Deep Analytics on Power Consumption Data Set

Rayon Susan Koshy

In this project we are doing domain research and exploratory data analysis ona electric power consumption data. The electric power consumption data set that we used in the project can be downloaded from the UC Irvine Machine Learning Repository:

http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption

We can use RMySQL package to connect to the database. Here we have combined the data set to a single data frame named as "newDf"

Understanding the Data Set

```
str(newDf)
```

```
## 'data.frame':    2027288 obs. of  6 variables:
##  $ Date               : chr  "2007-01-01" "2007-01-01" "2007-01-01" "2007-01-01" ...
##  $ Time               : chr  "00:00:00" "00:01:00" "00:02:00" "00:03:00" ...
##  $ Global_active_power: num  2.58 2.55 2.55 2.55 2.55 ...
##  $ Sub_metering_1     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sub_metering_2     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sub_metering_3     : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
summary(newDf)
```

```
##      Date               Time           Global_active_power Sub_metering_1
##  Length:2027288     Length:2027288     Min.   : 0.076      Min.   : 0.000
##  Class :character   Class :character   1st Qu.: 0.308      1st Qu.: 0.000
##  Mode  :character   Mode  :character   Median : 0.594      Median : 0.000
##                                        Mean   : 1.083      Mean   : 1.121
##                                        3rd Qu.: 1.520      3rd Qu.: 0.000
##                                        Max.   :11.122      Max.   :88.000
##  Sub_metering_2   Sub_metering_3
##  Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.: 0.000
##  Median : 0.000   Median : 1.000
##  Mean   : 1.289   Mean   : 6.448
##  3rd Qu.: 1.000   3rd Qu.:17.000
##  Max.   :80.000   Max.   :31.000
```

```
head(newDf)
```

```
##         Date     Time Global_active_power Sub_metering_1 Sub_metering_2
## 1 2007-01-01 00:00:00               2.580              0              0
## 2 2007-01-01 00:01:00               2.552              0              0
## 3 2007-01-01 00:02:00               2.550              0              0
## 4 2007-01-01 00:03:00               2.550              0              0
## 5 2007-01-01 00:04:00               2.554              0              0
```

```
## 6 2007-01-01 00:05:00              2.550              0              0
##   Sub_metering_3
## 1              0
## 2              0
## 3              0
## 4              0
## 5              0
## 6              0
```

Let us perform some data wrangling, which sometimes referred as data munching on our data set.

Since the Date and Time columns are separate they will need to be combined within the dataset in order to convert them to the correct format to complete the appropriate analysis

```
#Combine Date and Time attribute values
#in a new attribute column
newDf <- cbind(newDf,paste(newDf$Date,newDf$Time),
               stringsAsFactors =  FALSE)
ncol(newDf)
```

```
## [1] 7
```

```
colnames(newDf)[7] <- "DateTime"
```

After creating the new attribute lets us convert it to a DateTime data type called POSIXct. After converting to POSIXct we will add the time zone to prevent warning messages. The data description suggests that the data is from France.

```
#*You will now want to convert the new DateTime
#*attribute to a DateTime data type called POSIXct.
#*After converting to POSIXct we will add the
#*time zone to prevent warning messages.

newDf$DateTime <- as.POSIXct(newDf$DateTime,
                             "%Y/%m/%d %H:%M:%S")
attr(newDf$DateTime,"tzone") <- "Europe/Paris"
```

Now it is time to take a look at our data set to see all the attributes.

```
str(newDf)
```

```
## 'data.frame':    2027288 obs. of  7 variables:
##  $ Date               : chr  "2007-01-01" "2007-01-01" "2007-01-01" "2007-01-01" ...
##  $ Time               : chr  "00:00:00" "00:01:00" "00:02:00" "00:03:00" ...
##  $ Global_active_power: num  2.58 2.55 2.55 2.55 2.55 ...
##  $ Sub_metering_1     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sub_metering_2     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sub_metering_3     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ DateTime           : POSIXct, format: "2007-01-01 01:00:00" "2007-01-01 01:01:00" ...
```

Lubridate is one of the package makes working with DateTime much easier.One of the package's capabilities is extracting DateTime information into individual attributes like Year and Month. These attributes can be used by dplyr's filter command to subset the data into useful data sets for visualization.

```
#adding new column year, season
newDf$Year <- year(newDf$DateTime)
newDf$Month <- month(newDf$DateTime)
newDf$Hour <- hour(newDf$DateTime)
newDf$JustDate <- mday(newDf$DateTime)
```

