```
import numpy as np
import pandas as pd

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

reading the data

getting the shape

```
data = pd.read_csv('/content/abalone.csv')

# getting the shape
data.shape
```

```
(4177, 9)
```

looking at the head of the data

```
data.head()
```

```
  Sex  Length  Diameter  Height  Whole weight  Shucked weight  Viscera
weight  \
0   M   0.455     0.365   0.095        0.5140          0.2245
0.1010
1   M   0.350     0.265   0.090        0.2255          0.0995
0.0485
2   F   0.530     0.420   0.135        0.6770          0.2565
0.1415
3   M   0.440     0.365   0.125        0.5160          0.2155
0.1140
4   I   0.330     0.255   0.080        0.2050          0.0895
0.0395

   Shell weight  Rings
0         0.150     15
1         0.070      7
2         0.210      9
3         0.155     10
4         0.055      7
```

describe the data

```
data.describe()
```

```
            Length     Diameter       Height  Whole weight  Shucked
weight  \
count  4177.000000  4177.000000  4177.000000   4177.000000
4177.000000
mean      0.523992     0.407881     0.139516      0.828742
0.359367
std       0.120093     0.099240     0.041827      0.490389
```

```
0.221963
min       0.075000     0.055000     0.000000      0.002000
0.001000
25%       0.450000     0.350000     0.115000      0.441500
0.186000
50%       0.545000     0.425000     0.140000      0.799500
0.336000
75%       0.615000     0.480000     0.165000      1.153000
0.502000
max       0.815000     0.650000     1.130000      2.825500
1.488000

       Viscera weight  Shell weight        Rings
count     4177.000000   4177.000000  4177.000000
mean         0.180594      0.238831     9.933684
std          0.109614      0.139203     3.224169
min          0.000500      0.001500     1.000000
25%          0.093500      0.130000     8.000000
50%          0.171000      0.234000     9.000000
75%          0.253000      0.329000    11.000000
max          0.760000      1.005000    29.000000
```

information of the data

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4177 entries, 0 to 4176
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Sex             4177 non-null   object
 1   Length          4177 non-null   float64
 2   Diameter        4177 non-null   float64
 3   Height          4177 non-null   float64
 4   Whole weight    4177 non-null   float64
 5   Shucked weight  4177 non-null   float64
 6   Viscera weight  4177 non-null   float64
 7   Shell weight    4177 non-null   float64
 8   Rings           4177 non-null   int64
dtypes: float64(7), int64(1), object(1)
memory usage: 293.8+ KB
```

