Literacy Rate in ASIA(2010-2018)

1	F22-3138	Hamza Abdul Ali	BsAI-3B
2	F22-3130	Shehroz Abbas	BsAI-3B
3	F22-3177	Hassan Tahir	BsAI-3B
4	F21-9629	Almas Aina	BsCs-5B
5	F21-9180	Syeda Daniya	BsCs-5B

1.Problem Statement

The problem statement of "Literacy Rate in Asia (2010-2018)" is to analyze the historical Literacy rate of Asia over the period of 2010-2018. The study aims to comprehensively analyze a diverse dataset encompassing socio-demographic variables such as gender, age, country, region, and the critical metric of literacy rate. The primary objective is to understand the multifaceted relationships and dependencies between these demographic factors and literacy rates across various populations. Through extensive exploratory data analysis (EDA) and statistical modeling techniques, the research endeavors to uncover underlying patterns, correlations, and disparities influencing literacy rates. The study's focus extends to examining regional disparities, gender dynamics, and age-related influences on literacy rates, intending to derive actionable insights that can inform targeted policies, interventions, and educational strategies. By leveraging data-driven methodologies and predictive modeling approaches, the study aims to contribute insights beneficial for educators, policymakers, and social organizations, ultimately striving to enhance literacy rates and foster equitable access to education."

2.Objective

The objectives of this project are:

□ To perform exploratory data analysis on the Literacy rate dataset and visualize any patterns or trends in the data.

□ To calculate descriptive statistical measures such as mean, median, and standard deviation for the temperature data.

□ To analyze the linearity of the data and use regression models to the data to make predictions about Literacy rate in Asia.

3.Data Description

The dataset used in this project is taken from Kaggle and contains Literacy rate data for countries in Asia from the year 2010-2018. The data includes the following fields: Region,Country,Year,Age,Gender,Literacy rate

The dataset has 889 entries.

LINK:

https://www.kaggle.com/datasets/thedevastator/youth-adult-literacy-rates-in-2019

4.Results

The results of this analysis will be presented through a Desktop Application, which will display the following:

□ Simple BarCharts's, Multiple BarCharts's

□ Pie Chart’s, Histogram, Boxplot's

□ Descriptive statistical measures such as mean, median, and standard deviation.

□ Uniform Distribution Method predicted and calculated values for probability(x),Expected_Value(x),variance(x),std(x).

□ A Scatter Plot and Co-relation Coefficient to analyze the linearity of the Data

□ Confidence Interval for Descriptive Measures

5.Code in Python and Output’s

Libraries Used in python

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy import stats
```

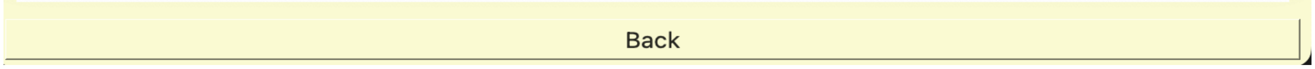
Reading the DATASET

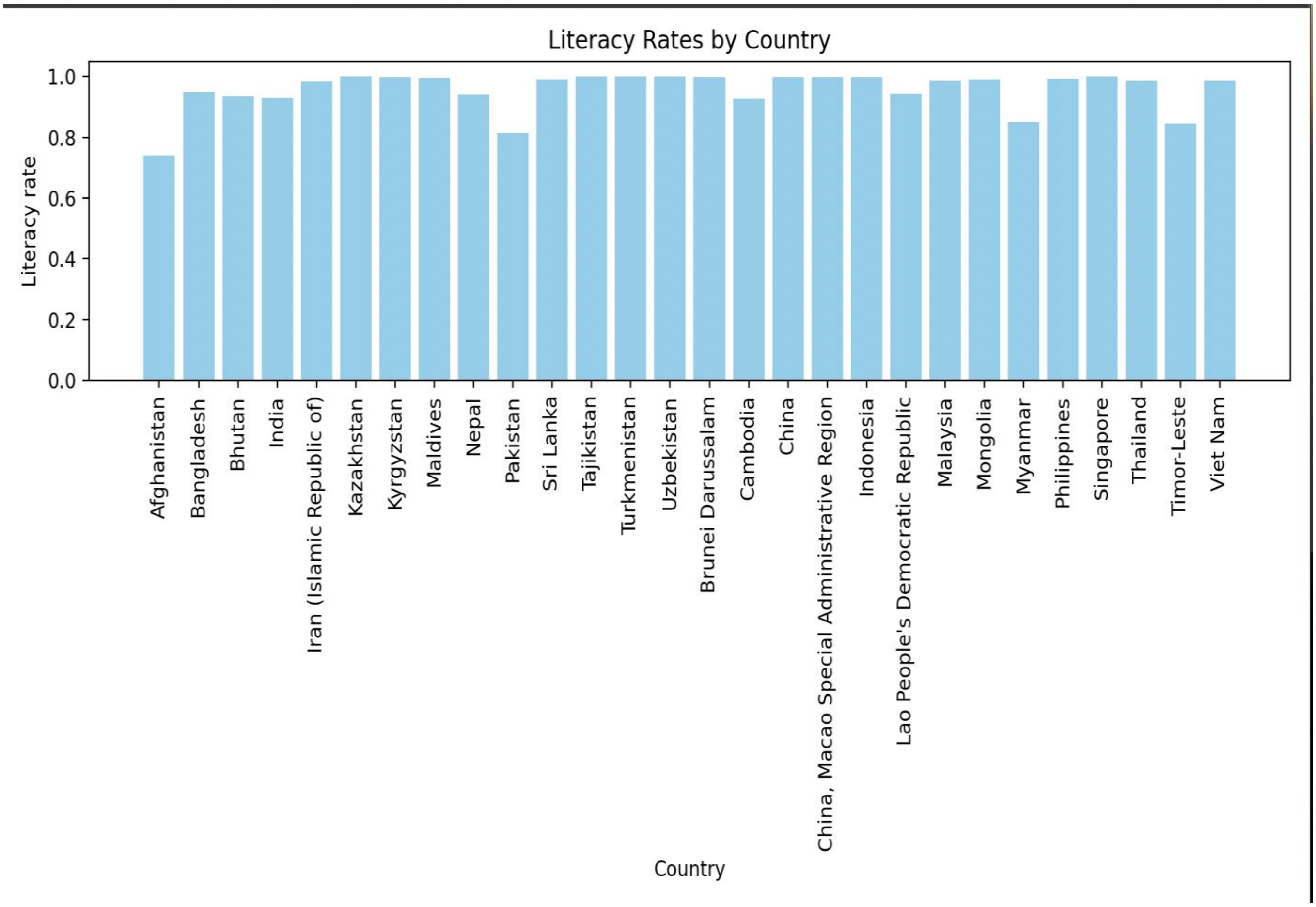
```
file_path = '/Users/syedadaniya/Desktop/LR.csv'

# Read the CSV file

df = pd.read_csv(file_path)data = pd.read_csv(file_path)
```

SIMPLE BAR CHART’S

```
# SIMPLE BAR CHART

# dataset with columns 'Country' and 'Literacy Rate'

data = pd.read_csv(file_path)
```

```
countries = data['Country']
literacy_rates = data['Literacy rate']


# Plotting a bar chart
plt.figure(figsize=(10, 6)) # Set the figure size (width, height)


plt.bar(countries, literacy_rates, color='skyblue') # Plotting the bar chart
plt.xlabel('Country') # Label for x-axis
plt.ylabel('Literacy rate') # Label for y-axis
plt.title('Literacy Rates by Country') # Title of the plot
plt.xticks(rotation=90) # Rotate x-axis labels for better readability


