# Experiment No. 3

# Title: Exploratory data analysis using PANDAS

**Batch: B-1** **Roll No: 16010422234** **Name: Chandana Ramesh Galgali**

**Experiment No.: 4**

**Aim:** To perform exploratory data analysis using python Pandas

---

**Resources needed:** Python IDE

---

**Theory:**

Pandas is a Python library that provides extensive means for data analysis. Data scientists often work with data stored in table formats like .csv, .tsv, or .xlsx. Pandas makes it very convenient to load, process, and analyze such tabular data using SQL-like queries. Python has long been great for data munging and preparation, but less so for data analysis and modeling. *pandas* helps fill this gap, enabling you to carry out your entire data analysis workflow in Python. In conjunction with Matplotlib and Seaborn, Pandas provides a wide range of opportunities for visual analysis of tabular data.



Installing pandas library:

Conda install pandas

Or

pip install pandas

The main data structures in Pandas are implemented with Series and DataFrame classes. The former is a one-dimensional indexed array of some fixed data type. The latter is a two-dimensional data structure - a table - where each column contains data of the same type. You can see it as a dictionary of Series instances. DataFrames are great for representing real data: rows correspond to instances (examples, observations, etc.), and columns correspond to features of these instances.

A series can be created using list ,dictionary etc. with index(implicit indexing) or without index(explicit indexing).

```
import pandas as pd
data1=pd.Series({2:'a', 1:'b', 3:'c'}) #implicit indexing
data2=pd.Series({2:'a', 1:'b', 3:'c'}, index=[1,2,3]) # explicit indexing
        #loc attribute allows indexing and slicing that always references the explicit index:
 data2.loc[2]
```

```
#iloc attribute allows indexing and slicing that always references the implicit
#Python-style index
data1.iloc[1]
```

Following are the various series related operations

- Append(): s3.append(s1) # Stitch s1 to s3
- Drop: s4.drop('e') #Delete the value whose index is e
- Addition: s4.add(s3)#addition according to the index, and it would be filled with NaN (null value) if the indexes are different.
- Subtraction: s4.sub(s3) #substraction according to the index, and it would be filled with NaN (null value) if the in\dexes are different.
- Multiplication: s4.mul(s3) #multiplication according to the index, and it would be filled with NaN (null value) if the indexes are different.
- Division: s4.div(s3)
- Median: s4.median()
- Sum: s4.sum()
- Maximum & Minimum : s4.max() s4.min()

A data frame object keeps track of both data (numerical as well as text), and column and row headers. It can have multiple columns of data.
convert a numpy array to a pandas data frame with pd.Data frame().

```
import numpy as np
h = [[1,2],[3,4]]
df_h = pd.DataFrame(h)
print('Data Frame:', df_h)
```

data frame can read data from dictionaries and files as well.

Reading and writing data from files:
CSVs don't have indexes like our DataFrames, so all we need to do is just designate the index_col when reading.

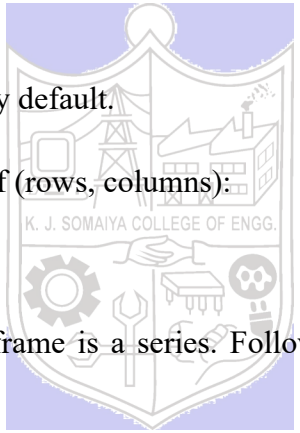
```
import pandas as pd
df=pd.read_csv("C:/Users/Admin/Desktop/ADVANCED
PYTHON/DATA/SalesJan2009.csv",index_col=0)
#Reading the dataset in a dataframe using Pandas
print(df)
```

To write data to a new csv file use to_csv()

```
df3.to_csv('animal.csv')
df3.to_excel('animal.xlsx', sheet_name='Sheet1')
```

**Following functions of dataframe can be used to explore the dataset to get a summary of it.**

- **info()** provides the essential details about your dataset, such as the number of rows and columns, the number of non-null values, what type of data is in each column, and how much memory your DataFrame is using.
  df.info()
- **describe()** is used to get a summary of numeric values in your dataset. It calculates the mean, standard deviation, minimum value, maximum value, 1st percentile, 2nd percentile, 3rd percentile of the columns with numeric values. It also counts the number of variables in the dataset.
  df.describe()
  describe() can also be used on a categorical variable to get the count of rows, unique count of categories, top category, and freq of top category
  temp_df['product'].describe()
- **head()** outputs the first five rows of your DataFrame by default, but we could also pass a number as well
  print(df.head)
- **tail()** outputs last five rows by default.
  print(df.tail)
- **shape()** outputs just a tuple of (rows, columns):
  df.shape




**Row and column selection:**

Each row and column of the dataframe is a series. Following functions can be used for row selection and column selection

