A

Course End Project Report on

WORLD HAPPINESS REPORT

Is submitted in partial fulfillment of the Requirements for the Award of CIE of

DATA ANALYSIS AND VISUALIZATION- 22ADE01

in

B.E, IV-SEM, INFORMATION TECHNOLOGY

Submitted by

MOHAMMED ABDUL RAFE SAJID,
160123737051,
IT-1

COURSE TAUGHT BY:

Dr Ramakrishna Okikiolu Professor, Dept of IT.

DEPARTMENT OF INFORMATION TECHNOLOGY

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY(A)

(Affiliated to Osmania University; Accredited by NBA,NAAC,ISO)

kokapet(V),GANDIPET(M),HYDERABAD- 500075

Website:www.cbit.ac.in 2024-2025

CERTIFICATE

This is to certify that the course-end project work entitled **“WORLD HAPPINESS REPORT”** has been submitted by **Mohammed Abdul Rafe Sajid (160123737051)** in partial fulfillment of the requirements for the award of **CIE Marks** for **DATA ANALYSIS AND VISUALIZATION (22ADE01)** as part of the **B.E., IV-Semester, Information Technology** program at **Chaitanya Bharathi Institute of Technology (A)**, affiliated with **Osmania University, Hyderabad**.

This report is a **bona fide** record of the work carried out by the candidate under my supervision and guidance. The results presented in this report have not been submitted to any other university or institute for the award of any other degree or diploma.

Signature of Course Faculty
Dr Ramakrishna Kolikipogu
Professor of IT

Kokapet(V),Gandipet(M),Ranga Reddy (Dist.)–500075, Hyderabad, T.S.

Acknowledgement

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr Ramakrishna Kolikipogu, Professor of IT** for his able guidance and useful suggestions, which helped us in completing the Course End Project in time.

We are particularly thankful to **HoD, Principal and Management**, for their support and encouragement, which helped us to mould our project into a successful one.

We also thank all the staff members of IT Department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

Mohammed abdul Rafe Sajid,
160123737051

Abstract

Happiness is a crucial indicator of well-being, influenced by multiple economic, social, and political factors. This project analyzes the World Happiness Report dataset, sourced from Kaggle, to identify key determinants of happiness scores across various countries. The dataset includes metrics such as GDP per capita, social support, life expectancy, freedom, government trust, and generosity, which contribute to a country's overall happiness ranking.

The project follows a structured data analysis pipeline, starting with data cleaning and preprocessing, followed by exploratory data analysis (EDA) to uncover trends and correlations. Various data visualization techniques using Matplotlib and Seaborn help in understanding the distribution of happiness scores and the impact of different factors. Statistical analysis is also performed to determine the most significant contributors to happiness.

Findings suggest that economic stability, strong social support systems, and low corruption levels significantly influence happiness scores. Countries with higher GDP per capita and better healthcare systems tend to rank higher in happiness. This project provides valuable insights into how governments and policymakers can enhance societal well-being by focusing on these key factors.

The analysis is conducted in Google Colab, and the final project is documented in a GitHub repository, making it accessible for further research and improvements.

.

Table of Contents

Title **Page No.**

Abbreviations

Abbreviation	**Description**
DAV	Data Analysis and Visualization
SB	Sea Born
PD	pandas
CSV	Comma Separated Value
HIST	Histogram

CHAPTER 1
Introduction

1.1 Definition of Problem

The **World Happiness Report** provides insights into global well-being by ranking countries based on various socioeconomic indicators. This project aims to analyze the dataset to understand the key factors influencing happiness scores across different nations. The study will explore correlations between GDP, social support, life expectancy, and other metrics to determine their impact on national happiness rankings. Additionally, data visualization techniques will be employed to illustrate trends, disparities, and significant patterns across regions. The ultimate goal is to derive meaningful conclusions that can inform policymakers on enhancing societal well-being.

1.1 Objectives and Outcomes

Objectives:

- **Data Exploration**: Conduct a thorough analysis of the dataset to understand its structure, features, and key attributes.
- **Insight Generation**: Identify correlations between happiness scores and factors such as GDP, government trust, and life expectancy.
- **Visualization**: Use Seaborn, Matplotlib, and Pandas to create impactful data visualizations for clear insight presentation.
- **Predictive Analysis**: If applicable, develop models to predict happiness scores based on key socioeconomic indicators.
- **Policy Recommendations**: Provide data-driven insights that could help in shaping policies to improve well-being.

Outcomes:

- **Comprehensive Dataset Understanding**: Gain a structured overview of the dataset, recognizing trends and potential biases.
- **Insightful Analysis:** Extract key findings on the most influential factors affecting global happiness.
- **Effective Communication**: Present results using compelling visualizations that simplify complex data

insights.

- **Predictive Models (**if applicable): Develop reliable models to estimate happiness scores based on historical data.
- **Informed Decision-Making**: Equip policymakers, researchers, and organizations with actionable insights to enhance societal well-being.

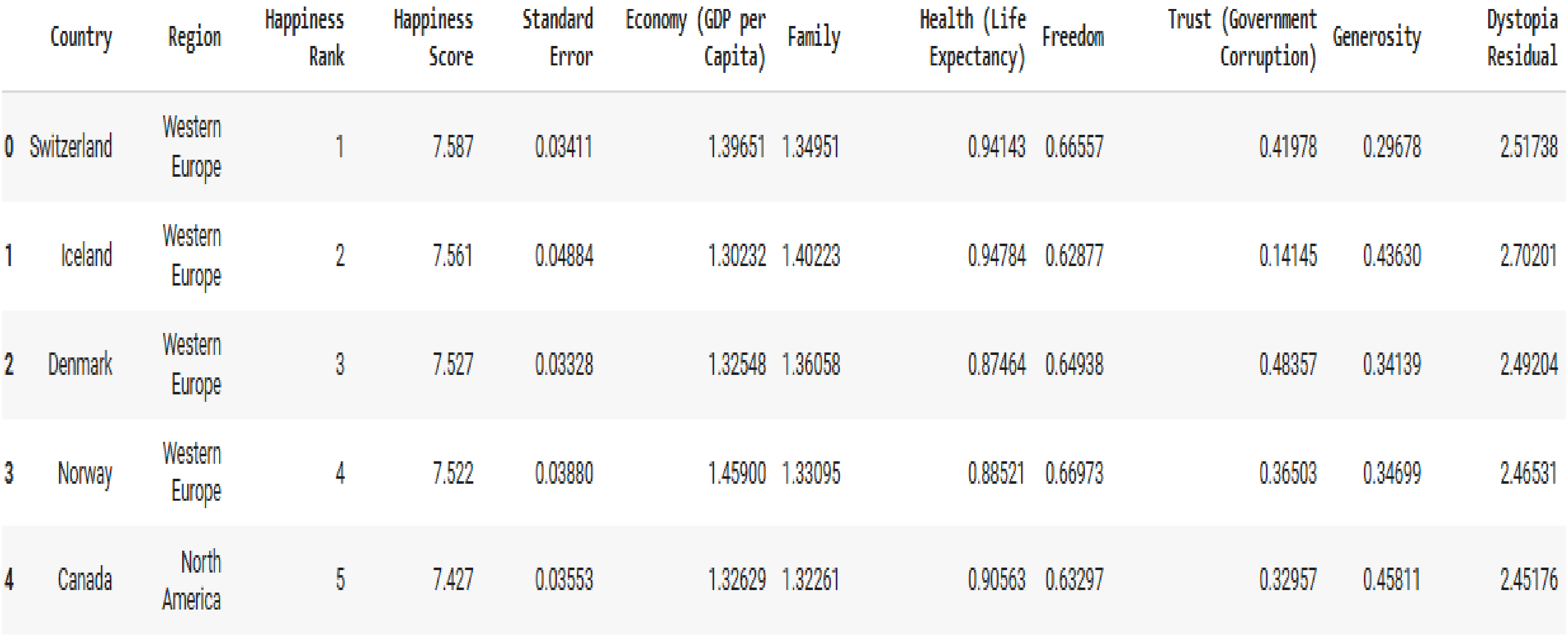
CHAPTER 2
Methodology

3.1 Data collection and Dataset description

The dataset used in this study is sourced from Kaggle and provides crucial insights into global happiness rankings. It contains detailed information on various socioeconomic factors that contribute to the well-being of nations. Each record represents a country's happiness score, calculated based on multiple indicators such as GDP per capita, social support, life expectancy, freedom, trust in government, generosity, and dystopia residuals.

Additionally, the dataset categorizes countries by region, enabling comparative analysis across different parts of the world. It includes numerical values for each contributing factor, allowing for statistical evaluation and visualization of patterns. By analyzing these factors, we aim to uncover significant trends, correlations, and disparities in happiness levels across nations.

The dataset serves as a foundation for exploratory data analysis (EDA), visualization, and potential predictive modeling to understand how different elements impact happiness scores. Through this project, we seek to provide actionable insights that can inform policies aimed at improving global well-being.



	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176

Figure 3.1: Dataset

3.2 Data cleaning and preprocessing

Data cleaning and preprocessing are essential steps in ensuring that our dataset is accurate, consistent, and free from errors. This phase enhances the dataset's reliability and prepares it for meaningful analysis.

Handling Unnecessary Columns

The first step was removing irrelevant columns that do not contribute significantly to the analysis. We examined the dataset and eliminated any redundant features using Pandas' drop() function. This streamlined our dataset, making it more efficient for analysis.

Managing Missing Values

Missing data can lead to biased or inaccurate results. We identified missing values using Pandas' isnull() function and applied different strategies for handling them. If a column had a small percentage of missing values, we filled them using the mean or mode of the respective column via fillna(). However, if a column had a substantial amount of missing data and was not crucial for analysis, we dropped it using dropna().

Removing Duplicate Entries

To ensure data integrity, we checked for duplicate rows using Pandas' duplicated() function. Any identified duplicates were removed using drop_duplicates(), ensuring that each entry was unique and did not distort our analysis.

Correcting Data Types

Data types were reviewed using Pandas' dtypes attribute to ensure numerical and categorical data were in the correct format. Columns containing Happiness Scores and GDP per Capita, initially stored as strings, were converted into numerical formats using astype(). This step ensured accurate computations and visualizations.

Handling Outliers

Outliers can significantly impact statistical analysis. We identified outliers using methods such as boxplots and Z-score analysis. Depending on their influence, we either removed extreme outliers or capped them within a reasonable range using Pandas and NumPy functions.

Addressing Zero Values

Certain columns, such as GDP per Capita or Life Expectancy, should not contain zero values. We replaced them with the column mean or median using replace(), ensuring logical consistency in the dataset.

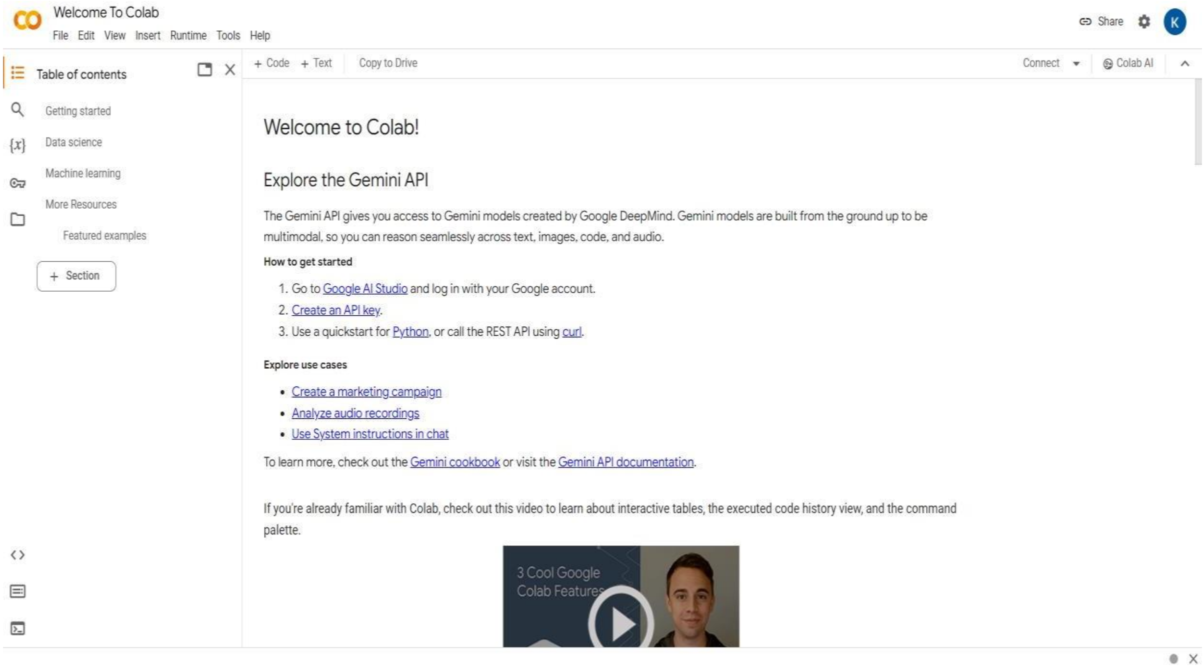
By following this structured approach, we ensured that our dataset was clean, reliable, and well-prepared for exploratory data analysis (EDA). This step provided a strong foundation for visualizing patterns and uncovering meaningful insights into global happiness trends.

