**Report**

**on**

**Assignment 2: Energy Forecasting**

* Krutika Dedhia
* Kinjal Gada
* Ankur Vora

**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

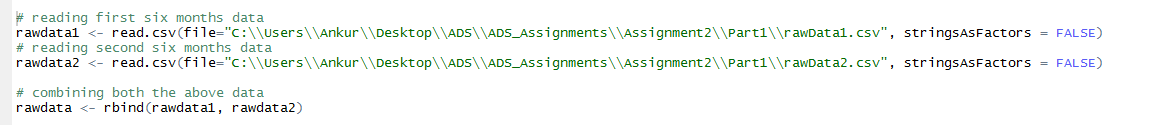
We have used to get the weather conditions for each day and each hour of year 2014.

* zoo

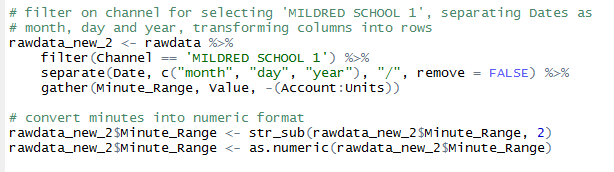
It aims at performing calculations containing irregular time series of numeric vectors, matrices & factors

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

**R-Script:**

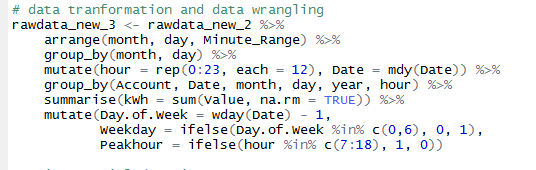


In this part of code, we have taken data from two CSV files ‘rowData1’ and ‘rowData2’ and stored them in two different data frames. Then, we merged them to one single file ‘rawdata’



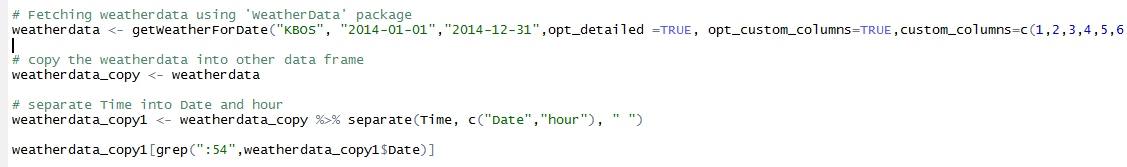
Here, we are filtering our data frame for channel ‘MILDRED SCHOOL 1’, and separate the Date into month, day and year. And gather will transform the each column value to row value and assign the corresponding value of Kwh(Value).

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



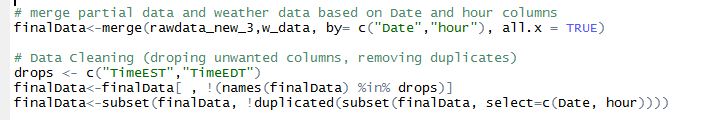
Here, we modified the data to match the format as of sampleformat.csv by using different functions together with the help of pipeline which Passes object on lef hand side as first argument (or . argument) of function on right-hand side.

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.



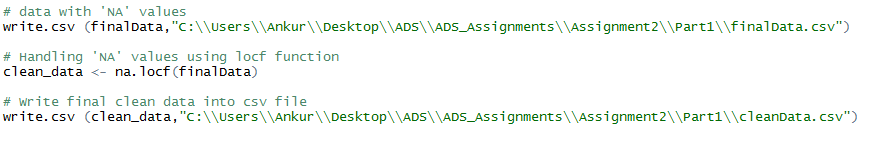
Here, we are getting weatherData using getWeatherForDate() function. And then we are separting the Time to Date and hour.

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. **Multiple – Linear Regression**

**Libraries or Packages used:**

* Forecast – to predict the values of a dependent variable
* data.table – for fast aggregation of data
* broom – used tidy function to clean the messy dataframe

**3 methods of feature selection for multiple linear regression used:**

* Backward – Start with all the variables and discard the ones that produce lower AIC values since we want to minimize the AIC.

