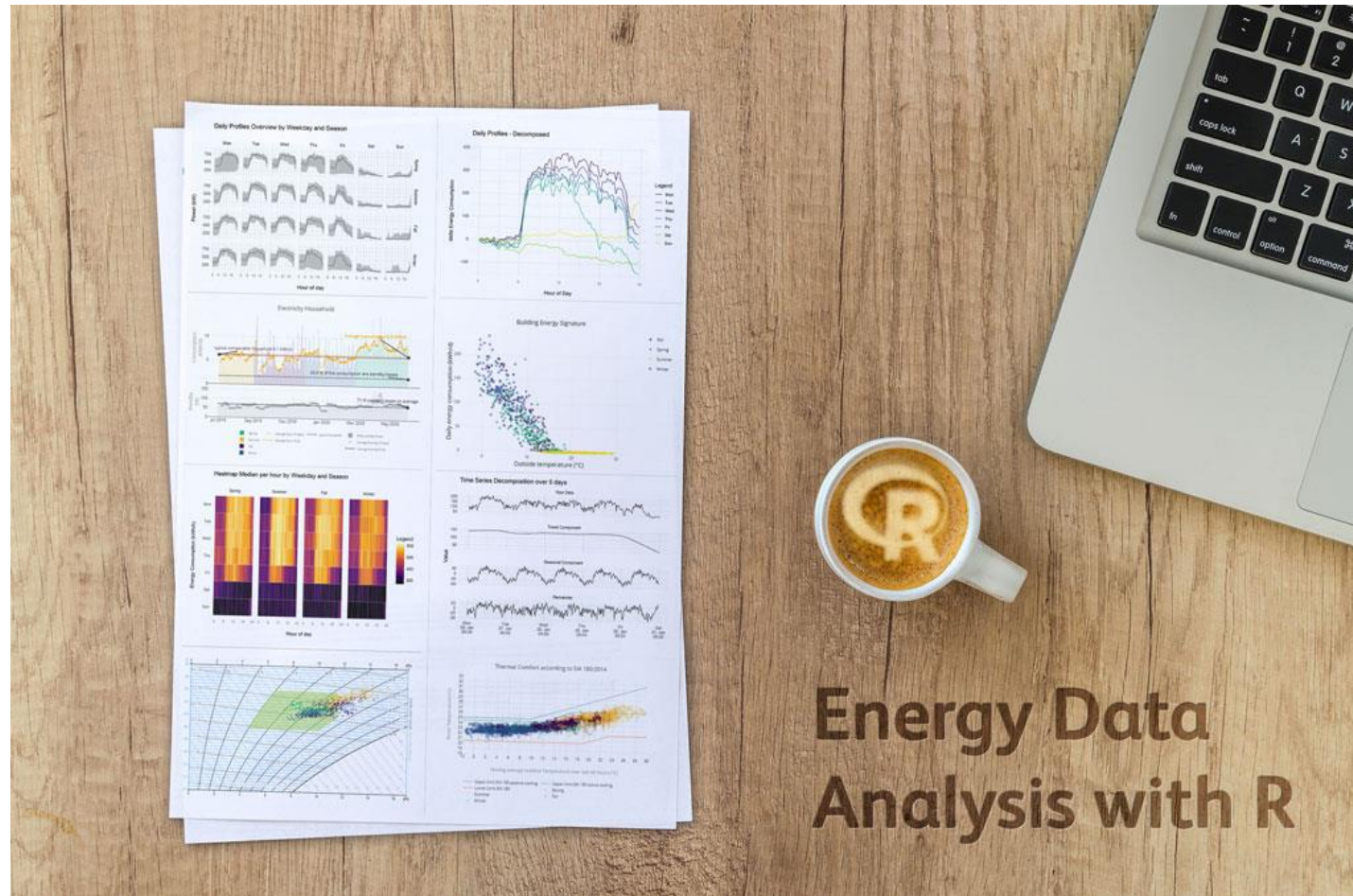


# 4. R로 하는 에너지 데이터 분석



# R, R Studio 설치

통계학과 데이터 마이닝을 위한 오픈소스 프로그램



<https://cran.r-project.org/>

- 설치파일 다운로드 :

[Download R for Linux](#) ([Debian](#), [Fedora/Redhat](#), [Ubuntu](#))

[Download R for macOS](#)

