

Analysis of Rental Housing Market

Fiseha Amine,Martina Schüpbach-Wolf

02 02 2021

Contents

Introduction	3
Aim of Analysis	3
Libraries	3
Data Import	4
Join second data set	5
Data Preparation	7
Data Modification	8
Graphical Analysis	9
Missing values	9
Graphical Visualisation and Analysis	10
Corellation matrix	10
1. Aim: Analysis of rental prices in relation to living space	11
2. Aim: Analysis of type of apartments that are mostly available (per canton)	13
Interactive Graphs	13
Further graphical visualisations for 2. aim	14
Analysis	16
3. Aim: The most expensive apartments	16
Tabullar presentation of most expensive apartments	17
4. Aim: The biggest apartments	18
5. Rental price in relation to size of apartments, number of rooms, population & location	19
Graphical interractive visualization	19
Fitting Linear Model	19
Residual analysis	21

-> Questions:

Am Schluss: - Datum am Schluss der Arbeit noch anpassen - viele (head) und (str) "unterdrücken / und auch das Zeigen der ganzen Datensätze - code der nicht mehr gebraucht wird - löschen - alle"labels" benannt? {r}

- Die Ziele (unten) können wir ja noch anpassen...
- Text "Missing values" unvollständig. Was wollen wir hier genau wissen??? Jenachdem Output anpassen. - Dann code anpassen. - Tabelle einbauen? - Export Bild

Introduction

This analysis is done as part of the R-Bootcamp module from HS 2020. The aim of the analysis within this scope is to experiment with the different functionalities and methods introduced in this course to answer some analytical and hypothetical research questions.

Aim of Analysis

Within this work we will analyse the house renting market in Switzerland with the help of open source data from the year 2019. Some of the analysis done in this work are:

1. Analyze the rental price in relation to living space
2. Analyze per canton what types of apartments are mostly available
3. Show where are the most expensive apartments (canton)
4. Show where are the biggest apartments
5. Rental price in relation to size of apartments, number of rooms, population & location

Libraries

```
library(readxl)
library(tidyverse)
library(dplyr)
library(mlbench)
library(mice)
library(ggplot2)
library(plotly)
library(shiny)
library(esquisse)
library(fBasics)
library(lmtest)
```

```
### needed to export plotly to PDF
#install.packages("webshot")
#webshot::install_phantomjs()

###package for esquisser
#install.packages("esquisse")
```

Data Import

The data set was downloaded from “Datenportal Schweiz_Wohungsmieten”: <https://datenportal.info/wohnungsmarkt/wohnungsmieten/>

It contains the following attributes (in German): ID, Year, Quarter, Apartment type, Room, Area, Rent Gross, Square Meter, Price Gross, Address, Postcode and City, Move-In Date, Building Coordinates, and Canton.

```
#Because the data contain "Umlaute" we use "UTF-8" for the read in.
data <- read.csv2("mietinserate_v1.csv", header = TRUE, fileEncoding = "UTF-8")
head(data)

##   ID Jahr Quartal      Wohnungstyp Zimmer Fläche Mietpreis_Brutto
## 1  1 2019       4            Wohnung    4.5          1850
## 2  2 2019       2            Wohnung    3.5          75        1180
## 3  3 2019       4  Maisonettewohnung    3.5          1400
## 4  4 2019       2            Wohnung    4.5          1750
## 5  5 2019       2            Wohnung    5.5          2450
## 6  6 2019       4            Wohnung     2          1200
##   Quadratmeterpreis_Brutto      Adresse      PLZ_Ort Bezugsdatum  GKODE
## 1                               NA Aachstrasse 13 9327 Tübach 2752059
## 2                               16 Aachstrasse 29 8586 Erlen 2734483
## 3                               NA Aachstrasse 31b 8586 Erlen 2734427
## 4                               NA Aachstrasse 5 9327 Tübach 2751994
## 5                               NA Aachstrasse 5 9327 Tübach 2751994
## 6                               NA Aachstrasse 5 9327 Tübach 2751994
##   GKODN KT
## 1 1261281 SG
## 2 1268102 TG
## 3 1268073 TG
## 4 1261267 SG
## 5 1261267 SG
## 6 1261267 SG

str(data)

## 'data.frame': 56777 obs. of 14 variables:
## $ ID           : int  1 2 3 4 5 6 7 8 9 10 ...
## $ Jahr         : int  2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 ...
## $ Quartal      : int  4 2 4 2 2 4 2 4 2 4 ...
## $ Wohnungstyp  : chr  "Wohnung" "Wohnung" "Maisonettewohnung" "Wohnung" ...
## $ Zimmer       : chr  "4.5" "3.5" "3.5" "4.5" ...
## $ Fläche       : chr  "" "75" "" "" ...
## $ Mietpreis_Brutto : int  1850 1180 1400 1750 2450 1200 1840 1200 1850 1500 ...
## $ Quadratmeterpreis_Brutto: int  NA 16 NA NA NA NA NA NA NA 19 ...
## $ Adresse       : chr  "Aachstrasse 13" "Aachstrasse 29" "Aachstrasse 31b" "Aachstrasse 5 ...
## $ PLZ_Ort       : chr  "9327 Tübach" "8586 Erlen" "8586 Erlen" "9327 Tübach" ...
## $ Bezugsdatum   : chr  "" "" "" "" ...
## $ GKODE         : chr  "2752059" "2734483" "2734427" "2751994" ...
## $ GKODN         : chr  "1261281" "1268102" "1268073" "1261267" ...
## $ KT           : chr  "SG" "TG" "TG" "SG" ...
```

```

summary(data)

##      ID          Jahr        Quartal    Wohnungstyp
##  Min.   : 1   Min.   :2019   Min.   :2.000  Length:56777
##  1st Qu.:14195 1st Qu.:2019   1st Qu.:2.000  Class  :character
##  Median :28389  Median :2019   Median :2.000  Mode   :character
##  Mean   :28389  Mean   :2019   Mean   :2.919
##  3rd Qu.:42583 3rd Qu.:2019   3rd Qu.:4.000
##  Max.   :56777  Max.   :2019   Max.   :4.000
##
##      Zimmer        Fläche     Mietpreis_Brutto
##  Length:56777  Length:56777  Min.   : 120
##  Class  :character  Class  :character  1st Qu.: 1300
##  Mode   :character  Mode   :character  Median  : 1600
##                                         Mean   : 1707
##                                         3rd Qu.: 1950
##                                         Max.   :31800
##
##      Quadratmeterpreis_Brutto  Adresse          PLZ_Ort
##  Min.   : 3.00           Length:56777  Length:56777
##  1st Qu.: 17.00          Class  :character  Class  :character
##  Median : 20.00          Mode   :character  Mode   :character
##  Mean   : 21.43
##  3rd Qu.: 23.00
##  Max.   :149.00
##  NA's   :8323
##      Bezugsdatum       GKODE        GKODN          KT
##  Length:56777  Length:56777  Length:56777  Length:56777
##  Class  :character  Class  :character  Class  :character  Class  :character
##  Mode   :character  Mode   :character  Mode   :character  Mode   :character
##
##      
```

Join second data set

To also have an overview of the number of residents in the cantons, we add a new data set. We extracted the data set from: <https://www.citypopulation.de/de/switzerland/cities/>. It contains the column names: Kanton, KT, Hauptstädte, G.Fläche, Einwohner_1980, Einwohner_1990, Einwohner_2000, Einwohner_2010, and Einwohner_2019.

```

#Read new data set and join new data set to our data set.
data_canton <- read_xlsx("SchweizKantone_Einwohner_org.xlsx", skip = 3)
head(data_canton)

```

```

## # A tibble: 6 x 9
##   Schweiz CHE Bern `41285` `6365960` `6873687` `7288010` `7870134` `8603899`
##   <chr>  <chr> <chr> <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 Aargau AG   Aarau  1404     453442   507508   547493   611466   685845
## 2 Appenze~ AR   Heri~   243      47611    52229    53504    53017    55445
## 3 Appenze~ AI   Appe~   173      12844    13870    14618    15688    16128

