Assignment 2 OpenRefine with Airbnb dataset

Storyline:
Your family is visiting you in Illinois for the very first time, and you decide to take them to Chicago for a short trip. You wish to give them the best Chicago experience, but the 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 gathered the 2018 Airbnb Chicago listing dataset, and started to put your OpenRefine skills that you've learned in class into practice.

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

STEP 1: (if you haven't yet:) Download and install **OpenRefine 3.1. (Please do not use any other version beside 3.1, otherwise it will affect the autograding results).**
https://github.com/OpenRefine/OpenRefine/releases/tag/3.1

STEP 2: Load the **airbnb_dirty.csv** into OpenRefine. Make sure it is loaded as CSV format. Click the blank box next to 'Encoding', a pop up window will show up. Select the encoding to "**ISO-8859-1**". Click Create Project on the upper right corner after everything is set up.

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 ***id*** column
- We suggest directly importing the airbnb_dirty.csv dataset into OpenRefine. Do **NOT** open the airbnb_dirty.csv file, nor you final cleaned dataset (after exporting from OpenRefine) using any spreadsheet software - this may create some weird character encodings in the spreadsheet
- Do the tasks in sequential order, step by step. Do **NOT** jump steps.

1. **TRIM and COLLAPSE WHITE SPACES**.
   - Description: It is very common to see unnecessary white spaces in datasets. A lot of times white spaces are hidden at the beginning or the end of a string, and sometimes they are hidden as two consecutive white spaces in a phrase. Here's what you can do to help clean up white spaces.
   - Tasks:
     - Trim all the leading and trailing white spaces in ALL columns that are texts (strings). This includes the **name**, **host_name**, **neighbourhood**, and **room_type** columns.

- Collapse consecutive white spaces 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**.
   - Description: Incorrect data types is almost always the second 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 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 number will be shown in green.

3. **CASES**.
   - Description: Sometimes you want your data all in lower cases, sometimes upper. When you're going through the Airbnb dataset, you noticed that most of the neighbourhood 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**.
     - Transform the **neighbourhood_case** column to title case.

4. **FACETS**.
   - Description: Faceting is also a useful technique to clean up datasets. According to your datatypes, there might be numeric facets, text facets, or scatterplot facets in your dataset. OpenRefine provides facet feature which can help to get an overview of data and enhance consistency of data.
   - Tasks:
     - Add a new column based on the **neighbourhood_case**, name the new column as **neighbourhood_loop**.
     - Using the **neighbourhood_loop** column, create a text facet. You should notice a box ('facet') appeared right on the left of your interface. Sort it by **count**.
     - In the original dataset, the Loop is spelled with diacritic characters "Lóóp". We have replaced these special characters with ? in the dirty dataset. Fix these placeholder ? from **L??p** to **Loop** using facets.
     - You can close (remove) the **neighbourhood_loop** text fact after you are done with all the above instructions.

5. **CLUSTERING.**
   - Description: Clustering helps us group similar texts together. In the case of data cleaning, in OpenRefine we can cluster similar text together based on different methods and key functions. Sometimes we have the same word but due to misspellings, typos, or punctuation mark differences, they 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 **OHare**. Please use the spelling "O**'H**are" with the apostrophe (**'**), 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".
     - Make sure to close (remove) the **neighbourhood_cluster** text facet window after you are done with this task.

     (**NOTE**: Quotes have different typographies. There is apostrophe (') which is an ASCII character but there are open single quote (‘) and closing single quote (’) too 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.**
   - Descriptions: In the **host_name** column, a lot of cells include two (or more) people's name joint by "**And**". For instance, there's an instance of "Michael And Veronica".
   - Tasks:
     - **Split** these joint host names into separate columns so each of the cell only contain 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 expression** when you split columns (remember to tick the 'regular expression' box).
     - And you should keep the original **host_name** column (tick OFF the 'remove this column' box in the split column window).

7. **DELETE IRRELEVANT COLUMN.** 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.

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 need. You decided to clean this column on your own by applying some regular expressions.

- 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, 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** (*) by:
        - Create a new column based on the **name_grel** column and name it as **name_grel_star**
        - Using GREL to remove all the exclatmation 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, explore the use of numeric facets to clean up your datasets.

- Task 1:
    - Create a numeric facet for the **price** column
    - You noticed there are a lot of unreasonable pricing for a listing. 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 up, 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, 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 the columns that are strings for one last time. 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 a lot of messy cells in this dataset (e.g.weird characters in the name column), but you think it looks relatively clean now compared to the original dataset. You've completed the tasks, now it's time to save your projects and move on with life. To push to the finish line, please complete Step 4.

STEP 4: Please refer to the Autograding_instructions for instructions on how and what to submit the assignment files.

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