Weather Extraction

Sameer, Lalit, Lipsa

November 18, 2016

#### Data from weather API

We have used the WeatherData Package to pull all the weather related information from wunderground.com.The weatherData package takes a date range and Location as an input .

We first calculated the minium and max date for our observed dataset

mindate <- min(aggData.long$Date)  
maxdate <- max(aggData.long$Date)

Converted the date range in to a desired format

mindate <- as.Date(mindate, "%m/%d/%Y")  
maxdate <- as.Date(maxdate, "%m/%d/%Y")

We got the station code for Boston

getStationCode("Boston")

## [[1]]  
## Station State airportCode  
## 656 Boston MA KBOS  
##   
## [[2]]  
## [1] "USA MA BOSTON KBOS BOS 72509 42 22N 071 01W 6 X U A 0 US"  
## [2] "USA MA BOSTON/TAUNTON KBOX BOX 41 57N 071 08W 36 X F 8 US"  
## [3] "USA MA BOSTON/RFC KTAR TAR 41 57N 071 08W 36 R 8 US"

The station\_id for Boston is "KBOS"

* getWeatherForDate(): Getting data for a range of dates, it has certain parameters
* station\_id: is a valid 3- or 4-letter Airport code or a valid Weather Station ID (example: "KBOS" for Boston).
* start\_date: string representing a date in the past ("YYYY-MM-DD", all numeric)
* end\_date : If an interval is to be specified,end\_date is a strin grepresenting a date in the past ("YYYY-MM-DD", all numeric) and greater than the start date
* opt\_detailed:indicates if detailed records for the station are desired. (default FALSE). By default only one records per date is returned.
* opt\_custom\_columns: to indicate if only a user-specified set of columns are to be returned. (default FALSE) If TRUE, then the desired columns must be specified via custom\_columns
* custom\_columns: Vector of integers specified by the user to indicate which columns to fetch. The Date column is always returned as the first column.

Once we fetched the respective inputs for the WeatherData .We tried to extract the weather information witht the applied inputs.

WeatherData <- getWeatherForDate("KBOS", start\_date=mindate,  
 end\_date = maxdate,  
 opt\_detailed=T,opt\_custom\_columns=T,  
 custom\_columns=c(2:13))

## [1] "TimeEST" "TemperatureF" "Dew\_PointF"   
## [4] "Humidity" "Sea\_Level\_PressureIn" "VisibilityMPH"   
## [7] "Wind\_Direction" "Wind\_SpeedMPH" "Gust\_SpeedMPH"   
## [10] "PrecipitationIn" "Events" "Conditions"   
## [13] "WindDirDegrees" "DateUTC"   
## [1] "TimeEST" "TemperatureF" "Dew\_PointF"   
## [4] "Humidity" "Sea\_Level\_PressureIn" "VisibilityMPH"   
## [7] "Wind\_Direction" "Wind\_SpeedMPH" "Gust\_SpeedMPH"   
## [10] "PrecipitationIn" "Events" "Conditions"   
## [13] "WindDirDegrees" "DateUTC"   
## [1] "Time" "TemperatureF" "Dew\_PointF"   
## [4] "Humidity" "Sea\_Level\_PressureIn" "VisibilityMPH"   
## [7] "Wind\_Direction" "Wind\_SpeedMPH" "Gust\_SpeedMPH"   
## [10] "PrecipitationIn" "Events" "Conditions"   
## [13] "WindDirDegrees"

head(WeatherData)

