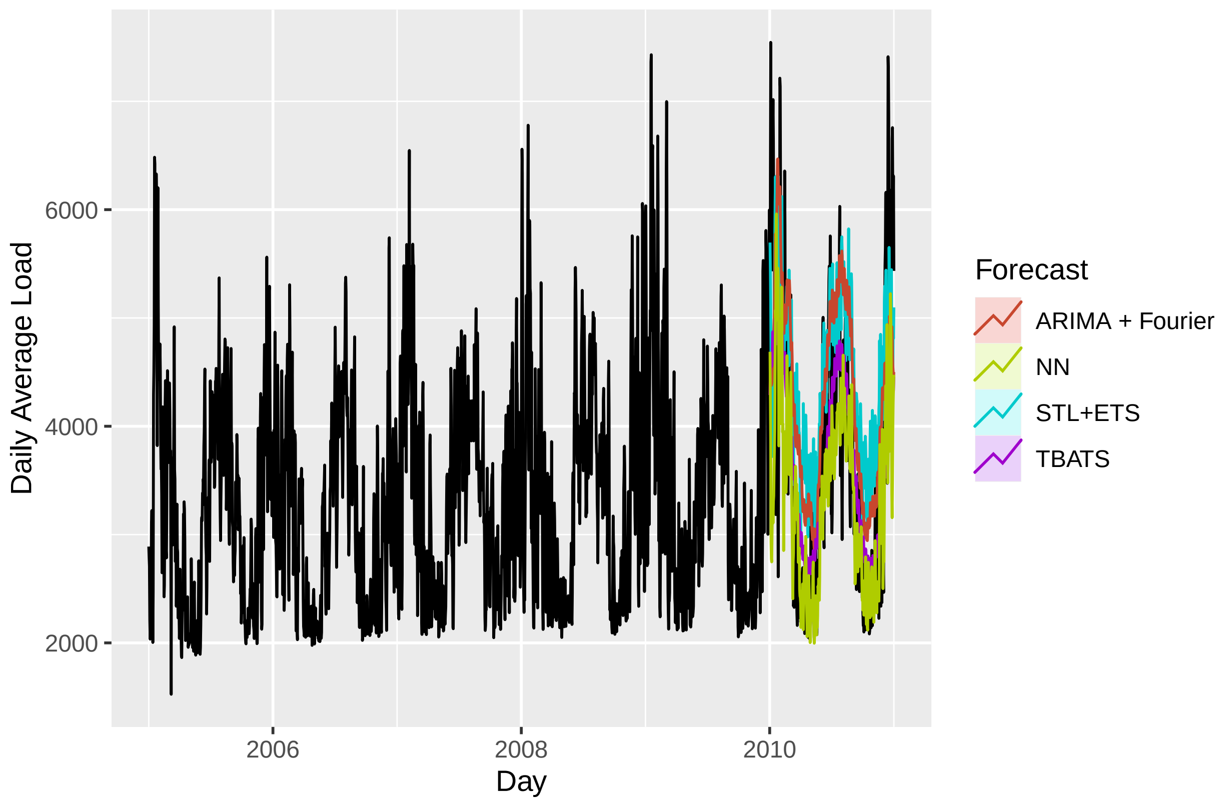
```r
cat("The best model by MAPE is:", row.names(scores[best_model_index_MAPE,]))
```

```
## The best model by MAPE is: NN
```

Table 1: Forecast Accuracy for Daily Average Load

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
STL+ETS	-664.6880	1230.3995	1039.9902	-27.15911	33.22800	0.79970	2.80673
ARIMA+Fourier	-506.1649	1080.7349	898.1974	-20.91309	27.48365	0.84141	2.29946
TBATS	118.6917	912.2884	688.5789	-3.18910	18.35719	0.82733	1.54891
NN	441.6998	1115.3130	763.6788	6.48168	17.85107	0.81983	1.65161

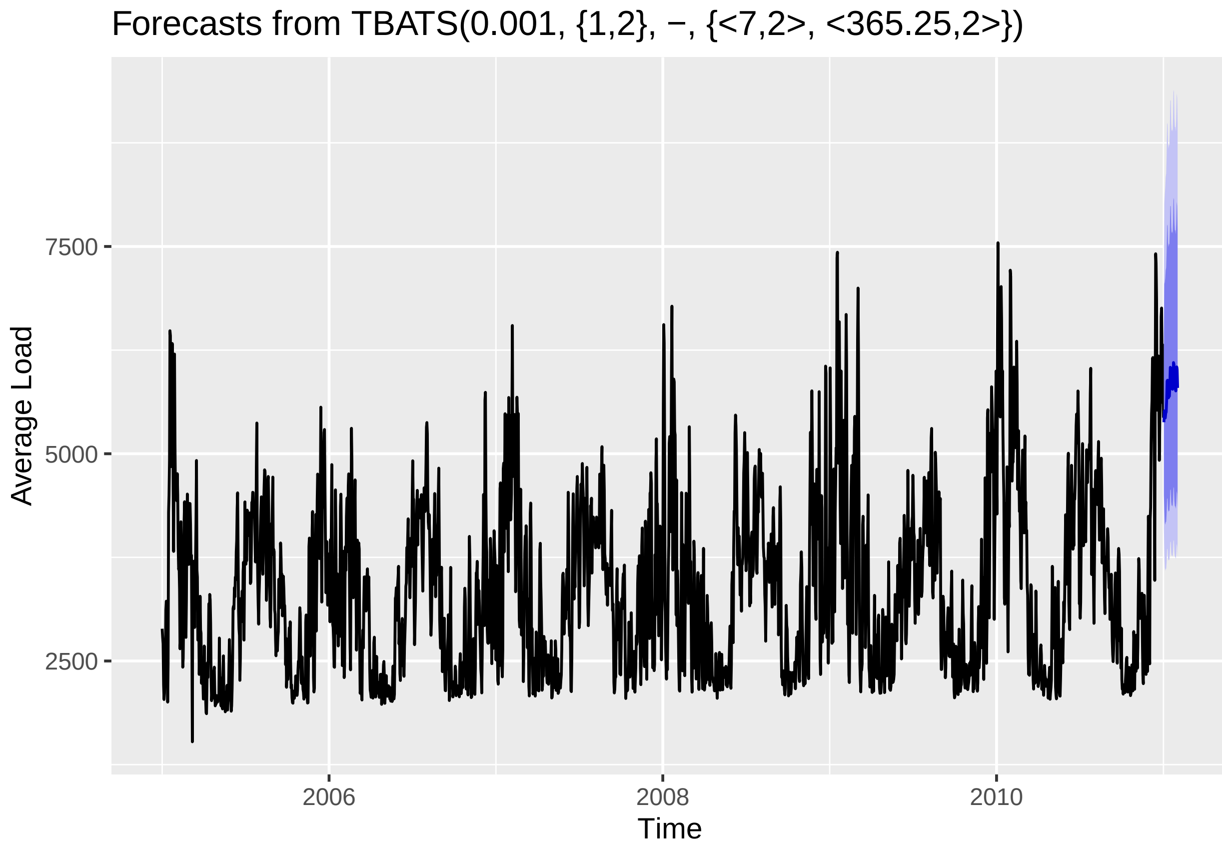
```r
autoplot(ts_DailyAvgLoad) +
  autolayer(ETS_fit, PI=FALSE, series="STL+ETS") +
  autolayer(ARIMA_Four_for, PI=FALSE, series="ARIMA + Fourier") +
  autolayer(TBATS_for,PI=FALSE, series="TBATS") +
  autolayer(NN_for,PI=FALSE, series="NN") +
  xlab("Day") + ylab("Daily Average Load") +
  guides(colour=guide_legend(title="Forecast"))
```



