Data Analysis

CDP-Climate Change

Handling Missing Data

Unlocking Climate Solutions

Abdel-Nasser Ateeq

March 22, 2021

Contents

List of Figures

1 Abstract

The climate is a worthwhile, lucrative field if it is taken care of. Climate changes strongly affect people's lifestyles, i.e. the tourism sector, consumed energy, levels of emissions, social sector, health sector, and more. Accordingly, communities are drifted to study climate changes due to mitigating their risks and trying to find opportunities that could help people to adapt to these changes. In this article, a deep study is accomplished in furtherance of concluding the relationship between climate changes and its effects on people's lives, which results in mitigating its side effects and discovering opportunities.

2 Introduction

The climate changes control people's lives and force them to change their lifestyle due to its changes. Climate changes have a strong effects on people's lives; e.g. economic growth, health services, energy consumption, tourism growth, social sector, transport sector, and water supply. Accordingly, the role of governments and societies is to set rules, set mitigation plans, taking mitigation actions, do a risk assessment, and set climate adaptation plan, in order to saving lives, detect new opportunities that help in economic growth, mitigate climate hazards, increase energy efficiency, and help people to adapt to these changes.

Hence, the climate changes are analyzed, and a correlation is calculated between the collected features to discover any related features that could help people, governments, societies to adapt to these changes.

3 Problem Statement

CDP company [1] has been collected a large data-set for the last three years by accomplishing questioners. In these questioners, the people's opinions about climate change effects were collected.

In this research, this data-set is analyzed in detail to anticipate the actions that could be taken to reduce climate hazards. Also, the intersection between environmental issues and social issues is discussed. Generally, this analysis is trying to answer the following questions:

1. How do you help cities adapt to a rapidly changing climate amidst a global pandemic, but do it in a way that is socially equitable?

2. What are the projects that can be invested in that will help pull cities out of a recession, mitigate climate issues, but not perpetuate racial/social inequities?

3. What are the practical and actionable points where city and corporate ambition join, i.e. where do cities have problems that corporations affected by those problems could solve, and vice versa?

4. How can we measure the intersection between environmental risks and social equity, as a contributor to resiliency?

4 Data-set Description

The collected data-set consists of three sub-data-sets which are countries, corporations, and supplementary data. The content of these data-sets are shown in detail in Figures 1, 2, and 3.

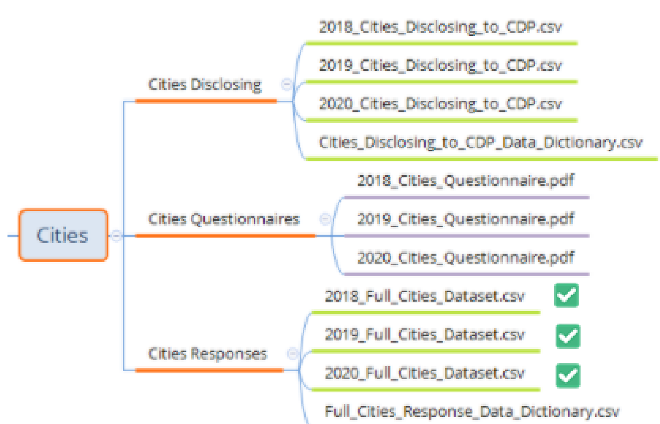

Figure 1: The Content of Cities Data-Set

The main data-sets that will be analyzed deeply and the relations will be discovered from are listed below:

- 2018_Full_Climate_Change_Dataset.csv

- 2019_Full_Climate_Change_Dataset.csv
- 2020_Full_Climate_Change_Dataset.csv
- 2018_Full_Water_Security_Dataset.csv
- 2019_Full_Water_Security_Dataset.csv
- 2020_Full_Water_Security_Dataset.csv
- 2018_Full_Cities_Dataset.csv
- 2019_Full_Cities_Dataset.csv
- 2020_Full_Cities_Dataset.csv

Notice that these files appear with a check-mark beside, in the previously mentioned figures. Each data-set is analyzed deeply in section 5.

