International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 12 | Dec 2022 www.irjet.net p-ISSN: 2395-0072

An Overview of Python for Data Analytics

Shailendra Chaudekar[1]

[1]PG Diploma, Big Data Analytics, CDAC Kolkata, India

Abstract - *The data analysis in this research is done with Python. The programming language is being explored. This article quickly describes the fundamental processes of data analysis, such as cleaning, transforming, and modelling data, with an emphasis on exploratory data analysis. Data analysis of an existing dataset and discovery of insights Using several Python libraries and methods, some graphical analysis of data from the dataset will be demonstrated. Here, A dataset titled "World Happiness Report 2022" is used to examine and extract data in both numerical and visual formats.*

Key Words: Python, Data Analytics, Data Science, Data Visualization, Python Libraries, Numpy, Pandas, Graphical Exploratory Data Analytics

1. INTRODUCTION

Python is a high-level, general-purpose programming language that has recently gained popularity. It enables programmers to write code in fewer lines, which is not achievable with other languages. Python programming is notable for its support of many programming paradigms. Python includes a broad, extendable standard library. Python's primary characteristics include simple and easy to learn, freeware and open source, and high-level programming language. Platform independence, portability, dynamically typed, procedure and object-oriented, Extensible, Embedded, and Extensive Library.

We hope to provide a quick overview of Python in the areas of data science, IoT, and machine learning in this article. Python is well-known for having a plethora of libraries that aid with data analysis and scientific computing. For example, we may create a Python programme to assist data analysts in analysing massive volumes of data for scientific computing. This paper requires a basic understanding of statistics as well as some experience with any C-style language. A working understanding of Python is advantageous but not required.

1.1 Introduction to Data Science

Data science is a multidisciplinary field that extracts knowledge and insights from organised and unstructured data using scientific techniques, procedures, tools, and systems. Data analytics, data mining, and big data are all connected to data science. It comprehends the data phenomena. It incorporates techniques and theories from a variety of domains, including mathematics, statistics, computer science, and information science. Statistics is one of the most essential disciplines for providing tools and methods for finding structure in and providing greater insight into data, as well as for analysing and quantifying uncertainty. Python has a number of preconfigured modules for working on data science tasks.

1.2 Python in Data Science Research and Education

Data Science is a booming field of study that combines statistics, computer science, and a variety of applied scientific fields. As is customary in such transdisciplinary settings, fields of study, Data Science education, mentorship, and research takes ideas and inspiration from a variety of other domains, including the mathematical sciences, computer science, and numerous business and application domains.

This field presents an overview to Data Science, including a wide range of essential issues and approaches for working with huge data. Data collection, integration, management, modelling, analysis, visualisation, prediction, and informed decision-making are among the topics to be covered, as are data security and data privacy. This integrates databases, data warehousing, statistics, data mining, data visualisation, high-performance computing, cloud computing, and business intelligence into the basic disciplines of data science. Professional abilities such as communication, presentation, and data storytelling will be cultivated. Through hands-on projects and case studies in a range of business, engineering, social sciences, and biological sciences areas, students will get a practical grasp of data science

2. DATA ANALYSIS USING PYTHON

The analysis of varied data basically entails cleaning the data, translating it into comprehensible form, and then modelling data to extract some relevant information for commercial or organisational usage. It is mostly employed in commercial decisions. There are several libraries accessible for conducting the analysis. NumPy, Pandas, Seaborn, Matplotlib, Sklearn, and more are examples.

- NUMPY : NumPy is the foundational Python library for scientific computing. It is a Python library that includes a multidimensional array object, various derived objects (such as masked arrays and matrices), and a variety of routines for performing fast array operations such as mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic

statistical operations, random simulation, and much more.

- PANDAS : Pandas is a data manipulation and analysis software package created for the Python computer language. It provides data structures and functions for manipulating numerical tables and time series in particular. It is free software distributed under the BSD three-clause licence. The name is taken from the word "panel data," which is an econometrics term for data sets that comprise observations for the same persons over several periods. Its name is an allusion to the term "Python data analysis." Wes McKinney began developing what would become pandas while working as a researcher at AQR Capital from 2007 to 2010.

