Assignment 2: OpenRefine (Airbnb dataset)

Storyline:
Your family is visiting you in Illinois, and you decide to take them to Chicago for a short trip. You wish to give them the best Chicago experience, but the nice hotels in Chicago are just way beyond your budget. Instead, you decide to stay at a Bed & Breakfast (BnB). You know that in order to choose a perfect BnB, you have to scrutinize and carefully inspect the listings. Therefore, you gather the latest Airbnb Chicago listing dataset, and start to put your OpenRefine skills that you learned in class into practice.

Your ultimate goal is: to clean the dataset to a certain, acceptable level, so that it is ready to be used for further data analysis (not just for your family).

STEP 1: (if you haven't yet:) Download and install OpenRefine **3.4.1**
https://openrefine.org/download.html
Note: While there are newer versions available, we suggest to use that version to avoid incompatibility issues with the grading tool.

STEP 2: Create Project → Load the **airbnb_dirty.csv** into OpenRefine. Make sure it is loaded in CSV format. **Uncheck** the default option '**Trim leading & trailing whitespace from strings**'. (In order to record all cleaning steps in the OpenRefine operation history/recipe, we ask you to trim whitespaces not automatically on import, but rather explicitly in the UI.)

STEP 3: Complete the data cleaning tasks below.
IMPORTANT NOTE: **READ THIS FIRST**

- For the purpose of grading and track changes, do **NOT** do any edits on the ***ids*** column.
- Do **NOT** open the **airbnb_dirty.csv** file in any other application. Otherwise your final clean dataset may have some non-standard ("weird") characters introduced by the spreadsheet (or other) software.
- Execute the tasks in sequential order, step by step. Do **NOT** jump steps.

1. **TRIM and COLLAPSE WHITESPACES**.
   - Description: It is very common to see unnecessary whitespaces in datasets. A lot of times whitespaces are hidden at the beginning or the end of a string, and sometimes they are hidden as two or more consecutive whitespaces in a phrase. Here's what you can do to clean up (or trim) unnecessary whitespaces.
   - Tasks:
     - **Trim all the leading and trailing whitespaces** in ALL columns that are texts (strings). This includes the **name**, **host_name**, **neighbourhood**, and **room_type** columns.
     - **Collapse consecutive whitespaces** in ALL columns that are texts (strings). This includes the **name**, **host_name**, **neighbourhood**, and **room_type** columns.
     - Note that these two actions are iterative, meaning you might have to do them AGAIN after you did other following operations.

2. **NUMBER TYPES**.
   - Description: Incorrect data types are almost always the next thing you inspect in a dataset. Usually numeric data will be seen (or converted to) as text data in a lot of platforms. To correct these, you can do the following:
   - Tasks:
     - Transform all columns that should be in numeric form to type number.
     - This includes the **host_id, latitude, longitude, price, mininum_nights, number_of_reviews, reviews_per_month, calculated_host_listings_count**, and **availaibility_365** columns.
     - Note that whatever you have converted to a number will be shown in green.

3. **Titlecase, UPPERCASE, lowercase**
   - Description: Sometimes you want your text data all in lowercase, sometimes uppercase. When you're going through the Airbnb dataset, you noticed that most of the values in the **neighbourhood** column are using title cases (e.g. **L**ogan **S**quare), but some are not.
   - Tasks:
     - **Add** a new column based on the **neighbourhood** column, name the new column as **neighbourhood_case**.
       - **Note the (British) spelling of neighbourhood_case!**
     - Transform the **neighbourhood_case** column to **Titlecase**.

4. **FACETS**.
   - Description: Faceting is a useful technique when cleaning data. Depending on the data type, there might be numeric facets, text facets, or scatterplot facets that you can use for your dataset. OpenRefine provides facet features to profile your data, i.e., you get a better overview of the data and improve its consistency based on your findings.
   - Tasks:
     - **Add** a new column based on the **neighbourhood_case**, name the new column as **neighbourhood_loop**.

    - **again: neighbourhood_loop, not neighborhood_loop**
- Using the **neighbourhood_loop** column, create a **text facet**. You should notice a box ('facet') that appears on the left of your user interface. Sort it by **count**.
- In the original dataset, the Loop is spelled with diacritic characters "Lóóp". In your reference (dirty) dataset these special characters appear as question marks "?". Replace the "?" placeholders by changing **L??p** to **Loop** using facets.
- You should close ("x out of") the **neighbourhood_loop** text facet after you have done the above steps (to avoid subsequent operations to only use the facet).

5. **CLUSTERING to Normalize ("Canonicalize") Names.**
   - Description: Clustering is used to group similar items together. In OpenRefine we can cluster similar text (often: names) using different similarity measures and key functions. Often we should have the same string value but due to spelling variations, typos, or punctuation mark differences, they values look different.
   - Tasks:
     - **Add** a new column based on the **neighbourhood_loop** column, and name the new column as **neighbourhood_cluster**.
     - Using the **neighbourhood_cluster** column, create a text facet.
     - On the **text facet** box for **neighbourhood_cluster**, click **Cluster**.
     - You'll immediately see many different spellings of **O'Hare**. Please use the spelling "O**'H**are" with the apostrophe (**'**), then tick **Merge?**, then click **Merge Selected and Recluster**.
     - Experiment using different combinations of Method and Key functions and fix other clusters. **Hint:** you should be able to unify "West Garfield Park" as well, but be careful, you won't want to mix up "East Garfield Park" with "West Garfield Park"!
     - Again, make sure to **close** the **neighbourhood_cluster** text facet window after you're done with this task.

