Loading The Dataset

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

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

```
df.head()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass
0	Adelie	Torgersen	39.1	18.7	181.0	375
1	Adelie	Torgersen	39.5	17.4	186.0	380
2	Adelie	Torgersen	40.3	18.0	195.0	325
3	Adelie	Torgersen	NaN	NaN	NaN	N
4	Adelie	Torgersen	36.7	19.3	193.0	345

```
df.shape
```

```
(344, 7)
```

Checking NULL Values

```
df.isnull().any()
```

```
species              False
island               False
culmen_length_mm      True
culmen_depth_mm       True
flipper_length_mm     True
body_mass_g           True
sex                   True
dtype: bool
```

```
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
```

Dealing With NULL Values

```
df.culmen_length_mm.median()
```

```
44.45
```

```
df.culmen_depth_mm.median()
```

```
17.3
```

```
df.flipper_length_mm.median()
```

```
197.0
```

```
df.body_mass_g.median()
```

```
4050.0
```

```
df.sex.mode()
```

```
0    MALE
Name: sex, dtype: object
```

```
##Dealing with NULL values

## Below features are type of float therefore we will deal the NULL values with median
df['culmen_length_mm'].fillna(df['culmen_length_mm'].median(),inplace=True)
df['culmen_depth_mm'].fillna(df['culmen_depth_mm'].median(),inplace=True)
df['flipper_length_mm'].fillna(df['flipper_length_mm'].median(),inplace=True)
df['body_mass_g'].fillna(df['body_mass_g'].median(),inplace=True)

## Below one is object type therefore we will deal the NULL values with mode
df['sex'].fillna('MALE',inplace=True)
```

```
df.head()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass
0	Adelie	Torgersen	39.10	18.7	181.0	375
1	Adelie	Torgersen	39.50	17.4	186.0	380
2	Adelie	Torgersen	40.30	18.0	195.0	325
3	Adelie	Torgersen	44.45	17.3	197.0	405
4	Adelie	Torgersen	36.70	19.3	193.0	345

```
df.isnull().any()
```

```
species              False
island               False
culmen_length_mm     False
culmen_depth_mm      False
flipper_length_mm    False
body_mass_g          False
sex                  False
dtype: bool
```

```
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
```

Descriptive Analysis

```
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.describe()
```

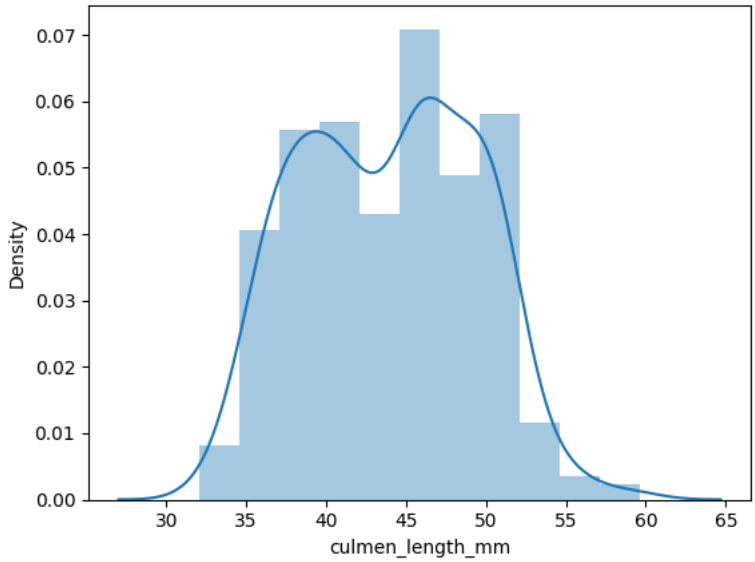
	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
count	344.000000	344.000000	344.000000	344.000000
mean	43.925000	17.152035	200.892442	4200.872093
std	5.443792	1.969060	14.023826	799.696532
25%	39.275000	15.600000	190.000000	3550.000000

Univariate, Bi-Variate, and Multi-Variate Analysis

1.Univariate

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

```
<ipython-input-133-24e9b5890c61>: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'>
```