- MatPlotlib : Matplotlib is a Python package that allows you to create static, animated, and interactive visualisations. Matplotlib makes simple things simple and difficult things possible.

    1. Produce plots suitable for publishing.
    2. Create interactive figures that can be zoomed, panned, and updated.
    3. Change the visual style and layout.
    4. Export to a variety of file formats.
    5. Incorporate JupyterLab and Graphical User Interfaces.

- Sklearn : The Sklearn Library is mostly used for data modelling and provides effective, user-friendly tools for any type of predictive data analysis. This library's principal use cases may be divided into six groups, which are as follows:

    1. Pre-processing
    2. Regression
    3. Classification
    4. Clustering
    5. Model Selection
    6. Dimensionality Reduction

A. Data Requirements

The most crucial unit in every study is data. Data must be given as inputs to the analysis based on the needs of the analysis. The word "experimental unit" refers to the sort of organisation employed to collect data (for example, a person or community of individuals). Specific demographic variables (such as height, weight, age, and pay) can be identified and obtained. It makes no difference whether the data is numerical or categorical.

B. Data Collection:

Depending on the demands of the study, data is acquired from a range of sources, including relational databases, cloud databases, and other sources. Data sources can also include field sensors such as traffic cameras, satellites, and monitoring systems.

C. Data processing

Data must be processed or structured before it can be analysed. These may include, for example, organising data into rows and columns in a tabular format (known as structured data) for further analysis, generally using a spreadsheet or statistical software.

D. Data cleaning:

Data cleaning is the process of cleansing data after it has been processed and arranged. It searches for and eliminates data inconsistencies, duplication, and mistakes. Record matching, identifying data inaccuracy, data sorting, outlier data detection, textual data spell checking, and data quality maintenance are all part of the data cleaning process. As a result, it prevents unexpected consequences and aids us in supplying high-quality data, which is critical for a successful outcome.

E. Exploratory data analysis:

After the datasets have been cleaned and checked for errors, they may be examined. A range of approaches, such as exploratory data analysis understanding the messages contained within the gathered data and descriptive statistics—finding the average, median, and so on can be used. Data visualisation is a technique in which data is displayed graphically in order to get more insights into the information contained within the data.

F. Data Modelling and algorithms:

Mathematical formulae or models can be applied to data to detect links between variables, such as correlation or causation.

G. Data product

A data product is a computer application that takes data inputs and generates outputs that are then fed back into the environment. It might be built on a model or an algorithm.

This review will explore data analysis in Python. The most fundamental concepts, such as why Python is used for data analysis, will be grasped. Furthermore, how anyone may begin using Python will be demonstrated. The essential

libraries, platforms, and datasets for doing the analysis will be introduced. The use of several Python functions for numerical analysis is explained, as well as various ways for generating graphs or charts.

