Energy Data Analysis with R

Lucerne University of Sciences and Arts, Engineering & Architecture, Institute of Building Techn

2020-10-30

Contents

Preface

This document gives you a short overview of the statistical software R and its ability to analyze and visualize time series in the context of building energy and comfort.

Disclaimer The authors decline any liability or responsibility in connection with the published documentation

Chapter 1

Introduction

This book is aimed at R beginners as well as advanced R users. The recipes in this book will show you how to complete certain tasks. Simple examples are shown so that you can understand the basic principle and reproduce it with your own data.

The book is strongly inspired by the [R Graphics Cookbook] (https://r-graphics.org/). The goal of this book is to additionaly provide specific recipes for energy and comfort related tasks that can be performed with the R or RStudio program.

1.1 Why R and RStudio

Spreadsheet programs like Excel quickly reach their limits when working with large data sets or creating complex graphics. Also the interactive ability of the graphics is limited. The open source programming language R and its graphical user interface RStudio offer many more possibilities for data analysis and data visualization.

Chapter 2

Explorative Data Analysis

2.1 Get overview

Get an overview of the whole data set and specific series of it

2.1.1 Load data

Load test data set in a data frame (e.g. from a csv-file)

```
df <- read.csv("https://github.com/retomarek/r/raw/master/datasets/buildingMonitoringTestDataSet.
```

2.1.2 Names

show the column headers of the data frame

```
names(df)
```

```
##  [1] "time"                  "WthStnPress"           "WthStnHum"
##  [4] "WthStnRain"            "WthStnSolRad"          "WthStnTemp"
##  [7] "WthStnWindDir"         "WthStnWindSpd"         "BldgEnergyHotwater"
## [10] "BldgEnergyHeating"     "FlatHum"               "FlatTemp"
## [13] "FlatVolFlowColdwater" "FlatVolFlowHotwater"
```

2.1.3 Structure

show the structure of the data frame

```
str(df)
```

```
## 'data.frame':    16394 obs. of  14 variables:
##  $ time               : chr  "2018-09-30T22:00:00.000Z" "2018-09-30T23:00:00.000Z"
##  $ WthStnPress        : num  1012 1012 1011 1011 1011 ...
##  $ WthStnHum          : num  87 87.5 87.5 86.5 88 89 86.5 81 78 80.5 ...
##  $ WthStnRain         : num  0.8 1.1 0.5 0.5 0.6 0.1 0.2 0 0 0 ...
##  $ WthStnSolRad       : num  0 0 0 0 0 0 0 0 3 24.5 ...
##  $ WthStnTemp         : num  12.8 12.4 11.9 11.9 11.6 ...
##  $ WthStnWindDir      : num  157.5 11.2 146.2 157.5 146.2 ...
##  $ WthStnWindSpd      : num  3.2 1.6 2.4 0.8 2.4 0.8 0.8 3.2 4 3.2 ...
##  $ BldgEnergyHotwater  : num  0 19 0 0 0 ...
##  $ BldgEnergyHeating   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ FlatHum             : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ FlatTemp            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ FlatVolFlowColdwater: num  0.006 0 0 0 0.006 ...
##  $ FlatVolFlowHotwater : num  0 0 0 0 0 ...
```