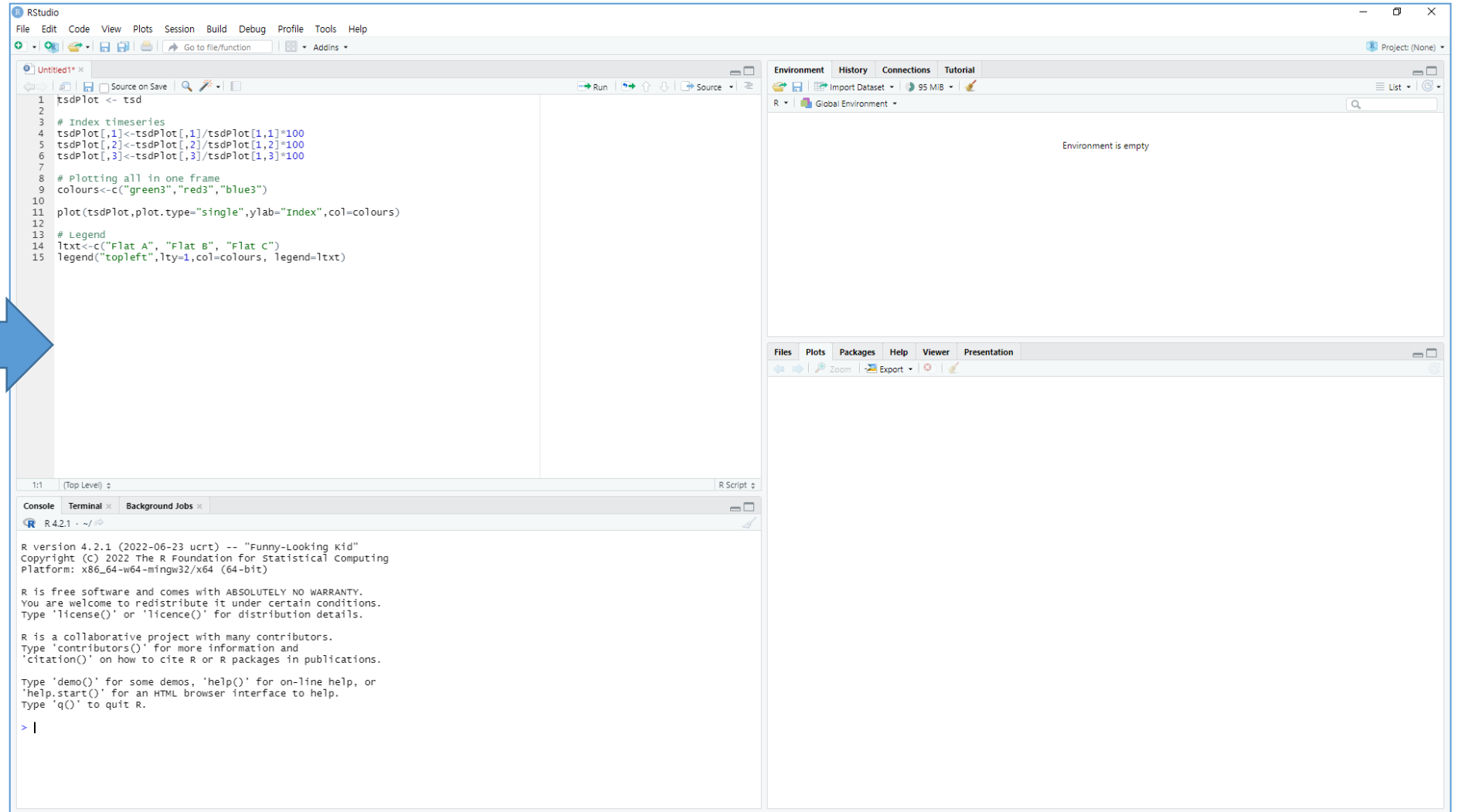
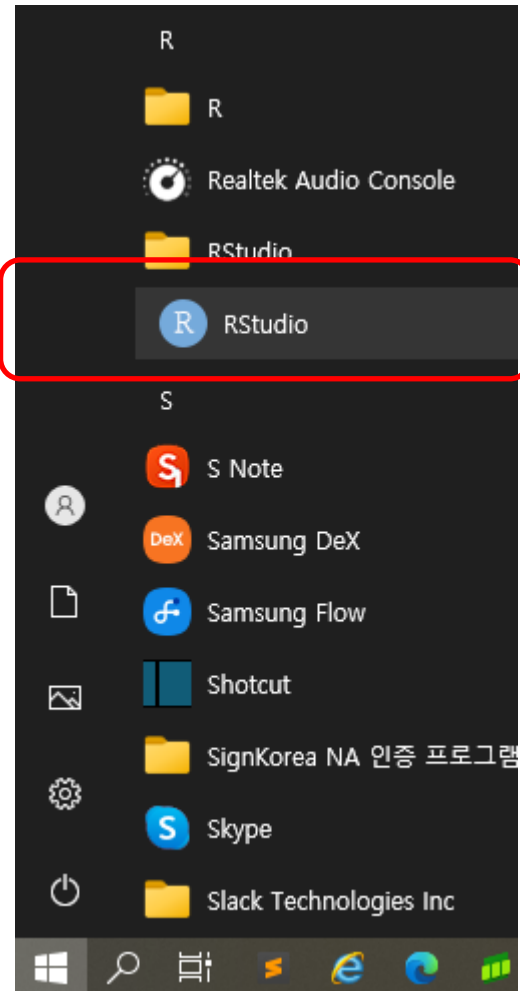
[Download R for Windows](#)



<https://www.rstudio.com/>

- 설치파일 다운로드 : <https://www.rstudio.com/products/rstudio/download/>
- 참고 : <https://bit.ly/3LkO202>

# R Studio(IDE)



■ R 기본 : <https://gomguard.tistory.com/235>

# 패키지 설치

```
install.packages("ggplot2")
```

```
install.packages("devtools")
```

```
library(devtools)
```

```
install_github("hslu-ige-laes/redutils")
```

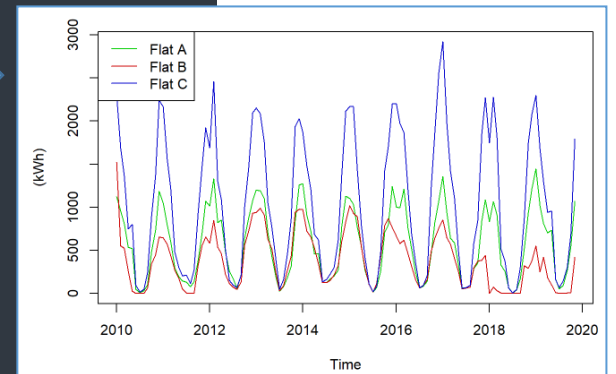
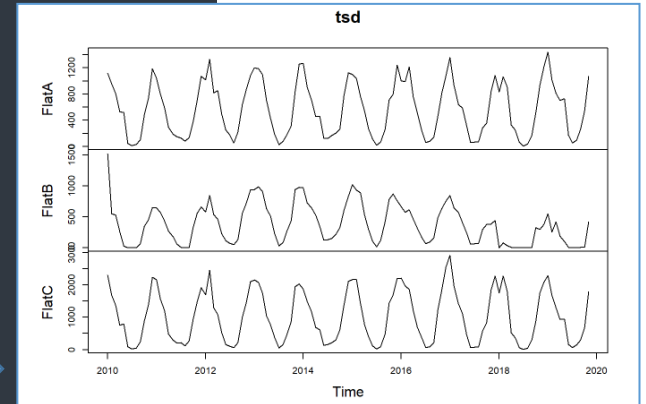
# 데이터 로드

```
1 # read data from current folder
2 df <- read.csv("datafile.csv")
3
4 # read data from a specific folder
5 df <- read.csv("C:/Desktop/datafile.csv")
6
7 # read data from a file in the internet
8 df <- read.csv2("https://github.com/hslu-ige-laes/edar/raw/master/sampleData/flatElectricity.csv")
9
10 # add arguments on how to parse the content
11 df <- read.csv("datafile.csv",
12               header=FALSE,
13               stringsAsFactors=FALSE,
14               sep = ",",
15               na.strings = c("", "NA"))
```

