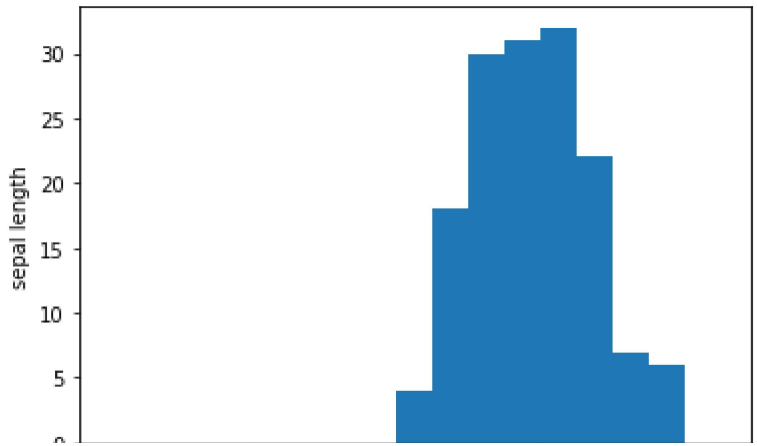
Session 6

Introduction to statistics 1

```
1 import pandas as pd
2 import numpy as np
3 from scipy import stats
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 %matplotlib inline
7
```

```
1
2 # read dataset
3 df = pd.read_csv('iris.csv')
4
5 def histo():
6     # create histogram
7     bin_edges = np.arange(0, df['sepal.length'].max() + 1, 0.5)
8     fig = plt.hist(df['sepal.length'], bins=bin_edges)
9
10    # add plot labels
11    plt.xlabel('count')
12    plt.ylabel('sepal length')
13
14
15 histo()
16 plt.show()
```




```
1 x = df['sepal.length'].values
2 x.dtype
```

```
dtype('float64')
```

Sample Mean:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} = x_i$$

```
1 print("Sample-Mean=",sum(i for i in x) / len(x))
2 x_mean = np.mean(x)
3 x_mean
4 print("Sample-Mean=",x_mean)
```

```
Sample-Mean= 5.843333333333335
Sample-Mean= 5.843333333333334
```

Sample Variance:

$$Var_x = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Bessel's correction to correct the bias of the population variance estimate Note the $unit$ of the variable is now $unit^2$

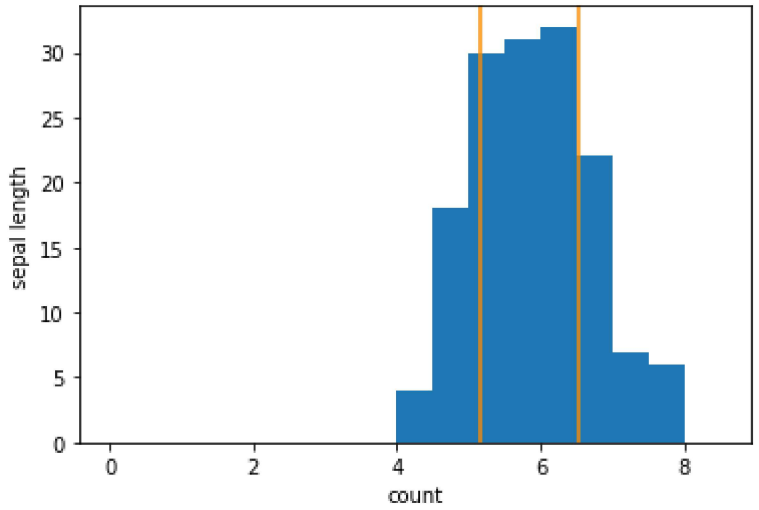
```
print("Sample Variance=",sum([(i - x_mean)**2 for i in x]) / (len(x) - 1))
print("Sample Variance=",np.var(x, ddof=1))
```

```
Sample Variance= 0.6856935123042504
Sample Variance= 0.6856935123042507
```

```
var=df['sepal.length'].var()
print("Sample Variance= ",df['sepal.length'].var(),) # note that Bessel's correction is the default
```

```
Sample Variance=  0.6856935123042505
```

```
histo()
plt.axvline(x_mean + var, color='darkorange')
plt.axvline(x_mean - var, color='darkorange')
plt.show()
```



Sample Standard Deviation:

$$Std_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

```
print("Standard Deviation=",(sum([(i - x_mean)**2 for i in x]) / (len(x) - 1))**0.5)
print("Standard Deviation=",np.sqrt(np.var(x, ddof=1)))
```

```
Standard Deviation= 0.8280661279778628
Standard Deviation= 0.828066127977863
```

```
std = np.std(x, ddof=1)
std
print("Standard Deviation=",std)

df['sepal.length'].std() # note that Bessel's correction is the default
```

```
Standard Deviation= 0.828066127977863
0.8280661279778629
```

Min/Max:

```
#Min/Max:
print("Minimum=",np.min(x))
print("Maximum=",np.max(x))
```

```
Minimum= 4.3
Maximum= 7.9
```

Mode

```
lst = list(x)
mode = max(set(lst), key=lst.count)
mode
```

```
5.0
```