2.1.4 Head/Tail

```
head(df)
```

```
##                       time WthStnPress WthStnHum WthStnRain WthStnSolRad
## 1 2018-09-30T22:00:00.000Z     1012.30      87.0        0.8            0
## 2 2018-09-30T23:00:00.000Z     1011.90      87.5        1.1            0
## 3 2018-10-01T00:00:00.000Z     1011.45      87.5        0.5            0
## 4 2018-10-01T01:00:00.000Z     1010.90      86.5        0.5            0
## 5 2018-10-01T02:00:00.000Z     1010.55      88.0        0.6            0
## 6 2018-10-01T03:00:00.000Z     1010.20      89.0        0.1            0
##   WthStnTemp WthStnWindDir WthStnWindSpd BldgEnergyHotwater BldgEnergyHeating
## 1      12.80        157.50           3.2                  0                 0
## 2      12.35         11.25           1.6                 19                 0
## 3      11.90        146.25           2.4                  0                 0
## 4      11.90        157.50           0.8                  0                 0
## 5      11.60        146.25           2.4                  0                 0
## 6      11.75         22.50           0.8                  0                 0
##   FlatHum FlatTemp FlatVolFlowColdwater FlatVolFlowHotwater
## 1      NA       NA                0.006                   0
## 2      NA       NA                0.000                   0
## 3      NA       NA                0.000                   0
## 4      NA       NA                0.000                   0
## 5      NA       NA                0.006                   0
## 6      NA       NA                0.000                   0
```

```
tail(df)
```

```
##                           time WthStnPress WthStnHum WthStnRain WthStnSolRad
## 16389 2020-08-13T18:00:00.000Z    1011.650     74.75    2.19964            9
## 16390 2020-08-13T19:00:00.000Z    1012.000     79.00    2.19964            0
## 16391 2020-08-13T20:00:00.000Z    1011.950     78.25    2.19964            0
## 16392 2020-08-13T21:00:00.000Z    1012.025     76.50    2.19964            0
## 16393 2020-08-13T22:00:00.000Z    1012.250     73.00    0.00000            0
## 16394 2020-08-13T23:00:00.000Z          NA        NA         NA           NA
##       WthStnTemp WthStnWindDir WthStnWindSpd BldgEnergyHotwater
## 16389     22.000        162.00      0.000000                 NA
## 16390     20.175        124.25      1.609340                 NA
## 16391     19.350        125.00      0.402335                 NA
## 16392     19.900         93.00      1.609340                 NA
## 16393     20.625        116.25      2.414010                 NA
## 16394         NA            NA            NA                 NA
##       BldgEnergyHeating FlatHum FlatTemp FlatVolFlowColdwater
## 16389                NA      NA       NA                   NA
## 16390                NA      NA       NA                   NA
## 16391                NA      NA       NA                   NA
## 16392                NA      NA       NA                   NA
## 16393                NA      NA       NA                   NA
## 16394                NA      NA       NA                   NA
##       FlatVolFlowHotwater
## 16389                  NA
## 16390                  NA
## 16391                  NA
## 16392                  NA
## 16393                  NA
## 16394                  NA
```

2.1.5 Five number summary

reveals details of a specific series

```
summary(df$WthStnTemp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   -5.25    5.50   11.25   11.99   17.35   40.30      12
```

2.2 Data Wrangling

2.2.1 season from date

```
# install redutils library
# devtools::install_github("retomarek/redutils", ref = "master")

# get season from a date
redutils::season(as.Date("2019-04-01"))
```

```
## [1] "Spring"
```

```
redutils::season(as.Date("2019-04-01"), c("Winter","Frühling","Sommer","Herbst"))
```

```
## [1] "Frühling"
```

```
# apply it for a data frame
df.season <- dplyr::mutate(df, season = redutils::season(df$time))
head(df.season)
```

```
##                       time WthStnPress WthStnHum WthStnRain WthStnSolRad
## 1 2018-09-30T22:00:00.000Z     1012.30      87.0        0.8            0
## 2 2018-09-30T23:00:00.000Z     1011.90      87.5        1.1            0
## 3 2018-10-01T00:00:00.000Z     1011.45      87.5        0.5            0
## 4 2018-10-01T01:00:00.000Z     1010.90      86.5        0.5            0
## 5 2018-10-01T02:00:00.000Z     1010.55      88.0        0.6            0
## 6 2018-10-01T03:00:00.000Z     1010.20      89.0        0.1            0
##   WthStnTemp WthStnWindDir WthStnWindSpd BldgEnergyHotwater BldgEnergyHeating
## 1      12.80        157.50           3.2                  0                 0
## 2      12.35         11.25           1.6                 19                 0
## 3      11.90        146.25           2.4                  0                 0
## 4      11.90        157.50           0.8                  0                 0
## 5      11.60        146.25           2.4                  0                 0
## 6      11.75         22.50           0.8                  0                 0
##   FlatHum FlatTemp FlatVolFlowColdwater FlatVolFlowHotwater season
## 1      NA       NA                0.006                   0   Fall
## 2      NA       NA                0.000                   0   Fall
## 3      NA       NA                0.000                   0   Fall
## 4      NA       NA                0.000                   0   Fall
## 5      NA       NA                0.006                   0   Fall
## 6      NA       NA                0.000                   0   Fall
```