```

```

## 4 Basel-L~ BL    Lies~      518    233447    248484    259374    274404    289468
## 5 Basel-S~ BS    Basel       37     203915    199411    188079    184950    195844
## 6 Bern     BE    Bern      5941    898397    943196    957197    979802    1039474

str(data_canton)

## # tibble [26 x 9] (S3: tbl_df/tbl/data.frame)
## $ Schweiz: chr [1:26] "Aargau" "Appenzell Ausserrhoden" "Appenzell Innerrhoden" "Basel-Landschaft"
## $ CHE   : chr [1:26] "AG" "AR" "AI" "BL" ...
## $ Bern  : chr [1:26] "Aarau" "Herisau" "Appenzell" "Liestal" ...
## $ 41285 : num [1:26] 1404 243 173 518 37 ...
## $ 6365960: num [1:26] 453442 47611 12844 233447 203915 ...
## $ 6873687: num [1:26] 507508 52229 13870 248484 199411 ...
## $ 7288010: num [1:26] 547493 53504 14618 259374 188079 ...
## $ 7870134: num [1:26] 611466 53017 15688 274404 184950 ...
## $ 8603899: num [1:26] 685845 55445 16128 289468 195844 ...

#change names of original 4th row (now 1st) of the data set to have those as the column names and for the
data_canton <- rename(data_canton, Kanton = Schweiz, KT = CHE, Hauptstädte = Bern, G.Fläche = "41285", .after = 4)

head(data_canton)

## # A tibble: 6 x 9
##   Kanton KT    Hauptstädte G.Fläche Einwohner_1980 Einwohner_1990 Einwohner_2000
##   <chr>  <chr> <chr>      <dbl>        <dbl>        <dbl>        <dbl>
## 1 Aargau AG    Aarau       1404        453442      507508      547493
## 2 Appen~ AR    Herisau      243         47611       52229       53504
## 3 Appen~ AI    Appenzell    173         12844       13870       14618
## 4 Basel~ BL    Liestal      518         233447      248484      259374
## 5 Basel~ BS    Basel        37          203915      199411      188079
## 6 Bern     BE    Bern       5941        898397      943196      957197
## # ... with 2 more variables: Einwohner_2010 <dbl>, Einwohner_2019 <dbl>

#join data sets
data_left_join <- left_join(data, data_canton, by = "KT")
head(data_left_join)

##   ID Jahr Quartal    Wohnungstyp Zimmer Fläche Mietpreis_Brutto
## 1  1  2019     4      Wohnung    4.5           1850
## 2  2  2019     2      Wohnung    3.5          75     1180
## 3  3  2019     4 Maisonettewohnung  3.5           1400
## 4  4  2019     2      Wohnung    4.5           1750
## 5  5  2019     2      Wohnung    5.5           2450
## 6  6  2019     4      Wohnung    2            1200
##   Quadratmeterpreis_Brutto      Adresse PLZ_Ort Bezugsdatum GKODE
## 1                               NA Aachstrasse 13 9327 Tübach      2752059
## 2                               16 Aachstrasse 29 8586 Erlen      2734483
## 3                               NA Aachstrasse 31b 8586 Erlen      2734427
## 4                               NA Aachstrasse 5 9327 Tübach      2751994
## 5                               NA Aachstrasse 5 9327 Tübach      2751994
## 6                               NA Aachstrasse 5 9327 Tübach      2751994
##   GKODN KT    Kanton Hauptstädte G.Fläche Einwohner_1980 Einwohner_1990

```

```

## 1 1261281 SG St. Gallen St. Gallen 2026 391995 427501
## 2 1268102 TG Thurgau Frauenfeld 991 183795 209362
## 3 1268073 TG Thurgau Frauenfeld 991 183795 209362
## 4 1261267 SG St. Gallen St. Gallen 2026 391995 427501
## 5 1261267 SG St. Gallen St. Gallen 2026 391995 427501
## 6 1261267 SG St. Gallen St. Gallen 2026 391995 427501
## Einwohner_2000 Einwohner_2010 Einwohner_2019
## 1 452837 478907 510734
## 2 228875 248444 279547
## 3 228875 248444 279547
## 4 452837 478907 510734
## 5 452837 478907 510734
## 6 452837 478907 510734

```

Data Preparation

For the upcoming analysis the data needs to get formatted and cleaned. We transform the data into a data frame. We also transform numeric data into numeric and transform the date data into the correct format. we also rename column names for better readability.

```

data_2 <- as.data.frame(data_left_join)

#transform data
data_2$Zimmer = as.numeric(data_left_join$Zimmer)
data_2$Fläche = as.numeric(data_left_join$Fläche)
data_2[data_2 == ""] <- NA
data_2$Bezugsdatum = as.Date(data_left_join$Bezugsdatum, "%d.%m.%Y")

#rename data columns for a better readability
data_1 <- rename(data_2, W.Type = Wohnungstyp, Miete = Mietpreis_Brutto, Quad_m = Quadratmeterpreis_Brutto)
head(data_1)

```

```

## ID Jahr Quartal W.Type Zimmer W.Fläche Miete Quad_m
## 1 1 2019 4 Wohnung 4.5 NA 1850 NA
## 2 2 2019 2 Wohnung 3.5 75 1180 16
## 3 3 2019 4 Maisonettewohnung 3.5 NA 1400 NA
## 4 4 2019 2 Wohnung 4.5 NA 1750 NA
## 5 5 2019 2 Wohnung 5.5 NA 2450 NA
## 6 6 2019 4 Wohnung 2.0 NA 1200 NA
## Adresse PLZ_Ort Bezug GKODE GKODN KT Kanton Hauptstädte
## 1 Aachstrasse 13 9327 Tübach <NA> 2752059 1261281 SG St. Gallen St. Gallen
## 2 Aachstrasse 29 8586 Erlen <NA> 2734483 1268102 TG Thurgau Frauenfeld
## 3 Aachstrasse 31b 8586 Erlen <NA> 2734427 1268073 TG Thurgau Frauenfeld
## 4 Aachstrasse 5 9327 Tübach <NA> 2751994 1261267 SG St. Gallen St. Gallen
## 5 Aachstrasse 5 9327 Tübach <NA> 2751994 1261267 SG St. Gallen St. Gallen
## 6 Aachstrasse 5 9327 Tübach <NA> 2751994 1261267 SG St. Gallen St. Gallen
## G.Fläche Einwohner_1980 Einwohner_1990 Einwohner_2000 Einwohner_2010
## 1 2026 391995 427501 452837 478907
## 2 991 183795 209362 228875 248444
## 3 991 183795 209362 228875 248444
## 4 2026 391995 427501 452837 478907
## 5 2026 391995 427501 452837 478907

```

```

## 6      2026      391995      427501      452837      478907
##   Einwohner_2019
## 1      510734
## 2      279547
## 3      279547
## 4      510734
## 5      510734
## 6      510734

```

##Delete Columns

Here we delete columns that are not needed within the scope of this analysis. Those columns appear in the dataset after joining two different datasets in the previous step.

```

#delete columns that are not needed anymore
data_3 <- data_1 %>% select(-Einwohner_1980, -Einwohner_1990, -Einwohner_2000, -Einwohner_2010)
head(data_3)