Extracting a column using square brackets will return a *Series.*

```
prod_col=temp_df['product']
#selecting multiple columns
subset=temp_df[['product','price']]
```

accessing rows:

.loc - locates by name

```
prom = movies_df.loc["Prometheus"]
prom
```

.iloc- locates by numerical index

```
prod = df.iloc[1]
prod
```

**Further analysis using pandas dataframe:**

**value_counts()** can tell us the frequency of all values in a column.

```
temp_df['product'].value_counts().head(10)
```

**nunique()** to count number of unique values that occur in dataset or in a column

```
df.nunique() #to see the counts of unique numbers in each column
df["Embarked"].nunique() #to get the unique count of a column
```

**corr()** generate the relationship between each continuous variable:

```
temp_df.corr()
```

Correlation tables are a numerical representation of the bivariate relationships in the dataset.

**astype()** can be used to change the datatype of that column

```
df["Embarked"] = df["Embarked"].astype("category")
df["Embarked"].dtype
```

**column clean up funtions:**

**append()** will return a copy after appending without affecting the original DataFrame(if inplace attribute is used).

```
temp_df = df.append(df)
temp_df.shape
```

**drop_duplicates()** method will return a copy of DataFramewith duplicates removed.

```
temp_df = temp_df.drop_duplicates()
#inorder to avoid reasignment it can be done
temp_df.drop_duplicates(inplace=True)
temp_df.shape
```

**.columns** print the column names of dataset also can be used for renaming

```
temp_df.columns
```

**drop()** can be used to delete columns

```
df.drop(columns=['A', 'C'])
```

**rename()** method is used to rename certain or all columns via a dict.

```
temp_df.rename(columns={
'Account_Created': 'Acc_Created',
'Last_Login': 'Lst_Login'
  }, inplace=True)
temp_df.columns
```

**Handling null values using pandas:**

Mostly Python's None or NumPy's np.nan indicates missing or null values.

- **isnull()** checks which cells in our DataFrame are null. It returns a DataFrame where each cell is either True or False depending on that cell's null status.

```
temp_df.isnull()
```

To count the number of nulls in each column we use an aggregate function .sum() for summing:

```
temp_df.isnull().sum()
```

To get rid of rows or columns with nulls. Removing null data is only suggested if you have a small amount of missing data. .dropna() will delete any row with at least a single null value, but it will return a new DataFrame without altering the original one.

```
temp_df.dropna()
```

- Or drop columns with null values by setting axis=1.

```
temp_df.dropna(axis=1)
```

- Replace nulls with non-null values, a technique known as imputation. Normally null value is replaced with mean or the median of that column.

**Conditional selections/ Filtering**

Comparison operators are used for filtering

Take a column from the DataFrame and apply a Boolean condition to it.

```
condition = (movies_df['Director'] == "Ridley Scott")
```

It returns a Series of True and False values.Some more examples on conditionals

Select movies_df where movies_df Director equals Ridley Scott.

```
movies_df[movies_df['Director'] == "Ridley Scott"]
movies_df[movies_df['Rating'] >= 8.6].head(3)
movies_df[(movies_df['Director'] == 'Christopher Nolan') | (movies_df['Director'] == 'Ridley Scott')].head()
movies_df[movies_df['Director'].isin(['Christopher Nolan', 'Ridley Scott'])].head()
```

**Summary statistics Functions/ Aggregate Functions**

Aggregating functions are the ones that reduce the dimension of the returned objects. DataFrames include some aggregate functions to understand the overall properties of a dataset

df.count(): Count the number of rows /items

df.mean(): To find mean average of data frame

Syntax: data.Population.mean()#where Population is column name

df.median(): To find median of data frame

df.quantile():

df.sum(): Do a summation operation on any column in the DataFrame

df.prod(): To find Product of all items

df.std():To find standard deviation of a data frame

df.var():To find variance of data frame

df.min(), df.max(): To find Minimum and maximum

df.first(), df.last(): First and last item

**GROUP BY and aggregation**

"group by" process can involve one or more of the following steps:

–Splitting the data into groups based on some criteria.

–Applying a function to each group independently.

–Combining the results into a data structure

Apply step involves following

–Aggregation: compute a summary statistic (or statistics) for each group. Some examples:

- •Compute group sums or means.
- •Compute group sizes / counts.

–Transformation: perform some group-specific computations and return a like-indexed object. Some examples:

- •Standardize data (zscore) within a group.
- •Filling NAs within groups with a value derived from each group.

