Exploratory Data Analysis on Housing Market in Italy

Supervised Learning & Visualization

Daniel Anadria   Kyuri Park   Ernst-Paul Swens   Emilia Löscher

October 3, 2022

Contents

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 a couple of interesting research questions which can be answered by means of data visualization and summary statistics.

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. We do not have any information on the representativeness of our dataset. Hence, we advise caution when drawing inferences from our findings.

In order to plot the statistics of interest to maps of Italy, we use the regional and provincial shape files, which are 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 to cluster the municipalities in our `location` variable into their respective provinces and regions.

For each housing sale post, the dataset contains the following variables:

Table 1: Description of Variables in the Italy Housing Dataset

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 housing dataset as well as the ISTAT shape files.

2.1 Load Packages & Import Data

```
## load packages
library(tidyverse)   # for wrangling data
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
library(mice)        # for imputation procedure

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

## import istat shape files
# municipality
municipalities <- st_read("data/italy_shape_2022_files/Com01012022_g")
# extract only relevant geometric info
municipalities <- municipalities[c("COD_REG", "COD_PROV", "COMUNE")]
# province
provinces <- st_read("data/italy_shape_2022_files/ProvCM01012022_g")
# region
regions <- st_read("data/italy_shape_2022_files/Reg01012022_g")
```

3 Exploratory Research Questions

The goal of our analysis is:

1. Exploring the missingness in the dataset.
2. Investigating whether there are any geographical trends in the housing prices on regional and/or provincial level.

4 Data Cleaning

Note: We base the following data cleaning on the summary of the raw dataset, which can be found in the *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*) as they are deemed irrelevant. In addition, we exclude two columns that have only one unique value (`status` and `availibility`), as these are not variables but constants.

We observe that the 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 `property_age` 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 later as of now. Thus, we filter out those with `year_of_construction` > 2026.

For the second exploratory research question, our variable of interest is `price`. We notice that it is highly skewed to the right given that the mean (239,939) is far off to the right of the median (135,000). We examine the distribution of `price` with a boxplot (see *Figure 1*).

```r
## create our own theme that can be used throughout
custom.theme = theme(
  axis.title.x = element_text(size = 14),
  axis.text.x = element_text(size = 13),
  axis.title.y = element_text(size = 14),
  axis.text.y = element_text(size = 13))

## boxplot of price
houses %>%
   ggplot(aes(x=price)) +
   geom_boxplot() +
   # add comma on the x-axis labels
   scale_x_continuous(labels=scales::label_comma(),
   # rotate the x-axis labels
   guide = guide_axis(angle = 25)) +
   # apply minimal theme plus our own customized theme
   theme_minimal() + custom.theme
```

```r
## density plot of price (cleaned dataset)
houses %>%
  # filter the price over a million
  filter(price <= 1e6 | is.na(price)) %>%
  # create a ggplot
  ggplot(aes(price)) +
  # add histogram
  geom_histogram(aes(y=..density..), bins = 30, color = 1, fill="white") +
  # add density line
  geom_density(lwd=0.5, color = "#165e70", fill = "#165e70", alpha = 0.2) +
```

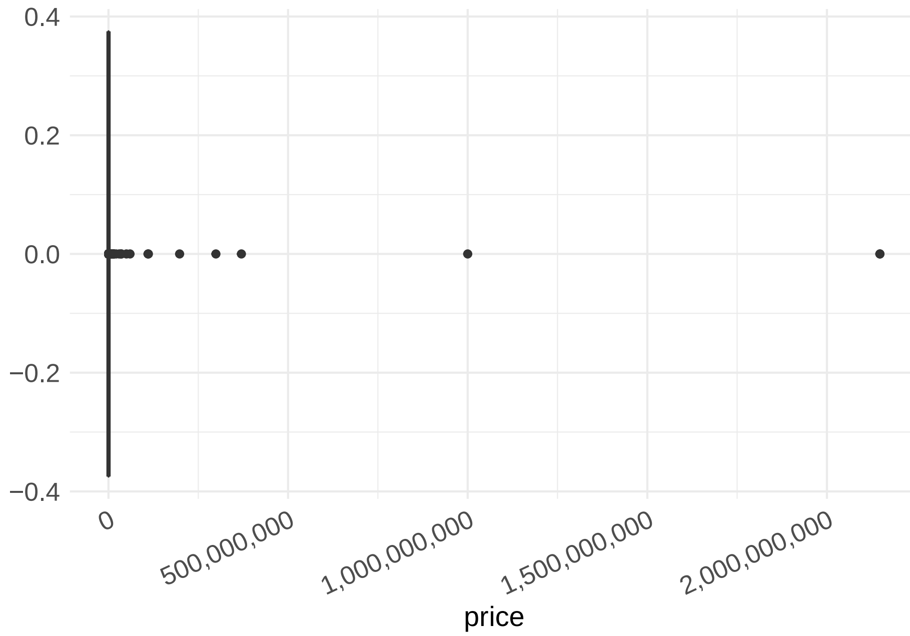


Figure 1: Boxplot of Housing Prices in Italy

```
  # apply our theme
  theme_minimal() + custom.theme
```

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. The distribution of housing prices after filtering can be seen in *Figure 2*.

As shown in *Figure 2*, the distribution of housing prices still has a long right tail after eliminating extreme outliers, but that is to be expected with housing prices in any country. Based on this, we conclude that when working with this housing price data, it is advisable to use centrality and spread measures that are robust to skewed data. For this reason, we will use *median* and *median absolute deviation (MAD)* instead of the mean and variance in our exploration of the present dataset.

```
## cleaning up the housing data
houses.cleaned <- houses %>%
  # only select variables that have more than one unique value
  select(where(~n_distinct(.) > 1)) %>%
  # fix the data type (convert them to factor)
  mutate(across(c(starts_with("has"), is_furnished, heating, energy_class, n_rooms,
                  n_bathrooms, location), factor)) %>%
  # filter out houses whose price is over a million (while keeping NAs)
  filter(price <= 1e6 | is.na(price),
  # filter out houses with a built year over 2026 (while keeping NAs)
         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))
```