```

```

##   ID Jahr Quartal      W.Type Zimmer W.Fläche Miete Quad_m
## 1  1 2019     4      Wohnung    4.5      NA  1850      NA
## 2  2 2019     2      Wohnung    3.5      75 1180      16
## 3  3 2019     4 Maisonettewohnung  3.5      NA 1400      NA
## 4  4 2019     2      Wohnung    4.5      NA 1750      NA
## 5  5 2019     2      Wohnung    5.5      NA 2450      NA
## 6  6 2019     4      Wohnung    2.0      NA 1200      NA
##          Adresse     PLZ_Ort Bezug    GKODE   GKODN KT      Kanton Hauptstädte
## 1 Aachstrasse 13 9327 Tübach <NA> 2752059 1261281 SG St. Gallen St. Gallen
## 2 Aachstrasse 29 8586 Erlen  <NA> 2734483 1268102 TG Thurgau Frauenfeld
## 3 Aachstrasse 31b 8586 Erlen <NA> 2734427 1268073 TG Thurgau Frauenfeld
## 4 Aachstrasse 5 9327 Tübach <NA> 2751994 1261267 SG St. Gallen St. Gallen
## 5 Aachstrasse 5 9327 Tübach <NA> 2751994 1261267 SG St. Gallen St. Gallen
## 6 Aachstrasse 5 9327 Tübach <NA> 2751994 1261267 SG St. Gallen St. Gallen
##   G.Fläche Einwohner_2019
## 1      2026      510734
## 2      991       279547
## 3      991       279547
## 4      2026      510734
## 5      2026      510734
## 6      2026      510734

```

Data Modification

In this step we split the column “PLZ_Ort” which consists a postal code and city name to PLZ and Ort.

```

data_sep <- data_3$PLZ_Ort%>%
  str_match('(.*\d) (.*)') %>%
  as_tibble() %>%
  select(2:3) %>%
  set_names(c('PLZ', 'Ort'))

## Warning: The 'x' argument of 'as_tibble.matrix()' must have unique column names if '.name_repair' is
## Using compatibility '.name_repair'.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_warnings()' to see where this warning was generated.

```

```
head(data_sep)
```

```
## # A tibble: 6 x 2
##   PLZ     Ort
##   <chr>  <chr>
## 1 9327   Tübach
## 2 8586   Erlen
## 3 8586   Erlen
## 4 9327   Tübach
## 5 9327   Tübach
## 6 9327   Tübach
```

As these columns are saved separately from the main dataset, we merge it again to the main dataset and remove the old column “PLZ_Ort”.

```
data_sort <- data_3%>%select(-PLZ_Ort)
data_clean <- bind_cols(data_sort, data_sep)
head(data_clean)
```

```
##   ID Jahr Quartal      W.Type Zimmer W.Fläche Miete Quad_m
## 1  1 2019     4      Wohnung    4.5      NA  1850      NA
## 2  2 2019     2      Wohnung    3.5      75 1180      16
## 3  3 2019     4 Maisonettewohnung  3.5      NA 1400      NA
## 4  4 2019     2      Wohnung    4.5      NA 1750      NA
## 5  5 2019     2      Wohnung    5.5      NA 2450      NA
## 6  6 2019     4      Wohnung    2.0      NA 1200      NA
##   Adresse Bezug   GKODE   GKODN KT      Kanton Hauptstädte G.Fläche
## 1 Aachstrasse 13 <NA> 2752059 1261281 SG St. Gallen St. Gallen 2026
## 2 Aachstrasse 29 <NA> 2734483 1268102 TG Thurgau Frauenfeld 991
## 3 Aachstrasse 31b <NA> 2734427 1268073 TG Thurgau Frauenfeld 991
## 4 Aachstrasse 5  <NA> 2751994 1261267 SG St. Gallen St. Gallen 2026
## 5 Aachstrasse 5  <NA> 2751994 1261267 SG St. Gallen St. Gallen 2026
## 6 Aachstrasse 5  <NA> 2751994 1261267 SG St. Gallen St. Gallen 2026
##   Einwohner_2019 PLZ     Ort
## 1             510734 9327 Tübach
## 2             279547 8586 Erlen
## 3             279547 8586 Erlen
## 4             510734 9327 Tübach
## 5             510734 9327 Tübach
## 6             510734 9327 Tübach
```

Graphical Analysis

With the help of graphical visualization we try to see if there are some patterns, outliers, missing values, and if the distribution of the dataset is normal.

Missing values

Below a graphical and numerical analysis of all missing data is applied to have an overview of the missing values and their distribution within the dataset. This is necessary to avoid bias in the dataset and have a

balanced representation of all cantons equally. For this reason we observe which cantons have how many NAs and if any action (deleting, imputing, replacing, etc.) is needed.

```
colSums(is.na(data_clean)) ##counts all NAs shown in a table  
md.pattern(data_clean, rotate.names = TRUE) ##shows a table and a plot of the data for the NAs
```

The graphical presentation of missing values shows that there is high missing values in number of rooms, square meter of rooms, and entrance date. There is no action which is taken to remove or replace missing values and we will continue our analysis with the missing values included.

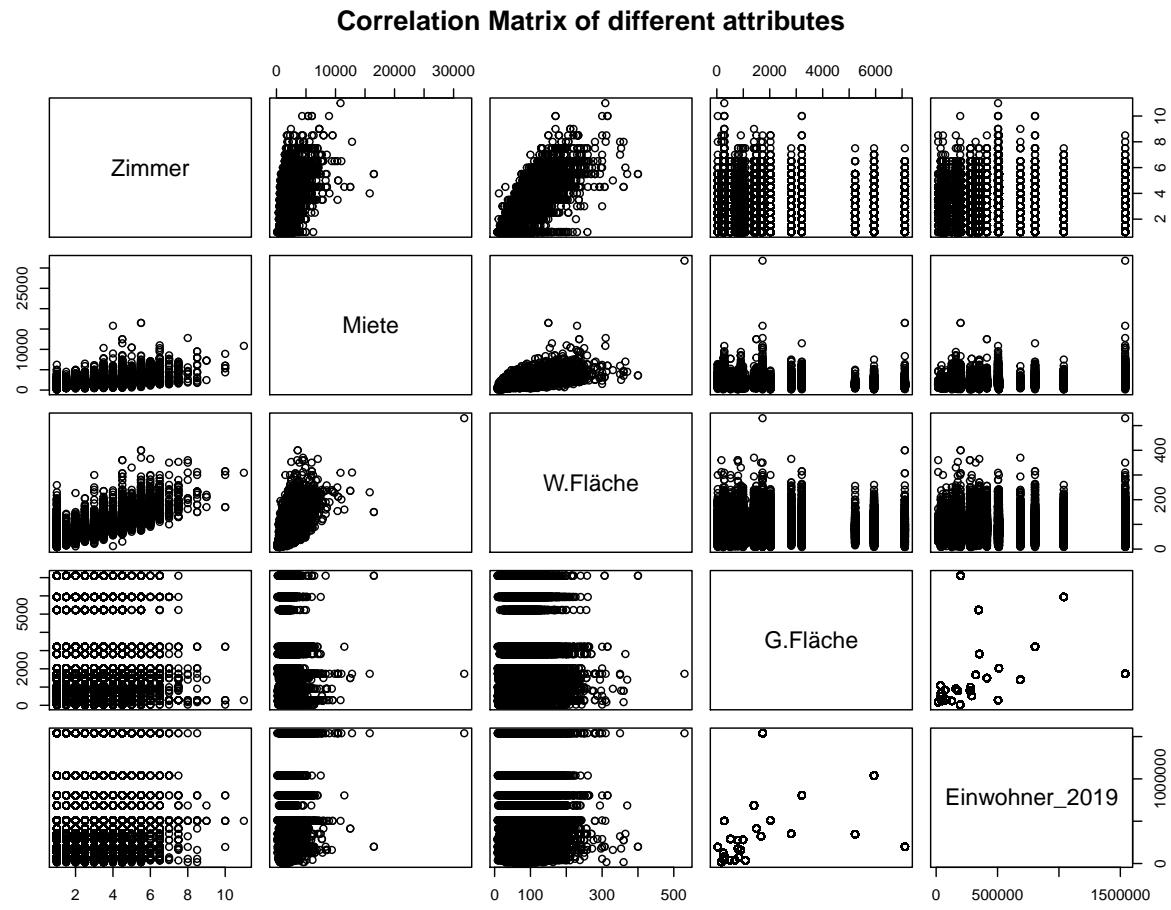
```
#some example searches for NAs  
  
#data_clean %>%  
#  group_by(KT) %>%  
#  summarise(NAs_Zimmer_count = sum(is.na(data_clean$Zimmer))) #Lists all cantons and shows the sum of  
  
#data_clean %>%  
#  filter(is.na(data_clean$Zimmer))%>%  
#  count(data_clean$Zimmer) #Lists all cantons including "NAs" for the column "Zimmer" and sums them
```

