

```
#Assignment 3
#Pavithra S
#21BIT0389
```

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

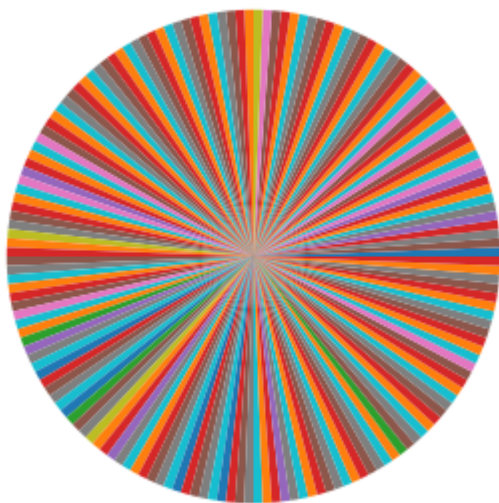
	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	MALE
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	FEMALE
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	FEMALE
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	FEMALE
...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0	FEMALE
341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0	MALE
342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0	FEMALE
343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0	MALE

344 rows × 7 columns

## Univariate Analysis- Pie Chart

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.figure(figsize=(4,4))  
condition=df['sex']=='MALE'  
plt.pie(condition)  
plt.show()
```



## Distribution Plot

```
plt.figure(figsize=(4,4))  
sns.distplot(df['body_mass_g'])  
plt.show()
```

```
<ipython-input-8-e30e5dc45fc7>:2: UserWarning:
```

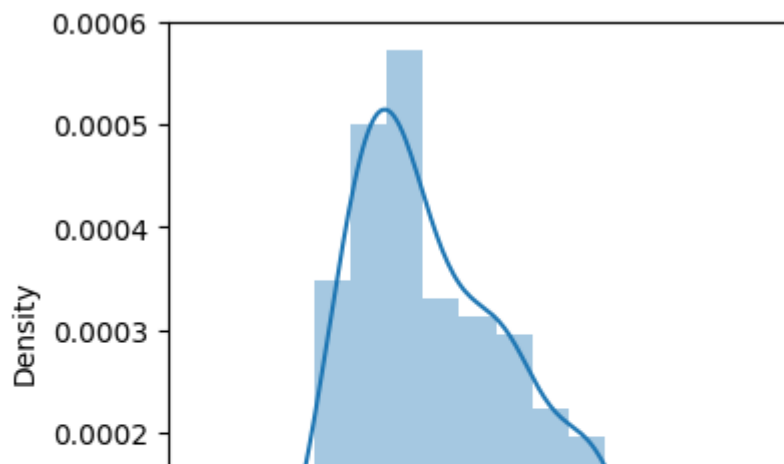
```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

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

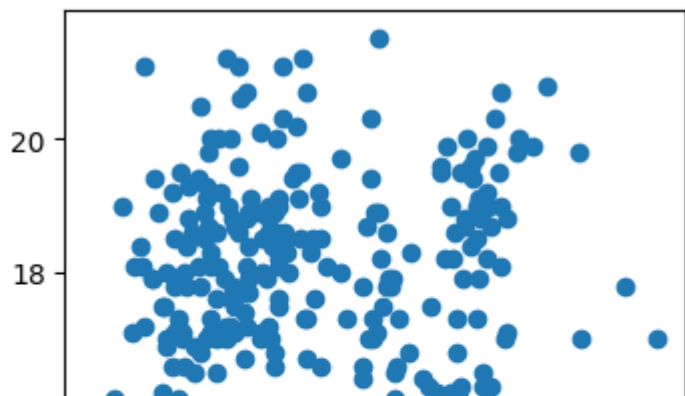
<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df['body_mass_g'])
```



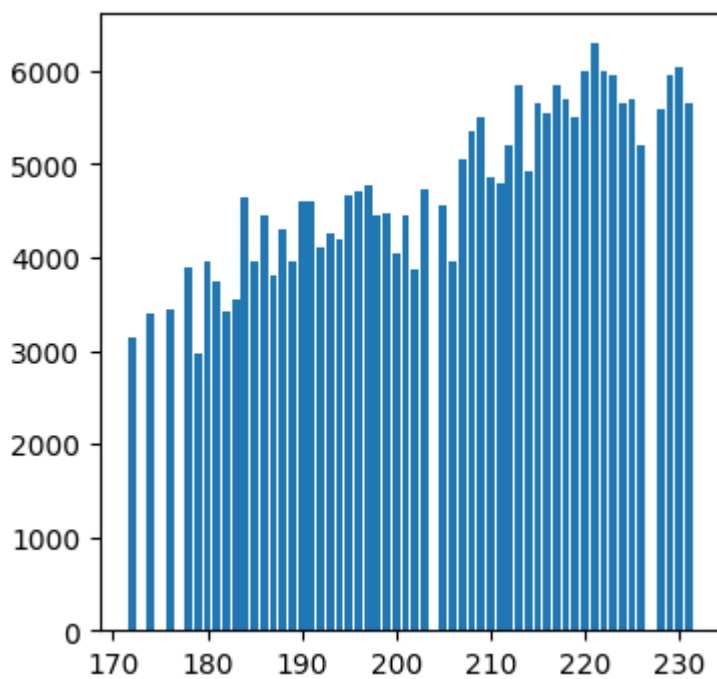
Bivariate Analysis-Scatter Graph