## Time TemperatureF Dew\_PointF Humidity  
## 1 2014-01-01 00:54:00 23.0 5.0 46  
## 2 2014-01-01 01:54:00 21.9 3.9 46  
## 3 2014-01-01 02:54:00 21.9 3.9 46  
## 4 2014-01-01 03:54:00 21.9 3.0 44  
## 5 2014-01-01 04:54:00 21.0 3.0 46  
## 6 2014-01-01 05:54:00 21.0 3.0 46  
## Sea\_Level\_PressureIn VisibilityMPH Wind\_Direction Wind\_SpeedMPH  
## 1 30.20 10 WNW 8.1  
## 2 30.23 10 WNW 11.5  
## 3 30.25 10 WSW 12.7  
## 4 30.27 10 WSW 11.5  
## 5 30.29 10 West 9.2  
## 6 30.30 10 West 11.5  
## Gust\_SpeedMPH PrecipitationIn Events Conditions WindDirDegrees  
## 1 - N/A <NA> Clear 290  
## 2 - N/A <NA> Partly Cloudy 290  
## 3 - N/A <NA> Clear 240  
## 4 19.6 N/A <NA> Clear 250  
## 5 - N/A <NA> Clear 260  
## 6 20.7 N/A <NA> Clear 270

We calculated the date and hour using the "Lubricate" package we have used.

WeatherData$date = date(WeatherData$Time)  
WeatherData$hour = hour(WeatherData$Time)

After looking in to the information pulled by the WeatherData package ,we got a picture that data is spread on hourly interval. We tried to confirm with the following function.

head(table(WeatherData$date))

##   
## 2014-01-01 2014-01-02 2014-01-03 2014-01-04 2014-01-05 2014-01-06   
## 24 54 31 24 36 46

After looking at the tabular values, we deduced that although most of the days had 24 observations, some of them have more than 24 .

The details revealed that in some instances observations were taken more than once for each hour,as illustrated in the following case :

View(WeatherData[which(WeatherData$date == "2014-06-05"),])

Detail Observation :

* we got -999999 value in columns TempratureF, DewPointF, Sea\_Level\_PressureIn, Visibility MPH
* We converted the data to the respective data types
* WindSpeed "Calm" which mean 0: Converting to character as it is in factor

WeatherData$date <- as.Date(WeatherData$date,"%m/%d/%Y")  
WeatherData$TemperatureF <- as.numeric(WeatherData$TemperatureF)  
WeatherData$Dew\_PointF <- as.numeric(WeatherData$Dew\_PointF)  
WeatherData$Sea\_Level\_PressureIn <- as.numeric(WeatherData$Sea\_Level\_PressureIn)  
WeatherData$VisibilityMPH <- as.numeric(WeatherData$VisibilityMPH)  
WeatherData$WindDirDegrees <- as.numeric(WeatherData$WindDirDegrees)  
WeatherData$Humidity <- as.numeric(WeatherData$Humidity)  
  
WeatherData$Wind\_SpeedMPH[WeatherData$Wind\_SpeedMPH == "Calm"] <- 0  
WeatherData$Wind\_SpeedMPH <- as.numeric(WeatherData$Wind\_SpeedMPH)

We need our data to fall in normal range to remove outliers

### Handling Outliers

* We used the approach of subsituting the previous or the next value of the observation. For example, if the record 8999 has Temperature as -9999 we used the record of 8998 so that this is still acceptable.
* We tried to handle the outliers with the following function

remove\_out <- function(param,index,min\_v,max\_v)  
{  
 val = NULL  
 val = param[index]  
   
 if(val < min\_v | val > max\_v | is.na(val)){  
 if(index-1 >= 1){  
 val = param[index-1]  
 } else if (index-1 <= 0){  
 val = param[index+1]  
 }   
 return(val)  
 } else{  
 print("Nothing changed")  
 return(val) #Normal Value return  
 }  
}

With the above function removed the outliers for Temperature. We found out the records where Temperature is falling out of the range defined in the table

Temperature

index <- which(WeatherData$TemperatureF < 0 | WeatherData$TemperatureF > 100 | is.na(WeatherData$Dew\_PointF))  
print(index)

## [1] 8206

We had an insight in to the data records WeatherData[8206,]

We found that it was indeed an outlier,could be a machine input error. We tried to remove this implementing the function and checked the record again after the function

for (i in index){  
WeatherData$TemperatureF[i] = remove\_out(WeatherData$TemperatureF,i,0,100)  
}  
WeatherData[8206,]

