Chapter 7: Statistics, Probability, and Interpolation

- Engineers use statistics to predict behavior of physical systems that have randomness.
- Histograms are used to present the frequency that data occurs.
- The Normal Distribution is a particular way of modeling randomness.
- Random number generators within MATLAB can be used to generate models for accounting for randomness.
- Engineers often have to use interpolation between data points for analysis.

Histograms

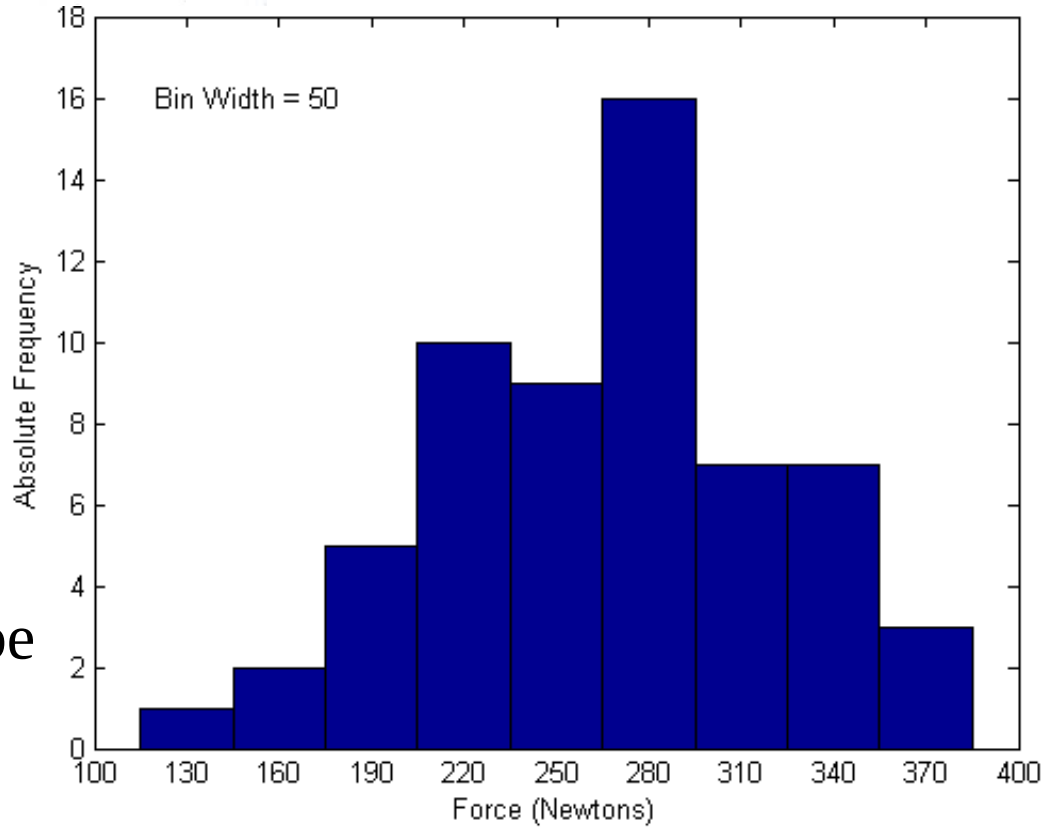
3. The following list gives the measured breaking force in newtons for a sample of 60 pieces of certain type of cord. Plot the absolute frequency histogram. Try bin widths of 10, 30, and 50 N. Which gives the most meaningful histogram? Try to find a better value for the bin width.

311	138	340	199	270	255	332	279	231	296	198	269
257	236	313	281	288	225	216	250	259	323	280	205
279	159	276	354	278	221	192	281	204	361	321	282
254	273	334	172	240	327	261	282	208	213	299	318
356	269	355	232	275	234	267	240	331	222	370	226

Histograms show the frequency of the occurrence of the data versus the data itself.

The data are grouped into subranges called Bins.

The shape of the histogram can be affected by the width of the bins.



Histograms

The Absolute Frequency is the number of times a particular value for a variable has been observed to occur.

The Relative Frequency is calculated by dividing the Absolute Frequency by the total number of values for the variable.

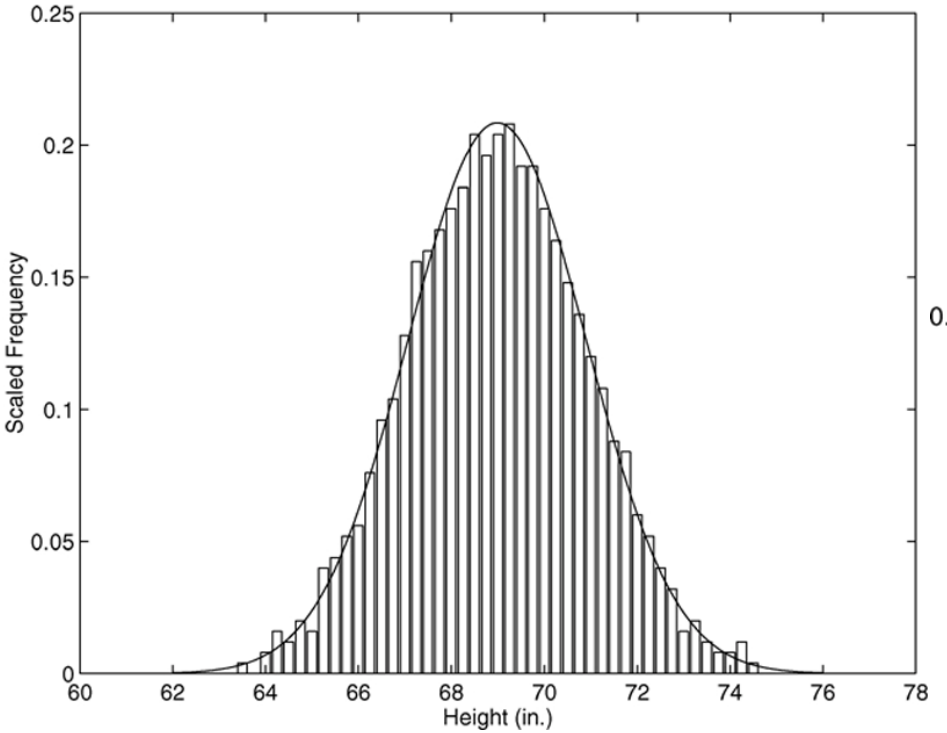
A Scaled Frequency Histogram is the Absolute Frequency Histogram divided by the total area of the histogram. Since the total area under the histogram rectangles is one, the area under the rectangles for a particular range is the probability of occurrence.