checking if there is any NULL data

```
data.isnull().sum()
```

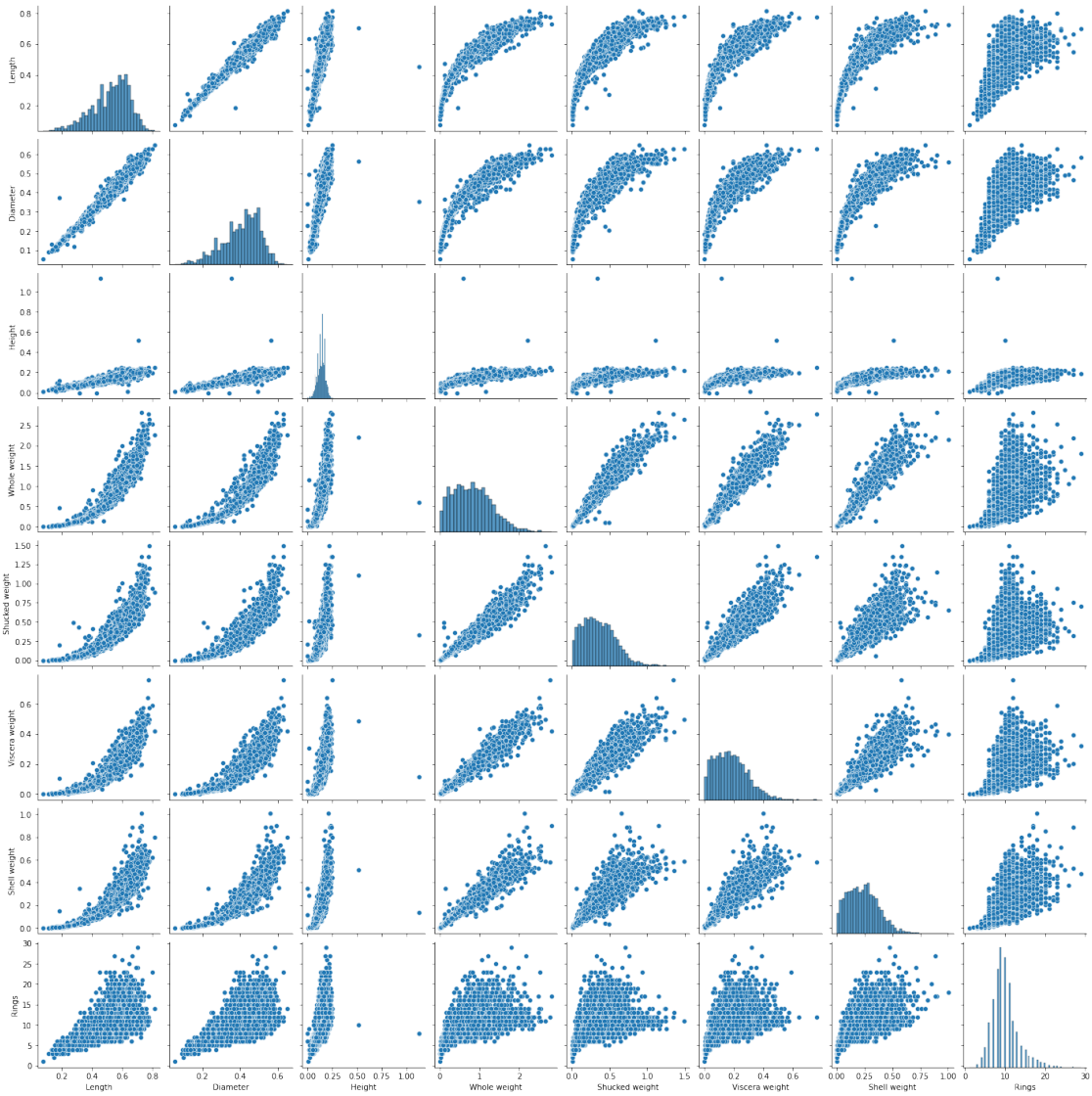
```
Sex               0
Length            0
Diameter          0
Height            0
```

```
Whole weight      0
Shucked weight    0
Viscera weight    0
Shell weight      0
Rings             0
dtype: int64
```

pairplot

```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7fc9851302d0>
```