Step: AIC=73870.78

kWh ~ hour + month + day + Day.of.Week + Weekday + Peakhour +

Temperature + Dew\_PointF + Humidity + Wind\_Direction + Conditions

Df Sum of Sq RSS AIC

<none> 39786754 73871

- hour 1 124490 39911245 73896

- Temperature 1 132259 39919013 73898

- day 1 236203 40022957 73921

- Dew\_PointF 1 287204 40073959 73932

- Wind\_Direction 17 474446 40261200 73941

- Humidity 1 391393 40178148 73955

- month 1 402943 40189698 73957

- Conditions 24 615213 40401968 73957

- Day.of.Week 1 598634 40385389 74000

- Weekday 1 9681792 49468547 75777

- Peakhour 1 26538740 66325494 78345

Residual standard error: 67.59 on 8709 degrees of freedom

Multiple R-squared: 0.596, Adjusted R-squared: 0.5937

F-statistic: 257 on 50 and 8709 DF, p-value: < 2.2e-16

Coefficients:

(Intercept) as.factor(hour)1

80.5284 0.6941

as.factor(hour)2 as.factor(hour)3

1.4657 2.9962

as.factor(hour)4 as.factor(hour)5

11.3519 15.5325

as.factor(hour)6 as.factor(hour)7

100.6369 157.5196

as.factor(hour)8 as.factor(hour)9

172.8060 178.6468

as.factor(hour)10 as.factor(hour)11

179.9446 179.7130

as.factor(hour)12 as.factor(hour)13

176.4410 170.5849

as.factor(hour)14 as.factor(hour)15

149.9721 126.8434

as.factor(hour)16 as.factor(hour)17

136.7477 109.1930

as.factor(hour)18 as.factor(hour)19

73.1478 64.6890

as.factor(hour)20 as.factor(hour)21

59.1818 8.1477

as.factor(hour)22 as.factor(hour)23

0.3421 -1.3985

month day

-2.3036 -0.5938

Day.of.Week Weekday

4.0607 74.6673

Peakhour Temperature

NA -1.7752

Dew\_PointF Humidity

3.2058 -1.6951

Wind\_DirectionEast Wind\_DirectionENE

6.5612 16.2454

Wind\_DirectionESE Wind\_DirectionNE

8.8144 -4.8083

Wind\_DirectionNNE Wind\_DirectionNNW

-1.9571 -0.3260

Wind\_DirectionNorth Wind\_DirectionNW

3.8552 -9.1155

Wind\_DirectionSE Wind\_DirectionSouth

4.2138 5.7919

Wind\_DirectionSSE Wind\_DirectionSSW

-7.9263 -0.8535

Wind\_DirectionSW Wind\_DirectionVariable

6.8153 -0.7858

Wind\_DirectionWest Wind\_DirectionWNW

-13.5349 -1.7597

Wind\_DirectionWSW ConditionsClear

-6.0664 60.1494

ConditionsDrizzle ConditionsFog

65.0482 65.6296

ConditionsHaze ConditionsHeavy Rain

40.2486 52.6317

ConditionsHeavy Snow ConditionsHeavy Thunderstorms and Rain

54.5110 155.8725

ConditionsIce Pellets ConditionsLight Drizzle

34.1049 60.9055

ConditionsLight Freezing Drizzle ConditionsLight Freezing Rain

81.3563 97.9023

ConditionsLight Ice Pellets ConditionsLight Rain

62.5055 62.1334

ConditionsLight Snow ConditionsLight Thunderstorms and Rain

82.7391 145.4961

ConditionsMostly Cloudy ConditionsOvercast

57.9975 59.4064

ConditionsPartly Cloudy ConditionsPatches of Fog

65.8524 198.1927

ConditionsRain ConditionsScattered Clouds

66.8487 69.5872

ConditionsSnow ConditionsThunderstorm

67.0262 197.2734

ConditionsThunderstorms and Rain

56.2502

* **Forward – Start with minimum 1 variable and keep on adding variables that produce lower AIC values.**

Step: AIC=73870.78

kWh ~ Peakhour + Weekday + Temperature + month + Day.of.Week +

Conditions + Wind\_Direction + day + Humidity + Dew\_PointF +

hour

Df Sum of Sq RSS AIC

<none> 39786754 73871

+ VisibilityMPH 1 2383 39784372 73872

+ Sea\_Level\_PressureIn 1 1813 39784941 73872

+ WindDirDegrees 1 28 39786727 73873

+ Wind\_SpeedMPH 32 221443 39565311 73886

Residual standard error: 67.59 on 8709 degrees of freedom

Multiple R-squared: 0.596, Adjusted R-squared: 0.5937

F-statistic: 257 on 50 and 8709 DF, p-value: < 2.2e-16

Coefficients:

(Intercept) Peakhour Weekday

178.4004 126.6474 74.3307

Temperature month Day.of.Week

-2.6181 -2.2292 4.1852

ConditionsClear ConditionsDrizzle ConditionsFog

19.8171 19.3622 34.6300

ConditionsHaze ConditionsHeavy Rain ConditionsHeavy Snow

-7.5979 34.3505 43.2777

ConditionsHeavy Thunderstorms and Rain ConditionsIce Pellets ConditionsLight Drizzle

158.6011 14.0986 45.8223

ConditionsLight Freezing Drizzle ConditionsLight Freezing Rain ConditionsLight Ice Pellets

