Competition Report

AyoungLeines

2025-04-23

Github repository

https://github.com/nicoleleines95/AyoungLeines_ENV797_TSA_ForecastCompetition_S25

Packages

```
library(tidyverse)
library(readxl)
library(ggplot2)
library(forecast)
library(Kendall)
library(tseries)
library(outliers)
library(smooth)
library(zoo)
library(kableExtra)
library(tsibble)
library(tibble)
library(forecastHybrid)
library(purrr)
library(tictoc)
library(janitor)
```

Directory

```
base_dir <- "D:/Geani/Box/Home Folder gnl13/Private/1 Academics/3 Time series/AyoungLeines_ENV797_TSA_F
data_dir <- file.path(base_dir, "Data")
output_dir <- file.path(base_dir, "Forecast")

file1 <- "load.xlsx"
file2 <- "temperature.xlsx"
file3 <- "relative_humidity.xlsx"
file4 <- "submission_template.xlsx"

file_path1 <- file.path(data_dir, file1)
file_path2 <- file.path(data_dir, file2)
file_path3 <- file.path(data_dir, file3)
```

```r
file_path4 <- file.path(data_dir, file4)

load_raw <- read_excel(file_path1) %>% clean_names()
temp_raw <- read_excel(file_path2) %>% clean_names()
hum_raw <- read_excel(file_path3) %>% clean_names()
template <- read_excel(file_path4)
```

Wrangling data - Aggregate the hourly data to daily using averages

```r
#Demand

# from wide (h1-h24) to long format, convert hour to integer
load_long <- load_raw %>%
  pivot_longer(
    cols        = starts_with("h"),
    names_to    = "hour",
    names_prefix= "h",
    values_to   = "load_kwh"
  ) %>%
  mutate(
    date     = as_date(date),
    hour     = as.integer(hour),
    meter_id = factor(meter_id)
  )

# compute per-meter daily mean
daily_load <- load_long %>%
  group_by(meter_id, date) %>%
  summarise(
    daily_load_kwh = mean(load_kwh, na.rm = TRUE),
    .groups = "drop"
  )
```

```r
# Temperature

# daily mean across all hours and stations
daily_temp <- temp_raw %>%
  pivot_longer(
    cols         = starts_with("t_ws"),
    names_to     = "station",
    names_prefix = "t_ws",
    values_to    = "temp_c"
  ) %>%
  mutate(date = as_date(date)) %>%
  group_by(date) %>%
  summarise(
    mean_temp_c = mean(temp_c, na.rm = TRUE),
    .groups     = "drop"
  )

# Relative humidity
```

```r
daily_hum <- hum_raw %>%
  pivot_longer(
    cols         = starts_with("rh_ws"),
    names_to     = "station",
    names_prefix = "rh_ws",
    values_to    = "rh_pct"
  ) %>%
  mutate(date = as_date(date)) %>%
  group_by(date) %>%
  summarise(
    mean_rh_pct = mean(rh_pct, na.rm = TRUE),
    .groups     = "drop"
  )
```

```r
daily_data <- daily_load %>%
  inner_join(daily_temp, by = "date") %>%
  inner_join(daily_hum,  by = "date")

# total system demand per day, plus averaged covariates
agg_daily <- daily_data %>%
  group_by(date) %>%
  summarise(
    demand_kwh = sum(daily_load_kwh, na.rm = TRUE),
    temp_c     = mean(mean_temp_c,  na.rm = TRUE),
    rh_pct     = mean(mean_rh_pct,  na.rm = TRUE),
    .groups    = "drop"
  )
```

```r
agg_daily %>%
  slice_head(n = 6) %>%
  kable(
    caption = "First six days of aggregated daily demand, temperature, and humidity",
    digits  = 2
  ) %>%
  kable_styling(full_width = FALSE)
```

Table 1: First six days of aggregated daily demand, temperature, and humidity

date	demand_kwh	temp_c	rh_pct
2005-01-01	2889.12	53.57	76.71
2005-01-02	2788.96	53.76	80.48
2005-01-03	2708.46	55.91	81.23
2005-01-04	2211.58	61.69	74.84
2005-01-05	2035.12	60.43	76.07
2005-01-06	2109.62	62.00	77.99

Define training and testing sets

```r
train <- agg_daily %>%
  filter(date >= ymd("2005-01-01") & date <= ymd("2009-12-31"))

test <- agg_daily %>%
  filter(date >= ymd("2010-01-01") & date <= ymd("2010-02-28"))

full_train <- agg_daily %>%
  filter(date >= ymd("2005-01-01") & date <= ymd("2010-12-31"))

# 5.2 Construct msts objects with weekly and yearly seasonality
y_train <- msts(
  train$demand_kwh,
  seasonal.periods = c(7, 365.25),
  start             = c(2005, 1)
)

y_test <- msts(
  test$demand_kwh,
  seasonal.periods = c(7, 365.25),
  start             = c(2010, 1)
)

y_full <-  msts(
  full_train$demand_kwh,
  seasonal.periods = c(7, 365.25),
  start             = c(2005, 1)
)
```

Forecasting until 2010

```r
# 1) Forecast horizon
h <- length(y_test)

# 2) Fit each model

# ARIMA + Fourier + weather regressors
f_tr          <- fourier(y_train, K = c(2,6))
f_ts          <- fourier(y_train, K = c(2,6), h = h)
fit_reg       <- auto.arima(
                   y_train,
                   seasonal = FALSE,
                   xreg     = cbind(f_tr, train$temp_c, train$rh_pct)
                 )
fc_reg        <- forecast(
                   fit_reg,
                   h    = h,
                   xreg = cbind(f_ts, test$temp_c, test$rh_pct)
                 )


# ETS with Box-Cox (lambda chosen to stabilize MAPE)
```

```r
fc_ets_bc     <- forecast(ets(y_train, lambda="auto"), h = h)

fc_naive      <- naive(y_train, h = h)
fc_tbats      <- forecast(tbats(y_train),   h = h)

# TBATS with weather covariates
xreg_temp      <- matrix(train$temp_c,  ncol = 1)
xreg_test_temp <- matrix(test$temp_c,   ncol = 1)
xreg_hum       <- matrix(train$rh_pct,  ncol = 1)
xreg_test_hum  <- matrix(test$rh_pct,   ncol = 1)
xreg_both      <- cbind(train$temp_c,  train$rh_pct)
xreg_test_both <- cbind(test$temp_c,   test$rh_pct)

fc_tbats_temp  <- forecast(tbats(y_train, xreg = xreg_temp),  h = h, xreg = xreg_test_temp)
fc_tbats_hum   <- forecast(tbats(y_train, xreg = xreg_hum),   h = h, xreg = xreg_test_hum)
fc_tbats_both  <- forecast(tbats(y_train, xreg = xreg_both),  h = h, xreg = xreg_test_both)
```

```r
# 3) Extract accuracy metrics
models <- list(
    `ETS-BoxCox`   = fc_ets_bc,
    Naive          = fc_naive,
    TBATS          = fc_tbats,
  `TBATS+Temp`   = fc_tbats_temp,
  `TBATS+Hum`    = fc_tbats_hum
)

accuracy_tbl <- purrr::map_df(models, function(fit) {
  acc <- accuracy(fit, y_test)
  # if accuracy() returned two rows (train & test), use the test row
  idx <- if (nrow(acc) == 2) 2 else 1
  # pull out the six metrics we want
  acc[idx, c("ME","RMSE","MAE","MAPE","MASE","ACF1"), drop=FALSE] %>%
    as_tibble(rownames = "dummy") %>%
    select(-dummy)
}, .id = "Model")

# 4) Render comparison table, sorted by MAPE
accuracy_tbl %>%
  arrange(MAPE) %>%
  kable(
    caption = "Forecast accuracy comparison (all models)",
    digits  = 2
  ) %>%
  kable_styling(full_width = FALSE) %>%
  row_spec(1, bold = TRUE, background = "#F0F0F0")  # highlight best MAPE
```

Table 2: Forecast accuracy comparison (all models)

Model	ME	RMSE	MAE	MAPE	MASE	ACF1
ETS-BoxCox	**485.94**	**1239.18**	**984.90**	**19.28**	**1.28**	**0.8**
Naive	486.05	1239.22	984.94	19.28	1.28	0.8
TBATS	438.30	1244.69	1008.28	20.35	1.31	0.8

TBATS+Temp	438.30	1244.69	1008.28	20.35	1.31	0.8
TBATS+Hum	438.30	1244.69	1008.28	20.35	1.31	0.8

Forecasting until 2011- top 5 models - with all information

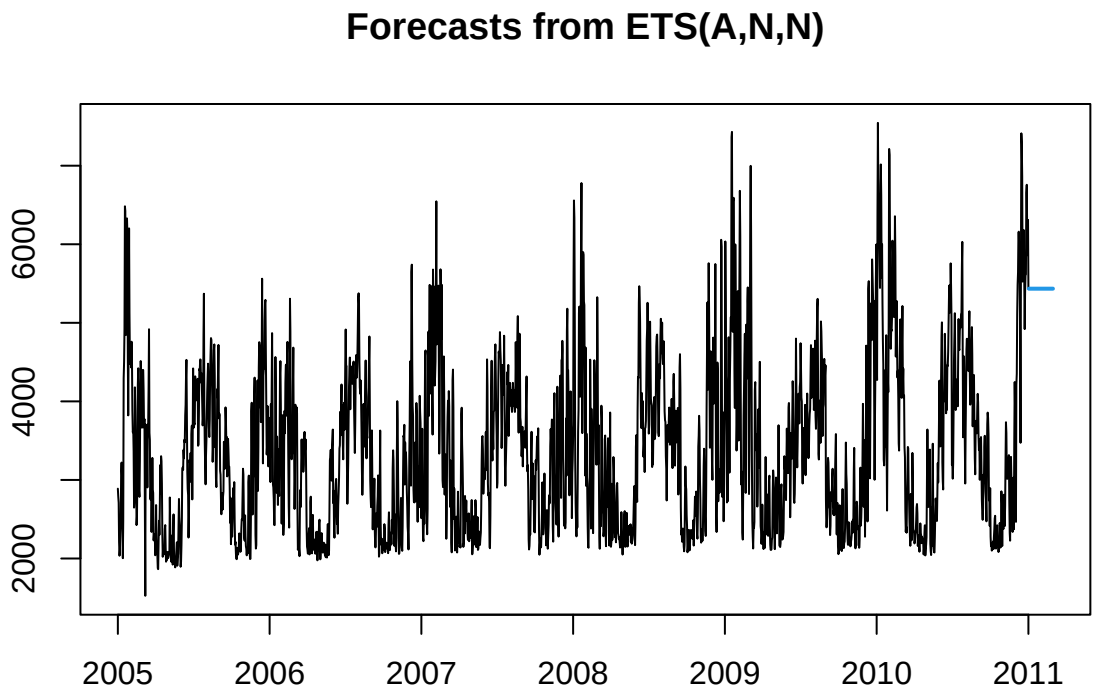
```
forecast_dates <- seq(as.Date("2011-01-01"), as.Date("2011-02-28"), by = "day")
h_future       <- length(forecast_dates)  # 59
```

```
#Model 1: # ETS with Box-Cox (lambda chosen to stabilize MAPE)
fc_ets_bc_full     <- forecast(ets(y_full, lambda="auto"), h = h_future)

print(fc_ets_bc_full)
```