```
plt.figure(figsize=(4,4))  
plt.scatter(df['culmen_length_mm'],df['culmen_depth_mm'])  
plt.show()
```



Bivariate Analysis-Bar Graph

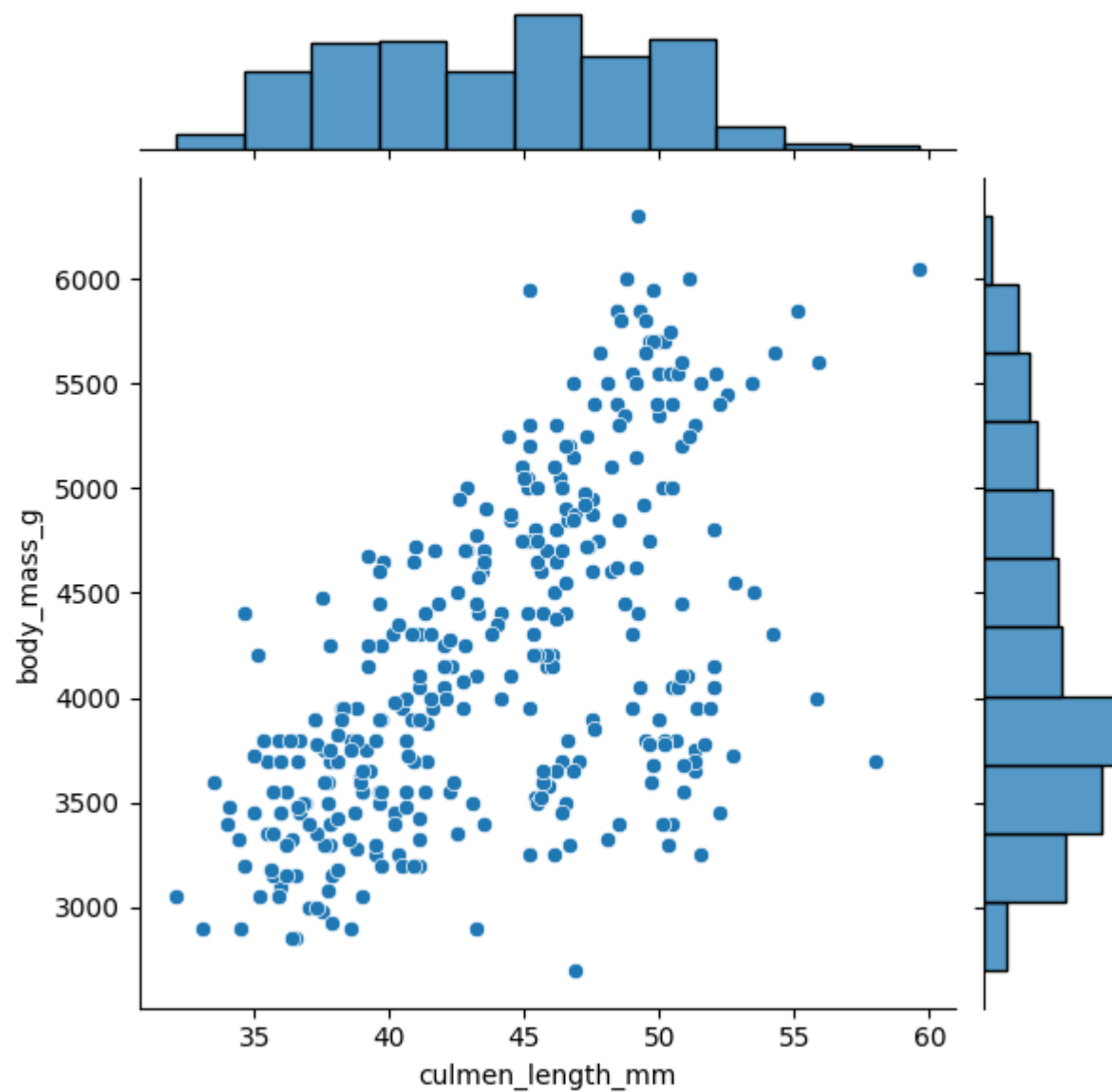
```
plt.figure(figsize=(4,4))
plt.bar(df['flipper_length_mm'], df['body_mass_g'])
plt.show()
```



## Bivariate Analysis-Jointpoint

