Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 07, July-2021

Data Analysis using Python

Kiranbala Nongthombam
University Institute of Sciences
(Mathematics Department) Chandigarh University,
Punjab, India

Deepika Sharma
University Institute of Sciences
(Mathematics Department)
Chandigarh University, Punjab, India

Abstract- **In this paper, the analysis of data using Python Programming Language is studied. The very basic processes of data analysis like cleaning, transforming, modeling of data is briefly explained in this paper and focus more on exploratory data analysis of an already existing dataset and finding the insights. Some graphical analysis of the data from the dataset will be shown using different libraries and functions of Python. Here, a dataset named "World Happiness report 2021" is used to analyze and extract various information in both numerical and pictorial form.**

Keywords:- Data analysis; python; data visualization; pandas; seaborn; exploratory data analysis

I. INTRODUCTION

Data are those raw facts and figures with no proper information hence need to be processed to get the desired information. While information is those results which we get after processing the raw data in different levels or extracted conclusions from a given dataset through a process called data analysis.

Data Analysis is simply the analysis of various data means cleaning the data, transforming it into understandable form, and then modeling data to extract some useful information for business use or an organizational use. It is mainly used in taking business decisions. Many libraries are available for doing the analysis. For example, NumPy, Pandas, Seaborn, Matplotlib, Sklearn, etc. [7].

- NumPy: NumPy is a library written in Python, used for numerical analysis in Python. It stores the data in the form of nd-arrays (n-dimensional arrays).
- Pandas: Pandas is mainly used for converting data into tabular form and hence, makes the data more structured and easily to read.
- Matplotlib: Matplotlib is a data visualisation and graphical plotting package for Python and its numerical extension NumPy that runs on all platforms.
- Seaborn: Seaborn is a Python data visualisation package based on matplotlib that is tightly connected with pandas data structures. The core component of Seaborn is visualisation, which aids in data exploration and comprehension.
- Sklearn: Scikit-learn is the most useful library for machine learning in Python. It includes numerous useful tools for classification, regression, clustering, and dimensionality reduction.

Data visualization will help the data analysis to make it more understandable and interactive by plotting or displaying the data in pictorial form. Pandas, a Python open-source package that deals with three different data structures: series, data frames, and panels, solves that need of analyzing and visualization of data [2].

Data analysis using Python makes task easier since Python Programming language has many advantages over any other programming language. It has prominent features like being a high-level programming language (the codes are in human readable form) it is easy to understand and use by any programmer or user. Many libraries and functions for statistical, numerical analysis are available in Python. Moreover, the source code is freely available to anyone (free and open source).

This paper includes all the basic terms and functions which are much needed by a beginner to know what data analysis is. The paper is divided broadly into 4 sections. In section II, the main steps in data analysis will be discussed. In section III, data analysis using python will be studied with all the basic needs of python in doing data analysis and data visualization will aid the analysis by representing them in picture format. In section IV, conclusion of the paper is given.

II. MAIN PHASES IN DATA ANALYSIS

A. Data requirements

Data are the most important unit in any study. Data must be provided as inputs to the analysis based on the analysis' requirements. The term "experimental unit" refers to the type of organization that would be used to gather data (e.g., a person or population of people). It is possible to identify and obtain specific population variables (such as height, weight, age, and salary). It doesn't matter whether the data is numerical or categorical.

B. Data Collecting:

The collecting of data is simply known as Data Collecting. Data is gathered from a variety of sources, including relational databases, cloud databases, and other sources, depending on the study' needs. Field sensors, such as traffic cameras, satellites, monitoring systems, and so on, can also be used as data sources.

C. Data processing

Data that are collected must be processed or organized for analysis. For instance, these may involve arranging data into rows and columns in a table format (known as structured data) for further analysis, often through the use of spreadsheet or statistical software.