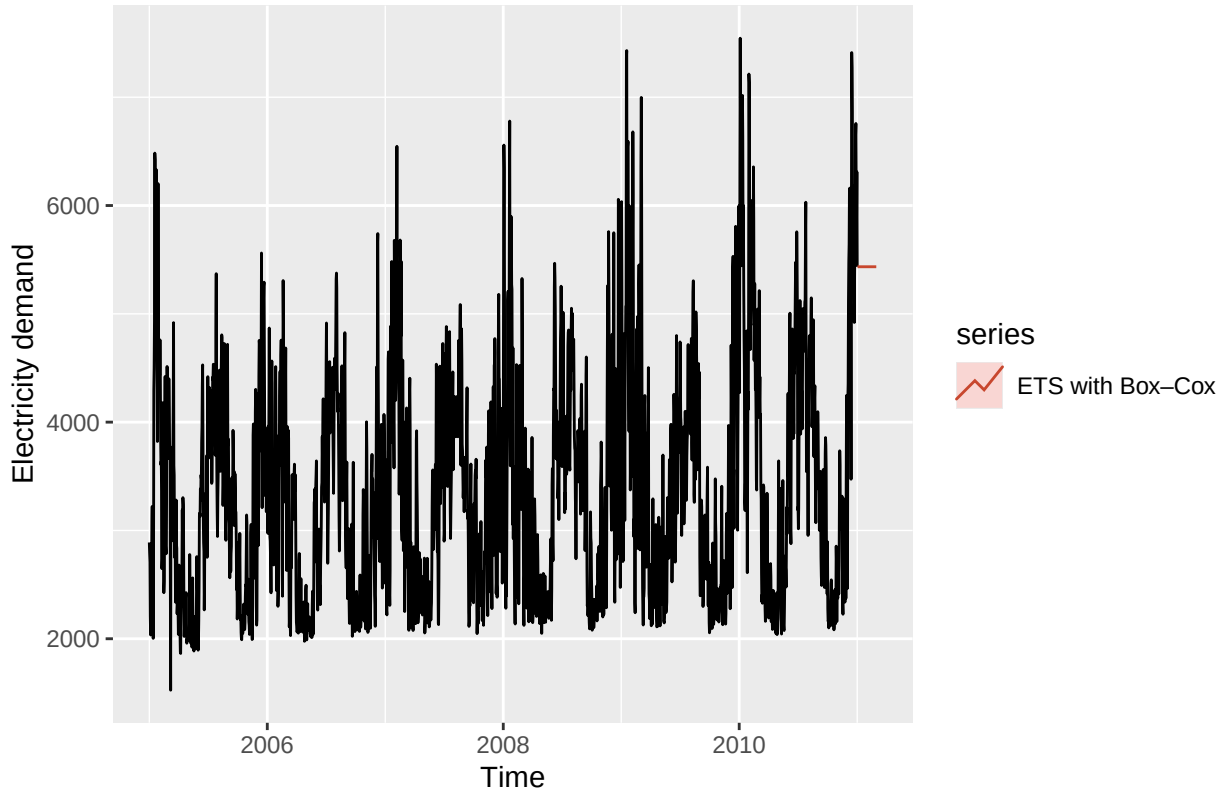
```
##           Point Forecast    Lo 80      Hi 80    Lo 95      Hi 95
## 2011.0027       5434.701 4095.479   7951.106 3614.502   10422.48
## 2011.0055       5434.701 3709.794   9768.706 3164.111   16263.85
## 2011.0082       5434.701 3457.804  11797.605 2885.135   27561.24
## 2011.0110       5434.701 3269.399  14245.249 2683.988   61189.65
## 2011.0137       5434.701 3118.904  17357.289 2527.699 1086869.88
## 2011.0164       5434.701 2993.767  21521.223 2400.613         NA
## 2011.0192       5434.701 2886.840  27437.657 2294.034         NA
## 2011.0219       5434.701 2793.647  36546.352 2202.621         NA
## 2011.0247       5434.701 2711.184  52358.137 2122.862         NA
## 2011.0274       5434.701 2637.341  86170.280 2052.322         NA
## 2011.0301       5434.701 2570.571 203285.559 1989.248         NA
## 2011.0329       5434.701 2509.710         NA 1932.334         NA
## 2011.0356       5434.701 2453.857         NA 1880.582         NA
## 2011.0384       5434.701 2402.301         NA 1833.215         NA
## 2011.0411       5434.701 2354.472         NA 1789.614         NA
## 2011.0438       5434.701 2309.904         NA 1749.280         NA
## 2011.0466       5434.701 2268.213         NA 1711.805         NA
## 2011.0493       5434.701 2229.078         NA 1676.849         NA
## 2011.0521       5434.701 2192.229         NA 1644.130         NA
## 2011.0548       5434.701 2157.433         NA 1613.408         NA
## 2011.0575       5434.701 2124.494         NA 1584.478         NA
## 2011.0603       5434.701 2093.241         NA 1557.165         NA
## 2011.0630       5434.701 2063.523         NA 1531.318         NA
## 2011.0658       5434.701 2035.211         NA 1506.805         NA
## 2011.0685       5434.701 2008.189         NA 1483.510         NA
## 2011.0712       5434.701 1982.357         NA 1461.331         NA
## 2011.0740       5434.701 1957.624         NA 1440.179         NA
## 2011.0767       5434.701 1933.908         NA 1419.974         NA
## 2011.0795       5434.701 1911.138         NA 1400.645         NA
## 2011.0822       5434.701 1889.249         NA 1382.127         NA
## 2011.0849       5434.701 1868.181         NA 1364.364         NA
## 2011.0877       5434.701 1847.883         NA 1347.304         NA
## 2011.0904       5434.701 1828.304         NA 1330.900         NA
## 2011.0932       5434.701 1809.401         NA 1315.110         NA
## 2011.0959       5434.701 1791.135         NA 1299.894         NA
## 2011.0986       5434.701 1773.467         NA 1285.219         NA
## 2011.1014       5434.701 1756.365         NA 1271.051         NA
```

```
## 2011.1041       5434.701 1739.796         NA 1257.362         NA
## 2011.1068       5434.701 1723.733         NA 1244.123         NA
## 2011.1096       5434.701 1708.149         NA 1231.311         NA
## 2011.1123       5434.701 1693.019         NA 1218.902         NA
## 2011.1151       5434.701 1678.320         NA 1206.874         NA
## 2011.1178       5434.701 1664.032         NA 1195.208         NA
## 2011.1205       5434.701 1650.135         NA 1183.886         NA
## 2011.1233       5434.701 1636.609         NA 1172.890         NA
## 2011.1260       5434.701 1623.439         NA 1162.205         NA
## 2011.1288       5434.701 1610.608         NA 1151.817         NA
## 2011.1315       5434.701 1598.101         NA 1141.710         NA
## 2011.1342       5434.701 1585.904         NA 1131.873         NA
## 2011.1370       5434.701 1574.004         NA 1122.293         NA
## 2011.1397       5434.701 1562.388         NA 1112.959         NA
## 2011.1425       5434.701 1551.045         NA 1103.860         NA
## 2011.1452       5434.701 1539.964         NA 1094.987         NA
## 2011.1479       5434.701 1529.134         NA 1086.329         NA
## 2011.1507       5434.701 1518.546         NA 1077.879         NA
## 2011.1534       5434.701 1508.190         NA 1069.628         NA
## 2011.1562       5434.701 1498.058         NA 1061.567         NA
## 2011.1589       5434.701 1488.141         NA 1053.690         NA
## 2011.1616       5434.701 1478.432         NA 1045.990         NA
```

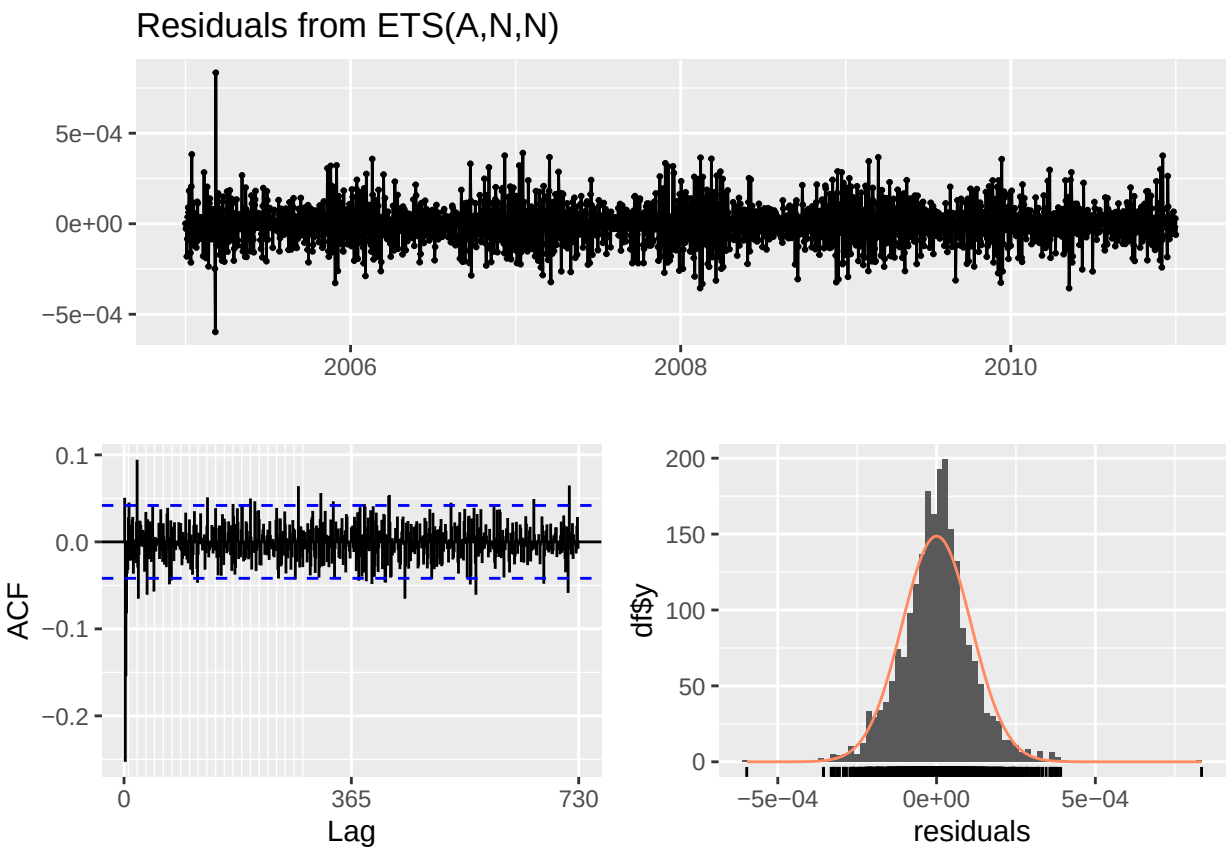
```
plot(fc_ets_bc_full)
```



```
#Plot model + observed data
autoplot(y_full) +
  autolayer(fc_ets_bc_full, series="ETS with Box-Cox",PI=FALSE) +
  ylab("Electricity demand")
```




```
checkresiduals(fc_ets_bc_full)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,N,N)
## Q* = 790.65, df = 438, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 438
```

```
# Export to CSV
submission <- template %>%
  mutate(
    date = format(forecast_dates, "%Y-%m-%d"),
    load = as.numeric(fc_ets_bc_full$mean)
  )
out_name <- "submission_final_ETS_with_Box-Cox.csv"
readr::write_csv(submission, file.path(output_dir, out_name))
```

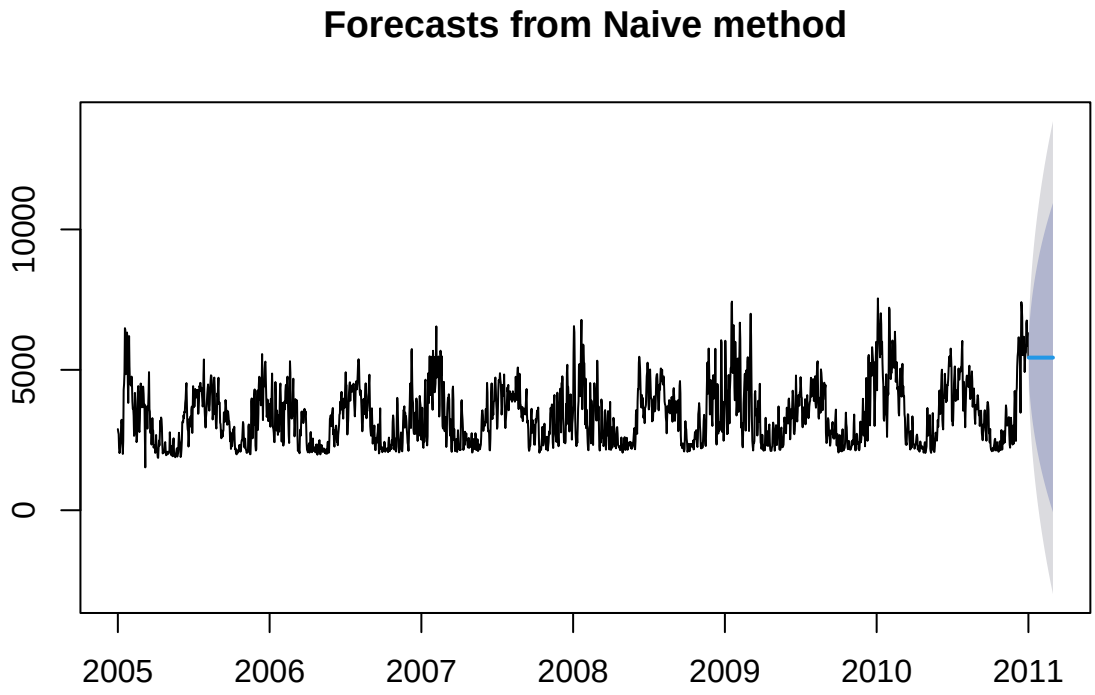
```
#Model 2: Naive

fc_naive_full <- naive(y_full, h = h_future)
print(fc_naive_full)
```

```
##           Point Forecast      Lo 80     Hi 80       Lo 95     Hi 95
## 2011.0027       5434.625 4717.82576  6151.424  4338.37516  6530.875
## 2011.0055       5434.625 4420.91779  6448.332  3884.29361  6984.956
```