Packages used: 1. Numpy 2.Pandas 3. Seaborn 4. Matplotlib

Platform used: Anaconda (Jupyter Notebook

Dataset used: World Happiness record 2022

RANK	Country	Happiness	Whisker-hi	Whisker-lo	Dystopia (:	GDP per ca	Social supp	Healthy lif	Freedom t	Generosity	Perceptions of corruption
1	Finland	7.821	7.886	7.756	2.518	1.892	1.258	0.775	0.736	0.109	0.534
2	Denmark	7.636	7.71	7.563	2.226	1.953	1.243	0.777	0.719	0.188	0.532
3	Iceland	7.557	7.651	7.464	2.32	1.936	1.32	0.803	0.718	0.27	0.191
4	Switzerlan	7.512	7.586	7.437	2.153	2.026	1.226	0.822	0.677	0.147	0.461
5	Netherlan	7.415	7.471	7.359	2.137	1.945	1.206	0.787	0.651	0.271	0.419
6	Luxembou	7.404	7.501	7.307	2.042	2.209	1.155	0.79	0.7	0.12	0.388
7	Sweden	7.384	7.454	7.315	2.003	1.92	1.204	0.803	0.724	0.218	0.512
8	Norway	7.365	7.44	7.29	1.925	1.997	1.239	0.786	0.728	0.217	0.474
9	Israel	7.364	7.426	7.301	2.634	1.826	1.221	0.818	0.568	0.155	0.143
10	New Zeala	7.2	7.279	7.12	1.954	1.852	1.235	0.752	0.68	0.245	0.483
11	Austria	7.163	7.237	7.089	2.148	1.931	1.165	0.774	0.623	0.193	0.329
12	Australia	7.162	7.244	7.081	2.011	1.9	1.203	0.772	0.676	0.258	0.341
13	Ireland	7.041	7.121	6.961	1.743	2.129	1.166	0.779	0.627	0.19	0.408
14	Germany	7.034	7.122	6.947	2.142	1.924	1.088	0.776	0.585	0.163	0.358
15	Canada	7.025	7.107	6.943	1.924	1.886	1.188	0.783	0.659	0.217	0.368
16	United Sta	6.977	7.065	6.888	2.214	1.982	1.182	0.628	0.574	0.22	0.177
17	United Kin	6.943	7.018	6.867	1.967	1.867	1.143	0.75	0.597	0.289	0.329
18	Czechia	6.92	7.029	6.811	2.263	1.815	1.26	0.715	0.66	0.158	0.048
19	Belgium	6.805	6.89	6.72	2.283	1.907	1.106	0.764	0.492	0.049	0.204
20	France	6.687	6.758	6.615	1.895	1.863	1.219	0.808	0.567	0.07	0.266
21	Bahrain	6.647	6.779	6.514	2.092	1.854	1.029	0.625	0.693	0.199	0.155
22	Slovenia	6.63	6.718	6.542	1.885	1.81	1.249	0.769	0.685	0.118	0.115
23	Costa Rica	6.582	6.683	6.481	2.346	1.584	1.054	0.744	0.661	0.089	0.102
24	United Ara	6.576	6.66	6.492	1.809	1.998	0.98	0.633	0.702	0.204	0.25
25	Saudi Arab	6.523	6.637	6.409	2.075	1.87	1.092	0.577	0.651	0.078	0.18
26	Taiwan Pr	6.512	6.596	6.429	2.002	1.897	1.095	0.733	0.542	0.075	0.168
27	Singapore	6.48	6.569	6.392	0.932	2.149	1.127	0.851	0.672	0.163	0.587
28	Romania	6.477	6.575	6.379	2.446	1.719	1.006	0.655	0.605	0.039	0.006

Fig -1: A view of the dataset (World Happiness record 2022)

A. Import libraries:

Libraries that will be utilised in the analysis process should be imported first. Here are the import codes for the libraries.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

import sklearn

import csv

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

Fig -2: Importing libraries

B. Import Dataset

The dataset (World Happiness Report 2022) is imported into the Jupyter notebook in this case.

data = pd.read_csv("Path\World Happiness Report 2022.csv")

data

```
data = pd.read_csv("C:\\Users\\Lenovo\\Desktop\\MyWork\\World Happiness Report 2022.csv")
```

```
data
```

	RANK	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.83) + residual	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
0	1	Finland	7.821	7.886	7.756	2.518	1.892	1.258	0.775	0.736	0.109	0.534
1	2	Denmark	7.636	7.710	7.563	2.226	1.953	1.243	0.777	0.719	0.188	0.532
2	3	Iceland	7.557	7.651	7.464	2.320	1.936	1.320	0.803	0.718	0.270	0.191
3	4	Switzerland	7.512	7.586	7.437	2.153	2.026	1.226	0.822	0.677	0.147	0.461
4	5	Netherlands	7.415	7.471	7.359	2.137	1.945	1.206	0.787	0.651	0.271	0.419
...	...	...	...	...	...	...	...	...	...	...	...	...
141	142	Botswana*	3.471	3.667	3.275	0.187	1.503	0.815	0.280	0.571	0.012	0.102
142	143	Rwanda*	3.268	3.462	3.074	0.536	0.785	0.133	0.462	0.621	0.187	0.544
143	144	Zimbabwe	2.995	3.110	2.880	0.548	0.947	0.690	0.270	0.329	0.106	0.105
144	145	Lebanon	2.955	3.049	2.862	0.216	1.392	0.498	0.631	0.103	0.082	0.034
145	146	Afghanistan	2.404	2.469	2.339	1.263	0.758	0.000	0.289	0.000	0.089	0.005

