R Assignment 1 Group 7

Pavlo N., Chiara S., Viktor S.

2022-06-05

Make sure to install and load following packages first:

```
library(dplyr)
library(knitr)
library(stringr)
library(here)
library(stringi)
library(janitor)
library(magrittr)
library(lubridate)
library(tibble)
library(kableExtra)
library(ggplot2)
library(zoo)
library(stargazer)
library(modelr)
library(tidyr)
library(tidyverse)
library(rvest)
library(DBI)
library(RSQLite)
```

Exercise Number 1)

First prepare to scrape the data for **Essen** on page 1 of immowelt. The scraped data will contain 40 observations for each city (Essen and Bochum).

```
url_essen_1 <- ("https://www.immowelt.de/liste/essen/wohnungen/mieten?sort=relevanz")
url_essen_2 <- ("https://www.immowelt.de/liste/essen/wohnungen/mieten?d=true&sd=DESC&sf=RELE

html_essen_1 <- read_html(url_essen_1)
html_essen_2 <- read_html(url_essen_2)
```

a) Now scrape the URL for Essen

```
get_url_essen_1 <- function(html_essen_1){
  html_essen_1 %>%
```

```
    html_elements(".noProject-eaed4") %>%
    html_attr("href")
}
url.e1<- get_url_essen_1(html_essen_1)

get_url_essen_2 <- function(html_essen_2){
  html_essen_2 %>%
    html_elements(".noProject-eaed4") %>%
    html_attr("href")
}

url.e2 <-get_url_essen_2(html_essen_2)
```

b) Now scrape the title of the first 20 advertisements:

```
get_title_essen_1 <- function(html_essen_1){
  html_essen_1 %>%
    html_elements("h2") %>%
    html_text() %>%
    str_trim()
}
title.e1<-get_title_essen_1(html_essen_1)

get_title_essen_2 <- function(html_essen_2){
  html_essen_2 %>%
    html_elements("h2") %>%
    html_text() %>%
    str_trim()
}

title.e2 <- get_title_essen_2(html_essen_2)
```

c) Area of the city for Essen:

```
get_city_essen_1 <- function(html_essen_1){
  html_essen_1 %>%
    html_elements(".IconFact-e8a23:nth-child(1) span") %>%
    html_text() %>%
    str_replace_all(".*\\,", "") %>%
    str_trim()

}
city.e1<- get_city_essen_1(html_essen_1)

get_city_essen_2 <- function(html_essen_2){
  html_essen_2 %>%
    html_elements(".IconFact-e8a23:nth-child(1) span") %>%
    html_text() %>%
    str_replace_all(".*\\,", "") %>%
```

```
    str_trim()
}
city.e2 <- get_city_essen_2(html_essen_2)
```

d) Zipcodes of Essen:

```
html_test <- read_html("https://www.immowelt.de/expose/25wua5q")
get_zip_essen_1 <- function(html_essen_1){
  html_test %>%
    html_elements("#exposeAddress div") %>%
    html_text()

}
zip.e1 <- get_zip_essen_1(html_essen_1)
```

e) Cold rent for Essen:

```
get_cold_rent_essen_1 <- function(html_essen_1){
  html_essen_1 %>%
    html_elements(".KeyFacts-efbce div:nth-child(1)") %>%
    html_text() %>%
    str_trim()
}
cold_rent.e1 <- get_cold_rent_essen_1(html_essen_1)

get_cold_rent_essen_2 <- function(html_essen_2){
  html_essen_2 %>%
    html_elements(".KeyFacts-efbce div:nth-child(1)") %>%
    html_text() %>%
    str_trim()
}

cold_rent.e2 <-get_cold_rent_essen_2(html_essen_2)
```

f) square meters for Essen:

```
get_square_meters_essen_1 <- function(html_essen_1){
  html_essen_1 %>%
    html_elements(".KeyFacts-efbce div:nth-child(2)") %>%
    html_text() %>%
    str_trim()
}
sq_mt.e1 <- get_square_meters_essen_1(html_essen_1)

get_square_meters_essen_2 <- function(html_essen_2){
  html_essen_2 %>%
    html_elements(".KeyFacts-efbce div:nth-child(2)") %>%
    html_text() %>%
```

```
    str_trim()
}
sq_mt.e2 <-get_square_meters_essen_2(html_essen_2)
```

g) Rooms for Essen:

```
get_rooms_essen_1 <- function(html_essen_1){
  html_essen_1 %>%
    html_elements(".KeyFacts-efbce div:nth-child(3)") %>%
    html_text() %>%
    str_trim()
}
rooms.e1 <- get_rooms_essen_1(html_essen_1)

get_rooms_essen_2 <- function(html_essen_2){
  html_essen_2 %>%
    html_elements(".KeyFacts-efbce div:nth-child(3)") %>%
    html_text() %>%
    str_trim()
}
rooms.e2 <- get_rooms_essen_2(html_essen_2)
```

Now we scrape the same type of data for the city of **Bochum**.

```
url_bochum_1 <- ("https://www.immowelt.de/liste/bochum/wohnungen/mieten?sort=relevanz")
url_bochum_2 <-
  ("https://www.immowelt.de/liste/bochum/wohnungen/mieten?d=true&sd=DESC&sf=RELEVANCE&sp=2")

html_bochum_1 <- read_html(url_bochum_1)
html_bochum_2 <- read_html(url_bochum_2)
```

a) Url for Bochum:

```
get_url_bochum_1 <- function(html_bochum_1){
  html_bochum_1 %>%
    html_elements(".noProject-eaed4") %>%
    html_attr("href")
}

url.b1 <-get_url_bochum_1(html_bochum_1)

get_url_bochum_2 <- function(html_bochum_2){
  html_bochum_2 %>%
    html_elements(".noProject-eaed4") %>%
    html_attr("href")
}
url.b2 <- get_url_bochum_2(html_bochum_2)
```

b) title of the advertisments for Bochum:

```
get_title_bochum_1 <- function(html_bochum_1){
  html_bochum_1 %>%
    html_elements("h2") %>%
    html_text() %>%
    str_trim()
}
title.b1 <- get_title_bochum_1(html_bochum_1)

get_title_bochum_2 <- function(html_bochum_2){
  html_bochum_2 %>%
    html_elements("h2") %>%
    html_text() %>%
    str_trim()
}
title.b2 <-get_title_bochum_2(html_bochum_2)
```

c) area of the city Bochum:

```
get_city_bochum_1 <- function(html_bochum_1){
  html_bochum_1 %>%
    html_elements(".IconFact-e8a23:nth-child(1) span") %>%
    html_text() %>%
    str_replace_all(".*\\,", "") %>%
    str_trim()
}
city.b1 <-get_city_bochum_1(html_bochum_1)

get_city_bochum_2 <- function(html_bochum_2){
  html_bochum_2 %>%
    html_elements(".IconFact-e8a23:nth-child(1) span") %>%
    html_text() %>%
    str_replace_all(".*\\,", "") %>%
    str_trim()
}
city.b2 <-get_city_bochum_2(html_bochum_2)
```

d) Cold rent for Bochum:

```
get_cold_rent_bochum_1 <- function(html_bochum_1){
  html_bochum_1 %>%
    html_elements(".KeyFacts-efbce div:nth-child(1)") %>%
    html_text() %>%
    str_trim()
}
cold_rent.b1 <-get_cold_rent_bochum_1(html_bochum_1)

get_cold_rent_bochum_2 <- function(html_bochum_2){
  html_bochum_2 %>%
    html_elements(".KeyFacts-efbce div:nth-child(1)") %>%
```

```
    html_text() %>%
    str_trim()
}
cold_rent.b2 <-get_cold_rent_bochum_2(html_bochum_2)
```

e) Square meters for Bochum:

```
get_square_meters_bochum_1 <- function(html_bochum_1){
  html_bochum_1 %>%
    html_elements(".KeyFacts-efbce div:nth-child(2)") %>%
    html_text() %>%
    str_trim()
}
sq_mt.b1 <-get_square_meters_bochum_1(html_bochum_1)

get_square_meters_bochum_2 <- function(html_bochum_2){
  html_bochum_2 %>%
    html_elements(".KeyFacts-efbce div:nth-child(2)") %>%
    html_text() %>%
    str_trim()
}
sq_mt.b2 <-get_square_meters_bochum_2(html_bochum_2)
```

f) Rooms for Bochum:

```
get_rooms_bochum_1 <- function(html_bochum_1){
  html_bochum_1 %>%
    html_elements(".KeyFacts-efbce div:nth-child(3)") %>%
    html_text() %>%
    str_trim()
}
rooms.b1 <-get_rooms_bochum_1(html_bochum_1)

get_rooms_bochum_2 <- function(html_bochum_2){
  html_bochum_2 %>%
    html_elements(".KeyFacts-efbce div:nth-child(3)") %>%
    html_text() %>%
    str_trim()
}
rooms.b2 <-get_rooms_bochum_2(html_bochum_2)
```

