Histograms

To create a histogram, you need a continuous variable. Seldom histograms are used for discrete variables as well. Histograms are univariate i.e one requires only one variable to plot a histogram.

A histogram is a bar plot where the axis representing the data variable is divided into a set of discrete bins and the count of observations falling within each bin is shown using the height of the corresponding bar. Histograms are also known as frequency distribution plots and it is a very common practice to use histograms to check the distribution of the feature or if the data is positively skewed or negatively skewed.

DRAWBACKS: Histograms are bin biased. You can change the number of bins in a histogram, but there is no *optimal number of bins*. Sometimes we use fewer bins and sometimes more, we cannot simply rely on the default bins on whichever software we wish to use to create a histogram.

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
plt.style.use('fivethirtyeight')
#run this to change style to default - plt.style.use('default')
```

In [3]:

```
#loading the dataset
scores = pd.read_csv(r'E:\Downloads\StudentHoursScores.csv')
scores.head()
```

Out[3]:

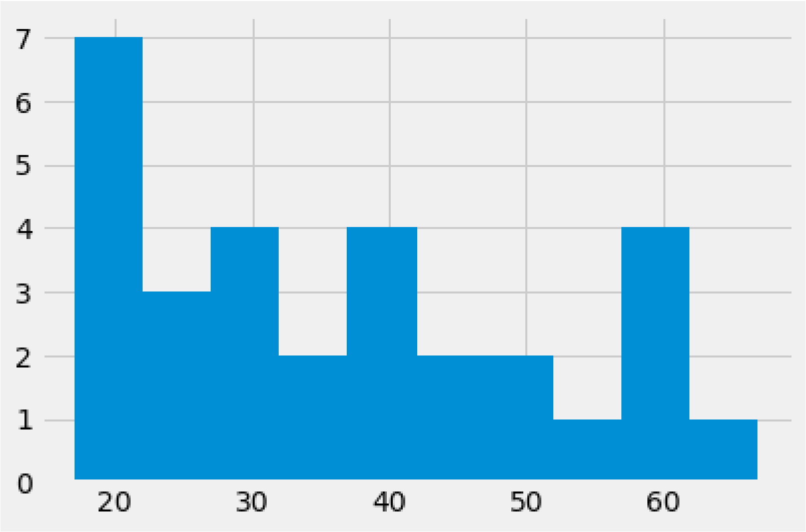
	Hours	Scores
0	7.7	79
1	5.9	60
2	4.5	45
3	3.3	33
4	1.1	12

In [4]:

```
sample_ages = [18, 20, 19, 21, 18, 19, 17, 23, 25,
               27, 29, 30, 35, 44, 56, 67, 33, 42, 48, 49,
              38, 40, 39, 39, 58, 59, 60, 58, 24, 29]
```

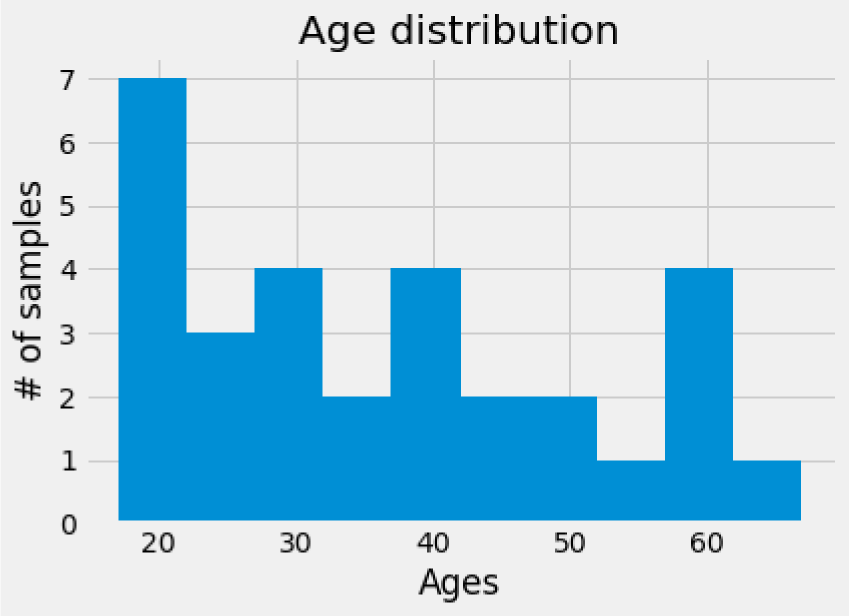
In [5]:

