GOVERNMENT POLYTECHNIC

NAGAMANGALA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vth Semester Diploma

Artificial Intelligence and Machine Learning (20CS51)

Assignment: 02

NAME: Nithya

ROLL NO: 158cs22031

ASSIGNMENT – WEEK 03

1. Download any two datasets from the internet and perform the following operations.
   a) Analyze the univariate dataset Ex- Mean, Mode, Median, Range, Std, and Variance and perform Univariate tests for the dataset.
   b) Analyze the multivariate of the dataset Ex- co-variance, co-relation.
   c) Visualize the univariate and multivariate with various plots.
   d) Push the code to your GitHub Repository.

Important Note:

1. **Last Date for Submission:31-07-2024.**
2. **Everyone must perform the above operation using datasets (Previously used).**
3. **Submit the report to the email aimlgptn@gmail.com**

1. Download any two datasets from the internet and perform the following operations.

a) Analyze the univariate dataset Ex- Mean, Mode, Median, Range, Std, and Variance and perform Univariate tests for the dataset.

```
import pandas as pd

data=pd.read_csv('/content/people (1).csv')

df=pd.read_csv('/content/Country-data.csv')
```

Mean:

```
data.mean(numeric_only=True)

df.mean(numeric_only=True)
```

Output:

```
Height(cm)    166.15

Weight(kg)     74.35

dtype: float64



child_mort      38.270060

exports        41.108976

health         6.815689

imports        46.890215

income      17144.688623

inflation       7.781832

life_expec      70.555689

total_fer       2.947964

gdpp        12964.155689

dtype: float64
```

Median:

```
data.median(numeric_only=True)

df.median(numeric_only=True)
```

Output:

```
Height(cm)    164.0

Weight(kg)    64.0

dtype: float64


child_mort     19.30

exports        35.00

health         6.32

imports        43.30

income         9960.00

inflation      5.39

life_expec     73.10

total_fer      2.41

gdpp           4660.00

dtype: float64
```

Mode:

```
data.mode(numeric_only=True)

df.mode(numeric_only=True)
```

Output:

```
Height(cm)   Weight(kg)

0            160          64



child_mort   exports   health   imports   income   inflation   life_expec   total_fer   gdpp

0            4.5       23.8     5.07      19.4     1390.0      16.6         70.4        1.57      1310.0

1            NaN       35.0     5.20      23.8     12700.0     NaN          74.5        1.92      NaN

2            NaN       NaN      NaN       26.8     15300.0     NaN          76.3        NaN       NaN

3            NaN       NaN      NaN       28.0     15400.0     NaN          76.4        NaN       NaN

4            NaN       NaN      NaN       28.5     20100.0     NaN          79.5        NaN       NaN

5            NaN       NaN      NaN       37.4     21100.0     NaN          80.4        NaN       NaN

6            NaN       NaN      NaN       42.1     28300.0     NaN          NaN         NaN       NaN

7            NaN       NaN      NaN       43.0     28700.0     NaN          NaN         NaN       NaN

8            NaN       NaN      NaN       43.3     29600.0     NaN          NaN         NaN       NaN

9            NaN       NaN      NaN       46.2     36200.0     NaN          NaN         NaN       NaN

10           NaN       NaN      NaN       49.6     41100.0     NaN          NaN         NaN       NaN

11           NaN       NaN      NaN       51.3     NaN         NaN          NaN         NaN       NaN

12           NaN       NaN      NaN       57.5     NaN         NaN          NaN         NaN       NaN

13           NaN       NaN      NaN       58.9     NaN         NaN          NaN         NaN       NaN

14           NaN       NaN      NaN       62.9     NaN         NaN          NaN         NaN       NaN

15           NaN       NaN

             NaN       63.6     NaN       NaN      NaN         NaN          NaN
```

Range:

```
data.max(numeric_only=True)-data. min(numeric_only=True)

df.max(numeric_only=True)-df. min(numeric_only=True)
```

Output:

```
Height(cm)    46
Weight(kg)    91
dtype: int64

child_mort      205.4000
exports         199.8910
health           16.0900
imports         173.9341
income       124391.0000
inflation       108.2100
life_expec       50.7000
total_fer         6.3400
gdpp         104769.0000
dtype: float64
```

Variance:

```
data.var(numeric_only=True)

df. var(numeric_only=True)
```

Output:

```
Height(cm)    144.450000
Weight(kg)    883.081579
dtype: float64

child_mort    1.626423e+03
exports       7.514183e+02
health        7.545116e+00
imports       5.861042e+02
income        3.716439e+08
inflation     1.117398e+02
life_expec    7.908851e+01
total_fer     2.291734e+00
gdpp          3.359414e+08
dtype: float64
```