CHAPTER 4

System Architecture and Implementation

4.1 Google Colab

Google Colaboratory, commonly known as Google Colab, is a free online cloud-based Jupyter notebook environment tailored for training machine learn- ing and deep learning models. This article explores the functionalities, benefits, and features of Google Colab, elucidating its significance in the realm of data science and machine learning.



Welcome to Colab!

Explore the Gemini API

The Gemini API gives you access to Gemini models created by Google DeepMind. Gemini models are built from the ground up to be multimodal, so you can reason seamlessly across text, images, code, and audio.

How to get started

1. Go to Google AI Studio and log in with your Google account.
2. Create an API key.
3. Use a quickstart for Python, or call the REST API using curl.

Explore use cases

- Create a marketing campaign
- Analyze audio recordings
- Use System instructions in chat

To learn more, check out the Gemini cookbook or visit the Gemini API documentation.

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.

Figure 4.1: Google Colab

4.1.1 What is Google Colab?

Google Colab offers a cloud-based environment accessible via any web browser, eliminating the need for local software installation. Users can leverage its computing resources, including CPUs, GPUs, and TPUs, facilitating efficient model training and execution.

4.2 Benefits of Google Colab

Accessibility: Users can access Google Colab from any location with internet connectivity, streamlining collaboration and workflow.
Power: The platform provides access to potent computing resources like GPUs and TPUs, enabling swift and effective model training.
Collaboration: Google Colab simplifies collaborative efforts by allowing real-time editing and sharing of notebooks among team members.

Education: It serves as an invaluable educational tool for learning about machine learning and data science, offering a plethora of tutorials and resources.

4.2.1 Why Choose Google Colab?

Google Colab stands out as an ideal choice for students, data scientists, researchers, and enthusiasts due to its:
Ease of Use: With no setup requirements, users can swiftly start coding after creating an account.
Affordability: The platform is largely free to use, with paid plans available for more demanding tasks.
Flexibility: Users can seamlessly train models, process data, create visualizations, and collaborate with others, making it a versatile tool for various applications.

4.2.2 Notebook in Google Colab

In Google Colab, a notebook serves as a web-based environment for code creation and execution. Notebooks offer several advantages, including real-time code execution and visualization, support for markdown for documentation, and collaboration features, making them indispensable for data scientists and machine learning practitioners.

4.2.3 Google Colab Features

Google Colab boasts several features that enhance its usability and effec- tiveness:

Free Access to GPUs and TPUs: Users can leverage powerful computing resources without any additional cost.

Web-based Interface: The intuitive and user-friendly interface eliminates the need for local software installation.

Collaboration Tools: Multiple users can collaborate on the same notebook simultaneously, streamlining teamwork.

Markdown Support: Notebooks support markdown, enabling users to include formatted text, equations, and images alongside their code.

Pre-installed Libraries: Google Colab comes pre-installed with popular libraries and tools for machine learning and deep learning, such as TensorFlow and PyTorch, saving time on setup and configuration.

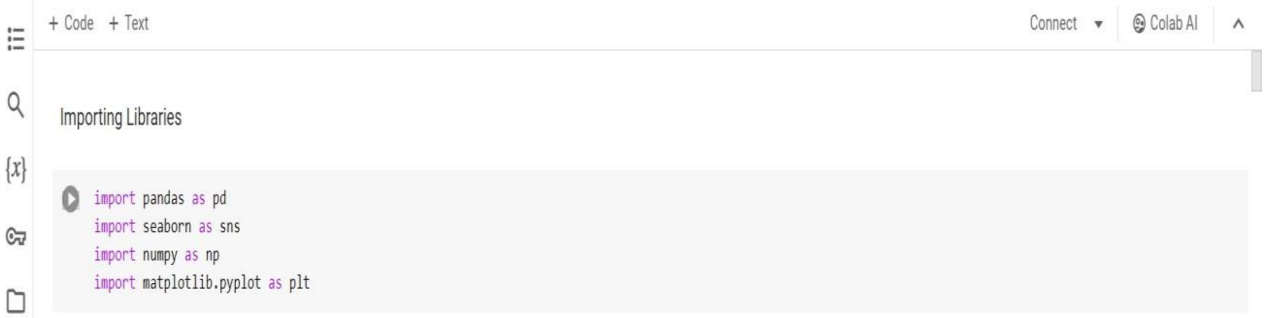
Google Colab emerges as a versatile and indispensable tool for machine learning and data science tasks, offering accessibility, power, and flexibility. Its user-friendly interface, collaborative features, and integration with powerful computing resources make it an invaluable asset for individuals and teams alike, driving innovation and progress in the field of machine learning and beyond.

4.3 Code Snippets

4.3.1 Importing libraries and Data loading

To begin our project, we first import the necessary libraries for data analysis and visualization. We import pandas as pd and numpy as np for data handling and numerical operations, respectively. We use Matplotlib and Seaborn for data visualization.

Next, we load the dataset into our code using the read csv function from pandas, assuming the dataset is stored in a CSV file named '15.csv' which actually contains they entire data of world happiness report. We assign the loaded dataset to a variable named 'df'.



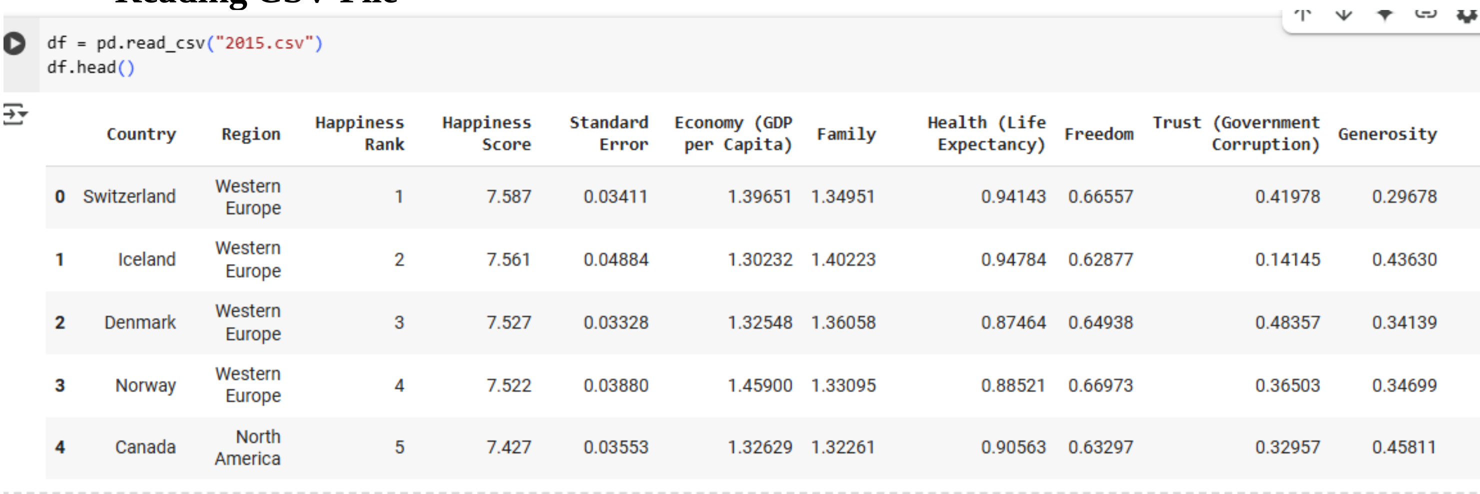
Importing Libraries

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

Figure 4.2: importing libraries and Dataset loading

To ensure that the dataset has been loaded successfully, we display the first few rows of the dataset using the head() function. This allows us to inspect the structure and content of the dataset, confirming that it has been imported correctly and is ready for further processing.

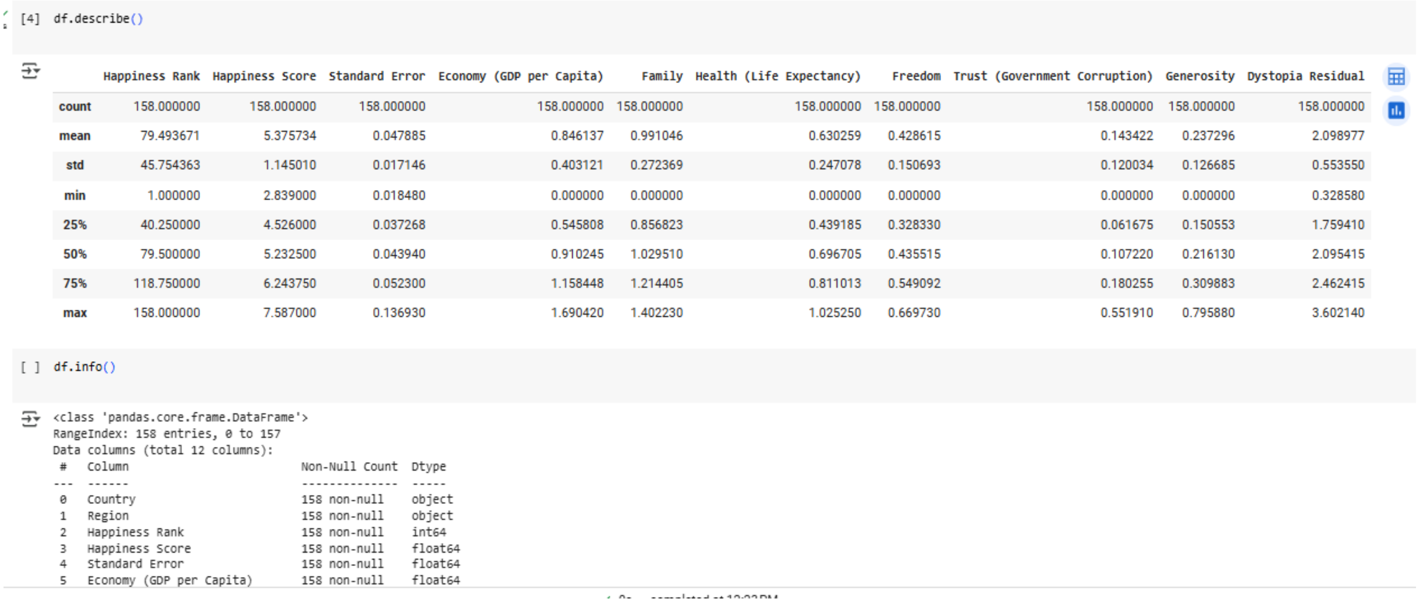
Reading CSV File



```
df = pd.read_csv("2015.csv")
df.head()
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811

Figure 4.4: Reading CSV



```
[4] df.describe()
```

	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
count	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000	158.000000
mean	79.493671	5.375734	0.047885	0.846137	0.991046	0.630259	0.428615	0.143422	0.237296	2.098977
std	45.754363	1.145010	0.017146	0.403121	0.272369	0.247078	0.150693	0.120034	0.126685	0.553550
min	1.000000	2.839000	0.018480	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.328580
25%	40.250000	4.526000	0.037268	0.545808	0.856823	0.439185	0.328330	0.061675	0.150553	1.759410
50%	79.500000	5.232500	0.043940	0.910245	1.029510	0.696705	0.435515	0.107220	0.216130	2.095415
75%	118.750000	6.243750	0.052300	1.158448	1.214405	0.811013	0.549092	0.180255	0.309883	2.462415
max	158.000000	7.587000	0.136930	1.690420	1.402230	1.025250	0.669730	0.551910	0.795880	3.602140

```
[ ] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158 entries, 0 to 157
Data columns (total 12 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Country                        158 non-null    object
 1   Region                         158 non-null    object
 2   Happiness Rank                 158 non-null    int64
 3   Happiness Score                158 non-null    float64
 4   Standard Error                 158 non-null    float64
 5   Economy (GDP per Capita)       158 non-null    float64
```

Figure 4.5: Statistics for Numerical features

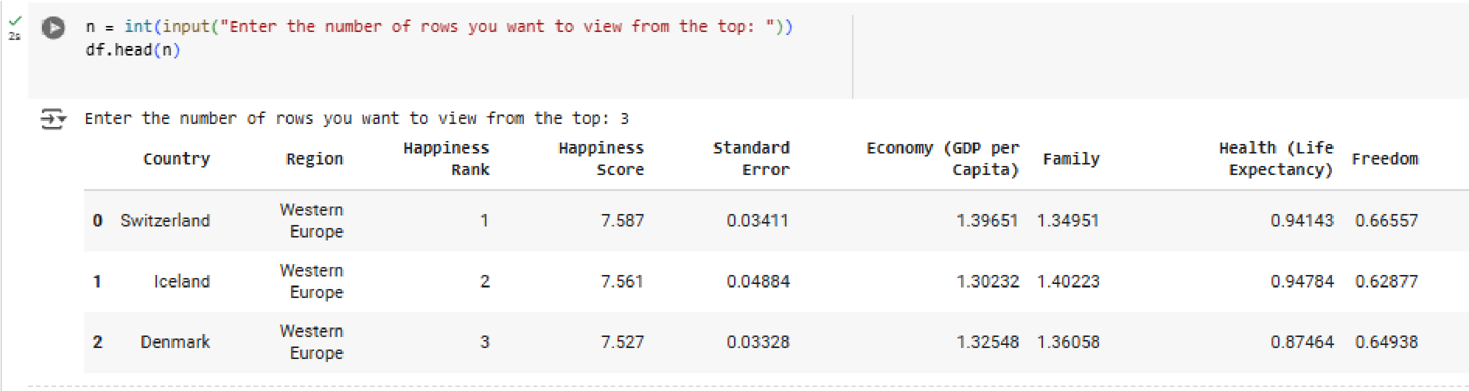
This code snippet prints out summary statistics for numerical features in a DataFrame (df). It provides key statistical measures such as count, mean, standard deviation, minimum, maximum, and quartile values for each numerical column in the DataFrame.

4.3.2 Data cleaning and preprocessing

The data cleaning process for our dataset was carried out in a structured manner to ensure quality and reliability for subsequent analysis

1. Viewing Data

Users can specify the number of rows to view from the top of the DataFrame. This feature helps in getting a quick look at the data's initial entries.