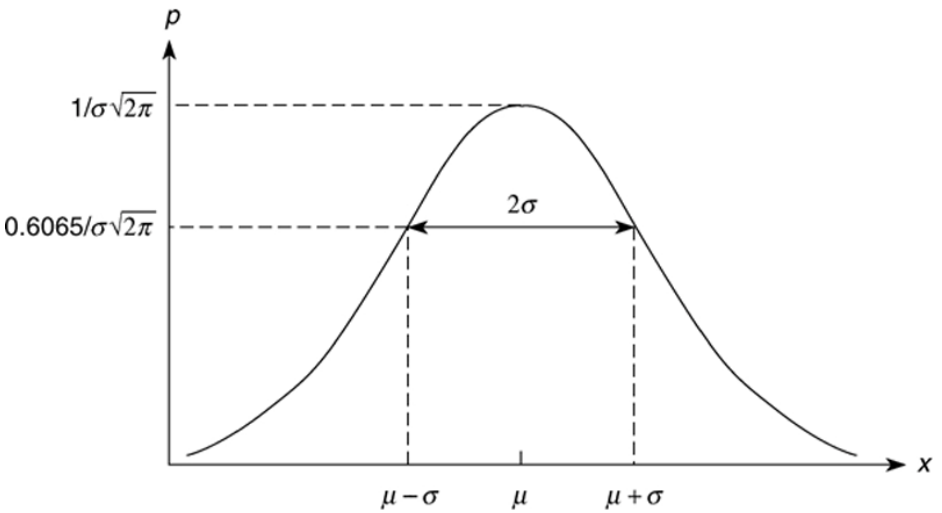
Normal Distribution

For Scaled Frequency Histograms with vast amounts of data, the plot can be approximated as a continuous variable as opposed to discrete rectangles (Bell-Shaped Curve). The Normal Probability Function is:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

where μ is the Mean, σ is the Standard Deviation, and σ^2 is the Variance.





Normal Distribution

For a **Normal Probability Distribution Curve**, it can be shown that approximately 68 percent of the area lies between the limits of:

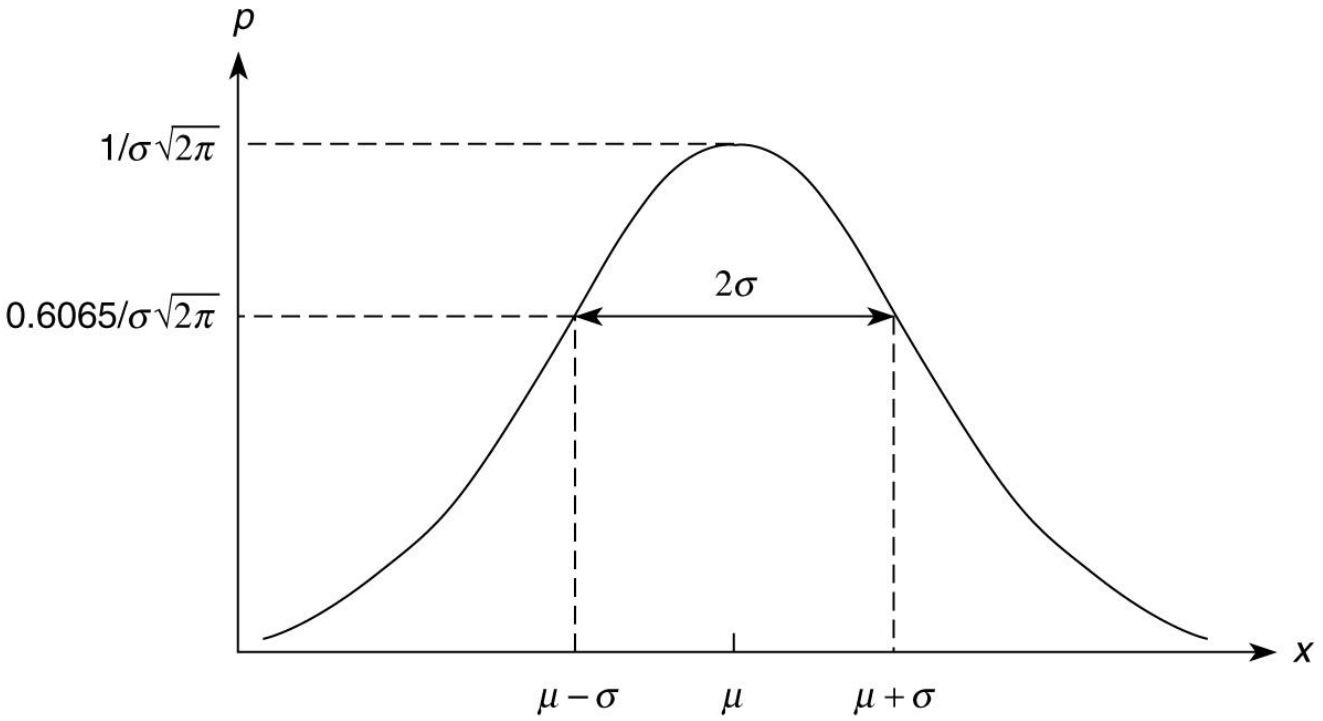
$$\mu - \sigma \leq x \leq \mu + \sigma$$

Approximately 96 percent of the area lies between the limits of

$$\mu - 2\sigma \leq x \leq \mu + 2\sigma$$

Approximately 99.7 percent of the area lies between the limits of

$$\mu - 3\sigma \leq x \leq \mu + 3\sigma$$



Normal Distribution

For Normally Distributed outcomes, the probability that a random variable x is less than or equal to b is:

$$P(x \leq b) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{b-\mu}{\sigma\sqrt{2}}\right)\right]$$

The probability that the random variable x is no less than a and no greater than b is:

$$P(a \leq x \leq b) = \frac{1}{2}\left[\text{erf}\left(\frac{b-\mu}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{a-\mu}{\sigma\sqrt{2}}\right)\right]$$

where erf (x) is the Error Function.

