**Explore COVID-19 confirmed cases**

**Source of COVID19 dataset:** <https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset#COVID19_open_line_list.csv>

**Supplemental file:**

<https://uofi.box.com/s/9jxnvy1xvmz5c9d7el6ukph3caxpaizm>

**Introduction and Overview**

Our project is about the COVID-19 confirmed cases dataset. The data set is COVID-19 confirmed cases from Kaggle. Our focus is to enhance the data quality of the original data by cleaning the data via Openrefine and Python first, and then we aim to find if there are certain patterns in the infection of COVID-19.

The situation of COVID-19 spread in the world is very serious currently. There is massive news and suggestions coming from different sources every day. However, some of the information is true while some other is unauthentic. People in anxiety are very likely to feel panic about some fake news. We try to analyze the existing data of confirmed cases and to verify whether some hypotheses are true or not.

We have three hypotheses here. First is that people with travel history are more likely to have the disease. Second is that all the confirmed cases have common attributes. The third is elderly people are more likely to have the disease.

**Initial Assessment of the dataset and Use cases**

The COVID-19 dataset includes 33 columns and 13479 rows. The dataset shows detailed personal information of each patient affected by coronavirus, like personal profile, living area, activity area, symptom, recorded date, resources origin, etc. The goal of analyzing this data is to verify the validity of existing hypotheses based on current cases.

**ID:** Identical numbers for each COVID-19 patience.

**Age:** patient age

**Sex:** patient sex, female/ male/ unknown

**City:** Hometown city of patient

**Province:** Living province of patient

**County:** Living country of patient

**Wuhan(0)\_not\_in\_wuhan(1):** whether the patient was living in Wuhan province, 0 for yes and 1 for no

**Latitude/longitude:** Precise location of patient

**Geo\_resolution:** Filler

**Date\_onset\_symptom:** The date of showing coronavirus symptoms

**date\_admission\_hospital:** The date of being sent to the hospital

**date\_confirmation:** The date of being confirmed as COVID-19

**symptom:** clinical symptoms of coronavirus

**lives\_in\_wuhan:** whether the patience has lived in Wuhan

**travel\_history\_dates:** the date of previous travel

**travel\_history\_location:** the city name of travel history, including pass-by city

**reported\_market\_exposure:** whether the patient has exposed to coronavirus break out market

**additional\_information:** extra information about the patient travel history if available

**chronic\_disease\_binary:** chronic disease if available

**source:** link where the data come from

**sequence\_available:** sequence if available

**outcome:** the clinical condition of patient: death/discharge/recover etc.

**date\_death\_or\_discharge:** specific date of death or discharge

**notes\_for\_discussion:** additional information left for further discussion about the patience

**location:** living area if patients live outside of China

**admin1/2/3:** location information of admin

**country\_new:** country information

**admin\_id:** ID number of admin

**data\_moderate/ column 1 - 12:** blank columns

At this pandemic time in the US, previous COVID-19 data can help researchers to find certain infection patterns of the virus and provide related plans to fight against it. So far, there are several conjectures of the infection rate of coronavirus and personal physical features. Our goal is to verify the validity of the existing conjectures. First, since the initial outbreak of pneumonia in China started in Wuhan and the most serious illness has concentrated in Wuhan, we proposed that people with travel history, especially those who have been to Wuhan are more likely to be infected with the coronavirus. Secondly, doctors notice that almost all the patients share similar symptoms like cough and fever and we would like to test the find out the most common symptoms among the COVID-19 patients. Lastly, we would like to check the relationship between age and infection rate. In other words, whether older people are more likely to have the disease based on the current dataset.

At first glance, the dataset has been clean enough without messy words or gibberish and is good enough for us to have a clear overview and understanding of the whole dataset. But there do exist data quality issues that impede our analysis. Above all, the format problems. In the text column, there are white spaces in the leading and trailing. Besides, the numerical and date columns don’t have uniform formats, which will impede further analysis. Besides, in the symptom column, the descriptions are a little messy since they don’t have a uniform format and exist repetition. According to our use cases, we will find out the patterns existing in the patient’s symptoms thus what we need to do is classify symptoms of all patience into multiple groups as well as merge the same symptoms but written in different order into the same groups. When we analyze the age column, we also found some problems. Some of the records are too vague and only offer a range of ages. For example, a patient is at the age between 20 to 80. We cannot group this record by any age since the range is too large and inaccurate. In addition, we noticed that there are many redundant columns that are not related to our topic so we need to remove them.

**Data Cleaning methods and process​**

* **OpenRefine**

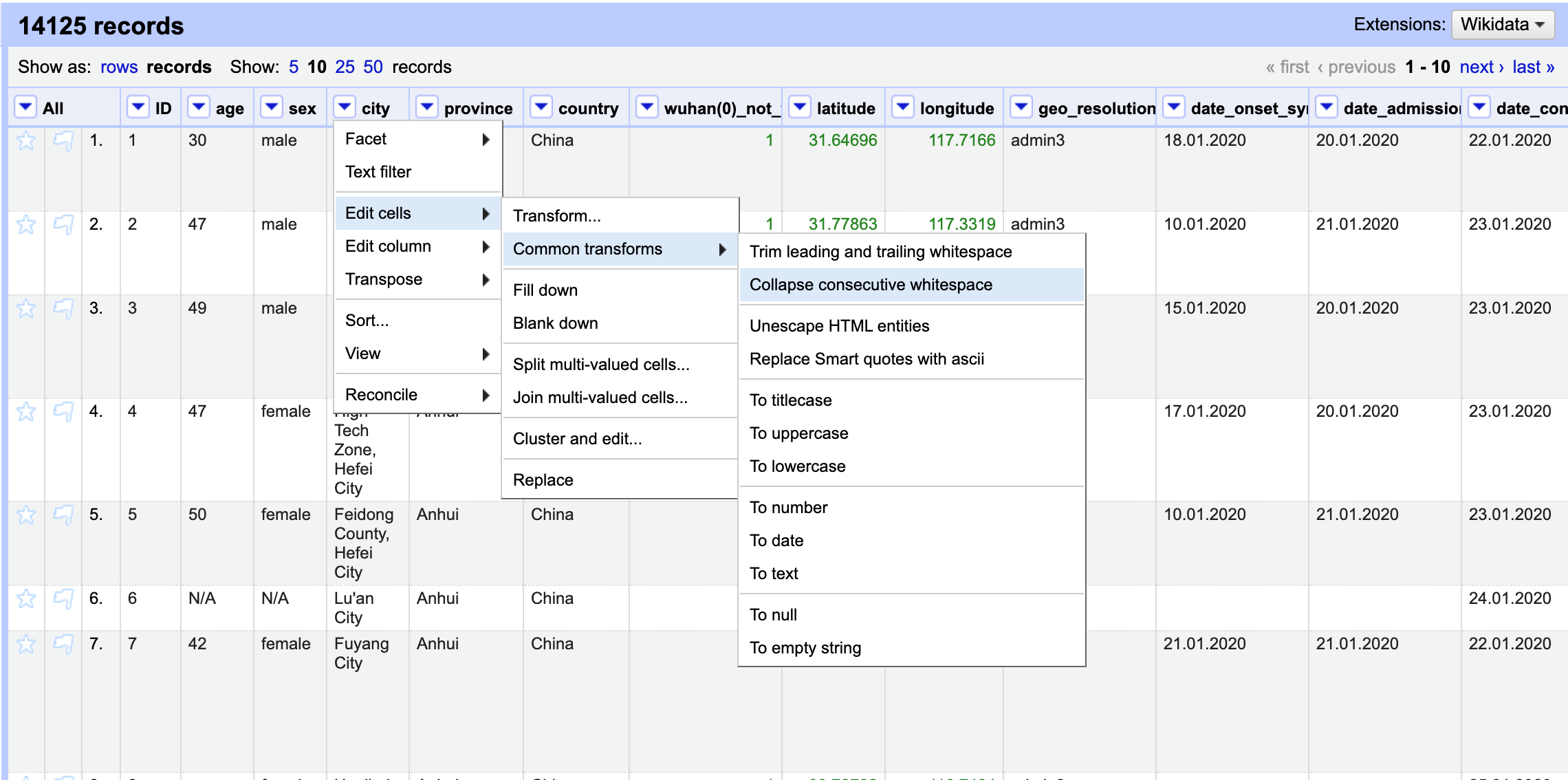
We used OpenRefine to do the initial data cleaning and used Python packages including Pandas and Numpy to do further data visualization to test our hypothesis. We chose OpenRefine since the main task of our data cleaning will transform the messy format into a unified format and textual facet, and OpenRefine provides useful functions like to numeric and text facet that exactly fit our goal. Besides, the cluster function of text facet is also suitable for us to merge repetitive textual information into the same groups. It is especially useful to verify our second hypothesis through the cluster and merge.