plt.tight_layout() # Adjust layout to prevent clipping of labels
plt.show() # Display the plot
```

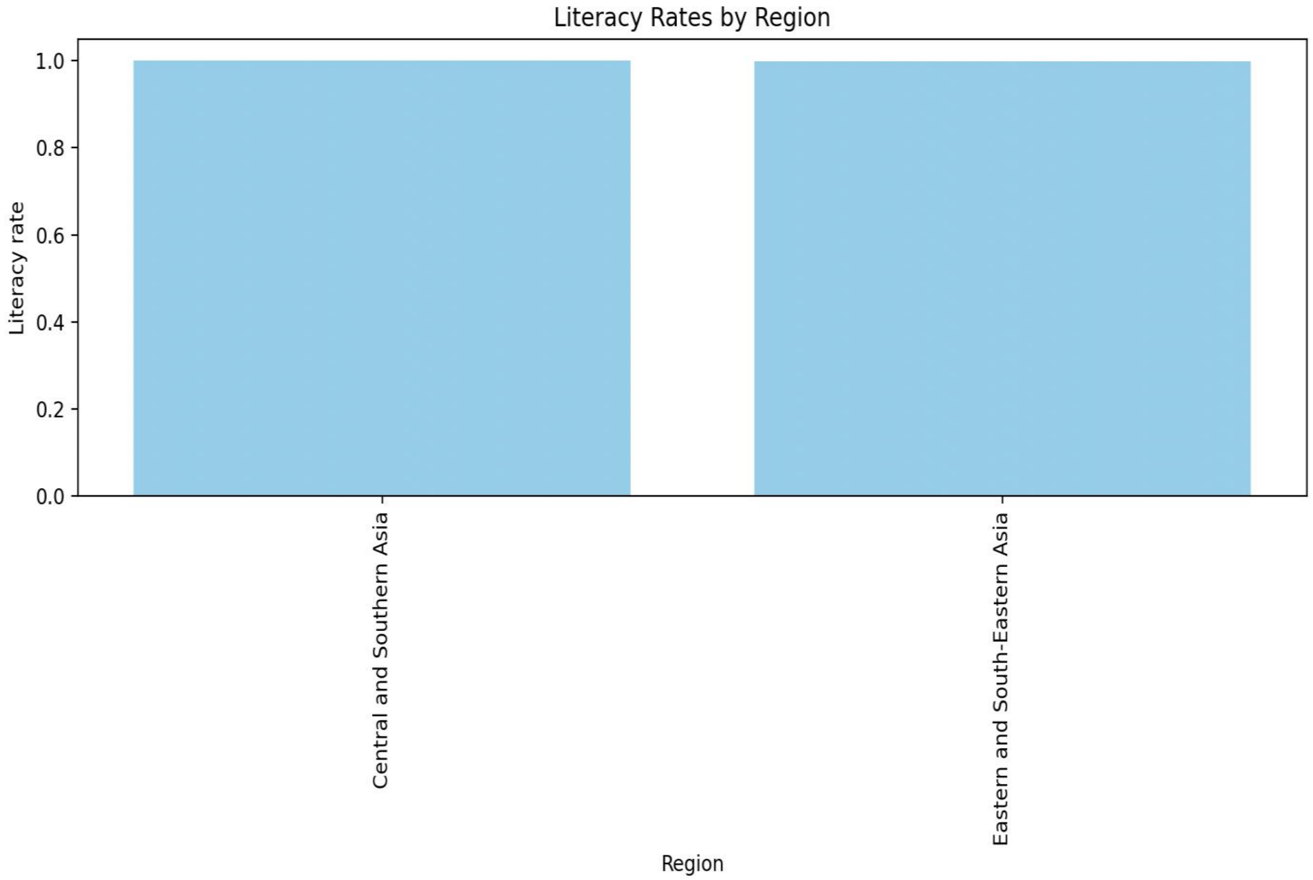




SIMPLE BAR CHART - Literacy Rates by Country

This bar chart utilizes the 'Country' variable on the x-axis and 'Literacy Rate' on the y-axis. It represents the literacy rates across different countries, providing a visual comparison of literacy rates among various nations.

```
#2

region = data['Region']

literacy_rates = data['Literacy rate']



# Plotting a bar chart

plt.figure(figsize=(10, 6)) # Set the figure size (width, height)



plt.bar(region, literacy_rates, color='skyblue') # Plotting the bar chart
plt.xlabel('Region') # Label for x-axis
plt.ylabel('Literacy rate') # Label for y-axis
plt.title('Literacy Rates by Region') # Title of the plot
plt.xticks(rotation=90) # Rotate x-axis labels for better readability

plt.tight_layout() # Adjust layout to prevent clipping of labels
plt.show() # Display the plot
```



SIMPLE BAR CHART - Literacy Rates by Region
Similarly, this bar chart uses the 'Region' variable on the x-axis and 'Literacy Rate' on the y-axis. It illustrates the literacy rates categorized by different regions, facilitating a comparison of literacy levels across geographical regions.

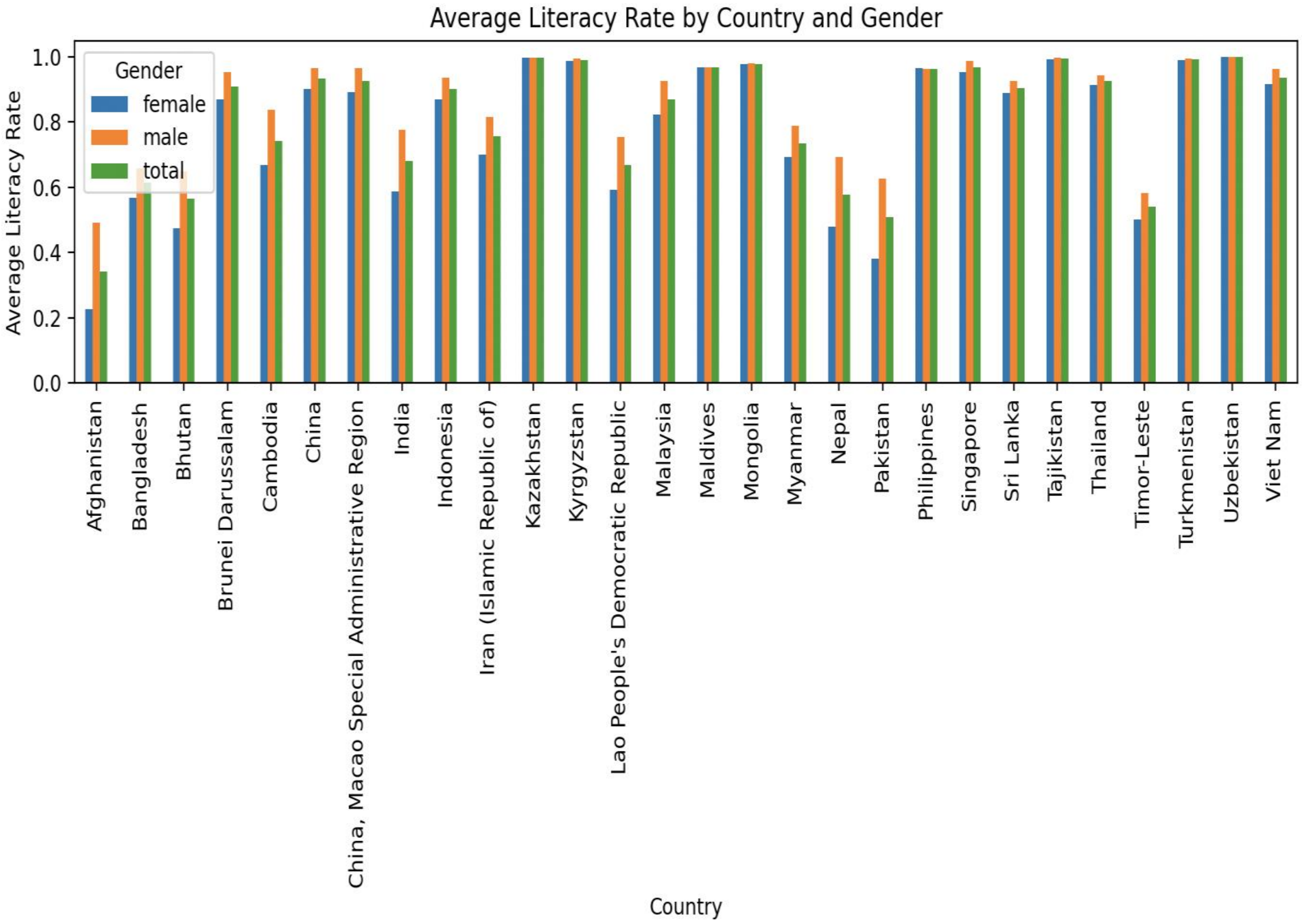
```
#3

gender = data['Gender']

literacy_rates = data['Literacy rate']



# Plotting a bar chart

plt.figure(figsize=(10, 6)) # Set the figure size (width, height)



plt.bar(gender, literacy_rates, color='skyblue') # Plotting the bar chart
plt.xlabel('Gender') # Label for x-axis
plt.ylabel('Literacy rate') # Label for y-axis
plt.title('Literacy Rates by Gender') # Title of the plot
plt.xticks(rotation=90) # Rotate x-axis labels for better readability


plt.tight_layout() # Adjust layout to prevent clipping of labels
```

SIMPLE BAR CHART - Literacy Rates by Gender

In this graph, the 'Gender' variable is plotted on the x-axis, while the 'Literacy Rate' is on the y-axis. It presents the literacy rates categorized by gender, offering insights into literacy levels concerning gender distribution

MULTIPLE BAR CHART'S

```
# MULTIPLE BAR CHART


data = pd.read_csv(file_path)
# Group data by 'Country' and 'Gender', and calculate mean Literacy Rate
grouped_data = data.groupby(['Country', 'Gender'])['Literacy rate'].mean().unstack()