checking the columns of the data

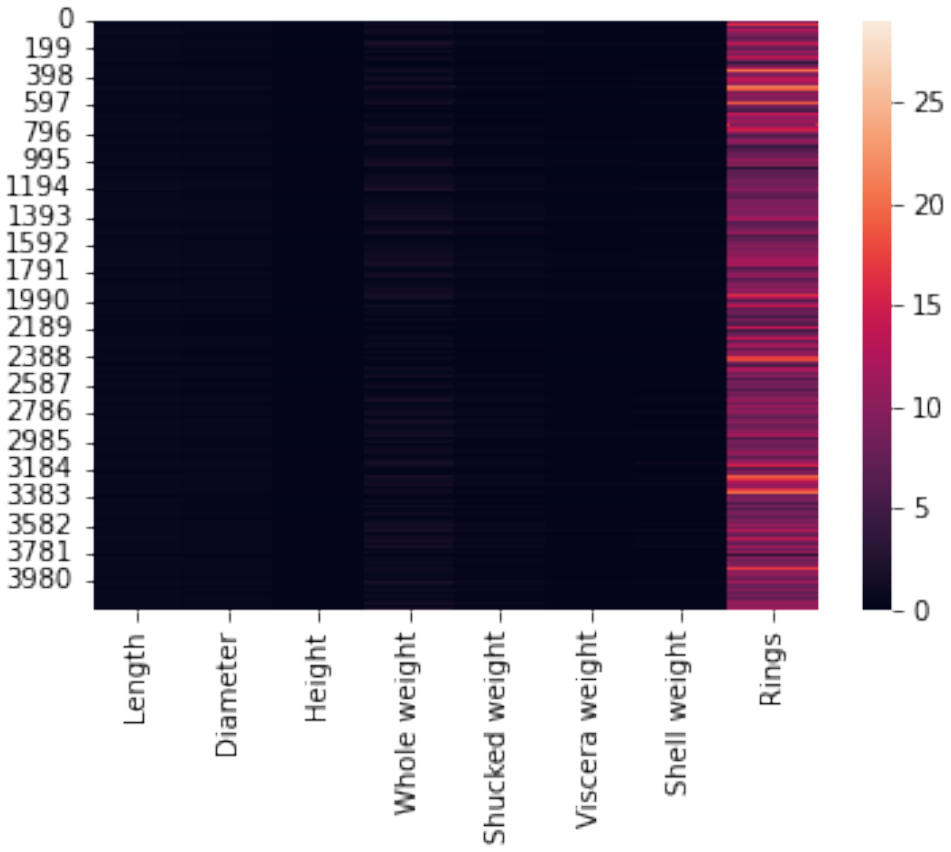
```
data.columns
```

```
Index(['Sex', 'Length', 'Diameter', 'Height', 'Whole weight', 'Shucked
weight',
       'Viscera weight', 'Shell weight', 'Rings'],
      dtype='object')
```

heatmap

```
sns.heatmap(data[[ 'Length', 'Diameter', 'Height', 'Whole weight',
'Shucked weight',
       'Viscera weight', 'Shell weight', 'Rings']])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc980b6b590>
```



checkig the values of sex

```
data['Sex'].value_counts()
```

```
M    1528
I    1342
F    1307
Name: Sex, dtype: int64
```

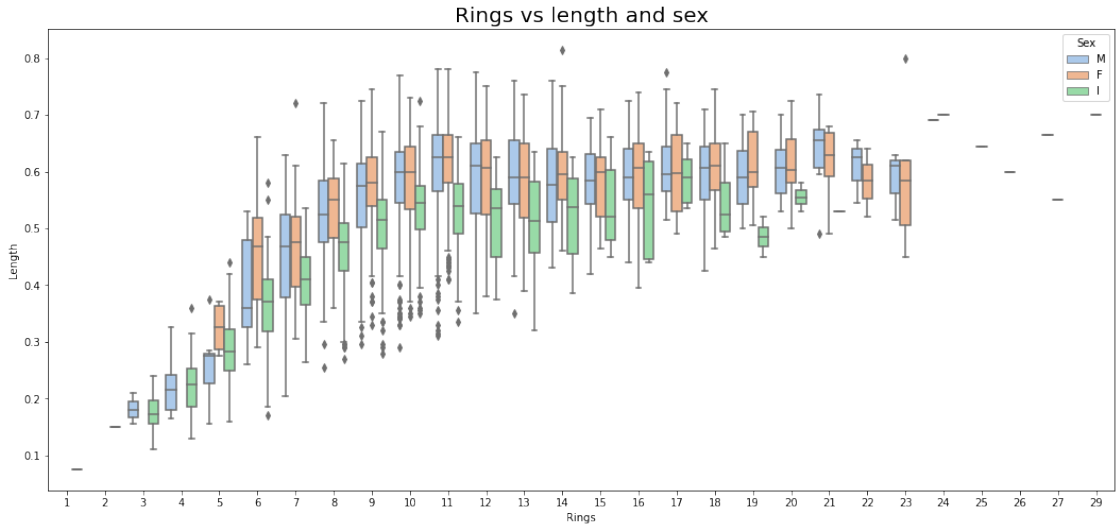
plotting a hue plot

```
plt.rcParams['figure.figsize'] = (18, 8)
sns.boxplot(data['Rings'], data['Length'], hue = data['Sex'], palette
```

```
= 'pastel')
plt.title('Rings vs length and sex', fontsize = 20)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  FutureWarning
```

```
Text(0.5, 1.0, 'Rings vs length and sex')
```