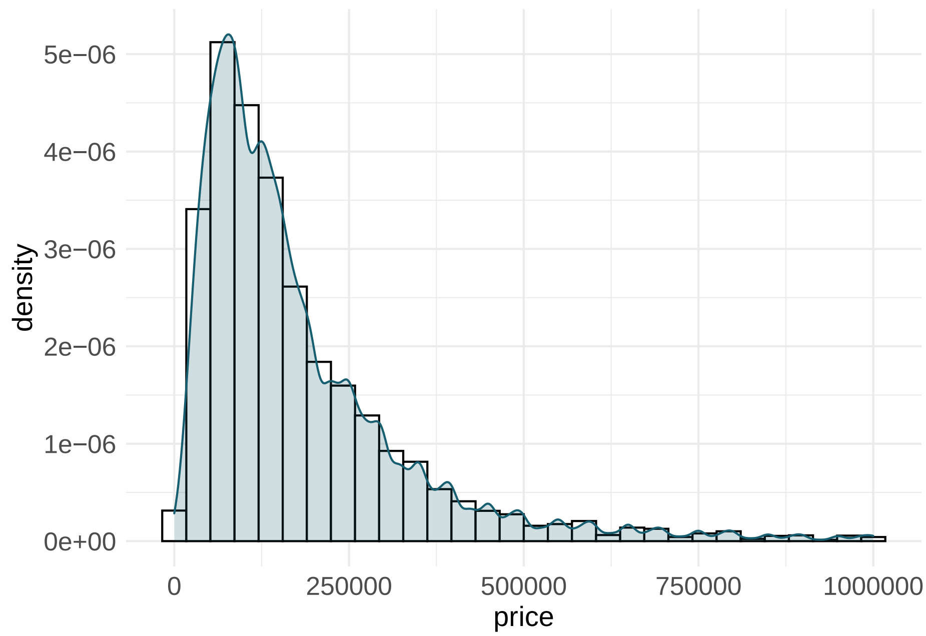


Figure 2: Histogram and Density Plot of Housing Price After Filtering

In order to employ the geometric data that we imported earlier (*ISTAT* shape files), we append them to the cleaned housing dataset. To this end, we use *fuzzy string 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.

```
## function to join geospatial data
join.geo <- function(data) {
  data.geo <-
    # fuzzy join by municipality
    stringdist_left_join(houses.cleaned, municipalities,
      by = c("location" = "COMUNE"), distance_col = "distance",
      ignore_case = TRUE) %>%
    # keep the closest match for each location
    group_by(location) %>%
    slice_min(distance) %>%
    # remove geometry and distance
    select(-geometry,-distance) %>%
    # left join region information, remove geometry and COMUNE
    left_join(., as.data.frame(regions[,c("COD_REG","DEN_REG")])) %>%
    select(-geometry, -COMUNE) %>%
    # left join province information
    left_join(., as.data.frame(provinces[,c("DEN_UTS", "COD_PROV")],
      by = "COD_PROV")) %>%
    # rename DEN_REG and DEN_UTS
    rename(region = DEN_REG, province = DEN_UTS) %>%
    # remove geometry, regional and provincial codes
    select(-geometry, -COD_REG, -COD_PROV) %>%
    # move region and province to after location
    relocate(c(region, province), .after = location)
```

```
}

## join geospatial data to the cleaned house data
data.geo <- join.geo(houses.cleaned)
```

```
## load the cleaned house data including the geospatial data
data.geo <- readRDS("data/datageo.rds")
```

4.1 Data Summary

After data cleaning, we take a look at the summary statistics to get a better overview of our data. We skim through our cleaned dataset using the `skimr` package.

```
## settings: round up by 2 decimal places & disable scientific notation
options(digits = 3, scipen = 999)
## specify skimming function
custom.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 statistics table
custom.skim(houses.cleaned)
```

Table 2: Data summary

Name	houses.cleaned
Number of rows	220748
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	58297	0.74	4	3: 56766, 4: 47144, 5: 30944, 2: 27597
n_bathrooms	12677	0.94	3	1: 107372, 2: 79843, 3: 20856
energy_class	638	1.00	12	g: 115238, f: 25396, e: 17124, a: 15931
heating	0	1.00	2	aut: 197849, oth: 22899
has_garage	0	1.00	2	0: 180669, 1: 40079
has_terrace	0	1.00	2	0: 196111, 1: 24637

skim_variable	n_missing	complete_rate	n_unique	top_counts
has_garden	0	1.00	2	0: 184426, 1: 36322
has_balcony	0	1.00	2	0: 198058, 1: 22690
has_fireplace	0	1.00	2	0: 208817, 1: 11931
has_alarm	0	1.00	2	0: 218752, 1: 1996
has_air_conditioning	0	1.00	2	0: 155058, 1: 65690
has_pool	0	1.00	2	0: 216473, 1: 4275
has_parking	0	1.00	2	0: 217364, 1: 3384
has_elevator	0	1.00	2	0: 208067, 1: 12681
is_furnished	0	1.00	2	0: 203644, 1: 17104

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	median	min	max	n_unique
price	39113	0.82	177784.71	153812.51	130000	1	1000000	2555
floor	71398	0.68	1.82	1.13	2	1	52	20
mq	3343	0.98	156.08	124.68	116	1	999	971
property_age	10	1.00	56.07	74.36	42	-3	1022	375

From the data summary table, 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 presented in a table for factor and numeric variables, separately.

From the table of factor variables, we again see that `location` has 7023 unique values (i.e., municipalities). Also, there are some missing values for `energy_class` and a lot of missing values for `n_rooms` and `n_bathrooms`. We look into this more in detail when we address the first exploratory question which concerns the missingness in the dataset. Looking at the `top_counts`, we see that three rooms (`n_rooms`), one bathroom (`n_bathrooms`), autonomous heating (`heating`) and energy class of *g* (`energy_class`) are the most frequent value of the respective factors.

In the numeric table, we see that all variables except for the `property age` have some missing values. Furthermore, we see that the mean and median of `floor` is around 2 and the maximum is 52, indicating that most houses are with more less than 2 floors with some exception of high-rise multi-story houses. The average size in square meters (`mq`) for this dataset is 156.1 m^2. Given that its standard deviation is 124.7, there seems quite some deviations in size of properties. The mean and median of property age is about 56.1 and 42 respectively, and the oldest one according to the dataset is 1022 years old. Considering the standard deviation of property age is also not small (74.4), we assume that there are probably few really old properties, while most of them are likely to fall under 100 years. We acknowledge that the minimum value of `price` and `mq` is 1, which may not be realistic. Yet, we decide to keep them as we are not completely certain if they were due to measurement/entry error.

5 Answering the Exploratory Questions

5.1 Question 1: Exploring missingness in the dataset.

5.1.1 Visual Exploration

First, we calculate and plot the proportion of missing values for all the variables (see *Figure 3*).

```r
## barplot to show the missingness per variable
houses.cleaned %>%
   # get the average missing proportion
   is.na() %>% colMeans() %>% stack() %>%
   # create ggplot for the missing proportion
   ggplot(aes(x = reorder(ind, values), y=values)) +
   # specify the bar plot
   geom_bar(stat="identity", width = .2, fill = "black",
            position = position_dodge2(padding = 0.8)) +
   # flip the bar plot and add the points
   geom_point() + coord_flip() +
   # apply our custom theme
   theme_minimal() + custom.theme +
   # specify axis labels
   labs(x = "", y = "Missingness (%)")
```