```
plt.hist(sample_ages)
plt.show()
```



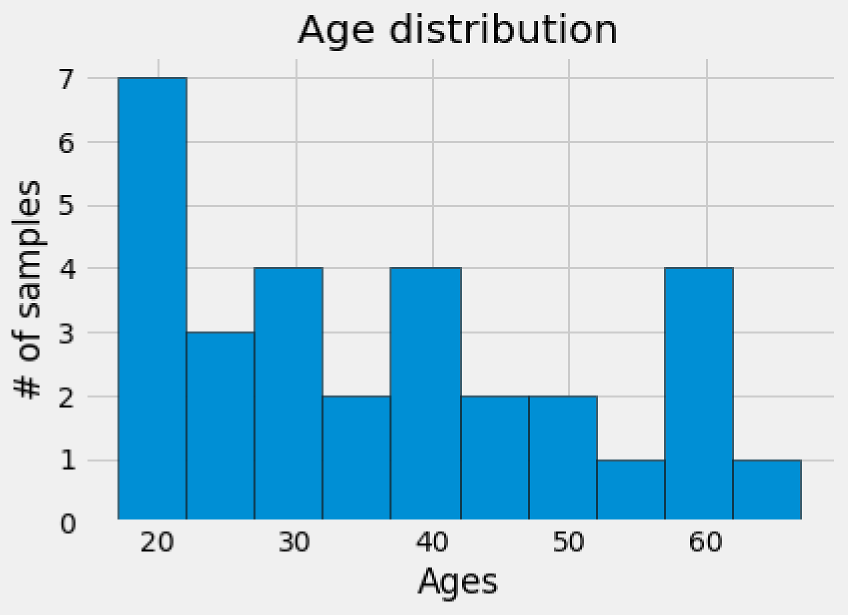
In [6]:

```
plt.hist(sample_ages)
plt.xlabel('Ages')
plt.ylabel('# of samples')
plt.title('Age distribution')
plt.show()
```



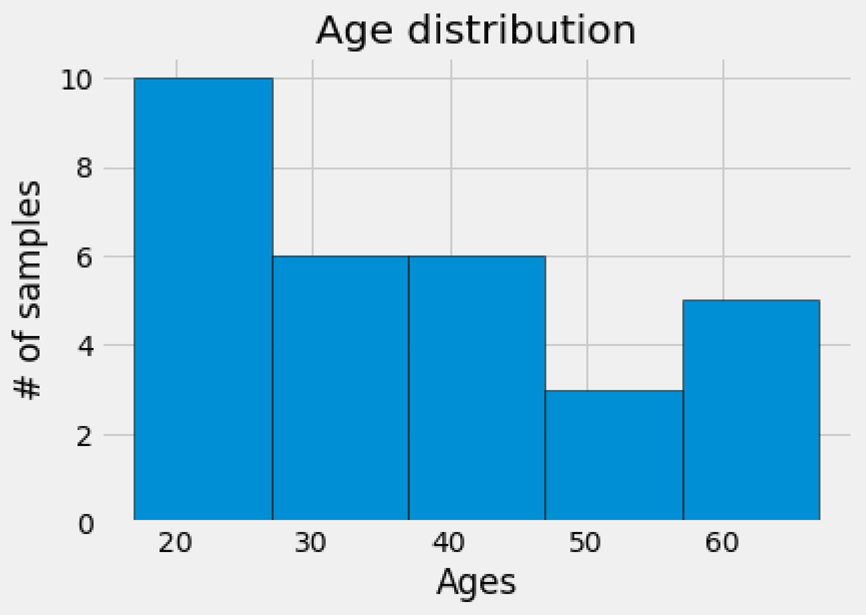
In [7]:

```
plt.hist(sample_ages, ec = 'k')
plt.xlabel('Ages')
plt.ylabel('# of samples')
plt.title('Age distribution')
plt.show()
```

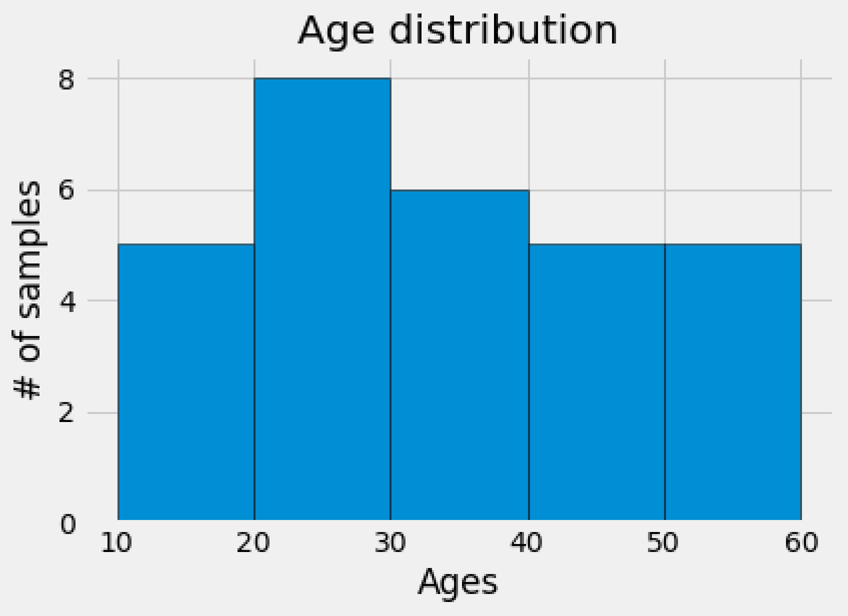


In [8]:

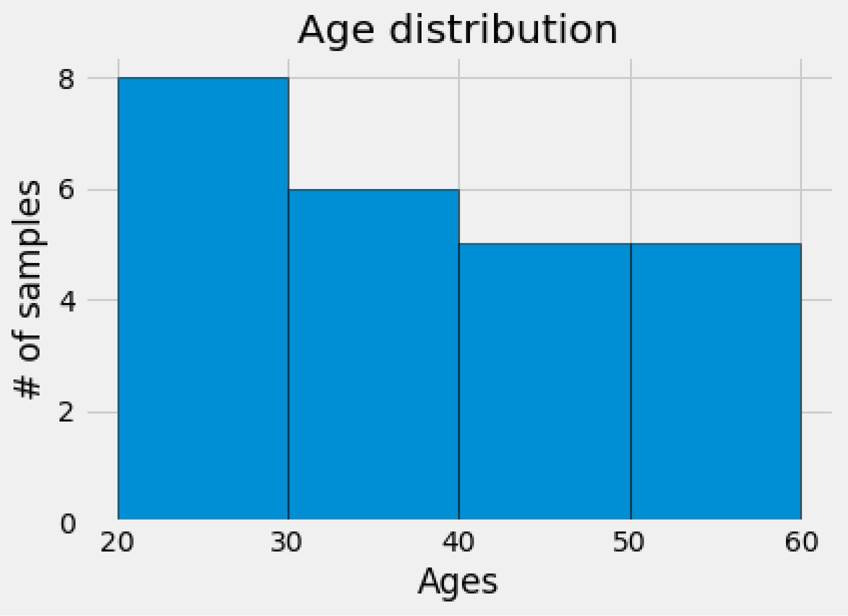
```
plt.hist(sample_ages, ec = 'k', bins = 5)
plt.xlabel('Ages')
plt.ylabel('# of samples')
plt.title('Age distribution')
plt.show()
```



In [9]:
```
plt.hist(sample_ages, ec = 'k', bins = [10,20,30,40,50,60])
plt.xlabel('Ages')
plt.ylabel('# of samples')
plt.title('Age distribution')
plt.show()
```



In [10]:
```
plt.hist(sample_ages, ec = 'k', bins = [20,30,40,50,60])
plt.xlabel('Ages')
plt.ylabel('# of samples')
plt.title('Age distribution')
plt.show()
```

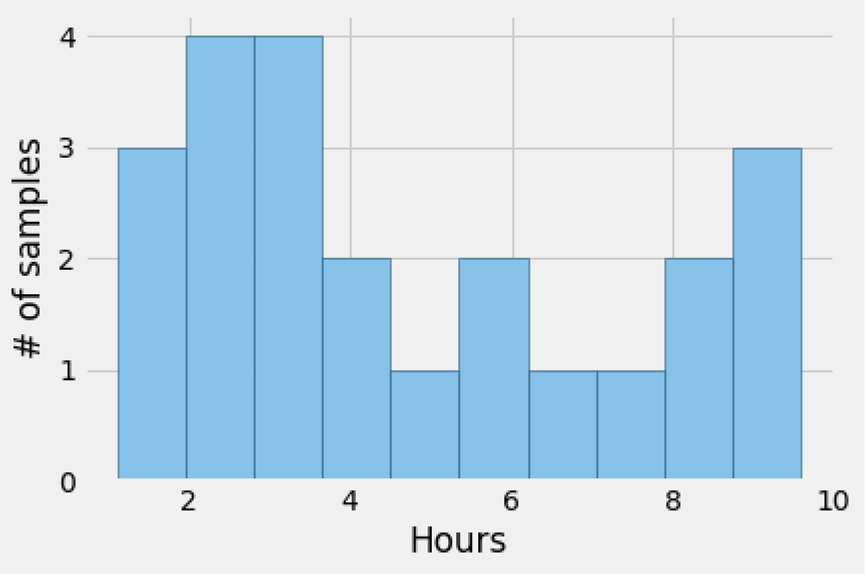


In [11]:
```
scores.head()
```

Out[11]:

	Hours	Scores
0	7.7	79
1	5.9	60
2	4.5	45
3	3.3	33
4	1.1	12

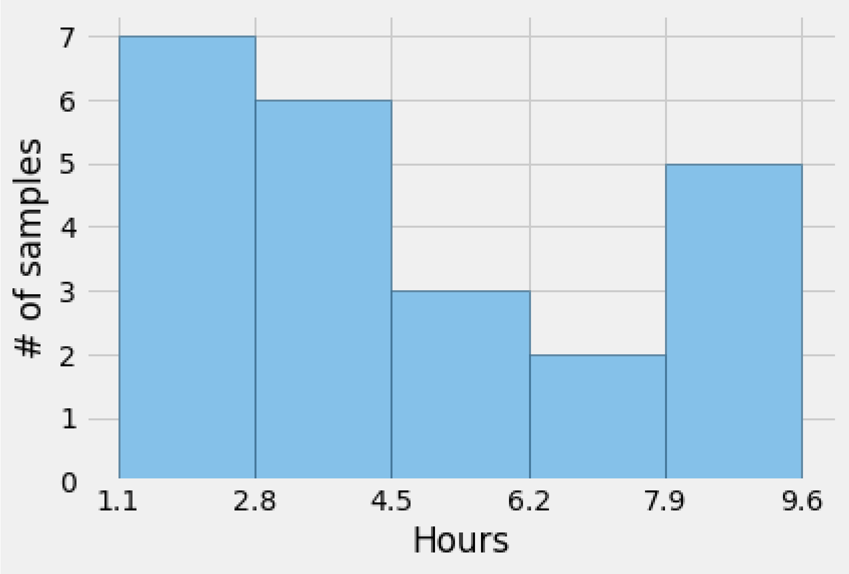
In [12]:
```
plt.hist(scores['Hours'], color = '#85C1E9', ec = '#1B4F72')
plt.xlabel('Hours')
plt.ylabel('# of samples')
plt.show()
```



In [13]:
```
count, bin_edges = np.histogram(scores['Hours'], bins = 5)
```

In [14]:

```
plt.hist(scores['Hours'], color = '#85C1E9', ec = '#1B4F72', bins = 5)
plt.xlabel('Hours')
plt.ylabel('# of samples')
plt.xticks(bin_edges)
plt.show()
```

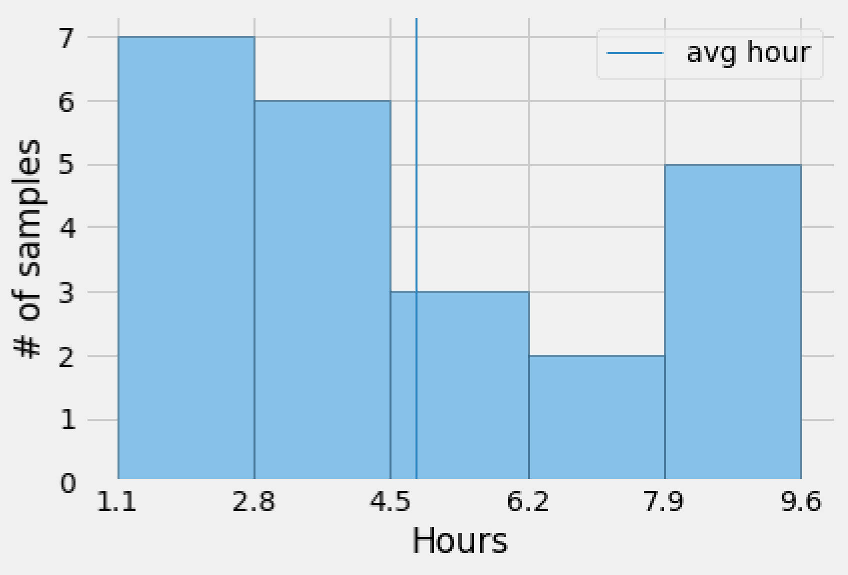


In [15]:

```
hour_mean = scores['Hours'].mean()
```

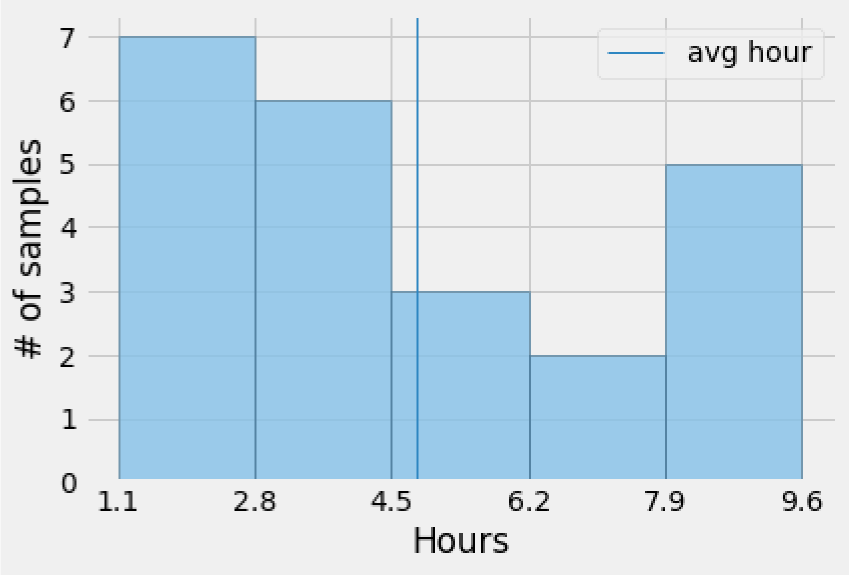
In [16]:

```
plt.hist(scores['Hours'], color = '#85C1E9', ec = '#1B4F72', bins = 5)
plt.xlabel('Hours')
plt.ylabel('# of samples')
plt.xticks(bin_edges)
plt.axvline(hour_mean, label = 'avg hour',
            color = '#2E86C1', linewidth = 1)
plt.legend()
plt.show()
```



In [17]:

```
plt.hist(scores['Hours'], color = '#85C1E9', ec = '#1B4F72', bins = 5,
        alpha = 0.8)
plt.xlabel('Hours')
plt.ylabel('# of samples')
plt.xticks(bin_edges)
plt.axvline(hour_mean, label = 'avg hour',
            color = '#2E86C1', linewidth = 1)
plt.legend()
plt.show()
```


avg hour
# of samples
Hours