# Plotting a multiple bar chart
grouped_data.plot(kind='bar', figsize=(10, 6))
plt.xlabel('Country')
plt.ylabel('Average Literacy Rate')
plt.title('Average Literacy Rate by Country and Gender')
plt.legend(title='Gender')
plt.tight_layout()

plt.show()
```

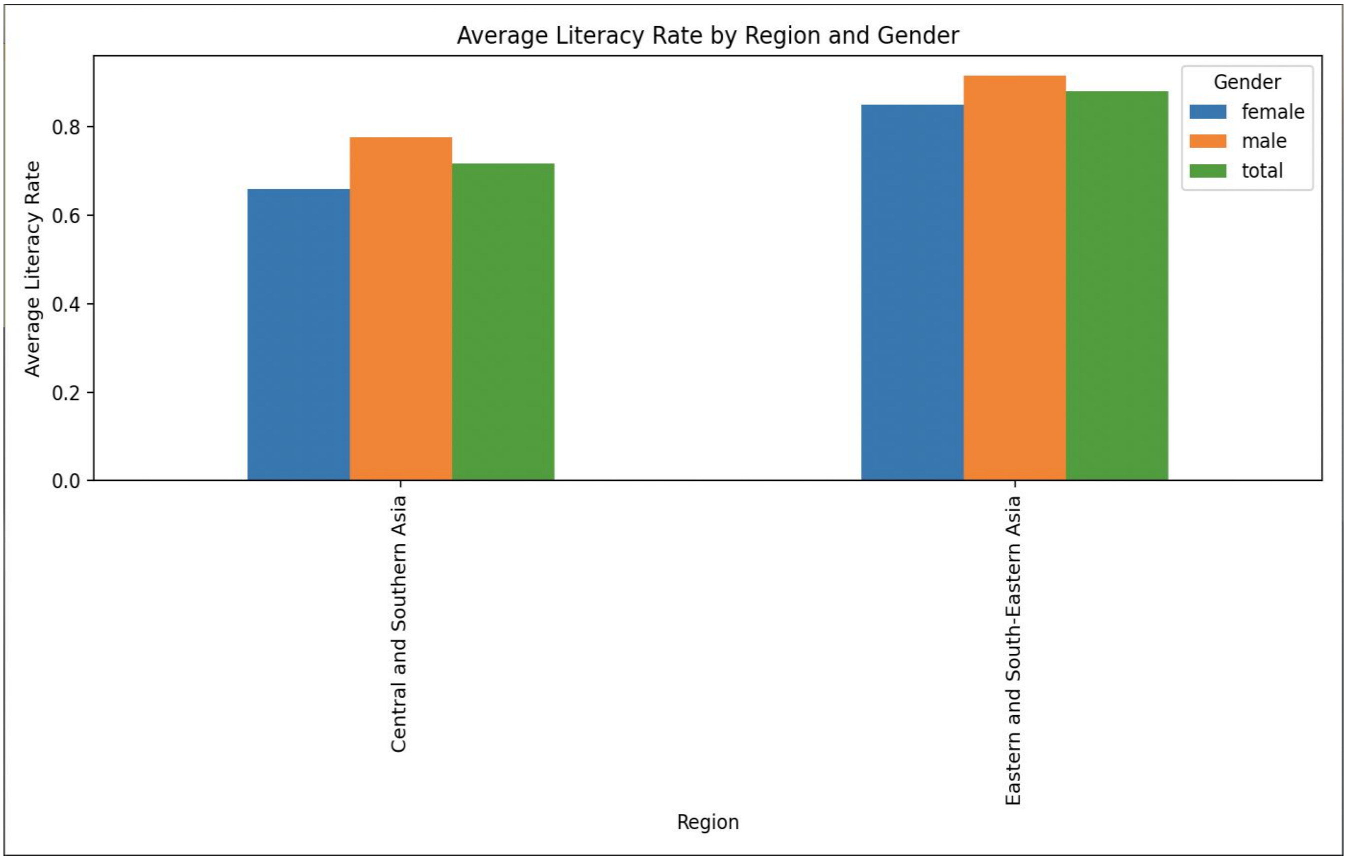


MULTIPLE BAR CHART - Average Literacy Rate by Country and Gender

This multiple bar chart showcases the average literacy rates across various countries, categorized by gender. Each country has bars representing different genders, enabling a comparison of average literacy rates based on both country and gender.

```
#2

data = pd.read_csv(file_path)

# Group data by 'Country' and 'Gender', and calculate mean Literacy Rate
```

```
grouped_data = data.groupby(['Region', 'Gender'])['Literacy rate'].mean().unstack()


# Plotting a multiple bar chart
grouped_data.plot(kind='bar', figsize=(10, 6))
plt.xlabel('Region')
plt.ylabel('Average Literacy Rate')
plt.title('Average Literacy Rate by Region and Gender')
plt.legend(title='Gender')


plt.tight_layout()
plt.show()
```



MULTIPLE BAR CHART - Average Literacy Rate by Region and Gender

Similar to the previous graph, this multiple bar chart represents the average literacy rates categorized by both region and gender. Each region contains bars for different genders, providing insights into literacy rates across regions and genders

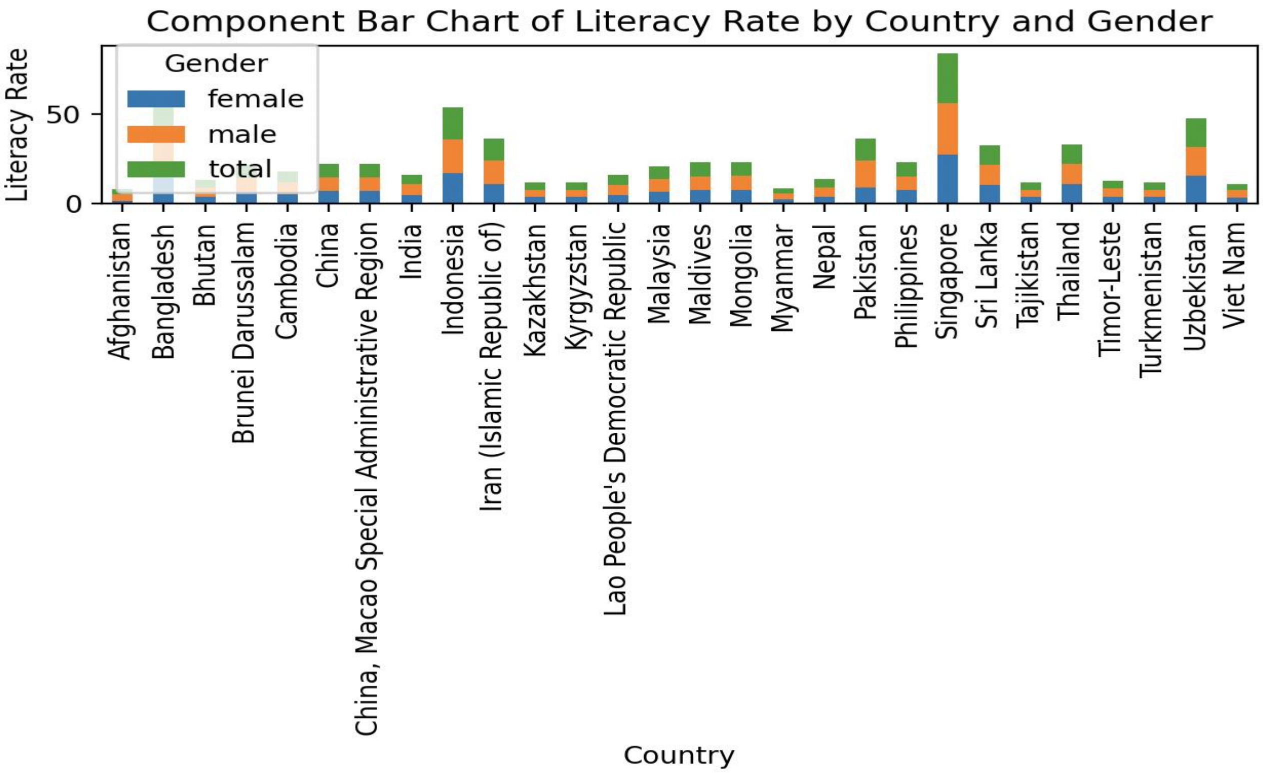
```
#3

data = pd.read_csv(file_path)

# Group data by 'Country' and 'Gender', and calculate mean Literacy Rate

grouped_data = data.groupby(['Region', 'Country'])['Literacy rate'].mean().unstack()



# Plotting a multiple bar chart

grouped_data.plot(kind='bar', figsize=(10, 10))

plt.xlabel('Region')

plt.ylabel('Average Literacy Rate')

plt.title('Average Literacy Rate by Region and Country')
plt.legend(title='Country')


plt.tight_layout()
plt.show()
```

MULTIPLE BAR CHART - Average Literacy Rate by Region and Country

This multiple bar chart displays the average literacy rates categorized by both region and country. Each region contains bars representing different countries, offering a comparative view of average literacy rates across regions for multiple countries

COMPONENT BAR CHART’S

```
#COMPONENT BAR CHART
grouped_data = data.groupby(['Country', 'Gender'])['Literacy rate'].sum().unstack()


# Plotting the component bar chart
plt.figure(figsize=(10, 6))


grouped_data.plot(kind='bar', stacked=True)
plt.xlabel('Country')
plt.ylabel('Literacy Rate')
plt.title('Component Bar Chart of Literacy Rate by Country and Gender')


plt.legend(title='Gender')
plt.tight_layout()
plt.show()
```



COMPONENT BAR CHART - Literacy Rate by Country and Gender

Utilizing 'Country' as the x-axis variable, this stacked bar chart visualizes the components of literacy rates categorized by gender for different countries. It illustrates the gender distribution within each country concerning literacy rates

```
#2
grouped_data = data.groupby(['Region', 'Gender'])['Literacy
rate'].sum().unstack()


# Plotting the component bar chart
plt.figure(figsize=(10, 6))


grouped_data.plot(kind='bar', stacked=True)
plt.xlabel('Region')
plt.ylabel('Literacy Rate')
plt.title('Component Bar Chart of Literacy Rate by Region and Gender')


plt.legend(title='Gender')


plt.tight_layout()
plt.show()
```

Region

COMPONENT BAR CHART - Literacy Rate by Region and Gender
Similarly, this stacked bar chart employs 'Region' as the x-axis variable to display the components of literacy rates categorized by gender for different regions. It provides insights into gender distribution within regions concerning literacy rates.