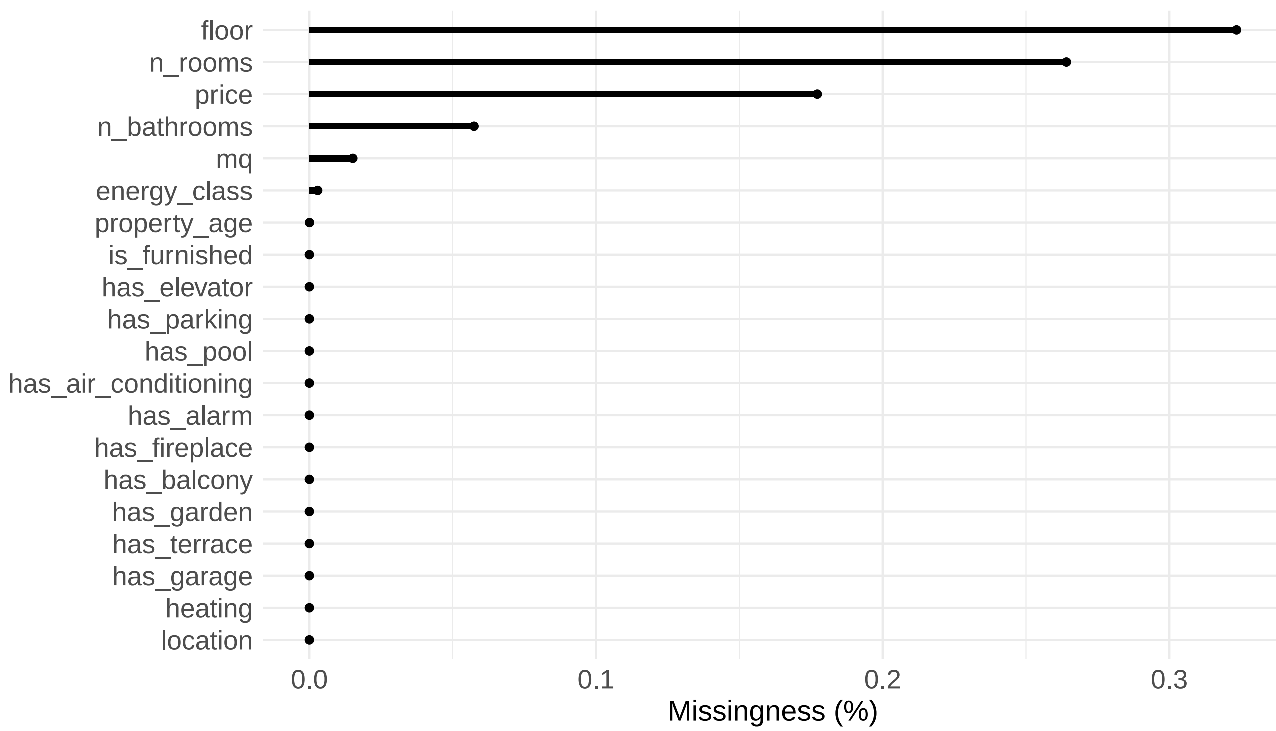


Figure 3: Proportion of Missing Values

Figure 3 shows that `floor` and `n_rooms` have high percentage of missing values, which lies above 30% and 25% respectively. The variable `price` also has quite substantial missing values of about 18%, followed by `n_bathrooms` that has about 6% missing values. Except for `energy_class`, which has about 1% missing, the remaining variables have no missing values.

To explore further on the missingness of `price`, which is the variable of our main interest, we compute the correlation between missingness of *price* and other variables to see if there are any relations.

```r
houses.cleaned %>%
  # create missingness indicator of price
```

```
  mutate(na_ind = ifelse(is.na(price), 1, 0)) %>%
  ungroup() %>%
  # compute correlation
  summarize(across(n_rooms:property_age, ~cor(as.numeric(.x), na_ind,
                                          use = "pairwise.complete.obs"))) %>%
  # sort them in descending order based on their absolute values
  t(.) %>% .[order(abs(.), decreasing = TRUE),] %>%
  # create a neat table
  knitr::kable(col.names = "correlation", caption = "Correlation between missingness
             of price and other variables", align="c")
```

Table 5: Correlation between missingness of price and other variables

	correlation
energy_class	-0.272
has_garden	-0.114
heating	0.109
has_garage	-0.097
has_fireplace	-0.087
has_air_conditioning	-0.086
has_terrace	-0.078
has_elevator	-0.077
n_rooms	0.062
has_balcony	-0.056
has_parking	-0.038
mq	0.033
has_alarm	-0.032
property_age	-0.020
n_bathrooms	0.017
floor	0.010
has_pool	-0.010
is_furnished	0.005

As shown in *Table 5*, the missingness of `price` appears to be moderately correlated with the `energy_class` (*cor = -0.271*). Hence, we visualize the count of missing values of `price` across different energy classes to see we can recognize any patterns (see *Figure 4*).

```
## create a barplot for missing values in price vs energy classes
houses.cleaned %>%
  # add price missingness indicator
  mutate(na_ind = ifelse(is.na(price), 1, 0)) %>%
  # group by energy class
  group_by(energy_class) %>%
  # sum up all the missingness in price per energy class
  summarize(n = sum(na_ind)) %>%
  # create a ggplot for the sum of missingness
  ggplot(aes(x = energy_class, y = n, fill = energy_class)) +
```

```r
  # turn off legend
  geom_col(show.legend=FALSE) +
  # customize the color
  scale_fill_manual(values = c("#F8766D", "#999999","#F8766D",
                               rep("#999999", 8),"#F8766D")) +
  # apply the themes to the bar plot
  theme_minimal() + custom.theme +
  # change the labels
  labs(x = "Energy Class", y = "Missingness in price")
```

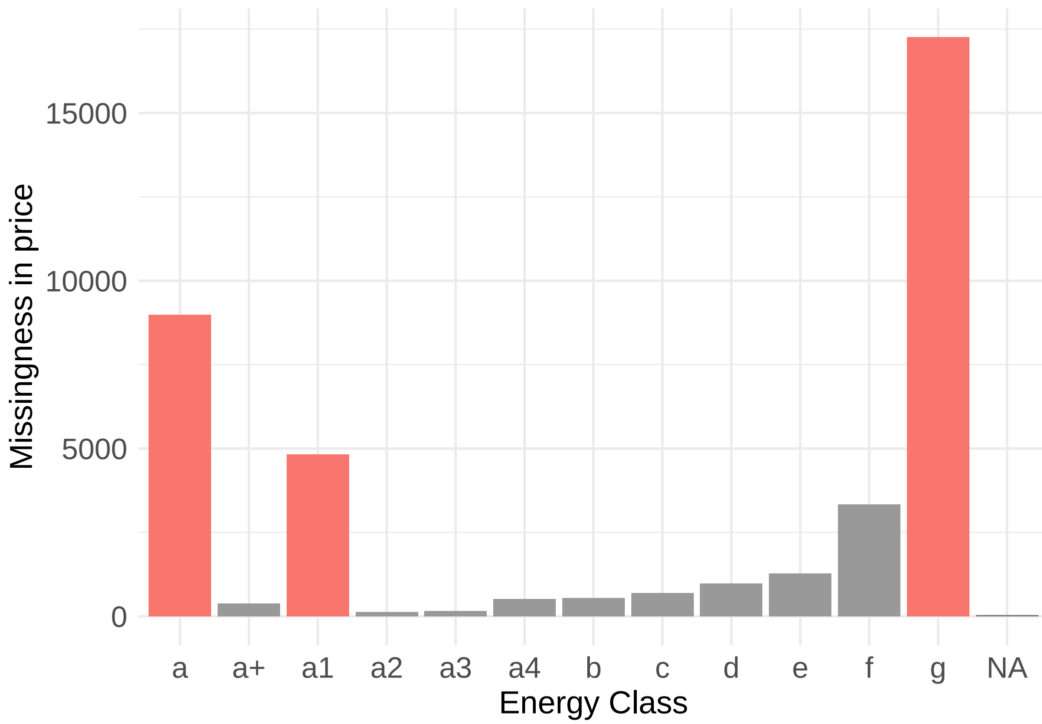


Figure 4: Missing Values in Price Across Different Energy Classes

Figure 4 shows that there is higher missingness in price for houses that either have good energy class of *a* or *a1*, or fall into the very inefficient energy class *g* (marked in red). There is low missingness in *price* for the other energy classes and for houses whose *energy_class* is missing (NA).

