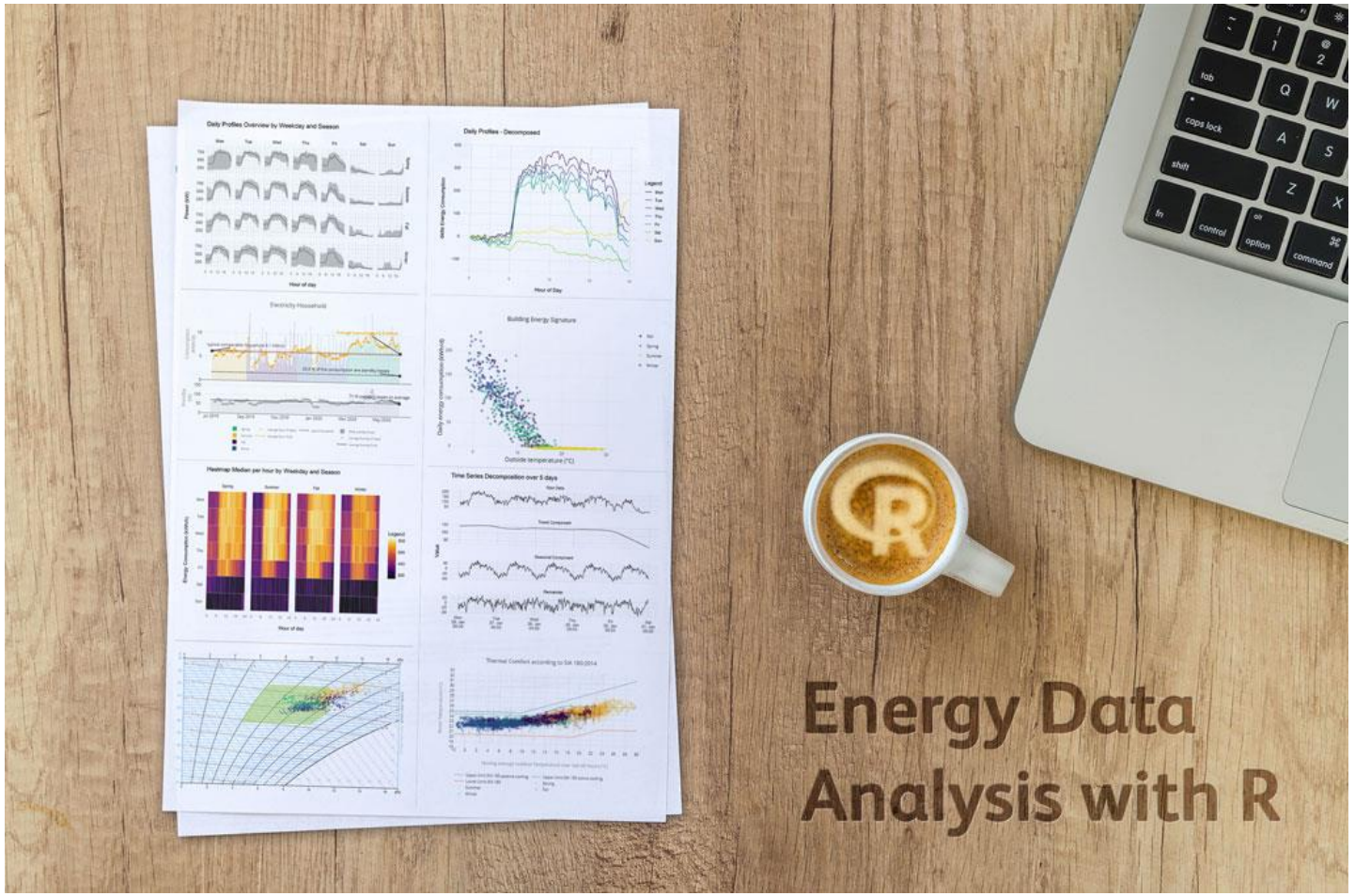
4. R로 하는 에너지 데이터 분석



박경규

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)