In addition to the extraction of the attributes let us create a new attribute Season, which corresponds to each season of the year and Day which corresponds to the time of day like Morning,Afternoon,Evening & Night.

```
##               Date     Time Global_active_power Sub_metering_1 Sub_metering_2
## 2027283 2010-11-26 20:57:00               0.946              0              0
## 2027284 2010-11-26 20:58:00               0.946              0              0
## 2027285 2010-11-26 20:59:00               0.944              0              0
## 2027286 2010-11-26 21:00:00               0.938              0              0
## 2027287 2010-11-26 21:01:00               0.934              0              0
## 2027288 2010-11-26 21:02:00               0.932              0              0
##         Sub_metering_3            DateTime Year Month Hour JustDate Season
## 2027283              0 2010-11-26 21:57:00 2010    11   21       26   Fall
## 2027284              0 2010-11-26 21:58:00 2010    11   21       26   Fall
## 2027285              0 2010-11-26 21:59:00 2010    11   21       26   Fall
## 2027286              0 2010-11-26 22:00:00 2010    11   22       26   Fall
## 2027287              0 2010-11-26 22:01:00 2010    11   22       26   Fall
## 2027288              0 2010-11-26 22:02:00 2010    11   22       26   Fall
##             Day
## 2027283 Evening
## 2027284 Evening
## 2027285 Evening
## 2027286 Evening
## 2027287 Evening
## 2027288 Evening
```

```
## 'data.frame':    2027288 obs. of  13 variables:
##  $ Date               : chr  "2007-01-01" "2007-01-01" "2007-01-01" "2007-01-01" ...
##  $ Time               : chr  "00:00:00" "00:01:00" "00:02:00" "00:03:00" ...
##  $ Global_active_power: num  2.58 2.55 2.55 2.55 2.55 ...
##  $ Sub_metering_1     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sub_metering_2     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sub_metering_3     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ DateTime           : POSIXct, format: "2007-01-01 01:00:00" "2007-01-01 01:01:00" ...
##  $ Year               : num  2007 2007 2007 2007 2007 ...
##  $ Month              : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Hour               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ JustDate           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Season             : Factor w/ 4 levels "Winter","Spring",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Day                : Factor w/ 4 levels "Night","Morning",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Exploratory Data Analysis

The below gives you total power consumed by of the Sub-meters from the year 2007 to 2010

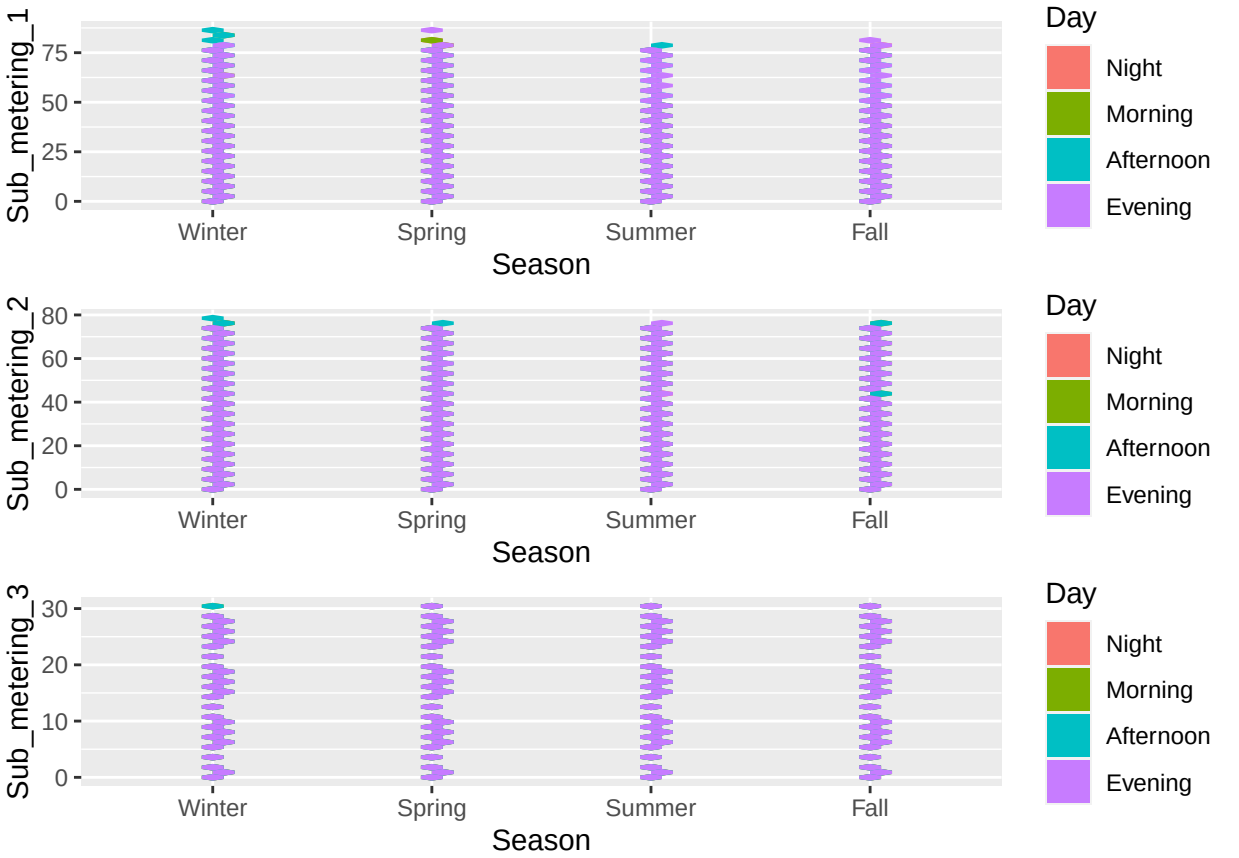
```
#year
plot1 <- ggplot(data = newDf,
       aes(x=Year,y=Sub_metering_1))+
  geom_bar(stat = "identity",fill = "steelblue")+
  scale_y_continuous(labels = comma)
plot2 <- ggplot(data = newDf,
                aes(x=Year,y=Sub_metering_2))+
  geom_bar(stat = "identity", fill = "steelblue")+
  scale_y_continuous(labels = comma)
plot3 <- ggplot(data = newDf,
                aes(x=Year,y=Sub_metering_3))+
  geom_bar(stat = "identity", fill = "steelblue")+
```

