SLV Assignment1

Daniel Anadria Kyuri Park Ernst-Paul Swens Emilia Loescher

September 29, 2022

Frontpage with toc

1 Introduction

1.1 The Dataset

We explore a dataset from kaggle, which contains information about the housing market in Italy. The data were scraped from one of the most relevant housing sales websites in Italy during the month of *August 2022*. The data consist of more than 223,000 sales posts spread over 7,023 (89% of the total 7,904) Italian municipalities (*comuni*). Some of the entries were removed in dataset construction due to translation limitations (e.g., extended text-based description, specific url of the post).

In order to plot the statistics of interest to maps of Italy, we use the regional and provincial shape files which we obtained from the Italian National Institute of Statistics. These files contain the regional and provincial coding and geographical shape information which can be used cluster the comuni in our data (`location`) into (107) provinces and (20) regions.

For each sale, the dataset contains the following variables:

Table 1: Description of

Variable	**Description**
id	ID of the sale
timestamp	Timestamp consisting of 10 digits
location	Location on municipality level
title	Short description of property
price	Price in Euros
n_rooms	Number of rooms
floor	Floor
mq	Size in square meters
n_bathrooms	Number of bathrooms
year_of_construction	Year of construction
availability	Availability of property
energy_class	Energy class ranging from a+ to g
status	Status of the property
heating	Type of heating
has_garage	Garage present: yes (1), no (0)
has_terrace	Terrace present: yes (1), no (0)
has_garden	Garden present: yes (1), no (0)
has_balcony	Balcony present: yes (1), no (0)
has_fireplace	Fireplace present: yes (1), no (0)
has_alarm	Alarm present: yes (1), no (0)
has_air_conditioning	Air Conditioning present: yes (1), no (0)
has_pool	Pool present: yes (1), no (0)
has_parking	Parking present: yes (1), no (0)
has_elevator	Elevator present: yes (1), no (0)
is_furnished	Furniture present: yes (1), no (0)

1.2 Exploratory Questions (*EDIT LATER*)

We focus on four exploratory questions concerning housing data set in Italy.

Firstly, we explore if there are any geographical trends in the mean and median housing prices in Italy. (E.g., Is housing more/less expensive in Northern Italy compared to Southern Italy?)

Secondly, we examine if there are differences in mean regional variances of housing prices between the different provinces and regions.

Thirdly, we explore if there is a correlation between the missingness of housing price and other variables.

Fourthly, we identify the most important predictors of housing prices in Italy.

2 Preparation

In order to start our exploratory analysis, we first load relevant packages and import the full data set.

2.1 Load Packages & Import Data

```
## load packages
library(tidyverse) # for wrangling data
library(magrittr) # for using pipes
library(skimr) # for skimming data
library(sf) # for spatial analysis
library(sp) # for spatial analysis
library(ggplot2) # for plotting
library(fuzzyjoin) # for joining on not-exact matches
library(ggpubr) # for arranging ggplots


## import Italy housing data
housing <- read.csv("data/housing_data_italy_august2022.csv", na.strings=c("","NA"), header = TRUE)

## import Italy shape data
# municipality level
munic_2022 <- st_read("data/italy_shape_2022_files/Com01012022_g")[c("COD_REG","COD_PROV", "COMUNE")]
# province level
prov_2022 <- st_read("data/italy_shape_2022_files/ProvCM01012022_g")
# region level
reg_2022 <- st_read("data/italy_shape_2022_files/Reg01012022_g")
```

3 Preliminary analysis

We skim through our data using the `skimr` package. This summary of the raw data set can be found in the section 6 Appendix. The original data consist of 223,409 rows (sales) and 25 columns (variables). Given our questions, we conclude that `id` (ID of the sale), `timestamp` (timestamp of the sale), and `title` (description of the property) are irrelevant and, hence, we exclude them from the dataset for further analysis. In addition, we remove two columns that have only one unique value (`status`: “other” and `availibility`: “not free/other”), as these variables do not provide information specific to certain sales.

Furthermore, we observe that types of some variables are wrongly specified. We convert them to a correct type (e.g., `heating`: character –> factor, `has_xxx`: numeric –> factor, `is_furnished`: numeric –> factor).

We create a new variable `age` which contains the age of the property as of 2022 by subtracting the `year_of_construction` from 2022. In the original dataset, there are some unreasonable years of construction (e.g., 2209). Some properties may be sold before construction is completed, but we deem it unlikely for properties whose `year_of_construction` is more than 4 years later from now. Thus, we exclude the `age` of property lower than -4 (i.e., exclude `year_of_construction` > 2026).