Data cleaning process using OpenRefine:

1. Import dataset into OpenRefine

2. Trim leading and trailing white spaces and collapse white spaces on City, Province, Country, Symptom columns. (fig. 1)

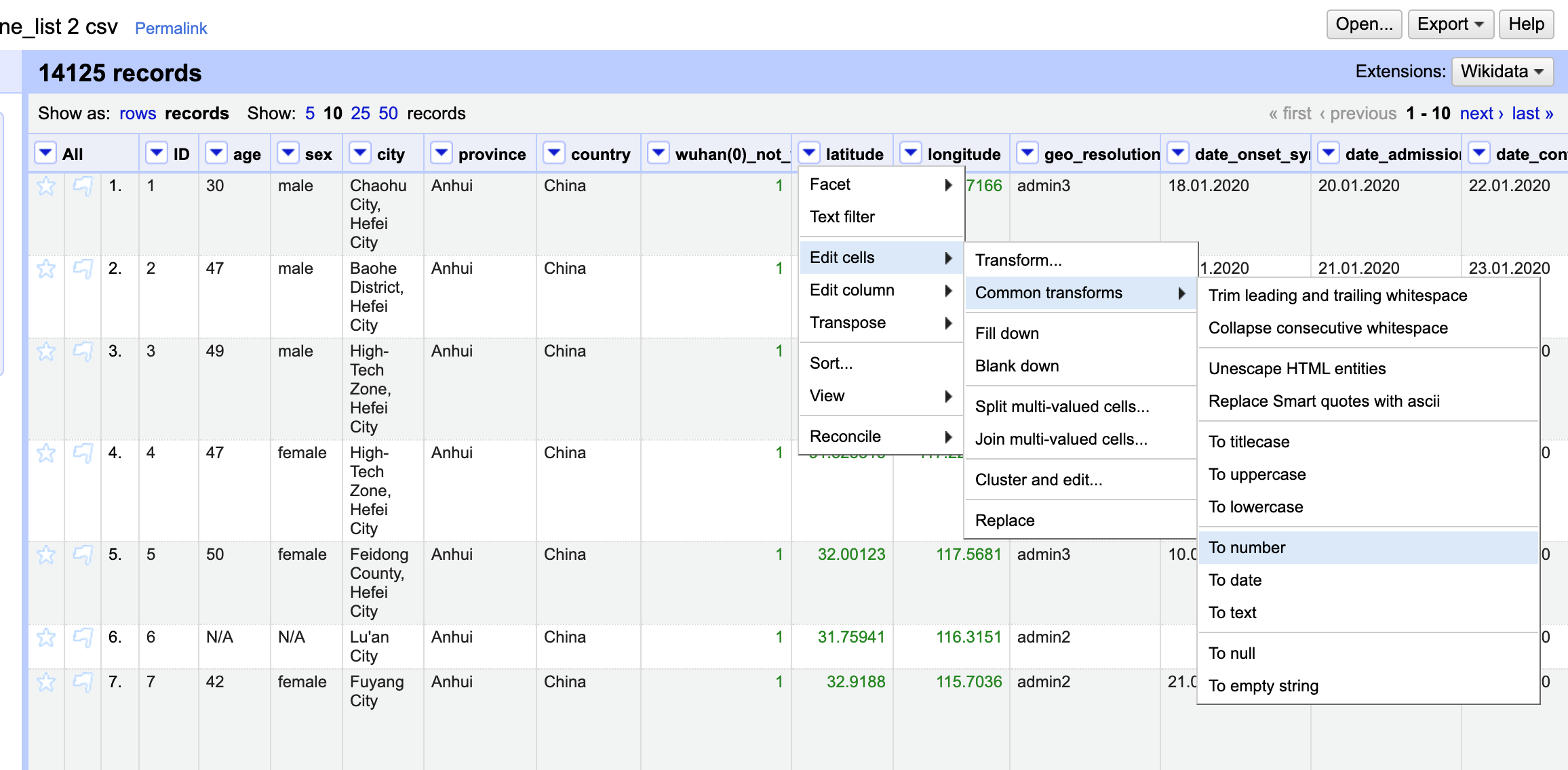
To avoid the effect of the white spaces, we trim all the white spaces before our next cleaning steps.



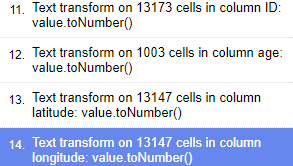
(fig 1: trimming white spaces)

3. To number on ID, age, latitude, longitude columns. (fig 2, fig 3)

The formats of these four columns are text format. Since we have further calculations according to ID and age columns, we transform their formats to number.



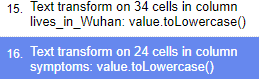
(fig 2: process to transform records format to number)



(fig 3: record of number transformation)

4. To lowercase on lives\_in\_wuhan, symptom columns. (fig 5)

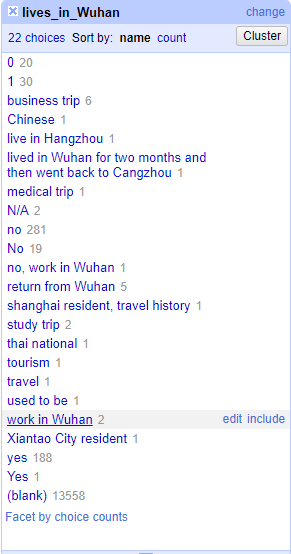
We found the capital form and the lowercase form is not consistent in these two columns. This may cause some confusion in the further group steps. So we transformed all the records into uniformed lowercase.



(fig 5: record of lowercase transformation)

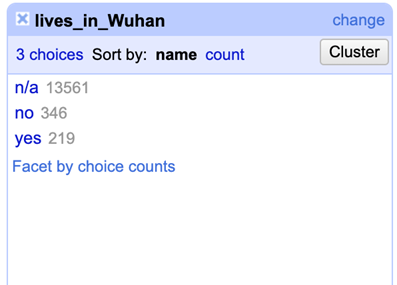
5. Text facet and cluster on lives\_in\_wuhan (fig 6, fig 7)

There are several types of reasons that people live in Wuhan, but since our analysis only focus on whether the patient has travel history, we can simplify the answer to just yes and no.



(fig 6: text facet of column live\_in\_wuhan)

As a result, we classified different reasons which people not living in Wuhan as No, and living in Wuhan as Yes. If the information is not available, we assign n/a to it.



