# 1.0 Abstract

**Project title: A Prediction of Individual household Electricity Consumption**

This study aims to explore the association of household electric power consumption with changes in weather conditions such as ambient temperature, precipitation, levels of sunlight, wind, humidity, time and cloud cover etc. By employing dataset for a single household consider other factors such as design, occupancy, activity levels, the condition of windows, equipment, insulation and the usage rate of heavy loads such as ac, washer, dryers, stoves (where installed) etc are constant. Effective prediction of levels of electricity consumption can inform infrastructure planning, shutdown maintenance, family finance as well as statutory activities.

UCI Machine Learning’s “Individual household electric power consumption Data Set” spanning 47 months has been selected this study. The metered power covers lighting, ac, dish washer, washing machine & dryer and other kitchen For this case study the cooking stove and furnace are gas fired loads. Electric stove and furnace are excluded as they are gas fired. This dataset was augmented with publicly available historical weather with 13 attributes for the same city of Sceax, France. The combined dataset would have some 2 million instances and 20 variables.

Some of the research questions we aim to search for includes: which weather features are most strongly associated with household behaviour such as dish washing, the use of AC and laundry? On the average, which weather pattern is more closely associated with people not being at home – i.e. vacant? How does the peak power- hour vary for the 4 seasons of the year? All things being equal, when should the Utility company schedule their 3-week turnaround maintenance? What incentives could be enacted in order to balance the daily and annual consumption? For algorithms I will consider Decision Tree Classification (Support Vector and Naïve Bayes) as well as Multivariable Regression algorithms.

# 2.0 LITERATURE REVIEW

# 2.1 Introduction

Electricity consumption levels is known to vary throughout the year, with occupancy levels of homes, behaviour of occupants and their socio-economic status. The largest single driver of consumption recently is the increasing use of HVAC (heating ventilation and air conditioning) in homes of developed world (Beccali et al 2007). There is a growing interest on this due to increasing energy costs, environmental awareness, statutory interests (e.g. vacancy taxes), development of Codes, evaluation of home building materials and fittings, and sizing of home appliances such as air conditioners (Baccali et al, 2007; Edwards et al 2012).

Continues on Project Report

### 2.1 Drivers of Residential Energy Electricity Consumption

Lucas et al (2001) studied the behavioral factors that influence residential energy consumption in the suburbs through a survey method. They concluded the number of members of the family, their ages and the amount of time they spend at home are important factors that influence energy consumption. They propose an index of consumption ratio per inhabit and hours at home. Worldwide, the Building sector energy consumption has been rising (2% in Europe and 105 in China). They consume 30% of the total energy worldwide and as much as 40% in Europe (Catalina et al 2013). One reason for the growing energy demand is that the performance of buildings degrade with time. One way to arrest this growth in demand is to frequently renew homes (Li et al 2010). Facility managers have perennially struggled with the objective of forecasting and evaluating the energy demand of buildings with air conditioning systems. The consumption tended to “changes in the external climate conditions, occupants’ fluctuations along the day, and the internal loads installed in the building” Neto and Fiorelli (2008). The authors (Neto and Fiorelli, 2008) found that simplified ANN based models are more accurate than the proprietary EnergyPlus simulator (10% versus 13% error rate). Their analysis concluded further that of the effects of dry bulb temperature was more significant than those of humidity than radiation.

Continues on Project Report

### 2.3.1 Study Research Questions

Some of the questions we want to establish include the following

1. What are the most important predictors of electricity consumption for the various levels of resolution(hourly, daily, weekly and monthly)?
2. Which are the most suitable algorithms for predicting electricity consumption in these levels of analysis, in terms of accuracy (RMSE, AIC or otherwise)?
3. What months of the year, days of the week, hours of the day do we have peak and minimum electricity consumption?
4. What is the predicted monthly, weekly and hourly consumption over the next set of cycles (months, weeks, days and hours), as appropriate?
5. What scope is there for introducing incentives to encourage a most balance load profile?

## 2.4 Methods Review and Selection

**Table 1 Methods Screen (adapted from Z Li et al, 2014). Assumes our 2millon dataset**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Method** | **Easy to use?** | **Accuracy** | **Data Use. 2m dataset** | **Data Resolution** | **Non-Linear** | **Calibration Required?** |
| Bin Method | Yes | Medium | Yes. Low | Multiple resolution | No | No |
| Linear Regression | Yes | Medium | Yes. High | Multiple resolution | No | No |
| Support Vector Regression | Yes | High | Yes. High | Multiple resolution | Yes | No |
| Gaussian Process Regression | Yes | High. Calculated | Yes. High | Multiple resolution | Yes | No |
| Artificial Neural Network | No | Medium | Yes. High | Hourly | Yes | Yes |
| Decision Tree | Yes | High | Yes. High | Annual | Yes | No |
| Bayesian Network | Yes | High | Yes. High | Daily | Yes | No |
| RC Network | No | Medium | Yes | Hourly | Yes | Yes |
| Normative | No | Medium | No | Hourly, monthly | Yes | Yes |
| Detailed Simulation | No | High | No | Sub-Hourly | Yes | Yes |

# 3 Datasets

## 3.1 Individual Household Electricity Consumption Dataset

This dataset was donated to UCI Machine Learning by Georges Hebrail on 30th August of 2012. It consists of meter readings of electric power consumption, taken every minute, in one household in Sceax, near Paris France, over a period of 47 months, between December 2006 and November 2010. Other electrical quantities and some sub-metering values are available making a total of 9 attributes. There are some 2,075, 259 instances.

Donors of the dataset reports that it has some missing values (meter values not provided) of around 1.25% of the instances (all calendar timestamps are present, only the readings are missing). We will discuss how these missing values will be treated later in the report.

There is no indication whether or not dataset has Outliers or extreme values in meter readings. We will have to text for it ourselves, using range functions.

### 3.1.1 Attribute Description (taken from UCI Summary)

1.date: Date in format dd/mm/yyyy

2.time: time in format hh:mm:ss

3.global\_active\_power: household global minute-averaged active power (in kilowatt)

4.global\_reactive\_power: household global minute-averaged reactive power (in kilowatt)

5.voltage: minute-averaged voltage (in volt)

6.global\_intensity: household global minute-averaged current intensity (in ampere)

7.sub\_metering\_1: energy sub-metering No. 1 (in watt-hour of active energy). It corresponds to the kitchen, containing mainly a dishwasher, an oven and a microwave (hot plates are not electric but gas powered).

8.sub\_metering\_2: energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room, containing a washing-machine, a tumble-drier, a refrigerator and a light.

9.sub\_metering\_3: energy sub-metering No. 3 (in watt-hour of active energy). It corresponds to an electric water-heater and an air-conditioner

Total electricity consumption in Kilowatt-Hour = active power + [smter1 + smeter2 +smeter3]/1000

## 3.2 Second Dataset on Weather

Temperature, humidity and radiation influence comfort levels and the tendency to use heating and air-conditioning (Neto and Fiorelli, 2008). Wind speed and solar radiation sources have indirect effects on ambient temperature. So also do the levels of precipitation especially snow. Although it is possible to analyse electricity consumption as a time series model purely on its own dataset (e.g Beccali et al, 2007), we would also like to explore its relationships with weather attributes as we know that these affects household occupancy levels and their behavior. Secondly, it is known that some weather attributes e.g. temperature follow a repeated pattern in time so determining the proportion of the variations that can be adduced to weather would be useful for prediction. Indeed, some of the authors reviewed explored weather effects and found some significance with some weather attributes especially temperature and humidity (Heden, 2016; Beccali et al 2007). This is not surprising when you consider household electricity bills also follow a pattern in time, peaking at certain periods of the year and lower during the other months. This is due to the fact that weather and time of day / year has a strong relationship on the deployment of the heavier loads of the home such as air conditioning (Neto and Fiorelli, 2008).

The donors of the data did not collect weather information. When contacted, Mr. Georges Hebmail offered data from the nearest weather station (openly NOAA sourced- see below). There are several sources of weather data referencing the weather station near the location where the individual household data was collected. Once accessed, we shall match the consumption data and testing for some association, with weather information giving us some indication of both behaviour and occupancy of the household.

The open sourced weather data from NOAA had too much missing values (more that 70%). Instead we acquired 30 years data from Sceax, France from openly available weather forecast group meteoblue.com for Euro100 – with Captone Supervisor and Coordinator approval. I was given access for 30 years which we downloaded. For modelling we also downloaded for those 47 months of interest December 2006 to December 2010.

### 3.2.1 Attribute Description

Year; Month; Day; Hour; Minute;

Temperature [2 m above gnd] deg C or °C;

Relative Humidity [2 m above gnd] %;

Mean Sea Level Pressure [MSL] hPa;

Total Precipitation [mm];

Snowfall amount raw [cm];

Total Cloud Cover [%];

High Cloud Cover [high cld lay] %;

Medium Cloud Cover [mid cld lay] %;

Low Cloud Cover [low cld lay] %;

Sunshine Duration [min];

Shortwave Radiation [W/m2];

Wind Speed [10 m above gnd] Km/h;

Wind Direction [10 m above gnd] degrees;

Wind Speed [80 m above gnd] Km/h;

Wind Direction [80 m above gnd] degrees;

Wind Speed [900 mb] Km/h;

Wind Direction [900 mb] degrees;

Wind Gust [Km/h]

### 3.2.2 Weather variables information (Taken from Meteorblue.com)

**Temperature (2m) and relative humidity (2m):** Comparable to measurements at 2 meters above ground.

**Pressure:** Atmospheric air pressure reduced to mean sea level is the most commonly used for weather reports. The local pressure varies with altitude. Locations at higher elevation have a lower local atmospheric pressure.

**Precipitation amount:** Total precipitation amount including rain, convective precipitation and snow.

**Snowfall amount:** Fraction of total precipitation that falls down as snow and is converted to cm instead of mm.

**Total cloud cover:** Percentage of the sky that is covered with clouds: 50% means half of the sky is covered. 0-25% clear sky, 25-50% partly cloudy, 50-85% mostly cloudy and above 85% overcast.

**Low, mid and high cloud cover**: Cloud cover at different altitudes. High clouds (8-15 km) like cirrus are less significant for total cloud cover than low (below 4 km) like stratus, cumulus and fog or mid clouds (4-8 km) like alto cumulus and alto stratus.

**Solar radiation**: Global radiation (diffuse and direct) on a horizontal plane given in Watt per square meter.

**Wind speed:** Hourly average wind speeds at given altitude levels 10 or 80 meters above ground or pressure level "900 hPa". We expect 10m altitude to be more relevant houses are rather low in profile

**Wind direction:** Wind direction in degrees seamless from 0° (wind blowing from north), 90° (east wind), 180° (south wind) and 270° (west wind).

**Wind gusts**: Short term wind speed turbulence in an hour. Gusts indicate the level of turbulence as such they could be lower than regular wind speeds.

# 4.0 Analysis Approach

**Step1:** Data Exploration to review the individual household electricity consumption dataset, treat missing values and outliers, check for correlations. Perform same treatments for the weather download for Sceax, France. Aggregate the data to the same level of resolution (hourly averages) and combine them. Daily, Weekly and Monthly versions of the electricity consumption data are also generated for trend identification and coarse time series analyses.

**Step 2:** Transform and Normalize the numerical attributes, set levels for the categorical variables. Decide on the response variable total consumption global active power or sub-meter 1, sub-meter 2 or sub-meter 3. Attempt prediction with time series models. Run Decision Tree with non Linear methods e.g. SVM, Logistics Regression and possibly Neural Networks (Multilayer Perceptrons). These will be attempted in R (base and packages) as well as WEKA. The results and accuracy of each of these models will be compared robustly by using WEKA experiments.

**Step 3:** Carryout Predictions based on the 3 methods and determine the accuracy of each

## 4.1 Explore Clean Correlate Aggregate and Combine

.

### 4.1.1 Explore, Clean & Check for Correlations

We will explore clean and check the dataset for missing values and outliers in order to reduce their effects on the model results.

#### 4.1.1.1 Read Dataset and Explore

hholdpcon <- read.csv("D:/BigDataAnalytics/Capstone/household\_power\_consumption/hholdpcon.txt", header= TRUE, sep=";")

View(hholdpcon)

summary(hholdpcon)

str(hholdpcon)

nrow(hholdpcon)



The following codes include header names on he dataset and then transforms them to numeric

hdrnames <- c("wdate", "wtime", "apower", "rpower", "voltage", "intensity", "smeter1", "smeter2", "smeter3")

colnames(hholdpcon) <- hdrnames

hholdpcon$wdate <- as.Date(hholdpcon$wdate, format = "%d/%m/%Y")

hholdpcon$apower <- as.numeric(as.character(hholdpcon$apower))

hholdpcon$rpower <- as.numeric(as.character(hholdpcon$rpower))

hholdpcon$voltage <- as.numeric(as.character(hholdpcon$voltage))

hholdpcon$intensity <- as.numeric(as.character(hholdpcon$intensity))

hholdpcon$smeter1 <- as.numeric(as.character(hholdpcon$smeter1))

hholdpcon$smeter2 <- as.numeric(as.character(hholdpcon$smeter2))

hholdpcon$smeter3 <- as.numeric(as.character(hholdpcon$smeter3))

# install.packages("lubridate")

# library(lubridate)

hholdpcon$wtime<- hms(hholdpcon$wtime)

### 4.1.2 Treat Outliers, Missing values

The donors of the dataset reported 1.25% missing values. As the dataset is huge, with 2 million instances, we would seek to remove these missing values as means of cleaning them. The effects of the missing values (and outliers) are also significantly reduced by the aggregation carried out subsequently. Moving from minute data to hourly takes the average over 60 values for each grouping. From hourly to daily is 24 and daily to monthly is 30.

str(hholdpcon)

table(is.na(hholdpcon)) # How many not missing(False) and missing (True)?

'data.frame': 2075259 obs. of 9 variables:

$ wdate : Date, format: "2006-12-16" "2006-12-16" "2006-12-16" "2006-12-16" ...

$ wtime : Factor w/ 1440 levels "00:00:00","00:01:00",..: 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 ...

$ apower : num 4.22 5.36 5.37 5.39 3.67 ...

$ rpower : num 0.418 0.436 0.498 0.502 0.528 0.522 0.52 0.52 0.51 0.51 ...

$ voltage : num 235 234 233 234 236 ...

$ intensity: num 18.4 23 23 23 15.8 15 15.8 15.8 15.8 15.8 ...

$ smeter1 : num 0 0 0 0 0 0 0 0 0 0 ...

$ smeter2 : num 1 1 2 1 1 2 1 1 1 2 ...

$ smeter3 : num 17 16 17 17 17 17 17 17 17 16 ...

FALSE TRUE

18495478 181853 #missing value information

#removes all the missing cases

hholdclean2 <- hholdpcon[rowSums(is.na(hholdpcon[ , 1:10])) == 0, ]

table(is.na(hholdclean2)) # False ???, True ???. There are ????? missing data

FALSE

19637890

The following code calculates the total missing value per attribute. And the correlation with missing values ignored.

sapply(hholdpcon, function(x) sum(is.na(x))) # Breakdown below

wdate wtime apower rpower voltage intensity smeter1 smeter2 smeter3

0 0 25979 25979 25979 25979 25979 25979 25979

cor(hholdpcon[, 3:9 ], method = "pearson", use = "complete.obs")

# correlation with missing values ignored

The purpose of this correlation run is to check and confirm strong relation between the attributes. Power is current x voltage, or intensity x voltage so this is reflected in the correlation results. We will eventually use active power or total power (including all the subs for the modelling).



### 4.1.3 Test for Outliers and Treat Them

We have for the first past sought to remove the outliers in the dataset. We also carried out preliminary analysis with outliers intact. A thirds pass would be done (not shown) with outliers treated with nearest neighbour.

#### 4.1.3.1 Relevant Code Outlier Test and Removal

head(hholdclean2)

hholdx <- hholdclean2[, c(3:9)] # dataset for mirror for outlier detection, excludes date & time. We have not determined the Class variable though we expect it to be some combination of active power and sub-meters

m\_dist <- mahalanobis(hholdx, colMeans(hholdx), cov(hholdx)) # mahalanobis function check help

#cmchMD <- 1 - pchisq(m\_dist, 6) # with dof set as 8Calculate the cumulative chi squared function for the 4521 MV variable 7 variables dof is 6

hholdx$MD <- cmchMD

hholdx$outlier <- "No"

hholdx$outlier[hholdx$MD <0.01] <- "Yes" # Threshold set to 99% Confidence

hholdclean2$outlier <- hholdx$outlier

hholdT <- subset(hholdx, hholdx$outlier == "No")

hholdclean22 <- subset(hholdclean2, hholdclean2$outlier == "No") # clean data with NA.RM with outliers as well.

nrow(hholdx) #2,049,280

nrow(hholdT) #1,888,036

nrow(hholdclean22)

# for electricity consumption data, meaning 161,244 were removed. Thus figure was arrived at after tuning the confidence interval to 99% and using a degree of freedoom of 8.

opar <- par(no.readonly=TRUE)

par(mfrow=c(2,2))

boxplot(hholdclean2$apower)$out

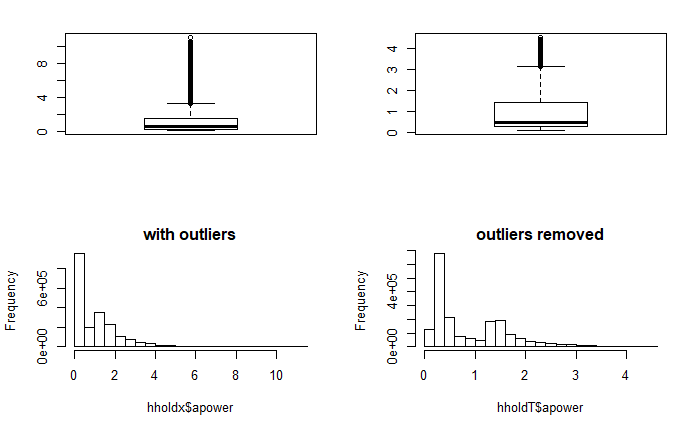
boxplot(hholdx$apower)$out

hist(hholdx$apower)

hist(hholdx$smeter3)

par(opar)

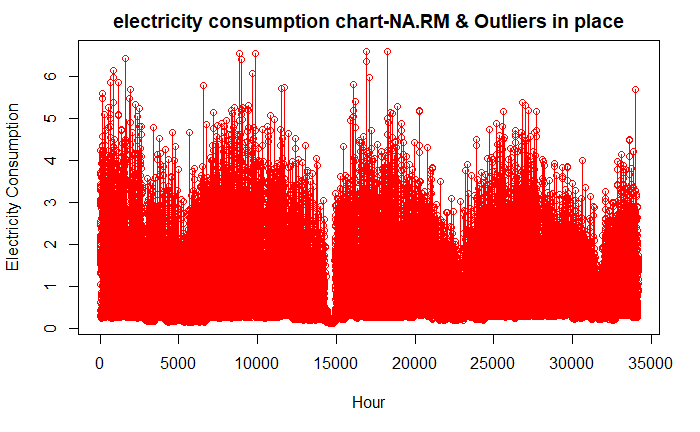
The plots that follow show the boxplots of active power before and after outliers were removed