The variable of our main interest `price` has some missing values, and we see that it is highly skewed to the right given that mean is far off to the right of the median (see section 6 Appendix). We look into the distribution of the `price` further in detail. As shown in Figure 1, there are extreme outliers in `price` that seem unrealistic (e.g., over 2 billion). We decide to filter out the houses whose price is over a million ***to prevent severe bias in our subsequent analyses … KP: NOT SURE IF THIS IS CORRECT TO SAY*** (see Figure 2 for the distribution of the price for the filtered houses).

```r
## cleaning up the housing data
cleaned_housing <- housing %>%
  # select variables that have more than one unique value
  select(where(~n_distinct(.) > 1)) %>%
  # fix the data type
  mutate(across(c(starts_with("has"), is_furnished, heating, energy_class, n_rooms, n_bathrooms, location)
  # filter out houses whose price is over a million (while keeping NAs)
  filter(price < 1e6 | is.na(price),
  # filter out houses whose construction year is more than 4 years later as of today
         year_of_construction < 2026 |is.na(year_of_construction)) %>%
  # create property age variable
  mutate(property_age = 2022 - as.numeric(year_of_construction)) %>%
            # remove id, timestamp, title and year_of_construction
  select(-c(id, timestamp, title, year_of_construction))
```

```r
## boxplot of price
housing %>%
ggplot(aes(x=price)) +
  geom_boxplot() + theme_minimal()
```

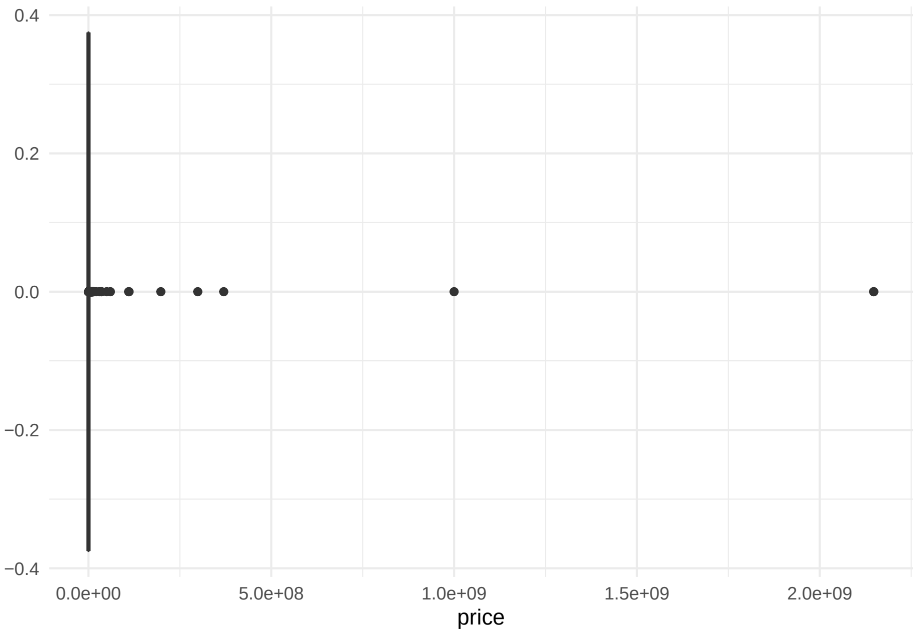


Figure 1: Distribution of housing price

```r
## density plot of price (under million)
housing %>%
  # filter out houses whose price is over a million
  filter(price < 1e6) %>%
  ggplot(aes(price)) +
  geom_histogram(aes(y=..density..), color=1, fill="white") +
  geom_density(lwd=0.5, color = 4, fill=4, alpha=0.2) + theme_minimal()
```

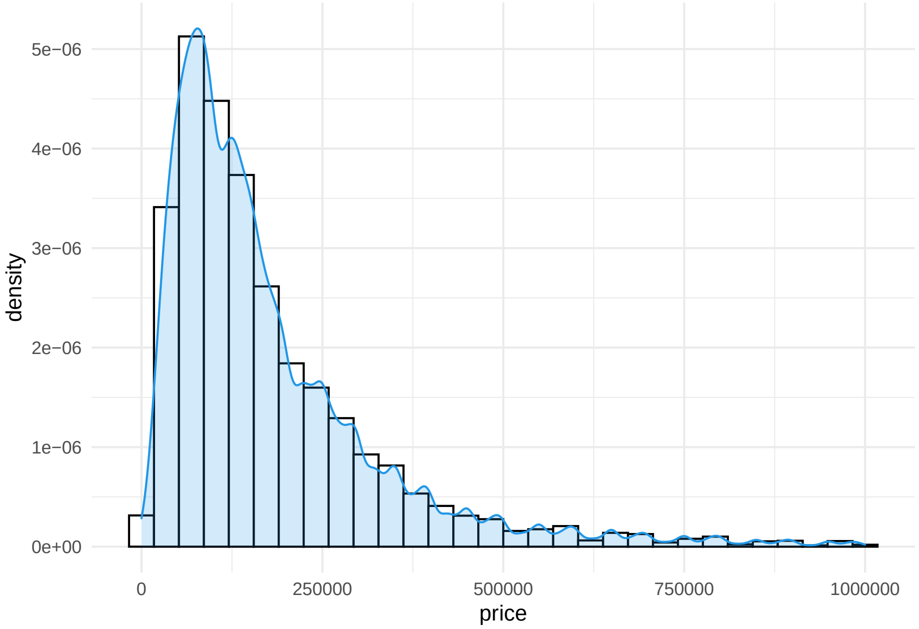


Figure 2: Histogram of housing price (under a million)