```
sns.jointplot(x='culmen_length_mm', y='body_mass_g', data=df)
```

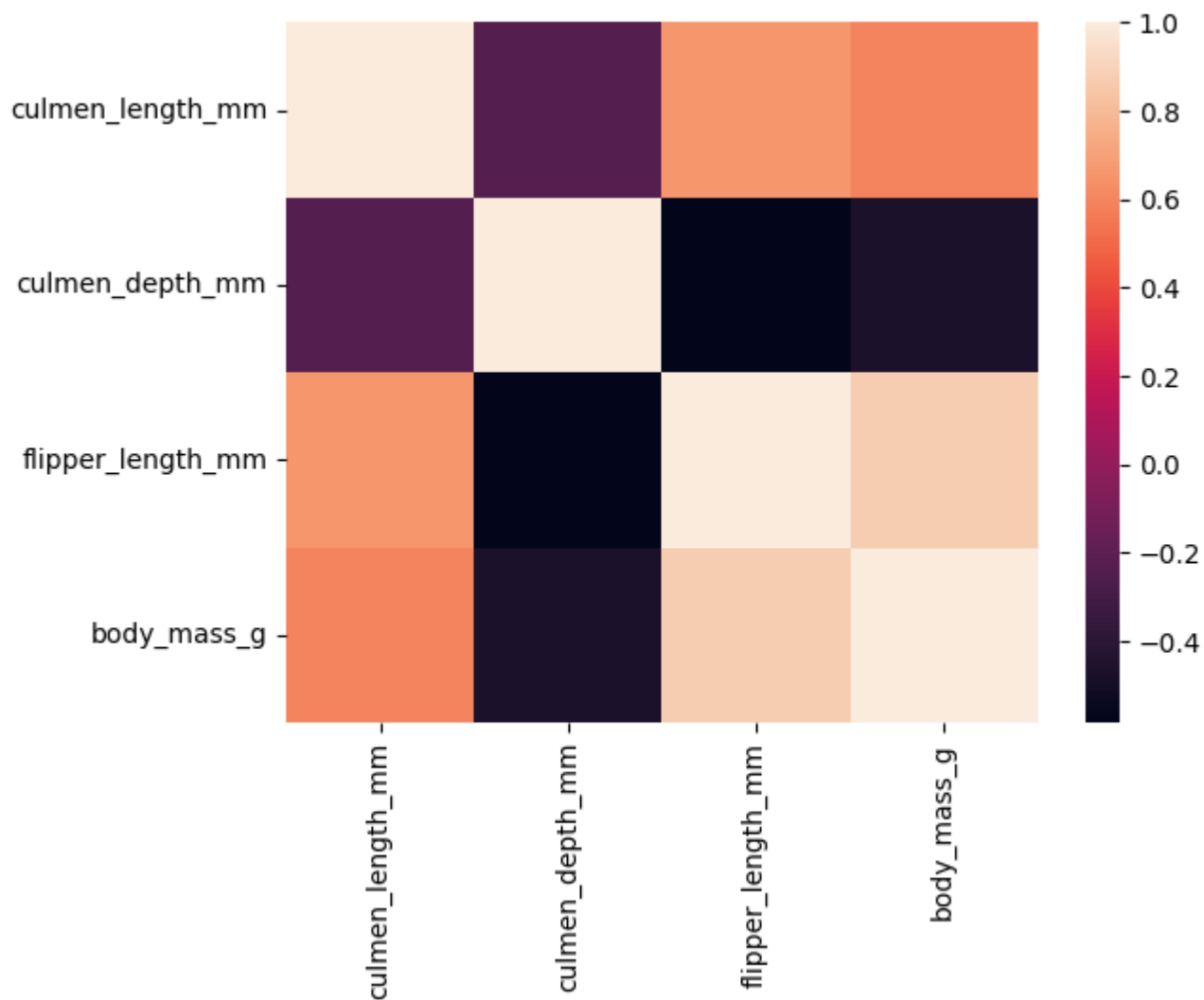
```
<seaborn.axisgrid.JointGrid at 0x7a99b799cc10>
```



## Multivariate Analysis- Heatmap

