SLV Assignment1

Daniel Anadria Kyuri Park Ernst-Paul Swens Emilia Löscher

September 30, 2022

Frontpage with toc

1 Introduction

This is an exploratory data analysis of the Italian housing market in 2022. For context, Italy contains a total of 20 regions (*regioni*), 107 provinces (*province*) and 7,904 municipalities (*comuni*). In the present work, we pose several interesting research questions which can be answered by means of data visualization and predictive model building.

1.1 The Dataset

Our dataset originates from Kaggle. It contains information about the housing market in Italy in 2022. The data were scraped from one of the most prominent 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% coverage) Italian municipalities. Some of the entries were removed during dataset construction due to translation limitations (e.g., extended text-based description, specific url of the post). We do not have any information on the representativeness of our dataset. Hence, we advise caution when drawing inferences from our findings. Any inferences would require the analysis to be repeated with a representative dataset.

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* ISTAT. These files contain the regional and provincial coding and geographical shape information which can be used cluster the municipalities in our `location` variable into their respective provinces and 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)

2 Preparation

In order to start our exploratory analysis, we first load relevant packages and import the dataset as well as ISTAT shape files.

2.1 Load Packages & Import Data

```r
# 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 ISTAT shape files
## municipality
muni_2022 <- st_read("data/italy_shape_2022_files/Com01012022_g")[c("COD_REG","COD_PROV", "COMUNE")]
## province
prov_2022 <- st_read("data/italy_shape_2022_files/ProvCM01012022_g")
## region
reg_2022 <- st_read("data/italy_shape_2022_files/Reg01012022_g")
```

3 Data Cleaning

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 research questions, we exclude: `id` (*ID of the sale*), `timestamp` (*timestamp of the sale*), and `title` (*description of the property*). In addition, we exclude two columns that have only one unique value (`status`: "other" and `availibility`: "not free/other"), as these are not variables but constants.

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).

Next we create a new variable `age` (*age of the property*) by subtracting the `year_of_construction` from 2022. In the original dataset, there are some unreasonable years of construction (e.g., 2209). While some properties may be sold before their construction is completed, we deem it unlikely for properties whose `year_of_construction` is more than 4 years removed from 2022. Thus, we exclude the `age` of property lower than -4 (i.e., `year_of_construction` $>$ 2026).

The variable of our main interest `price` is highly skewed to the right given that mean (239,939) is far off to the right of the median (135,000) (see section 6 Appendix). We look into the distribution of the `price` further in detail. We create a boxplot to examine the outliers (see *Figure 1*).

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

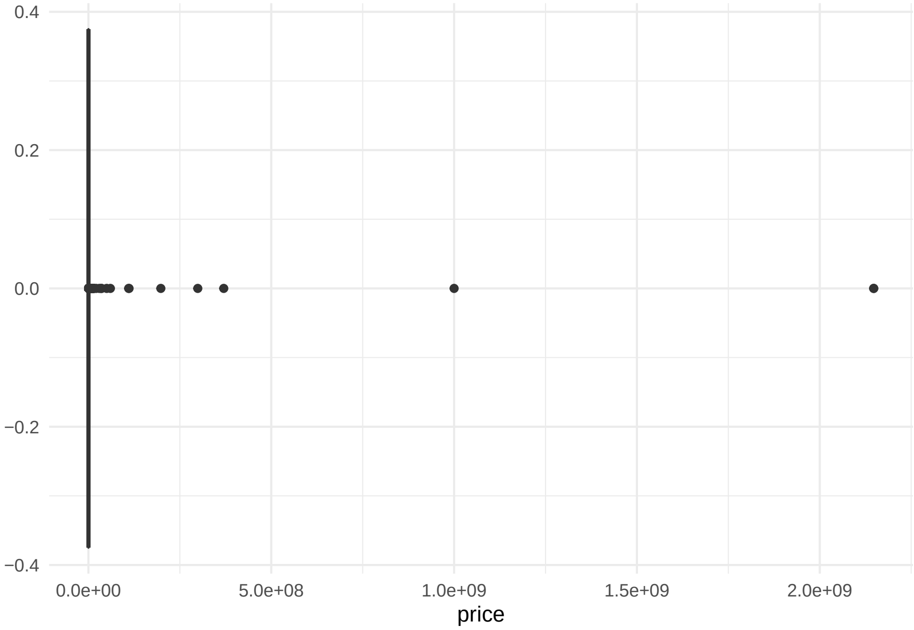


Figure 1: Boxplot of Housing Prices in Italy

From *Figure 1*, we observe that there are extreme outliers in `price`. Some housing prices in the dataset are exorbitant (e.g., over €2B). We decide to focus the scope of our analysis on the houses whose price is less than or equal to €1M (using the `filter` function), which are more likely to be affordable to an average Italian. The distribution of housing prices after filtering can be seen in *Figure 2*.

```
# 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))
```

```
# density plot of price (cleaned dataset)
cleaned_housing %>%
  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()
```