D. Data cleaning:

The method of cleaning data after it has been processed and organized is known as data cleaning. It scans for data

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 07, July-2021

inconsistencies, duplicates, and errors, and then removes them. The data cleaning process includes tasks such as record matching, identifying data inaccuracy, data sort, outlier data identification, textual data spell checker, and data quality maintenance. As a consequence, it keeps us from having unexpected outcomes and assists us in delivering high-quality data, which is essential for a successful outcome.

E. *Exploratory data analysis:*

Once the datasets are cleaned and free of error, it can then be analyzed. A variety of techniques can be applied such as exploratory data analysis- understanding the messages contained within the obtained data and descriptive statistics- finding average, median, etc. Data visualization is also a technique used, in which the data is represented in a graphical format in order to obtain additional insights, regarding the information within the data [4].

F. *Modeling and algorithms:*

Mathematical formulas or models (known as algorithms), may be applied to the data in order to identify relationships among the variables; for example, using correlation or causation.

G. *Data product*

A data product, is a computer application that takes data inputs and generates outputs, feeding them back into the environment. It may be based on a model or algorithm.

III. DATA ANALYSIS USING PYTHON

In this section, data analysis using python will be studied. The most basic things like why using python for data analysis will be understood. Moreover, how anyone can start using python will be shown. The important libraries, the platforms, the dataset to carry out the analysis will be introduced. Usage of various python functions for numerical analysis are given along with various methods of plotting graphs or charts are discussed.

A. *Why using Python?*

Python is a high-level, interpreted, multi-purpose programming language. Many programming paradigms like procedural programming language, object-oriented programming is supported in python. It can be used for many applications, that includes statistical computing with various packages and functions. Moreover, it is easy to learn. It can be picked up by anyone including those who has less programming skills [9].

Some features of Python are as listed below:

- Open source and free
- Interpreted language
- Dynamic typesetting
- Portable
- Numerous IDE

B. *Packages used:*

- Numpy
- Pandas
- Seaborn
- Matplotlib

C. *Platform used:*

- Anaconda (Jupyter Notebook)

D. *Dataset used:*

- World Happiness record 2021

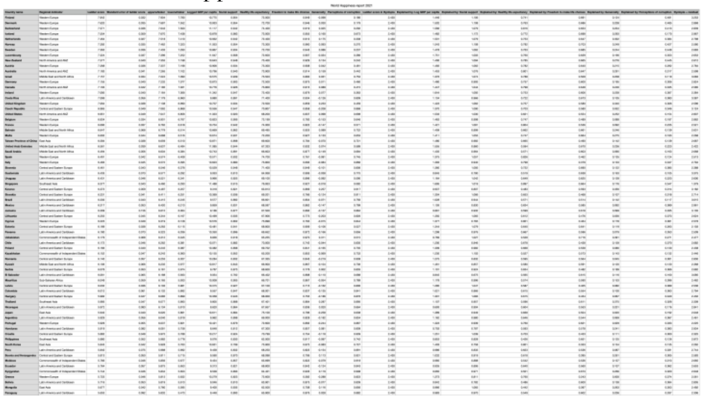

Fig. 1. A view of the dataset (World Happiness record 2021)

E. *Working with dataset*

- Importing libraries:

Libraries that would be used in the process of analysis are to be imported first. Here are the codes to import the libraries.

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
```

Fig. 2. Importing libraries

- Importing dataset

Here, the dataset (World Happiness report 2021) is imported in the jupyter notebook.

mydata=pd.read csv(“World Happiness report 2021.csv”)
mydata

