Lucrative Locations and Sectors to Work in

CS513: Theory & Practice of Data Cleaning

Team 63 - Final Report

Authors (equal contributions, reverse alphabetical): [1] Chon Long Chiang (clc9@illinois.edu), [2] Kautuk Khare (kkhare2@illinois.edu), [3] Gabriel Angelo Sandoval (grs4@illinois.edu), UIUC, Summer 2021.

1. Points received for initial Phase-1 submission : __93__ (out of 100)

2. Our team chooses the following Phase-1 option:
   a. [ ] (A) No change to Phase-1 report
   b. [X] (B) Improved (or extended) Phase-1 report

Phase 1 Change Report

Phase 1 Remarks (TA Feedback)	Student Comments/Summary
[-5] no discussion on the connection between data quality issues and use cases (how these data quality issues affect your use case);	Added content for ***Impact of Data Quality Issues*** under Section 4 of Phase 1 Report
[-2] Did not specify how you will document changes made to the original dataset after cleaning (e.g. using a summary table).	Considering the number of cleaning changes we were anticipating, we were planning to add the metrics for change/improvement in data quality along with the cleaning steps in the narrative (We have done this in Phase 2 Report)

Lucrative Locations and Sectors to Work in

CS513: Theory & Practice of Data Cleaning - Project Report: Phase 1

Authors (equal contributions, reverse alphabetical): [1] Chon Long Chiang (clc9@illinois.edu), [2] Kautuk Khare (kkhare2@illinois.edu), [3] Gabriel Angelo Sandoval (grs4@illinois.edu), UIUC, Summer 2021.

1. INTRODUCTION

Students and budding professionals alike have a keen curiosity for knowing which industry, roles and locations they should target for work so that their earning potential is maximized. This project tries to answer these questions using a salary related questionnaire floated by www.askamanger.org. The data collected is not immediately ready for analysis and needs profiling and cleaning. This document elaborates on the dataset, the use case and the process we plan to deploy for cleaning the data and answering the questions posed by the use-cases.

2. DATA

Source

We came across this salary dataset at Kaggle (https://www.kaggle.com/filco306/salaries-across-industries-crowdsourced). The data was in turn collected by www.askamanager.org via the survey floated here: https://www.askamanager.org/2019/04/how-much-money-do-you-make-3.html. This is a live survey, though the dataset on Kaggle was uploaded in May 2020, and has temporal coverage from April 2020 to May 2020.

Schema

The data comprises of 10 fields:

Timestamp	When was the survey filled by the participant
Age	Age Range of the survey participant
Industry	Industry domain or sector which the participant works in
Job Title	Title or Role of the participant
Salary	Earning range of the participant
Currency	Currency of the salary
Location	City, State, Country where the participants work
Experience	Post college number (range) of years of experience
Job Title Context	Additional information filled by the participant regarding their job title
Other	Any other information which the participants want to fill up

Scale

The data has ~34000 cross-sectional data points.

3. USE CASES

The key question which we plan to answer using this data is:

U1: Which locations and Industries are most lucrative to work in?

This use case should help us identify top N locations and Industries for students and budding professionals to target if they want to maximize their earning potential post college degrees. To answer this use case we need to get a canonical list of location and industries, along with standardized salaries - all of which will require appropriate data cleaning steps.

A use cases which does not need data cleaning for answering is:

U0: Which age group contributes the most in the workforce?

Since the data already has the age brackets provided for each record, and they don't have any data quality issues, a simple SQL query like below should be able to get us answer for above question:

SELECT AGE, COUNT(*)

FROM DATA

GROUP BY AGE

ORDER BY COUNT(*) DESC;

A use case which we won't be able to answer irrespective of the amount of data cleaning we do is:

U2: Will the participants be better off working in the US compared to other countries?

Majority of the entries are located in the US. So we don't have enough samples of a particular job outside of the US to answer the question. Moreover, even though the data captures the raw salary range of the participants, which can be compared across the locations, it does not capture information on purchasing power parity of one location compared to other. So even if we are able to properly clean location data and are able to comment on which location is more lucrative compared to others based on the dollar amount, we will be unable to comment on whether the participant is better off at one location compared to another.

4. DATA QUALITY

Obvious data quality problems are the following

1. Inconsistent formatting of annual salary

   Examples: *88,000, 56K*

2. Some salaries are rated hourly/yearly.

   Examples: *$40/hour*

3. Some salaries are stated in different currencies so the numerical values are not directly comparable.
4. Some annual salaries include a currency symbol, some don't.