Create a tibble to store the scraped data in and export to a csv.file:

```
# Tibble Essen Page 1
extract_page1_data_essen <- function(url){
  html <- read_html(url)
  tibble(city = get_city_essen_1(html),
         cold_rent = get_cold_rent_essen_1(html),
         rooms = get_rooms_essen_1(html),
```

```
        square_meter = get_square_meters_essen_1(html),
        title = get_title_essen_1(html),
        url=get_url_essen_1(html))
}
t1<-extract_page1_data_essen("https://www.immowelt.de/liste/essen/wohnungen/mieten?sort=rele
t1
```

```
## # A tibble: 20 x 6
##    city                  cold_rent rooms   square_meter title            url
##    <chr>                 <chr>     <chr>   <chr>        <chr>            <chr>
##  1 Essen (Horst)         313,49 €  2 Zi.   57 m²        Bezugsfertige 2-~ http~
##  2 Essen (Kupferdreh)    491,30 €  3.5 Zi. 49.13 m²     Sofort einziehen~ http~
##  3 Essen (Steele)        455,69 €  2 Zi.   61.58 m²     Frisch modernisi~ http~
##  4 Essen (Kray)          441,69 €  2 Zi.   47.29 m²     Gemütliche 2-Zim~ http~
##  5 Essen (Katernberg)    379 €     2 Zi.   45.48 m²     STOP! Erste eige~ http~
##  6 Essen (Stoppenberg)   310 €     2 Zi.   61 m²        Helle 2-Zimmer-W~ http~
##  7 Essen (Nordviertel)   588 €     3 Zi.   84 m²        Mein Zuhause im ~ http~
##  8 Essen (Leithe)        590 €     3 Zi.   67.45 m²     Nachmieter gesuc~ http~
##  9 Essen (Katernberg)    375 €     2 Zi.   53.65 m²     Teilrenovierte 2~ http~
## 10 Essen (Huttrop)       565 €     2 Zi.   56.53 m²     ++ Erstbezug nac~ http~
## 11 Essen (Borbeck-Mitte) 359,32 €  2 Zi.   65.57 m²     Helle 2-Zimmer-W~ http~
## 12 Essen (Kray)          326 €     2.5 Zi. 50.1 m²      Hier schlägt bes~ http~
## 13 Essen (Frohnhausen)   590 €     2.5 Zi. 69 m²        Erstbezug nach S~ http~
## 14 Essen (Altenessen-Süd) 650 €    3 Zi.   104 m²       Sehr großzügige,~ http~
## 15 Essen (Bedingrade)    450 €     2 Zi.   58 m²        Schöne Wohnung m~ http~
## 16 Essen (Katernberg)    800 €     3 Zi.   85 m²        3 1/2  Zimmer Wo~ http~
## 17 Essen (Werden)        790 €     3.5 Zi. 79 m²        Renovierte Wohnu~ http~
## 18 Essen (Bredeney)      1.250 €   3.5 Zi. 120 m²       Großzügig geschn~ http~
## 19 Essen (Altendorf)     430 €     2 Zi.   60 m²        Helle 2 R.-Whg. ~ http~
## 20 Essen (Frohnhausen)   480 €     2.5 Zi. 74 m²        Helle Dachwohnung http~
```

```
#Tibble essen Page 2
extract_page2_data_essen <- function(url){
  html <- read_html(url)
  tibble(city = get_city_essen_2(html),
         cold_rent = get_cold_rent_essen_2(html),
         rooms = get_rooms_essen_2(html),
         square_meter = get_square_meters_essen_2(html),
         title = get_title_essen_2(html),
         url=get_url_essen_2(html))
}
t2<-extract_page2_data_essen("https://www.immowelt.de/liste/essen/wohnungen/mieten?d=true&so
t2
```

```
## # A tibble: 20 x 6
##    city                  cold_rent rooms   square_meter title            url
##    <chr>                 <chr>     <chr>   <chr>        <chr>            <chr>
##  1 Essen (Heidhausen)    653,60 €  2 Zi.   76 m²        "Schöne, renovi~ http~
##  2 Essen (Heidhausen)    576,20 €  2 Zi.   67 m²        "Schöne, renovi~ http~
##  3 Essen (Bergerhausen)  320 €     2 Zi.   46 m²        "2-Zi.-Mietwohn~ http~
```

```
##  4 Essen (Frintrop)        430 €     2 Zi.   50 m²       "2-Zimmer Erdge~ http~
##  5 Essen (Karnap)          533,16 €  2.5 Zi. 67.66 m²    "Wir haben für ~ http~
##  6 Essen (Frillendorf)     548 €     3.5 Zi. 73.94 m²    "Wohnen, wo das~ http~
##  7 Essen (Huttrop)         551 €     2 Zi.   58.04 m²    "Erstbezug nach~ http~
##  8 Essen (Huttrop)         645,74 €  2 Zi.   72.15 m²    "Erstbezug nach~ http~
##  9 Essen (Bredeney)        1.110 €   3 Zi.   100 m²       "Individuelle W~ http~
## 10 Essen (Kettwig)         580 €     3 Zi.   80.23 m²    "Renovierte Eig~ http~
## 11 Essen (Holsterhausen)   300 €     1 Zi.   33 m²       "IHRE GELEGENHE~ http~
## 12 Essen (Altenessen-Süd)  840 €     3 Zi.   110 m²       "Altenessen-Süd~ http~
## 13 Essen (Holsterhausen)   465 €     3 Zi.   65.48 m²    "3-Zimmerwohnun~ http~
## 14 Essen (Stadtkern)       475 €     2 Zi.   65.44 m²    "Zentral gelege~ http~
## 15 Essen (Überruhr-Hinsel) 455 €     1.5 Zi. 35.25 m²    "Betreutes Wohn~ http~
## 16 Essen (Überruhr-Hinsel) 465 €     2 Zi.   36.06 m²    "Betreutes Wohn~ http~
## 17 Essen (Überruhr-Hinsel) 417 €     1 Zi.   32.3 m²     "Betreutes Wohn~ http~
## 18 Essen (Kray)            425 €     2 Zi.   58.96 m²    "Ruhige Lage I ~ http~
## 19 Essen (Südostviertel)   440 €     1 Zi.   39 m²       "Südostviertel:~ http~
## 20 Essen (Bochold)         360 €     2.5 Zi. 55 m²       "Gemütliche 2,5~ http~
```

```
# Tibble Bochum Page 1
extract_page1_data_bochum <- function(url){
  html <- read_html(url)
  tibble(city = get_city_bochum_1(html),
         cold_rent = get_cold_rent_bochum_1(html),
         rooms = get_rooms_bochum_1(html),
         square_meter = get_square_meters_bochum_1(html),
         title = get_title_bochum_1(html),
         url=get_url_bochum_1(html))

}

t3<-extract_page1_data_bochum("https://www.immowelt.de/liste/bochum/wohnungen/mieten?sort=re

# tibble Bochum page 2
extract_page2_data_bochum <- function(url){
  html <- read_html(url)
  tibble(city = get_city_bochum_2(html),
         cold_rent = get_cold_rent_bochum_2(html),
         rooms = get_rooms_bochum_2(html),
         square_meter = get_square_meters_bochum_2(html),
         title = get_title_bochum_2(html),
         url=get_url_bochum_2(html))

}

t4<-extract_page2_data_bochum("https://www.immowelt.de/liste/bochum/wohnungen/mieten?d=true&
t4
```

```
## # A tibble: 20 x 6
##    city                      cold_rent rooms   square_meter title         url
##    <chr>                     <chr>     <chr>   <chr>        <chr>         <chr>
##  1 Bochum (Hamme)            365,26 €  2 Zi.   50.59 m²     2-Zimmer-Woh~ http~
```

```
##  2 Bochum (Hamme)             454,91 €  2 Zi.   55.68 m²     Moderne 2-Zi~ http~
##  3 Bochum (Hamme)             454,82 €  2 Zi.   55.67 m²     Moderne 2-Zi~ http~
##  4 Bochum (Hamme)             363,32 €  3 Zi.   53.43 m²     3-Zimmer-Woh~ http~
##  5 Bochum (Weitmar)           384,90 €  1 Zi.   39.68 m²     Nette Nachba~ http~
##  6 Bochum (Weitmar)           595 €     3 Zi.   75 m²        zentral - re~ http~
##  7 Bochum (Linden)            400 €     3 Zi.   70 m²        3 Zi Wohnung~ http~
##  8 Bochum (Innenstadt)        999 €     2 Zi.   79 m²        Schöne Altba~ http~
##  9 Bochum / Stiepel (Stiepel) 995 €     4 Zi.   106 m²       Erstbezug na~ http~
## 10 Bochum (Höntrop)           790 €     2.5 Zi. 73.77 m²     Modern gesch~ http~
## 11 Bochum (Innenstadt)        515 €     2 Zi.   73.9 m²      Charmante Wo~ http~
## 12 Bochum (Höntrop)           695 €     2.5 Zi. 67.1 m²      Barrierearme~ http~
## 13 Bochum (Höntrop)           790 €     2.5 Zi. 70.88 m²     Modern gesch~ http~
## 14 Bochum (Weitmar)           450 €     2 Zi.   64 m²        Renovierte 2~ http~
## 15 Bochum (Wattenscheid)      765 €     4 Zi.   109 m²       Helle Maison~ http~
## 16 Bochum (Hofstede)          570 €     3.5 Zi. 77.5 m²      Renovierte W~ http~
## 17 Bochum (Weitmar)           595 €     2 Zi.   76.47 m²     BO - WEITMAR~ http~
## 18 Bochum (Altenbochum)       490 €     3.5 Zi. 64 m²        Altenbochum:~ http~
## 19 Bochum (Wiemelhausen)      645 €     3.5 Zi. 86 m²        Helle und mo~ http~
## 20 Bochum (Westenfeld)        600,60 €  3 Zi.   66 m²        Nette Nachba~ http~
```

Combine the data in one tibble and export the data into a csv. file:

```
Full_data<-rbind(t1,t2,t3,t4)
Full_data
```

```
## # A tibble: 80 x 6
##    city                cold_rent rooms   square_meter title                url
##    <chr>               <chr>     <chr>   <chr>        <chr>                <chr>
##  1 Essen (Horst)       313,49 €  2 Zi.   57 m²        Bezugsfertige 2-Rau~ http~
##  2 Essen (Kupferdreh)  491,30 €  3.5 Zi. 49.13 m²     Sofort einziehen: S~ http~
##  3 Essen (Steele)      455,69 €  2 Zi.   61.58 m²     Frisch modernisiert~ http~
##  4 Essen (Kray)        441,69 €  2 Zi.   47.29 m²     Gemütliche 2-Zimmer~ http~
##  5 Essen (Katernberg)  379 €     2 Zi.   45.48 m²     STOP! Erste eigene ~ http~
##  6 Essen (Stoppenberg) 310 €     2 Zi.   61 m²        Helle 2-Zimmer-Wohn~ http~
##  7 Essen (Nordviertel) 588 €     3 Zi.   84 m²        Mein Zuhause im Elt~ http~
##  8 Essen (Leithe)      590 €     3 Zi.   67.45 m²     Nachmieter gesucht:~ http~
##  9 Essen (Katernberg)  375 €     2 Zi.   53.65 m²     Teilrenovierte 2-Zi~ http~
## 10 Essen (Huttrop)     565 €     2 Zi.   56.53 m²     ++ Erstbezug nach S~ http~
## # ... with 70 more rows
```

```
write.csv(Full_data, "Webscraping_Data.csv")
```

Excercise Number 2)

a)

Load the immowelt data into R

```
load(here::here("rent_advertisements.RData"))
```

Omit column "heating_cost_excluded" as it contains no data (NAs) :

```
immowelt %<>%
  select(-heating_cost_excluded)

head(immowelt)
```

```
## # A tibble: 6 x 13
##   title       zipcode city  cold_rent heating_cost_in~ service_charges warm_rent
##   <chr>         <dbl> <chr>     <dbl>            <dbl>           <dbl>     <dbl>
## 1 2-Zi.-Miet~   45136 essen      320               NA              90      410
## 2 *** Das is~   45355 essen      480               90             236      716
## 3 *** Modern~   45355 essen      590               77             224      814
## 4 Wir renovi~   45307 essen      496.              45             117      658.
## 5 Schöne dre~   45134 essen      690               NA             160      850
## 6 Schöne Aus~   45279 essen      450               75             150      675
## # ... with 6 more variables: deposit <dbl>, square_meter <dbl>, rooms <dbl>,
## #   building_year <chr>, efficiency_class <chr>, energy_demand <dbl>
```

```
names(immowelt)
```

```
##  [1] "title"                "zipcode"              "city"
##  [4] "cold_rent"            "heating_cost_included" "service_charges"
##  [7] "warm_rent"            "deposit"              "square_meter"
## [10] "rooms"                "building_year"        "efficiency_class"
## [13] "energy_demand"
```

Overview of the data:

```
immowelt %>%
  knitr::kable(booktabs = TRUE, linesep = "",
               escape = TRUE, caption = 'Immowelt'
  ) |>
  kableExtra::kable_styling(font_size = 10,
                            latex_options = c("striped", "hold_position")) %>%
  kableExtra::row_spec(0, bold = TRUE)
```

Clean all non-available data (NA).

```
immowelt_clean <- na.omit(immowelt)
immowelt_clean
```

```
## # A tibble: 134 x 13
##    title      zipcode city  cold_rent heating_cost_in~ service_charges warm_rent
##    <chr>        <dbl> <chr>     <dbl>            <dbl>           <dbl>     <dbl>
##  1 Wir renov~   45307 essen      496.              45             117       658.
```

```
##  2 Da kommt ~   45144 essen     1600             106             273       1979
##  3 Wer will ~   45128 essen      540.             36              96        672.
##  4 Preisgüns~   45144 essen      453.             56             106        615.
##  5 3 Kaltmie~   45141 essen      422.             20.4            93.8      536.
##  6 2 Raum mi~   45326 essen      397              50              68        515
##  7 Individue~   45279 essen      294.             30             129        453.
##  8 Ein ruhig~   45279 essen      346.             89             218        652.
##  9 Wohnung m~   45279 essen      632.             38             148        818.
## 10 Gemütlich~   45329 essen      455              64              66        585
## # ... with 124 more rows, and 6 more variables: deposit <dbl>,
## #   square_meter <dbl>, rooms <dbl>, building_year <chr>,
## #   efficiency_class <chr>, energy_demand <dbl>
```

Ensure that all variables of the dataset are stored using a proper class.

```
char_v <- c("title","building_year")

immowelt_clean %<>%
  mutate(across(where(is.character) &! any_of(char_v),
                as.factor))

immowelt_clean %<>%
  mutate(building_year = as.numeric(building_year))
```

b) Calculate the cold_rent and warm_rent per square meter:

```
immowelt_clean %<>%
  mutate(cold_rent_qm = cold_rent / square_meter) %>%
  mutate(warm_rent_qm = warm_rent / square_meter)

immowelt_clean$cold_rent_qm
```

```
##   [1]  8.000000 10.113141 10.500097  8.999205 11.259931  8.446809  5.779965
##   [8]  4.731580  8.100038  9.214257  8.416654  8.177905  7.456140  8.200000
##  [15]  5.192753  7.501028 11.259931  7.356760 11.095101  5.723975  7.893971
##  [22]  8.376264  6.880263 11.259931  9.306607  8.000000  9.000000  9.347826
##  [29]  8.448980  8.376264  7.280025  9.170000  7.530000  8.430985  4.731580
##  [36]  9.273342  9.500000  8.855520  9.069767  7.011773  7.250000  8.149248
##  [43]  9.590058  8.500000  7.560606  9.200000  5.254690  9.000000  9.561567
##  [50]  8.550009  8.397063  8.746762  5.987647  9.500000  8.949965  8.439024
##  [57]  7.739986  5.364741  7.389933  6.864407  8.400000 10.357675  8.505905
##  [64] 10.199948  9.166667  8.800000  7.220004  6.641509  5.592552  4.925403
##  [71]  6.799925  8.799898  7.031250  9.700010  9.530498  6.759965  9.500000
##  [78]  4.940000  8.500090  8.700000 10.708960  9.000000  8.799965  8.270047
##  [85] 12.645570  7.879980  7.139987  8.750055  7.500101  7.500104  9.700101
##  [92]  6.230038  8.300038 10.200026 10.500061  7.600040  8.170079  8.800000
##  [99]  9.500000  8.701473  6.395349  8.535407  6.589935  7.549928  7.000000
## [106]  8.599921  7.239941  8.859973  9.339958  7.500000  7.945664  8.300007
## [113]  8.350000  8.500000  8.799928  8.562992  8.250060  8.000000 11.145598
## [120]  9.200000 10.200000  9.450061 10.199936  8.620690  8.169930  9.700119
```

```
## [127]  8.880007  7.800115  8.799968 10.200000 12.804878 10.510036  8.900000
## [134]  7.599970
```

```
immowelt_clean$warm_rent_qm
```

```
##   [1] 10.610377 12.508691 13.064698 12.219881 14.302586 10.957447  8.909270
##   [8]  8.935908 10.482516 11.846902 12.013132 11.559746  9.649123 10.272727
##  [15]  8.227025 11.097411 14.302586  9.745543 14.265130  8.904099 10.496912
##  [22] 11.925205 10.474430 14.302586 12.170178 10.870662 12.151467 11.376812
##  [29] 10.959184 11.963366  9.931073 11.659796  9.880877 12.448638  9.319365
##  [36] 12.146177 12.658363 10.780633 11.627907 10.734072 10.330645 11.787637
##  [43] 12.175881 10.857143 11.060606 12.461538  8.313672 13.166667 13.269590
##  [50] 11.189621 11.884368 12.094043 10.084457 11.807917 10.834927 11.000000
##  [57] 10.092088  8.554533 11.248993 10.254237 12.974850 13.412817 11.791888
##  [64] 13.493552 11.770833 12.300000 11.015220  8.716981  9.254306  8.048845
##  [71] 10.430844 12.512077 11.312500 13.345791 12.038523 10.192205 11.548656
##  [78]  8.749524 11.625651 12.160076 13.704758 12.800000 12.814501 11.932019
##  [85] 15.367089 11.795540 10.239133 12.257666 10.624467 11.460294 13.581149
##  [92] 10.170584 11.961322 13.487628 12.973802 11.605705 11.348958 12.030047
##  [99] 13.066879 12.427488  8.488372 12.380752  8.749045 10.621161 10.124366
## [106] 12.161456 10.111192 11.989194 11.876396 10.887487 11.295125 11.982249
## [113] 11.788320 11.776190 11.960848 11.240157 12.034491 11.087313 14.150677
## [120] 11.760606 13.493681 12.499591 13.478795 11.724138 11.151787 12.499404
## [127] 11.861865 11.256336 11.147612 13.493681 17.439024 13.199719 12.831818
## [134] 10.570267
```