From *Figure 2*, we take that the distribution of housing prices after filtering appears a lot more ordinary. There is still a long right tail, but that is to be expected with housing prices in any country. The extreme outliers have been

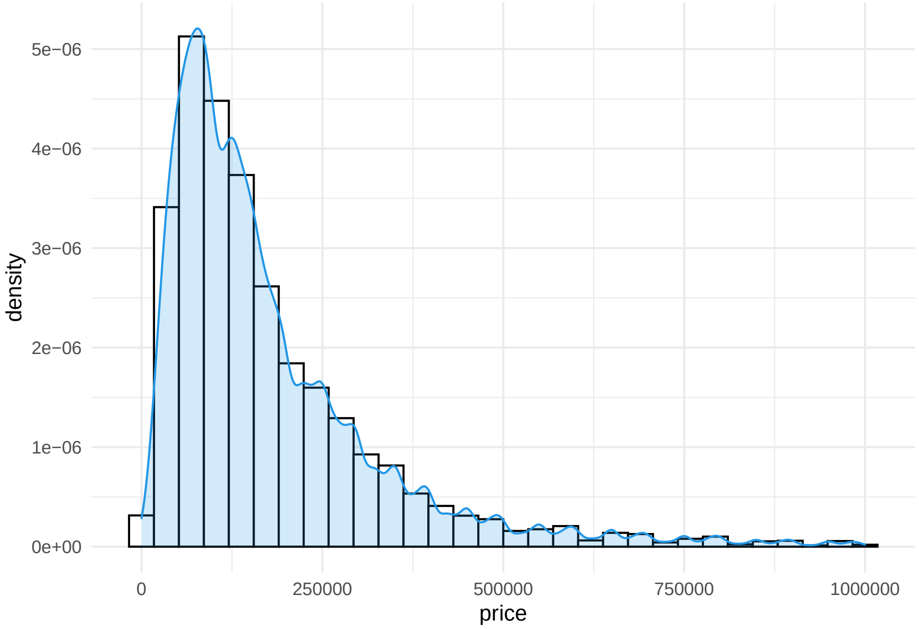


Figure 2: Histogram and Density Plot of Housing Price After Filtering

eliminated. From this plot, we also conclude that when working with housing price data, it is more informative to use centrality and spread measures which are robust to skewed data. For this reason, we will use the median and median absolute deviation instead of the mean and variance in our exploration of the present dataset.

3.1 Data Summary

After data cleaning, we take a look at the summary statistics 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

From the output, we see that our cleaned dataset has 220,607 rows and 20 columns, 16 of which are factors, and 4 of which are numeric types.

The output is separated into two tables: one for factors, and one for numeric variables.

For every factor, the table provides insights into: the number of missing values (`n_missing`), the proportion of the variable that is not missing (`complete_rate`), the number of unique observations (`n_unique`), and the most frequent levels (`top_counts`).

For the numeric variables, the table shows: the number of missing values (`n_missing`), the proportion of the variable that is not missing (`complete_rate`), the mean (`mean`), standard deviation (`sd`), the median (`median`), minimum (`min`), maximum (`max`) and the number of unique observations (`n_unique`). Importantly, a significant proportion (17.5%) of `price`, our main outcome of interest is missing. We discuss how we deal with this in the section **??**.

Having transformed the data and examined its variables, we start the exploratory analysis.

4 Exploratory Data Analysis

A good way to start an exploratory data analysis is to identify exploratory questions which will be answered.

We focus on four questions concerning the housing prices in Italy:

1. Are there any regional trends in the median housing prices and their absolute deviations in Italy?

2. Are there any provincial trends in the median housing prices and their absolute deviations in Italy?

3. Is there a correlation between the missingness of housing price and other variables?

4. What are the most important predictors of housing prices in Italy?

To answer our research questions, we first have to prepare our dataset for geographical plotting.

4.1 Data Preparation for Geographical Plotting

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`.

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 to its corresponding province and region. We use the data from *ISTAT* to append the province and region information to every observed municipality in our dataset. 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, muni_2022,by = c("location" = "COMUNE"), distance
  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 two exploratory question, we aggregate our data on two levels: 1) regional and 2) provincial level by computing two aggregate statistics: 1) the median housing price and 2) the median absolute deviation in housing price on the two respective levels. This yields two datasets, one per aggregation level. To each, we attach geometric information needed for geographic plotting and convert it to an `sf` object which is a requirement for plotting maps.

