Project #1 - data merging

1. We open Jupyter and import Pandas, re, matplotlib, seaborn, fuzzywuzzy, and numpy.

2. We import the CSV file.

3. We check the head of the data.

4. We notice 34,706 rows and 22 columns.

5. Now, let's do a quick interpretation of the columns and requirements:

    - We have three sources with data about a company in each row.

    - Based on the requirement, each source has about 80% accuracy.

    - We need to perform a merge between the sources and keep only one.

6. We check the data types and notice that the phone number in Source 1 and the phone in Source 2 are int and float, respectively.

    - For phone numbers, it would be better to declare them as str, so we will make the conversion.

7. We check the number of null elements in each column and observe:

    - Source1-full address has 1 null.

    - Source1-phone number, Source1-country, and Source3-phone have 0 null data.

    - The rest of the columns have more than 4000 null values.

8. We apply lower and strip to all columns to transform everything to lowercase and remove extra spaces.

9. For the phone numbers, we create a function to:

    - Remove anything that isn’t a digit or a comma.

    - Retain phone numbers with more than 10 characters.

10. We check for duplicate rows and notice that there are 1,166 duplicates.

    - We delete them, leaving us with 33,540 rows.

11. We will rename the columns to make them more readable.

12. Now, let's see what's going on with the addresses:

    - What countries we have, which regions, and which cities.

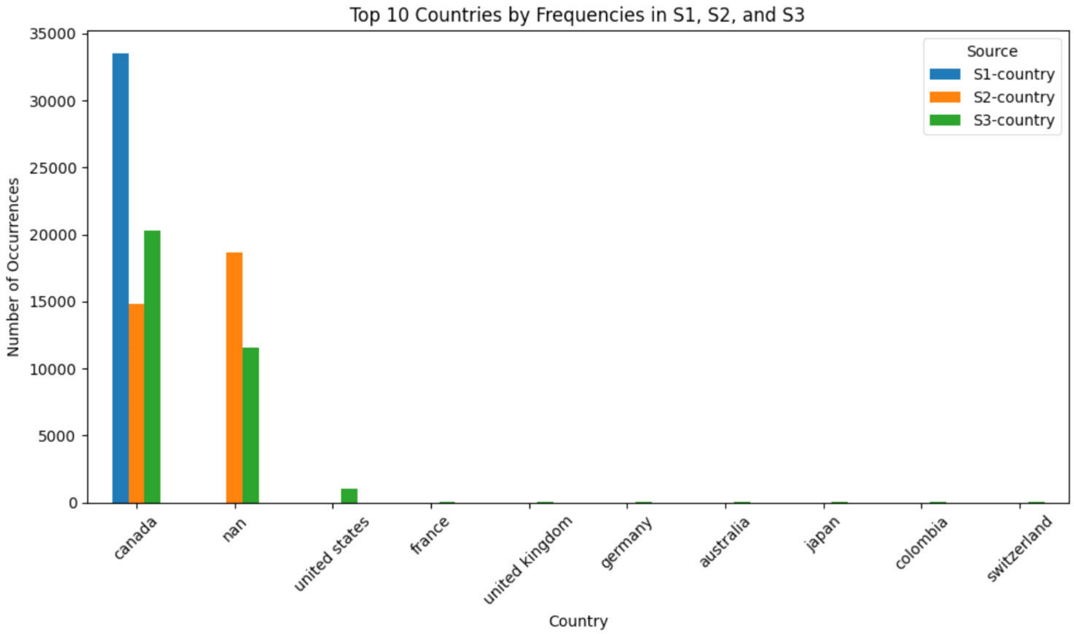
    - We display the frequency of countries from S1, S2, and S3, and observe:

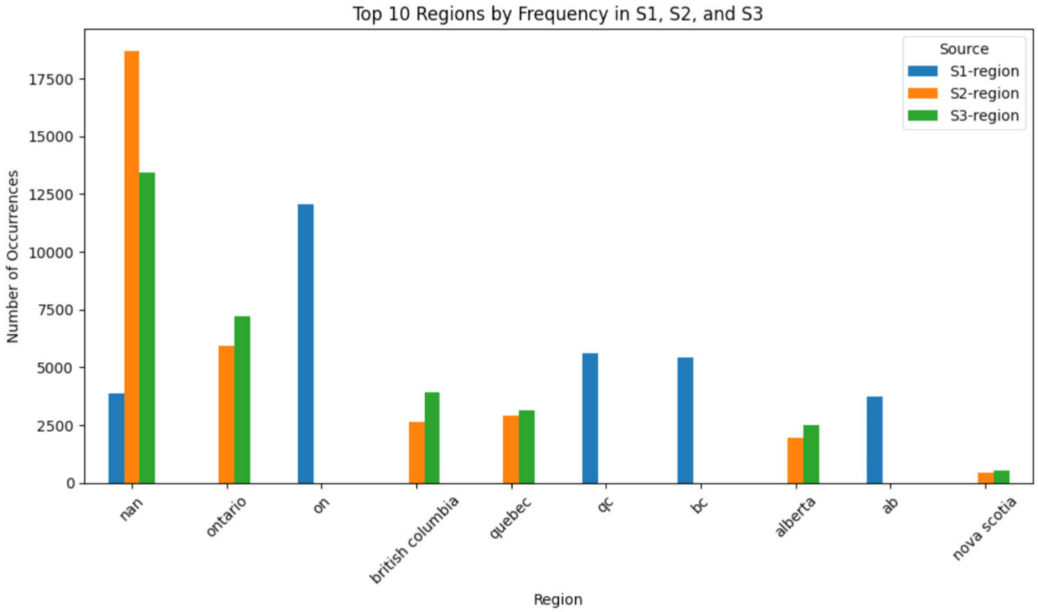
        - In S1, it's only Canada.