In addition, we check the missingness in `price` across different regions to see if there are any geographical patterns.

```r
## check missingness in price w.r.t regions
data.geo %>%
   # add price missingness indicator
   mutate(na_ind = ifelse(is.na(price), 1, 0)) %>%
   group_by(region) %>%
   # get the total missingness proportion per region
   summarize(`Average missing proportion (%)` = sum(na_ind) / n()) %>%
   # add the spatial data
   left_join(., regions, by = c("region" = "DEN_REG")) %>%
```

```
# convert it to `sf` object
st_as_sf() %>%
ggplot() +
# plot the italy map
geom_sf(fill=NA) +
# add scatter plots of missingness proportion per region
geom_point(color = alpha("red", 0.4),
aes(size = `Average missing proportion (%)`, geometry = geometry),
stat = "sf_coordinates") +
scale_size(range = c(1, 5)) +
# remove unnecessary coordinates
theme_void() +
# relocate the legend
theme(legend.position = "bottom")
```

Figure 5: Missingness of Price per Region

Figure 5 shows that the proportion of missing data in `price` is not equal across different regions. However, there seems not to be any regions, which are drastically more/less likely to have missing values in `price` either.

Next, we examine the relationships between the missingness of other variables that have high percentage of missing data (i.e., `floor`, `n_rooms`, `n_bathrooms`, `mq`) and `price`.

For each variable, we construct a missingness indicator and plot the density of price (on a log10-scale) while splitting the densities by the missingness indicator.

```r
## function to create a density plot of price with missing indicator
missing.plots <- function(x) {
  plot <- houses.cleaned %>% ungroup() %>%
    # create the missingness indicator
    mutate(missing = is.na(.[,x])) %>%
    # specify ggplot of price on log-10 scale
    ggplot(aes(x = log10(price), fill = missing)) +
    geom_density(alpha = 0.5, color = NA) +
    theme_classic() +
    # speicfy the axis label
    labs(x = expression(paste(Log["10"],"(Price)"), y = "Density")) +
    # set the x-lim and y-lim
    scale_x_continuous(limits = c(2, 7)) + ylim(0, 1.5)
  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"),
          # add labels and legend
          labels = c("        A. Floor", "B. Number of rooms", "C. Number of bathrooms",
                     "D. Squared meter(m2)"),
                     font.label=list(color="black",size = 10),
          ncol = 2, nrow = 2, common.legend = TRUE, legend = "bottom")
```

The missingness of number of floors (*Figure 6A*) and number of rooms (*Figure 6B*) do not seem to be dependent much on `price` information. Whereas for the number of bathrooms (*Figure 6C*) and the meters squared (*Figure 6D*) missingness show a different resulting density of `price`, namely, missingness tends to occur at higher house prices.

5.1.2 Conclusion

For the first question, we explored missingness in the dataset. We saw that there are quite a lot of missing values for the *floor*, the *number of rooms*, the *number of bathrooms*, and *house price*. As *house price* is of interest for the second question, we took a closer look and realized that missingness on *price* is correlated, e.g., with *energy class* and that the amount of missingness varies somewhat across regions.

Additionally, we also examined if the missingness of the other variables and *house price* are related. From our results, it appears that they are. Hence, it is unlikely that the missingness mechanism is missing completely at random (*MCAR*). Therefore, before we go on to the next question of exploring geographical patterns of housing price, we decide to first perform multiple imputation procedure.

5.1.3 Imputation

We run a multiple imputation procedure ($m = 5$) on our data using a selection of predictors derived from `quickpred`. We kept the default imputation method. This resulted in predictive mean matching for `price`, `floor`, `mq`, `property_age` and polynomial regression for `n_rooms`, `n_bathrooms`, and `energy_class`. Note