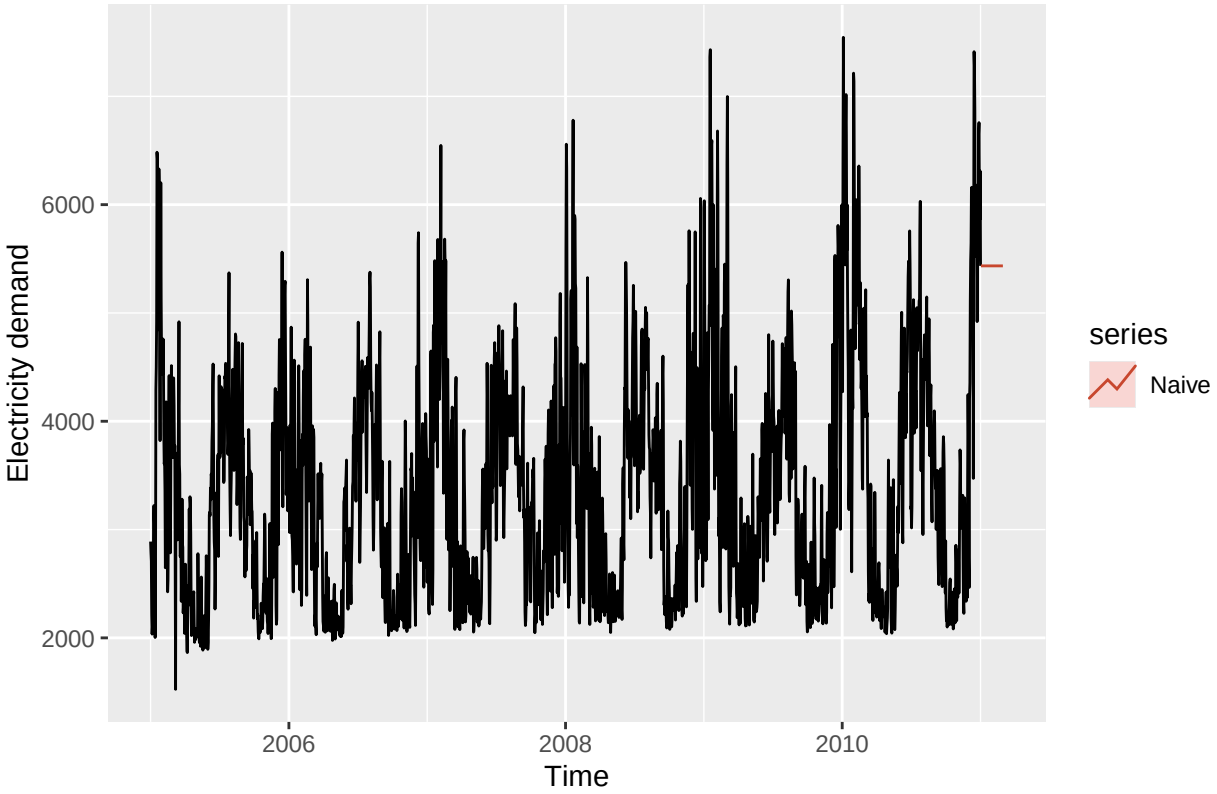
```
## 2011.0082       5434.625 4193.09230  6676.158  3535.86458  7333.385
## 2011.0110       5434.625 4001.02652  6868.223  3242.12532  7627.125
## 2011.0137       5434.625 3831.81317  7037.437  2983.33584  7885.914
## 2011.0164       5434.625 3678.83261  7190.417  2749.37227  8119.878
## 2011.0192       5434.625 3538.15247  7331.098  2534.22055  8335.029
## 2011.0219       5434.625 3407.21058  7462.039  2333.96222  8535.288
## 2011.0247       5434.625 3284.22728  7585.023  2145.87549  8723.375
## 2011.0274       5434.625 3167.90677  7701.343  1967.97863  8901.271
## 2011.0301       5434.625 3057.27087  7811.979  1798.77561  9070.474
## 2011.0329       5434.625 2951.55959  7917.690  1637.10417  9232.146
## 2011.0356       5434.625 2850.16858  8019.081  1482.04000  9387.210
## 2011.0384       5434.625 2752.60783  8116.642  1332.83370  9536.416
## 2011.0411       5434.625 2658.47348  8210.777  1188.86763  9680.382
## 2011.0438       5434.625 2567.42804  8301.822  1049.62565  9819.624
## 2011.0466       5434.625 2479.18602  8390.064   914.67113  9954.579
## 2011.0493       5434.625 2393.50338  8475.747   783.63083 10085.619
## 2011.0521       5434.625 2310.16955  8559.080   656.18274 10213.067
## 2011.0548       5434.625 2229.00134  8640.249   532.04668 10337.203
## 2011.0575       5434.625 2149.83822  8719.412   410.97714 10458.273
## 2011.0603       5434.625 2072.53854  8796.711   292.75748 10576.493
## 2011.0630       5434.625 1996.97661  8872.273   177.19547 10692.055
## 2011.0658       5434.625 1923.04022  8946.210    64.11953 10805.130
## 2011.0685       5434.625 1850.62880  9018.621   -46.62419 10915.874
## 2011.0712       5434.625 1779.65168  9089.598  -155.17432 11024.424
## 2011.0740       5434.625 1710.02689  9159.223  -261.65625 11130.906
## 2011.0767       5434.625 1641.67994  9227.570  -366.18389 11235.434
## 2011.0795       5434.625 1574.54296  9294.707  -468.86105 11338.111
## 2011.0822       5434.625 1508.55387  9360.696  -569.78265 11439.033
## 2011.0849       5434.625 1443.65573  9425.594  -669.03578 11538.286
## 2011.0877       5434.625 1379.79617  9489.454  -766.70055 11635.951
## 2011.0904       5434.625 1316.92686  9552.323  -862.85087 11732.101
## 2011.0932       5434.625 1255.00311  9614.247  -957.55507 11826.805
## 2011.0959       5434.625 1193.98350  9675.266 -1050.87650 11920.127
## 2011.0986       5434.625 1133.82956  9735.420 -1142.87403 12012.124
## 2011.1014       5434.625 1074.50544  9794.745 -1233.60244 12102.852
## 2011.1041       5434.625 1015.97772  9853.272 -1323.11285 12192.363
## 2011.1068       5434.625  958.21518  9911.035 -1411.45304 12280.703
## 2011.1096       5434.625  901.18855  9968.061 -1498.66774 12367.918
## 2011.1123       5434.625  844.87041 10024.380 -1584.79891 12454.049
## 2011.1151       5434.625  789.23499 10080.015 -1669.88594 12539.136
## 2011.1178       5434.625  734.25804 10134.992 -1753.96592 12623.216
## 2011.1205       5434.625  679.91674 10189.333 -1837.07378 12706.324
## 2011.1233       5434.625  626.18951 10243.060 -1919.24247 12788.492
## 2011.1260       5434.625  573.05602 10296.194 -2000.50314 12869.753
## 2011.1288       5434.625  520.49699 10348.753 -2080.88524 12950.135
## 2011.1315       5434.625  468.49418 10400.756 -2160.41667 13029.667
## 2011.1342       5434.625  417.03031 10452.220 -2239.12387 13108.374
## 2011.1370       5434.625  366.08896 10503.161 -2317.03194 13186.282
## 2011.1397       5434.625  315.65452 10553.595 -2394.16476 13263.415
## 2011.1425       5434.625  265.71217 10603.538 -2470.54500 13339.795
## 2011.1452       5434.625  216.24776 10653.002 -2546.19429 13415.444
## 2011.1479       5434.625  167.24784 10702.002 -2621.13320 13490.383
## 2011.1507       5434.625  118.69956 10750.550 -2695.38139 13564.631
## 2011.1534       5434.625   70.59065 10798.659 -2768.95761 13638.208
```

```
## 2011.1562       5434.625   22.90941 10846.341 -2841.87978 13711.130
## 2011.1589       5434.625  -24.35538 10893.605 -2914.16503 13783.415
## 2011.1616       5434.625  -71.21444 10940.464 -2985.82978 13855.080
```

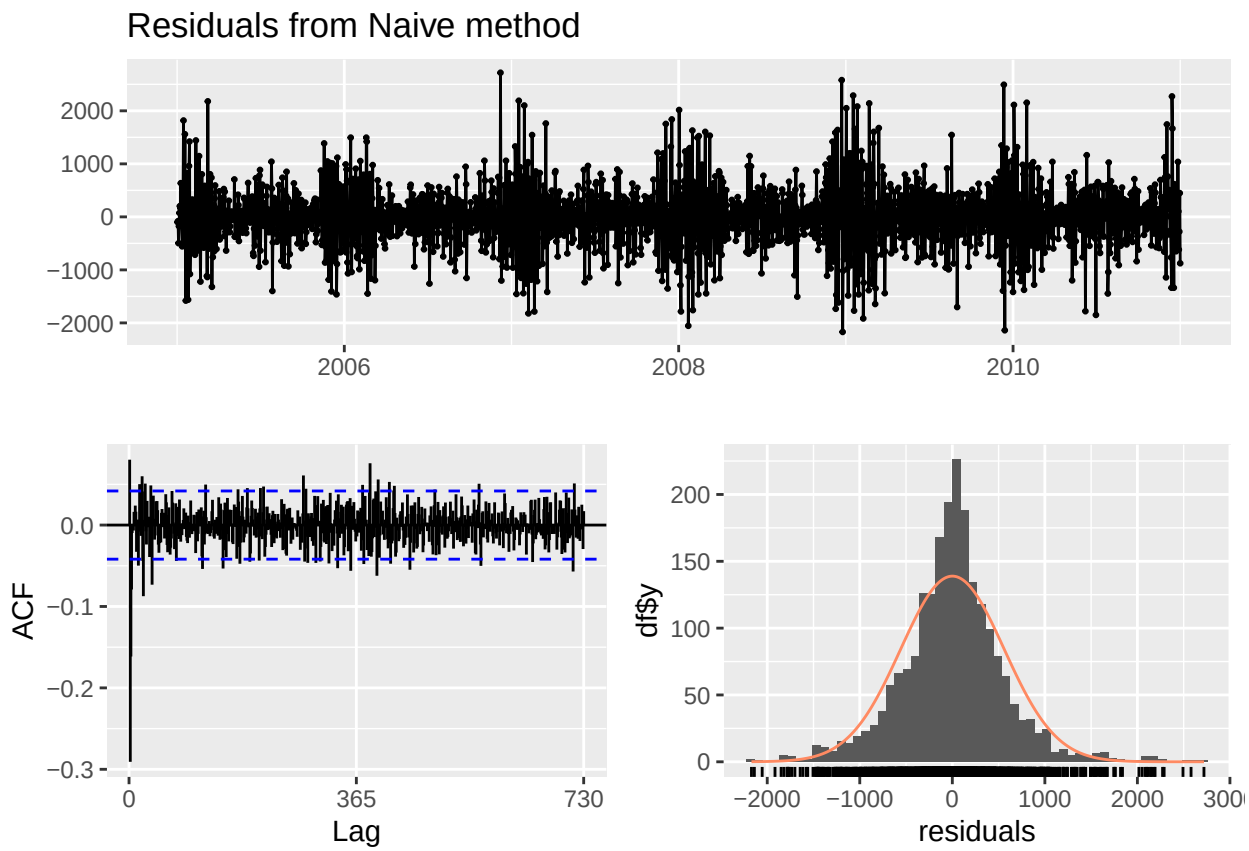
```r
plot(fc_naive_full)
```




```r
#Plot model + observed data
autoplot(y_full) +
  autolayer(fc_naive_full, series="Naive",PI=FALSE) +
  ylab("Electricity demand")
```




```
checkresiduals(fc_naive_full)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Naive method
## Q* = 888.66, df = 438, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 438
```

```
# Export to CSV
submission <- template %>%
  mutate(
    date = format(forecast_dates, "%Y-%m-%d"),
    load = as.numeric(fc_naive_full$mean)
  )
out_name <- "submission_final_Naive.csv"
readr::write_csv(submission, file.path(output_dir, out_name))
```