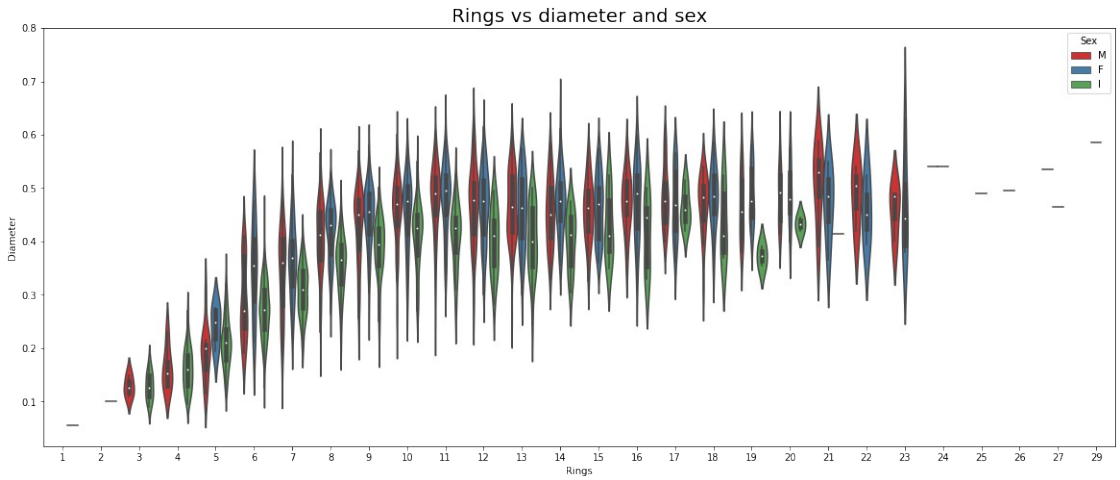


rings vs diameter and sex

```
plt.rcParams['figure.figsize'] = (20, 8)
sns.violinplot(data['Rings'], data['Diameter'], hue = data['Sex'],
palette = 'Set1')
plt.title('Rings vs diameter and sex', fontsize = 20)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  FutureWarning
```

```
Text(0.5, 1.0, 'Rings vs diameter and sex')
```