```
mydata=pd.read_csv("World Happiness report 2021.csv")
mydata
```

	Country name	Regional indicator	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Ladder score in Dystopia	Explai by: GDP ca
0	Finland	Western Europe	7.842	0.032	7.904	7.780	10.775	0.954	72.000	0.949	-0.098	0.186	2.43	1.
1	Denmark	Western Europe	7.620	0.035	7.687	7.552	10.933	0.954	72.700	0.946	0.030	0.179	2.43	1.
2	Switzerland	Western Europe	7.571	0.036	7.643	7.500	11.117	0.942	74.400	0.919	0.025	0.292	2.43	1.
3	Iceland	Western Europe	7.554	0.059	7.670	7.438	10.878	0.983	73.000	0.955	0.160	0.673	2.43	1.
4	Netherlands	Western Europe	7.464	0.027	7.518	7.410	10.932	0.942	72.400	0.913	0.175	0.338	2.43	1.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
144	Lesotho	Sub-Saharan Africa	3.512	0.120	3.748	3.276	7.926	0.787	48.700	0.715	-0.131	0.915	2.43	0.
145	Botswana	Sub-Saharan Africa	3.467	0.074	3.611	3.322	9.782	0.784	59.269	0.824	-0.246	0.801	2.43	1.
146	Rwanda	Sub-Saharan Africa	3.415	0.068	3.548	3.282	7.676	0.552	61.400	0.897	0.061	0.167	2.43	0.
147	Zimbabwe	Sub-Saharan Africa	3.145	0.058	3.259	3.030	7.943	0.750	56.201	0.677	-0.047	0.821	2.43	0.
148	Afghanistan	South Asia	2.523	0.038	2.596	2.449	7.695	0.463	52.493	0.382	-0.102	0.924	2.43	0.

Fig. 3. Importing dataset

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

- Cleaning Data

Removing unwanted data or null values are done in the process of data cleaning. So, first we need to check the dataset whether it contains any null value or empty cells [6].
# isnull() returns true in the entry where there is no value or NA value. And sum() is used together with isnull() to find the total number of null values in every columns.
mydata.isnull().sum()

```
mydata.isnull().sum()

Country name                                  0
Regional indicator                            0
Ladder score                                  0
Standard error of ladder score                0
upperwhisker                                  0
lowerwhisker                                  0
Logged GDP per capita                         0
Social support                                0
Healthy life expectancy                       0
Freedom to make life choices                  0
Generosity                                    0
Perceptions of corruption                     0
Ladder score in Dystopia                      0
Explained by: Log GDP per capita              0
Explained by: Social support                  0
Explained by: Healthy life expectancy         0
Explained by: Freedom to make life choices    0
Explained by: Generosity                      0
Explained by: Perceptions of corruption       0
Dystopia + residual                           0
dtype: int64
```

Fig. 4. Checking null values in the dataset

According to our needs for the analysis, we can extract some particular rows or records from the dataset. Here is an example to extract the top most and last rows from the dataset.
#head() is used to extract the top-most data in the dataset. 5 is the default value of the head(). Here, top 10 rows from the dataset is taken.
headdata=mydata.head(10) headdata

```
headdata=mydata.head(10)
headdata
```

	Country name	Regional indicator	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Ladder score in Dystopia	Explained by: Lo GDP p capi
0	Finland	Western Europe	7.842	0.032	7.904	7.780	10.775	0.954	72.0	0.949	-0.098	0.186	2.43	1.44
1	Denmark	Western Europe	7.620	0.035	7.687	7.552	10.933	0.954	72.7	0.946	0.030	0.179	2.43	1.50
2	Switzerland	Western Europe	7.571	0.036	7.643	7.500	11.117	0.942	74.4	0.919	0.025	0.292	2.43	1.56
3	Iceland	Western Europe	7.554	0.059	7.670	7.438	10.878	0.983	73.0	0.955	0.160	0.673	2.43	1.48
4	Netherlands	Western Europe	7.464	0.027	7.518	7.410	10.932	0.942	72.4	0.913	0.175	0.338	2.43	1.50
5	Norway	Western Europe	7.392	0.035	7.462	7.323	11.053	0.954	73.3	0.960	0.093	0.270	2.43	1.54
6	Sweden	Western Europe	7.363	0.036	7.433	7.293	10.867	0.934	72.7	0.945	0.086	0.237	2.43	1.47
7	Luxembourg	Western Europe	7.324	0.037	7.396	7.252	11.647	0.908	72.6	0.907	-0.034	0.386	2.43	1.75
8	New Zealand	North America and ANZ	7.277	0.040	7.355	7.198	10.643	0.948	73.4	0.929	0.134	0.242	2.43	1.40
9	Austria	Western Europe	7.268	0.036	7.337	7.198	10.906	0.934	73.3	0.908	0.042	0.481	2.43	1.49

Fig. 5. Top 10 rows of the dataset

#tail() is used to extract the last rows in the dataset. 5 is the default value of the tail(). taildata=mydata.tail(10) taildata