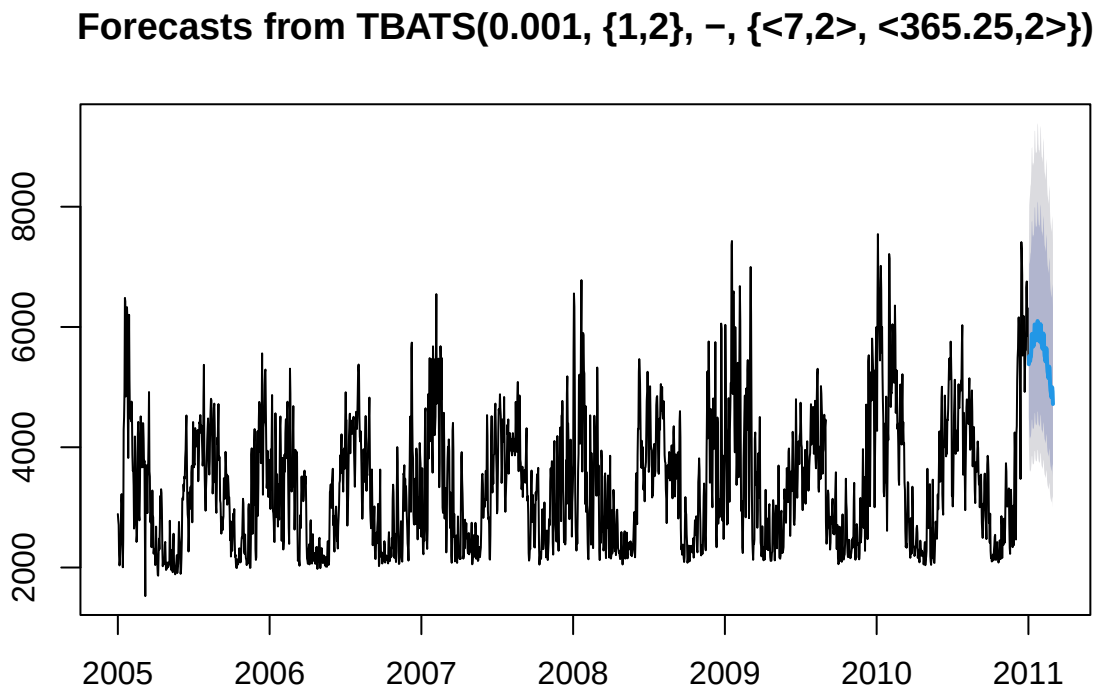
```
#Model 3: TBATS

fc_tbats_full       <- forecast(tbats(y_full),   h = h_future)
print(fc_tbats_full)
```

```
##           Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2011.0027       5380.085 4500.220 6431.809 4094.244 7069.322
## 2011.0055       5517.120 4318.157 7048.632 3792.779 8024.480
```

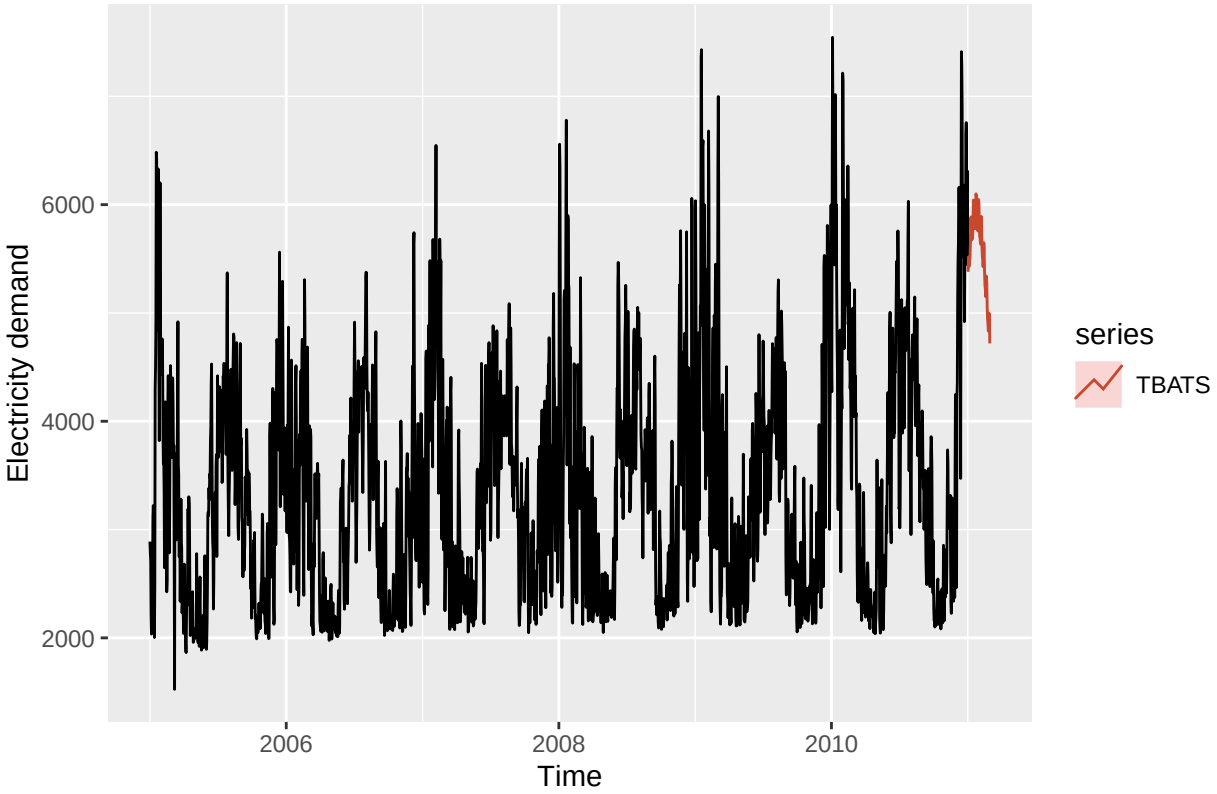
```
## 2011.0082       5429.911 4170.880 7068.590 3627.191 8127.492
## 2011.0110       5431.497 4143.706 7119.079 3590.553 8215.164
## 2011.0137       5488.393 4174.928 7214.637 3612.029 8338.276
## 2011.0164       5502.866 4179.752 7244.364 3613.371 8379.185
## 2011.0192       5641.948 4281.575 7434.080 3699.650 8602.675
## 2011.0219       5876.817 4457.006 7748.428 3849.954 8969.423
## 2011.0247       5886.765 4462.254 7765.537 3853.437 8991.674
## 2011.0274       5719.353 4333.369 7548.151 3741.230 8742.081
## 2011.0301       5674.506 4297.585 7492.105 3709.511 8679.097
## 2011.0329       5702.158 4316.805 7531.617 3725.314 8726.710
## 2011.0356       5693.586 4308.633 7523.231 3717.493 8718.793
## 2011.0384       5817.851 4401.000 7690.345 3796.424 8914.255
## 2011.0411       6042.138 4568.961 7989.797 3940.533 9263.191
## 2011.0438       6035.782 4562.421 7984.425 3934.101 9258.824
## 2011.0466       5848.697 4419.302 7739.918 3809.915 8977.119
## 2011.0493       5787.844 4371.665 7662.290 3768.090 8888.863
## 2011.0521       5801.133 4380.058 7682.764 3774.574 8914.384
## 2011.0548       5777.578 4360.632 7654.446 3757.086 8883.294
## 2011.0575       5888.533 4442.730 7804.333 3827.070 9059.019
## 2011.0603       6099.819 4600.453 8087.322 3962.167 9389.322
## 2011.0630       6077.697 4582.050 8061.014 3945.534 9360.636
## 2011.0658       5874.103 4426.870 7793.950 3811.141 9052.340
## 2011.0685       5797.953 4367.842 7695.799 3759.575 8940.118
## 2011.0712       5796.233 4364.921 7696.378 3756.321 8942.549
## 2011.0740       5757.776 4334.343 7648.166 3729.270 8888.285
## 2011.0767       5853.211 4404.567 7777.789 3788.953 9040.682
## 2011.0795       6047.635 4549.217 8039.060 3912.633 9346.174
## 2011.0822       6010.266 4519.425 7992.360 3886.243 9293.710
## 2011.0849       5794.126 4355.250 7707.856 3744.319 8964.676
## 2011.0877       5704.534 4286.308 7591.499 3684.321 8831.087
## 2011.0904       5688.513 4272.693 7572.975 3671.900 8811.259
## 2011.0932       5636.700 4232.207 7506.775 3636.393 8735.946
## 2011.0959       5716.005 4290.188 7615.166 3685.500 8863.794
## 2011.0986       5891.483 4420.299 7851.777 3796.547 9140.945
## 2011.1014       5840.991 4380.796 7787.361 3761.882 9067.725
## 2011.1041       5617.573 4211.648 7492.309 3615.911 8725.897
## 2011.1068       5517.787 4135.304 7361.944 3549.671 8575.745
## 2011.1096       5489.650 4112.709 7327.087 3529.590 8536.798
## 2011.1123       5427.375 4064.559 7246.633 3487.588 8444.702
## 2011.1151       5491.567 4111.144 7334.994 3526.884 8549.308
## 2011.1178       5647.908 4226.672 7546.516 3625.306 8797.514
## 2011.1205       5587.656 4180.047 7468.750 3584.618 8708.546
## 2011.1233       5362.831 4010.361 7170.912 3438.423 8362.919
## 2011.1260       5256.980 3929.760 7031.955 3368.662 8202.458
## 2011.1288       5219.955 3900.663 6984.971 3343.074 8149.223
## 2011.1315       5150.960 3847.700 6895.163 3297.044 8046.000
## 2011.1342       5202.327 3884.674 6966.425 3328.094 8130.701
## 2011.1370       5340.965 3986.781 7154.616 3414.928 8351.914
## 2011.1397       5274.970 3936.084 7068.787 3370.853 8253.312
## 2011.1425       5054.418 3770.113 6775.744 3228.082 7912.715
## 2011.1452       4946.868 3688.545 6633.987 3157.630 7748.667
## 2011.1479       4904.658 3655.750 6579.759 3128.957 7686.798
## 2011.1507       4832.918 3600.971 6485.867 3081.479 7578.562
## 2011.1534       4874.508 3630.664 6544.015 3106.302 7647.949
```

```
## 2011.1562       4998.020 3721.347 6712.196 3183.292 7845.965
## 2011.1589       4930.330 3669.618 6623.686 3138.442 7743.986
## 2011.1616       4718.870 3510.936 6341.933 3002.143 7416.024
```