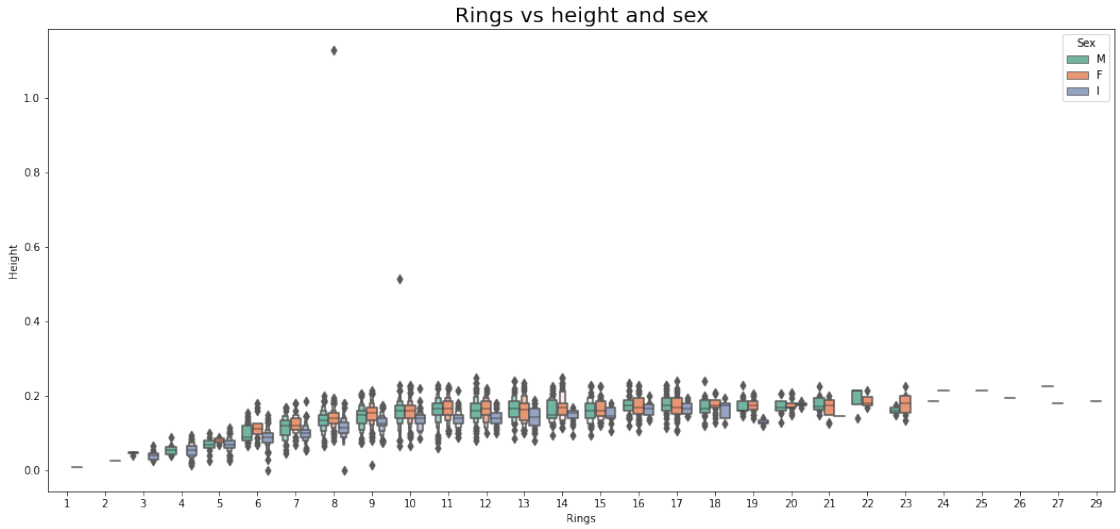


rings vs height and sex

```
plt.rcParams['figure.figsize'] = (18, 8)
sns.boxenplot(data['Rings'], data['Height'], hue = data['Sex'],
palette = 'Set2')
plt.title('Rings vs height and sex', fontsize = 20)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  FutureWarning
```

```
Text(0.5, 1.0, 'Rings vs height and sex')
```



ring vs weight

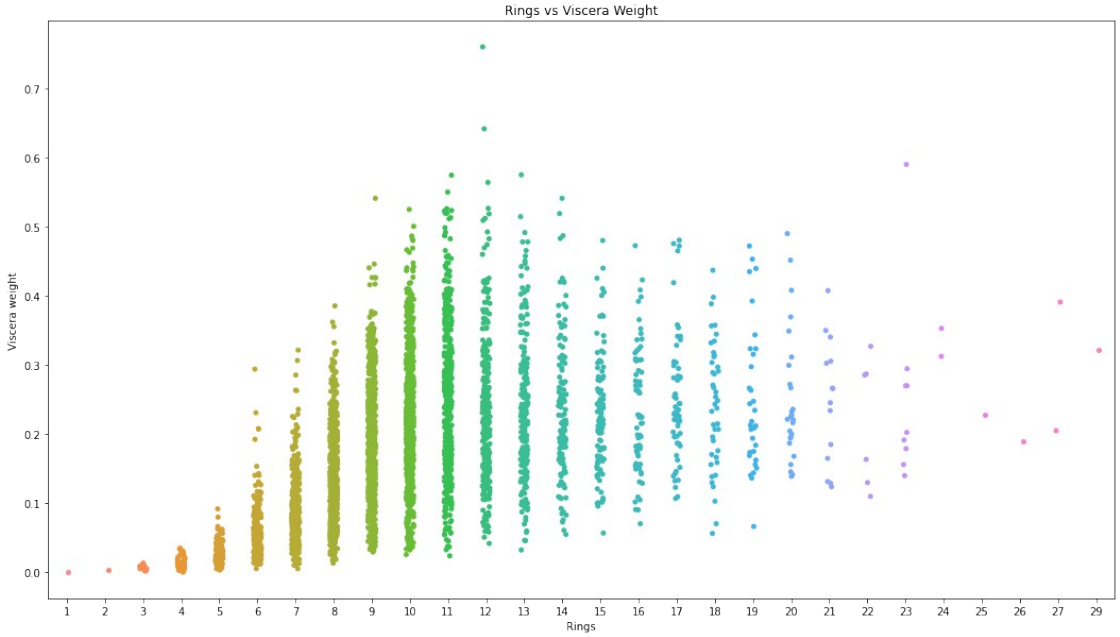
```
plt.rcParams['figure.figsize'] = (18, 10)
sns.swarmplot(data['Rings'], data['Whole weight'])
plt.title('Rings vs weight')
```

plt.rcParams['figure.figsize'] = (18, 10) sns.swarmplot(data['Rings'], data['Shucked weight'], palette = 'dark') plt.title('Rings vs shucked weight')

```
plt.rcParams['figure.figsize'] = (18, 10)
sns.stripplot(data['Rings'], data['Viscera weight'])
plt.title('Rings vs Viscera Weight')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  FutureWarning
```