146 rows × 12 columns

Fig -3: Importing dataset

C. Data Cleaning:

Data cleaning involves the removal of unnecessary data or null values. So, initially, we must examine the dataset to see whether it includes any null values or empty cells. *# isnull()* returns true in entries with no value or a NA value. And sum() is used in conjunction with *isnull()* to get the total number of null values in each column.

data.isnull().sum()

```
In [39]: 1 data.isnull().sum()

Out[39]: RANK                          0
         Country                       0
         Happiness score               0
         Whisker-high                  0
         Whisker-low                   0
         Dystopia (1.83) + residual    0
         GDP per capita                0
         Social support                0
         Healthy life expectancy       0
         Freedom to make life choices  0
          Generosity                   0
         Perceptions of corruption     0
         dtype: int64
```

Fig. 4. Checking null values in the dataset

We can extract specific rows or records from the dataset based on our analytical needs. Here's an example of extracting the first and end rows of a dataset.

#head() returns the data at the top of the dataset. The head's default value is 5. (). The top ten rows of the dataset are used in this case.

headdata = data.head(10)

headdata

```
1 headdata = data.head(10)
2 headdata
```

	RANK	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.83) + residual	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
0	1	Finland	7.821	7.886	7.756	2.518	1.892	1.258	0.775	0.736	0.109	0.534
1	2	Denmark	7.636	7.710	7.563	2.226	1.953	1.243	0.777	0.719	0.188	0.532
2	3	Iceland	7.557	7.651	7.464	2.320	1.936	1.320	0.803	0.718	0.270	0.191
3	4	Switzerland	7.512	7.586	7.437	2.153	2.026	1.226	0.822	0.677	0.147	0.461
4	5	Netherlands	7.415	7.471	7.359	2.137	1.945	1.206	0.787	0.651	0.271	0.419
5	6	Luxembourg*	7.404	7.501	7.307	2.042	2.209	1.155	0.790	0.700	0.120	0.388
6	7	Sweden	7.384	7.454	7.315	2.003	1.920	1.204	0.803	0.724	0.218	0.512
7	8	Norway	7.365	7.440	7.290	1.925	1.997	1.239	0.786	0.728	0.217	0.474
8	9	Israel	7.364	7.426	7.301	2.634	1.826	1.221	0.818	0.568	0.155	0.143
9	10	New Zealand	7.200	7.279	7.120	1.954	1.852	1.235	0.752	0.680	0.245	0.483

Fig. 5. Top 10 rows of the dataset

#tail() is used to retrieve the dataset's final rows. The tail() default value is 5.

taildata=data.tail(10)

Taildata

```
1 taildata = data.tail(10)
2 taildata
```

	RANK	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.83) + residual	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
136	137	Zambia	3.760	3.902	3.618	1.135	0.930	0.577	0.306	0.525	0.203	0.083
137	138	Malawi	3.750	3.941	3.560	1.661	0.648	0.279	0.388	0.477	0.140	0.157
138	139	Tanzania	3.702	3.847	3.558	0.735	0.848	0.597	0.425	0.578	0.248	0.270
139	140	Sierra Leone	3.574	3.732	3.416	1.556	0.686	0.416	0.273	0.387	0.202	0.055
140	141	Lesotho*	3.512	3.748	3.276	1.312	0.839	0.848	0.000	0.419	0.076	0.018
141	142	Botswana*	3.471	3.667	3.275	0.187	1.503	0.815	0.280	0.571	0.012	0.102
142	143	Rwanda*	3.268	3.462	3.074	0.536	0.785	0.133	0.462	0.621	0.187	0.544
143	144	Zimbabwe	2.995	3.110	2.880	0.548	0.947	0.690	0.270	0.329	0.106	0.105
144	145	Lebanon	2.955	3.049	2.862	0.216	1.392	0.498	0.631	0.103	0.082	0.034
145	146	Afghanistan	2.404	2.469	2.339	1.263	0.758	0.000	0.289	0.000	0.089	0.005

Fig. 6. Last 10 rows of the dataset

D. Exploratory Data Analysis In statistics:

Exploratory data analysis is a process of analysing data sets in order to summarise their major properties, which is frequently done with statistical graphics and other data visualisation approaches. A statistical model may or may not be utilised, but the primary goal of EDA is to explore what the data can tell us beyond the formal modelling or hypothesis testing work. John Tukey championed exploratory data analysis to encourage statisticians to investigate the data and maybe create hypotheses that could lead to future data gathering and experiments.