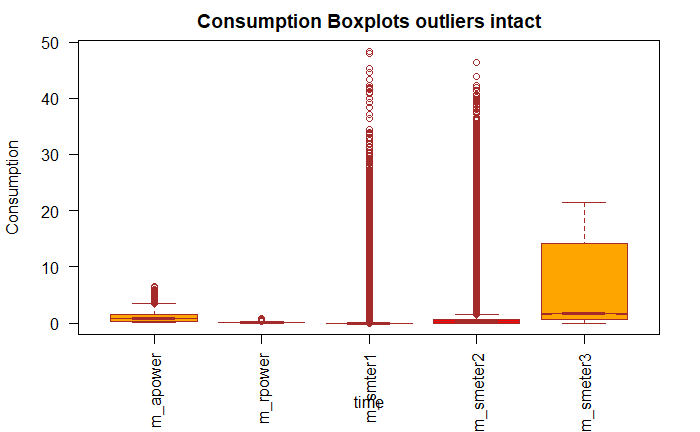


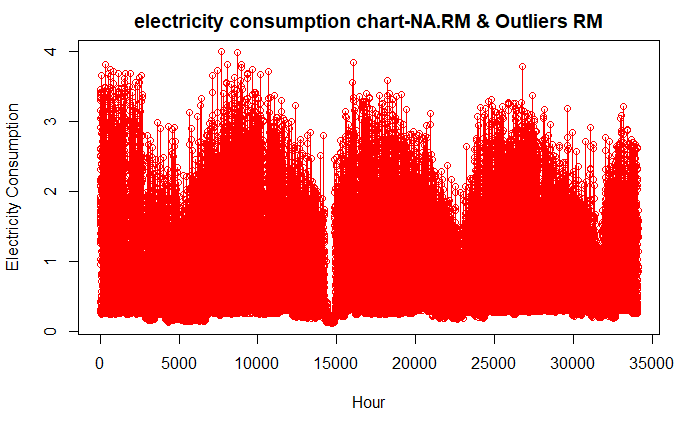
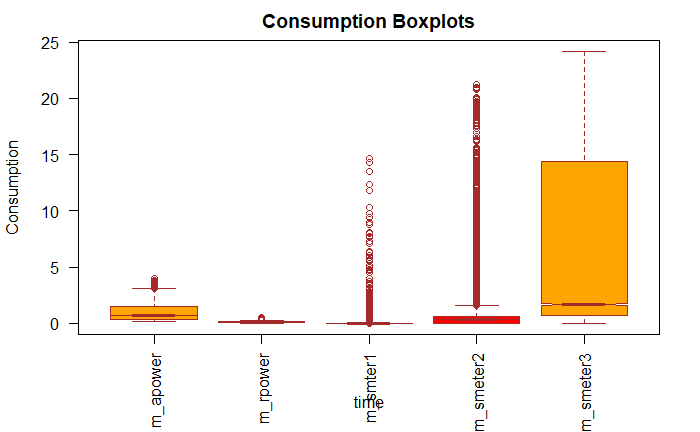
**Head(hholdclean2)**

| **wdate**  <date> | **wtime**  <S4: Period> | **apower**  <dbl> | **rpower**  <dbl> | **voltage**  <dbl> | **intensity**  <dbl> | **smeter1**  <dbl> | **smeter2**  <dbl> | **smeter3**  <dbl> | **outlier**  <chr> |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2006-12-16 | 17H 24M 0S | 4.216 | 0.418 | 234.84 | 18.4 | 0 | 1 | 17 | Yes |
| 2006-12-16 | 17H 25M 0S | 5.360 | 0.436 | 233.63 | 23.0 | 0 | 1 | 16 | Yes |
| 2006-12-16 | 17H 26M 0S | 5.374 | 0.498 | 233.29 | 23.0 | 0 | 2 | 17 | Yes |
| 2006-12-16 | 17H 27M 0S | 5.388 | 0.502 | 233.74 | 23.0 | 0 | 1 | 17 | Yes |
| 2006-12-16 | 17H 28M 0S | 3.666 | 0.528 | 235.68 | 15.8 | 0 | 1 | 17 | Yes |
| 2006-12-16 | 17H 29M 0S | 3.520 | 0.522 | 235.02 | 15.0 | 0 | 2 | 17 | Yes |

Maybe the dataset is better with outliers treated intead of removed





### 4.1.4 Aggregation and Combination of Datasets

Aggregation of household energy data would be done with the Data.table package. This is necessary in order be able to check the possible frequency orders in which the household is varying (weekly, hourly, daily, monthly).

Using the fast read function we read the household data into memory

hholdpcon <- fread("D:/BigDataAnalytics/Capstone/household\_power\_consumption/hholdpcon.txt", header= TRUE, sep=";")

install.packages(data.table)

library(data.table)

hholdpcon <- data.table(hholdpcon)

head(hholdpcon)

| **wdate**  <date> | **wtime**  <fctr> | **apower**  <dbl> | **rpower**  <dbl> | **voltage**  <dbl> | **intensity**  <dbl> | **smeter1**  <dbl> | **smeter2**  <dbl> | **smeter3**  <dbl> |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2006-12-16 | 17:24:00 | 4.216 | 0.418 | 234.84 | 18.4 | 0 | 1 | 17 |
| 2006-12-16 | 17:25:00 | 5.360 | 0.436 | 233.63 | 23.0 | 0 | 1 | 16 |
| 2006-12-16 | 17:26:00 | 5.374 | 0.498 | 233.29 | 23.0 | 0 | 2 | 17 |
| 2006-12-16 | 17:27:00 | 5.388 | 0.502 | 233.74 | 23.0 | 0 | 1 | 17 |
| 2006-12-16 | 17:28:00 | 3.666 | 0.528 | 235.68 | 15.8 | 0 | 1 | 17 |
| 2006-12-16 | 17:29:00 | 3.520 | 0.522 | 235.02 | 15.0 | 0 | 2 | 17 |

# The following code introduces hourly data attribute we will use for grouping

install.packages("lubridate")

library(lubridate)

library(chron)

hholdpcon$hrofday <- lubridate::hour(lubridate::hms((hholdpcon$wtime)))

head(hholdpcon))

| **wdate**  <date> | **wtime**  <S4: Period> | **apower**  <dbl> | **rpower**  <dbl> | **voltage**  <dbl> | **intensity**  <dbl> | **smeter1**  <dbl> | **smeter2**  <dbl> | **smeter3**  <dbl> | **hrofday**  <dbl> |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2006-12-16 | 17H 24M 0S | 4.216 | 0.418 | 234.84 | 18.4 | 0 | 1 | 17 | 17 |
| 2006-12-16 | 17H 25M 0S | 5.360 | 0.436 | 233.63 | 23.0 | 0 | 1 | 16 | 17 |
| 2006-12-16 | 17H 26M 0S | 5.374 | 0.498 | 233.29 | 23.0 | 0 | 2 | 17 | 17 |
| 2006-12-16 | 17H 27M 0S | 5.388 | 0.502 | 233.74 | 23.0 | 0 | 1 | 17 | 17 |
| 2006-12-16 | 17H 28M 0S | 3.666 | 0.528 | 235.68 | 15.8 | 0 | 1 | 17 | 17 |
| 2006-12-16 | 17H 29M 0S | 3.520 | 0.522 | 235.02 | 15.0 | 0 | 2 | 17 | 17 |

# The following codes generates the hourly averages of the original minutes data for each of the variables grouped by date and hour

hholdhrly <- hholdpcon[, list(m\_apower=mean(apower, na.rm = TRUE), m\_rpower=mean(rpower, na.rm = TRUE), m\_voltage= mean(voltage, na.rm = TRUE), m\_intensity=mean(intensity, na.rm = TRUE), m\_smeter1=mean(smeter1, na.rm = TRUE), m\_smeter2=mean(smeter2, na.rm = TRUE), m\_smeter3=mean(smeter3, na.rm = TRUE)), by="wdate,hrofday"] #data set is grouped by date,then hour

head(hholdhrly)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **wdate**  <date> | **hrofday**  <dbl> | **m\_apower**  <dbl> | **m\_rpower**  <dbl> | **m\_voltage**  <dbl> | **m\_intensity**  <dbl> | **m\_smeter1**  <dbl> | **m\_smeter2**  <dbl> | **m\_smeter3**  <dbl> |
| 2006-12-16 | 17 | 4.222889 | 0.22900000 | 234.6439 | 18.100000 | 0 | 0.5277778 | 16.861111 |
| 2006-12-16 | 18 | 3.632200 | 0.08003333 | 234.5802 | 15.600000 | 0 | 6.7166667 | 16.866667 |
| 2006-12-16 | 19 | 3.400233 | 0.08523333 | 233.2325 | 14.503333 | 0 | 1.4333333 | 16.683333 |
| 2006-12-16 | 20 | 3.268567 | 0.07510000 | 234.0715 | 13.916667 | 0 | 0.0000000 | 16.783333 |
| 2006-12-16 | 21 | 3.056467 | 0.07666667 | 237.1587 | 13.046667 | 0 | 0.4166667 | 17.216667 |
| 2006-12-16 | 22 | 2.200133 | 0.05616667 | 238.7600 | 9.523333 | 0 | 0.1333333 | 4.433333 |

| **wdate**  <date> | **hrofday**  <dbl> | **m\_apower**  <dbl> | **m\_rpower**  <dbl> | **m\_voltage**  <dbl> | **m\_intensity**  <dbl> | **m\_smeter1**  <dbl> | **m\_smeter2**  <dbl> | **m\_smeter3**  <dbl> |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2010-11-26 | 16 | 1.0679333 | 0.21563333 | 240.4578 | 4.610000 | 0 | 0.95000000 | 0.00000 |
| 2010-11-26 | 17 | 1.7259000 | 0.06140000 | 237.0697 | 7.216667 | 0 | 0.00000000 | 12.86667 |
| 2010-11-26 | 18 | 1.5734667 | 0.05370000 | 237.5318 | 6.620000 | 0 | 0.00000000 | 0.00000 |
| 2010-11-26 | 19 | 1.6593333 | 0.06003333 | 236.7410 | 7.056667 | 0 | 0.06666667 | 0.00000 |
| 2010-11-26 | 20 | 1.1637000 | 0.06116667 | 239.3960 | 4.913333 | 0 | 1.06666667 | 0.00000 |
| 2010-11-26 | 21 | 0.9346667 | 0.00000000 | 239.6900 | 3.800000 | 0 | 0.00000000 | 0.00000 |

This summarized dataset **(hholdhrly)** has 34,589 observations of 9 variables. In The process of summarizing, the missing values were removed by setting NA.RM = TRUE for all attributes as the codes above demonstrates. The averaging process from minutes to hours also reduces the effect of the outliers

### 4.1.5 Total Power Consumption

The total power consumption in the household include the energy in KWh expended in the sub-meters We will need to calculate this as it is likely to be our response variable.

hholdhrly$power\_subs=hholdhrly$m\_smeter1+hholdhrly$m\_smeter2+hholdhrly$m\_smeter3

hholdhrly$power\_kwh = 0.001\*hholdhrly$power\_subs + hholdhrly$m\_apower

vv <- hholdhrly$power\_kwh

tt <- hholdhrly$m\_apower

plot(vv,type = "o",col = "red", xlab = "Hour", ylab = "Electricity Consumption",

main = "Total Electricity Consumption chart - NA.RM =TRUE")

plot(tt,type = "o",col = "red", xlab = "Hour", ylab = "Electricity Consumption",

main = "Electricity Consumption chart - NA.RM =TRUE")

attach(hholdhrly)

boxplot(m\_apower, m\_rpower, m\_smeter1, m\_smeter2, m\_smeter3, data=hholdhrly, main="Consumption Boxplots",

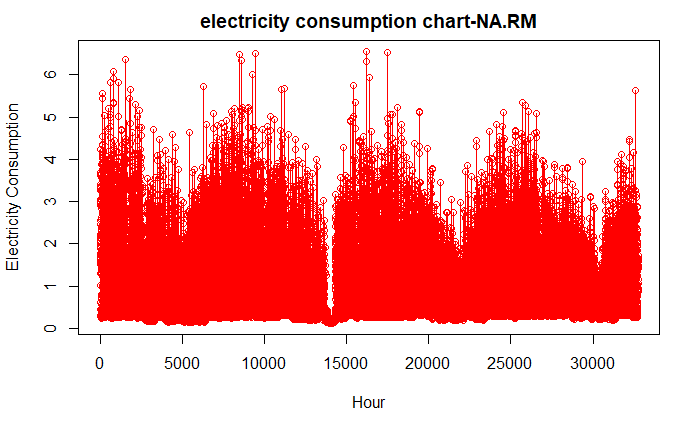
xlab="time", ylab="Consumption",

names = c("m\_apower", "m\_rpower", "m\_smter1", "m\_smeter2", "m\_smeter3"),

las = 2, col = c("orange","red"), border = "brown", horizontal = FALSE, notch = TRUE

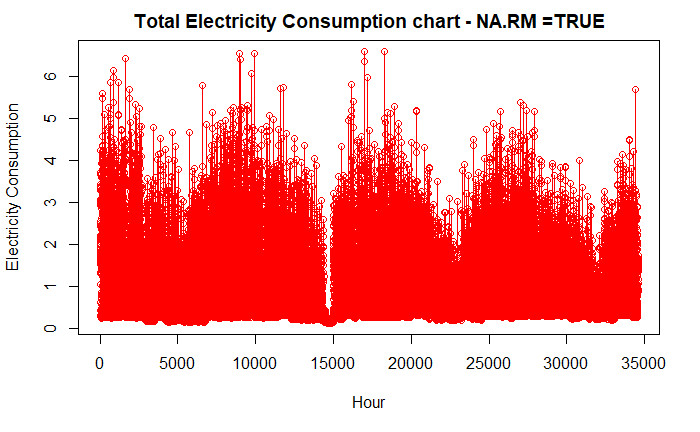
)

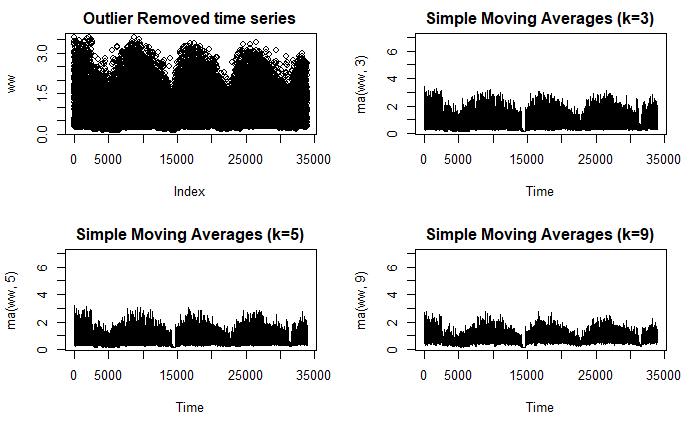
detach(hholdhrly)

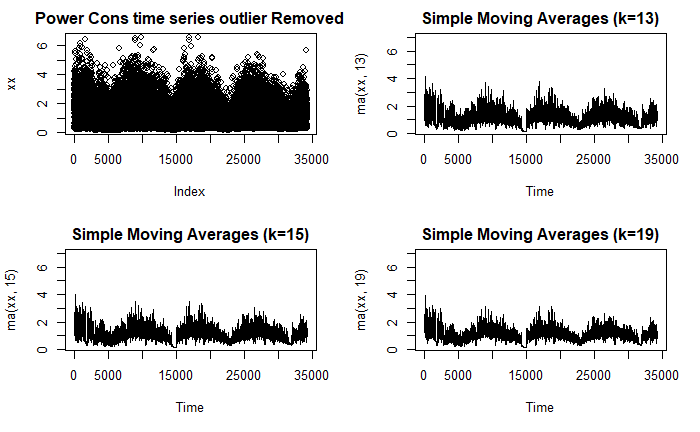


Both plots overall power consumption and the active power without the subs have strong time series components. Thus one approach for predicting this power consumption is to consider it a time series and then using one of the available R package to analysis it. Another option is to deter the factors driving the behavior and use any of the algorithms good for non-linear data to handle it.

Clear cased of outliers







# 4.2 Weather dataset

### 4.2.1 Source of Weather Data

The weather dataset for Sceax is downloaded for the same 47month period we now have household electricity data. The boundaries of Sceax weather data we are interested in is given by 17H of 16 December 2006 TO 21H of 26Nov 2010. This data was sourced from the metroblue website after the NOAA one proved to have far too many missing values.

Finally, we will combine this with the hourly dataset of weather which we describe below.

#### 4.2.1.1 Read the weather data into R

scxwedar <- read.csv("D:/BigDataAnalytics/Capstone/scxwedar.csv", header= TRUE, sep=";")

View(scxwedar)

head(scxwedar)

#Reassign header names

hdrnames2 <- c("Year", "Month", "Day", "Hour", "Minute", "Temp", "Humd", "Pressure", "Precipt", "Snowfall", "CCloudT", "CCloudH","CCloudM", "CCloudL", "SunShD", "Radiatn", "WndSpeed", "WndDir", "WndSpeed80m", "WndDir80m", "WndSpeed900mb", "WndDir900mb", "WndGusts")

colnames(scxwedar) <- hdrnames2

head(scxwedar)

install.packages("lubridate")

library(lubridate)

#Merge the Date data in the same format as consumption data

scxwedar$Date <- as.Date(with(scxwedar, paste(Year, Month, Day, sep="-")),

"%Y-%m-%d ")

scxwedar <- scxwedar[, c(24, 4:23)]

head(scxwedar)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| #Shown truncated | | | | | | | | | | | |
|  |
| **Date**  <date> | | **Hour**  <int> | | **Temp**  <dbl> | | **Humd**  <dbl> | | **Pressure**  <dbl> | | | **Precipt**  <dbl> | | **Snowfall**  <dbl> | | **CCloudT**  <dbl> | | **CCloudH**  <dbl> | |  |
|  | 2006-12-01 | | 0 | | 6.76 | | 95 | | 1025.8 | 0 | | 0 | 15.9 | 53 | |  | |
|  | 2006-12-01 | | 1 | | 5.70 | | 97 | | 1024.9 | 0 | | 0 | 15.0 | 50 | |  | |
|  | 2006-12-01 | | 2 | | 4.99 | | 98 | | 1025.7 | 0 | | 0 | 26.0 | 37 | |  | |
|  | 2006-12-01 | | 3 | | 5.28 | | 97 | | 1025.8 | 0 | | 0 | 100.0 | 36 | |  | |
|  | 2006-12-01 | | 4 | | 5.45 | | 96 | | 1026.1 | 0 | | 0 | 100.0 | 30 | |  | |
|  | 2006-12-01 | | 5 | | 5.41 | | 96 | | 1026.6 | 0 | | 0 | 100.0 | 28 | |  | |

### 4.2.2 COMBINATION of DATASET

# To combine the two datasets we will first align both 17H of 16 December 2006 TO 21H of 26Nov 2010

#### 4.2.2.1 Relevant Code for Aligning Period of Datasets

scxwedar1 <- scxwedar[scxwedar$Date >= as.Date("2006-12-16"), ]

scxwedar1 <- scxwedar1[scxwedar1$Date <= as.Date("2010-11-26"), ]

head(scxwedar1)

tail(scxwedar1)

scxwedar2 <- scxwedar1[-c(1:17),]

scxwedar3 <- scxwedar2[-c(34590:34591),]

head(scxwedar3)

tail(scxwedar3)

summary(scxwedar3)

# With these the two dataset are fully aligned



### 4.2.3 MERGE of TWO DATASETS

The dataset was merged on an hourly basis. The aim here is to have power consumption attributes next to the recorded weather attributes for the exact hour as downloaded from metroblue. Now the weather data was already downloaded in a hourly average format. To facilitate the merger the household energy consumption was rendered in an hourly format through aggregation. The date and hour attributes in both datasets were aligned in terms of their names and then merged. The code is shown below.