```
#3
grouped_data = data.groupby(['Region', 'Country'])['Literacy
rate'].sum().unstack()


# Plotting the component bar chart


plt.figure(figsize=(10, 8))
grouped_data.plot(kind='bar', stacked=True)
plt.xlabel('Region')
plt.ylabel('Literacy Rate')
plt.title('Component Bar Chart of Literacy Rate by Region and Country')


plt.legend(title='Country')
plt.tight_layout()
plt.show()
```

COMPONENT BAR CHART - Literacy Rate by Region and Country

Using 'Region' on the x-axis, this stacked bar chart visualizes the components of literacy rates categorized by countries within regions. It illustrates the contribution of each country to the overall literacy rate within specific regions.

PIE CHART’S

```
#PIE-CHART


# Grouping data by region and calculating the mean literacy rate
grouped_data = data.groupby('Region')['Literacy rate'].mean().reset_index()


# Plotting a pie chart showing the distribution of mean literacy rate by region
fig, ax = plt.subplots()
colors = plt.get_cmap('tab20c').colors # Using a colormap for colors


# Plotting the pie chart
ax.pie(grouped_data['Literacy rate'], labels=grouped_data['Region'], autopct='%1.1f%%', colors=colors, startangle=90)
ax.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle


plt.title('Distribution of Literacy Rate by Region')
plt.show()
```

PIE CHART - Distribution of Literacy Rate by Region

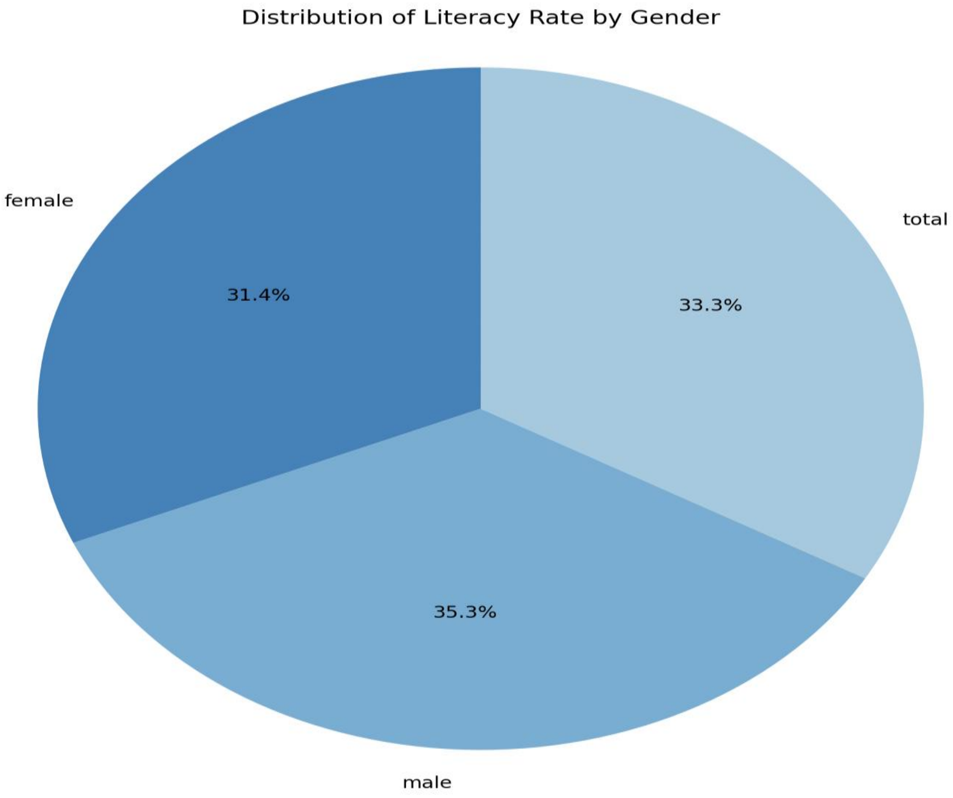
This pie chart represents the distribution of mean literacy rates across different regions. Each section of the pie chart corresponds to a region, indicating the proportion of literacy rates contributed by each region to the total.

```
#2


# Grouping data by region and calculating the mean literacy rate
grouped_data = data.groupby('Country')['Literacy rate'].mean().reset_index()


# Plotting a pie chart showing the distribution of mean literacy rate by region
fig, ax = plt.subplots()
colors = plt.get_cmap('tab20c').colors # Using a colormap for colors


ax.pie(grouped_data['Literacy rate'], labels=grouped_data['Country'], autopct='%1.1f%%', colors=colors, startangle=90)
ax.axis('equal')


plt.title('Distribution of Literacy Rate by Country')
plt.show()
```

PIE CHART - Distribution of Literacy Rate by Country

Similar to the previous pie chart, this one illustrates the distribution of mean literacy rates across various countries. Each segment represents a country, showcasing its contribution to the overall literacy rate distribution.

```
#3

# Grouping data by region and calculating the mean literacy rate
grouped_data = data.groupby('Gender')['Literacy rate'].mean().reset_index()

# Plotting a pie chart showing the distribution of mean literacy rate by region
fig, ax = plt.subplots()
colors = plt.get_cmap('tab20c').colors # Using a colormap for colors
ax.pie(grouped_data['Literacy rate'], labels=grouped_data['Gender'], autopct='%1.1f%%', colors=colors, startangle=90)
ax.axis('equal')

plt.title('Distribution of Literacy Rate by Gender')
plt.show()
plt.figure(figsize=(8, 6))
```

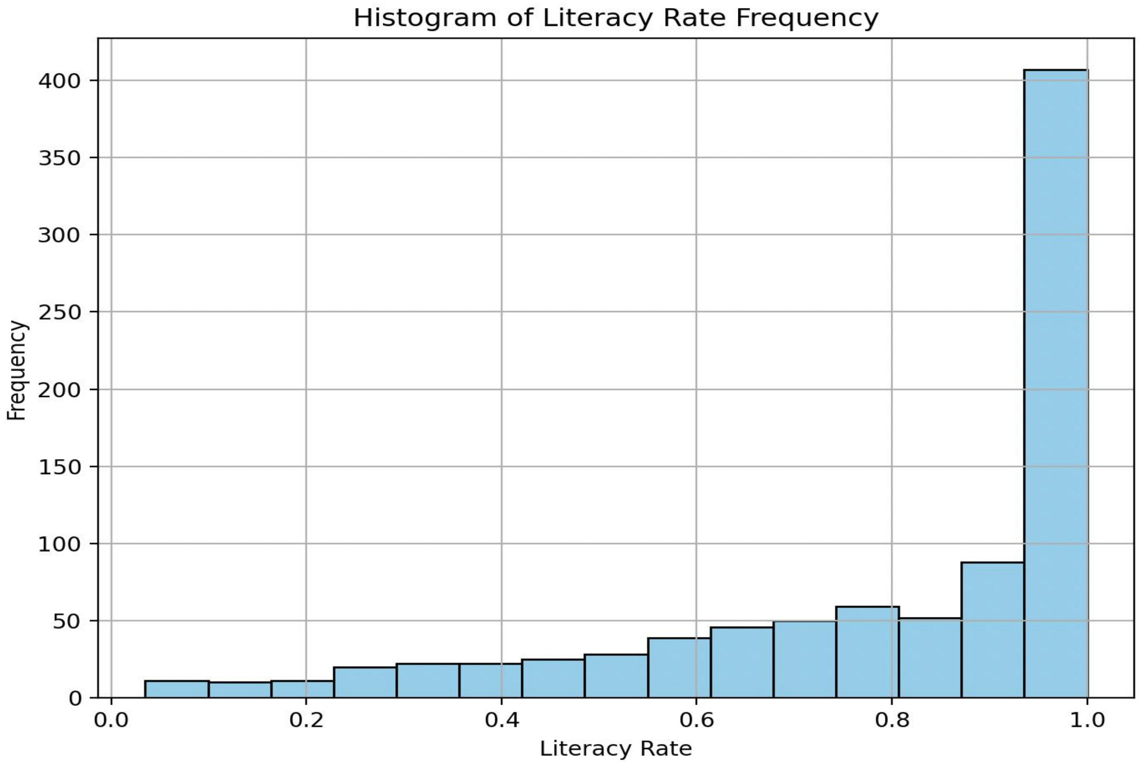


PIE CHART - Distribution of Literacy Rate by Gender
This pie chart displays the distribution of mean literacy rates based on gender. Each segment corresponds to a gender category, illustrating the proportion of literacy rates contributed by each gender.

HISTOGRAM'S