   Example: *50,000, $50,000, 50,000$*

5. Some annual salary comes with a short descriptions

   Example: *$150,000 with cash bonus, 42,000ish*

6. Industry type, Job Title, Location can be clustered.

   Example:

   Sacramento, CA

   Sacramento, CA, USA

   Sacramento/CA/USA

   Sacramento CA

7. Trailing whitespaces and extra white spaces between words in column values.

Impact of Data Quality Issues

Due to these data quality issues:

a) We will be unable to use Salary as a numeric field and hence will be unable to compare & aggregate it across industries or locations to answer the question posed by U1
b) The non-standardization of Location and Industry fields will distribute the records across multiple buckets, which may be actually representing the same Location (or Industry). This will result in incorrect analysis results

5. CLEANING PLAN

1. Remove unnecessary rows and comments in the original .xlsx file.
2. Convert file to csv format.
3. Use OpenRefine to address syntactic errors stated above.
4. Use SQLite to check constraints (age aligns with years of experience).
5. Document data cleaning flow.

Tools: Excel, OpenRefine, SQLite, Python, Pandas

.

Lucrative Locations and Sectors to Work in

CS513: Theory & Practice of Data Cleaning - Project Report: Phase 2

Authors (equal contributions, reverse alphabetical): [1] Chon Long Chiang (clc9@illinois.edu), [2] Kautuk Khare (kkhare2@illinois.edu), [3] Gabriel Angelo Sandoval (grs4@illinois.edu), UIUC, Summer 2021.

1. INTRODUCTION

Students and budding professionals alike have a keen curiosity for knowing which industry, roles and locations they should target for work so that their earning potential is maximized. This project tries to answer these questions using a salary related questionnaire floated by www.askamanger.org. The data collected is not immediately ready for analysis and needs profiling and cleaning. This document elaborates on the dataset, the use case and the process we plan to deploy for cleaning the data and answering the questions posed by the use-cases.

2. DATA

Source

We came across this salary dataset at Kaggle (https://www.kaggle.com/filco306/salaries-across-industries-crowdsourced). The data was in turn collected by www.askamanager.org via the survey floated here: https://www.askamanager.org/2019/04/how-much-money-do-you-make-3.html. This is a live survey, though the dataset on Kaggle was uploaded in May 2020, and has temporal coverage from April 2020 to May 2020.

Schema

The data comprises of 10 fields:

Timestamp	When was the survey filled by the participant
Age	Age Range of the survey participant
Industry	Industry domain or sector which the participant works in
Job Title	Title or Role of the participant
Salary	Earning range of the participant
Currency	Currency of the salary
Location	City, State, Country where the participants work
Experience	Post college number (range) of years of experience
Job Title Context	Additional information filled by the participant regarding their job title
Other	Any other information which the participants want to fill up

Scale

The data has ~34000 cross-sectional data points.

3. USE CASES

The key question which we answer using this data is:

U1: Which locations and Industries are most lucrative to work in?

This use case should help us identify top N locations and Industries for students and budding professionals to target if they want to maximize their earning potential post college degrees. To answer this use case we need to get a canonical list of location and

industries, along with standardized salaries - all of which will require appropriate data cleaning steps.

A use cases which does not need data cleaning for answering is:

U0: Which age group contributes the most in the workforce?

Since the data already has the age brackets provided for each record, and they don't have any data quality issues, a simple SQL query like below should be able to get us answer for above question:

```
SELECT AGE, COUNT(*)
FROM DATA
GROUP BY AGE
ORDER BY COUNT(*) DESC;
```

A use case which we won't be able to answer irrespective of the amount of data cleaning we do is:

U2: Will the participants be better off working in the US compared to other countries?

Majority of the entries are located in the US. So we don't have enough samples of a particular job outside of the US to answer the question. Moreover, even though the data captures the raw salary range of the participants, which can be compared across the locations, it does not capture information on purchasing power parity of one location compared to other. So even if we are able to properly clean location data and are able to comment on which location is more lucrative compared to others based on the dollar amount, we will be unable to comment on whether the participant is better off at one location compared to another.