```
n = int(input("Enter the number of rows you want to view from the top: "))
df.head(n)
```

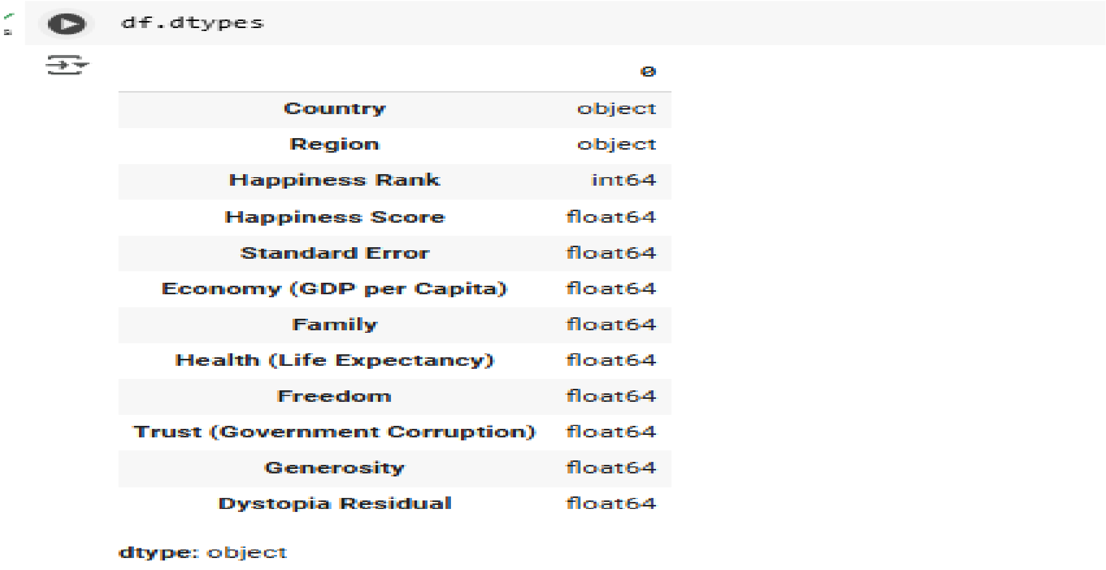
```
Enter the number of rows you want to view from the top: 3
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938

Figure 4.6: Viewing Data

2. Addressing Data Types:

We reviewed the data types using Pandas' dtypes attribute.



```
df.dtypes
```

	0
Country	object
Region	object
Happiness Rank	int64
Happiness Score	float64
Standard Error	float64
Economy (GDP per Capita)	float64
Family	float64
Health (Life Expectancy)	float64
Freedom	float64
Trust (Government Corruption)	float64
Generosity	float64
Dystopia Residual	float64

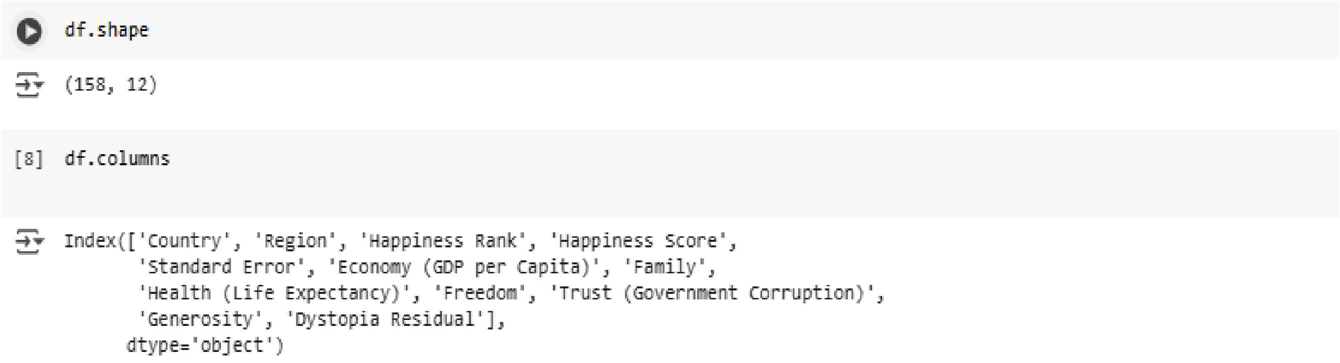
```
dtype: object
```

Figure 4.7:Addressing Data Types

5. Basic Data Exploration

The script provided insights into the structure of the DataFrame:

- **Shape**: Number of rows and columns.
- **Columns**: Names of the columns.
- **Index**: Indexes of the DataFrame.
- **Data Types**: Types of data in each column.
- **Missing Values**: Count of missing values in each column.

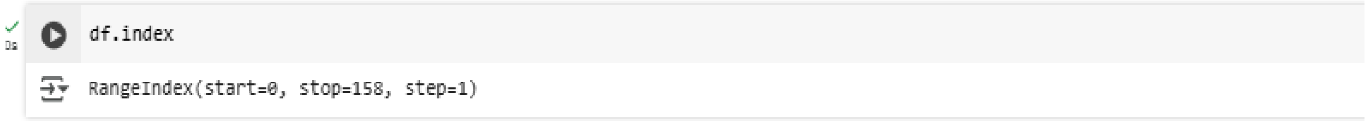


```
df.shape
```

```
(158, 12)
```

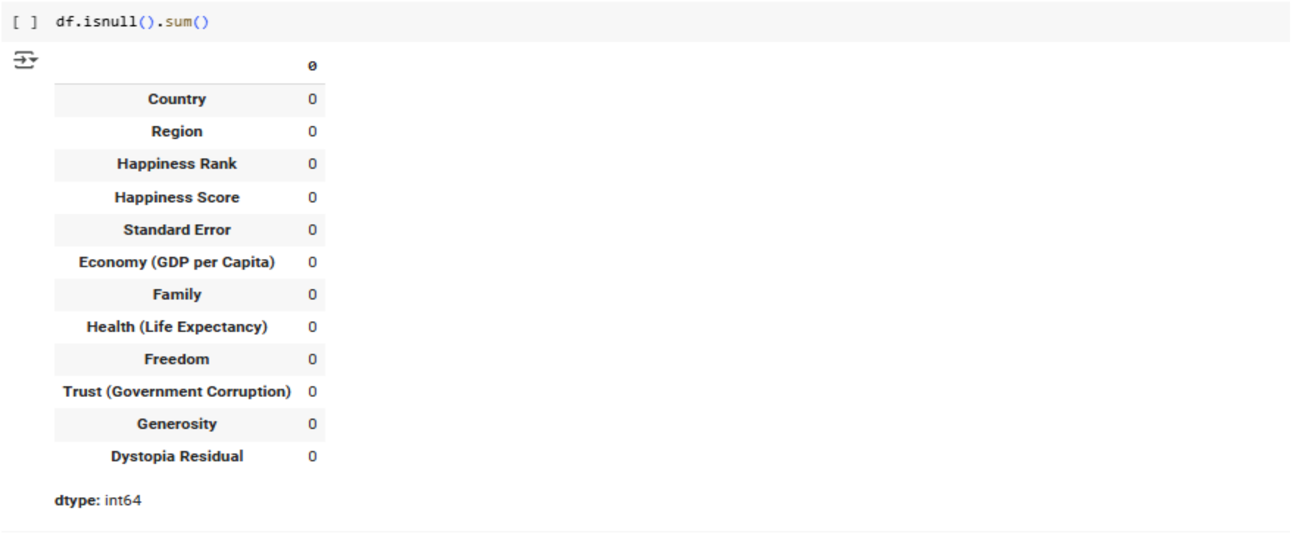
```
[8] df.columns
```

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



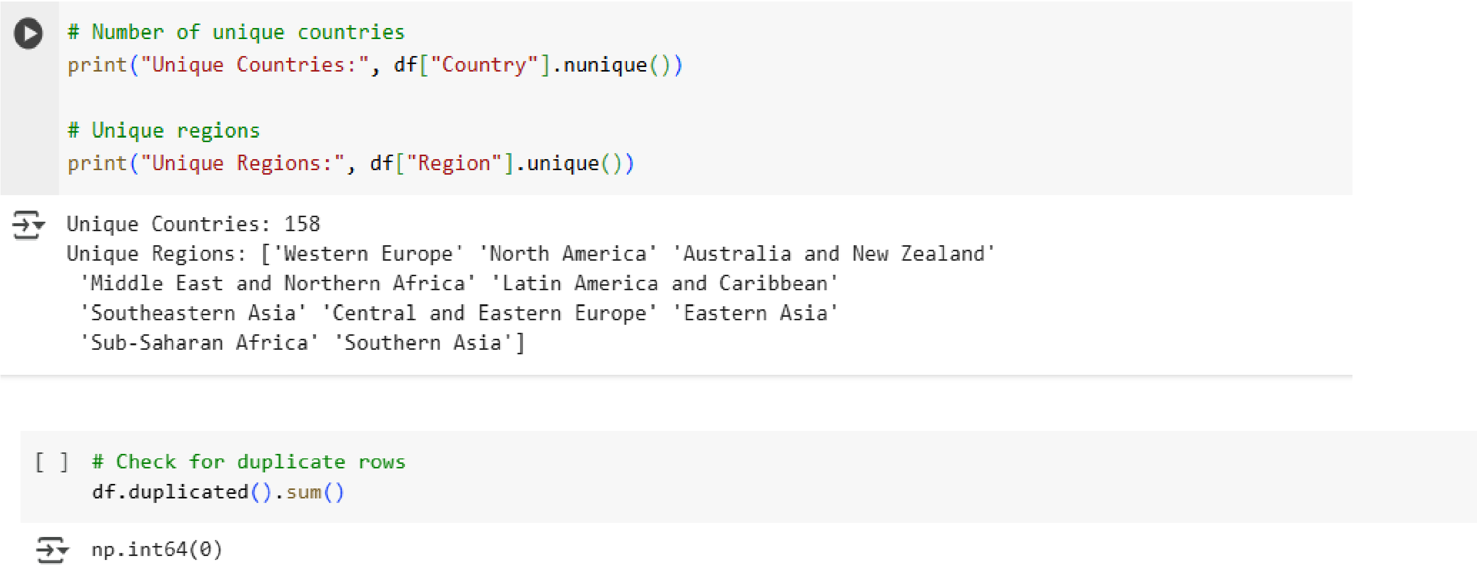
```
df.index
```

```
RangeIndex(start=0, stop=158, step=1)
```



```
[ ] df.isnull().sum()
```

```
                                0
                      Country   0
                       Region   0
               Happiness Rank   0
              Happiness Score   0
               Standard Error   0
     Economy (GDP per Capita)   0
                       Family   0
     Health (Life Expectancy)   0
                      Freedom   0
Trust (Government Corruption)   0
                   Generosity   0
            Dystopia Residual   0

dtype: int64
```



```
# Number of unique countries
print("Unique Countries:", df["Country"].nunique())

# Unique regions
print("Unique Regions:", df["Region"].unique())
```

```
Unique Countries: 158
Unique Regions: ['Western Europe' 'North America' 'Australia and New Zealand'
 'Middle East and Northern Africa' 'Latin America and Caribbean'
 'Southeastern Asia' 'Central and Eastern Europe' 'Eastern Asia'
 'Sub-Saharan Africa' 'Southern Asia']
```

```
# Check for duplicate rows
df.duplicated().sum()
```

