Part 1: Algorithm Implementation

* 1. **Data Wrangling and cleansing**

Packages or Libraries used:

* dplyr

It provides a flexible grammar of data manipulation. It's the next iteration of plyr, focused on tools for working with data frames (hence the *d* in the name).

In our script we have used below functions of dplyr

**filter** to select a subset of rows in a data frame, (channel ‘MILDRED SCHOOL 1’)

**arrange** to reorder the columns in our dataframe

**mutate** to add new columns based on the functions of existing columns (hour, Date)

**summerise** to collapse a data frame to a single row

* tidyr

It uses to tidy your data.

We have used,

**gather** takes multiple columns and collapses into key-value pairs, duplicating all other columns as needed. (to combine all the minutes columns)

**separate** turns a single character column into multiple columns (Split Date into Month, day and year)

* stringr

It providing a clean, modern interface to common string operations.

**str\_sub** will recycle all arguments to be the same length as the longest argument.

* lubridate

Functions to work with date-times and time-spans.

Used:

hour, month, day, year

* weatherData

Functions that help in fetching weather data from websites. Given a location and a date range, these functions help fetch weather data (temperature, pressure etc.) for any weather related analysis

We have used to get the weather conditions for each day and each hour of year 2014.

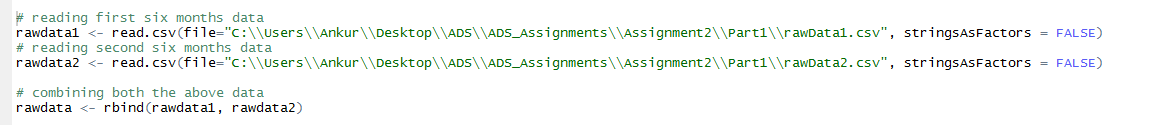
* zoo

It aims at performing calculations containing irregular time series of numeric vectors, matrices & factors

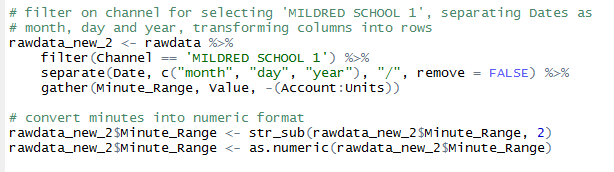
Usage:

na.locf() Generic function for replacing each NA with the most recent non-NA prior to it.

**R-Script:**

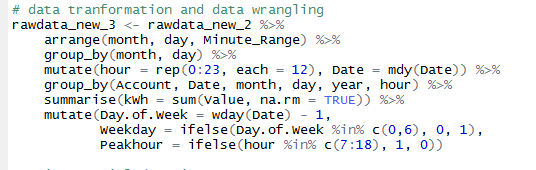


In this part of code, we have taken data from two CSV files ‘rowData1’ and ‘rowData2’ and stored them in two different data frames. Then, we merged them to one single file ‘rawdata’



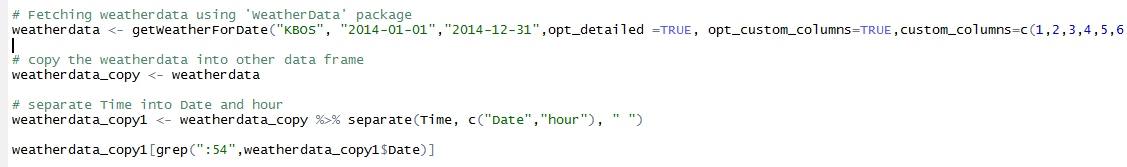
Here, we are filtering our data frame for channel ‘MILDRED SCHOOL 1’, and separate the Date into month, day and year. And gather will transform the each column value to row value and assign the corresponding value of Kwh(Value).

Then, we have modified the Minute\_Range column to make it standardized format and then convert it into numeric value.



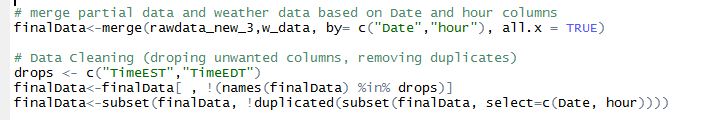
Here, we modified the data to match the format as of sampleformat.csv by using different functions together with the help of pipeline which Passes object on lef hand side as first argument (or . argument) of function on right-hand side.