3.1 Data summary

After the cleaning up the data, we take a look at the summary statistics for the data set to get a better overview of our data.

```
# round up by 2 decimal places + disable scientific notation
options(digits = 2, scipen = 999)
# specify skimming function
my_skim <- skim_with(character = sfl(whitespace=NULL, min = NULL, max = NULL),
                     factor = sfl(ordered = NULL),
                     numeric = sfl(p0 = NULL, p25=NULL, p50=NULL, p75=NULL,
                                   p100=NULL, hist=NULL, median = ~median(., na.rm=T),
                                   min = ~min(., na.rm=T), max = ~max(., na.rm=T), n_unique=n_unique))
# summary table
my_skim(cleaned_housing)
```

Table 2: Data summary

Name	cleaned_housing
Number of rows	220607
Number of columns	20
Column type frequency:	
factor	16
numeric	4
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	n_unique	top_counts
location	0	1.00	7023	pis: 192, leg: 190, la : 188, bar: 187
n_rooms	58208	0.74	4	3: 56756, 4: 47126, 5: 30926, 2: 27591
n_bathrooms	12592	0.94	3	1: 107355, 2: 79819, 3: 20841
energy_class	638	1.00	12	g: 115161, f: 25382, e: 17111, a: 15924
heating	0	1.00	2	aut: 197714, oth: 22893
has_garage	0	1.00	2	0: 180548, 1: 40059
has_terrace	0	1.00	2	0: 195984, 1: 24623
has_garden	0	1.00	2	0: 184307, 1: 36300
has_balcony	0	1.00	2	0: 197918, 1: 22689
has_fireplace	0	1.00	2	0: 208689, 1: 11918
has_alarm	0	1.00	2	0: 218611, 1: 1996
has_air_conditioning	0	1.00	2	0: 154964, 1: 65643
has_pool	0	1.00	2	0: 216350, 1: 4257
has_parking	0	1.00	2	0: 217228, 1: 3379
has_elevator	0	1.00	2	0: 207928, 1: 12679
is_furnished	0	1.00	2	0: 203515, 1: 17092

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	median	min	max	n_unique
price	39113	0.82	177146.0	152154.7	130000	1	999999	2554
floor	71354	0.68	1.8	1.1	2	1	52	20
mq	3309	0.99	155.9	124.4	116	1	999	970
property_age	10	1.00	56.1	74.3	42	-3	1022	374

DECRIBE THE SUMMARY TABLE as per changes

From the tables, we can see that there are 12 different energy classes and 7023 different locations. When taking a closer look at the `location` variable, one can see that they are given on a municipality level.

For the numeric variables, we observe that several have lots of missing values (e.g., `price`, `n_rooms`, `floor`, `mq`, `n_bathrooms`). We will discuss how we want to deal with this in section 4.2.

4 Main (exploratory) Analysis

4.1 Question 1: Geographical Differences in the Mean and Median Housing Price

Each sale in our dataset is assigned to one of 7023 municipalities. In order to create plots which visualize the differences in average housing prices across Italy, we assign each municipality (*comune*) to its corresponding province (*provincia*) and region (*regione*). We use the data from the *Italian National Institute of Statistics* (*ISTAT*) to append the province and region information to every observed municipality in our dataset.

4.1.1 Preparation

At the beginning of the assignment, we loaded the *ISTAT* shape files. These files are useful for two reasons. First, they contain the list of all Italian municipalities, their respective provinces and regions. Therefore, we can use this data to append our original dataset with additional location indicators. Second, they contain the shapes of Italy divided into provinces and regions. This is particularly useful for creating map plots using `ggplot2`.

For completeness of our dataset, we append the province and region information. We use fuzzy matching for inexact matches as we found that there were some minor inconsistencies in how the municipalities were named in our dataset as opposed to their names in the ISTAT shape files. The result of the following chunk of code is that all the municipalities are assigned their regions and provinces.

```
cleaned_housing <- stringdist_left_join(cleaned_housing, munic_2022,by = c("location" = "COMUNE"), distanc
  group_by(location) %>% slice_min(distance) %>%
  select(-geometry,-distance) %>%
  left_join(., as.data.frame(reg_2022[,c("COD_REG","DEN_REG")])) %>%
  select(-geometry, -COMUNE) %>%
  left_join(., as.data.frame(prov_2022[,c("DEN_UTS", "COD_PROV")], by = "COD_PROV")) %>%
  select(-geometry, -COD_REG, -COD_PROV) %>%
  rename(., "region" = "DEN_REG", "province" = "DEN_UTS") %>%
  relocate(c(region, province), .after=location)
```

To answer the first exploratory question, we aggregate our data on two levels: 1) regional and 2) provincial level by computing two aggregate statistics: 1) the mean housing price and 2) the variance in housing price on the two respective levels. This yields two datasets, one per aggregation level. To each, we attach geometric information needed for plotting and convert it to an `sf` object which is a requirement for plotting maps.

```
price_by_reg <- cleaned_housing %>% group_by(region) %>%
  summarize(mean = mean(price, na.rm=T), median = median(price, na.rm=T), variance = var(price, na.rm=T),
```

```
price_by_prov <- cleaned_housing %>% group_by(province) %>%
  summarize(mean = mean(price, na.rm=T), median = median(price, na.rm=T), variance = var(price, na.rm=T),
```

