Energy Data Modeling

JaeSeok Jeong
Department of Statistics, SKKU

1. 수정한 것

- 1시간 단위 기상청 데이터 (기온, 습도, 바람) 추가
- setkey설정 하여 merge로 묶음
- Korean Holiday 변수 추가
- Training data rolling period 형식으로 변경
- usage & temperature 시각화

2. 1시간 단위 기상청 데이터 (기온, 습도, 바람) 추가 & setkey설정 하여 merge로 묶음

```
#weather variables(temperature, wind, humidity)
#colnames 지정 / 날짜형식 변환
temperature <- fread("weather_variable.csv", header = T)
colnames(temperature) <- c("date", "temp", "wind", "humidity")
temperature$date <- ymd_hm(temperature$date)
temperature$date_time <- temperature$date
temperature_1 <- temperature[, c(2:5)]

str(temperature)
```

```
## Classes 'data.table' and 'data.frame':   8784 obs. of  5 variables:
##  $ date     : POSIXct, format: "2016-01-01 00:00:00" "2016-01-01 01:00:00" ...
##  $ temp     : num  -1.9 -2.1 -2.2 -2.5 -2.9 -3.2 -3.1 -2.6 -2.4 -2 ...
##  $ wind     : num  0.1 1.6 0.4 1.8 1.9 2 1.3 1.6 2.4 2 ...
##  $ humidity : int  85 83 86 90 90 92 90 88 88 84 ...
##  $ date_time: POSIXct, format: "2016-01-01 00:00:00" "2016-01-01 01:00:00" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
# NA 없음
table(is.na(temperature))
```

```
##
## FALSE
## 43920
```

```
#1월 1일 제외
temperature_1 <- temperature_1[-c(1:24),]

# set the ON clause as keys of the tables:
setkey(energy_3,date_time)
setkey(temperature_1,date_time)
```

```
final_data <- merge(energy_3,temperature_1)
```

3. Korean Holiday 변수 추가

공휴일	날짜	요일
신정	1/1	Monday
설날	2/7 ~2/10	Sun - Wed
삼일절	3/1	Tue
국회의원 선거일	4/14	Wed
근로자의날	5/1	Sun
어린이날	5/5	Thr
석가탄신일	5/5	Sat
현충일	6/6	Mon
광복절	8/15	Mon
추석연휴	9/14 ~ 9/16	Wed -Fri
개천절	10/3	Mon
한글날	10/9	Sun
크리스마스	12/25	Sun

```
final_data$week <- ifelse(final_data$usage_week == 7 | final_data$usage_week == 1, "weekend", "weekday")
final_data$week <- as.factor(final_data$week)
final_data$usage_month <- as.factor(final_data$usage_month)
holidays <- c("2016-02-07","2016-02-08","2016-02-09","2016-02-10",
              "2016-03-01", "2016-04-13", "2016-05-01", "2016-05-05","2016-05-14","2016-06-06", "2016-08
holidays <- ymd(holidays)
final_data$holiday <- ifelse(final_data$date %in% holidays, "Y", "N")
table(final_data$holiday)
```

```
##
##    N    Y
## 8352  408
```

```
final_data$holiday <- as.factor(final_data$holiday)
```

4. final_data 요약

```
head(final_data, 1)
```

```
##     date_time       date usage_year usage_month usage_day usage_week
## 1: 2016-01-02 2016-01-02       2016           1         2           7
##    usage_hour hour_ave      V24      V23      V22      V21      V20
## 1:          0 93650.99 104540.5 109838.4 113069.3 111388.4 112008.3
##         V19      V18      V17     V16      V15      V14      V13      V12
## 1: 114519.3 104201.6 93796.64 92625.6 95064.66 96010.29 97557.34 101733.5
##         V11      V10       V9      V8       V7       V6       V5       V4
## 1: 98983.01 96524.49 88299.99 78225.6 70764.84 68364.21 67804.51 72473.51
##          V3       V2       V1 temp wind humidity    week holiday
```

```
## 1: 78284.92 86295.65 104899.1  3.6  0.6       69 weekend       N
```

```
str(final_data)
```