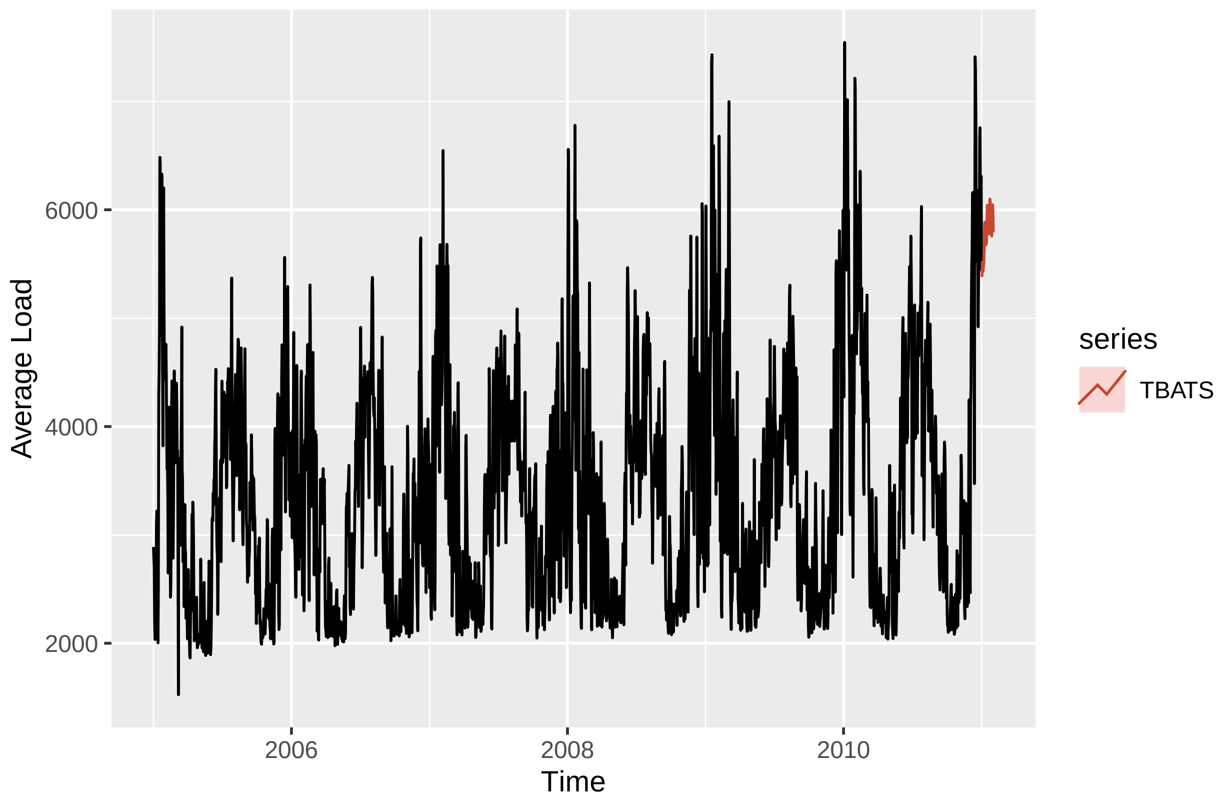
#Forecasting Daily Demand for January 2011

Since TBATS was determined to be the best model by RMSE:

```
TBATS11_fit <- tbats(ts_DailyAvgLoad)

TBATS11_for <- forecast::forecast(TBATS11_fit, h=31)

#Plot foresting results
autoplot(TBATS11_for) +
  ylab("Average Load")
```



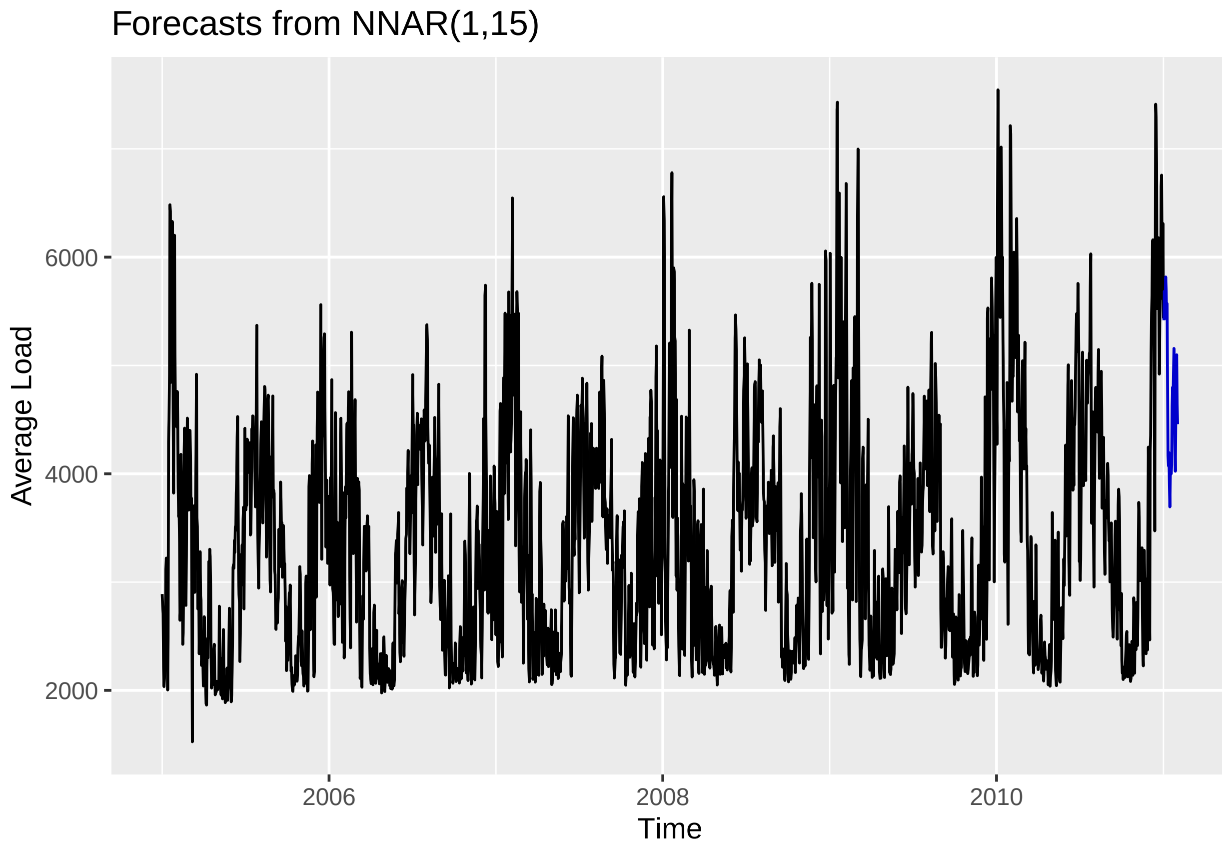

```r
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(TBATS11_for, series="TBATS",PI=FALSE)+
  ylab("Average Load")
```



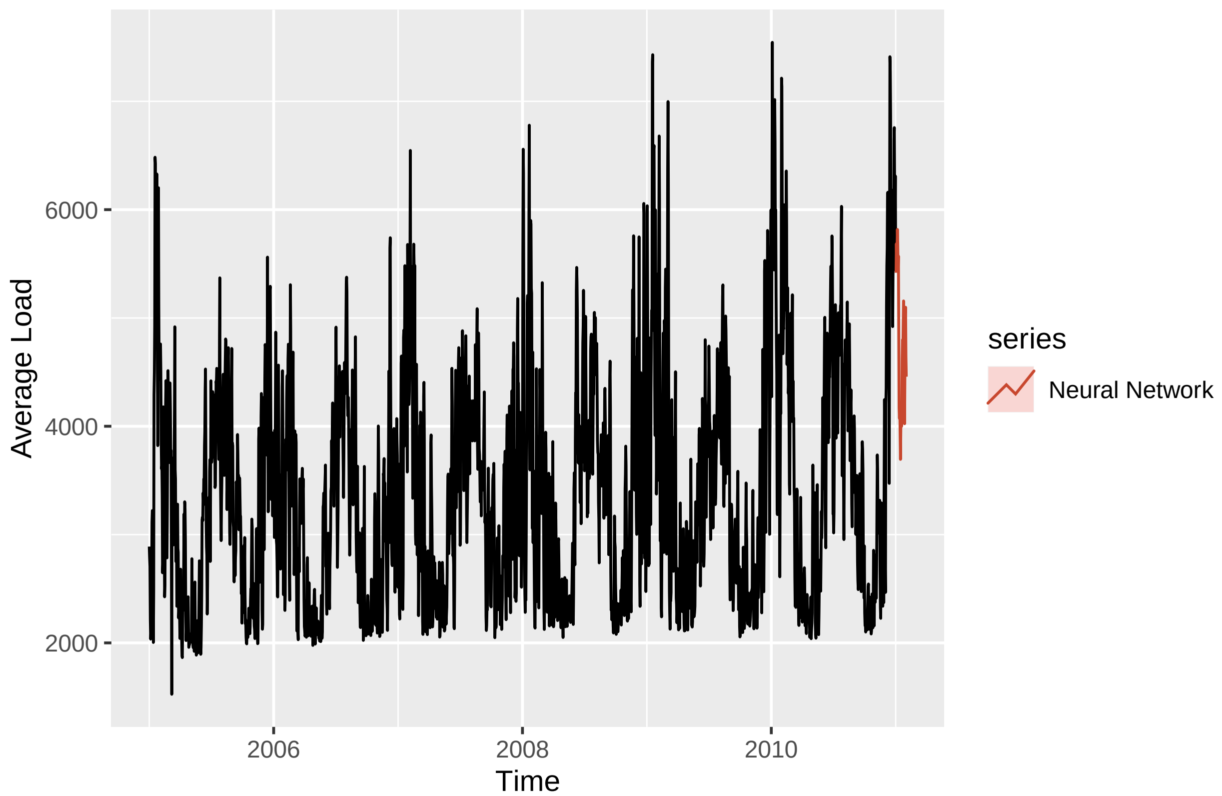

```
#write.csv(TBATS11_for, file="./Outputs/TBATS_forecast.csv")
```

