Aarjav Jain(21BIT0466)

```
#import necessary libraries

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


#Load the Dataset

df=pd.read_csv("/content/penguins_size.csv")
df.head()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0
3	Adelie	Torgersen	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0

```
df.shape
```

```
(344, 7)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   species            344 non-null    object 
 1   island             344 non-null    object 
 2   culmen_length_mm   342 non-null    float64
 3   culmen_depth_mm    342 non-null    float64
 4   flipper_length_mm  342 non-null    float64
 5   body_mass_g        342 non-null    float64
 6   sex                334 non-null    object 
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

```
df.isnull().sum()
```

```
species               0
island                0
culmen_length_mm      2
culmen_depth_mm       2
flipper_length_mm     2
body_mass_g           2
sex                  10
dtype: int64
```

```
df.sex.value_counts()
```

```
MALE      169
FEMALE    165
Name: sex, dtype: int64
```

```
df.sex = df.sex.replace(".","MALE")
```

```
df.sex.value_counts()
```

```
MALE      169
FEMALE    165
Name: sex, dtype: int64
```

```
df.sex=df.sex.fillna('MALE')
```

```
df.median()
```

```
<ipython-input-10-6d467abf240d>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future ver
  df.median()
culmen_length_mm       44.45
culmen_depth_mm        17.30
flipper_length_mm     197.00
body_mass_g          4050.00
dtype: float64
```

```
df = df.fillna(df.median())
```

```
<ipython-input-11-a187aa03e3ee>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future ver
  df = df.fillna(df.median())
```

```
df.isnull().sum()
```

```
species              0
island               0
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   species            344 non-null    object 
 1   island             344 non-null    object 
 2   culmen_length_mm   344 non-null    float64
 3   culmen_depth_mm    344 non-null    float64
 4   flipper_length_mm  344 non-null    float64
 5   body_mass_g        344 non-null    float64
 6   sex                344 non-null    object 
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

```
df.corr()
```

```
<ipython-input-14-2f6f6606aa2c>:1: FutureWarning: The default value of num
  df.corr()
```

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	
culmen_length_mm	1.000000	-0.235000	0.655858	
culmen_depth_mm	-0.235000	1.000000	-0.583832	
flipper_length_mm	0.655858	-0.583832	1.000000	
body_mass_g	0.594925	-0.471942	0.871221	

```
df.describe()
```

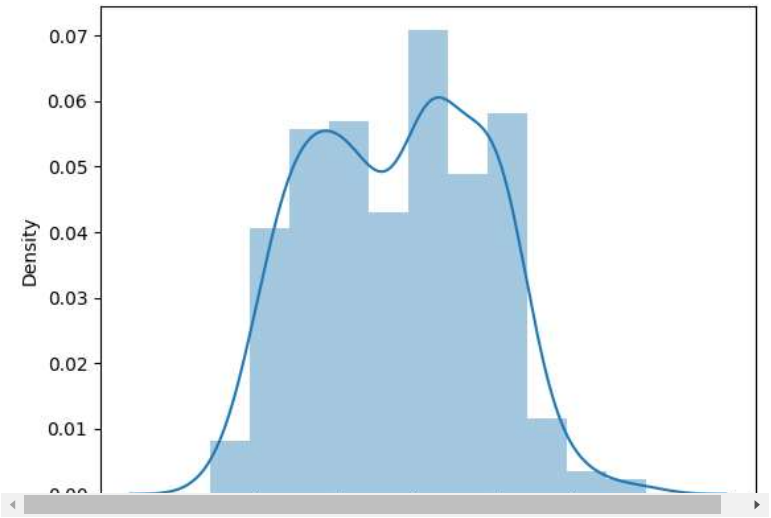
	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
count	344.000000	344.000000	344.000000	344.000000

VISUALISATION

Univariate Analysis