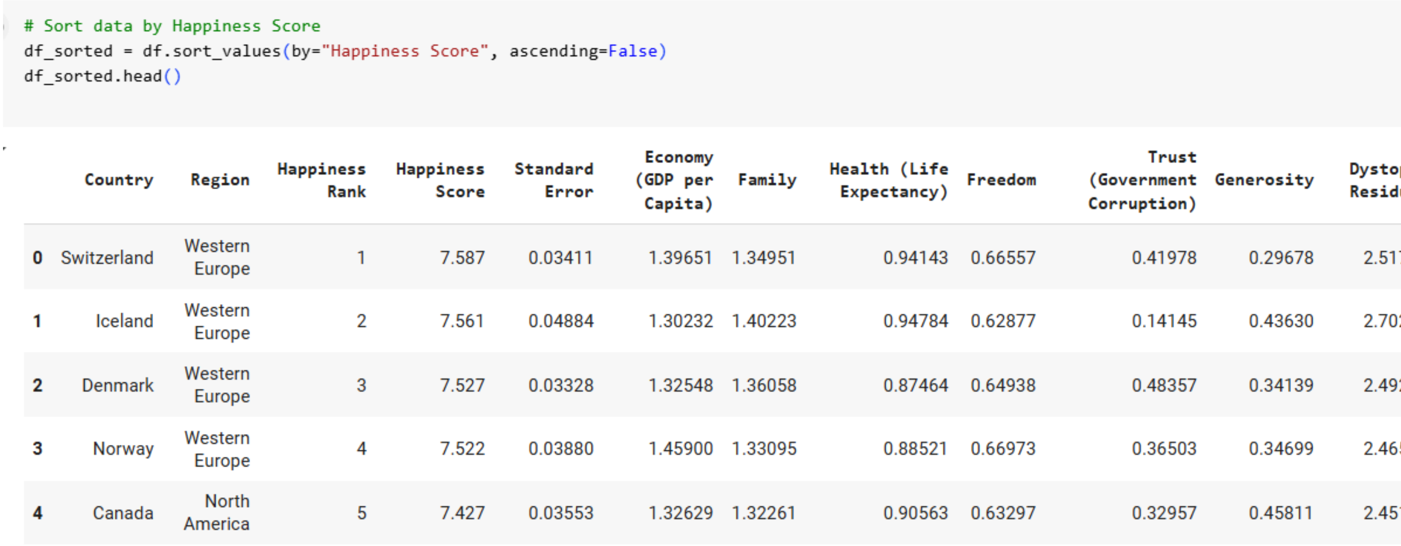
```
np.int64(0)
```

Figure 4.8: Basic Data Exploration

Data Manipulation

Sorting

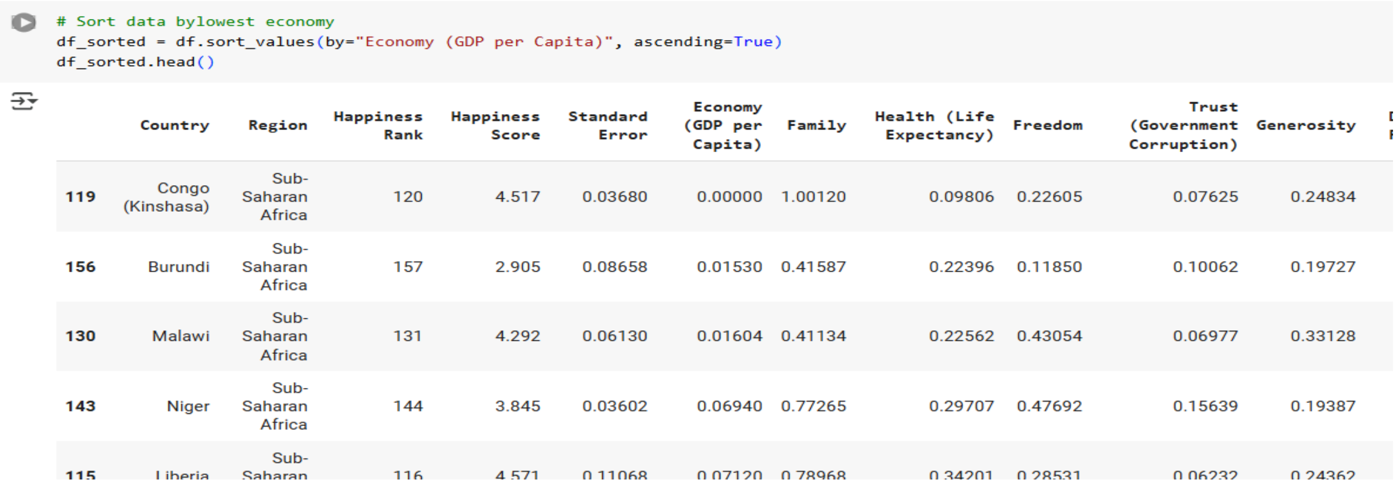
- The DataFrame was sorted by the happiness score in the descending order and found that Switzerland is the happiest country, followed by Iceland and Norway.



```
# Sort data by Happiness Score
df_sorted = df.sort_values(by="Happiness Score", ascending=False)
df_sorted.head()
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dysto Resid
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45

- Also sorted by lowest Economy (GDP per Capita)



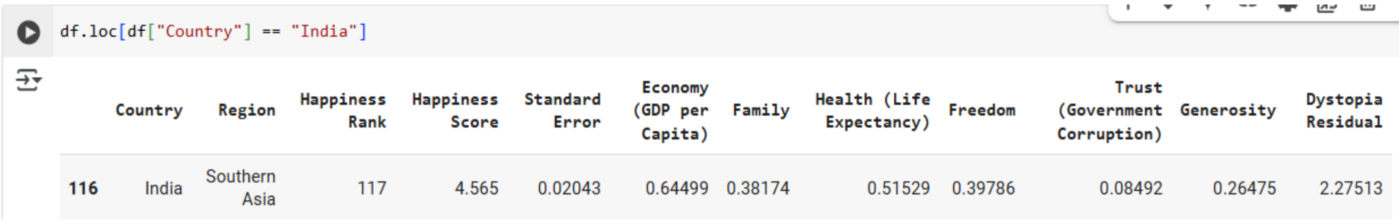
```
# Sort data bylowest economy
df_sorted = df.sort_values(by="Economy (GDP per Capita)", ascending=True)
df_sorted.head()
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity
119	Congo (Kinshasa)	Sub-Saharan Africa	120	4.517	0.03680	0.00000	1.00120	0.09806	0.22605	0.07625	0.24834
156	Burundi	Sub-Saharan Africa	157	2.905	0.08658	0.01530	0.41587	0.22396	0.11850	0.10062	0.19727
130	Malawi	Sub-Saharan Africa	131	4.292	0.06130	0.01604	0.41134	0.22562	0.43054	0.06977	0.33128
143	Niger	Sub-Saharan Africa	144	3.845	0.03602	0.06940	0.77265	0.29707	0.47692	0.15639	0.19387
115	Liberia	Sub-Saharan	116	4.571	0.11068	0.07120	0.78968	0.34201	0.28531	0.06232	0.24362

Figure 4.10: Sorting

Filtering

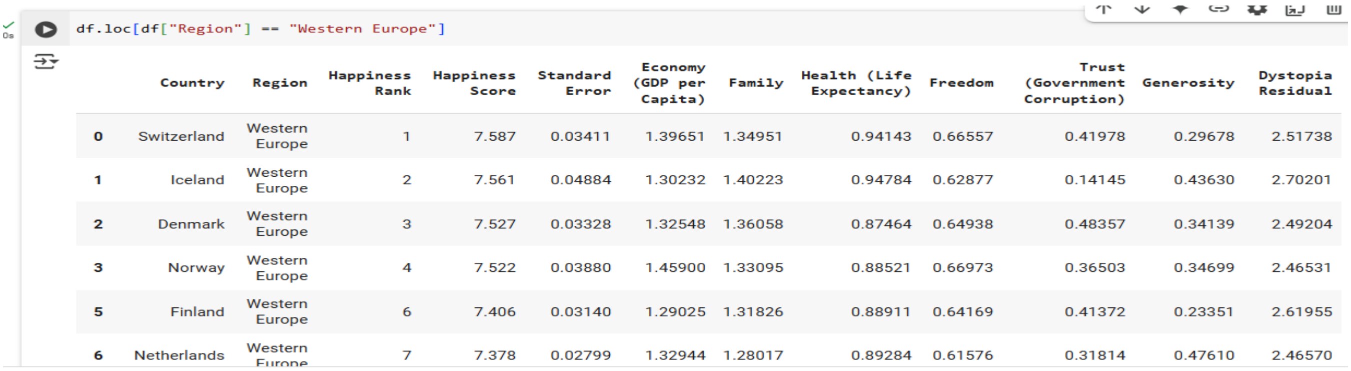
- Filter by a Specific Country



```
df.loc[df["Country"] == "India"]
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
116	India	Southern Asia	117	4.565	0.02043	0.64499	0.38174	0.51529	0.39786	0.08492	0.26475	2.27513

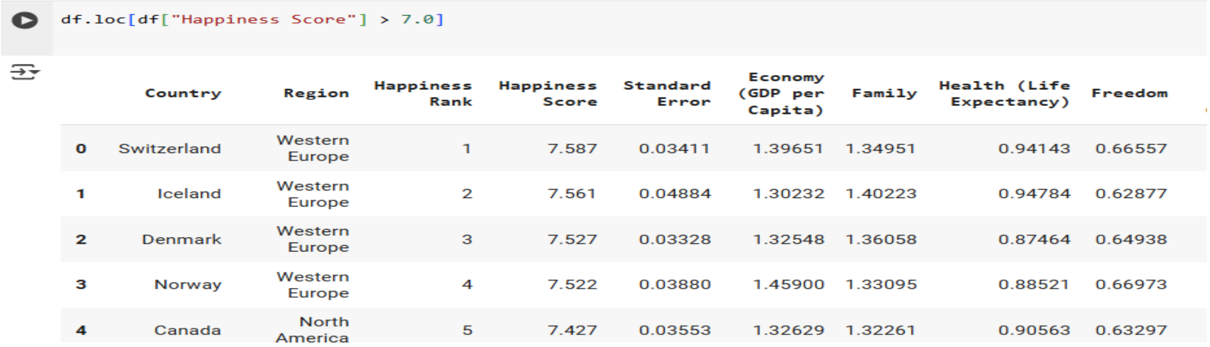
- Filter by Region



```
df.loc[df["Region"] == "Western Europe"]
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531
5	Finland	Western Europe	6	7.406	0.03140	1.29025	1.31826	0.88911	0.64169	0.41372	0.23351	2.61955
6	Netherlands	Western Europe	7	7.378	0.02799	1.32944	1.28017	0.89284	0.61576	0.31814	0.47610	2.46570

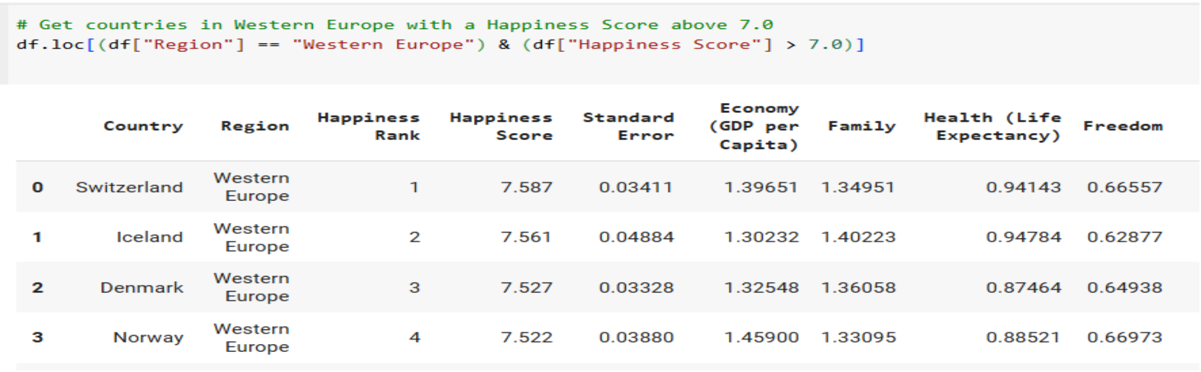
- Filter Countries with a Happiness Score Above a Certain Threshold



```
df.loc[df["Happiness Score"] > 7.0]
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973
4	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297

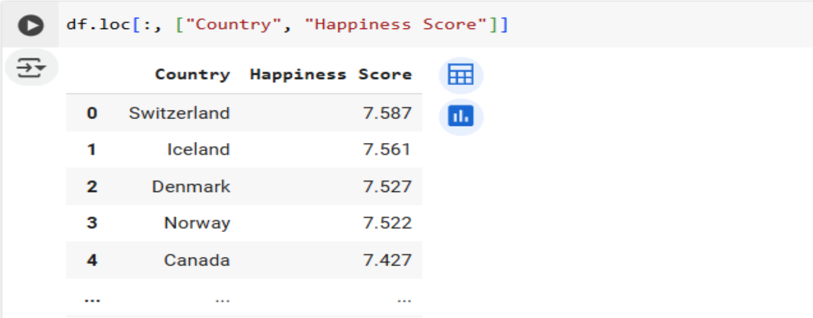
- Filter by Multiple Conditions



```
# Get countries in Western Europe with a Happiness Score above 7.0
df.loc[(df["Region"] == "Western Europe") & (df["Happiness Score"] > 7.0)]
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938
3	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973

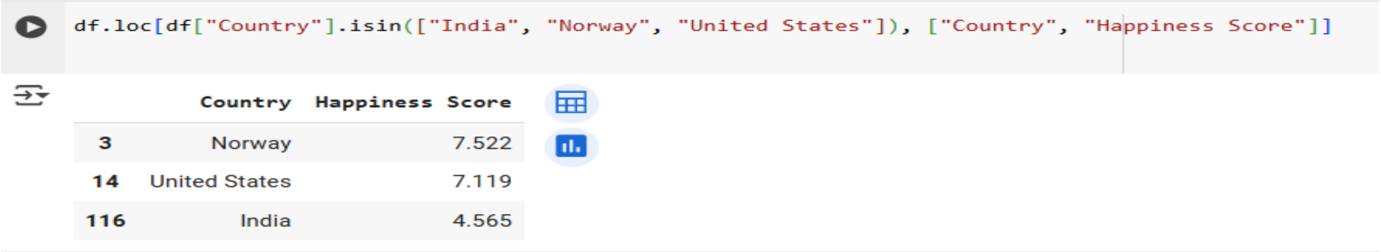
- Selecting Specific Columns with loc



```
df.loc[:, ["Country", "Happiness Score"]]
```

	Country	Happiness Score
0	Switzerland	7.587
1	Iceland	7.561
2	Denmark	7.527
3	Norway	7.522
4	Canada	7.427
...	...	...

