**Text Extraction From pdf, image Files**

Code: phase\_1 -test multi(final update).ipynb

Input data: all pdf, image files

Output data: Extracted and translated text

1. **Main code**

phase\_1 -test multi(final update).ipynb

This code uses multiprocess technique to process files. It is recommended to run files seperately, because it stores data in the memory first, which means the memory might be full and the program crashs if too many files have been run. Here we run 1k files at a time.

1. **Other tools**

Deal\_with\_the\_rest\_of\_data.ipynb

Some files cannot be processed normally due to some reasons. This code splits and reruns those files.

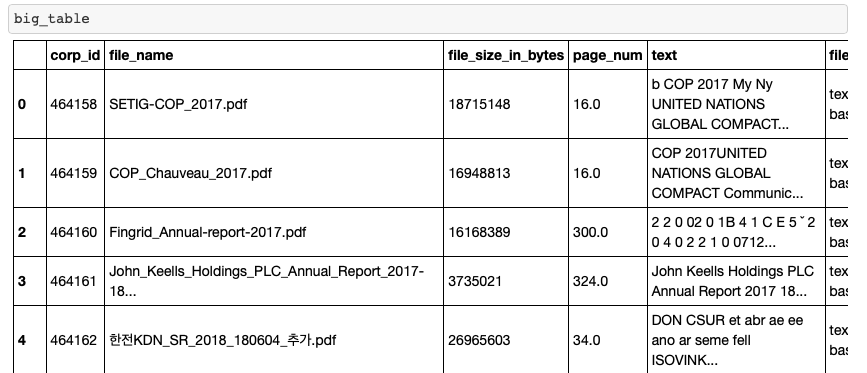
Wash.ipynb

While we process a image file, we create a pdf version behind it and extract the texts. However, if we run our code for this image file at the second time, the code will read both of the image file and pdf version of it. This can waste a huge amount of time. Therefore, it is recommended to remove all extra files from original folders. This code removes all 'converted.pdf' and 'extractable.pdf' files from folders.

Sample\_data.ipynb

We used this code to sample some data to test ‘phase\_1 -test multi(final update).ipynb’

1. **Sample result**

****

**2017 Key Words Extraction from Translated Text**

Code: 1\_unctad\_2017\_extraction.ipynb

1. **2018 data after extraction**

2018 data after extraction was found in the big computer, and is got by the same code (1\_unctad\_2017\_extraction.ipynb) on the translated text.

* unctad\_2018\_old.xlsx



1. **2017 corresponding data before extraction**

Input data (all translated data was found in big computer):

* Phase2.5\_2017\_p1\_EN.csv,
* Phase2.5\_2017\_EN\_p2.csv,
* Phase2.5\_2017\_p1.csv,
* Phase2.5\_2017\_p2.csv

Output data:

* selected\_2017\_co.csv

1. **2017 data after extraction**

Input data:

wikirate\_to\_indicator.csv,

cop\_files\_2017-forGlobalAI 1.xlsx,

selected\_2017\_co.csv,

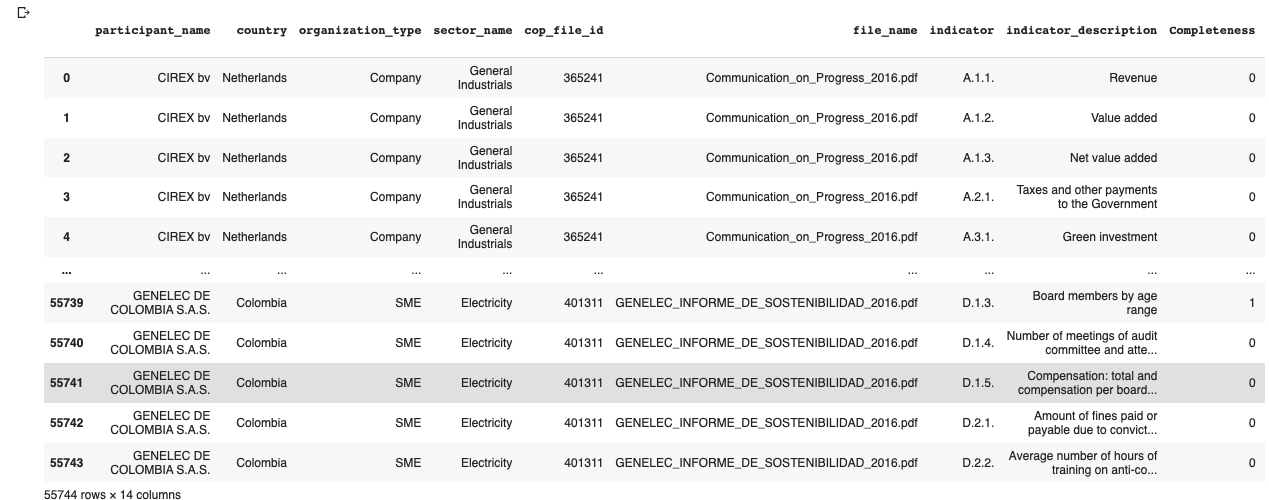
wikirate indicator.txt

Output data:

sent\_df\_2017.csv,

num\_df\_2017.csv,

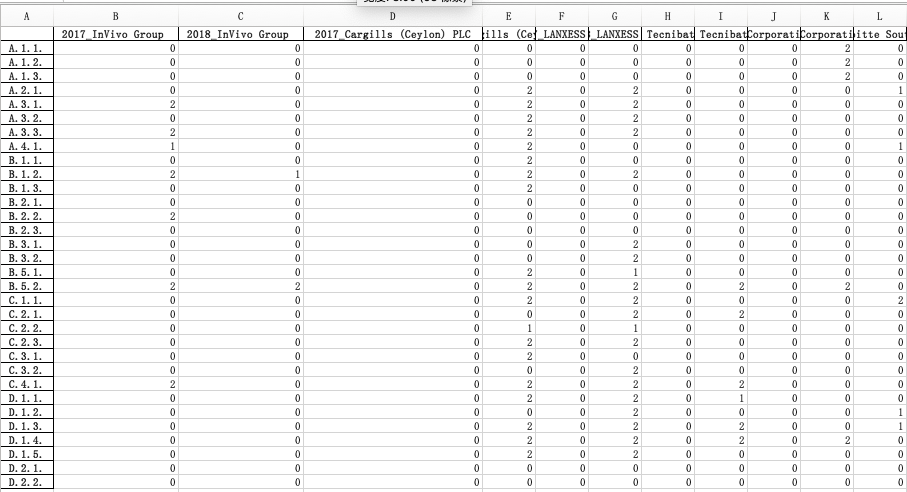
2017\_result.csv



1. **Combine 2017 and 2018 data**

Input: 2017\_result.csv

Output: two\_year\_result.xlsx



**UNCTAD: SCORE AND DASHBOARD**

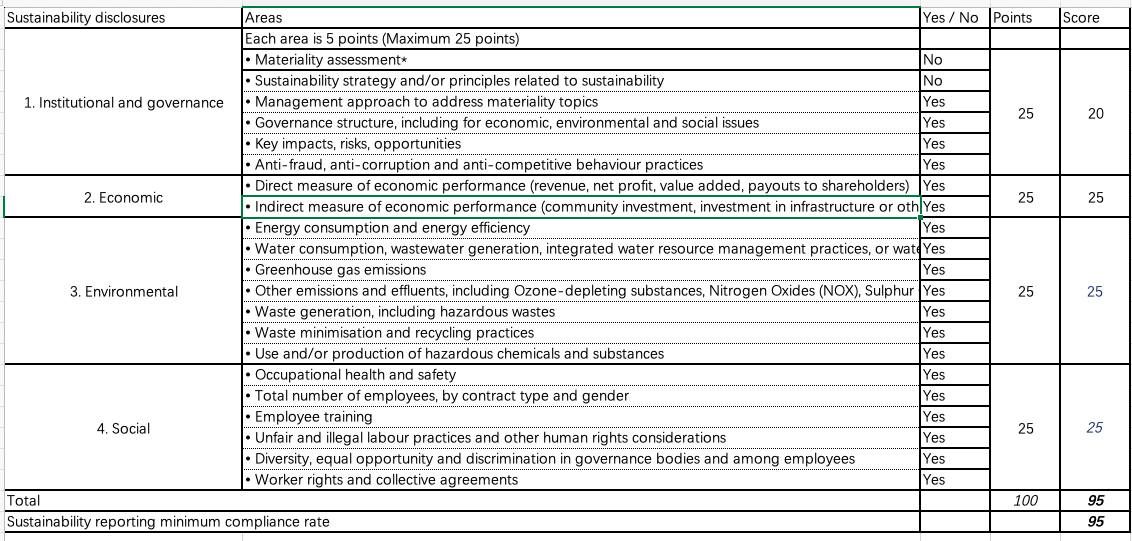
**Code**: 2\_template\_to\_calculate\_score\_for\_each\_corp.ipynb And 3\_UNCTAD\_visualization.ipynb

