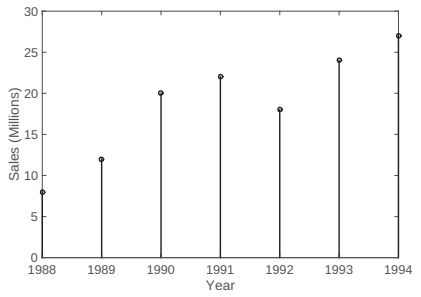
Stem Plot

Function Format

```
stem(x,y)
```




```
yr=[1988:1994];
sle=[8 12 20 22 18 24 27];
stem(yr,sle)
xlabel('Year')
ylabel('Sales (Mil-
lions)')
```

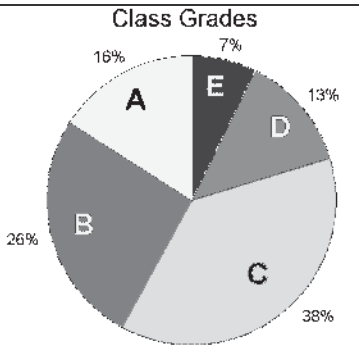
Pie charts are useful for visualizing the relative sizes of different but related quantities. For example, the table below shows the grades that were assigned to a class. The data is used to create the pie chart that follows.

Grade	A	B	C	D	E
Number of students	11	18	26	9	5

Pie Plot

Function format:

```
pie(x)
```




```
grd=[11 18 26 9 5];
pie(grd)
title('Class Grades')
```

MATLAB draws the sections in different colors. The letters (grades) were added using the Plot Editor.

5.8 HISTOGRAMS

Histograms are plots that show the distribution of data. The overall range of a given set of data points is divided into subranges (bins), and the histogram shows how many data points are in each bin. The histogram is a vertical bar plot in which the width of each bar is equal to the range of the corresponding bin and the height of the bar corresponds to the number of data points in the bin. Histograms are created in MATLAB with the `hist` command. The simplest form of the command is:

```
hist(y)
```

`y` is a vector with the data points. MATLAB divides the range of the data points into 10 equally spaced subranges (bins) and then plots the number of data points in each bin.

For example, the following data points are the daily maximum temperature (in °F) in Washington, DC, during the month of April 2002: 58 73 73 53 50 48

56 73 73 66 69 63 74 82 84 91 93 89 91 80 59 69 56 64 63 66 64 74 63 69 (data from the U.S. National Oceanic and Atmospheric Administration). A histogram of this data is obtained with the commands:

```
>> y=[58 73 73 53 50 48 56 73 73 66 69 63 74 82 84 91 93 89
91 80 59 69 56 64 63 66 64 74 63 69];
>> hist(y)
```

The plot that is generated is shown in Figure 5-11 (the axis titles were added using the Plot Editor). The smallest value in the data set is 48 and the largest is 93, which means that the range is 45 and the width of each bin is 4.5. The range of the first bin is from 48 to 52.5 and contains two points. The range of the second bin is from 52.5 to 57 and contains three points, and so on. Two of the bins (75 to 79.5 and 84 to 88.5) do not contain any points.

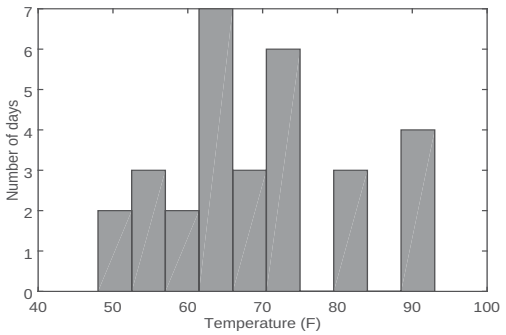


Figure 5-11: Histogram of temperature data.

Since the division of the data range into 10 equally spaced bins might not be the division that is preferred by the user, the number of bins can be defined to be different than 10. This can be done either by specifying the number of bins, or by specifying the center point of each bin as shown in the following two forms of the `hist` command:

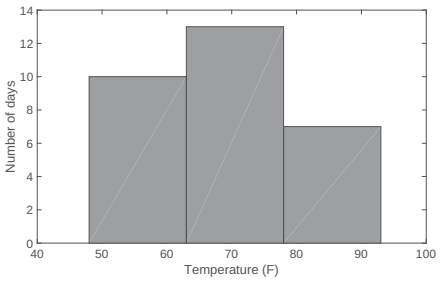
`hist(y,nbins)` or `hist(y,x)`

`nbins` is a scalar that defines the number of bins. MATLAB divides the range in equally spaced subranges.

`x` is a vector that specifies the location of the center of each bin (the distance between the centers does not have to be the same for all the bins). The edges of the bins are at the middle point between the centers.

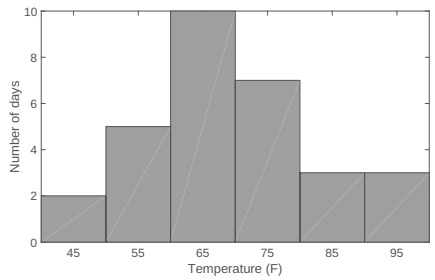
In the example above the user might prefer to divide the temperature range into three bins. This can be done with the command:

```
>> hist(y,3)
```



As shown in the top graph, the histogram that is generated has three equally spaced bins.

The number and width of the bins can also be specified by a vector `x` whose elements define the centers of the bins. For example, shown in the lower graph is a histogram that displays the temperature data from above in six bins with an equal width of 10 degrees. The elements of the vector `x` for this plot are 45, 55, 65, 75, 85, and 95. The plot was obtained with the following commands:




```
>> x=[45:10:95]
x =
   45   55   65   75   85   95
>> hist(y,x)
```

The `hist` command can be used with options that provide numerical output in addition to plotting a histogram. An output of the number of data points in each bin can be obtained with one of the following commands:

`n=hist(y)` `n=hist(y,nbins)` `n=hist(y,x)`

The output `n` is a vector. The number of elements in `n` is equal to the number of bins, and the value of each element of `n` is the number of data points (frequency count) in the corresponding bin. For example, the histogram in Figure 5-11 can also be created with the following command:

```
>> n = hist(y)
n =
   2   3   2   7   3   6   0   3   0   4
```

The vector `n` shows how many elements are in each bin.

The vector `n` shows that the first bin has two data points, the second bin has three data points, and so on.