**Data Quality Plan**

**Summary of Problems & Solutions**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Issue Description** | **Chosen Solution** |
| Address | Addresses in current format have little value | Change format to list to allow for information extraction |
| Postal Code | Missing Postal Codes | Extract Postal Codes from the Address Field using lists |
| Postal Code | Some Dublin Postal Codes are not in Dublin | Remove Postal Code from the rows |
| Property Size Description | Missing Property Size Descriptions | No action |
| Property Size Description | Duplicated Property Size Description Category | Merge categories |
| VAT Exclusive | Not all new build properties are inclusive of VAT | Add VAT to properties exclusive of VAT |
| Not Full Market Price | Some prices not at market value | No action |
| Price (€) | Dataset contains block purchases | Remove properties identified as duplicates from a new price column |
| Price (€) | Prices are not normalised - significant outliers | Remove log prices that are 2 standard deviations above the mean of each county |

**Discussion of Problems & Solutions**

**Issue 1:** Addresses in current format have little value

**Issue Description:** In their current format, the addresses pose little value. We can clearly see that the format applied is inconsistent. For example, some end with the county, some with the postal code. In addition, I note that some addresses contain the postal code when it is not contained in the postal code field. Therefore, it should be possible to extract useful information from this field to be additive to Postal Code. A difficulty with this is that the location of the Postal Code is not the same for each address.

**Potential Solutions:**

* **Remove Feature:** In its current format I have no real use for Address feature. I could choose to simply drop this feature and ignore it from my analysis. This would be quite hasty however, as there is very useful information contained in them, albeit subject to spelling mistakes, different formats, and missing aspects of the address.
* **Link to External API to look up Address:** It would be possible to lookup the addresses externally, through something like Google Maps. This would allow the extraction of postal codes, regardless of whether they were included in the address, as well as towns and the verification of the County field. Mistakes would certainly remain, as not all addresses would be filled in correctly, as well as issues where two locations share the same address. This approach, however, would be quite involved and other options are possible.
* **Change format to list to allow for information extraction:** Given that the address is currently a string it would be difficult to extract information. Therefore, I can convert this field to a list separated by commas which will allow me to iterate through the fields separately.

**Chosen Solution:**

* **Change format to list to allow for information extraction:** I decided to convert each address in my dataframe to list. This allowed me to extract postal codes from the database, easily check for block purchases and in section 4 enabled me to extract the town for those addresses where this was provided.

**Issue 2:** Missing Postal Codes

**Issue Description:** 80.76% of the data points do not have associated postal codes. I discovered in Part 1 that the Postal Code fields were only used for addresses in Dublin City, however, not all addresses in Dublin City have postal codes.

**Potential Solutions:**

* **Link to External API to look up Address:** As mentioned in Issue 1, it would be possible to lookup the addresses externally through Google Maps to allow for the extraction of postal codes, regardless of whether they were included in the address. Mistakes would certainly remain, as not all addresses would be filled in correctly, as well as issues where two locations share the same address. This approach, however, would be quite involved and other options are possible.
* **Extract Postal Codes from the Address Field using lists:** Given that I know some addresses contain the address field I can take the postal code from these, looping through each address and checking if any fields match a list of postal codes. Of course, this will not capture all postal codes for Dublin City, especially where they are not provided but also if the postal code is not spelled correctly.
* **Extract Postal Codes from the Address Field using RegEx:** I could loop through each address to check if any field contained a number, and perhaps the letter D (e.g., D1 provided instead of Dublin 1 to ensure all were captured, as well as other spelling mistakes. However, this would also capture some places that are not referring to a postal code, for example, if someone lived in apartment block D1. While using RegEx would likely capture more postal codes correctly, it would also capture more incorrectly.

**Chosen Solution:**

* **Extract Postal Codes from the Address Field using lists:** I decided to use lists to extract the postal codes from the address field. To limit mistakes, I included a check that the County for the row must be in Dublin, to prevent any addresses outside the capital being linked with an incorrect postal code. This solution increased the number of addresses with postal codes by 190.

**Issue 3:** Some Dublin Postal Codes are not in Dublin

**Issue Description:** It was discovered in section 1 that some addresses outside of Dublin were assigned Dublin Postal Codes. Clearly this is an error.

**Potential Solutions:**

* **Drop the affected rows:** It would not be unreasonable to delete the rows affected by this error, given that it calls into question the integrity of the rest of the datapoints for the row. However, this is likely a simple data entry error and we can see that the actual county provided in the County field also matches that of the Address field.
* **Remove Postal Code from the rows:** Simply remove the datapoint from the rows.

**Chosen Solution:**

* **Remove Postal Code from the rows:** I decided to delete the Postal Code from the row and set these to NaN, just like the other entries in the dataframe that are missing postal codes.

**Issue 4:** Missing Property Size Description Field

**Issue Description:** 89.47% of the datapoints do not have an entry for Property Size Description. Only New Dwellings have a description for the size of the property, however, there are 1665 new properties in the database that are new, meaning that some new properties do not have a Property Size Description. Through some analysis I discovered that the most recent entries do not have a Property Size Description feature associated, as well as some entries from the beginning. It looks as though this is information no longer collected.

**Potential Solutions:**

* **Drop feature:** This is clearly a feature that is not collected any more for new properties, and oftentimes was omitted for earlier properties too. Nevertheless, it is worth keeping it to analyse trends in size and sale price. Clearly however it will not serve any purpose in price prediction.
* **Use description from duplicated addresses:** I could check for duplicated addresses to see if one has the field filled in, to copy this across. However, the property could have changed in the period between both sales, and this is only filled in for new properties so would not serve any purpose to fill it in for a second-hand property.
* **No action:** I could ignore the fact that there is missing data and utilise what is available for analysis, to ascertain if there was a link in the past between size and price for new properties.