15.8951 44.9395 26.7241

ConditionsLight Rain ConditionsLight Snow ConditionsLight Thunderstorms and Rain

34.5778 56.4534 101.7142

ConditionsMostly Cloudy ConditionsOvercast ConditionsPartly Cloudy

29.3369 27.5544 37.0115

ConditionsPatches of Fog ConditionsRain ConditionsScattered Clouds

146.2437 35.1243 42.3104

ConditionsSnow ConditionsThunderstorm ConditionsThunderstorms and Rain

40.8107 145.7871 -2.0269

Wind\_DirectionEast Wind\_DirectionENE Wind\_DirectionESE

11.0668 22.1808 6.9126

Wind\_DirectionNE Wind\_DirectionNNE Wind\_DirectionNNW

3.0068 5.5635 2.0486

Wind\_DirectionNorth Wind\_DirectionNW Wind\_DirectionSE

5.0412 -4.4780 10.7987

Wind\_DirectionSouth Wind\_DirectionSSE Wind\_DirectionSSW

10.1296 -9.3669 -0.5943

Wind\_DirectionSW Wind\_DirectionVariable Wind\_DirectionWest

8.0619 11.7741 -11.8156

Wind\_DirectionWNW Wind\_DirectionWSW day

-1.8213 -3.1780 -0.5961

Humidity Dew\_PointF hour

-2.1927 4.1036 -0.5710

* **Stepwise – It is the combination if both forward and backward**

Step: AIC=73870.78

kWh ~ hour + month + day + Day.of.Week + Weekday + Peakhour +

Temperature + Dew\_PointF + Humidity + Wind\_Direction + Conditions

Df Sum of Sq RSS AIC

<none> 39786754 73871

+ VisibilityMPH 1 2383 39784372 73872

+ Sea\_Level\_PressureIn 1 1813 39784941 73872

+ WindDirDegrees 1 28 39786727 73873

+ Wind\_SpeedMPH 32 221443 39565311 73886

- hour 1 124490 39911245 73896

- Temperature 1 132259 39919013 73898

- day 1 236203 40022957 73921

- Dew\_PointF 1 287204 40073959 73932

- Wind\_Direction 17 474446 40261200 73941

- Humidity 1 391393 40178148 73955

- month 1 402943 40189698 73957

- Conditions 24 615213 40401968 73957

- Day.of.Week 1 598634 40385389 74000

- Weekday 1 9681792 49468547 75777

- Peakhour 1 26538740 66325494 78345

Residual standard error: 67.59 on 8709 degrees of freedom

Multiple R-squared: 0.596, Adjusted R-squared: 0.5937

F-statistic: 257 on 50 and 8709 DF, p-value: < 2.2e-16

Coefficients:

(Intercept) hour month

178.4004 -0.5710 -2.2292

day Day.of.Week Weekday

-0.5961 4.1852 74.3307

Peakhour Temperature Dew\_PointF

126.6474 -2.6181 4.1036

Humidity Wind\_DirectionEast Wind\_DirectionENE

-2.1927 11.0668 22.1808

Wind\_DirectionESE Wind\_DirectionNE Wind\_DirectionNNE

6.9126 3.0068 5.5635

Wind\_DirectionNNW Wind\_DirectionNorth Wind\_DirectionNW

2.0486 5.0412 -4.4780

Wind\_DirectionSE Wind\_DirectionSouth Wind\_DirectionSSE

10.7987 10.1296 -9.3669

Wind\_DirectionSSW Wind\_DirectionSW Wind\_DirectionVariable

-0.5943 8.0619 11.7741

Wind\_DirectionWest Wind\_DirectionWNW Wind\_DirectionWSW

-11.8156 -1.8213 -3.1780

ConditionsClear ConditionsDrizzle ConditionsFog

19.8171 19.3622 34.6300

ConditionsHaze ConditionsHeavy Rain ConditionsHeavy Snow

-7.5979 34.3505 43.2777

ConditionsHeavy Thunderstorms and Rain ConditionsIce Pellets ConditionsLight Drizzle

158.6011 14.0986 45.8223

ConditionsLight Freezing Drizzle ConditionsLight Freezing Rain ConditionsLight Ice Pellets

15.8951 44.9395 26.7241

ConditionsLight Rain ConditionsLight Snow ConditionsLight Thunderstorms and Rain

34.5778 56.4534 101.7142

ConditionsMostly Cloudy ConditionsOvercast ConditionsPartly Cloudy

29.3369 27.5544 37.0115

ConditionsPatches of Fog ConditionsRain ConditionsScattered Clouds

146.2437 35.1243 42.3104

ConditionsSnow ConditionsThunderstorm ConditionsThunderstorms and Rain

40.8107 145.7871 -2.0269

* **Feature and Model Selection:**

The **Akaike information criterion (AIC)** is a measure of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Hence, AIC provides a means for model selection.