Since Neural Network was determined to be the best model by MAPE:

```
NN11_fit <- nnetar(ts_DailyAvgLoad,p=1,P=0,xreg=fourier(ts_DailyAvgLoad, K=c(2,12)))

NN11_for <- forecast::forecast(NN11_fit, h=31,xreg=fourier(ts_DailyAvgLoad,
                                          K=c(2,12),h=31))

#Plot foresting results
autoplot(NN11_for) +
  ylab("Average Load")
```



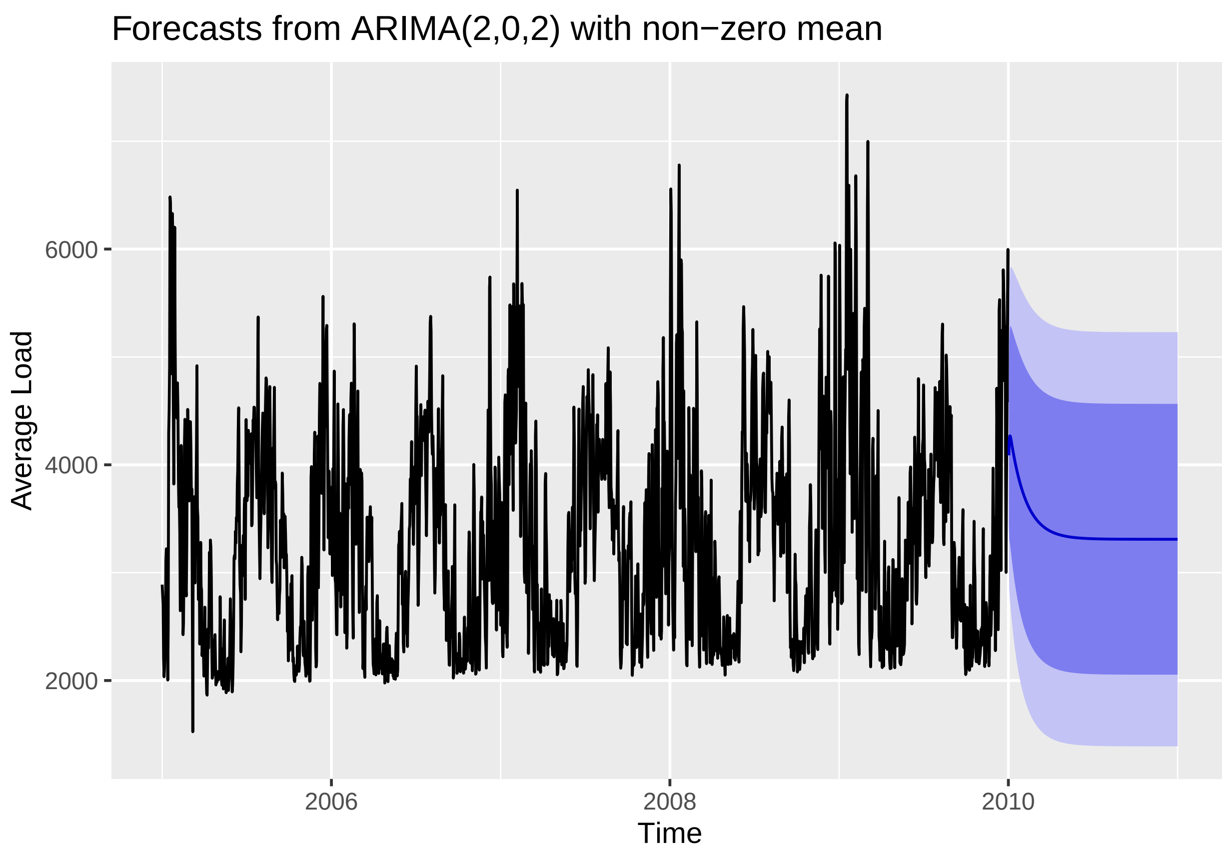

```
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(NN11_for, series="Neural Network",PI=FALSE)+
  ylab("Average Load")
```



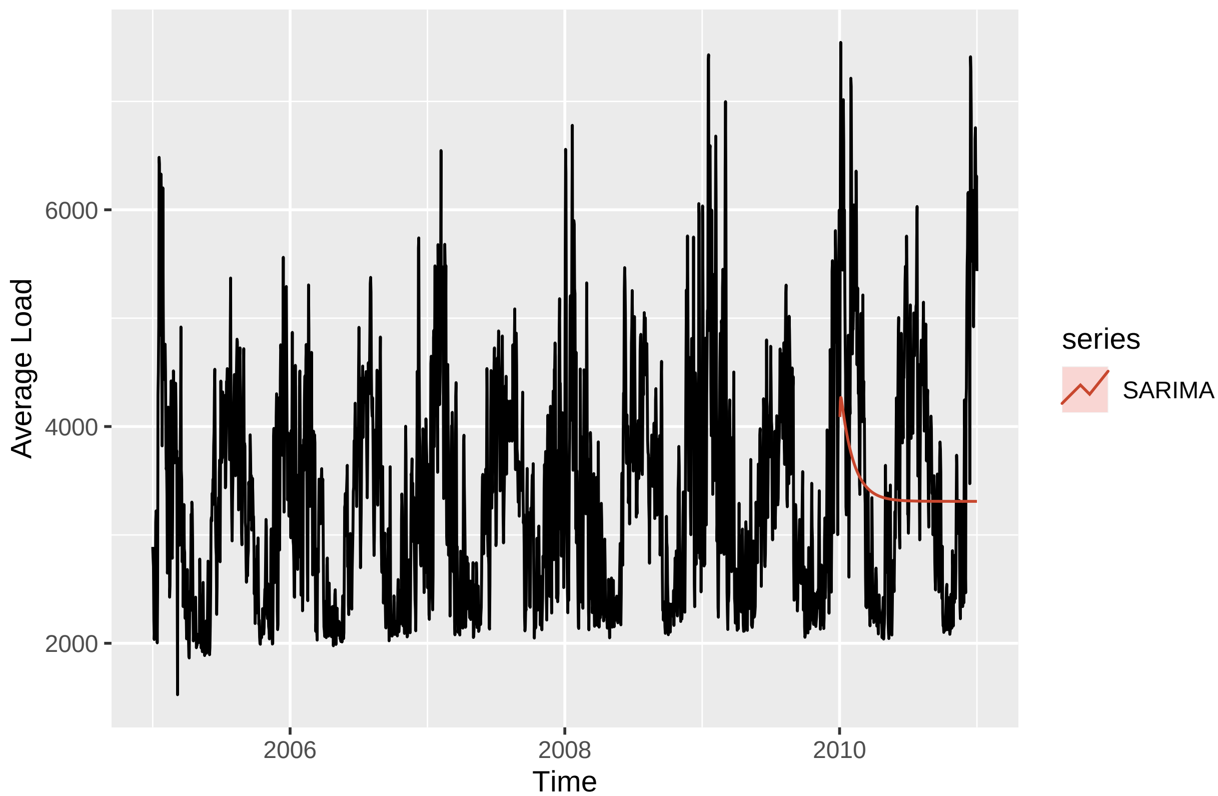

```
#write.csv(NN11_for, file="./Outputs/NN_forecast.csv")
```