–Filtration: discard some groups, according to a group-wise computation that evaluates True or False. Some examples:

- •Discard data that belongs to groups with only a few members.
- •Filter out data based on the group sum or mean.

```
# group the data on team value.
    gk = df.groupby('Team')
# Finding the values contained in the "Boston Celtics" group
    gk.get_group('Boston Celtics')
```

**Applying functions**

To iterate over a DataFrame or Series we can use list, but doing so — especially on large datasets — is very slow.

An efficient alternative is to apply() a function to the dataset. Using apply() will be much faster than iterating manually over rows because pandas is utilizing vectorization.

Combining datasets:

**Combining Datasets**

Concat: s to append either columns or rows from one DataFrame to another.

Joining two dataframe on the index

merge two dataframes on key attribute

---

**Activities:**

1. Download data set with at least 1500 rows and 10-20 columns (numeric and non numeric) from valid data sources
2. Read same in pandas DataFrame
3. Perform in detail Exploratory data analysis of this dataset
   - Get information and description of dataset.
   - See if any null values are present. Display count of null values.
   - Choose appropriate technique to handle missing values.(imputation with use of inplace)
   - Use sorting of data in dataframe to display topmost 5 or 8 records based on one or more column values(conditional filtering)

- Get frequency listing of any one relevant column(2 cases)
- Sorting of rows and columns,( implicit and explicit indexing)
- Accessing particular row based on certain condition and displaying only one or few columns from it.(3 cases with compound conditions)
- Minimum and maximum values related analysis
- Use of group by on one or more columns(2 cases)
- Add a new column to the existing dataframe and populate the same using existing columns data.
- Use of appropriate aggregate functions with groupby(2 cases)
- Selection on particular groups based on name or condition
- Find correlation between any two column values.
- Try transformation(normalization using any technique) on data set
- Joining , merging and concatenation of data in dataframe.

Write down observations for your dataset for each of the above listed tasks of analysis.