# EDA(Explorative Data Analysis)

```
4 df <- read.csv2("https://github.com/hslu-ige-laes/edar/raw/master/sampleData/flatHeatAndHotWater.csv",
5                 stringsAsFactors=FALSE)
6 # filter flat
7 df <- df %>% select(timestamp, Adr01_energyHeat, Adr02_energyHeat, Adr03_energyHeat)
8 #df <- df %>% filter(timestamp > "2014-12-01")
9
10 df <- df %>% dplyr::mutate(FlatA = lead(Adr01_energyHeat) - Adr01_energyHeat)
11 df <- df %>% dplyr::mutate(FlatB = lead(Adr02_energyHeat) - Adr02_energyHeat)
12 df <- df %>% dplyr::mutate(FlatC = lead(Adr03_energyHeat) - Adr03_energyHeat)
13
14 # remove counter value column
15 df <- df %>% select(-Adr01_energyHeat, -Adr02_energyHeat, -Adr03_energyHeat) %>% na.omit()
16 df$timestamp <- parse_date_time(df$timestamp,
17                                 orders = "YmdHMS",
18                                 tz = "Europe/Zurich")
19
20 tsd <- ts(df %>% select(-timestamp), frequency = 12, start = min(year(df$timestamp)))
21
22 # plot FlatA_Ele
23 plot(tsd)
24
25 # Plotting all in one frame
26 colours<-c("green3","red3","blue3")
27
28 plot(tsd,plot.type="single",ylab="(kWh)",col=colours)
29
30 # Legend
31 ltxt<-c("Flat A", "Flat B", "Flat C")
32 legend("topleft",lty=1,col=colours, legend=ltxt)
```

파이프 연산자(%>%): 데이터 전달하기  
<https://thebook.io/080217/ch07/01/02/>

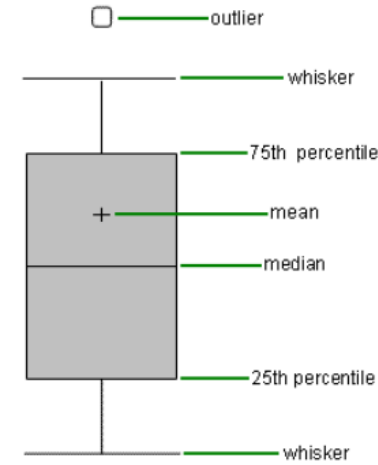


# Boxplot

```

8 df <- read.csv("https://github.com/hslu-ige-laes/edar/raw/master/sampleData/flatTempHum.csv",
9               stringsAsFactors=FALSE,
10               sep = ";")
11
12 df$time <- parse_date_time(df$time,
13                           order = "YmdHMS",
14                           tz = "Europe/Zurich")
15
16 df <- df %>%
17   select(time, FlatA_Temp, FlatB_Temp, FlatC_Temp, FlatD_Temp)
18
19 df <- as.data.frame(tidyr::pivot_longer(df,
20                                       cols = -time,
21                                       names_to = "room",
22                                       values_to = "Temp",
23                                       values_drop_na = TRUE))
24
25 minY <- round(min(df %>% select(Temp), na.rm = TRUE) - 1, digits = 0)
26 maxY <- round(max(df %>% select(Temp), na.rm = TRUE) + 1, digits = 0)
27
28
29 plot <- ggplot(df, aes(x = room, y = Temp)) +
30   geom_boxplot(outlier.alpha = 0.01, outlier.shape = 1, outlier.colour = "darkgrey") +
31   scale_y_continuous(limits=c(minY,maxY), breaks = seq(minY, maxY, by = 2)) +
32   ggtitle("Boxplot per Flat/Room") +
33   theme_minimal() +
34   theme(
35     legend.position="none",
36     plot.title = element_text(hjust = 0.5)
37   )
38
39 yaxis <- list(
40   title = "Temp<sub>Room</sub> in \u00B0C\n",
41   automargin = TRUE,
42   titlefont = list(size = 14, color = "darkgrey")
43 )
44
45 # create interactive plot
46 ggplotly(plot + ylab(" ") + xlab(" ")) %>%
47   plotly::config(modeBarButtons = list(list("toImage")),
48                 displaylogo = FALSE,
49                 toImageButtonOptions = list(
50                   format = "svg"
51                 )
52   ) %>%
53   layout(yaxis = yaxis)

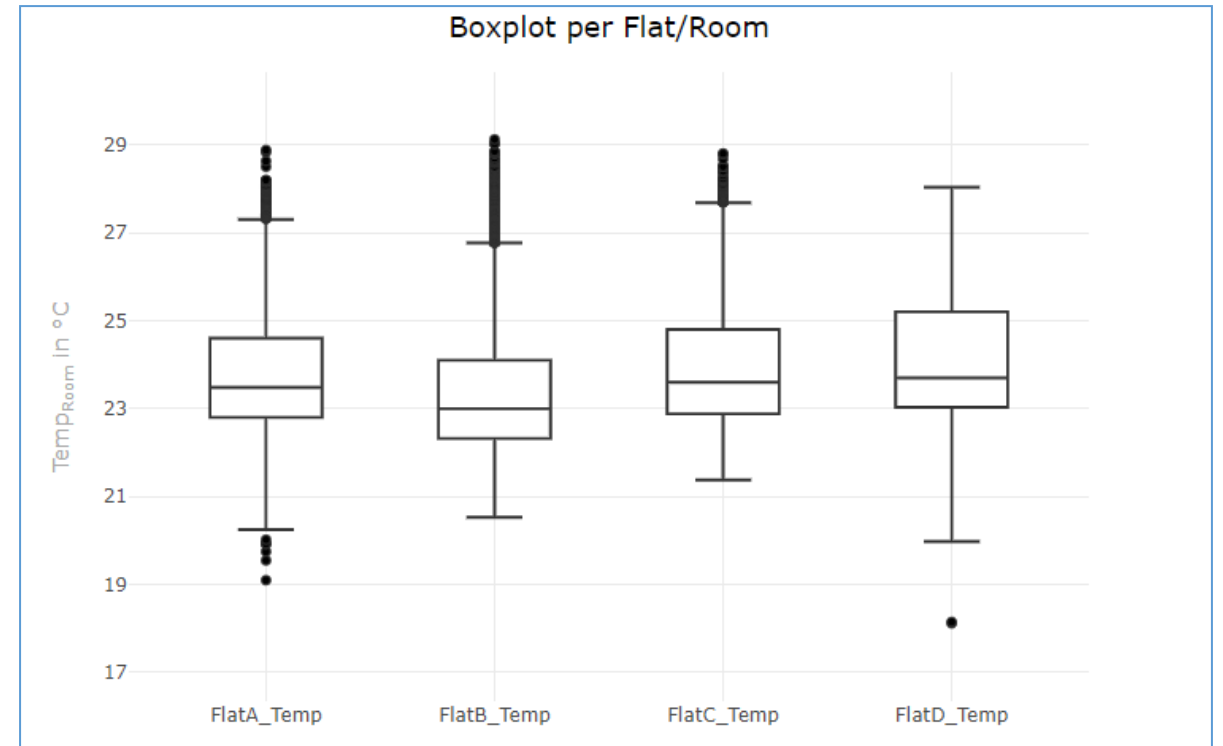
```