Random Number Generation

MATLAB can generate random numbers that are uniformly distributed or normally distributed. These sets of random numbers can be used to analyze outcomes.

x = rand(n): generates uniformly distributed random numbers in the range [0,1]. To generate uniformly distributed random numbers over the interval [a,b]:

$$y = (b - a)x + a$$

x = randn(n): generates normally distributed random numbers that have a mean of $\mu = 0$ and a standard deviation of $\sigma = 1$. To generate normally distributed random numbers that have a mean of μ and a standard deviation of σ:

$$y = \sigma x + \mu$$

Interpolation

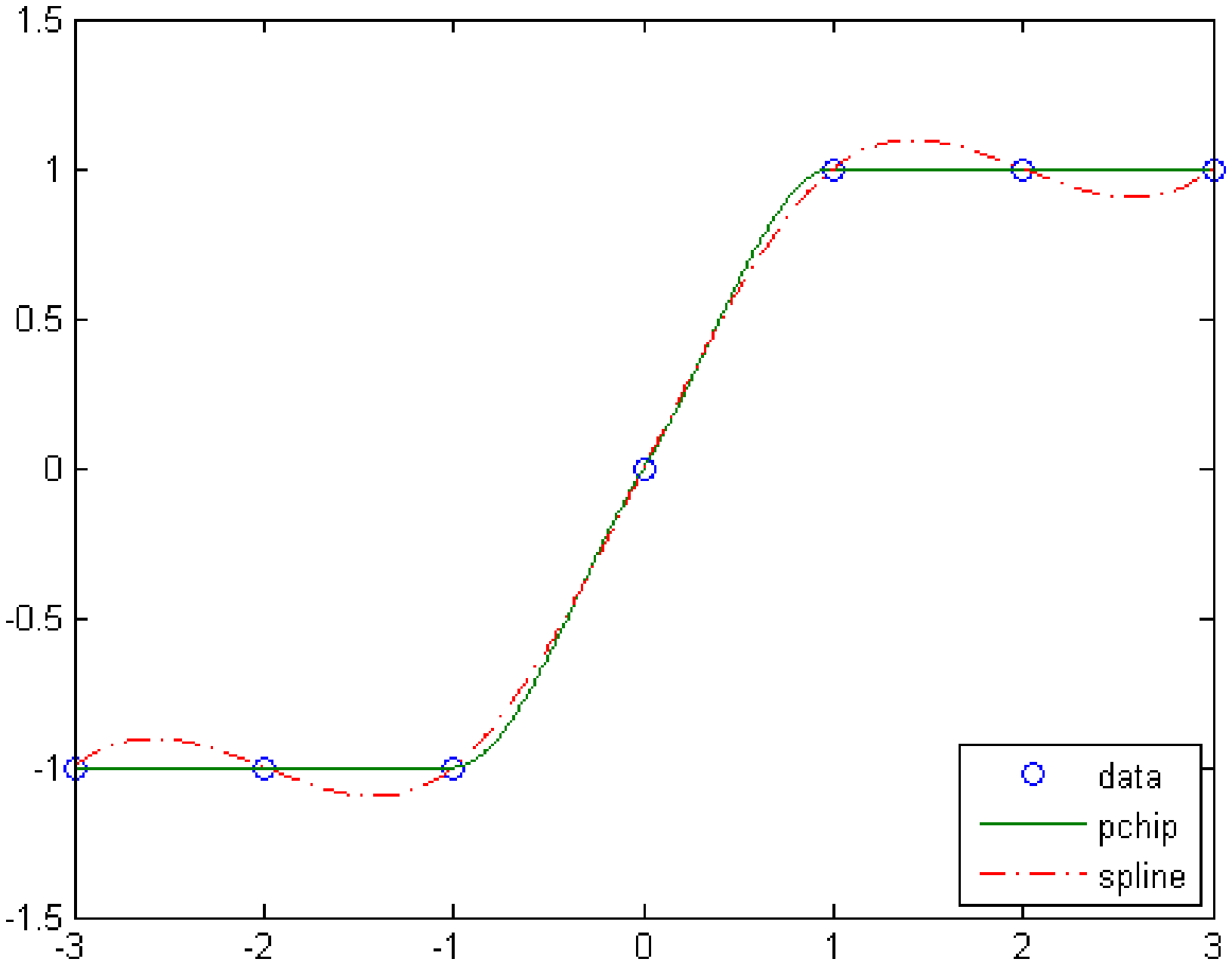
To estimate values between data points, several types of Interpolation can be used.

Linear Interpolation: The straight line connecting the two points are used for estimates of intermediate values.

Cubic Spline Interpolation: A third-order equation is derived between successive sets of three data points.

Piecewise Continuous Hermite Interpolation Polynomials (PCHIP): Less Overshoot than Cubic Spline.

Interpolation



The grade data below show how many students received A's, B's, etc. Download the student grade data to your computer. Place the file into the same folder as your matlab file:
http://cecs.wright.edu/people/faculty/sthomas/studentgrades.xlsx

Final Calculated Grade
98.8 / 100
59.2 / 100
24.7 / 100
32 / 100
92.4 / 100
74.3 / 100
46.3 / 100
65.2 / 100
29.3 / 100
92.9 / 100
33.1 / 100
53.5 / 100
116.1 / 100
99.5 / 100
75.3 / 100
87.8 / 100
50.1 / 100
72.2 / 100

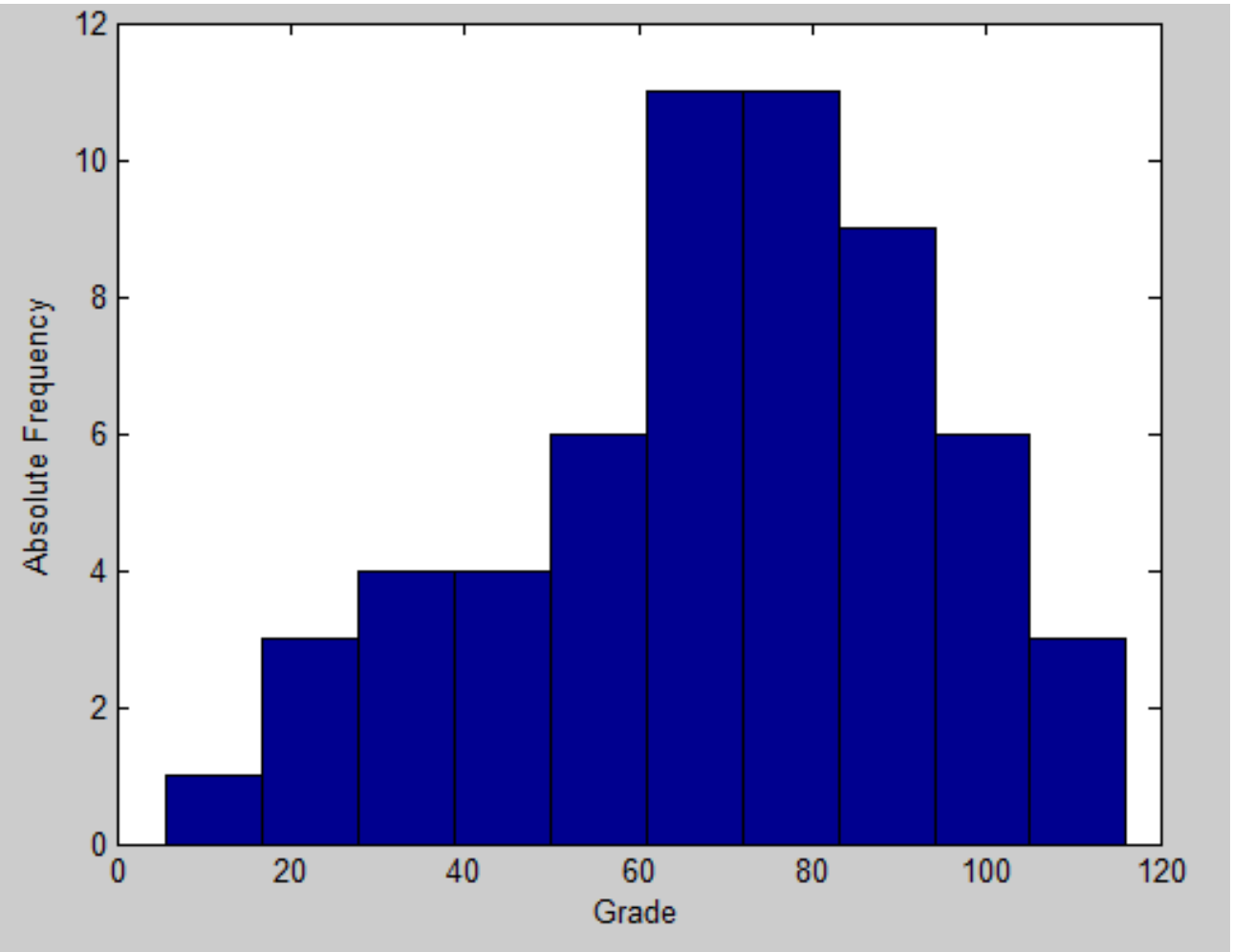
86.4 / 100	61.3 / 100	49.7 / 100	93.5 / 100
79.6 / 100	88.7 / 100	67.9 / 100	82.3 / 100
44.1 / 100	107.9 / 100	59 / 100	34 / 100
88.4 / 100	83.7 / 100	100 / 100	20.9 / 100
80.3 / 100	56.1 / 100	78.4 / 100	78.2 / 100
101.4 / 100	87 / 100	69.2 / 100	81.6 / 100
99 / 100	70.3 / 100	78.7 / 100	18.5 / 100
62.4 / 100	55.4 / 100	46.7 / 100	64.4 / 100
73.3 / 100	64.2 / 100	107.6 / 100	68.5 / 100
70.9 / 100	101.6 / 100	5.7 / 100	68.7 / 100