#Forecasting using ARIMA (Model 5)

```
autofit_SARIMA <- auto.arima(ts_DailyAvgLoad_train)

SARIMA_for <- forecast::forecast(object = autofit_SARIMA, h = n_forecast)

#Plot results
autoplot(SARIMA_for) +
  ylab("Average Load")
```



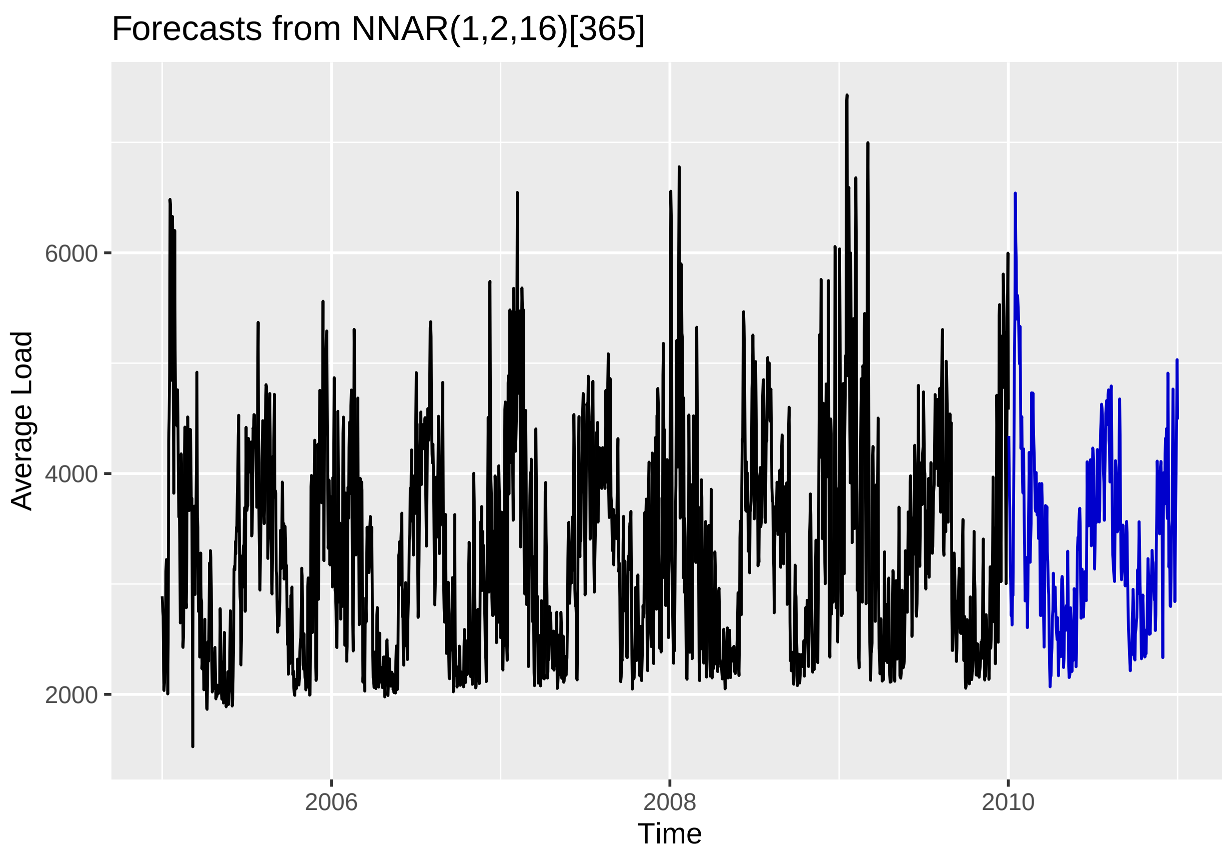

```r
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(SARIMA_for, series="SARIMA",PI=FALSE)+
  ylab("Average Load")
```



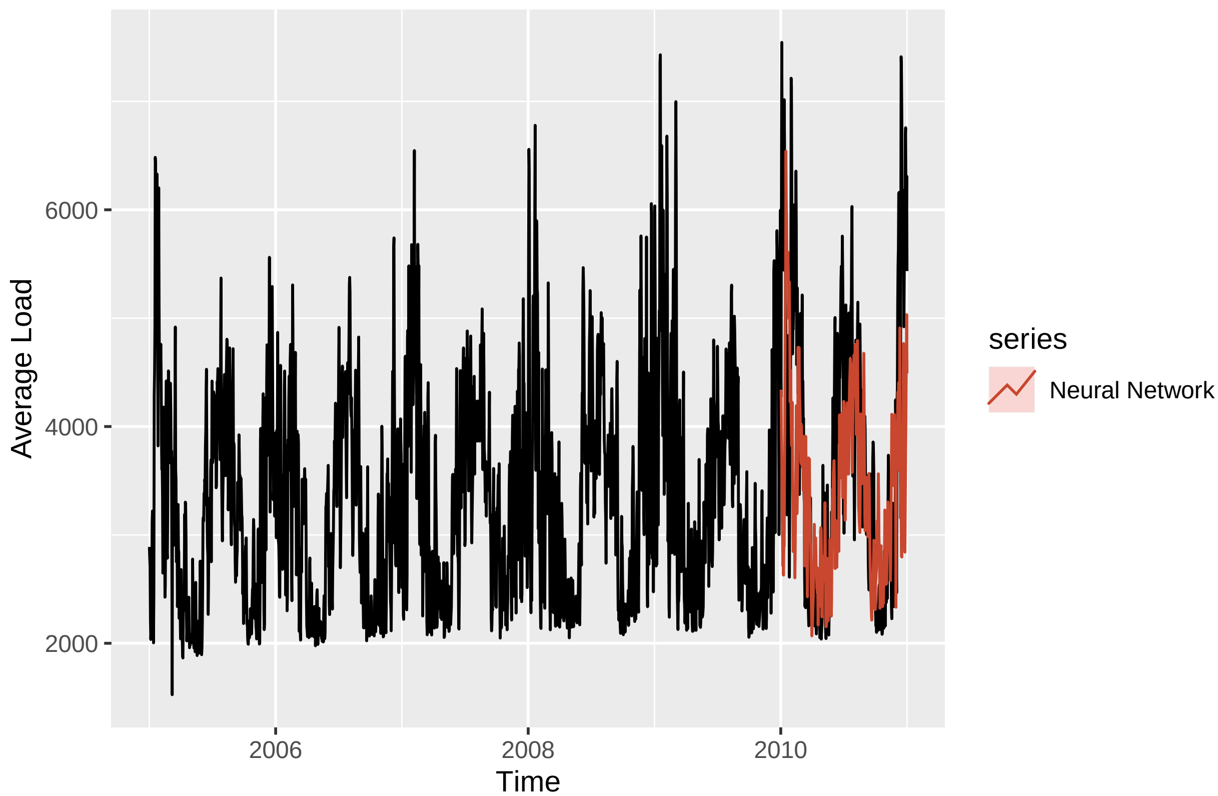
Not a great fit, but the results are valuable nonetheless. Since ARIMA(2,0,2) was selected, Will try incorporating P=2 into the neural network.