4.1.2 Plot

As the distribution of `price` is still quite skewed, we inspect the median housing price and the corresponding variance by region.

```
p1 <- ggplot(price_by_reg) +
  geom_sf(aes(fill = median))+
  theme_minimal()

p2 <- ggplot(price_by_reg) +
  geom_sf(aes(fill = variance))+
  theme_minimal()

ggarrange(p1, p2,
          labels = c("median", "variance"),
          legend = "bottom")
```

Price difference between high- and low-median groups We can show facet_wrap…

```
top2_med <- price_by_reg %>% slice_max(median, n = 2) %>% pull(region)
bottom2_med <- price_by_reg %>% slice_min(median, n = 2) %>% pull(region)
groups <- list(top2_med = top2_med, bottom2_med = bottom2_med)

p3 <- cleaned_housing %>%
  filter(region %in% top2_med) %>%
  ggplot(aes(x = price, group = region)) +
  geom_histogram() +
  facet_wrap(~region)

p4 <- cleaned_housing %>%
  filter(region %in% bottom2_med) %>%
  ggplot(aes(x = price, group = region)) +
  geom_histogram() +
```

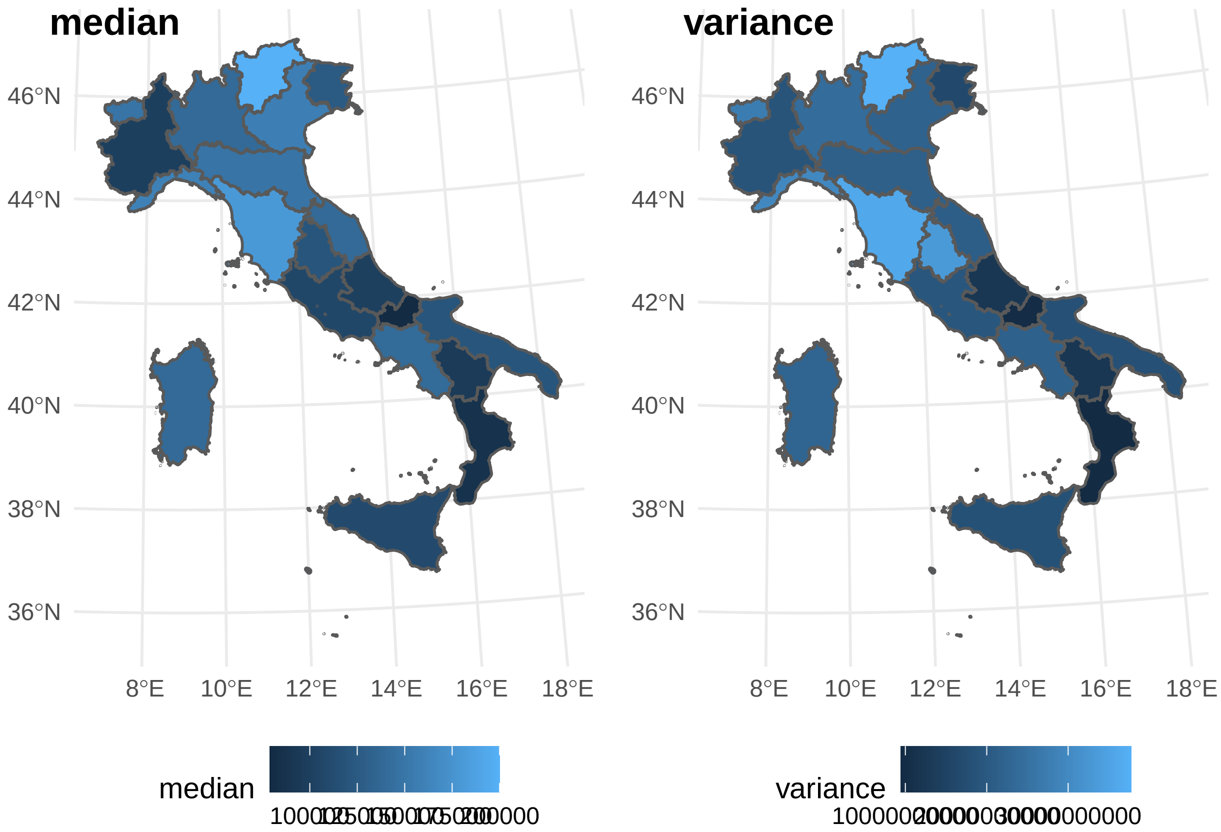
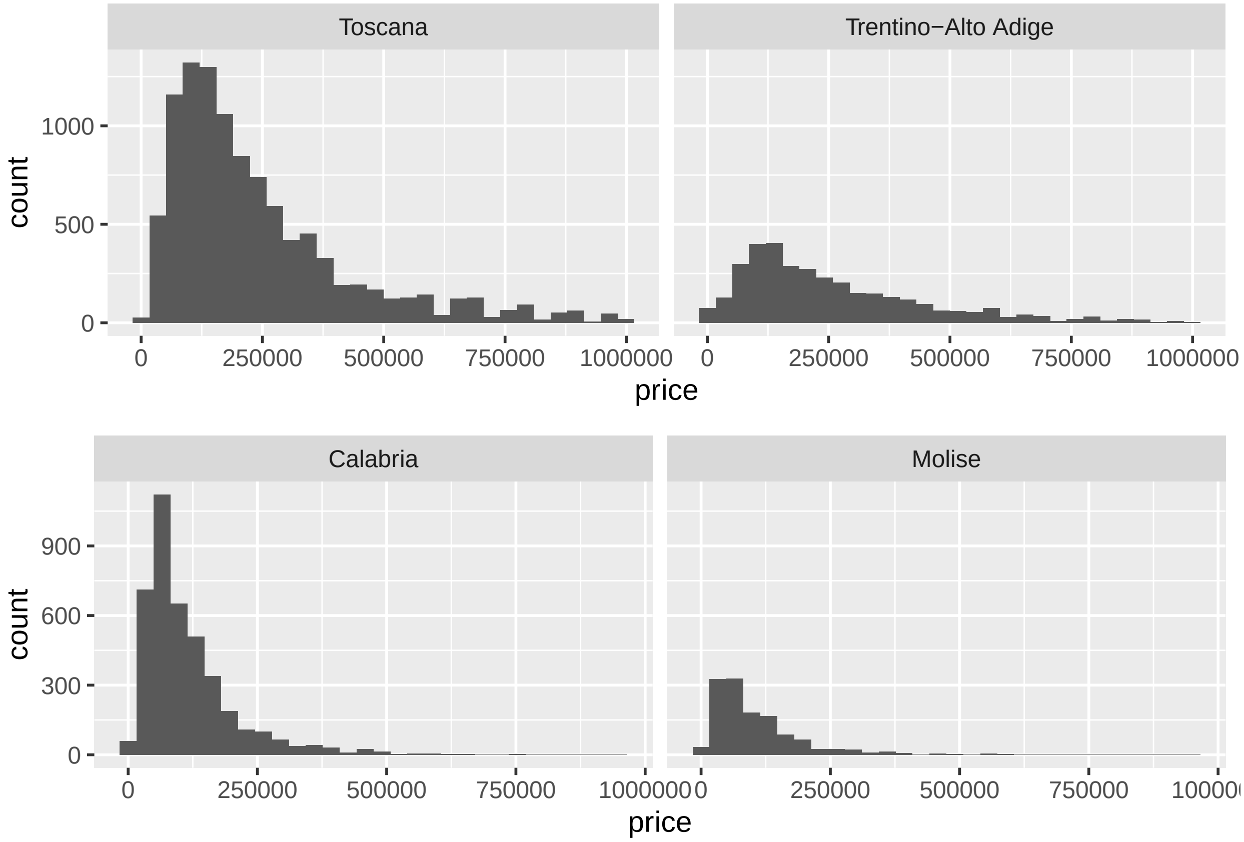
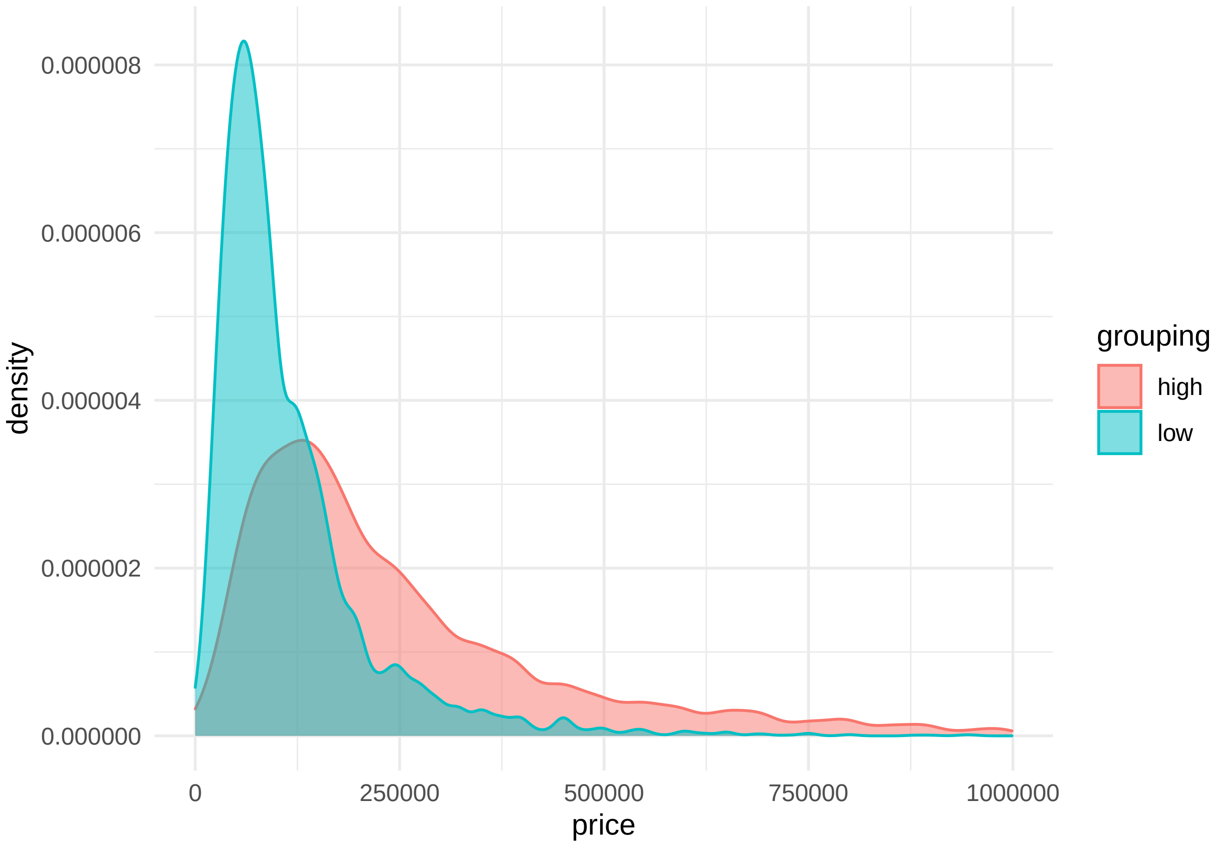