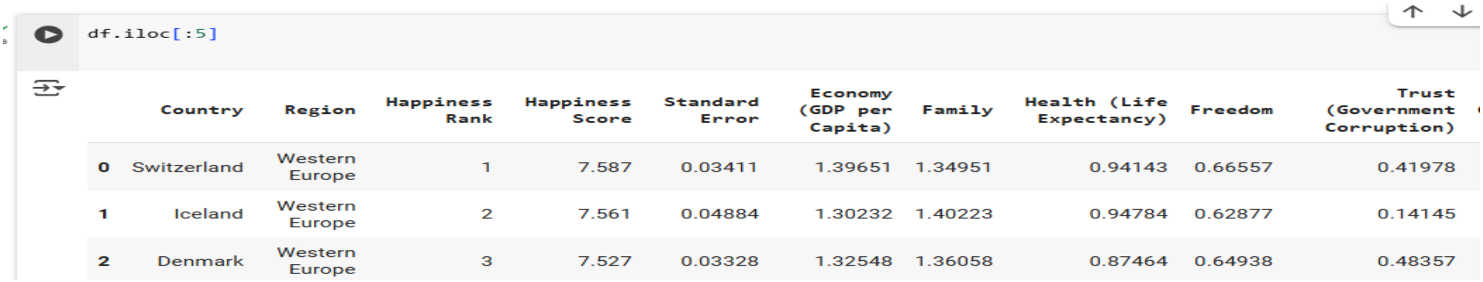
- Get specific countries with only their Happiness Score



```
df.loc[df["Country"].isin(["India", "Norway", "United States"]), ["Country", "Happiness Score"]]
```

	Country	Happiness Score
3	Norway	7.522
14	United States	7.119
116	India	4.565

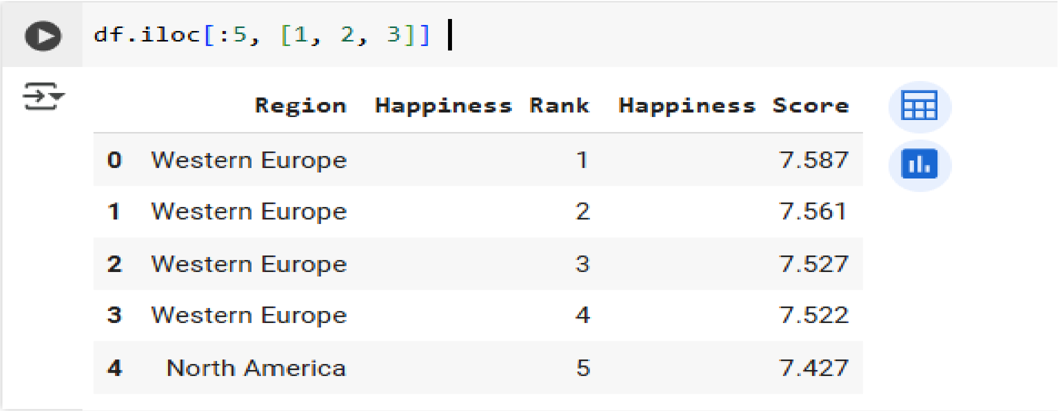
- Filtering with iloc (Index-Based Selection)



```
df.iloc[:5]
```

	Country	Region	Happiness Rank	Happiness Score	Standard Error	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)
0	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978
1	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145
2	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357

- Selecting first 5 rows and columns by index positions

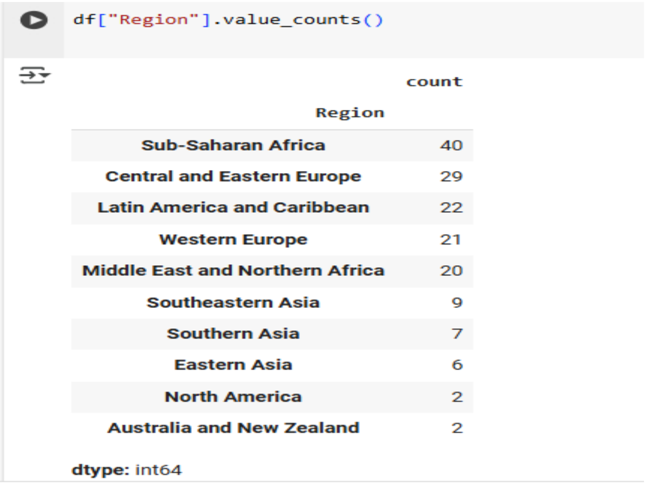


```
df.iloc[:5, [1, 2, 3]]
```

	Region	Happiness Rank	Happiness Score
0	Western Europe	1	7.587
1	Western Europe	2	7.561
2	Western Europe	3	7.527
3	Western Europe	4	7.522
4	North America	5	7.427

Value Counts

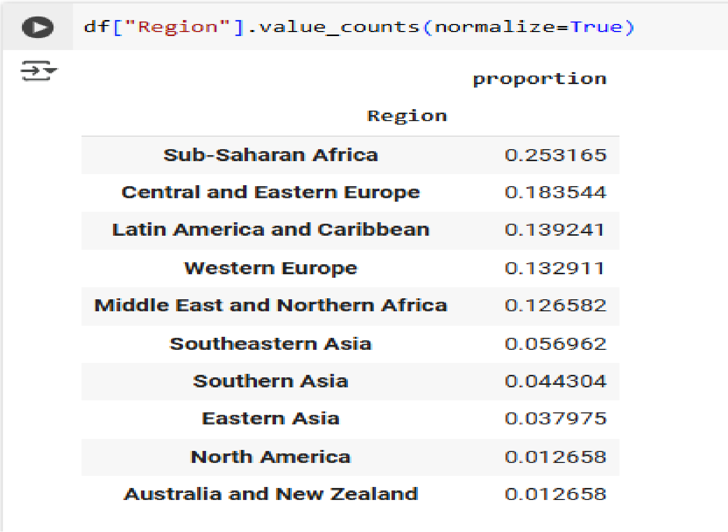
- Count Unique Values in a Column



```
df["Region"].value_counts()
```

```
                                       count
                              Region
               Sub-Saharan Africa         40
       Central and Eastern Europe         29
      Latin America and Caribbean         22
                   Western Europe         21
  Middle East and Northern Africa         20
               Southeastern Asia           9
                   Southern Asia           7
                    Eastern Asia           6
                   North America           2
       Australia and New Zealand           2

dtype: int64
```

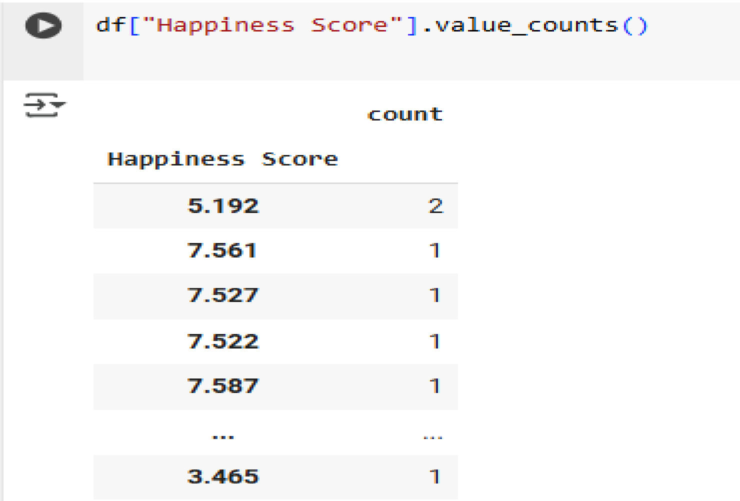
- Get proportion of each region in the dataset



```
df["Region"].value_counts(normalize=True)
```

```
                                        proportion
                              Region
               Sub-Saharan Africa         0.253165
       Central and Eastern Europe         0.183544
      Latin America and Caribbean         0.139241
                   Western Europe         0.132911
  Middle East and Northern Africa         0.126582
               Southeastern Asia          0.056962
                   Southern Asia          0.044304
                    Eastern Asia          0.037975
                   North America          0.012658
       Australia and New Zealand          0.012658
```

- Count Countries with a Specific Happiness Score



```
df["Happiness Score"].value_counts()
```

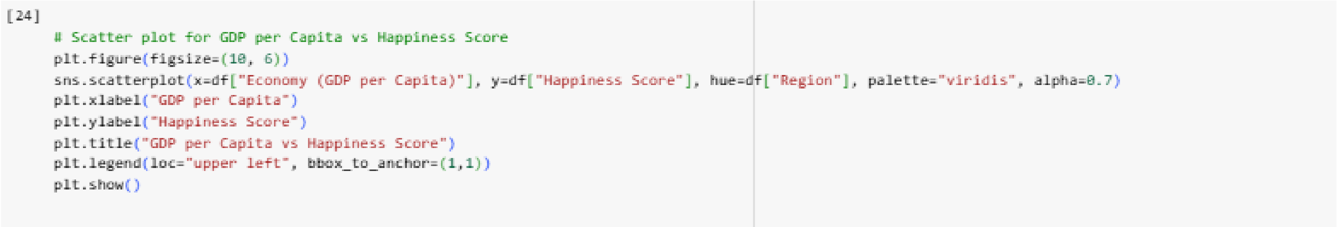
```
                      count
 Happiness Score
           5.192          2
           7.561          1
           7.527          1
           7.522          1
           7.587          1
             ...        ...
           3.465          1
```

Exploratory Data Analysis

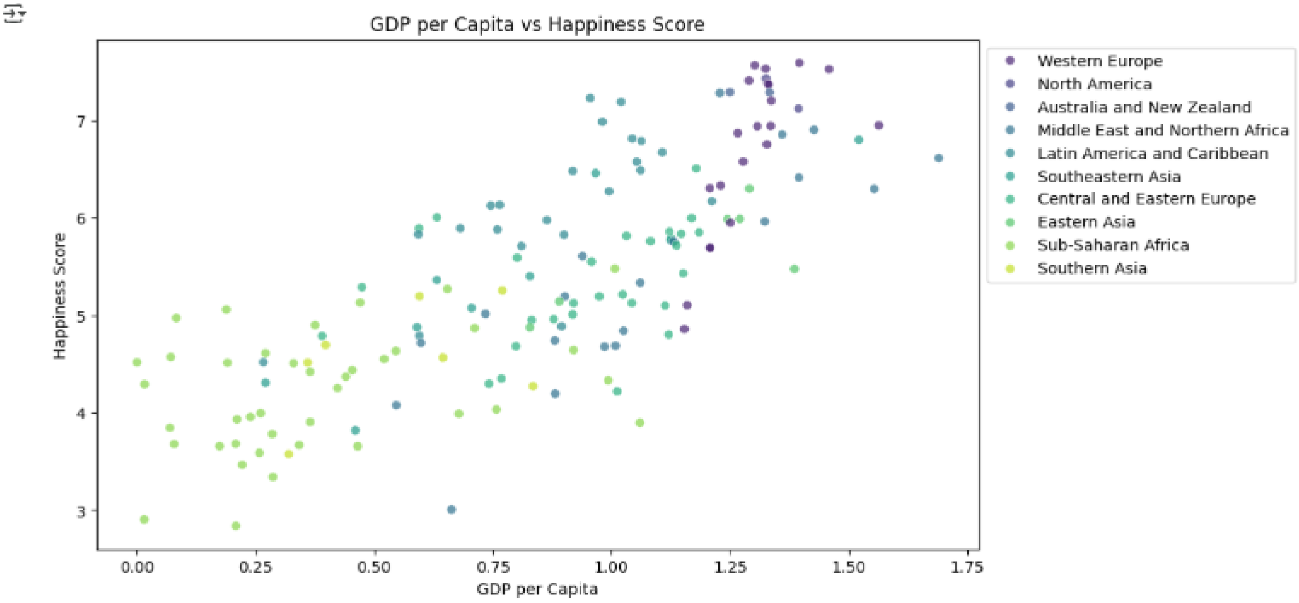
Exploratory Data Analysis (EDA) is a crucial preliminary step in data analysis, focusing on understanding the dataset's structure, identifying patterns, and uncovering relationships between variables. It involves visualizing data, summarizing key features, and detecting potential anomalies. EDA serves as a foundation for further analysis and model building.