1. **Data Source:** two\_year\_result.csv; UN FfD COUNTRY n REGIONS - UNSD — Methodology - M49.xlsx

The input file is generated by unctad\_2017.ipynb

1. **Rules we use to calculate the score:** POINTS SYSTEM - Definiton of Minimum REquirement.xlsx

**Goal** is to determine if EACH company meets the minimum criteria. So, if company has at least (1) in each category (4 categories), then it qualifies and the value for that company that year becomes One; otherwise Zero



We can use the weight dictionary to calculate the score

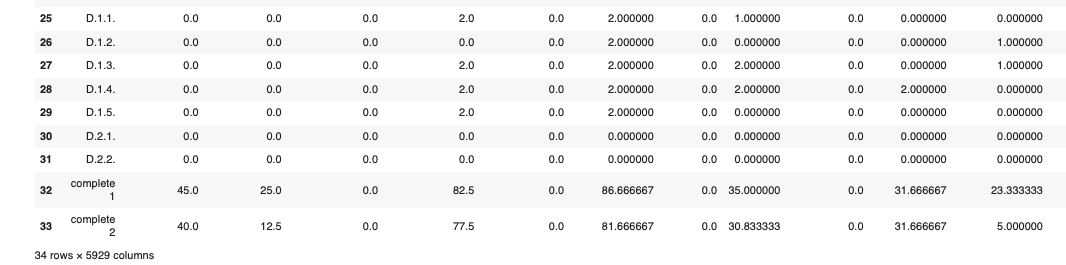
weight = {'1.1':5, '1.2':5, '1.3':5, '1.4':5, '1.5':5,

'2.1':12.5, '2.2':12.5,

'3.1':5, '3.2':5, '3.3':5, '3.4':5, '3.5':5,

'4.1':25/6, '4.2':25/6, '4.3':25/6, '4.4':25/6, '4.5':25/6, '4.6':25/6}

1. **Result of the score calculation result**



Here I considered two scenarios:

1) at least (one completeness=1 or completeness=2) in each category (4 categories), then it qualifies and the value for that company that year becomes One; otherwise Zero

2) at least (one completeness=2) in each category (4 categories), then it qualifies and the value for that company that year becomes One; otherwise Zero

1. **Using the function to create the metadata of the score data**

def metadata(df, col, threshold, complete):

if complete == 1:

ind = 32

else: ind = 33

counts = []

for i in range(len(threshold)):

count = 0

for item in col:

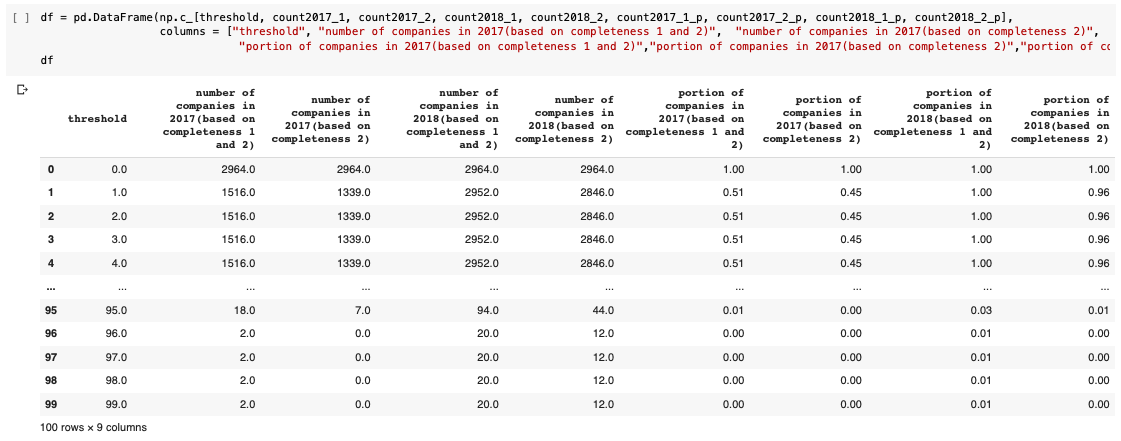
if df[item][ind] >= threshold[i]:

count += 1

counts.append(count)

return counts

And the result is as follow:

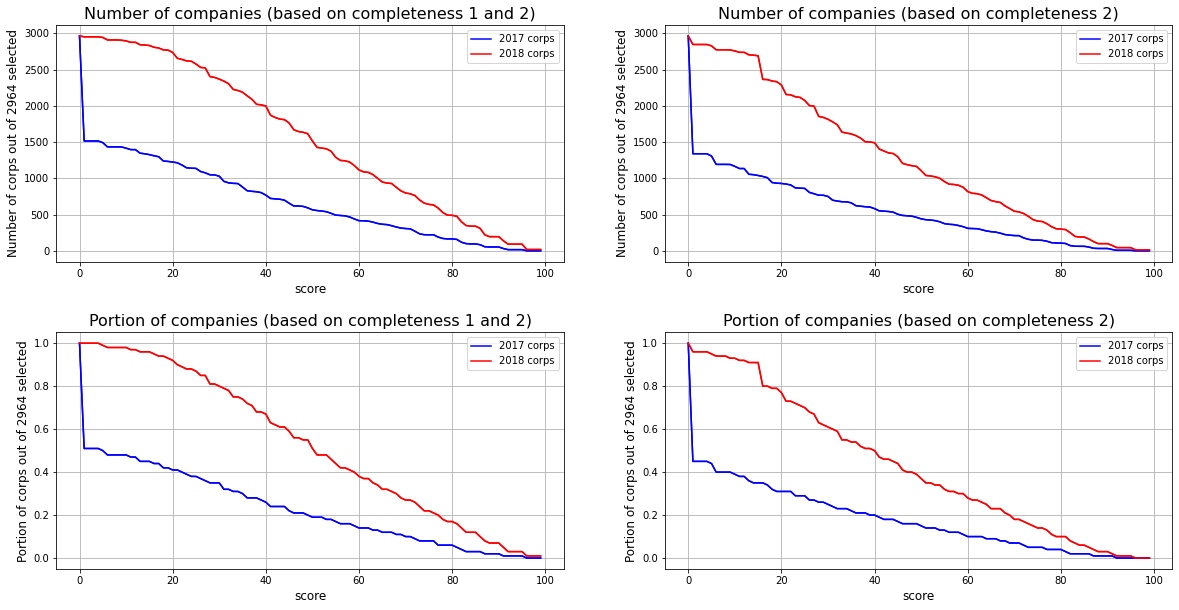


The selected row means that:

1. There are 1516 companies whose scores are higher than 4 based on the first scenario (considering completeness=1 and completeness=2) in 2017, accounting for 51%;
2. There are 1339 companies whose scores are higher than 4 based on the second scenario (considering completeness=2) in 2017, accounting for 45%;
3. There are 2952 companies whose scores are higher than 4 based on the first scenario (considering completeness=1 and completeness=2) in 2018, accounting for about 100%;
4. There are 2846 companies whose scores are higher than 4 based on the second scenario (considering completeness=2) in 2018, accounting for 96%;
5. **Visualization Template**

Finally, we use the template to create the dashboard based on the metadata. The two plots on the top show the number, and the two at the bottom show the portion.

