In [5]:
```
#Importing libraries we need
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import math
```

In [50]:
```
#import and read data
irisData = pd.read_csv('iris.csv')
#Read the top 5 using head function
irisData.head()
```

Out[50]:

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |

In [52]:
```
#How many columns and rows does our shape have?
irisData.shape
```

Out[52]:

| | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| **...** | ... | ... | ... | ... | ... |
| **145** | 6.7 | 3.0 | 5.2 | 2.3 | virginica |
| **146** | 6.3 | 2.5 | 5.0 | 1.9 | virginica |
| **147** | 6.5 | 3.0 | 5.2 | 2.0 | virginica |
| **148** | 6.2 | 3.4 | 5.4 | 2.3 | virginica |
| **149** | 5.9 | 3.0 | 5.1 | 1.8 | virginica |

150 rows × 5 columns

In [21]:
```
#Get descriptive statistics
descriStats = irisData.describe()
descriStats
```

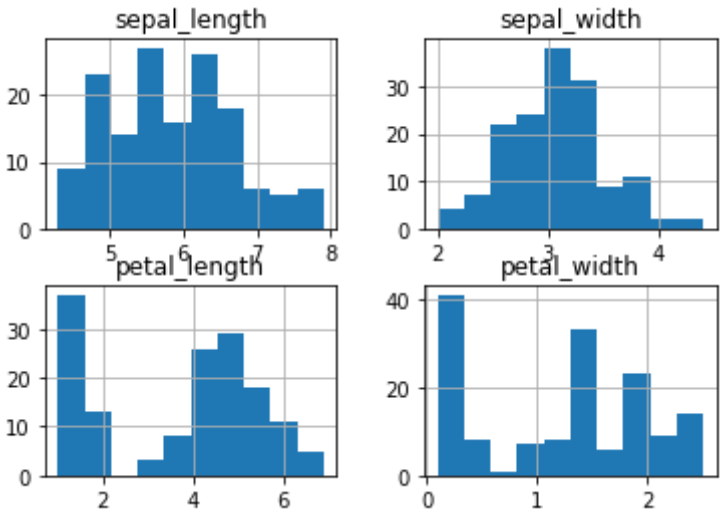
Out[21]:

| | sepal_length | sepal_width | petal_length | petal_width |
|---|---|---|---|---|
| **count** | 150.000000 | 150.000000 | 150.000000 | 150.000000 |
| **mean** | 5.843333 | 3.054000 | 3.758667 | 1.198667 |
| **std** | 0.828066 | 0.433594 | 1.764420 | 0.763161 |
| **min** | 4.300000 | 2.000000 | 1.000000 | 0.100000 |
| **25%** | 5.100000 | 2.800000 | 1.600000 | 0.300000 |
| **50%** | 5.800000 | 3.000000 | 4.350000 | 1.300000 |
| **75%** | 6.400000 | 3.300000 | 5.100000 | 1.800000 |
| **max** | 7.900000 | 4.400000 | 6.900000 | 2.500000 |

In [151]:
```
#Display species we have and count how many each has
irisData.groupby('species').species.count()
```

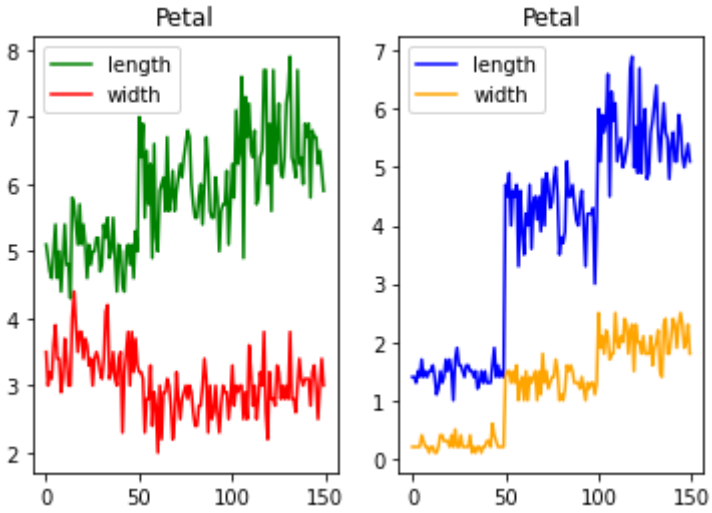
Out[151]:
```
species
setosa        50
versicolor    50
virginica     50
Name: species, dtype: int64
```