Create one table with the five districts with the most ads for each city:

```
ads_essen <- immowelt_clean %>%
  filter(city =="essen") %>%
  group_by(zipcode) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  slice(1:5)

ads_bochum <- immowelt_clean%>%
  filter(city=="bochum")%>%
  group_by(zipcode)%>%
  summarise(count=n())%>%
  arrange(desc(count))%>%
  slice(1:5)

full_join(ads_bochum,ads_essen)
```

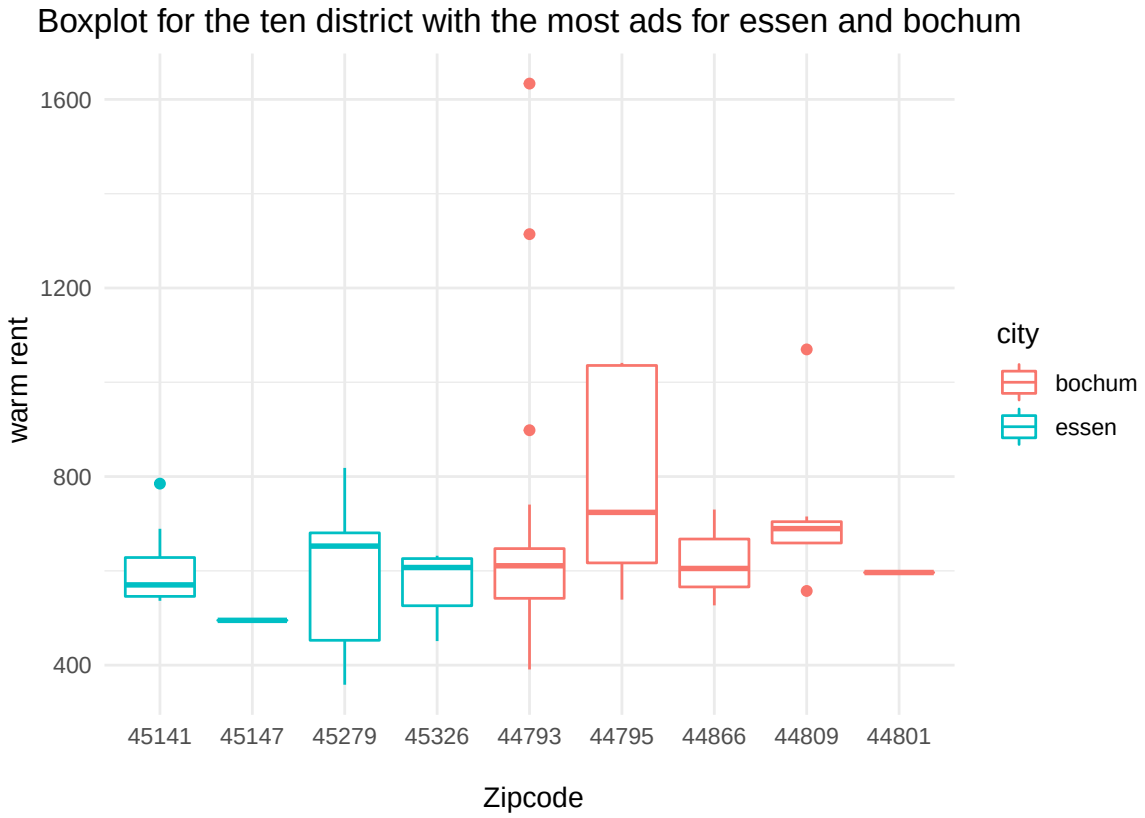
```
## Joining, by = c("zipcode", "count")
```

```
## # A tibble: 10 x 2
##    zipcode count
##      <dbl> <int>
```

```
##  1   44793    33
##  2   44795    11
##  3   44809     7
##  4   44867     6
##  5   44799     3
##  6   45141    14
##  7   45326     7
##  8   45327     7
##  9   45279     5
## 10   45329     5
```

d) Create a boxplot for the ten districts you found in (c), mapping the warm_rent.

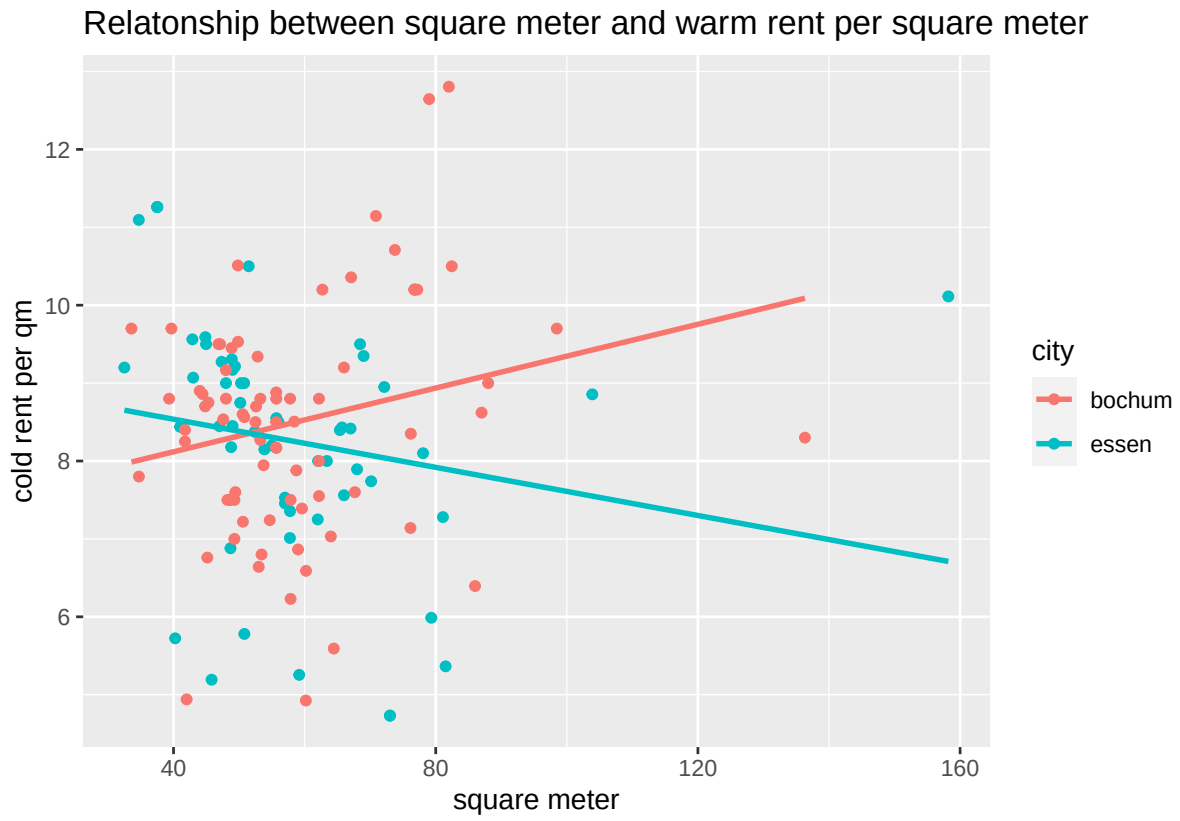
```
immowelt_clean %>%
 mutate(zipcode = factor(zipcode, c("45141","45147","45279","45326","45355",
                                    "44793","44795","44866","44809","44801"))) %>%
 drop_na() %>%
 ggplot(aes(x=zipcode, y=warm_rent, color=city)) +
 geom_boxplot() +
 labs(title = 'Boxplot for the ten district with the most ads for essen and bochum',
      x ="\n Zipcode", y ="\n warm rent")+
 theme_minimal() +
 theme(plot.title = element_text(hjust = 0.5))
```



e) Create a scatterplot showing the relationship between square_meter and cold_rent and warm_- rent per square meter separately. Color and shape the points differently for the two cities. Also, draw a regression line to point out the relationship in the plots.

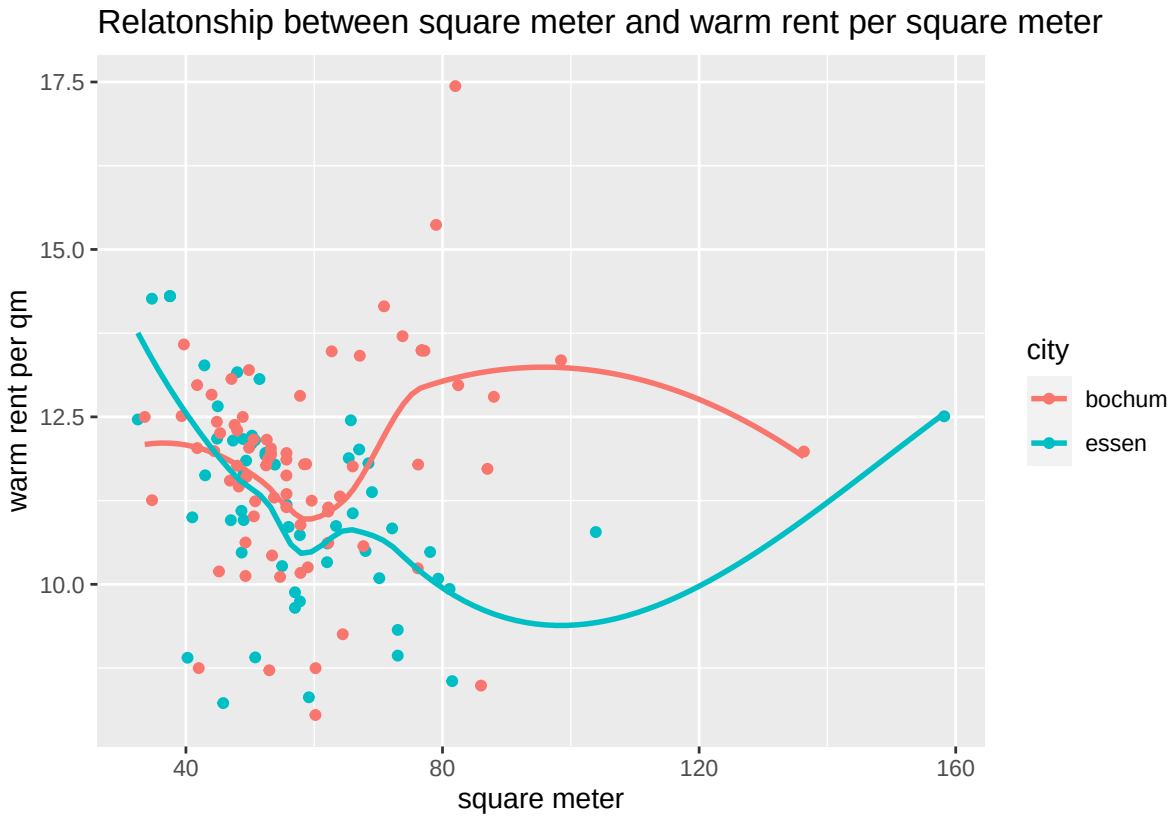
```
immowelt_clean %>%
 ggplot(aes(x = square_meter, y = cold_rent_qm, color = city)) +
 geom_point() +
 geom_smooth(se=FALSE, method = "lm") +
 labs(title = "Relatonship between square meter and warm rent per square meter",
      x ="square meter", y ="cold rent per qm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



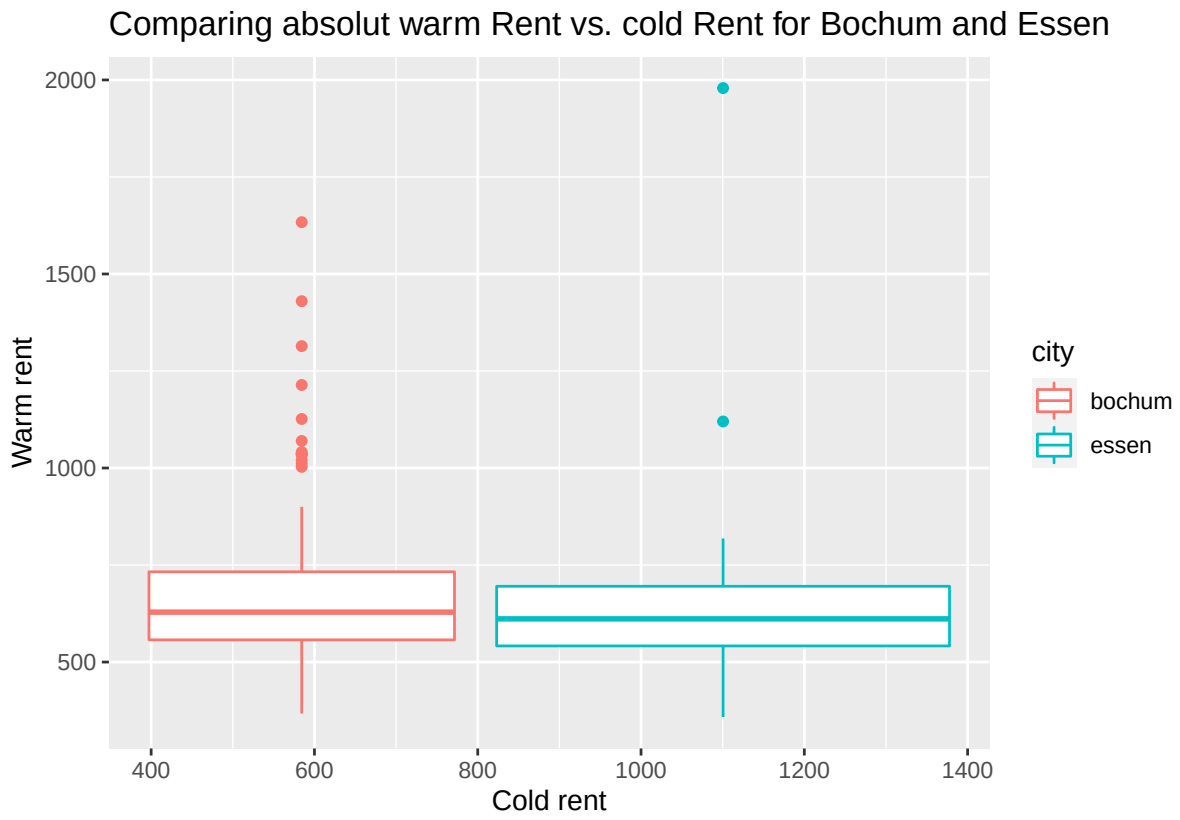