#Forecasting using Neural Network & p = 1, P = 2 (Model 6)

```
NN_fit_P2 <- nnetar(ts_DailyAvgLoad_train,p=1,P=2,xreg=fourier(ts_DailyAvgLoad_train, K=c(2,12)))

NN_for_P2 <- forecast::forecast(NN_fit_P2, h=365,xreg=fourier(ts_DailyAvgLoad_train,
                                                              K=c(2,12),h=365))

#Plot foresting results
autoplot(NN_for_P2) +
  ylab("Average Load")
```

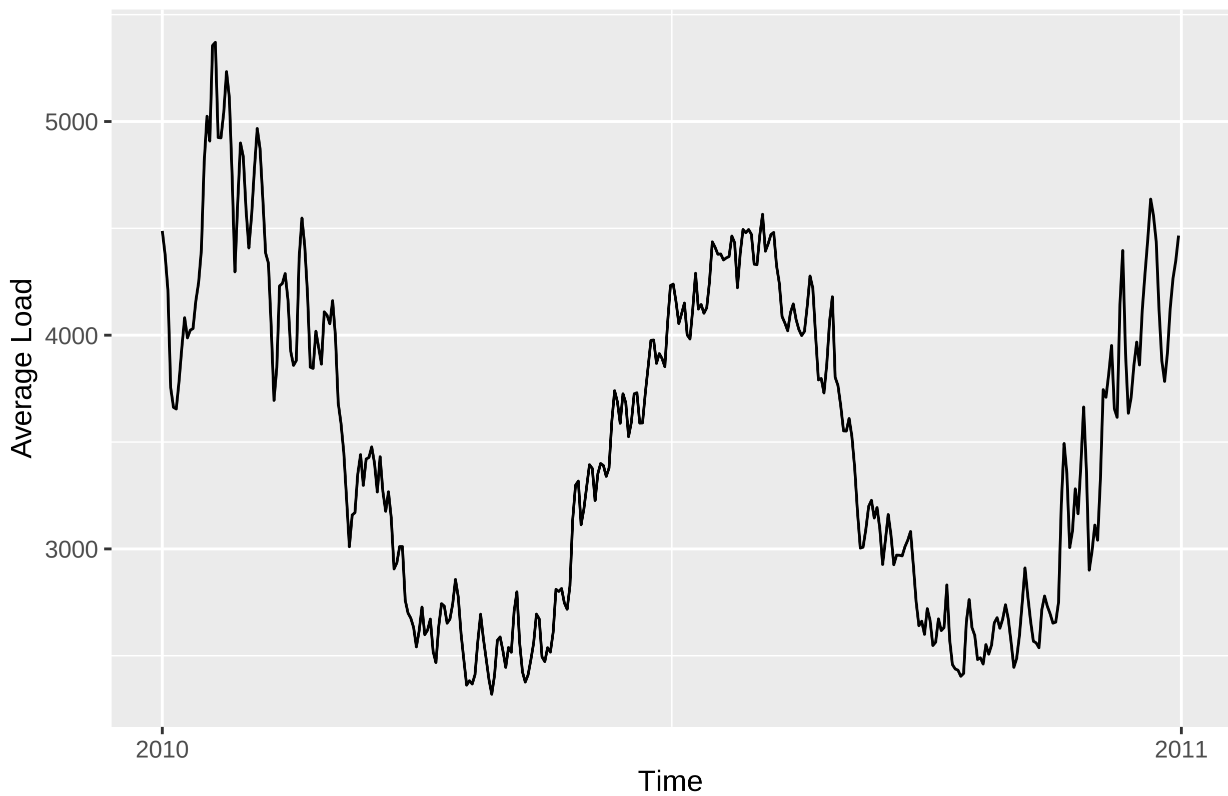



```r
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(NN_for_P2, series="Neural Network",PI=FALSE)+
  ylab("Average Load")
```



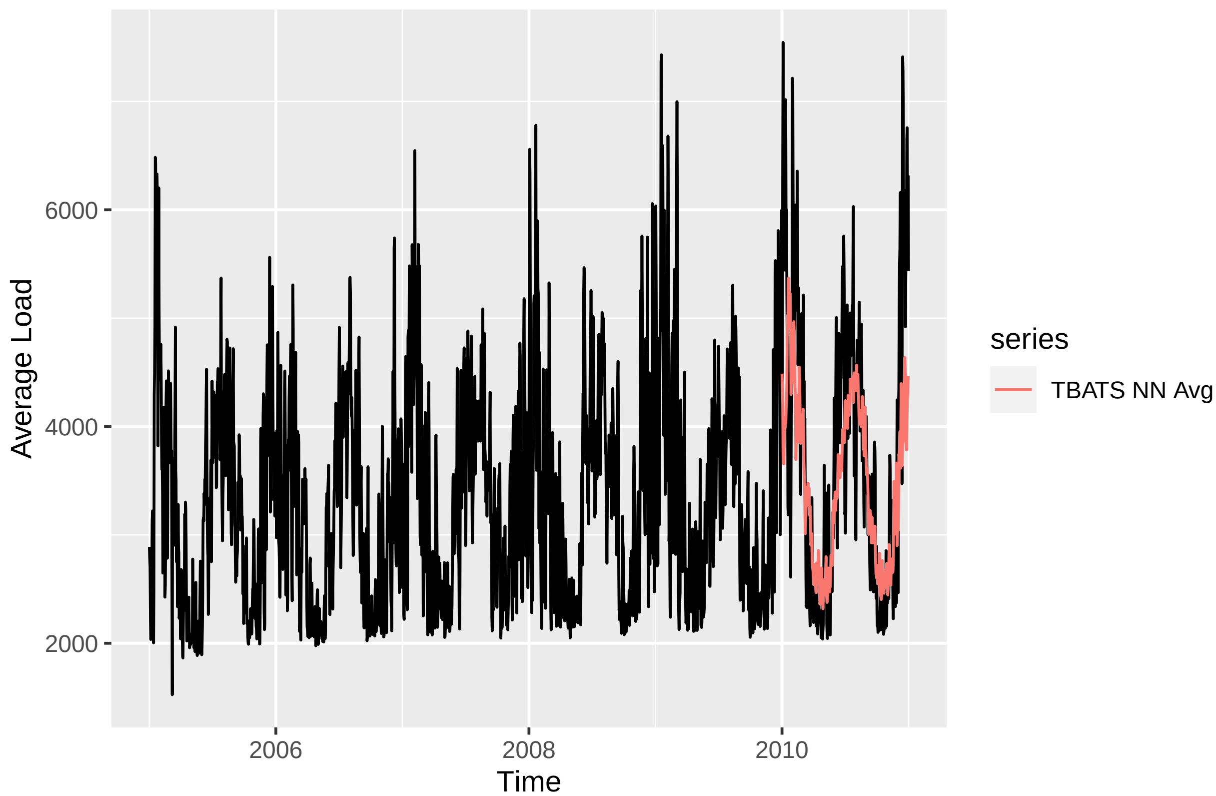
#Averaging forecast from Neural Network & TBATS (Model 7)

```
#Model 7: subset for testing
TBATS_NN_avg_test <- msts(rowMeans(cbind(TBATS_for$mean,NN_for$mean)),seasonal.periods =c(7,365.25),sta

#Plot foresting results
autoplot(TBATS_NN_avg_test) +
  ylab("Average Load")
```