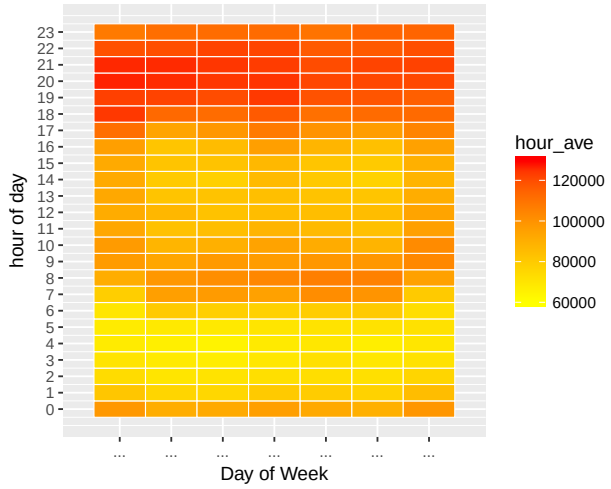
```
## Classes 'data.table' and 'data.frame':   8760 obs. of  37 variables:
##  $ date_time  : POSIXct, format: "2016-01-02 00:00:00" "2016-01-02 01:00:00" ...
##  $ date       : Date, format: "2016-01-02" "2016-01-02" ...
##  $ usage_year : int  2016 2016 2016 2016 2016 2016 2016 2016 2016 2016 ...
##  $ usage_month: Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ usage_day  : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ usage_week : num  7 7 7 7 7 7 7 7 7 7 ...
##  $ usage_hour : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ hour_ave   : num  93651 81992 70153 66545 63920 ...
##  $ V24        : num  104541 93651 81992 70153 66545 ...
##  $ V23        : num  109838 104541 93651 81992 70153 ...
##  $ V22        : num  113069 109838 104541 93651 81992 ...
##  $ V21        : num  111388 113069 109838 104541 93651 ...
##  $ V20        : num  112008 111388 113069 109838 104541 ...
##  $ V19        : num  114519 112008 111388 113069 109838 ...
##  $ V18        : num  104202 114519 112008 111388 113069 ...
##  $ V17        : num  93797 104202 114519 112008 111388 ...
##  $ V16        : num  92626 93797 104202 114519 112008 ...
##  $ V15        : num  95065 92626 93797 104202 114519 ...
##  $ V14        : num  96010 95065 92626 93797 104202 ...
##  $ V13        : num  97557 96010 95065 92626 93797 ...
##  $ V12        : num  101734 97557 96010 95065 92626 ...
##  $ V11        : num  98983 101734 97557 96010 95065 ...
##  $ V10        : num  96524 98983 101734 97557 96010 ...
##  $ V9         : num  88300 96524 98983 101734 97557 ...
##  $ V8         : num  78226 88300 96524 98983 101734 ...
##  $ V7         : num  70765 78226 88300 96524 98983 ...
##  $ V6         : num  68364 70765 78226 88300 96524 ...
##  $ V5         : num  67805 68364 70765 78226 88300 ...
##  $ V4         : num  72474 67805 68364 70765 78226 ...
##  $ V3         : num  78285 72474 67805 68364 70765 ...
##  $ V2         : num  86296 78285 72474 67805 68364 ...
##  $ V1         : num  104899 86296 78285 72474 67805 ...
##  $ temp       : num  3.6 3.4 3.4 3 2.6 2.2 2 1.5 1 1.9 ...
##  $ wind       : num  0.6 0.7 0.7 1.7 1.2 1.9 1.6 1.6 1.9 2 ...
##  $ humidity   : int  69 72 70 74 76 78 77 82 83 78 ...
##  $ week       : Factor w/ 2 levels "weekday","weekend": 2 2 2 2 2 2 2 2 2 2 ...
##  $ holiday    : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "sorted")= chr "date_time"
##  - attr(*, ".internal.selfref")=<externalptr>
```

5. 시각화

usage 시간당 사용량 heatmap

```
energy_map <- final_data %>%
  dplyr::select(date_time, date, hour_ave, usage_week)
```

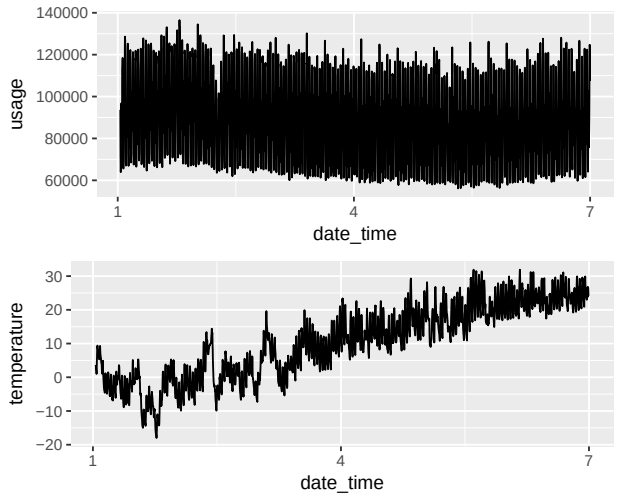
```r
energy_map$Hour <- hour(energy_map$date_time)

energy_map$Week <- week(energy_map$date_time)

var <- c("월","화","수","목","금","토","일")

energy_map_week1 <- energy_map %>%
  dplyr::filter(Week == 2)

gg1 <-ggplot(energy_map_week1,aes(x=usage_week,y=Hour,
                                  fill=hour_ave)) +
  scale_x_discrete(limits =var)


#min(energy_map_week1$hour_ave)
#max(energy_map_week1$hour_ave)


gg1 <- gg1 + geom_tile(color="white", size=0.3)+scale_fill_gradient(low="yellow", high="red",limit=c(60

#unique(energy_map$usage_week)

gg1 <- gg1 +scale_y_continuous(breaks=seq(0,23,1))
gg1 <- gg1 + labs(x="Day of Week", y="hour of day")
gg1
```