Graphical Visualisation and Analysis

Corellation matrix

To have a multi-dimensional visualization of several attributes, we visualize a correlation matrix.

```
pairs(~Zimmer+Miete+W.Fläche+G.Fläche+Einwohner_2019, data=data_clean,  
      main="Correlation Matrix of different attributes")
```



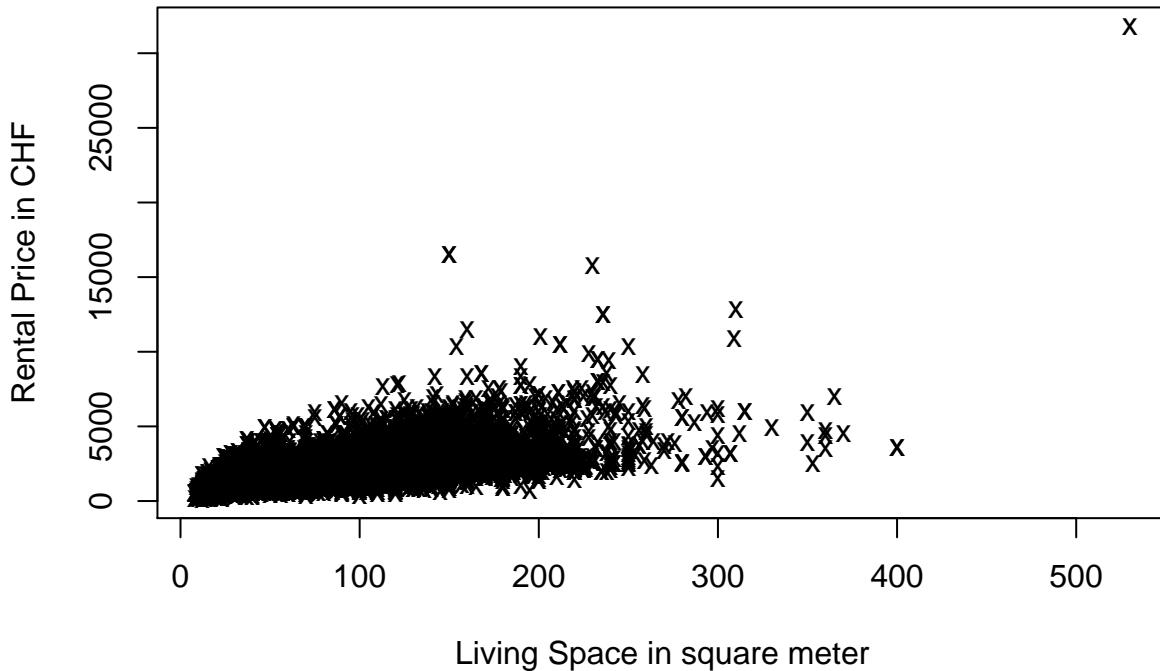
The correlation matrix shows whether there is a correlation among the different factors or not. From the above correlation matrix we can observe that there is a positive correlation between population of a canton and the living area in a commune.

1. Aim: Analysis of rental prices in relation to living space

Below we visualize the distribution of living area and correlating rental price.

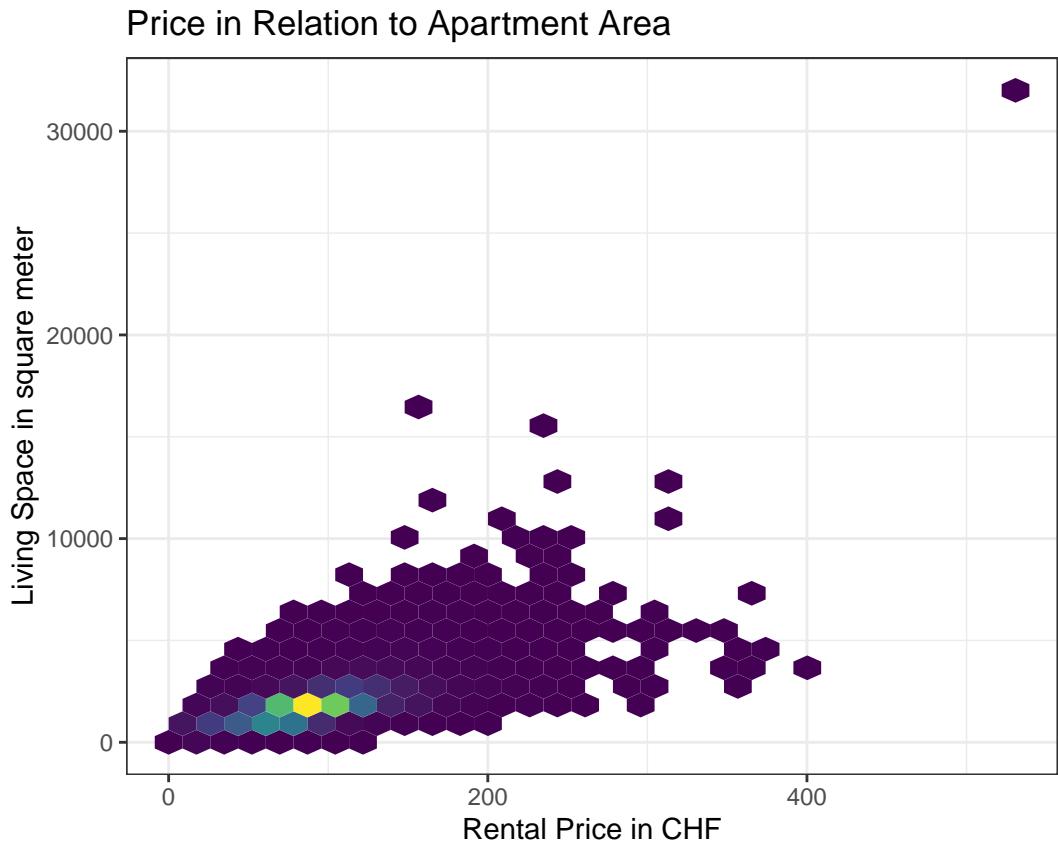
```
plot(y = data_clean$Miete,
      x = data_clean$W.Fläche,
      pch = "x",
      title("Rental Price per Apartment"),
      xlab = "Living Space in square meter",
      ylab = "Rental Price in CHF")
```

Rental Price per Apartment



The above plot shows that the range of the area of rental apartments lays between 0 and 200 square meter and costs less than 5000 Swiss Francs. There are some outliers with a living area of above 300 and rental price above 10'000 Swiss Francs.

```
g <- ggplot(data_clean, aes(x=W.Fläche, y=Miete, xlab = "Rental Price", ylab = "Apartment Area") ) +  
  geom_hex(bons = 70) +  
  scale_fill_continuous(type = "viridis") +  
  theme_bw()  
  
## Warning: Ignoring unknown parameters: bons  
  
g + ggtitle("Price in Relation to Apartment Area") +  
  xlab("Rental Price in CHF") + ylab("Living Space in square meter")  
  
## Warning: Removed 8262 rows containing non-finite values (stat_binhex).
```



The above plot shows the same result as seen with the first plot, however with numerical metric.