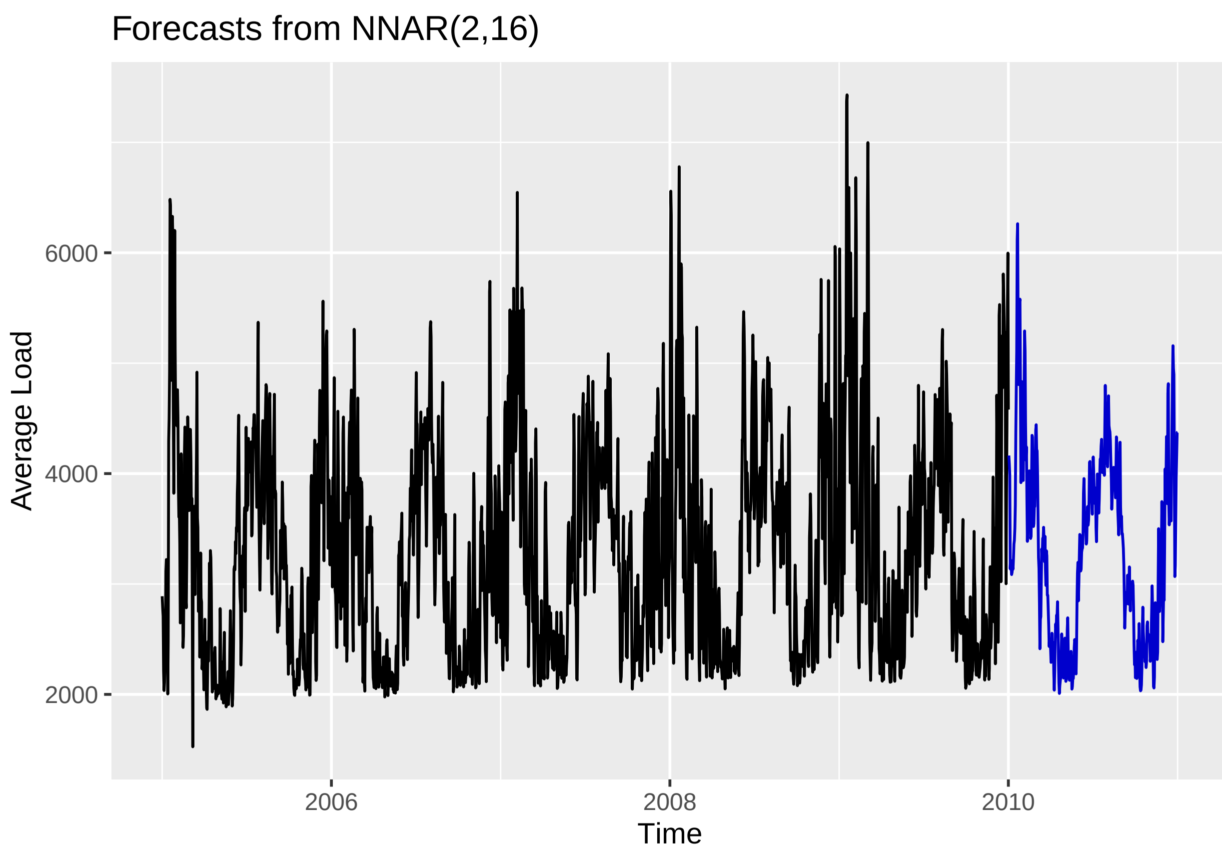

```
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(TBATS_NN_avg_test, series="TBATS NN Avg",PI=FALSE)+
  ylab("Average Load")
```

```
## Warning: Ignoring unknown parameters: PI
```

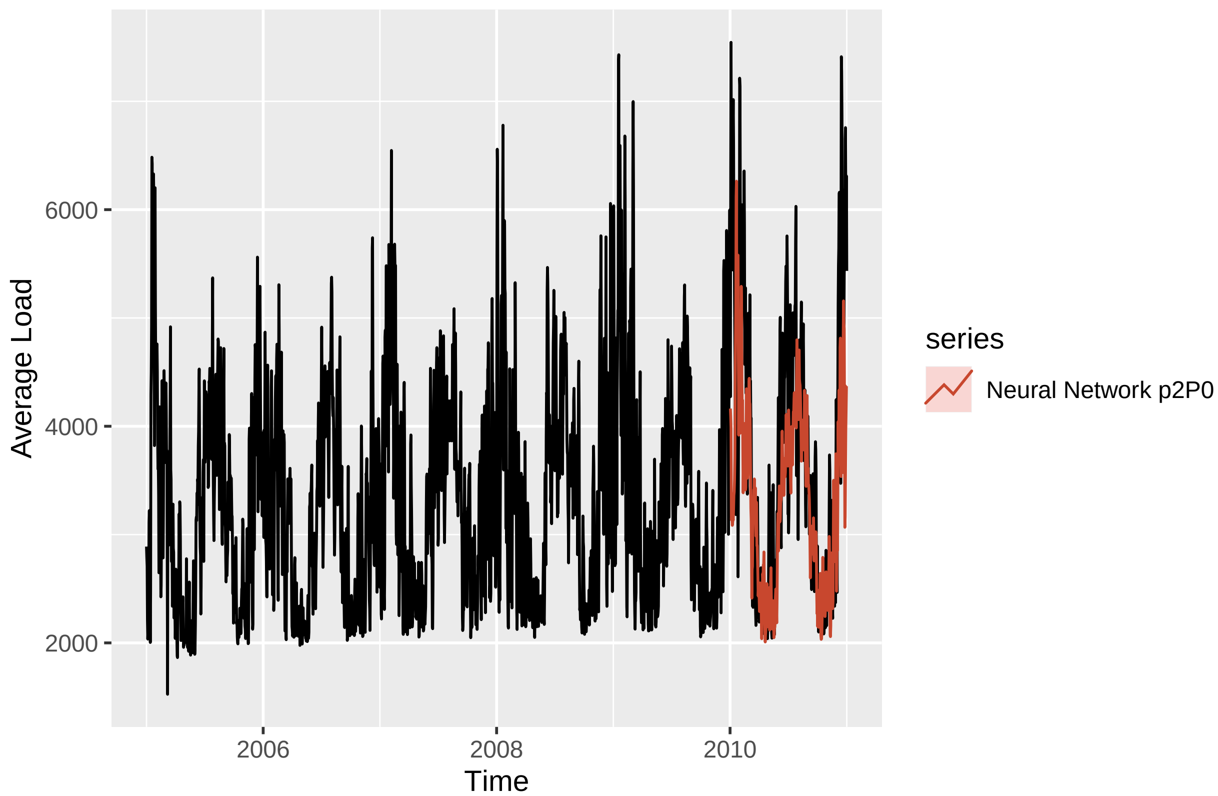


#Forecasting using Neural Network & different p=2, P=0 (Model 8)

```
NN_fit_p2P0 <- nnetar(ts_DailyAvgLoad_train,p=2,P=0,xreg=fourier(ts_DailyAvgLoad_train, K=c(2,12)))

NN_for_p2P0 <- forecast::forecast(NN_fit_p2P0, h=365,xreg=fourier(ts_DailyAvgLoad_train,
                                          K=c(2,12),h=365))

#Plot foresting results
autoplot(NN_for_p2P0) +
  ylab("Average Load")
```

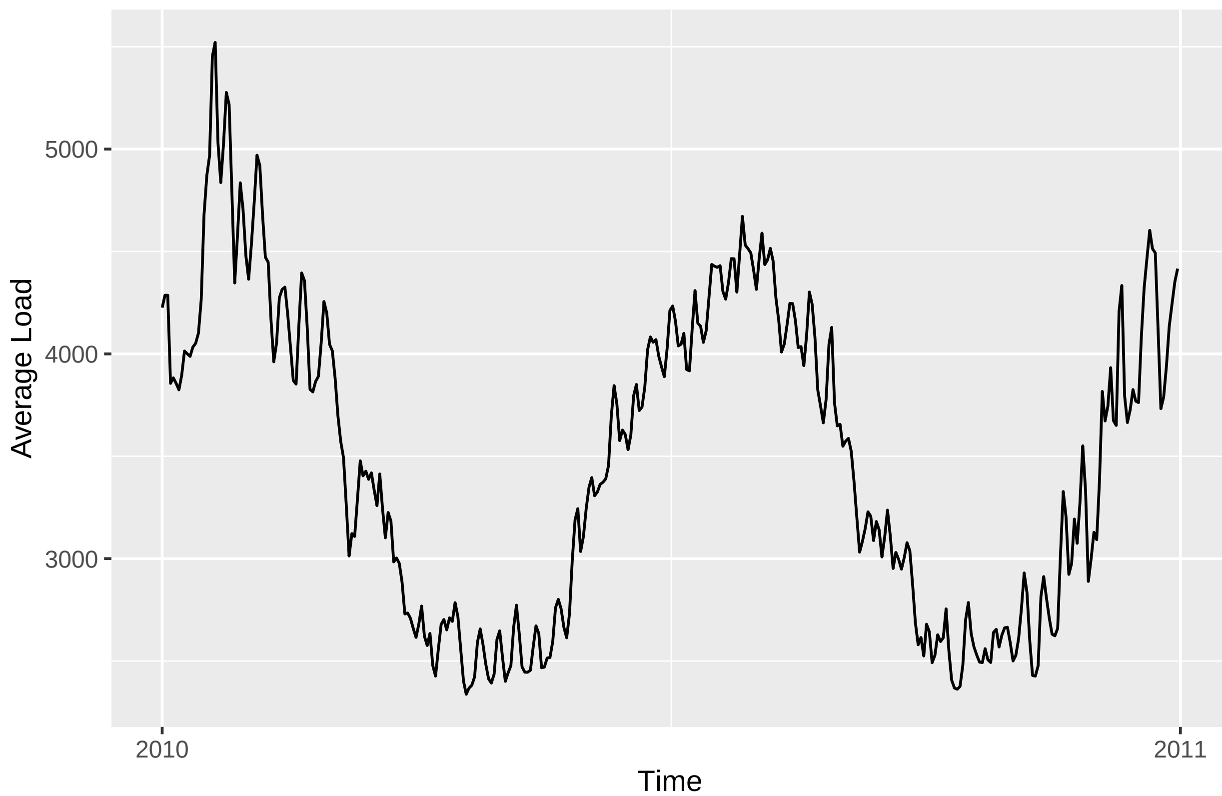



```r
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(NN_for_p2P0, series="Neural Network p2P0",PI=FALSE)+
  ylab("Average Load")
```