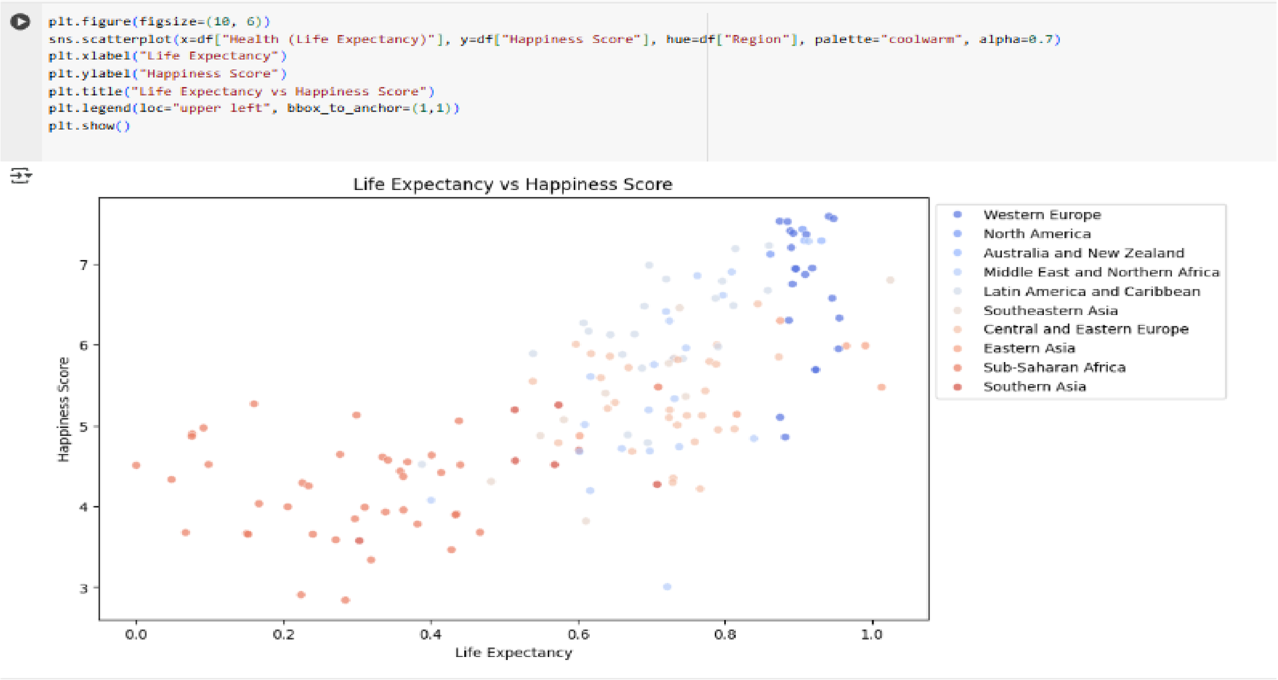
Scatter Plot Analysis:

Scatter plots are effective graphical tools for exploring relationships between two continuous variables. In the context of the World Happiness Report dataset, scatter plots can be used to visually assess potential associations between different factors influencing happiness scores. For instance, we can analyze how economic indicators, such as GDP per capita, correlate with Happiness Score, or explore the impact of Life Expectancy and Freedom on overall happiness. By leveraging scatter plots, we can identify trends, clusters, and outliers, providing valuable insights into the key drivers of happiness across different countries.



```
# Scatter plot for GDP per Capita vs Happiness Score
plt.figure(figsize=(10, 6))
sns.scatterplot(x=df["Economy (GDP per Capita)"], y=df["Happiness Score"], hue=df["Region"], palette="viridis", alpha=0.7)
plt.xlabel("GDP per Capita")
plt.ylabel("Happiness Score")
plt.title("GDP per Capita vs Happiness Score")
plt.legend(loc="upper left", bbox_to_anchor=(1,1))
plt.show()
```



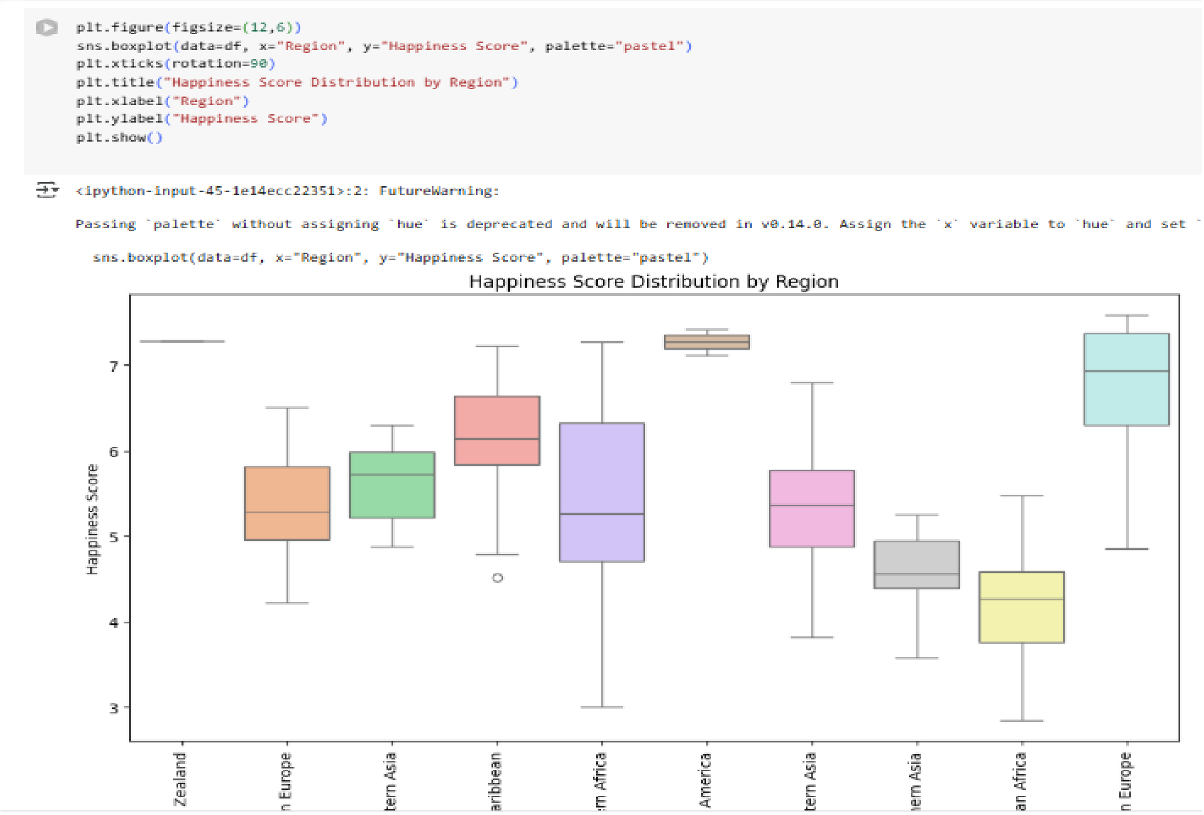


```
plt.figure(figsize=(10, 6))
sns.scatterplot(x=df["Health (Life Expectancy)"], y=df["Happiness Score"], hue=df["Region"], palette="coolwarm", alpha=0.7)
plt.xlabel("Life Expectancy")
plt.ylabel("Happiness Score")
plt.title("Life Expectancy vs Happiness Score")
plt.legend(loc="upper left", bbox_to_anchor=(1,1))
plt.show()
```

Life Expectancy vs Happiness Score
Happiness Score
3
4
5
6
7
0.0
0.2
0.4
0.6
0.8
1.0
Life Expectancy
Western Europe
North America
Australia and New Zealand
Middle East and Northern Africa
Latin America and Caribbean
Southeastern Asia
Central and Eastern Europe
Eastern Asia
Sub-Saharan Africa
Southern Asia

Boxplot: Happiness Score by Region

A box plot is a statistical visualization used to show the distribution of a numerical variable across different categories. It helps in identifying median values, spread, and potential outliers in the data.



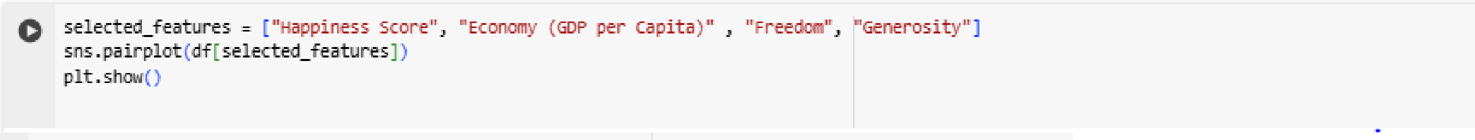
```
plt.figure(figsize=(12,6))
sns.boxplot(data=df, x="Region", y="Happiness Score", palette="pastel")
plt.xticks(rotation=90)
plt.title("Happiness Score Distribution by Region")
plt.xlabel("Region")
plt.ylabel("Happiness Score")
plt.show()
```

```
<ipython-input-45-1e14ecc22351>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `

  sns.boxplot(data=df, x="Region", y="Happiness Score", palette="pastel")
```

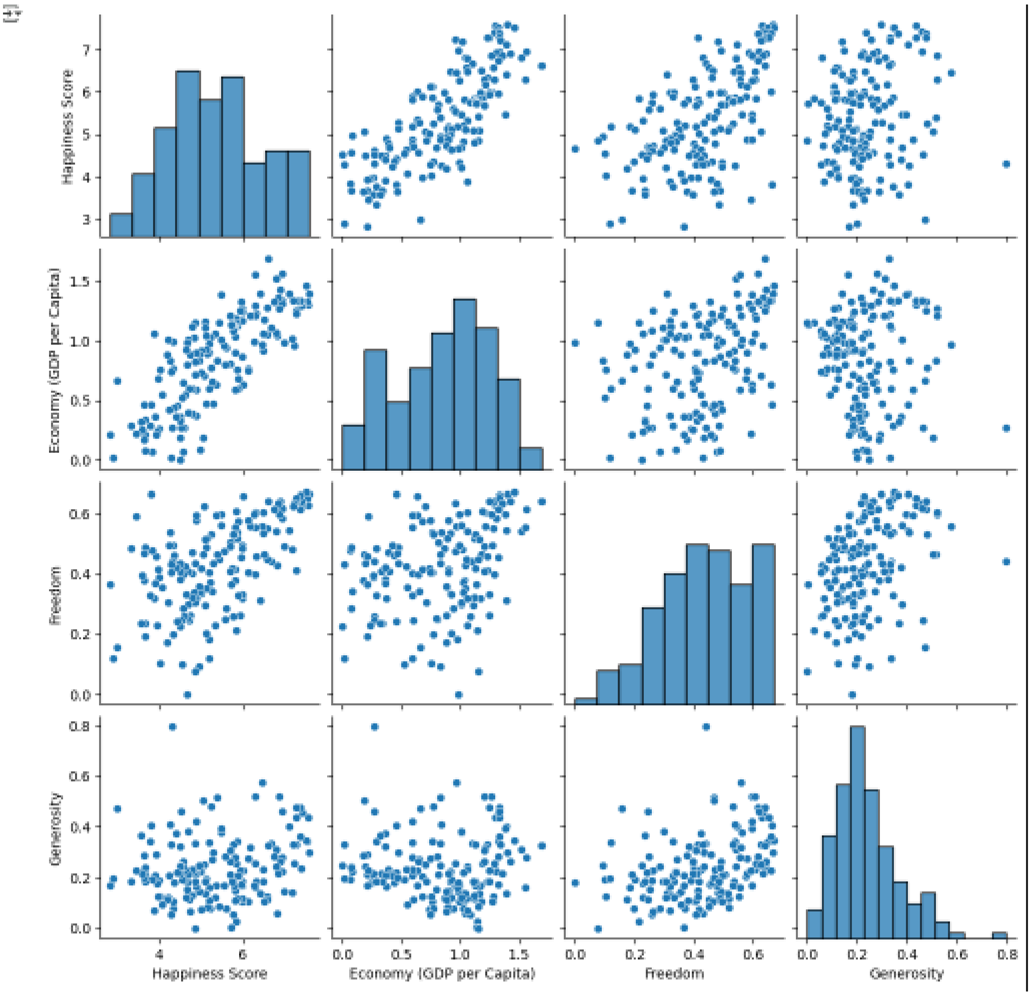
Pairplot: Correlation Between Key Features:

A pair plot is a powerful visualization tool in data analysis that displays pairwise relationships between multiple numerical variables in a dataset. It helps identify patterns, correlations, and potential outliers in the data.

In the World Happiness Report dataset, a pair plot can be used to explore relationships between key factors such as Happiness Score, GDP per Capita, Generosity, and Freedom. By analyzing these pairwise scatter plots, we can visually assess how different variables interact and influence happiness scores across different countries.



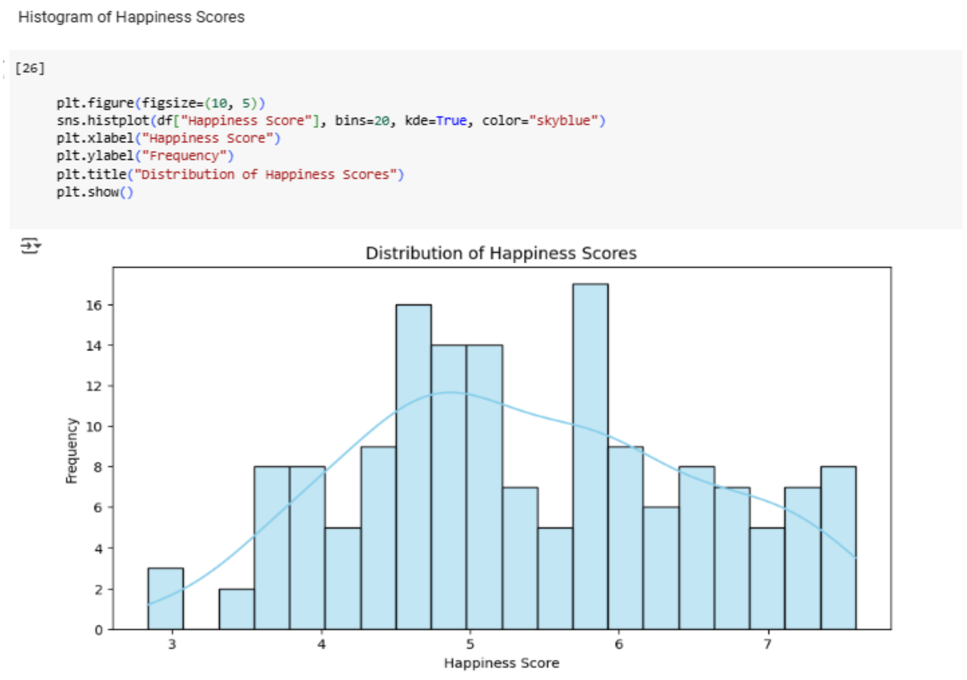
```
selected_features = ["Happiness Score", "Economy (GDP per Capita)" , "Freedom", "Generosity"]
sns.pairplot(df[selected_features])
plt.show()
```



Histogram

A histogram is a graphical representation that shows the distribution of a numerical variable by dividing it into bins and counting the number of observations in each bin. It helps in understanding the spread, skewness, and central tendency of the data.

