

Box plots are the best tool to visualise the data distribution.

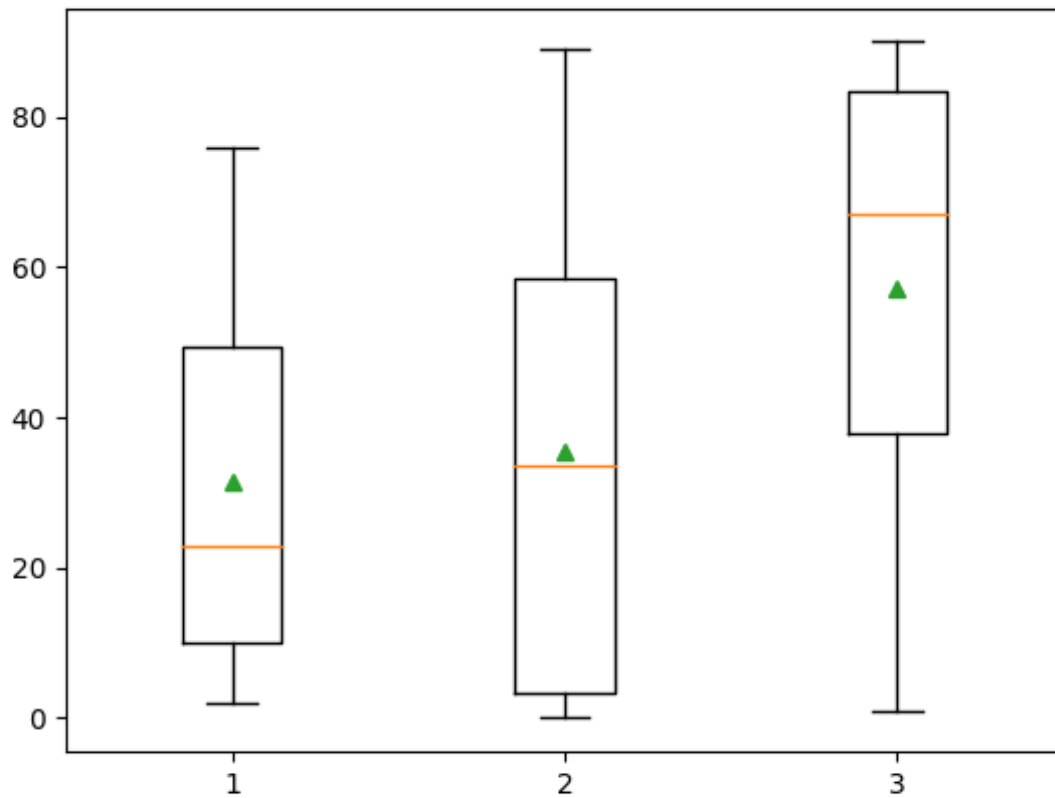
the line in the middle of the box plot shows the median of the data. boxplot is divide into 4 quartiles minimum, first quartile, second quartile, maximum and it's box enclose the middle 50% of data. the top and bottom whiskers shows the maximum and minimum respectively or to maximum or minimum values that fall within 1.5 times the height of the box, whichever yields the shorter whisker. 1.5 times height of the whisker in either direction are called the upper and ower fences. individuals values falls beyond the fences are called outliers.

in boxplot outliers are shown by rings.