```
#Histogram
```

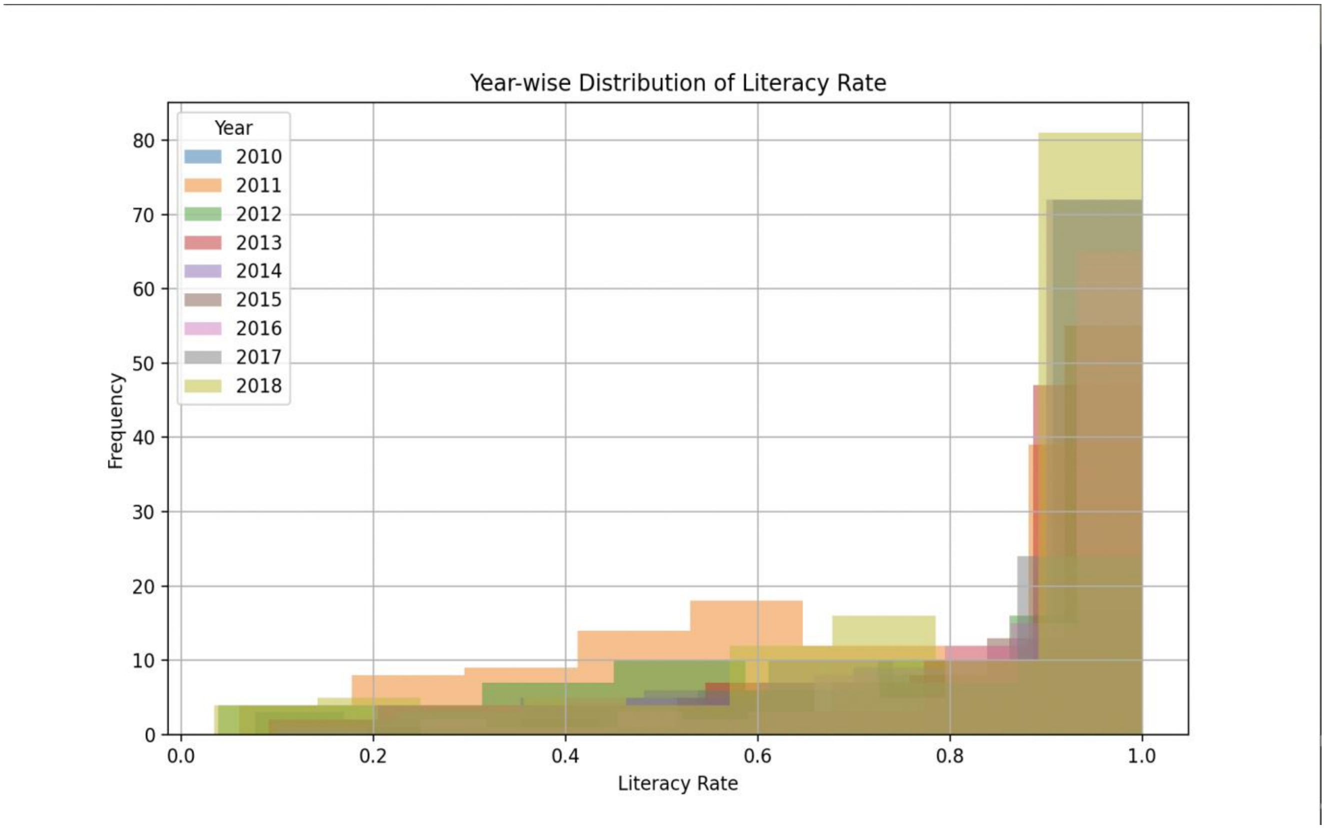
```
# Plotting a histogram of the literacy rate
plt.figure(figsize=(8, 6))
plt.hist(data['Literacy rate'], bins='auto', color='skyblue', edgecolor='black')


plt.xlabel('Literacy Rate')
plt.ylabel('Frequency')
plt.title('Histogram of Literacy Rate Frequency')
plt.grid(True)
plt.show()
```



HISTOGRAM - Histogram of Literacy Rate Frequency
The histogram represents the frequency distribution of literacy rates. It counts the occurrences of different literacy rate ranges, providing an overview of the distribution pattern.

```
#2
```

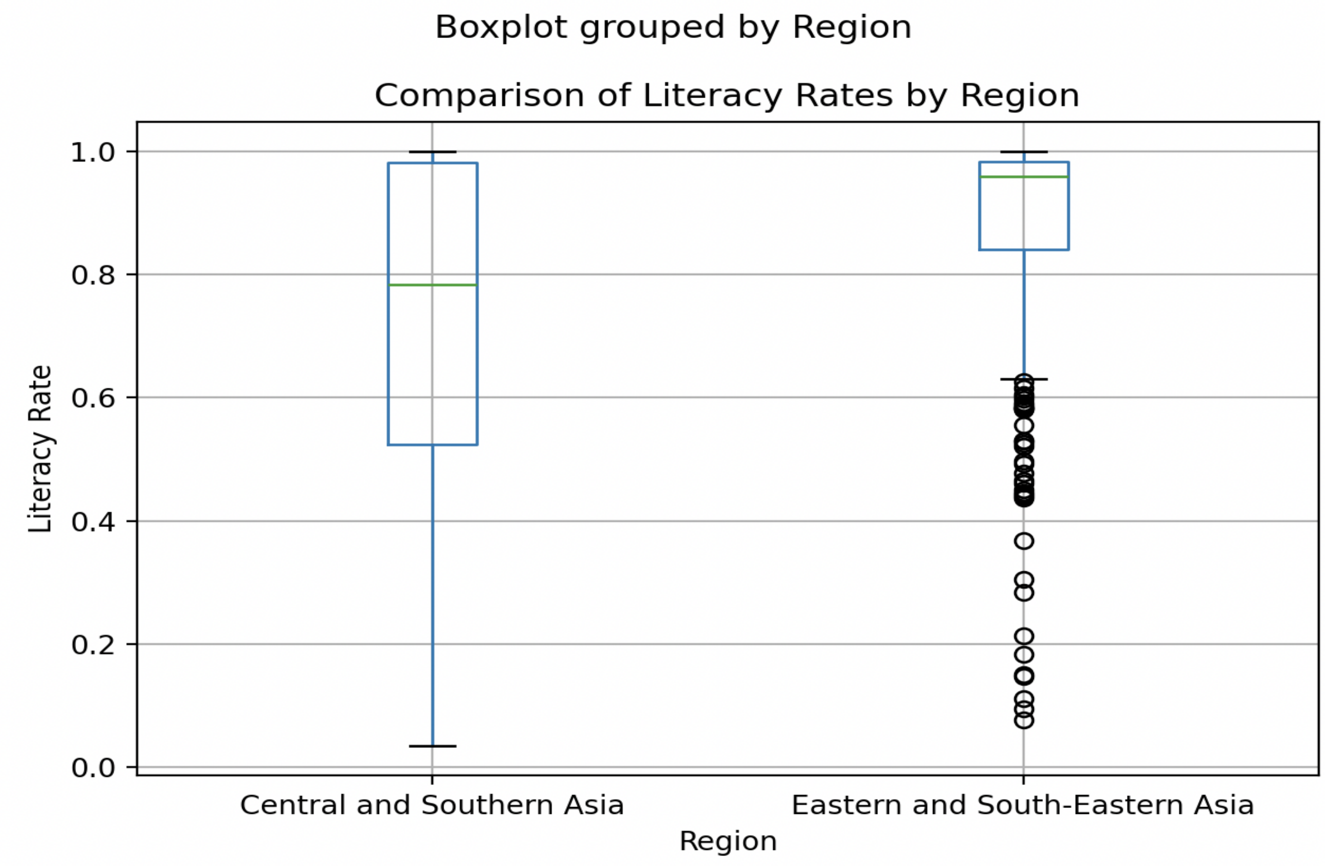
```
plt.figure(figsize=(10, 6))
for year, group_data in data.groupby('Year')['Literacy rate']:
plt.hist(group_data, bins='auto', alpha=0.5, label=str(year))


plt.xlabel('Literacy Rate')
plt.ylabel('Frequency')
plt.title('Year-wise Distribution of Literacy Rate')
plt.legend(title='Year')
plt.grid(True)
plt.show()
```



HISTOGRAM - Year-wise Distribution of Literacy Rate
This histogram visualizes the distribution of literacy rates across different years. It showcases the frequency distribution of literacy rates for each year, facilitating a comparison of literacy rate ranges over time.

BOXPLOT’S

```
#BOXPLOT


# Plotting side-by-side boxplots for literacy rates by region
plt.figure(figsize=(10, 6))
data.boxplot(column='Literacy rate', by='Region')
plt.xlabel('Region')
plt.ylabel('Literacy Rate')
plt.title('Comparison of Literacy Rates by Region')