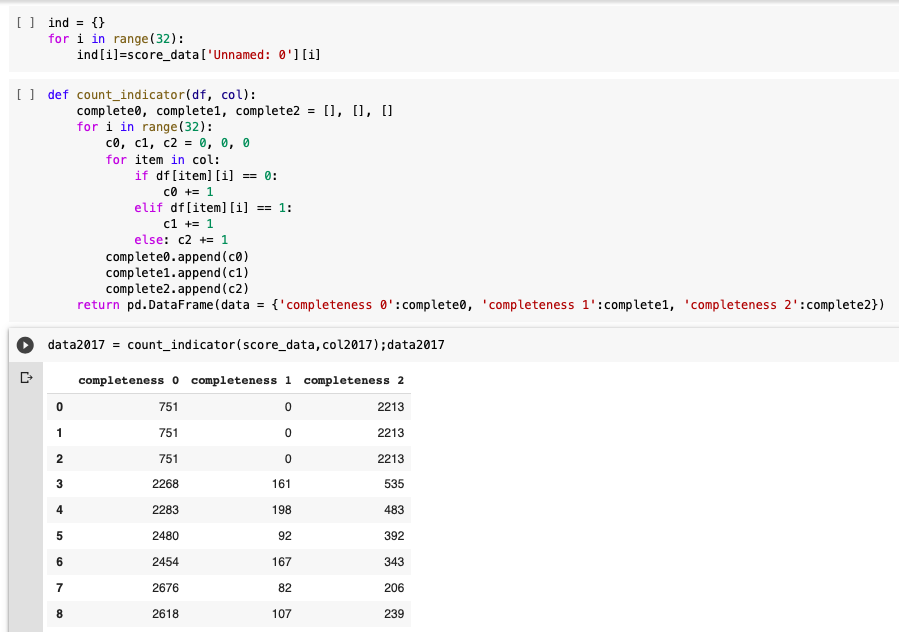
For example, we can see the improvement in the first plot that we have around 750 companies whose scores are above 40 in 2017 based on the first scenario, while we have around 2000 companies whose scores are above 40 in 2018.