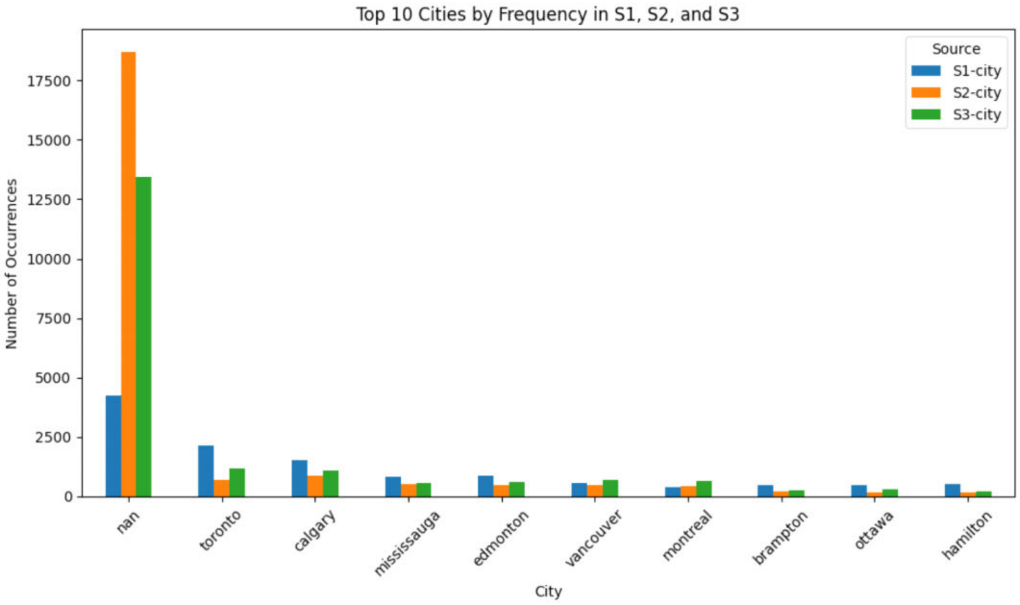
- In S2, it's also only Canada.

- In S3, there are different countries.

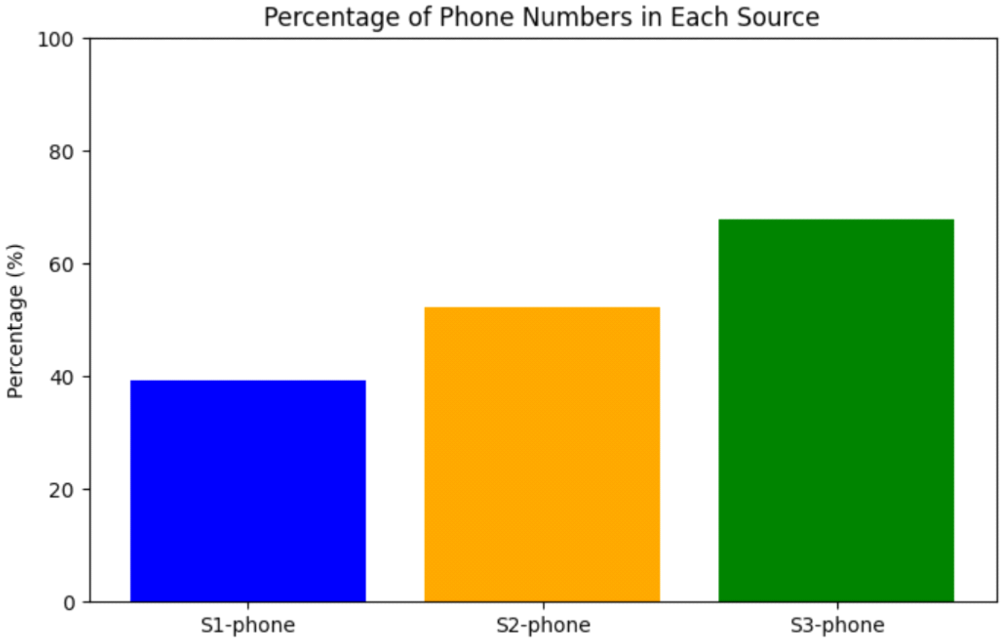
- This is something very important to keep in mind for later use.







13. Now, let’s take a look at the phone numbers and observe their uniqueness in each of the sources.



We will create a new column "Phone" and try to select the correct phone number based on the following considerations:

- If there is 100% similarity between the phone numbers, select one of them.

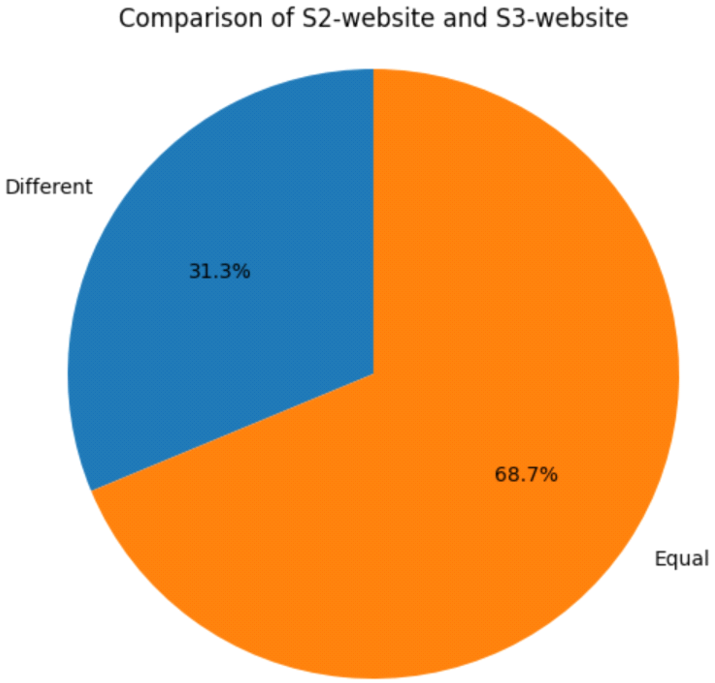
- If there is between 85-100% similarity, choose the first one that appears.

- If no rule applies, select from S3, as it is the source with the highest completeness percentage (around 65%).

14. Now we focus on the website part:

14.1: Check for uniqueness.

14.2: Check what % of S2-website is equal to S3-website and what % are different.