#### 4.2.3.1 Relevant Code for Merge of Datasets

head(hholdhrly22) #clean version of dataset with missing values and outliers removed

nrow(hholdhrly22) #33891

hdrnames3 <- c("Date", "Hour", "m\_apower", "m\_rpower", "m\_voltage", "m\_intensity", "m\_smeter1", "m\_smeter2", "m\_smeter3", "power\_subs", "power\_kwh")

colnames(hholdhrly22) <- hdrnames3

dstmdl <- merge(hholdhrly22,scxwedar3, by=c("Date","Hour"), all.x=TRUE, all.y=FALSE)

head(dstmdl)

# 33891 observations of 27 variables

summary(dstmdl)

cor(dstmdl[, 3:27 ], method = "pearson", use = "complete.obs")

library("corrgram")

corrgram(dstmdl[, 3:27 ], order=TRUE, lower.panel=panel.shade,

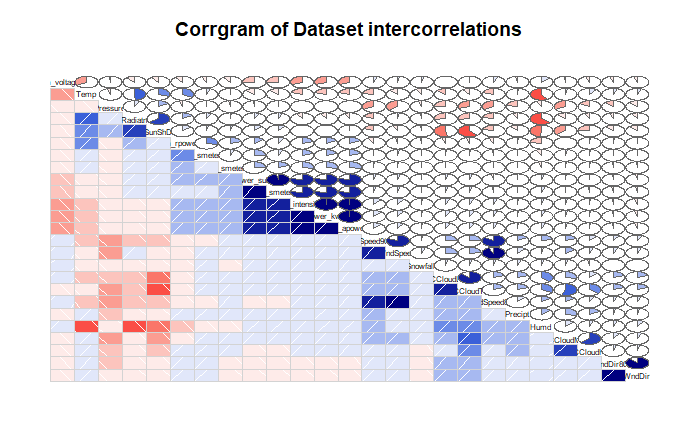
upper.panel=panel.pie, text.panel=panel.txt,

main="Corrgram of Dataset intercorrelations")

dstmdlx <- dstmdl[, c(1, 2, 3, 11, 12, 13, 14, 15, 16, 17, 21, 22, 23)]

head(dstmdlx)





### 4.2.4 Preliminary Variable Selection

A review of the weather data shows that we can drop the wind speed data other that those recorded at 2m. The wind speed at 80m is highly correlated with the ground level wind speed and so we choose those values at the regular height of a house. All the different versions of Windspeeds are highly correlated with each other (0.88, 0.91, 0.91).

Similarly, we can drop CCloudL cloud cover (low case below 4km) as it is highly correlated with CCloudT (total cloud cover, 0.85).

The total consumed electricity is the sum of active power and the submeter power readings in the same units. Given this, we can re-write the following expression for the total active power

Total electricity consumption in Kilowatt-Hour = active power + [smter1 + smeter2 +smeter3]/1000

And then replace those 4 attributes (mean active power and mean submeter 1/2/3 ) with the total active power.

Based on the forgoing the following features have been selected as a first pass. The reasons for selecting them are also shown. We will further rank these features using available tools as packages



Observation of the Corrgram plots indicates that active power is moderately correlated with temperature, voltage (expected), intensity (expected) and wind speed. The others are relatively weak. Other variables should be at play and we explore them using day of weak functions of lubridate.

### 4.2.5 Normalize and Transform Features using mean -max

The following transformations were considered

1. Date = extract day of the week (7 categories, Mon through Saturday)
2. Hrofday = 01-04 “sleepmorn”; 05-08 “risemorn”; 09-12 “workmorn”; 13-16 ‘workaft’; 17-20 ‘homeven’; 21-24 ‘sleepnigh’
3. Power-Kwh = cut levels 1, 2, 3
4. All other numeric variables = mean – max normalization

#### 4.2.5.4 Relevant Code for Regression with SVM

attach(dstmdlx2)

install.packages("e1071")

library(e1071)

#Import Library

require(e1071) #Contains the SVM

svtrain <- dstmdlx2[c(1:1400),]

svtest <- dstmdlx2[c(1500:3300),]

# there are various options associated with SVM training; like changing kernel, gamma and C value.

# create model

model <- svm(power\_kwh~Temp+Humd+Pressure+Precipt+SunShD+Snowfall+CCloudT+Radiatn+WndSpeed,data=svtrain,kernel='polynomial',gamma=0.2,cost=100)

#model <- svm(power\_kwh~Temp+Humd+Pressure+Precipt+SunShD,data=svtrain,kernel='polynomial',gamma=0.2,cost=100)

#model <- svm(power\_kwh~Temp+Precipt+SunShD,data=svtrain,kernel='polynomial',gamma=0.2,cost=100)

#Predict Output

preds <- predict(model,svtest)

table(preds)

plot(preds, main="SVM prediction plot. Indicates Missing Attributes")

#x <- subset(dstmdlx2, select = -power\_kwh)

#y <- power\_kwh

#model <- svm(x, y)

#print(model)

#summary(model)

# test with train data

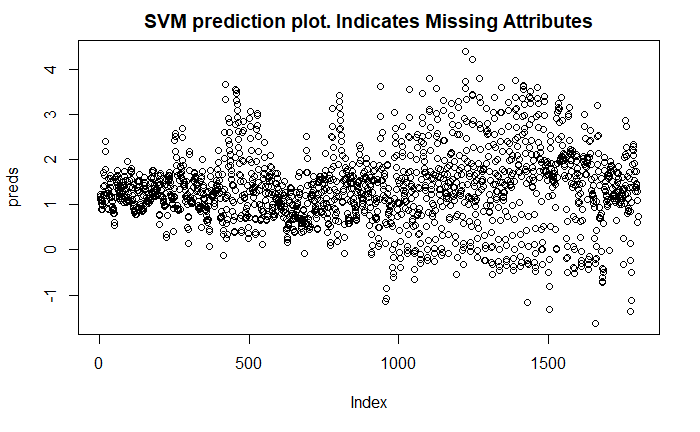
#pred <- predict(model, x)

# (same as:)

#pred <- fitted(model)

# Check accuracy:

#table(pred, y)



Clearly the model is not looking like the original data. We are missing the time related information in the variables we are predicting with.

# 5.0 Analysis Results

## 5.1 Daily Weekly and Monthly Time Series

### 5.1.1 Time Series Datasets

In order to be able to glean the frequencies active in the dataset, the monthly, weekly and daily averages were computed through aggregation so their time series can be viewed.

#### 5.1.1.1 Relevant Code for FOR AGGREGATION

head(hholdhrly2) #dataset with all all missing values removed

tail(hholdhrly2)

head(hholdhrly22) #dataset with all all missing values removed and outliers removed with Confidence Intv 99% abd DOF 6

tail(hholdhrly22)

library(data.table)

hholdTHR <- hholdhrly22 # mirror data without NA and Outliers

head(hholdTHR)

tail(hholdTHR)

# data for AGGREGATation

hholdTM <- hholdTHR

nrow((hholdTM)) # base for aggregate monthly data

head(hholdTM)

#hholdTM$monthabb <- dstmdlx3$monthabb

#hholdTM$dayofwk <- dstmdlx3$dayofwk

library(data.table)

library(zoo)

hholdTM$yearmon <- as.yearmon(hholdTM$wdate, "%m/%Y")

hholdTM$wkofyr <-lubridate::week(ymd(hholdTM$wdate))

hholdTM$yr <- strftime(ymd(hholdTM$wdate),'%Y')

head(hholdTM)

hholdTM <- as.data.table(hholdTM)

setkey(hholdTM, yr, yearmon, wkofyr, wdate)

# aggregate data by month, target is active power for time series

hholdTMLY <- as.data.frame(hholdTM[, j=list(active\_power= mean(m\_apower, na.rm = TRUE), react\_power = mean(m\_rpower, na.rm = TRUE)), by = list(yearmon)])

# aggregate data by WEEK, target is active power for time series

hholdWKLY <- as.data.frame(hholdTM[, j=list(W\_active\_power= mean(m\_apower, na.rm = TRUE), W\_react\_power = mean(m\_rpower, na.rm = TRUE)), by = list(yr, wkofyr)])

#data set is grouped by year ,then week of year

# aggregate data by DAY, target is active power for time series

hholdDAY <- as.data.frame(hholdTM[, j=list(D\_active\_power= mean(m\_apower, na.rm = TRUE), D\_react\_power = mean(m\_rpower, na.rm = TRUE)), by = list(wdate)])

#data set is grouped by year ,then week of year by = c('Date'))] #data set is grouped by Days dat

head(hholdTMLY)

head(hholdDAY)

head(hholdWKLY)

xx <- hholdTM$m\_apower

XMX <- hholdTMLY$active\_power

XDX <- hholdDAY$D\_active\_power

XWX <- hholdWKLY$W\_active\_power

par(mfrow=c(2,2))

ylim1 <- c(0.2, 3.0)

ylim2 <- c(0.2, 2.0)

plot(xx, main="Raw time series")

plot(XMX, main="Monthly Time Series Active Power)", ylim=ylim2,ylab="active power monthly avg", xlab= "month")

plot(XWX, main="Weekly Time Series Active Power", ylim=ylim2,ylab="active power weekly avg", xlab= "Week")

plot(XDX, main="Daily Time Series Active Power", ylim=ylim2, ylab="active power daily avg", xlab= "Day")

par(opar)

**MONTHLY TIME SERIES**

|  |
| --- |
|  |
|  | **yearmon**  <S3: yearmon> | **active\_power**  <dbl> | **react\_power**  <dbl> |  |
| 1 | Dec 2006 | 1.5617903 | 0.1258460 |  |
| 2 | Jan 2007 | 1.3245791 | 0.1278388 |  |
| 3 | Feb 2007 | 1.1990486 | 0.1101510 |  |
| 4 | Mar 2007 | 1.1026842 | 0.1094891 |  |
| 5 | Apr 2007 | 0.7761963 | 0.1162860 |  |
| 6 | May 2007 | 0.7953392 | 0.1114508 |  |

**WEEKLY TIMES SERIES**

|  | **yr**  <chr> | **wkofyr**  <dbl> | **W\_active\_power**  <dbl> | | **W\_react\_power**  <dbl> |
| --- | --- | --- | --- | --- | --- |
| 1 | 2006 | 50 | 2.931724 | 0.07665129 |
| 2 | 2006 | 51 | 1.480300 | 0.11821993 |
| 3 | 2006 | 52 | 1.505899 | 0.14026180 |
| 4 | 2006 | 53 | 2.164834 | 0.09227120 |
| 5 | 2007 | 1 | 1.345077 | 0.13039343 |
| 6 | 2007 | 2 | 1.349758 | 0.15098255 |

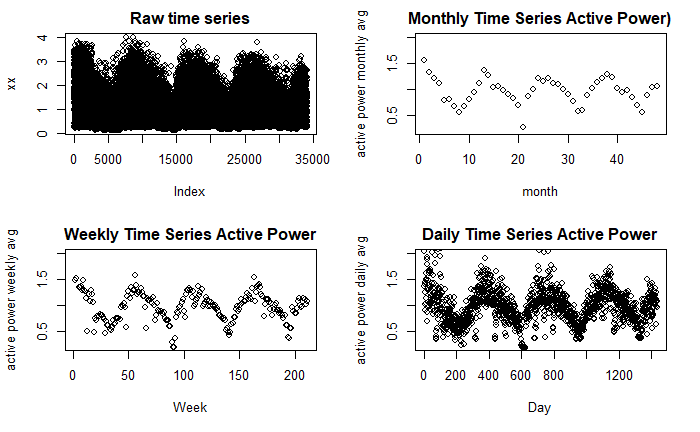
6 rows

**DAILY TIME SERIES**

|  | **wdate**  <date> | **D\_active\_power**  <dbl> | **D\_react\_power**  <dbl> |  |
| --- | --- | --- | --- | --- |
| 1 | 2006-12-16 | 2.931724 | 0.07665129 |  |
| 2 | 2006-12-17 | 2.036551 | 0.15010777 |  |
| 3 | 2006-12-18 | 1.359290 | 0.10398317 |  |
| 4 | 2006-12-19 | 0.898462 | 0.09821254 |  |
| 5 | 2006-12-20 | 1.395706 | 0.11162085 |  |
| 6 | 2006-12-21 | 1.017754 | 0.09052967 |  |

From the plots show below, it is clear that power consumption is broadly following a pattern described as varying almost like a sine wave with peaks in the winter and troughs in the summer months. This results is at variance with some of the other European studies that are beginning to see peaks in summer months. Another feature observed is that the difference between the hourly peaks (around 4kwh) and the troughs (around 2Kwh) is almost a factor of 2 i.e. 100% less. The range is even broader for monthly times series varying from around 0.5kwh for monthly average to 1.5Kwh peak (during Dec -Jan winter months). There is a lot of data overlaps which appears as noise on condensed plots but are actual datasets if we blow out the chats to hourly variations as we will.

The value of these monthly, weekly and hourly averages is that it “normalizes” of averages out the differences in the daily use patterns of the occupants. Thus, if shift changes at work if dish washer time changes during the day if laundry day changes these features have limited effect on the aggregated data. It also means that the aggregated data has “cycles” whereas the hourly data does not.



### 5.1.2 Prepare Dataset for Modeling Against External Attributes

In order to model power consumption we need to prepare them for modelling. Some of the models can perform well without normalization e.g. Support Vector Machine. Others like regression and ANN require normalised datasets.

#### 5.1.2.1 Relevant Code for Normalization of Dataset

dstmdlx3 <- dstmdlx

head(dstmdlx3)

normalize <- function(x) {

return ((x - min(x)) / (max(x) - min(x))) }

#Next we use the lapply function to efficiently normalize the entire dataframe

dstmdlx3 <- as.data.frame(lapply(dstmdlx3[ , 5:13], normalize))

dstmdlx3$power\_kwh <- dstmdlx$power\_kwh

dstmdlx3$m\_apower <- dstmdlx$m\_apower

dstmdlx3$wdate <- dstmdlx$wdate

dstmdlx3$hrofday <- dstmdlx$hrofday

#transform the potential class variables to categorical

dstmdlx3$cm\_apower <- cut(dstmdlx3$m\_apower, 4, labels=c('Low','Medium','High', 'VeryHigh'))

dstmdlx3$cpower\_kwh <- cut(dstmdlx3$power\_kwh, 4, labels=c('Low','Medium','High', 'VeryHigh'))

with(dstmdlx3, table(cm\_apower, cpower\_kwh))

cpower\_kwh

cm\_apower Low Medium High VeryHigh

Low 20474 33 0 0

Medium 12 9964 48 0

High 0 6 2867 2

VeryHigh 0 0 2 483

#these will offer numeric values

dstmdlx3$monthofyr <- month((dstmdlx3$Date))

dstmdlx3$dayofwk <- wday(dstmdlx3$Date)

# These will offer ordered factors

dstmdlx3$monthabb <- with(dstmdlx3, month.abb[dstmdlx3$monthofyr])

dstmdlx3$dayofwk <- strftime(dstmdlx3$Date,'%a')

dstmdlx3$monthabb <- factor(dstmdlx3$monthabb, ordered = TRUE,

levels = c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"))

dstmdlx3$dayofwk <- factor(dstmdlx3$dayofwk, ordered = TRUE,

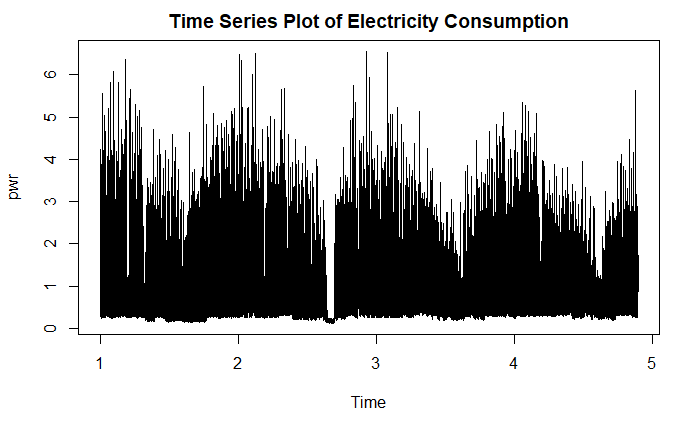
levels = c('Sun','Mon','Tue','Wed','Thur', 'Fri','Sat'))

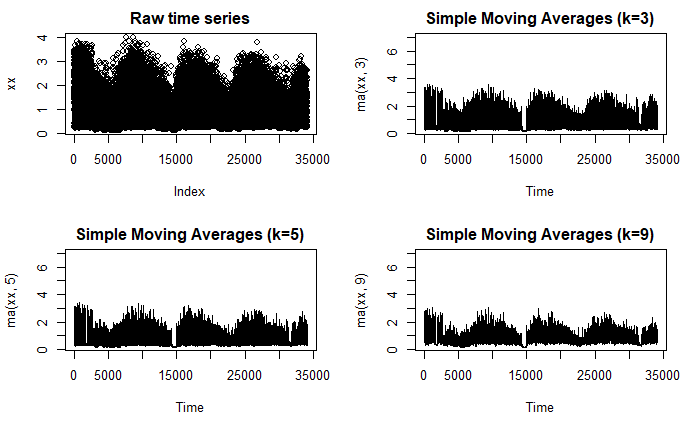
head(dstmdlx3)

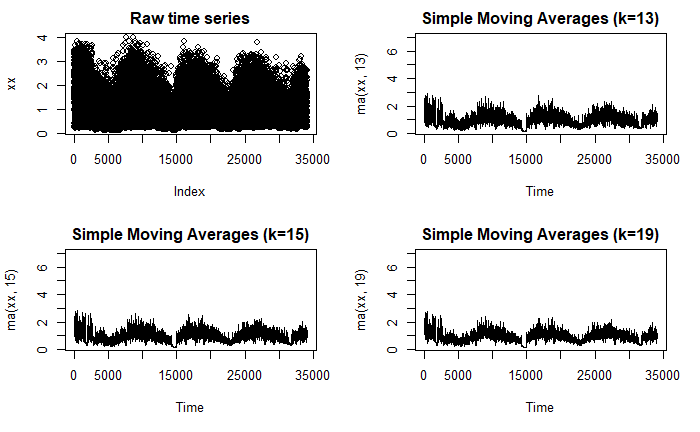
str(dstmdlx3)

write.csv(dstmdlx3, file = "D:/BigDataAnalytics/Capstone/dstmdlx3.csv",row.names=FALSE)

The missing values are clear near the middle of the plot







#### 5.1.2.2 Relevant Code for Moving Average Smoothening

install.packages("forecast")

library(forecast)