     (**REMARK**: Quotes come in many different forms. There is an apostrophe (') which is an ASCII character but there is also an open single quote (‘) and closing single quote (’), which are not ASCII. The original dataset had all three of these. For your convenience, we have replaced non-ASCII quotes with ? in the dirty dataset, so you will also see O?hare as one of the spellings. You can learn more about the different typography of quotes here.)

6. **SPLIT COLUMNS.**
   - Description: In the **host_name** column, many cells include two (or more) person names joined by "**And**". For instance, there's an instance of "Michael And Veronica".
   - Tasks:
     - **Split** these joint hostnames into separate columns so each of the cells only contains one name.
     - However, you would not want to split names such as **And**rea, **And**y, or **And**rew!
     - To achieve this, you will need to use **regular expressions** when you split columns (remember to tick the 'regular expression' box).

- Note that you should **keep** (not replace) the original **host_name** column (so **uncheck** the 'remove this column' box in the split column window).

7. **DELETE IRRELEVANT COLUMNS.** You noticed that there are almost no values on the **neighbourhood_group** column, and you decided that this is an irrelevant column for further analysis. Please **delete** this whole column.

8. **TO DATE**. The **last_review** column looks like it is in a date format. For your task, transform it into ISO standard date form.

9. **GREL**.

9.1 Although the **To date** transformation makes your date format into the ISO compliant YYYY-MM-DD format, it also contains time information that we don't have and don't need. Clean this column by applying GREL: toString and toDate functions.

- Tasks:
  - **Add** a new column based on the **last_review** column first. Enter **last_review_timeless** for the new column name.
  - On the **last_review_timeless** column do the **Operations: Edit cells** → **Transform** → **toString(toDate(value),"yyyy-MM-dd")**
  - Now it should look like the ISO standard date format without the time information.

9.2 For the **name** column,

- Tasks:
  - Add a new column based on the **name** column and name the new column as **name_grel**. Create a **text facet** on the **name_grel** column to see the distribution and how messy this column is.
  - Using GREL (and regular expression as needed), remove **the outermost parentheses** in each name, but not the inner ones. For example, the desired outcome looks like this:

```
Original: (Lincoln Park (Oasis) - Unit 2 ONLY)
Cleaned: Lincoln Park (Oasis) - Unit 2 ONLY
```

Hint: search on OpenRefine recipes. https://github.com/OpenRefine/OpenRefine/wiki/Recipes
You might also want to refer back to the regular expression notes on how to express the beginning and ending anchors. Also note that to use GREL, you might have to add outermost slashes in order to effectively transform using regex (e.g. / abc+ /)

  - Also clean **the exclamation marks** (!) and **the asterisks** (*) as follows:
    - **Create** a new column based on the **name_grel** column and name it as **name_grel_star**
    - Using GREL to remove all the exclamation marks and the asterisks as well.
    - Your desired outcome should like this:

*Original: ******* Luxury in Chicago**!!!** 2BR/ 2Ba / Parking / *BBQ****!!***
Cleaned: Luxury in Chicago 2BR/ 2Ba / Parking / BBQ

- Close the text facet for the **name_grel** column after the tasks.

10. **ADVANCED FACETS**. Now you know the power of using facets for column operations, explore the use of numeric facets to clean up your dataset.
    - Task 1:
        - Create a **numeric facet** for the **price** column
        - You notice there are a lot of unreasonable prices for a listing. Out of curiosity, you want to inspect those that are **$5000 and above** per night. To take note of these outrageous listings, you can do this:
        - Adjust the range of the numeric facets to $5000 and up. You will see the table is being filtered according to the records in this range. Based on the **price** column, add a new column **price_crazy**, and in the Expression box, enter "**1**".
        - Remove the numeric facet for **price** after this task.
        - You should notice that only the listings that are $5000 and above have been marked with '1' in the price_crazy column.

    - Task 2:
        - Create a numeric facet for the **minimum_nights** column
        - After you have adjusted the range of the numeric facets to 300 nights and above. You will see the table is being filtered according to the records in this range. Based on the **minimum_nights** column, add a new column **minimum_nights_long**, and in the Expression box, enter "**1**".
        - Remove the numeric facet for minimum_nights after this task.

11. Refer back to the first task and **TRIM the leading and trailing whitespaces**, as well as **COLLAPSING consecutive whitespaces** for all columns that have a string data type for one last time (in case some extraneous whitespace has been introduced in other steps). This includes the **name_grel, name_grel_star, host_name 1, host_name 2, neighbourhood_case, neighbourhood_loop,** and **neighbourhood_cluster** columns.

There are still many "messy" cells in this dataset (e.g., "weird", i.e., non-standard, characters in the name column), but you think it looks "clean enough" for your purposes. You've completed the tasks, so now it's time to save your project and move on with life. To push to the finish line, please complete Step 4.

STEP 4: Please refer to the Submission & Autograding guide for instructions on how, what, and where to submit the assignment files.

Congratulations! Hope you have a nice stay with your family in Chicago!