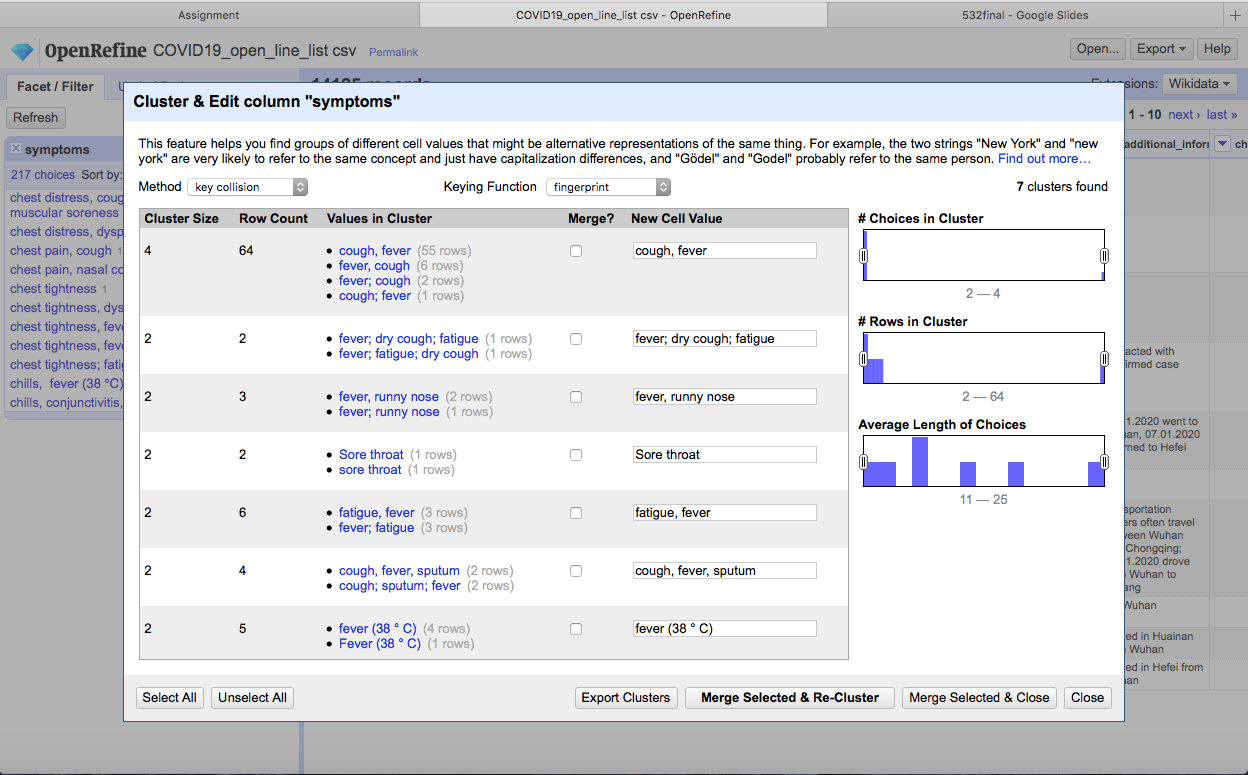
(fig 7: final result of clustering column live\_in\_wuhan)

This step helps us to explore our first hypothesis, we can use the sorted result to check whether people with travel history to Wuhan are more likely to infect the disease.

6. Text facet and cluster on symptom (fig 8, fig 9, fig 10)

To address our second hypothesis, the cleaning of symptoms is necessary and is one of our most important steps.

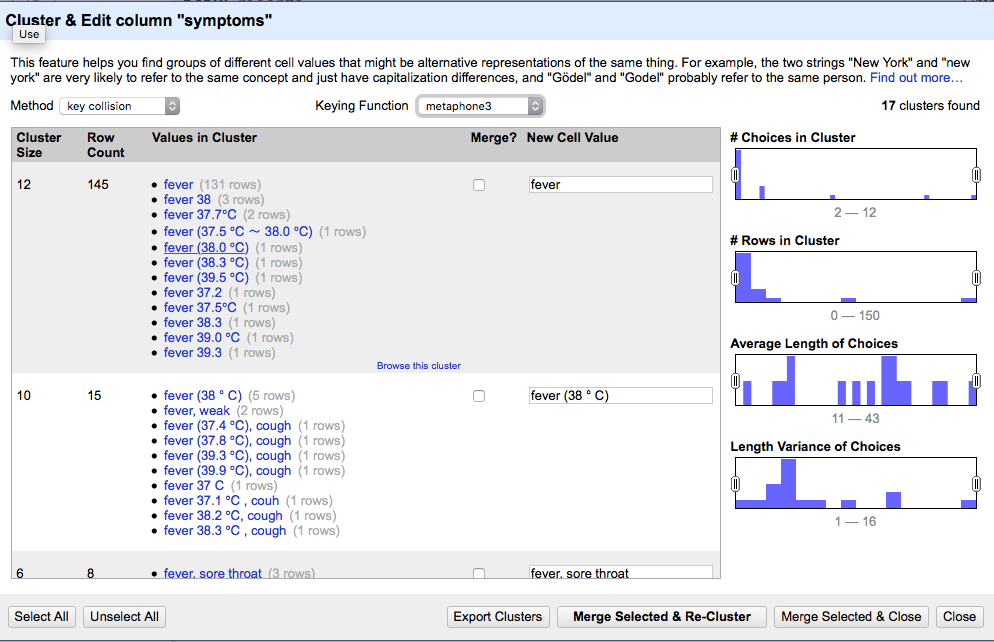
There are two main tasks in dealing with the symptom column. First, we merged the same symptom but wrote in different orders (eg. cough fever & fever cough) into the same group using different functions like megaphone 3 and fingerpoint.



(fig 8: clustering column symptoms)

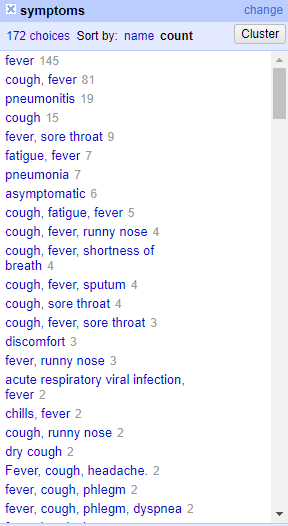
It helps us to calculate the appearing probability of different symptoms without duplication, as well as find out the most common symptoms among all the patient.

We also classified similar symptoms into same cluster (eg. fever 38℃ & fever 39℃).



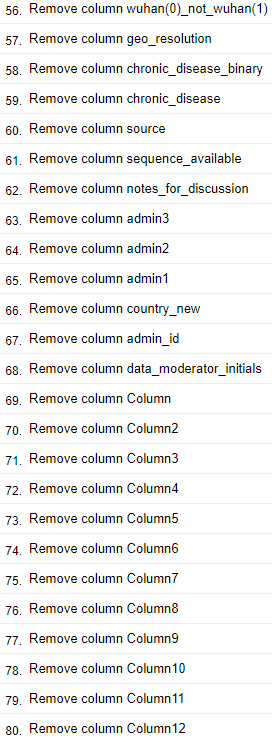
(fig 9: clustering column symptoms)

As a result, the symptom description is relatively concise and uniform without significant repetition now, though further cleaning is still needed.



(fig 10: final result of clustering column symptoms)

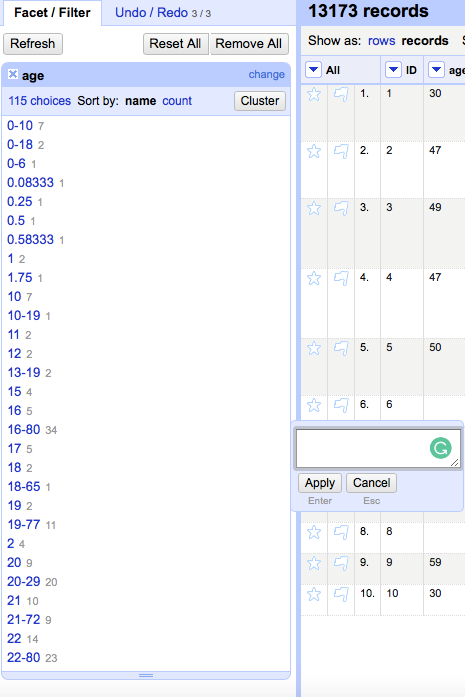
7. Remove unnecessary and blank columns on wuhan(0)\_not\_wuhan(1), geo resolution, source, note\_for\_disccusion, admin\_id, data\_moderator\_ initials and other empty columns. The information of these columns is not related to our goals. (fig 11)



(fig 11: records of removing unnecessary columns)

8. Remove abnormal age range on age column (fig 12)

We noticed that there are some weird float ages so what we did is get rid of these ages and replace them with integer numbers. Besides, the age range is not helping us to do further visualization, so instead of keeping these ranges, we chose one integer to represent the range. For example, for the range of 10 - 19, we kept the records as 11 since we group ages by the interval of 10. The records between 10 and 19 will cluster in one group in the further cleaning. Meanwhile, we remove some abnormal age ranges like 20 - 80. This range is too large so that we cannot group it by any set.