	A	B	C	D	F
Grade	>= 90.0	89.9 to 80.0	79.9 to 70.0	69.9 to 60.0	< 60.0
# of Students	12	9	10	9	18

Each grade can be thought of as a **Bin**. The number of grades in each **Bin** is called the **Absolute Frequency**. A plot of the **Absolute Frequency** versus the **Bin Range** is called a **Histogram**.

Use the following MATLAB Script file to create a **Histogram Plot** of the data:

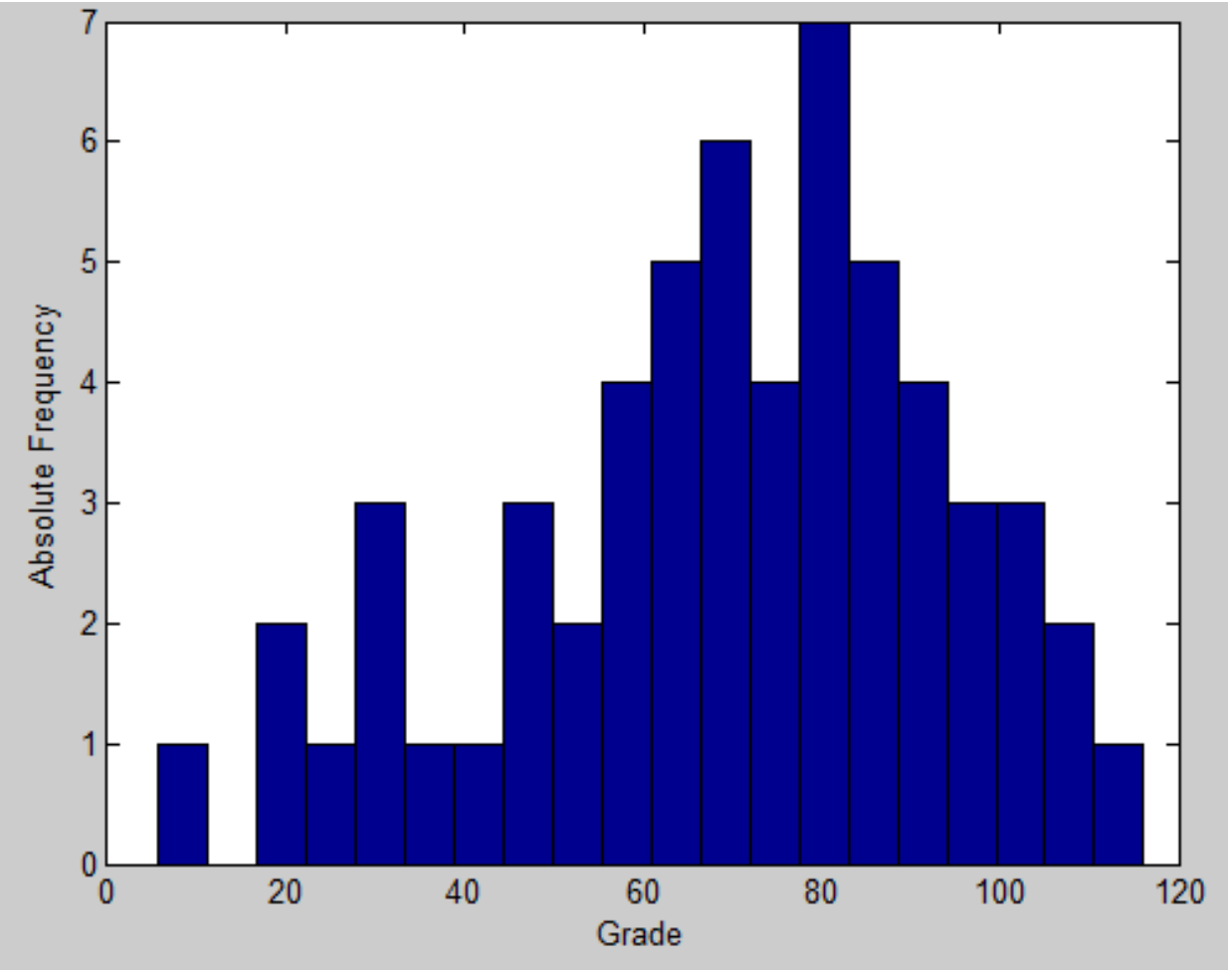
```
grades = xlsread('studentgrades');
hist(grades)
xlabel('Grade'), ylabel('Absolute Frequency')
```



This method simply divides the **Range** into **10 Bins** and plots the **Absolute Frequency** in each **Bin**.