```
  scale_y_continuous(labels = comma)

ggarrange(plot1,plot2,plot3,
          ncol = 1, nrow = 3)
```

From the above we can come to the below points: - Energy usage by the Sub_metering_3 appliances are comparatively high while it is low for Sub_metering_1 appliances - For both Sub_metering_1 & Sub_metering_2, 2007 has the highest consumption while it is less for 2010. - There is gradual reduction energy consumption for both Sub_metering_1 & Sub_metering_2 - In case Sub_metering_3, high energy usage is during the year 2009

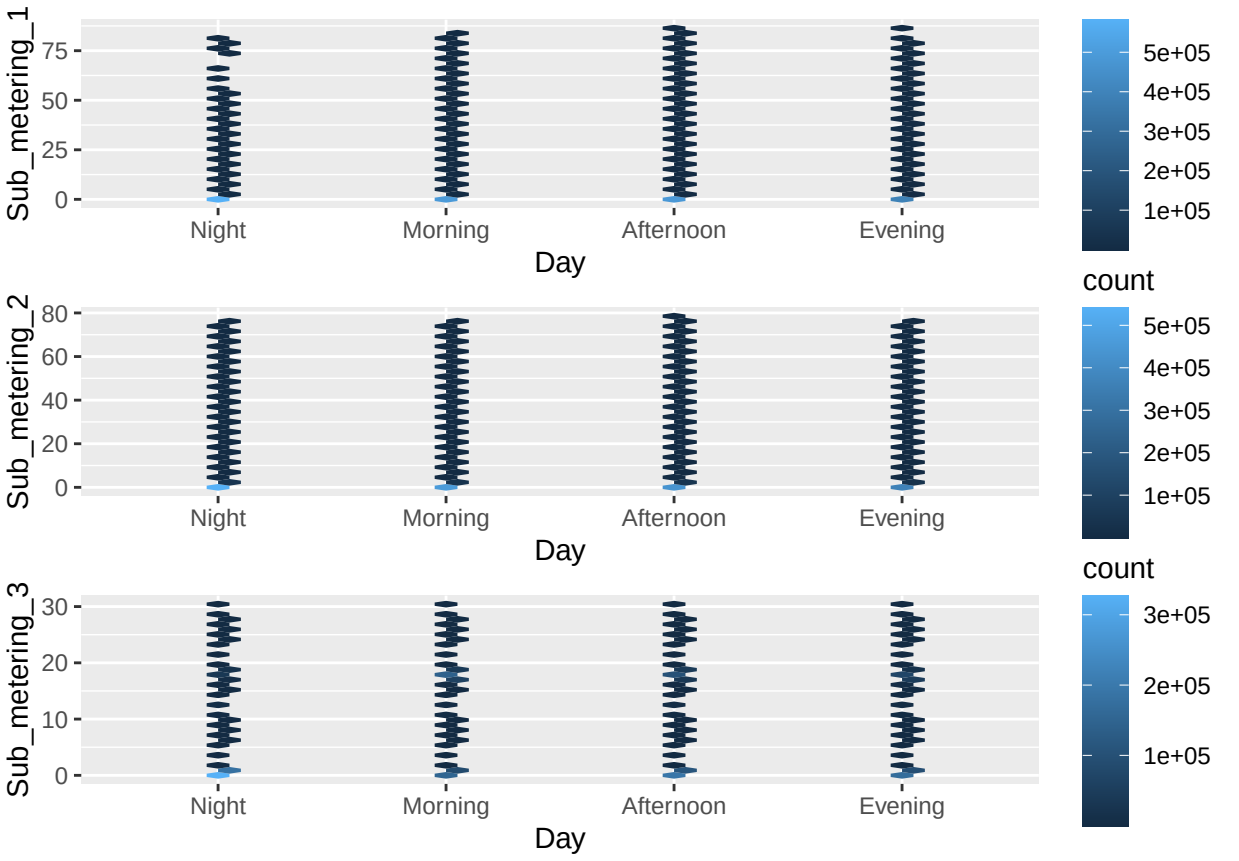
Let us try plotting the power usage of each submeters during each season & see how much power is consumed by each time of the day. To do so, first us try using geom_hex.

```
#Season
plot1 <- ggplot(data = newDf)+
  geom_hex(mapping = aes(x=Season,y=Sub_metering_1,fill = Day))
plot2 <- ggplot(data = newDf)+
  geom_hex(mapping = aes(x=Season,y=Sub_metering_2,fill = Day))
plot3 <- ggplot(data = newDf)+
  geom_hex(mapping = aes(x=Season,y=Sub_metering_3,fill = Day))