Count

```
1 lst.count(mode)
```

```
10
```

Mode

```
1 stats.mode(x)
2
```

```
ModeResult(mode=array([5.]), count=array([10]))
```

25th and 75th Percentile:

```
1 y = np.sort(x)
2 percentile_25th = y[round(0.25 * y.shape[0]) + 1]
3 percentile_25th
```

```
5.1
```

```
1 percentile_75th = y[round(0.75 * y.shape[0]) - 1]
2 percentile_75th
```

```
6.4
```

Using np array 25th and 75th Percentile:

```
1 np.percentile(x, q=[25, 75], interpolation='lower')
```

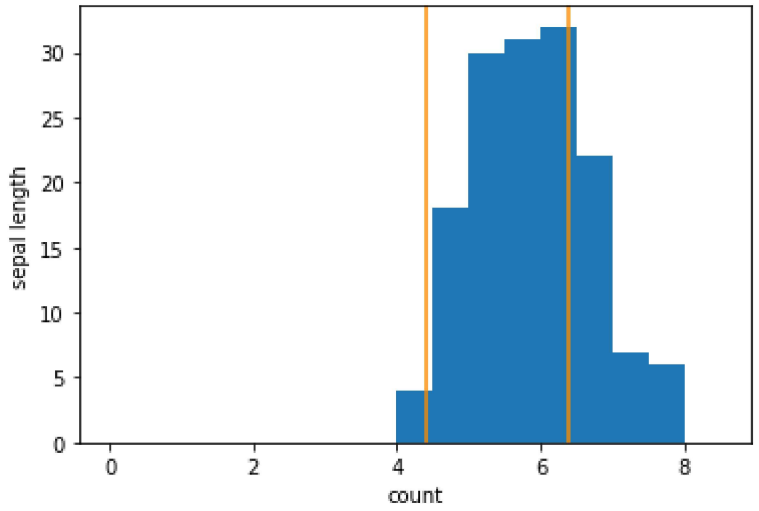
```
array([5.1, 6.4])
```

```
1 print(df['sepal.length'].quantile(0.25, interpolation='lower'))
2 print(df['sepal.length'].quantile(0.75, interpolation='lower'))
```

```
5.1
```

```
6.4
```

```
histo()
plt.axvline(percentile_75th, color='darkorange')
plt.axvline(percentile_25th - var, color='darkorange')
plt.show()

```



Median (50th Percentile):

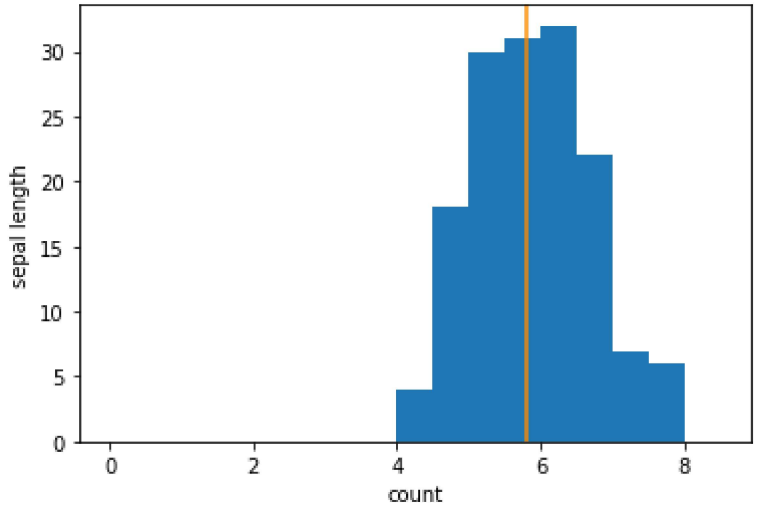
```
x = np.sort(x)

tmp = round(0.5 * x.shape[0])

if x.shape[0] % 2:
    median = x[tmp - 1]
else:
    median = x[tmp - 1] + (x[tmp] - x[tmp - 1]) / 2.

median
```

```
5.8
```