(fig 12: clustering and edit the age column)

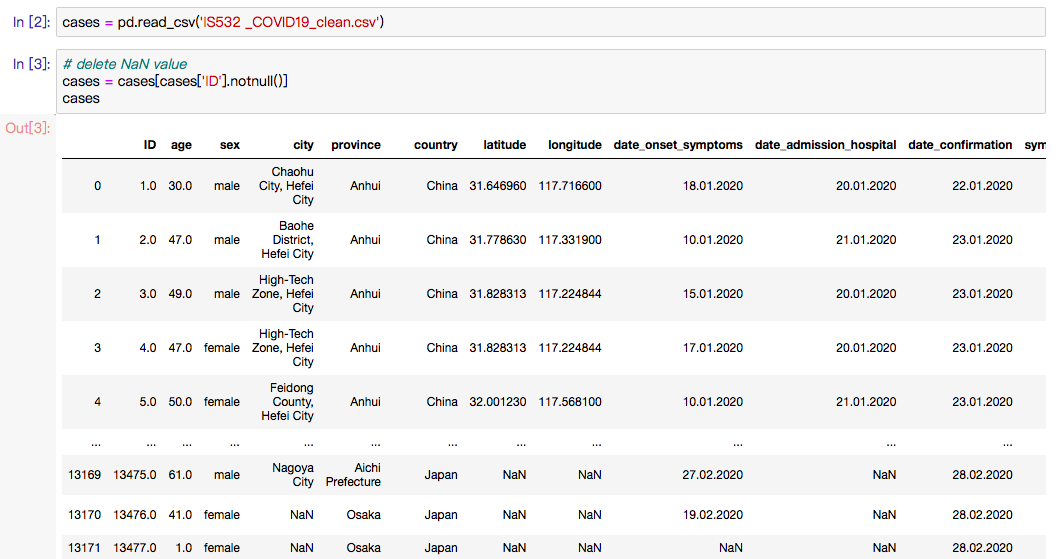
Classifying the age range paves the way to test our third hypothesis that older people are more likely to infect the COVID-19. Now we have an initial clear age data and next step we will design a balanced age distribution based on existing data to find out the age range that has the largest number of people infected.

* Python

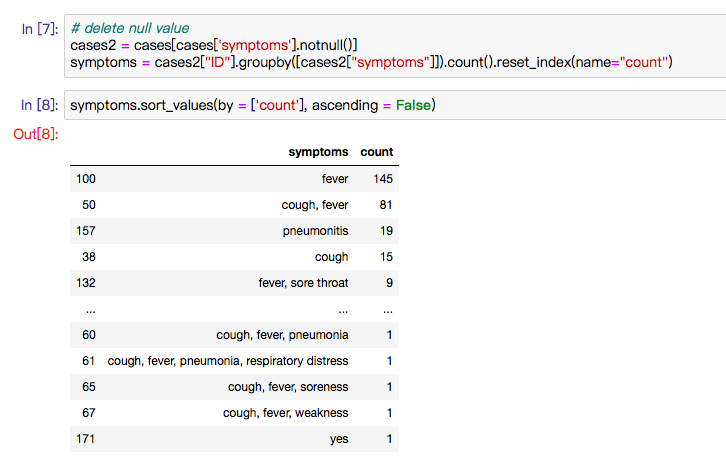
The original file is not complete and contains a lot of null values. The null values are unordered. If we delete or fill up the null values uniformly, the information will lose. For example, one record has complete data of age and location but has a null value of symptoms. If we delete it in the original file, we can get a good result when we analyze symptoms but an inaccurate result when we analyze age and location. To keep the completeness of the original file, we only did the basic clean on the OpenRefine and did a further clean in python.

At first, we delete the rows without id in the original file. ID is the primary key in this table. This is unique and necessary. So the results without id cannot be counted in any analysis (fig 13).

Then we selected specific columns in python according to each hypothesis, so that we can delete the null values and clean the selected data again while not affecting the file itself (fig 14). For example, when we analyze hypothesis 1, people who live in Wuhan are more likely to have the disease, we need to learn the data in the column 'lives\_in\_Wuhan'. So we extract this column out and delete the null values by pandas package before visualizing. Pandas package enables users to extract data and show the results as a new dataframe, which is independent of the original file. This operation will not affect the original file so that when we analyze another hypothesis, we can still ensure the accuracy of the results. The complete Jupyter notebook is attached in the end.

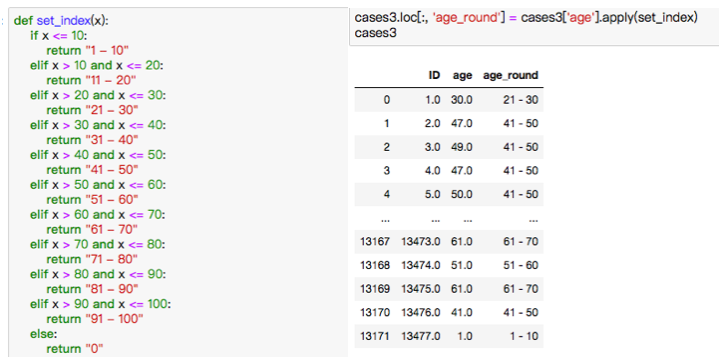


(fig 13: the process to remove the null primary key before further cleaning. It will be saved as an original file.)



(fig 14: the cleaning process after extracted out according to specific needs.)