4. DATA QUALITY

Some of the key data quality problems we encountered were:

1. Inconsistent formatting of annual salary

   Examples: *88,000, 56K*

2. Some salaries are rated hourly/yearly.

   Examples: *$40/hour*

3. Some salaries are stated in different currencies so the numerical values are not directly comparable.
4. Some annual salaries include a currency symbol, some don't.

   Example: *50,000, $50,000, 50,000$*

5. Some annual salary comes with a short descriptions

   Example:

   $150,000 with cash bonus, 42,000ish

6. There were salaries in multiple currencies
7. Industry type, Job Title, Location were clustered.

   Example:

   Sacramento, CA
   Sacramento, CA, USA
   Sacramento/CA/USA
   Sacramento CA

8. A lot of Industries and Locations had very few records, hence making the data insufficient to be included in the analysis.
9. Trailing whitespaces and extra white spaces between words in column values.

Impact of Data Quality Issues

Due to these data quality issues:

c) We will be unable to use Salary as a numeric field and hence will be unable to compare & aggregate it across industries or locations to answer the question posed by U1
d) The non-standardization of Location and Industry fields will distribute the records across multiple buckets, which may be actually representing the same Location (or Industry). This will result in incorrect analysis results

5. CLEANING STEPS

Our full data cleaning task is a four-stage process:

First, we transform raw data in Excel so that it is ready for ingestion in the OpenRefine tool. Then in the second stage we address syntactic issues by using OpenRefine to transform and filter out values

across the different columns of the dataset. Once syntactic issues are addressed, we export this refined dataset.

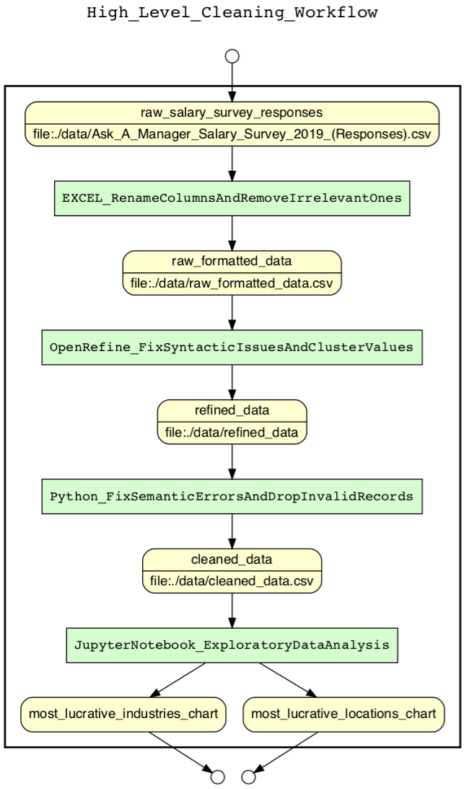
Third, we used Python with Pandas library import the refined dataset from OpenRefine to check semantic issues and further make sure that the data is high in integrity by dropping records that violate the following constraints:

- Industry, Salary_Local and Location should be not *NULL*
- Age should be greater than the number of years of experience
- Currencies should be identifiable:

```
'USD'   'GBP'   'CAD'   'EUR'   'SEK'
'AUD/NZD' 'JPY' 'CHF' 'HKD' 'ZAR'
```

Finally, we used Jupyter Notebooks/Python to do exploratory data analysis, remove outliers and under-represented categorical values records and try to answer questions posed by the use case.

This four stage process is shown in the diagram below.



We used YesWorkflow and OpenRefine History to keep the provenance of data and processes for reproducibility.

Step 1: Prepare data for OpenRefine ingestion

Removed the top row, which was the instructions on how to add more data to this dataset and renamed the columns to shorter strings (earlier they were the full questions except for the first column Timestamp). The final columns were:

1. Age
2. Industry
3. Job_Title
4. Salary
5. Currency
6. Location
7. Experience
8. Context
9. Other

Tool Used in this Step: Excel

Step 2: Cleaned Salary, Industry, Job Title and Location information in OpenRefine

For this we firsted Imported the data and created a new project in OpenRefine, and then tackled each of the columns one by one:

Salary

To answer U1**, Salary** is the most critical field. While analysing this field we realized that it is not a numeric field in the data collected and contains text of a variety of types. To get it to an analyzable state, we took the following actions via OpenRefine capabilities:

Removed trailing spaces

Filtered the data to see what values are in it which are non numbers. Used Regex "***^[0-9]+[\.]*[0-9]*$***" for it, and then inverted the filter