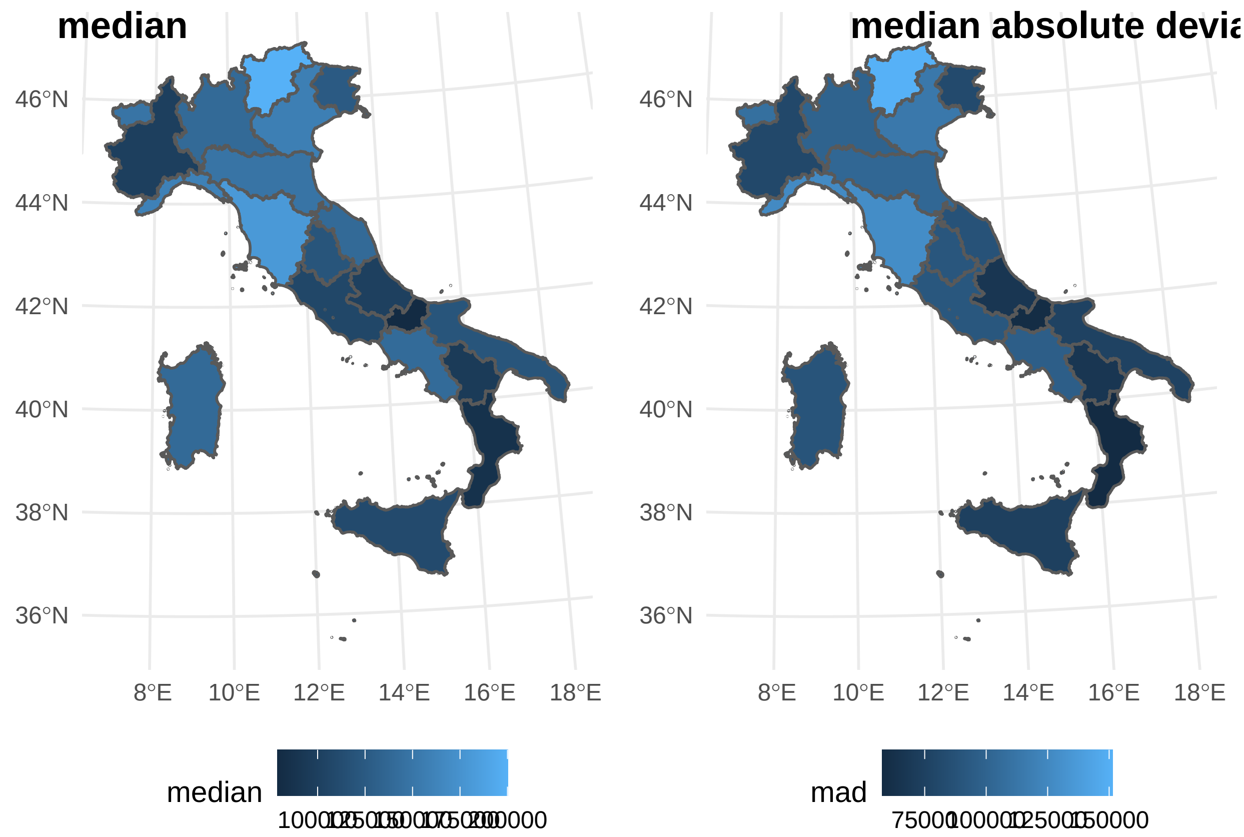
```
price_by_reg <- cleaned_housing %>% group_by(region) %>%
  summarize(median = median(price, na.rm=T), mad = mad(price,na.rm=T)) %>% left_join(.,reg_2022, by = c("r
```

```
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), 
```

Having done this, we are ready to start answering our exploratory questions.

4.2 Question 1: Regional Trends in the Median Housing Price and Absolute Deviations in Italy

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

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

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



@Kyuri: reverse the color scheme, see if blue is color-blind friendly, fix the values on the legend @Emilia: interpret the plots above, name the regions with highest and lowest values for the median and mad

Interpretation goes here.

4.3 Question 2: Provincial Trends in the Median Absolute Deviation of Housing Prices

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

p2 <- ggplot(price_by_prov) +
  geom_sf(aes(fill = mad))+
  theme_minimal()