```
sns.distplot(df.culmen_length_mm)
```

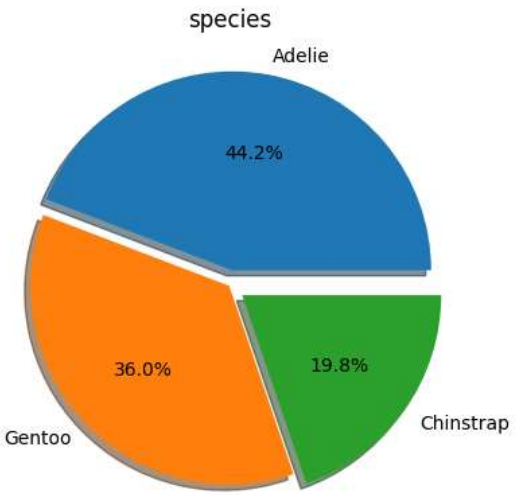
```
<ipython-input-16-24e9b5890c61>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.culmen_length_mm)
<Axes: xlabel='culmen_length_mm', ylabel='Density'>
```



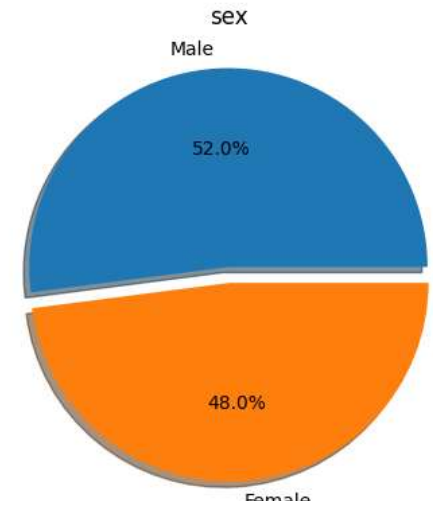

```
sns.distplot(df.flipper_length_mm)
```

```
<ipython-input-17-4c42e92ff055>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0

Please adapt your code to use either `displot` (a figure-level function wi
similar flexibility) or `histplot` (an axes-level function for histograms)

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.flipper_length_mm)
<Axes: xlabel='flipper_length_mm', ylabel='Density'>
```

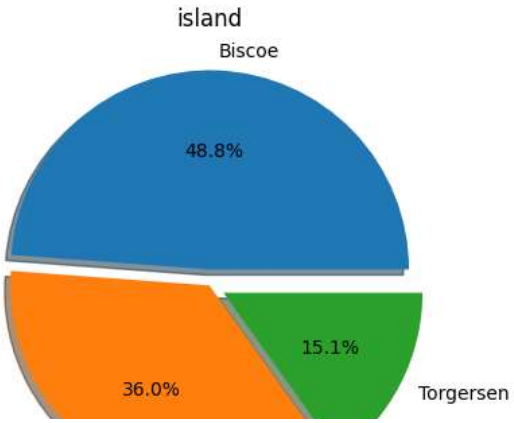
```
plt.pie(df.species.value_counts(),[0.08,0,0.08],labels=['Adelie','Gentoo','Chinstrap'],autopct='%1.1f%%', shadow=True)
plt.title('species')
plt.show()
```




```
plt.pie(df.sex.value_counts(),[0.08,0],labels=['Male','Female'],autopct='%1.1f%%', shadow=True)
plt.title('sex')
plt.show()
```

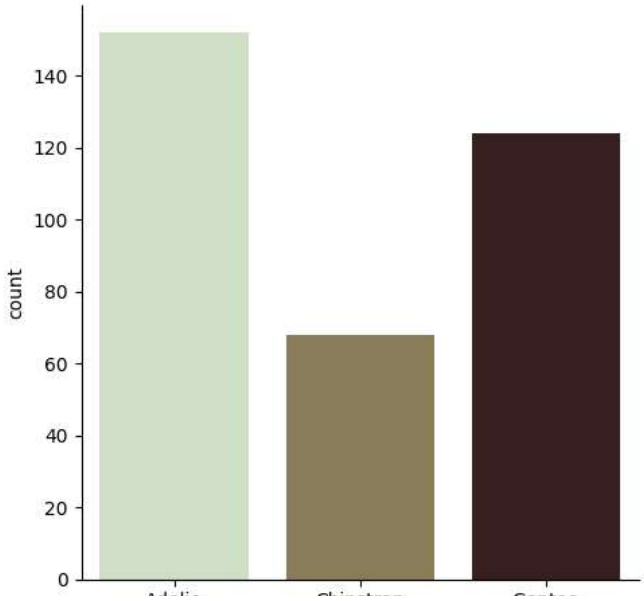