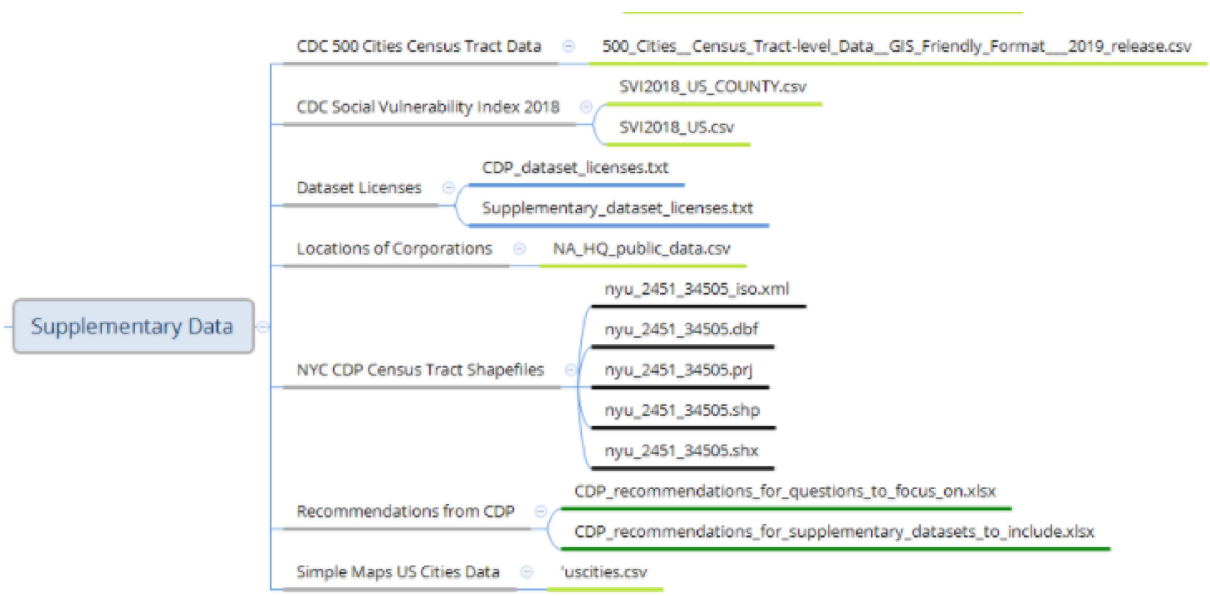

Figure 2: The Content of Supplementary data

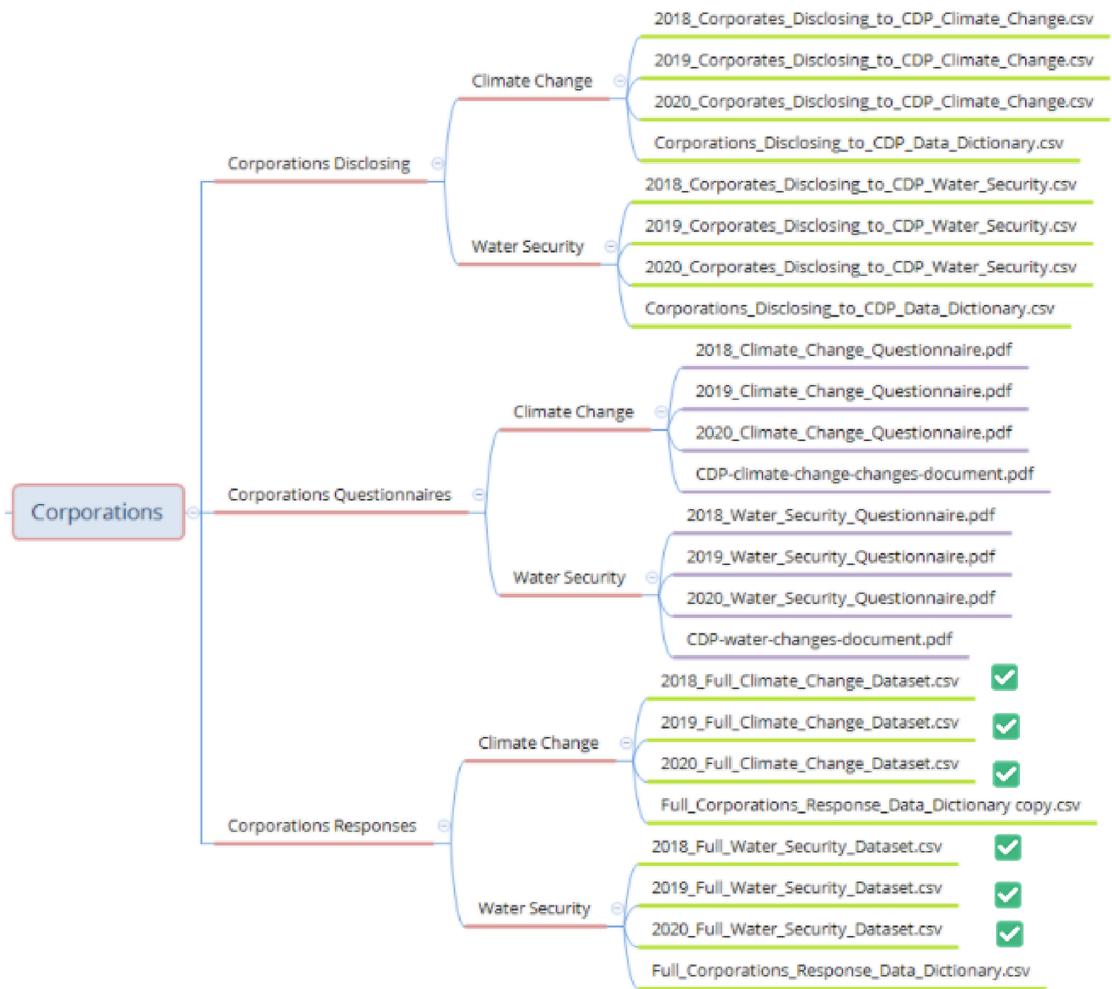

Figure 3: The Content of Corporations Data-Set

5 Experimental Results

Data mining is a complex technique that consists of a set of steps that the data has to pass through to discover the hidden patterns. The first step of data mining is to handle the missing values to prepare the data for the subsequent processes. Each sub-data-set is handled independently, by arranging to subsections. Also, each feature is handled in an appropriate way due to its importance and its level of usefulness.

5.1 Handling Missing Data

First of all, the required libraries have to be imported as shown in Figure 4, as well the path of the main folder as shown in Figure 5. Also, a set of functions are established in order to facilitate the handling-missing process, as shown in Figures 6, 7, 8, 9, and 10.

```
# Basic libraries
import os
import pandas as pd
import numpy as np

# Visualization libraries
import matplotlib.pyplot as plt
```

Figure 4: Importing the required libraries

```
DATA_PATH = ".\Course-Project\cdp-unlocking-climate-solutions"

list(os.listdir(DATA_PATH))
```

```
['Cities', 'Corporations', 'Supplementary Data']
```

Figure 5: Import data-set path

```
def missing_ratio(df):
    DataFrame = df.isna().sum().rename('Num_0f_missing')

    # Rename the index to Feature
    DataFrame = DataFrame.reset_index().rename(columns={'index':'Feature'})

    Num_of_samples = df.shape[0]
    DataFrame['Percent_of_msising'] = DataFrame['Num_0f_missing'] / Num_of_samples

    # Select the features that have a missing values
    DataFrame = DataFrame[DataFrame['Num_0f_missing'] != 0]
    DataFrame = DataFrame.sort_values(by=['Percent_of_msising'],ascending=False)
    return DataFrame
```