plt.tight_layout()
plt.show()
```



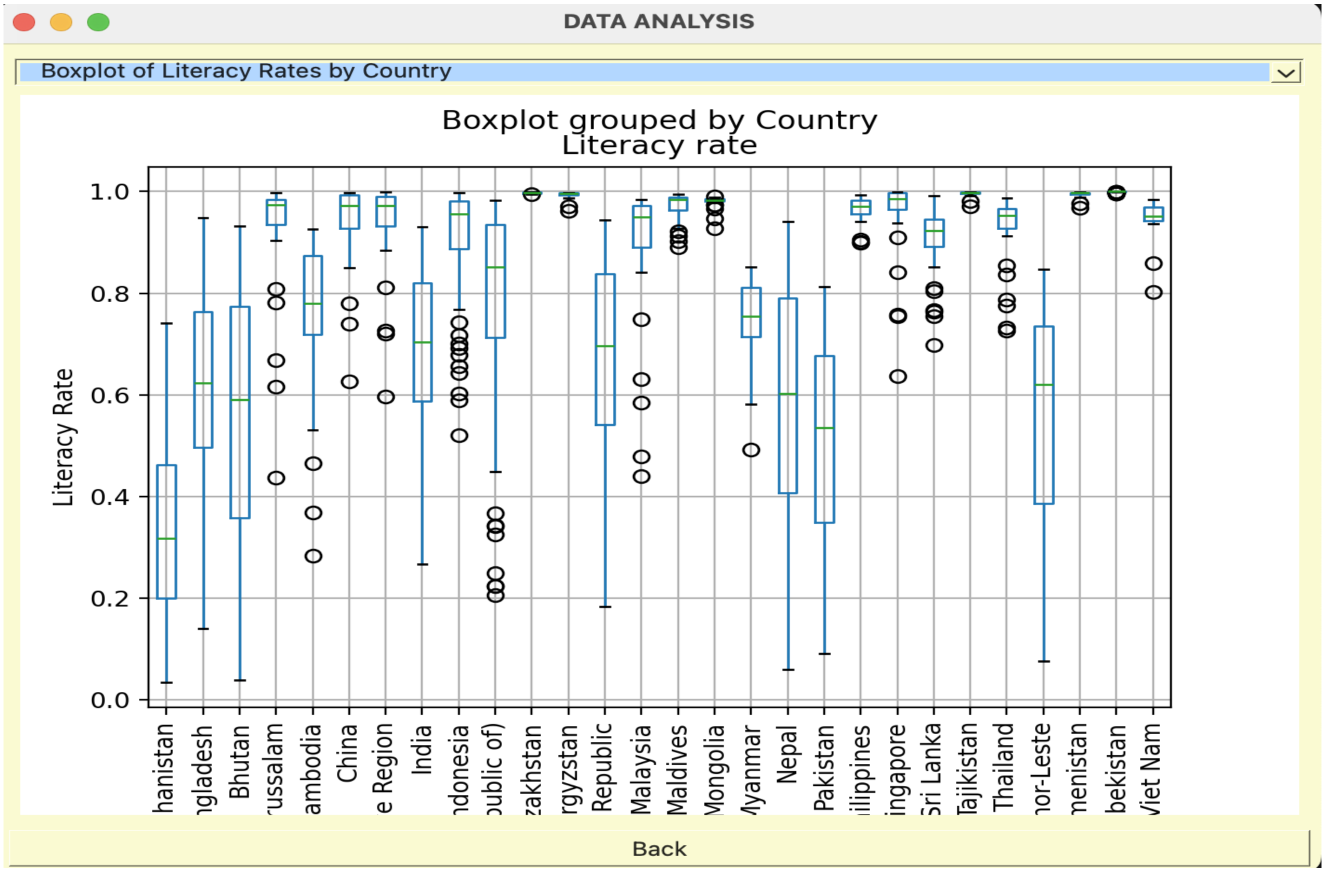
BOXPLOT - Comparison of Literacy Rates by Region
This boxplot compares literacy rates across different regions, displaying their distribution (minimum, maximum, median, quartiles). It helps in understanding the spread and central tendency of literacy rates in each region.

```
#2


# Plotting side-by-side boxplots for literacy rates by gender
plt.figure(figsize=(8, 6))
data.boxplot(column='Literacy rate', by='Gender')
plt.xlabel('Gender')
plt.ylabel('Literacy Rate')
plt.title('Boxplot of Literacy Rate by Gender')
plt.grid(True)
plt.show()
```

BOXPLOT - Boxplot of Literacy Rate by Gender

Similar to the previous graph, this boxplot compares literacy rates based on gender. It illustrates the spread and central tendency of literacy rates among different gender categories.

```
#3


# Plotting side-by-side boxplots for literacy rates by country
plt.figure(figsize=(10, 6))
data.boxplot(column='Literacy rate', by='Country')
plt.xlabel('Country')


plt.ylabel('Literacy Rate')
plt.xticks(rotation=90)
plt.title('Boxplot of Literacy Rate by Country')
plt.grid(True)
plt.show()
```



BOXPLOT - Boxplot of Literacy Rate by Country
This boxplot showcases the distribution of literacy rates for various countries. It represents the spread, central tendency, and outliers of literacy rates among different countries.

SCATTER PLOT

```
#SCATTER PLOT
#plotting scatter plot to check the linearity of data
x = data['Year']
y = data['Literacy rate']
```

```
# Calculate linear regression parameters (slope and intercept) using scipy.stats.linregress
slope, intercept, r_value, p_value, std_err = stats.linregress(x, y)


# Create scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(x, y, alpha=0.5, label='Data')


# Plot the linear regression line
plt.plot(x, slope * x + intercept, color='red', label='Linear Regression')


plt.title('Scatter Plot of Literacy Rate vs Year with Linear Regression')
plt.xlabel('Year')
plt.ylabel('Literacy Rate')
plt.legend()
plt.grid(True)
plt.show()
```

SCATTER PLOT - Scatter Plot of Literacy Rate vs Year with Linear Regression
This scatter plot represents the relationship between 'Year' and 'Literacy Rate.' It includes a linear regression line that indicates the direction and strength of the relationship between these variables.

ANALYSIS OF LINEAR RELATIONSHIP USING CO-RELATION

```
# Display the correlation coefficient (r-value) to assess linearity
print(f"Correlation Coefficient (r-value): {r_value}")
print(f"Correlation Coefficient value depicts that there is weak or no linear relationship")

# Load the dataset (replace 'file_path' with the path to your CSV file)

file_path = '/Users/syedadaniya/Desktop/LR.csv'

df = pd.read_csv(file_path)
```

Finding Auto-Corelation for Lags within the Literacy rate

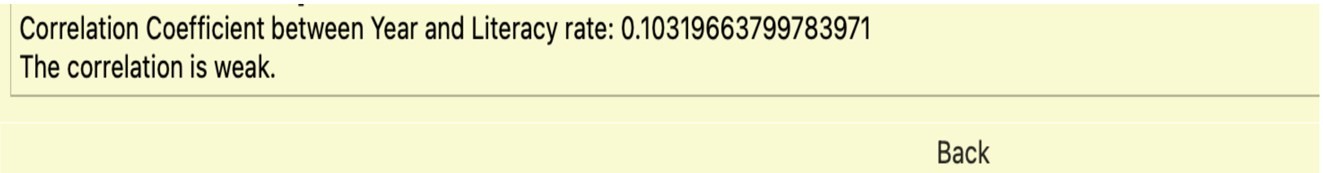
```
import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.graphics.tsaplots import plot_acf



# Load the dataset (replace 'file_path' with the path to your CSV file)



# Assuming 'Literacy rate' is a time series data column and 'Year' is the time index

# Set 'Year' as the index of the DataFrame (assuming it represents time)

df['Year'] = pd.to_datetime(df['Year'], format='%Y') # Convert 'Year' to datetime if needed

df.set_index('Year', inplace=True)



# Compute and plot autocorrelation function (ACF)

plt.figure(figsize=(12, 6))

plot_acf(df['Literacy rate'], lags=20) # Adjust 'lags' parameter as needed

plt.xlabel('Lag')

plt.ylabel('Autocorrelation')

plt.title('Autocorrelation Function (ACF) for Literacy Rate')

plt.show()
```

Interpreting the ACF plot assists in understanding temporal dependencies or lags within the 'Literacy rate' variable. Significant correlations at specific lags may indicate patterns or influences from past observations that can be useful for forecasting or understanding the behavior of the variable over time.

OUTPUT

Correlation Coefficient (r-value): 0.10319663799783971

Correlation Coefficient value depicts that there is weak or no linear relationship



```
Correlation Coefficient between Year and Literacy rate: 0.10319663799783971
The correlation is weak.
```

DESCRIPTIVE STATISTICAL MEASURES

```
#DESCRIPTIVE STATISTICAL MEASURES


# Calculate Mean, Median, and Mode
mean = data['Literacy rate'].mean()
median = data['Literacy rate'].median()


# Measures of Dispersion (Variability)
```

```python
data_min = data['Literacy rate'].min()

data_max = data['Literacy rate'].max()

data_range = data_max - data_min

variance = data['Literacy rate'].var()

std_deviation = data['Literacy rate'].std()

q1 = data['Literacy rate'].quantile(0.25)

q3 = data['Literacy rate'].quantile(0.75)

iqr = q3 - q1


# Measures of Distribution Shape

skewness = data['Literacy rate'].skew()

# Frequency Distribution

frequency_distribution = data['Literacy rate'].value_counts()


# Measures of Position

quartiles = np.percentile(data['Literacy rate'], [25, 50, 75]) # Quartiles

deciles = np.percentile(data['Literacy rate'], np.arange(10, 100, 10)) # Deciles

percentiles_100 = np.percentile(data['Literacy rate'], np.arange(1, 100, 1)) # Percentiles (1 to 99)