In [152]:
```
# Visualize each attribute using histograms
hists = irisData.hist(bins=10)
```



In [153]:
```
fig,(axSep,axPet) = plt.subplots(nrows = 1,ncols = 2)
axSep.plot(irisData['sepal_length'], color='green', marker='o', linestyle='
axSep.plot(irisData['sepal_width'], color='red', marker='o', linestyle='sol
axPet.plot(irisData['petal_length'], color='blue', marker='o', linestyle='s
axPet.plot(irisData['petal_width'], color='orange', marker='o', linestyle='
axSep.legend()
axPet.legend()
axPet.set_title('Petal')
axSep.set_title('Petal')
```

Out[153]:
```
Text(0.5, 1.0, 'Petal')
```




**Get the sample mean of Sepal data**

In [6]:
```
sepLength = irisData['sepal_length']
```

In [66]:
```
count = 0
for i in sepLength:
    count = count + i
sepal_len_mean = round((count/sepLength.count()),2)
print('The sample mean of Sepal length data is : ',sepal_len_mean)
```

```
The sample mean of Sepal length data is :  5.84
```

**Now let's calculate the coefficient of variation in sepal lenth data**

In [99]:
```
sepal_length_sdv = descriStats.iloc[2:3,0]
sepal_length_sdv = float(sepal_length_sdv)
```

In [100]:
```
coeff_of_var = (sepal_length_sdv/sepal_len_mean)*100
```

In [101]:
```
print('The coefficient of variation in sepal lenth data is equal', math.flo
```

```
The coefficient of variation in sepal lenth data is equal 14 %
```

## Apply Chevychev rule of intervals

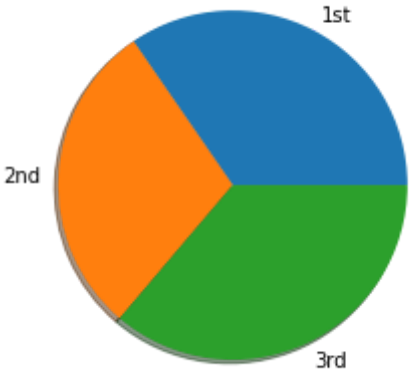
In [106]:
```
lowerLimit1 = round(sepal_len_mean - (3*sepal_length_sdv),2)
upperLimit1 = round(sepal_len_mean + (3*sepal_length_sdv),2)
lowerLimit2 = round(sepal_len_mean - (2*sepal_length_sdv),2)
upperLimit2 = round(sepal_len_mean + (2*sepal_length_sdv),2)
lowerLimit3 = round(sepal_len_mean - (4*sepal_length_sdv),2)
upperLimit3 = round(sepal_len_mean + (4*sepal_length_sdv),2)
```

In [113]:
```
: sepal lenth data is between ',lowerLimit, 'and ',upperLimit,'\n 75% of sep
of sepal lenth data is between ',lowerLimit3, 'and ',upperLimit3)
```

```
 88.9% of sepal lenth data is between  3.36 and  8.32
 75% of sepal lenth data is between  4.18 and  7.5
 93.8% of sepal lenth data is between  2.53 and  9.15
```

## Display how much data in 1st,second,and 3rd interval from the mean

In [117]:
```
counts = [ 88.9,75,93 ]
lab = ["1st","2nd","3rd"]
plt.pie(counts,shadow = True, labels= lab);
```