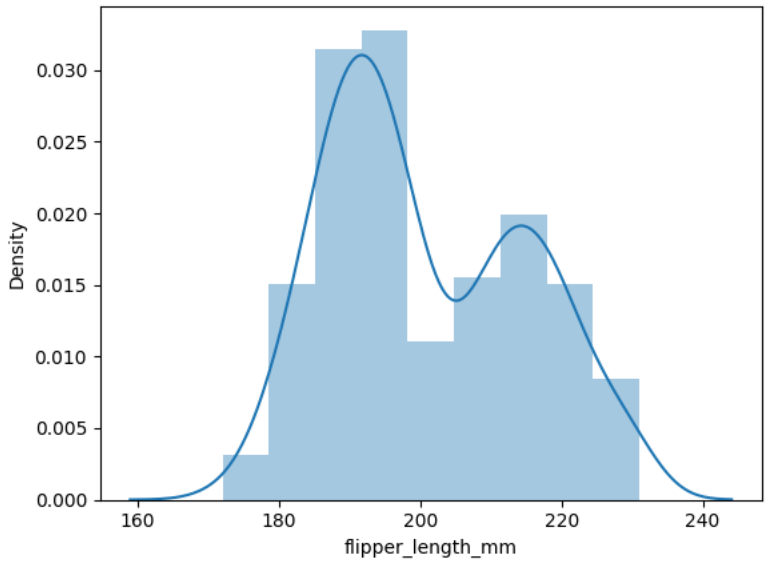



```
sns.distplot(df.culmen_depth_mm)
```

```
<ipython-input-134-4b07ffb4fe44>: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).
```

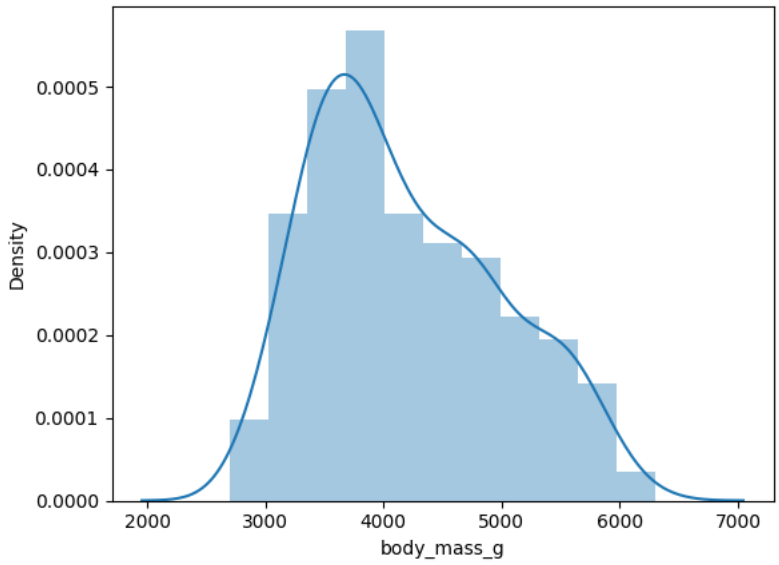
```
sns.distplot(df.flipper_length_mm)
```

```
<ipython-input-135-4c42e92ff055>: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 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.flipper_length_mm)
<Axes: xlabel='flipper_length_mm', ylabel='Density'>
```



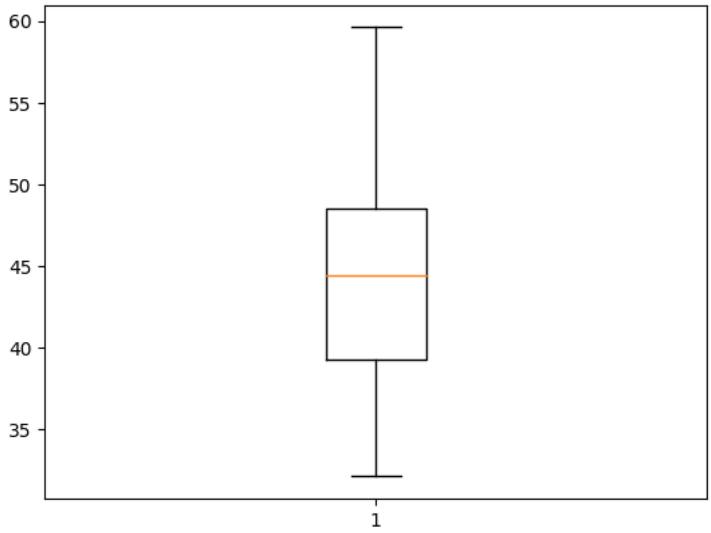

```
sns.distplot(df.body_mass_g)
```

```
<ipython-input-136-176964dae727>: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.body_mass_g)
<Axes: xlabel='body_mass_g', ylabel='Density'>
```