Arranging the columns month, day and Minute-Range, then use group by using month and day to get the data based on month and year format. Mutate to denote hour as 0 to 23, and there are 12 intervals of 5 minutes, that’s why each is 12. Summerise will sum all the values of kWh and assigned them to a column named ‘kWh’. At the end, to find Day of week, week day and Peakhour based on the criteria given in the problem statement.

e

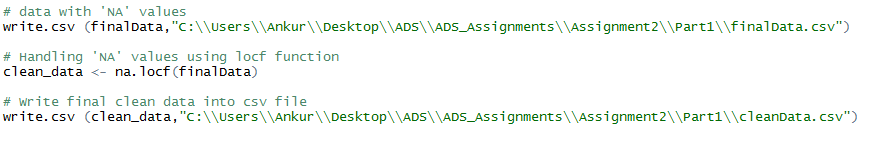
Here, we are getting weatherData using getWeatherForDate() function. And then we are separting the Time to Date and hour.

Mergin and Cleansing:



Merging the data from rawData and WeatherData based on common columns(Date and hour). After that, we have drop the unwanted columns and then removed the duplicates as a part of data cleansing process.

**Handing NA values and final output:**



Here, we are handing NA values in our dataset using na.locf() function of zoo package. And at the end, we are writing our clean data set to CSV file in required format.

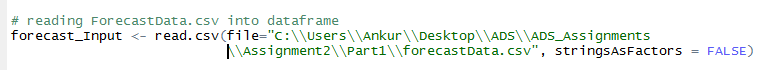
* 1. **Forecast**

Packages or Libraries used:

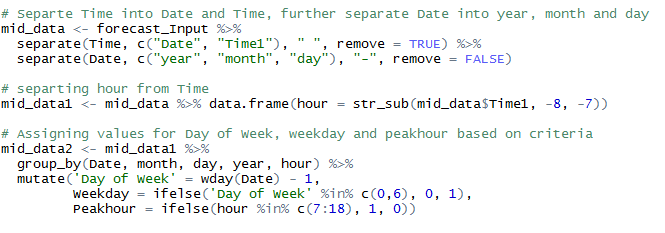
1. dplyr
2. tidyr
3. stringr
4. lubridate
5. chron
6. forecast

**R-Script:**

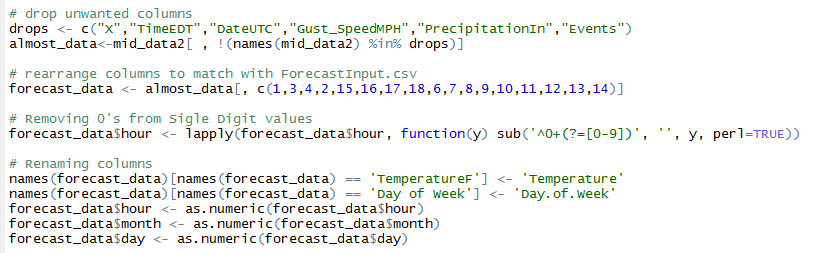
Reading data from input file:



Data handling and formatting:

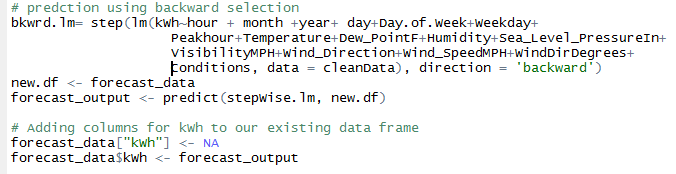


We are separating the Time in Date and Time. And again, separating Date in year, month and day as required in the sample format. We need to separate hour from time. And assign the values of Day of Week, weekday and Peakhour based on the criteria.



Dropping the unwanted columns, and rearranging the columns as needed. We are renaming the columns to match the column names.

Predicting the kWh:



Here, we are using backward selection (regression model) to predict the kWh values for the input file.