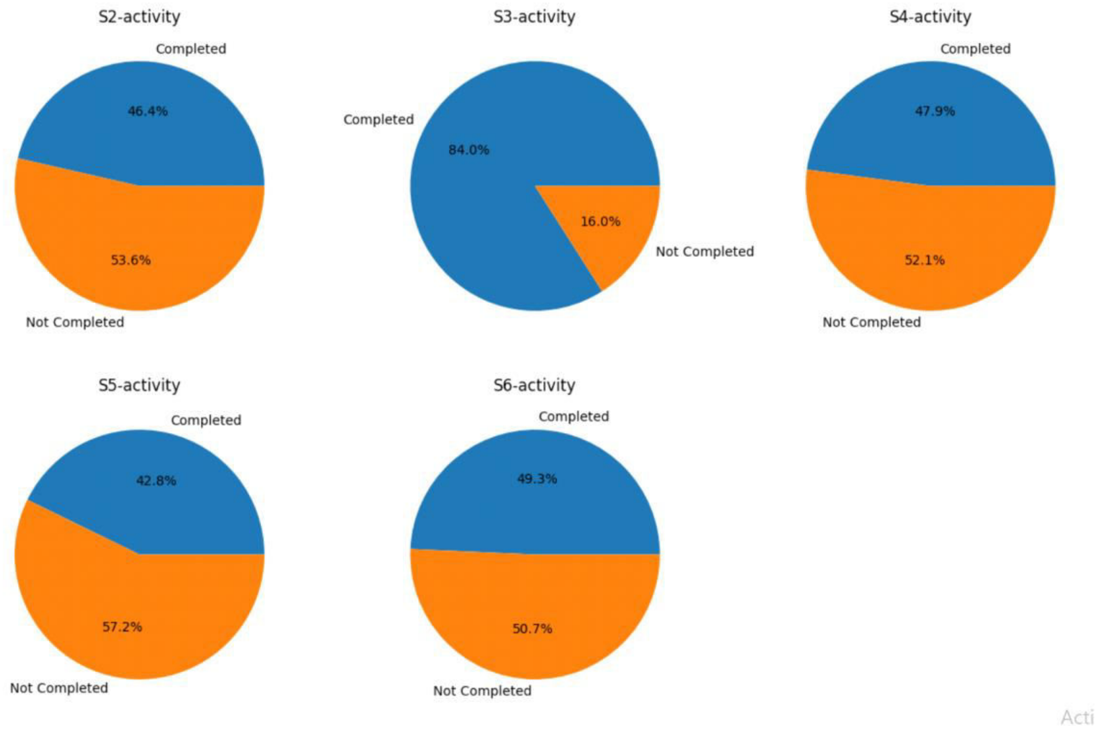


4.3: I want to see a few lines where S2-website != S3-website, and we notice something interesting: whenever they are different, one is null, and the other is filled. There is no case where both are non-null and also different from each other. Therefore, it's very simple to correctly choose the Website: if they are equal, we pick one, and if one exists while the other is null, we choose the non-null one.

15. Now let's focus on the Activity column, where we have data from quite a few sources: S2, S3, S4, S5, S6.

15.1: Check for uniqueness in each source.

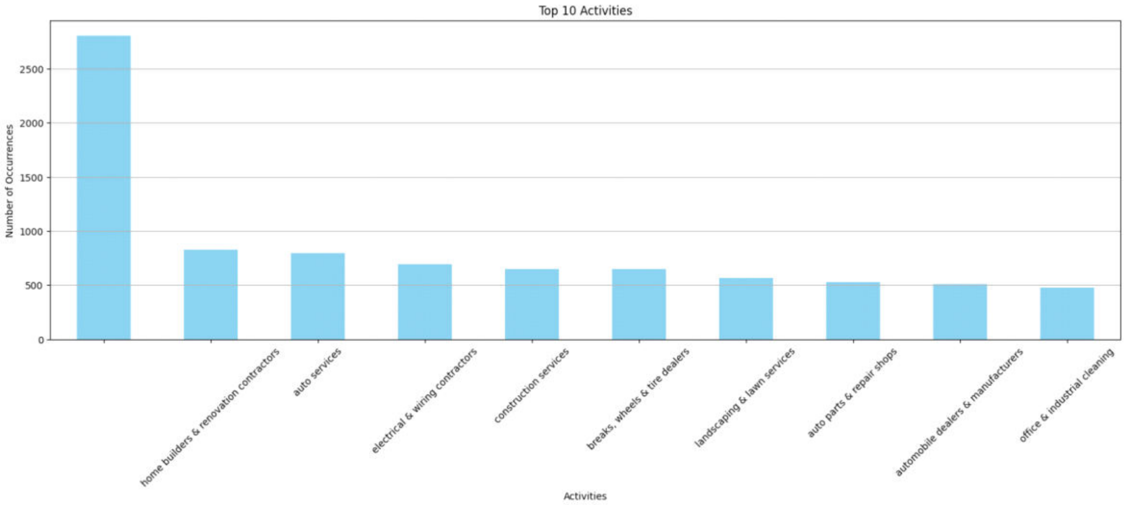
15.2: Verify with a pie chart what % are filled and what % are not.



15.3: We observe that there are 19,301 cases where there are at least two columns with the same activity (>50%). Therefore, the first idea for selecting the activity will be to choose the one that appears most frequently in the row. We will use FuzzyWuzzy with a minimum ratio of 80% (we tried without it, but it works better with FuzzyWuzzy, as it combines activities like "electrical & wiring contractors" with "electrician," which is very useful).

15.4: We will create a combination function that works as follows: if there are at least two activities with a similarity of >80%, it will keep only one of those values. If there is no similarity between columns (40% of cases), it will choose the most complex one, meaning the longest one with the most data.

15.5: After that, let's check the top 10 activities. We notice that most of them have empty activities, followed by "home builders & renovation contractors," and then "auto services,"



etc.