median - 중앙값

IQR - 25% ~ 75%에 해당하는 부분  
 75th Percentile - 제 3사분위수  
 25th Percentile - 제 1사분위수  
 Whisker - 최댓값, 최솟값

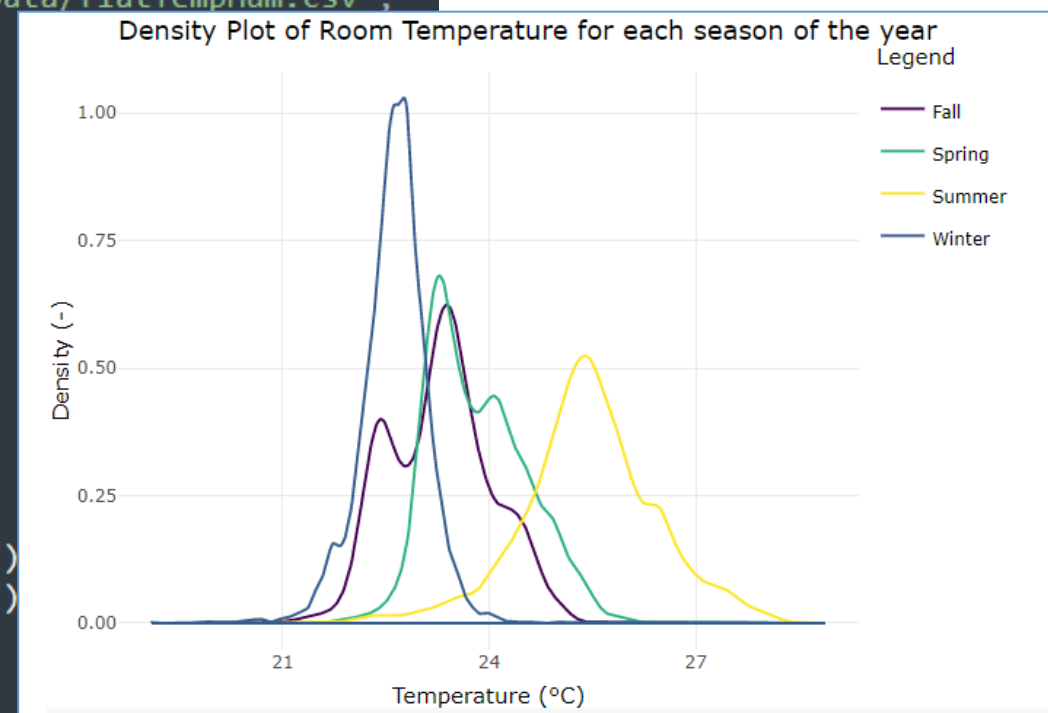
Outlier :  $Q1 - 1.5(Q3 - Q1) < data < Q3 + 1.5(Q3 - Q1)$  를 벗어나는 데이터