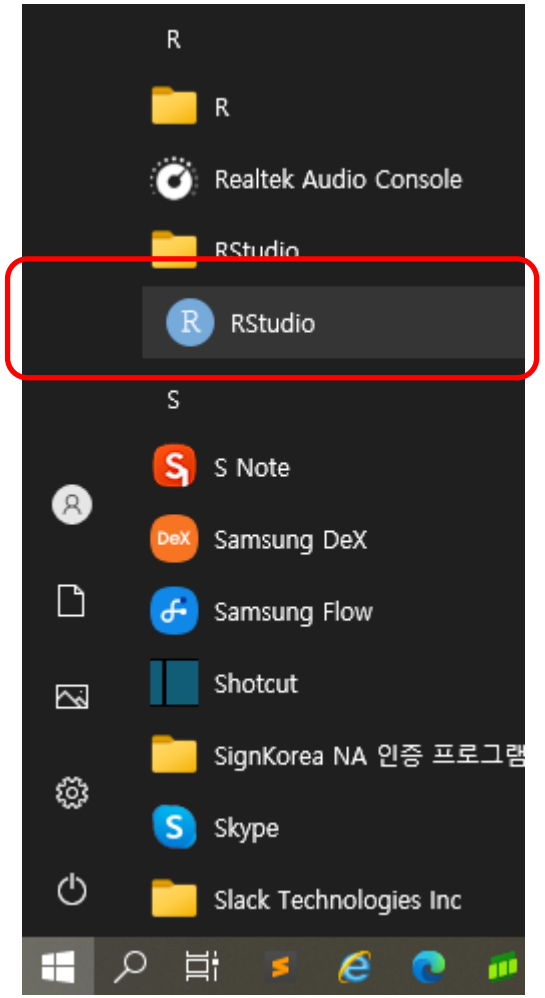
  Download R for macOS

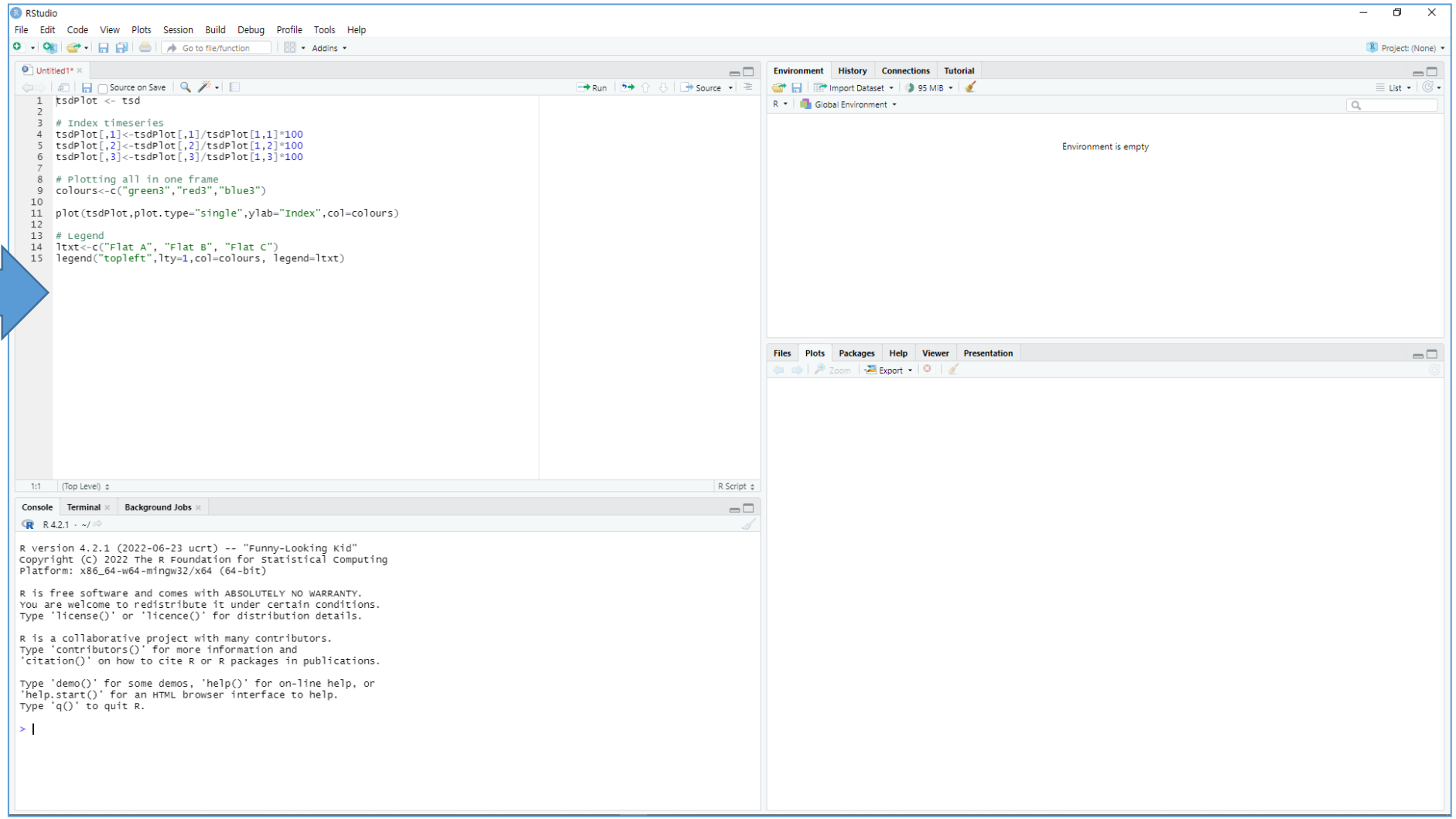
  Download R for Windows

https://www.rstudio.com/

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

R Studio(IDE)


R
R
Realtek Audio Console
RStudio
RStudio
S
S Note
Samsung DeX
Samsung Flow
Shotcut
SignKorea NA 인증 프로그램
Skype
Slack Technologies Inc


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Untitled1*
Source on Save
Run
Source
tsdPlot <- tsd
# Index timeseries
tsdPlot[,1]<-tsdPlot[,1]/tsdPlot[1,1]*100
tsdPlot[,2]<-tsdPlot[,2]/tsdPlot[1,2]*100
tsdPlot[,3]<-tsdPlot[,3]/tsdPlot[1,3]*100
# Plotting all in one frame
colours<-c("green3","red3","blue3")
plot(tsdPlot,plot.type="single",ylab="Index",col=colours)
# Legend
ltxt<-c("Flat A", "Flat B", "Flat C")
legend("topleft",lty=1,col=colours, legend=ltxt)
(Top Level)
R Script
Environment History Connections Tutorial
Import Dataset
Global Environment
Environment is empty
Files Plots Packages Help Viewer Presentation
Export
Console Terminal Background Jobs
R 4.2.1 · ~/
R version 4.2.1 (2022-06-23 ucrt) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>

패키지 설치

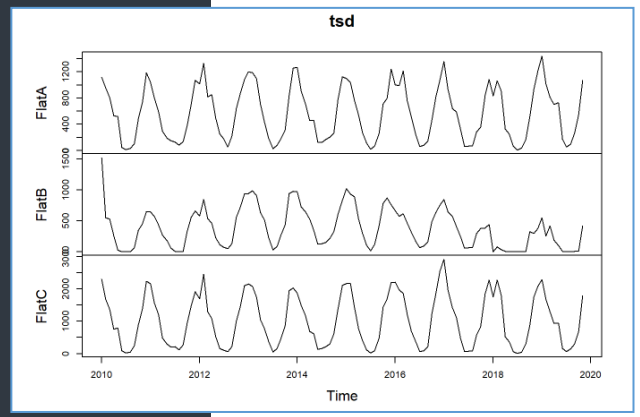
```
install.packages("ggplot2")


install.packages("devtools")
library(devtools)
install_github("hslu-ige-laes/redutils")
```