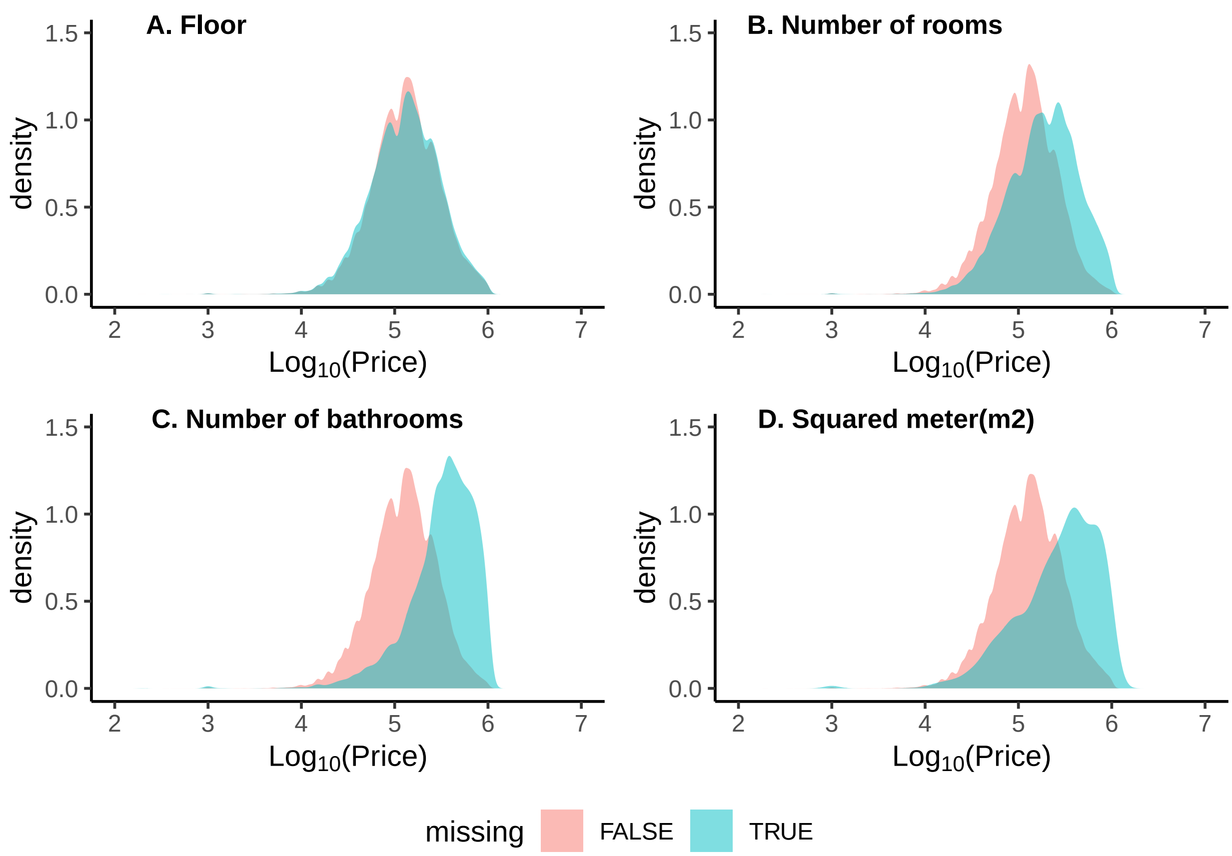


Figure 6: Density Plots for Price based on the Missing Indicators

that some of the following code chunks are set to `eval=FALSE`. They take a long time to run and need not be rerun as we saved the resulting objects and load them in subsequent chunks.

```
## create predictor matrix
predictors <- houses.cleaned %>% ungroup() %>% quickpred()

## multiple imputation procedure (m = 5)
imputations <- mice(houses.cleaned, m = 5, seed = 1, predictorMatrix = predictors)

## join geospatial data to the imputed data
imputed <- lapply(complete(imputations, "all"), join.geo)
```

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 (see *Appendix*).

```
## load imputation results
imputations <- readRDS("data/imputation.rds")

## load the imputed data including the geospatial data
imputed <- readRDS("data/imputed.rds")
```

5.2 Question 2: Regional and Provincial Trends in the Median Housing Price and the Median Absolute Deviations in Italy

We use the imputed dataset to explore whether there are geographical trends in the median and the median absolute deviation (MAD) of *housing price* (as we previously discussed that `price` is highly skewed and therefore probably it is a good idea to use *median* and *MAD* rather than *mean* and *variance*).

First we plot the median and the MAD on the regional and the provincial level:

```
## function to group by region and obtain price estimates
group.region <- function(data, estimate) {
   data %>%
      group_by(region) %>%
      summarize(estimate = ifelse(estimate, median(price), mad(price))) %>%
      select(-region)
}

# derive pooled estimates per region
region.median <- lapply(imputed, group.region, estimate = T) %>%
   do.call(cbind, .) %>% rowMeans()
region.mad <- lapply(imputed, group.region, estimate = F) %>%
   do.call(cbind, .) %>% rowMeans()

# vector for the labels in the plots
labels <- c("Trentino-Alto Adige", "Molise", "Piemonte", "Bolzano", "Calabria")

# combine region results into one dataframe
```

```r
price.by.region <- data.frame(region = sort(regions$DEN_REG),
   median = region.median, mad = region.mad) %>%
   left_join(.,regions, by = c("region" = "DEN_REG")) %>% st_as_sf() %>%
   mutate(label = ifelse(region %in% labels, region, NA))
```

```r
## function to group by province and obtain price estimates
group.province <- function(data, estimate) {
   data %>%
      group_by(province) %>%
      summarize(estimate = ifelse(estimate, median(price), mad(price))) %>%
      select(-province)
}

# derive pooled estimates per province
province.median <- lapply(imputed, group.province, estimate = T) %>%
   do.call(cbind, .) %>% rowMeans()
province.mad <- lapply(imputed, group.province, estimate = F) %>%
   do.call(cbind, .) %>% rowMeans()

# combine province results into one dataframe
price.by.province <- data.frame(province = sort(provinces$DEN_UTS),
   median = province.median, mad = province.mad) %>%
   left_join(., provinces, by = c("province" = "DEN_PROV")) %>% st_as_sf()
```

```r
plot.list.1 <- list()

## median & mad of price per region
plot.list.1 <- map(
  c("median", "mad"),
  function(var) {
    ggplot(price.by.region) +
      # map each statistic
      geom_sf(aes(fill = .data[[var]])) +
      # void theme: remove all unncessary coordinates
      theme_void() +
      # add labels to specific regions
      geom_sf_text(aes(label = label), color = "black", size = 3.9, fontface = "bold") +
      # color-scheme (color-blind friendly???)
      scale_fill_viridis_c(option = "E", direction = -1) +
      # lengthen the legend
      theme(legend.key.width= unit(2, 'cm'))
     }
  )
plot.list.2 <- list()

## median & mad of price per province
plot.list.2 <- map(
  c("median", "mad"),
  function(var) {
```

```
    ggplot(price.by.province) +
      # map each statistic
      geom_sf(aes(fill = .data[[var]])) +
      # void theme: remove all unncessary coordinates
      theme_void() +
      # color-scheme (color-blind friendly???)
      scale_fill_viridis_c(option = "E", direction = -1) +
      # lengthen the legend
      theme(legend.key.width= unit(2, 'cm'))
    }
  )

## combine the plot lists
plot.list <- c(plot.list.1, plot.list.2)
```

```
## plot the median of price
ggarrange(plotlist = plot.list[c(1,3)], nrow = 1, ncol = 2, common.legend = TRUE,
          legend = "bottom")
```

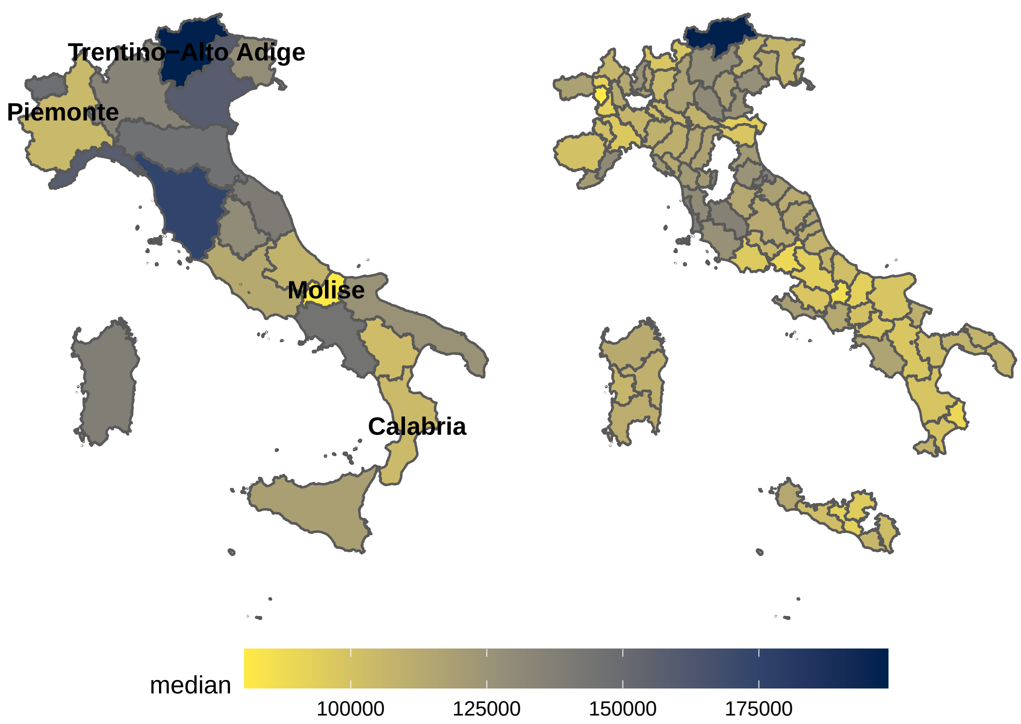


Figure 7: Median of Price per Region (left) and Province (right)

```
## plot the mad of price
ggarrange(plotlist = plot.list[c(2,4)], nrow = 1, ncol = 2, common.legend = TRUE,
          legend = "bottom")
```