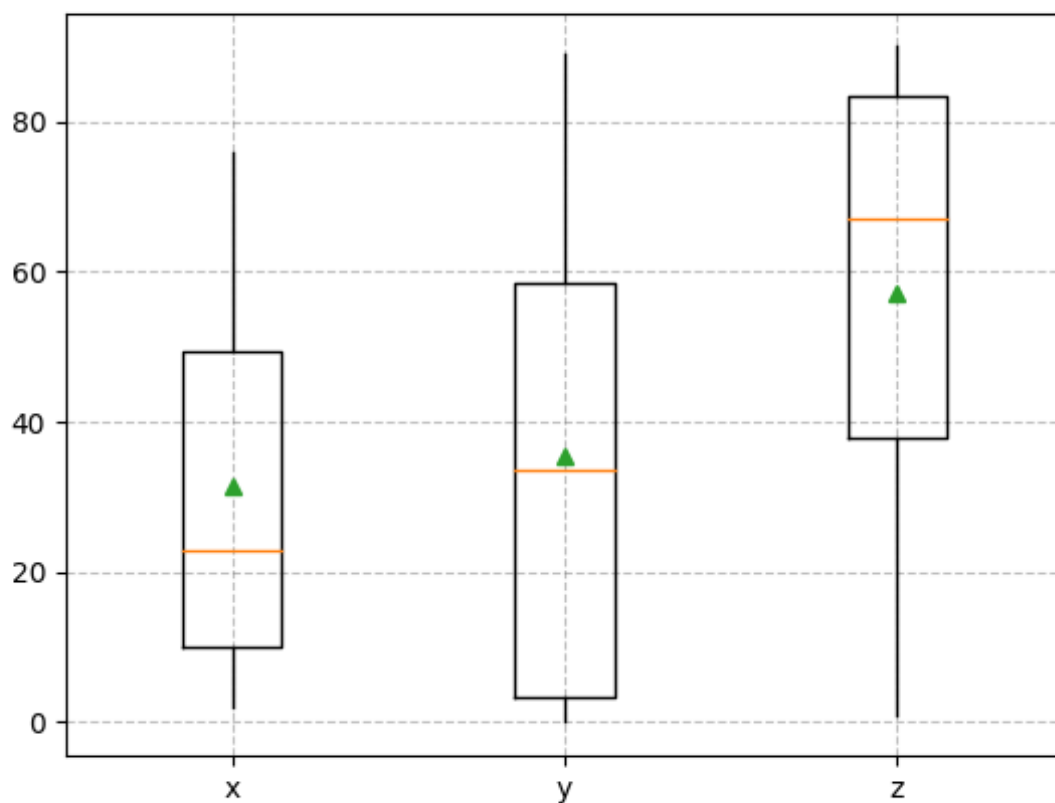
outliers - those data points which are abnormally far from its whole data points.

```
data -  
x = [8, 56, 23, 12, 2, 76, 43]  
#max = 76  
#min = 2  
#x = [2, 8, 12, 23, 43, 56, 76]  
#median = 23
```

```
import matplotlib.pyplot as plt  
import numpy as np  
x = [8, 56, 23, 12, 2, 76, 43]  
y = [4, 89, 45, 56, 22, 1, 0, 66]  
z = [33, 90, 1, 43, 67, 89, 78]  
data = [x, y, z]  
#triangles in box shows the mean of the data  
#line middle of the box shows the median of data  
plt.boxplot(data, showmeans=True)  
plt.show()
```