```
taildata=mydata.tail(10)
taildata
```

	Country name	Regional indicator	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Ladder score in Dystopia	Expla by: GDP ca
139	Burundi	Sub-Saharan Africa	3.775	0.107	3.985	3.565	6.635	0.490	53.400	0.626	-0.024	0.607	2.43	0
140	Yemen	Middle East and North Africa	3.658	0.070	3.794	3.521	7.578	0.832	57.122	0.602	-0.147	0.800	2.43	0
141	Tanzania	Sub-Saharan Africa	3.623	0.071	3.762	3.485	7.876	0.702	57.999	0.833	0.183	0.577	2.43	0
142	Haiti	Latin America and Caribbean	3.615	0.173	3.953	3.276	7.477	0.540	55.700	0.593	0.422	0.721	2.43	0
143	Malawi	Sub-Saharan Africa	3.600	0.092	3.781	3.419	6.958	0.537	57.948	0.780	0.038	0.729	2.43	0
144	Lesotho	Sub-Saharan Africa	3.512	0.120	3.748	3.276	7.926	0.787	48.700	0.715	-0.131	0.915	2.43	0
145	Botswana	Sub-Saharan Africa	3.467	0.074	3.611	3.322	9.782	0.784	59.269	0.824	-0.246	0.801	2.43	1
146	Rwanda	Sub-Saharan Africa	3.415	0.068	3.548	3.282	7.676	0.552	61.400	0.897	0.061	0.167	2.43	0
147	Zimbabwe	Sub-Saharan Africa	3.145	0.058	3.259	3.030	7.943	0.750	56.201	0.677	-0.047	0.821	2.43	0
148	Afghanistan	South Asia	2.523	0.038	2.596	2.449	7.695	0.463	52.493	0.382	-0.102	0.924	2.43	0

Fig. 6. Last 10 rows of the dataset

F. Exploratory Data Analysis

In statistics, exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task. Exploratory data analysis was promoted by John Tukey to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments [4][8].

- *Data types:* Datatype refers to the type of data- int, object, float are the basic datatypes in python. Printing the types of data of all the columns in the dataset using dtypes-
  mydata.dtypes

```
mydata.dtypes

Country name                                   object
Regional indicator                             object
Ladder score                                  float64
Standard error of ladder score                float64
upperwhisker                                  float64
lowerwhisker                                  float64
Logged GDP per capita                         float64
Social support                                float64
Healthy life expectancy                       float64
Freedom to make life choices                  float64
Generosity                                    float64
Perceptions of corruption                     float64
Ladder score in Dystopia                      float64
Explained by: Log GDP per capita              float64
Explained by: Social support                  float64
Explained by: Healthy life expectancy         float64
Explained by: Freedom to make life choices    float64
Explained by: Generosity                      float64
Explained by: Perceptions of corruption       float64
Dystopia + residual                           float64
dtype: object
```

Fig. 7. Datatypes of the whole coumns in the dataset

- *Describing the dataset:* Describing data of a dataset means extracting the summary of the given dataframe such as mean, count, min, max, etc. It can be done using describe() function-

For the whole dataset: mydata.describe()