using np

```
1 np.median(x)
```

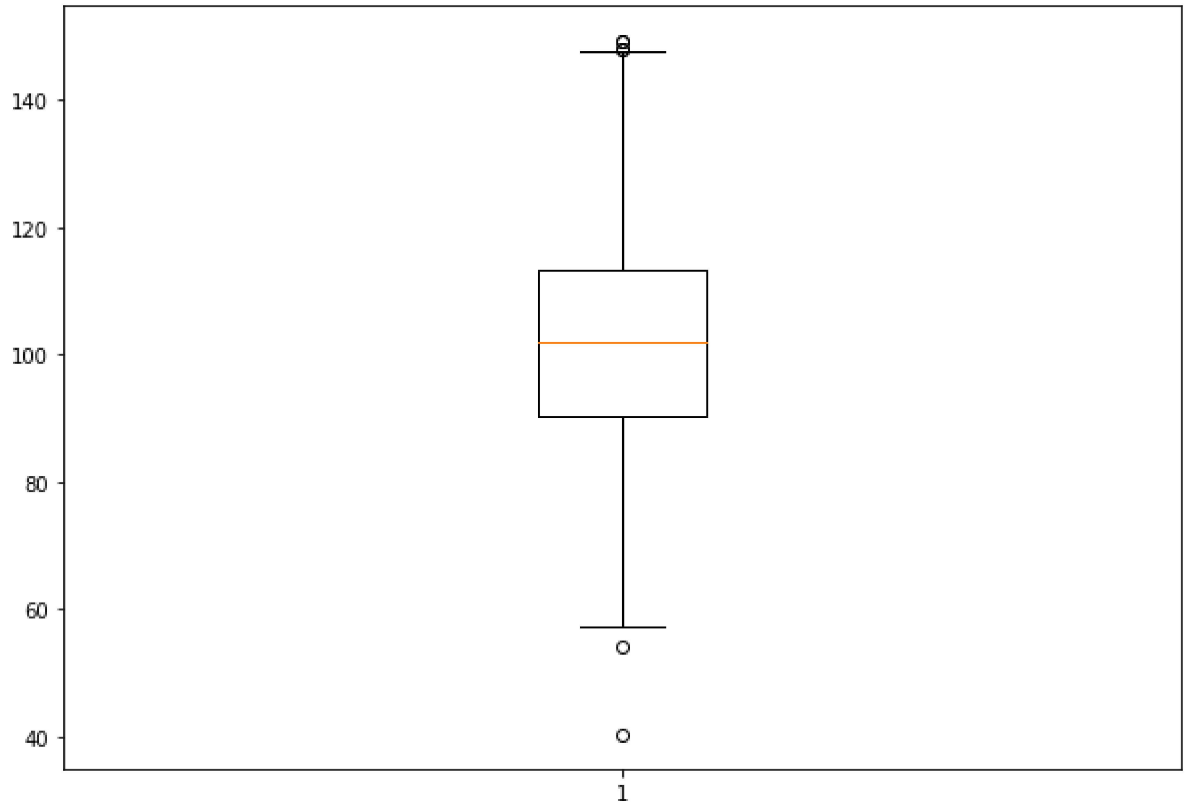
```
1 histo()
2 plt.axvline(median, color='darkorange')
3 plt.show()
```



Box Plot

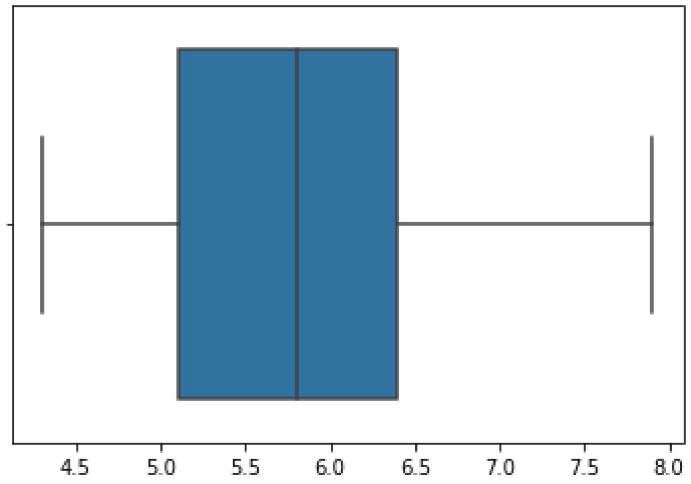
```
1 # Import libraries
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5
6 # Creating dataset
7 np.random.seed(10)
8 data = np.random.normal(100, 20, 200)
```

```
 9
10 fig = plt.figure(figsize =(10, 7))
11
12 # Creating plot
13 plt.boxplot(data)
14
15 # show plot
16 plt.show()
17
```

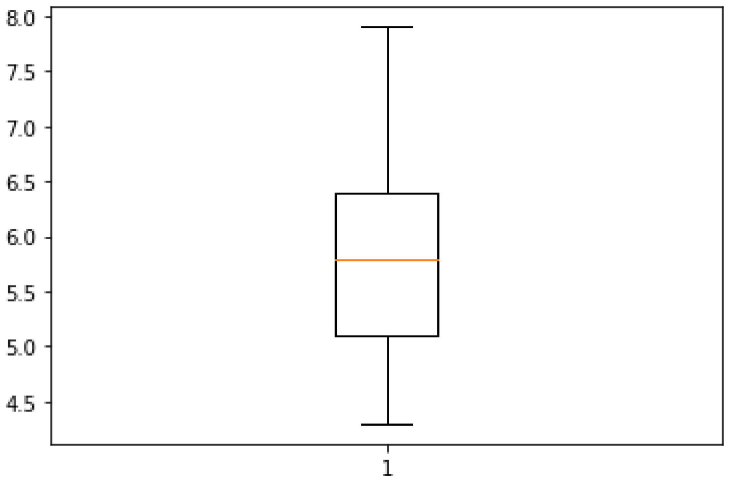



```
1 # Creating plot using seaborn
2
3 sns.boxplot(x)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keywc
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fbc622162d0>
```




```
1 # Creating plot using matplotlib
2 plt.boxplot(x)
3
4 # show plot
5 plt.show()
```



✓ 0s completed at 2:58 PM