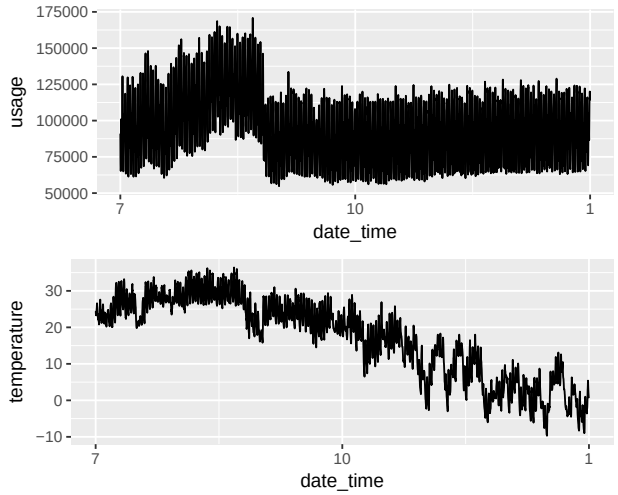
상반기 /하반기 사용량 및 온도 그래프

```r
#1~6월 그래프
unique_month <- c(1:6)
first_half <- final_data %>%
  dplyr::filter(usage_month %in% unique_month)

first_half_table <- data.table(usage = first_half$hour_ave,
                               date_time = first_half$date_time,
```

```
                                temperature = first_half$temp)

#사용량 그래프
first_half_usage <- ggplot(first_half_table, aes(date_time, usage)) +
  geom_line()

#기온 추이 그래프
first_half_temperature <- ggplot(first_half_table, aes(date_time, temperature)) +
  geom_line()


grid.arrange(first_half_usage, first_half_temperature, nrow=2)
```



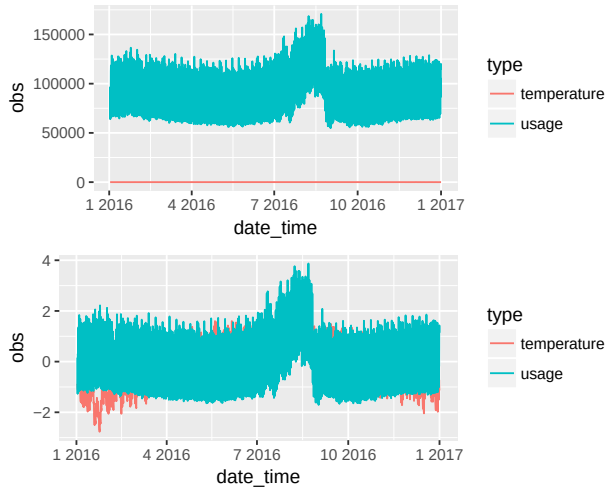

```
# 7~ 12월 그래프

unique_month <- c(7:12)
second_half <- final_data %>%
  dplyr::filter(usage_month %in% unique_month)

second_half_table <-data.table(usage = second_half$hour_ave,
                  date_time = second_half$date_time,
                  temperature = second_half$temp)



#사용량 그래프
second_half_usage <- ggplot(second_half_table, aes(date_time, usage)) +
  geom_line()

#기온 추이 그래프
second_half_temperature <- ggplot(second_half_table, aes(date_time, temperature)) +
  geom_line()


grid.arrange(second_half_usage, second_half_temperature, nrow=2)
```



usage와 temperature를 한 그래프에 같이 나타낸 것

축설정 어려움..단위 다를 때.

```
  usage_temp1 <-data.table(obs = c(final_data$hour_ave, final_data$temp),
                date_time = rep(final_data$date_time, 2),
                type = rep(c("usage", "temperature"), each = 8760))


  usage_temp2 <-data.table(obs = c(scale(final_data$hour_ave),scale(final_data$temp)),
                date_time = rep(final_data$date_time, 2),
                type = rep(c("usage", "temperature"), each = 8760))

usage_temp_plot <- ggplot(usage_temp1, aes(date_time, obs,color=type, group=type)) +
  geom_line()

usage_temp_plot2 <- ggplot(usage_temp2, aes(date_time, obs,color=type, group=type)) +
  geom_line()

grid.arrange(usage_temp_plot, usage_temp_plot2, nrow=2)
```