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

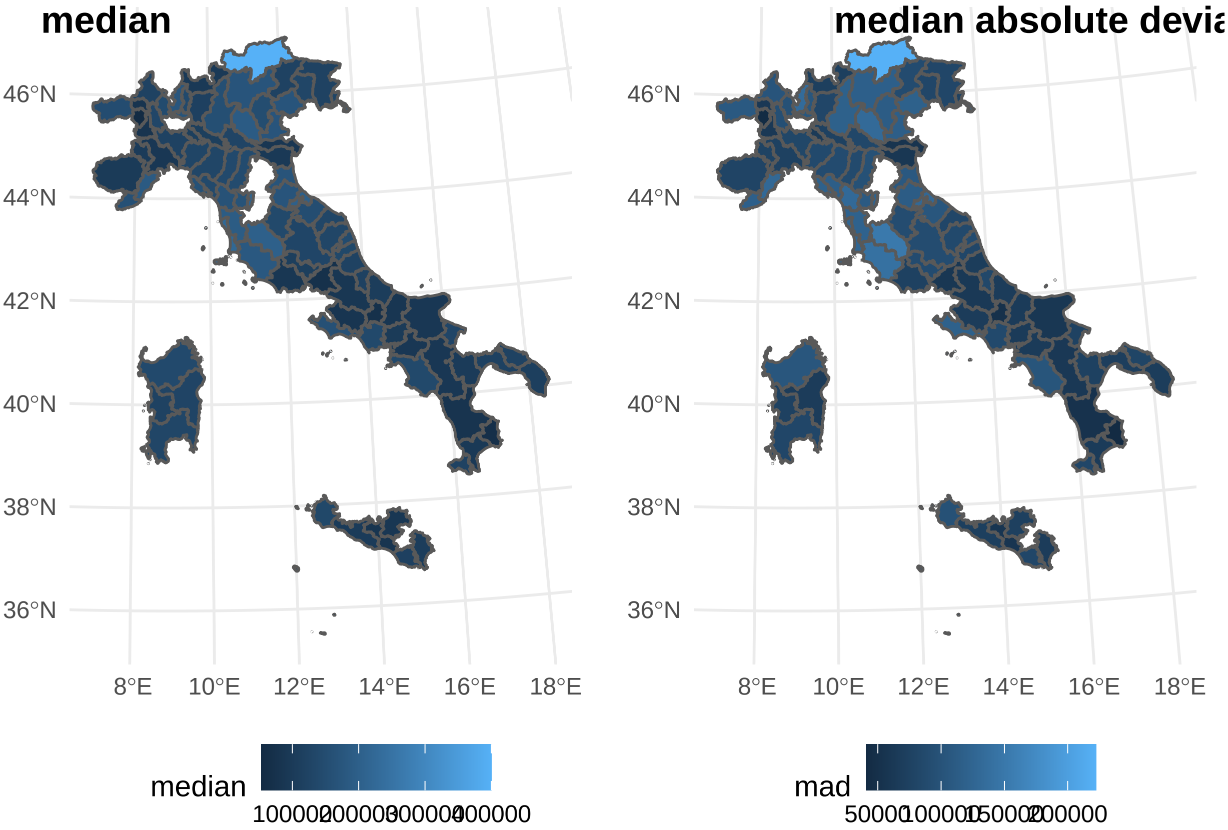


Figure 3: Median and MAD of Price per Province

Interpretation goes here.

4.4 Question 3: Correlation Between Missingness of Price and Other Variables

For this exploratory research, we are interested which variables can predict house prices. However, from an initial inspection, it became clear that some potential predictors contain missing values. Additionally, the outcome itself includes missing values.

Therefore, we are interesting to explore if the missingness of the predictors and the outcome variable: house prices are related. First, we investigated the correlation between the missingness indicator for price and the predictors. Second, we visually examined the relationship between the predictors' missingness indicators and the price densities. Lastly, we visually inspected the percentage of missingness per region. These results could provide us some indication of the missing data mechanism. Specifically, if missing completely at random (MCAR) is plausible to assume or not.

```
# barplot to show the missingness per variable
cleaned_housing %>%
   is.na() %>% colMeans() %>% stack() %>%
   ggplot(aes(x = reorder(ind, values), y=values)) +
   geom_bar(stat="identity", width = .2, fill = "black") +
   geom_point() +
   theme_minimal() +
   coord_flip() +
   labs(x = "Variable", y = "Missingness (%)")
```