On the regional level, we see that the median of price is the highest for region *Trentino-Alto Adige* (€200,000; the darkest blue), and the lowest for region *Molise* (€79,000; the brightest yellow). We

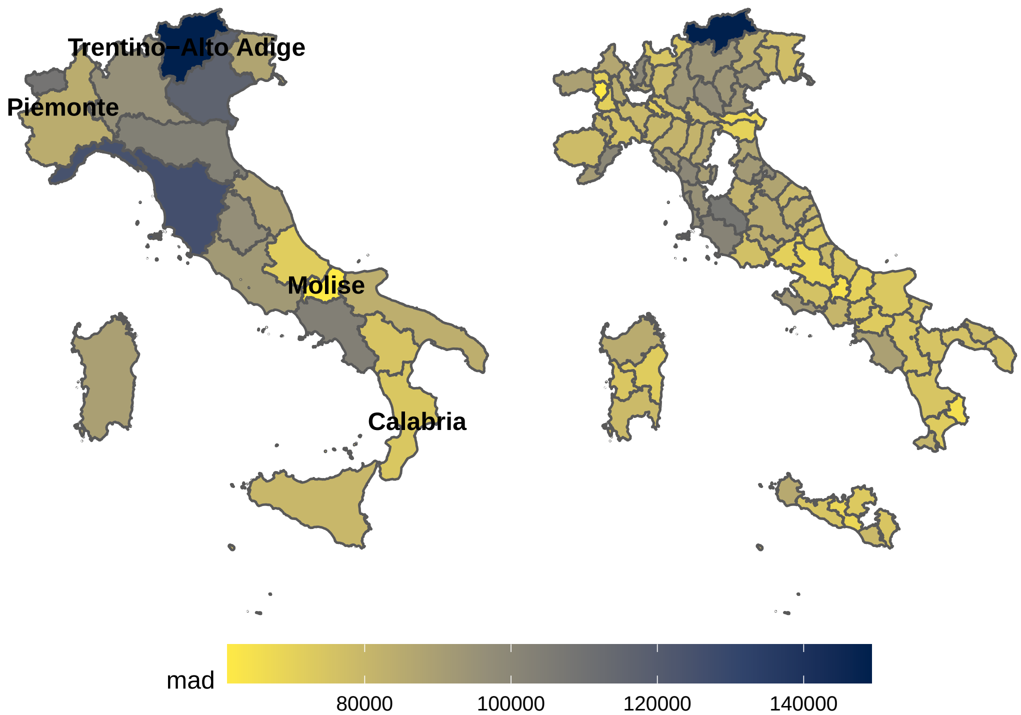


Figure 8: MAD of Price per Region (left) and Province (right)

recognize that there is a geographical trend in housing price such that the median price is in general lower for the Southern regions compared to the Northern regions in Italy. The only exception from this is the region *Piemonte* (€99,000), which has a lower median price than the surrounding regions.

On the provincial level, it can be seen that the high median of *Trentino-Alto Adige* region is mainly due to the high median of the province *Bolzano* (€400,000). The general trend of Northern regions being more expensive than the Southern regions can still be observed on the provincial level. In addition, we notice that there is no housing sales data available in some provinces (see the empty space/missing provinces in the province map).

Regarding the MAD, a measure of variability in housing price, we see that it is higher for the regions with high median prices, and lower for the regions with low median prices. This is recognizable as the overall color pattern in the median plot matches with the pattern in the MAD plot, which implies that again Northern parts tend to have more variability in price compared to the Southern parts of Italy.

Given that the overall geographical pattern for median and MAD of `price` correspond to each other, it is perhaps interesting to investigate further the difference in the distribution of `price` between high- and low-median regions. We proceed by selecting the top 2 highest median regions and bottom 2 lowest median regions, then plotting the distribution of `price` for each of the corresponding regions. See *Figure 9*.

```
## top 2 high median regions
top.2.median <- price.by.region %>% slice_max(median, n = 2) %>% pull(region)

## lowest 2 low median regions
lowest.2.median <- price.by.region %>% slice_min(median, n = 2) %>% pull(region)

## plot the histograms for each region in the high- and low-median groups
```

```r
imputed[[1]] %>%
  # subset top two and bottom two countries
  filter(region %in% c(top.2.median, lowest.2.median)) %>%
  # group by the regions
  group_by(region) %>%
  # create the grouping variable for coloring
  mutate(grouping = ifelse(region %in% top.2.median, "high median", "low median"),
         # get the median price for each region
         med_price = median(price, na.rm=T)) %>%
  # create ggplot for price (coloring by groups)
  ggplot(aes(x = price, fill = grouping)) +
  geom_histogram(bins = 30) +
  # create a panel of plots per region
  theme_bw() + facet_wrap(~region) +
  # indicate the median price by a vertical line
  geom_vline(aes(xintercept = med_price, group=region), linetype="dashed") +
  # change legend title
  labs(fill = "high/low regions")
```

