

Use case :

Loading the data :

```
: #Read in the csv file and convert to a Pandas dataframe
World_Happiness_2015 = pd.read_csv("datafrom_2015.csv ")
World_Happiness_2016 = pd.read_csv("datafrom_2016.csv ")
World_Happiness_2017 = pd.read_csv("datafrom_2017.csv ")
World_Happiness_2018 = pd.read_csv("datafrom_2018.csv ")
World_Happiness_2019 = pd.read_csv("datafrom_2019.csv ")
#World_Happiness = pd.concat([World_Happiness_2015,World_Happiness_2016,World_Happiness_2017,World_Happiness_2018,World_Happiness_2019])
```

merge all datagrams :

```
: # merge all dataframes
World_Happiness = pd.concat([World_Happiness_2015,World_Happiness_2016,World_Happiness_2017,World_Happiness_2018,World_Happiness_2019])
```

Viewing the dataframe :

Viewing the dataframe

We can get a quick sense of the size of our dataset by using the shape method. This returns a tuple with the number of rows and columns in the dataset.

```
[117]: World_Happiness
```

```
[117]:
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	...	Trust..Government.Corruption.	Dystopia.Residua
0	Switzerland	Western Europe	1.0	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	...	NaN	NaN
1	Iceland	Western Europe	2.0	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	...	NaN	NaN
2	Denmark	Western Europe	3.0	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	...	NaN	NaN
3	Norway	Western Europe	4.0	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	...	NaN	NaN
4	Canada	North America	5.0	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	...	NaN	NaN
...
151	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
152	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
153	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
154	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN

Data Profiling before do consistency processes:

1. Data Profiling:

Data profiling is a comprehensive process of examining the data available in an existing dataset and collecting statistics and information about that data.

```
[119]: World_Happiness.info
```

```
[119]: <bound method DataFrame.info of
0      Switzerland  Western Europe      1.0      7.587
1      Iceland     Western Europe      2.0      7.561
2      Denmark     Western Europe      3.0      7.527
3      Norway      Western Europe      4.0      7.522
4      Canada      North America      5.0      7.427
..      ...
151     NaN         NaN              NaN      NaN
152     NaN         NaN              NaN      NaN
153     NaN         NaN              NaN      NaN
154     NaN         NaN              NaN      NaN
155     NaN         NaN              NaN      NaN

Standard Error  Economy (GDP per Capita)  Family \
0      0.03411      1.39651      1.34951
1      0.04884      1.30232      1.40223
2      0.03328      1.32548      1.36058
3      0.03880      1.45900      1.33095
4      0.03553      1.32629      1.32261
..      ...
151     NaN         NaN              NaN
152     NaN         NaN              NaN
153     NaN         NaN              NaN
154     NaN         NaN              NaN
155     NaN         NaN              NaN
```

```
[119]: World_Happiness.info
```

```
[121]: World_Happiness.shape
```

```
[121]: (782, 30)
```

```
[123]: World_Happiness.describe()
```

```
[123]:
```

	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual	...	Health..Life.Expectancy.	Trust..Go
count	315.000000	315.000000	158.000000	315.000000	470.000000	315.000000	470.000000	315.000000	782.000000	315.000000	...	155.000000	
mean	79.238095	5.378949	0.047885	0.899837	0.990347	0.594054	0.402828	0.140532	0.218576	2.212032	...	0.551341	
std	45.538922	1.141531	0.017146	0.410780	0.318707	0.240790	0.150356	0.115490	0.122321	0.558728	...	0.237073	
min	1.000000	2.839000	0.018480	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.328580	...	0.000000	
25%	40.000000	4.510000	0.037268	0.594900	0.793000	0.419645	0.297615	0.061315	0.130000	1.884135	...	0.369866	
50%	79.000000	5.286000	0.043940	0.973060	1.025665	0.640450	0.418347	0.106130	0.201982	2.211260	...	0.606042	
75%	118.500000	6.269000	0.052300	1.229000	1.228745	0.787640	0.516850	0.178610	0.278832	2.563470	...	0.723008	
max	158.000000	7.587000	0.136930	1.824270	1.610574	1.025250	0.669730	0.551910	0.838075	3.837720	...	0.949492	

8 rows x 27 columns

```
[125]: World_Happiness.columns
```

```
[125]: Index(['Country', 'Region', 'Happiness Rank', 'Happiness Score',
        'Standard Error', 'Economy (GDP per Capita)', 'Family',
        'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)',
        'Generosity', 'Dystopia Residual', 'Lower Confidence Interval',
        'Upper Confidence Interval', 'Happiness.Rank', 'Happiness.Score',
        'Whisker.high', 'Whisker.low', 'Economy..GDP.per.Capita.',
        'Health..Life.Expectancy.', 'Trust..Government.Corruption.',
        'Dystopia.Residual', 'Overall rank', 'Country or region', 'Score',
        'GDP per capita', 'Social support', 'Healthy life expectancy',
        'Freedom to make life choices', 'Perceptions of corruption'],
        dtype='object')
```

Data Quality Checks :

Data Quality Checks

Data quality checks involve the process of ensuring that the data is accurate, complete, consistent, relevant, and reliable.

Here are typical steps involved in checking data quality:

1. Reliability:

Evaluate the data's source and collection process to determine its trustworthiness.