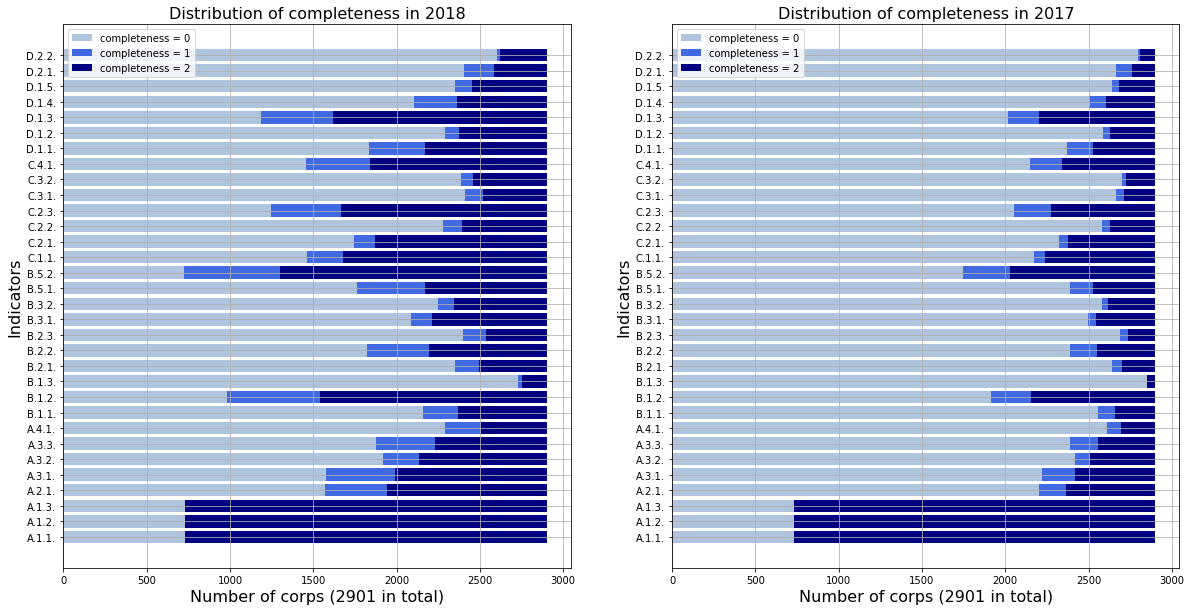


1. Visualization

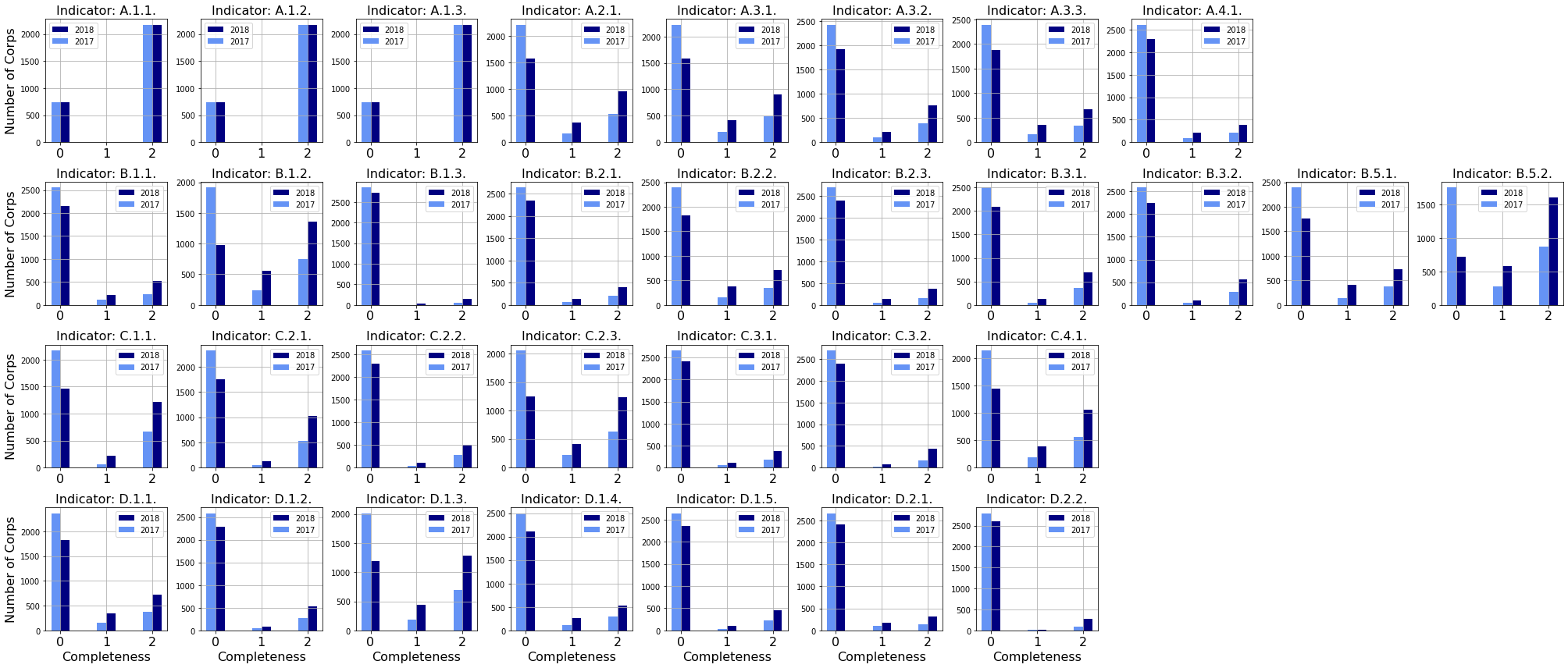
We assume that the data of A.1.1, A.1.2, and A.1.3 in 2017 are the same as the data in 2018. These three indicators have different data source (not from the extraction)

* 1. We use the following code to count the completeness for each indicator and plot their distribution