Standard deviation:

```
data. std(numeric_only=True)
```

```
df. std(numeric_only=True)
```

Output:

```
Height(cm)    12.018735

Weight(kg)    29.716689

dtype: float64


child_mort      40.328931

exports        27.412010

health         2.746837

imports        24.209589

income       19278.067698

inflation      10.570704

life_expec      8.893172

total_fer       1.513848

gdpp        18328.704809

dtype: float64
```

Univariate tests

T-test:

```
import pandas as pd

import scipy.stats as stats

df = pd.read_csv('/content/Country-data.csv')

df.head()

imports1 = df[df['imports'] == 'imports1']['child_mort']

income2 = df[df['income'] == 'income2']['child_mort']

t_stat, p_value = stats.ttest_ind(imports1,income2)

print("T-test Results:")

print("T-statistic:", t_stat)
```

Output:

```
T-test Results:

T-statistic: nan
```

Chi-square test:

```
observed_values = pd.crosstab(df['child_mort'],df['life_expec'])
```

```
chi2,p,dof,expected = stats.chi2_contingency(observed_values)

print("Chi-square Test Results:")

print("Chi-square statistic:",chi2)
```

Output:

```
Chi-square Test Results:

Chi-square statistic: 18031.36111111111
```

Anova test:

```
healths = [df[df['health'] == group]['income'] for group in df['health'].unique()]

f_stat, p_value = stats.f_oneway(*healths)

print("ANOVA Test Results:")

print("F-statistic:", f_stat)
```

Output:

```
ANOVA Test Results:

F-statistic: 2.7949074494831665
```

b) Analyze the multivariate of the dataset Ex- co-variance,

co-relation.

Co-variance:

```
data.cov(numeric_only=True)

df. cov(numeric_only=True)
```

Output:

```
             Height(cm)    Weight(kg)

Height(cm)  144.450000    231.260526

Weight(kg)  231.260526    883.081579
```

```
            child_mort          exports     health      imports     income      inflation   life_expec  total_fer   gdpp

child_mort  1626.422713         -351.651128             -22.199943  -124.201982             -4.076360e+05           122.893627 -318.008262
51.801162   -3.570463e+05

exports     -351.651128         751.418298 -8.614534    489.350622 2.730946e+05             -31.090078  77.110598   -13.279671  2.103785e+05

health      -22.199943  -8.614534  7.545116     6.365141    6.861669e+03            -7.415093   5.146808    -0.817828   1.741797e+04

imports     -124.201982         489.350622 6.365141     586.104198 5.712872e+04             -63.208898  11.710284   -5.829066   5.125005e+04

income      -407635.982270      273094.598023           6861.669071             57128.721588            3.716439e+08            -30110.122438
104916.785517           -14645.727927           3.164430e+08

inflation   122.893627 -31.090078  -7.415093    -63.208898  -3.011012e+04           111.739781 -22.533965  5.071509     -4.294042e+04
```

```
life_expec   -318.008262            77.110598   5.146808    11.710284   1.049168e+05           -22.533965  79.088507   -10.243585  9.781472e+04

total_fer    51.801162   -13.279671  -0.817828   -5.829066   -1.464573e+04          5.071509    -10.243585  2.291734    -1.262233e+04

gdpp         -357046.306154         210378.470377          17417.971217           51250.050217           3.164430e+08           -42940.421636
97814.722603             -12622.333657          3.359414e+08
```

Co-relation:

data.corr(numeric_only=True)

data.corr(numeric_only=True)

Output:

```
            Height(cm)   Weight(kg)

Height(cm)   1.000000    0.647504

Weight(kg)   0.647504    1.000000
```

```
            child_mort   exports     health      imports     income      inflation   life_expec  total_fer   gdpp

child_mort   1.000000    -0.318093   -0.200402   -0.127211   -0.524315   0.288276    -0.886676   0.848478    -0.483032

exports      -0.318093   1.000000    -0.114408   0.737381    0.516784    -0.107294   0.316313    -0.320011   0.418725

health       -0.200402   -0.114408   1.000000    0.095717    0.129579    -0.255376   0.210692    -0.196674   0.345966

imports      -0.127211   0.737381    0.095717    1.000000    0.122406    -0.246994   0.054391    -0.159048   0.115498

income       -0.524315   0.516784    0.129579    0.122406    1.000000    -0.147756   0.611962    -0.501840   0.895571

inflation    0.288276    -0.107294   -0.255376   -0.246994   -0.147756   1.000000    -0.239705   0.316921    -0.221631

life_expec   -0.886676   0.316313    0.210692    0.054391    0.611962    -0.239705   1.000000    -0.760875   0.600089

total_fer    0.848478    -0.320011   -0.196674   -0.159048   -0.501840   0.316921    -0.760875   1.000000    -0.454910

gdpp         -0.483032   0.418725    0.345966    0.115498    0.895571    -0.221631   0.600089    -0.454910   1.000000
```