Data types:

In Python, datatype refers to the type of data- int, object, and float are the main datatypes. Using dtypes, print the data types of all columns in the dataset.

data.dtypes

```
1 data.dtypes
```

```
RANK                          int64
Country                      object
Happiness score             float64
Whisker-high                float64
Whisker-low                 float64
Dystopia (1.83) + residual  float64
GDP per capita              float64
Social support              float64
Healthy life expectancy     float64
Freedom to make life choices float64
 Generosity                 float64
Perceptions of corruption   float64
dtype: object
```

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

Describing the Dataset:

Describing data from a dataset entails obtaining a summary of the supplied data frame, such as mean, count, min, max, and so on. It is possible to accomplish this using the describe() function-

data.describe()

```
1 data.describe()
```

	RANK	Happiness score	Whisker-high	Whisker-low	Dystopia (1.83) + residual	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
count	146.000000	146.000000	146.000000	146.000000	146.000000	146.000000	146.000000	146.000000	146.000000	146.000000	146.000000
mean	73.500000	5.553575	5.673589	5.433568	1.831808	1.410445	0.905863	0.586171	0.517226	0.147377	0.154781
std	42.290661	1.086843	1.065621	1.109380	0.534994	0.421663	0.280122	0.176336	0.145859	0.082799	0.127514
min	1.000000	2.404000	2.469000	2.339000	0.187000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	37.250000	4.888750	5.006250	4.754750	1.555250	1.095500	0.732000	0.463250	0.440500	0.089000	0.068250
50%	73.500000	5.568500	5.680000	5.453000	1.894500	1.445500	0.957500	0.621500	0.543500	0.132500	0.119500
75%	109.750000	6.305000	6.448750	6.190000	2.153000	1.784750	1.114250	0.719750	0.626000	0.197750	0.198500
max	146.000000	7.821000	7.886000	7.756000	2.844000	2.209000	1.320000	0.942000	0.740000	0.468000	0.587000

Fig. 8. Summary of the whole dataset

taildata.describe()

```
1 taildata.describe()
```

	RANK	Happiness score	Whisker-high	Whisker-low	Dystopia (1.83) + residual	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
count	10.00000	10.000000	10.000000	10.000000	10.000000	10.000000	10.00000	10.000000	10.000000	10.000000	10.000000
mean	141.50000	3.339100	3.492700	3.185800	0.914900	0.933600	0.48530	0.332400	0.401000	0.134500	0.137300
std	3.02765	0.437869	0.475988	0.403095	0.539941	0.288021	0.28091	0.163592	0.206852	0.073775	0.162556
min	137.00000	2.404000	2.469000	2.339000	0.187000	0.648000	0.00000	0.000000	0.000000	0.012000	0.005000
25%	139.25000	3.063250	3.198000	2.928500	0.539000	0.764750	0.31325	0.274750	0.343500	0.083750	0.039250
50%	141.50000	3.491500	3.699500	3.275500	0.935000	0.843500	0.53750	0.297500	0.448000	0.123000	0.092500
75%	143.75000	3.670000	3.822250	3.522500	1.299750	0.942750	0.66675	0.415750	0.559500	0.198250	0.144000
max	146.00000	3.760000	3.941000	3.618000	1.661000	1.503000	0.84800	0.631000	0.621000	0.248000	0.544000

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

Correlations:

Correlation demonstrates the relationship between any two variables in a dataset. Correlation measures the strength of a linear relationship between two variables. Using corr() to print the correlation of multiple characteristics