## Time TemperatureF Dew\_PointF Humidity  
## 8206 2014-10-17 04:07:00 60.8 -9999 NA  
## Sea\_Level\_PressureIn VisibilityMPH Wind\_Direction Wind\_SpeedMPH  
## 8206 -9999 -9999 SW 8.1  
## Gust\_SpeedMPH PrecipitationIn Events Conditions WindDirDegrees  
## 8206 - N/A <NA> Mostly Cloudy 220  
## date hour  
## 8206 2014-10-17 4

We were successful in getting in to shape.We implemtened the same thing for the other features:

Dew Point

index <- which(WeatherData$Dew\_PointF < -20 | WeatherData$Dew\_PointF > 80 | is.na(WeatherData$Dew\_PointF))  
for (i in index){  
 WeatherData$Dew\_PointF[i] = remove\_out(WeatherData$Dew\_PointF,i,-20,80)  
}

Humidity

index <- which(WeatherData$Humidity < 10 | WeatherData$Humidity > 100 | is.na(WeatherData$Humidity))  
for (i in index){  
 WeatherData$Humidity[i] = remove\_out(WeatherData$Humidity,i,10,100)  
}

Wind\_SpeedMPH

index <- which(WeatherData$Wind\_SpeedMPH < 0 | WeatherData$Wind\_SpeedMPH > 50 | is.na(WeatherData$Wind\_SpeedMPH))  
for (i in index){  
 WeatherData$Wind\_SpeedMPH[i] = remove\_out(WeatherData$Wind\_SpeedMPH,i,0,50)  
}

Sea\_Level\_Pressure

index <- which(WeatherData$Sea\_Level\_PressureIn < 28 | WeatherData$Sea\_Level\_PressureIn > 32 | is.na(WeatherData$Sea\_Level\_PressureIn))  
for (i in index){  
 WeatherData$Sea\_Level\_PressureIn[i] = remove\_out(WeatherData$Sea\_Level\_PressureIn,i,28,32)  
}

VisibilityMPH

index <- which(WeatherData$VisibilityMPH < 0 | WeatherData$VisibilityMPH > 10 | is.na(WeatherData$VisibilityMPH))  
for (i in index){  
 WeatherData$VisibilityMPH[i] = remove\_out(WeatherData$VisibilityMPH,i,0,10)  
}

WindDirDegree

index <- which(WeatherData$WindDirDegrees < 0 | WeatherData$WindDirDegrees > 360 | is.na(WeatherData$WindDirDegrees))  
for (i in index){  
 WeatherData$WindDirDegrees[i] = remove\_out(WeatherData$WindDirDegrees,i,0,360)  
}

The data was clean and consistent. We aggregated the dataset as done in part 1, so that we get records for each hour and we can take average values for numeric values and frequency count for character values. We followed the below steps: 1) Remove non essential features like Time, Gust\_speedMH,P,E 2) Group the data by Date and hour 3) summarise base on mean and frequency count

WeatherData.Agg <- WeatherData %>%   
 select(-c(Time,Gust\_SpeedMPH,  
 PrecipitationIn,Events)) %>%  
 group\_by(date,hour) %>%   
 summarise(TemperatureF = mean(TemperatureF),  
 Dew\_PointF = mean(Dew\_PointF),  
 Humidity = mean(Humidity),  
 Sea\_Level\_PressureIn = mean(Sea\_Level\_PressureIn),  
 VisibilityMPH = mean (VisibilityMPH),  
 Wind\_SpeedMPH = mean(Wind\_SpeedMPH),  
 WindDirDegrees = mean(WindDirDegrees),  
 Conditions = names(table(Conditions))[which.max(table(Conditions))],  
 Wind\_Direction = names(table(Wind\_Direction))[which.max(table(Wind\_Direction))])  
  
head(WeatherData.Agg)