#msts modelling of time series power consumption using dual seasonal data, Daily(24hrs) and Annual (8766hours)

xx <- hholdTM$m\_apower

XMX <- hholdTMLY$active\_power # monthly aggregate

XDX <- hholdDAY$D\_active\_power # daily aggregate

XWX <- hholdWKLY$W\_active\_power # weekly aggregate

pwr <- msts(xx, start=c(2006-12-16), seasonal.periods = 8760)

pwr.fit <- tbats(pwr)

plot(forecast(pwr.fit))

plot(pwr, main="Time Series Plot of Electricity Consumption")

# Moving Average Smoothening with periods 3, 5, 9

library(forecast)

opar <- par(no.readonly=TRUE)

par(mfrow=c(2,2))

ylim <- c(0.2, 7.0)

plot(xx, main="Raw time series")

plot(ma(xx, 3), main="Simple Moving Averages (k=3)", ylim=ylim)

plot(ma(xx, 5), main="Simple Moving Averages (k=5)", ylim=ylim)

plot(ma(xx, 9), main="Simple Moving Averages (k=9)", ylim=ylim)

par(opar)

# Moving Average Smoothening with periods 13, 15, 99

library(forecast)

opar <- par(no.readonly=TRUE)

par(mfrow=c(2,2))

ylim <- c(0.2, 7.0)

plot(xx, main="Raw time series")

plot(ma(xx, 13), main="Simple Moving Averages (k=13)", ylim=ylim)

plot(ma(xx, 15), main="Simple Moving Averages (k=15)", ylim=ylim)

plot(ma(xx, 19), main="Simple Moving Averages (k=19)", ylim=ylim)

par(opar)

#Exponential smoothening

yy <- hholdhrly2$m\_apower

xx <- hholdhrly$power\_kwh

require(xts)

require(forecast)

hholdhrly2<-as.data.frame(hholdhrly2)

yy <- hholdhrly2$m\_apower

xx <- hholdhrly2$power\_kwh

library(lubridate)

hholdhrly2$ddhrd <- ymd\_hms(hholdhrly2$wdate)

head(hholdhrly2)

require(xts)

require(forecast)

pwrdata <- xts(xx, order.by = hholdhrly2$ddhrd)

ets(pwrdata)

ETS(M,A,N)

Call:

ets(y = pwrdata)

Smoothing parameters:

alpha = 0.9622

beta = 1e-04

Initial states:

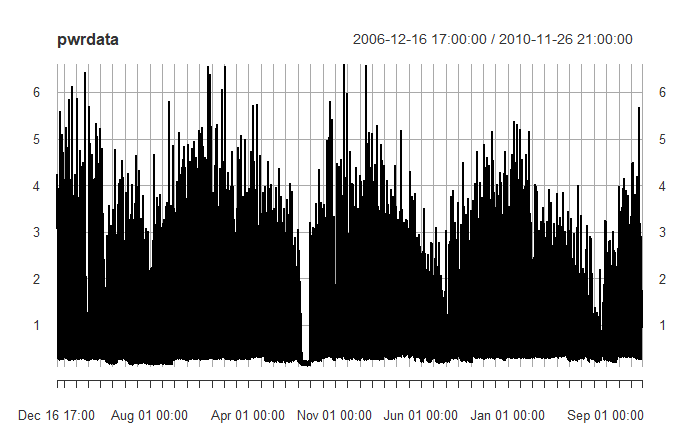
l = 3.0913

b = 0.3924

sigma: 0.7474

AIC AICc BIC

329862.2 329862.2 329904.4

plot(pwrdata)

M,A,N M means errors are multiplicative, A means trend type Addictive and N means there is no seasonality (very strange)

## 5.2 Artificial Neural Nets -Dataset for daily weekly and monthly resolution

### 5.2.1 Prepare ANN Datasets

In order to perform ANN models, we need to render the power consumption data in several levels of resolution to make it amenable for prediction. The data is first rendered in monthly average format. The average monthly figures is a good starting place fo8r prediction of the similar month data. Due to the repeated nature of time series, if there is a monthly sequence, we expected to see some features of it in the monthly data.

#### 5.2.1.1 Relevant Code Render Power Datasets in daily, weekly and monthly resolution

mgdata <- dstmdlx

head(mgdata)

library(data.table)

library(zoo)

mgdata$yearmon <- as.yearmon(mgdata$Date, "%m/%Y")

mgdata$wkofyr <-lubridate::week(ymd(mgdata$Date))

mgdata$yr <- strftime(ymd(mgdata$Date),'%Y')

head(mgdata)

mgdata <- as.data.table(mgdata)

setkey(mgdata, yr, yearmon, wkofyr, Date)

# aggregate data by month, target is active power for time series

mgdata\_mnthly <- as.data.frame(mgdata[, j=list(apower\_mna= mean(m\_apower, na.rm = TRUE), powerkwh\_mna = mean(power\_kwh, na.rm = TRUE), Temp\_mna = mean(Temp, na.rm = TRUE), Humd\_mna = mean(Humd, na.rm = TRUE), Pressure\_mna = mean(Pressure, na.rm = TRUE), WndSpd\_mna = mean(WndSpeed, na.rm = TRUE)), by = list(yearmon)])

# aggregate data by WEEK, target is active power for time series

mgdata\_wkly <- as.data.frame(mgdata[, j=list(apower\_wka= mean(m\_apower, na.rm = TRUE), powerkwh\_wka = mean(power\_kwh, na.rm = TRUE), Temp\_wka = mean(Temp, na.rm = TRUE), Humd\_wka = mean(Humd, na.rm = TRUE), Pressure\_wka = mean(Pressure, na.rm = TRUE), WndSpd\_wka = mean(WndSpeed, na.rm = TRUE)), by = list(yr, wkofyr)])

#data set is grouped by year ,then week of year

# aggregate data by DAY, target is active power for time series

mgdata\_dly <- as.data.frame(mgdata[, j=list(apower\_dla= mean(m\_apower, na.rm = TRUE), powerkwh\_dla = mean(power\_kwh, na.rm = TRUE), Temp\_dla = mean(Temp, na.rm = TRUE), Humd\_dla = mean(Humd, na.rm = TRUE), Pressure\_dla = mean(Pressure, na.rm = TRUE), WndSpd\_dla = mean(WndSpeed, na.rm = TRUE)), by = list(Date)])

#data set is grouped by year ,then week of year by = c('Date'))] #data set is grouped by Days dat

head(mgdata\_mnthly)

head(mgdata\_wkly)

head(mgdata\_dly)

library(zoo)

plot(mgdata\_mnthly)

plot(mgdata\_wkly)

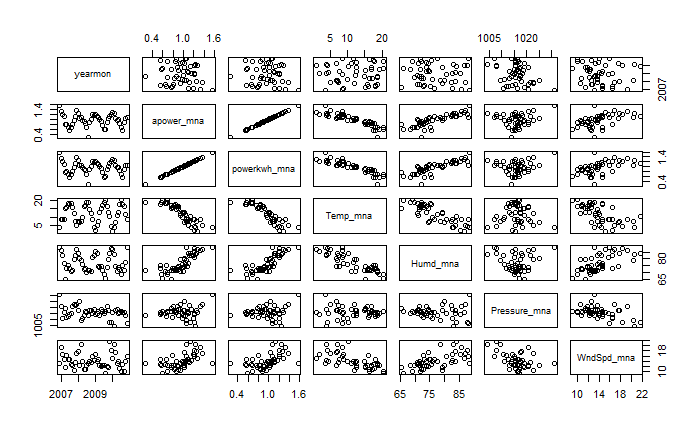
plot(mgdata\_dly)

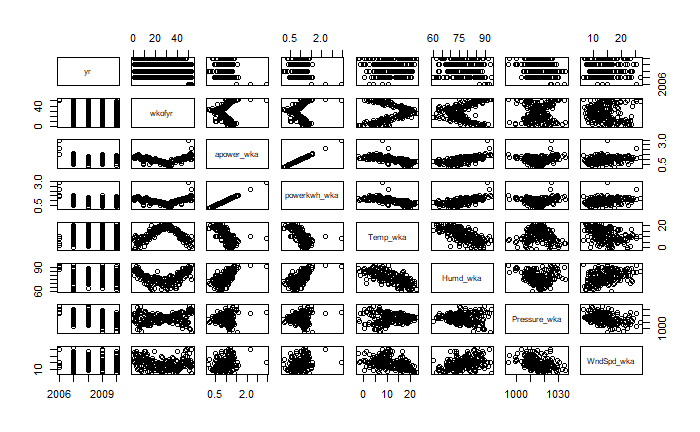
| **yearmon**  <S3: yearmon> | **apower\_mna**  <dbl> | **powerkwh\_mna**  <dbl> | | **Temp\_mna**  <dbl> | **Humd\_mna**  <dbl> | **Pressure\_mna**  <dbl> | **WndSpd\_mna**  <dbl> | **month\_abb**  <chr> |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dec 2006 | 1.5617903 | 1.5701595 | 3.700388 | | 88.51247 | 1031.286 | 13.12202 | Dec | |
| Jan 2007 | 1.3245791 | 1.3326584 | 8.410605 | | 85.83333 | 1020.698 | 19.26058 | Jan | |
| Feb 2007 | 1.1990486 | 1.2064259 | 8.795928 | | 82.66617 | 1010.651 | 16.48470 | Feb | |
| Mar 2007 | 1.1026842 | 1.1100946 | 8.853396 | | 76.57951 | 1016.134 | 14.70751 | Mar | |
| Apr 2007 | 0.7761963 | 0.7815646 | 15.106282 | | 65.31411 | 1019.280 | 11.49668 | Apr | |
| May 2007 | 0.7953392 | 0.8012562 | 15.967322 | | 72.92463 | 1011.865 | 16.38353 | May |

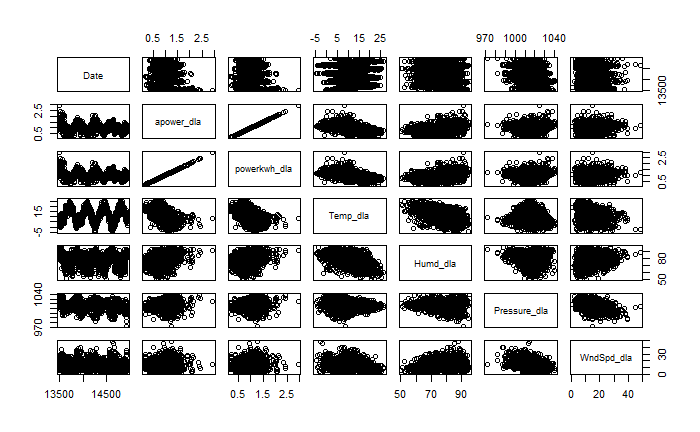
6 rows

| **yearmon**  <S3: yearmon> | **apower\_mna**  <dbl> | | **powerkwh\_mna**  <dbl> | | | **Temp\_mna**  <dbl> | | **Humd\_mna**  <dbl> | | **Pressure\_mna**  <dbl> | | **WndSpd\_mna**  <dbl> | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dec 2006 | | 1.5617903 | | 1.5701595 | 3.700388 | | 88.51247 | | 1031.286 | | 13.12202 | |
| Jan 2007 | | 1.3245791 | | 1.3326584 | 8.410605 | | 85.83333 | | 1020.698 | | 19.26058 | |
| Feb 2007 | | 1.1990486 | | 1.2064259 | 8.795928 | | 82.66617 | | 1010.651 | | 16.48470 | |
| Mar 2007 | | 1.1026842 | | 1.1100946 | 8.853396 | | 76.57951 | | 1016.134 | | 14.70751 | |
| Apr 2007 | | 0.7761963 | | 0.7815646 | 15.106282 | | 65.31411 | | 1019.280 | | 11.49668 | |
| May 2007 | | 0.7953392 | | 0.8012562 | 15.967322 | | 72.92463 | | 1011.865 | | 16.38353 | |
|  | |  | |  |  | |  | |  | |  | |

| **yr**  <chr> | **wkofyr**  <dbl> | **apower\_wka**  <dbl> | **powerkwh\_wka**  <dbl> | | **Temp\_wka**  <dbl> | **Humd\_wka**  <dbl> | **Pressure\_wka**  <dbl> | **WndSpd\_wka**  <dbl> |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2006 | 50 | 2.931724 | | 2.944936 | 8.052857 | 91.42857 | 1025.114 | 14.98143 |
| 2006 | 51 | 1.480300 | | 1.489508 | 4.066786 | 89.63690 | 1034.414 | 13.08458 |
| 2006 | 52 | 1.505899 | | 1.513649 | 2.227805 | 86.71951 | 1029.268 | 11.45963 |
| 2006 | 53 | 2.164834 | | 2.169873 | 10.495000 | 92.36364 | 1024.405 | 25.20864 |
| 2007 | 1 | 1.345077 | | 1.351109 | 9.603036 | 86.79762 | 1021.258 | 22.65060 |
| 2007 | 2 | 1.349758 | | 1.358546 | 10.814107 | 87.90476 | 1020.174 | 21.45119 |







## 5.3 SVM Model of Monthly Data

### 5.3.1 SVM Base Model with Time

SVM models are quite robust for non-linear data. We used the svm function of E1071 Package to carryout this model. The primary choice is that of which variables to model it against. We remained focus on the time, temperature, humidity and pressure and windspeed that we have found to have some variation around the year (not constant).

We modelled the monthly data using the first 365 days data from Jan 1, 2007 until the last day of the year. The model was done using the e1071 package and the results is as follows. The model with active power (daily average) W/b/RMSE values of 0.496/-0-471/0.284 as show below. The same model was repeated for powerkwh (the total power consumption with submeters added and the value of the W, b and RMSE were compared. They were exactly the same.

Next we conducted an optimization of the 2007 prediction using the tune R code and the Best parameters received were W/b/RMSE = //0.28537

#### 5.3.1.1 Relevant Code First Prediction based on days of the year

library(e1071)

#Scatter Plot

mgdata\_dly\_1$ndofyr <- mgdata\_dly\_1$dyofyr - 182

head(mgdata\_dly\_1)

mgdatad1 <- mgdata\_dly\_1[ , c(2, 9)]

plot(mgdatad1$apower\_dla ~ mgdatad1$ndofyr, main="Scatterplot overlaid with SVM monthly model Yr2007 with Jan1 = -181")

#Regression with SVM

mgdata1\_svm = svm(apower\_dla~ndofyr, mgdatad1)

#Predict using SVM regression

predYsvm\_1 = predict(mgdata1\_svm, mgdatad1)

predYsvm\_1

#Overlay SVM Predictions on Scatter Plot

points(mgdatad1$ndofyr, predYsvm\_1, col = "red", pch=16 )

##Calculate parameters of the SVR model

#Find value of W

W = t(mgdata1\_svm$coefs) %\*% mgdata1\_svm$SV # 0.4962209

#Find value of b

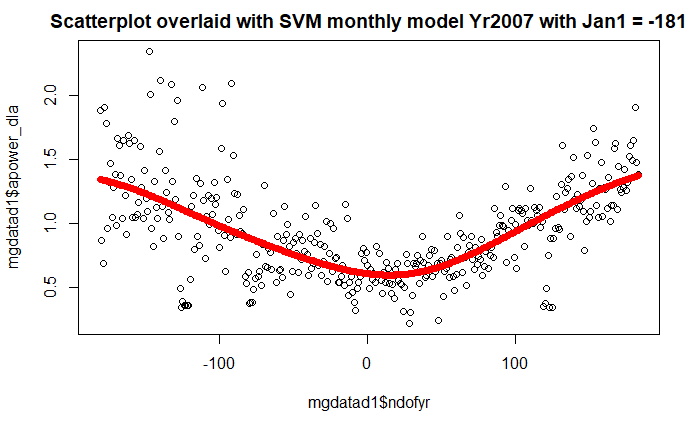
b = mgdata1\_svm$rho # -0.4714167

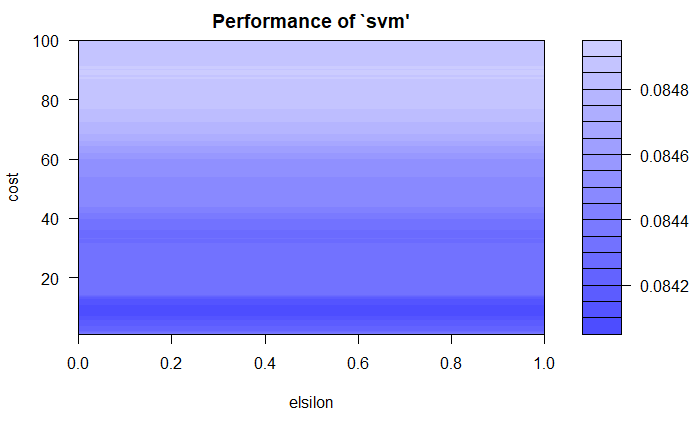
## RMSE for SVR Model

#Calculate RMSE

library(met)

RMSEsvm=rmse(predYsvm\_1,mgdatad1$apower\_dla) # 0.2844464



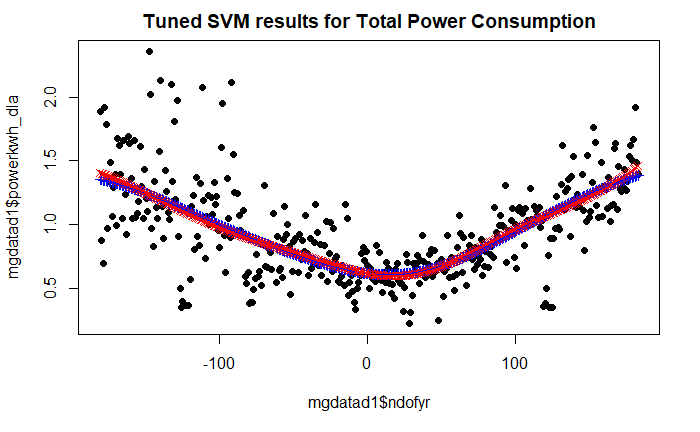


best parameters:

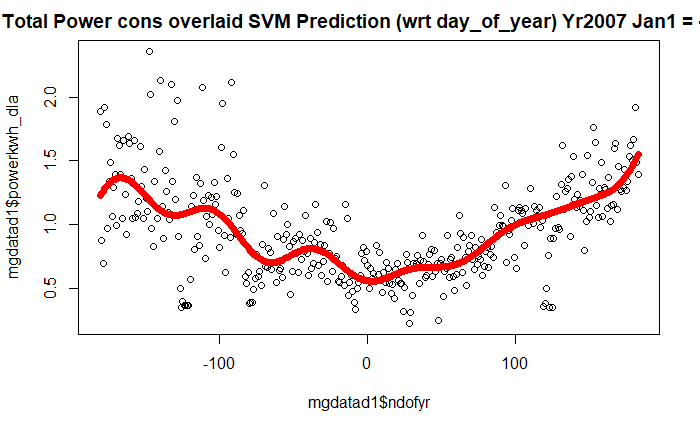
elsilon cost

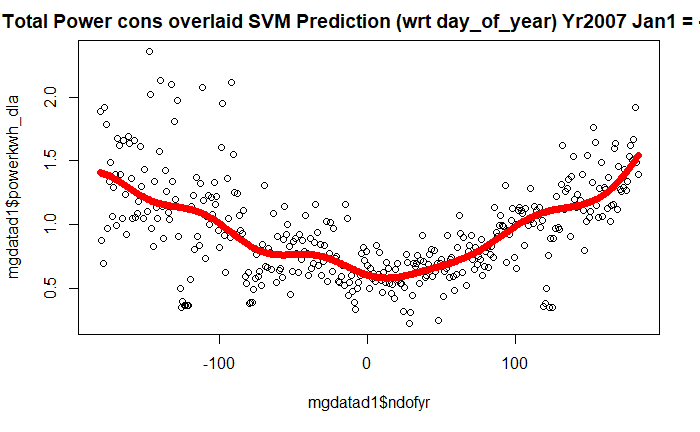
0 10

- best performance: 0.08408243



A similar plot this time of Total Power Consumption (with all the sub meters included at the same units) and using a gamma of 3 and cost of 300 vs gamma =2 and cost of 200 are shown below (latter is lower).