data.corr()

```
1 data.corr()
```

	RANK	Happiness score	Whisker-high	Whisker-low	Dystopia (1.83) + residual	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
RANK	1.000000	-0.980856	-0.978345	-0.982031	-0.444633	-0.792137	-0.773773	-0.750945	-0.623040	-0.032831	-0.403199
Happiness score	-0.980856	1.000000	0.999333	0.999383	0.498990	0.763677	0.777889	0.740260	0.624822	0.063785	0.416216
Whisker-high	-0.978345	0.999333	1.000000	0.997434	0.514099	0.752104	0.773375	0.727611	0.622934	0.065728	0.413718
Whisker-low	-0.982031	0.999383	0.997434	1.000000	0.483762	0.773844	0.781302	0.751530	0.625926	0.061820	0.418162
Dystopia (1.83) + residual	-0.444633	0.498990	0.514099	0.483762	1.000000	-0.073423	0.083606	-0.006886	0.117695	0.066915	-0.051075
GDP per capita	-0.792137	0.763677	0.752104	0.773844	-0.073423	1.000000	0.722421	0.815386	0.458591	-0.164472	0.377589
Social support	-0.773773	0.777889	0.773375	0.781302	0.083606	0.722421	1.000000	0.666760	0.480466	-0.002339	0.223352
Healthy life expectancy	-0.750945	0.740260	0.727611	0.751530	-0.006886	0.815386	0.666760	1.000000	0.433166	-0.098133	0.362626
Freedom to make life choices	-0.623040	0.624822	0.622934	0.625926	0.117695	0.458591	0.480466	0.433166	1.000000	0.176800	0.402474
Generosity	-0.032831	0.063785	0.065728	0.061820	0.066915	-0.164472	-0.002339	-0.098133	0.176800	1.000000	0.096107
Perceptions of corruption	-0.403199	0.416216	0.413718	0.418162	-0.051075	0.377589	0.223352	0.362626	0.402474	0.096107	1.000000

Fig. 10. Correlation of the whole dataset

data[['Country', 'Happiness score', 'GDP per capita', 'Social support', 'Healthy life expectancy', 'Perceptions of corruption']].corr()

corrdata

```
1 corrdata = data[['Country','Happiness score','GDP per capita','Social support',
2                  'Healthy life expectancy','Perceptions of corruption']].corr()
3 corrdata
```

	Happiness score	GDP per capita	Social support	Healthy life expectancy	Perceptions of corruption
Happiness score	1.000000	0.763677	0.777889	0.740260	0.416216
GDP per capita	0.763677	1.000000	0.722421	0.815386	0.377589
Social support	0.777889	0.722421	1.000000	0.666760	0.223352
Healthy life expectancy	0.740260	0.815386	0.666760	1.000000	0.362626
Perceptions of corruption	0.416216	0.377589	0.223352	0.362626	1.000000

Fig. 11. Correlation of some attributes in the dataset

E. Graphical Exploratory Data Analysis

Graphical exploratory data analysis is fundamentally the graphical version of non-graphical exploratory data analysis. EDA evaluates data sets to summarise their statistical properties by focusing on the same four major elements, such as measures of central tendency, measures of spread, distribution shape, and the presence of outliers. We also classified GEDA into three types: univariate, bivariate, and multivariate. In the coming paragraphs and GEDA features [5,] we will go over these significant types in further depth. To begin, a subset of the data frame is selected for analysis or visualisation.

```
subdata = data[['Country','Happiness score','GDP per capita','Social support',
                'Healthy life expectancy','Perceptions of corruption']]
subdata
```

	Country	Happiness score	GDP per capita	Social support	Healthy life expectancy	Perceptions of corruption
0	Finland	7.821	1.892	1.258	0.775	0.534
1	Denmark	7.636	1.953	1.243	0.777	0.532
2	Iceland	7.557	1.936	1.320	0.803	0.191
3	Switzerland	7.512	2.026	1.226	0.822	0.461
4	Netherlands	7.415	1.945	1.206	0.787	0.419
...	...	...	...	...	...	...
141	Botswana*	3.471	1.503	0.815	0.280	0.102
142	Rwanda*	3.268	0.785	0.133	0.462	0.544
143	Zimbabwe	2.995	0.947	0.690	0.270	0.105
144	Lebanon	2.955	1.392	0.498	0.631	0.034
145	Afghanistan	2.404	0.758	0.000	0.289	0.005

146 rows × 6 columns

Fig. 12. A subset of the dataframe

F. Univariate Graphical Exploratory Data Analysis

1. Histogram:

A histogram is a data representation that resembles a bar graph and groups various outcomes into columns along the x-axis. The y-axis can be used to depict numerical counts or percentages of occurrences in each column to demonstrate data distributions. Matplotlib.pyplot.hist() can be used to create a histogram in Python.