---

**Result: (script and output)**

```
import pandas as pd
df = pd.read_excel(r'C:\Users\daxay\Downloads\GlobalDataOnSustainableEnergy.xls')
print("Dataframe & its Info:\n", df)
print(df.info())
print("\nDataframe Description:\n", df.describe())
print("\nDataframe Head:\n", df.head())
print("\nDataframe Tail:\n", df.tail())
print("\nDataframe Shape:\n", df.shape)
print("\nDataframe Missing Values:\n", df.isnull())
print("\nTotal Number of Missing Values:\n", df.isnull().sum())
print("\nHandling Missing Values (putting 0 in place of null values):\n", df.fillna(0))
print("\nDeleting Rows with Missing Values:\n", df.dropna())
print("\nDeleting Columns with Missing Values:\n", df.dropna(axis=1))
print("\nSorting the DataFrame based on one or more columns:\n", df.sort_values(by=['Year', 'Renewable energy share in the total final energy consumption (%)']))
print("\nFrequency Listing of a Relevant Column:\n",
```

```
df['Renewable-electricity-generating-capacity-per-capita'].value_counts())
df_filtered = df[df['Financial flows to developing countries (US $)'] > 786330000]
print(df_filtered)
print("\nMinimum Value:\n", df.min())
print("\nMaximum Value:\n", df.max())
print("\nGroup by one or more columns and perform aggregation:\n", df.groupby('Entity').agg({'Electricity from fossil fuels (TWh)': sum, 'Electricity from nuclear (TWh)': sum, 'Electricity from renewables (TWh)': sum}))
df['Electricity generated'] = df['Electricity from fossil fuels (TWh)'] + df['Electricity from nuclear (TWh)'] + df['Electricity from renewables (TWh)']
print("\nDataFrame with a New Column:\n", df)
print("\nCorrelation between column1 and column2:\n", df['Renewable-electricity-generating-capacity-per-capita'].corr(df['gdp_per_capita']))
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df_normalized = pd.DataFrame(scaler.fit_transform(df['Energy intensity level of primary energy (MJ/$2017 PPP GDP)'].values.reshape(-1, 1)), columns=['Energy intensity level of primary energy (MJ/$2017 PPP GDP)'])
print("\nNormalized DataFrame:\n", df_normalized)
df1 = pd.DataFrame(df['Entity'])
df2 = pd.DataFrame(df['Year'])
df3 = pd.DataFrame(df['Value_co2_emissions_kt_by_country'])
df_concatenated = pd.concat([df1, df2, df3], axis=1)
print("\nConcatenated DataFrame:\n", df_concatenated)
```

```
Dataframe & its Info:
          Entity  Year  Access to electricity (% of population)  Access to clean fuels for cooking  ...  Density\n(P/Km2)  Land Area(Km2)   Latitude  Longitude
0     Afghanistan  2000                                1.613591                                6.2  ...              60.0        652230.0  33.939110  67.709953
1     Afghanistan  2001                                4.074574                                7.2  ...              60.0        652230.0  33.939110  67.709953
2     Afghanistan  2002                                9.409158                                8.2  ...              60.0        652230.0  33.939110  67.709953
3     Afghanistan  2003                               14.738506                                9.5  ...              60.0        652230.0  33.939110  67.709953
4     Afghanistan  2004                               20.064968                               10.9  ...              60.0        652230.0  33.939110  67.709953
...           ...   ...                                     ...                                ...  ...               ...             ...        ...        ...
3644     Zimbabwe  2016                               42.561730                               29.8  ...              38.0        390757.0 -19.015438  29.154857
3645     Zimbabwe  2017                               44.178635                               29.8  ...              38.0        390757.0 -19.015438  29.154857
3646     Zimbabwe  2018                               45.572647                               29.9  ...              38.0        390757.0 -19.015438  29.154857
3647     Zimbabwe  2019                               46.781475                               30.1  ...              38.0        390757.0 -19.015438  29.154857
3648     Zimbabwe  2020                               52.747670                               30.4  ...              38.0        390757.0 -19.015438  29.154857

[3649 rows x 21 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3649 entries, 0 to 3648
Data columns (total 21 columns):
 #   Column                                                 Non-Null Count  Dtype
---  ------                                                 --------------  -----
 0   Entity                                                 3649 non-null   object
 1   Year                                                   3649 non-null   int64
 2   Access to electricity (% of population)                3639 non-null   float64
 3   Access to clean fuels for cooking                      3480 non-null   float64
 4   Renewable-electricity-generating-capacity-per-capita   2718 non-null   float64
 5   Financial flows to developing countries (US $)         1560 non-null   float64
 6   Renewable energy share in the total final energy consumption (%)  3455 non-null   float64
 7   Electricity from fossil fuels (TWh)                    3628 non-null   float64
 8   Electricity from nuclear (TWh)                         3523 non-null   float64
 9   Electricity from renewables (TWh)                      3628 non-null   float64
 10  Low-carbon electricity (% electricity)                 3607 non-null   float64
 11  Primary energy consumption per capita (kWh/person)     3649 non-null   float64
 12  Energy intensity level of primary energy (MJ/$2017 PPP GDP)  3442 non-null   float64
 13  Value_co2_emissions_kt_by_country                      3221 non-null   float64
 14  Renewables (% equivalent primary energy)               1512 non-null   float64
 15  gdp_growth                                             3332 non-null   float64
 16  gdp_per_capita                                         3367 non-null   float64
 17  Density\n(P/Km2)                                       3648 non-null   float64
 18  Land Area(Km2)                                         3648 non-null   float64
 19  Latitude                                               3648 non-null   float64
 20  Longitude                                              3648 non-null   float64
dtypes: float64(19), int64(1), object(1)
memory usage: 598.8+ KB
None

Dataframe Description:
              Year  Access to electricity (% of population)  Access to clean fuels for cooking  ...  Land Area(Km2)     Latitude    Longitude
count  3649.000000                              3639.000000                        3480.000000  ...    3.648000e+03  3648.000000  3648.000000
mean   2010.038367                                78.933702                          63.255287  ...    6.332135e+05    18.246388    14.822695
std       6.054228                                30.275541                          39.043658  ...    1.585519e+06    24.159232    66.348148
