Forecasting Competition Final Report - McNeill/Turatkhan

GitHub Repository Link

Jenn McNeill and Zhanylai Turatkhan kyzy

04/01/2024

```
load <- read_excel("./Data_NoGIT/Raw/load.xlsx")
humidity <- read_excel("./Data_NoGIT/Raw/relative_humidity.xlsx")
temperature <- read_excel("./Data_NoGIT/Raw/temperature.xlsx")
head(data)
```

```
##
## 1 function (..., list = character(), package = NULL, lib.loc = NULL,
## 2     verbose = getOption("verbose"), envir = .GlobalEnv, overwrite = TRUE)
## 3 {
## 4     fileExt <- function(x) {
## 5         db <- grepl("\\\\.[^.]+\\\\.(gz|bz2|xz)$", x)
## 6         ans <- sub(".*\\\\.", "", x)
```

```
head(humidity)
```

```
## # A tibble: 6 x 30
##   date                   hr rh_ws1 rh_ws2 rh_ws3 rh_ws4 rh_ws5 rh_ws6 rh_ws7
##   <dttm>              <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 2005-01-01 00:00:00     1     99     93     93     90     87     93     93
## 2 2005-01-01 00:00:00     2     76     93     97     89     87     97     80
## 3 2005-01-01 00:00:00     3     79     93     93     90     93     97     83
## 4 2005-01-01 00:00:00     4     79     93     93     90     87     89     90
## 5 2005-01-01 00:00:00     5     79     93     96     93     87     90     93
## 6 2005-01-01 00:00:00     6     82     93     97     97     87     93     97
## # i 21 more variables: rh_ws8 <dbl>, rh_ws9 <dbl>, rh_ws10 <dbl>,
## #   rh_ws11 <dbl>, rh_ws12 <dbl>, rh_ws13 <dbl>, rh_ws14 <dbl>, rh_ws15 <dbl>,
## #   rh_ws16 <dbl>, rh_ws17 <dbl>, rh_ws18 <dbl>, rh_ws19 <dbl>, rh_ws20 <dbl>,
## #   rh_ws21 <dbl>, rh_ws22 <dbl>, rh_ws23 <dbl>, rh_ws24 <dbl>, rh_ws25 <dbl>,
## #   rh_ws26 <dbl>, rh_ws27 <dbl>, rh_ws28 <dbl>
```

```
head(temperature)
```

```
## # A tibble: 6 x 30
##   date                   hr t_ws1 t_ws2 t_ws3 t_ws4 t_ws5 t_ws6 t_ws7 t_ws8
##   <dttm>              <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 2005-01-01 00:00:00     1    43    46    40    47    48    46    44    52
## 2 2005-01-01 00:00:00     2    41    46    38    46    48    45    51    50
## 3 2005-01-01 00:00:00     3    40    46    37    45    45    45    49    48
## 4 2005-01-01 00:00:00     4    39    46    37    47    48    48    45    50
## 5 2005-01-01 00:00:00     5    38    46    37    44    48    49    43    50
## 6 2005-01-01 00:00:00     6    37    45    36    45    48    48    40    50
## # i 20 more variables: t_ws9 <dbl>, t_ws10 <dbl>, t_ws11 <dbl>, t_ws12 <dbl>,
## #   t_ws13 <dbl>, t_ws14 <dbl>, t_ws15 <dbl>, t_ws16 <dbl>, t_ws17 <dbl>,
## #   t_ws18 <dbl>, t_ws19 <dbl>, t_ws20 <dbl>, t_ws21 <dbl>, t_ws22 <dbl>,
## #   t_ws23 <dbl>, t_ws24 <dbl>, t_ws25 <dbl>, t_ws26 <dbl>, t_ws27 <dbl>,
## #   t_ws28 <dbl>
```

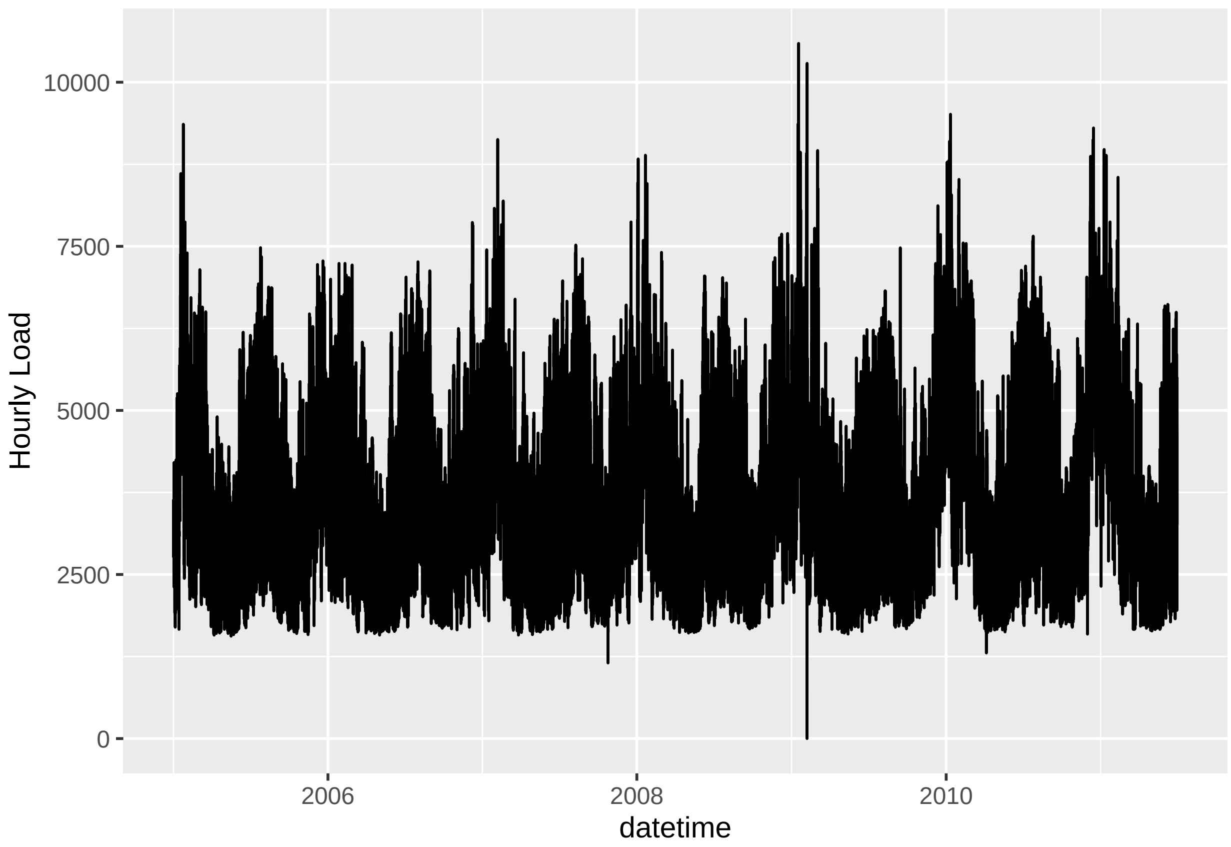
```
# create df with hourly values
hourly_data <- load %>%
  pivot_longer(h1:h24, names_to = "hour", values_to = "load") %>%
  mutate(hour = as.numeric(str_replace(hour, "h", ""))) %>%
  mutate(hour = hour - 1) %>%
  mutate(datetime = ymd_h(paste(date, hour, sep = " "))) %>%
  select(date, hour, datetime, load)

# create df with daily averages
daily_data <- hourly_data %>%
  filter(!is.na(load)) %>%
  group_by(date) %>%
  summarise(average_load = mean(load))

# check for NAs
summary(hourly_data$load)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0    2568    3352    3629    4520   10592       7
```