- Histogram of Happiness score

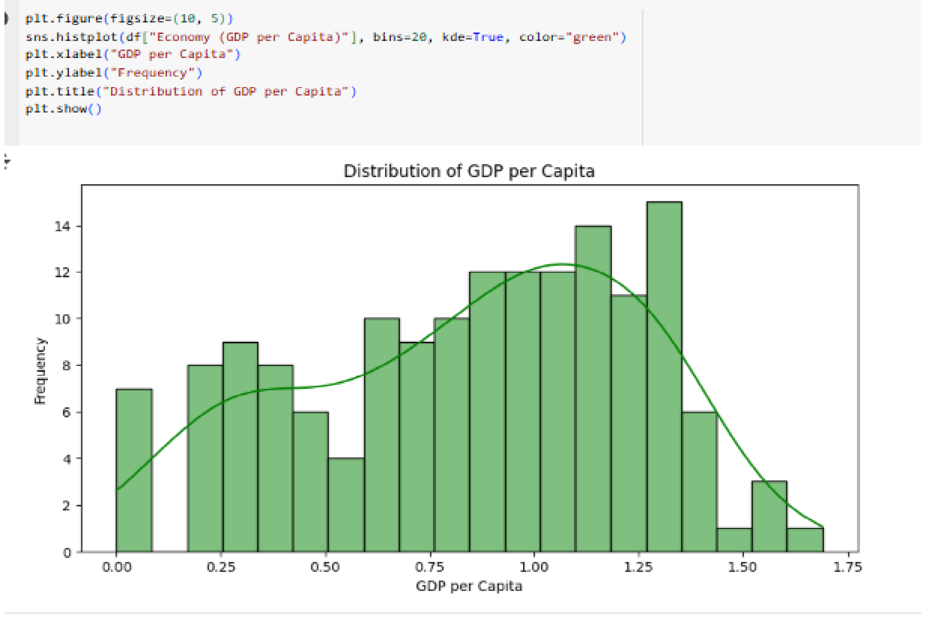


Histogram of Happiness Scores

[26]

```
plt.figure(figsize=(10, 5))
sns.histplot(df["Happiness Score"], bins=20, kde=True, color="skyblue")
plt.xlabel("Happiness Score")
plt.ylabel("Frequency")
plt.title("Distribution of Happiness Scores")
plt.show()
```

- Histogram of GDP per Capita



```
plt.figure(figsize=(10, 5))
sns.histplot(df["Economy (GDP per Capita)"], bins=20, kde=True, color="green")
plt.xlabel("GDP per Capita")
plt.ylabel("Frequency")
plt.title("Distribution of GDP per Capita")
plt.show()
```

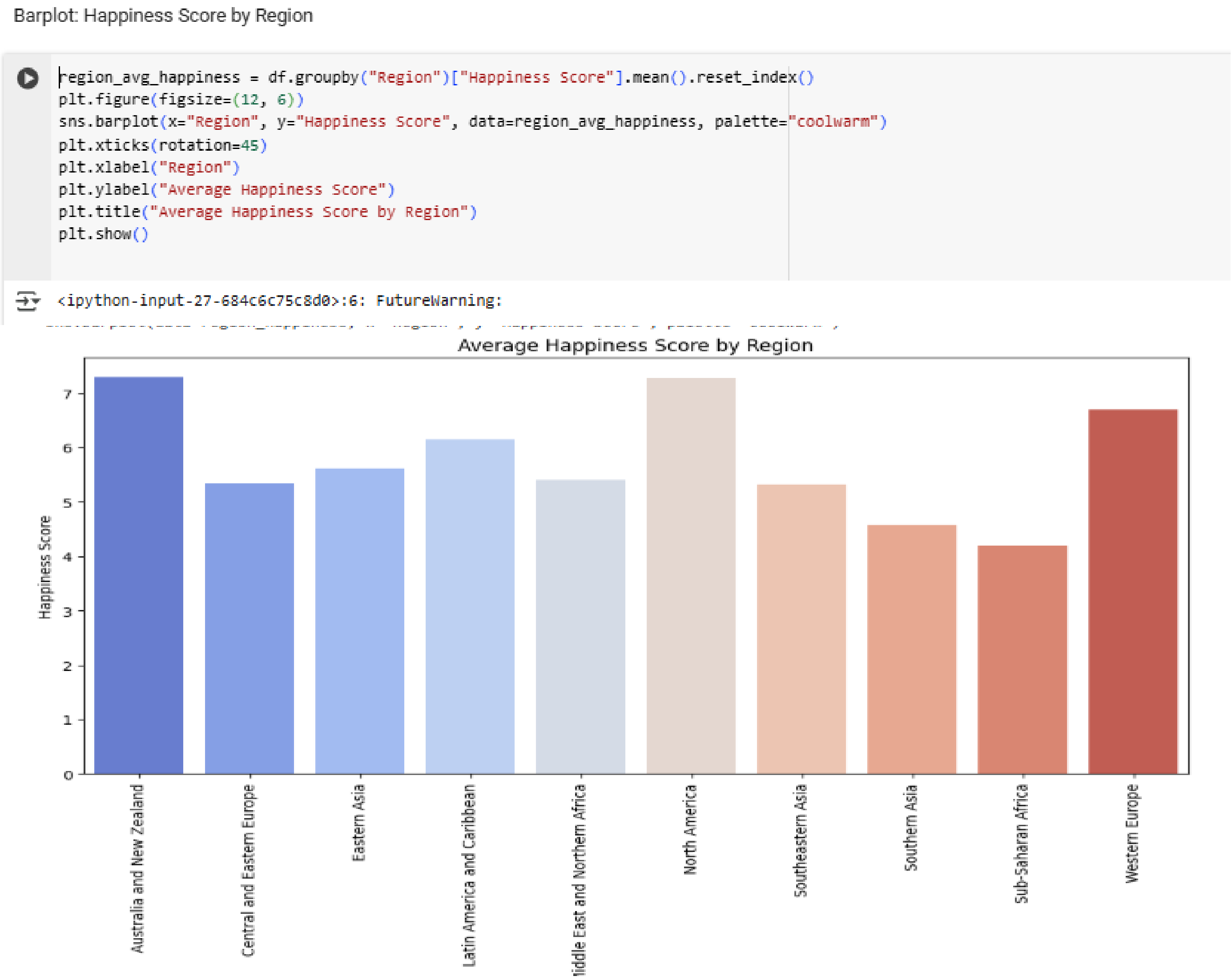
Bar Plot

A bar plot is a graphical representation that displays the average values of a numerical variable across different categories. It is useful for comparing group-wise statistics and identifying patterns.

In the World Happiness Report, a bar plot of Happiness Score by Region helps us understand:

1. Which regions have the highest and lowest average happiness scores
2. The variation in happiness levels across different parts of the world
3. How different socioeconomic factors might influence regional happiness

This visualization provides a clear comparison of happiness levels across regions, helping to identify trends and disparities.



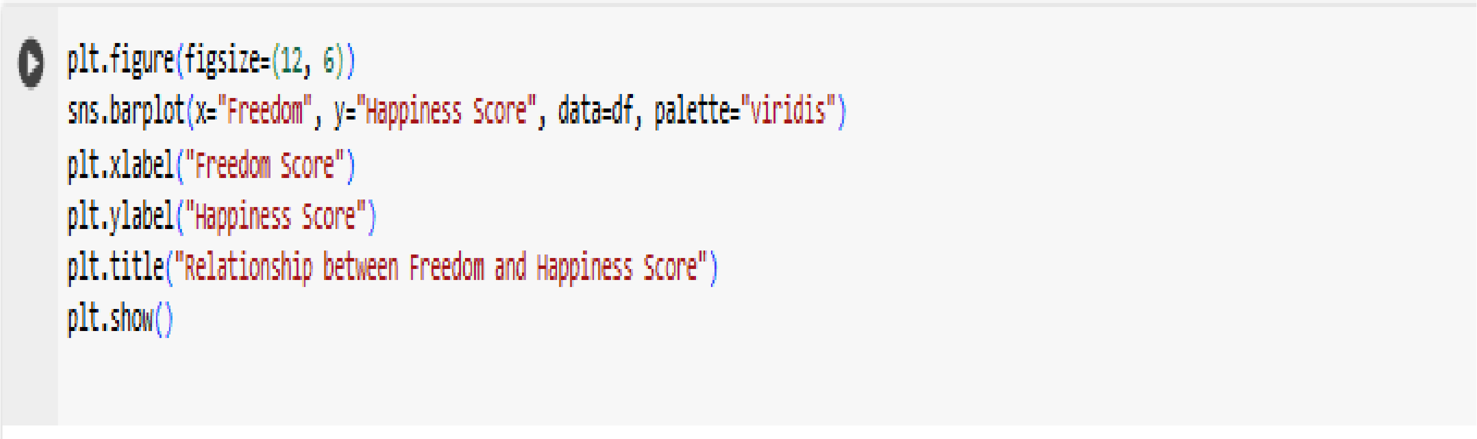
Barplot: Happiness Score by Region

```
region_avg_happiness = df.groupby("Region")["Happiness Score"].mean().reset_index()
plt.figure(figsize=(12, 6))
sns.barplot(x="Region", y="Happiness Score", data=region_avg_happiness, palette="coolwarm")
plt.xticks(rotation=45)
plt.xlabel("Region")
plt.ylabel("Average Happiness Score")
plt.title("Average Happiness Score by Region")
plt.show()
```

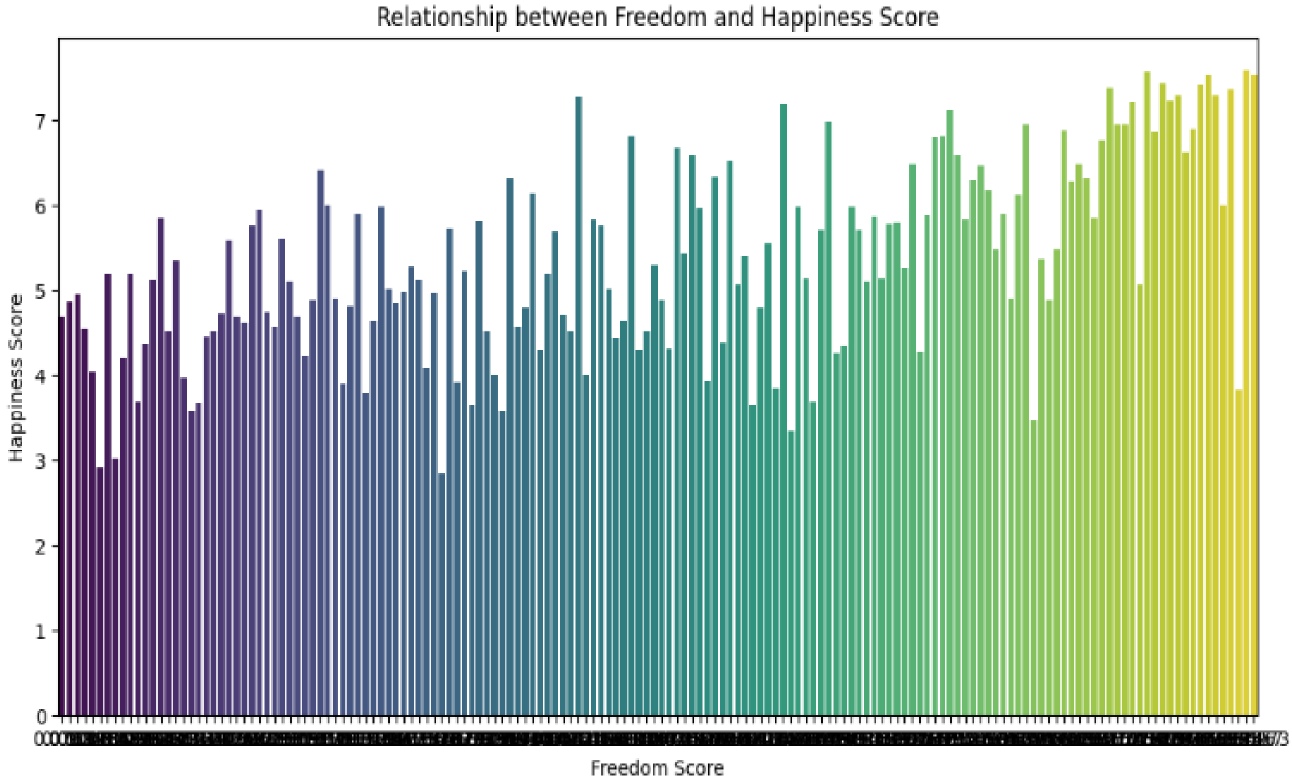
```
<ipython-input-27-684c6c75c8d0>:6: FutureWarning:
```

Average Happiness Score by Region
Happiness Score
0
1
2
3
4
5
6
7
Australia and New Zealand
Central and Eastern Europe
Eastern Asia
Latin America and Caribbean
Middle East and Northern Africa
North America
Southeastern Asia
Southern Asia
Sub-Saharan Africa
Western Europe