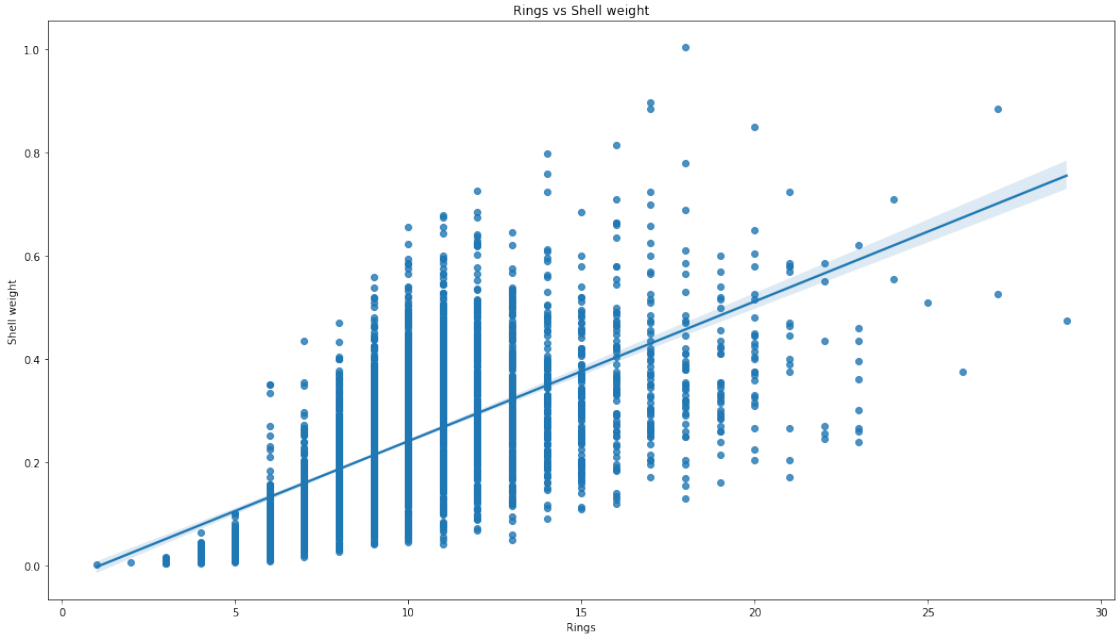
```
Text(0.5, 1.0, 'Rings vs Viscera Weight')
```




```
plt.rcParams['figure.figsize'] = (18, 10)
sns.regplot(data['Rings'], data['Shell weight'])
plt.title('Rings vs Shell weight')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  FutureWarning
```