# 밀도분포(Density plot)

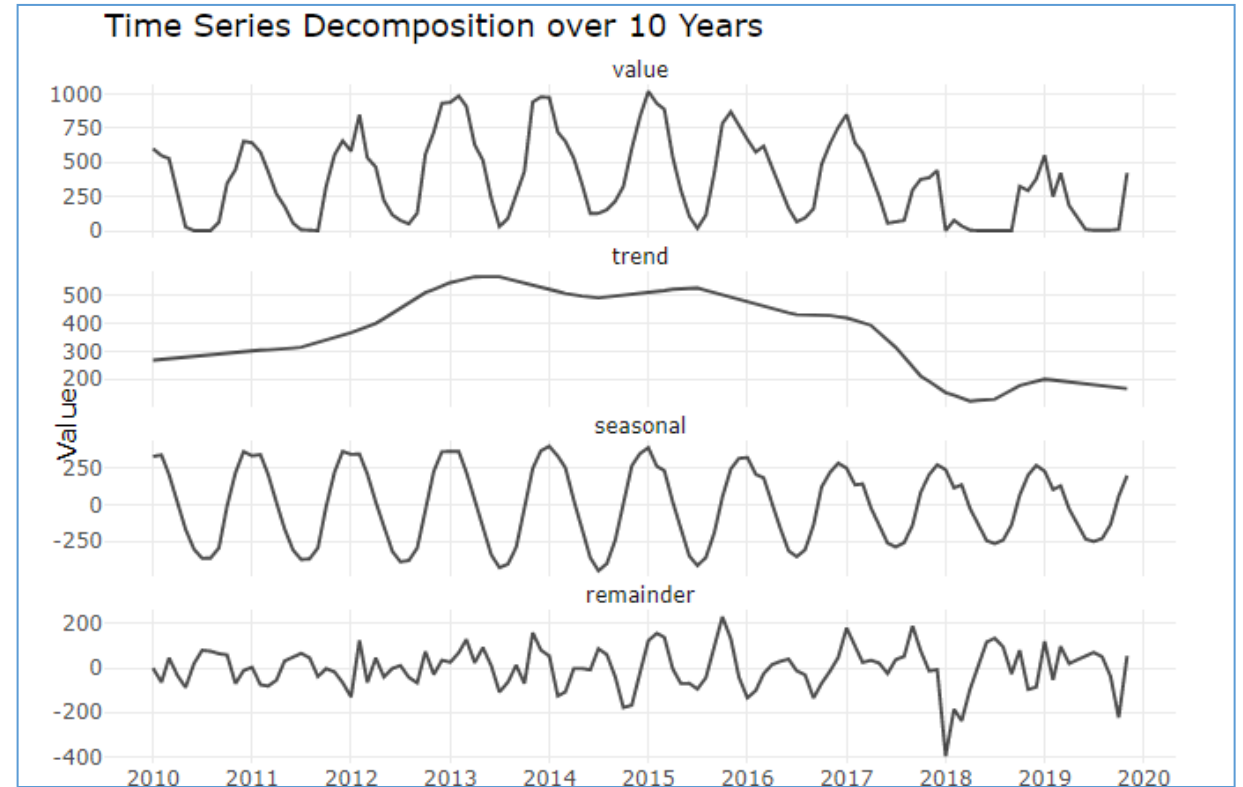
```
5 # load time series data and aggregate daily mean values
6 library(dplyr)
7 library(lubridate)
8 library(reduutils)
9
10 # read and print data
11 df <- read.csv("https://github.com/hslu-ige-laes/edar/raw/master/sampleData/flatTempHum.csv",
12               stringsAsFactors=FALSE,
13               sep = ";")
14
15 # select temperature and remove empty cells
16 df <- df %>% select(time, FlatA_Temp) %>% na.omit()
17
18 colnames(df) <- c("time", "value")
19
20 df$season = getSeason(df$time)
21
22 # static chart with ggplot
23 plot <- ggplot(df) +
24   geom_density(aes(x = value, colour = season)) +
25   scale_color_manual(values=c("#440154", "#2db27d", "#fde725", "#365c8d"))
26   ggtitle("Density Plot of Room Temperature for each season of the year")
27   labs(x = "Temperature (\u00B0C)",
28        y = "Density (-)",
29        colour = "Legend") +
30   theme_minimal()
31
32 # interactive chart
33 plotly::ggplotly(plot)
```