Figure 3: Median and Variance of price per region

```
  facet_wrap(~region)

ggarrange(p3, p4, nrow=2)
```



```
## OR WE CAN PUT ALL IN ONE PLOT SHOWING THE HIGH MEDIAN GROUP VS LOW MEDIAN GROUP
cleaned_housing %>%
  filter(region %in% c(top2_med, bottom2_med)) %>%
  mutate(grouping = case_when(region %in% top2_med ~ "high", TRUE ~ "low")) %>%
  ggplot(aes(x = price, color = grouping, fill = grouping)) +
  geom_density(alpha=0.5) + theme_minimal()
```



4.1.3 Conclusion

4.2 Question 3: Missingness and Imputation

4.2.1 Preparation

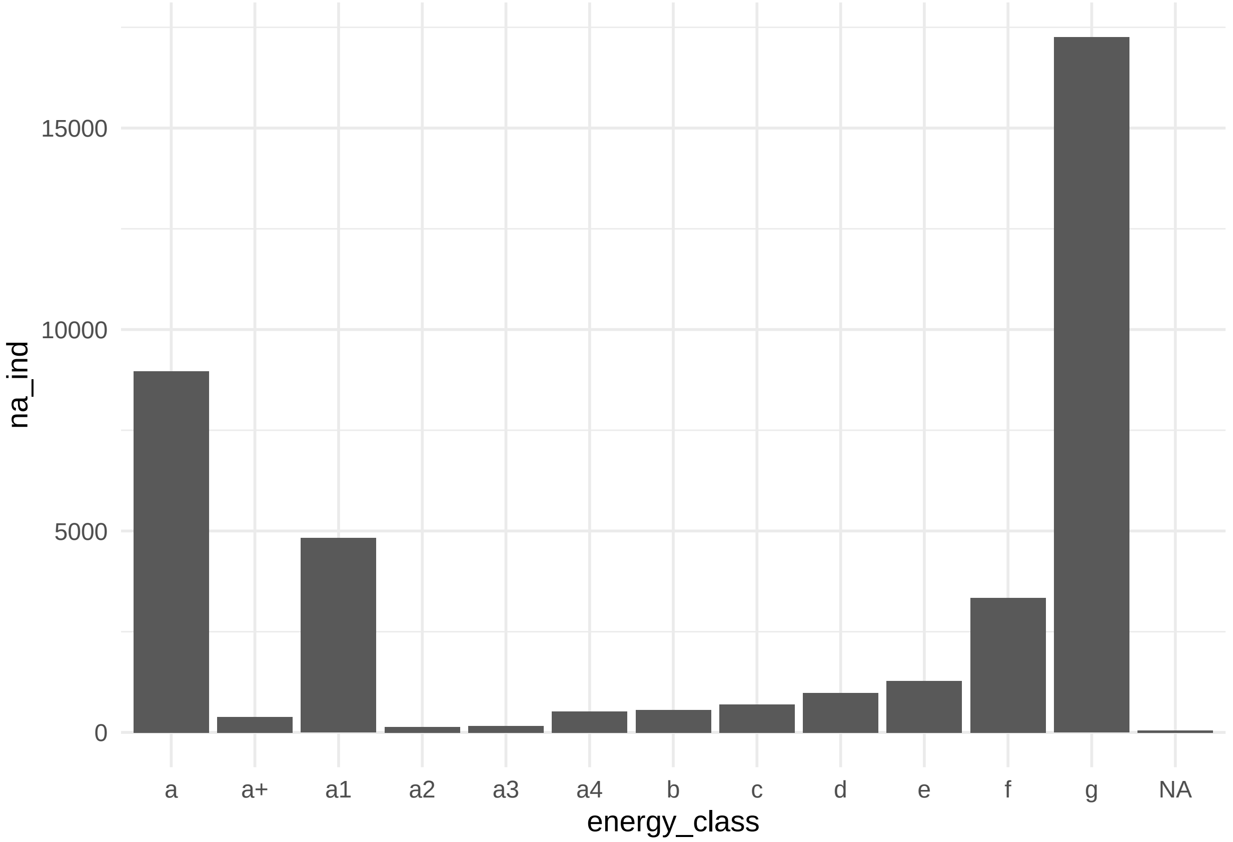
```
# create missingness indicator of price
cleaned_housing %>%
  mutate(na_ind = ifelse(is.na(price), 1, 0)) %>%
  ungroup() %>%
  summarize(across(n_rooms:property_age, ~cor(as.numeric(.x), na_ind, use = "pairwise.complete.obs"))) %>%
```

```
        is_furnished             has_pool                floor
              0.0052              -0.0097               0.0104
         n_bathrooms         property_age            has_alarm
              0.0172              -0.0198              -0.0321
                  mq          has_parking          has_balcony
              0.0344              -0.0383              -0.0556
             n_rooms         has_elevator          has_terrace
              0.0618              -0.0771              -0.0783
has_air_conditioning        has_fireplace           has_garage
             -0.0855              -0.0867              -0.0968
             heating           has_garden         energy_class
              0.1088              -0.1137              -0.2715
```