#### 5.3.1.2 Relevant Code SVM Model with Temperature Alone

head(svtrain)

plot(svtrain$powerkwh\_dla ~ svtrain$Temp\_dla, main="Total Power overlaid with SVM (wrt Temp) Daily Average")

model\_Tsvm <- svm(powerkwh\_dla~Temp\_dla,data=svtrain, kernel='radial',gamma=1,cost=300)

#Predict Output

predsT <- predict(model\_Tsvm,svtrain)

table(predsT)

points(svtrain$Temp\_dla, predsT, col = "red", pch=16 )

Tuning of Power Consumption Relationship with Day of the Year

We carried out a tuning of the Total power consumption using the codes inspired by XXX

As follows:

#### 5.3.1.3 Tune the SVM model

Optmgdata1\_svm=tune(svm, powerkwh\_dla~ndofyr, data=mgdatad1, ranges=list(elsilon=seq(0, 3, 0.5), cost=10:300))

#Print optimum value of parameters

print(Optmgdata1\_svm)

#Plot the perfrormance of SVM Regression model

plot(Optmgdata1\_svm)

## Select the best model out of 1100 trained models and compute RMSE

#Find out the best model

Bstmgdata\_Model=Optmgdata1\_svm$best.model

#Predict Y using best model

PredYBst1=predict(Bstmgdata\_Model,mgdatad1)

#Calculate RMSE of the best model

RMSEBst=rmse(PredYBst1,mgdatad1$powerkwh\_dla)

## Plotting SVR Model and Tuned Model in same plot

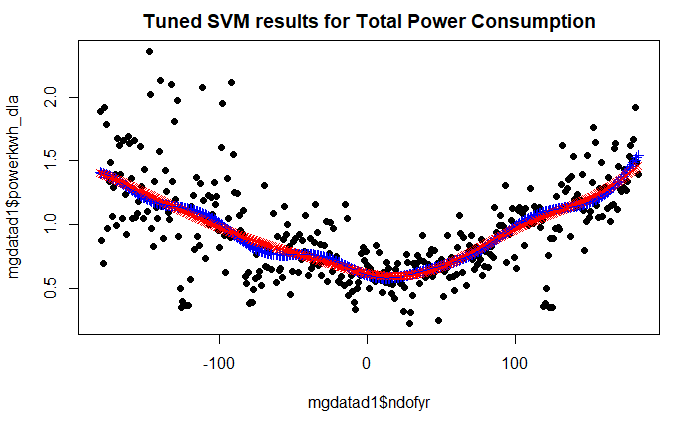
plot(mgdatad1$powerkwh\_dla~mgdatad1$ndofyr, pch=16,main= "Tuned SVM results for Total Power Consumption")

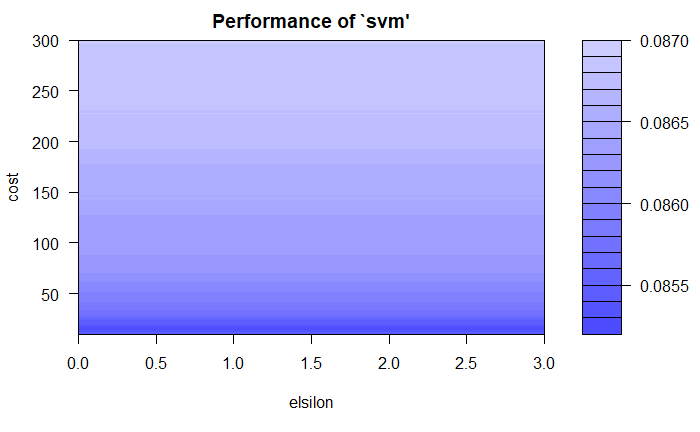
points(mgdatad1$ndofyr, predYsvm\_1, col = "blue", pch=3)

points(mgdatad1$ndofyr, PredYBst1, col = "red", pch=4)

points(mgdatad1$ndofyr, predYsvm\_1, col = "blue", pch=3, type="l")

points(mgdatad1$ndofyr, PredYBst1, col = "red", pch=4, type="l")





Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

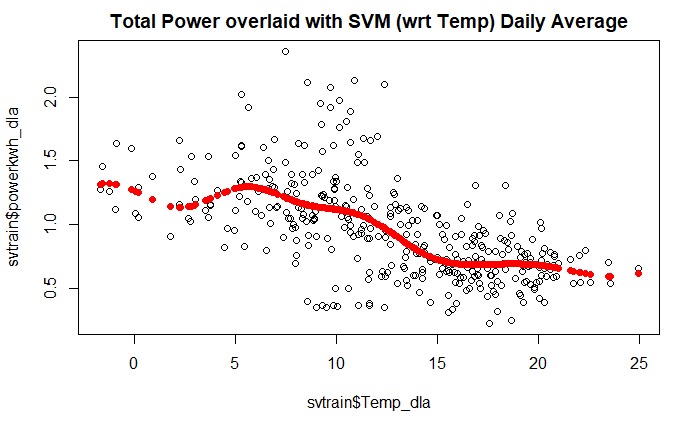
- best parameters:

elsilon cost

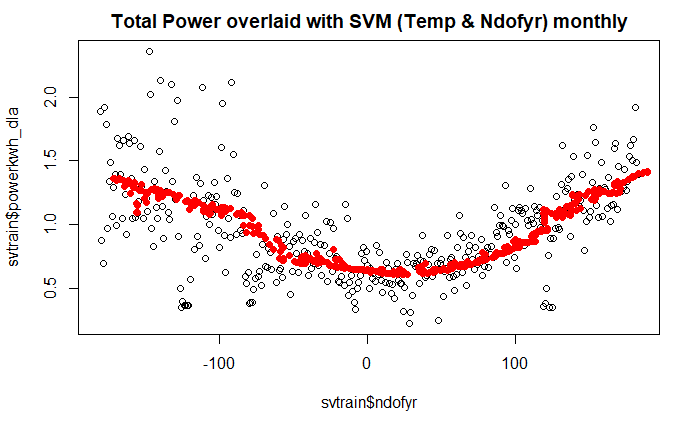
0 15

- best performance: 0.08526917

### 5.3.2 Power Consumption Relationship with Temperature



It is clear that power consumption has an inverse polynormal relationship with temperature. The results shown was achieved with a gamma of 1 and cost of 300 on the e1071 package svm function. With a gamma of 0 the result is a straight line which is clearly not suitable for the obvious trend. With a gamma of 0.2 the negative slope emerges which is almost useless.



### 5.3.3 Power Consumption Relationship with Humidity

We also explored relationship of power consumption with Humidity alone using the svm function in the e1071 package of R. See codes below

#### *Relevant Code for Humidity Alone*

plot(svtrain$powerkwh\_dla ~ svtrain$Humd\_dla, main="Total Power overlaid with SVM (wrt Humidity) Daily Average")

model\_Hsvm <- svm(powerkwh\_dla~Humd\_dla,data=svtrain, kernel='radial',gamma=1,cost=300)

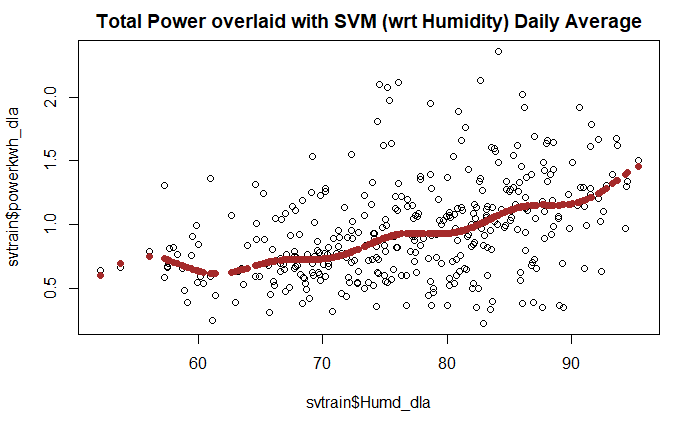
#Predict Output

predsH <- predict(model\_Hsvm,svtrain)

table(predsH)

points(svtrain$Humd\_dla, predsH, col = "brown", pch=16 )

The plot shows a positive correlation between power consumption and humidity both of them are obviously responding to temperature in the atmospheric conditions.



### 5.3.4 Power Consumption Relationship with Wind Speed

The relationship with Wind speed is shown below along with the codes

#### 5.3.4.1 Relevant Code for Model with Wind Speed Alone

head(svtrain)

plot(svtrain$powerkwh\_dla ~ svtrain$WndSpd\_dla, main="Total Power overlaid with SVM (wrt WindSpeed) Daily Average")

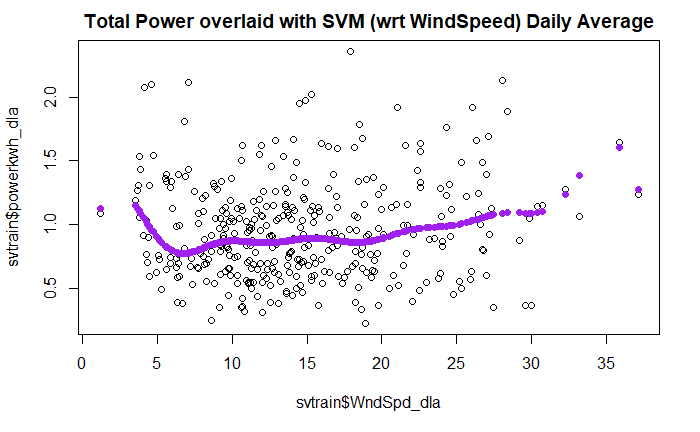
model\_Wsvm <- svm(powerkwh\_dla~WndSpd\_dla,data=svtrain, kernel='radial',gamma=1,cost=300)

#Predict Output

predsW <- predict(model\_Wsvm,svtrain)

table(predsW)

points(svtrain$WndSpd\_dla, predsW, col = "purple", pch=16 )



### 5.3.5 Full Period Prediction Temperature, Humidity and Day if Year

Power consumption was modelled against Temperature, Humidity and Day of Year separately with the entire dataset (daily average) over the 47 months. The results is as follows: Codes are similar so only one set is written here

#### 5.3.5.1Relevant Code SVM Model with Temperature entire dataset Dec06 to Nov10

plot(mgdata\_dly$powerkwh\_dla ~ mgdata\_dly$Temp\_dla, main="Total Power overlaid with SVM Prediction (wrt Temp 47months) DLA")

model\_TFsvm <- svm(powerkwh\_dla~Temp\_dla,data=mgdata\_dly, kernel='radial',gamma=2.5,cost=200)

#Predict Output

predsTF <- predict(model\_TFsvm,mgdata\_dly)

points(mgdata\_dly$Temp\_dla, predsTF, col = "violet", pch=16 )

#Model with Humidity Alone with entire dataset Dec2006 to Nov 2010

plot(mgdata\_dly$powerkwh\_dla ~ mgdata\_dly$Humd\_dla, main="Total Power overlaid with SVM Prediction (wrt Humidity 47months) DLA")

model\_HFsvm <- svm(powerkwh\_dla~Humd\_dla,data=mgdata\_dly, kernel='radial',gamma=2.5,cost=200)

#Predict Output

predsHF <- predict(model\_HFsvm,mgdata\_dly)

points(mgdata\_dly$Humd\_dla, predsHF, col = "brown", pch=16 )

#Model with Day of the Year Alone with entire dataset Dec2006 to Nov 2010

head(mgdata\_dly)

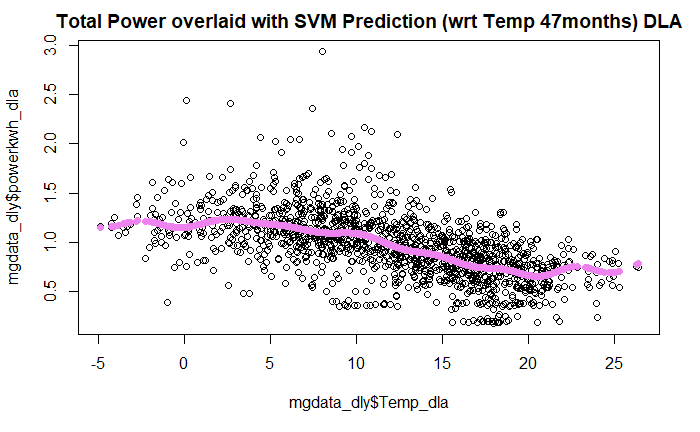
plot(mgdata\_dly$powerkwh\_dla ~ mgdata\_dly$Date, main="Total Power overlaid with SVM Prediction (wrt Date 47months) DLA")

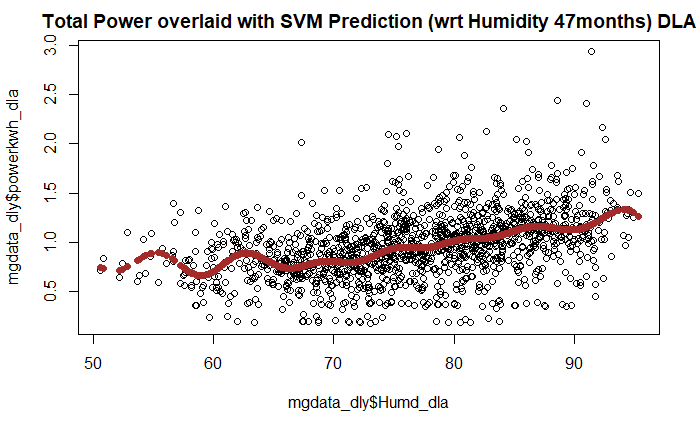
model\_DFsvm <- svm(powerkwh\_dla~Date,data=mgdata\_dly, kernel='radial',gamma=3,cost=200)

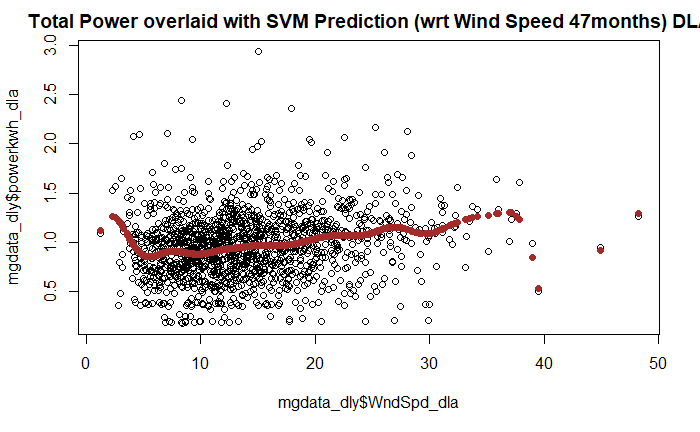
#Predict Output

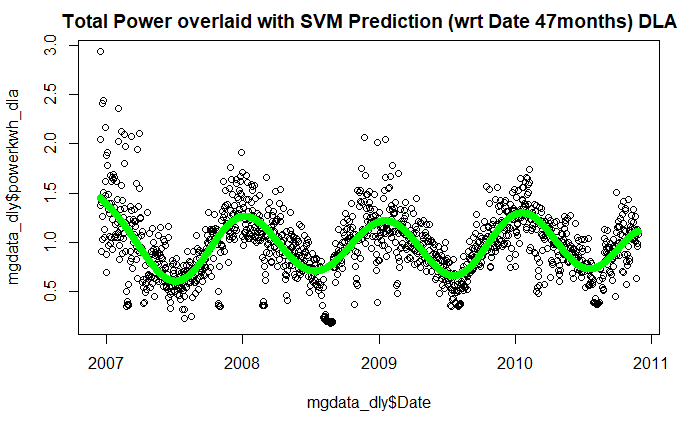
predsDF <- predict(model\_DFsvm,mgdata\_dly)

points(mgdata\_dly$Date, predsDF, col = "green", pch=16 )









The following is note on results.

Either of temperature, humidity and day of the year could be used to get a reasonable estimate of the daily average temperature for the location. Day of the year appears to give the best model. For the temperature and humidity, the spread across these predicted averages seem significant, suggesting that other factors appear to be active. It is not clear if these are polynomials of existing attributes or variations visible under the hourly resolution.

It is necessary to check if a combination of these features would be more effective and more accurate. One way to achieve this is to use Neural Nets, or some other package robust for complex relationships.

It is clear that if SVM is to be used, the most relevant base for prediction would be day of the year. The power consumption them follows a near sine wave with a peak of 1.5kwh and trough of around 0.6kwh. The prediction also features a growing tread as the time progresses.

## 5.4 Artificial Neural Networks with NeuralNet

Analytics Vidya (2017) demonstrates how neural nets can be used to mine data particularly complex datasets. Its interconnected neurons, or interconnected information processing units works like the human brain, robust for non-linear data and quite fast. The information processing units do not work in a linear manner. The sort of power consumption data have been successfully modelled with neural networks by several authors (references)

The following code is inspired by Analytics Vidya (2017) on neural nets.