Modify the MATLAB file as follows:

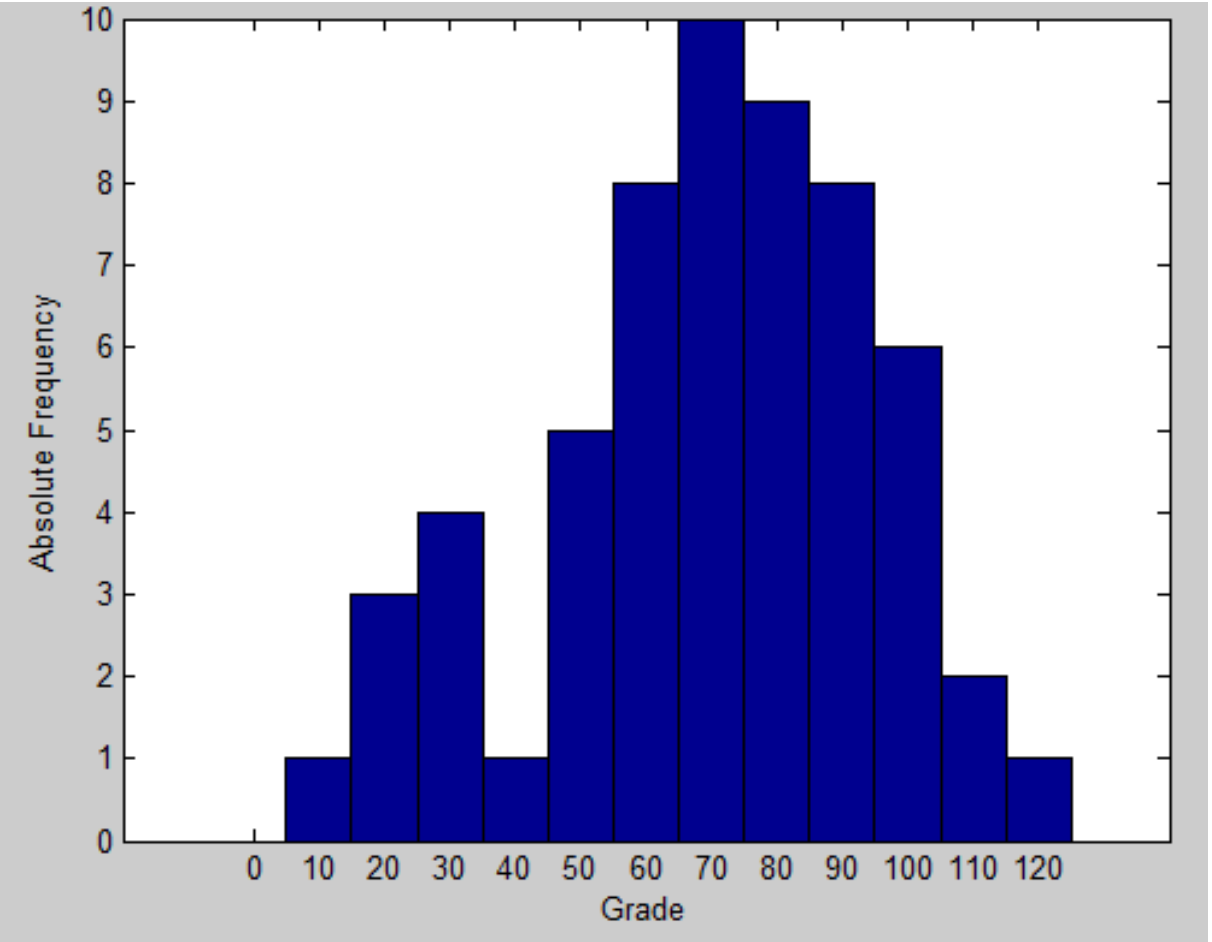
```
grades = xlsread('studentgrades');
n = 20
hist(grades,n)
xlabel('Grade'), ylabel('Absolute Frequency')
```



In this **Histogram Plot**, we have specified the number of **Bins** to be **20**.

Modify the MATLAB file as follows:

```
grades = xlsread('studentgrades');
x = 0:10:120
hist(grades,x)
xlabel('Grade'), ylabel('Absolute Frequency')
```



In this **Histogram Plot**, the **Bin Centers** are defined by the vector **x**, and the **Bin Width** is the distance between centers.

The **Ranges** of the **Bins** are:

$\text{Grade} \leq 5$

$5 < \text{Grade} \leq 15$

$15 < \text{Grade} \leq 25$

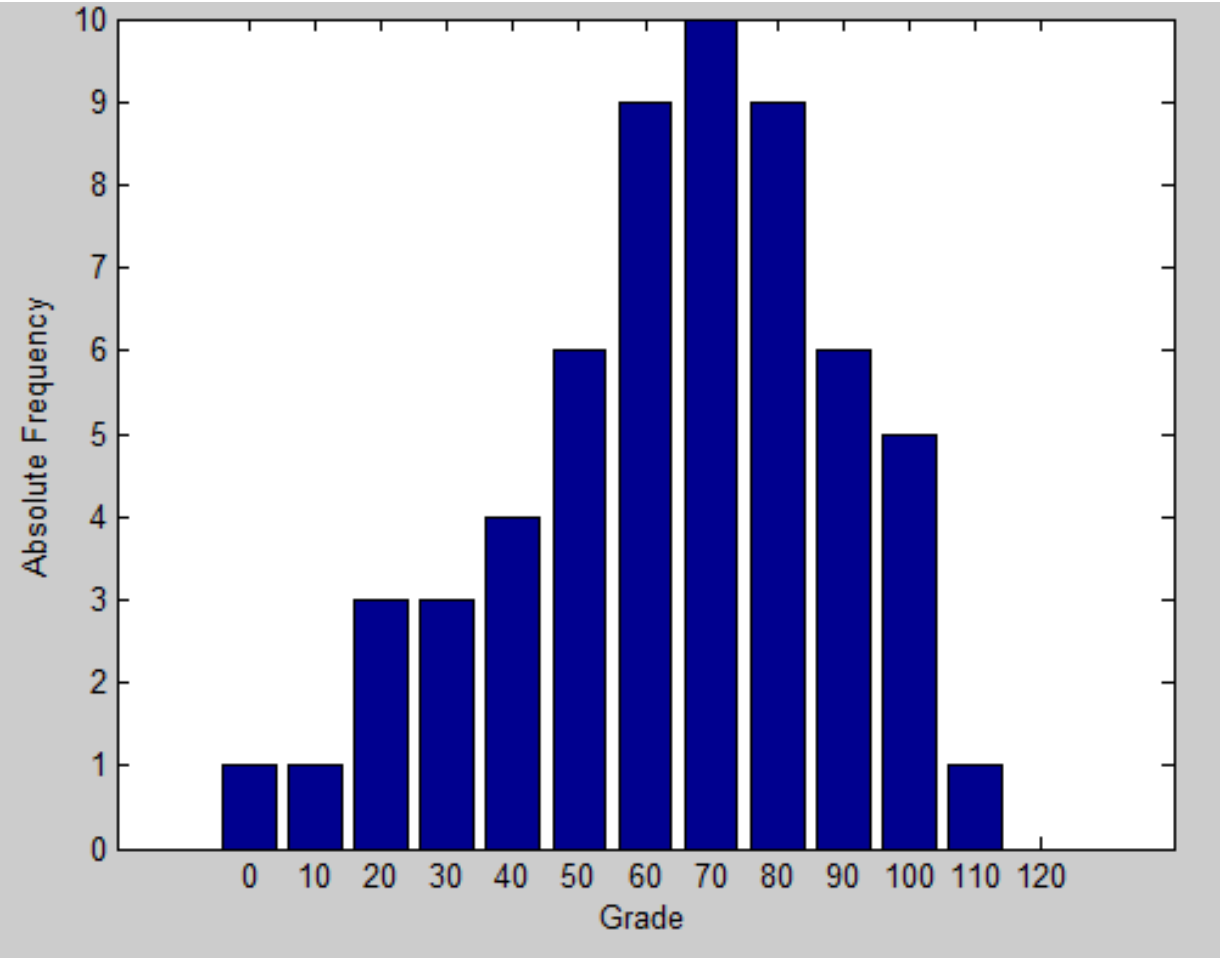
$\vdots$

$105 < \text{Grade} \leq 115$

$\text{Grade} > 115$

Modify the MATLAB file as follows:

```
grades = xlsread('studentgrades');
edges = [0 10 20 30 40 50 60 70 80 90 100 110 120]
n = histc(grades,edges)
bar(edges,n)
xlabel('Grade'), ylabel('Absolute Frequency')
```

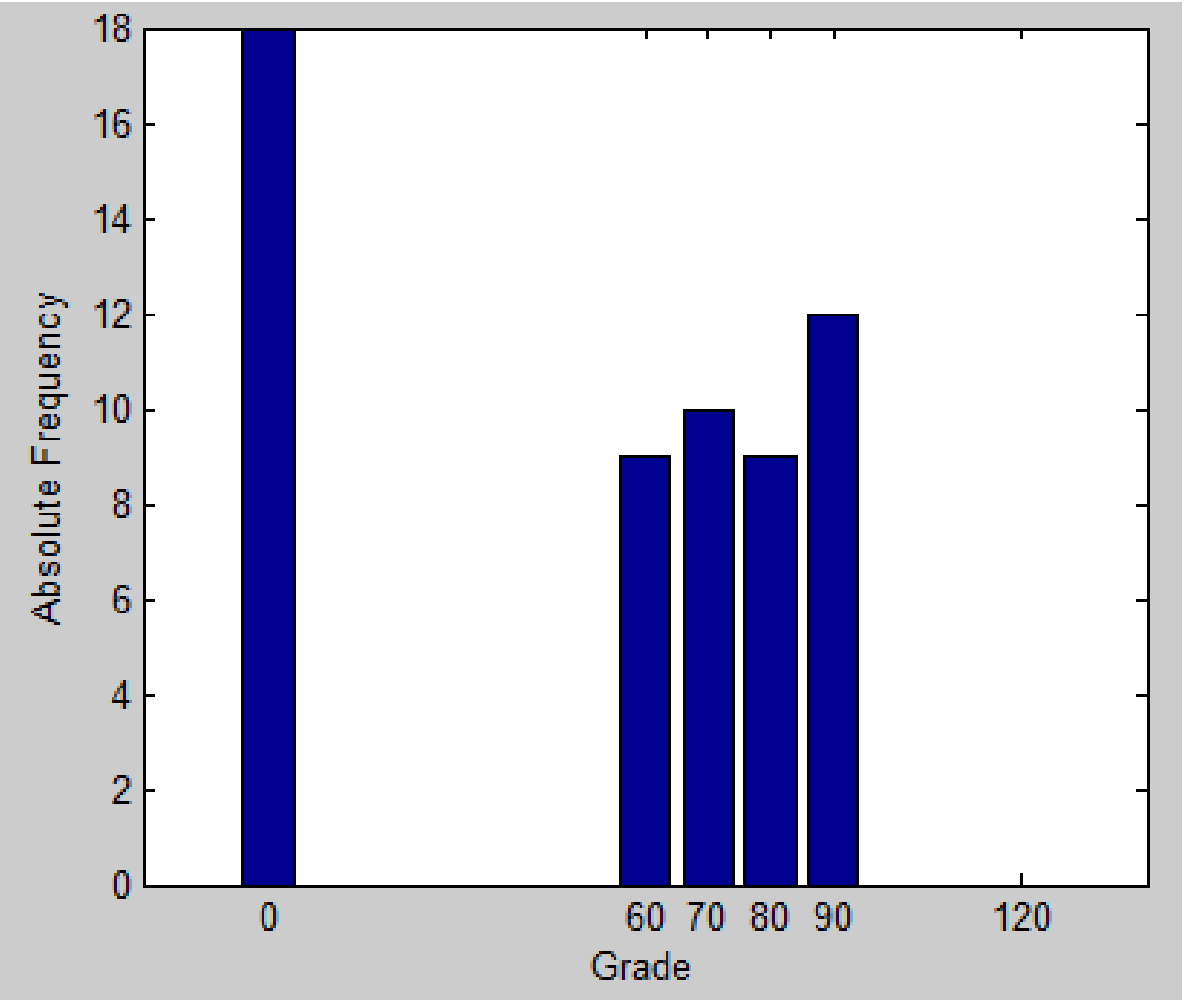


This type of **Histogram Plot** was created as a Bar Plot. The function **histc** was used because it made sense to define the **Bins** in terms of **Edges** instead of **Centers** for this problem. The **Ranges** of the **Bins** are:

$$0 \le \text{Grade} < 10$$
$$10 \le \text{Grade} < 20$$
$$\vdots$$
$$100 \le \text{Grade} < 110$$
$$110 \le \text{Grade} < 120$$

Modify the MATLAB file as follows:

```
grades = xlsread('studentgrades');
edges = [0 60 70 80 90 120]
n = histc(grades,edges)
bar(edges,n)
xlabel('Grade'), ylabel('Absolute Frequency')
```



This type of **Histogram Plot** specifies the **Edges** of the **Bins**.