Figure 6: Missing-ratio function

```python
def handle_missing(df,list_columns,handle_type,value):

    if handle_type == "bfill":
        # no inplace
        df[list_columns] = df[list_columns].fillna(method='bfill')

    elif handle_type == "ffill":
        df[list_columns] = df.loc[:,list_columns].fillna(method='ffill')

    elif handle_type == "mean":
        df[list_columns] = df[list_columns].fillna(df.mean()).round(2)

    elif handle_type == "dropna0":
        df[list_columns] = df[list_columns].dropna(axis=0, how='any')

    elif handle_type == "dropna1":
        df[list_columns] = df[list_columns].dropna(axis=1, how='any')

    else:
        df[list_columns] = df.loc[:,list_columns].fillna(value)
```

Figure 7: Handle-missing function

```python
def high_missing_columns(df):
    stutus  = missing_ratio(df)
    Columns = stutus[stutus['Percent_of_msising'] > 0.7]['Feature'].values.tolist()
    df = df.drop(Columns,1)
    return df
```

Figure 8: High-missing-columns function

```python
def fill_using_relation(df,primary_industry,primary_sector,questionnaire_sector):
    cols_related = ['primary_sector','primary_industry','primary_questionnaire_sector']

    a = df[(df['primary_industry'] == primary_industry)\
                            & (df['primary_sector']== primary_sector)\
                            & (df['primary_questionnaire_sector'].isna())].loc[:,cols_related]

    #Get a list of rows that satisfy the condition
    row_list = a.index.tolist()
    #Replace the Nan values with the new value
    df.loc[row_list,'primary_questionnaire_sector'] = questionnaire_sector
```

Figure 9: Fill using relation function

```python
def merge_df(dataframes):

    if all([set(dataframes[0].columns) == set(df.columns) for df in dataframes]):

        result = pd.concat(dataframes,ignore_index=True)
        return result
    else:
        print('Some have different columns')
```

Figure 10: Merge data frame function

5.1.1 Cities

Cities Responses

Since the cities responses look like to have the same features but for three different years, the tricky way is to merge all of them in one data frame. So, after importing those files as shown in Figure 11 to check if they have the same features, the script in Figure 12 is used.

```
full_ct_dataset_2020 = pd.read_csv(f"{DATA_PATH}/Cities/Cities Responses/2020_Full_Cities_Dataset.csv")
full_ct_dataset_2019 = pd.read_csv(f"{DATA_PATH}/Cities/Cities Responses/2019_Full_Cities_Dataset.csv")
full_ct_dataset_2018 = pd.read_csv(f"{DATA_PATH}/Cities/Cities Responses/2018_Full_Cities_Dataset.csv")
```

Figure 11: Importing cities-responses files

```
dataframes = [full_ct_dataset_2020,full_ct_dataset_2019,full_ct_dataset_2018]
df_cities_response = merge_df(dataframes)
```

Figure 12: Check if merging is possible

By using the missing-ratio function, the missing ratio is shown in Figure 13. Also, there are two features with a missing ratio of more than 70%. Therefore, those features have to be deleted as shown in Figure 14.

```
stutus_cities_response = missing_ratio(df_cities_response)
stutus_cities_response
```

	Feature	Num_0f_missing	Percent_of_msising
16	File Name	1537241	0.996593
15	Comments	1497767	0.971002
13	Row Name	925226	0.599824
14	Response Answer	401188	0.260090
6	Parent Section	274302	0.177830
11	Column Name	74091	0.048033

Figure 13: Ratio of missing in cities-responses

Concerning the rest of the features, there is no clear pattern that could be used to fill the missing values. Also, the rows of those corresponding features which their values are missing, have useful information that could be used for patterns mining. Then, the best solution is to

```
#--------------------------------------------------------
#drop fatures with missing ratio more than 70% -------
#--------------------------------------------------------
df_cities_response = high_missing_columns(df_cities_response)
missing_ratio(df_cities_response)
```