min    2000.000000                                 1.252269                           0.000000  ...    2.100000e+01   -40.900557  -175.198242
25%    2005.000000                                59.800890                          23.175000  ...    2.571300e+04     3.202778   -11.779889
50%    2010.000000                                98.361570                          83.150000  ...    1.176000e+05    17.189877    19.145136
75%    2015.000000                               100.000000                         100.000000  ...    5.131200e+05    38.969719    46.199616
max    2020.000000                               100.000000                         100.000000  ...    9.984670e+06    64.963051   178.065032

[8 rows x 20 columns]

Dataframe Head:
       Entity  Year  Access to electricity (% of population)  Access to clean fuels for cooking  ...  Density\n(P/Km2)  Land Area(Km2)  Latitude  Longitude
0  Afghanistan  2000                                1.613591                                6.2  ...              60.0        652230.0  33.93911  67.709953
1  Afghanistan  2001                                4.074574                                7.2  ...              60.0        652230.0  33.93911  67.709953
2  Afghanistan  2002                                9.409158                                8.2  ...              60.0        652230.0  33.93911  67.709953
3  Afghanistan  2003                               14.738506                                9.5  ...              60.0        652230.0  33.93911  67.709953
4  Afghanistan  2004                               20.064968                               10.9  ...              60.0        652230.0  33.93911  67.709953

[5 rows x 21 columns]

Dataframe Tail:
        Entity  Year  Access to electricity (% of population)  Access to clean fuels for cooking  ...  Density\n(P/Km2)  Land Area(Km2)   Latitude  Longitude
3644  Zimbabwe  2016                               42.561730                               29.8  ...              38.0        390757.0 -19.015438  29.154857
3645  Zimbabwe  2017                               44.178635                               29.8  ...              38.0        390757.0 -19.015438  29.154857
3646  Zimbabwe  2018                               45.572647                               29.9  ...              38.0        390757.0 -19.015438  29.154857
3647  Zimbabwe  2019                               46.781475                               30.1  ...              38.0        390757.0 -19.015438  29.154857
3648  Zimbabwe  2020                               52.747670                               30.4  ...              38.0        390757.0 -19.015438  29.154857

[5 rows x 21 columns]

Dataframe Shape:
 (3649, 21)

Dataframe Missing Values:
      Entity   Year  Access to electricity (% of population)  Access to clean fuels for cooking  ...  Density\n(P/Km2)  Land Area(Km2)  Latitude  Longitude
0      False  False                                   False                              False  ...             False           False     False      False
1      False  False                                   False                              False  ...             False           False     False      False
2      False  False                                   False                              False  ...             False           False     False      False
3      False  False                                   False                              False  ...             False           False     False      False
4      False  False                                   False                              False  ...             False           False     False      False
...      ...    ...                                     ...                                ...  ...               ...             ...       ...        ...
3644   False  False                                   False                              False  ...             False           False     False      False
3645   False  False                                   False                              False  ...             False           False     False      False
3646   False  False                                   False                              False  ...             False           False     False      False
3647   False  False                                   False                              False  ...             False           False     False      False
3648   False  False                                   False                              False  ...             False           False     False      False

[3649 rows x 21 columns]

Total Number of Missing Values:
 Entity                                                    0
Year                                                       0
Access to electricity (% of population)                   10
Access to clean fuels for cooking                        169
Renewable-electricity-generating-capacity-per-capita     931
Financial flows to developing countries (US $)          2089
Renewable energy share in the total final energy consumption (%)     194
Electricity from fossil fuels (TWh)                       21
Electricity from nuclear (TWh)                           126
Electricity from renewables (TWh)                         21
```

```
Low-carbon electricity (% electricity)                          42
Primary energy consumption per capita (kWh/person)               0
Energy intensity level of primary energy (MJ/$2017 PPP GDP)    207
Value_co2_emissions_kt_by_country                               428
Renewables (% equivalent primary energy)                       2137
gdp_growth                                                      317
gdp_per_capita                                                  282
Density\n(P/Km2)                                                  1
Land Area(Km2)                                                    1
Latitude                                                          1
Longitude                                                         1
dtype: int64

Handling Missing Values (putting 0 in place of null values):
          Entity  Year  Access to electricity (% of population)  Access to clean fuels for cooking  ...  Density\n(P/Km2)  Land Area(Km2)   Latitude  Longitude
0    Afghanistan  2000                                1.613591                               6.2  ...              60.0        652230.0  33.939110  67.709953
1    Afghanistan  2001                                4.074574                               7.2  ...              60.0        652230.0  33.939110  67.709953
2    Afghanistan  2002                                9.409158                               8.2  ...              60.0        652230.0  33.939110  67.709953
3    Afghanistan  2003                               14.738506                               9.5  ...              60.0        652230.0  33.939110  67.709953
4    Afghanistan  2004                               20.064968                              10.9  ...              60.0        652230.0  33.939110  67.709953