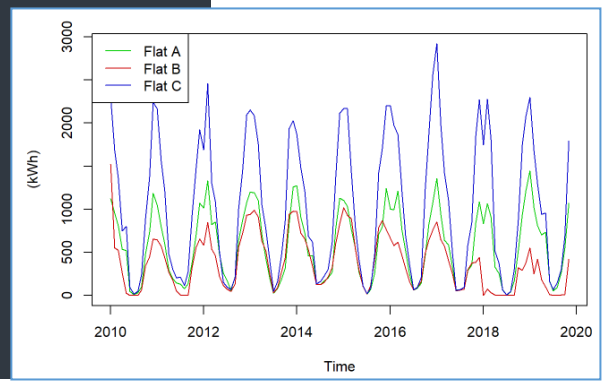
데이터 로드

```r
# read data from current folder
df <- read.csv("datafile.csv")

# read data from a specific folder
df <- read.csv("C:/Desktop/datafile.csv")

# read data from a file in the internet
df <- read.csv2("https://github.com/hslu-ige-laes/edar/raw/master/sampleData/flatElectricity.csv")

# add arguments on how to parse the content
df <- read.csv("datafile.csv",
               header=FALSE,
               stringsAsFactors=FALSE,
               sep =",",
               na.strings = c("", "NA"))
```

EDA(Explorative Data Analysis)

```r
df <- read.csv2("https://github.com/hslu-ige-laes/edar/raw/master/sampleData/flatHeatAndHotWater.csv",
                stringsAsFactors=FALSE)
# filter flat
df <- df %>% select(timestamp, Adr01_energyHeat, Adr02_energyHeat, Adr03_energyHeat)
#df <- df %>% filter(timestamp > "2014-12-01")

df <- df %>% dplyr::mutate(FlatA = lead(Adr01_energyHeat) - Adr01_energyHeat)
df <- df %>% dplyr::mutate(FlatB = lead(Adr02_energyHeat) - Adr02_energyHeat)
df <- df %>% dplyr::mutate(FlatC = lead(Adr03_energyHeat) - Adr03_energyHeat)

# remove counter value column
df <- df %>% select(-Adr01_energyHeat, -Adr02_energyHeat, -Adr03_energyHeat) %>% na.omit()
df$timestamp <- parse_date_time(df$timestamp,
                                orders = "YmdHMS",
                                tz = "Europe/Zurich")

tsd <- ts(df %>% select(-timestamp), frequency = 12, start = min(year(df$timestamp)))

# plot FlatA_Ele
plot(tsd)

# Plotting all in one frame
colours<-c("green3","red3","blue3")

plot(tsd,plot.type="single",ylab="(kWh)",col=colours)

# Legend
ltxt<-c("Flat A", "Flat B", "Flat C")
legend("topleft",lty=1,col=colours, legend=ltxt)
```