```
mydata.describe()
```

	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Ladder score in Dystopia
count	149.000000	149.000000	149.000000	149.000000	149.000000	149.000000	149.000000	149.000000	149.000000	149.000000	1.490000e+02
mean	5.532839	0.058752	5.648007	5.417631	9.432208	0.814745	64.992799	0.791597	-0.015134	0.727450	2.430000e+00
std	1.073924	0.022001	1.054330	1.094879	1.158601	0.114889	6.762043	0.113332	0.150657	0.179226	5.347044e-15
min	2.523000	0.026000	2.596000	2.449000	6.635000	0.463000	48.478000	0.382000	-0.288000	0.082000	2.430000e+00
25%	4.852000	0.043000	4.991000	4.706000	8.541000	0.750000	59.802000	0.718000	-0.126000	0.667000	2.430000e+00
50%	5.534000	0.054000	5.625000	5.413000	9.569000	0.832000	66.603000	0.804000	-0.036000	0.781000	2.430000e+00
75%	6.255000	0.070000	6.344000	6.128000	10.421000	0.905000	69.600000	0.877000	0.079000	0.845000	2.430000e+00
max	7.842000	0.173000	7.904000	7.780000	11.647000	0.983000	76.953000	0.970000	0.542000	0.939000	2.430000e+00

Fig. 8. Summary of the whole dataset

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 07, July-2021

For some selected rows: taildata.describe()

```
taildata.describe()
```

	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Ladder score in Dystopia	Explained by: Log GDP per capita	Explain by: Soc supp
count	10.000000	10.000000	10.00000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.00000	10.00	10.000000	10.000
mean	3.433300	0.087100	3.60370	3.262500	7.754600	0.643700	56.023200	0.692900	0.000700	0.70620	2.43	0.391000	0.408
std	0.362624	0.038336	0.41074	0.325312	0.829727	0.140316	3.672269	0.151356	0.191256	0.22111	0.00	0.289745	0.316
min	2.523000	0.038000	2.59600	2.449000	6.635000	0.463000	48.700000	0.382000	-0.246000	0.16700	2.43	0.000000	0.000
25%	3.428000	0.068500	3.56375	3.276000	7.502250	0.537750	53.975000	0.608000	-0.123750	0.63550	2.43	0.302750	0.169
50%	3.556000	0.072500	3.75500	3.302000	7.685500	0.627000	56.661500	0.696000	-0.035500	0.76450	2.43	0.367000	0.371
75%	3.621000	0.103250	3.79075	3.468500	7.913500	0.775500	57.986250	0.813000	0.055250	0.81600	2.43	0.446500	0.705
max	3.775000	0.173000	3.98500	3.565000	9.782000	0.832000	61.400000	0.897000	0.422000	0.92400	2.43	1.099000	0.831

Fig. 9. Summary of some selected entries(10 last rows)

- *Correlations:* Correlation shows the relation between any two variables in the dataset. The strength of a linear relation between two variables is measured by correlation. Printing Correlation of various attributes using corr() [1].

# For whole dataset-
mydata.corr()

```
mydata.corr()
```

	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Ladder score in Dystopia	Explained by: L GDP p capi
Ladder score	1.000000	-0.470787	0.999347	0.999396	0.789760	0.756888	0.768099	0.607753	-0.017799	-0.421140	NaN	0.7897
Standard error of ladder score	-0.470787	1.000000	-0.438612	-0.501150	-0.645799	-0.530815	-0.583805	-0.275182	0.138349	0.276997	NaN	-0.6457
upperwhisker	0.999347	-0.438612	1.000000	0.997489	0.777995	0.749215	0.758455	0.607797	-0.012616	-0.417560	NaN	0.7779
lowerwhisker	0.999396	-0.501150	0.997489	1.000000	0.800064	0.763299	0.776364	0.606944	-0.022794	-0.423976	NaN	0.8000
Logged GDP per capita	0.789760	-0.645799	0.777995	0.800064	1.000000	0.785299	0.859461	0.432323	-0.199286	-0.342337	NaN	1.0000
Social support	0.756888	-0.530815	0.749215	0.763299	0.785299	1.000000	0.723256	0.482930	-0.114946	-0.203207	NaN	0.7852
Healthy life expectancy	0.768099	-0.583805	0.758455	0.776364	0.859461	0.723256	1.000000	0.461494	-0.161750	-0.364374	NaN	0.8594
Freedom to make life choices	0.607753	-0.275182	0.607797	0.606944	0.432323	0.482930	0.461494	1.000000	0.169437	-0.401363	NaN	0.4323
Generosity	-0.017799	0.138349	-0.012616	-0.022794	-0.199286	-0.114946	-0.161750	0.169437	1.000000	-0.163962	NaN	-0.1992
Perceptions of corruption	-0.421140	0.276997	-0.417560	-0.423976	-0.342337	-0.203207	-0.364374	-0.401363	-0.163962	1.000000	NaN	-0.3423
Ladder score in Dystopia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
Explained by: Log GDP per capita	0.789745	-0.645776	0.777981	0.800048	1.000000	0.785287	0.859446	0.432350	-0.199229	-0.342310	NaN	1.0000

Fig. 10. Correlation of the whole dataset

For some selected coulmns or attributes-
mydata[['Country name', 'Regional indicator', 'Ladder score', 'Standard error of ladder score', 'Logged GDP per capita', 'Social support', 'Healthy life expectancy', 'Generosity', 'Perceptions of corruption']].corr()