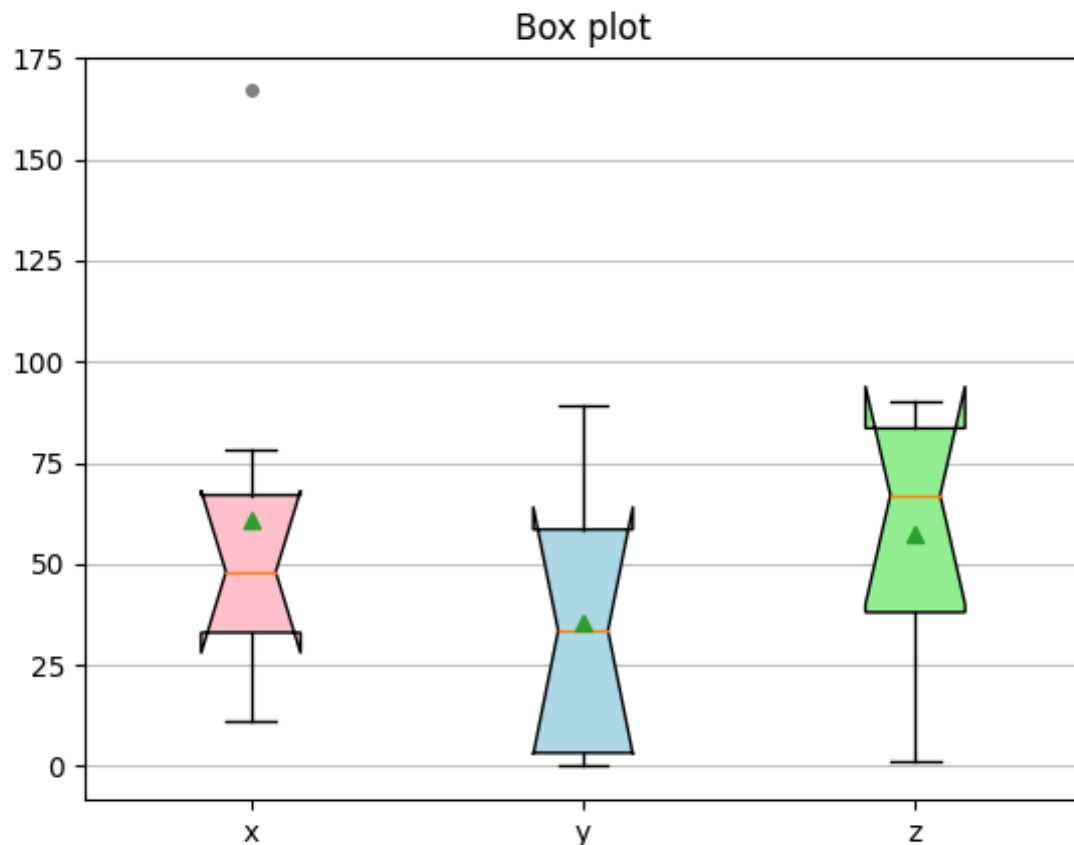
```
#showcaps parameter is use to show caps on whiskers, takes only
boolean value(default True)
plt.boxplot(data, showmeans=True, showcaps=False)
plt.grid(visible=True, color='grey', alpha=0.5, linestyle='--')
plt.xticks([1, 2, 3], ['x', 'y', 'z'])
plt.show()
```



```

#outliers
x = [48, 78, 43, 23, 11, 56, 167]
y = [4, 89, 45, 56, 22, 1, 0, 66]
z = [33, 90, 1, 43, 67, 89, 78]
data = [x, y, z]
marker_properties = {'markerfacecolor':'grey',
                    'markeredgecolor':'white', 'marker':'o'}
#markerprops used to customise marker properties.
#notch(default False) makes a notch in box at median
value.
#patch_artist(default False) used to customise properties of
box.
box = plt.boxplot(data, showmeans=True, showfliers=True,
                  patch_artist=True, flierprops=marker_properties, notch=True)
colors = ['pink', 'lightblue', 'lightgreen']
#set face color of boxes
for patch, color in zip(box['boxes'], colors):
    patch.set_facecolor(color)
plt.xticks([1, 2, 3], ['x', 'y', 'z'])
plt.grid(visible=True, color='grey', alpha=0.5, linestyle='-',
        axis='y')
plt.title('Box plot')
plt.show()

```



```
#horizontal boxplot
```

```
x = np.random.normal(0, 40, 80)
y = np.random.normal(0, 60, 80)
z = np.random.normal(0, 80, 80)
data = [x, y, z]
marker_properties = {'markerfacecolor':'grey',
                    'markeredgecolor':'white', 'marker':'o'}
#vert(by default True) argument is used plot boxes
horizontally, takes only boolean value.
box = plt.boxplot(data, showmeans=True, showfliers=True,
                  patch_artist=True, flierprops=marker_properties ,notch=True,
                  vert=False)
colors = ['pink', 'lightblue', 'lightgreen']
for patch, color in zip(box['boxes'], colors):
    patch.set_facecolor(color)
plt.yticks([1, 2, 3], ['x', 'y', 'z'])
plt.grid(visible=True, color='grey', alpha=0.5, linestyle='-',
        axis='x')
plt.title('Box plot')
plt.show()
```