	Feature	Num_0f_missing	Percent_of_msising
13	Row Name	925226	0.599824
14	Response Answer	401188	0.260090
6	Parent Section	274302	0.177830
11	Column Name	74091	0.048033

Figure 14: Drop features with high missing ratio

fill those rows with constant "Missing" as shown in Figure 15. Besides, it's noticed that there are no missing values anymore in this data-set.

```
fill_with_missing = ['Row Name','Response Answer','Parent Section','Column Name']
handle_missing(df_cities_response,fill_with_missing,"","Missing")
missing_ratio(df_cities_response)
```

Feature	Num_0f_missing	Percent_of_msising

Figure 15: Fill with constant

Cities Disclosing

Fist, all of the data-set is merged in one data frame, because they have the same features using the same scripts in the first section. To check the missing ratio the script in Figure 16 is used.

```
missing_ratio(df_cities_Disclosing)
```

	Feature	Num_0f_missing	Percent_of_msising
11	City Location	438	0.211799
3	City	352	0.170213
9	Population	284	0.137331

Figure 16: Ratio of missing in cities-disclosing

When taking a deep look at the features, it's noticed that the most important ones are city, city location, and population. So, if there are any rows where the value of those three features are missing, they should be dropped as shown in Figures 17 and 18. Then, the ratio of missing will be as shown in Figure 19.

```
#Check the number of rows where all of the three features have missing values
cols = ['City','Population','City Location']
three_col = df_cities_Disclosing[df_cities_Disclosing[cols].isna().all(1)]
three_col.head(3)
```

Figure 17: Check where the three features are missing

```
#Since there are 64 rows have missing values in the 3 columns ['City','Population','City Location']
#which are the most important features in the data frame, so there is no need to keep saving those rows
df_cities_Disclosing.dropna(subset=cols,inplace=True, how='all')
missing_ratio(df_cities_Disclosing)
```

Figure 18: Drop rows where the three features are missing

	Feature	Num_0f_missing	Percent_of_msising
11	City Location	374	0.186627
3	City	288	0.143713
9	Population	220	0.109780

Figure 19: Missing ratio after dropping

Even though there are almost 374 rows where city or city location values are missing, and there are no useful features that could help to predict the values of those cells, we still can get the benefit of them by keeping the population. So, the best solution is to fill them with constant "Missing" as shown in Figure 20.

```
Fill_with_constant_set1 = ['City','City Location']
handle_missing(df_cities_Disclosing,Fill_with_constant_set1,"","Missing")
missing_ratio(df_cities_Disclosing)
```

	Feature	Num_0f_missing	Percent_of_msising
9	Population	220	0.10978

Figure 20: Filling missing with constant

After that, there still 220 rows where the population is missing, but we can still get the benefit of them by filling them with constant "0" as shown in Figure 21. Also, it's noticed there is no missing values anymore.

```
#Since there are 220 cells where population values are missing,
#there still the values of city and city location exist which could be benefit somehow.
#So, the best solution is to fill with constant ZERO
handle_missing(df_cities_Disclosing,['Population'],"",0)
missing_ratio(df_cities_Disclosing)
```

Feature	Num_0f_missing	Percent_of_msising

Figure 21: Filling missing with constant

5.1.2 Corporations/Corporations Disclosing

Climate Change

The manner for handling missing data is the same as the previous section. First, all of the data-set is merged in one data frame, because they have the same features. By using the missing-ratio function, the missing ratio is shown in Figure 22. Also, it's noticed that the "region" feature has a missing ratio more than 80%. Therefore, this feature has to be deleted as shown in Figure 23.

```
#Check the missing ratio
missing_ratio(Corp_Disclosing_climateChange)
```

	Feature	Num_0f_missing	Percent_of_msising
4	region	2561	0.986898
7	samples	987	0.380347
21	primary_ticker	836	0.322158
22	tickers	832	0.320617
9	minimum_tier	348	0.134104
20	primary_questionnaire_sector	23	0.008863
17	primary_activity	10	0.003854
18	primary_sector	10	0.003854
19	primary_industry	10	0.003854