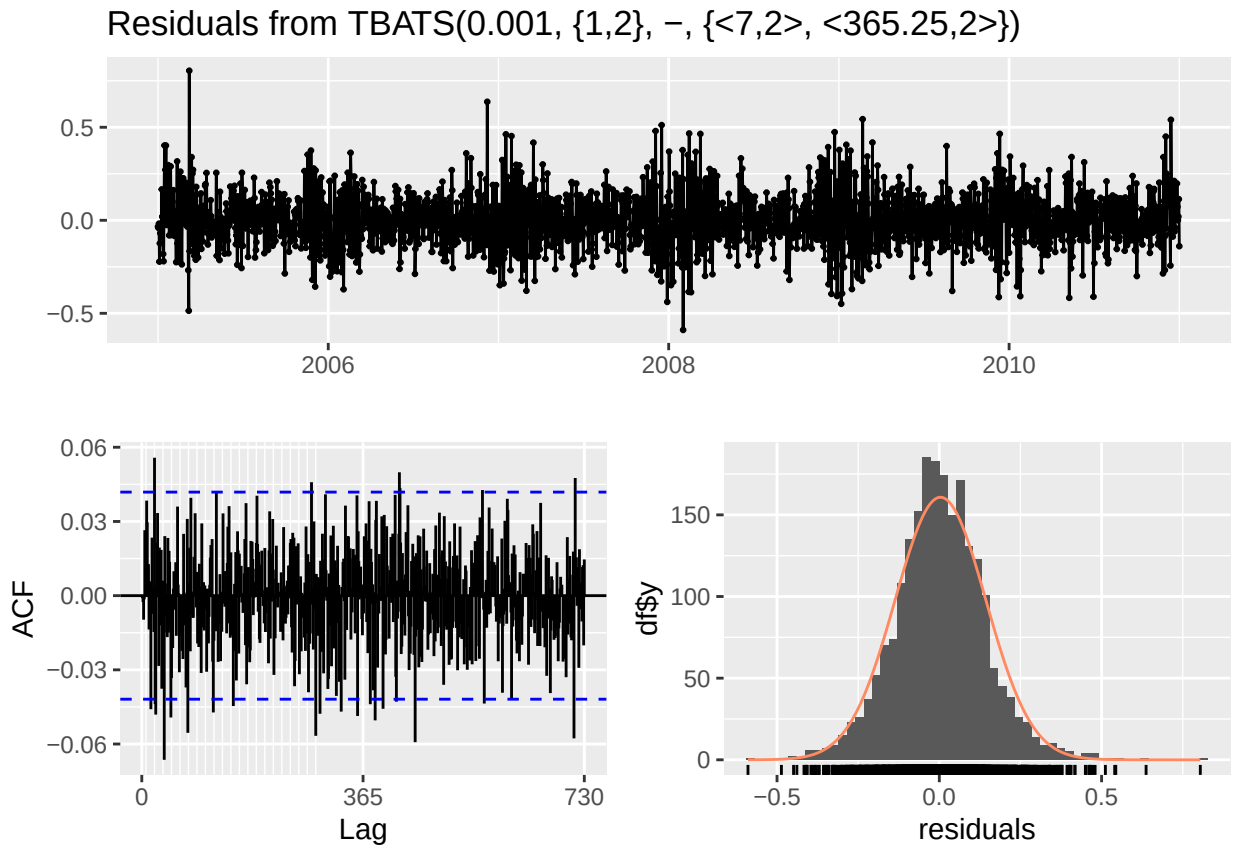
```r
plot(fc_tbats_full)
```




```r
#Plot model + observed data
autoplot(y_full) +
  autolayer(fc_tbats_full, series="TBATS",PI=FALSE) +
  ylab("Electricity demand")
```




```
checkresiduals(fc_tbats_full)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from TBATS(0.001, {1,2}, -, {<7,2>, <365.25,2>})
## Q* = 437.14, df = 438, p-value = 0.5027
##
## Model df: 0.   Total lags used: 438
```

```
# Export to CSV
submission <- template %>%
  mutate(
    date = format(forecast_dates, "%Y-%m-%d"),
    load = as.numeric(fc_tbats_full$mean)
  )
out_name <- "submission_final_TBATS.csv"
readr::write_csv(submission, file.path(output_dir, out_name))
```

```
#Model 4-6: TBATS + regressors


#-- 1) Forecast temp & humidity into 2011 (no drift/mean) --#
fit_temp <- auto.arima(full_train$temp_c, seasonal=FALSE,
                       allowdrift=FALSE, allowmean=FALSE)
fit_hum  <- auto.arima(full_train$rh_pct, seasonal=FALSE,
                       allowdrift=FALSE, allowmean=FALSE)
```

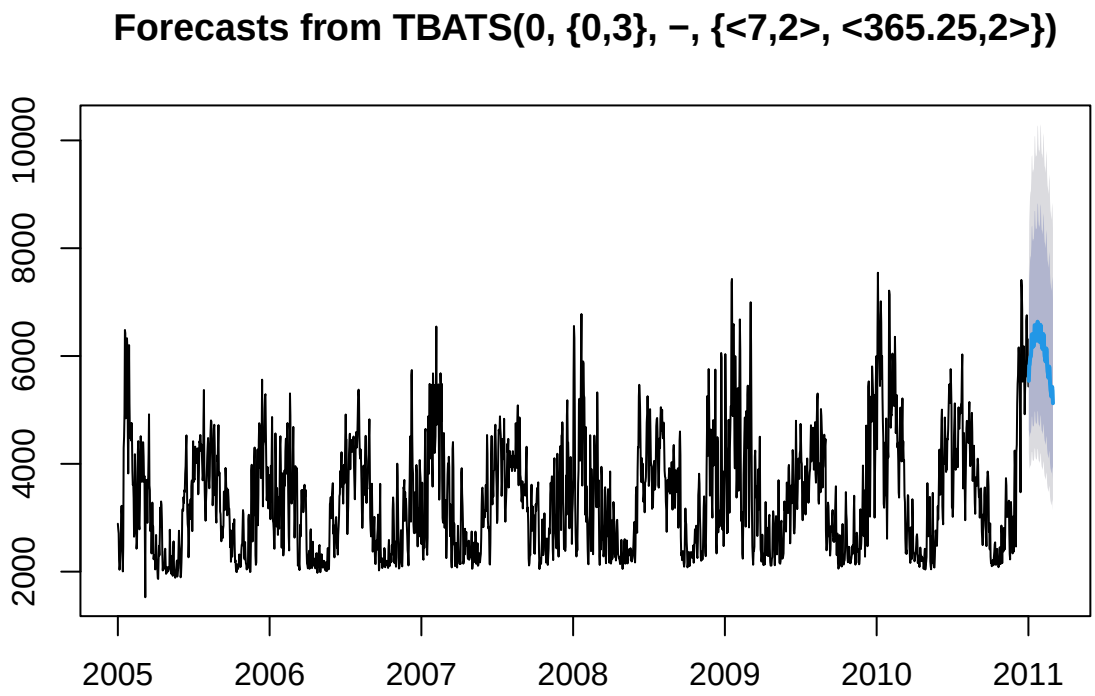
```r
fc_temp <- forecast(fit_temp, h = h_future)$mean
fc_hum  <- forecast(fit_hum,  h = h_future)$mean

#-- 2) Build xreg matrices - just the covariates --#
# Historical (2005-2010):
xreg_full_temp <- matrix(full_train$temp_c, ncol=1)
xreg_full_hum  <- matrix(full_train$rh_pct, ncol=1)
xreg_full_both <- cbind(full_train$temp_c, full_train$rh_pct)

# Future (2011):
xreg_future_temp <- matrix(fc_temp, ncol=1)
xreg_future_hum  <- matrix(fc_hum,  ncol=1)
xreg_future_both <- cbind(as.numeric(fc_temp), as.numeric(fc_hum))
```

```r
# Model 4: TBATS + Temp
fit_tb_temp_full   <- tbats(y_full, xreg = xreg_full_temp)
fc_tbats_temp_full <- forecast(fit_tb_temp_full,
                          h    = h_future,
                          xreg = xreg_future_temp)

print(fc_tbats_temp_full)
```

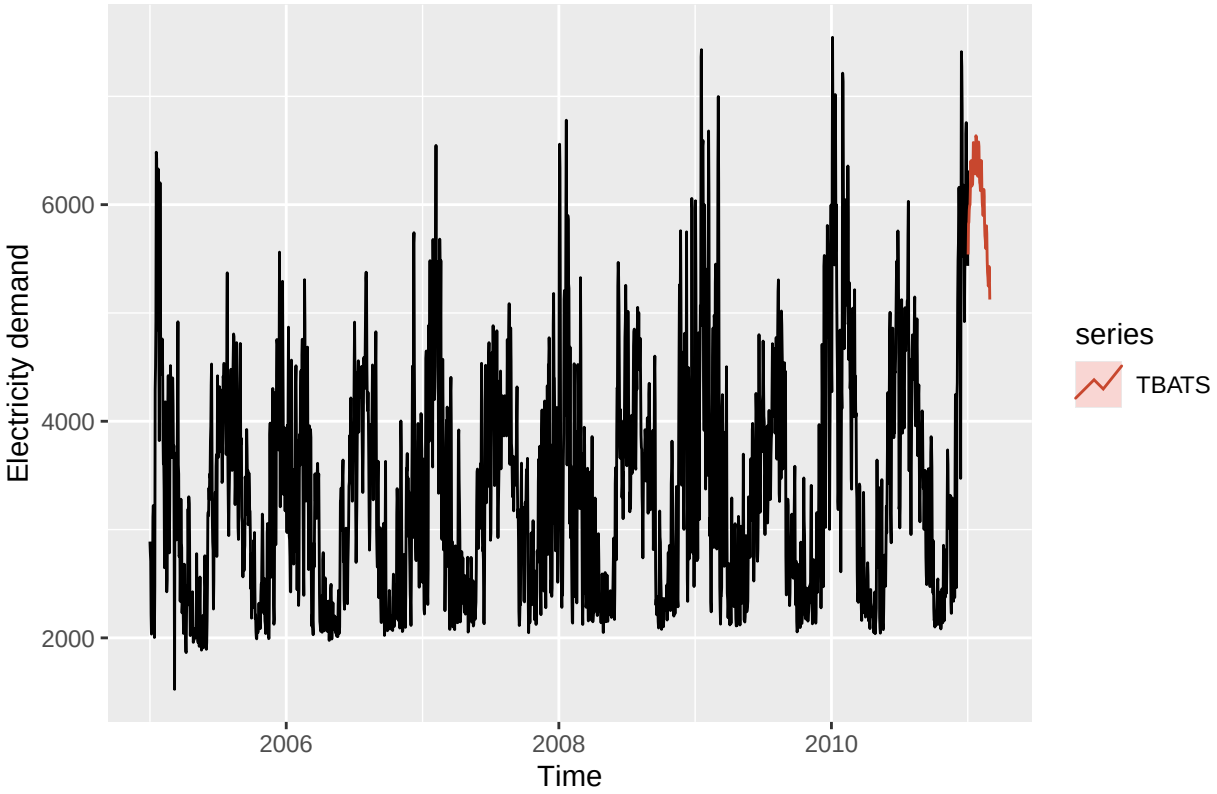
```
##           Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
## 2011.0027       5539.938 4631.264 6626.897 4212.228  7286.146
## 2011.0055       5823.408 4555.477 7444.244 4000.187  8477.625
## 2011.0082       5838.615 4487.720 7596.156 3904.159  8731.567
## 2011.0110       5949.202 4551.732 7775.724 3950.214  8959.770
## 2011.0137       5991.887 4579.571 7839.754 3972.163  9038.580
## 2011.0164       5996.749 4578.487 7854.341 3969.020  9060.423
## 2011.0192       6143.125 4685.347 8054.470 4059.410  9296.422
## 2011.0219       6397.117 4873.989 8396.223 4220.524  9696.213
## 2011.0247       6406.109 4875.788 8416.737 4219.768  9725.233
## 2011.0274       6221.596 4730.475 8182.742 4091.773  9460.020
## 2011.0301       6171.773 4687.774 8125.560 4052.632  9399.024
## 2011.0329       6201.652 4705.643 8173.269 4065.874  9459.341
## 2011.0356       6192.060 4693.569 8168.967 4053.249  9459.475
## 2011.0384       6328.028 4791.749 8356.852 4135.805  9682.260
## 2011.0411       6573.677 4972.710 8690.076 4289.688 10073.747
## 2011.0438       6566.744 4962.450 8689.684 4278.549 10078.681
## 2011.0466       6361.758 4802.712 8426.898 4138.622  9779.093
## 2011.0493       6294.974 4747.531 8346.800 4088.900  9691.286
## 2011.0521       6309.433 4753.679 8374.345 4092.027  9728.418
## 2011.0548       6283.603 4729.499 8348.381 4069.064  9703.376
## 2011.0575       6405.093 4816.149 8518.261 4141.434  9906.042
## 2011.0603       6636.562 4985.251 8834.851 4284.596 10279.605
## 2011.0630       6612.384 4962.190 8811.355 4262.549 10257.624
## 2011.0658       6389.327 4790.092 8522.488 4112.576  9926.504
## 2011.0685       6305.797 4722.839 8419.317 4052.732  9811.425
## 2011.0712       6303.822 4716.746 8424.913 4045.409  9823.028
## 2011.0740       6261.675 4680.645 8376.746 4012.376  9771.910
## 2011.0767       6366.160 4754.123 8524.810 4073.266  9949.754
## 2011.0795       6579.136 4908.407 8818.548 4203.296 10297.876
## 2011.0822       6538.257 4873.204 8772.219 4171.018 10249.010
```

```
## 2011.0849       6301.490 4692.221 8462.683 4014.071  9892.396
## 2011.0877       6203.254 4614.645 8338.747 3945.704  9752.470
## 2011.0904       6185.622 4597.129 8323.005 3928.736  9738.991
## 2011.0932       6128.860 4550.599 8254.501 3887.008  9663.712
## 2011.0959       6215.662 4610.658 8379.381 3936.324  9814.856
## 2011.0986       6407.841 4748.702 8646.663 4052.142 10133.018
## 2011.1014       6352.599 4703.316 8580.225 4011.405 10060.191
## 2011.1041       6107.928 4517.913 8257.526 3851.358  9686.656
## 2011.1068       5998.565 4432.854 8117.294 3776.970  9526.890
## 2011.1096       5967.679 4405.899 8083.072 3752.141  9491.433
## 2011.1123       5899.488 4351.484 7998.182 3703.967  9396.401
## 2011.1151       5969.720 4399.182 8100.952 3742.719  9521.837
## 2011.1178       6140.882 4521.105 8340.976 3844.554  9808.791
## 2011.1205       6074.975 4468.441 8259.104 3797.908  9717.276
## 2011.1233       5828.862 4283.459 7931.821 3638.906  9336.769
## 2011.1260       5712.914 4194.388 7781.202 3561.502  9163.937
## 2011.1288       5672.323 4160.761 7733.020 3531.231  9111.624
## 2011.1315       5596.811 4101.610 7637.073 3479.340  9002.940
## 2011.1342       5652.985 4138.989 7720.785 3509.348  9106.032
## 2011.1370       5804.699 4246.193 7935.232 3598.506  9363.478
## 2011.1397       5732.539 4189.599 7843.711 3548.836  9259.938
## 2011.1425       5491.223 4009.603 7520.329 3394.742  8882.422
## 2011.1452       5373.485 3920.089 7365.735 3317.368  8703.990
## 2011.1479       5327.252 3882.858 7308.949 3284.293  8641.011
## 2011.1507       5248.781 3822.223 7207.770 3231.466  8525.451
## 2011.1534       5294.240 3851.867 7276.725 3254.980  8611.104
## 2011.1562       5429.333 3946.618 7469.093 3333.468  8842.939
## 2011.1589       5355.357 3889.373 7373.900 3283.566  8734.360
## 2011.1616       5124.122 3718.131 7061.780 3137.520  8368.591
```