```
sns.heatmap(df.corr())
```

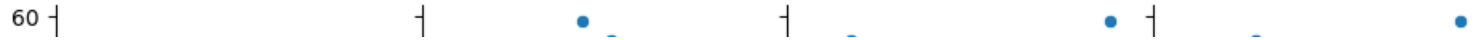
```
<ipython-input-13-aa4f4450a243>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is
  sns.heatmap(df.corr())
<Axes: >
```



## Multivariate Analysis- PairPlot

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

```
<seaborn.axisgrid.PairGrid at 0x7a99b38a4cd0>
```



## Descriptive Statistics



```
df.describe()
```

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	
<b>count</b>	342.000000	342.000000	342.000000	342.000000	
<b>mean</b>	43.921930	17.151170	200.915205	4201.754386	
<b>std</b>	5.459584	1.974793	14.061714	801.954536	
<b>min</b>	32.100000	13.100000	172.000000	2700.000000	
<b>25%</b>	39.225000	15.600000	190.000000	3550.000000	
<b>50%</b>	44.450000	17.300000	197.000000	4050.000000	
<b>75%</b>	48.500000	18.700000	213.000000	4750.000000	
<b>max</b>	59.600000	21.500000	231.000000	6300.000000	



## Checking and handling 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.sex.value_counts()
```

```
MALE      168
FEMALE    165
.          1
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-22-60f9aa6de9d9>:2: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated.
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())
```

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

```
<ipython-input-23-fea379c4db1f>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated.
df=df.fillna(df.median())
species              0
island               0
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

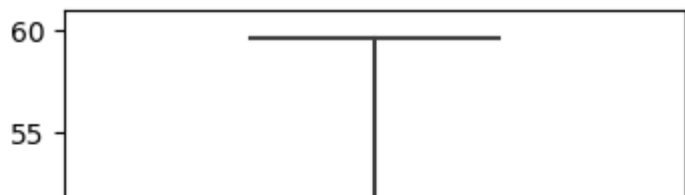
```
df.info()
```