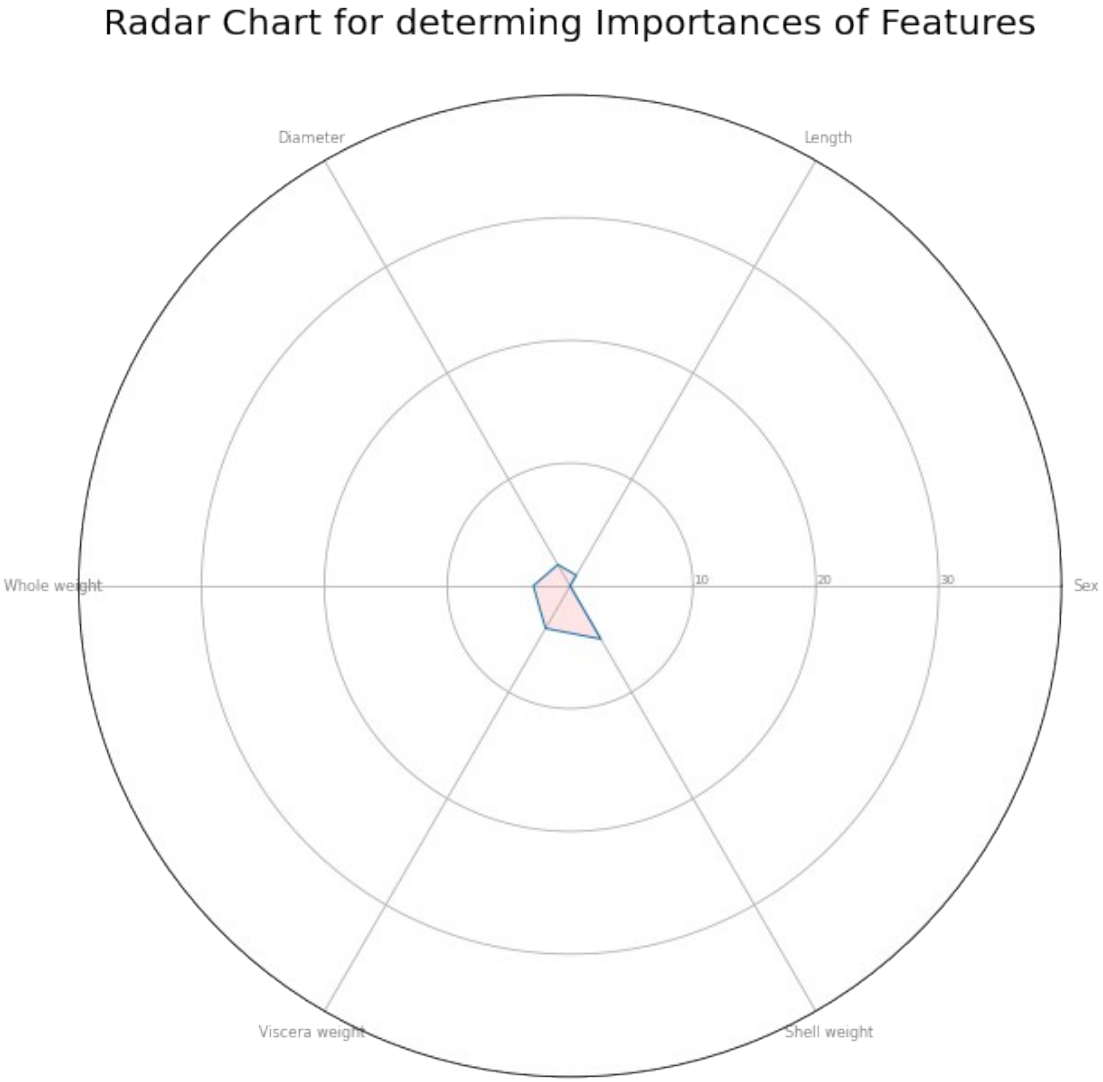
```
Text(0.5, 1.0, 'Rings vs Shell weight')
```




```
from math import pi

# Set data
df = pd.DataFrame({
'group': [i for i in range(0, 4177)],
'Sex': data['Sex'],
'Length': data['Length'],
'Diameter': data['Diameter'],
'Whole weight':  data['Whole weight'],
'Viscera weight': data['Viscera weight'],
'Shell weight': data['Shell weight']
})

# number of variable
categories=list(df)[1:]
N = len(categories)

# We are going to plot the first line of the data frame.
# But we need to repeat the first value to close the circular graph:
values = df.loc[0].drop('group').values.flatten().tolist()
values += values[:1]
values

# What will be the angle of each axis in the plot? (we divide the plot
/ number of variable)
angles = [n / float(N) * 2 * pi for n in range(N)]
angles += angles[:1]

# Initialise the spider plot
ax = plt.subplot(111, polar=True)
```

```
# Draw one axe per variable + add labels labels yet
plt.xticks(angles[:-1], categories, color='grey', size=8)

# Draw ylabels
ax.set_rlabel_position(0)
plt.yticks([10,20,30], ["10","20","30"], color="grey", size=7)
plt.ylim(0,40)

# Plot data
ax.plot(angles, values, linewidth=1, linestyle='solid')
plt.title('Radar Chart for determing Importances of Features',
fontsize = 20)
# Fill area
ax.fill(angles, values, 'red', alpha=0.1)
```

```
[<matplotlib.patches.Polygon at 0x7fc97dadd890>]
```




```
data = pd.get_dummies(data)
```

```
data.head()
```

```
  Sex  Length  Diameter  Height  Whole weight  Shucked weight  Viscera
weight  \
0   M   0.455     0.365   0.095        0.5140          0.2245
0.1010
1   M   0.350     0.265   0.090        0.2255          0.0995
0.0485
2   F   0.530     0.420   0.135        0.6770          0.2565
0.1415
3   M   0.440     0.365   0.125        0.5160          0.2155
0.1140
4   I   0.330     0.255   0.080        0.2050          0.0895
0.0395

   Shell weight  Rings
0         0.150     15
1         0.070      7
2         0.210      9
3         0.155     10
4         0.055      7
```

```
y = data['Rings']
data = data.drop(['Rings'], axis = 1)
x = data

# getting the shapes
print("Shape of x:", x.shape)
print("Shape of y:", y.shape)
```

```
Shape of x: (4177, 8)
Shape of y: (4177,)
```