#### #Relevant Code for Artificial Neural Network

tail(mgdata\_dly)

startD <- lubridate::ymd\_hms("2006-12-16 00:59:00", tz = "Europe/Paris")

mgdata\_dly$tnofdays <- as.duration(startD%--%mgdata\_dly$Date)/ddays((1))

# scale data for nls

mgdata\_dsc <- mgdata\_dly[, c(2:7)]

max = apply(mgdata\_dsc , 2 , max)

min = apply(mgdata\_dsc, 2 , min)

scaled = as.data.frame(scale(mgdata\_dsc, center = min, scale = max - min))

tail(scaled)

scaled$tnofdays <- mgdata\_dly$tnofdays

# Random sampling

samplesize = 0.65 \* nrow(mgdata\_dsc)

set.seed(500)

index = sample( seq\_len ( nrow ( mgdata\_dsc ) ), size = samplesize )

# Create training and test set

mgdata\_dsctrain = mgdata\_dsc[ index, ]

mgdata\_dsctest = mgdata\_dsc[ -index, ]

## Fit neural network

# install library

install.packages("neuralnet")

# load library

library(neuralnet)

# creating training and test set

ptrainNN = scaled[index , ]

ptestNN = scaled[-index , ]

# fit neural network

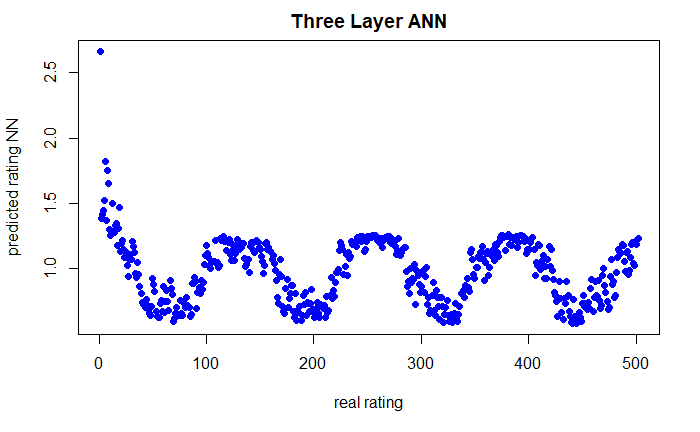
set.seed(200)

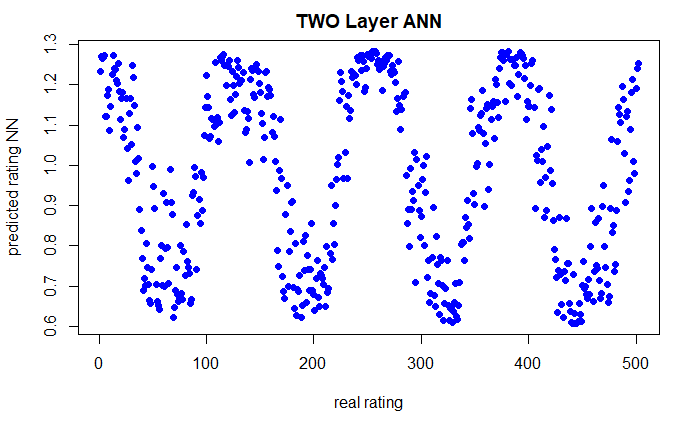
NN = neuralnet(powerkwh\_dla ~ Temp\_dla + Humd\_dla + Pressure\_dla + WndSpd\_dla + sin((pi\*tnofdays/366)), ptrainNN, hidden = 3 , linear.output = FALSE )

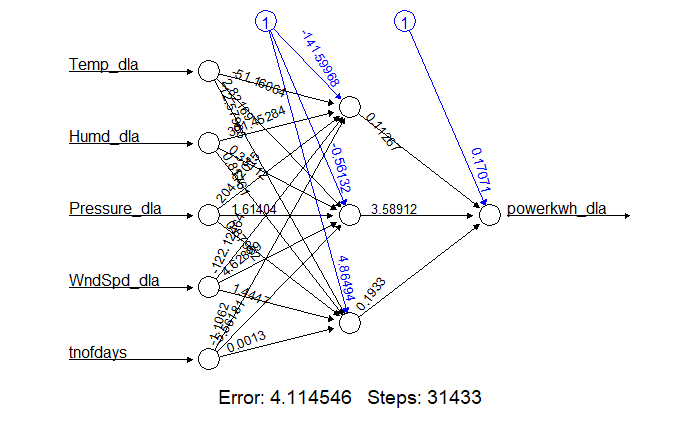
# plot neural network

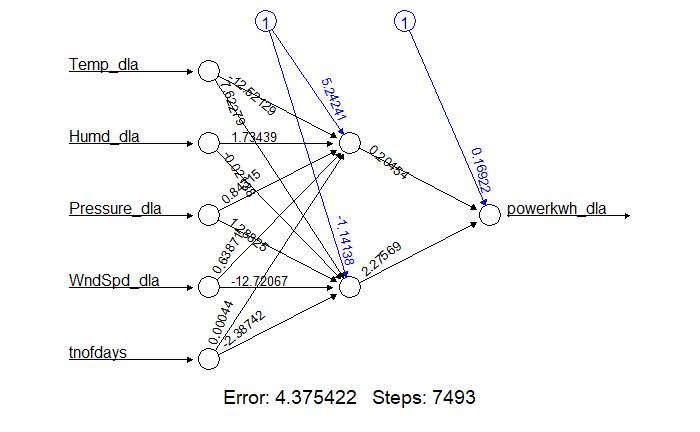
plot(NN)

Using two layers converged faster than using 3 layers Both results in a time varying power consumption, which is what we expect. The plots for hidden layers set at 1, 4 and 5 returned non-useful results (a flat horizontal line).









## 5.5 Hourly Prediction

In order to predict he hourly consumption data we have return to the hourly dataset dstmdl and make a copy. First we check the forms of the power consumption with time and and other variables. It is clear that the monthly / daily trend has its effect. With a certain month and day, there are peaks and lows that are possible and this are governed by the prediction already done in the previous section. To refine these further into hourly predictions we have to start with averages for the given hours and given month and then build one this. One way to extend this maybe to look as the mean, max and min for each month and the impose a distribution of a time varying form. We already know the averages (see below). We can also calculate the max and mean for given hour and month but these may not be very useful as the dynamic drivers for the varying portions may not time alone.

Another option is to seat on the historical averages for the current month and impose a time varying structure based on recent events or predictions based on external predictors of temperature, humidity and time of the day.

#### 5.5.1.1 Relevant Code for Hourly ANN Forecasts

##Hourly forecasts for Power Consumption

pwrstfrc <- dstmdlx

head(pwrstfrc)

nrow(pwrstfrc)

#the last 100days and most relevant attributes pwr, temp, Humd, wndspd, press date and Hr

pwrstfrc\_1 <- pwrstfrc[c(33880:34120), c(1,2,4,5,6, 7, 13)]

head(pwrstfrc\_1, 60)

plot(pwrstfrc\_1$power\_kwh)

lines(pwrstfrc\_1$power\_kwh)

## The power consumption is lowests midnight to around 4am and then peaks again at about 7am and then rises to around middday. The peaks for each day is different based on the daily averages we carried out earlier.

## From these perhaps an appropriate method is predicting hourly temperatures is to have a handle on the month hourly averages.

library(data.table)

library(zoo)

pwrstfrc$yearmon <- as.yearmon(pwrstfrc$Date, "%m/%Y")

pwrstfrc <- as.data.table(pwrstfrc)

setkey(pwrstfrc, Hour, yearmon)

# aggregate data by month and hour, target is active power for time series

pwrstfrc\_agg <- as.data.frame(pwrstfrc[, j=list(apower\_agg= mean(m\_apower, na.rm = TRUE), powerkwh\_agg = mean(power\_kwh, na.rm = TRUE), Temp\_agg = mean(Temp, na.rm = TRUE), Humd\_agg = mean(Humd, na.rm = TRUE), Pressure\_agg = mean(Pressure, na.rm = TRUE), WndSpd\_agg = mean(WndSpeed, na.rm = TRUE)), by = list(yearmon, Hour)])

yearmon Hour apower\_agg powerkwh\_agg Temp\_agg Humd\_agg Pressure\_agg WndSpd\_agg

1 Dec 2006 17 2.018467438 2.029236583 6.302500 78.6250 1030.11250 14.390625

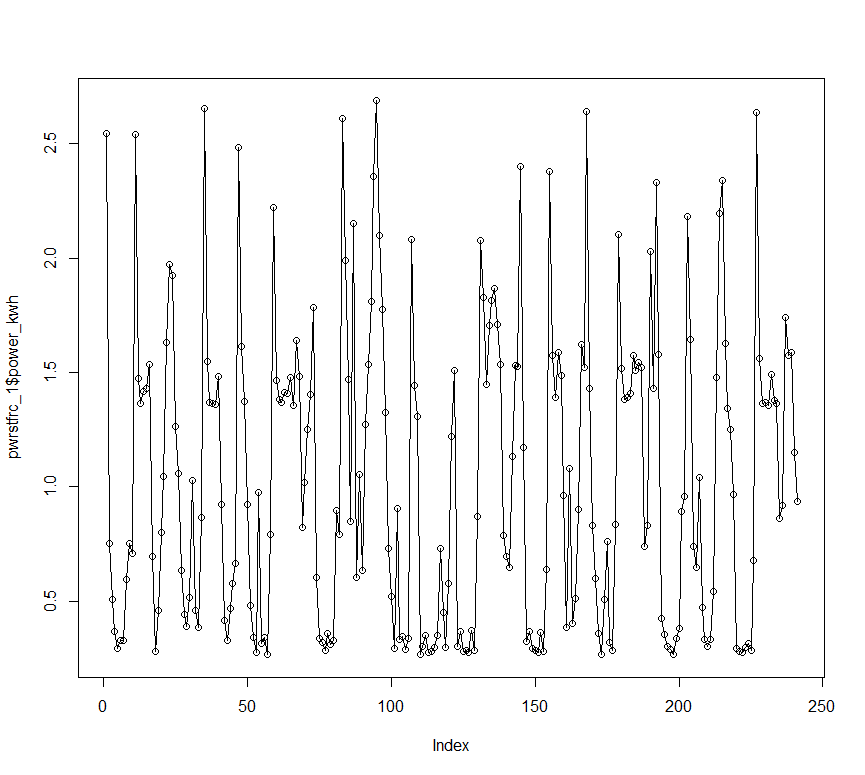
2 Dec 2006 18 2.240951154 2.251901567 5.395000 83.5000 1029.97500 13.723125

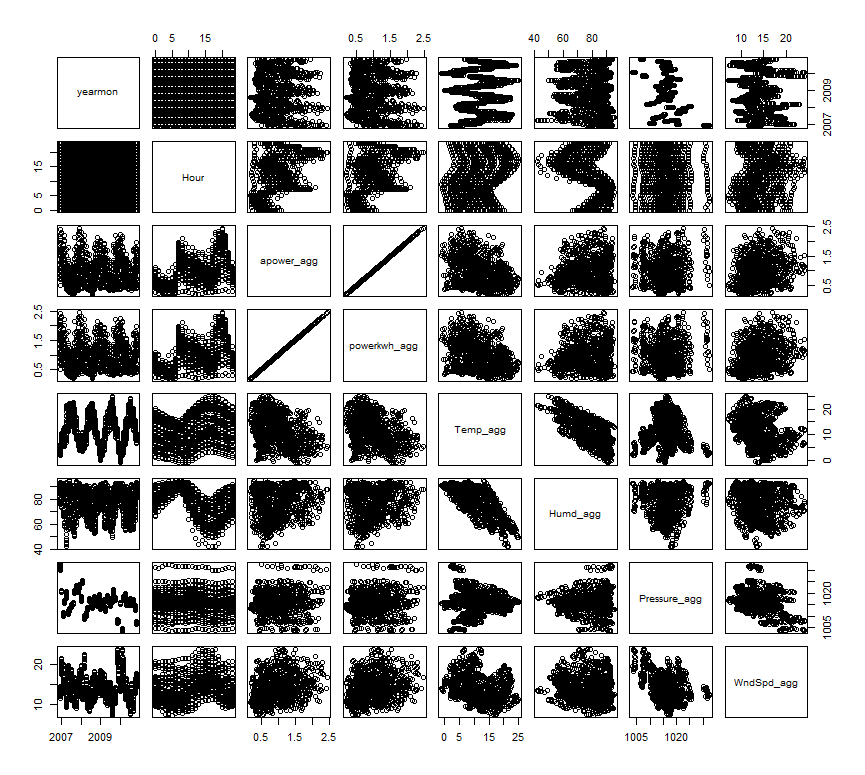
3 Dec 2006 19 2.236095682 2.246212683 4.938125 86.3125 1030.13125 13.848125

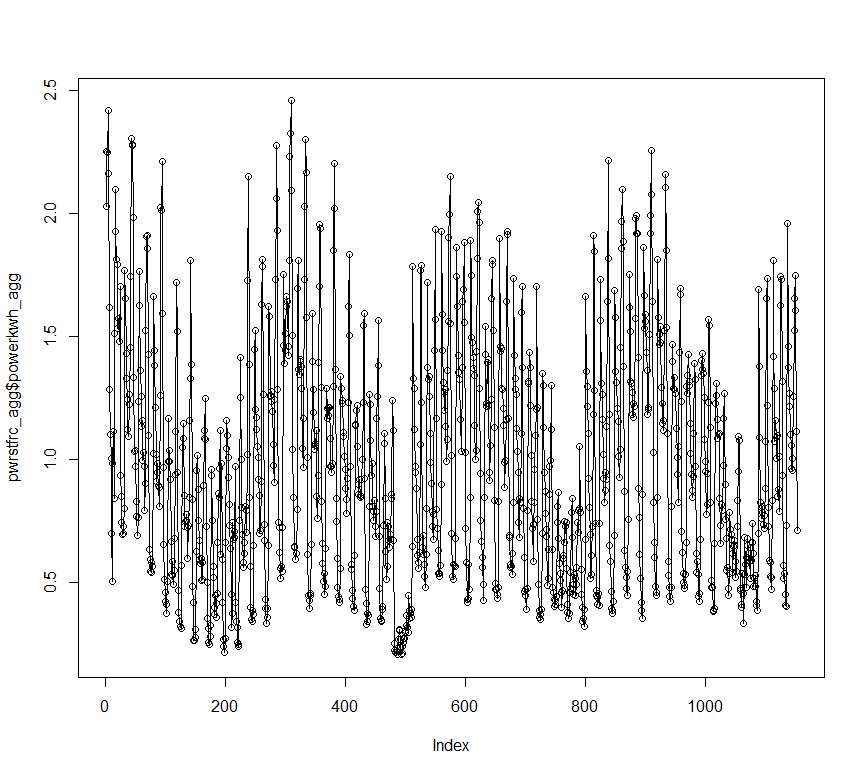
4 Dec 2006 20 2.407870389 2.419192362 4.650625 88.1250 1030.55625 13.975000

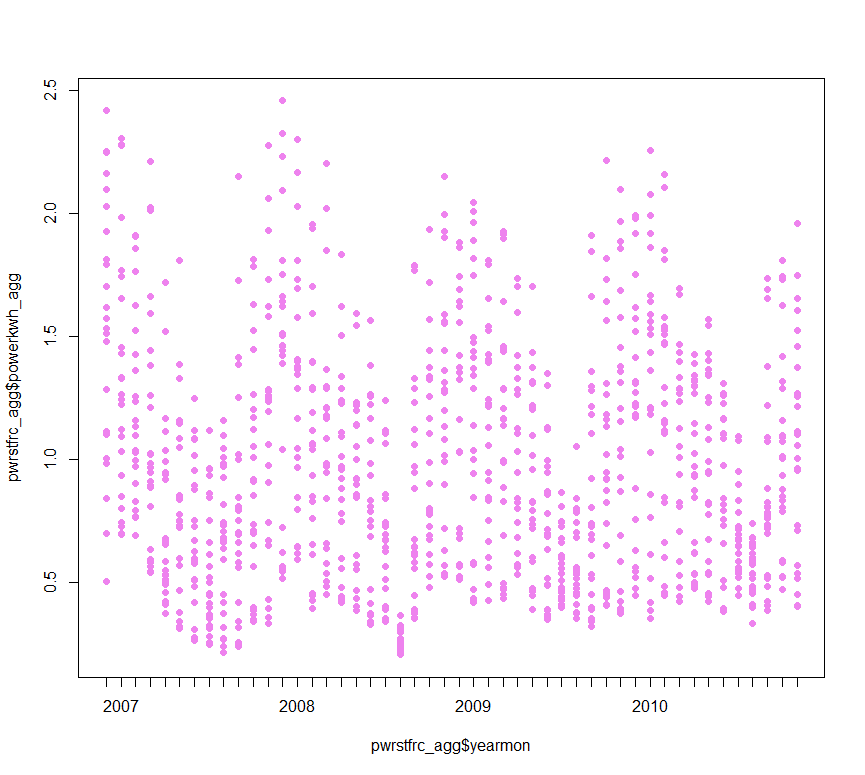
5 Dec 2006 21 2.150050592 2.160364404 4.378750 89.2500 1030.98125 14.084375

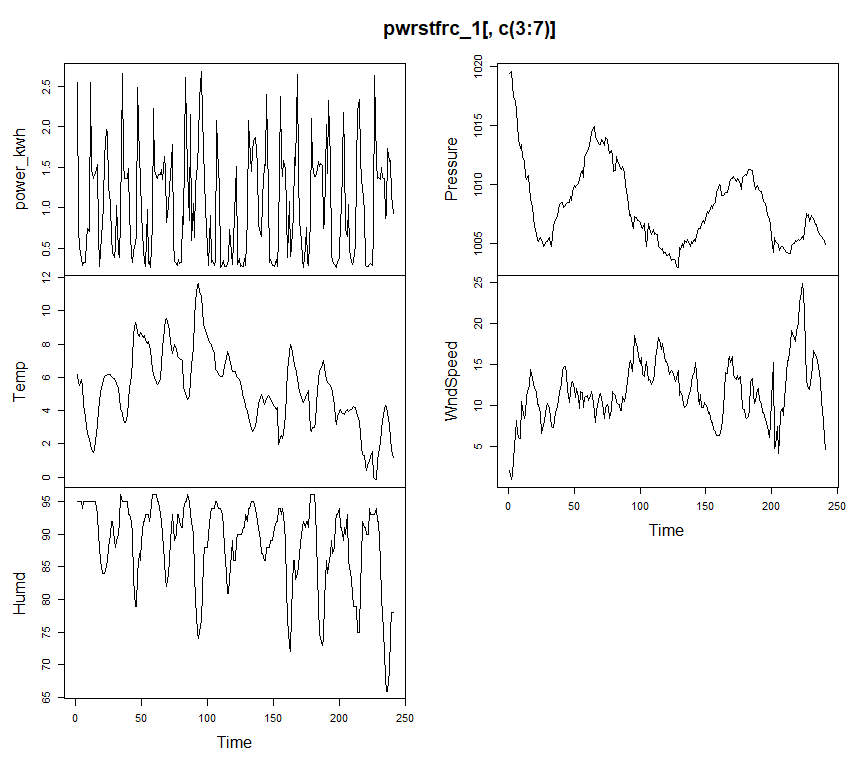
6 Dec 2006 22 1.611389210 1.616206543 4.095000 90.5000 1031.28750 13.968750