**Chosen Solution:**

* **No action:** I decided to leave the dataframe in its current format and keep the feature. I will be able to use it in task 3 to see if there is a connection between size and price.

**Issue 5:** Duplicated Property Size Description Category

**Issue Description:** In task 1 I observed that 2 of the description fields are essentially the same. We can update 'greater than 125 sq metres' to 'greater than or equal to 125 sq metres'. In this section I will also reduce the size of 'greater than or equal to 38 sq metres and less than 125 sq metres' to 'between 38 and 125 sq metres' so that it is more legible in charts further on.

**Potential Solutions:**

* **Ignore duplication:** Clearly both features I obversed to be the same are not the exact same, as 'greater than 125 sq metres' is a subset of 'greater than 125 sq metres' to 'greater than or equal to 125 sq metres'. However, clearly both categories should be merged, unless all properties in the latter are exactly 125 square metres which is obviously not the case.
* **Merge categories:** Remove the subset category so that there are now only 4 categories for this feature.

**Chosen Solution:**

* **Merge categories:** I decided to merge both categories, to create a larger category of 208 properties

**Issue 6:** Not all new build properties are inclusive of VAT

**Issue Description:** As discovered in task 1, only new properties are marked as not inclusive of VAT, however, not all new properties are marked as exclusive. The sale prices for these will need to be updated to reflect this. Not all new properties are exclusive of VAT, and so the conversion is justified for uniformity.

**Potential Solutions:**

* **Remove VAT from properties inclusive of VAT:** I could remove the VAT from those new properties that are inclusive of VAT. However, this is not the price that consumers pay for the properties and so would not be correct approach.
* **Add VAT to properties exclusive of VAT:** Add on the 13.5% VAT to those new properties that were exclusive of VAT.

**Chosen Solution:**

* **Add VAT to properties exclusive of VAT:** I updated all new properties marked as VAT exclusive to include the 13.5% VAT rate, to correctly reflect what consumers paid and create uniformity in the price field.

**Issue 7:** Some prices not at market value

**Issue Description:** I saw in part 1 that a considerable amount of the properties in the dataframe were not listed at full market price. This means that the data we have for some properties is not reflective of the actual valuation.

**Potential Solutions:**

* **Delete rows not at full market price:** The type of data associated with those properties sold not at market price is quite different to the remainder of the dataframe. Clearly, the prices of these will be considerably smaller, resulting in measures of central tendency artificially reduced.
* **No action:** It is worth noting that although these properties were sold at prices below their market value, they were still sold at this price regardless. Properties will continue to be sold below their market price in future and therefore it is worth considering these in my analysis. Deleting the rows would mean removing a considerable percentage of the dataset.

**Chosen Solution:**

* **No action:** I decided to keep all rows. I will be able to explore the relationship between being sold at market price and below market price in part 3.

**Issue 8:** Dataset contains block purchases

**Issue Description:** As seen in part 1, there are quite a few entries that relate to block purchases of properties, which distorts the distribution of the data that is supposed to represent single property purchases.

**Potential Solutions:**

* **Ignore the issue:** Unfortunately, it would not be possible to be sure all block purchases were identified. However, it would still be helpful to remove as many as possible and reduce the distortion in the distribution.
* **Remove properties identified as duplicates from a new price column:** Using the keywords ‘-‘, ‘inclusive’ and ‘units’ from the first element of each address, I was able to identify properties that were duplicates. I could create a new column of cleaned prices that exclude these properties.

**Chosen Solution:**

* **Remove properties identified as duplicates from a new price column:** I decided to create a new column termed ‘CleanedPrice(€), consisting of the same information as the Price column. I then removed the properties with the above keywords if their price was above one standard deviation above the mean. It was these properties that would cause the most significant distortion in the distribution. The prices for 11 properties were removed from the new column.

**Issue 8:** Prices are not normalised - significant outliers

**Issue Description:** As mentioned in part 1, it appears as though the prices follow a lognormal distribution, a feature generally associated with house price distributions.

**Potential Solutions:**

* **Remove prices that are 3 standard deviations above the mean of each county from the CleanedPrice(€) column:** It would be fair to assume that any price 3 standard deviations above the mean is an outlier, i.e., it is significantly different to the trends observed in the rest of the dataset. It would be necessary to conduct this removal on a county basis as the distribution varies considerably for each. Note that prices would not be deleted from the dataset, but rather from the newly created price column, so that it would include data more appropriate for analysing.
* **Remove log prices that are 3 standard deviations above the mean of each county:** As I observed the distribution to be lognormal, it may be more appropriate to carry out the above approach but first convert the prices to log and then check for prices 2 standard deviations above the mean. I would then convert the prices back to €.
* **Ignore problem and chose an arbitrary max price to focus analysis on**: I could simply choose a max price to focus on, arbitrarily defined. A max price of €1 million would likely still produce interesting results. However, looking at prices above 2 standard deviations above the mean would be more statistically relevant.

**Chosen Solution:**

* **Remove log prices that are 2 standard deviations above the mean of each county:** 167 counties were removed in total from the CleanedPrice(€) column to result in a more normally distributed set of prices. This reduced the standard deviation compared to the raw Price(€) column by 66.4%, which should make for more interesting observations, less impacted by outliers.