The happiness index raw data are 5 csv files from 2015 to 2019 using Kaggle API to extract.

The inconsistency among those csv files are:

1. Some of the data in one year does not exist in other years. E.g. in 2015 there is a “standard error”, but it is not in the rest years, same as “Lower confidence interval”, “upper confident interval” in 2016. “Dystopia Residual” in 2017 etc.
2. The name of the same variable in each year does not consistent among years, e.g. it is “Happiness Rank” in 2015 but it is “Happiness.Rank”; it is “Family” in 2015, 2016, 2017, but it named as “social support” in 2018, 2019 etc.
3. The order of the columns isn’t consist too, e.g in 2017 the order of the columns is “country, happiness.rank, happiness score”, but in 2018 the order is “overall rank, country or region, score”

The cleaning process I took is:

1. Inserting year into the fist column for each of the csv file and transform to a pandas DataFrame use a **self-define function** **formatDt**.**(Used df.insert(loc, column, value) and pd.read\_csv(path))**
2. Print out column names of all the years to compare any inconsistency.
3. List column names that exists in one year but does not exist in the other year use a **self-define function** **intersection**. **(Used list comprehension )** (which print out the column name that exists in the former year but does not exist in the later year and print out the column name that exists in the later year but does not exist in the former year.)
4. Drop columns that from step 3 if it is also irrelevant to happiness but just statistics use **df.drop(inplace = True).**
5. Repeat step 3 and 4 until the last year. When comes to that the discrepancy is due to a different order of column name, change the order of the columns by using **df.reindex(columns= new order)**, then repeat step 3 and 4 again.
6. After reaching to the last year, print column index of all the years to compare if there is any real discrepancy exist. Also compare the length of the column index for each year to check if any inconsistency exists.
7. Setting the column names for all the years in the same naming: used the 2015 column name and rename the columns of the rest years.
8. Concatenate 5 dataframe by using **pd.concat([df1, df2, df3, df4, df5])**
9. Drop nan values and set year as row index by **df.set\_index(‘Year’)**
10. Export dataFrame as a csv file for later use by **to\_csv(path).**