# Displaying the calculated values

print(f"Mean: {mean}")

print(f"Median: {median}")

print(f"Range: {data_range}")
print(f"Variance: {variance}")
print(f"Standard Deviation: {std_deviation}")
print(f"First Quartile (Q1): {q1}")
print(f"Third Quartile (Q3): {q3}")
print(f"IQR: {iqr}")
print(f"Skewness: {skewness}")
print(f"Frequency Distribution: \n{frequency_distribution}")
print(f"Quartiles: {quartiles}")
print(f"Deciles: {deciles}")
print(f"Percentiles (1-99): {percentiles_100}")
```

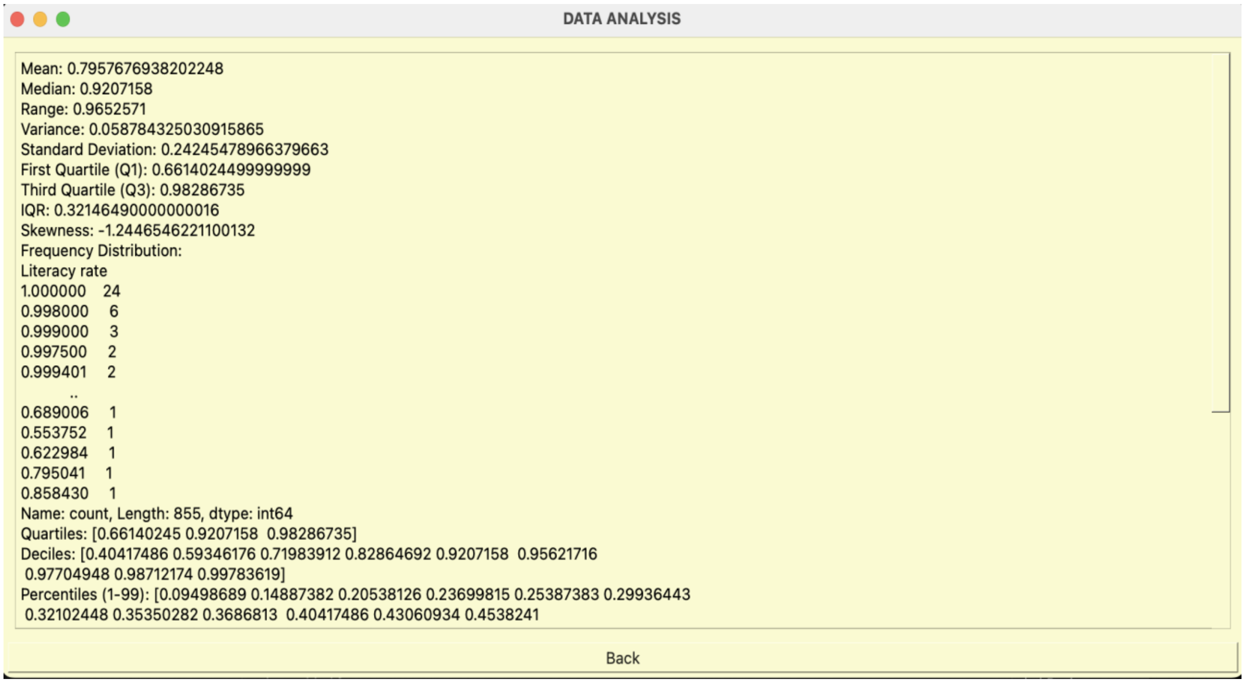
OUTPUT

```
Mean: 0.7957676938202248
Median: 0.9207158
Range: 0.9652571
Variance: 0.058784325030915865
Standard Deviation: 0.24245478966379663
First Quartile (Q1): 0.6614024499999999
Third Quartile (Q3): 0.98286735
IQR: 0.32146490000000016
Skewness: -1.2446546221100132
```

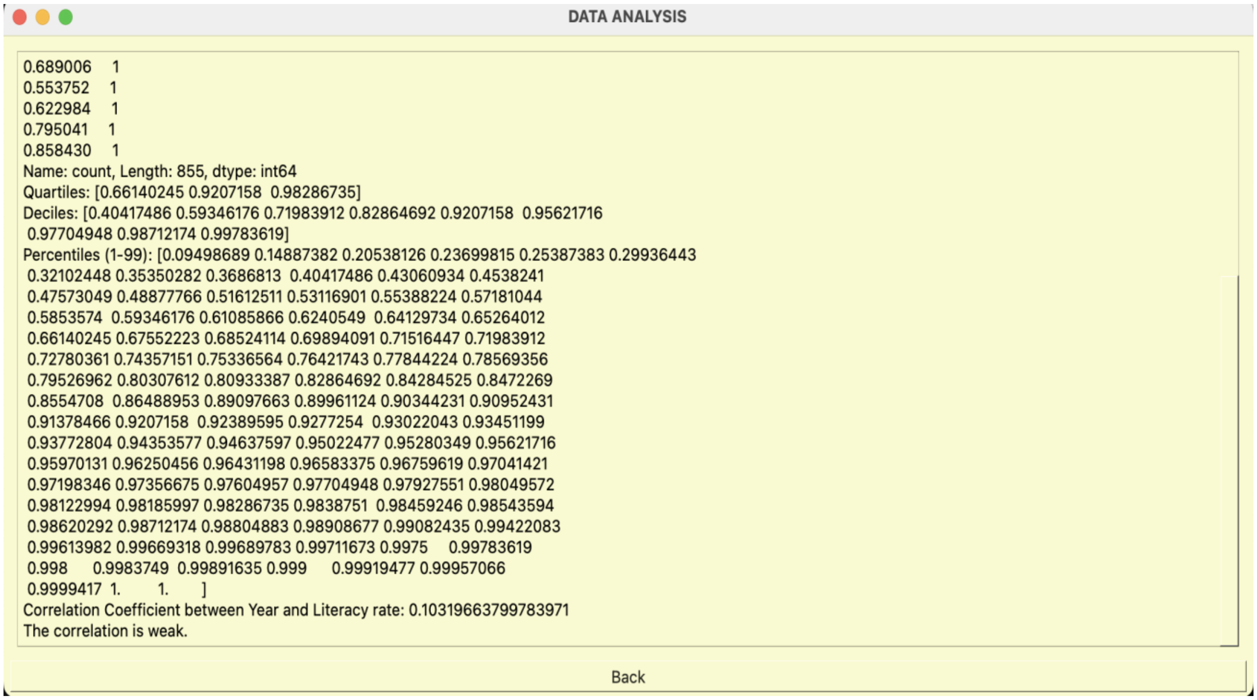
TABULAR REPRESENTATION

```
Frequency Distribution:
```

```
Literacy rate
1.000000    24
0.998000     6
0.999000     3
0.997500     2
0.999401     2
            ..
0.689006     1
0.553752     1
0.622984     1
0.795041     1
0.858430     1
Name: count, Length: 855, dtype: int64
Quartiles: [0.66140245 0.9207158  0.98286735]
Deciles: [0.40417486 0.59346176 0.71983912 0.82864692 0.9207158  0.95621716
0.97704948 0.98712174 0.99783619]
Percentiles (1-99): [0.09498689 0.14887382 0.20538126 0.23699815 0.25387383
0.29936443
0.32102448 0.35350282 0.3686813  0.40417486 0.43060934 0.4538241
0.47573049 0.48877766 0.51612511 0.53116901 0.55388224 0.57181044
0.5853574  0.59346176 0.61085866 0.6240549  0.64129734 0.65264012
0.66140245 0.67552223 0.68524114 0.69894091 0.71516447 0.71983912
0.72780361 0.74357151 0.75336564 0.76421743 0.77844224 0.78569356
0.79526962 0.80307612 0.80933387 0.82864692 0.84284525 0.8472269
0.8554708  0.86488953 0.89097663 0.89961124 0.90344231 0.90952431
0.91378466 0.9207158  0.92389595 0.9277254  0.93022043 0.93451199
0.93772804 0.94353577 0.94637597 0.95022477 0.95280349 0.95621716
0.95970131 0.96250456 0.96431198 0.96583375 0.96759619 0.97041421
0.97198346 0.97356675 0.97604957 0.97704948 0.97927551 0.98049572
0.98122994 0.98185997 0.98286735 0.9838751  0.98459246 0.98543594
0.98620292 0.98712174 0.98804883 0.98908677 0.99082435 0.99422083
0.99613982 0.99669318 0.99689783 0.99711673 0.9975     0.99783619
0.998      0.9983749  0.99891635 0.999      0.99919477 0.99957066
0.9999417  1.         1.        ]
```