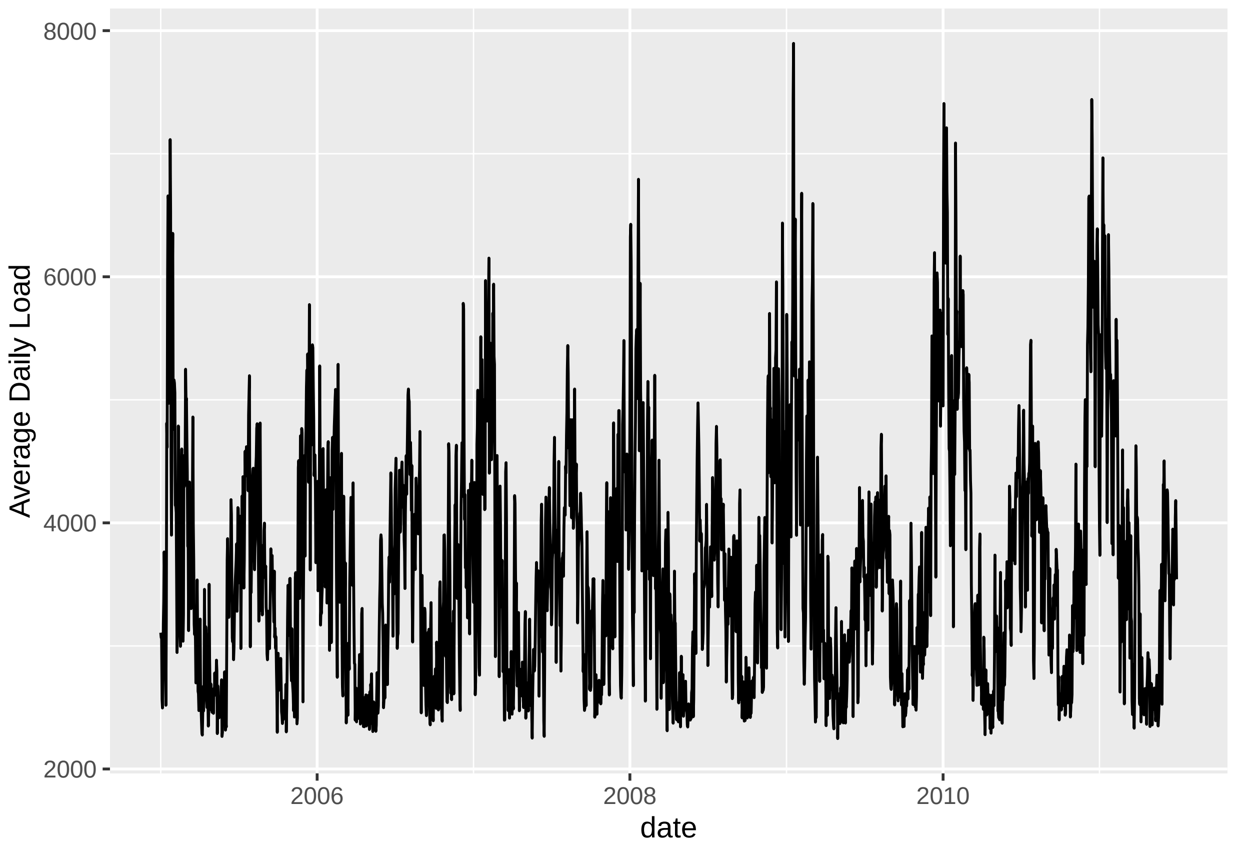
```
# there are 7 missing hourly values, so we will need to run tsclean if we are using hourly data to make

summary(daily_data$average_load)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2247    2798    3506    3629    4211    7897
```

```
# there are no NAs in the daily average data, so we can make a time series without running the tsclean

# plot the hourly values
ggplot(hourly_data, aes(x = datetime, y = load)) +
  geom_line() +
  ylab("Hourly Load")
```



```
# plot the daily averages
ggplot(daily_data, aes(x = date, y = average_load)) +
  geom_line() +
  ylab("Average Daily Load")
```