```
plt.pie(df.island.value_counts(),[0.08,0,0.08],labels=['Biscoe','Dream','Torgersen'],autopct='%1.1f%%', shadow=True)
plt.title('island')
plt.show()
```




```
sns.catplot(data=df, x='species',kind='count', palette="ch:.75")
```

```
<seaborn.axisgrid.FacetGrid at 0x7dd7119f8d00>
```



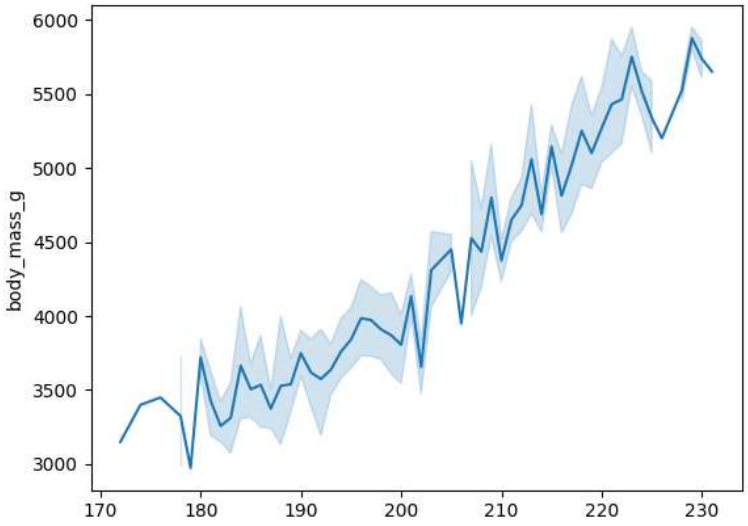
Bivariate Analysis

```
sns.lineplot(x=df.culmen_length_mm, y=df.culmen_depth_mm)
```

```
<Axes: xlabel='culmen_length_mm', ylabel='culmen_depth_mm'>
```

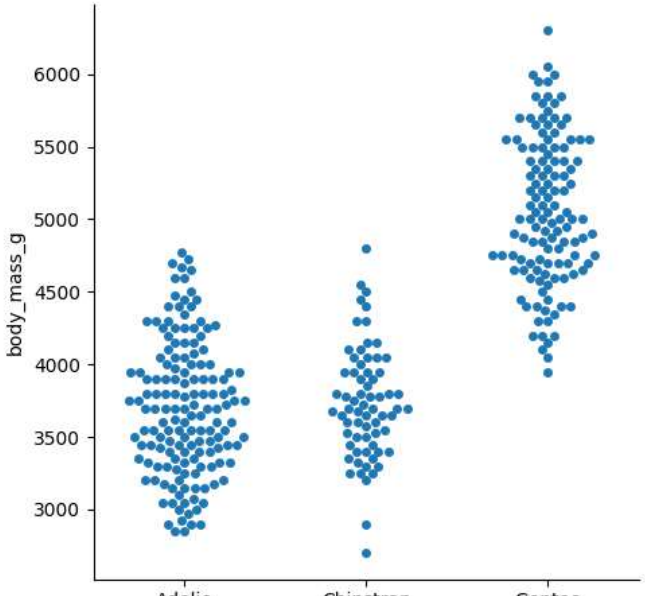
```
sns.lineplot(x=df.flipper_length_mm, y=df.body_mass_g)
```