ggarrange(plot1,plot2,plot3,
          ncol = 1, nrow = 3)
```



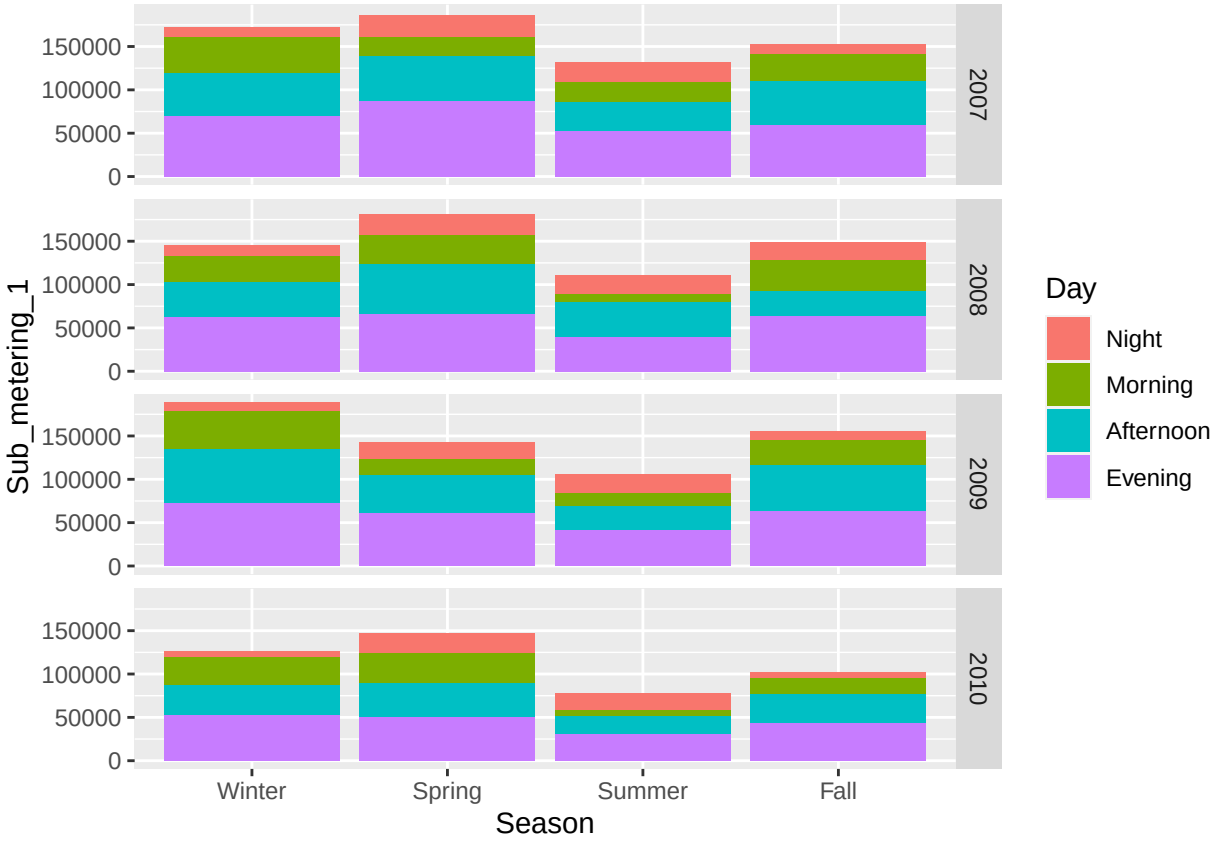
Geom_hex divides the plot plane into regular hexagons and counts the number of cases in each hexagon.Count gives us number of points in bin. The below shows you the plot for each submeter's usage during each time of the day.

```
#Time of the day
plot1 <- ggplot(data = newDf,
                aes(x=Day,y=Sub_metering_1))+
  geom_hex()
plot2 <- ggplot(data = newDf,
                aes(x=Day,y=Sub_metering_2))+
  geom_hex()
plot3 <- ggplot(data = newDf,
                aes(x=Day,y=Sub_metering_3))+
  geom_hex()