```
immowelt_clean %>%
  ggplot(aes(x = square_meter, y = warm_rent_qm, color = city)) +
  geom_point() +
  geom_smooth(se=FALSE) +
  labs(title = "Relatonship between square meter and warm rent per square meter",
       x="square meter", y="warm rent per qm")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

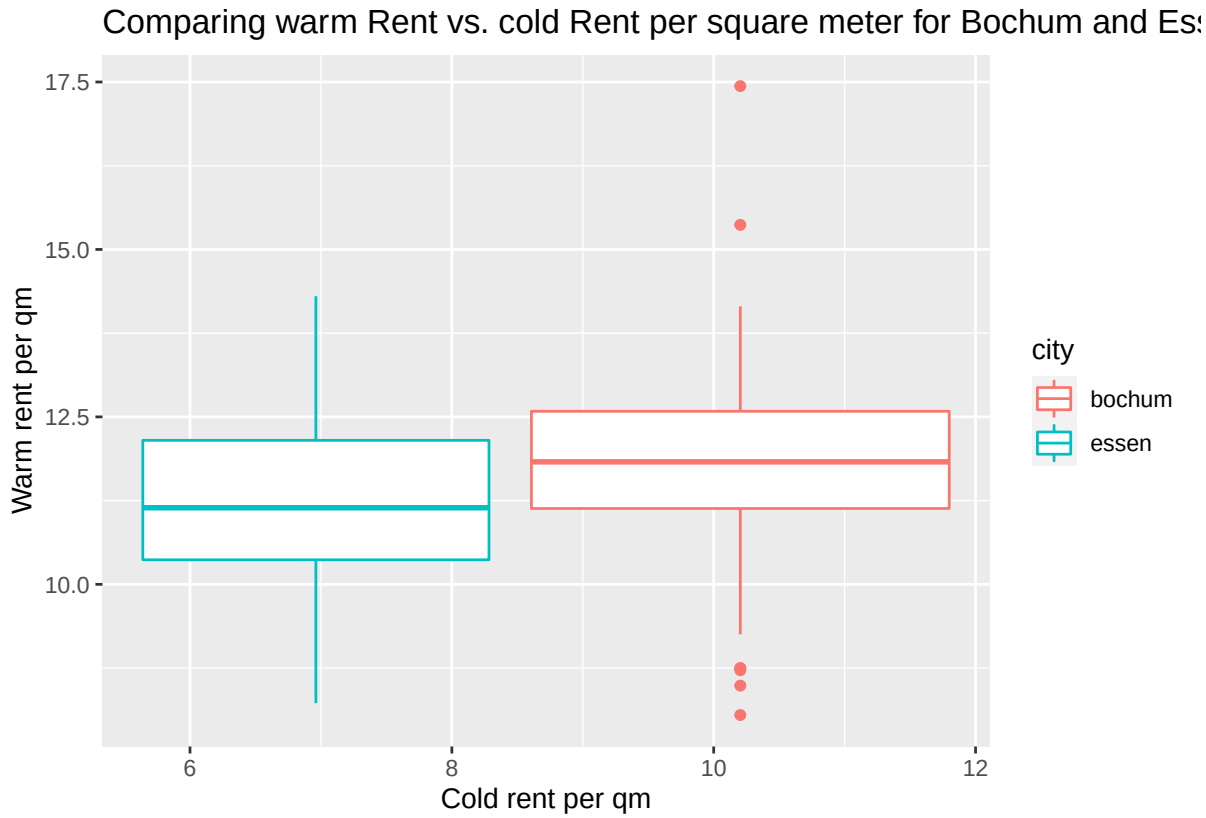


f) To compare the housing market of the two cities graphically, create a boxplot on which both the cold_rent and the warm_rent are plotted. Do this for the absolute rents as well as for the rents per square meter.

```
immowelt_clean%>%
  ggplot(aes(x=cold_rent,y=warm_rent, color=city))+
  geom_boxplot()+
  labs(title = "Comparing absolut warm Rent vs. cold Rent for Bochum and Essen",
       x="Cold rent",y="Warm rent" )
```

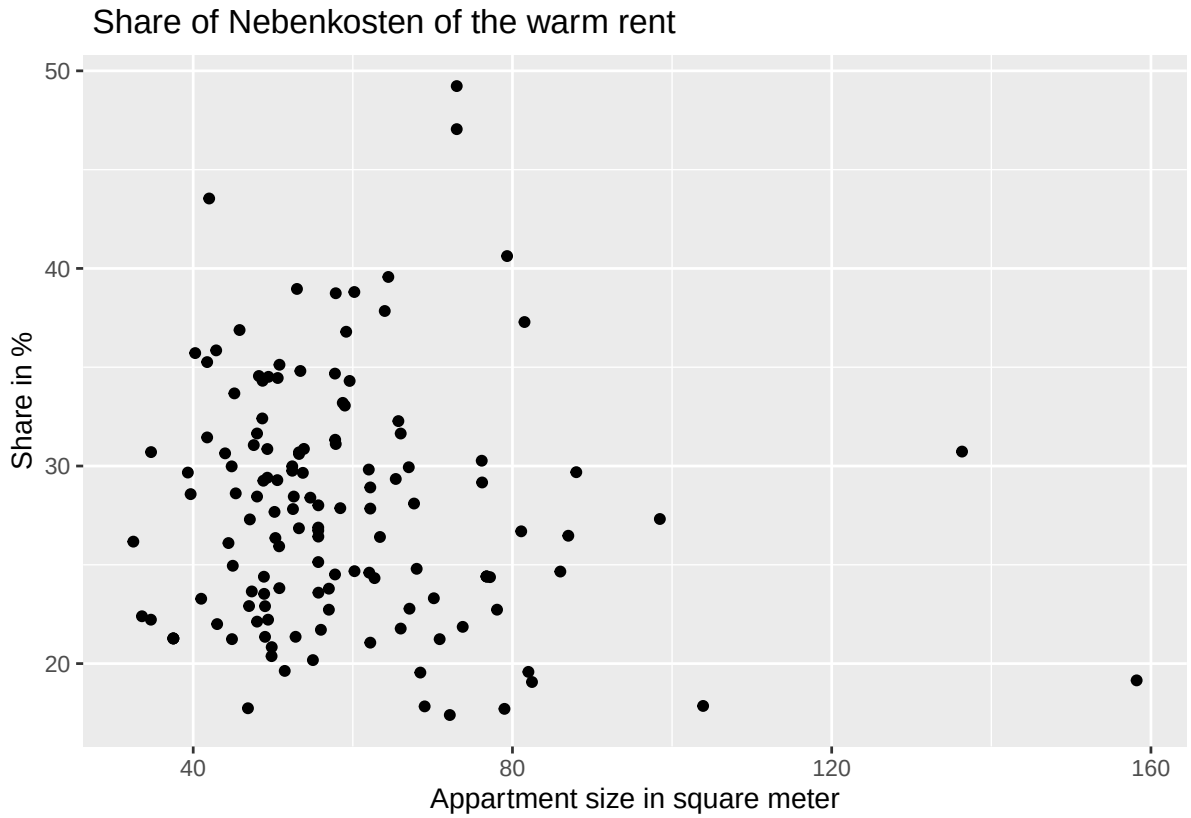



```
immowelt_clean%>%
  ggplot(aes(x=cold_rent_qm, y=warm_rent_qm, color=city))+
  geom_boxplot()+
  labs(title = "Comparing warm Rent vs. cold Rent per square meter for Bochum and Essen",
       x ="Cold rent per qm", y = "Warm rent per qm")
```



g) Calculate the share of Nebenkosten (Nebenkosten = service_charges + heating_cost_-included) of the warm_rent and plot the percentage against the apartment size.

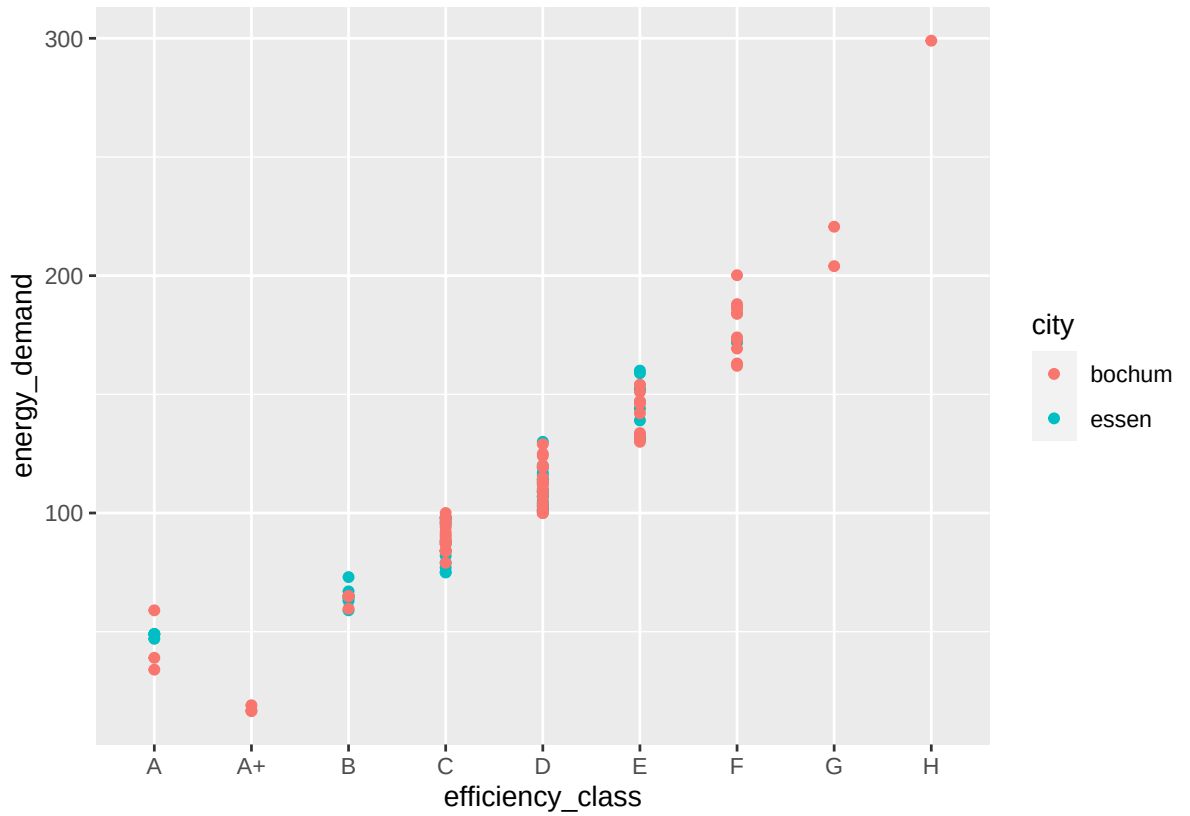
```
immowelt_clean %>%
  select(heating_cost_included, square_meter, warm_rent) %>%
  mutate(share_nebenkosten = (heating_cost_included+immowelt_clean$service_charges)
         /warm_rent*100) %>%
  ggplot(aes(x=square_meter, y=share_nebenkosten))+
  geom_point()+
  labs(title = " Share of Nebenkosten of the warm rent",
       x="Appartment size in square meter", y="Share in %")
```



h) Next, you want to get an overview of the different efficiency classes of efficiency_class. Create appropriate plots and compute statistics (if necessary) to answer the following questions:

i) Do the efficiency classes differ in terms of their energy demand? Discuss your findings shortly:

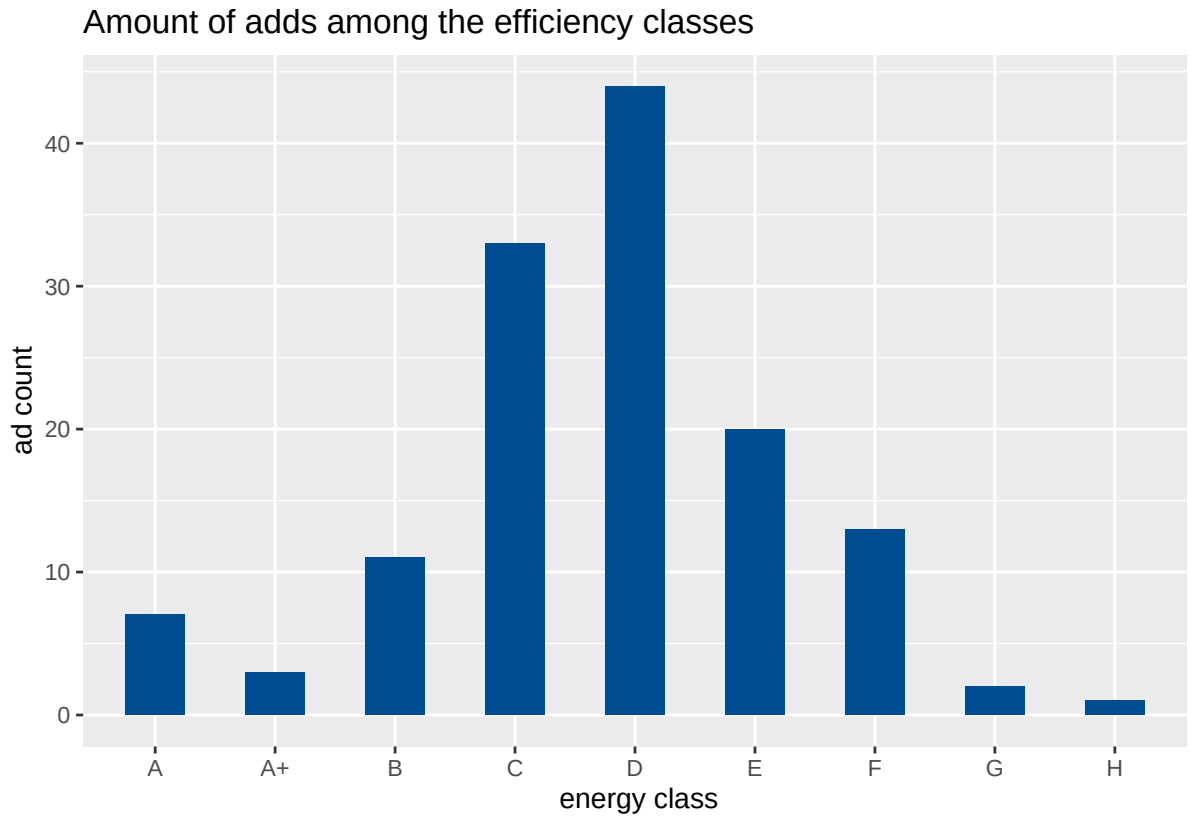
```
immowelt_clean %>%
  select(efficiency_class,energy_demand, city)%>%
  mutate(efficiency_class = factor(efficiency_class,c(
    "A","A+","B","C","D","E",
    "F","G","H"))) %>%
  drop_na() %>%
  ggplot(aes(efficiency_class, energy_demand, color = city))+
  geom_point()
```



The efficiency classes differ in terms of their energy demand. Low energy classes seem to correlate with high energy demand. High energy classes seem to correlate with low energy demands.

ii) How is the amount of ads distributed among the efficiency classes?

```
immowelt_clean %>%
  select(efficiency_class, city)%>%
  drop_na()%>%
  mutate(efficiency_class=factor(efficiency_class,c(
    "A","A+","B","C","D","E",
    "F","G","H"))) %>%
  ggplot(aes(x=efficiency_class))+
  geom_bar(stat = 'count', width = 0.5, fill = '#004c93')+
  labs(title = "Amount of adds among the efficiency classes",
       x=" energy class", "amount", y = "ad count")
```



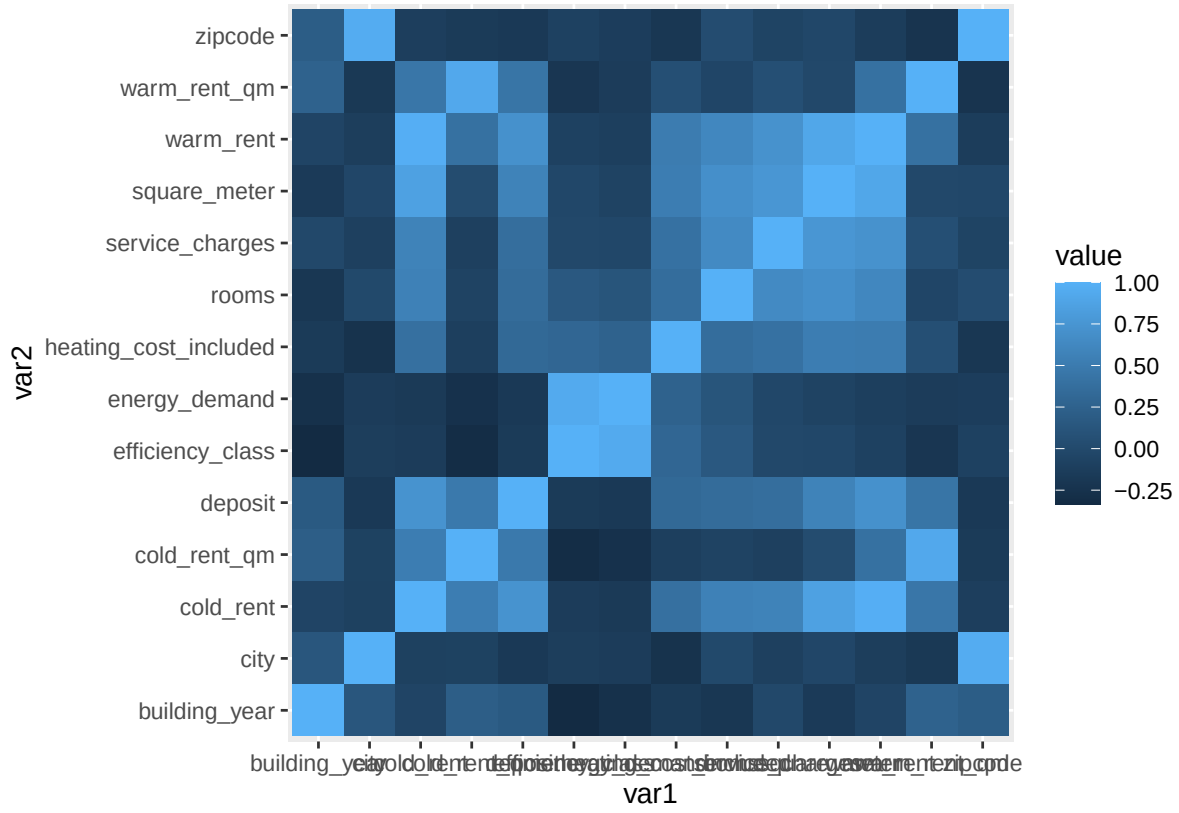
Most ads are distributed at energy class “D”.

iii) & vi)

```
cor<-immowelt_clean %>%
  mutate(across(where(is.character), as.numeric))%>%
  mutate(across(where(is.factor), as.numeric))%>%
  select_if(is.numeric) %>%
  select(-title) %>%
  cor()
```

```
## Warning in mask$eval_all_mutate(quo): NAs durch Umwandlung erzeugt
```

```
cor%>%
  as_tibble() %>%
  mutate(var1 = colnames(cor))%>%
  pivot_longer(-var1, names_to = "var2", values_to = "value")%>%
  ggplot(aes(var1, var2, fill=value))+
  geom_tile()
```




```
cor
```

```
##                            zipcode         city   cold_rent
## zipcode                 1.00000000  0.957527366 -0.12099509
## city                    0.95752737  1.000000000 -0.09318625
## cold_rent              -0.12099509 -0.093186253  1.00000000
## heating_cost_included  -0.19823579 -0.246442769  0.39461326
## service_charges        -0.06061003 -0.098511247  0.57463885
## warm_rent              -0.13247898 -0.123264705  0.97738289
## deposit                -0.17211919 -0.177268145  0.72903506
## square_meter           -0.02593832 -0.032882213  0.85739296
## rooms                   0.03264287 -0.008410934  0.55626793
## building_year           0.20139668  0.132082931 -0.05383605
## efficiency_class       -0.08735620 -0.122510813 -0.14201658
## energy_demand          -0.12728184 -0.141042999 -0.16512656
## cold_rent_qm           -0.15669800 -0.082687229  0.51899217
## warm_rent_qm           -0.22909810 -0.178868008  0.44956791
##                        heating_cost_included service_charges   warm_rent
## zipcode                           -0.19823579     -0.06061003 -0.13247898
## city                              -0.24644277     -0.09851125 -0.12326471
## cold_rent                          0.39461326      0.57463885  0.97738289
## heating_cost_included              1.00000000      0.40449750  0.50557518
## service_charges                    0.40449750      1.00000000  0.71386758
## warm_rent                          0.50557518      0.71386758  1.00000000
## deposit                            0.31864149      0.36840542  0.70371126
## square_meter                       0.52130250      0.75535488  0.90608598
## rooms                              0.36313622      0.64436542  0.61696009
```

```
## building_year                  -0.15762575     -0.01642518 -0.05488153
## efficiency_class                0.29700277     -0.01805435 -0.08874809
## energy_demand                   0.25362829     -0.02337416 -0.11243906
## cold_rent_qm                   -0.10788920     -0.10085479  0.39912837
## warm_rent_qm                    0.06075776      0.05924705  0.39685471
##                          deposit square_meter        rooms building_year
## zipcode                -0.1721192  -0.02593832  0.032642872    0.20139668
## city                   -0.1772681  -0.03288221 -0.008410934    0.13208293
## cold_rent               0.7290351   0.85739296  0.556267925   -0.05383605
## heating_cost_included   0.3186415   0.52130250  0.363136224   -0.15762575
## service_charges         0.3684054   0.75535488  0.644365423   -0.01642518
## warm_rent               0.7037113   0.90608598  0.616960086   -0.05488153
## deposit                 1.0000000   0.57275007  0.351892449    0.16715592
## square_meter            0.5727501   1.00000000  0.692276185   -0.16009479
## rooms                   0.3518924   0.69227619  1.000000000   -0.20172152
## building_year           0.1671559  -0.16009479 -0.201721522    1.00000000
## efficiency_class       -0.1583222  -0.02301613  0.156291944   -0.33254517
## energy_demand          -0.1735921  -0.07155952  0.118439275   -0.26938478
## cold_rent_qm            0.4815515   0.02604321 -0.071165760    0.20974012
## warm_rent_qm            0.4391582  -0.01398242 -0.041679506    0.25393548
##                        efficiency_class energy_demand cold_rent_qm warm_rent_qm
## zipcode                     -0.08735620   -0.12728184  -0.15669800  -0.22909810
## city                        -0.12251081   -0.14104300  -0.08268723  -0.17886801
## cold_rent                   -0.14201658   -0.16512656   0.51899217   0.44956791
## heating_cost_included        0.29700277    0.25362829  -0.10788920   0.06075776
## service_charges             -0.01805435   -0.02337416  -0.10085479   0.05924705
## warm_rent                   -0.08874809   -0.11243906   0.39912837   0.39685471
## deposit                     -0.15832216   -0.17359207   0.48155151   0.43915819
## square_meter                -0.02301613   -0.07155952   0.02604321  -0.01398242
## rooms                        0.15629194    0.11843928  -0.07116576  -0.04167951
## building_year               -0.33254517   -0.26938478   0.20974012   0.25393548
## efficiency_class             1.00000000    0.94063739  -0.30896094  -0.20726742
## energy_demand                0.94063739    1.00000000  -0.26100727  -0.14316665
## cold_rent_qm                -0.30896094   -0.26100727   1.00000000   0.92294041
## warm_rent_qm                -0.20726742   -0.14316665   0.92294041   1.00000000
```

```
cor[11,10]
```

```
## [1] -0.3325452
```

```
cor[11,3]
```