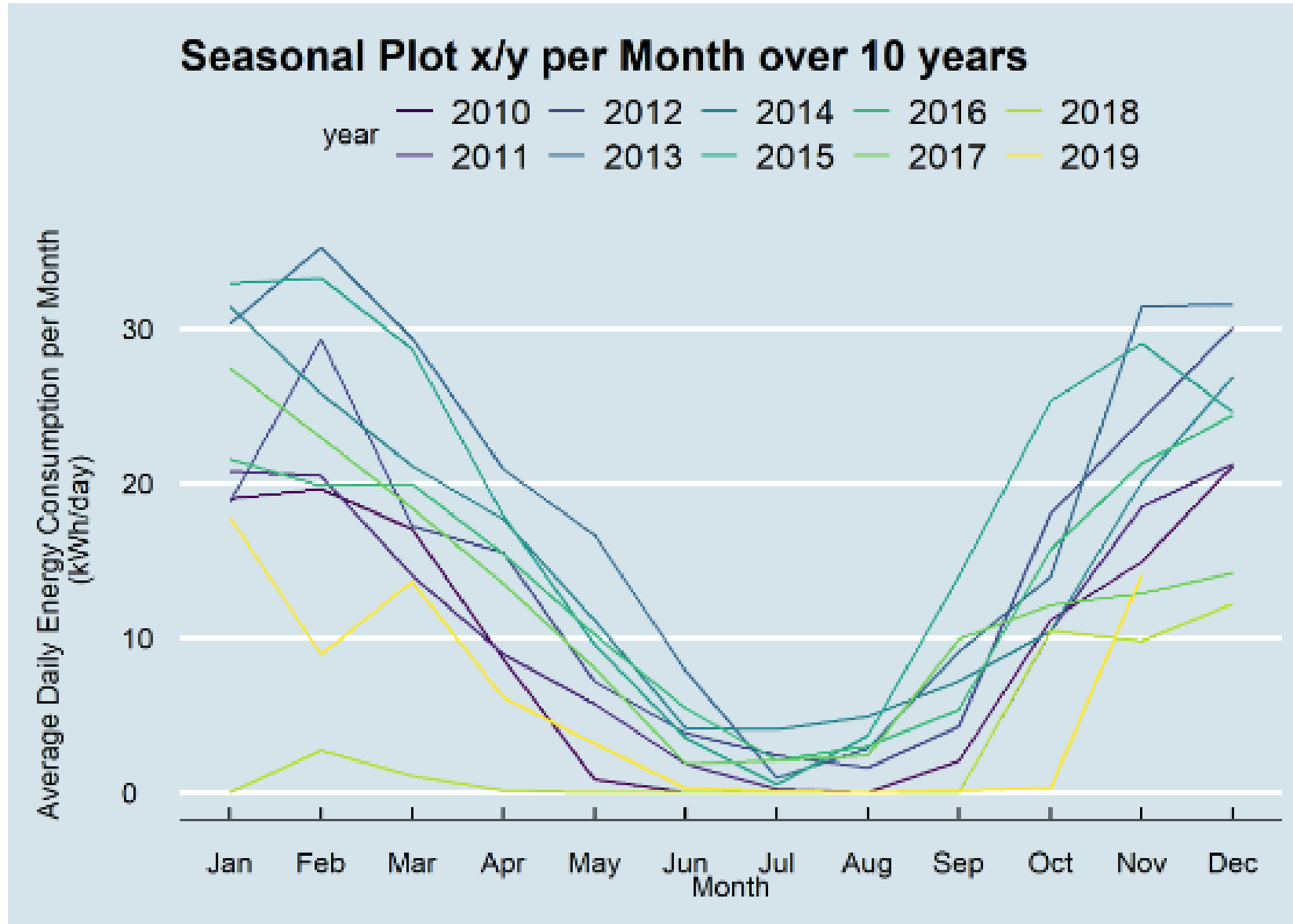


# 시계열 분해(Time Series Decomposition)

```
8 df <- read.csv2("https://github.com/hslu-ige-laes/edar/raw/master/sampleData/flatHeatAndHotWater.csv",
9               stringsAsFactors=FALSE)
10
11 # filter flat
12 df <- df %>% select(timestamp, ADR02_energyHeat)
13 colnames(df) <- c("Time", "meterValue")
14 df$Time <- parse_date_time(df$Time, orders = "YmdHMS", tz = "Europe/Zurich")
15
16 # calculate consumption value per month
17 df <- df %>% dplyr::mutate(value = lead(meterValue) - meterValue)
18
19 # remove counter value column
20 df <- df %>% select(-meterValue) %>% na.omit()
21 df[1,2] <- 600
22 df.ts <- ts(df %>% select(value) %>% na.omit(), frequency = 12, start = min(year(df$Time)))
23 df.decompose <- df.ts[,1] %>%
24   stl(s.window = 7)
25
26 df.decompose <- df.decompose$time.series
27 df.decompose <- as.data.frame(df.decompose)
28 df.decompose <- cbind(df, df.decompose)
29 data <- as.data.frame(tidy::pivot_longer(df.decompose,
30                                       cols = -Time,
31                                       names_to = "Component",
32                                       values_to = "Value",
33                                       values_drop_na = TRUE)
34 )
35 data$component <- as.factor(data$Component)
36 data$component <- factor(data$Component, c("value",
37                                           "trend",
38                                           "seasonal",
39                                           "remainder"))
40 data$Value <- round(data$Value, digits = 1)
41
42 plot <- ggplot(data) +
43   geom_path(aes(x = Time,
44                y = Value
45                ),
46            color = "black",
47            alpha = 0.7) +
48   facet_wrap(~component, ncol = 1, scales = "free_y") +
49   scale_x_datetime(date_breaks = "years", date_labels = "%Y") +
50   theme_minimal() +
51   theme(panel.spacing = unit(1, "lines"),
52         legend.position = "none") +
53   labs(x = "") +
54   ggtitle("Time Series Decomposition over 10 Years")
55 ggplotly(plot)
```