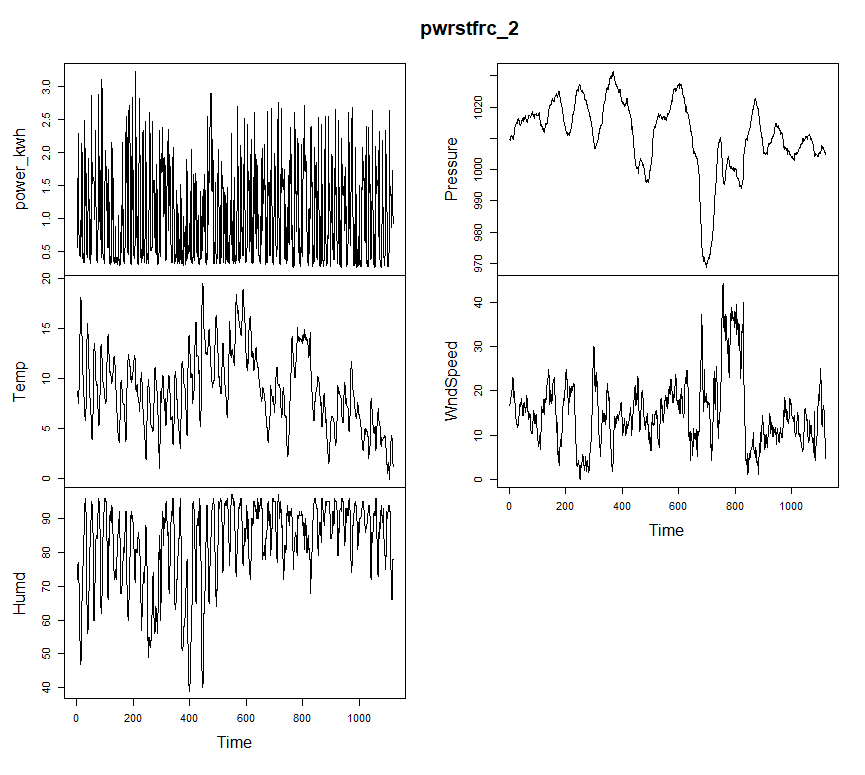
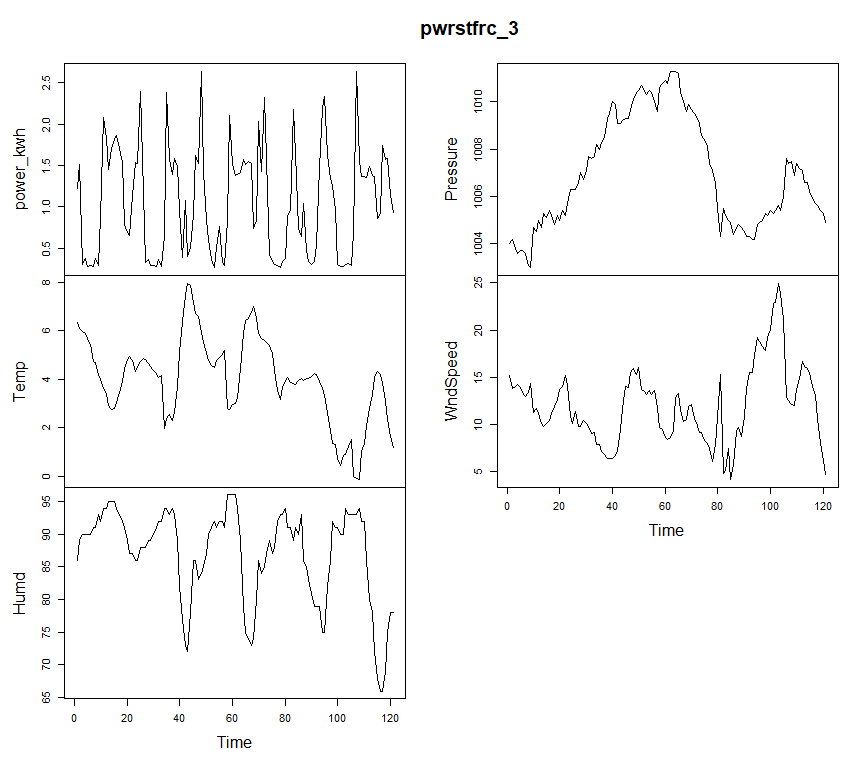








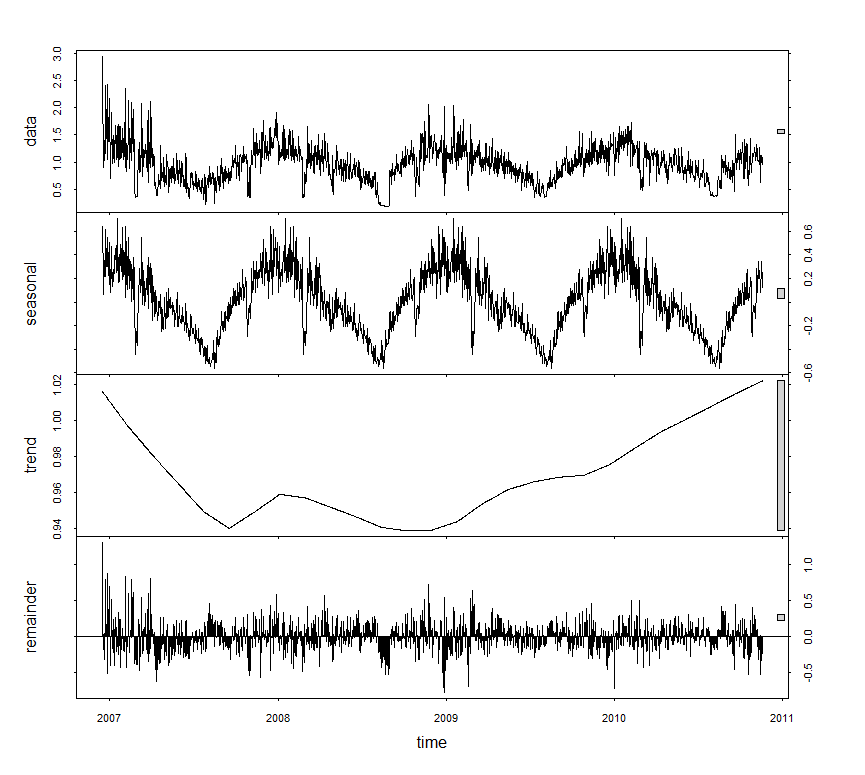




## 5.6 Time Series Forecasts

The stats and forecast packages are the basis of exponential and ARIMA time series forecasts

### 5.6.1 Time Series daily Forecasts



Results of Exopential ETS Fit Model

|  |
| --- |
| ETS(A,Ad,N)  Call:  ets(y = tspwr)  Smoothing parameters:  alpha = 0.2443  beta = 0.0001  phi = 0.8  Initial states:  l = 2.2585  b = -0.2764  sigma: 0.2346  AIC AICc BIC  6266.366916 6266.425822 6297.972069 |
|  |
| |  | | --- | |  | |

The model was requested to select the best model and the ETS(A, Ad, N) means that Seasonality is Additive, Trend is Additive and Irregular is Negative. The forecast is shown the dark line, and the 80% and 95% confidence intervals are the light and dark gray, respectively.

#### #Relevant R Code Daily Time Series

`{r}

#### #First We explore the daily time series

#### library(stats)

#### head(mgdata\_dly)

#### prds <- seq(as.Date("2006-12-16"), as.Date("2010-11-26"), by = "day") # start and end dates

#### tspwr <- ts(mgdata\_dly[ , 3], start=c(2006, as.numeric(format(prds[1], "%j"))), frequency=365)

#### # decompose stationary and other components

#### tspwr.fit <- stl(tspwr, s.window= "periodic", t.window=)

#### plot(tspwr)

#### plot(tspwr.fit)

#### # visualize seasonal decomposition

#### library(forecast)

#### monthplot(tspwr, xlab="", ylab="", choice = "seasonal", col = "green", pch=16, main="Monthplot of daily Electricity Consumption TS")

#### ## use ets exponential fit to choose ARIMA terms

#### library(forecast)

#### fitpwr1 <- ets(tspwr)

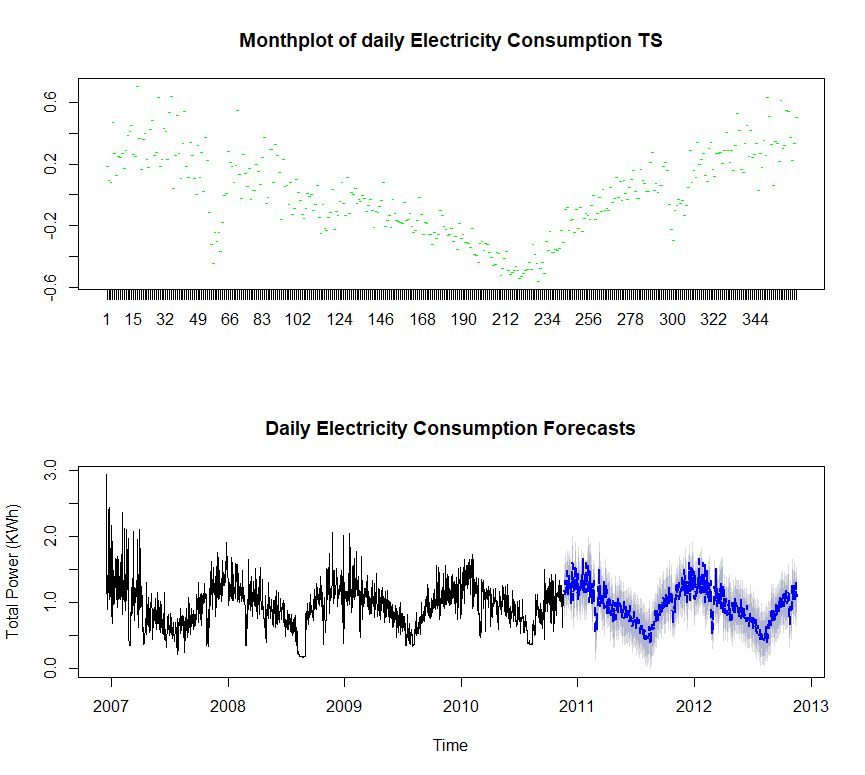
#### fitpwr1

#### forecast(fitpwr1, 10)

#### accuracy(fitpwr1)

#### plot(forecast(tspwr.fit, method=c("arima")), main="Daily Electricity Consumption Forecasts",

#### ylab="Total Power (KWh)", xlab="Time", flty=2)

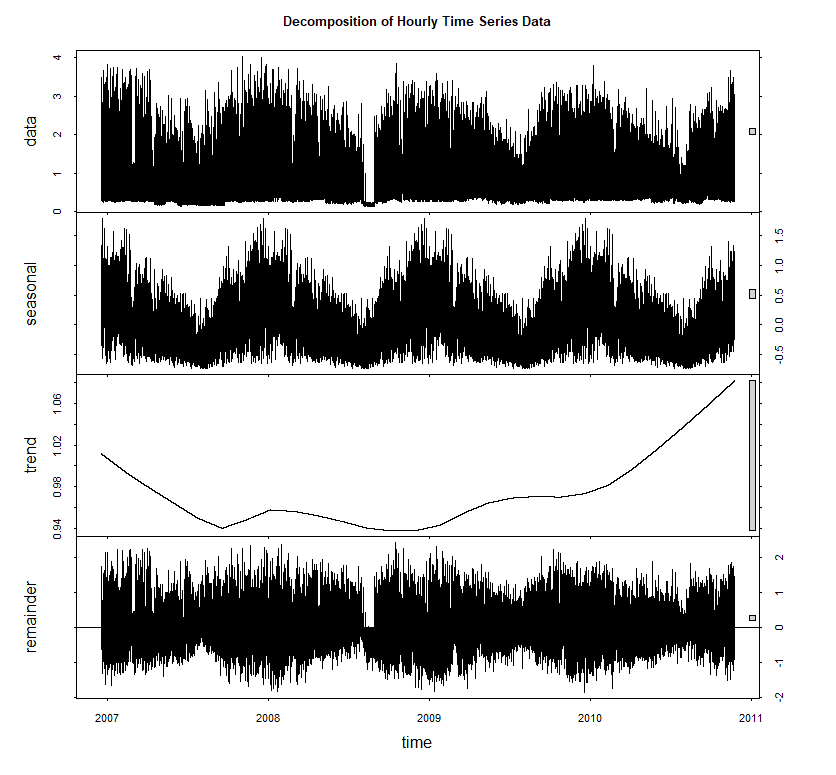


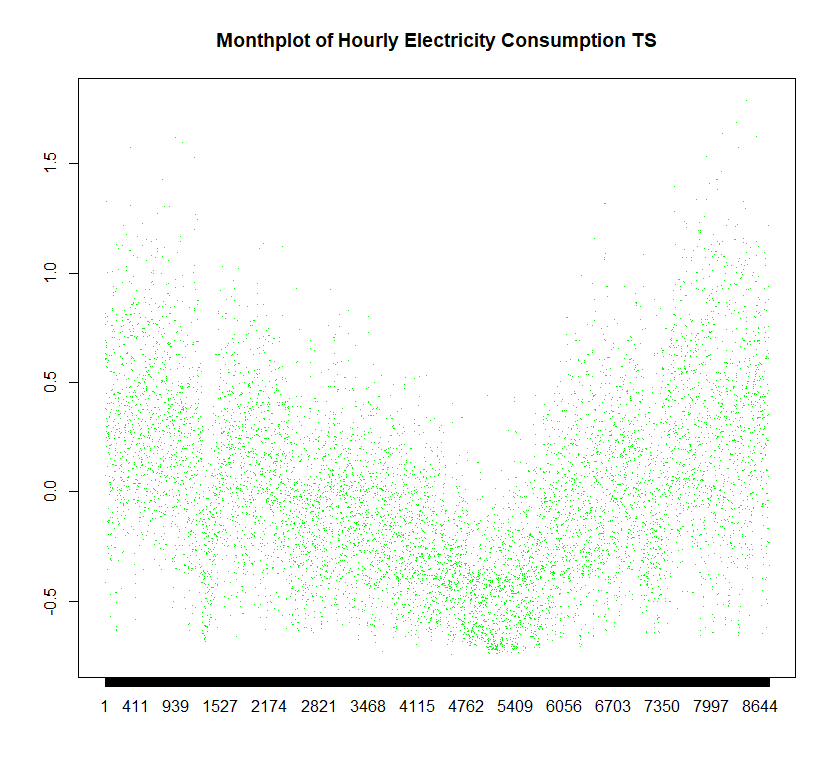
### Results of Ten Step Forecast

|  |
| --- |
| forecast(fitpwr1, 10)  Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  2010.882192 1.036255078 0.7355664964 1.336943659 0.5763915764 1.496118579  2010.884932 1.036251240 0.7267135995 1.345788881 0.5628542705 1.509648210  2010.887671 1.036248170 0.7181030645 1.354393275 0.5496872190 1.522809121  2010.890411 1.036245714 0.7097164975 1.362774930 0.5368623718 1.535629056  2010.893151 1.036243749 0.7015376666 1.370949831 0.5243549695 1.548132528  2010.895890 1.036242177 0.6935521665 1.378932187 0.5121430331 1.560341321  2010.898630 1.036240919 0.6857471467 1.386734692 0.5002069511 1.572274888  2010.901370 1.036239913 0.6781110903 1.394368736 0.4885291432 1.583950683  2010.904110 1.036239108 0.6706336307 1.401844586 0.4770937816 1.595384435  2010.906849 1.036238465 0.6633053999 1.409171529 0.4658865606 1.606590369  accuracy(fitpwr1)  ME RMSE MAE MPE MAPE MASE ACF1  Training set -0.0003326604091 0.2342188581 0.1688133739 -5.92030091 19.76891723 0.6943318791 0.1165607424 |
|  |
| |  | | --- | |  | |

The daily forecast is very useful as it has face validity. You see it and you immediately recognise that it represents the historical trend. Another useful feature of it as that it provides confidence intervals for 80% and 95% for the high and low cases. These neatly plotted and provides an effective boundary for the forecasts going forward.

### 5.6.2 Time Series Hourly Forecasts





forecast(fitpwrhrly2, 10)

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

2010.895548 2.905250516 2.2088932354 3.601607796 1.84026395843 3.970237073

2010.895662 2.905250516 1.9505168176 3.859984214 1.44511132696 4.365389705

2010.895776 2.905250516 1.7484794031 4.062021628 1.13612176577 4.674379266

2010.895890 2.905250516 1.5768219781 4.233679053 0.87359438899 4.936906643

2010.896005 2.905250516 1.4249379297 4.385563102 0.64130778220 5.169193249

2010.896119 2.905250516 1.2872491792 4.523251852 0.43073101003 5.379770022

2010.896233 2.905250516 1.1603919835 4.650109048 0.23671967134 5.573781360

2010.896347 2.905250516 1.0421524855 4.768348546 0.05588796407 5.754613067

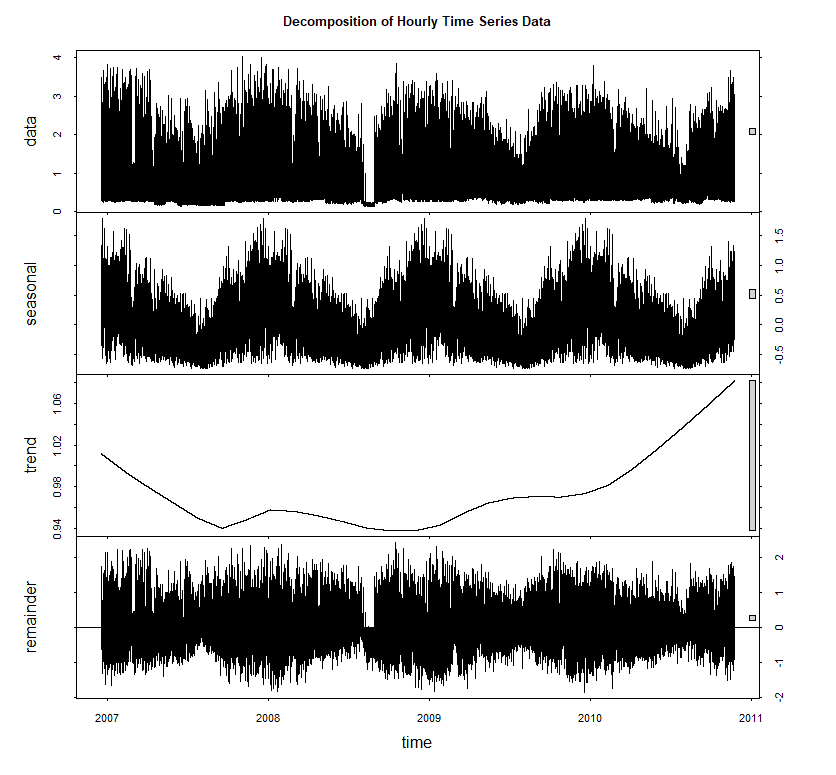
2010.896461 2.905250516 0.9309817289 4.879519303 -0.11413303629 5.924634068

2010.896575 2.905250516 0.8257457152 4.984755316 -0.27507763048 6.085578662

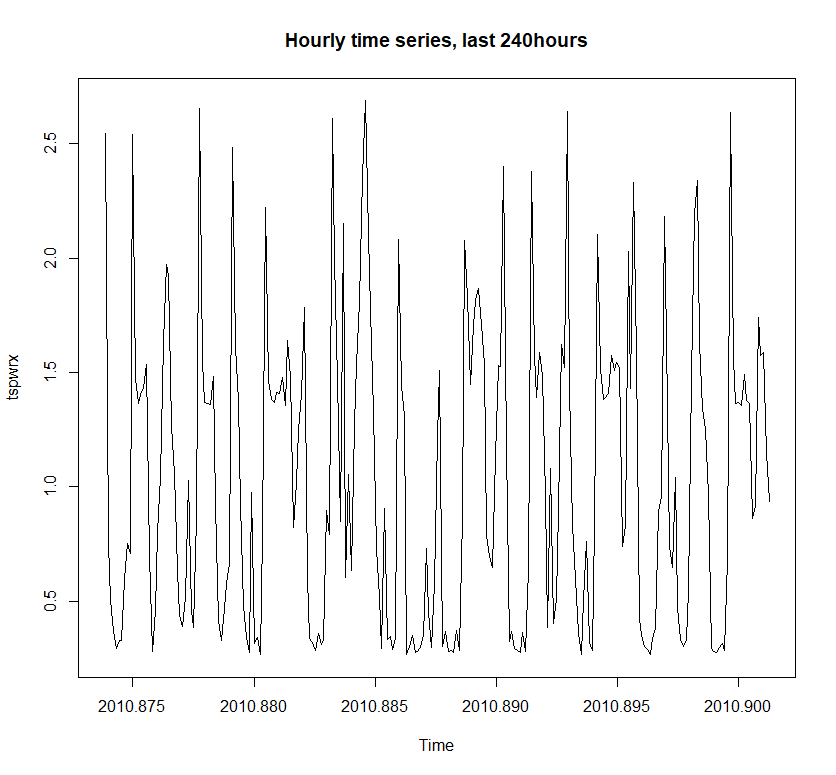
accuracy(fitpwrhrly2)