...          ...   ...                                     ...                               ...  ...               ...             ...        ...        ...
3644     Zimbabwe  2016                               42.561730                              29.8  ...              38.0        390757.0 -19.015438  29.154857
3645     Zimbabwe  2017                               44.178635                              29.8  ...              38.0        390757.0 -19.015438  29.154857
3646     Zimbabwe  2018                               45.572647                              29.9  ...              38.0        390757.0 -19.015438  29.154857
3647     Zimbabwe  2019                               46.781475                              30.1  ...              38.0        390757.0 -19.015438  29.154857
3648     Zimbabwe  2020                               52.747670                              30.4  ...              38.0        390757.0 -19.015438  29.154857

[3649 rows x 21 columns]

Deleting Rows with Missing Values:
          Entity  Year  Access to electricity (% of population)  Access to clean fuels for cooking  ...  Density\n(P/Km2)  Land Area(Km2)   Latitude  Longitude
43       Algeria  2001                               98.96687                              97.30  ...              18.0       2381741.0  28.033886   1.659626
44       Algeria  2002                               98.95306                              97.80  ...              18.0       2381741.0  28.033886   1.659626
45       Algeria  2003                               98.93401                              98.00  ...              18.0       2381741.0  28.033886   1.659626
46       Algeria  2004                               98.91208                              98.20  ...              18.0       2381741.0  28.033886   1.659626
47       Algeria  2005                               98.88961                              98.50  ...              18.0       2381741.0  28.033886   1.659626
```

```
...          ...   ...                                     ...                               ...  ...               ...             ...        ...        ...
3559  Uzbekistan  2015                              100.00000                              85.35  ...              79.0        447400.0  41.377491  64.585262
3560  Uzbekistan  2016                              100.00000                              85.20  ...              79.0        447400.0  41.377491  64.585262
3561  Uzbekistan  2017                              100.00000                              84.90  ...              79.0        447400.0  41.377491  64.585262
3562  Uzbekistan  2018                              100.00000                              84.30  ...              79.0        447400.0  41.377491  64.585262
3563  Uzbekistan  2019                              100.00000                              84.60  ...              79.0        447400.0  41.377491  64.585262

[343 rows x 21 columns]

Deleting Columns with Missing Values:
          Entity  Year  Primary energy consumption per capita (kWh/person)
0    Afghanistan  2000                                          302.59482
1    Afghanistan  2001                                          236.89185
2    Afghanistan  2002                                          210.86215
3    Afghanistan  2003                                          229.96822
4    Afghanistan  2004                                          204.23125
...          ...   ...                                               ...
3644     Zimbabwe  2016                                         3227.68020
3645     Zimbabwe  2017                                         3068.01150
3646     Zimbabwe  2018                                         3441.98580
3647     Zimbabwe  2019                                         3003.65530
3648     Zimbabwe  2020                                         2680.13180

[3649 rows x 3 columns]

Sorting the DataFrame based on one or more columns:
                 Entity  Year  Access to electricity (% of population)  ...  Land Area(Km2)   Latitude   Longitude
84    Antigua and Barbuda  2000                               97.689260  ...           443.0  17.060816  -61.796428
231               Bahamas  2000                              100.000000  ...         13878.0  25.025885  -78.035889
252               Bahrain  2000                              100.000000  ...           765.0  26.066700   50.557700
630        Cayman Islands  2000                              100.000000  ...           264.0  19.329900   81.252400
1786               Kuwait  2000                              100.000000  ...         17818.0  29.311660   47.481766
...                   ...   ...                                     ...  ...             ...        ...         ...
3543              Uruguay  2020                              100.000000  ...        176215.0 -32.522779  -55.765835
3564           Uzbekistan  2020                              100.000000  ...        447400.0  41.377491   64.585262
3585              Vanuatu  2020                               67.333270  ...         12189.0 -15.376706  166.959158
```

```
3606                Yemen  2020                               73.757930  ...        527968.0  15.552727   48.516388
3627               Zambia  2020                               44.524475  ...        752618.0 -13.133897   27.849332

[3649 rows x 21 columns]

Frequency Listing of a Relevant Column:
 Renewable-electricity-generating-capacity-per-capita
0.00      229
0.06       11
0.01       11
0.26       10
0.05       10
         ...
731.32      1
714.92      1
711.64      1
675.23      1
80.61       1
Name: count, Length: 2110, dtype: int64
          Entity  Year  Access to electricity (% of population)  Access to clean fuels for cooking  ...  Density\n(P/Km2)  Land Area(Km2)   Latitude   Longitude
119    Argentina  2014                              100.000000                              99.7  ...              17.0       2780400.0 -38.416097  -63.616672
606     Cameroon  2018                               62.200000                              22.1  ...              56.0        475440.0   7.369722   12.354722
705        Chile  2012                              100.000000                             100.0  ...              26.0        756096.0 -35.675147  -71.542969
706        Chile  2013                               99.600000                             100.0  ...              26.0        756096.0 -35.675147  -71.542969
707        Chile  2014                              100.000000                             100.0  ...              26.0        756096.0 -35.675147  -71.542969
730        China  2016                              100.000000                              70.6  ...             153.0       9596960.0  35.861660  104.195397
751     Colombia  2016                               98.400000                              90.7  ...              46.0       1138910.0   4.570868  -74.297333