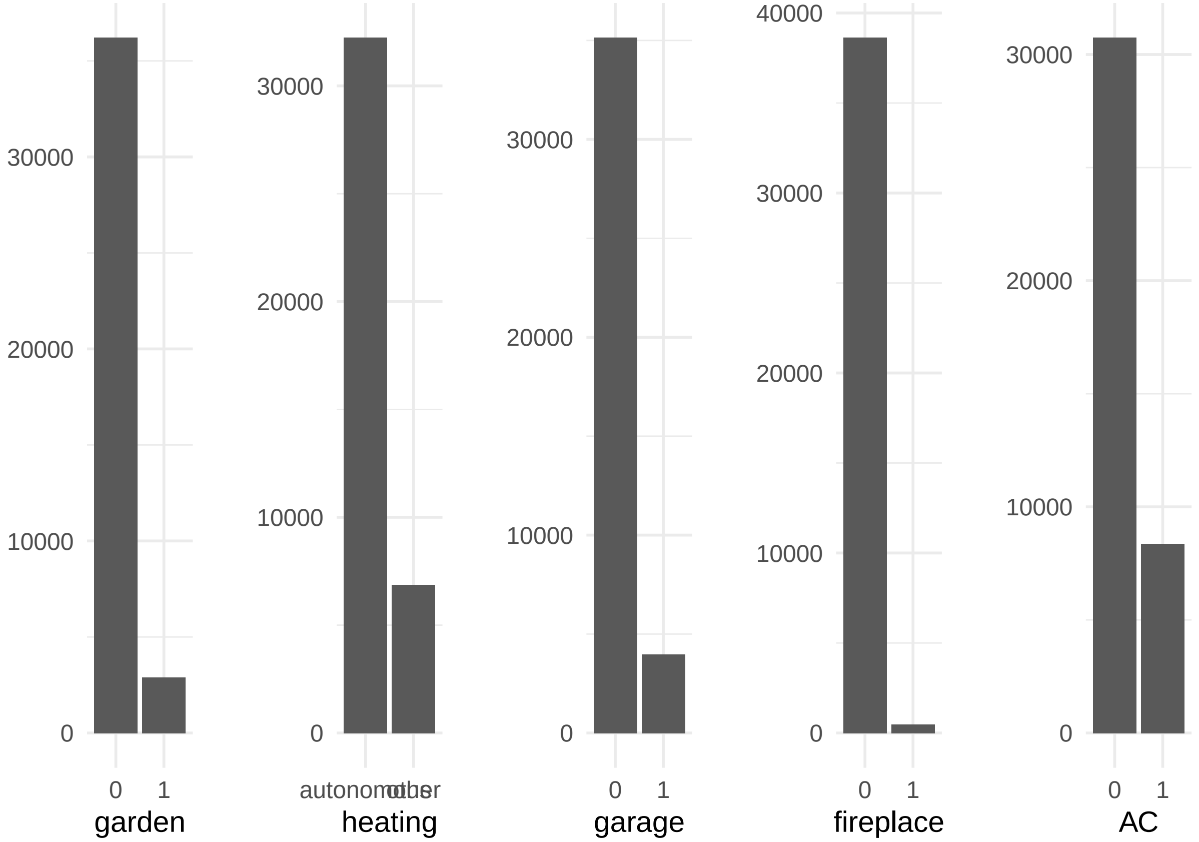
4.2.2 Plots

- energy_class(cor = 0.271): a lot of missing values in class a, a1, g .. what does that mean?
- has_garden (cor = -0.114) : lots of houses without a garden are missing.
- heating (cor = 0.109): lots of houses with autonomous heating are missing.
- has_garage (cor = -0.097) : lots of houses without a garage are missing.
- has_fireplace (cor = -0.087) : lots of houses without a fireplace are missing.
- has_air_conditioning (cor = -0.086) : lots of houses without AC are missing.

```
# add price missingness indicator
house_naind <- cleaned_housing %>%
  mutate(na_ind = ifelse(is.na(price), 1, 0))

## create plots for each of the correlated vars

# energy_class (cor = 0.27 the highest)
house_naind %>%
  ggplot(aes(x=energy_class, y = na_ind)) +
  geom_col() + theme_minimal()
```



```

plot_list <- list()

plot_list <- map2(
  c("has_garden", "heating", "has_garage", "has_fireplace", "has_air_conditioning"),
  c("garden", "heating", "garage", "fireplace", "AC"),
  function(var, lab) {
    ggplot(house_naind, aes(x = .data[[var]], y = na_ind)) +
      geom_col() +
      theme_minimal() +
      labs(x = lab, y = "")
     }
  )
ggarrange(plotlist = plot_list, nrow = 1, ncol = 5)
```



```
## check missingness w.r.t regions

# get the total missing prices per region
total_missing_region <- house_naind %>%
  group_by(region) %>%
  summarize(total_missing = sum(na_ind)) %>%
  left_join(.,reg_2022, by = c("region" = "DEN_REG")) %>% st_as_sf()

# plot the missing prices over the map
ggplot(total_missing_region) +
  geom_sf(fill=NA) +
  geom_point(
    aes(color = "coral1", size = total_missing, geometry = geometry),
    stat = "sf_coordinates") +