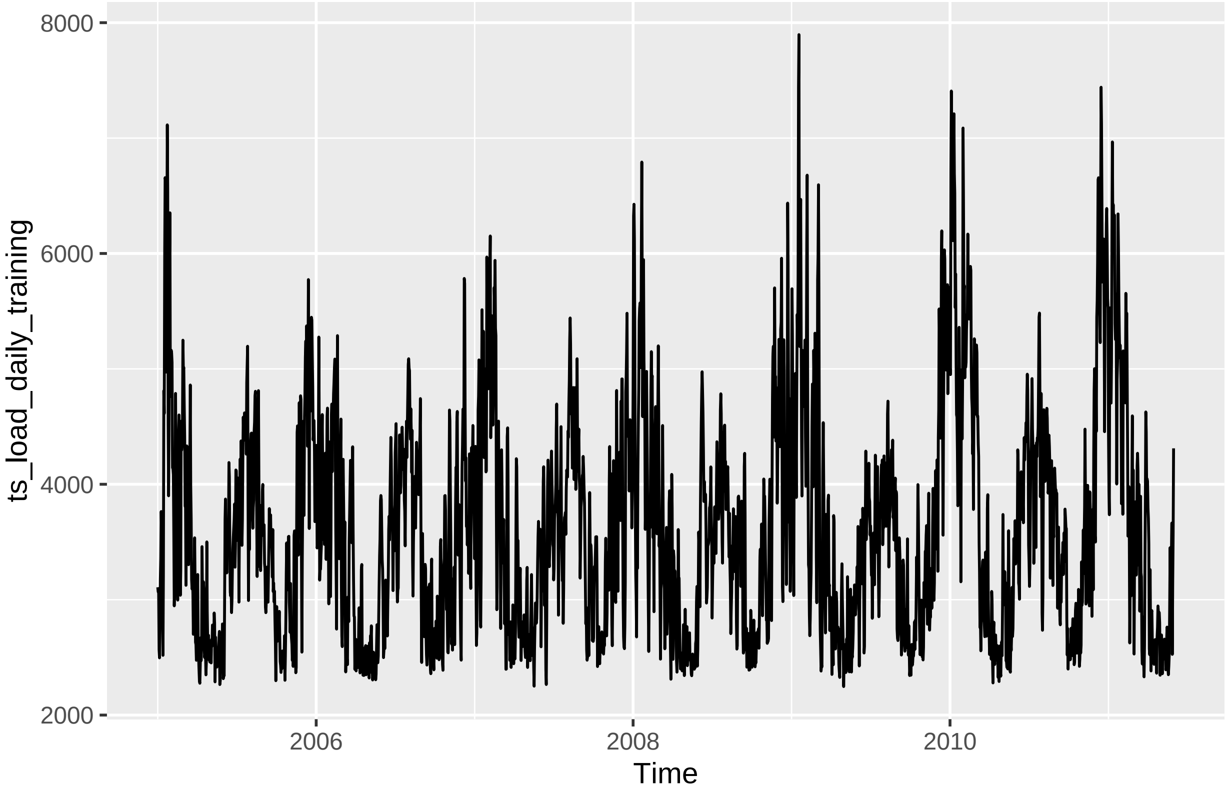


```r
# create df with daily temp averages
daily_temp <- temperature %>%
  pivot_longer(t_ws1:t_ws28, names_to = "site", values_to = "temperature") %>%
  group_by(date) %>%
  summarise(average_temp = mean(temperature)) %>%
  slice(1:2372)

# create df with daily relative humidity averages
daily_humidity <- humidity %>%
  pivot_longer(rh_ws1:rh_ws28, names_to = "site", values_to = "humidity") %>%
  group_by(date) %>%
  summarise(average_humidity = mean(humidity))
```

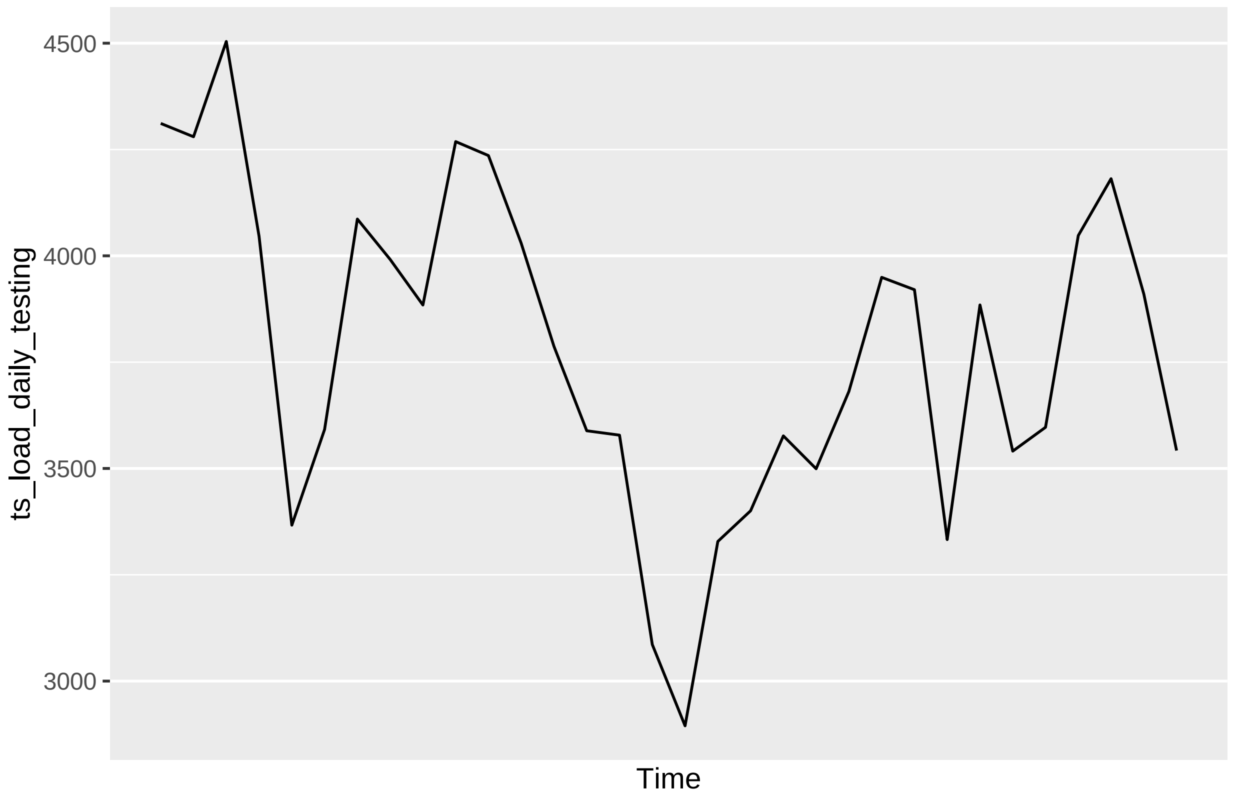
```r
# create a subset of the time series that excludes one month
n_for = 31
ts_load_daily_training <- subset(ts_load_daily, end = length(ts_load_daily) - n_for)

# create a subset of the time series that only includes the last month
ts_load_daily_testing <- subset(ts_load_daily, start = length(ts_load_daily) - n_for)

# repeat the process for temperature regressor
ts_temp_daily_training <- subset(ts_temp_daily, end = length(ts_temp_daily) - n_for)
ts_temp_daily_testing <- subset(ts_temp_daily, start = length(ts_temp_daily) - n_for)

# repeat the process for humidity regressor
ts_humidity_daily_training <- subset(ts_humidity_daily, end = length(ts_humidity_daily) - n_for)
```

```
ts_humidity_daily_testing <- subset(ts_humidity_daily, start = length(ts_humidity_daily) - n_for)
```