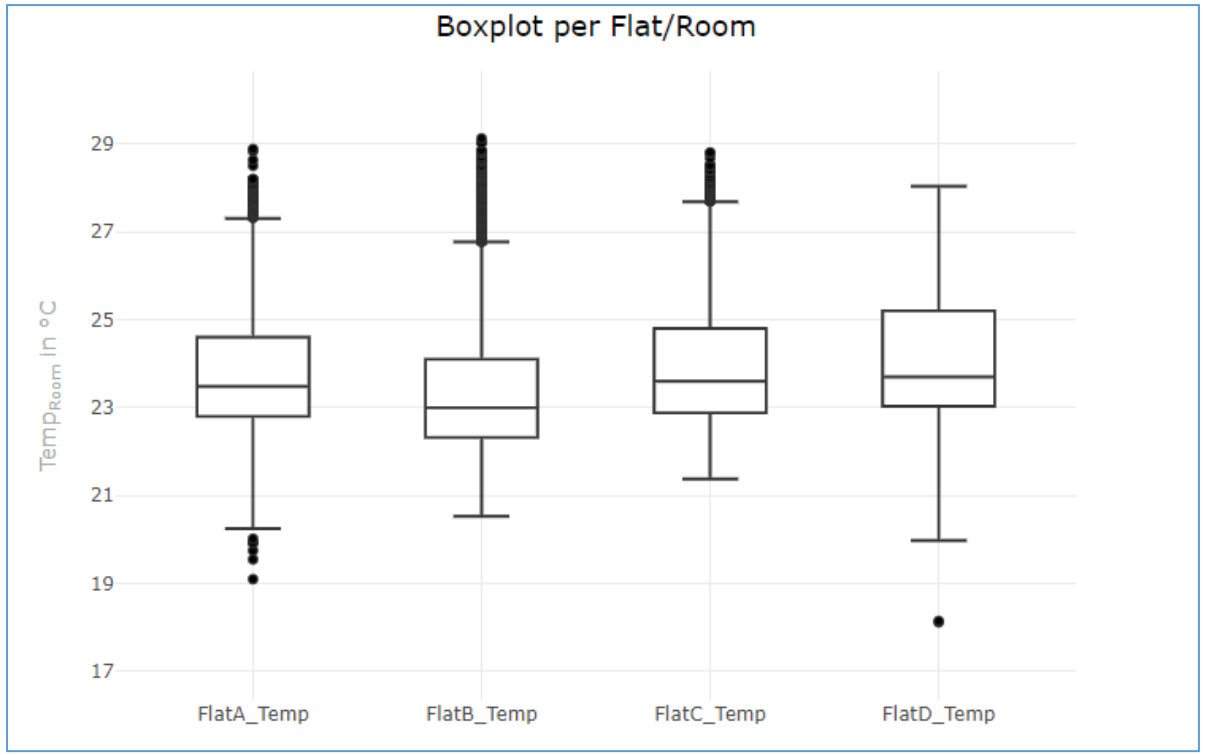




https://hslu-ige-laes.github.io/edar/quick-data-visualizations.html

Boxplot

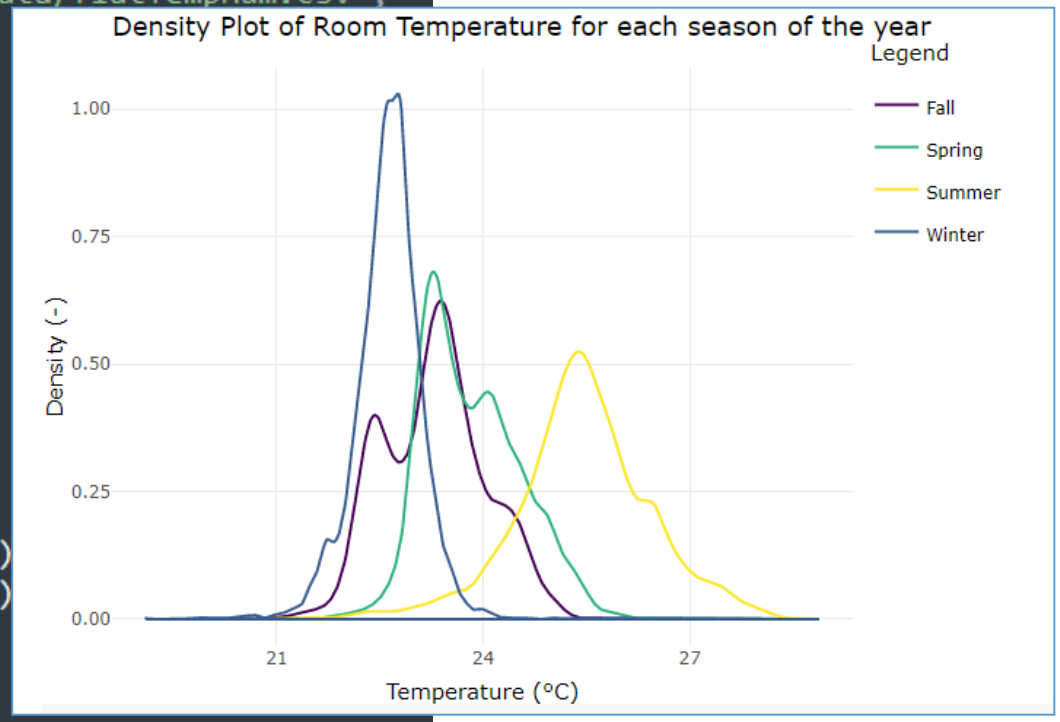
```r
df <- read.csv("https://github.com/hslu-ige-laes/edar/raw/master/sampleData/flatTempHum.csv",
                        stringsAsFactors=FALSE,
                        sep =";")

df$time <- parse_date_time(df$time,
                           order = "YmdHMS",
                           tz = "Europe/Zurich")

df <- df %>%
  select(time, FlatA_Temp, FlatB_Temp, FlatC_Temp, FlatD_Temp)

df <- as.data.frame(tidyr::pivot_longer(df,
                                        cols = -time,
                                        names_to = "room",
                                        values_to = "Temp",
                                        values_drop_na = TRUE))

minY <- round(min(df %>% select(Temp), na.rm = TRUE) - 1, digits = 0)
maxY <- round(max(df %>% select(Temp), na.rm = TRUE) + 1, digits = 0)


plot <- ggplot(df, aes(x = room, y = Temp)) +
  geom_boxplot(outlier.alpha = 0.01, outlier.shape = 1, outlier.colour = "darkgrey") +
  scale_y_continuous(limits=c(minY,maxY), breaks = seq(minY, maxY, by = 2)) +
  ggtitle("Boxplot per Flat/Room") +
  theme_minimal() +
  theme(
    legend.position="none",
    plot.title = element_text(hjust = 0.5)
  )

yaxis <- list(
  title = "Temp<sub>Room</sub> in \u00B0C\n",
  automargin = TRUE,
  titlefont = list(size = 14, color = "darkgrey")
)

# create interactive plot
ggplotly(plot + ylab(" ") + xlab(" ")) %>%
  plotly::config(modeBarButtons = list(list("toImage")),
                 displaylogo = FALSE,
                 toImageButtonOptions = list(
                   format = "svg"
                 )
  ) %>%
  layout(yaxis = yaxis)
```