2. Aim: Analysis of type of apartments that are mostly available (per canton)

Interactive Graphs

With the help of graphical functions we can produce quality interactive graphs for presentation, reporting, publication, etc. For this purpose we will produce some graphs using “plotly”.

```
count_W_Type <- data_clean %>%
  group_by(KT, W.Type) %>%
  count(KT, sort = TRUE)

fig <- plot_ly(
  count_W_Type, x = ~KT, y = ~n, type = 'bar', color = ~W.Type)

fig <- fig %>% layout(barmode = "stack",
                        title = 'Apartments per Canton',
                        xaxis = list(title = "Canton"),
                        yaxis = list(title = "Number Apartments"))

fig
```

PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, please

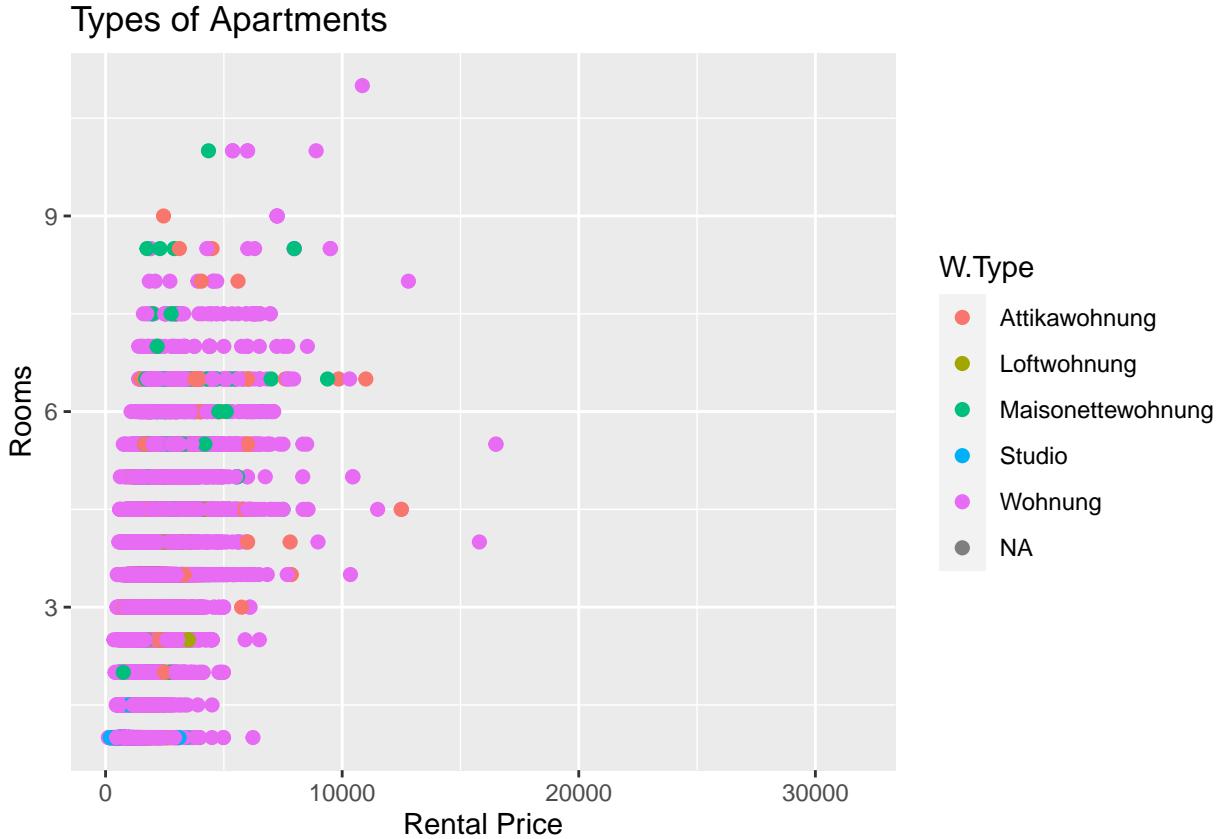
Further graphical visualisations for 2. aim

```

g <- ggplot(data_clean, aes(x=Miete, y=Zimmer, color=W.Type))+
  geom_point(size=2)
g + ggtitle("Types of Apartments") +
  xlab("Rental Price") + ylab("Rooms")

```

Warning: Removed 224 rows containing missing values (geom_point).



The above plot shows the type of rental housing available for rent colorfully. It shows that most of the rental objects available are apartments (pink colour).

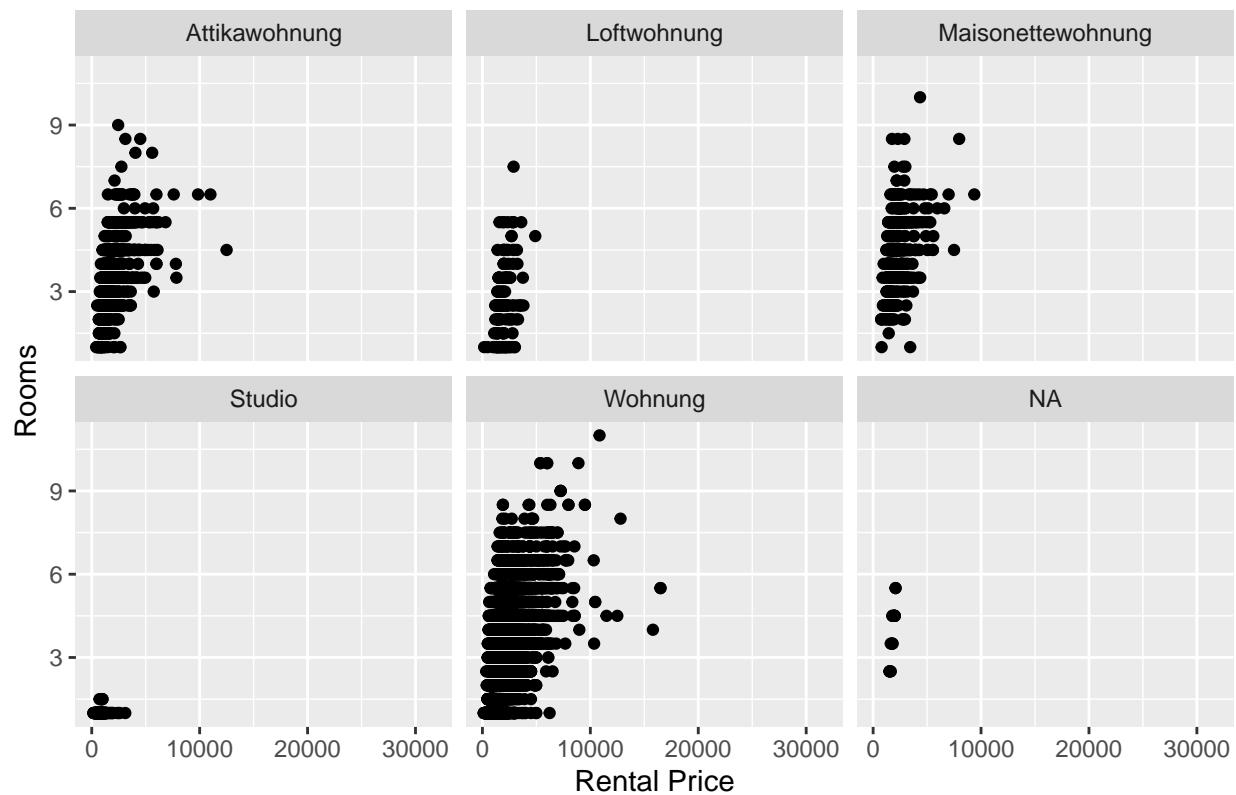
```

##  
  

qplot(y = Zimmer, x = Miete, data = data_clean, facets = ~ W.Type, main = "Types of Apartments", xlab =
## Warning: Removed 224 rows containing missing values (geom_point).