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

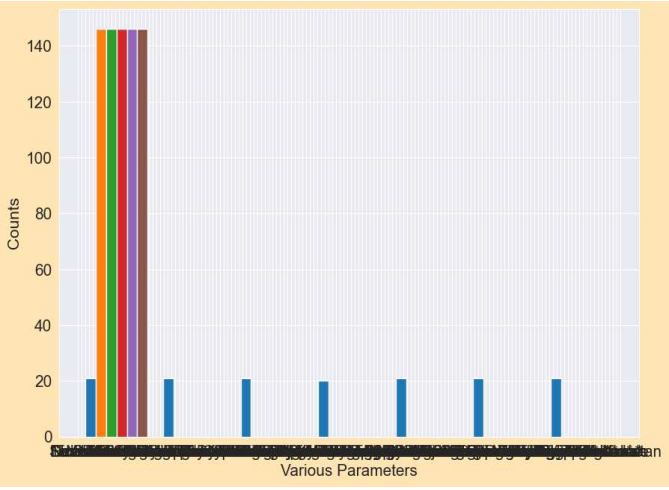

Chart. 1. Histogram

2. Stem Plot:

A stem plot establishes a marker at each x-point and draws vertical lines from the baseline to the y-axis. The x-positions are not required. The formats can be supplied as keyword or positional parameters. Matplotlib.pyplot.stem can be used to create a stem plot in Python ()

```
plt.stem(data['Happiness score'])
```

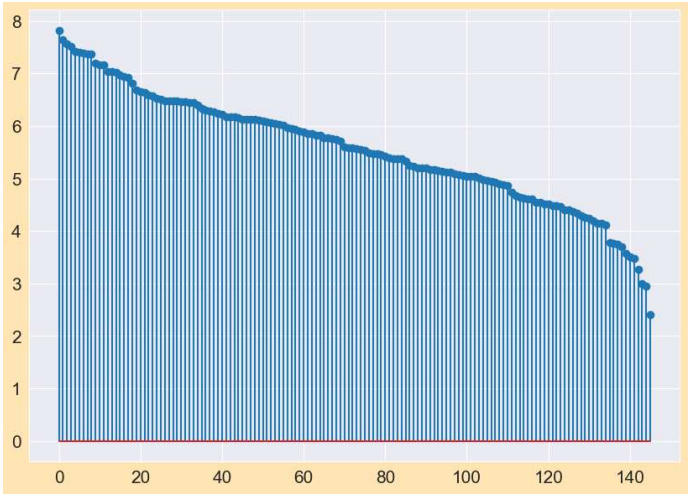

Chart. 2. Stem plot

3. Box Plot:

A box plot is a visual depiction and comparison of data groupings. The box plot illustrates the level, spread, and symmetry of a data distribution by employing the median, approximate quartiles, outliers, and the smallest and largest data points (extreme values).

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

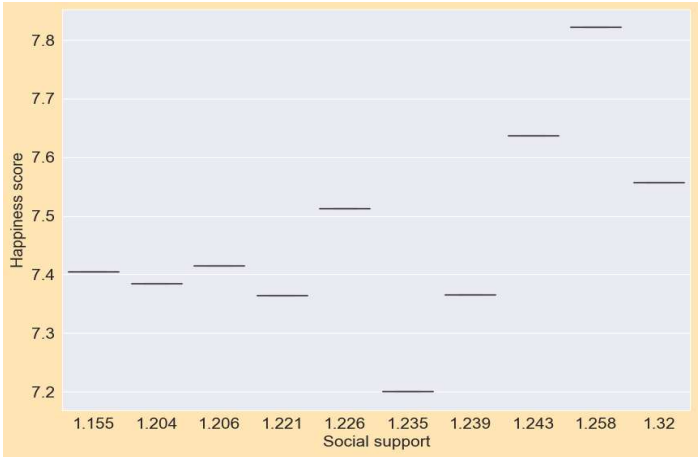

Chart. 3. Boxplot

Multivariate Graphical Exploratory Data Analysis

1. Scatter plot:

In a scatter plot, dots represent the values of two separate numerical variables. The positions of each dot on the horizontal and vertical axes represent the values for each data point. Scatter plots are used to show how variables relate to one another. The scatter plot of "Happiness score" vs "GDP per Capita" is shown below-

```
plt.scatter(x="Happiness score",y="GDP per capita",
            data = subdata,marker="*",color="Red")
```