ggarrange(plot1,plot2,plot3,
          ncol = 1, nrow = 3)
```

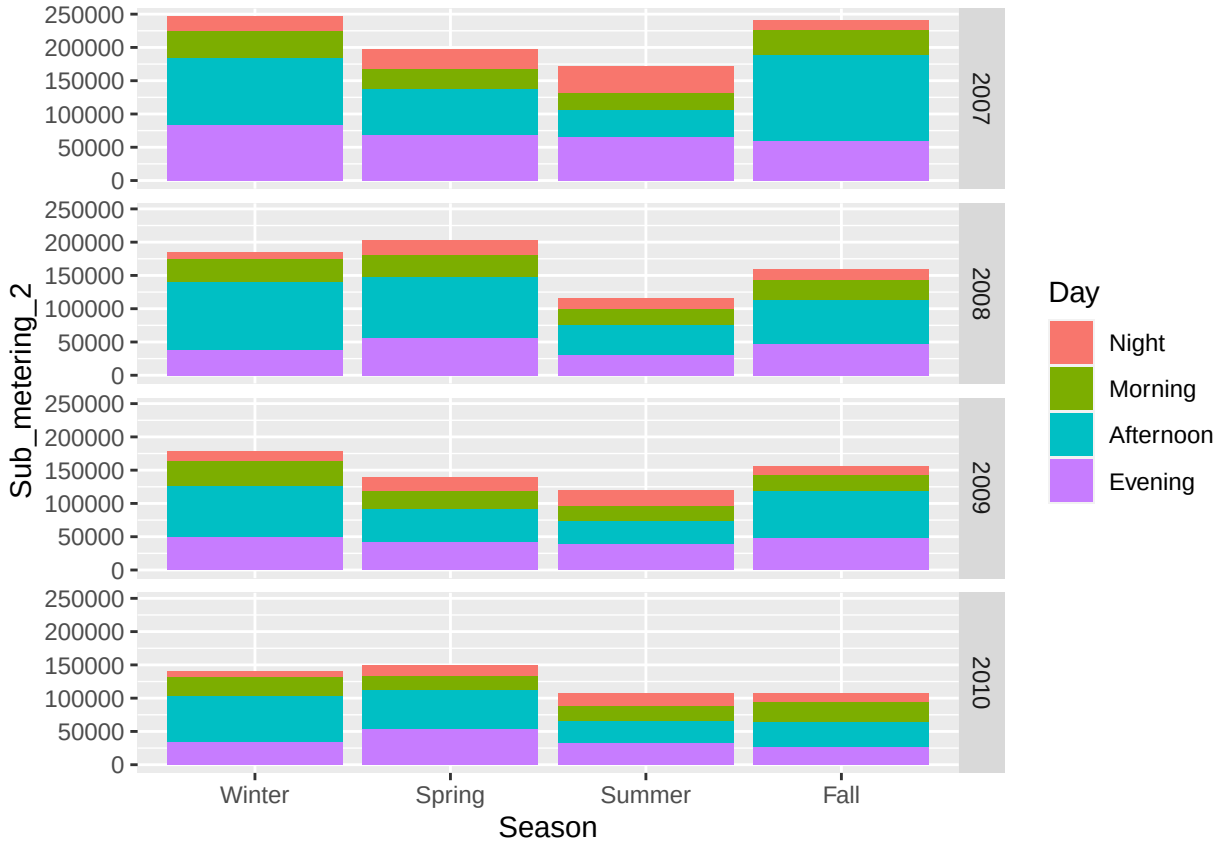


The graph below shows Sub-meter1,2 & 3 usage during each season from the year 2007 to 2010. The graph also tells us which time of the day consumes more comparatively.

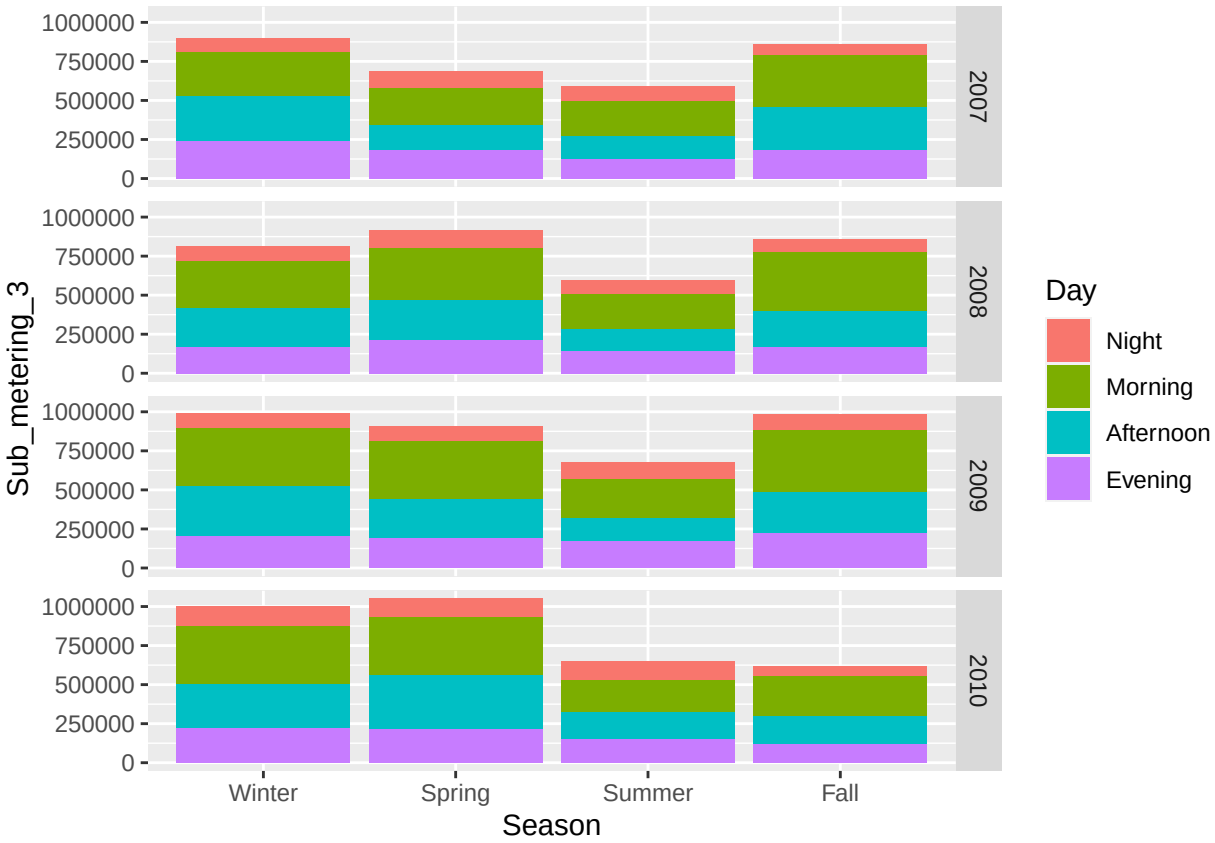
```
plot1 <- ggplot(data = newDf,
                aes(x=Season, y = Sub_metering_1,
                    fill= Day))+
  geom_bar(stat = "identity")+
  facet_grid(rows = vars(Year))
plot1
```