```

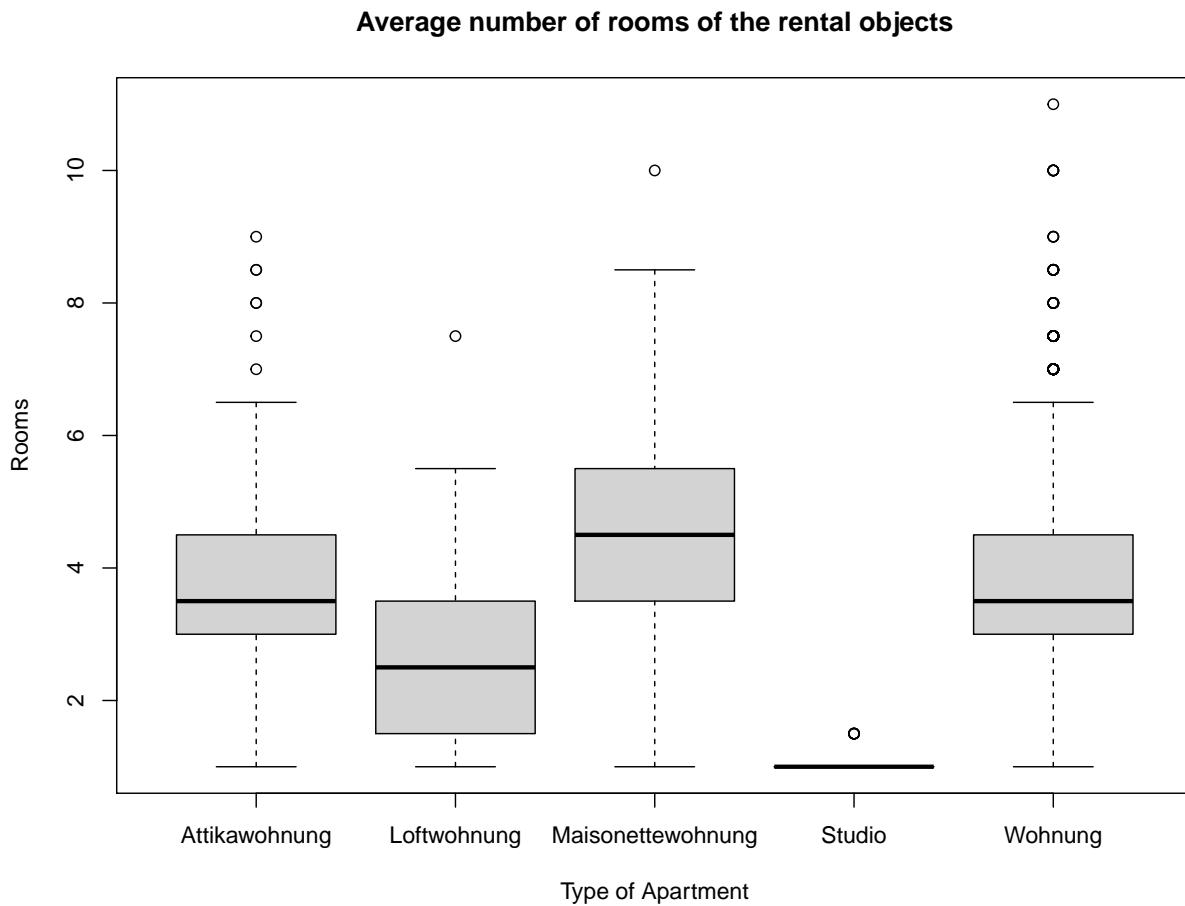
Types of Apartments



The above graphs shows the same information as above however in a separately displayed plots. The plot is produced with qplot.

```
##
```

```
boxplot(Zimmer ~ W.Type, data = data_clean, main = "Average number of rooms of the rental objects", xla
```



The above boxplots visualize the average number of rooms for different types of rental objects. Rental objects of type “Maisonettewohnung” have in average 5 rooms, where as Rental objects of type “Studio” have in average one room.

Analysis

3. Aim: The most expensive apartments

```
top10 <- data_clean%>%top_n(n = 10, wt = Miete)

top10_sorted <- top10[order(top10$Miete, na.last=TRUE) , ]
top10_sorted
```

ID	Jahr	Quartal	W.Type	Zimmer	W.Fläche	Miete	Quad_m	
2	33986	2019	2	Wohnung	11.0	309	10852	35
5	44712	2019	4	Attikawohnung	6.5	201	11000	55
1	9970	2019	4	Wohnung	4.5	160	11500	72
6	48785	2019	2	Wohnung	4.5	236	12500	53
7	48786	2019	4	Attikawohnung	4.5	236	12500	53
8	50134	2019	4	Wohnung	8.0	310	12800	41

```

## 4 42859 2019      2     Wohnung   4.0    230 15800    69
## 9 52755 2019      2     Wohnung   5.5    150 16500   110
## 10 52756 2019     4     Wohnung   5.5    150 16500   110
## 3 42857 2019      2     Wohnung   NA     530 31800    60
##               Adresse   Bezug   GKODE   GKODN KT   Kanton
## 2 Place Edouard-Claparède 3 2019-05-01 2500530 1116881 GE Genève [Genf]
## 5           Seestrasse 500 2019-10-01 2683140 1243527 ZH Zürich
## 1   Chemin des Mouettes 16 2019-10-01 2537468 1151218 VD Vaud [Waadt]
## 6           Tivolistrasse 8 <NA> 2667477 1212161 LU Luzern
## 7           Tivolistrasse 8 <NA> 2667477 1212161 LU Luzern
## 8           Utoquai 37 <NA> 2683708 1246462 ZH Zürich
## 4   Scheideggstrasse 79 <NA> 2682462 1245216 ZH Zürich
## 9           Via Tinus 52 <NA> 2783943 1152625 GR Graubünden
## 10          Via Tinus 52 <NA> 2783943 1152625 GR Graubünden
## 3   Scheideggstrasse 79 <NA> 2682462 1245216 ZH Zürich
##   Hauptstädte G.Fläche Einwohner_2019 PLZ   Ort
## 2   Genève     282     504128 1205   Genève
## 5   Zürich    1729     1539275 8038   Zürich
## 1   Lausanne   3212     805098 1007   Lausanne
## 6   Luzern    1493     413120 6006   Luzern
## 7   Luzern    1493     413120 6006   Luzern
## 8   Zürich    1729     1539275 8008   Zürich
## 4   Zürich    1729     1539275 8038   Zürich
## 9   Chur       7105     199021 7500 St. Moritz
## 10  Chur       7105     199021 7500 St. Moritz
## 3   Zürich    1729     1539275 8038   Zürich

```

The most expensive apartment is located in the canton of Zurich and it costs 31'800 Swiss Francs per month. The 10th most expensive apartment costs 10'852 and it is located in the canton of geneve. Here we consider only price, however other factors like apartment area and specific location could be also studied.