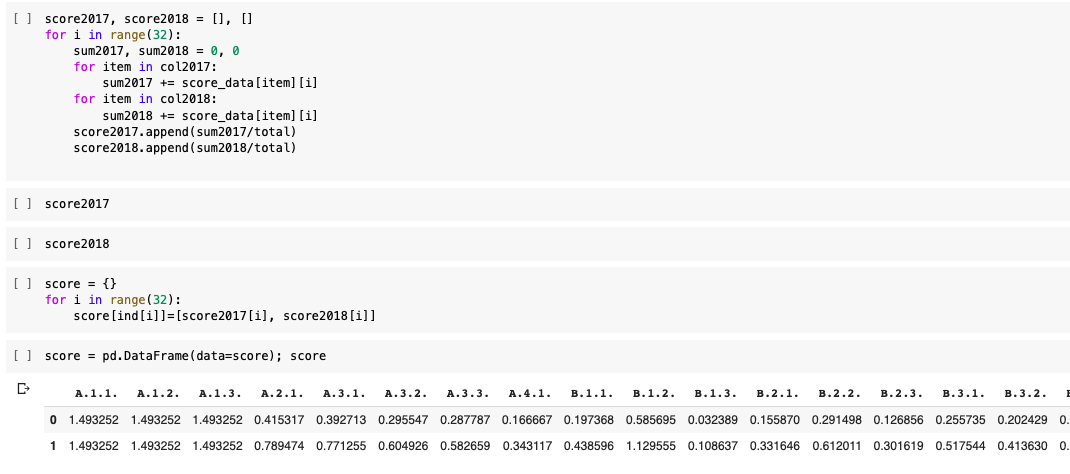


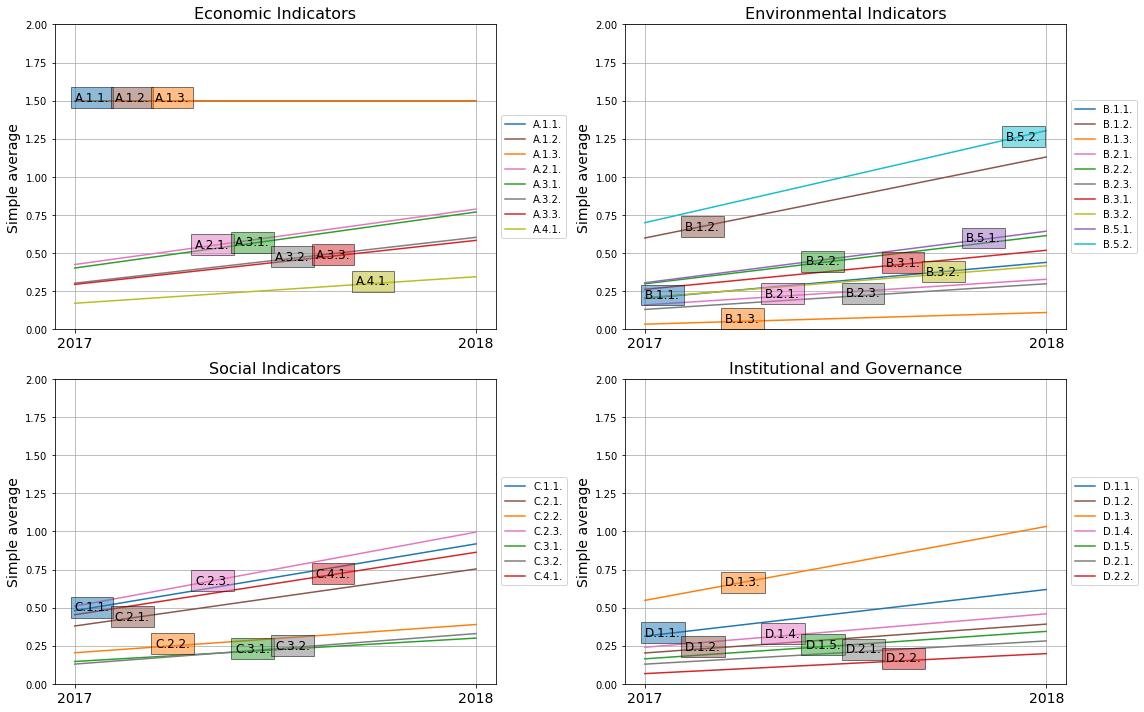


* 1. And plot the comparison between the two year for each indicator

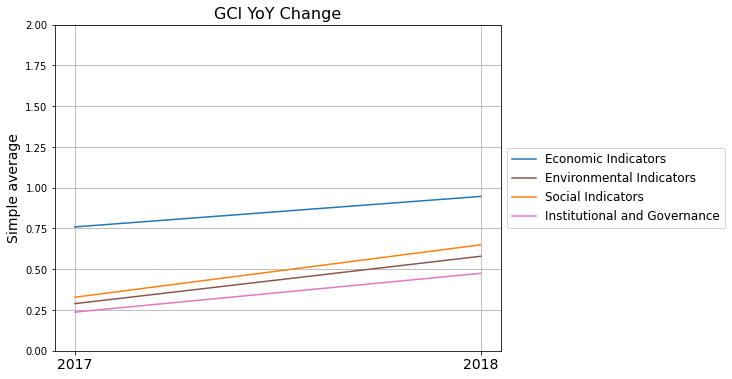


* 1. We use the following code to calculate the score(simple average with the maximum of 2) for each indicator, and visualized the trend for each indicator, and grouped by the categories of indicators





* 1. We calculated the simple average for each category and visualized the trend for each category



* 1. We calculated the simple average for each sub region and the trend
     + - 1. Average all indicators to calculate the score of each category
         2. Average four categories to calculate the score of each corp
         3. Average the score of all corps in a region to get the score of that region