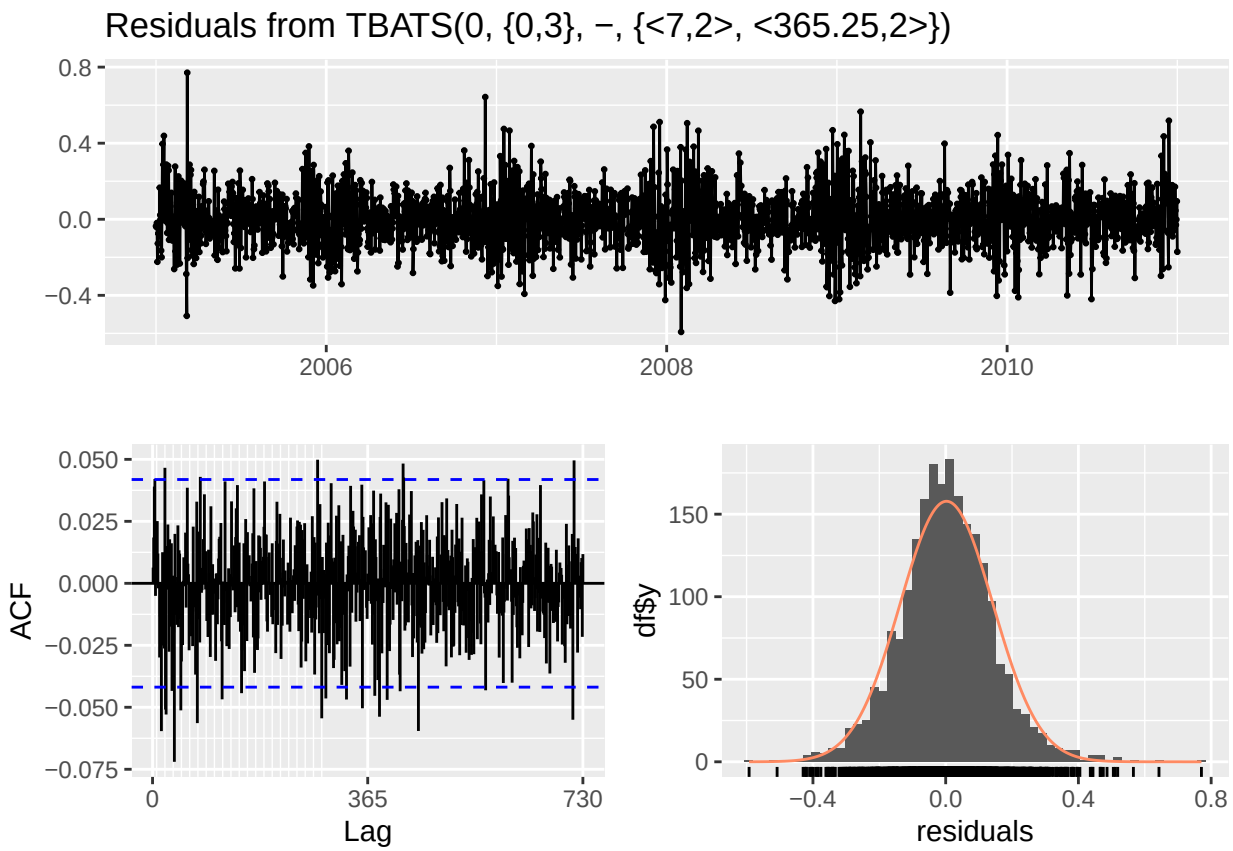
```
plot(fc_tbats_temp_full)
```




```
#Plot model + observed data
autoplot(y_full) +
  autolayer(fc_tbats_temp_full, series="TBATS",PI=FALSE) +
  ylab("Electricity demand")
```




```
checkresiduals(fc_tbats_temp_full)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from TBATS(0, {0,3}, -, {<7,2>, <365.25,2>})
## Q* = 451.67, df = 438, p-value = 0.3157
##
## Model df: 0.   Total lags used: 438
```

```
# Export to CSV
submission <- template %>%
  mutate(
    date = format(forecast_dates, "%Y-%m-%d"),
    load = as.numeric(fc_tbats_temp_full$mean)
  )
out_name <- "submission_final_TBATS_temp.csv"
readr::write_csv(submission, file.path(output_dir, out_name))
```

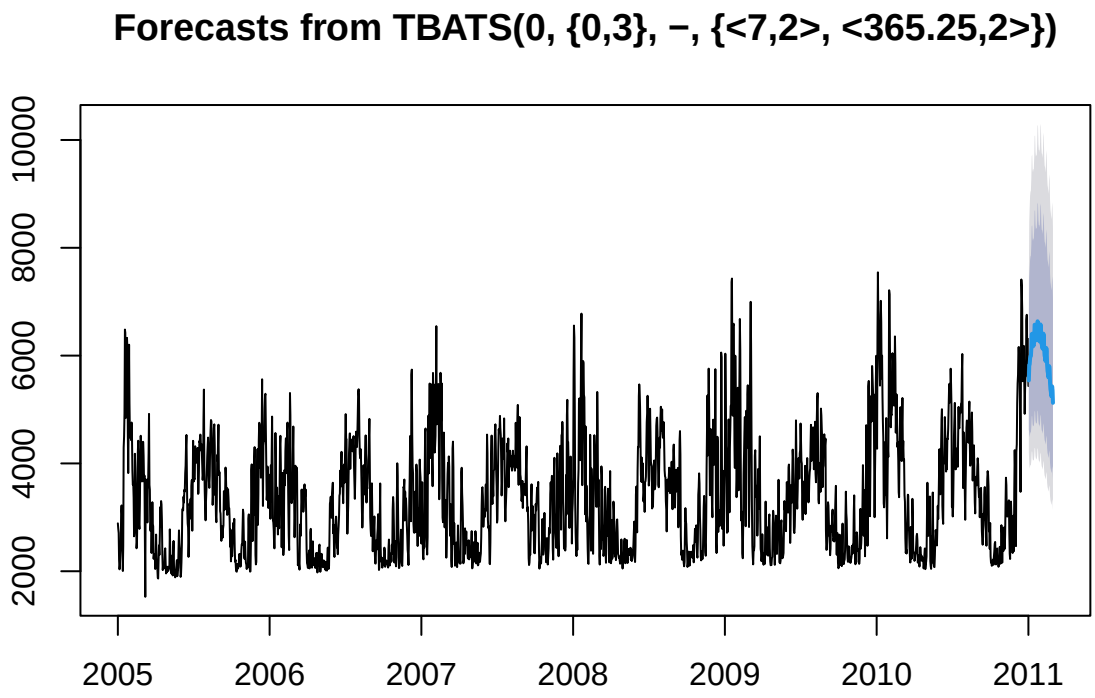
```
# Model TBATS + Hum
fit_tb_hum_full   <- tbats(y_full, xreg = xreg_full_hum)
fc_tbats_hum_full <- forecast(fit_tb_hum_full,
                          h    = h_future,
                          xreg = xreg_future_hum)

print(fc_tbats_hum_full)
```

```
##           Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
```