```
<Axes: xlabel='flipper_length_mm', ylabel='body_mass_g'>
```




```
sns.catplot(data=df, x="species", y="body_mass_g", kind="swarm")
```

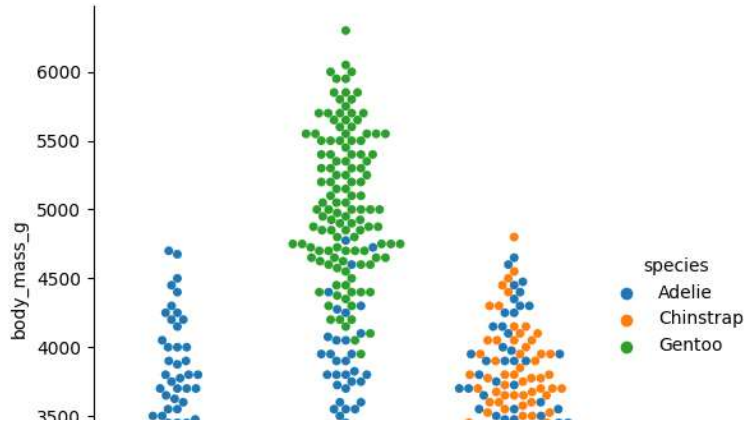
```
<seaborn.axisgrid.FacetGrid at 0x7dd7119b12a0>
```



MULTIVARIATE ANALYSIS

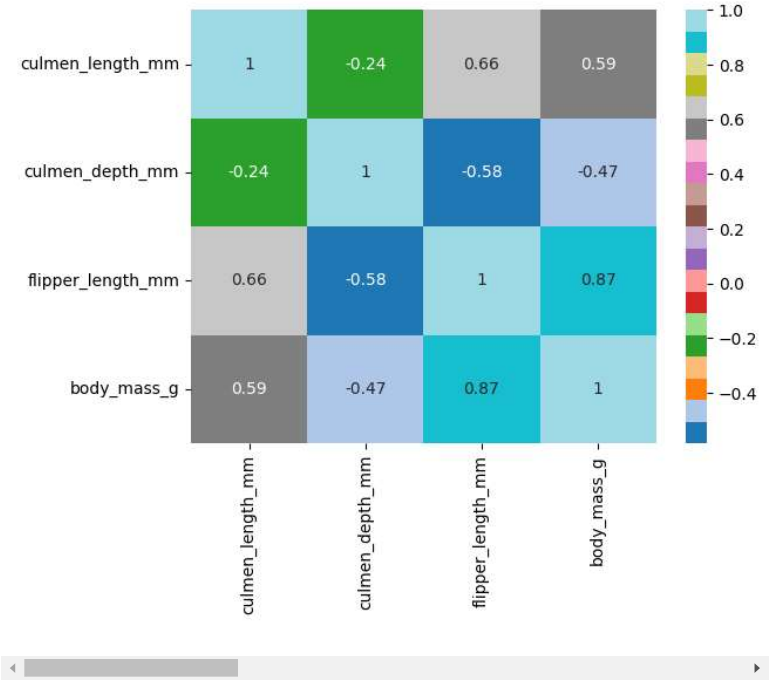
```
sns.catplot(data=df, x="island", y="body_mass_g", hue="species", kind="swarm")
```

```
<seaborn.axisgrid.FacetGrid at 0x7dd715d02320>
```




```
sns.heatmap(df.corr(), annot=True, cmap="tab20")
```

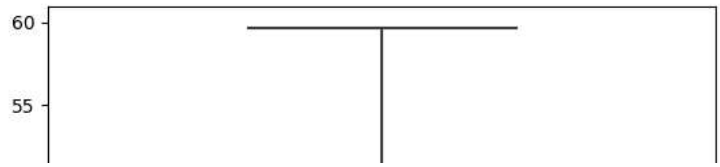
```
<ipython-input-26-056e9d0b5df0>:1: FutureWarning: The default value of num
  sns.heatmap(df.corr(), annot=True, cmap="tab20")
<Axes: >
```



Outliers Detecttion and Replacement

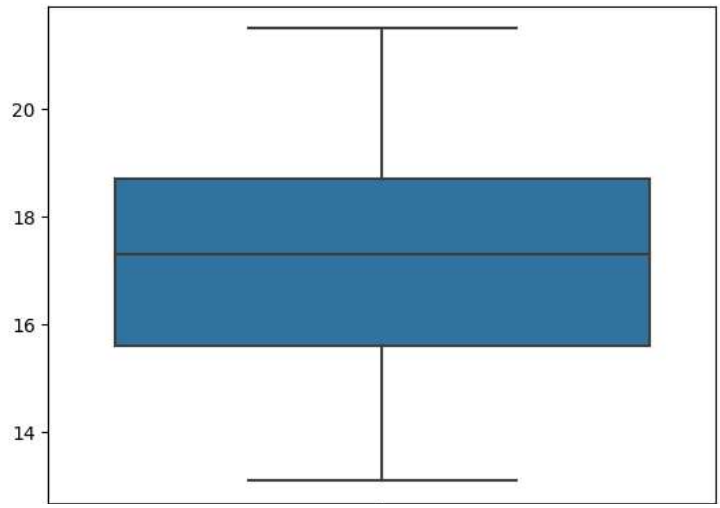
```
sns.boxplot(df['culmen_length_mm']) #No outliers present
```