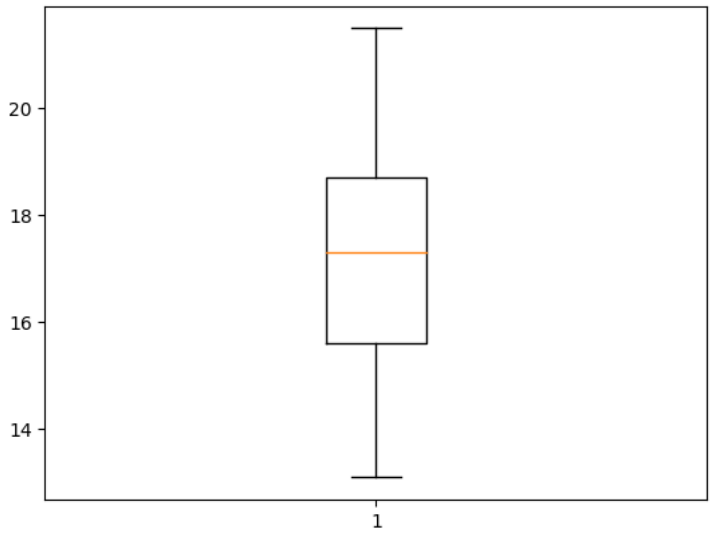

```
plt.boxplot(df.culmen_length_mm)
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x792965cce170>,
  <matplotlib.lines.Line2D at 0x792965cce410>],
 'caps': [<matplotlib.lines.Line2D at 0x792965cce6b0>,
  <matplotlib.lines.Line2D at 0x792965cce950>],
 'boxes': [<matplotlib.lines.Line2D at 0x792965ccded0>],
 'medians': [<matplotlib.lines.Line2D at 0x792965ccebf0>],
 'fliers': [<matplotlib.lines.Line2D at 0x792965ccee90>],
 'means': []}
```




```
plt.boxplot(df.culmen_depth_mm)
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x792965d5c070>,
  <matplotlib.lines.Line2D at 0x792965d5c310>],
 'caps': [<matplotlib.lines.Line2D at 0x792965d5c5b0>,
  <matplotlib.lines.Line2D at 0x792965d5c850>],
 'boxes': [<matplotlib.lines.Line2D at 0x792965d2fd90>],
 'medians': [<matplotlib.lines.Line2D at 0x792965d5caf0>],
 'fliers': [<matplotlib.lines.Line2D at 0x792965d5cd90>],
 'means': []}
```

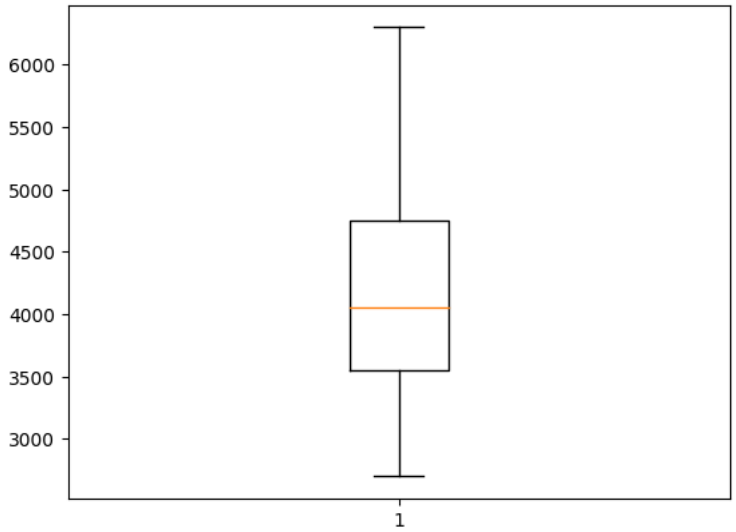



```
plt.boxplot(df.flipper_length_mm)
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x792965bb16f0>,
  <matplotlib.lines.Line2D at 0x792965bb1990>],
 'caps': [<matplotlib.lines.Line2D at 0x792965bb1c30>,
  <matplotlib.lines.Line2D at 0x792965bb1ed0>],
 'boxes': [<matplotlib.lines.Line2D at 0x792965bb1450>],
 'medians': [<matplotlib.lines.Line2D at 0x792965bb2170>],
 'fliers': [<matplotlib.lines.Line2D at 0x792965bb2410>],
 'means': []}
```


```
plt.boxplot(df.body_mass_g)
```