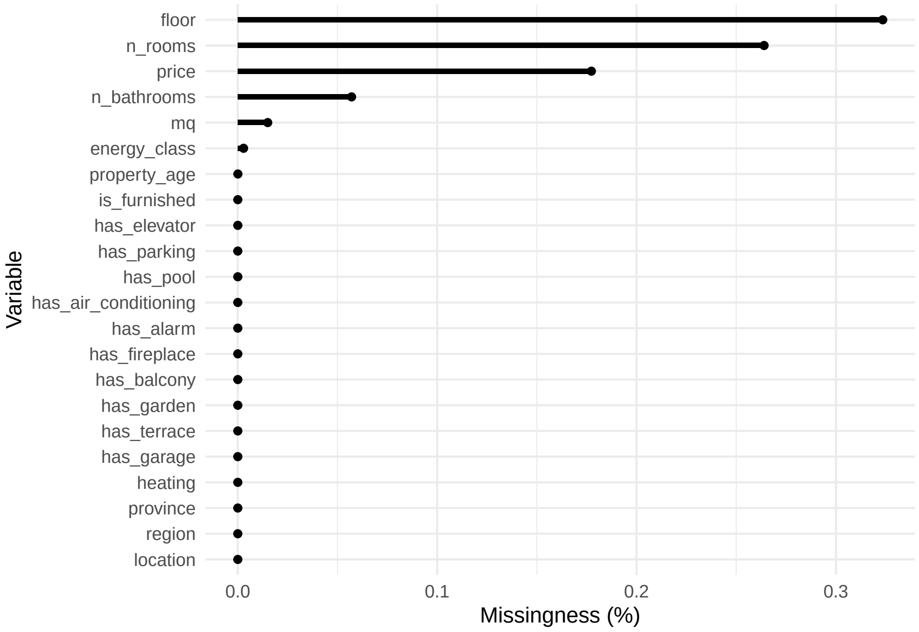


Figure 4: Distribution of housing price

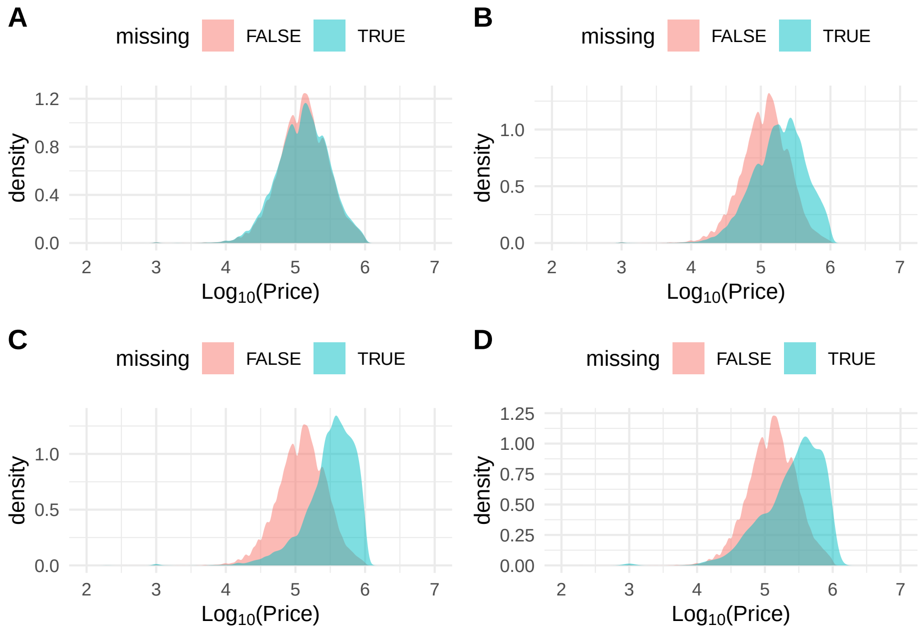
4.4.1 Results

```
# missingness correlation between price and predictors
cleaned_housing %>%
   mutate(missing = ifelse(is.na(price), 1, 0)) %>%
   ungroup() %>%
   summarize(across(n_rooms:property_age, ~cor(as.numeric(.x), missing,
      use = "pairwise.complete.obs"))) %>% t(.) %>% .[order(abs(.)),] %>%
   round(2)
```

```
        is_furnished             has_pool                floor 
                0.01                -0.01                 0.01 
         n_bathrooms         property_age            has_alarm 
                0.02                -0.02                -0.03 
                  mq          has_parking          has_balcony 
                0.03                -0.04                -0.06 
             n_rooms         has_elevator          has_terrace 
                0.06                -0.08                -0.08 
has_air_conditioning        has_fireplace           has_garage 
               -0.09                -0.09                -0.10 
             heating           has_garden         energy_class 
                0.11                -0.11                -0.27 
```

The missingness of price appears to be moderately correlation with the energy class. Further, the missingness of price seems to be weakly correlated to having a garden and heating, and the type of home heating system.

```
# function to create a density plot of price with missing indicator
missing_plots <- function(x) {
  plot <- cleaned_housing %>% ungroup() %>%
    mutate(missing = is.na(.[,x])) %>%
    ggplot(aes(x = log10(price), fill = missing)) +
    geom_density(alpha = 0.5, color = NA) +
    theme_minimal() +
    theme(legend.position = "top") +
    labs(x = expression(paste(Log["10"],"(Price)"), y = "Density")) +
    scale_x_continuous(limits = c(2, 7))
  return(plot)
}

# multiple plots for floor, no. of rooms, no. of bathrooms, and meters squared
ggarrange(missing_plots("floor"),
          missing_plots("n_rooms"),
          missing_plots("n_bathrooms"),
          missing_plots("mq"),
          labels = c("A", "B", "C", "D"),
          ncol = 2, nrow = 2)
```



The missingness of number of floors (@ref(fig:missingness_density) A) and number of rooms (@ref(fig:missingness_density) B) does not seem to be dependent on the observed price information. Whereas for the number of bathrooms (@ref(fig:missingness_density) C) and the meters squared (@ref(fig:missingness_density) D) missingness shows a different result, namely missingness tends to occur at higher house prices.

```
## check missingness w.r.t regions
cleaned_housing %>%
   mutate(na_ind = ifelse(is.na(price), 1, 0)) %>%
   group_by(region) %>%
   summarize(total_missing = sum(na_ind) / n()) %>%
   left_join(.,reg_2022, by = c("region" = "DEN_REG")) %>%
   st_as_sf() %>%
   ggplot() +
      geom_sf(fill=NA) +
      geom_point(color = "coral1",
      aes(size = total_missing, geometry = geometry),
      stat = "sf_coordinates") +
      theme_void() +
      theme(legend.position = "bottom")
```

Finally, figure @ref(fig:missing_reg) shows that the proportion of missing data is not equal for each region.