https://hslu-ige-laes.github.io/edar/boxplot.html

Density Plot

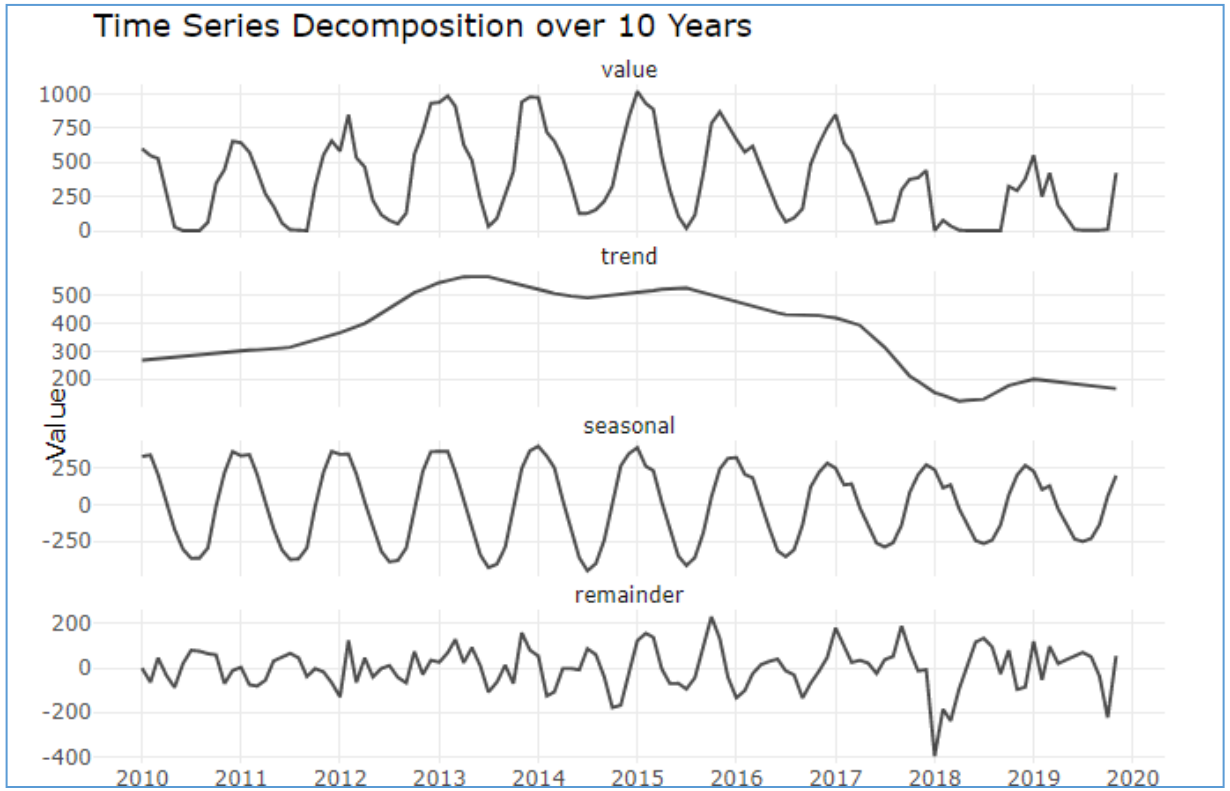
```r
# load time series data and aggregate daily mean values
library(dplyr)
library(lubridate)
library(redutils)

# read and print data
df <- read.csv("https://github.com/hslu-ige-laes/edar/raw/master/sampleData/flatTempHum.csv",
               stringsAsFactors=FALSE,
               sep =";")

# select temperature and remove empty cells
df <- df %>% select(time, FlatA_Temp) %>% na.omit()

colnames(df) <- c("time", "value")

df$season = getSeason(df$time)

# static chart with ggplot
plot <- ggplot(df) +
  geom_density(aes(x = value, colour = season)) +
  scale_color_manual(values=c("#440154", "#2db27d", "#fde725", "#365c8d")) +
  ggtitle("Density Plot of Room Temperature for each season of the year") +
  labs(x = "Temperature (\u00B0C)",
       y = "Density (-)",
       colour = "Legend") +
  theme_minimal()

# interactive chart
plotly::ggplotly(plot)
```