```
Mean: 0.7957676938202248
Median: 0.9207158
Range: 0.9652571
Variance: 0.058784325030915865
Standard Deviation: 0.24245478966379663
First Quartile (Q1): 0.6614024499999999
Third Quartile (Q3): 0.98286735
IQR: 0.32146490000000016
Skewness: -1.2446546221100132
Frequency Distribution:
Literacy rate
1.000000    24
0.998000     6
0.999000     3
0.997500     2
0.999401     2
            ..
0.689006     1
0.553752     1
0.622984     1
0.795041     1
0.858430     1
Name: count, Length: 855, dtype: int64
Quartiles: [0.66140245 0.9207158  0.98286735]
Deciles: [0.40417486 0.59346176 0.71983912 0.82864692 0.9207158  0.95621716
 0.97704948 0.98712174 0.99783619]
Percentiles (1-99): [0.09498689 0.14887382 0.20538126 0.23699815 0.25387383 0.29936443
 0.32102448 0.35350282 0.3686813  0.40417486 0.43060934 0.4538241
```



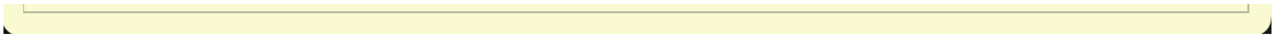
```
0.689006    1
0.553752    1
0.622984    1
0.795041    1
0.858430    1
Name: count, Length: 855, dtype: int64
Quartiles: [0.66140245 0.9207158  0.98286735]
Deciles: [0.40417486 0.59346176 0.71983912 0.82864692 0.9207158  0.95621716
 0.97704948 0.98712174 0.99783619]
Percentiles (1-99): [0.09498689 0.14887382 0.20538126 0.23699815 0.25387383 0.29936443
 0.32102448 0.35350282 0.3686813  0.40417486 0.43060934 0.4538241
 0.47573049 0.48877766 0.51612511 0.53116901 0.55388224 0.57181044
 0.5853574  0.59346176 0.61085866 0.6240549  0.64129734 0.65264012
 0.66140245 0.67552223 0.68524114 0.69894091 0.71516447 0.71983912
 0.72780361 0.74357151 0.75336564 0.76421743 0.77844224 0.78569356
 0.79526962 0.80307612 0.80933387 0.82864692 0.84284525 0.8472269
 0.8554708  0.86488953 0.89097663 0.89961124 0.90344231 0.90952431
 0.91378466 0.9207158  0.92389595 0.9277254  0.93022043 0.93451199
 0.93772804 0.94353577 0.94637597 0.95022477 0.95280349 0.95621716
 0.95970131 0.96250456 0.96431198 0.96583375 0.96759619 0.97041421
 0.97198346 0.97356675 0.97604957 0.97704948 0.97927551 0.98049572
 0.98122994 0.98185997 0.98286735 0.9838751  0.98459246 0.98543594
 0.98620292 0.98712174 0.98804883 0.98908677 0.99082435 0.99422083
 0.99613982 0.99669318 0.99689783 0.99711673 0.9975     0.99783619
 0.998      0.9983749  0.99891635 0.999      0.99919477 0.99957066
 0.9999417  1.         1.        ]
Correlation Coefficient between Year and Literacy rate: 0.10319663799783971
The correlation is weak.
```

APPLYING DISTRIBUTION METHOD

```
#APPLYING UNIFORM DISTRIBUTION


self.literacy_rate_input = QLineEdit()
layout.addWidget(QLabel('Enter the value of Literacy Rate:'))
layout.addWidget(self.literacy_rate_input)


self.lower_bound_input = QLineEdit()
layout.addWidget(QLabel('Enter the lower bound:'))
layout.addWidget(self.lower_bound_input)


self.upper_bound_input = QLineEdit()
layout.addWidget(QLabel('Enter the upper bound:'))
layout.addWidget(self.upper_bound_input)
```

```
# Button to calculate distribution
calculate_button = QPushButton('Calculate')
calculate_button.clicked.connect(self.calculate_uniform_distribution)
layout.addWidget(calculate_button)


# Result display
self.result_text_edit = QTextEdit()
self.result_text_edit.setReadOnly(True)
layout.addWidget(self.result_text_edit)


self.setLayout(layout)


def calculate_uniform_distribution(self):
try:
literacy_rate = float(self.literacy_rate_input.text())
lower_bound = float(self.lower_bound_input.text())
upper_bound = float(self.upper_bound_input.text())


if lower_bound >= upper_bound:
self.result_text_edit.clear()
self.result_text_edit.append('Lower bound must be less than upper bound.')
else:
range_x = upper_bound - lower_bound
p_x = 1 / range_x
e_x = (upper_bound + lower_bound) / 2
variance_x = ((upper_bound - lower_bound) ** 2) / 12
```

```
std_x = np.sqrt(variance_x)


self.result_text_edit.clear()
self.result_text_edit.append(f'For Literacy Rate = {literacy_rate} '
f'and the given range [{lower_bound}, {upper_bound}]:')
self.result_text_edit.append(f'P(x): {p_x}')
self.result_text_edit.append(f'E(x): {e_x}')
self.result_text_edit.append(f'Range: {range_x}')
self.result_text_edit.append(f'Variance(x): {variance_x}')
self.result_text_edit.append(f'Std(x): {std_x}')


except ValueError:
self.result_text_edit.clear()
self.result_text_edit.append('Please enter valid numerical values.')
```



OUTPUT

```
Enter lower bound of literacy rate range (between 0.0 and 1.0): 0.3
Enter upper bound of literacy rate range (between 0.0 and 1.0): 0.45


Statistics for the specified range:
{'Range': '0.3 < X <= 0.45', 'Probability': 0.05730337078651685, 'Expected Value':
0.375, 'Variance': 0.0018750000000000006, 'Standard Deviation':
0.04330127018922194}
```

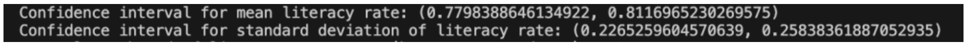
```
04330127018922194}
```

CONFIDENCE INTERVAL FOR DESCRIPTIVE MEASURES

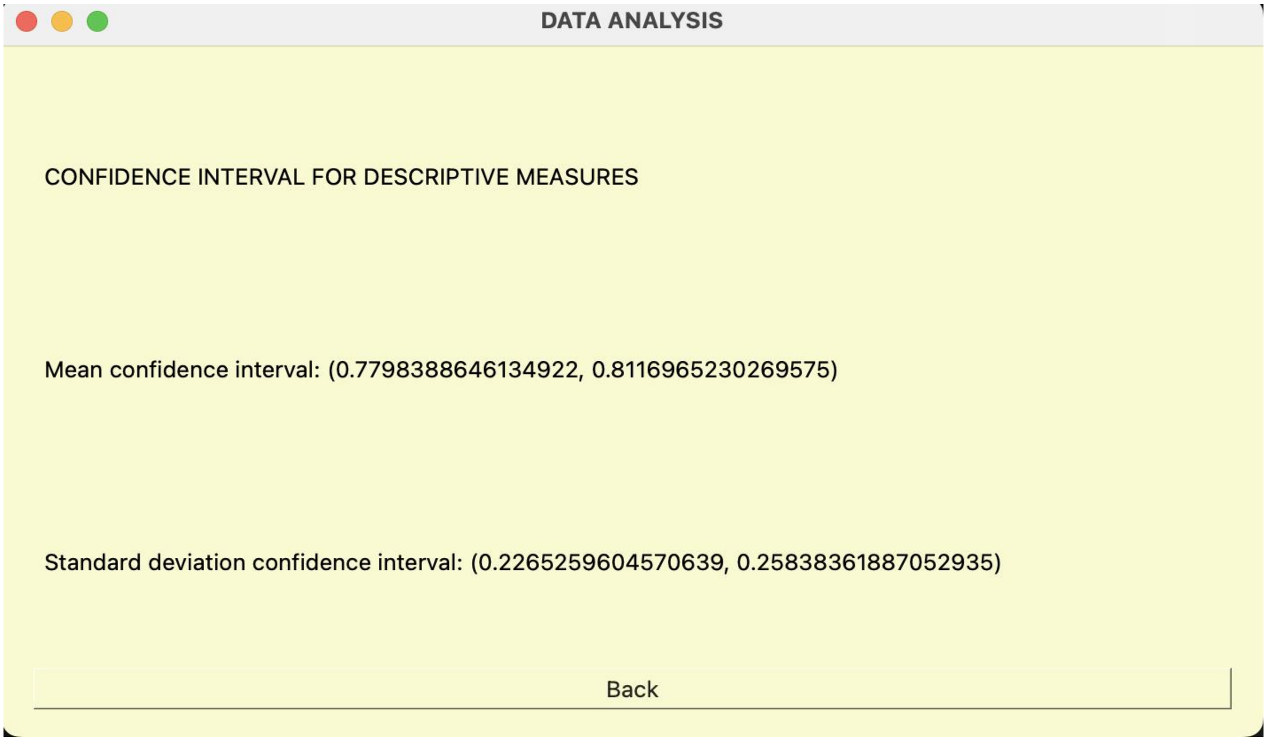
```
#CONFIDENCE INTERVAL FOR DESCRIPTIVE MEASURES
# Calculate mean and standard deviation of literacy rate
mean_literacy = data['Literacy rate'].mean()
std_dev_literacy = data['Literacy rate'].std()


# Set confidence level and calculate z-score (for a 95% confidence interval)
confidence_level = 0.95
z_score = stats.norm.ppf((1 + confidence_level) / 2)


# Calculate the margin of error for mean
margin_of_error_mean = z_score * (std_dev_literacy / np.sqrt(len(data)))


# Calculate confidence intervals for mean and standard deviation
confidence_interval_mean = (mean_literacy - margin_of_error_mean, mean_literacy + margin_of_error_mean)
confidence_interval_std = (std_dev_literacy - margin_of_error_mean, std_dev_literacy + margin_of_error_mean)
```

OUTPUT

```
Confidence interval for mean literacy rate: (0.7798388646134922, 0.8116965230269575)
Confidence interval for standard deviation of literacy rate: (0.2265259604570639, 0.25838361887052935)
```



```
Confidence interval for mean literacy rate: (0.7798388646134922, 0.8116965230269575)
Confidence interval for standard deviation of literacy rate: (0.2265259604570639, 0.25838361887052935)
```



CONFIDENCE INTERVAL FOR DESCRIPTIVE MEASURES

Mean confidence interval: (0.7798388646134922, 0.8116965230269575)

Standard deviation confidence interval: (0.2265259604570639, 0.25838361887052935)

6.Conclusion

In conclusion, this project aims to analyze Literacy rate in Asia over the period of 2010-2018 using statistical methods in Python. By visualizing the data and fitting various models, we hope to identify any patterns or trends in the data. The Desktop app will make the results of the analysis easily accessible and understandable to a wider audience.