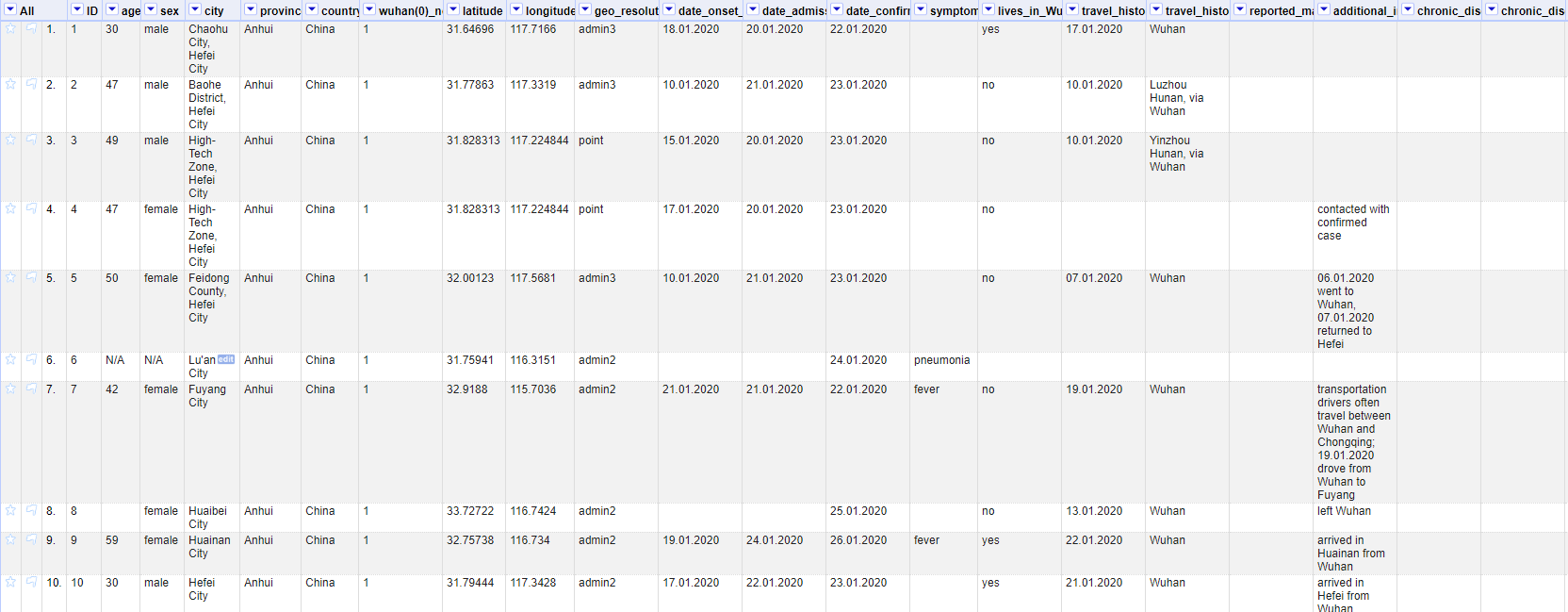
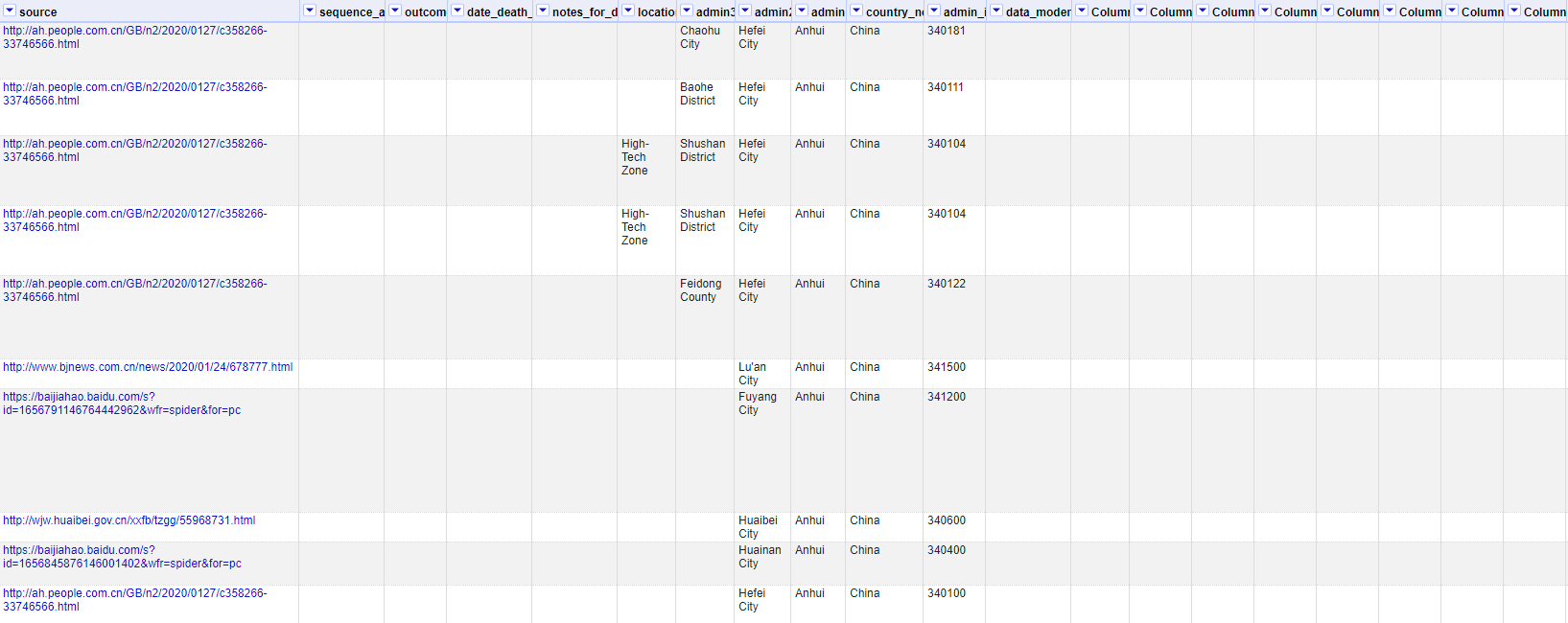
Also, some advanced group and cluster processes were done in python. We regrouped age by function in this part. There are too many ages from 0 to 100. So we try to simplify the records as 10 ranges (fig 15). This step made the data cleaner and is easily analyzed.



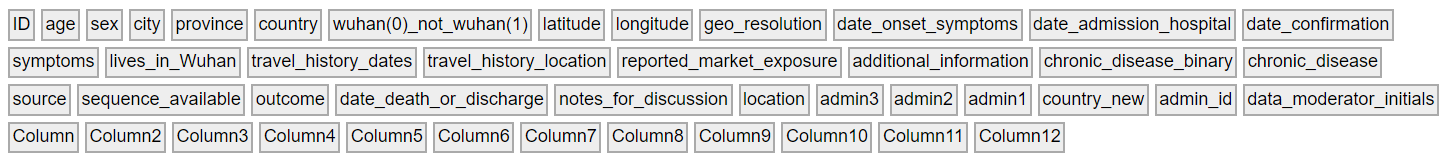
(fig 15: advanced group process)

**Results**

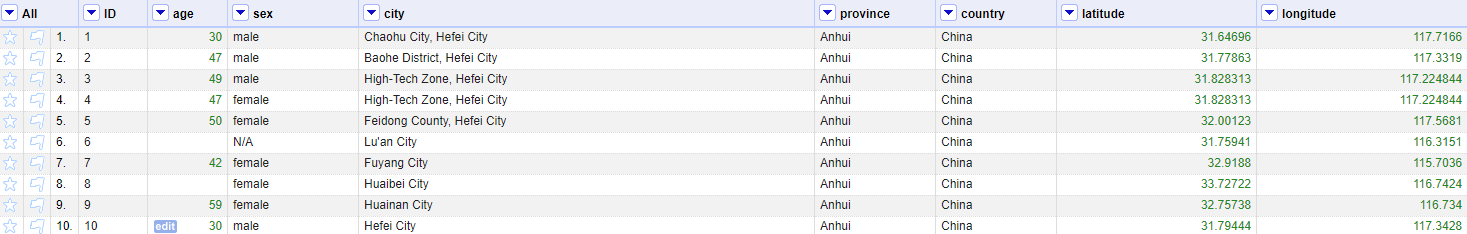
* Showcase the ​overall changes  
  Before cleaning, the dataset has 33 columns and 13479 rows with all the detailed COVID-19 information (fig 16, fig 17).

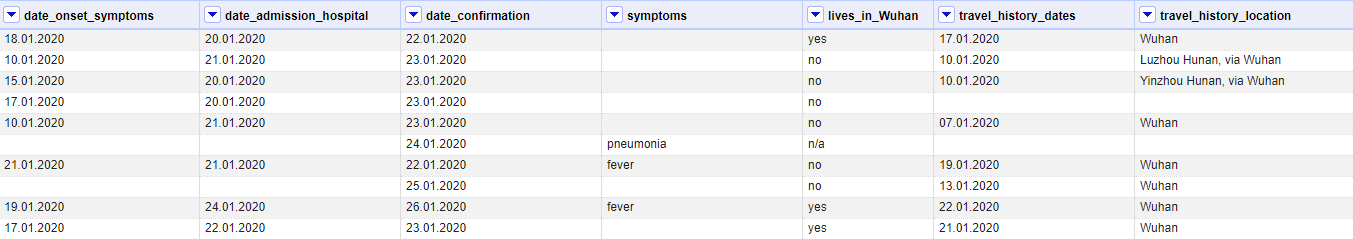


(fig 16: show all the columns in OpenRefine before cleaning)

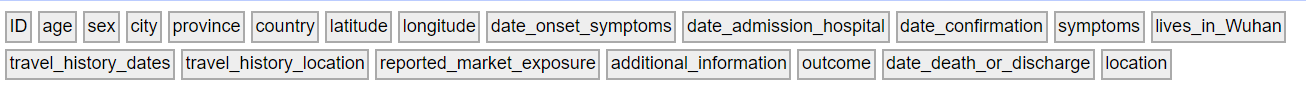


(fig 17: all the columns’ names before cleaning)

After data cleaning through removing the unnecessary columns and resorting, now we have a concise dataset with 13173 rows and 20 columns ready for further analysis (fig 18, fig 19).

(fig 18: show all the columns in OpenRefine after cleaning)

We only kept the necessary columns related to patient personal physical and geographical information and related to verify our hypothesis.