```
<Axes: >
```



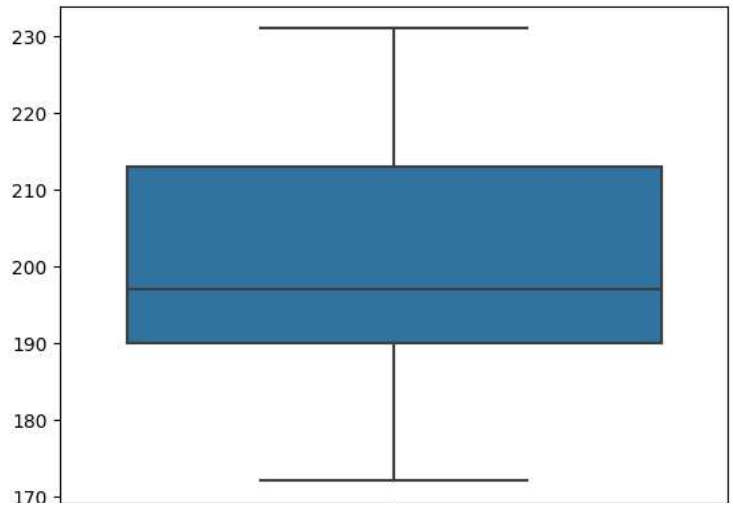

```
sns.boxplot(df['culmen_depth_mm']) #No outliers present
```

```
<Axes: >
```



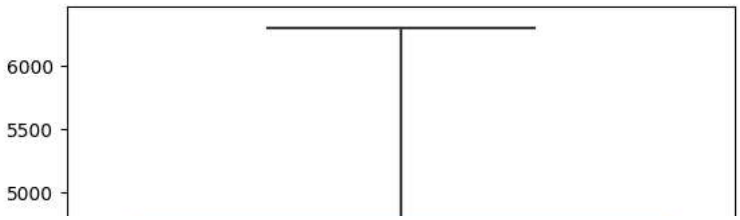

```
sns.boxplot(df['flipper_length_mm']) #No outliers present
```

```
<Axes: >
```

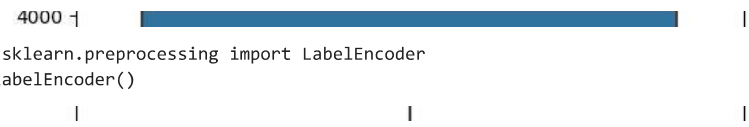



```
sns.boxplot(df['body_mass_g']) #No outliers present
```

```
<Axes: >
```



Performing Label Encoding for categorical values




```
from sklearn.preprocessing import LabelEncoder
le =LabelEncoder()
```

```
df['species']=le.fit_transform(df['species'])
df['sex']=le.fit_transform(df['sex'])
df['island']=le.fit_transform(df['island'])
```

```
df.head()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm
0	0	2	39.10	18.7	181.0
1	0	2	39.50	17.4	186.0
2	0	2	40.30	18.0	195.0
3	0	2	44.45	17.3	197.0

Independent(X) and dependent(Y) variable split

```
y = df.species
```

```
y.head()
```

```
0    0
1    0
2    0
3    0
4    0
Name: species, dtype: int64
```

```
X = df.drop(columns =['species'],axis =1)
X.head()
```

	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass
0	2	39.10	18.7	181.0	375
1	2	39.50	17.4	186.0	380
2	2	40.30	18.0	195.0	325
3	2	44.45	17.3	197.0	405

Scaling

```
from sklearn.preprocessing import MinMaxScaler
scale= MinMaxScaler()
```

```
x_scaled=pd.DataFrame(scale.fit_transform(X),columns=X.columns)
x_scaled.head()
```