1120 rows out of ~34k total rows did not have Salary as a proper number. This pertained to nearly ***3.2% of records***. Some examples are shown in the below screenshot

All	Timestamp	Age	Industry	Job_Title	Salary
13.	Wed Apr 24 11:43:37 GST 2019	25-34	Telecommunications	Marketing Manager	73000$
55.	Wed Apr 24 11:44:05 GST 2019	25-34	Health insurance	Project specialist	42,000ish
58.	Wed Apr 24 11:44:07 GST 2019	35-44	Public Libraries	Library Director	51,000includinghealthinsurancestipend
93.	Wed Apr 24 11:44:51 GST 2019	55-64	Legal	Attorney	125,00
108.	Wed Apr 24 11:45:07 GST 2019	25-34	Financial services	Copywriter	£25000
112.	Wed Apr 24 11:45:10 GST 2019	45-54	Social Work	Lead Facilitator	35K
119.	Wed Apr 24 11:45:15 GST 2019	35-44	Civil Service	Human Service Specialist 4	59K
137.	Wed Apr 24 11:45:32 GST 2019	35-44	Veterinary medicine	Veterinary technician	$18.53/hror$35,600/yr
158.	Wed Apr 24 11:46:03 GST 2019	25-34	Manufacturing	Human Resources Generalist/Payroll Admin	60-90K
192.	Wed Apr 24 11:46:41 GST	55-64	Consulting	Vice President	175K+bonus

To fix these, we:

1. Removed $ sign which updated 306 rows
2. Removed £ sign which updated 82 rows
3. Removed “,” which updated 481 rows
4. Replaced K with 000, which updated 119 rows
5. Replaced k with 000, which updated 312 rows
6. Removed yr and year which updated 24 rows
7. Removed ~, which updated 24 rows
8. Fixed rows mentioning Approx, Around, Roughly etc
9. Splitted Salary to get the first number in the column

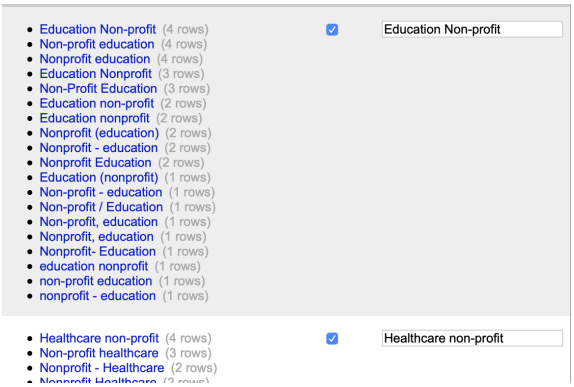
All these steps got the Salary column to not have proper numbers in 129 rows (compared to 1120 rows before cleaning), which ***brought down bad data quality from 3.2% of records to 0.3% of records.***

We removed these 129 rows (examples shown below) resulting in Salary column having proper numbers and then transformed it to a number type

Salary
Basepay49900withbonusopportunitiesseasonalOThaveearnedupto95000
Variesonhowmuchwor000youget.Jobpays15/hr(minwageinNY).
Varies>10000

Industry

It being a free flow text field during data entry process, Industry had a lot of similar (denoting the same) but differently worded text, as shown below:



To standardize it we took the following steps

1. Removed trailing spaces
2. Removed consecutive spaces
3. Analyzed the clusters and noticed there were 7040 unique clusters. We merged them via multiple techniques:
   a. Method: '`key collision`', Keying function '`fingerprint`'. This resulted in reducing number of clusters to 5621
   b. Method: '`key collision`', Keying function '`ngram-fingerprint (ngram_size: 2)`'. This resulted in reducing number of clusters to 5381
   c. Method: '`key collision`', Keying function '`metaphone3`'. This resulted in reducing number of clusters to 4024
   d. Method: '`key collision`', Keying function '`Daitch-Mokotoff`' This resulted in reducing number of clusters to 2722

With this effort we ***were able to reduce the number of clusters from 7040 to 2722, an improvement of ~60%***.

Location

Similar to Industry, Location was also a free text column during data collection, resulting in a lot of noisy and clustered text.