Barplot of Freedom vs Happiness score



```
plt.figure(figsize=(12, 6))
sns.barplot(x="Freedom", y="Happiness Score", data=df, palette="viridis")
plt.xlabel("Freedom Score")
plt.ylabel("Happiness Score")
plt.title("Relationship between Freedom and Happiness Score")
plt.show()
```

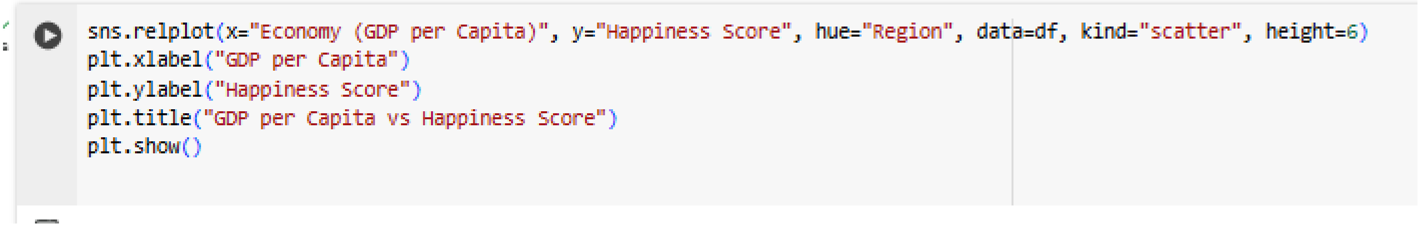


Seaborn

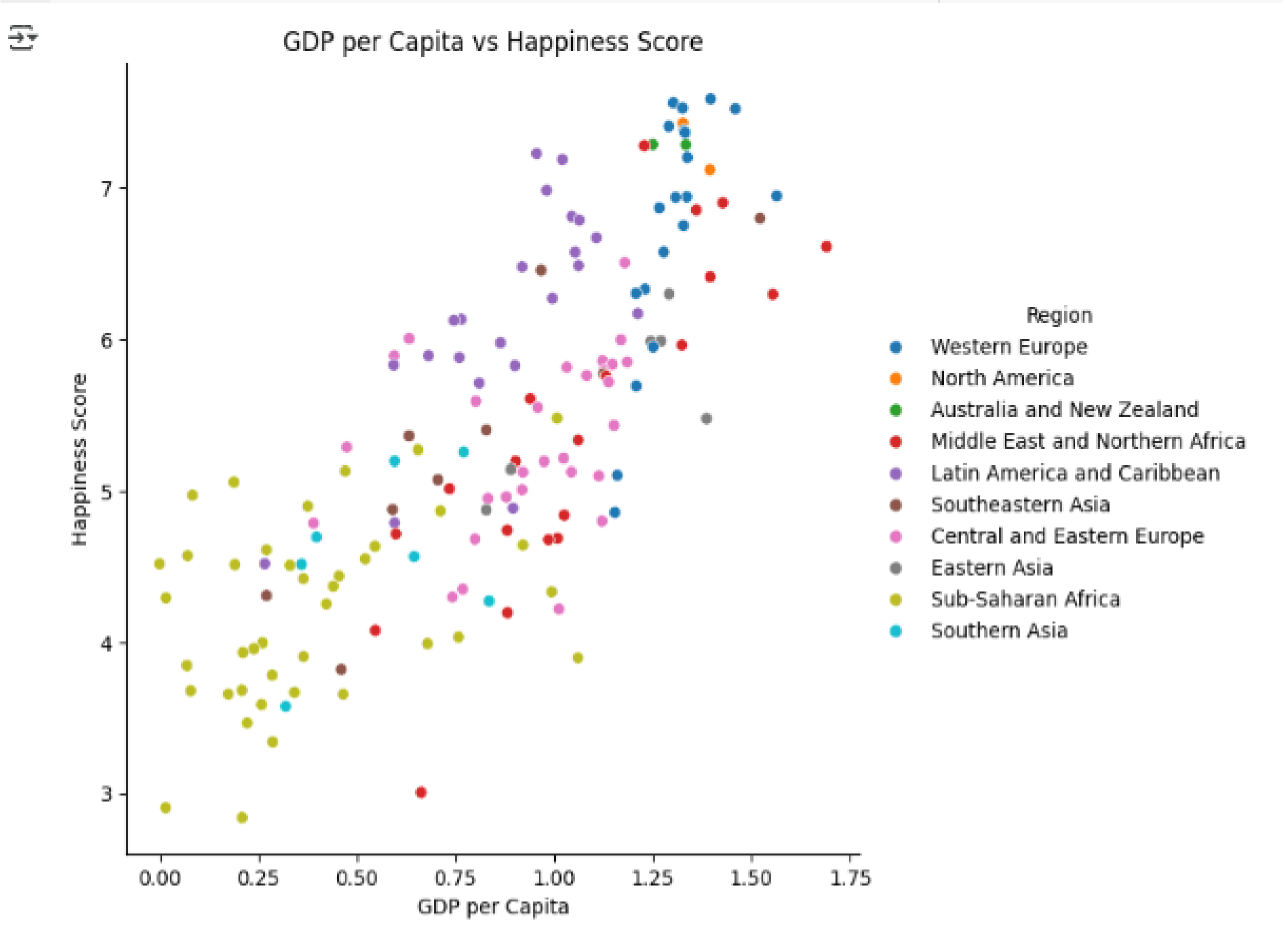
Seaborn is a powerful Python data visualization library built on Matplotlib, designed for creating attractive, informative, and statistical graphics with minimal code. It integrates well with Pandas DataFrames, making it easier to visualize trends, relationships, and distributions.

Key Features:

- Simplifies complex visualizations like pair plots, violin plots, and heatmaps.
- In-built themes for better aesthetics.
- Works well with categorical & numerical data.
- Supports statistical analysis, like regression trends.

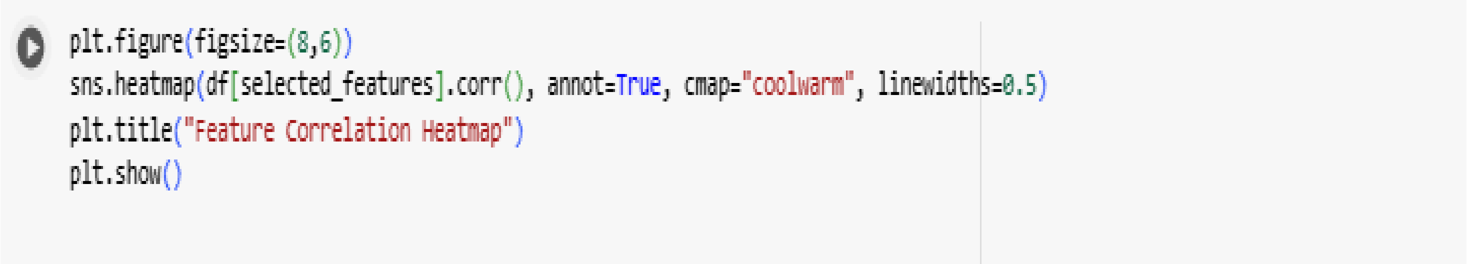


```
sns.relplot(x="Economy (GDP per Capita)", y="Happiness Score", hue="Region", data=df, kind="scatter", height=6)
plt.xlabel("GDP per Capita")
plt.ylabel("Happiness Score")
plt.title("GDP per Capita vs Happiness Score")
plt.show()
```

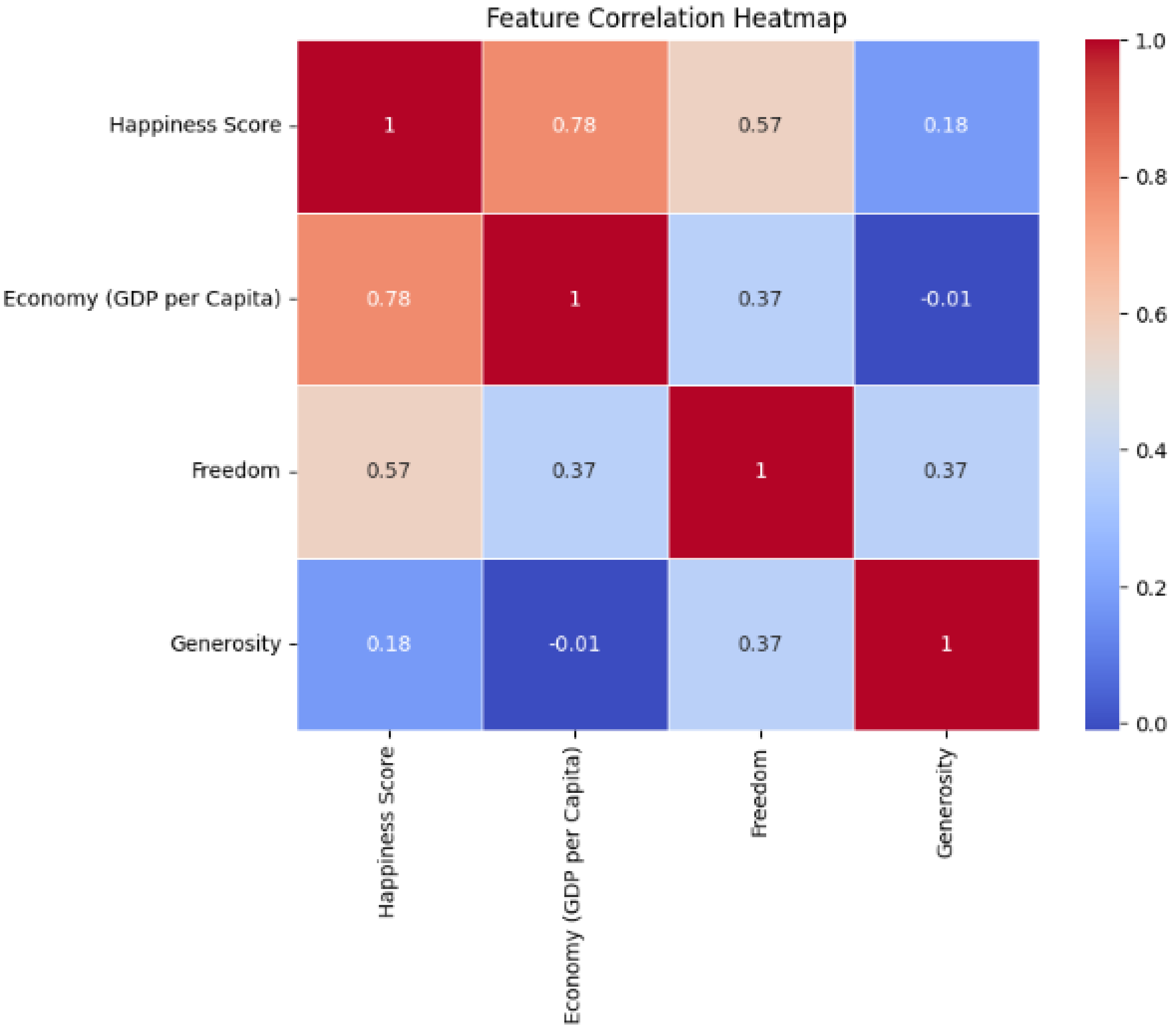


HEATMAP

A heatmap is a graphical representation of data using colors to indicate different values. In the context of a correlation matrix, a heatmap helps visualize relationships between numerical variables in the dataset.



```
plt.figure(figsize=(8,6))
sns.heatmap(df[selected_features].corr(), annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Feature Correlation Heatmap")
plt.show()
```



	Happiness Score	Economy (GDP per Capita)	Freedom	Generosity
Happiness Score	1	0.78	0.57	0.18
Economy (GDP per Capita)	0.78	1	0.37	-0.01
Freedom	0.57	0.37	1	0.37
Generosity	0.18	-0.01	0.37	1

4.4 Conclusion

This project provided a comprehensive exploration and analysis of the World Happiness Report dataset, uncovering key insights into the factors influencing happiness across different countries. Through data visualization, correlation analysis, and feature exploration, we identified significant relationships between economic stability, social support, life expectancy, and perceived freedom in determining happiness scores.

By leveraging statistical methods and visualization techniques, the study highlighted disparities in well-being across regions and provided a clearer understanding of the underlying socioeconomic patterns. These insights serve as a valuable foundation for further research and policy recommendations aimed at enhancing global happiness. The findings emphasize the importance of economic development, governance, and social welfare policies in improving the overall quality of life.

This analysis sets the stage for future studies that could incorporate predictive modeling or time-series analysis to assess trends in global happiness and forecast potential changes.

References

1. **Pandas Documentation:**
   - McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 51-56.
   - Pandas Documentation
2. **Matplotlib Documentation:**
   - Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3), 90-95.
   - Matplotlib Documentation
3. **Seaborn Documentation:**
   - Waskom, M. L. (2021). Seaborn: Statistical Data Visualization. *Journal of Open Source Software*, 6(60), 3021.
   - Seaborn Documentation

4.**Exploratory Data Analysis (EDA):**

- Behrens, J. T. (1997). Principles and Procedures of Exploratory Data Analysis.
  Psychological Methods, 2(2), 131-160.
- Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

4. **Visualization Techniques:**
   - Few, S. (2012). *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press.
   - Kalik, C. N. (2015). *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley.
5. **Python for Data Analysis:**
   - McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and I Python*. O'Reilly Media.

6. **VanderPlas, J. (Python Data Science Handbook)**:

- Essential Tools for Working with Data. O'Reilly Media