	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass
0	1.0	0.254545	0.666667	0.152542	0.2916
1	1.0	0.269091	0.511905	0.237288	0.3055
2	1.0	0.298182	0.583333	0.389831	0.1527
3	1.0	0.449091	0.500000	0.423729	0.3750

Train test split

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x_scaled,y,test_size =0.3,random_state=0)
```

```
x_test.head()
```

	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_ma
141	0.5	0.309091	0.488095	0.254237	0.21
6	1.0	0.247273	0.559524	0.152542	0.25
60	0.0	0.130909	0.452381	0.220339	0.12
249	0.0	0.650909	0.261905	0.813559	0.79

```
x_train.head()
```

	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_ma
219	0.5	0.658182	0.666667	0.440678	0.29
271	0.0	0.596364	0.119048	0.813559	0.72
266	0.0	0.487273	0.095238	0.644068	0.41
335	0.0	0.836364	0.345238	0.983051	0.87

```
y_test.head()
```

```
141    0
6      0
60     0
249    2
54     0
Name: species, dtype: int64
```

```
y_train.head()
```

```
219    1
271    2
266    2
335    2
217    1
Name: species, dtype: int64
```

```
x_test.shape
```

```
(104, 6)
```

```
x_train.shape
```

```
(240, 6)
```

```
from sklearn.linear_model import LinearRegression

model=LinearRegression()

model.fit(x_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
Y_Pred=model.predict(x_test)
Y_Pred
```

```
array([ 2.88574491e-01,  1.10715338e-01,  2.29384016e-01,  2.05896280e+00,
       -6.08913996e-02,  1.86758378e+00,  2.19971521e-01,  1.21457725e+00,
        1.81656167e+00,  1.85783292e-01,  3.27810391e-01,  1.81544855e+00,
        1.47213984e-01,  1.84774774e+00,  1.88251377e+00,  2.91780752e-01,
        1.88664000e-01,  1.76895103e+00,  4.20247376e-01,  1.80530654e-02,
        1.75001096e+00,  1.47256735e-01,  8.95558523e-01,  9.02257546e-02,
        2.01785560e+00,  2.23439175e+00,  1.63536605e+00,  1.10997938e-01,
        2.90478891e-01,  1.64944473e+00,  1.90892006e+00,  7.94562103e-01,
        2.74891660e-02,  2.81019458e-01, -1.24151576e-01,  9.57016545e-01,
        1.98326481e+00,  8.36268849e-01, -3.74758014e-03,  4.52041298e-01,
       -7.25766521e-02,  3.98603764e-01,  5.93981320e-02,  1.88540886e+00,
        1.80596227e+00, -3.59432541e-02,  2.02807113e+00,  1.20664123e+00,
        1.06009313e-01,  2.03655994e+00,  1.68612863e-02,  1.56430761e-01,
        1.64771762e-01,  4.78593048e-01,  1.54053934e-02,  7.13790417e-01,
        1.21612983e-01,  2.17691615e+00,  9.14992887e-01,  2.00316136e+00,
        9.81321715e-01,  1.02823904e+00,  5.54453706e-01,  1.64510967e+00,
        1.70533775e+00,  2.00374506e+00, -1.34013018e-01,  1.60209598e+00,
        1.70265211e+00, -1.20460916e-01,  2.02517569e+00,  1.63653039e+00,
        2.42814897e-01,  8.18375101e-01,  2.27017999e+00,  2.24309406e-01,
        9.37416706e-05,  2.01384491e+00,  9.10887124e-01,  5.81943394e-01,
        4.91555220e-01,  8.14054439e-01,  2.47188347e-01,  6.41225399e-01,
        5.40424758e-01,  1.61840134e+00,  4.88468650e-03,  2.10365250e+00,
        3.51861223e-01,  1.05452735e-01,  2.08933995e+00,  1.61458342e+00,
        3.70123305e-01,  8.58007076e-01,  2.02277857e+00,  1.72611257e+00,
        1.01294002e+00,  1.95934336e+00,  2.39788829e-01,  5.65304663e-01,
        3.07820674e-01,  4.16480566e-02, -3.21384544e-01,  1.95745792e+00])
```

✓ 0s completed at 8:13 PM

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.