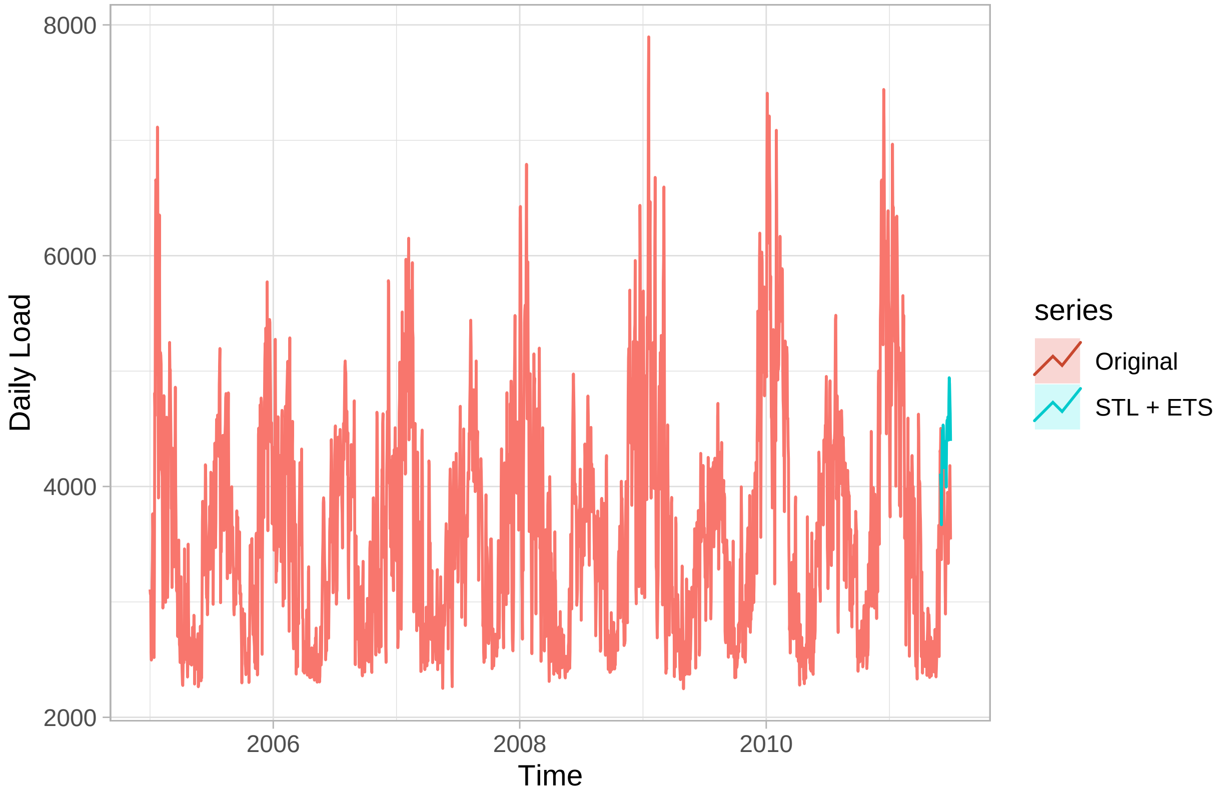
```
autoplot(ts_load_daily_training)
```



```
autoplot(ts_load_daily_testing)
```



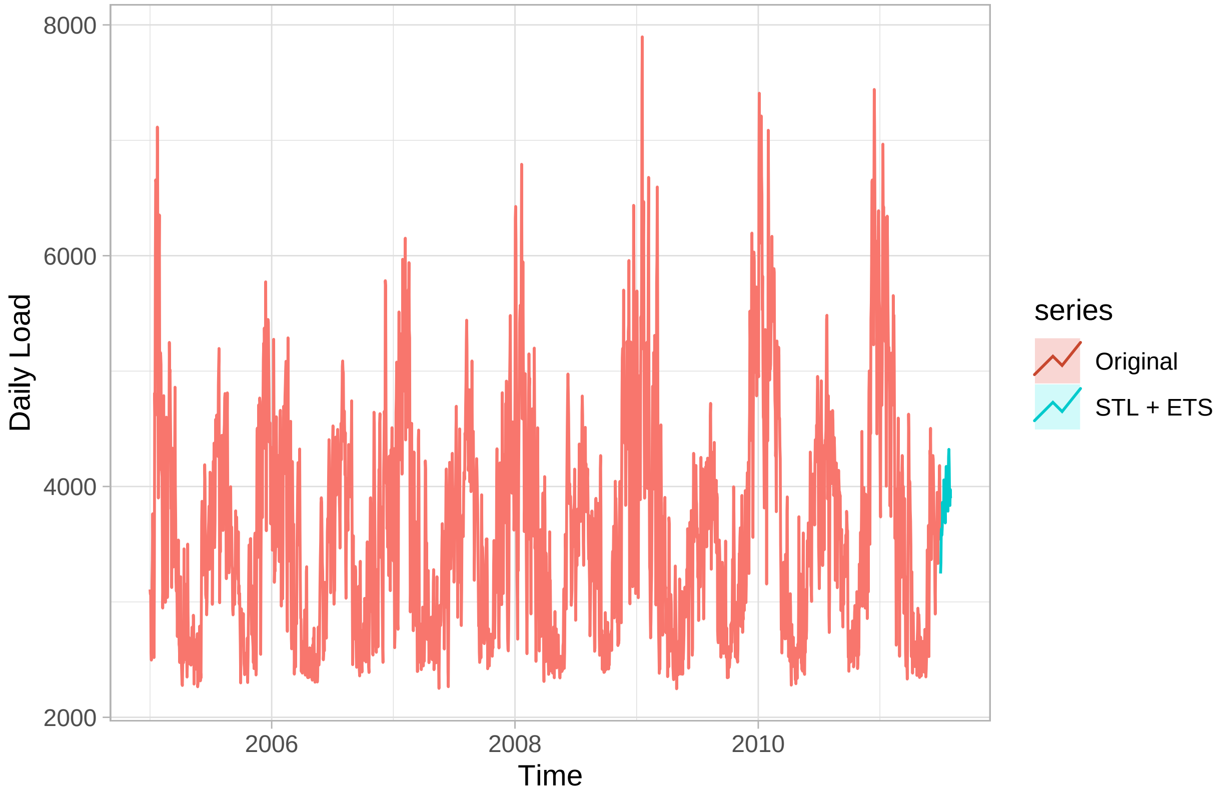
```
# fit and forecast STL + ETS model to data
STL_ETS_test <-  stlf(ts_load_daily_training, h = 31)

# plot model + observed data
autoplot(ts_load_daily, series = "Original") +
  autolayer(STL_ETS_test, series = "STL + ETS", PI = FALSE) +
  ylab("Daily Load") +
  theme_light()
```



```
# check the MAPE
STL_ETS_scores <- accuracy(STL_ETS_test$mean, ts_load_daily_testing)
print(STL_ETS_scores)
```

```
##                   ME     RMSE      MAE       MPE    MAPE      ACF1 Theil's U
## Test set -619.2312 754.1782 667.3514 -17.50235 18.5994 0.6472146  2.450204
```

```
#use this model on the whole dataset to predict july 2011
STL_ETS_forecast <- stlf(ts_load_daily, h = 31)

autoplot(ts_load_daily, series = "Original") +
  autolayer(STL_ETS_forecast, series = "STL + ETS", PI = FALSE) +
  ylab("Daily Load") +
  theme_light()
```