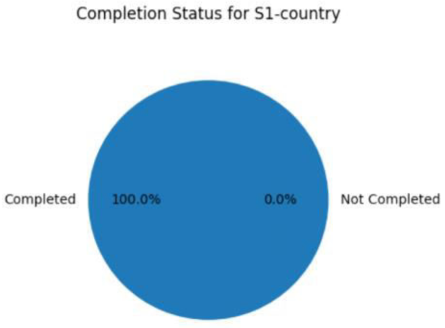
16. Now we arrive at location, and we need to choose the most appropriate one. We have:

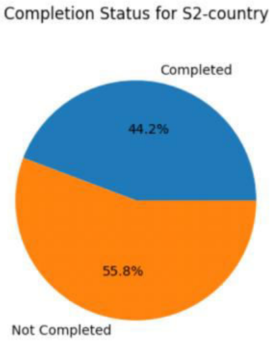
S1-full address		
S1-country	S2-country	S3-country
S1-region	S2-region	S3-region
S1-city	S2-city	S3-city

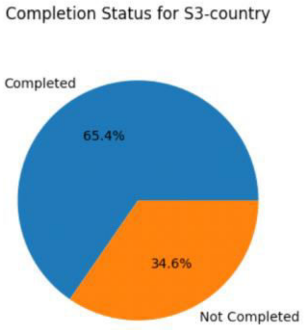
16.1: We check how many rows have all three countries equal and find (11,880), which is <50%.

16.2: However, I believe it's more important to look at cases where at least two out of three countries are equal to determine a majority. We see that there are 23,240 cases where at least two countries match, which is >50%.

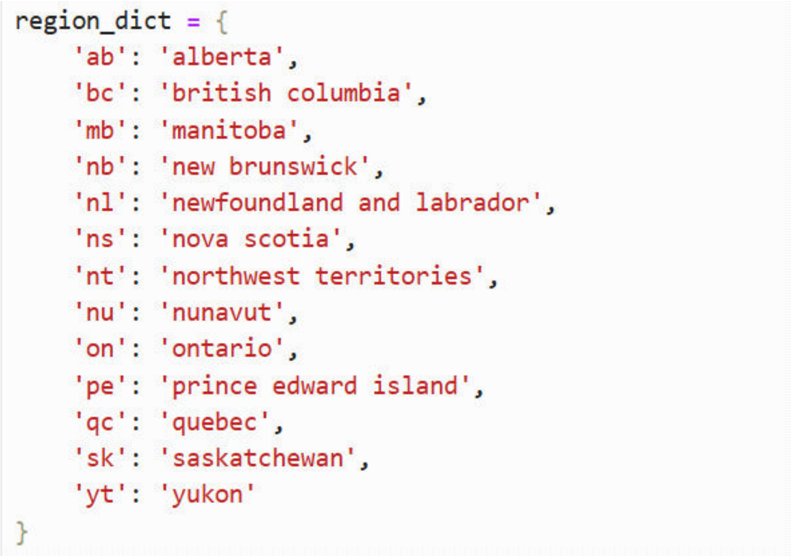
16.3: We also observe the completion rate, and in S1, it's 100% complete, so we could use it as a benchmark as a last resort.







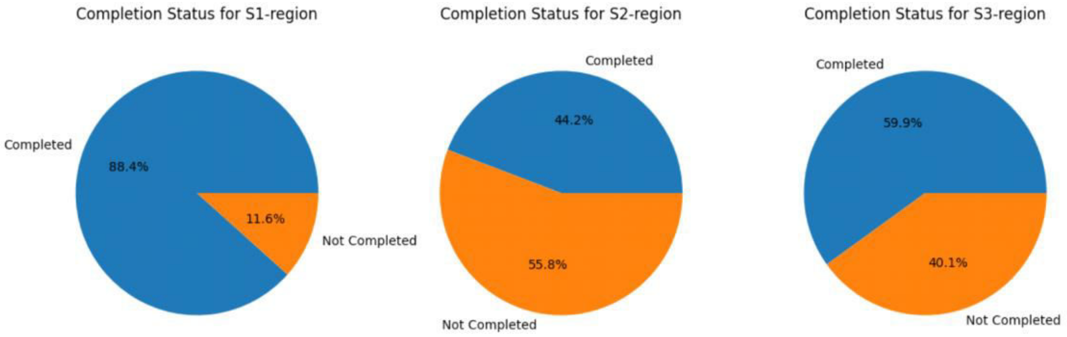
16.4: Moving on to regions, we observe that there are (9,402) cases where at least two regions are equal, which is <50%. However, in S1, we notice that the regions are abbreviated. At the same time, we remember that S1's country is only Canada, so we'll search on Google for the regions of Canada (13 in total), and then create a dictionary that we will use to change the S1-region from abbreviated to full region names.



```
region_dict = {
    'ab': 'alberta',
    'bc': 'british columbia',
    'mb': 'manitoba',
    'nb': 'new brunswick',
    'nl': 'newfoundland and labrador',
    'ns': 'nova scotia',
    'nt': 'northwest territories',
    'nu': 'nunavut',
    'on': 'ontario',
    'pe': 'prince edward island',
    'qc': 'quebec',
    'sk': 'saskatchewan',
    'yt': 'yukon'
}
```

16.5: We check again how many rows have at least two equal regions and now we have 11,420, which is slightly better.

16.6: We observe that S1 has a completion rate of 88.4%, while S2 and S3 are below 60%. Therefore, we will use S1 as the benchmark in cases where there are no duplicates.

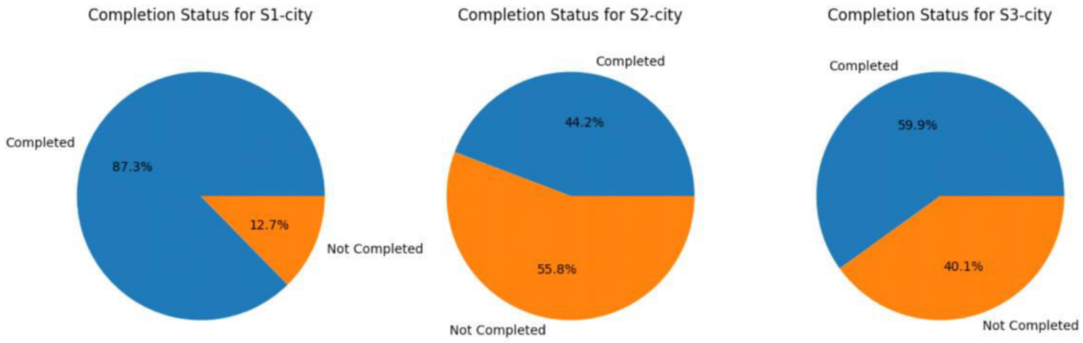


16.7: Now we will do the same for City:

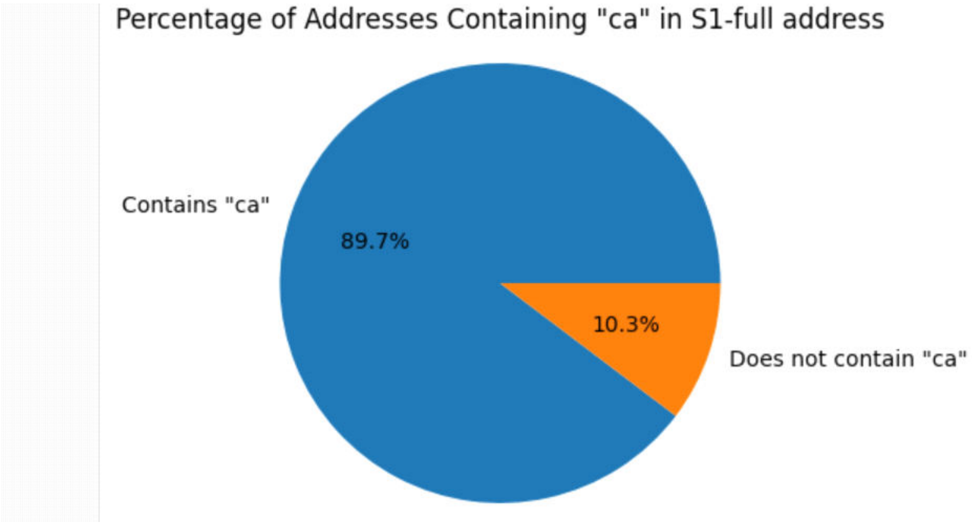
- We see that 9,260 rows have at least two equal cities.

- We observe that 87.3% in S1 indicates that they are completed for City, so we will take it as the benchmark in cases where there are no duplicates.

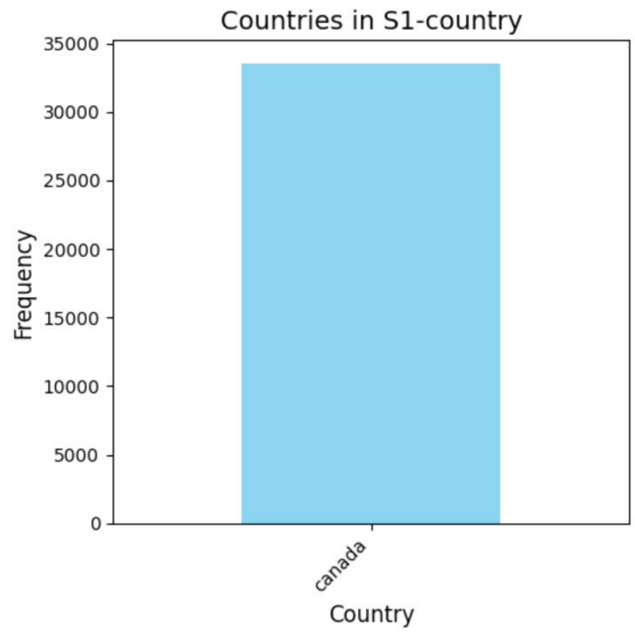
- We will create a function where if there are two identical cities, one of them will be chosen. If one city is completed while the others are NaN, only that city will remain. If they are different, we will keep the one from S1, as it is 87% completed. If S1 does not exist, we will take from S3 since it has a higher completeness than S2.



17. Well, now let's check the data from S1-full address and we want to see the main country. We notice that we have the abbreviation for the country "ca", so we check what percentage (%) it contains and we find it is about 90%. Therefore, we expect that the final product will have the majority in Canada.



18. We now observe that 100% of the country data is Canada, so it is clear that the data is sourced from Canada.



19. Now, we will combine Country, Region, and City into a single column named Location.

20. We will keep only the columns Location, Phone, Website and Activity.

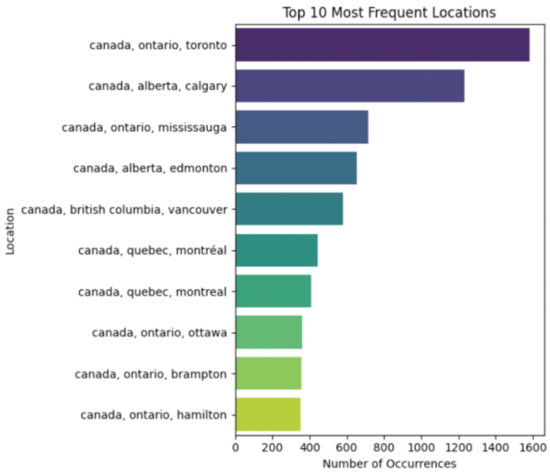
21. Now let's check the data in the final DataFrame.

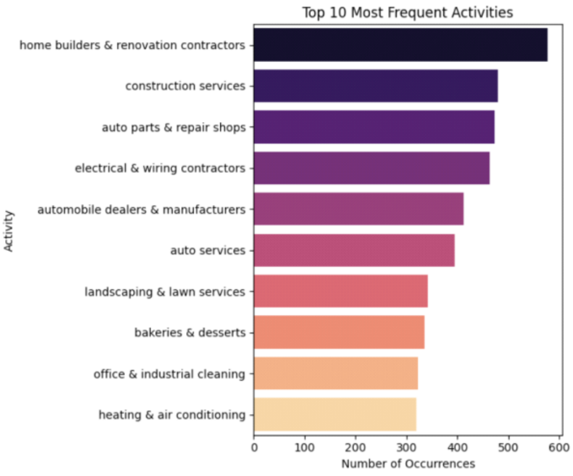
21.1: We check for duplicate rows and observe 2,219, so we will combine them.

21.2: We now observe the 10 most popular locations and the 10 most frequent activities, noting that most activities are NaN.