```
corrdata=mydata[['Country name',
        'Ladder score',
        'Standard error of ladder score',
        'Logged GDP per capita',
        'Social support',
        'Healthy life expectancy',
        'Generosity','Perceptions of corruption']].corr()
corrdata
```

	Ladder score	Standard error of ladder score	Logged GDP per capita	Social support	Healthy life expectancy	Generosity	Perceptions of corruption
Ladder score	1.000000	-0.470787	0.789760	0.756888	0.768099	-0.017799	-0.421140
Standard error of ladder score	-0.470787	1.000000	-0.645799	-0.530815	-0.583805	0.138349	0.276997
Logged GDP per capita	0.789760	-0.645799	1.000000	0.785299	0.859461	-0.199286	-0.342337
Social support	0.756888	-0.530815	0.785299	1.000000	0.723256	-0.114946	-0.203207
Healthy life expectancy	0.768099	-0.583805	0.859461	0.723256	1.000000	-0.161750	-0.364374
Generosity	-0.017799	0.138349	-0.199286	-0.114946	-0.161750	1.000000	-0.163962
Perceptions of corruption	-0.421140	0.276997	-0.342337	-0.203207	-0.364374	-0.163962	1.000000

Fig. 11. Correlation of some attributes in the dataset

G. *Graphical EDA*

Fundamentally, graphical exploratory data analysis is the graphical equivalent to conventional non-graphical exploratory data analysis. EDA that examines data sets in order to summarise their statistical characteristics by focusing on the same four main features, such as measures of central tendency, measures of spread, distribution form, and the presence of outliers. We also divided GEDA into three categories: Univariate GEDA, Bivariate GEDA, and Multivariate GEDA. We'll go through these important varieties in more detail in the following paragraphs and aspects of GEDA [5].

First, a subset of the dataframe is taken to analyse or visualize using it.

```
subdata=mydata[['Ladder score',
        'Standard error of ladder score', 'upperwhisker', 'lowerwhisker',
        'Logged GDP per capita', 'Social support', 'Healthy life expectancy',
        'Freedom to make life choices', 'Generosity',
        'Perceptions of corruption']]
subdata
```

	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
0	7.842	0.032	7.904	7.780	10.775	0.954	72.000	0.949	-0.098	0.186
1	7.620	0.035	7.687	7.552	10.933	0.954	72.700	0.946	0.030	0.179
2	7.571	0.036	7.643	7.500	11.117	0.942	74.400	0.919	0.025	0.292
3	7.554	0.059	7.670	7.438	10.878	0.983	73.000	0.955	0.160	0.673
4	7.464	0.027	7.518	7.410	10.932	0.942	72.400	0.913	0.175	0.338
...	...	...	...	...	...	...	...	...	...	...
144	3.512	0.120	3.748	3.276	7.926	0.787	48.700	0.715	-0.131	0.915
145	3.467	0.074	3.611	3.322	9.782	0.784	59.269	0.824	-0.246	0.801
146	3.415	0.068	3.548	3.282	7.676	0.552	61.400	0.897	0.061	0.167
147	3.145	0.058	3.259	3.030	7.943	0.750	56.201	0.677	-0.047	0.821
148	2.523	0.038	2.596	2.449	7.695	0.463	52.493	0.382	-0.102	0.924

Fig. 12. A subset of the dataframe

1. *Univariate GEDA*

- Histogram: A histogram is a data representation that looks like a bar graph that buckets a variety of outcomes into columns along the x-axis. The y-axis can be used to illustrate data distributions by representing the numerical count or percentage of occurrences in each column. Histogram in python can be drawn using matplotlib.pyplot.hist()-