  theme_void() +
  theme(legend.position = "bottom") + guides(guides(colour=FALSE)) ## not sure how to change the color of
```

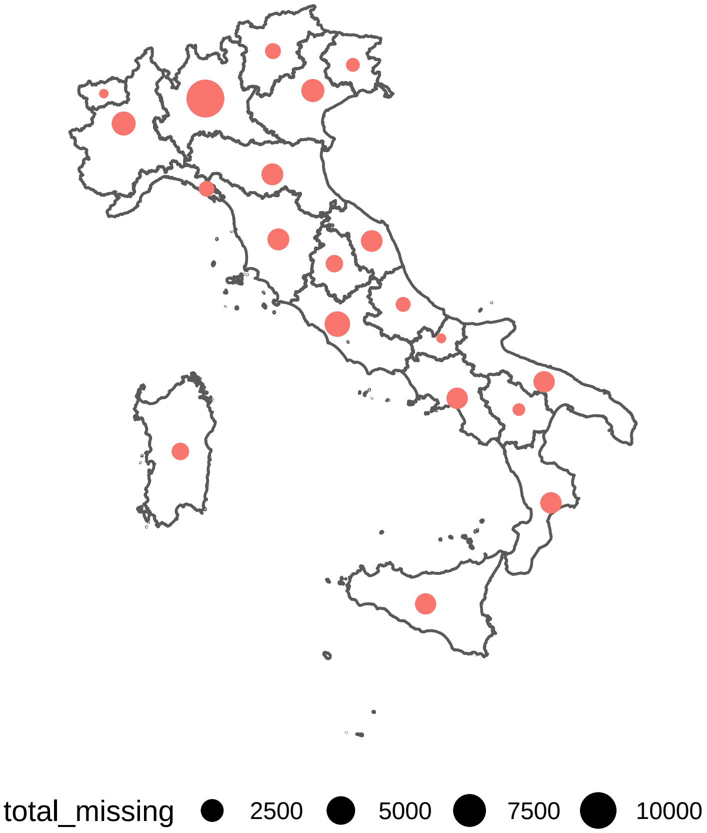


Figure 4: missingness of price per region

4.2.3 Conclusion

4.3 Question 4: Relevant Predictors for Housing Price

4.3.1 Preparation

4.3.2 Analysis

4.3.3 Conclusion

5 Overall Conclusion

6 Appendix

Summary of the raw data using the my_skim function.

```
my_skim(housing)
```

Table 5: Data summary

Name	housing
Number of rows	223409
Number of columns	25
Column type frequency:	
character	6
numeric	19
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	empty	n_unique
location	0	1	0	7023
title	0	1	0	199305
availability	0	1	0	1
energy_class	679	1	0	12
status	0	1	0	1
heating	0	1	0	2

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	median	min	max	n_unique
id	0	1.00	111705.00	64492.77	111705	1	223409	223409
timestamp	0	1.00	1661135705.37	12645.42	1661135577	1661114079	1661158618	42238
price	39116	0.82	239938.98	7562062.01	135000	1	2147483647	2852
n_rooms	60323	0.73	3.50	0.99	3	2	5	4
floor	72365	0.68	1.82	1.13	2	1	52	22
mq	4034	0.98	158.63	128.68	117	1	999	976
n_bathrooms	14397	0.94	1.59	0.67	1	1	3	3

skim_variable	n_missing	complete_rate	mean	sd	median	min	max	n_unique
year_of_construction	10	1.00	1965.13	76.75	1980	1000	2209	389
has_garage	0	1.00	0.18	0.38	0	0	1	2
has_terrace	0	1.00	0.11	0.32	0	0	1	2
has_garden	0	1.00	0.17	0.37	0	0	1	2
has_balcony	0	1.00	0.10	0.30	0	0	1	2
has_fireplace	0	1.00	0.05	0.23	0	0	1	2
has_alarm	0	1.00	0.01	0.10	0	0	1	2
has_air_conditioning	0	1.00	0.30	0.46	0	0	1	2
has_pool	0	1.00	0.02	0.15	0	0	1	2
has_parking	0	1.00	0.02	0.12	0	0	1	2
has_elevator	0	1.00	0.06	0.23	0	0	1	2
is_furnished	0	1.00	0.08	0.27	0	0	1	2