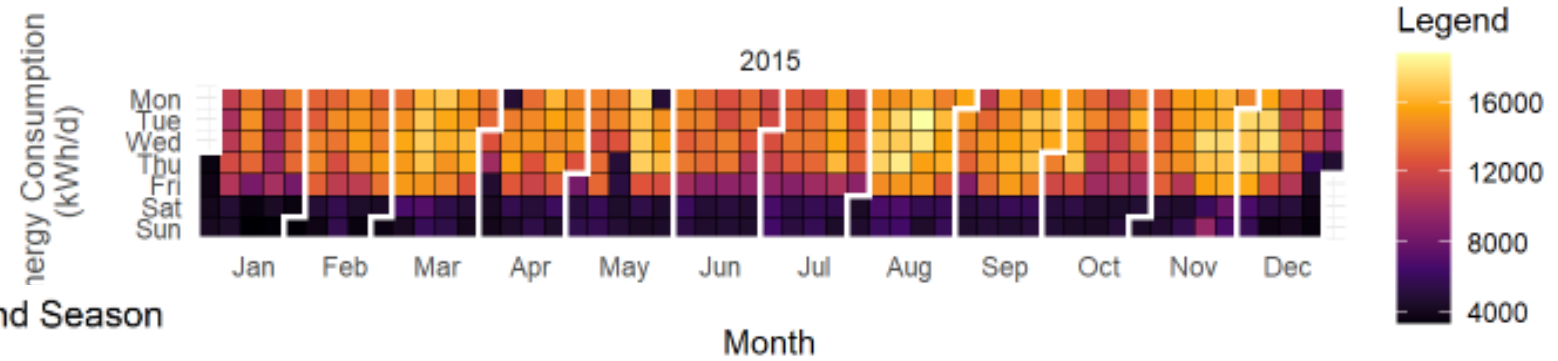


# 계절성 그래프(Seasonal Plots)

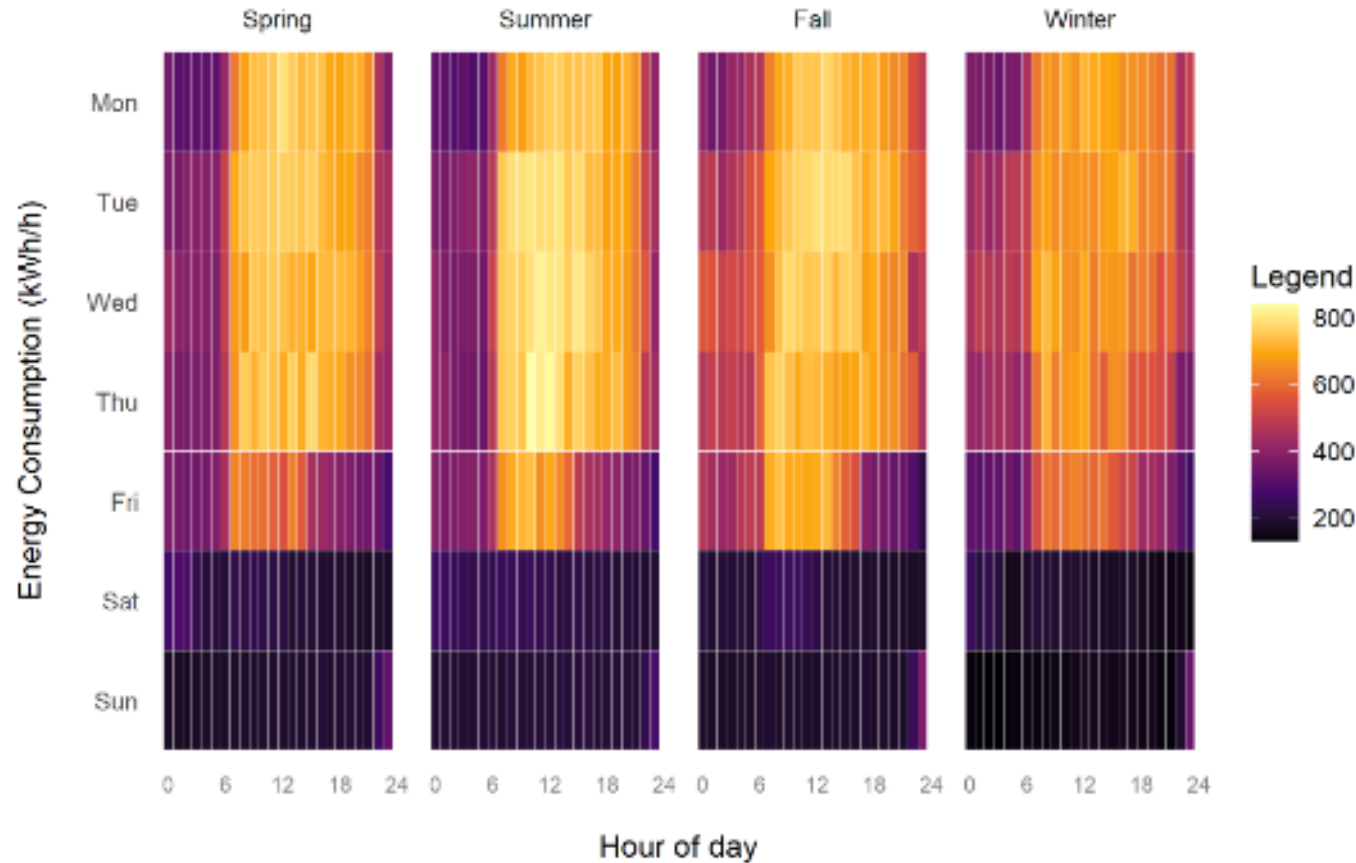


# Heat Map

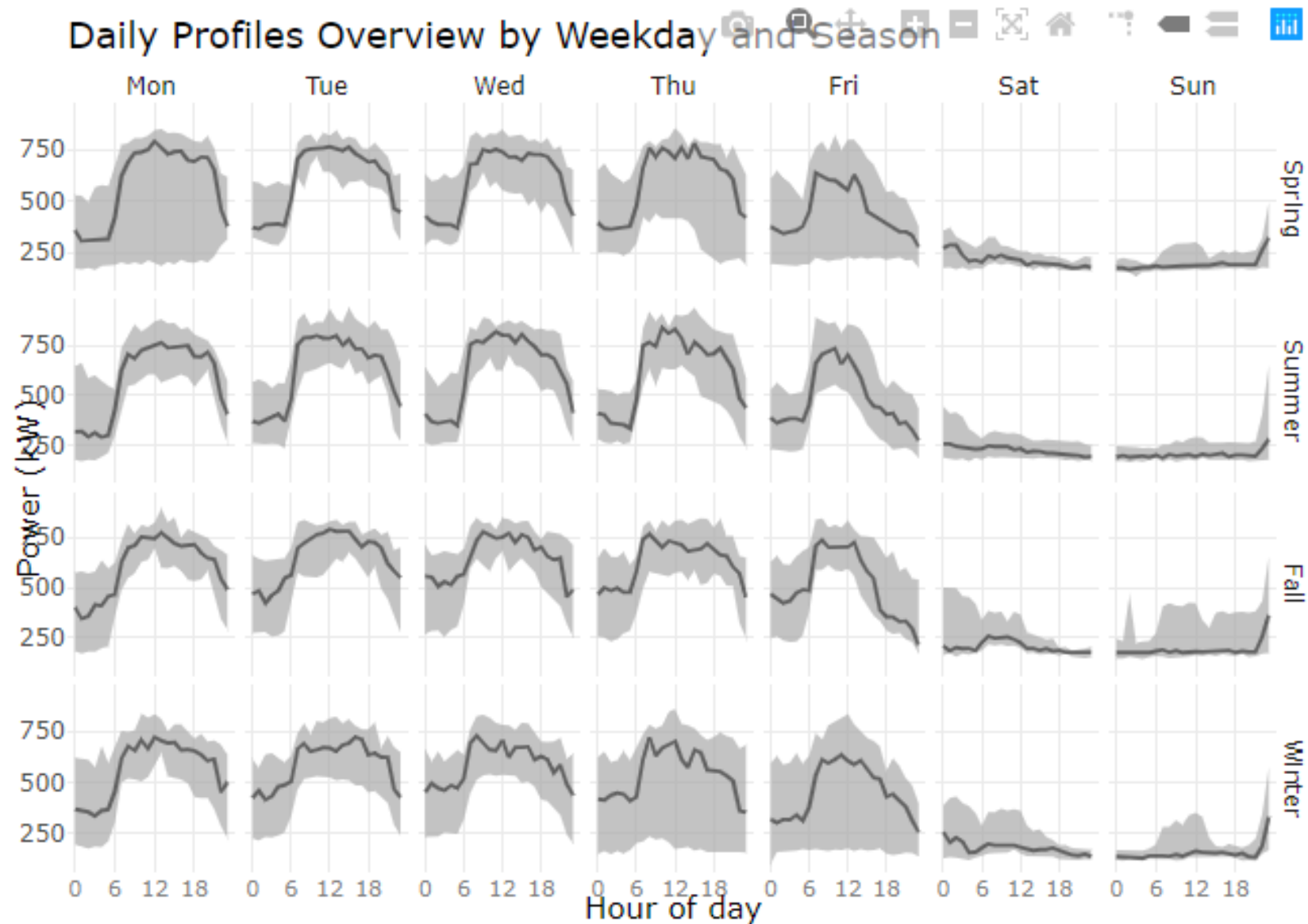