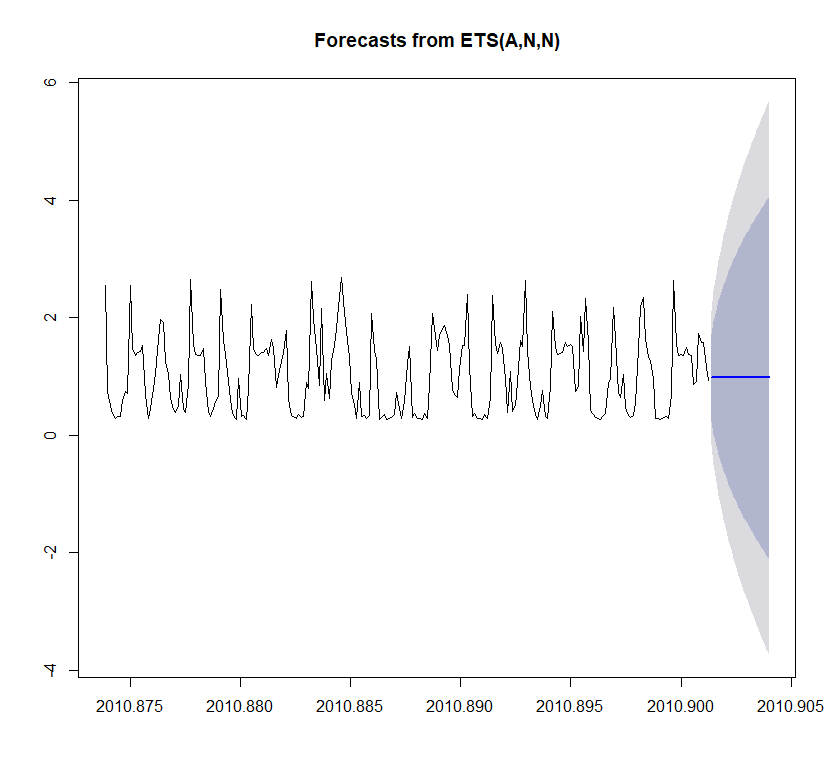
ME RMSE MAE MPE MAPE MASE ACF1

Training set -0.000006587124666 0.5433547087 0.3587381601 -18.93071716 45.6710283 0.4589759348 0.008432482588



|  |
| --- |
| head(pwrhrx)  Hour power\_kwh thr2  1: 21 2.5475000000 2010-11-16 21:00:00  2: 22 0.7507000000 2010-11-16 22:00:00  3: 23 0.5062166667 2010-11-16 23:00:00  4: 0 0.3687166667 2010-11-17 00:00:00  5: 1 0.2940666667 2010-11-17 10:00:00  6: 2 0.3284666667 2010-11-17 20:00:00  > tail(pwrhrx)  Hour power\_kwh thr2  1: 16 0.9171568627 2010-11-26 16:00:00  2: 17 1.7387666667 2010-11-26 17:00:00  3: 18 1.5734666667 2010-11-26 18:00:00  4: 19 1.5879259259 2010-11-26 19:00:00  5: 20 1.1491694915 2010-11-26 20:00:00  6: 21 0.9346666667 2010-11-26 21:00:00 |
|  |
| |  | | --- | |  | |





#### #Time Series Forecasts for Power Consumption HOURLY

```{r cars}

require(forecast)

pwrhrly <- hholdTHR

pwrhrly$power\_subs=pwrhrly$m\_smeter1+pwrhrly$m\_smeter2+pwrhrly$m\_smeter3

pwrhrly$power\_kwh = 0.001\*pwrhrly$power\_subs + pwrhrly$m\_apower

pwrhrly$thr <- (paste(pwrhrly$Date, paste(pwrhrly$Hour, "00", sep=""), sep=" "))

head(pwrhrly)

tail(pwrhrly)

class(pwrhrly$thr)

pwrhrly <- pwrhrly[ , c(1,2, 11, 13)]

library(lubridate)

pwrhrly$thr2 <- ymd\_hm(as.character(pwrhrly$thr), tz ="Europe/Paris")

head(pwrhrly)

tail(pwrhrly)

nrow(pwrhrly) #34120

class(pwrhrly$thr2)

## in order to have finer resolution we are going to subset the last 24 \* 10 = 240hours of data, power\_kwh and hourly information

pwrhrx <- pwrhrly[c(33880:34120), c(2,3,5)]

dat <- pwrhrx$thr2

head(pwrhrx)

tail(pwrhrx)

#First We explore the daily time series

library(stats)

##Here's how to use the ts() function in base R (assuming your data X are contained in the data frame dat). You'll need to specify the first year and hour for start (you don't need end), and frequency will be the number of hours in a year.

startHour <- 24\*(as.Date("2010-11-16 21:00:00")-as.Date("2010-1-1 00:00:00"))

tspwrx <- ts(pwrhrx$power\_kwh,start=c(2010,startHour),frequency=24\*365)

# decompose into seasonal, trend and irregular components

tspwrx.fit <- decompose(tspwrx, type = c("additive", "multiplicative"))

#tspwrx.fit <- stl(tspwrx, s.window= "", t.window=)

plot(tspwrx, main="Hourly time series, last 240hours")

#plot(tspwrx.fit, main ="Decomposition of Hourly Time Series Data, c(2010, 7845) Start")

## use ets exponential fit ETS to choose ARIMA terms

library(forecast)

fitpwrx <- ets(tspwrx)

fitpwrx

forecast(fitpwrx, 10)

accuracy(fitpwrx)

plot(forecast(fitpwrx, method=c("rwdrift")), main="Hourly Electricity Forecasts",

ylab="Total Power (KWh)", xlab="Time", flty=2) # model with ARIMA

## AUto.Arima

pwr\_aa = auto.arima(tspwrx)

fpwr\_aa = forecast(pwr\_aa, h=24)

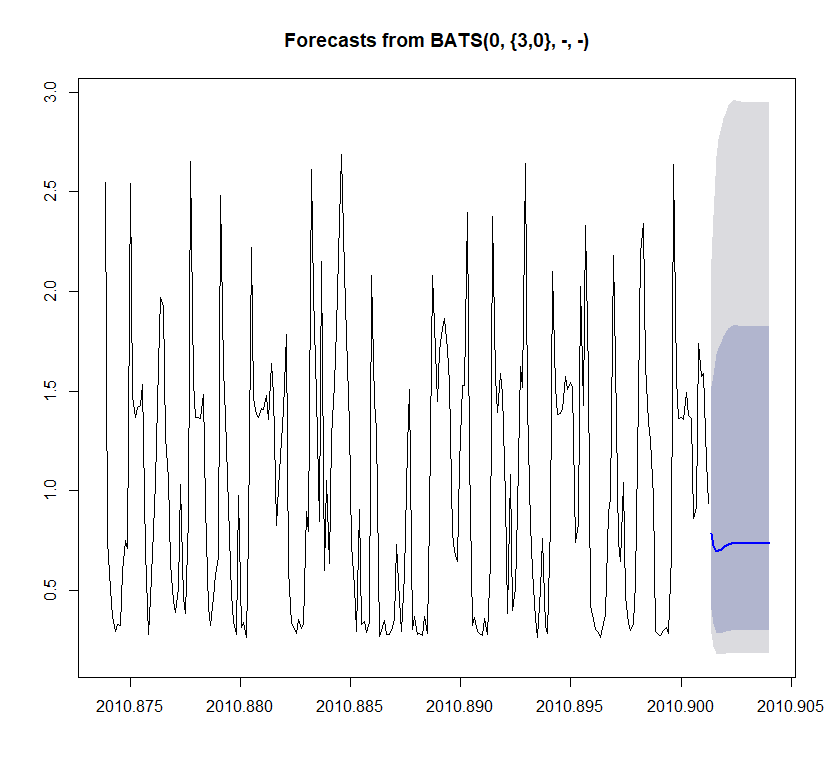
plot(fpwr\_aa)

## TBATS

pwr\_tbats = tbats(tspwrx)

fpwr\_tbats = forecast(pwr\_tbats, h=24)

plot(fpwr\_tbats)

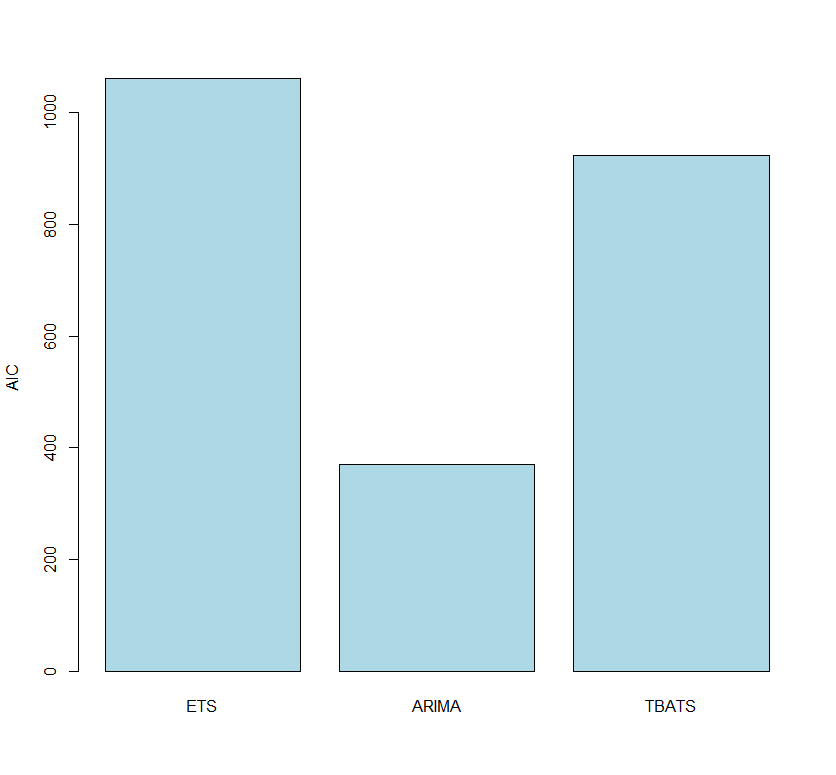


We use AIC to compare the three models and the models with the smallest AIC value is the best of the three

barplot(c(ETS=fitpwrx$aic, ARIMA=pwr\_aa$aic, TBATS=pwr\_tbats$AIC),

col="light blue",

ylab="AIC")



Unfortunately, the bounds for the hourly forecasts are much too broad to be useful. The same time series derived (e.g. ETS) package used for the daily forecast are less effective for hourly data because it looses it seasonality for hourly data.

Thus, we had to resort to using TBATS package to have more meaningful results. This is also time series derived but handles multiple frequencies or zero frequencies much better. The results were only marginally better.

### 5.6.3 Hourly Time Series Forecasts Version 2

head(pwrhrx2)

Hour power\_kwh thr2

1: 13 0.3420666667 2007-02-22 13:00:00

2: 14 0.6082666667 2007-02-22 14:00:00

3: 15 0.4885000000 2007-02-22 15:00:00

4: 16 0.6482068966 2007-02-22 16:00:00

5: 17 0.7028333333 2007-02-22 17:00:00

6: 18 1.5254000000 2007-02-22 18:00:00

> tail(pwrhrx2)

Hour power\_kwh thr2

1: 16 0.9171568627 2010-11-26 16:00:00

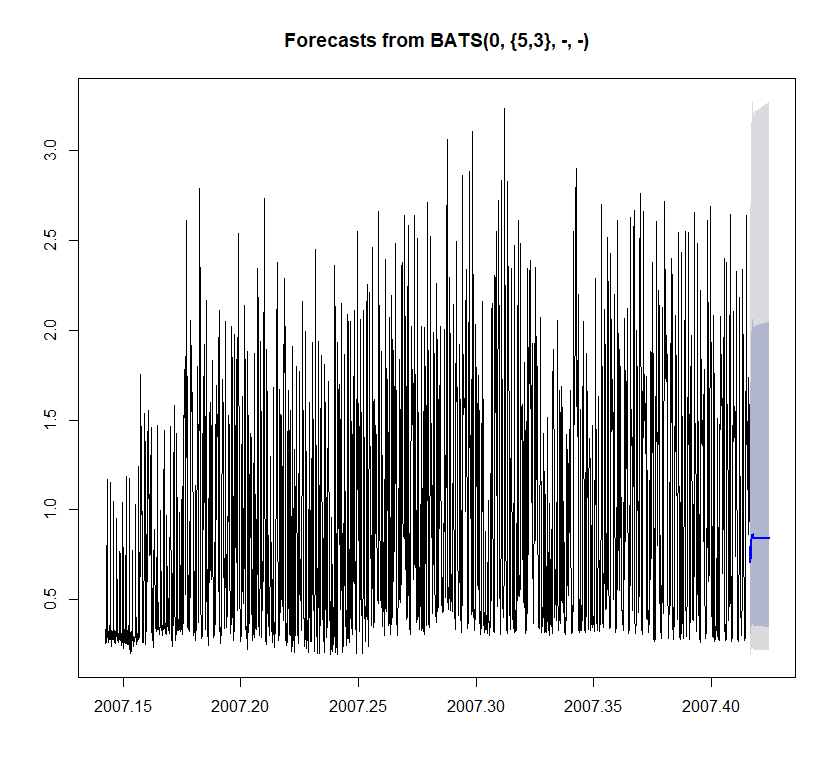
2: 17 1.7387666667 2010-11-26 17:00:00

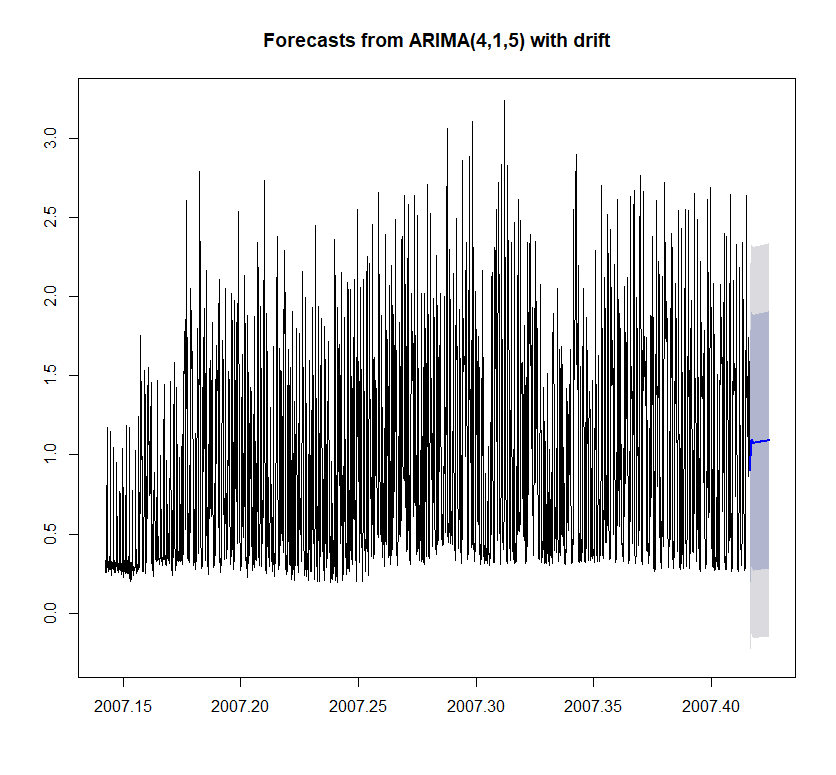
3: 18 1.5734666667 2010-11-26 18:00:00

4: 19 1.5879259259 2010-11-26 19:00:00

5: 20 1.1491694915 2010-11-26 20:00:00

6: 21 0.9346666667 2010-11-26 21:00:00





# References

Yu Z., Haghighat F., Fung B.C.M and Yoshino H. “A decision tree method for building energy demand modeling” (2010) Elsevier Energy and Buildings 42 (2010) 1637–1646

Lucas B. I., E. Hidalgo, Gomez W. and Rose R. (2001) “Behavioral Factors Study of Residential Users which Influence the Energy Consumption” Elsevier Renewable Energy 24 (2001) 521–527.

Neto A.H. and Fiorelli F. A. S. (2008) “Comparison between Detailed Model Simulation and Artificial Neural Network for Forecasting Building Energy Consumption” Elsevier Energy and Buildings 40 (2008) 2169–2176.

Li Z., Han Y. and Xu P. (2014) “Methods for benchmarking building energy consumption against its past or intended performance: An overview” Elsevier Applied Energy 124 (2014) 325–334

Catalina T., Iordache V. and Caracaleanu B. (2013)”Multiple regression model for fast prediction of the heating energy demand” Elsevier Energy and Buildings 57 (2013) 302–312

Chujai P., Kerdprasop N., and Kerdprasop Kittisak (2013) “Time Series Analysis of Household Electric Consumption with ARIMA and ARMA Models”. Proceedings of the International MultiConference of Engineers and Computer Scientists Vol I, March 13 - 15, 2013, Hong Kong

Edwards R. E. , Newb J. and Lynne E. Parker L.E. (2012) a “Predicting future hourly residential electrical consumption: A machine learning case study” Elsevier Energy and Buildings ENB-3661

Beccali M., M. Cellura M., Brano V. L and Marvuglia A. (2007) “Short-term prediction of household electricity consumption: Assessing weather sensitivity in a Mediterranean area”

Heden W. (2016) “Predicting Hourly Residential Energy Consumption using Random Forest and Support Vector Regression” KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF ENGINEERING SCIENCES. <https://kth.diva-portal.org/smash/get/diva2:932582/FULLTEXT01.pdf> Master Thesis

Chaitanya S. (2017) “Building Regression Models in R Using Support Vector Regression” <https://www.kdnuggets.com/2017/03/building-regression-models-support-vector-regression.html>

Analytics Vidya BLOG (2017) “Creating and Visualizing Neural Networks in R” <https://www.analyticsvidhya.com/blog/2017/09/creating-visualizing-neural-network-in-r/> September 2017. Downloaded 18Jun2018.

Soteris A. Kalogirou S. A and Bojic M. (1999) “Artificial neural networks for the prediction of the energy consumption of a passive solar building” Energy 25 Elsevier.

Kobacoff R.I. (2015) “ R in Action Data Analysis and Graphics with R” Manning Second Edition