To standardize it we took the following steps

1. Removed trailing spaces

2. Removed consecutive spaces
3. Analyzed the clusters and noticed there were 10142 unique clusters. We merged them via multiple techniques:
    a. Method: '`key collision`', Keying function `'fingerprint'`. This resulted in reducing number of clusters to 7107
    b. Method: `'key collision'`, Keying function '`ngram-fingerprint (ngram_size: 2)`'. This resulted in reducing number of clusters to 6051
    c. Method: `'key collision'`, Keying function '`metaphone3`'. This resulted in reducing number of clusters to 4678
    d. Method: '`key collision`', Keying function '`Daitch-Mokotoff`' This resulted in reducing number of clusters to 3343

With this effort we ***were able to reduce the number of clusters from 10142 to 3343, an improvement of ~67%***.

Even though it was not necessary to answer U1, we also standardized the Job_Title field reducing clusters from 14616 to 8645, using `fingerprint`, `ngram-fingerprint` and `metaphone3` collision techniques.

Tool Used in this Step: OpenRefine[1]

Step 3: Drop records with NULL values

It was necessary to drop records with *NULL* values since those data wouldn't contribute anything at all, based on our use case, for the data analysis phase.

Below shows the number of records dropped based on columns:

```
Industry - 960
Salary_Local - 11
Location - 1357
```

With this effort we ***were able to remove a total of 2328 invalid records.***

Tool Used in this Step: Python

Step 4: Drop records with invalid age based on experience

We also filtered out the records showing invalid age based on their number of years of experience in the industry.

By satisfying this constraint that *Age should be greater than the number of years of experience*, **we were able to remove 9 more invalid records.**

Tool Used in this Step: Python, re library for regex processing

Step 5: Remove records with undetermined currencies

Without knowing the currency of the salary, we have no way to convert the person's salary to US Dollars. Recognized currencies in this dataset include the following:

`USD, GBP, CAD, EUR, SEK, AUD/NZD, JPY, CHF, HKD, ZAR`

By satisfying this constraint that *Salary currency should be within the list of recognized currencies*, **we were able to remove 174 invalid records.**

Tool Used in this Step: Python

Step 6: Standardized Salary Currency

Even though the salary data resulting from OpenRefine analysis was syntactically clean (all values were numbers), it was not actionable since those salary numbers were pertaining to different currencies. To standardize these different currencies, we used Python code to transform salary to salary_USD using specific exchange rates (as of the analysis day):

Currency	Exchange Rate
USD	1
GBP	1.37
CAD	0.8
EUR	1.18
SEK	0.12
AUD/NZD	0.74
JPY	0.0091
CHF	1.09
HKD	0.13
ZAR	0.067

During this standardization, we also removed the records with undetermined currencies and where Age-Experience numbers were a mismatch.

[1] *OpenRefine History file provided as part of the submission*

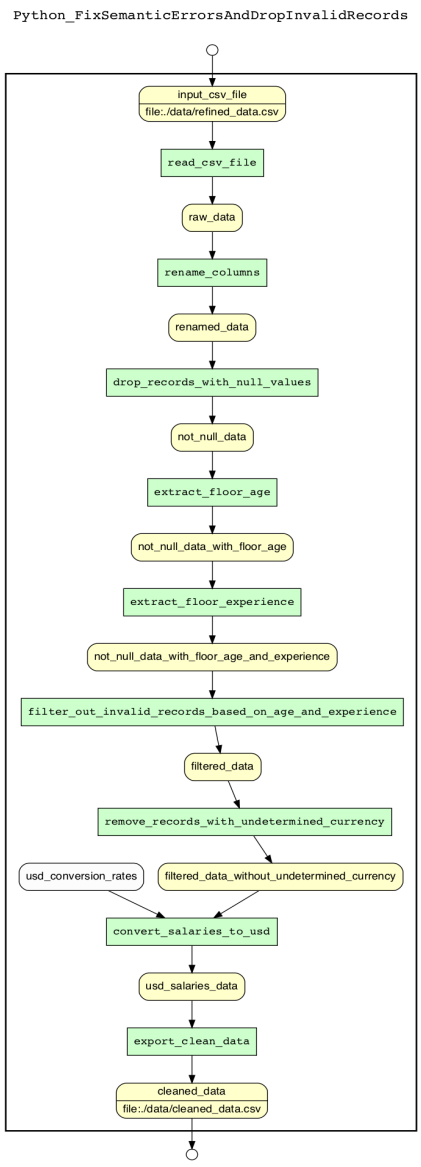
Tools Used in this Step: Python

Provenance

Since we used a Python script to automate the workflow, we used YesWorkflow to ensure that the whole process is reproducible.