```
plot2 <- ggplot(data = newDf,
                aes(x=Season, y = Sub_metering_2,
                    fill= Day))+
  geom_bar(stat = "identity")+
  facet_grid(rows = vars(Year))
plot2
```

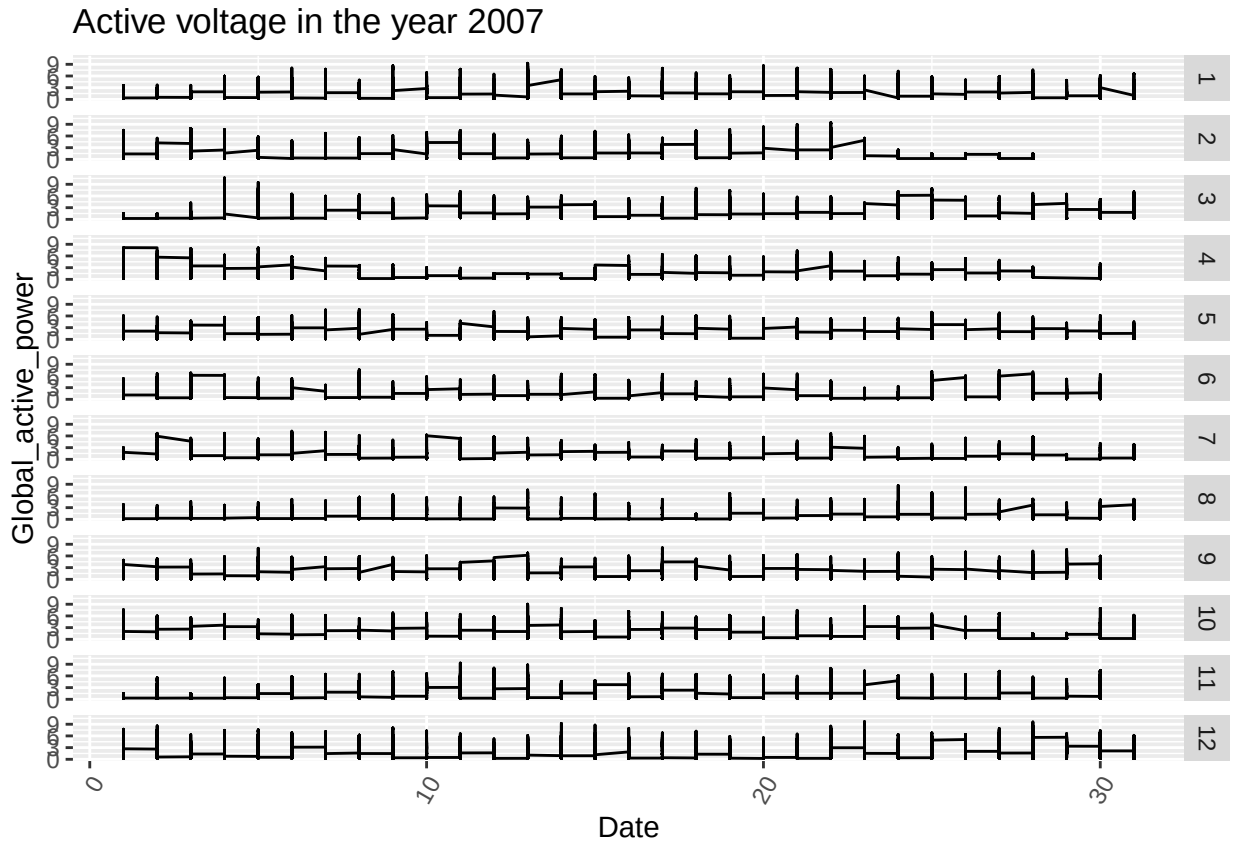