Tabular presentation of most expensive apartments

Subsetting a datafram to have a summarized view of the needed factors.

```
data_subset <- top10_sorted %>% select(W.Type, Zimmer, W.Fläche, Miete, Quad_m, KT, Ort)
data_subset
```

```

##       W.Type Zimmer W.Fläche Miete Quad_m KT   Ort
## 2     Wohnung   11.0    309 10852    35 GE   Genève
## 5 Attikawohnung   6.5    201 11000    55 ZH   Zürich
## 1     Wohnung   4.5    160 11500    72 VD   Lausanne
## 6     Wohnung   4.5    236 12500    53 LU   Luzern
## 7 Attikawohnung   4.5    236 12500    53 LU   Luzern
## 8     Wohnung   8.0    310 12800    41 ZH   Zürich
## 4     Wohnung   4.0    230 15800    69 ZH   Zürich
## 9     Wohnung   5.5    150 16500   110 GR St. Moritz
## 10    Wohnung   5.5    150 16500   110 GR St. Moritz
## 3     Wohnung   NA     530 31800    60 ZH   Zürich

```

4. Aim: The biggest apartments

```
top10 <- data_clean%>%top_n(n = 10, wt = W.Fläche)

top10_sorted <- top10[order(top10$W.Fläche, na.last=TRUE) , ]
top10_sorted
```

```
##      ID Jahr Quartal      W.Type Zimmer W.Fläche Miete Quad_m
## 5    7371 2019       4    Wohnung    7.0     350   5950     17
## 7   11356 2019       2    Wohnung    8.0     350   3900     11
## 6   9594 2019       2    Wohnung    7.5     353   2500      7
## 3   5455 2019       2    Wohnung    4.5     360   4390    12
## 4   5456 2019       4    Wohnung    4.5     360   3500    10
## 8  19073 2019       2    Wohnung    8.0     360   4700    13
## 11 50771 2019       2    Wohnung    6.0     365   7000    19
## 9   22052 2019       4    Wohnung    5.5     370   4500    12
## 1    654 2019        2    Wohnung    5.5     400   3600      9
## 2    655 2019        4 Loftwohnung  5.5     400   3600      9
## 10  42857 2019       2    Wohnung    NA     530  31800     60
##                               Adresse      Bezug   GKODE   GKODN KT
## 5          Breitingerstrasse 35 2019-10-01 2682569 1246256 ZH
## 7          Doktorgässli 19 2019-04-01 2579950 1193463 FR
## 6          Chemin de Naefels 13 2019-04-01 2553764 1215919 NE
## 3          Bergstrasse 38      <NA> 2595885 1227664 SO
## 4          Bergstrasse 38      <NA> 2595885 1227664 SO
## 8          Haslenstrasse 13 2019-04-01 2748543 1244789 AI
## 11         Via Canonico Ghiringhelli 18      <NA> 2721857 1116291 TI
## 9          Holzgasse 305      <NA> 2656221 1255088 AG
## 1          Albulastrasse 81      <NA> 2754563 1174909 GR
## 2          Albulastrasse 81      <NA> 2754563 1174909 GR
## 10         Scheideggstrasse 79      <NA> 2682462 1245216 ZH
##                               Kanton Hauptstädte G.Fläche Einwohner_2019 PLZ
## 5          Zürich      Zürich     1729     1539275 8002
## 7          Fribourg [Freiburg]  Fribourg    1671     321783 3212
## 6          Neuchâtel [Neuenburg] Neuchâtel    803     176496 2300
## 3          Solothurn    Solothurn    791     275247 2540
## 4          Solothurn    Solothurn    791     275247 2540
## 8          Appenzell Innerrhoden Appenzell    173     16128 9050
## 11         Ticino [Tessin]   Bellinzona   2812     351491 6500
## 9          Aargau        Aarau     1404     685845 5246
## 1          Graubünden    Chur      7105     199021 7411
## 2          Graubünden    Chur      7105     199021 7411
## 10         Zürich      Zürich     1729     1539275 8038
##                               Ort
## 5          Zürich
## 7          Gurmels
## 6          La Chaux-de-Fonds
## 3          Grenchen
## 4          Grenchen
## 8          Appenzell
## 11         Bellinzona
## 9          Scherz
## 1          Sils im Domleschg
```

```
## 2 Sils im Domleschg
## 10 Zürich
```

The biggest apartment available for rental is located in the canton of Graubünden and it has an area of 400 square meter. The 10th biggest Apartment has an area of 350 square meter and it is located in the canton of Zurich. This comparision is however independent of the price, as it varies due to the location of the apartment.

5. Rental price in relation to size of apartments, number of rooms, population & location

Graphical interactive visualization

```
fig2 <- plot_ly(data_clean, y = ~G.Fläche, x = ~Einwohner_2019, type = 'scatter', mode = 'markers', size = ~Zimmer)
      hoverinfo = 'text',
      text = ~paste('</br> Canton: ', Kanton,
                    '</br> Residents: ', Einwohner_2019,
                    '</br> Community area: ', G.Fläche))

fig2 <- fig2 %>% layout(title = 'Residents in 2019 against community area via canton',
                           xaxis = list(title = "Residents in 2019"),
                           yaxis = list(title = "Community area"))

fig2

tmpFile <- tempfile(fileext = ".png")
export(fig2, file = tmpFile)

## Warning: 'export' is deprecated.
## Use 'orca' instead.
## See help("Deprecated")

## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, please
## NULL
```

The above plot shows that the distribution of the population in relation to commune area is normal. Zurich with high population and small communal area and Graubünden with low population and high communal area could be seen as outliers.

Fitting Linear Model

```
data_scaled <- data_clean %>%
  mutate_at(c("W.Fläche", "Miete", "Einwohner_2019", "G.Fläche", "Zimmer"), ~(scale(.) %>% as.vector))

simple.fit <- lm(Miete~W.Fläche, data=data_scaled)
summary(simple.fit)
```

```

## 
## Call:
## lm(formula = Miete ~ W.Fläche, data = data_scaled)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -4.753 -0.377 -0.098  0.222 33.164 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 0.063144  0.003491 18.09   <2e-16 ***
## W.Fläche    0.677904  0.003492 194.16   <2e-16 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.769 on 48513 degrees of freedom
##   (8262 observations deleted due to missingness)
## Multiple R-squared:  0.4373, Adjusted R-squared:  0.4373 
## F-statistic: 3.77e+04 on 1 and 48513 DF,  p-value: < 2.2e-16

```

The above applied simple regression analysis shows that the rental price varies depending on the apartment area. The adjusted R-squared value is however only 0.43, meaning that only 43% of the data are modeled by the chosen model.

```

multi.fit = lm(Miete~W.Fläche + G.Fläche + Zimmer + Einwohner_2019, data=data_scaled)
summary(multi.fit)

```

```

## 
## Call:
## lm(formula = Miete ~ W.Fläche + G.Fläche + Zimmer + Einwohner_2019,
##      data = data_scaled)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -4.6406 -0.3572 -0.0918  0.2097 19.9448 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 0.057880  0.003282 17.635   <2e-16 ***
## W.Fläche    0.670255  0.005469 122.564   <2e-16 *** 
## G.Fläche    -0.094021  0.003451 -27.242   <2e-16 *** 
## Zimmer     -0.010488  0.005456  -1.922   0.0546 .  
## Einwohner_2019 0.220525  0.003365  65.542   <2e-16 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.7214 on 48388 degrees of freedom
##   (8384 observations deleted due to missingness)
## Multiple R-squared:  0.4866, Adjusted R-squared:  0.4866 
## F-statistic: 1.147e+04 on 4 and 48388 DF,  p-value: < 2.2e-16

```

The above multi-regression analysis shows that the price of an apartment is dependent on many factors such as apartment area, population & commune area, but not on the number of rooms. Here, the adjusted

R-squared value is better than within the regression analysis with multiple factors is 0.49, meaning that 49% of the data are modeled by the chosen model. To see whether this R-value is good or not, we do residual analysis.