The **Ranges** of the **Bins** are:

< 60

$60 \leq \text{Grade} < 70$

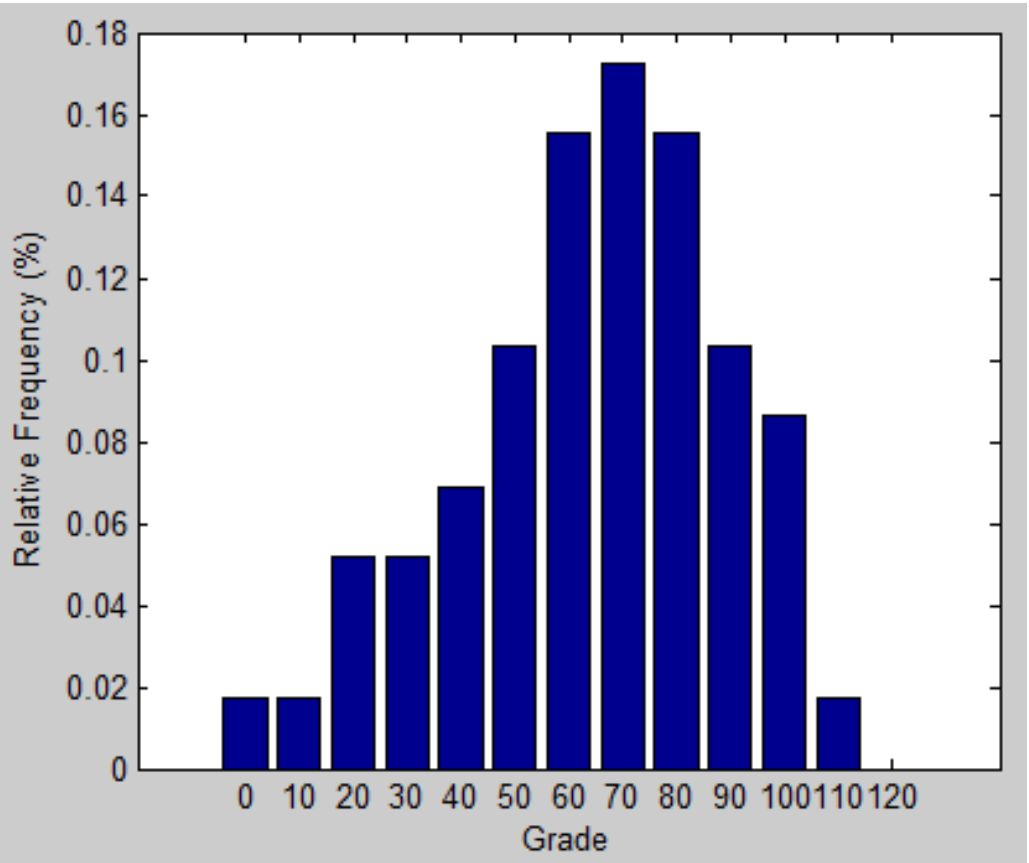
$70 \leq \text{Grade} < 80$

$80 \leq \text{Grade} < 90$

$\text{Grade} \geq 90$

Modify the MATLAB file as follows:

```
grades = xlsread('studentgrades');
edges = 0:10:120;
n = histc(grades,edges)
Number_Students = sum(n)
Rel_Freq = n/Number_Students
sum_Relative_Frequency = sum(Rel_Freq)
bar(edges,Rel_Freq)
xlabel('Grade'), ylabel('Relative Frequency (%)')
```



The total number of students in the class was 58. The **Relative Frequency** can be calculated by dividing the number in each bin by the total number of students.

Problem 7.4:

For the data given in Problem 1:

a. Plot the Absolute Frequency Histogram, the Relative Frequency Histogram, and the Scaled Frequency Histogram.

b. Compute the mean and standard deviation and use them to estimate the lower and upper limits of gas mileage corresponding to 68 percent of the cars of this model.

Go to the following webpage to download the data for this problem:
www.cs.wright.edu/~sthomas/prob7_1.xlsx

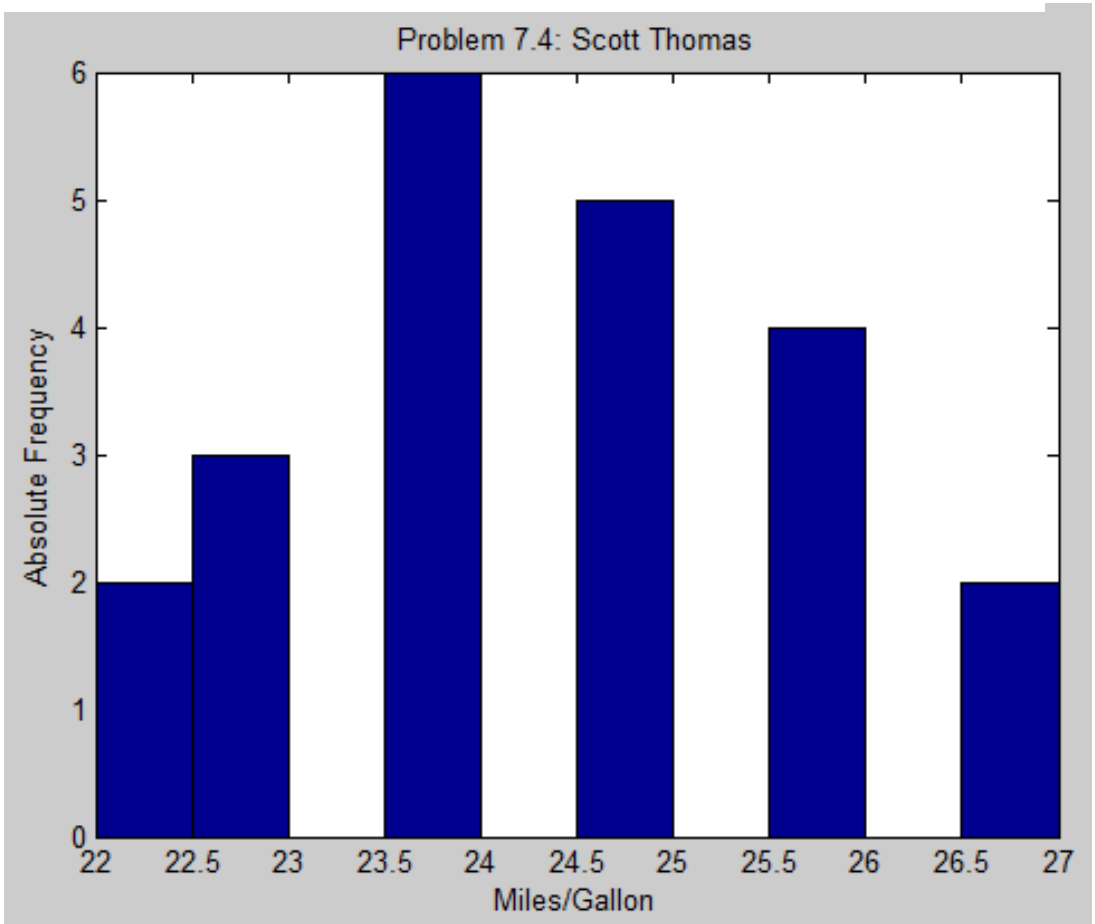
1. The following list gives the measured gas mileage in miles per gallon for 22 cars of the same model. Plot the absolute frequency histogram and the relative frequency histogram.