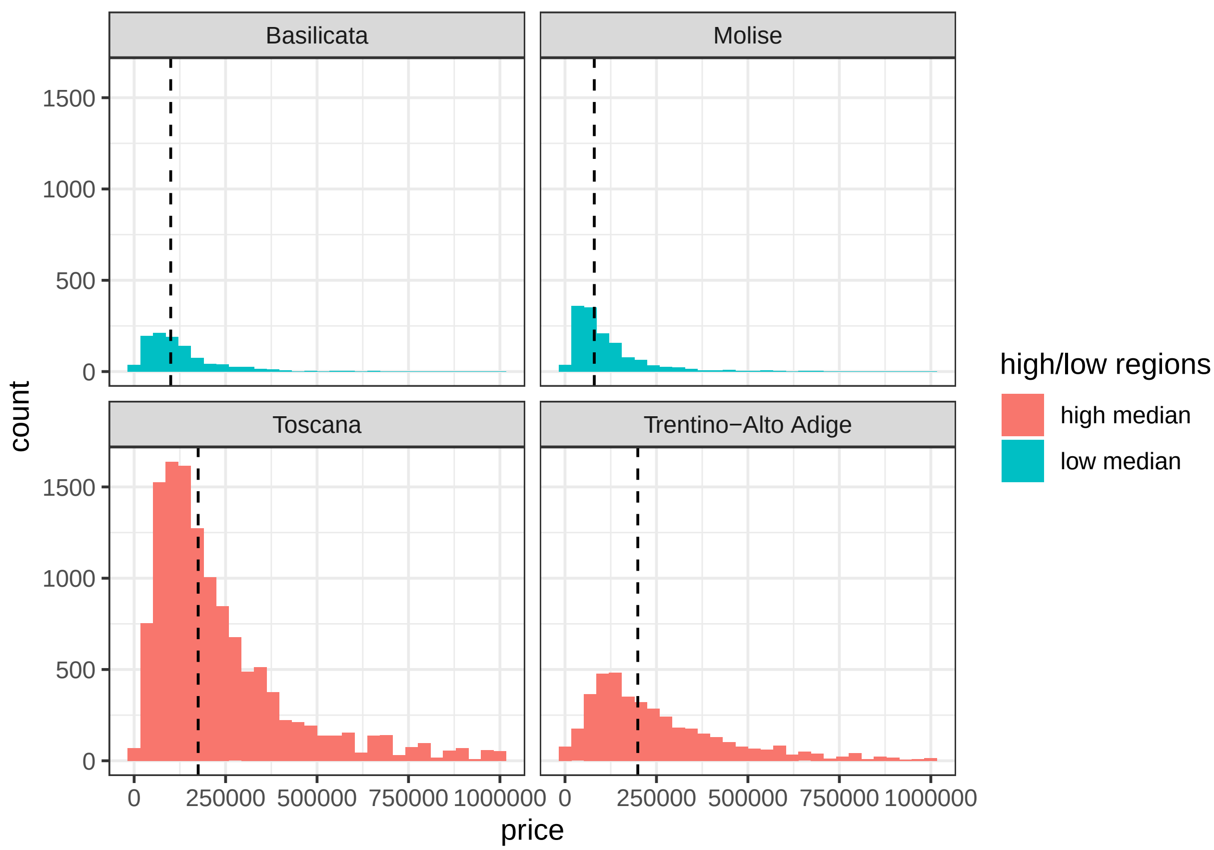


Figure 9: group differences in price distribution

An interesting take-away from *Figure 9* is that all the densities for the *house price* are right skewed. This is reasonable as there is a fixed lower boundary to how cheap houses can be, the bulk represents what most people can afford, but there is no set limit on the maximum house price. Very expensive houses are rare, but theoretically there is no fixed upper limit to a house cost. There is a geographic trend to house prices in Italy. The most expensive sales in the dataset are from the *Toscana* region where there are many more expensive houses than in the regions of *Calabria* and *Molise* where the median house price tends to be lower.

5.2.1 Conclusion

All things considered, we found a general decrease in the median price when moving from northern to southern Italy. We also showed that higher medians on a regional level are due to a higher median on the province level with one region having a particular high median. Consequently, such regions where there is one province with a higher median of *house price*, the MAD is higher too if the other provinces in that region do not have a high median. Furthermore, by including the plots on the province level, it was recognizable that there were no sales in some provinces. ***How many provinces did not have data?***

6 Overall Conclusion

All things considered, we identified that there is quite some missingness present - especially for *house price* for which about 18% of values were missing. The absolute correlation of missingness in *house price* was highest with *energy class* (-0.27). As we were not able to conclude from our analysis that the missingness mechanism is MCAR, we decided to impute the missing values.

Using the imputed dataset, we explored the median and the MAD on the regional and provincial level in Italy. We found a trend in the median price from more expensive to cheaper when going from North to South in Italy. We also have shown that these higher medians on a regional level are due to a higher median on the provincial level. In most cases, the MAD for such regions was high, too, as there frequently is one province with a higher median of *house price* and the MAD is consequently higher if the other provinces in that region do not have a high median, too.

7 Appendix

7.1 Summary of the raw data using the my_skim function.

```
custom.skim(houses)
```

Table 6: Data summary

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

Variable type: character

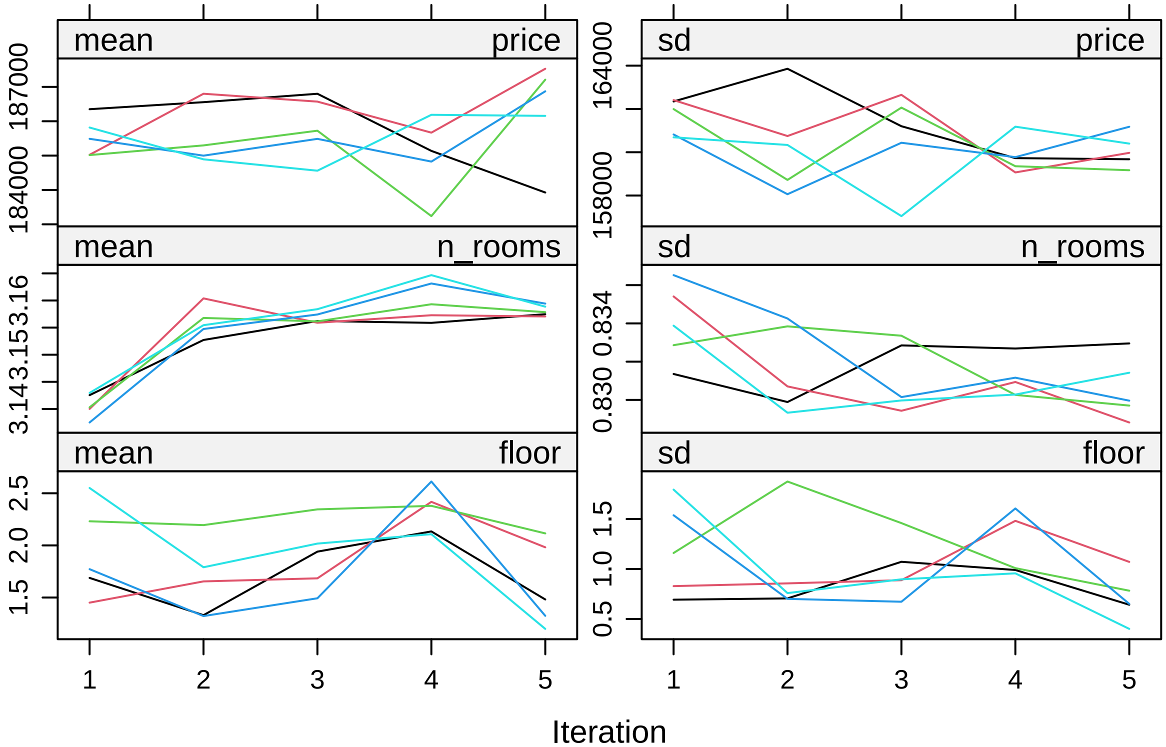
skim_variable	n_missing	complete_rate	empty	n_unique
location	0	1	0	7023
title	0	1	0	199305
availability	0	1	0	1
energy_class	679	1	0	12
status	0	1	0	1
heating	0	1	0	2