```
<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
```

## Find and replace Outliers

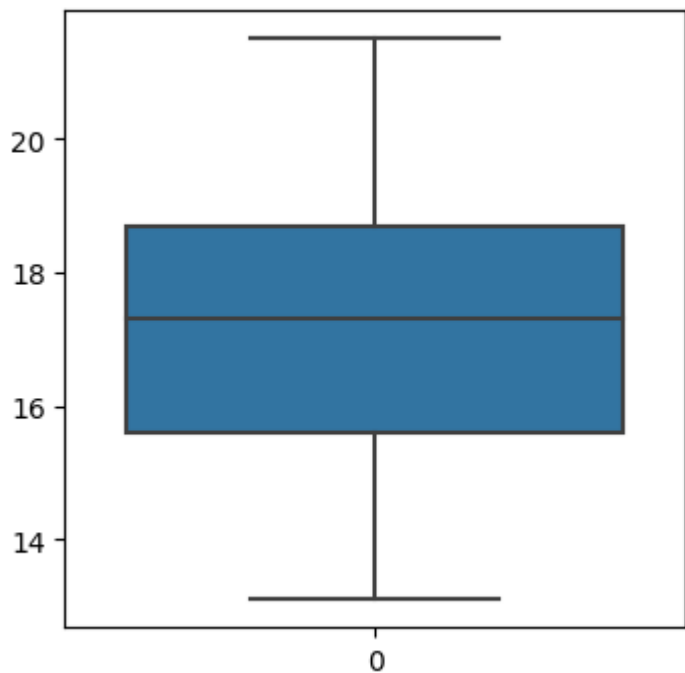
```
plt.figure(figsize=(4,4))
sns.boxplot(df.culmen_length_mm)
```

&lt;Axes: &gt;



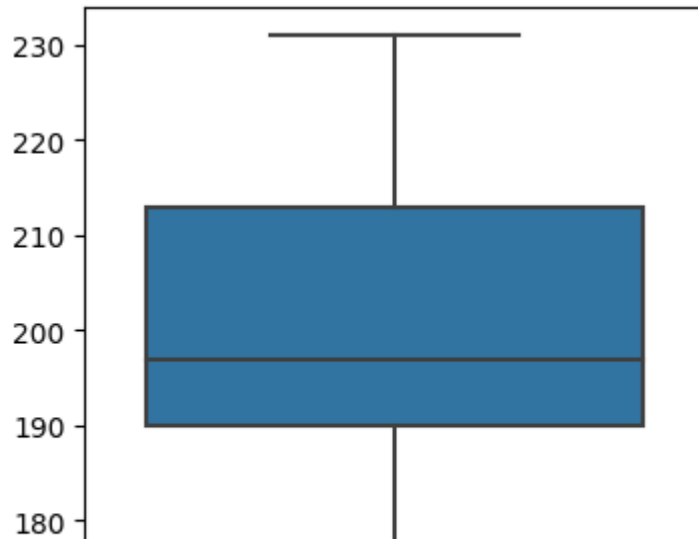
```
plt.figure(figsize=(4,4))  
sns.boxplot(df.culmen_depth_mm)
```

&lt;Axes: &gt;



```
plt.figure(figsize=(4,4))  
sns.boxplot(df.flipper_length_mm)
```

&lt;Axes: &gt;



```
plt.figure(figsize=(4,4))  
sns.boxplot(df.body_mass_g)
```

no outliers

Check for Categorical Columns and perform Encoding

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
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['sex'] = le.fit_transform(df['sex'])
df['species'] = le.fit_transform(df['species'])
df['island'] = le.fit_transform(df['island'])
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	1	

Check Correlation of Independent Variables with the target

2	0	2	40.30	18.0	195.0	3250.0	0
---	---	---	-------	------	-------	--------	---

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

```
species          1.000000
flipper_length_mm 0.850819
body_mass_g       0.747547
culmen_length_mm  0.728706
sex               0.010240
island           -0.635659
culmen_depth_mm   -0.741282
Name: species, dtype: float64
```

Split the data into dependent and independent variables

```
x=df.drop(columns=['species'], axis=1)
y=df.species
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	1	
1	2	39.50	17.4	186.0	3800.0	0	
2	2	40.30	18.0	195.0	3250.0	0	
3	2	44.45	17.3	197.0	4050.0	1	
4	2	36.70	19.3	193.0	3450.0	0	



```
y.head()
```

```
0    0
1    0
2    0
3    0
4    0
```

```
Name: species, dtype: int64
```

## Scaling Data

```
from sklearn.preprocessing import MinMaxScaler
scale=MinMaxScaler()
x_s=pd.DataFrame(scale.fit_transform(x),columns=x.columns)
x_s.head()
```

	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex	
<b>0</b>	1.0	0.254545	0.666667	0.152542	0.291667	1.0	
<b>1</b>	1.0	0.269091	0.511905	0.237288	0.305556	0.0	
<b>2</b>	1.0	0.298182	0.583333	0.389831	0.152778	0.0	
<b>3</b>	1.0	0.449091	0.500000	0.423729	0.375000	1.0	
<b>4</b>	1.0	0.167273	0.738095	0.355932	0.208333	0.0	

## Split 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_s,y,test_size=0.2,random_state=0)
```

## Check training and testing Shape

```
x_train.shape
```

```
(275, 6)
```

```
x_test.shape
```

```
(69, 6)
```

```
y_train.shape
```

```
↳ (275,)
```

---

[+ Code](#)[+ Text](#)

```
y_test.shape
```

```
(69,)
```

[Colab paid products](#) - [Cancel contracts here](#)



✓ 0s completed at 7:41 PM