We generated Data View, Process View, and Combined (Data+Process) View to achieve this.

A detailed (combined view) YesWorkflow diagram[2] for this workflow above is shown below:



Step 7: Removing Outliers

When we tried doing analysis using the cleaned data from Step 3, we realized it is having outliers which is resulting in incorrect and non-realistic results. To fix these we removed two key types of outliers in the analysis done:

1. **Records with very low or very high salaries.**
    a. There were records with salaries lower than 100 dollars, which may have been artefact of hourly rates, and there were also salaries more than a couple of million dollars likely an artefact of incorrect currency data
2. **Location and Industries which are not having enough records to make meaningful conclusion about them**
    a. There were certain industries which were present only in a single record and were hence incorrectly surfacing as the top industry due to high salary value of that record

Tool Used in this Step: Python, Jupyter Notebooks

6. ANALYSIS AND RESULTS

We analyzed the cleaned data in the Jupyter notebook, and identified the answers to the questions posed in U1, using the “group by” aggregation methods provided by Python. An example of this is:

```
top_industries = cleaned_data
      .groupby("Industry")
      .aggregate(func=np.median)
      .sort_values(by="Salary_USD",
                   ascending=False)
```

We also ingested this cleaned data in SQLite database, and did similar analysis, but then decided to continue using Jupyter notebook + Python combination due to the powerful data visualization libraries they offer.

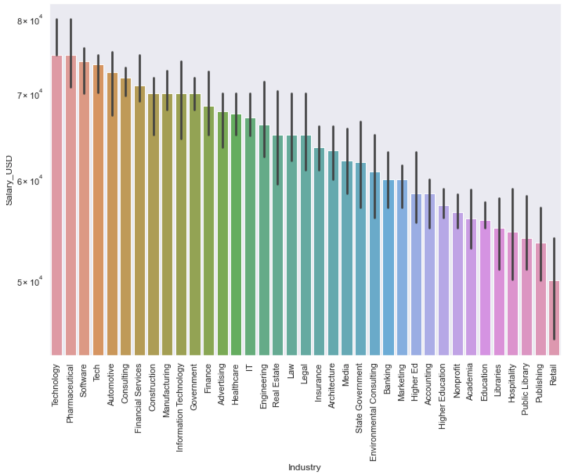
We present below the key results we discovered with this analysis. [3]

[2] *Clean_Workflow.py file with YesWorkflow annotations attached with the submission*

[3] *Detailed code and all results are provided in explore.html (html export of ipynb file) as part of submission*

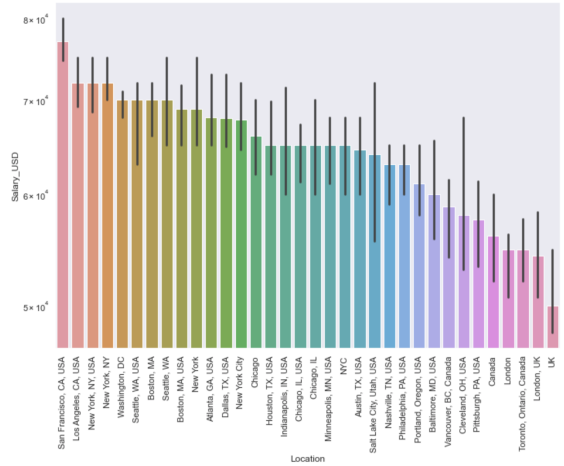
Most Lucrative Industries

Technology, Pharma and Software popped up as the most lucrative industries to work in followed by Automotive, Consulting and Financial Services.



Most Lucrative Locations

In terms of location, San Francisco and Los Angeles top the list along with New York, not lagging much behind. This is likely a manifestation of Silicon Valley presence in SFO Area. In terms of other (non USA) countries, UK and Canada show up in the top 20 locations.