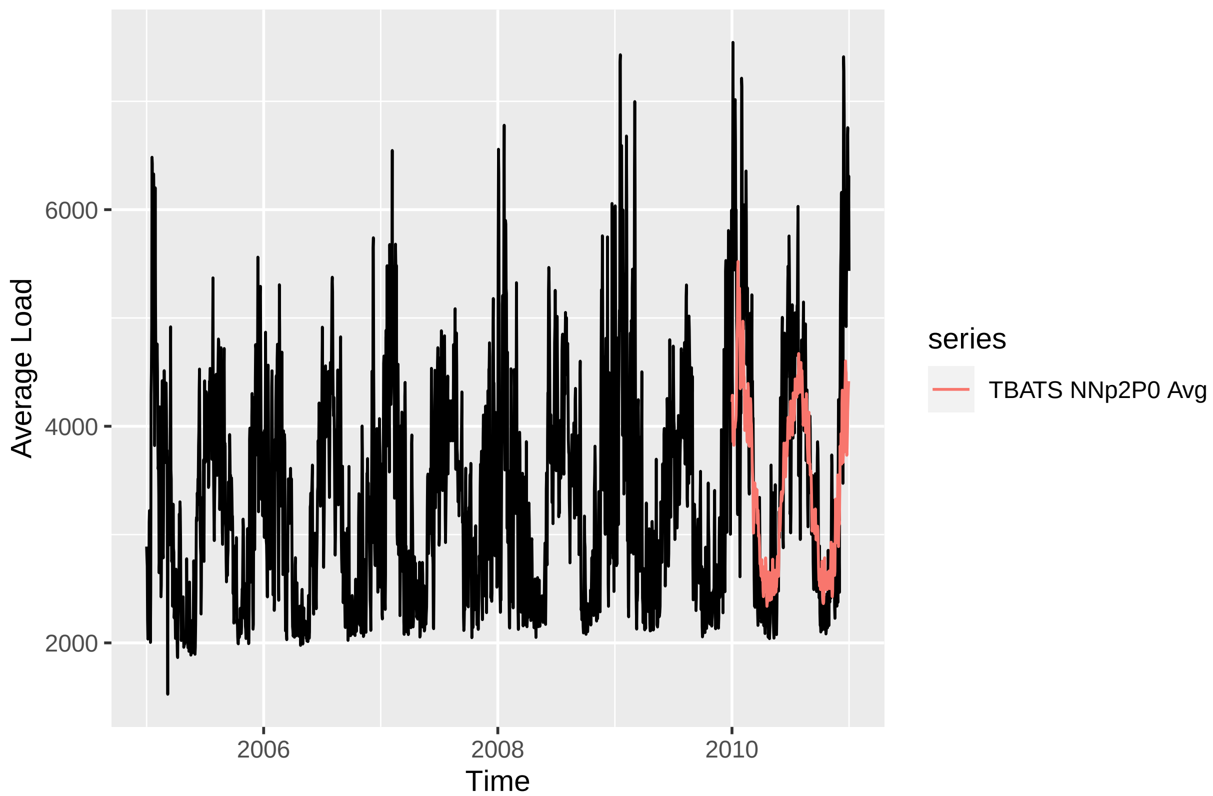
#Averaging forecast from Neural Network (p=2,P=0) & TBATS (Model 9)

```
#Model 9: subset for testing
TBATS_NNp2P0_avg_test <- msts(rowMeans(cbind(TBATS_for$mean,NN_for_p2P0$mean)),seasonal.periods =c(7,365

#Plot foresting results
autoplot(TBATS_NNp2P0_avg_test) +
  ylab("Average Load")
```




```r
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(TBATS_NNp2P0_avg_test, series="TBATS NNp2P0 Avg ",PI=FALSE)+
  ylab("Average Load")
```

```
## Warning: Ignoring unknown parameters: PI
```



Checking accuracy of the new models

```
# Model 5:  SARIMA
SARIMA_scores <- accuracy(SARIMA_for$mean,ts_DailyAvgLoad_test)

# Model 6: Neural Network with P = 2

NN_P2_scores <- accuracy(NN_for_P2$mean,ts_DailyAvgLoad_test)

# Model 7 Average of Neural Network & TBATS
TBATS_NN_avg_scores <- accuracy(TBATS_NN_avg_test,ts_DailyAvgLoad_test)

# Model 8 Neural Network with p = 0, P = 2
NN_p2P0_scores <- accuracy(NN_for_p2P0$mean,ts_DailyAvgLoad_test)

# Model 9 Average of Neural Network (p=2,P=0) & TBATS
TBATS_NNp2P0_avg_scores <- accuracy(TBATS_NNp2P0_avg_test,ts_DailyAvgLoad_test)

scores <-  rbind(ETS_scores, ARIMA_scores, TBATS_scores, NN_scores,SARIMA_scores,NN_P2_scores,TBATS_NN_a
row.names(scores) <- c("STL+ETS","ARIMA+Fourier","TBATS","NN","SARIMA","NNP2","NN+TBATS Avg","NNp2P0","N
```

As indicated by higher MAPE and RMSE, the Neural Network with P=2 seems to be a worse fit than P=0. However, adjusting for non-seasonal lags instead, with p=2, performs better than seasonal lags of P=2. The

Table 2: Forecast Accuracy for Daily Average Load

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
STL+ETS	-664.6880	1230.3995	1039.9902	-27.15911	33.22800	0.79970	2.80673
ARIMA+Fourier	-506.1649	1080.7349	898.1974	-20.91309	27.48365	0.84141	2.29946
TBATS	118.6917	912.2884	688.5789	-3.18910	18.35719	0.82733	1.54891
NN	441.6998	1115.3130	763.6788	6.48168	17.85107	0.81983	1.65161
SARIMA	360.0236	1302.5216	1059.3662	-1.12886	28.26034	0.90536	2.28572
NNP2	368.3427	1260.2871	908.0166	2.58131	22.41993	0.85467	1.99283
NN+TBATS Avg	277.8536	977.0065	696.2904	1.60439	17.19193	0.81736	1.50953
NNp2P0	451.1385	1104.7318	755.8021	6.94118	17.71573	0.82371	1.64153
NNp2P0+TBATS Avg	282.5730	974.9215	692.2798	1.81521	17.08414	0.81235	1.50610