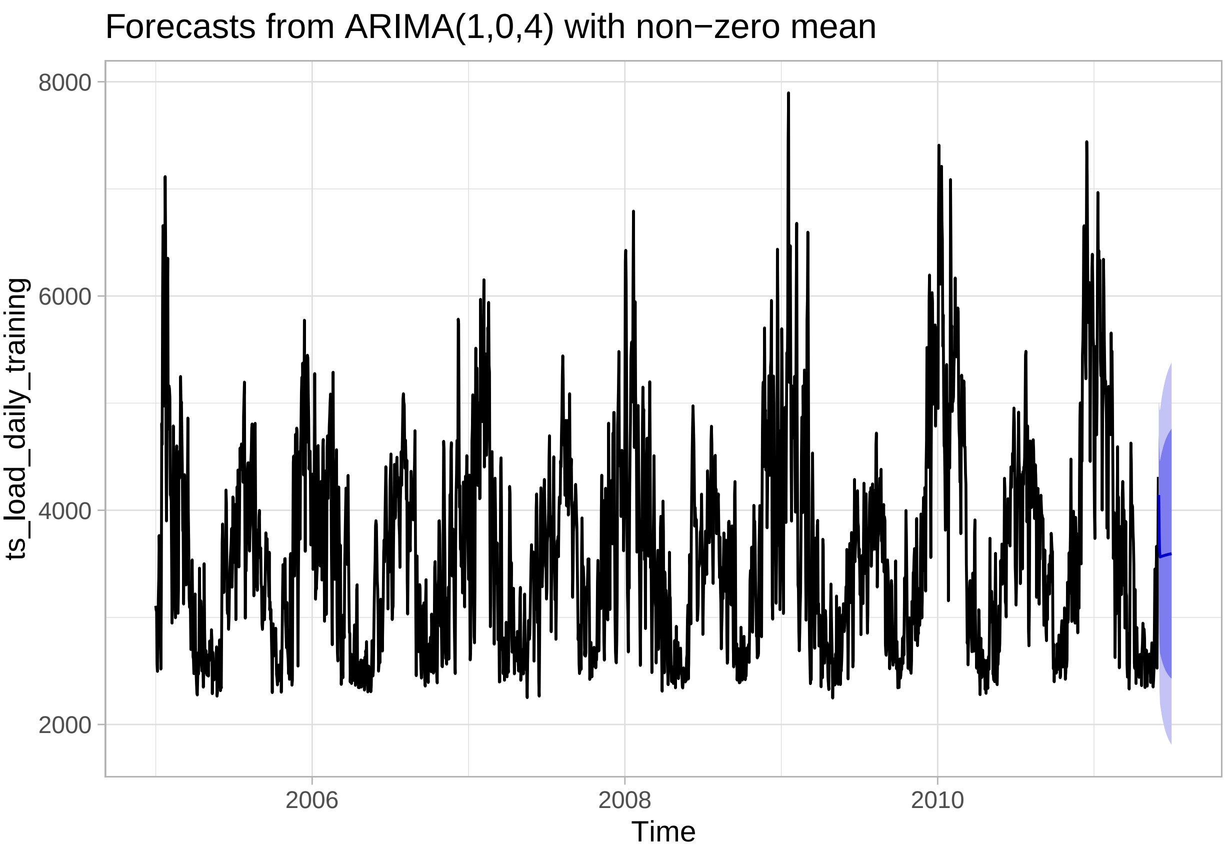


```r
STL_ETS_forecast_submission <- STL_ETS_forecast$mean

getwd()
```