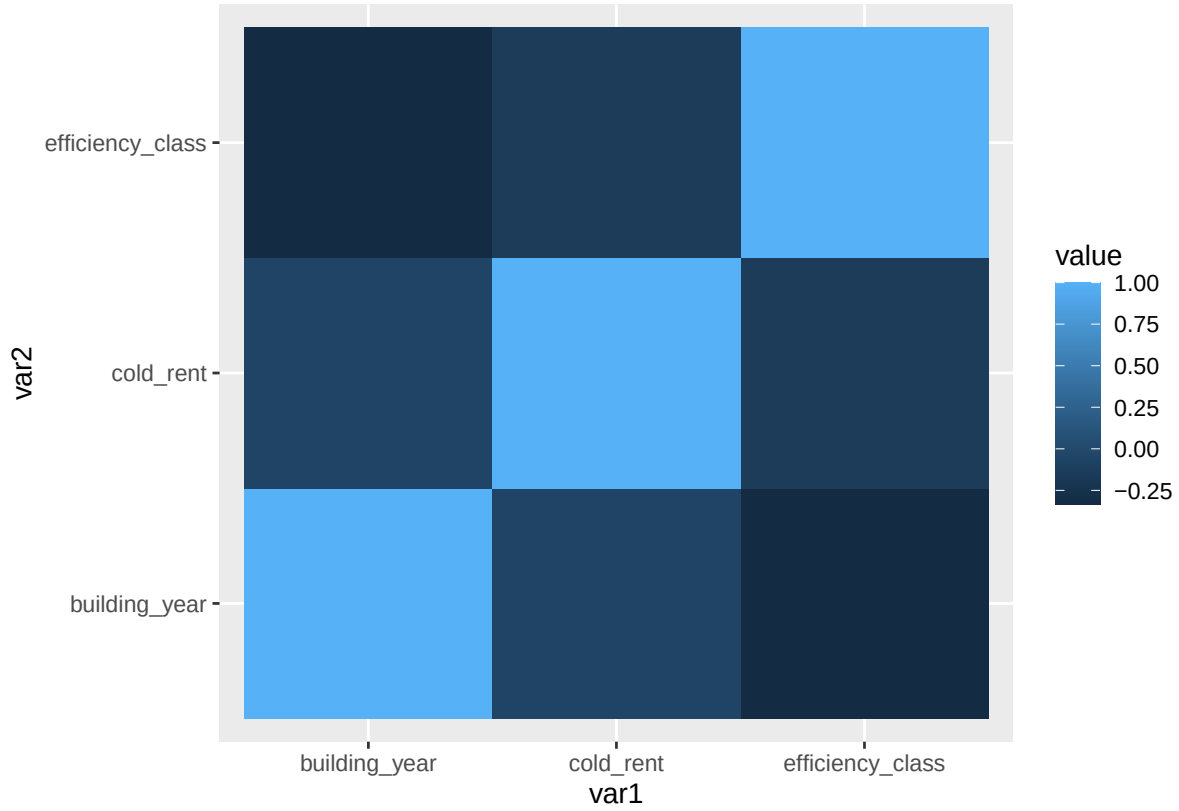
```
## [1] -0.1420166
```

The correlation between effiency_class and building_year is = -0.3325452. The correlation between effiency_class and cold_rent is = -0.14201658. In both cases, theres a slight negative correlation. The whole dataset is included in this calculation.

In the following the variables “building year”, “cold rent” and “efficiency class” will be considered for calculating the correlation:

```
cor<-immowelt_clean%>%
  select(efficiency_class, building_year, cold_rent)%>%
  mutate(across(where(is.character), as.numeric))%>%
  mutate(across(where(is.factor), as.numeric))%>%
  select_if(is.numeric) %>%
  cor()

cor%>%
  as_tibble()%>%
  mutate(var1=colnames(cor))%>%
  pivot_longer(-var1, names_to = "var2", values_to = "value")%>%
  ggplot(aes(var1, var2, fill=value))+
  geom_tile()
```




```
cor
```

```
##                  efficiency_class building_year   cold_rent
## efficiency_class        1.0000000   -0.33254517 -0.14201658
## building_year          -0.3325452    1.00000000 -0.05383605
## cold_rent              -0.1420166   -0.05383605  1.00000000
```

```
cor[1,2]
```

```
## [1] -0.3325452
```

```
cor[1,3]
```

```
## [1] -0.1420166
```

The results do not differ. a slight negative correlation between the efficiency class and the building year could be explained by improved building structure and heat efficient heating systems. It´s logical to assume that with decreasing building year,the efficiency class will rise to a better grading. The same argumentation can be applied to the correlation between the efficiency class and the cold rent. A well build and innovative house will generally cost a higher cold rent which results in better insulation and hence a lower (= better) efficiency class.

i) Perform two linear regressions to predict warm_rent. Both regressions use the variables efficiency_- class, rooms, building_year, square_meter, and service_charges. To investigate a systematical difference between the cities, use the variable city in the first regression, and zipcode in the other. Also, discuss why you cannot use city and zipcode in one regression and the advantages of each approach.

```
 reg1<-lm(warm_rent~efficiency_class+rooms+building_year+square_meter+service_charges+city,
          data = immowelt_clean)
summary(reg1)
```

```
##
## Call:
## lm(formula = warm_rent ~ efficiency_class + rooms + building_year +
##     square_meter + service_charges + city, data = immowelt_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -288.42  -47.90    6.01   39.61  436.39
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -618.6383   942.9045  -0.656   0.5130
## efficiency_classA+   35.3948    67.7365   0.523   0.6023
## efficiency_classB   -72.8506    53.3951  -1.364   0.1750
## efficiency_classC   -84.6780    45.7795  -1.850   0.0668 .
## efficiency_classD   -65.7440    43.6363  -1.507   0.1345
## efficiency_classE   -86.9238    48.0950  -1.807   0.0732 .
## efficiency_classF   -91.9057    50.3217  -1.826   0.0703 .
## efficiency_classG   -95.6573    80.1508  -1.193   0.2350
## efficiency_classH   -22.1844   105.3327  -0.211   0.8335
## rooms                 1.3814    13.1780   0.105   0.9167
## building_year         0.3479     0.4690   0.742   0.4596
## square_meter         11.6113     0.8252  14.071   <2e-16 ***
## service_charges       0.1711     0.2921   0.586   0.5592
## cityessen           -45.7939    18.0572  -2.536   0.0125 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 95.06 on 120 degrees of freedom
## Multiple R-squared:  0.8491, Adjusted R-squared:  0.8327
## F-statistic: 51.93 on 13 and 120 DF,  p-value: < 2.2e-16
```

```
reg2<-lm(warm_rent~efficiency_class+rooms+building_year+square_meter+service_charges+zipcode
         data = immowelt_clean)
summary(reg2)
```

```
##
## Call:
## lm(formula = warm_rent ~ efficiency_class + rooms + building_year +
##     square_meter + service_charges + zipcode, data = immowelt_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -285.84  -50.04    7.60   39.29  428.63
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4691.89866 1780.57309   2.635  0.00952 **
## efficiency_classA+   40.37003   66.37672   0.608  0.54421
## efficiency_classB   -58.95926   53.18408  -1.109  0.26982
## efficiency_classC   -70.34652   45.49021  -1.546  0.12464
## efficiency_classD   -52.91687   43.44679  -1.218  0.22562
## efficiency_classE   -73.48984   47.84768  -1.536  0.12719
## efficiency_classF   -79.48313   49.60163  -1.602  0.11169
## efficiency_classG   -83.62469   78.84231  -1.061  0.29098
## efficiency_classH   -15.62949  103.81992  -0.151  0.88059
## rooms                 4.07852   13.07305   0.312  0.75560
## building_year         0.55347    0.47508   1.165  0.24633
## square_meter         11.57357    0.81372  14.223  < 2e-16 ***
## service_charges       0.15977    0.28771   0.555  0.57972
## zipcode              -0.12776    0.04059  -3.148  0.00208 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93.78 on 120 degrees of freedom
## Multiple R-squared:  0.8531, Adjusted R-squared:  0.8372
## F-statistic: 53.61 on 13 and 120 DF,  p-value: < 2.2e-16
```

```
reg3<-lm(warm_rent~efficiency_class+rooms+building_year+square_meter+service_charges+zipcode
         data = immowelt_clean)
summary(reg3)  # for comparison
```

```
##
## Call:
## lm(formula = warm_rent ~ efficiency_class + rooms + building_year +
##     square_meter + service_charges + zipcode + city, data = immowelt_clean)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -267.39  -51.84    7.89   38.76  424.74
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        14404.2772  6188.4588   2.328   0.0216 *
## efficiency_classA+    62.2407    67.2555   0.925   0.3566
## efficiency_classB    -43.0922    53.6965  -0.803   0.4239
## efficiency_classC    -46.3948    47.4831  -0.977   0.3305
## efficiency_classD    -34.0573    44.6558  -0.763   0.4472
## efficiency_classE    -53.8407    49.0072  -1.099   0.2741
## efficiency_classF    -53.9380    51.6683  -1.044   0.2986
## efficiency_classG    -54.4464    80.2970  -0.678   0.4990
## efficiency_classH      2.0841   103.6656   0.020   0.9840
## rooms                  7.4645    13.1460   0.568   0.5712
## building_year          0.7922     0.4938   1.604   0.1113
## square_meter          11.4746     0.8103  14.160   <2e-16 ***
## service_charges        0.1895     0.2863   0.662   0.5094
## zipcode               -0.3556     0.1448  -2.455   0.0155 *
## cityessen            104.1071    63.5687   1.638   0.1041
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93.13 on 119 degrees of freedom
## Multiple R-squared:  0.8563, Adjusted R-squared:  0.8394
## F-statistic: 50.67 on 14 and 119 DF,  p-value: < 2.2e-16
```

If both zipcode and city are used in the regression analysis, it might result in multicolinearity, which will lead to possibly reduced accuracy of the predictions. To determine which model is more fitting, cross validation is used in the following:

```
set.seed(123)
train<-immowelt_clean%>%
  slice_sample(prop = 0.8)

test<-immowelt_clean%>%
  anti_join(train)
```

```
## Joining, by = c("title", "zipcode", "city", "cold_rent",
## "heating_cost_included", "service_charges", "warm_rent", "deposit",
## "square_meter", "rooms", "building_year", "efficiency_class", "energy_demand",
## "cold_rent_qm", "warm_rent_qm")
```

```
mod_city<-lm(warm_rent~efficiency_class+rooms+building_year+square_meter+service_charges+cit
             data = train)
summary(mod_city)
```

```
##
## Call:
## lm(formula = warm_rent ~ efficiency_class + rooms + building_year +
```

```
##     square_meter + service_charges + city, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -270.29  -47.48    5.92   37.20  447.66
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         815.7063  1227.3639   0.665   0.5080
## efficiency_classA+    8.1139    87.1850   0.093   0.9261
## efficiency_classB  -110.7968    73.6366  -1.505   0.1358
## efficiency_classC  -147.0374    65.8568  -2.233   0.0280 *
## efficiency_classD  -115.0612    63.9878  -1.798   0.0754 .
## efficiency_classE  -155.8857    68.2113  -2.285   0.0246 *
## efficiency_classF  -148.8190    70.2352  -2.119   0.0368 *
## efficiency_classG  -157.0446    94.9381  -1.654   0.1015
## efficiency_classH   -81.9826   119.0859  -0.688   0.4929
## rooms                -4.8661    16.2200  -0.300   0.7648
## building_year        -0.3341     0.6063  -0.551   0.5829
## square_meter         10.8994     0.9790  11.133   <2e-16 ***
## service_charges       0.2823     0.3476   0.812   0.4188
## cityessen           -54.6758    21.1237  -2.588   0.0112 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 98.77 on 93 degrees of freedom
## Multiple R-squared:  0.8162, Adjusted R-squared:  0.7905
## F-statistic: 31.76 on 13 and 93 DF,  p-value: < 2.2e-16
```

```
mod_zipcode<-lm(warm_rent~efficiency_class+rooms+building_year+square_meter+service_charges+
                data = train)

summary(mod_zipcode)
```

```
##
## Call:
## lm(formula = warm_rent ~ efficiency_class + rooms + building_year +
##     square_meter + service_charges + zipcode, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -264.66  -51.12    6.15   36.69  441.01
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6243.78881 2110.63391   2.958  0.00392 **
## efficiency_classA+   20.21662   86.42884   0.234  0.81557
## efficiency_classB   -92.93830   74.40000  -1.249  0.21474
## efficiency_classC  -125.78507   66.50587  -1.891  0.06169 .
## efficiency_classD   -95.02871   65.03338  -1.461  0.14732
## efficiency_classE  -133.74524   69.13513  -1.935  0.05609 .
```

```
## efficiency_classF  -126.97844   70.53946  -1.800  0.07509 .
## efficiency_classG  -134.18380   94.62180  -1.418  0.15950
## efficiency_classH   -65.39513  118.57393  -0.552  0.58260
## rooms                -1.97288   16.12926  -0.122  0.90291
## building_year        -0.07537    0.62553  -0.120  0.90435
## square_meter         10.87204    0.97418  11.160  < 2e-16 ***
## service_charges       0.27985    0.34579   0.809  0.42041
## zipcode              -0.13298    0.04774  -2.786  0.00647 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 98.25 on 93 degrees of freedom
## Multiple R-squared:  0.8181, Adjusted R-squared:  0.7927
## F-statistic: 32.18 on 13 and 93 DF,  p-value: < 2.2e-16
```