https://hslu-ige-laes.github.io/edar/density-plot-season.html

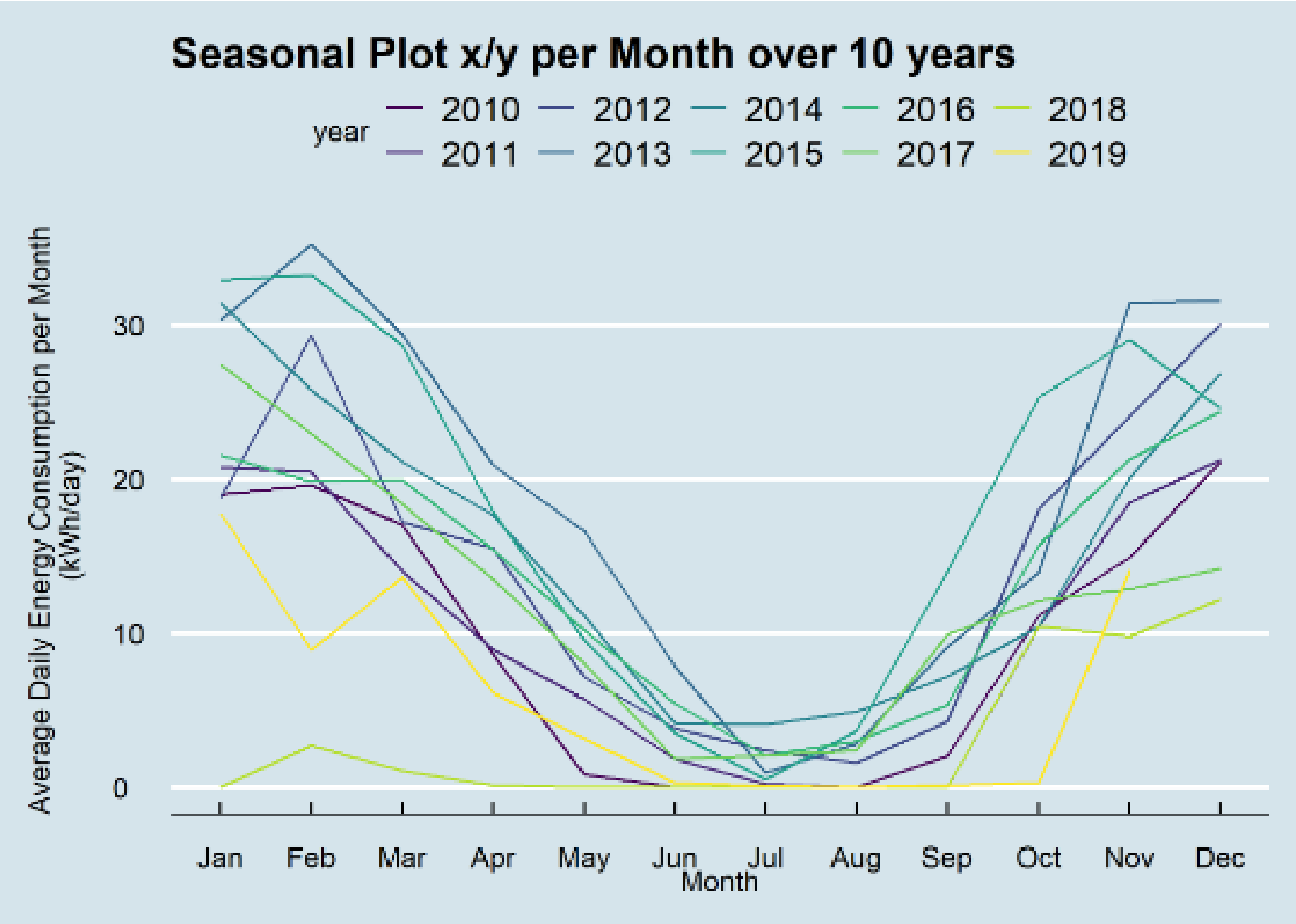
Time Series Decomposition

```r
df <- read.csv2("https://github.com/hslu-ige-laes/edar/raw/master/sampleData/flatHeatAndHotWater.csv",
                stringsAsFactors=FALSE)

# filter flat
df <- df %>% select(timestamp, Adr02_energyHeat)
colnames(df) <- c("Time", "meterValue")
df$Time <- parse_date_time(df$Time, orders = "YmdHMS", tz = "Europe/Zurich")

# calculate consumption value per month
df <- df %>% dplyr::mutate(value = lead(meterValue) - meterValue)

# remove counter value column
df <- df %>% select(-meterValue) %>% na.omit()
df[1,2] <- 600
df.ts <- ts(df %>% select(value) %>% na.omit(), frequency = 12, start = min(year(df$Time)))
df.decompose <- df.ts[,1] %>%
  stl(s.window = 7)

df.decompose <- df.decompose$time.series
df.decompose <- as.data.frame(df.decompose)
df.decompose <- cbind(df, df.decompose)
data <- as.data.frame(tidyr::pivot_longer(df.decompose,
                                          cols = -Time,
                                          names_to = "Component",
                                          values_to = "Value",
                                          values_drop_na = TRUE)
)
data$component <- as.factor(data$Component)
data$component <- factor(data$Component, c("value",
                                           "trend",
                                           "seasonal",
                                           "remainder"))
data$Value <- round(data$Value, digits = 1)

plot <- ggplot(data) +
  geom_path(aes(x = Time,
                y = Value
  ),
  color = "black",
  alpha = 0.7) +
  facet_wrap(~component, ncol = 1, scales = "free_y") +
  scale_x_datetime(date_breaks = "years" , date_labels = "%Y") +
  theme_minimal() +
  theme(panel.spacing = unit(1, "lines"),
        legend.position = "none") +
  labs(x = "") +
  ggtitle("Time Series Decomposition over 10 Years")
ggplotly(plot)
```



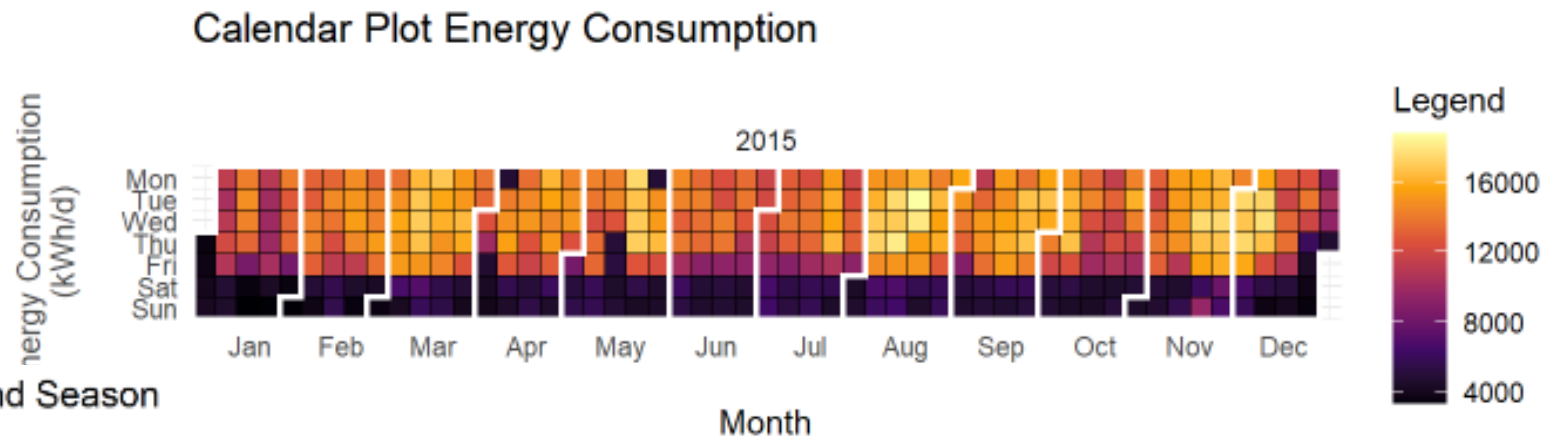
https://hslu-ige-laes.github.io/edar/time-series-decomposition.html