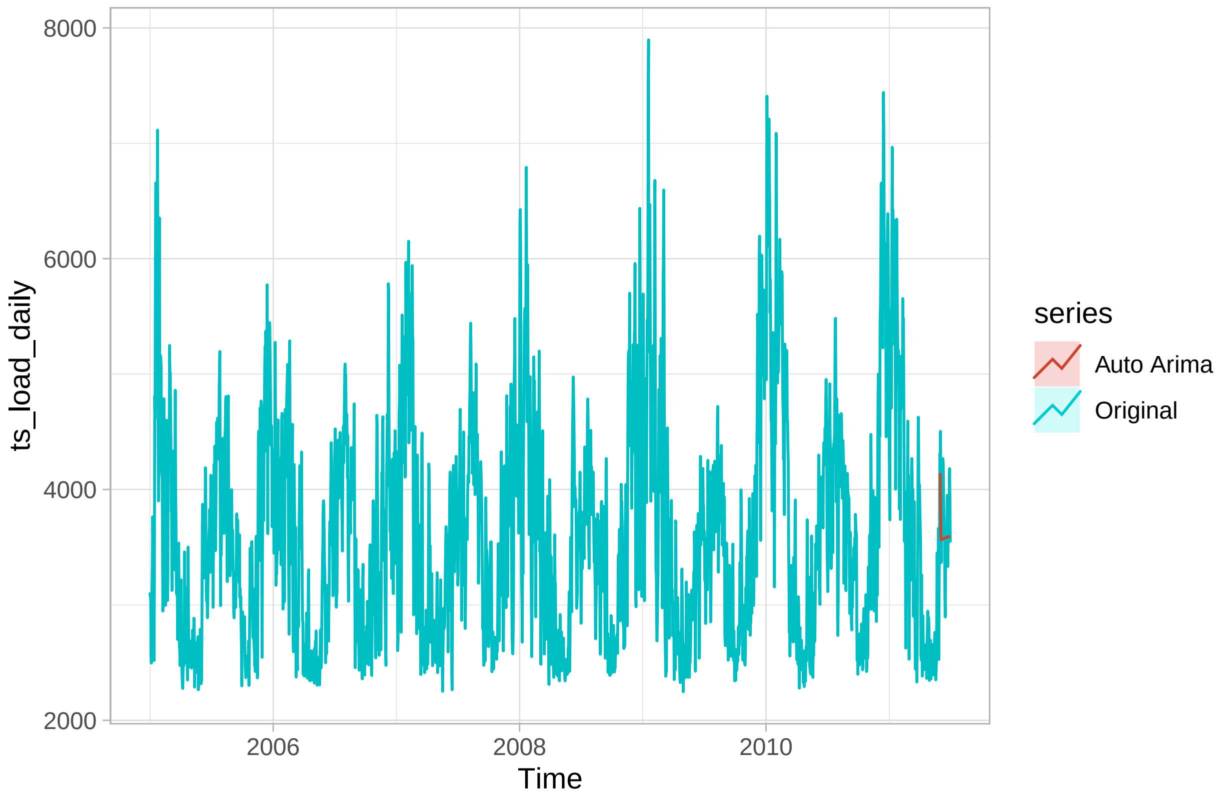
```
## [1] "/home/guest/TSA_Janka/Upd_McNeillTuratkhan_ENV797_TSA_Competition"
```

```r
submission_template <- read.csv(file="./submission_template.csv", header=TRUE)
submission_template$date <- as.Date(submission_template$date, format = "%m/%d/%y")
submission_template$load <- STL_ETS_forecast_submission
write.table(submission_template, "submission.csv", sep = ",", row.names = FALSE, quote = FALSE)
```

```r
auto_arima_train <- auto.arima(ts_load_daily_training, seasonal=FALSE)

auto_arima_test <- forecast(auto_arima_train, h=31)

#plot foresting results
autoplot(auto_arima_test) +
  theme_light()
```



```r
#plot model + observed data
autoplot(ts_load_daily, series = "Original") +
  autolayer(auto_arima_test, series = "Auto Arima", PI = FALSE) +
  theme_light()
```



```
#check the MAPE
auto_arima_scores <- accuracy(auto_arima_test$mean, ts_load_daily_testing)
print(auto_arima_scores)
```

```
##                 ME     RMSE      MAE      MPE     MAPE      ACF1 Theil's U
## Test set 154.6606 374.7045 300.4123 3.252613 7.872301 0.5483927  1.143661
```

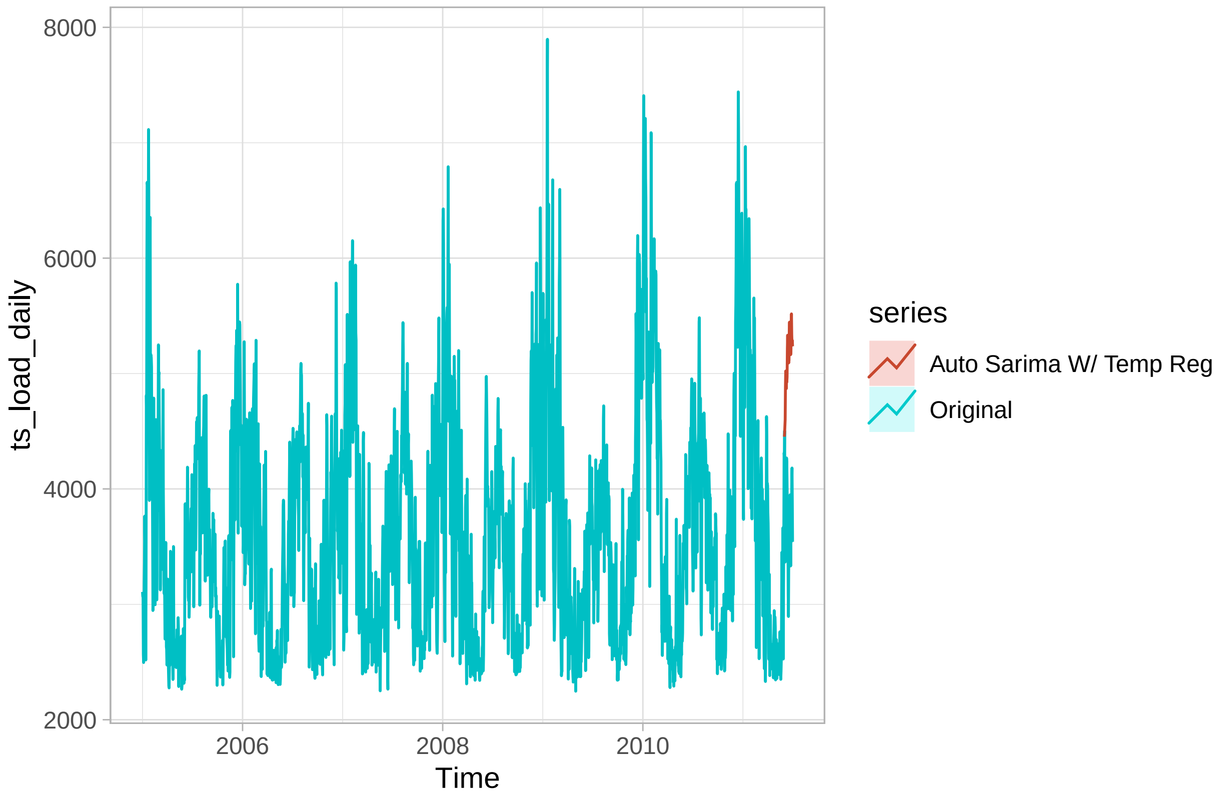
```
#use this model on the whole dataset to predict july 2011
auto_arima <- auto.arima(ts_load_daily, seasonal=FALSE)
auto_arima_forecast <- forecast(auto_arima, h=31)

auto_arima_forecast_submission <- auto_arima_forecast$mean

getwd()
```

```
## [1] "/home/guest/TSA_Janka/Upd_McNeillTuratkhan_ENV797_TSA_Competition"
```

```
submission_template <- read.csv(file="./submission_template.csv", header=TRUE)
submission_template$date <- as.Date(submission_template$date, format = "%m/%d/%y")
submission_template$load <- auto_arima_forecast_submission
write.table(submission_template, "submission.csv", sep = ",", row.names = FALSE, quote = FALSE)
```

```
auto_with_temp_reg_train <- auto.arima(ts_load_daily_training, seasonal=FALSE,
                            lambda=0, xreg=fourier(ts_temp_daily_training, K=c(2,12)))

auto_with_temp_reg_test <- forecast(auto_with_temp_reg_train,
                                    xreg=fourier(ts_temp_daily_training, K=c(2,12), h=31),
                                    h=31)
```