## Calculate percentiles using quantile function

In [37]:
```
 quantiles = irisData['sepal_width'].quantile(q =[0.25, 0.5,0.75])
quantiles
```

Out[37]:
```
0.25    2.8
0.50    3.0
0.75    3.3
Name: sepal_width, dtype: float64
```

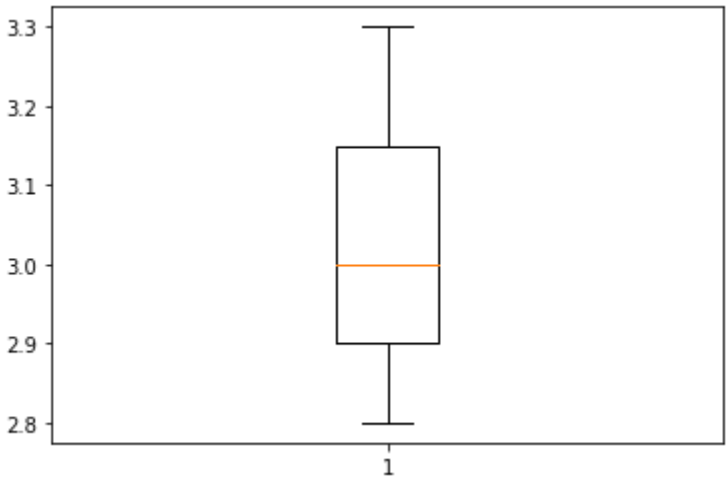
***Calculate the IQR(upper-lower value of 75th percentile in this case minus the 25th)***

In [38]:
```
iqrSeries = quantiles.to_numpy()
for i in range(len(iqrSeries)):
    iqr = iqrSeries[2]-iqrSeries[0]
print('The interquartile range on sepal with data is: ',iqr)
```

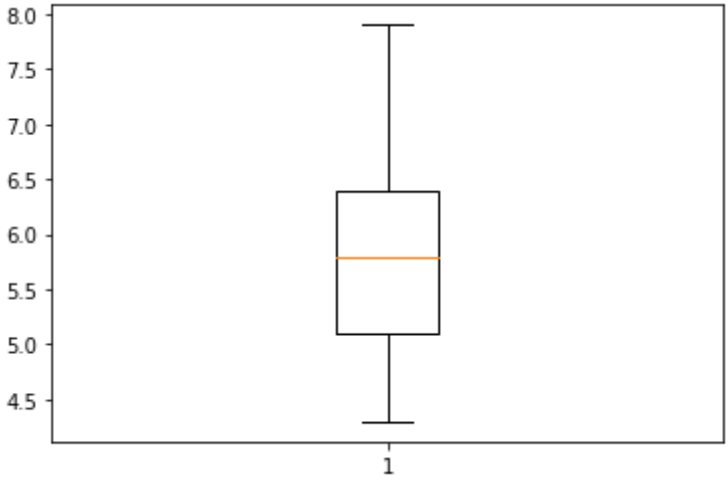
```
The interquartile range on sepal with data is:  0.5
```

**Box plot petal width column using quartile values**

In [46]:
```
sepLength = irisData['sepal_length']
plt.boxplot(iqrSeries)
plt.show()
#the range in sepal width is less than the range in sepal length data set
```



In [51]:
```
plt.boxplot(sepLength)
plt.show()
# We can take conclusion that legth data foe sepal is more spread out based
#Also from the box plot we can tell the minimums,maximums and medias of dat
```




In [71]:
```
#sepal width equal 3.0 constistute how much percent of the whole sepla with
sepWidth = irisData['sepal_width']
count = 0
for i in range(len(sepWidth)):
    if sepWidth[i] == 3.0:
        count = count + 1
print(round((count/len(sepWidth)*100),2),'% of sepal width data is equal to
```

```
17.33 % of sepal width data is equal to 3.0
```

# End