```
plt.hist(subdata,bins=7)
plt.xlabel("Various Parameters")
plt.ylabel("Counts")
```

```
Text(0, 0.5, 'Counts')
```

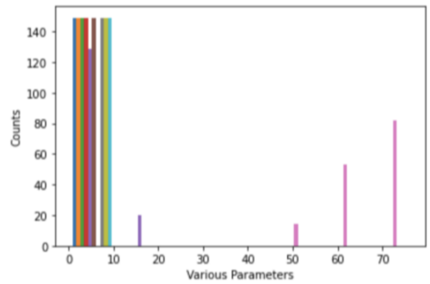

Fig. 13. Histogram

- Stem Plot: A stem plot draws vertical lines from the baseline to the y axis and sets a marker at each x point. The x-positions are not necessary. The formats can be specified as keyword-arguments or as positional arguments. Stem plot in python can be drawn using matplotlib.pyplot.stem()

```
plt.stem(mydata['Ladder score'])
```

```
<StemContainer object of 3 artists>
```

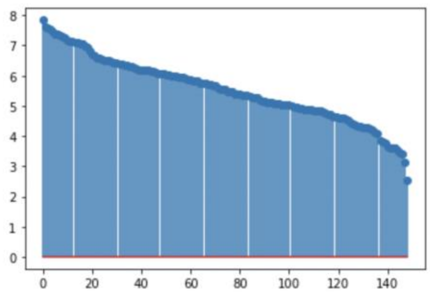

Fig. 14. Stem plot

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 07, July-2021

- Box Plot: Box plot is a visual representation of and comparison of groups of data. The box plot depicts the level, spread, and symmetry of a data distribution by using the median, approximate quartiles, outliers, and the lowest and highest data points (extreme values) [10].

```
sns.boxplot(x="Social support",y="Generosity",
            data=headdata,palette="coolwarm")
```

<AxesSubplot:xlabel='Social support', ylabel='Generosity'>

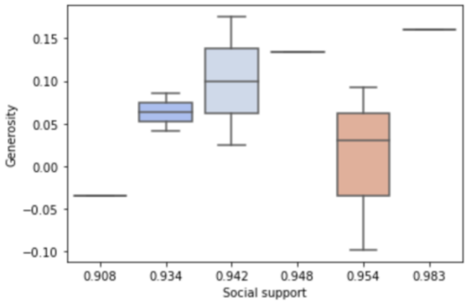

Fig. 15. Boxplot

2. *Multivariate GEDA*

- Scatter plot: Dots are used to indicate values for two different numeric variables in a scatter plot. The values for each data point are indicated by the position of each dot on the horizontal and vertical axes. Scatter plots are used to see how variables relate to one another.
  Here, scatter plot of "Ladder score" against "Standard error of ladder score" is plotted below-

```
plt.scatter(x="Ladder score",y="Standard error of ladder score",
            data=subdata,marker="*",color="green")
```

<matplotlib.collections.PathCollection at 0x7fcac03b2a00>

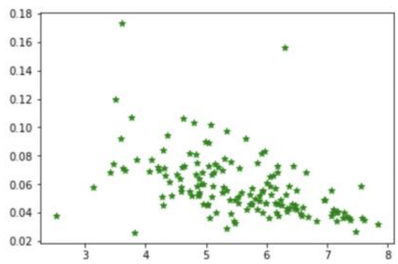

Fig. 16. Scatter Plot

- Heat Maps: A heatmap is a graphical depiction of data that uses a color-coding method to represent various values. It represents two- dimensional table of color-shades. This technique of plotting is popularly used in biology to represent gene expression and other multivariate data [3].
  A heatmap example is shown in the fig. 17.