4.4.2 Conclusion

We explored if the missingness of the predictors and the outcome variable: house prices are related. From our results it appears that it does. Hence, it is unlikely that the missingness mechanism is MCAR. Therefore, we include an multiple imputation procedure before the creation of the prediction model in the next section.

4.5 Question 4: The Most Important Predictors of Housing Price in Italy

As mentioned in the previous section, we will handle the missing data using a multiple imputation procedure. Subsequently, we are interested to find out which variables can predict house prices. For this purpose, we used two step-wise modeling strategies. The first strategy consists of a backward selection method based on the pooled

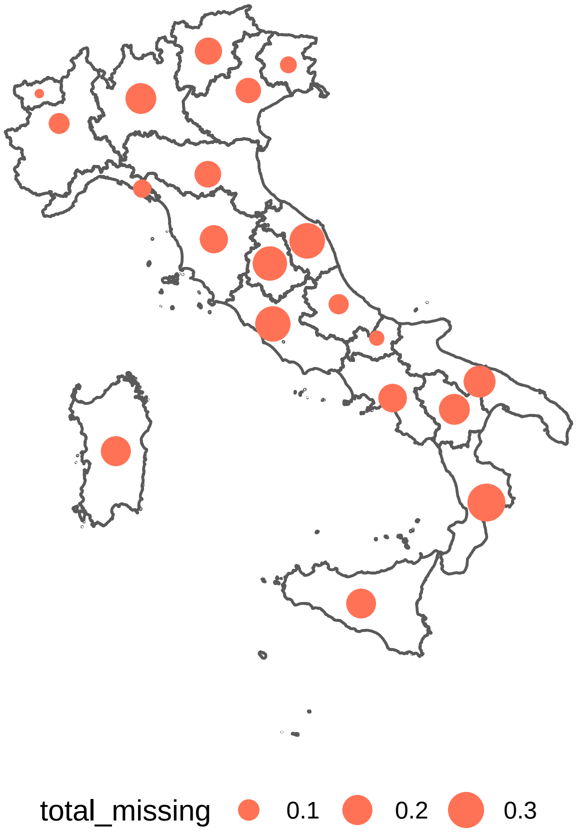


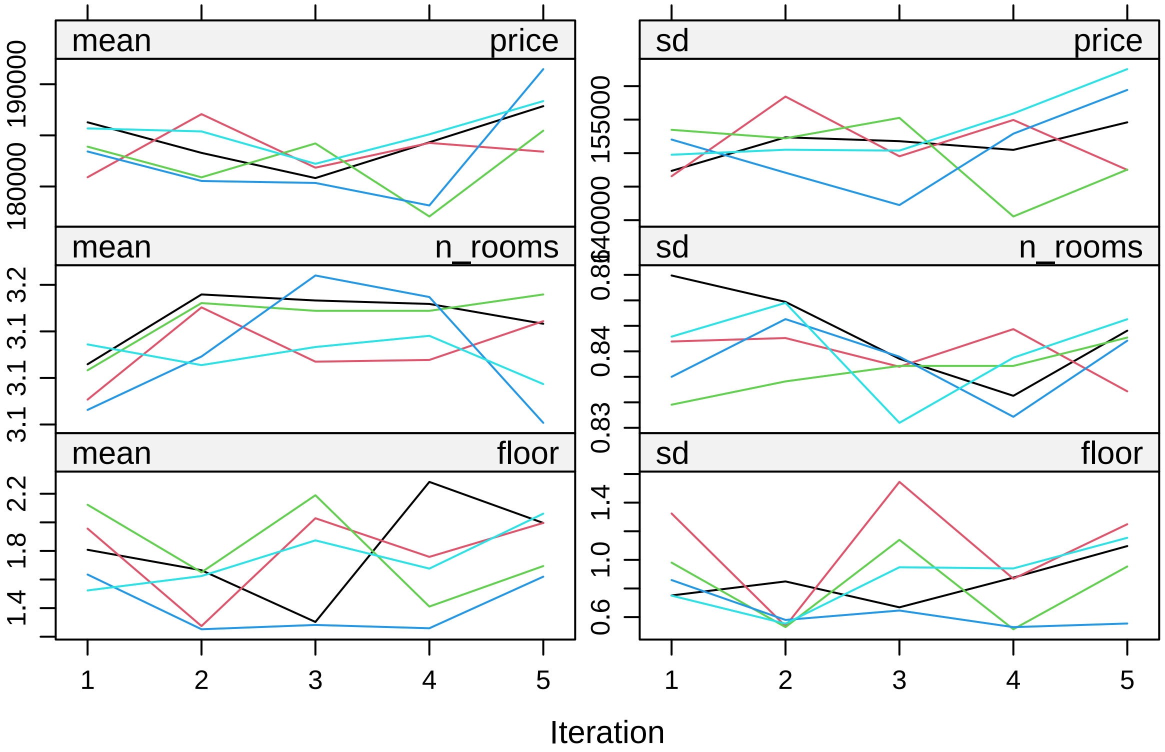
Figure 5: Missingness of Price per Region