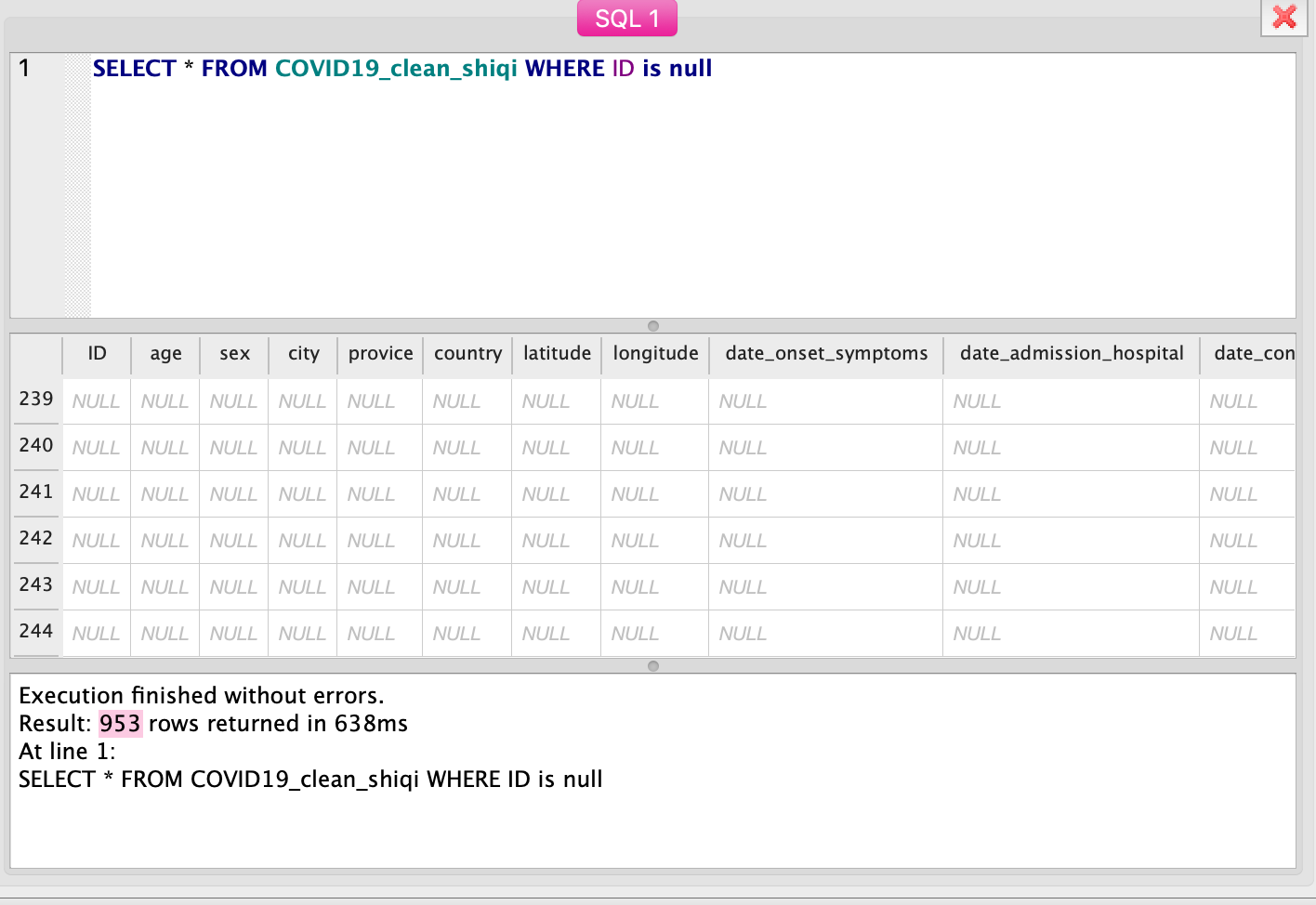


(fig 19: all the columns’ names after cleaning)

* IC violations checks

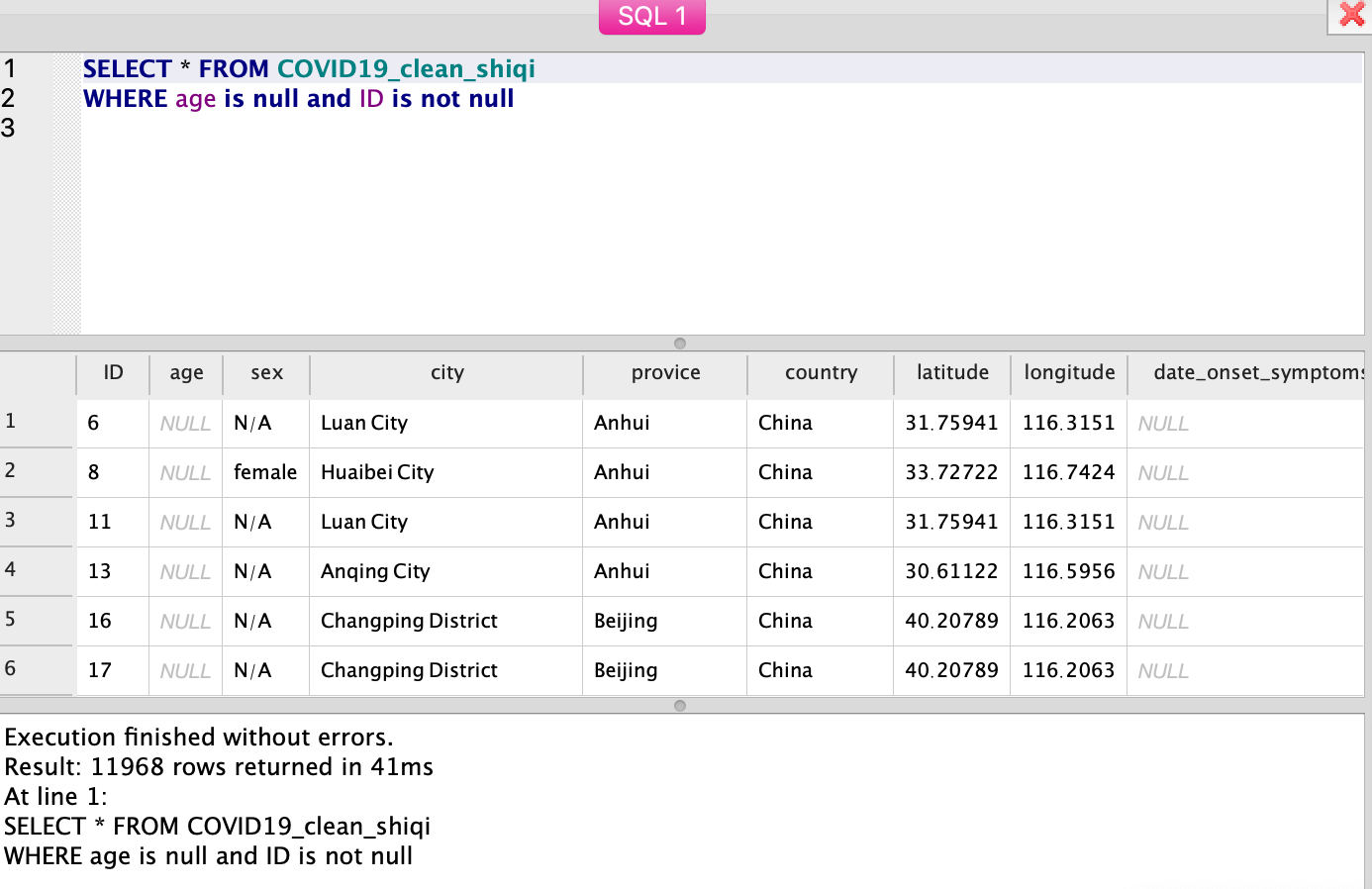
After cleaning the data through Openrefine and Python, we import our COVID-19 data into SQLite to check ​integrity constraints. We will focus on checking unique constraints and not null constraints here. Since there is only one table, we will not talk about foreign keys constraints.

Not null constraint ensures all rows in the table contain a definite value for the column which is specified as not null. In our data set, we think ID serves as the role of primary key. We check not null constraint on ID column using ‘SELECT \* FROM COVID19 WHERE ID is null’. It returns 953 rows. These 953 rows have null values not only in ID columns but also in other columns too. However, one of these rows has sex ‘4000’, which is obviously a error value here (fig 20).