Calendar Plot Energy Consumption



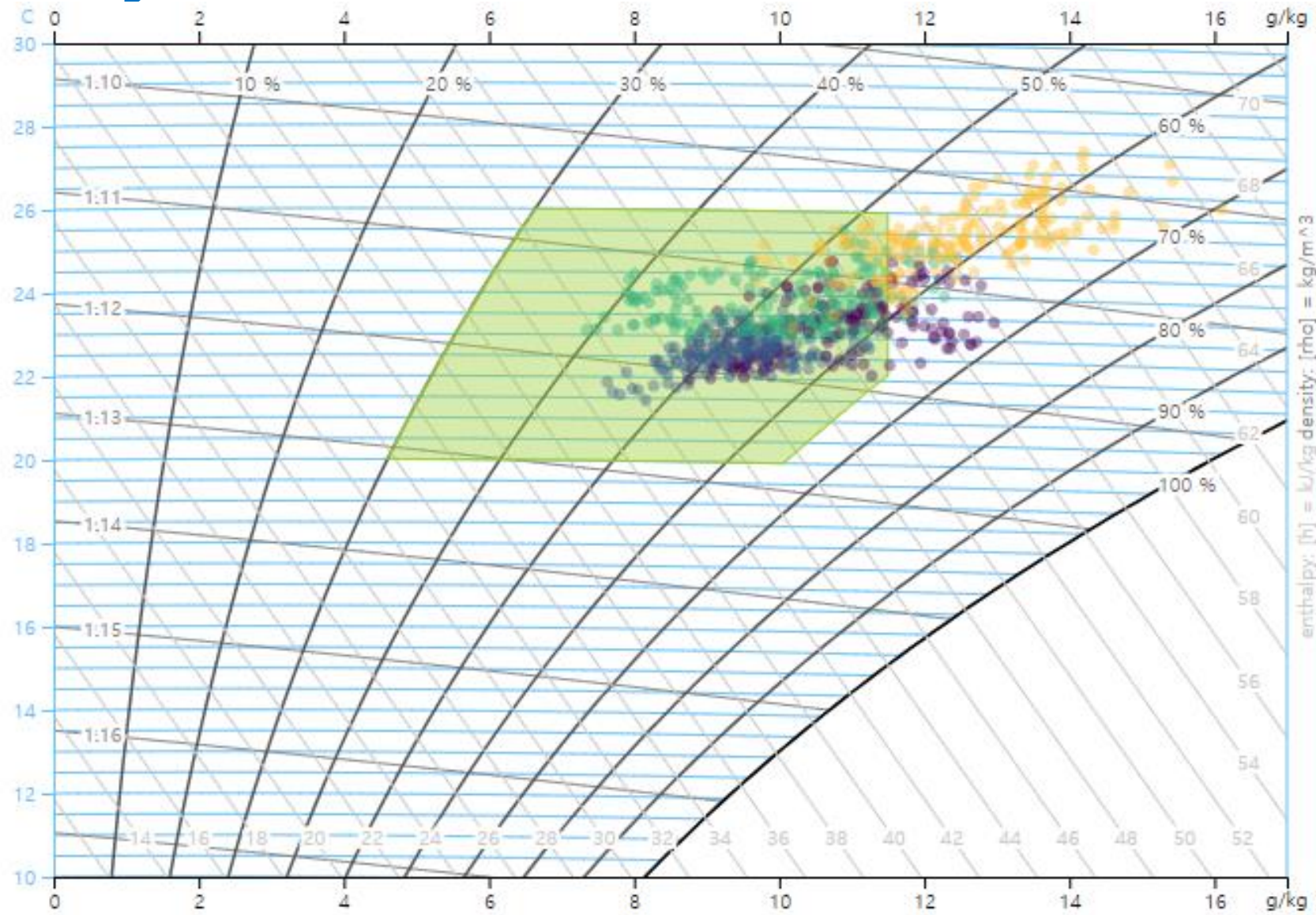
Heatmap Median per hour by Weekday and Season



# Typical Daily Profiles



# Typical Daily Profiles



# Thank you