Chart. 4. Scatter Plot

• Heat Maps:

A heatmap is a graphical representation of data that employs colour coding to indicate different values. It is a two-dimensional table of colour tones. This graphing approach is widely used in biology to display gene expression and other multivariate data.

```
plt.figure(figsize=[13, 5])
sns.heatmap(corrdata,annot=True)
```

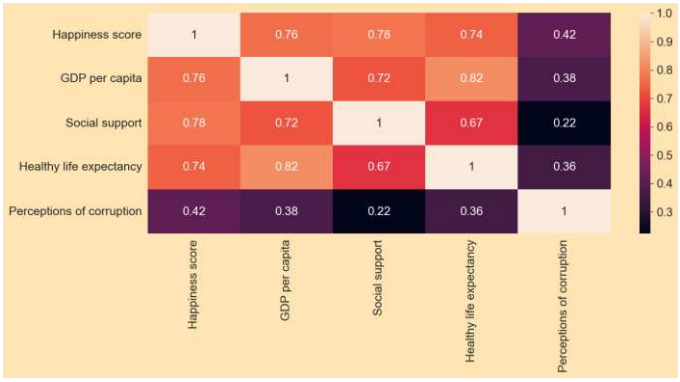

Chart. 5. Heatmap

2. Count Plot:

A Seaborn count plot is a graphical depiction of the number of occurrences or frequency for each category of data, shown by bars. The countplot() function is used to generate bars that represent the number of observations in each categorical category. The sub-data data frame is shown using the Count plot.

```
plt.figure(figsize=[15,5])
sns.countplot
```

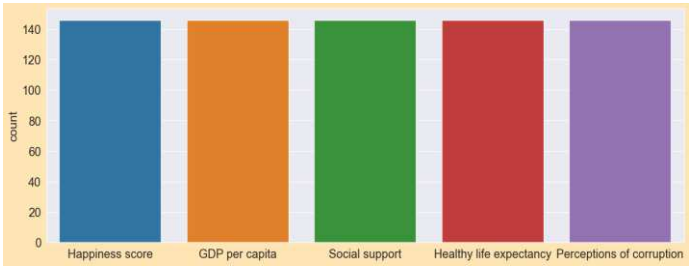

Chart. 6. Count plot

3. CONCLUSIONS

The many processes of data analysis, including data collection, cleaning, and analysis, are addressed briefly in this work. The primary focus of this course is exploratory data analysis. Python programming language is utilised for implementation. Jupyter notebook is used for in-depth study. Various Python libraries and packages are discussed. Numerous findings are gathered using various analytical and visualization approaches. The dataset "World Happiness Record 2022" is used to extract important information such as the difference in happiness scores of different countries, the dependence of one attribute in building up the score, how a variable affects another variable, and so on. Various graphs have been plotted using various attributes in the dataset to draw conclusions in an easy way. The primary focus of this course is exploratory data analysis. Python programming language is utilised for implementation. Jupyter notebook is used for in-depth study. Various Python libraries and packages are discussed. Numerous findings are gathered using various analytical and visualization approaches. The dataset "World Happiness Record 2022" is used to extract important information such as the difference in happiness scores of different countries, the dependence of one attribute in building up the score, how a variable affects another variable, and so on. Various graphs have been plotted using various attributes in the dataset to draw conclusions in an easy way.

REFERENCES

[1] Wolfram, S.: Mathematica: A System for Doing Mathematics by Computer. Addi-son Wesley Longman Publishing Co., Inc., Boston (1991)

[2] Mauriciusa Munhoz de Medeiros, Norberto Hoppen, Antonio Carlosa Gastaud Maçada, Data science for business: benefits, challenges and opportunities, Bottom Line (ISSN: 0888045X) 33 (2020) 149–163, http://dx.doi.org/10.1108/BL-12- 2019-0132/FULL/XML.

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

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

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

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

[7] https://stackoverflow.com

[8] https://github.com

[9] KD Nuggets poll result https://www.kdnuggets.com/