For both models, compute the RMSE on the 'test' dataset:

```
 test%>%
  rmse(mod_city,.)
```

```
## [1] 93.85988
```

```
test%>%
  rmse(mod_zipcode,.)
```

```
## [1] 87.5189
```

The prediction with the variable "zipcode" is a better fit.

```
mod_full<-lm(warm_rent~efficiency_class+rooms+building_year+square_meter+service_charges+
               zipcode+city, data = train)
 test%>%
  rmse(mod_full,.)
```

```
## [1] 84.69338
```

```
summary(mod_full)
```

```
##
## Call:
## lm(formula = warm_rent ~ efficiency_class + rooms + building_year +
##     square_meter + service_charges + zipcode + city, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -260.94  -50.53    7.44   32.87  439.65
##
## Coefficients:
```

```
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        8562.90502 7547.36795   1.135   0.2595
## efficiency_classA+   26.25710   88.87443   0.295   0.7683
## efficiency_classB   -86.89660   77.10624  -1.127   0.2627
## efficiency_classC  -117.46549   71.70298  -1.638   0.1048
## efficiency_classD   -87.83065   69.10841  -1.271   0.2070
## efficiency_classE  -125.38300   74.21839  -1.689   0.0945 .
## efficiency_classF  -118.03743   76.18479  -1.549   0.1247
## efficiency_classG  -124.28097   99.98575  -1.243   0.2170
## efficiency_classH   -58.30730  121.18927  -0.481   0.6316
## rooms                -0.67686   16.70549  -0.041   0.9678
## building_year         0.01938    0.69475   0.028   0.9778
## square_meter         10.86600    0.97910  11.098   <2e-16 ***
## service_charges       0.28053    0.34748   0.807   0.4216
## zipcode              -0.18913    0.18180  -1.040   0.3009
## cityessen            25.62082   80.02093   0.320   0.7496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 98.73 on 92 degrees of freedom
## Multiple R-squared:  0.8183, Adjusted R-squared:  0.7907
## F-statistic:  29.6 on 14 and 92 DF,  p-value: < 2.2e-16
```

```
summary(mod_city)
```

```
##
## Call:
## lm(formula = warm_rent ~ efficiency_class + rooms + building_year +
##     square_meter + service_charges + city, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -270.29  -47.48    5.92   37.20  447.66
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         815.7063  1227.3639   0.665   0.5080
## efficiency_classA+    8.1139    87.1850   0.093   0.9261
## efficiency_classB  -110.7968    73.6366  -1.505   0.1358
## efficiency_classC  -147.0374    65.8568  -2.233   0.0280 *
## efficiency_classD  -115.0612    63.9878  -1.798   0.0754 .
## efficiency_classE  -155.8857    68.2113  -2.285   0.0246 *
## efficiency_classF  -148.8190    70.2352  -2.119   0.0368 *
## efficiency_classG  -157.0446    94.9381  -1.654   0.1015
## efficiency_classH   -81.9826   119.0859  -0.688   0.4929
## rooms                -4.8661    16.2200  -0.300   0.7648
## building_year        -0.3341     0.6063  -0.551   0.5829
## square_meter         10.8994     0.9790  11.133   <2e-16 ***
## service_charges       0.2823     0.3476   0.812   0.4188
## cityessen           -54.6758    21.1237  -2.588   0.0112 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 98.77 on 93 degrees of freedom
## Multiple R-squared:  0.8162, Adjusted R-squared:  0.7905
## F-statistic: 31.76 on 13 and 93 DF,  p-value: < 2.2e-16
```

```
summary(mod_zipcode)
```

```
##
## Call:
## lm(formula = warm_rent ~ efficiency_class + rooms + building_year +
##     square_meter + service_charges + zipcode, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -264.66  -51.12    6.15   36.69  441.01
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6243.78881 2110.63391   2.958  0.00392 **
## efficiency_classA+   20.21662   86.42884   0.234  0.81557
## efficiency_classB   -92.93830   74.40000  -1.249  0.21474
## efficiency_classC  -125.78507   66.50587  -1.891  0.06169 .
## efficiency_classD   -95.02871   65.03338  -1.461  0.14732
## efficiency_classE  -133.74524   69.13513  -1.935  0.05609 .
## efficiency_classF  -126.97844   70.53946  -1.800  0.07509 .
## efficiency_classG  -134.18380   94.62180  -1.418  0.15950
## efficiency_classH   -65.39513  118.57393  -0.552  0.58260
## rooms                -1.97288   16.12926  -0.122  0.90291
## building_year        -0.07537    0.62553  -0.120  0.90435
## square_meter         10.87204    0.97418  11.160  < 2e-16 ***
## service_charges       0.27985    0.34579   0.809  0.42041
## zipcode              -0.13298    0.04774  -2.786  0.00647 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 98.25 on 93 degrees of freedom
## Multiple R-squared:  0.8181, Adjusted R-squared:  0.7927
## F-statistic: 32.18 on 13 and 93 DF,  p-value: < 2.2e-16
```

"Mod___full" has to much impact. A possible explanation could be multicollinearity due to using the variable "zipcode" and "city" in the same regression.

Exercise Number 3)

a) Create a function that uses a standard ggplot2 theme and customize the following elements according to your taste.

```
 theme_favourite <- function(){
  theme_classic() %+replace%
    theme(
      text = element_text(family = "serif", size = 16),
      panel.spacing = unit(2, "cm"),
      panel.grid.major = element_line(colour = "blue"),
      panel.grid.minor = element_line(color = "blue")

    )
}
```

b) Write a function ggscatt() that generates a scatter plot. The function should be able to create faceted plots. Your theme from (a) should be used.

```
ggscatt <- function(data,x,y){
  ggplot(data, aes(x = x, y=y)) +
    geom_point() +
    theme_favourite()+
    labs(title = "A scatter plot") +
    facet_grid()
}
```

c) Establish a relative connection to the database.

```
library(dplyr)
connection <- DBI::dbConnect(
  drv = RSQLite::SQLite(),
  dbname = here::here("assignment_1.sqlite3"),
)
# show the list of tables in the database
DBI::dbListTables(connection)
```

```
## [1] "metro"   "metro_2"
```

d) Load the table metro into R.

```
# create an object which is a reference to a table in the database
 metro<-tbl(connection, "metro")
head(metro)
```

```
## # Source:   lazy query [?? x 9]
## # Database: sqlite 3.38.5
## #   [C:\Users\chiara\Documents\GitHub\Assigments-Advanved-R\assignment_1.sqlite3]
##   holiday  temp rain_1h snow_1h clouds_all weather_main weather_description
##   <chr>   <dbl>   <dbl>   <dbl>      <dbl> <chr>        <chr>
## 1 None     288.       0       0         40 Clouds       scattered clouds
## 2 None     289.       0       0         75 Clouds       broken clouds
## 3 None     290.       0       0         90 Clouds       overcast clouds
```

```
## 4 None     290.       0       0         90 Clouds       overcast clouds
## 5 None     291.       0       0         75 Clouds       broken clouds
## 6 None     292.       0       0          1 Clear        sky is clear
## # ... with 2 more variables: date_time <dbl>, traffic_volume <dbl>
```

```
#actually pull the data into R.
metro_R<-metro %>%
  collect(n=Inf)
head(metro_R)
```

```
## # A tibble: 6 x 9
##   holiday  temp rain_1h snow_1h clouds_all weather_main weather_description
##   <chr>   <dbl>   <dbl>   <dbl>      <dbl> <chr>        <chr>
## 1 None     288.       0       0         40 Clouds       scattered clouds
## 2 None     289.       0       0         75 Clouds       broken clouds
## 3 None     290.       0       0         90 Clouds       overcast clouds
## 4 None     290.       0       0         90 Clouds       overcast clouds
## 5 None     291.       0       0         75 Clouds       broken clouds
## 6 None     292.       0       0          1 Clear        sky is clear
## # ... with 2 more variables: date_time <dbl>, traffic_volume <dbl>
```

e) Ensure that all variables are stored using a proper class.

```
 metro_R%<>%
  mutate(date_time=as_datetime(date_time))
tail(metro_R)
```

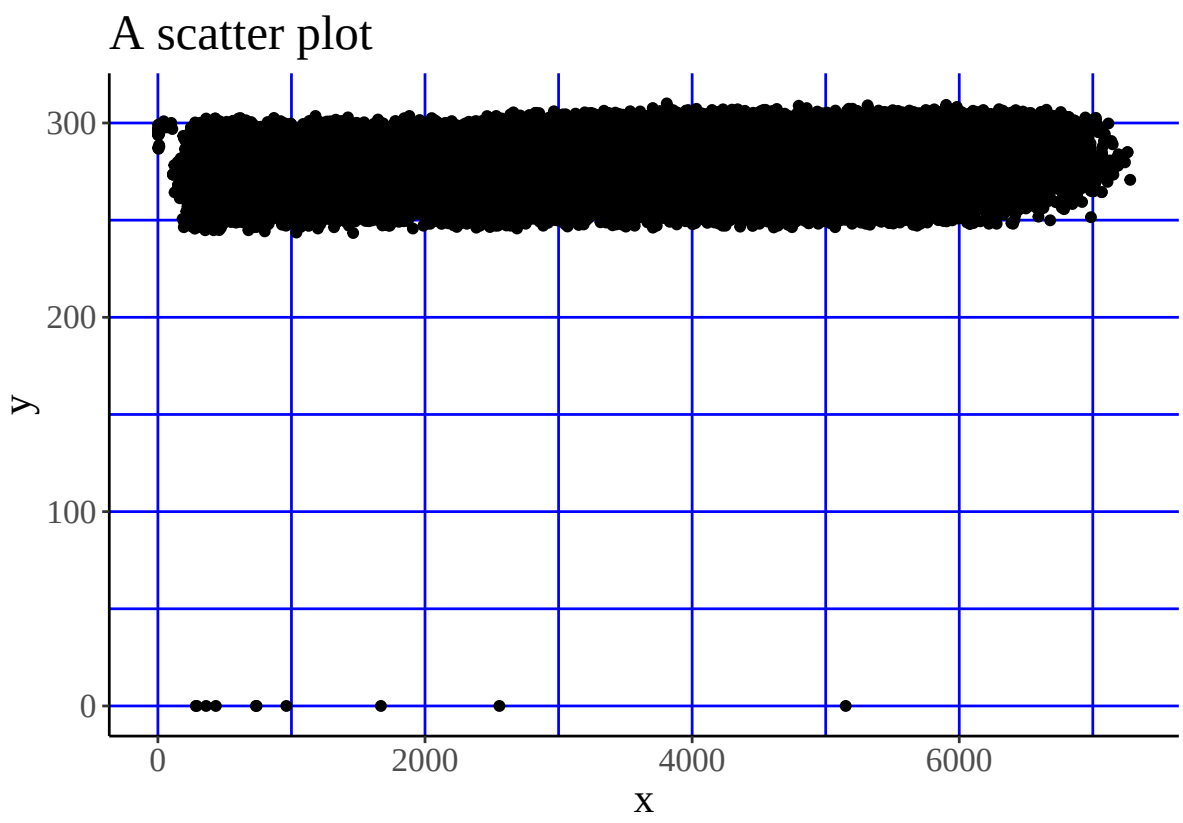
```
## # A tibble: 6 x 9
##   holiday  temp rain_1h snow_1h clouds_all weather_main weather_description
##   <chr>   <dbl>   <dbl>   <dbl>      <dbl> <chr>        <chr>
## 1 None     284.    0.25       0         75 Rain         light rain
## 2 None     283.    0          0         75 Clouds       broken clouds
## 3 None     283.    0          0         90 Clouds       overcast clouds
## 4 None     283.    0          0         90 Thunderstorm proximity thunderstorm
## 5 None     282.    0          0         90 Clouds       overcast clouds
## 6 None     282.    0          0         90 Clouds       overcast clouds
## # ... with 2 more variables: date_time <dttm>, traffic_volume <dbl>
```

f) create a new variable weekday that represents the weekdays.