997      Ecuador  2010                               97.462140                              93.6  ...              71.0        283561.0  -1.831239  -78.183406
1018       Egypt  2010                               99.394554                              99.1  ...             103.0       1001450.0  26.820553   30.802498
1025       Egypt  2017                              100.000000                              99.9  ...             103.0       1001450.0  26.820553   30.802498
1143    Ethiopia  2009                               23.690039                               1.9  ...             115.0       1104300.0   9.145000   40.489673
1310       Ghana  2007                               56.929794                              11.8  ...             137.0        238533.0   7.946527   -1.023194
1405      Guinea  2018                               45.000000                               1.4  ...              53.0        245857.0   9.945587   -9.696645
1406      Guinea  2019                               42.200000                               1.6  ...              53.0        245857.0   9.945587   -9.696645
1545       India  2011                               67.600000                              37.2  ...             464.0       3287263.0  20.593684   78.962880
1546       India  2012                               79.900000                              39.6  ...             464.0       3287263.0  20.593684   78.962880
```

```
1549      India  2015                            88.000000                            48.2  ...            464.0       3287263.0  20.593684   78.962880
1550      India  2016                            89.217800                            51.4  ...            464.0       3287263.0  20.593684   78.962880
1551      India  2017                            92.124950                            55.8  ...            464.0       3287263.0  20.593684   78.962880
1552      India  2018                            95.700000                            59.8  ...            464.0       3287263.0  20.593684   78.962880
1566  Indonesia  2011                            94.830000                            47.3  ...            151.0       1904569.0  -0.789275  113.921327
1569  Indonesia  2014                            97.010000                            62.8  ...            151.0       1904569.0  -0.789275  113.921327
1573  Indonesia  2018                            98.510000                            78.7  ...            151.0       1904569.0  -0.789275  113.921327
2211    Morocco  2012                            94.210520                            97.1  ...             83.0        446550.0  31.791702   -7.092620
2213    Morocco  2014                            91.600000                            97.4  ...             83.0        446550.0  31.791702   -7.092620
2217    Morocco  2018                            98.100000                            97.9  ...             83.0        446550.0  31.791702   -7.092620
2443    Nigeria  2013                            55.600000                             3.4  ...            226.0        923768.0   9.081999    8.675277
2447    Nigeria  2017                            54.400000                             8.7  ...            226.0        923768.0   9.081999    8.675277
2449    Nigeria  2019                            55.400000                            12.9  ...            226.0        923768.0   9.081999    8.675277
2526   Pakistan  2012                            70.958610                            38.2  ...            287.0        796095.0  30.375321   69.345116
2529   Pakistan  2015                            71.381546                            42.1  ...            287.0        796095.0  30.375321   69.345116
2530   Pakistan  2016                            71.550224                            43.3  ...            287.0        796095.0  30.375321   69.345116
2531   Pakistan  2017                            70.790000                            44.8  ...            287.0        796095.0  30.375321   69.345116
3148      Sudan  2003                            29.790567                            13.4  ...             25.0       1861484.0  12.862807   30.217636
3366     Turkey  2011                           100.000000                            94.3  ...            110.0        783562.0  38.963745   35.243322
3371     Turkey  2016                           100.000000                            95.1  ...            110.0        783562.0  38.963745   35.243322
3373     Turkey  2018                           100.000000                            95.2  ...            110.0        783562.0  38.963745   35.243322
3431     Uganda  2013                            13.900000                             0.8  ...            229.0        241038.0   1.373333   32.290275
3622     Zambia  2015                            31.100000                            14.0  ...             25.0        752618.0 -13.133897   27.849332

[39 rows x 21 columns]

Minimum Value:
 Entity                                                        Afghanistan
Year                                                                  2000
Access to electricity (% of population)                           1.252269
Access to clean fuels for cooking                                      0.0
Renewable-electricity-generating-capacity-per-capita                   0.0
Financial flows to developing countries (US $)                         0.0
Renewable energy share in the total final energy consumption (%)       0.0
Electricity from fossil fuels (TWh)                                    0.0
Electricity from nuclear (TWh)                                         0.0
Electricity from renewables (TWh)                                      0.0
Low-carbon electricity (% electricity)                                 0.0
Primary energy consumption per capita (kWh/person)                     0.0
Energy intensity level of primary energy (MJ/$2017 PPP GDP)           0.11
Value_co2_emissions_kt_by_country                                      10.0
Renewables (% equivalent primary energy)                               0.0
gdp_growth                                                          -62.07592
gdp_per_capita                                                     111.927225
Density\n(P/Km2)                                                      2.0
Land Area(Km2)                                                        21.0
Latitude                                                          -40.900557
Longitude                                                        -175.198242
dtype: object

Maximum Value:
 Entity                                                            Zimbabwe
Year                                                                  2020
Access to electricity (% of population)                              100.0
Access to clean fuels for cooking                                    100.0
Renewable-electricity-generating-capacity-per-capita               3060.19
Financial flows to developing countries (US $)                5202310000.0