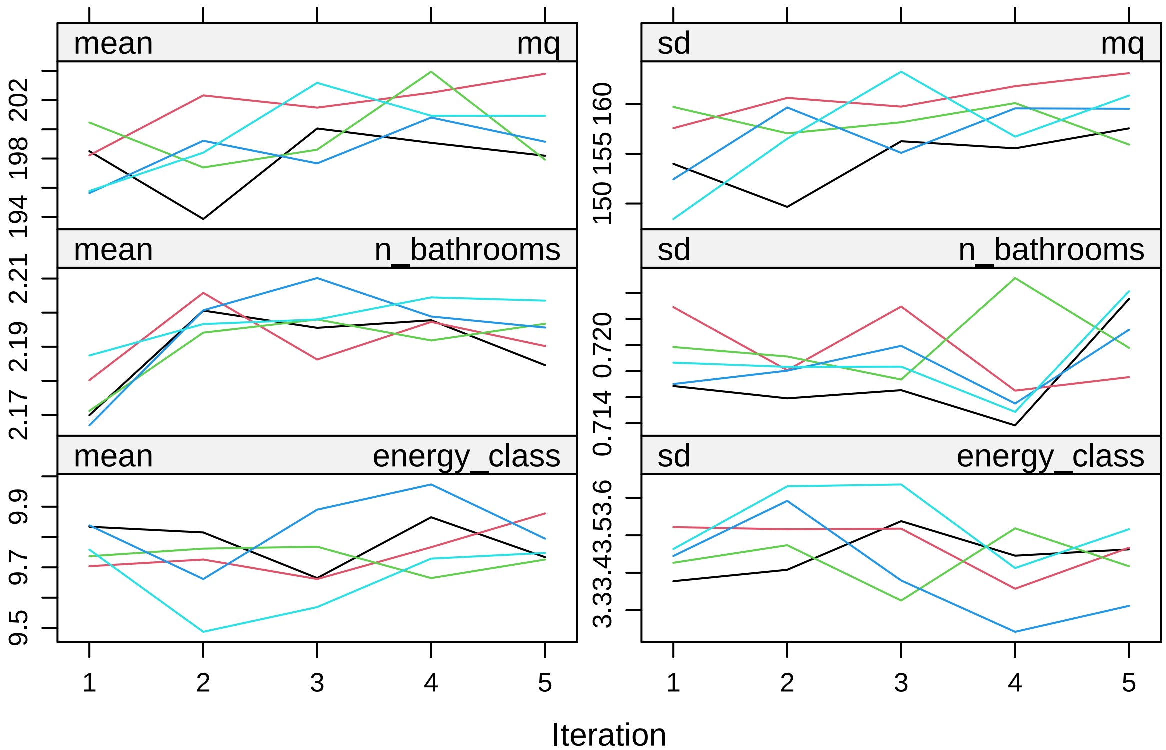
Variable type: numeric

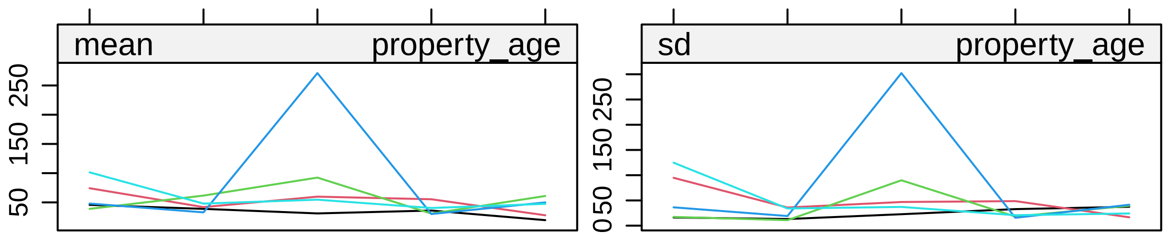
skim_variable	n_missing	complete_rate	mean	sd	median	min	max	n_unique
id	0	1.00	111705.00	64492.77	111705	1	223409	223409
timestamp	0	1.00	1661135705.37	12645.42	1661135577	1661114079	1661158618	42238
price	39116	0.82	239938.98	7562062.01	135000	1	2147483647	2852
n_rooms	60323	0.73	3.50	0.99	3	2	5	4
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

7.2 Convergence of the algorithm and plausability of the imputed dataset

```
## convergence of the algorithm
plot(imputations)
```

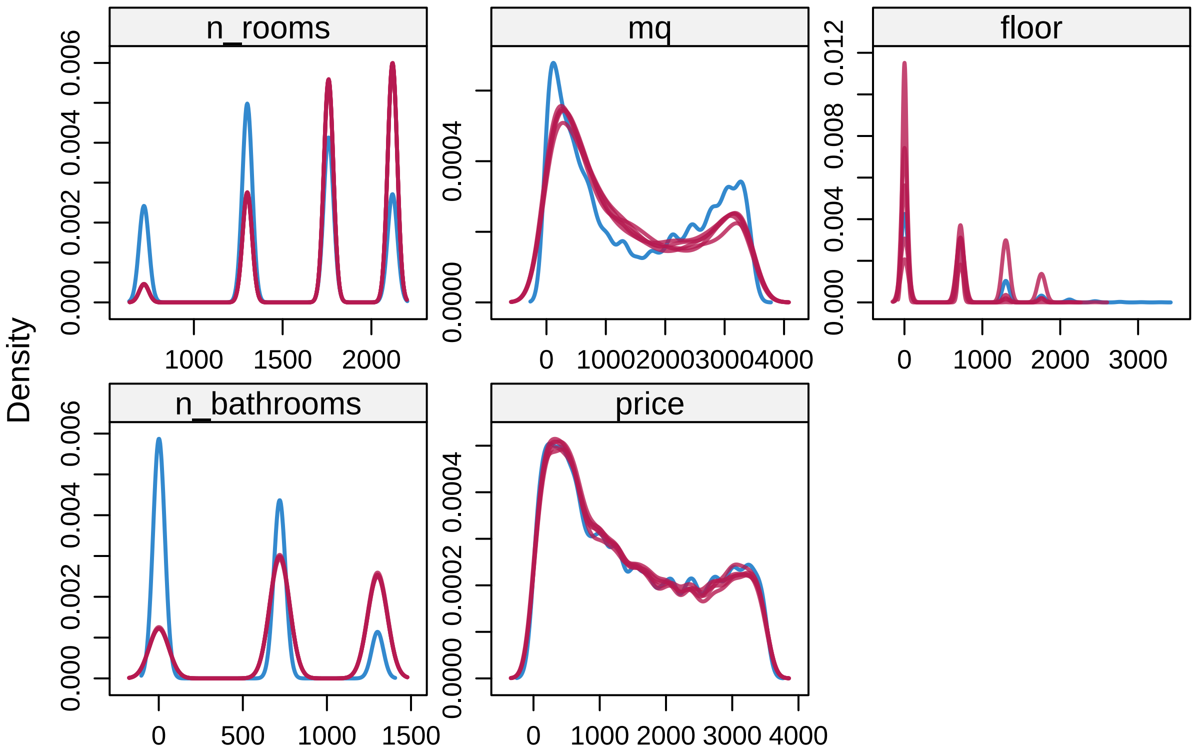

mean
price
sd
price
mean
n_rooms
sd
n_rooms
mean
floor
sd
floor
Iteration


mean
mq
sd
mq
mean
n_bathrooms
sd
n_bathrooms
mean
energy_class
sd
energy_class
Iteration



Iteration

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



Density
n_rooms
mq
floor
n_bathrooms
price
0.000
0.002
0.004
0.006
0.0000
0.0004
0.000
0.004
0.008
0.012
0.0000
0.0002
0.0004
1000
1500
2000
0
1000
2000
3000
4000
0
500
1000
1500