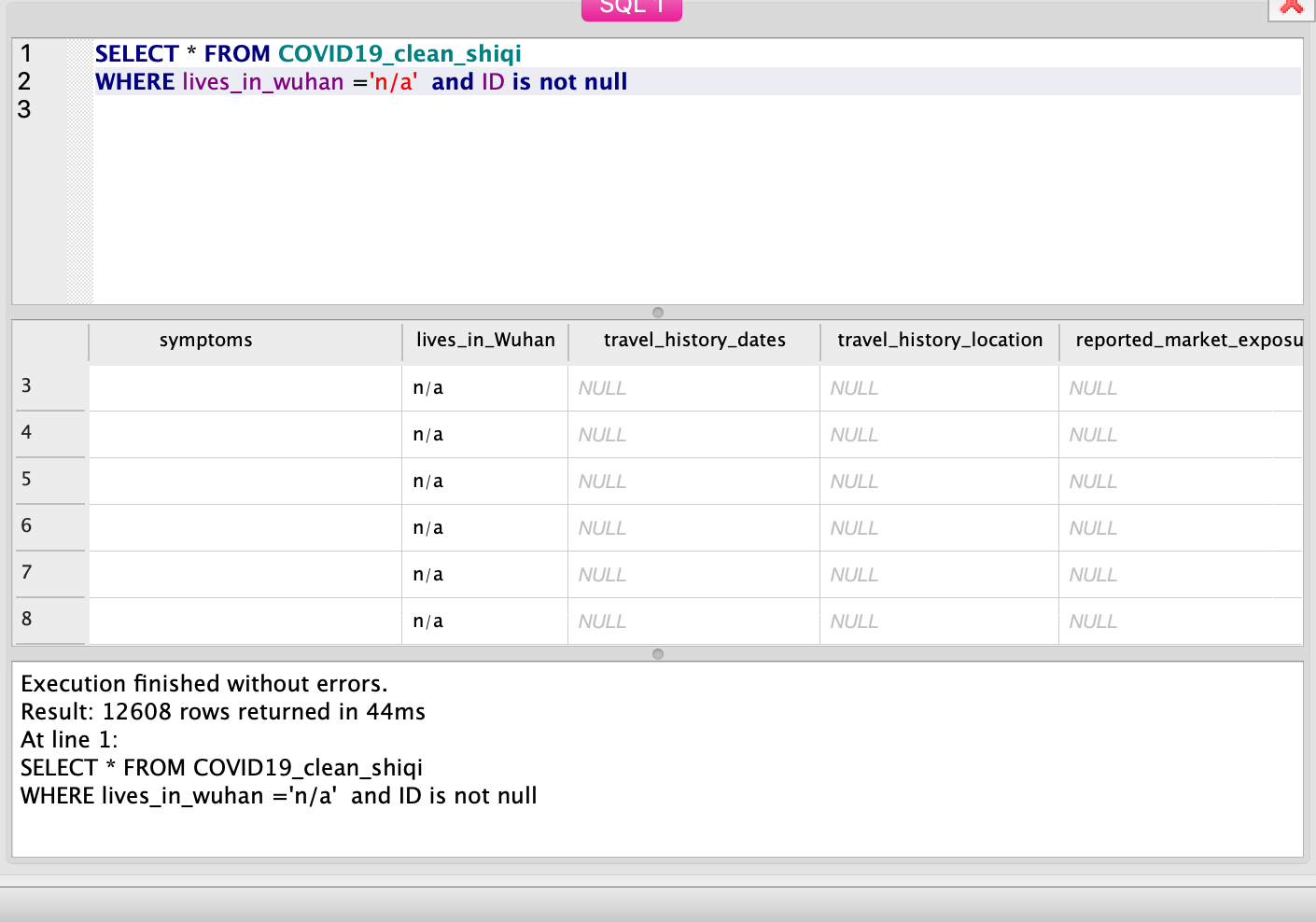
(fig 20: not null constraint check on id column)

We also check not null constraint on age because our hypotheses that older people are more likely to have diseases has to do with age. We use ‘ SELECT \* FROM COVID19 WHERE age is null and ID is not null ’and it returns 11968 rows. This means there are 11968 records which do not have age, which can affect our analytics about age (fig 21). So in the step of python cleaning, we removed these null values to ensure the accuracy of the results.



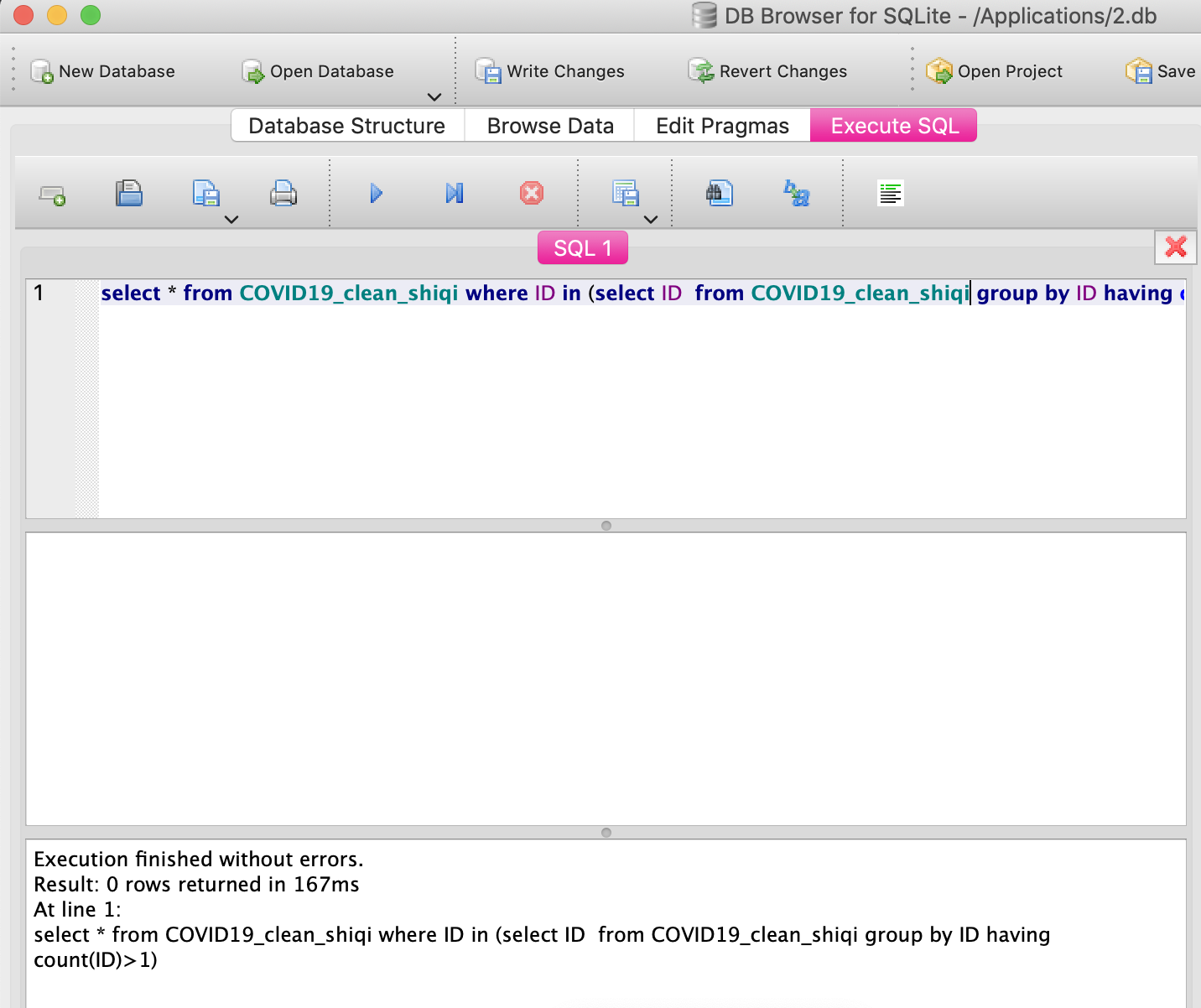
(fig 21: not null constraint check on age column)

Since our hypotheses also are related to lives\_in\_wuhan and symptoms, We also check lives\_in\_wuhan. ‘SELECT \* FROM COVID19\_clean\_shiqi WHERE lives\_in\_wuhan is null and ID is not null’ returns no null values, however, this happens because SQL does not identify n/a since the format of n/a is a text here. After we use ‘SELECT \* FROM COVID19\_clean\_shi WHERE lives\_in\_wuhan ='n/a‘ and ID is not null’, we find that lives\_in\_wuhan have 12608 null values. Symptoms column also has 12608 null values (fig 22). We also removed these null values respectively in the cleaning process of python.