Figure 22: Ratio of missing in climate change

After taking a deep look at the features and the missing ratio, it's noticed that there are three columns with the same missing ratio which are primary_activity, primary_sector, and primary_industry.Besides, it's noticed that no feature has the same ratio of missing which could help to predict their missing values. So, the best solution is to delete those rows.

```
#-----------------------------------------------------
#drop fatures with missing ratio more than 70% -------
#-----------------------------------------------------
Corp_Disclosing_climateChange = high_missing_columns(Corp_Disclosing_climateChange)
missing_ratio(Corp_Disclosing_climateChange)
```

Figure 23: Drop features with high missing ratio

Concerning the primary_questionnaire_sector feature, it's obviously related to primary_sector and primary_industry features as noticed in Figure 24. Then, a set of scripts are accomplished to understand this relation and picking up the possible values for the missing cells as shown in 25.

```
#Trying to predict the value of 'primary_questionnaire_sector' by checking its relation with
# the two columns'primary_sector','primary_industry'
cols_related = ['primary_sector','primary_industry','primary_questionnaire_sector']
Corp_Disclosing_climateChange[Corp_Disclosing_climateChange['primary_questionnaire_sector'].isna()]\
.loc[:,cols_related]
```

	primary_sector	primary_industry	primary_questionnaire_sector
1054	Rail transport	Transportation services	NaN
1105	Paper products & packaging	Manufacturing	NaN
1150	Intermodal transport & logistics	Transportation services	NaN
1243	Wood & paper materials	Materials	NaN
1331	Powered machinery	Manufacturing	NaN
1403	Paper products & packaging	Manufacturing	NaN
1436	Paper products & packaging	Manufacturing	NaN
1498	Transportation equipment	Manufacturing	NaN
1517	Transportation equipment	Manufacturing	NaN
1526	Energy utility networks	Infrastructure	NaN
1537	Transportation equipment	Manufacturing	NaN
1561	Wood & paper materials	Materials	NaN
1618	Chemicals	Materials	NaN

Figure 24: Predict the missing values in primary_questionnaire_sector

Then for every row of the 19 rows which their primary_questionnaire_sector value is missing, a script is generated to fill the missing cells with the appropriate constant as shown in Figure 26, and the same is done for all rows where primary_questionnaire_sector value is missing.

Concerning the minimum_tier feature, it's noticed that it is related to the selected_tier feature as shown in Figure 27, but there is no clear sequence to predict one from the other. In

```
Corp_Disclosing_climateChange[Corp_Disclosing_climateChange['primary_industry'] == "Transportation services"]\
.groupby(['primary_sector','primary_industry','primary_questionnaire_sector']).size().reset_index(name='Count')
```

	primary_sector	primary_industry	primary_questionnaire_sector	Count
0	Air transport	Transportation services	Transport services	19
1	Intermodal transport & logistics	Transportation services	General	23
2	Intermodal transport & logistics	Transportation services	Transport services	13
3	Marine transport	Transportation services	Transport services	7
4	Rail transport	Transportation services	Transport services	21
5	Road transport	Transportation services	Transport services	11

Figure 25: Possible values when primary_industry is transportation services

```
fill_using_relation(Corp_Disclosing_climateChange,'Transportation services','Rail transport','Transport services')
missing_ratio(Corp_Disclosing_climateChange)
```

Figure 26: Fill the missing cells with the appropriate constant

other hand, it's obvious from Figure 28 that the most frequent values are 'Climate Change - Full' and 'Climate Change - Minimum'. So, it's appropriate to choose forward filling, as shown in Figure 29.