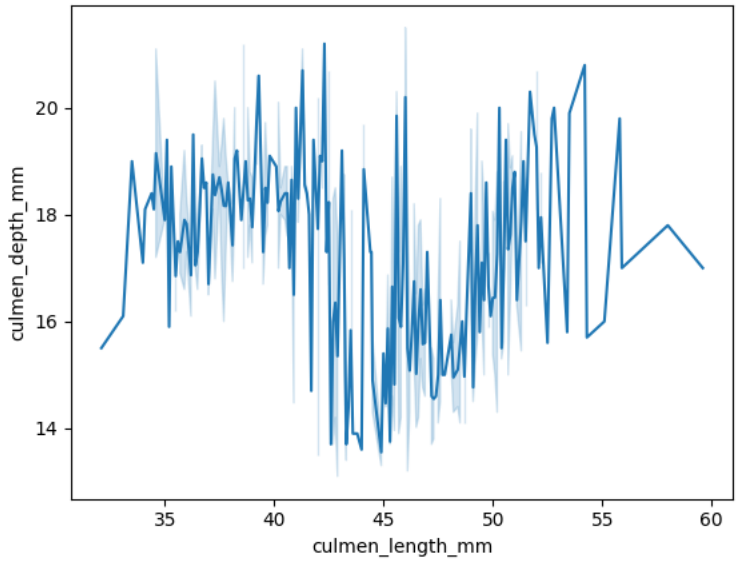
```
{'whiskers': [<matplotlib.lines.Line2D at 0x792965c2caf0>,
  <matplotlib.lines.Line2D at 0x792965c2cd90>],
 'caps': [<matplotlib.lines.Line2D at 0x792965c2d030>,
  <matplotlib.lines.Line2D at 0x792965c2d2d0>],
 'boxes': [<matplotlib.lines.Line2D at 0x792965c2c850>],
 'medians': [<matplotlib.lines.Line2D at 0x792965c2d570>],
 'fliers': [<matplotlib.lines.Line2D at 0x792965c2d810>],
 'means': []}
```



2. Bi-Variate Analysis

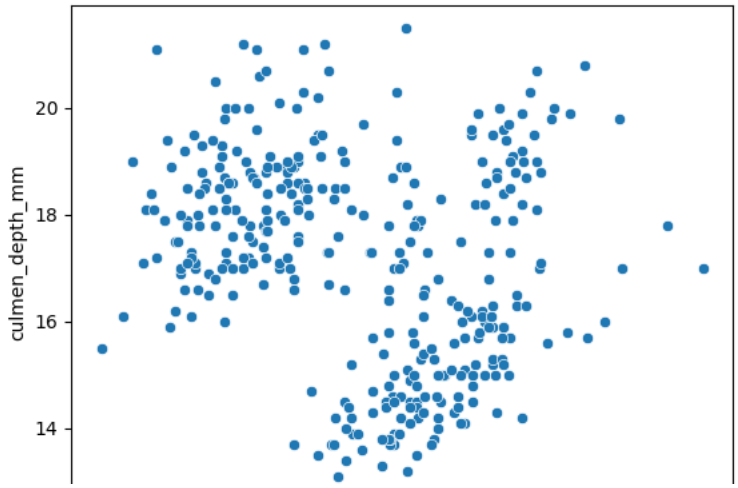
```
sns.lineplot(x=df.culmen_length_mm,y=df.culmen_depth_mm)
```