7.Training & Test set 구분

n_date를 조정하여 training 수 조절

```
#rolling period
n_date <- unique(final_data[, date]) # 1/2 ~ 12/31


training_data_temp<- final_data[(date %in% n_date[1:2]),-c(1:7)]
test_data_temp<- final_data[(date %in% n_date[3]),-c(1:7)]
test_value<-final_data[(date %in% n_date[3]),c(8)]
```

8. 시간별/요일별 median 사용량으로 id 구분

k-means

```
energy_2 <- energy_2 %>%
  dplyr::mutate(date_time = make_datetime(usage_year,usage_month, usage_day, usage_hour),
                usage_wday = wday(date_time))

id_hour <- energy_2 %>%
  dplyr::group_by(id, usage_hour) %>%
  dplyr::summarise(usage = median(hour_ave, na.rm = T))

id_hour_cast<-cast(id_hour, id~usage_hour, values="usage")
```

```
## Using usage as value column.  Use the value argument to cast to override this choice
```

```
id_wday <- energy_2 %>%
  dplyr::group_by(id, usage_wday) %>%
  dplyr::summarise(usage = median(hour_ave, na.rm = T))
```

```
id_wday_cast<-cast(id_wday, id~usage_wday, values="usage")
```

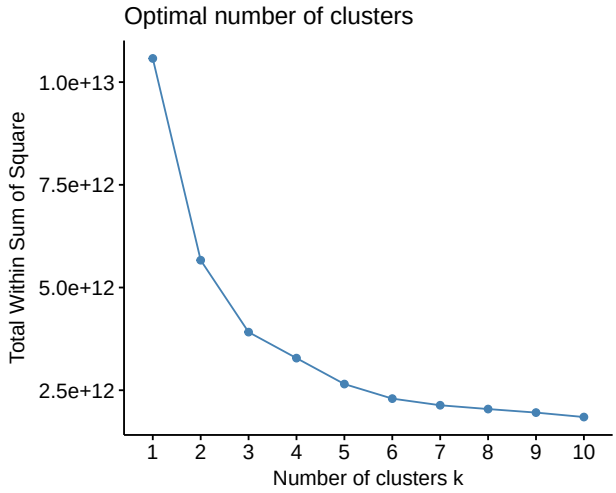
```
## Using usage as value column.  Use the value argument to cast to override this choice
```

```
colnames(id_wday_cast) <- c("id","Sun", "Mon", "Tue", "Wed", "Thr", "Fri", "Sat")
```

```
id_wday_cast <- as.data.table(id_wday_cast)
id_hour_cast <- as.data.table(id_hour_cast)

setkey(id_wday_cast,id)
setkey(id_hour_cast,id)

id_final <- merge(id_wday_cast, id_hour_cast)
id_final_1 <- id_final[, -1]

head(id_final, 3)
```

```
##            id      Sun      Mon      Tue      Wed      Thr      Fri
## 1: 012746aa5a 79142.62 69171.75 79364.25 71165.25 71793.00 71058.00
## 2: 01850eab5b 75766.25 77779.75 82461.50 80937.62 78374.75 79196.88
## 3: 0251cabf2d 43323.38 35839.00 35375.38 35703.00 35899.12 36074.25
##         Sat         0        1        2        3        4        5
## 1: 74681.76  85195.25 63789.50 59243.25 59330.00 63611.88 60562.12
## 2: 91815.00 111364.75 86369.50 73833.62 70452.50 63166.50 89798.25
## 3: 40227.75  39193.00 35031.75 34232.00 33106.75 32574.12 33846.75
##           6        7        8        9        10        11       12
## 1: 62282.25 73961.88 88485.25 88955.62 105087.50 110504.88 102934.4
## 2: 72247.38 89580.75 80859.75 62855.12  60015.25  59853.25  58051.0
## 3: 33558.25 42272.88 40598.50 35677.00  34479.25  33582.50  33476.0
##          13       14      15       16       17       18        19
## 1: 91486.88 62915.38 61235.5 62343.75 59757.50  66589.5 113028.25
## 2: 56955.12 58007.25 61721.0 63438.50 71553.25 104293.8 144572.50
## 3: 32104.96 33102.62 32959.0 33029.00 33615.25  37525.0  48875.38
##           20       21       22       23
## 1: 143501.25 144867.4 133064.5 115591.5
## 2: 146422.50 155741.0 151397.8 137359.0
## 3:  51687.25  52031.0  52412.0  49094.5
```

```
fviz_nbclust(id_final_1, kmeans, method = "wss")
```




```
# compute kmeans
set.seed(123)

km <- kmeans(id_final_1, 3, nstart=10)

id_final$cluster <- km$cluster

#visualization
plotcluster(id_final_1, km$cluster)
```