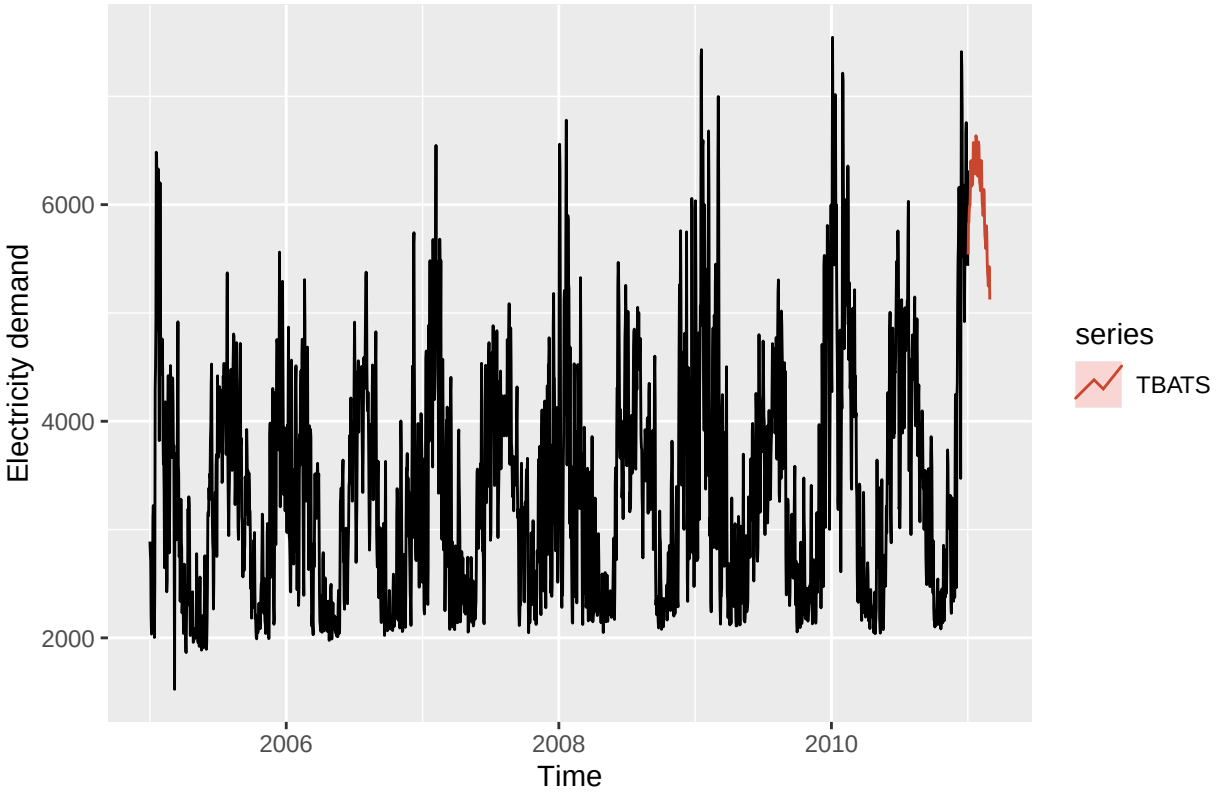
```
## 2011.0027       5539.938 4631.264 6626.897 4212.228  7286.146
## 2011.0055       5823.408 4555.477 7444.244 4000.187  8477.625
## 2011.0082       5838.615 4487.720 7596.156 3904.159  8731.567
## 2011.0110       5949.202 4551.732 7775.724 3950.214  8959.770
## 2011.0137       5991.887 4579.571 7839.754 3972.163  9038.580
## 2011.0164       5996.749 4578.487 7854.341 3969.020  9060.423
## 2011.0192       6143.125 4685.347 8054.470 4059.410  9296.422
## 2011.0219       6397.117 4873.989 8396.223 4220.524  9696.213
## 2011.0247       6406.109 4875.788 8416.737 4219.768  9725.233
## 2011.0274       6221.596 4730.475 8182.742 4091.773  9460.020
## 2011.0301       6171.773 4687.774 8125.560 4052.632  9399.024
## 2011.0329       6201.652 4705.643 8173.269 4065.874  9459.341
## 2011.0356       6192.060 4693.569 8168.967 4053.249  9459.475
## 2011.0384       6328.028 4791.749 8356.852 4135.805  9682.260
## 2011.0411       6573.677 4972.710 8690.076 4289.688 10073.747
## 2011.0438       6566.744 4962.450 8689.684 4278.549 10078.681
## 2011.0466       6361.758 4802.712 8426.898 4138.622  9779.093
## 2011.0493       6294.974 4747.531 8346.800 4088.900  9691.286
## 2011.0521       6309.433 4753.679 8374.345 4092.027  9728.418
## 2011.0548       6283.603 4729.499 8348.381 4069.064  9703.376
## 2011.0575       6405.093 4816.149 8518.261 4141.434  9906.042
## 2011.0603       6636.562 4985.251 8834.851 4284.596 10279.605
## 2011.0630       6612.384 4962.190 8811.355 4262.549 10257.624
## 2011.0658       6389.327 4790.092 8522.488 4112.576  9926.504
## 2011.0685       6305.797 4722.839 8419.317 4052.732  9811.425
## 2011.0712       6303.822 4716.746 8424.913 4045.409  9823.028
## 2011.0740       6261.675 4680.645 8376.746 4012.376  9771.910
## 2011.0767       6366.160 4754.123 8524.810 4073.266  9949.754
## 2011.0795       6579.136 4908.407 8818.548 4203.296 10297.876
## 2011.0822       6538.257 4873.204 8772.219 4171.018 10249.010
## 2011.0849       6301.490 4692.221 8462.683 4014.071  9892.396
## 2011.0877       6203.254 4614.645 8338.747 3945.704  9752.470
## 2011.0904       6185.622 4597.129 8323.005 3928.736  9738.991
## 2011.0932       6128.860 4550.599 8254.501 3887.008  9663.712
## 2011.0959       6215.662 4610.658 8379.381 3936.324  9814.856
## 2011.0986       6407.841 4748.702 8646.663 4052.142 10133.018
## 2011.1014       6352.599 4703.316 8580.225 4011.405 10060.191
## 2011.1041       6107.928 4517.913 8257.526 3851.358  9686.656
## 2011.1068       5998.565 4432.854 8117.294 3776.970  9526.890
## 2011.1096       5967.679 4405.899 8083.072 3752.141  9491.433
## 2011.1123       5899.488 4351.484 7998.182 3703.967  9396.401
## 2011.1151       5969.720 4399.182 8100.952 3742.719  9521.837
## 2011.1178       6140.882 4521.105 8340.976 3844.554  9808.791
## 2011.1205       6074.975 4468.441 8259.104 3797.908  9717.276
## 2011.1233       5828.862 4283.459 7931.821 3638.906  9336.769
## 2011.1260       5712.914 4194.388 7781.202 3561.502  9163.937
## 2011.1288       5672.323 4160.761 7733.020 3531.231  9111.624
## 2011.1315       5596.811 4101.610 7637.073 3479.340  9002.940
## 2011.1342       5652.985 4138.989 7720.785 3509.348  9106.032
## 2011.1370       5804.699 4246.193 7935.232 3598.506  9363.478
## 2011.1397       5732.539 4189.599 7843.711 3548.836  9259.938
## 2011.1425       5491.223 4009.603 7520.329 3394.742  8882.422
## 2011.1452       5373.485 3920.089 7365.735 3317.368  8703.990
## 2011.1479       5327.252 3882.858 7308.949 3284.293  8641.011
```

```
## 2011.1507       5248.781 3822.223 7207.770 3231.466  8525.451
## 2011.1534       5294.240 3851.867 7276.725 3254.980  8611.104
## 2011.1562       5429.333 3946.618 7469.093 3333.468  8842.939
## 2011.1589       5355.357 3889.373 7373.900 3283.566  8734.360
## 2011.1616       5124.122 3718.131 7061.780 3137.520  8368.591
```

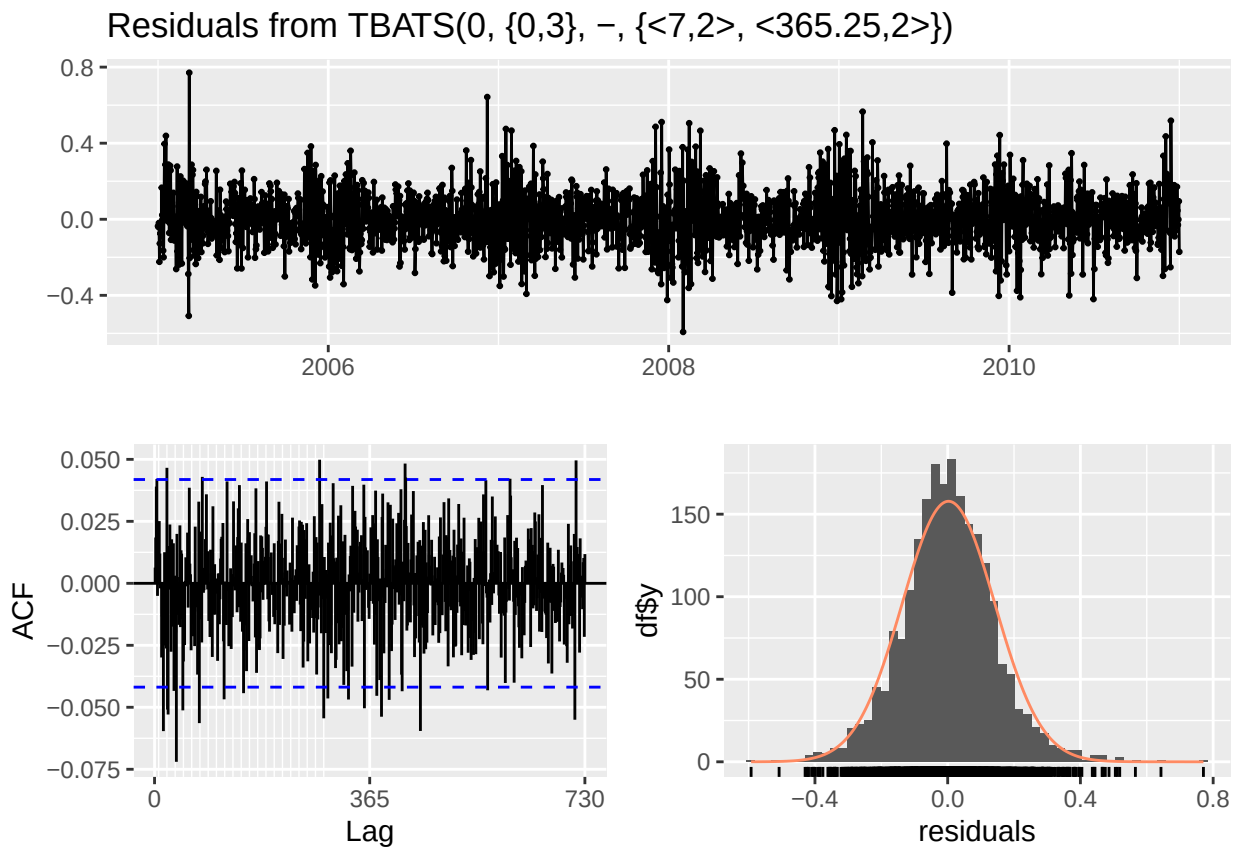
```r
plot(fc_tbats_hum_full)
```




```r
#Plot model + observed data
autoplot(y_full) +
  autolayer(fc_tbats_hum_full, series="TBATS",PI=FALSE) +
  ylab("Electricity demand")
```




```
checkresiduals(fc_tbats_hum_full)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from TBATS(0, {0,3}, -, {<7,2>, <365.25,2>})
## Q* = 451.67, df = 438, p-value = 0.3157
##
## Model df: 0.   Total lags used: 438
```

```r
# Export to CSV
submission <- template %>%
  mutate(
    date = format(forecast_dates, "%Y-%m-%d"),
    load = as.numeric(fc_tbats_hum_full$mean)
  )
out_name <- "submission_final_TBATS_hum.csv"
readr::write_csv(submission, file.path(output_dir, out_name))
```

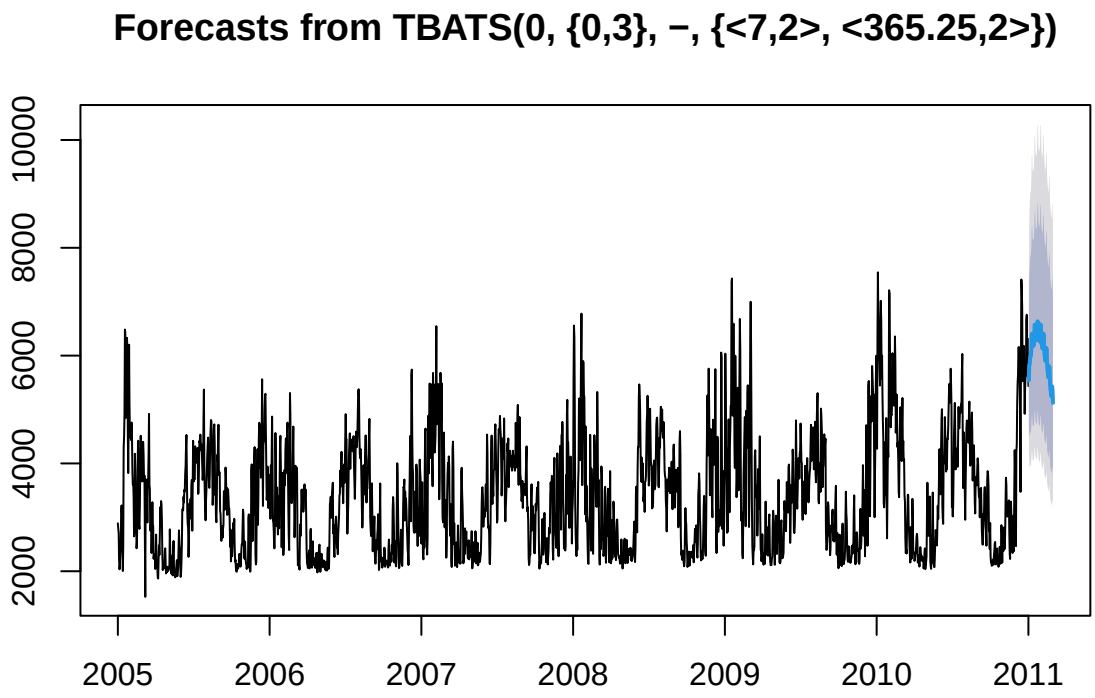
```r
# Model 6: TBATS + Both
fit_tb_both_full   <- tbats(y_full, xreg = xreg_full_both)
fc_tbats_both_full <- forecast(fit_tb_both_full,
                           h    = h_future,
                           xreg = xreg_future_both)

print(fc_tbats_both_full)
```

```
##           Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
```