```
plot3 <- ggplot(data = newDf,
                aes(x=Season, y = Sub_metering_3,
                    fill= Day))+
  geom_bar(stat = "identity")+
  facet_grid(rows = vars(Year))
plot3
```

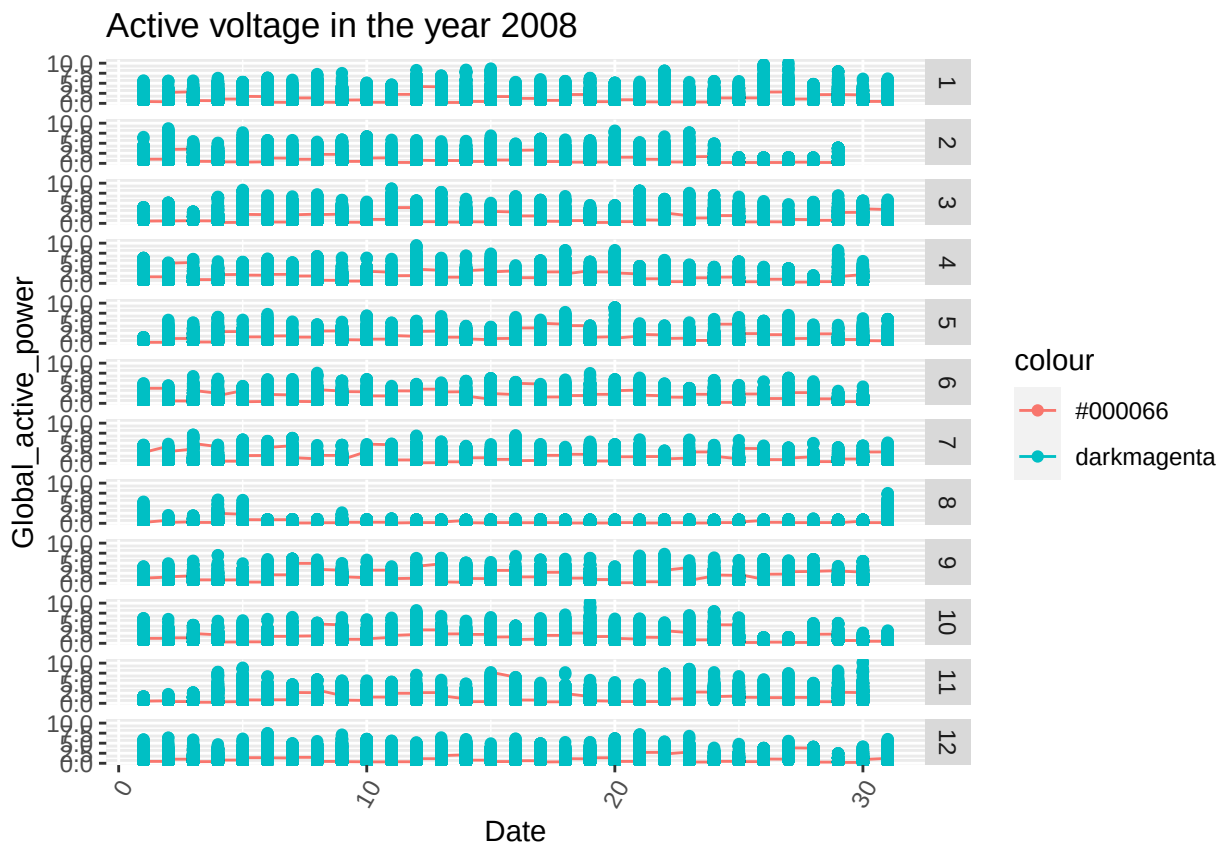


Let us take a look into the active voltage usage by using Geom_line. Geom_line gives a line graph which pictures the pattern of "Global_active_power" usage from the year 2007 to 2010

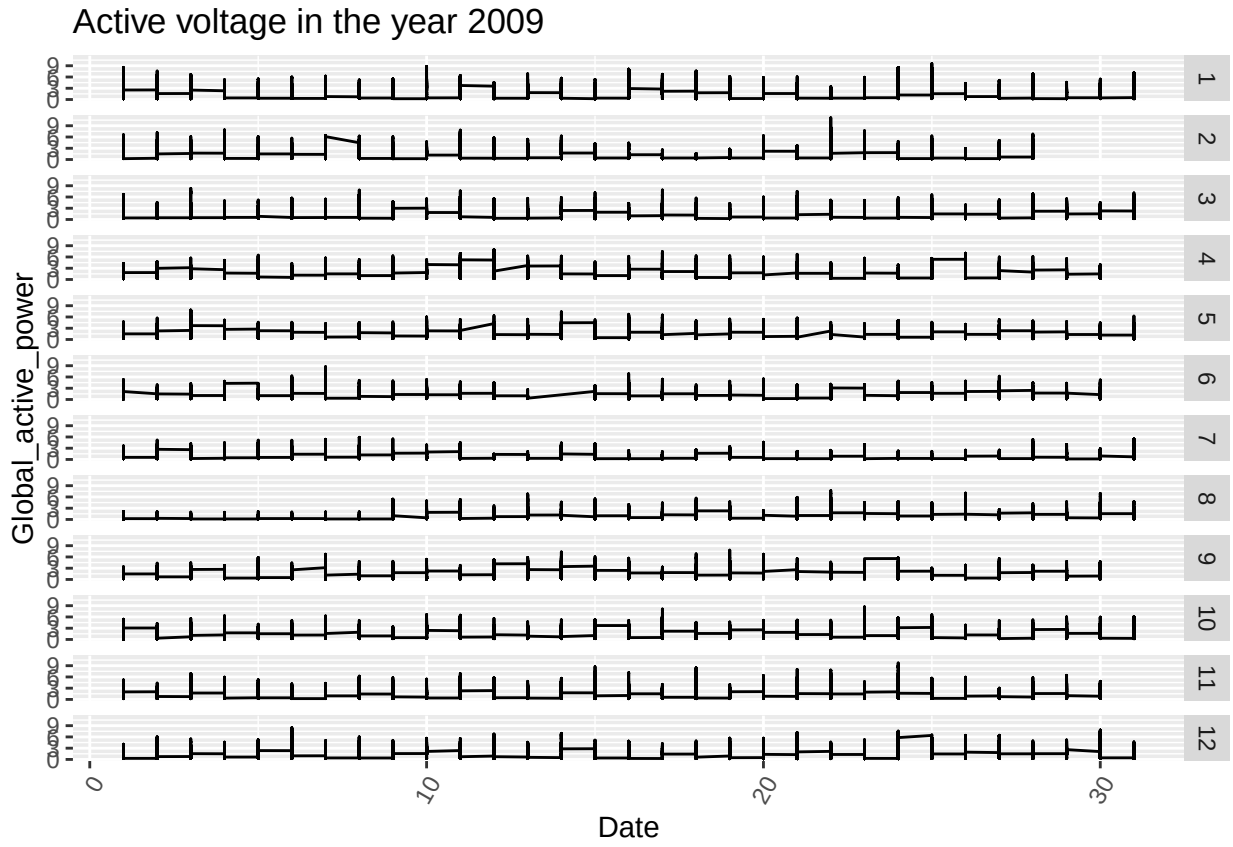
```
###### time series of active voltage