	minimum_tier	selected_tier
0	Climate Change - Minimum	Climate Change - Full
1	Climate Change - Minimum	Climate Change - Full
2	Climate Change - Full	Climate Change - Minimum
3	Climate Change - Full	Climate Change - Minimum
4	Climate Change - Full	Climate Change - Minimum
5	Climate Change - Full	Climate Change - Minimum
6	Climate Change - Full	Climate Change - Minimum
7	Climate Change - Full	Climate Change - Minimum
8	Climate Change - Full	Climate Change - Minimum
9	Climate Change - Minimum	Climate Change - Minimum
10	Climate Change - Minimum	Climate Change - Minimum
11	Climate Change - Minimum	Climate Change - Minimum
12	Climate Change - Minimum	Climate Change - Minimum
13	Climate Change - Minimum	Climate Change - Minimum

Figure 27: The relation between minimum_tier and selected_tier

Last but not least, according to primary_ticker and tickers features, a script is run to check if there are any relation between them, as shown in Figure 30. It's clear that the two features

```
#Display the most frequent values
Corp_Disclosing_climateChange['minimum_tier'].value_counts()[:2].index.tolist()
```

```
['Climate Change - Full', 'Climate Change - Minimum']
```

Figure 28: The most frequent values in minimum_tier

```
#As shown that each cell is related with the previous cell
#So, we can use ffill
handle_missing(Corp_Disclosing_climateChange,['minimum_tier'],"ffill",0)
```

Figure 29: Using filling forward for minimum_tier

have the same value per row, but the number of rows where both features are missing is roughly the same, so there is no ability to predict one value from the other. Then, the best solution is to fill the missing cells for those two features with constant "Missing".

	primary_ticker	tickers
0	GEI CN	GEI CN
1	OSB CN	OSB CN
37	BRC US	BRC US
38	GSIT US	GSIT US
44	CCL/B CN	CCL/A CN, CCL/B CN
...	...	...
2590	BCE CN	BCE CN
2591	BB CN	BB CN
2592	WSP CN	WSP CN
2593	MG CN	MG CN
2594	DOO CN	DOO CN

Figure 30: Check if there is any relation

Concerning the samples feature, it's noticed from Figure 31 that the filling forward option is not desirable, since the error ratio will be high; i.e. if Canada's rows filled with USA's solution and vice versa. Figure 32 shows that the samples feature is related with country, thus the most frequent value is used to fill the missing when the country is USA,as shown in Figure 33 and the same for Canada.

```
Corp_Disclosing_climateChange.loc[230:280,['samples','country']]
```

	samples	country
230	NaN	United States of America
231	Benchmark, CCGR1800, Continuity Climate Chang...	United States of America
232	Benchmark, CCGR1800, Continuity Climate Chang...	United States of America
233	Benchmark, CCGR1800, Continuity Climate Chang...	United States of America
234	NaN	Canada
235	NaN	Canada
236	NaN	Canada
237	NaN	Canada
238	NaN	Canada
239	NaN	Canada
240	NaN	Canada

Figure 31: Example declare why the forward filling isn't desirable

```
df_sample = Corp_Disclosing_climateChange[(Corp_Disclosing_climateChange['samples'].notna())\
                              & (Corp_Disclosing_climateChange['country'] == 'United States of America' ) ]\
.groupby(['samples','country']).size().reset_index(name='Count')

df_sample = df_sample.sort_values(by=['Count'],ascending=False).head(15)
df_sample
```

	samples	country	Count
60	Benchmark, CCGR1800, Continuity Climate Chang...	United States of America	164
162	Self selected companies climate change	United States of America	161
48	Benchmark, CCGR1800, Continuity Climate Chang...	United States of America	115
36	Benchmark, CCGR1000, CCGR1800, FTSE All-World...	United States of America	71
62	Benchmark, CCGR1800, Continuity Climate Chang...	United States of America	60

Figure 32: Declare the relation between samples and country

```
#Condition #10
cols_related = ['samples','country']
k = Corp_Disclosing_climateChange[(Corp_Disclosing_climateChange['samples'].isna())\
                             & (Corp_Disclosing_climateChange['country']=='United States of America')].loc[:,cols_related]
k
```

...

```
Fill_samples_USA = df_sample['samples'].head(1).values[0]
#Get a list of rows that satisfy the tenth condition
row_list = k.index.tolist()

#Replace the Nan values with the values of the variable Fill_samples_USA
Corp_Disclosing_climateChange.loc[row_list,'samples'] = Fill_samples_USA
```

Figure 33: Fill missing in samples where the country is USA

Water Security

The manner for handling missing data is the same as the previous section. First, all of the data-set is merged in one data frame, because they have the same features. Besides, these features the same as corporations-disclosing-climate-change data-set features. So, all of the features in this data-set are treated in the same manner as corporations-disclosing-climate-change data-set.