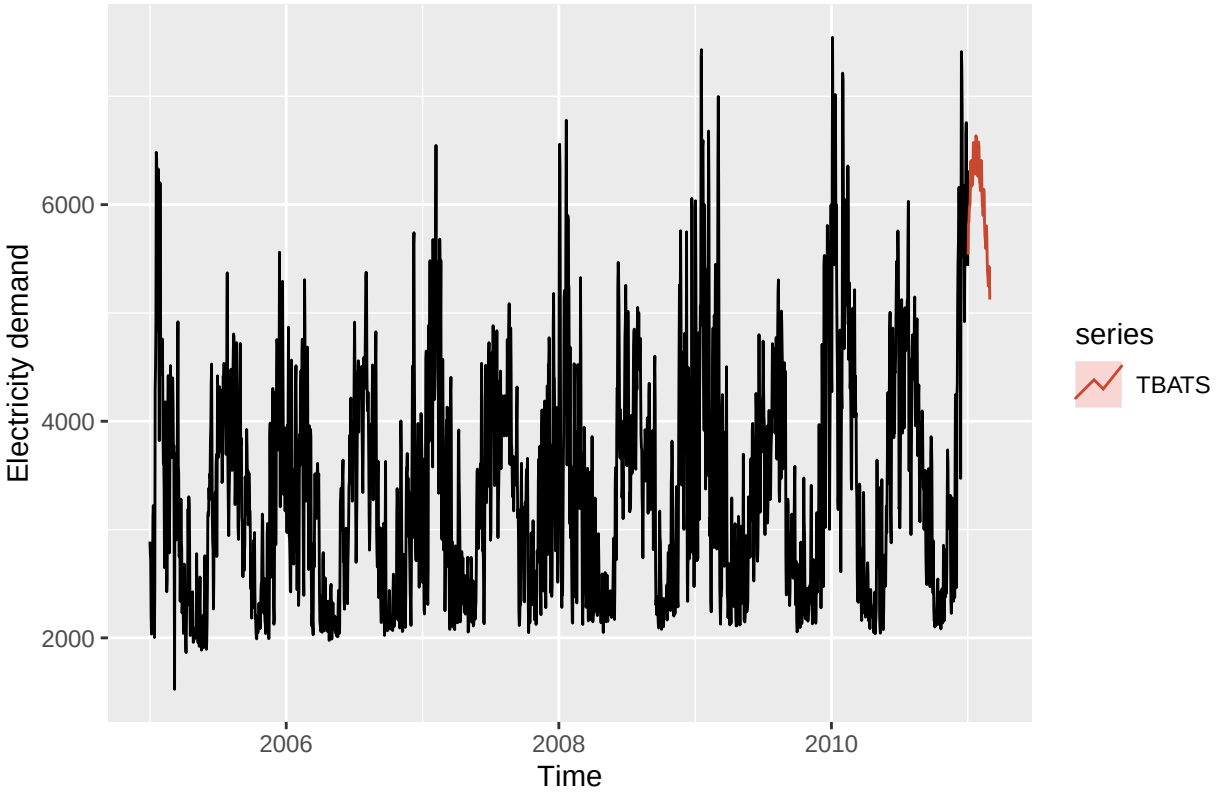
```
## 2011.0027       5539.938 4631.264 6626.897 4212.228  7286.146
## 2011.0055       5823.408 4555.477 7444.244 4000.187  8477.625
## 2011.0082       5838.615 4487.720 7596.156 3904.159  8731.567
## 2011.0110       5949.202 4551.732 7775.724 3950.214  8959.770
## 2011.0137       5991.887 4579.571 7839.754 3972.163  9038.580
## 2011.0164       5996.749 4578.487 7854.341 3969.020  9060.423
## 2011.0192       6143.125 4685.347 8054.470 4059.410  9296.422
## 2011.0219       6397.117 4873.989 8396.223 4220.524  9696.213
## 2011.0247       6406.109 4875.788 8416.737 4219.768  9725.233
## 2011.0274       6221.596 4730.475 8182.742 4091.773  9460.020
## 2011.0301       6171.773 4687.774 8125.560 4052.632  9399.024
## 2011.0329       6201.652 4705.643 8173.269 4065.874  9459.341
## 2011.0356       6192.060 4693.569 8168.967 4053.249  9459.475
## 2011.0384       6328.028 4791.749 8356.852 4135.805  9682.260
## 2011.0411       6573.677 4972.710 8690.076 4289.688 10073.747
## 2011.0438       6566.744 4962.450 8689.684 4278.549 10078.681
## 2011.0466       6361.758 4802.712 8426.898 4138.622  9779.093
## 2011.0493       6294.974 4747.531 8346.800 4088.900  9691.286
## 2011.0521       6309.433 4753.679 8374.345 4092.027  9728.418
## 2011.0548       6283.603 4729.499 8348.381 4069.064  9703.376
## 2011.0575       6405.093 4816.149 8518.261 4141.434  9906.042
## 2011.0603       6636.562 4985.251 8834.851 4284.596 10279.605
## 2011.0630       6612.384 4962.190 8811.355 4262.549 10257.624
## 2011.0658       6389.327 4790.092 8522.488 4112.576  9926.504
## 2011.0685       6305.797 4722.839 8419.317 4052.732  9811.425
## 2011.0712       6303.822 4716.746 8424.913 4045.409  9823.028
## 2011.0740       6261.675 4680.645 8376.746 4012.376  9771.910
## 2011.0767       6366.160 4754.123 8524.810 4073.266  9949.754
## 2011.0795       6579.136 4908.407 8818.548 4203.296 10297.876
## 2011.0822       6538.257 4873.204 8772.219 4171.018 10249.010
## 2011.0849       6301.490 4692.221 8462.683 4014.071  9892.396
## 2011.0877       6203.254 4614.645 8338.747 3945.704  9752.470
## 2011.0904       6185.622 4597.129 8323.005 3928.736  9738.991
## 2011.0932       6128.860 4550.599 8254.501 3887.008  9663.712
## 2011.0959       6215.662 4610.658 8379.381 3936.324  9814.856
## 2011.0986       6407.841 4748.702 8646.663 4052.142 10133.018
## 2011.1014       6352.599 4703.316 8580.225 4011.405 10060.191
## 2011.1041       6107.928 4517.913 8257.526 3851.358  9686.656
## 2011.1068       5998.565 4432.854 8117.294 3776.970  9526.890
## 2011.1096       5967.679 4405.899 8083.072 3752.141  9491.433
## 2011.1123       5899.488 4351.484 7998.182 3703.967  9396.401
## 2011.1151       5969.720 4399.182 8100.952 3742.719  9521.837
## 2011.1178       6140.882 4521.105 8340.976 3844.554  9808.791
## 2011.1205       6074.975 4468.441 8259.104 3797.908  9717.276
## 2011.1233       5828.862 4283.459 7931.821 3638.906  9336.769
## 2011.1260       5712.914 4194.388 7781.202 3561.502  9163.937
## 2011.1288       5672.323 4160.761 7733.020 3531.231  9111.624
## 2011.1315       5596.811 4101.610 7637.073 3479.340  9002.940
## 2011.1342       5652.985 4138.989 7720.785 3509.348  9106.032
## 2011.1370       5804.699 4246.193 7935.232 3598.506  9363.478
## 2011.1397       5732.539 4189.599 7843.711 3548.836  9259.938
## 2011.1425       5491.223 4009.603 7520.329 3394.742  8882.422
## 2011.1452       5373.485 3920.089 7365.735 3317.368  8703.990
## 2011.1479       5327.252 3882.858 7308.949 3284.293  8641.011
```

```
## 2011.1507       5248.781 3822.223 7207.770 3231.466  8525.451
## 2011.1534       5294.240 3851.867 7276.725 3254.980  8611.104
## 2011.1562       5429.333 3946.618 7469.093 3333.468  8842.939
## 2011.1589       5355.357 3889.373 7373.900 3283.566  8734.360
## 2011.1616       5124.122 3718.131 7061.780 3137.520  8368.591
```

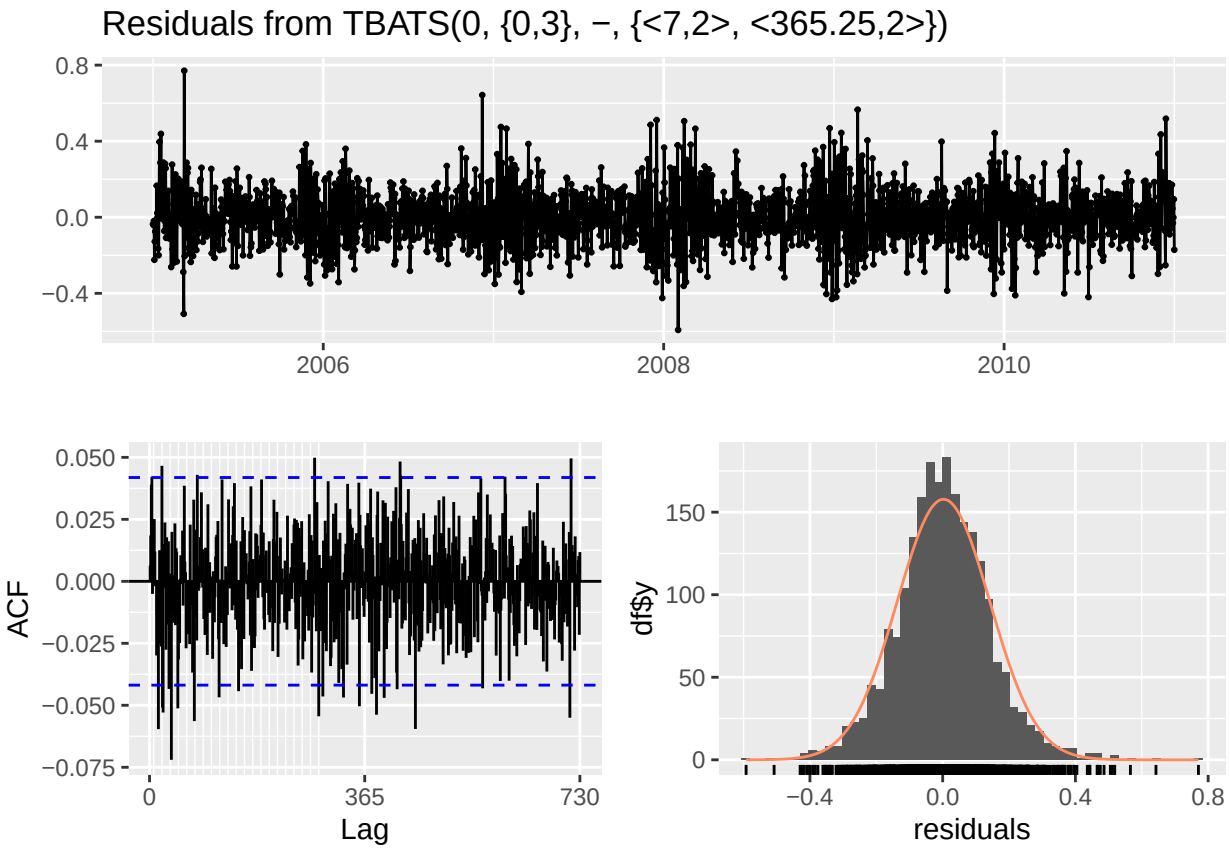
```r
plot(fc_tbats_both_full)
```




```r
#Plot model + observed data
autoplot(y_full) +
  autolayer(fc_tbats_both_full, series="TBATS",PI=FALSE) +
  ylab("Electricity demand")
```




```
checkresiduals(fc_tbats_both_full)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from TBATS(0, {0,3}, -, {<7,2>, <365.25,2>})
## Q* = 451.67, df = 438, p-value = 0.3157
##
## Model df: 0.   Total lags used: 438
```

```
# Export to CSV
submission <- template %>%
  mutate(
    date = format(forecast_dates, "%Y-%m-%d"),
    load = as.numeric(fc_tbats_both_full$mean)
  )
out_name <- "submission_final_TBATS_both.csv"
readr::write_csv(submission, file.path(output_dir, out_name))
```

Graph

```
autoplot(y_full) +
  autolayer(fc_ets_bc_full, PI=FALSE, series="ETS") +
  autolayer(fc_naive_full, PI=FALSE, series="NAIVE") +
  autolayer(fc_tbats_full,PI=FALSE, series="TBATS") +
```

```
autolayer(fc_tbats_temp_full,PI=FALSE, series="TBATS+T") +
autolayer(fc_tbats_hum_full,PI=FALSE, series="TBATS+H") +
autolayer(fc_tbats_both_full,PI=FALSE, series="TBATS+BOTH") +
xlab("Day") + ylab("Daily Electricity demand") +
guides(colour=guide_legend(title="Forecast"))
```