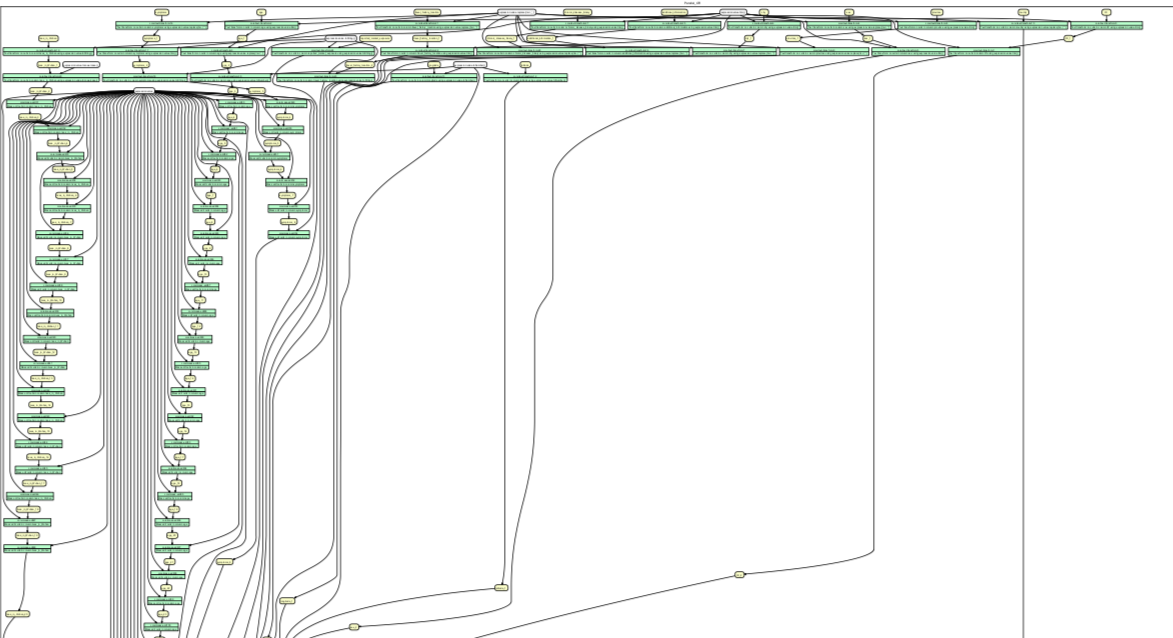
(fig 22: not null constraint check on lives\_in\_wuhan column)

After checking the null constraint, we also check the unique constraint. It ensures that a column or a group of columns in each row have a distinct value. It is okay to have duplicate values in our data because people can have the same ages or genders and they can all have traveled to Wuhan before. As for unique constraints here, we only want to make sure that ID do not have duplicate values. We use ‘select \* from COVID19 where ID in (select ID from COVID19 group by ID having count(ID)>1)’ to check if there are duplicate values. The result is 0 rows, which means there is no same ID value (fig 23).



(fig 23: unique constraint check)

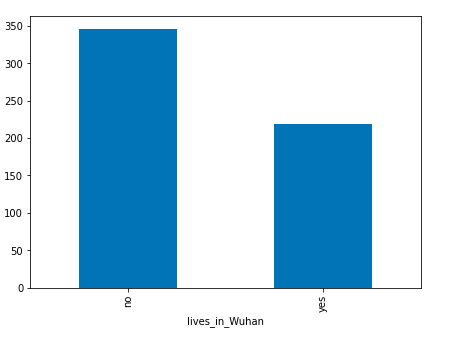
* Workflow Model

To show the data cleaning process and data provenance clearly, we used or2yw tool to create a YesWorkflow model according to the JSON records in our OpenRefine recipe. We transfer the JSON file to the yw file. Then we used the Graphviz tool to visualize the workflow. The input of the overall workflow is the raw data and the output is the clean dataset. We can see that the series of operations of the columns “lives\_in\_wuhan” and “age” are based on the dependencies. Some of the operations can only run after the former steps finished. So there are some dependencies. Since there are too many different but synonymous records in age and symptoms columns, the final workflow is very large and complex because of too many cleaning steps. This is the screenshot of part of the workflow (fig 24). It’s too large to show so the complete file is attached in the end.

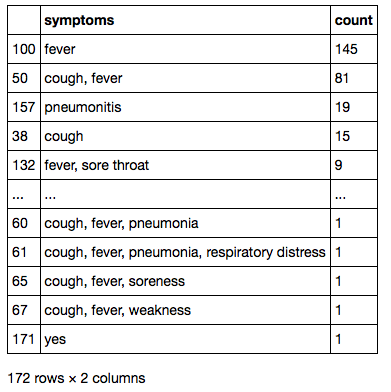
(fig 24: screenshot of part of the workflow model)

**Conclusions and future work**

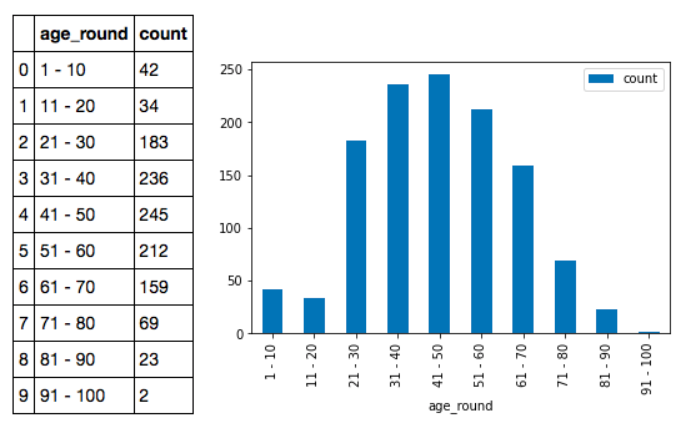
After all the cleaning works in OpenRefine and Jupyter notebook, we analyze the results according to our hypotheses. To verify our hypotheses, we also visualized specific data and analyzed the results. We found the first hypothesis is not successful since the number of patients who did not live in Wuhan is more than the patients in Wuhan (fig 25). The third hypothesis is not successful too. The patients are concentrated in the age range from 30-60 (fig 27). So we think it is not true that older people are more likely to have the disease. Hypothesis 2 is the only one that is successful. We grouped symptoms by each attribute and then sorted the result descending. We found most of the cases have the attributes of fever and cough. Some of them have other diverse complications (fig 26). So we think the second hypothesis is true that confirmed cases have common attributes of fever and cough.



(fig 25: visualization of hypothesis 1)



(fig 26: visualization of hypothesis 2)



(fig 27: visualization of hypothesis 3)

In the end, we got a clean COVID-19 dataset and the conclusion above. The format of the data is consistent and the records of some descriptive columns are simplified and concluded. The final file is cleaner and readable which is more comfortable and friendly to clients. With the workflow and the explanation notes in the OpenRefine and Jupyter notebook files, clients can have a clear sense of our process of clean and analysis.

During the cleaning process, processing abnormal and unclear values and clustering some columns must be some difficulties. Some records only offer a vague range so that we are hard to group and process them. And some columns with descriptive text, such as symptoms, contain too many kinds of synonymous words so that we were hard to classify and conclude. In the end, we had to delete several vague records to ensure the consistency and readability of the data. But meanwhile, we think accuracy must be affected to some extent.

Moreover, we think we may have something to do in the future. In the part of dealing with null and abnormal data, we think we can be more careful and clean it more properly. Compared with deleting these data, complementing them according to mean or other formulas may be a better way. This can save more details so that the results will be more accurate. Also, to improve the accuracy, we think we can find other relative datasets and make the same analysis to compare with our results. Only one of the three hypotheses was successful in this dataset. We think this may be because of the insufficiency of the cases. In the end, we only explore several columns in this project. This dataset contains much more information than we have used. So in the future, we can analyze other aspects in this dataset and do some further cleaning according to the columns we need.