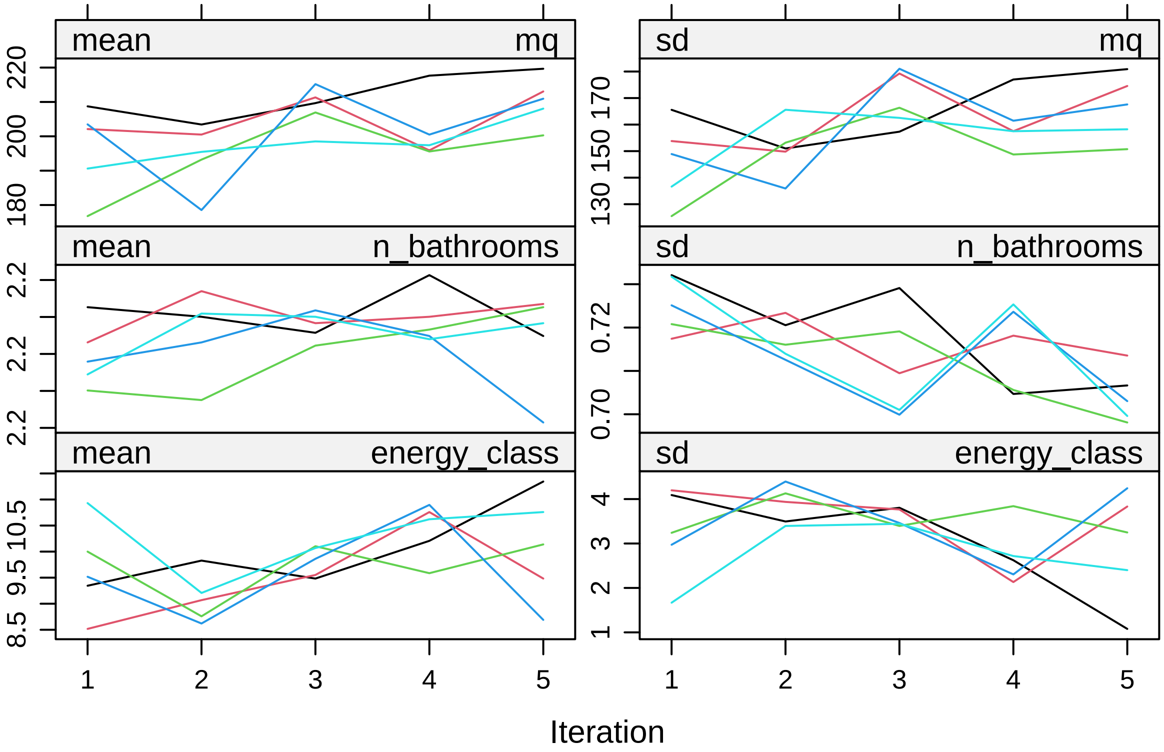
estimates. The second strategy is a forward selection method applied to each imputed data set ($m = 5$) and variables are selected in the final that appear the majority of the models. The models will be compared based on their included predictors, pooled coefficient of determination, and Bayesian information criterion (BIC).