21.3: We see duplicates in the Phone column and think that two similar phone numbers cannot exist for two different companies.

21.4: We observe a total of 7,338 duplicates for Website; two different companies cannot have the same Website.





22. After noticing that there are identical websites, I thought it must be the same company, as there can't be two companies with the same website. Therefore, I grouped by this column, but a problem arose: different locations, activities, and phone numbers were appearing for two companies with the same website. So, I came up with the following approach for a merge function:

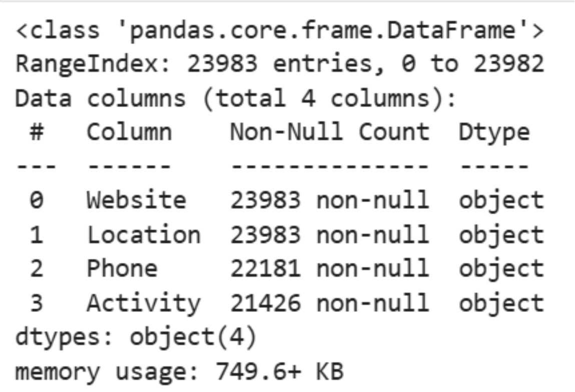
- If all the data is empty, return empty.

- If the data is more than 90% similar using fuzzy matching, return one of the values.

- If there is a majority of identical values, keep only one of the identical values. This applies when, for example, two values are the same, and one is completely different.

- If the values are not similar at all, keep the longest one, which, in my opinion, will be the most detailed.