Renewable energy share in the total final energy consumption (%)      96.04
Electricity from fossil fuels (TWh)                                5184.13
Electricity from nuclear (TWh)                                      809.41
Electricity from renewables (TWh)                                  2184.94
Low-carbon electricity (% electricity)                           100.00001
Primary energy consumption per capita (kWh/person)                262585.7
Energy intensity level of primary energy (MJ/$2017 PPP GDP)          32.57
Value_co2_emissions_kt_by_country                             10707219.73
Renewables (% equivalent primary energy)                         86.836586
gdp_growth                                                      123.139555
gdp_per_capita                                                 123514.1967
Density\n(P/Km2)                                                  8358.0
Land Area(Km2)                                                 9984670.0
Latitude                                                       64.963051
Longitude                                                     178.065032
dtype: object

Group by one or more columns and perform aggregation:
                  Electricity from fossil fuels (TWh)  Electricity from nuclear (TWh)  Electricity from renewables (TWh)
Entity
Afghanistan                                      3.98                            0.0                              15.56
Albania                                          1.02                            0.0                             108.87
Algeria                                        984.81                            0.0                               7.67
Angola                                          42.85                            0.0                              92.84
Antigua and Barbuda                              5.70                            0.0                               0.06
...                                               ...                            ...                                ...
Uzbekistan                                     913.01                            0.0                             135.20
Vanuatu                                          0.98                            0.0                               0.17
Yemen                                          103.06                            0.0                               1.93
Zambia                                          10.34                            0.0                             224.73
Zimbabwe                                        67.81                            0.0                             105.15

[176 rows x 3 columns]

DataFrame with a New Column:
          Entity  Year  Access to electricity (% of population)  Access to clean fuels for cooking  ...  Land Area(Km2)   Latitude  Longitude  Electricity gener
ated
0     Afghanistan  2000                              1.613591                                6.2  ...        652230.0  33.939110  67.709953                  0
.47
1     Afghanistan  2001                              4.074574                                7.2  ...        652230.0  33.939110  67.709953                  0
.59
2     Afghanistan  2002                              9.409158                                8.2  ...        652230.0  33.939110  67.709953                  0
.69
3     Afghanistan  2003                             14.738506                                9.5  ...        652230.0  33.939110  67.709953                  0
.94
4     Afghanistan  2004                             20.064968                               10.9  ...        652230.0  33.939110  67.709953                  0
.89
...           ...   ...                                   ...                                ...  ...             ...        ...        ...        ...
...
3644     Zimbabwe  2016                             42.561730                               29.8  ...        390757.0 -19.015438  29.154857                  6
.82
```

```
3645     Zimbabwe  2017                              44.178635                          29.8  ...       390757.0 -19.015438  29.154857                7
.35
3646     Zimbabwe  2018                              45.572647                          29.9  ...       390757.0 -19.015438  29.154857                9
.19
3647     Zimbabwe  2019                              46.781475                          30.1  ...       390757.0 -19.015438  29.154857                8
.24
3648     Zimbabwe  2020                              52.747670                          30.4  ...       390757.0 -19.015438  29.154857                7
.59

[3649 rows x 22 columns]

Correlation between column1 and column2:
 0.01726575405975201

Normalized DataFrame:
      Energy intensity level of primary energy (MJ/$2017 PPP GDP)
0                                                0.047135
1                                                0.050216
2                                                0.039741
3                                                0.039741
4                                                0.033580
...                                                   ...
3644                                             0.304683
3645                                             0.289587
3646                                             0.299445
3647                                             0.319162
3648                                             0.304683

[3649 rows x 1 columns]

Concatenated DataFrame:
          Entity  Year  Value_co2_emissions_kt_by_country
0     Afghanistan  2000                         760.000000
1     Afghanistan  2001                         730.000000
2     Afghanistan  2002                        1029.999971
3     Afghanistan  2003                        1220.000029
4     Afghanistan  2004                        1029.999971
...           ...   ...                                ...
3644     Zimbabwe  2016                       11020.000460
3645     Zimbabwe  2017                       10340.000150
3646     Zimbabwe  2018                       12380.000110
3647     Zimbabwe  2019                       11760.000230
3648     Zimbabwe  2020                                NaN

[3649 rows x 3 columns]
```

**Outcomes:**

Inculcate the knowledge of python libraries like numpy, pandas, matplotlib for scientific-computing and data visualization.

**Conclusion:** (Conclusion to be based on the objectives and outcomes achieved)

The experiment successfully demonstrated the use of Python Pandas for exploratory data analysis, providing valuable insights into the dataset through various tasks and analysis techniques.

**References:**


1. Daniel Arbuckle, Learning Python Testing, Packt Publishing, 1st Edition, 2014
2. Wesly J Chun, Core Python Applications Programming, O'Reilly, 3rd Edition, 2015
3. Wes McKinney, Python for Data Analysis, O'Reilly, 1st Edition, 2017
4. Albert Lukaszewsk, MySQL for Python, Packt Publishing, 1st Edition, 2010
5. Eric Chou, Mastering Python Networking, Packt Publishing, 2nd Edition, 2017