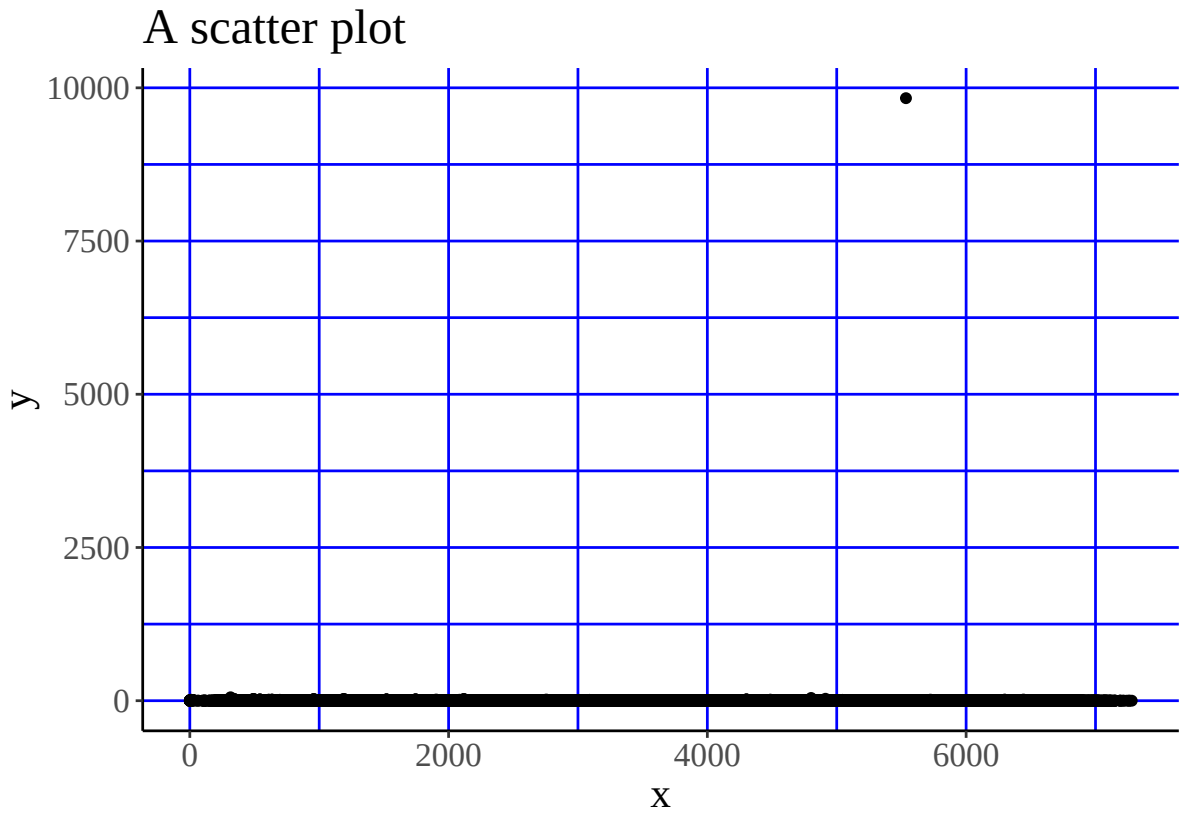
```
 metro_R %<>%
  mutate(weekday=weekdays(date_time))
```

g) Use your function ggscatt() from (b).

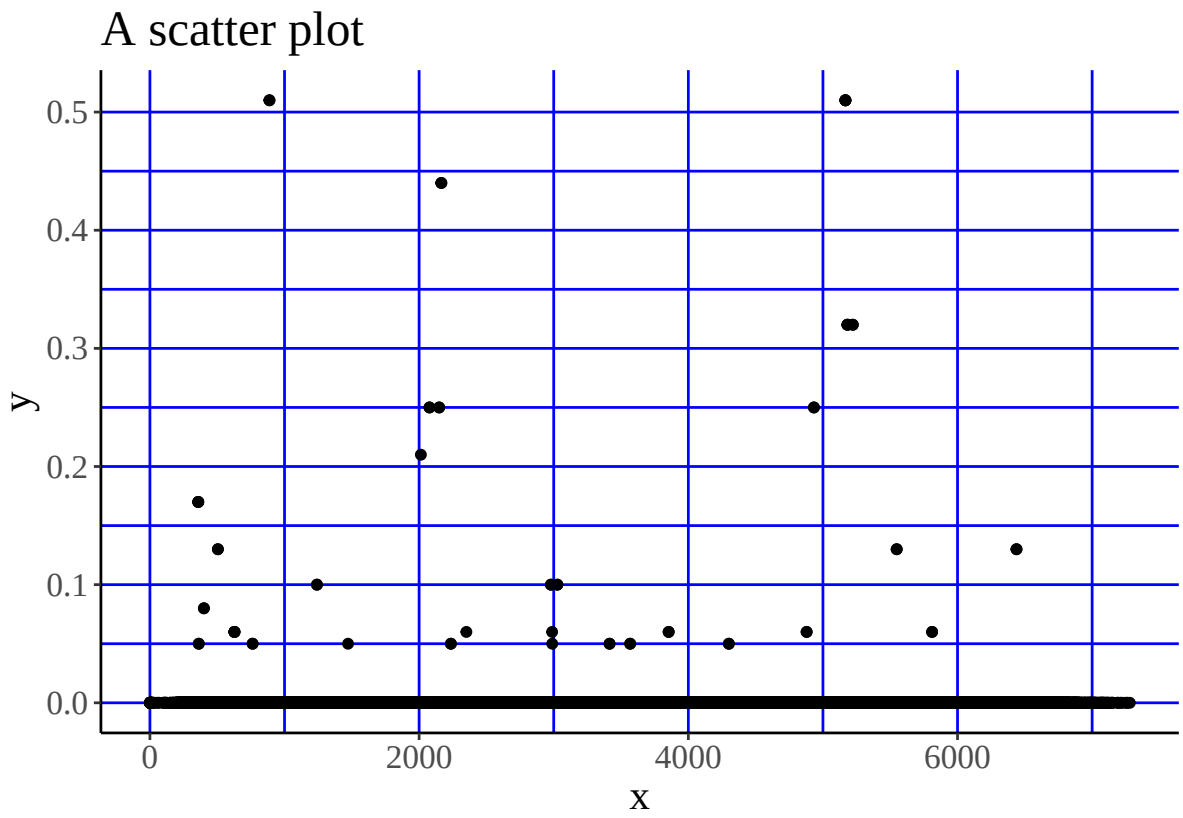
```
ggscatt(data = metro_R, metro_R$traffic_volume, metro_R$temp)
```



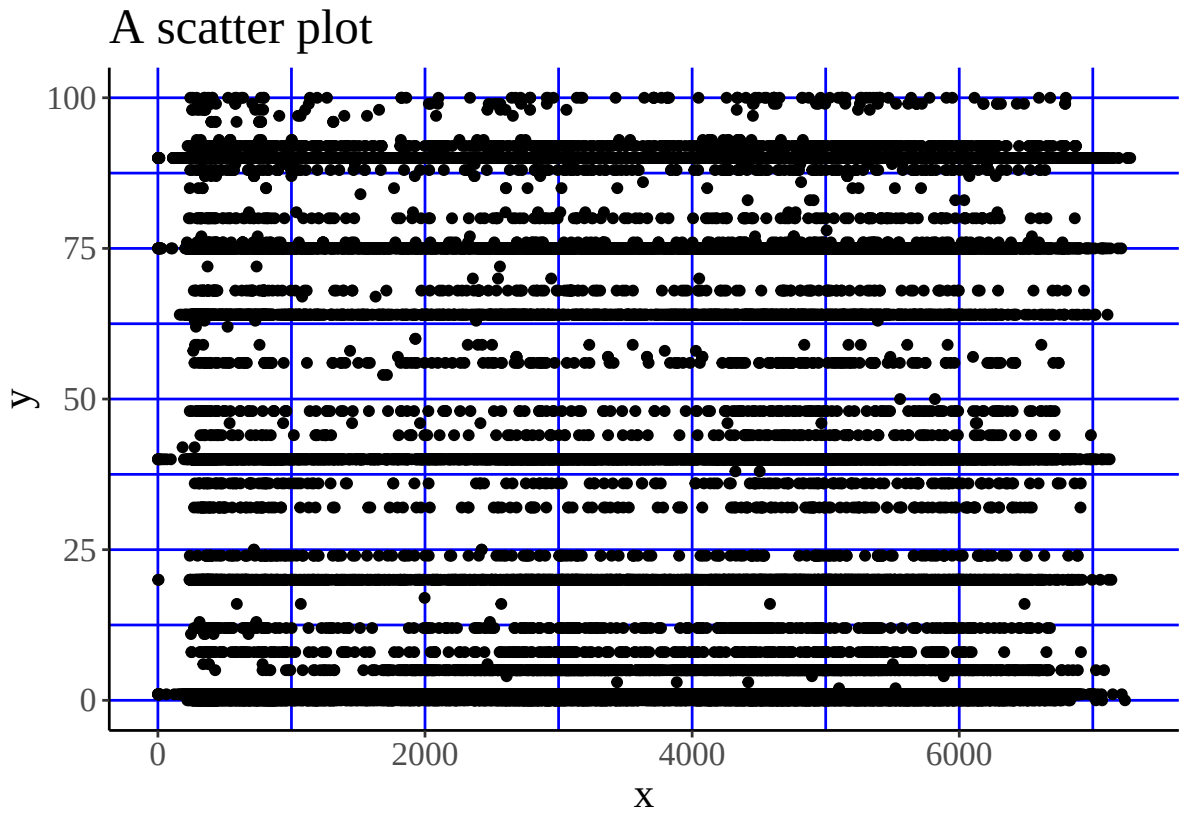

```
ggscatt(data = metro_R, metro_R$traffic_volume, metro_R$rain_1h)
```



```
ggscatt(data = metro_R, metro_R$traffic_volume, metro_R$snow_1h)
```

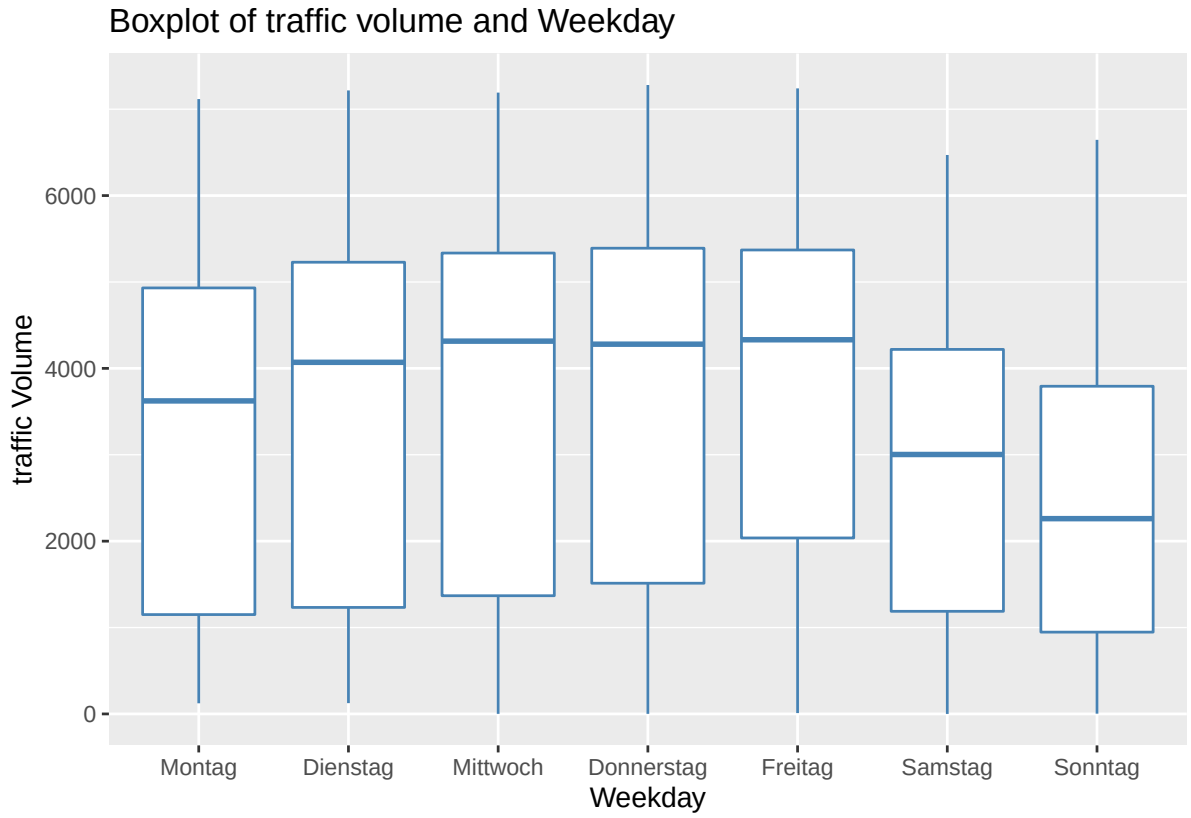



```
ggscatt(data = metro_R, metro_R$traffic_volume, metro_R$clouds_all)
```



h) Create a boxplot with the variable weekday on the x-axis and traffic_volume on the y-axis.

```
metro_R %>%
  mutate(weekday=factor(weekday, c("Montag","Dienstag","Mittwoch","Donnerstag","Freitag",
                                   "Samstag","Sonntag"))) %>%
  ggplot(aes(x=weekday, y=traffic_volume))+
  geom_boxplot(color="steelblue")+
  labs(title = "Boxplot of traffic volume and Weekday",
       x="Weekday", y="traffic Volume")
```



i) Add your modified dataframe as a table to the database under the same metro_2.

```
#dbWriteTable(connection,
                  #name="metro_2",
                  #value=metro_R)

#db_list_tables(connection)
```

Exercise Number 4)

a)

$ git rm byeGit.txt # removing and staging removal of byeGit.txt

$ git add HelloGit.txt # staging for commit

$ git status # make sure HelloGit.txt and byeGit.txt are staged

$ git commit -m “Fixes to file” # HelloGit.txt committed

$ git add assignment_1.sqlite3 # track files for staging $ git add rent_advertisment.RData

$ git commit -m “data added” # committed files

b)

$ git add -p HelloGit.txt ## break down file into hunks, Git will prompt you ## with a choice which hunks to stage for next commit $ git commit -m “Part 1”

$ git add HelloGit.txt ## stage rest of commitment

$ git commit -m “Part2” ## commit rest of commitment