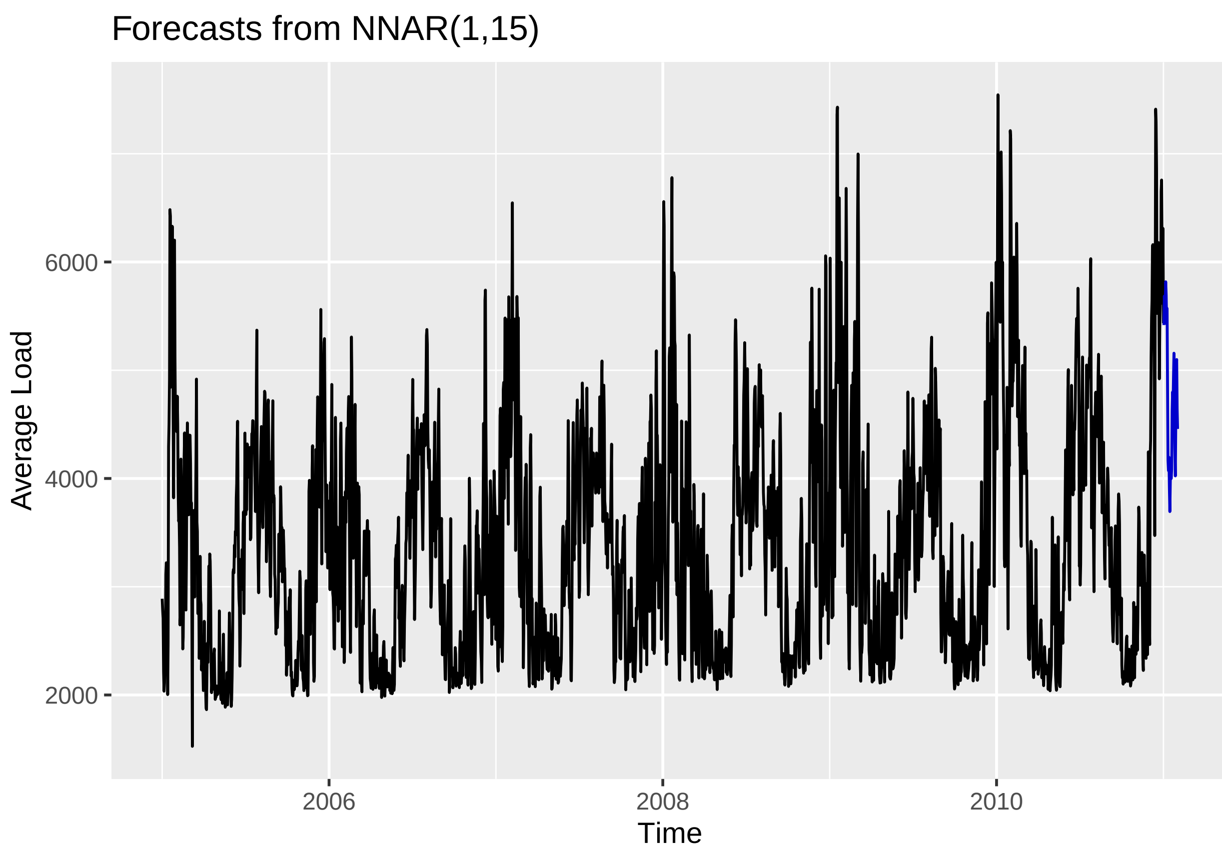
average of TBATS and NN performs the best by MAPE.

Creating Jan 2011 forecast using averages of Neural Network & TBATS

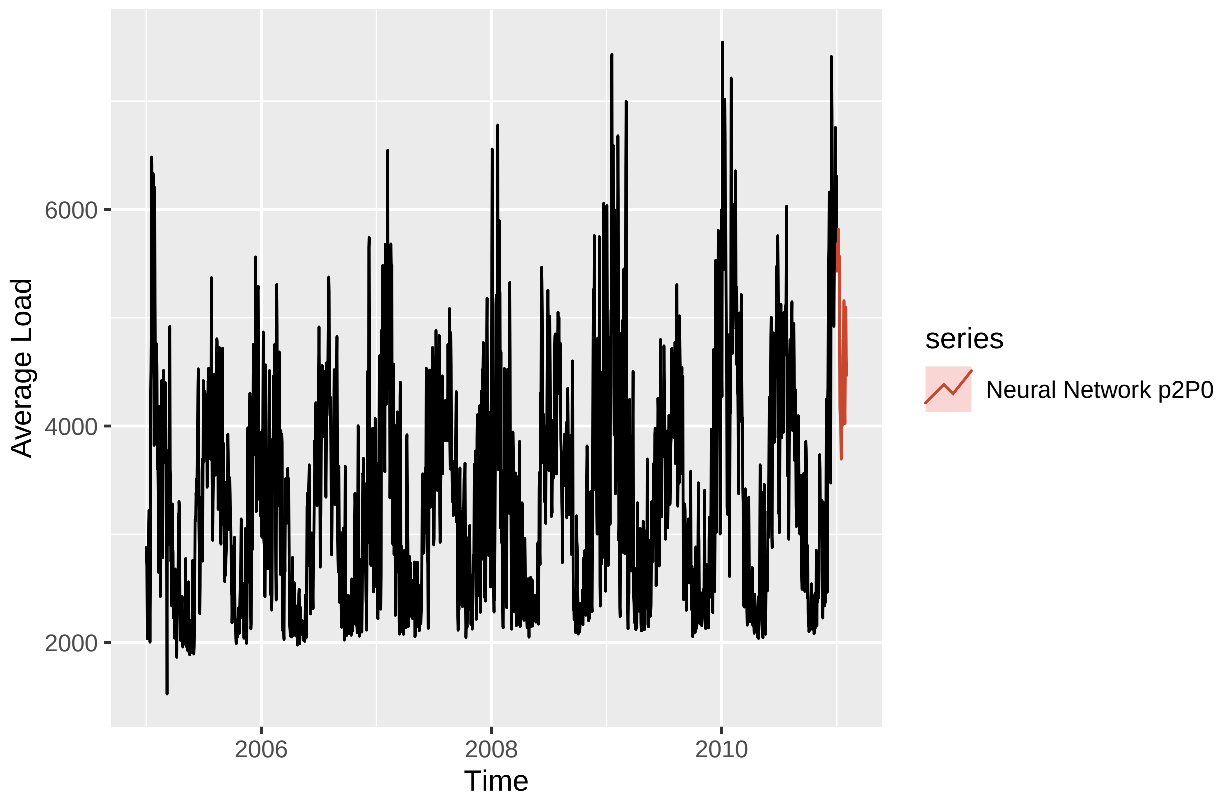
```r
#Create dates vector for forecast period
forecast_dates <- seq(as.Date("2011-01-01"), as.Date("2011-01-31"), by="days")


#Create df of average forecasted TBATS and NN load values
TBATS11_NN11_avg_for <- data.frame(cbind(forecast_dates,rowMeans(cbind(TBATS11_for$mean,NN11_for$mean))
colnames(TBATS11_NN11_avg_for) <- c("date","load")

#format dates
TBATS11_NN11_avg_for$date <- seq(as.Date("2011-01-01"), as.Date("2011-01-31"), by="days")

#write.csv(TBATS11_NN11_avg_for, file="./Outputs/TBATS_NN_avg_forecast.csv")
```

Creating Jan 2011 forecast using Neural Network (p=2, P=0)

```r
NN11_p2P0_fit <- nnetar(ts_DailyAvgLoad,p=1,P=0,xreg=fourier(ts_DailyAvgLoad, K=c(2,12)))

NN11_p2P0_for <- forecast::forecast(NN11_fit, h=31,xreg=fourier(ts_DailyAvgLoad,
                                            K=c(2,12),h=31))

#Plot foresting results
autoplot(NN11_p2P0_for) +
  ylab("Average Load")
```




```r
#Plot model + observed data
autoplot(ts_DailyAvgLoad) +
  autolayer(NN11_p2P0_for, series="Neural Network p2P0",PI=FALSE)+
  ylab("Average Load")
```




```
#write.csv(NN11_p2P0_for, file="./Outputs/NN11_p2P0_forecast.csv")
```

Creating Jan 2011 forecast using averages TBATS & Neural Network (p = 2, P = 0)

```
#Create df of average forecasted TBATS and NN load values
TBATS11_NN11_p2P0_avg_for <- data.frame(cbind(forecast_dates,rowMeans(cbind(TBATS11_for$mean,NN11_p2P0_
colnames(TBATS11_NN11_p2P0_avg_for) <- c("date","load")

#format dates
TBATS11_NN11_p2P0_avg_for$date <- seq(as.Date("2011-01-01"), as.Date("2011-01-31"), by="days")

#write.csv(TBATS11_NN11_p2P0_avg_for, file="./Outputs/TBATS_NNp2P0_avg_forecast.csv")
```