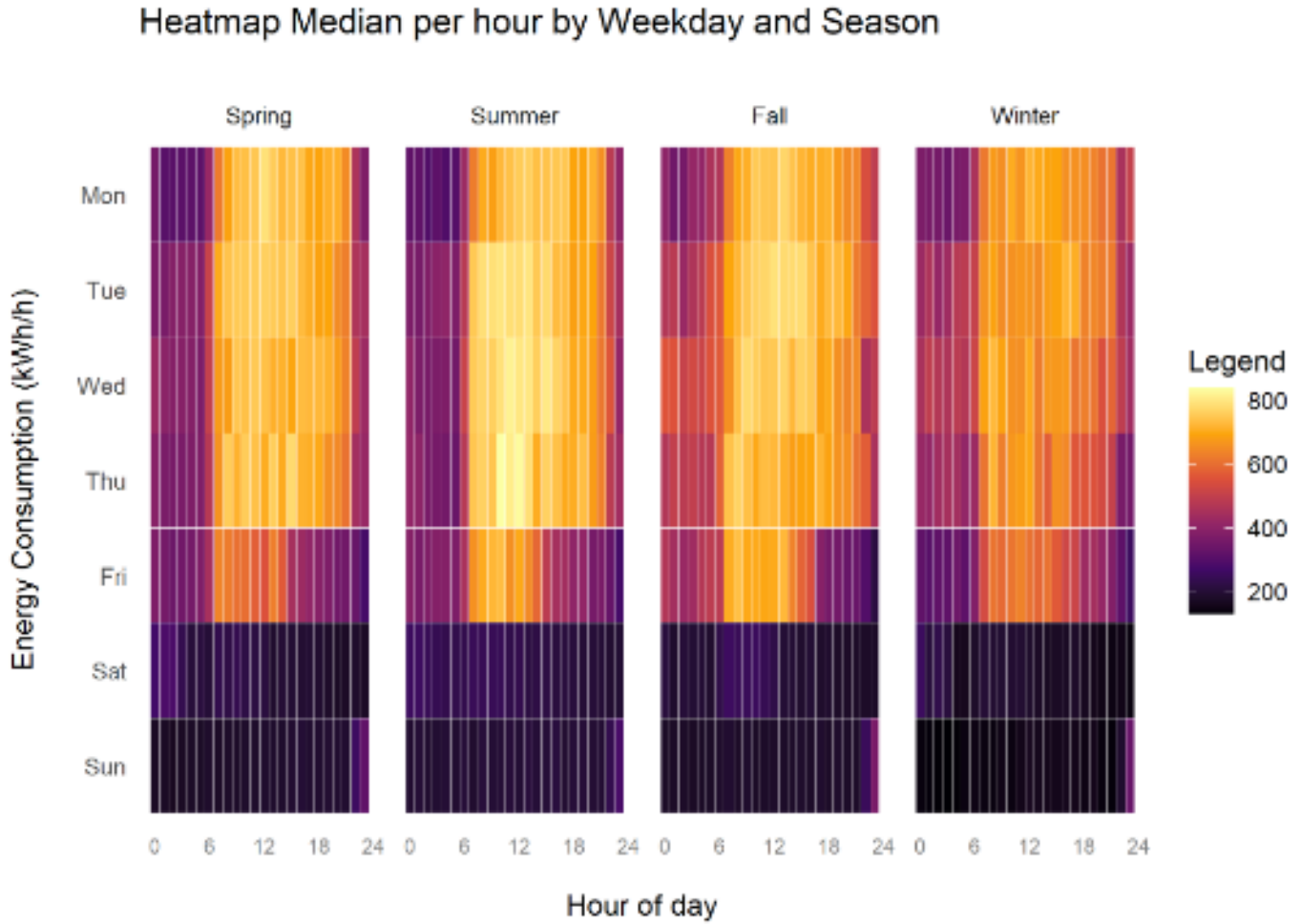
Seasonal Plots



https://hslu-ige-laes.github.io/edar/seasonal-plots.html

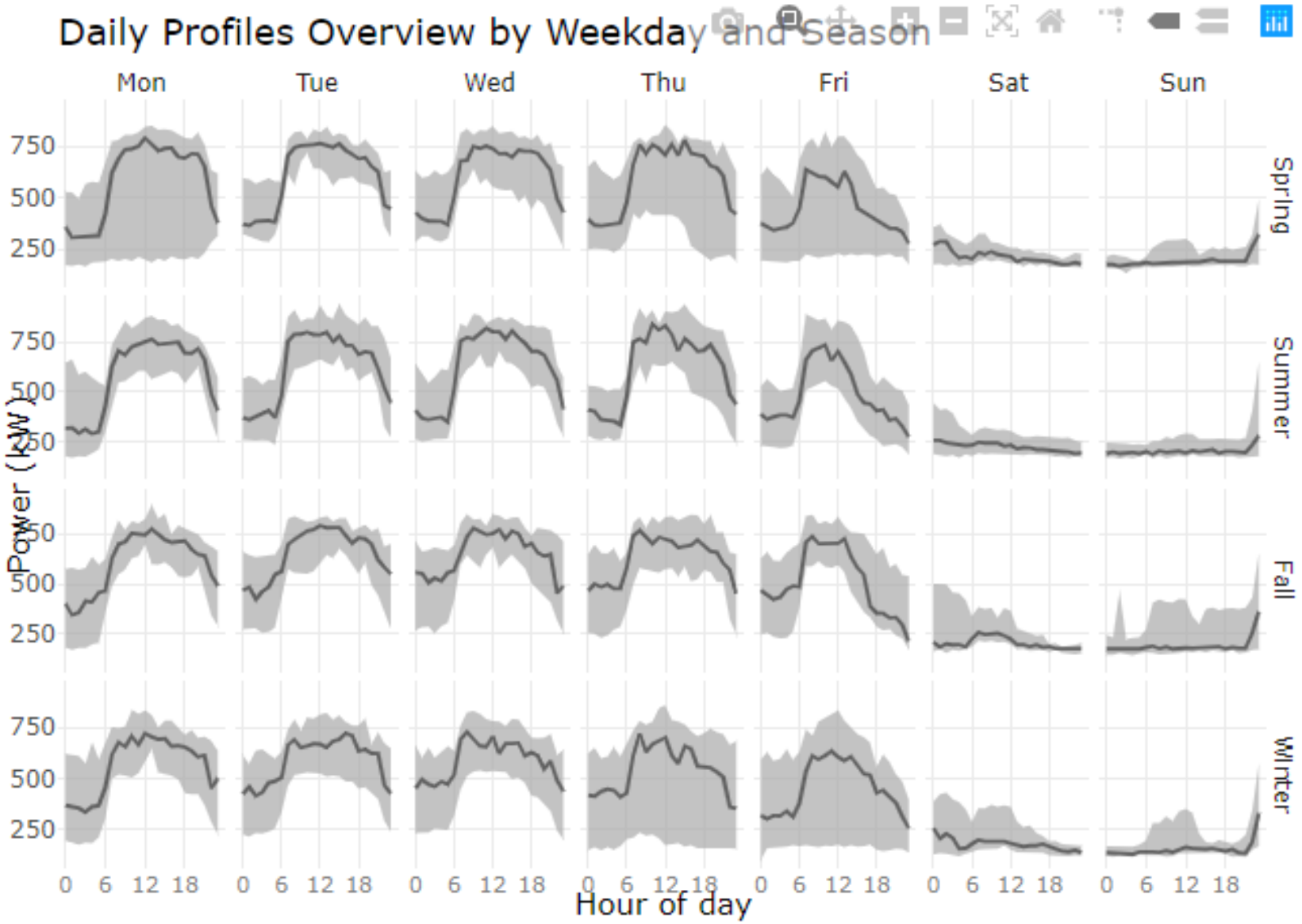
Heat Map





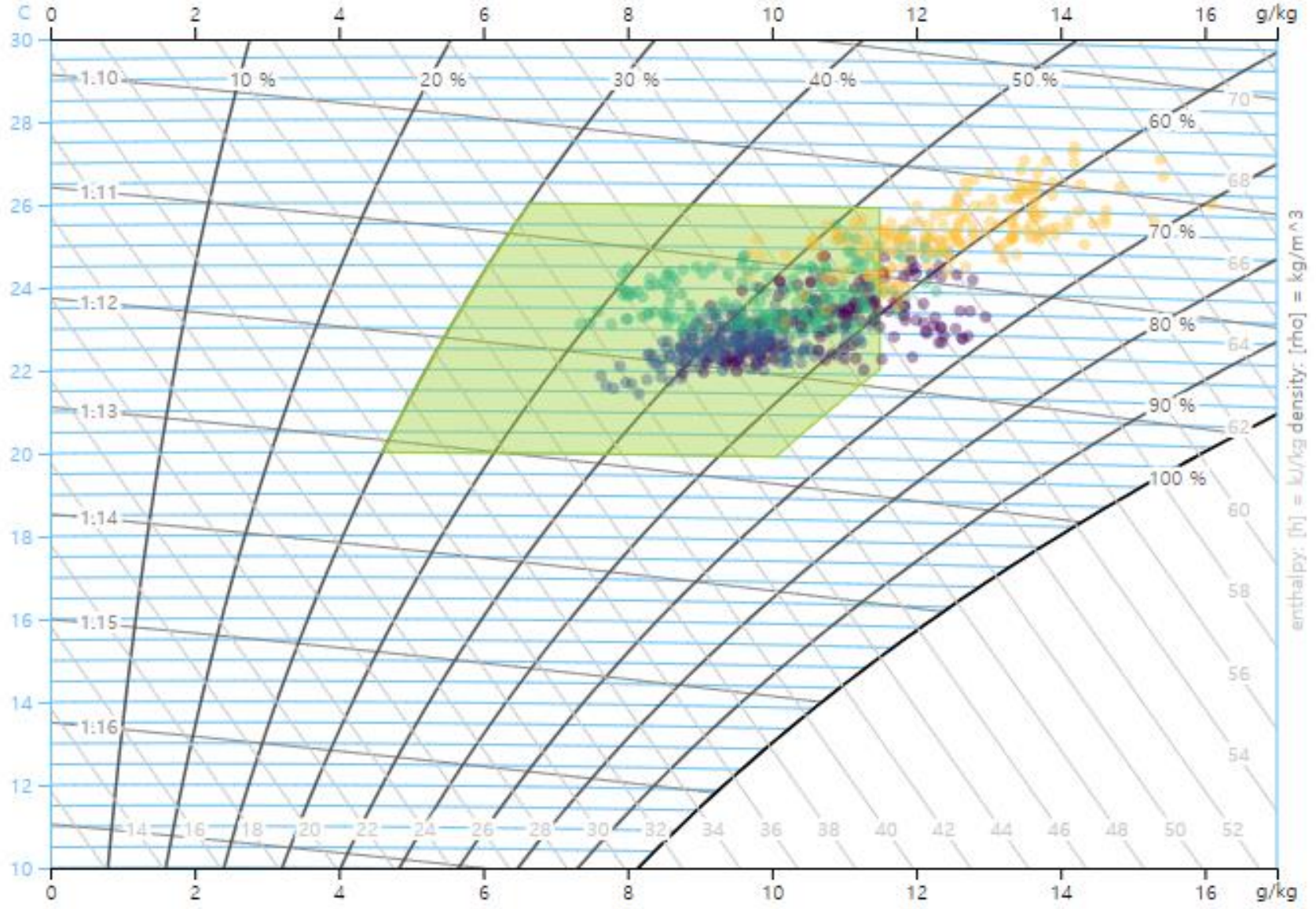
https://hslu-ige-laes.github.io/edar/heat-maps.html

Typical Daily Profiles



https://hslu-ige-laes.github.io/edar/typical-daily-profiles.html

Typical Daily Profiles



https://hslu-ige-laes.github.io/edar/comfort-plots.html

Thank you

kgpark88@gmail.com