1. Select the most optimized variables from the method that produces lesser AIC values. Used function lm.fit() to find the R squared values. More the R squared value better is the model.
2. Create the model based on the variables selected.
3. Split the dataset in train and test data.
4. Train the data using the model selected.
5. Predict the forecast accuracy on the test data.

We explored all the methods and selected backward regression. Forward selection has drawbacks, including the fact that each addition of a new variable may render one or more of the already included variables non-significant. In backward selection, one starts with fitting a model with all the variables of interest (following the initial screen). Then the least significant variable is dropped, so long as it is not significant at our chosen critical level. We continue by successively re-fitting reduced models and applying the same rule until all remaining variables are statistically significant.

* **Performance Metrics:**

>accuracy(pred, train$kWh)

Output:

               ME     RMSE      MAE       MPE     MAPE

Test set 3.084447 138.0408 108.0624 -28.55807 67.32483

**Mean Absolute Error (MAE):**

MAE is the average over the verification sample of the absolute values of the differences between forecast and the corresponding observation. The MAE measures the average magnitude of the errors in a set of forecasts, without considering their direction. It measures accuracy for continuous variables.

**Root Mean Square Error (RMSE):**

RMSE is the difference between forecast and corresponding observed values are each squared and then averaged over the sample. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. RMSE is most useful when large errors are particularly undesirable.

The RMSE will always be larger or equal to the MAE. If the RMSE is equal to MAE, then all the errors are of the same magnitude. 138.04 > 108.06

**Mean Absolute Percentage Error (MAPE):**

MAPE is a measure of how high or low are the differences between the predictions and actual data. For e.g. 15% MAPE means on average the predictions from a model will be 15% higher or lower than actual.

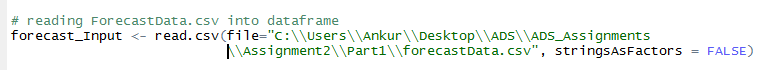
* 1. **Forecast**

Packages or Libraries used:

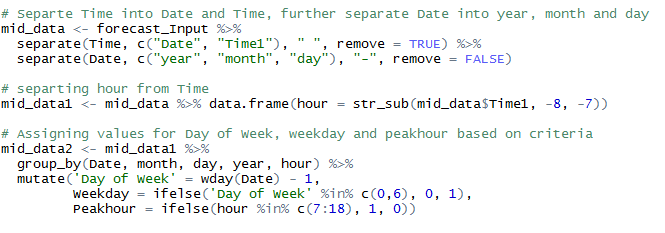
1. dplyr
2. tidyr
3. stringr
4. lubridate
5. chron
6. forecast

**R-Script:**

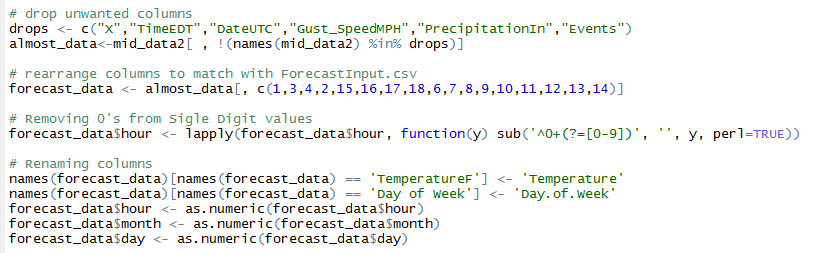
Reading data from input file:



Data handling and formatting:

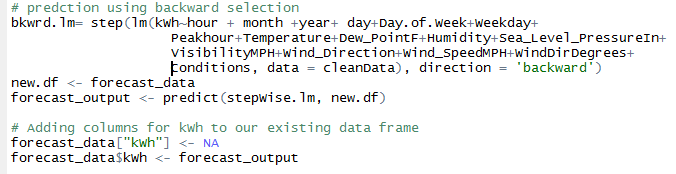


We are separating the Time in Date and Time. And again, separating Date in year, month and day as required in the sample format. We need to separate hour from time. And assign the values of Day of Week, weekday and Peakhour based on the criteria.



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.

**Flowchart**

Data Wrangling

Data Merging

Data Gathering

Raw Data

Weather Data

Data Gathering

Merging Raw and weather data based on Date and hour columns

Feature Seletion

Data cleansing and removing Duplicate rows

handling NA values using LOCF Package

Model Selection

Split into Train and Test Data

Forecast kWh with developed Model

Evaluate Performance Matrix

Lasso Implementation

Forecast the prediction with test data

Model the train data