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
0.2, random_state = 0)

# getting the shapes
print("Shape of x_train :", x_train.shape)
print("Shape of x_test :", x_test.shape)
print("Shape of y_train :", y_train.shape)
print("Shape of y_test :", y_test.shape)
```

```
Shape of x_train : (3341, 8)
Shape of x_test : (836, 8)
Shape of y_train : (3341,)
Shape of y_test : (836,)
```

```
# train test split
```

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
0.2, random_state = 0)

# getting the shapes
print("Shape of x_train :", x_train.shape)
print("Shape of x_test :", x_test.shape)
print("Shape of y_train :", y_train.shape)
print("Shape of y_test :", y_test.shape)
```

```
Shape of x_train : (3341, 8)
Shape of x_test : (836, 8)
Shape of y_train : (3341,)
Shape of y_test : (836,)
```

```
!pip install eli5
```

```
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: eli5 in /usr/local/lib/python3.7/dist-
packages (0.13.0)
Requirement already satisfied: attrs>17.1.0 in
/usr/local/lib/python3.7/dist-packages (from eli5) (22.1.0)
Requirement already satisfied: numpy>=1.9.0 in
/usr/local/lib/python3.7/dist-packages (from eli5) (1.21.6)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-
packages (from eli5) (1.15.0)
Requirement already satisfied: scikit-learn>=0.20 in
/usr/local/lib/python3.7/dist-packages (from eli5) (1.0.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-
packages (from eli5) (1.7.3)
Requirement already satisfied: jinja2>=3.0.0 in
/usr/local/lib/python3.7/dist-packages (from eli5) (3.1.2)
Requirement already satisfied: graphviz in
/usr/local/lib/python3.7/dist-packages (from eli5) (0.10.1)
Requirement already satisfied: tabulate>=0.7.7 in
/usr/local/lib/python3.7/dist-packages (from eli5) (0.8.10)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.7/dist-packages (from jinja2>=3.0.0->eli5)
(2.0.1)
Requirement already satisfied: joblib>=0.11 in
/usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.20->eli5)
(1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.20->eli5)
(3.1.0)
```

```
import eli5
from eli5.sklearn import PermutationImportance
```

```
perm = PermutationImportance(model, random_state = 0).fit(x_test,
y_test)
eli5.show_weights(perm, feature_names = x_test.columns.tolist())
```