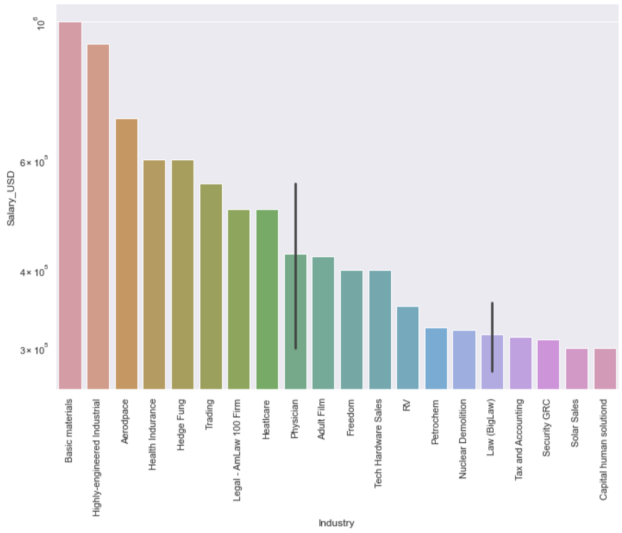
7. CONCLUSION

Based on the results of the analysis, and our experience of working on this project, we identify the following avenues of future work:

1. Even though we clustered the Industry and Location column substantially resulting in massive improvement in data quality, our merging was very algorithm driven and less based on contextual and semantic understanding. An example of this is Industry Clusters for IT and Information Technology, which could have been merged together, but we missed them. Focusing on context/semantics of data with respect to problem at hand was one of our key take-aways from the project

2. During the project execution we realized (again) that outliers can change the outcome of analysis. The cleaned Salary field had a few outliers, which most likely represented hourly or weekly compensation rates instead of yearly salary as expected. As part of the cleaning step we dropped such rows. As an improvement we can recognize hourly vs annual vs monthly rates based on the raw value on the Salary column and adjust the salary accordingly.

3. Less number of data points for a specific categorical variable can severely affect the result of analysis. We noticed this when we initially tried to identify the most lucrative industries, “Basic Materials” topped the chart which seemed a bit counter-intuitive to us (that “incorrect” chart shown below) given there was no error, indicated by the black vertical lines. There are multiple possible causes for this. First is more inclusive clustering. Second is to use statistical methods to remove outliers

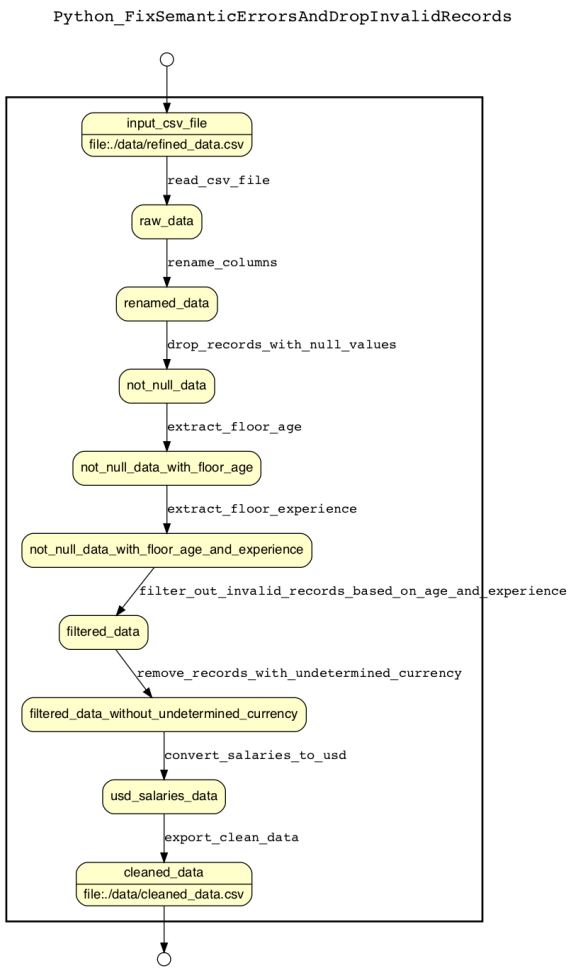


On analysis we realized that there is only a single record in data for “Basic Materials” with a salary of 1 Million USD. Because of that, this record is coming out on the top. We then used only those industries (and locations) in the analysis which had at a minimum of 100 records. As an improvement area, we can increase this threshold of 100 to get more robust results.

In terms of contribution, all team members contributed equally towards this work. Chon Long focused on data source identification and initial exploratory data analysis, Gabriel focused on Python analysis and Yes Workflow, and Kautuk focused on OpenRefine Analysis and Outlier identification. We all worked on the documentation and reporting.

Appendix

Provenance Graphs: Python Data Cleaning Workflow (Data View)



Provenance Graph: Python Data Cleaning Workflow (Process View)

Python_FixSemanticErrorsAndDropInvalidRecords