Residual analysis

To check whether the chosen model represents the dataset well or not we apply residual analysis.

```
#Testing normal distribution and independence assumptions
jarqueberaTest(simple.fit$resid) #Test residuals for normality
```

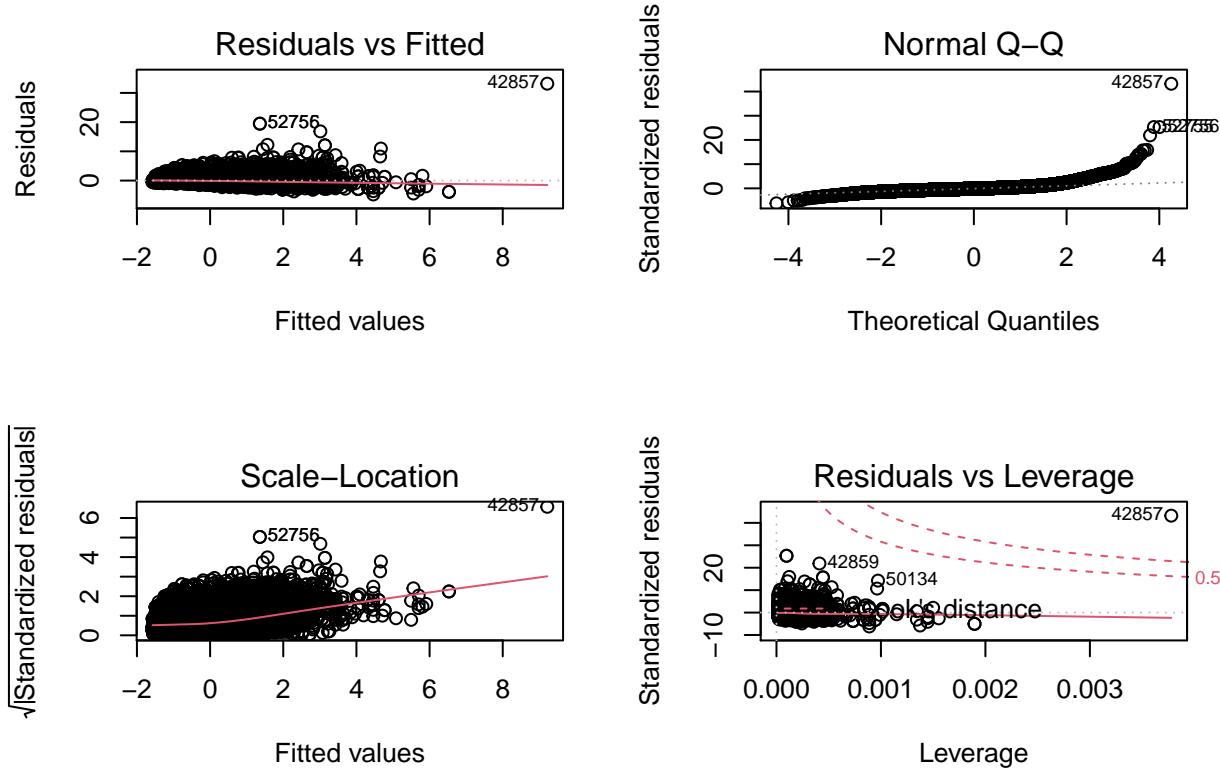
```
##
## Title:
## Jarque - Bera Normality Test
##
## Test Results:
##   STATISTIC:
##     X-squared: 25582190.1444
##   P VALUE:
##     Asymptotic p Value: < 2.2e-16
##
## Description:
##   Wed Feb 24 06:48:04 2021 by user: Fiseha Amine
```

```
dwttest(simple.fit) #Test for independence of residuals
```

```
##
## Durbin-Watson test
##
## data: simple.fit
## DW = 1.1172, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

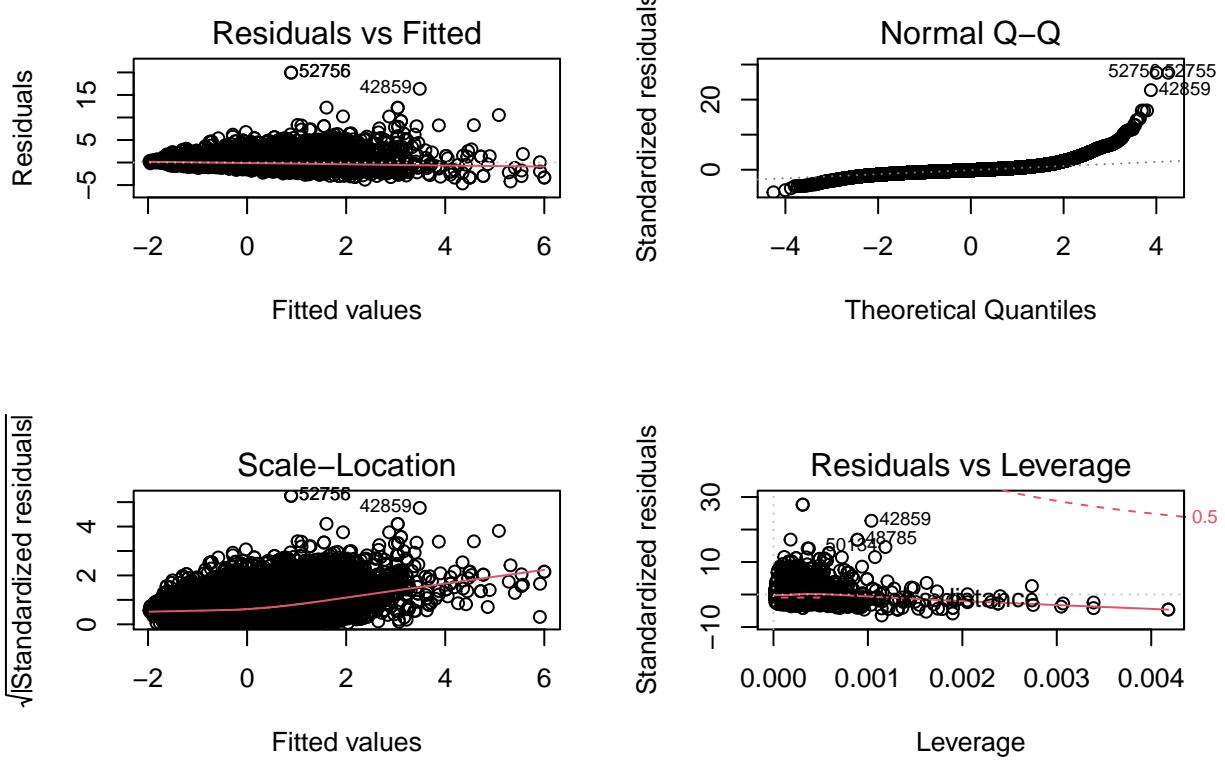
```
predicted <- predict(simple.fit)
residuals <- residuals(simple.fit)

par(mfrow = c(2, 2)) # Split the plotting panel into a 2 x 2 grid
plot(simple.fit)
```



The residuals analysis above shows that the plot residuals vs. fitted values can be presented in a linear model. Therefore we consider that the chosen “lm” model presents the dataset well. As normality is a precondition for such an analysis we consider also the Normal Q-Q plot, where the line shows a good fitting with the exception of some data at the end, which we can neglect in this scope. The scale-location plot tests the linear regression assumption of equal variance (homoscedasticity). Meaning that the data is spread equally above and below the line, which is mostly the case in our plot. The residuals vs leverage plot can be used to find influential cases in the dataset, this is however beyond the scope of the analysis.

```
multi.fit = lm(Miete~W.Fläche + G.Fläche + Zimmer + Einwohner_2019, data=data_scaled)
predicted <- predict(multi.fit)
residuals <- residuals(multi.fit)
par(mfrow = c(2, 2)) # Split the plotting panel into a 2 x 2 grid
plot(multi.fit)
```



The above residual analysis for the multi-regression analysis shows also that the residuals and the fitted values can be modeled with a linear model well. As described above, the Norma Q-Q plot shows that there are enough data not fitting the line at the beginning and at the end. precised analysis for normality might be needed, which however will not be done in this scope. The scale-location plot shows that the distribution of the data below and above the line somehow well presented is.

```
##Chapter of our choice
```

The “esquisse” package is an interactive “shiny” gadget to create “ggplot2” charts in a browser. Also the retrieved code from the new plot can get copied out and reused in the further R-code.

```
if (interactive()) {
# Launch with :
esquisser(data_clean)

#Launch esquisse in browser :
esquisser(data_clean, viewer = "browser")
}
```

As this visualization is done in an external browser, there will not be plots shown in this analytical part.