I also noticed there were duplicates in the phone numbers, but here I think I need more data to better group them. Finally, I was left with the following data:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23983 entries, 0 to 23982
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Website   23983 non-null  object
 1   Location  23983 non-null  object
 2   Phone     22181 non-null  object
 3   Activity  21426 non-null  object
dtypes: object(4)
memory usage: 749.6+ KB
```

I also have a few modifications in mind:

- We could delete all rows where there is at least one NA value (df.dropna()).

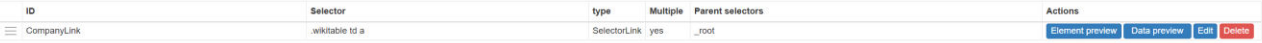
- Group by phone number and apply fuzzy matching on the websites.

In the end, I was left with approximately 24k companies.

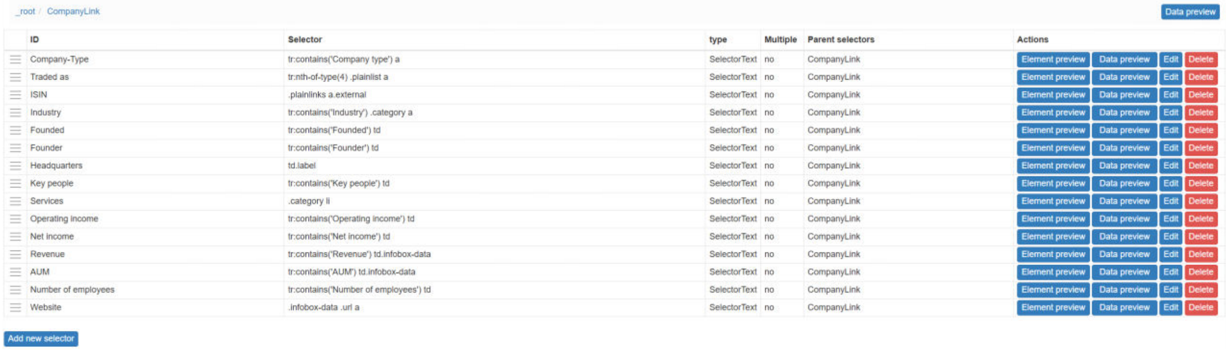
Have a good day! :)

Project #2 - simple scraper (optional)

I correctly installed the extension, then created a new sitemap where I added the link https://en.wikipedia.org/wiki/FTSE_100_Index and gave it a name. After that, I added a new selector in which I selected the entire table with the company links from the main link, set the selector to multiple, and saved it. After that, in Company-Link (where all the companies are located), I created sub-selectors for the separate company data and for retrieving the information from the box on the right side. Then I clicked Scrape to display the table, and afterwards I exported it in CSV format.



	ID	Selector	type	Multiple	Parent selectors	Actions
☰	CompanyLink	.wikitable td a	SelectorLink	yes	_root	Element preview Data preview Edit Delete

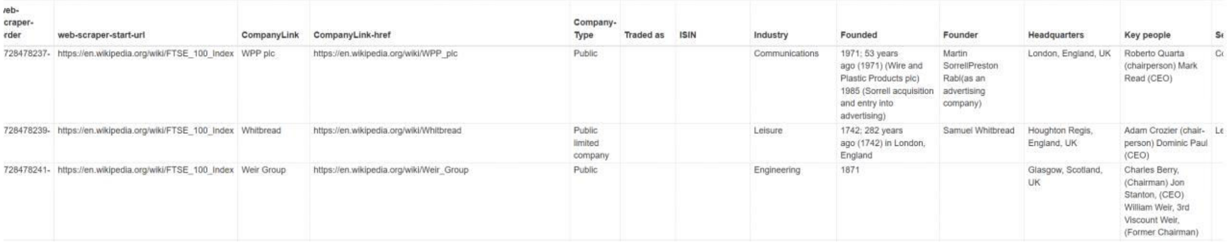


_root / CompanyLink

Data preview

	ID	Selector	type	Multiple	Parent selectors	Actions
☰	Company-Type	tr:contains('Company type') a	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Traded as	tr:nth-of-type(4) .plainlist a	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	ISIN	.plainlinks a.external	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Industry	tr:contains('Industry') .category a	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Founded	tr:contains('Founded') td	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Founder	tr:contains('Founder') td	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Headquarters	td.label	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Key people	tr:contains('Key people') td	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Services	.category li	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Operating income	tr:contains('Operating income') td	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Net income	tr:contains('Net income') td	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Revenue	tr:contains('Revenue') td.infobox-data	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	AUM	tr:contains('AUM') td.infobox-data	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Number of employees	tr:contains('Number of employees') td	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete
☰	Website	.infobox-data .url a	SelectorText	no	CompanyLink	Element preview Data preview Edit Delete

Add new selector

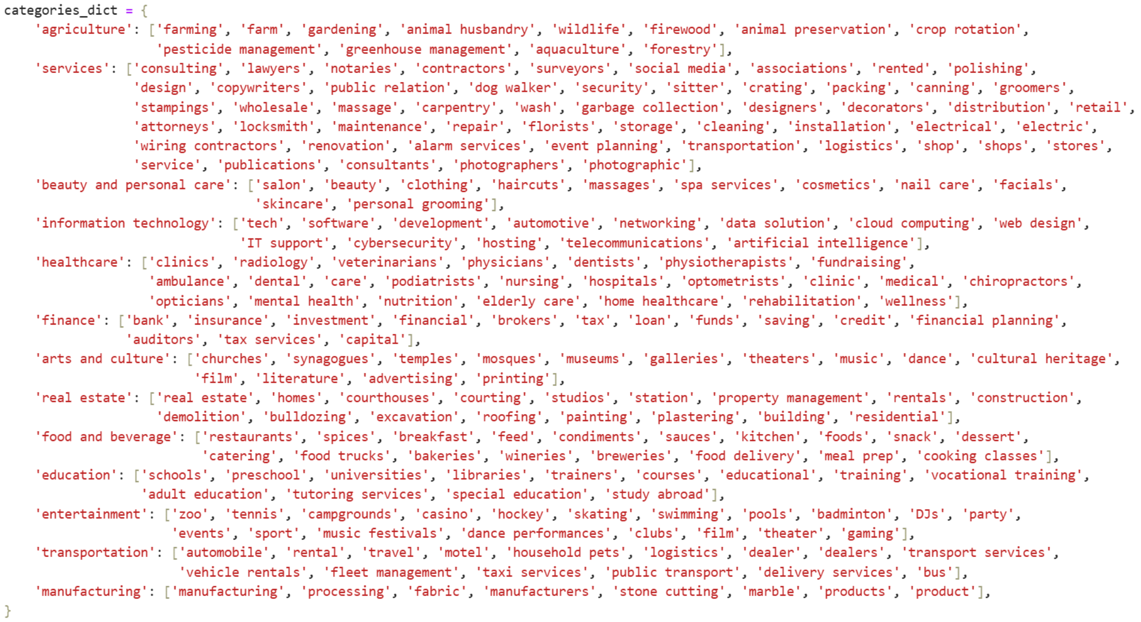


/eb-craper-rder	web-scraper-start-url	CompanyLink	CompanyLink-href	Company-Type	Traded as	ISIN	Industry	Founded	Founder	Headquarters	Key people	Se
728478237-	https://en.wikipedia.org/wiki/FTSE_100_Index	WPP plc	https://en.wikipedia.org/wiki/WPP_plc	Public			Communications	1971; 53 years ago (1971) (Wire and Plastic Products plc) 1985 (Sorrell acquisition and entry into advertising)	Martin SorrellPreston Rabl(as an advertising company)	London, England, UK	Roberto Quarta (chairperson) Mark Read (CEO)	Co
728478239-	https://en.wikipedia.org/wiki/FTSE_100_Index	Whitbread	https://en.wikipedia.org/wiki/Whitbread	Public limited company			Leisure	1742; 282 years ago (1742) in London, England	Samuel Whitbread	Houghton Regis, England, UK	Adam Crozier (chair-person) Dominic Paul (CEO)	Le
728478241-	https://en.wikipedia.org/wiki/FTSE_100_Index	Weir Group	https://en.wikipedia.org/wiki/Weir_Group	Public			Engineering	1871		Glasgow, Scotland, UK	Charles Berry, (Chairman) Jon Stanton, (CEO) William Weir, 3rd Viscount Weir, (Former Chairman)	

Project #3 - mitigate different taxonomies (optional)

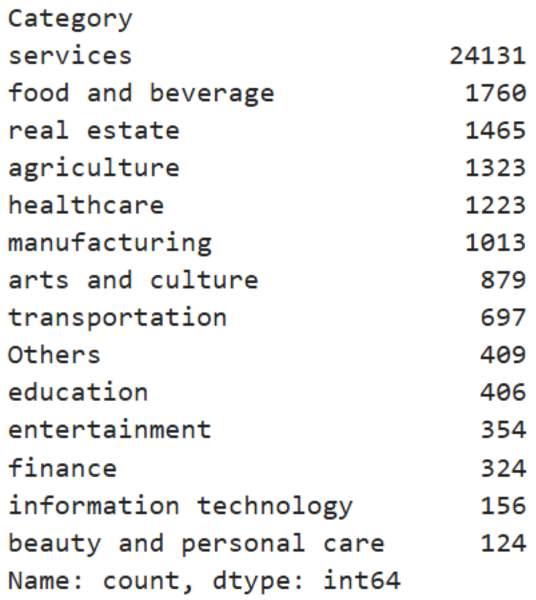
Essentially, I need to organize all this data into categories. I started by brainstorming potential categories and, upon examining the CSV, I noticed entries related to stores, restaurants, salons, DJs, installation services, construction, hosting, energy, etc. Initially, I considered categories like Agriculture, Services, Beauty, Finance, but there turned out to be more. For each category, I defined subcategories; for example, under Services, I included terms like consulting, notaries, polishing, creating a comprehensive list of categories and subcategories.

Next, I developed a simple function in Python that operates as follows: it combines "Activity in Taxonomy 1" with "Activity in Taxonomy2" to search for subcategories. If a match is found, it creates a separate column called "Category" that retains only the name of the category corresponding to the identified subcategory. If no match is found, the entry is automatically categorized as "Other."

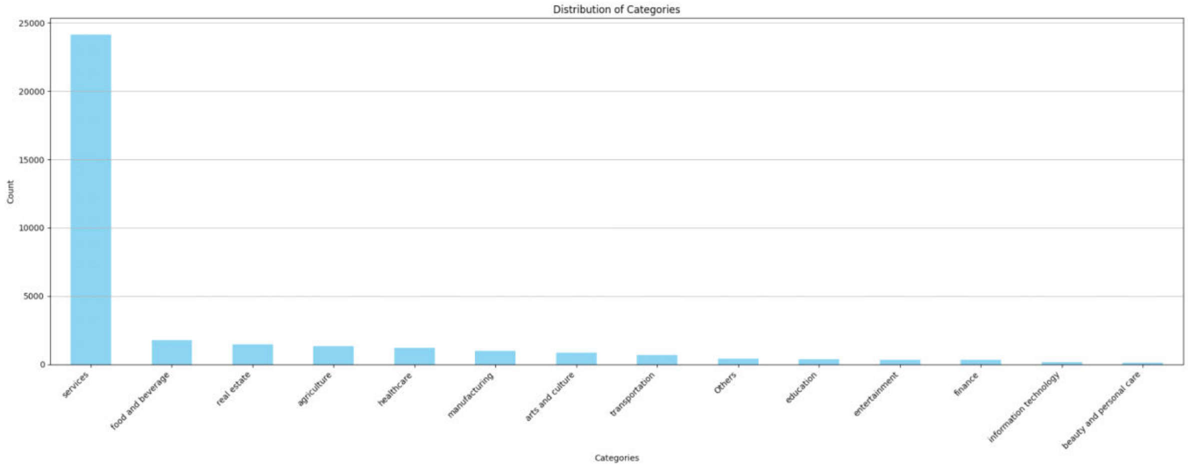