plot1 <- ggplot(data = filter(newDf, Year == 2007), aes(x=JustDate, y = Global_active_power))+
  geom_line()+
  xlab("Date")+
  theme(axis.text.x = element_text(angle = 60, hjust=1))+
  facet_grid(rows = vars(Month))+
  ggtitle("Active voltage in the year 2007")
plot1
```



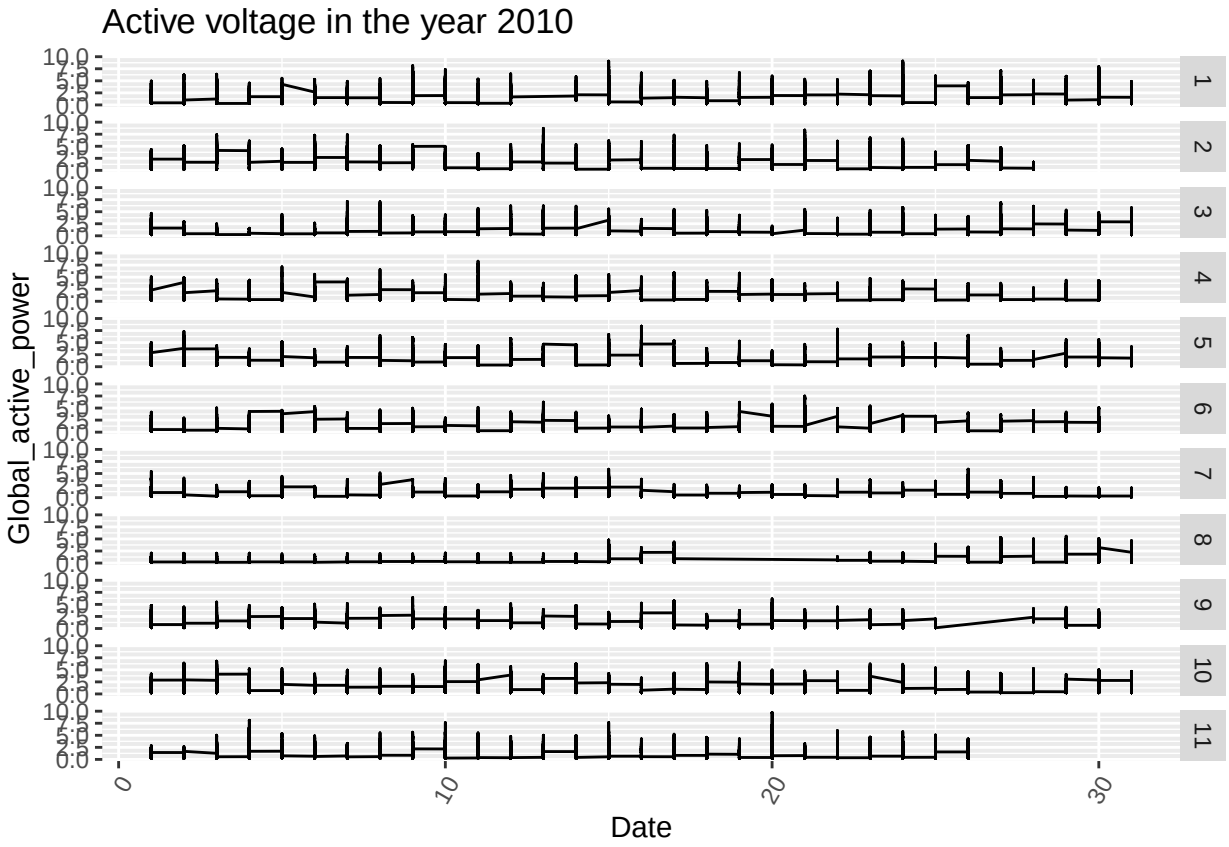

```
plot2 <- ggplot(data = filter(newDf, Year == 2008), aes(x=JustDate, y = Global_active_power))+
  geom_line(aes(color = "#000066"))+
  geom_point(aes(color = "darkmagenta"))+
  xlab("Date")+
  theme(axis.text.x = element_text(angle = 60, hjust=1))+
  facet_grid(rows = vars(Month))+
  ggtitle("Active voltage in the year 2008")
plot2
```



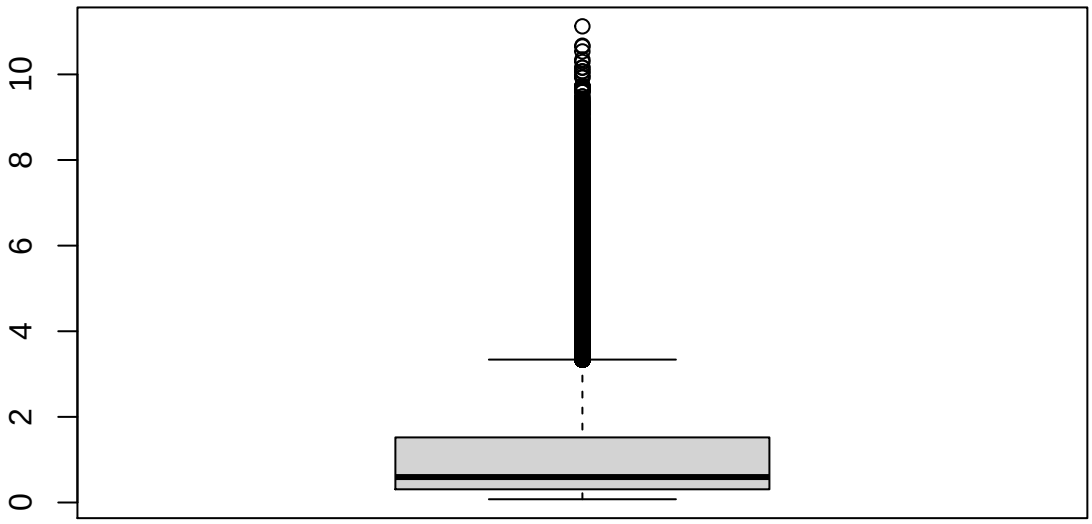

```
plot3 <- ggplot(data = filter(newDf, Year == 2009), aes(x=JustDate, y = Global_active_power))+
  geom_line()+
  xlab("Date")+
  theme(axis.text.x = element_text(angle = 60, hjust=1))+
  facet_grid(rows = vars(Month))+
  ggtitle("Active voltage in the year 2009")
plot3
```




```
plot4 <- ggplot(data = filter(newDf, Year == 2010), aes(x=JustDate, y = Global_active_power))+
  geom_line()+
  xlab("Date")+
  theme(axis.text.x = element_text(angle = 60, hjust=1))+
  facet_grid(rows = vars(Month))+
  ggtitle("Active voltage in the year 2010")
plot4
```




```
##BOXPLOT
boxplot(newDf$Global_active_power)
```


0
2
4
6
8
10