4.5.1 Preparation

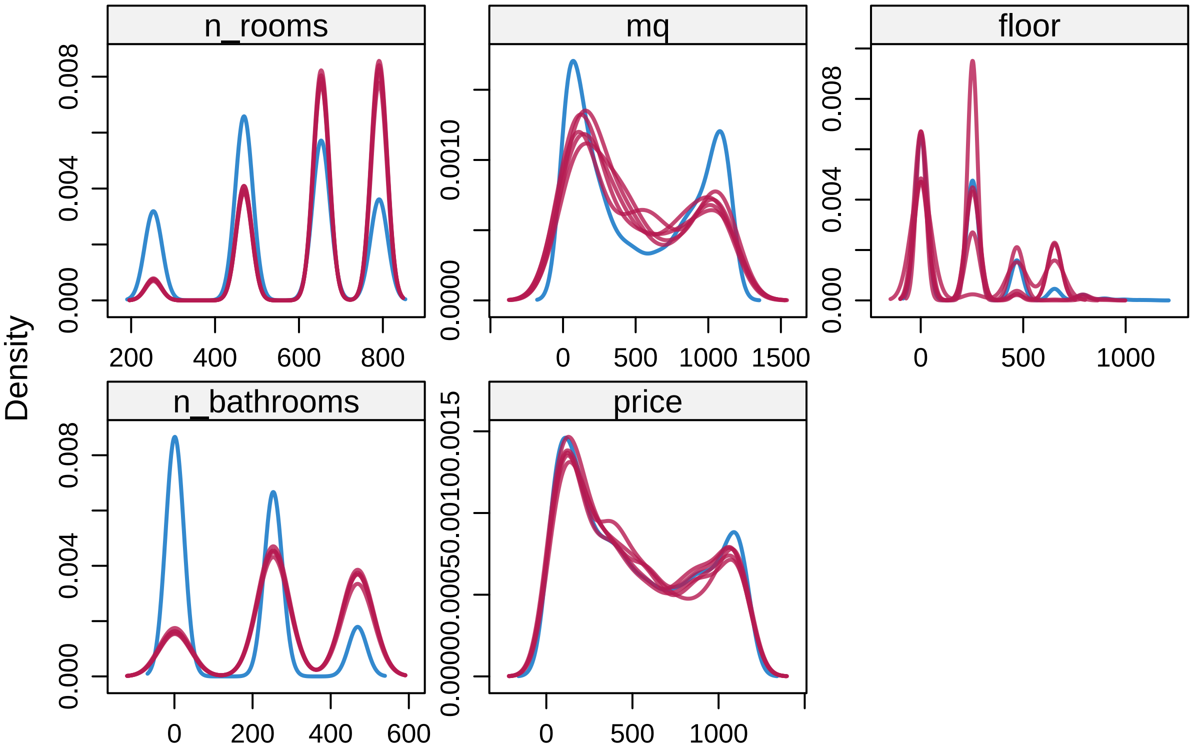
```r
# fix seed for reproducibility
set.seed(1)

# reduced data set
housing_sampled <- cleaned_housing %>%
   # ungroup on location
   ungroup() %>%
   # sample 10000 records
   sample_n(10000) %>%
   # ignore location, and province information
   select(-location, -province)
```

```r
# convergence of the algorithm
plot(imp)
```





```
# plausibility of the imputed data
densityplot(imp, ~n_rooms + mq + floor + n_bathrooms + price, lwd = 2)
```



To reduce the computation time of the imputation model, we sampled 10000 records from the initial data set. Further, for the scope of this exploratory research, we limited ourselves to simple random sampling and disregarding location and province as predictors. Furthermore, we limited the diagnostics for the imputation procedure to the convergence of the algorithm and plausibility of the imputed data. There appeared to be no convergence issues and the imputed data appeared to be plausible with respect to the observed data.

4.5.2 Analysis

The model based on backward selection method based on the pooled estimates contains 10 predictors: region, number of rooms, squared meters, number of bathrooms, energy class, type of home heating system, and the home including a terrace, alarm, pool, or elevator. The predictors in the model explain 40 percent of the variability observed in the house price.

```r
# define the scope of which predictors
scope <- list(upper = ~ region + n_rooms + floor + mq + n_bathrooms +
   energy_class + heating + has_garage + has_terrace + has_garden + has_balcony +
   has_fireplace + has_alarm + has_air_conditioning + has_pool + has_parking +
   has_elevator + is_furnished + property_age,
              lower = ~ 1)

# apply a forward selection method to each imputed data set
expr <- expression(f1 <- lm(price ~ 1),
                   f2 <- step(f1, scope = scope))
fit <- with(imp, expr)

# majority vote which predictors to include
formulas <- lapply(fit$analyses, formula)
terms <- lapply(formulas, terms)
```

```r
votes <- unlist(lapply(terms, labels))
table(votes)

# model based on forward selection method applied to each imputed data set
model_2 <- with(imp, lm(price ~ energy_class + floor + has_alarm + has_elevator +
   has_garden + has_pool + has_terrace + heating + mq + n_bathrooms +
   n_rooms + property_age + region))

# pooled effect estimates
summary(pool(model_2))

# pooled r-squared estimate
pool.r.squared(model_2, adjusted = TRUE)
```

The model based on backward selection method based on the pooled estimates contains 13 predictors: region, number of rooms, number of floor, squared meters, number of bathrooms, energy class, type of home heating system, property age and the home including a elevator, alarm, garden, pool, or terrace. The predictors in the model explain 40 percent of the variability observed in the house price. The code for the second model is reused from Gerko Vinks' mice vignettes, which are publicly available.

```r
# derive the BIC value for each model
model_1_bic <- model_1$analyses %>% sapply(BIC)
model_2_bic <- model_2$analyses %>% sapply(BIC)

# compare the BIC value
sum(model_1_bic < model_2_bic)
```

```
[1] 5
```

```r
model_2_bic - model_1_bic
```

```
[1] 17 20 10 13 16
```

To compare the models we used the BIC for each imputed data set. For all imputed data sets, the BIC favors model 1 over model 2. For all comparisons, the absolute difference in BIC score is at least 10 points. So, according to the BIC model 1 is preferred. This is congruent with the coefficient of determination results, considering adding three predictors: floor, property age, garden, did not yield a higher r-squared value.

4.5.3 Conclusion

5 Overall Conclusion

6 Appendix

Summary of the raw data using the my_skim function.

```r
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
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