```python
categories_dict = {
    'agriculture': ['farming', 'farm', 'gardening', 'animal husbandry', 'wildlife', 'firewood', 'animal preservation', 'crop rotation',
                    'pesticide management', 'greenhouse management', 'aquaculture', 'forestry'],
    'services': ['consulting', 'lawyers', 'notaries', 'contractors', 'surveyors', 'social media', 'associations', 'rented', 'polishing',
                 'design', 'copywriters', 'public relation', 'dog walker', 'security', 'sitter', 'crating', 'packing', 'canning', 'groomers',
                 'stampings', 'wholesale', 'massage', 'carpentry', 'wash', 'garbage collection', 'designers', 'decorators', 'distribution', 'retail',
                 'attorneys', 'locksmith', 'maintenance', 'repair', 'florists', 'storage', 'cleaning', 'installation', 'electrical', 'electric',
                 'wiring contractors', 'renovation', 'alarm services', 'event planning', 'transportation', 'logistics', 'shop', 'shops', 'stores',
                 'service', 'publications', 'consultants', 'photographers', 'photographic'],
    'beauty and personal care': ['salon', 'beauty', 'clothing', 'haircuts', 'massages', 'spa services', 'cosmetics', 'nail care', 'facials',
                                 'skincare', 'personal grooming'],
    'information technology': ['tech', 'software', 'development', 'automotive', 'networking', 'data solution', 'cloud computing', 'web design',
                               'IT support', 'cybersecurity', 'hosting', 'telecommunications', 'artificial intelligence'],
    'healthcare': ['clinics', 'radiology', 'veterinarians', 'physicians', 'dentists', 'physiotherapists', 'fundraising',
                   'ambulance', 'dental', 'care', 'podiatrists', 'nursing', 'hospitals', 'optometrists', 'clinic', 'medical', 'chiropractors',
                   'opticians', 'mental health', 'nutrition', 'elderly care', 'home healthcare', 'rehabilitation', 'wellness'],
    'finance': ['bank', 'insurance', 'investment', 'financial', 'brokers', 'tax', 'loan', 'funds', 'saving', 'credit', 'financial planning',
                'auditors', 'tax services', 'capital'],
    'arts and culture': ['churches', 'synagogues', 'temples', 'mosques', 'museums', 'galleries', 'theaters', 'music', 'dance', 'cultural heritage',
                         'film', 'literature', 'advertising', 'printing'],
    'real estate': ['real estate', 'homes', 'courthouses', 'courting', 'studios', 'station', 'property management', 'rentals', 'construction',
                    'demolition', 'bulldozing', 'excavation', 'roofing', 'painting', 'plastering', 'building', 'residential'],
    'food and beverage': ['restaurants', 'spices', 'breakfast', 'feed', 'condiments', 'sauces', 'kitchen', 'foods', 'snack', 'dessert',
                          'catering', 'food trucks', 'bakeries', 'wineries', 'breweries', 'food delivery', 'meal prep', 'cooking classes'],
    'education': ['schools', 'preschool', 'universities', 'libraries', 'trainers', 'courses', 'educational', 'training', 'vocational training',
                  'adult education', 'tutoring services', 'special education', 'study abroad'],
    'entertainment': ['zoo', 'tennis', 'campgrounds', 'casino', 'hockey', 'skating', 'swimming', 'pools', 'badminton', 'DJs', 'party',
                      'events', 'sport', 'music festivals', 'dance performances', 'clubs', 'film', 'theater', 'gaming'],
    'transportation': ['automobile', 'rental', 'travel', 'motel', 'household pets', 'logistics', 'dealer', 'dealers', 'transport services',
                       'vehicle rentals', 'fleet management', 'taxi services', 'public transport', 'delivery services', 'bus'],
    'manufacturing': ['manufacturing', 'processing', 'fabric', 'manufacturers', 'stone cutting', 'marble', 'products', 'product'],
}
```

After saving the results in a CSV file, I plan to check how many times "Other" appears in the Category column. Additionally, I will explore whether I can link certain activities from the two columns to integrate them into an existing subcategory or, if necessary, create a new category and define relevant subcategories.



```
Category
services                     24131
food and beverage             1760
real estate                   1465
agriculture                   1323
healthcare                    1223
manufacturing                 1013
arts and culture               879
transportation                 697
Others                         409
education                      406
entertainment                  354
finance                        324
information technology         156
beauty and personal care       124
Name: count, dtype: int64
```



*** I also tried an idea with fuzzy matching, namely to create a column that combines the two. If there is more than 75% similarity, we assume it will likely have the subcategory in the combined column. However, if there is less than 75%, we should take the taxonomy column from the client, assuming it would be better understood, and search for the subcategory only in the client's column. However, there were too many differences. ***