Assignment 2 OpenRefine with Airbnb dataset

This assignment will not be run on Coursera infrastructure but on a different server run by course staff. More details will be shared on Piazza.

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 latest 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 the latest version of OpenRefine 3.1

- http://openrefine.org/download.html

STEP 2: Create Project → Load the **airbnb_dirty.csv** into OpenRefine. Make sure it is loaded as CSV format.

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 the tasks in sequential order, step by step. Do **NOT** jump steps.

1. **TRIM and COLLAPSE WHITE SPACES**. 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.
    - 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**. 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:
    - 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**. 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. To fix this:
    - Add a new column based on the **neighbourhood** column first. Operations: Edit column → Add column based on this column → Enter **neighbourhood_case** for the new column name.
    - Transform the **neighbourhood_case** column to title case.

4. **FACETS**. 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. To explore the use of facets to clean datasets, you will:
    - Add a new column based on the **neighbourhood_case** column first. Operations: Edit column → Add column based on this column → Enter **neighbourhood_loop** for the new column name.
    - 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**.
    - As you scroll down the facet, you probably notice something weird with the neighbourhood "Loop". In the original dataset, the Loop is spelled with diacritic characters "Lóóp". We have replaced these special characters with ? in the dirty dataset. Your job is to replace these placeholder ? with ASCII equivalent of original character. Please edit **L??p** to **Loop** using facets.
    - You can close (remove) the **neighbourhood_loop** text fact after you done all the above instructions.

5. **CLUSTERING.** Clustering helps us group similar items 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. To help with these situations, you can:
    - Add a new column based on the **neighbourhood_loop** column first. Operations: Edit column → Add column based on this column → Enter **neighbourhood_cluster** for the new column name.
    - 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**. Is it O'hare or Ohare? Are they the same thing? Let's assume they all refer to the same O'hare, please use the spelling "O**'H**are" with the apostrophe ('), tick merge, then click Merge Selected and Recluster.
- **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. These spellings also have to be replaced with O'Hare. You can learn more about the different typography of quotes here.
- **NOTE:** Other special non-ASCII characters were also replaced with ?
- 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".
- After you have successfully merged the clusters, make sure to close the clustering window, and you can also close (remove) the **neighbourhood_cluster** text facet window.

6. **SPLIT COLUMNS.** You noticed that 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". For your task, you want to **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**. You noticed that the **last_review** column looks like it is in a date format. For your task, you want to transform it into ISO standard date.

9. **GREL**.
   9.1 You also noticed that although the **To date** transformation makes your date format into the ISO compliant YYYY-MM-DD format, it also contains time information that you don't really want. You decided to clean this column on your own by applying some regular expressions. To fix it, try the following:

- 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, first, also add a new column based on the name column. Then name the new column **name_grel**. Create a **text facet** on the **name_grel** column to see the distribution of the name column to have a sense of how messy this column is.

a. 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+ **/)**

b. You also think that **the exclamation marks** (!) and **the asterisks** (*) in the name column make everything look very messy. Even the ones that are in the middle of the text. To fix this:
    - Create a new column based on the **name_grel** column and name it **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

c. After all the above operations, you can close the text facet for the **name_grel** column now.

10. **ADVANCED FACETS**. Now you know the power of using facets, you want to explore the use of numeric facets to clean up your datasets.
    - 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:

        - After you have adjusted 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 you have done the above operation.
    - You should notice that only the listings that are $5000 and above have been marked with '1' in the price_crazy column.

    - Similarly, 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 you have done the above operation.

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: Submit openrefine operation history and clean file. Submission instructions will be shared via Piazza.

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