5.1.3 Corporations/Corporations Responses

Climate Change

The manner for handling missing data is the same as the climate change in the previous section. First, all of the data-set is merged in one data frame, because they have the same features. Secondly, the features with missing ratio more than 70% is dropped, then the response_value and row_name is filled with constant 'missing'. Finally, the column_number and data_point_id is filled with constant '0'.

Water Security

First, all of the data-set is merged in one data frame, because they have the same features. Secondly, the features with missing ratio more than 70% is dropped, then the response_value and row_name is filled with constant 'missing' by using the same scripts that used in the previous sections.

5.2 EDA

In the analysis step the question is considered important one and deserves analysis if it has responses for the three years, and its answer does not exceed the allowed length (15 chars).

There are some ambiguous questions that bounded in the following cases:

- Their responses are supposed to be numerical but they are categorical.
- Where their responses are supposed to be explanation (explanation in details), but they are yes/no.

- Where their responses are categorical and numerical at the same time, whereas they are not supposed to be.

  So, to solve this ambiguity, the questions in one of those cases are ignored.

After analysing the cities responses, climate-change, and water-security data-frames it is concluded that:

1. The fuel type 'L' is the most fuel used if compared with others, for the last three years, but it is noticed that; its ratio in 2020 was the least if compared with 2019 and 2018.
2. The cities councils are aware of the climate hazards and have mitigation plans for the last two years, but from people's perspectives, they expected in 2019 there will reduction in GHG emissions. On other hand, they expected in 2020 there will be infrastructure development.
3. People are more aware of the climate hazards for the last year and focus on enhanced resilience and finance secured as adaptation actions.
4. The people agreed that cities councils have published plan that addresses climate change adaptation.
5. The people strongly agreed that their cities face risks to public health or health systems associated with climate change.
6. The people agreed that their cities councils are focused on finding new opportunities by the development of energy efficiency measures and technologies, and the development of a sustainable transport sector.
7. The most diffused language is English since the vast majority of the people were used to fill the questionnaires.
8. The vast majority of people use USD Dollar for their financial transactions, whereas the EUR Euro is the second most one.
9. People agree with about 95% that there is no renewable energy in their cities.

10. Cities councils are emphasizing; extending the green areas.

11. People agree that their local government operations emissions have decreased for the last three years.

12. On the risks over the business side, people have believed in 2018 and 2019 that there will be physical risks in their businesses, but in 2020 they believe that there would be short-term risks.

13. On the opportunities over the business side, the people who agree that there would be short-term benefits for the last year are increased, if compared with 2018 and 2019.

14. The emissions level of HFCs gas has decreased for the last year if compared with 2019, whereas it was noticed that there is a vast decrease in the level of CO2 gas.

15. The vast majority of people depend on the water presence for continuing in their businesses.

16. In the metals and mining sector most of the organizations engaged in mining or processing metals.

6 Conclusion

From the detailed analysis, it is clear that the USD Dollar will be the most dominant currency in the world. Also, there is a clear lack of renewable energy, so, cities councils have to give more effort to find alternatives to adapt to climate risks. Besides, there has to be more attention in providing water resources, since most people's businesses depend on it. Finally, there should be more attention to the health sector, since the people's perspectives displayed that the health sector is not prepared to adapt the climate risks.

References

[1] *CDP*. URL: `http://engineering.purdue.edu/~mark/puthesis`.