Step 1. Open the data file and study the general information.

While conducting exploratory analysis, I discovered that there are 14 out of 22 columns that have null values. Additionally, there are several columns whose types don't make sense for its name. Such as, object type for date posted, float type for days listed, float type for balconies, float type for total floors, etc.

Step 2. Data preprocessing

For the airport_dist column, I identified null values and filled those values with the mean of the column. I decided to do so because every listing has to be a certain distance from the airport. I finished preprocessing this column by converting its values to float type.
For the balconies column, I identified null values and filled those values with 0. I decided to do so because you can't assume that a listing has a balcony so it's better to say otherwise. I finished preprocessing this column by converting its values to int type.
For the ceiling_height column, I identified null values and filled those values with the mean of the column. I decided to do so because every listing has to be a certain distance from the city center. I finished preprocessing this column by rounding the values to two decimal values.
For the days_listed column, I identified null values and filled those values with 0. I decided to do so because you can't assume that a listing has been on the market for any number of days. I finished preprocessing this column by converting its values to int type.
For the airport_dist column, I preprocessed this column by converting its values to a datetime type object.
For the floors_total column, I identified null values and filled those values with 0. I decided to do so because you can't assume that a listing has any number of floors. I finished preprocessing this column by converting its values to int type.
For the bike_parking column, I identified null values and filled those values with False. I decided to do so because you can't assume that a listening has bike parking.
For the kitchen_area column, I identified null values and filled those values with the mean of the column. I decided to do so because every listing has to have a certain kitchen area. I finished preprocessing this column by rounding the values to two decimal values.
For the last_price column, I preprocessed this column by converting its values to int type.
For the living_area column, I identified null values and filled those values with the mean of the column. I decided to do so because every listing has to have a certain living area. I finished preprocessing this column by rounding the values to two decimal values.
For the locality_name column, standardized the column values by lowercasing each entry and removing any instance of the word 'village'. I dropped any rows with null values because it's not possible to assume a listings locality.
For the parks_within_3000 column, I identified null values and filled those values with 0. I decided to do so because you can't assume that a listing has any parks within 3000 meters. I finished preprocessing this column by converting its values to int type.

For the park_dist column, I identified rows who have a null park_dist value but who have a value in parks_within_3000 greater than one and for those rows, I filled the null park_dist value with the mean of the park_dist column. Then, I identified rows who have a null park_dist value but who have a value in parks_within_3000 equal to 0 and for those rows, I filled the null park_dist value with 0.
For the ponds_within_3000 column, I identified null values and filled those values with 0. I decided to do so because you can't assume that a listing has any ponds within 3000 meters. I finished preprocessing this column by converting its values to int type.
For the pond_dist column, I identified rows who have a null pond_dist value but who have a value in ponds_within_3000 greater than one and for those rows, I filled the null pond_dist value with the mean of the pond_dist column. Then, I identified rows who have a null pond_dist value but who have a value in ponds_within_3000 equal to 0 and for those rows, I filled the null pond_dist value with 0.
All other columns were not found to have any null values or issues with the type.

Step 3. Make calculations and add them to the table

I added the price_per_sqm column to the data by dividing all entires of the last_price column by the total_area column and rounding the result to two decimal places.
I added the weekday_posted column to the data by extracting the weekday from the date_posted column using the datetime library.
I utilized the same technique to add the month_posted and year_posted column.
Utilizing the floor and floors_total column, I applied the determineFloor() function on each row to determine the type of floor that the listing is. The function works as follows: if the listing has a total number of 0 floors or if the listing itself is on the first floor, then it is a 'first' floor type; if the floor number / total floors in the building is less than one (meaning its on a floor between first and last), then it's an 'other' floor type; if the floor number / total floors in the building is exactly one (meaning it's on the 8th floor of an 8 floor building), then it's a 'last' floor type. Afterwards, the determineFloor() method with the pandas apply() method generated a column called floor_category.
I added the living_ratio column to the data by dividing all entries of the living_area column by the total_area column and rounding the result to two decimal places.
I added the kitchen_ratio column to the data by dividing all entries of the kitchen_area column bythe total_area column and rounding the result to two decimal places.

Step 4. Conduct exploratory data analysis and follow the instructions below:

These histograms show the distribution of the data for the last_price column, the ceiling_height column, the total_area column, the bedrooms column with the values of the listed columns on the x axis and the frequency of those values on the y axis.

Step 5. Overall conclusion

I found that outliers within a column can have a greater impact on the dispersion and basic calculations of that data than I expected as even a couple points can shift any kind of plot.
#
I found that there was a significant correlation between the last price a listing was sold at and the total area of the listing. Perhaps this correlation (0.64) isn't stronger because a smaller listing in a desirable location could be more expensive than a larger listing in an undesirable location. I found a weak correlation between the last price a listing was sold at and the number of bedrooms (0.35) which perhaps is for the same reason as to why the total area correlation isn't higher. I found that there is a negative correlation between the last price a listing was sold and the distance it is to the city center. Perhaps this correlation (-0.19) depends on the locality itself and requires further explortation. I found no significant correlations (very close to 0) between the last price a listing was sold and the floor number it is on, the day of the week, month, or year it was posted.
#
When examining the distance to the city center and last price listing was sold on the top 10 localities, I found there to be a significant price drop as the distance furthers from the city center. When graphing the distance to the city center and the price per sqm, there is a clear drop at about 2km. This indicates that 2km is the city center border so all listings that are less than or equal to 2km will be expensive (desirable to live in the city, prices rise) and all listnings that are greater than 2km will have a smaller price. When looking at the graph, the prices after 2km distance fluctuates - between 3km and 6km there is an increase in price (which could be an affluant suburb) afterwards a gradual decrease.
#
I found a significant correlation (0.72) between total area and bedrooms which makes sense as the larger a listing is, the most bedrooms it will have. I found a significant correlation (0.55) between total area and last price, although we know that this is not always the case when a listing is in city center.
#
#