```
[133]: #In the kaggle page mentioned, the data source is Creative Commons Organization
```

2. Timeliness:

Ensure the data is up-to-date and reflective of the current situation or the period of interest for the analysis.

```
[136]: #Data from 2015 to 2019
```

Consistency :

When I merged the data frames, I found that the order of the columns did not match, and their names did not match. Now I will perform some operations to correct these problems and make the data consistent.

1.Reorder the columns :

Dataframs 2015 & 2016:

```
new_or1 = new_or1.sort_index(axis=1, ascending=False)

[267]: #reorder dataframe for 2015
new_or1 = ['Country', 'Happiness Rank', 'Happiness Score',
           'Economy (GDP per Capita)', 'Family',
           'Health (Life Expectancy)', 'Freedom', 'Trust (Government Corruption)',
           'Generosity', 'Dystopia Residual', 'Region', 'Standard Error']
World_Happiness_2015 = World_Happiness_2015[new_or1]
World_Happiness_2015.columns

[267]: Index(['Country', 'Happiness Rank', 'Happiness Score',
        'Economy (GDP per Capita)', 'Family', 'Health (Life Expectancy)',
        'Freedom', 'Trust (Government Corruption)', 'Generosity',
        'Dystopia Residual', 'Region', 'Standard Error'],
        dtype='object')

[269]: #reorder dataframe for 2016
new_or2 = ['Country', 'Happiness Rank', 'Happiness Score',
           'Economy (GDP per Capita)', 'Family', 'Health (Life Expectancy)',
           'Freedom', 'Trust (Government Corruption)', 'Generosity',
           'Dystopia Residual', 'Region', 'Lower Confidence Interval', 'Upper Confidence Interval']
World_Happiness_2016 = World_Happiness_2016[new_or2]
World_Happiness_2016.columns

[269]: Index(['Country', 'Happiness Rank', 'Happiness Score',
        'Economy (GDP per Capita)', 'Family', 'Health (Life Expectancy)',
        'Freedom', 'Trust (Government Corruption)', 'Generosity',
        'Dystopia Residual', 'Region', 'Lower Confidence Interval',
        'Upper Confidence Interval'],
        dtype='object')
```

Dataframes 2017&2018&2019 :

```
[271]: #reorder dataframe for 2017
new_or3 = ['Country', 'Happiness.Rank', 'Happiness.Score',

           'Economy..GDP.per.Capita.', 'Family', 'Health..Life.Expectancy.',
           'Freedom', 'Trust..Government.Corruption.', 'Generosity',
           'Dystopia.Residual','Whisker.high',
           'Whisker.low']
World_Happiness_2017 = World_Happiness_2017[new_or3]
World_Happiness_2017.columns
```

```
[271]: Index(['Country', 'Happiness.Rank', 'Happiness.Score',
           'Economy..GDP.per.Capita.', 'Family', 'Health..Life.Expectancy.',
           'Freedom', 'Trust..Government.Corruption.', 'Generosity',
           'Dystopia.Residual', 'Whisker.high', 'Whisker.low'],
          dtype='object')
```

```
[273]: #reorder dataframe for 2018 & 2019
new_or4 = ['Country or region', 'Overall rank', 'Score',

           'GDP per capita', 'Social support', 'Healthy life expectancy',
           'Freedom to make life choices', 'Perceptions of corruption', 'Generosity',
           ]
World_Happiness_2018 = World_Happiness_2018[new_or4]
World_Happiness_2019 = World_Happiness_2019[new_or4]
World_Happiness_2018.columns
World_Happiness_2019.columns
```

```
[273]: Index(['Country or region', 'Overall rank', 'Score', 'GDP per capita',
           'Social support', 'Healthy life expectancy',
           'Freedom to make life choices', 'Perceptions of corruption',
           'Generosity'],
          dtype='object')
```

Rename Columns :

```
[275]: #rename columns in dataframes
#2015
columns = {World_Happiness_2015.columns[i]: standard_columns_nameandorder[i] for i in range(len(standard_columns_nameandorder))}
World_Happiness_2015.rename(columns=columns, inplace=True)
#2016
columns = {World_Happiness_2016.columns[i]: standard_columns_nameandorder[i] for i in range(len(standard_columns_nameandorder))}
World_Happiness_2016.rename(columns=columns, inplace=True)
#2017
columns = {World_Happiness_2017.columns[i]: standard_columns_nameandorder[i] for i in range(len(standard_columns_nameandorder))}
World_Happiness_2017.rename(columns=columns, inplace=True)
#2018
columns = {World_Happiness_2018.columns[i]: standard_columns_nameandorder[i] for i in range(len(standard_columns_nameandorder))}
World_Happiness_2018.rename(columns=columns, inplace=True)
#2019
columns = {World_Happiness_2019.columns[i]: standard_columns_nameandorder[i] for i in range(len(standard_columns_nameandorder))}
World_Happiness_2019.rename(columns=columns, inplace=True)
```

Merge dataframes again and view dataframe:

```
[277]: # new merge all data frame agin
World_Happiness_v2 = pd.concat([World_Happiness_2015,World_Happiness_2016,World_Happiness_2017,World
```

```
[279]: World_Happiness_v2
```

```
[279]:
```

	Country	Happiness Rank	Happiness Score	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity
0	Switzerland	1	7.587	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678
1	Iceland	2	7.561	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630
2	Denmark	3	7.527	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139
3	Norway	4	7.522	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699
4	Canada	5	7.427	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811
...
151	Rwanda	152	3.334	0.35900	0.71100	0.61400	0.55500	0.41100	0.21700
152	Tanzania	153	3.231	0.47600	0.88500	0.49900	0.41700	0.14700	0.27600
153	Afghanistan	154	3.203	0.35000	0.51700	0.36100	0.00000	0.02500	0.15800
154	Central African Republic	155	3.083	0.02600	0.00000	0.10500	0.22500	0.03500	0.23500
155	South Sudan	156	2.853	0.30600	0.57500	0.29500	0.01000	0.09100	0.20200

Profiling new dataframe :

Info :

```
[223]: print("Data info for dataframe After doing consistency processes : ",World_Happiness_v2.info)

Data info for dataframe After doing consistency processes : <bound method DataFrame.info of
re \
0          Switzerland          1          7.587
1          Iceland            2          7.561
2          Denmark            3          7.527
3          Norway             4          7.522
4          Canada             5          7.427
..          ...              ...          ...
151         Rwanda            152         3.334
152         Tanzania          153         3.231
153         Afghanistan       154         3.203
154  Central African Republic  155         3.083
155         South Sudan        156         2.853

Economy (GDP per Capita)  Family  Health (Life Expectancy)  Freedom \
0          1.39651      1.34951          0.94143      0.66557
1          1.30232      1.40223          0.94784      0.62877
2          1.32548      1.36058          0.87464      0.64938
3          1.45900      1.33095          0.88521      0.66973
4          1.32629      1.32261          0.90563      0.63297
..          ...              ...          ...          ...
151         0.35900      0.71100          0.61400      0.55500
152         0.47600      0.88500          0.49900      0.41700
153         0.35000      0.51700          0.36100      0.00000
154         0.02600      0.00000          0.10500      0.22500
155         0.30600      0.57500          0.29500      0.01000

Trust (Government Corruption)  Generosity  Dystopia  Residual \
0          0.41978      0.29678          2.51738
1          0.14145      0.43630          2.70201
2          0.48357      0.34139          2.49204
3          0.36503      0.34699          2.46531
4          0.32957      0.45811          2.45176
..          ...              ...          ...
```

Shape :

```
] : print("The shape of dataframe After doing consistency processes : ",World_Happiness_v2.shape)

The shape of dataframe After doing consistency processes : (782, 17)
```

Description:

```
[229]: print("The Description of dataframe After doing consistency processes : ")
World_Happiness_v2.describe()
```

The Description of dataframe After doing consistency processes :

```
[229]:
```

	Happiness Rank	Happiness Score	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual	Standard Error	Lower Confidence Interval	Upper Confidence Interval	Dystopia.Re
count	782.000000	782.000000	782.000000	782.000000	782.000000	782.000000	781.000000	782.000000	315.000000	158.000000	157.000000	157.000000	155.0
mean	78.698210	5.379018	0.916047	1.078392	0.612416	0.411091	0.125436	0.218576	2.212032	0.047885	5.282395	5.481975	1.8
std	45.182384	1.127456	0.407340	0.329548	0.248309	0.152880	0.105816	0.122321	0.558728	0.017146	1.148043	1.136493	0.5
min	1.000000	2.693000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.328580	0.018480	2.732000	3.078000	0.3
25%	40.000000	4.509750	0.606500	0.869363	0.440183	0.309768	0.054000	0.130000	1.884135	0.037268	4.327000	4.465000	1.5
50%	79.000000	5.322000	0.982205	1.124735	0.647310	0.431000	0.091000	0.201982	2.211260	0.043940	5.237000	5.419000	1.8
75%	118.000000	6.189500	1.236187	1.327250	0.808000	0.531000	0.156030	0.278832	2.563470	0.052300	6.154000	6.434000	2.1
max	158.000000	7.769000	2.096000	1.644000	1.141000	0.724000	0.551910	0.838075	3.837720	0.136930	7.460000	7.669000	3.1

Activate Win

Columns :

```
[231]: print("The Columns of dataframe After doing consistency processes : ")
World_Happiness_v2.columns
```

The Columns of dataframe After doing consistency processes :

```
[231]: Index(['Country', 'Happiness Rank', 'Happiness Score',
            'Economy (GDP per Capita)', 'Family', 'Health (Life Expectancy)',
            'Freedom', 'Trust (Government Corruption)', 'Generosity',
            'Dystopia Residual', 'Region', 'Standard Error',
            'Lower Confidence Interval', 'Upper Confidence Interval',
            'Dystopia.Residual', 'Whisker.high', 'Whisker.low'],
            dtype='object')
```