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



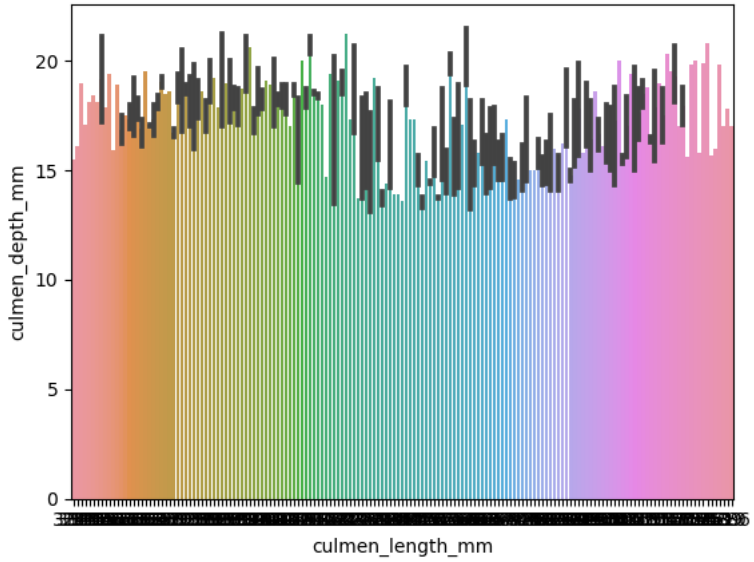

```
sns.scatterplot(x=df.culmen_length_mm,y=df.culmen_depth_mm)
```

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




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

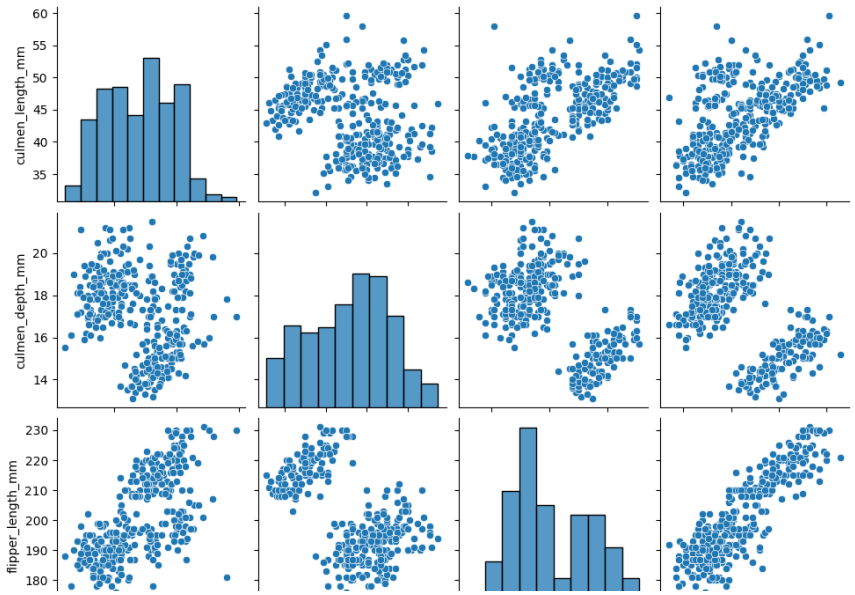
```
<Axes: xlabel='culmen_length_mm', ylabel='culmen_depth_mm'>
```