```
plt.figure(figsize=(13, 5))
sns.heatmap(cordata,annot=True)
```

<AxesSubplot:>

Fig. 17. Heatmap

- Count Plot: A Seaborn count plot is a graphical representation of the number of occurrences or frequency for each category data using bars to depict the number of occurrences or frequency. The countplot() function is used to visualize the number of observations in each categorical category as bars. Here, Count plot is plotted for the subdata dataframe.

```
plt.figure(figsize=(15,5))
sns.countplot(data=subdata)
```

<AxesSubplot:ylabel='count'>

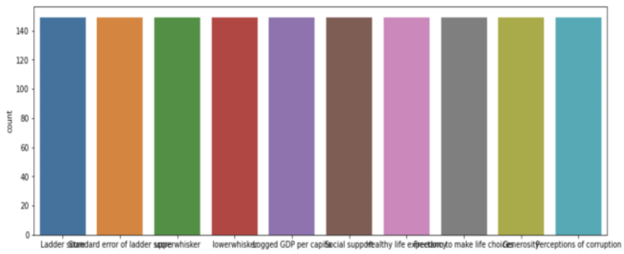

Fig. 18. Countplot

IV. CONCLUSION

In this paper, various phases of data analysis including data collection, cleaning and analysis are discussed briefly. Explorative data analysis is mainly studied here. For the implementation, Python programming language is used. For detailed research, jupyter notebook is used. Different Python libraries and packages are introduced. Using various analysis and visulaization methods, numerous results are extracted. The dataset "World Happiness Record 2021" is used and extract important informations like the difference in the score of happiness of different countries, the dependence of one attribute in building up the score, how a variable affects another variable, etc. are seen in this analysis and various graphs has been plotted using various attributes in the dataset and draw conclusions in an easy way.

V. ACKNOWLEDGMENT

I express my heartfelt gratitude towards my mentor Ms. Deepika Sharma for guiding me to accomplish such a great work. I offer my sincere appreciation towards the Head of Department, University Institute of Sciences (Mathematics Department), Chandigarh University for giving me such a chance to gain a wider view of knowledge.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

VI. REFERENCES

[1] Viv Bewick, Liz Cheek, and Jonathan Ball. *Statistics review 7: Correlation and regression. Critical care*, 2003.

[2] Dr Ossama Embarak, Embarak, and Karkal. *Data analysis and visualization using python*. Springer, 2018.

[3] Nils Gehlenborg and Bang Wong. Heat maps. *Nature Methods*, 2012.

[4] Michel Jambu. *Exploratory and multivariate data analysis*. Elsevier, 1991.

[5] Matthieu Komorowski, Dominic C Marshall, Justin D Salciccioli, and Yves Crutain. Exploratory data analysis. *Secondary analysis of electronic health records*, 2016.

[6] Wes McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.", 2012.

[7] Fabio Nelli. *Python data analytics: Data analysis and science using PANDAs, Matplotlib and the Python Programming Language*. Apress, 2015.

[8] Kabita Sahoo, Abhaya Kumar Samal, Jitendra Pramanik, and Subhendu Kumar Pani. Exploratory data analysis using python. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 2019.

[9] Guido Van Rossum et al. Python programming language. In *USENIX annual technical conference*, 2007.

[10] David F Williamson, Robert A Parker, and Juliette S Kendrick. The box plot: a simple visual method to interpret data. *Annals of internal medicine*, 1989.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)