c) Visualize the univariate and multivariate with various plots.

Univariate plots:

```
import pandas as pd

df=pd.read_csv('/content/people (1).csv')

import matplotlib.pyplot as plt

import seaborn as sns

plt.figure(figsize=(20, 15))
```

```
<Figure size 2000x1500 with 0 Axes>

<Figure size 2000x1500 with 0 Axes>
```

Bar plot:

```
plt.subplot(5, 5, 6)

plt.bar(df['Height(cm)'], df['Weight(kg)'])

plt.title("bar plot")

plt.xlabel('Height(cm)')

plt.ylabel('Weight(kg)')

plt. show()
```

Box plot:

```
plt.subplot(5, 5, 6)

plt.boxplot(df['Height(cm)'])

plt.title("box plot")

plt.xlabel('Height(cm)')

plt.ylabel('Weight(kg)')

plt. show()
```

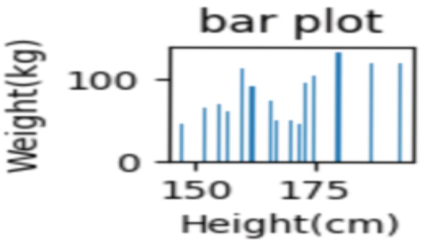
Histogram:

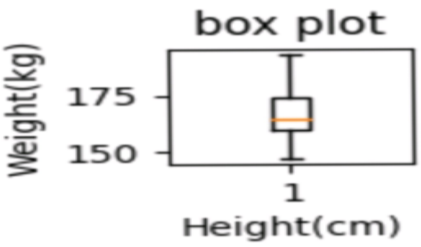
```
plt.subplot(5, 5, 6)

plt.hist(df['Height(cm)'], color='green')

plt.title("histogram")

plt.xlabel('Height(cm)')

plt.ylabel('Weight(kg)')

plt. show()
```

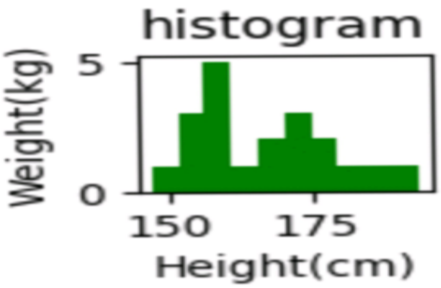
Violin plot:

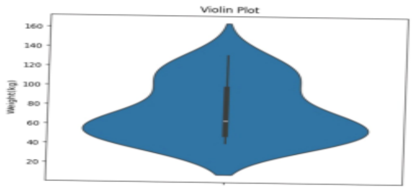
```
sns.violinplot(y=df['Weight(kg)'])

plt.title('Violin Plot')

plt.show()
```

Output:









```
import pandas as pd

df=pd.read_csv('/content/Country-data.csv')

import matplotlib.pyplot as plt

import seaborn as sns

plt.figure(figsize=(20, 15))
```

```
<Figure size 2000x1500 with 0 Axes>

<Figure size 2000x1500 with 0 Axes>
```

Bar plot:

```
plt.subplot(2, 3, 2)
plt.bar(df['child_mort'], df['health'])
plt.title("bar plot")
plt.xlabel('chil_mort')
plt.ylabel('health')
plt. show()
```

Box plot:

```
plt.subplot(2, 3, 6)
plt.boxplot(df['exports'])
plt.title("box plot")
plt.xlabel('exports')
plt.ylabel('imports')
plt. show()
```

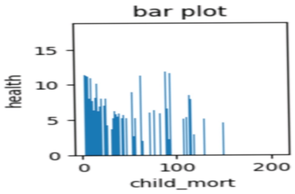
Histogram:

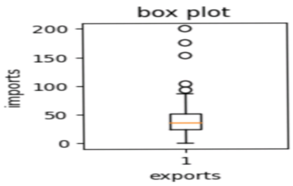
```
plt.subplot(2, 3, 6)
plt.hist(df['exports'], color='red')
plt.title("histogram")
plt.xlabel('exports')
plt.ylabel('health')
plt. show()
```

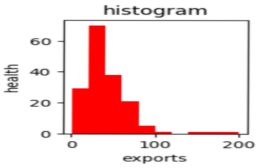
Violin plot:

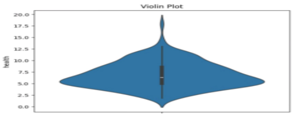
```
sns.violinplot(y=df['export'])
plt.title('Violin Plot')
plt.show()
```

Output:









Mulivariate plots:

```
import pandas as pd

df=pd.read_csv('/content/people (1).csv')

import matplotlib.pyplot as plt

import seaborn as sns

plt.figure(figsize=(20, 15))
```

```
<Figure size 2000x1500 with 0 Axes>

<Figure size 2000x1500 with 0 Axes>
```

Box plot:

```
sns.boxplot(x="Height(cm)", y="Weight(kg)", data=df)

plt.show()
```

Joint plot:

```
sns.jointplot(x="Weight(kg)", y="Height(cm)", data=df)

plt.show()
```

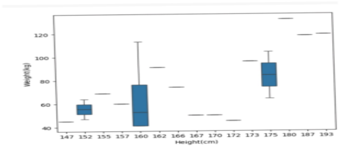
Bar plot:

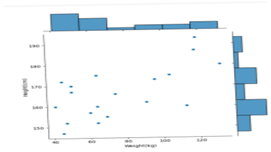
```
sns.barplot(x = 'Weight(kg)', y = 'Height(cm)', data = df)
```

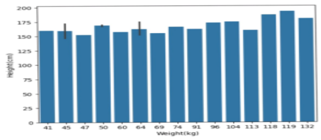
Scatter plot:

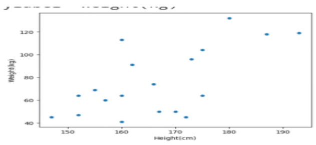
```
sns.scatterplot(x='Height(cm)',y='Weight(kg)',data = df)
```

Output:









```
import pandas as pd

data=pd.read_csv('/content/Country-data.csv')

import matplotlib.pyplot as plt

import seaborn as sns

plt.figure(figsize=(20, 15))
```

```
<Figure size 2000x1500 with 0 Axes>

<Figure size 2000x1500 with 0 Axes>
```

Box plot:

```
sns.boxplot(x="imports", y="health",data=data)

plt.show()
```

Joint plot:

```
sns.jointplot(x="child_mort", y="imports", data=data)

plt.show()
```

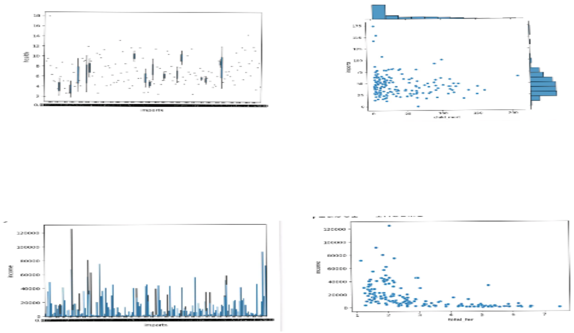
Bar plot:

sns.barplot(x='imports', y='income', data = data)

Scatter plot:

sns.scatterplot(x='total_fer',y='income',data = data)

Output:



d) Push the code to your GitHub Repository.

e)perform any probability calculation

import pandas as pd

file_path = ('/content/people (1).csv')

if 'Height(cm)' in data.columns:

```
    people_above_30 = data[data['Height(cm)'] > 30]

    probability_above_30 = len(people_above_30) / len(data)

    print(f"Probability of a person being older than 30: {probability_above_30:.4f}")

else:

    print("The 'Height(cm)' column is not found in the dataset.")
```

Output:

```
Probability of a person being older than 30: 1.0000
```

2)

```
import pandas as pd

file_path = ('/content/Country-data.csv')

if 'child_mort' in data.columns:

    Country_above_30 = data[data['child_mort'] > 30]

    probability_above_30 = len(Country_above_30) / len(data)

    print(f"Probability of a person being older than 30: {probability_above_30:.4f}")

else:

    print("The 'child_mort' column is not found in the  dataset.")
```

Outputs :

```
The 'child_mort' column is not found in the dataset.
```