3. Multi-Variate Analysis

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x79296550bdf0>
```

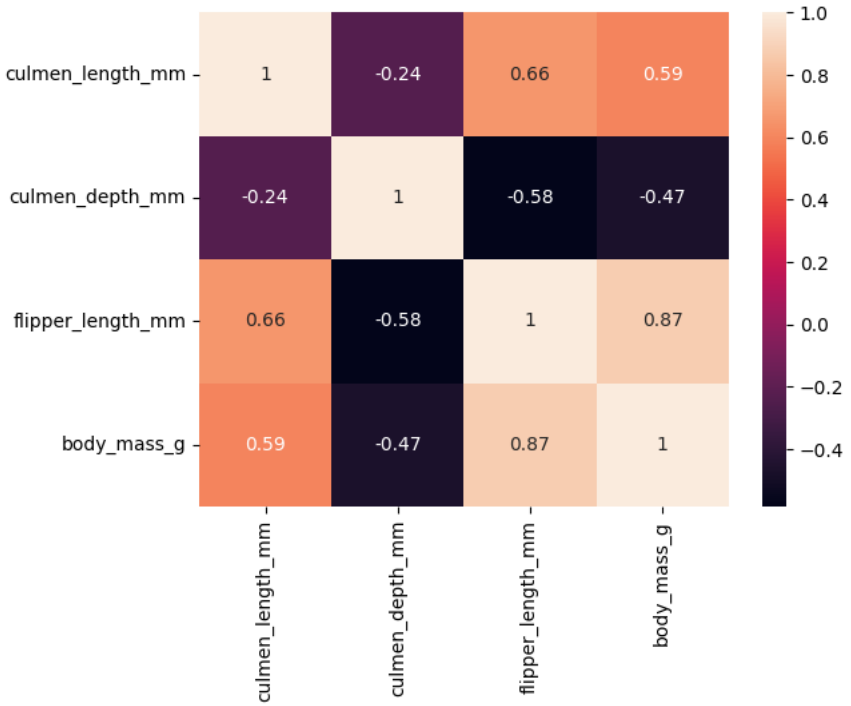


Double-click (or enter) to edit

Co-Relation

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

```
<ipython-input-145-8df7bcac526d>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
  sns.heatmap(df.corr(),annot=True)
<Axes: >
```




```
df.corr().flipper_length_mm.sort_values(ascending=False)
```

```
flipper_length_mm    1.000000
body_mass_g          0.871221
species              0.850819
culmen_length_mm     0.655858
sex                  0.225848
island              -0.562957
culmen_depth_mm     -0.583832
Name: flipper_length_mm, dtype: float64
```

Performing Encoding

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

```
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.head()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.10	18.7	181.0	3750.0	MALE
1	Adelie	Torgersen	39.50	17.4	186.0	3800.0	FEMALE
2	Adelie	Torgersen	40.30	18.0	195.0	3250.0	FEMALE
3	Adelie	Torgersen	44.45	17.3	197.0	4050.0	MALE
4	Adelie	Torgersen	36.70	19.3	193.0	3450.0	FEMALE

```
df.species = le.fit_transform(df.species)
df.island = le.fit_transform(df.island)
df.sex = le.fit_transform(df.sex)
```

```
df.head()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	0	2	39.10	18.7	181.0	3750.0	2
1	0	2	39.50	17.4	186.0	3800.0	1
2	0	2	40.30	18.0	195.0	3250.0	1
3	0	2	44.45	17.3	197.0	4050.0	2
4	0	2	36.70	19.3	193.0	3450.0	1

Splitting Data into Independent And Dependent Datas

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

```
y.head()
```

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

```
X.head()
```

	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	2	39.10	18.7	181.0	3750.0	2
1	2	39.50	17.4	186.0	3800.0	1
2	2	40.30	18.0	195.0	3250.0	1
3	2	44.45	17.3	197.0	4050.0	2
4	2	36.70	19.3	193.0	3450.0	1

Scaling Data

```
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_g	sex
0	1.0	0.254545	0.666667	0.152542	0.291667	1.0
1	1.0	0.269091	0.511905	0.237288	0.305556	0.5
2	1.0	0.298182	0.583333	0.389831	0.152778	0.5
3	1.0	0.449091	0.500000	0.423729	0.375000	1.0
4	1.0	0.167273	0.738095	0.355932	0.208333	0.5

Splitting The Data Into Training And Testing

```
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=2)
```

Checking The Shape Of Splitted Training And Testing Dataset

```
X_train.shape
```

```
(240, 6)
```

```
X_test.shape
```

```
(104, 6)
```

```
y_train.shape
```

```
(240,)
```

```
y_test.shape
```

```
(104,)
```