Chapter 3

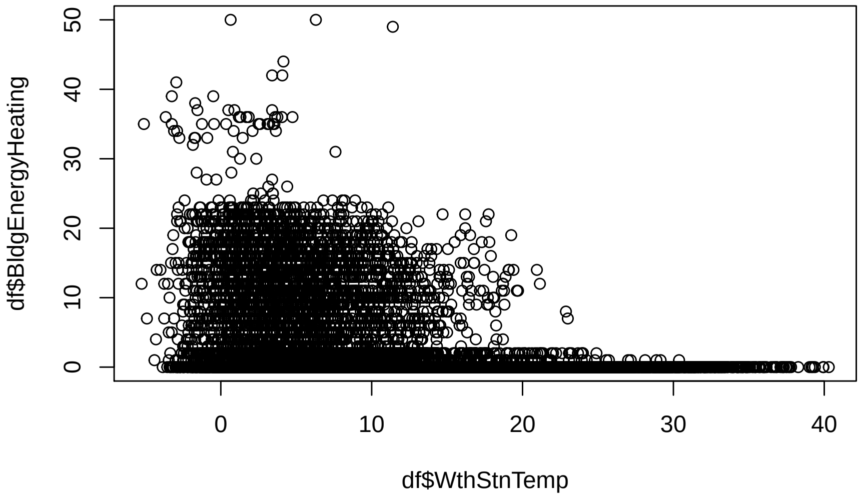
Data Visualizations

3.1 General Plots

3.1.1 Scatterplot

3.1.1.1 plot()

```
# load data set
df <- read.csv("https://github.com/retomarek/r/raw/master/datasets/buildingMonitoringTestDataSet.

# crate simple scatterplot
plot(df$WthStnTemp, df$BldgEnergyHeating)
```

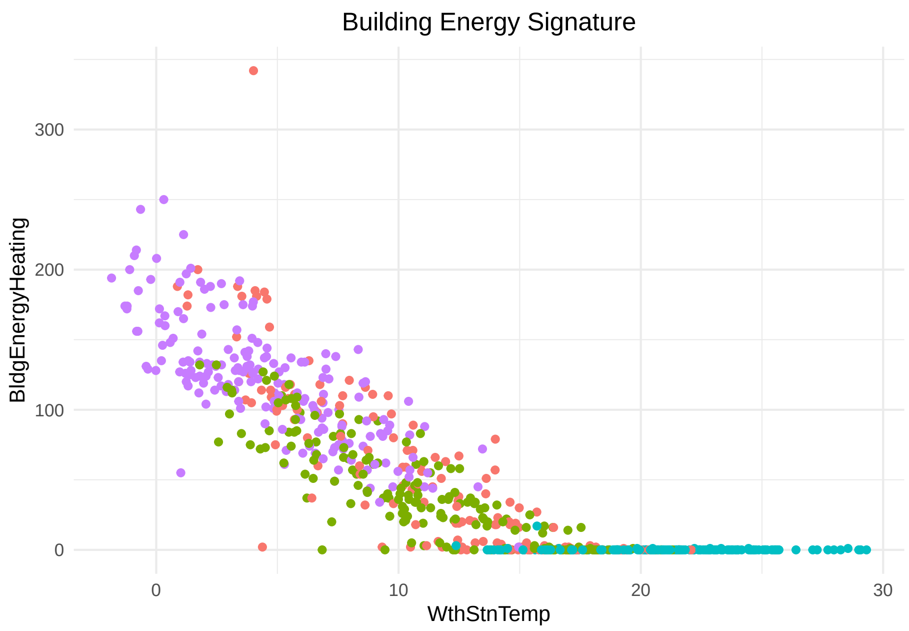


3.2 Building Energy Signature

3.2.1 static

```
library(ggplot2)
library(plotly)
library(dplyr)
library(redutils)
library(lubridate)

# load data set
df <- read.csv("https://github.com/retomarek/r/raw/master/datasets/buildingMonitoringT
               stringsAsFactors=FALSE, sep =",")

# select data and calculate season
data <- df %>%
  select(time, WthStnTemp, BldgEnergyHeating) %>%
  mutate(season = redutils::season(df$time)) %>%
  na.omit()

# Aggregate data to daily values
data$time <- parse_date_time(data$time, "YmdHMS", tz = "Europe/Zurich")
```