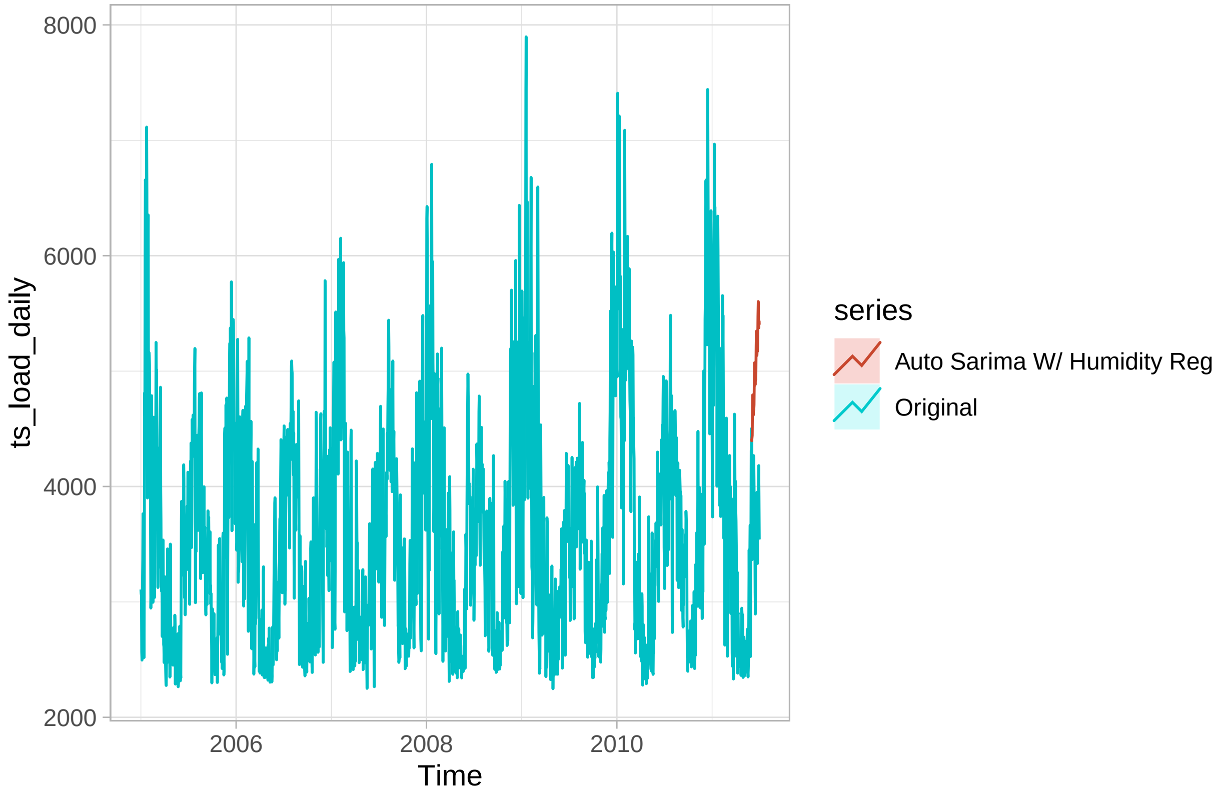
```
#plot model + observed data
autoplot(ts_load_daily, series = "Original") +
  autolayer(auto_with_temp_reg_test, series = "Auto Sarima W/ Temp Reg", PI = FALSE) +
  theme_light()
```



```
#check the MAPE
auto_with_temp_reg_scores <- accuracy(auto_with_temp_reg_test$mean, ts_load_daily_testing)
print(auto_with_temp_reg_scores)
```

```
##                   ME     RMSE     MAE       MPE     MAPE      ACF1 Theil's U
## Test set -1322.395 1427.407 1323.86 -36.75873 36.79128 0.7057242  4.639653
```

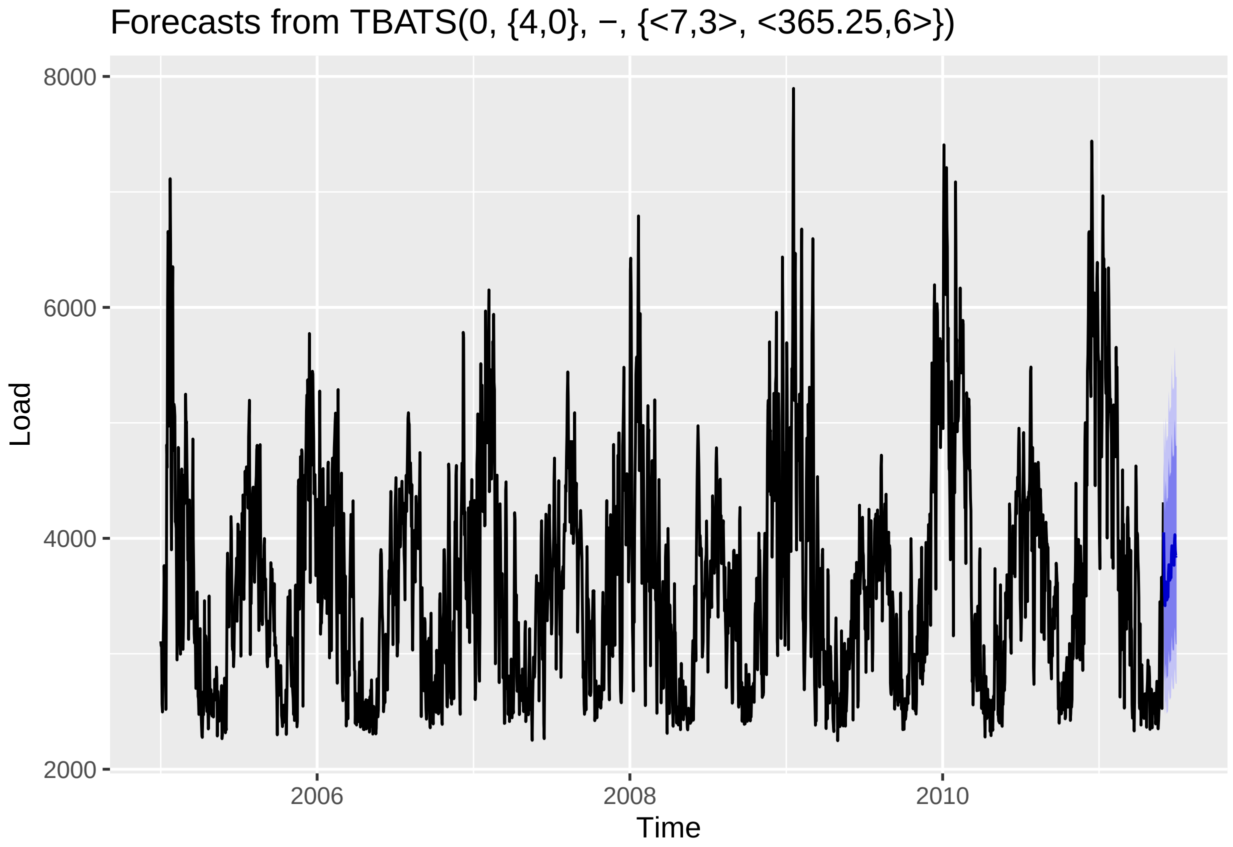
```
auto_with_humidity_reg_train <- auto.arima(ts_load_daily_training, seasonal=FALSE,
                            lambda=0, xreg=fourier(ts_humidity_daily_training, K=c(2,4)))

auto_with_humidity_reg_test <- forecast(auto_with_humidity_reg_train,
                                        xreg=fourier(ts_humidity_daily_training, K=c(2,4), h=31),
                                        h=31)

#plot model + observed data
autoplot(ts_load_daily, series = "Original") +
  autolayer(auto_with_humidity_reg_test, series = "Auto Sarima W/ Humidity Reg", PI = FALSE) +
  theme_light()
```