## Source: local data frame [6 x 11]  
## Groups: date [1]  
##   
## date hour TemperatureF Dew\_PointF Humidity Sea\_Level\_PressureIn  
## <date> <int> <dbl> <dbl> <dbl> <dbl>  
## 1 2014-01-01 0 23.0 5.0 46 30.20  
## 2 2014-01-01 1 21.9 3.9 46 30.23  
## 3 2014-01-01 2 21.9 3.9 46 30.25  
## 4 2014-01-01 3 21.9 3.0 44 30.27  
## 5 2014-01-01 4 21.0 3.0 46 30.29  
## 6 2014-01-01 5 21.0 3.0 46 30.30  
## # ... with 5 more variables: VisibilityMPH <dbl>, Wind\_SpeedMPH <dbl>,  
## # WindDirDegrees <dbl>, Conditions <chr>, Wind\_Direction <chr>

Once We had both the dataset wit in the desired format We merged the data with part 1 of the energy usage data by Date and hour

### Final Output Data

mergeData <- merge(aggData.long,WeatherData.Agg,by.x = c("Date","hour"),by.y = c("date","hour"))  
head(mergeData)

## Date hour Account Channel Units month day year  
## 1 2014-01-01 0 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## 2 2014-01-01 1 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## 3 2014-01-01 10 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## 4 2014-01-01 11 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## 5 2014-01-01 12 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## 6 2014-01-01 13 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## Day of Week weekday Kwh PeakHour TemperatureF Dew\_PointF Humidity  
## 1 4 1 132.37 0 23.0 5.0 46  
## 2 4 1 132.72 0 21.9 3.9 46  
## 3 4 1 129.11 1 26.1 5.0 41  
## 4 4 1 125.83 1 26.1 5.0 41  
## 5 4 1 120.91 1 27.0 5.0 39  
## 6 4 1 125.20 1 28.0 3.9 36  
## Sea\_Level\_PressureIn VisibilityMPH Wind\_SpeedMPH WindDirDegrees  
## 1 30.20 10 8.1 290  
## 2 30.23 10 11.5 290  
## 3 30.35 10 12.7 280  
## 4 30.34 10 13.8 260  
## 5 30.33 10 13.8 280  
## 6 30.33 10 8.1 300  
## Conditions Wind\_Direction  
## 1 Clear WNW  
## 2 Partly Cloudy WNW  
## 3 Partly Cloudy West  
## 4 Mostly Cloudy West  
## 5 Scattered Clouds West  
## 6 Mostly Cloudy WNW

Arranging the data by Date and hour

mergeData<- arrange(mergeData,Date,hour)  
head(mergeData)

## Date hour Account Channel Units month day year  
## 1 2014-01-01 0 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## 2 2014-01-01 1 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## 3 2014-01-01 2 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## 4 2014-01-01 3 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## 5 2014-01-01 4 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## 6 2014-01-01 5 26908650026 MILDRED SCHOOL 1 kWh 1 1 2014  
## Day of Week weekday Kwh PeakHour TemperatureF Dew\_PointF Humidity  
## 1 4 1 132.37 0 23.0 5.0 46  
## 2 4 1 132.72 0 21.9 3.9 46  
## 3 4 1 129.03 0 21.9 3.9 46  
## 4 4 1 125.76 0 21.9 3.0 44  
## 5 4 1 129.39 0 21.0 3.0 46  
## 6 4 1 132.51 0 21.0 3.0 46  
## Sea\_Level\_PressureIn VisibilityMPH Wind\_SpeedMPH WindDirDegrees  
## 1 30.20 10 8.1 290  
## 2 30.23 10 11.5 290  
## 3 30.25 10 12.7 240  
## 4 30.27 10 11.5 250  
## 5 30.29 10 9.2 260  
## 6 30.30 10 11.5 270  
## Conditions Wind\_Direction  
## 1 Clear WNW  
## 2 Partly Cloudy WNW  
## 3 Clear WSW  
## 4 Clear WSW  
## 5 Clear West  
## 6 Clear West

We write this output to csv file

write.csv(mergeData,"MergedData.csv")

Now we have the clean data to start with our model