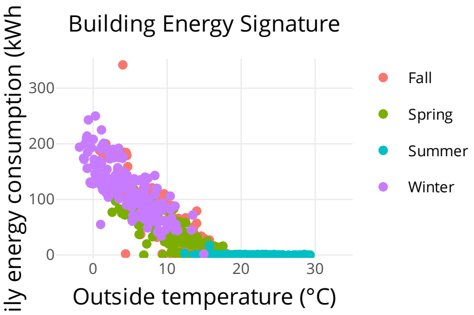
```
data$year <- as.Date(cut(data$time, breaks = "year"))
data$month <- as.Date(cut(data$time, breaks = "month"))
data$day <- as.Date(cut(data$time, breaks = "day"))

data <- data %>%
  select(day, WthStnTemp, BldgEnergyHeating, season) %>%
  group_by(day) %>%
  mutate(WthStnTemp = mean(WthStnTemp))

data <- data %>%
  group_by(day) %>%
  mutate(BldgEnergyHeating = sum(BldgEnergyHeating))

data <- data %>%
  unique()

# static chart with ggplot
p <- ggplot2::ggplot(data) +
  ggplot2::geom_point(aes(x = WthStnTemp,
                          y = BldgEnergyHeating, color=season,
                          text = paste("</br>Date:  ", as.Date(data$day),
                                       "</br>Temp: ", round(data$WthStnTemp, digits = 1), "\u00B0
                                       "</br>Energy: ", round(data$BldgEnergyHeating, digits = 0)
                                       "</br>Season: ", data$season))
                      ) +
  ggtitle("Building Energy Signature") +
        theme_minimal() +
        theme(
          legend.position="none",
          plot.title = element_text(hjust = 0.5)
        )
p
```



3.2.2 interactive

Make ggplot2 chart above interactive with plotly

```r
# continuation from upper ggplot code section
plotly::ggplotly(p, tooltip = c("text")) %>%
  layout(xaxis = list(title = "Outside temperature (\u00B0C)",
                      range = c(min(-5,min(data$WthStnTemp)), max(35,max(data$WthStnTem
         yaxis = list(title = "Daily energy consumption (kWh/d)",
                      range = c(-5, max(data$BldgEnergyHeating) + 10)),
         showlegend = TRUE
         ) %>%
  plotly::config(displayModeBar = FALSE, displaylogo = FALSE)
```


Building Energy Signature
ily energy consumption (kWh
300
200
100
0
0
10
20
30
Outside temperature (°C)
● Fall
● Spring
● Summer
● Winter

Appendix A

Installing R and R Studio

- Before we can start the first analysis, we have to install "R" and "RStudio".
- "R" is a programming language used for statistical computing while "RStudio" provides a graphical user interface.
- "R" may be used without "RStudio", but "RStudio" may not be used without "R". Both, "R" and "RStudio" are free of charge and there are no licencse fees.

A.1 How to Download and Install R

A.1.1 Windows

1. Open https://cran.r-project.org/bin/windows/base/ and press the link "Download R…"
2. Run the downloaded installer file and follow the installation wizard

The wizard will install R into your "Program Files" folders and add a shortcut in your Start menu. Note that you will need to have all necessary administration rights to install new software on your machine.

A.1.2 Mac OSX

1. Open https://cran.r-project.org/bin/macosx/ and download the latest *.pkg file
2. Run the downloaded installer file and follow the installation wizard

The installer allows you to customize your installation. However the default values will be suitable for most users.

A.1.3 Linux

R is part of many Linux distributions, therefore you should check with your Linux package management system if it's already installed.

The CRAN website provides files to build R from source on Debian, Redhat, SUSE, and Ubuntu systems under the link "Download R for Linux"

- Open https://cran.r-project.org/bin/linux/ and then follow the directory trail to the version of Linux you wish to install R on top of

The exact installation procedure will vary depending on your Linux operating system. CRAN supports the process by grouping each set of source files with documentation or README files that explain how to install on your system.

A.2 How to Download and Install RStudio

R Studio is a development environment for R.

1. Open https://rstudio.com/products/rstudio/download/ and download "RStudio Desktop Open Source"
2. Follow the on-screen instructions
3. Once you have installed R Studio, you can run it like any other application via