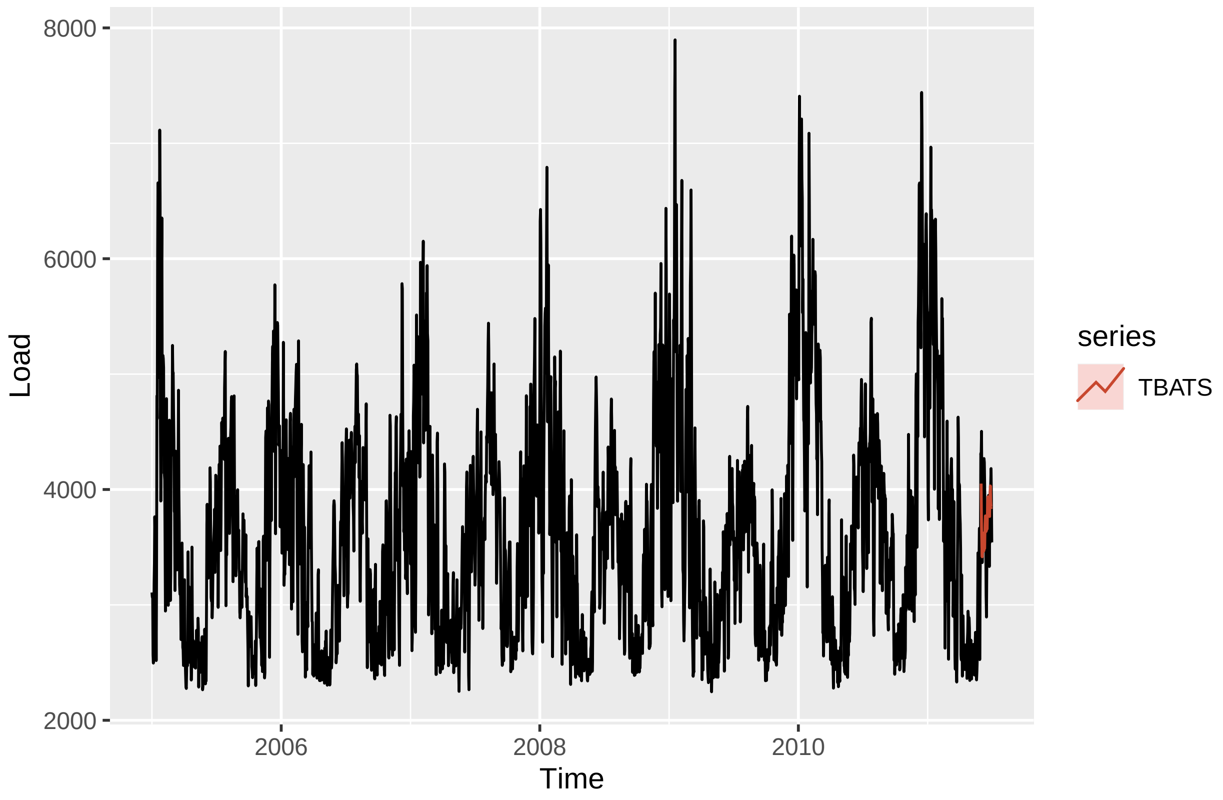
```r
#check the MAPE
auto_with_humidity_reg_scores <- accuracy(auto_with_humidity_reg_test$mean, ts_load_daily_testing)
print(auto_with_humidity_reg_scores)
```

```
##                  ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -1218.589 1346.502 1225.534 -33.94039 34.09459 0.7364895  4.366533
```

```r
TBATS_fit <- tbats(ts_load_daily_training)

TBATS_for <- forecast(TBATS_fit, h=31)

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



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



```
TBATS_scores <- accuracy(TBATS_for$mean, ts_load_daily_testing)
print(TBATS_scores)
```

```
##                ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set 45.70463 417.4762 349.6332 0.2120964 9.342013 0.6403287   1.27502
```

```
#retraining for the whole dataset to predict july 2011
TBATS_kaggle_fit <- tbats(ts_load_daily)
TBATS_kaggle_for <- forecast(TBATS_kaggle_fit, h=31)

TBATS_forecast_submission <- TBATS_kaggle_for$mean

# the best score as of now
getwd()
```

```
## [1] "/home/guest/TSA_Janka/Upd_McNeillTuratkhan_ENV797_TSA_Competition"
```

```
submission_template <- read.csv(file="./submission_template.csv", header=TRUE)
submission_template$date <- as.Date(submission_template$date, format = "%m/%d/%y")
submission_template$load <- TBATS_forecast_submission
write.table(submission_template, "submission.csv", sep = ",", row.names = FALSE, quote = FALSE)
```