23	25	26	25	27	25	24	22	23	25	26
26	24	24	22	25	26	24	24	24	27	23

Problem 7.4:

Step 1: Read in the data with the **xlsread** command and use the **hist** command to plot the **Absolute Frequency Histogram**.

```
mpg = xlsread('prob7_1');
hist(mpg)
xlabel('Miles/Gallon'), ylabel('Absolute Frequency')
title('Problem 7.4: Scott Thomas')
```

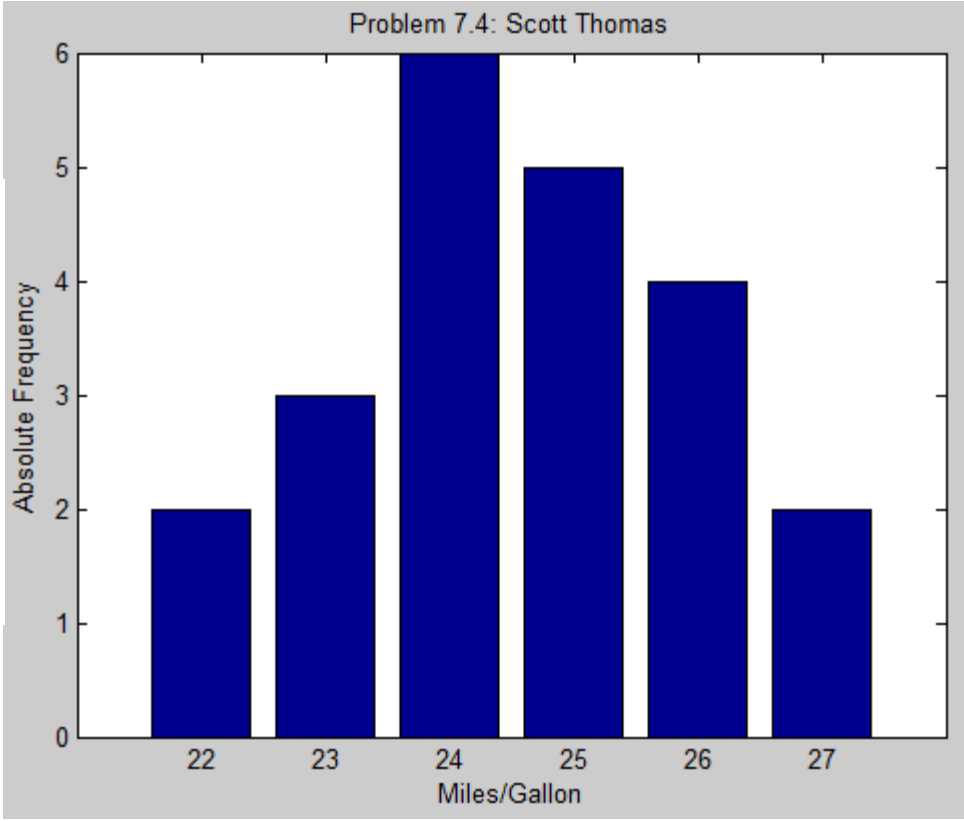


Problem 7.4:

Step 2: Change the bin centers using the **hist** command as shown, then plot the **Absolute Frequency Histogram** using the **bar** command.

`[n,xout] = hist(...)` returns vectors `n` and `xout` containing the frequency counts and the bin locations. You can use `bar(xout,n)` to plot the histogram.

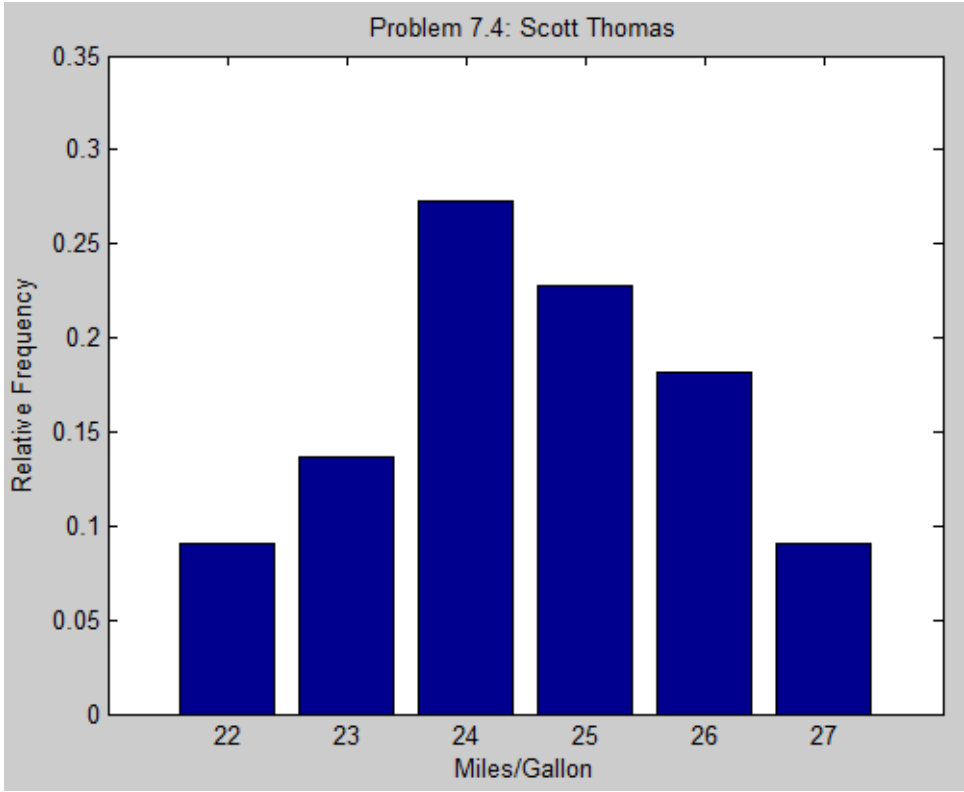
```
mpg = xlsread('prob7_1');
x = 22:27;
[n,xout] = hist(mpg,x)
bar(xout,n)
xlabel('Miles/Gallon')
ylabel('Absolute Frequency')
title('Problem 7.4: Scott Thomas')
```



Problem 7.4:

Step 3: The **Relative Frequency** is calculated by dividing the **Absolute Frequency** (the number of values in each bin) by the total number of values for the variable. Calculate the number of values and use it to plot the **Relative Frequency Histogram.**

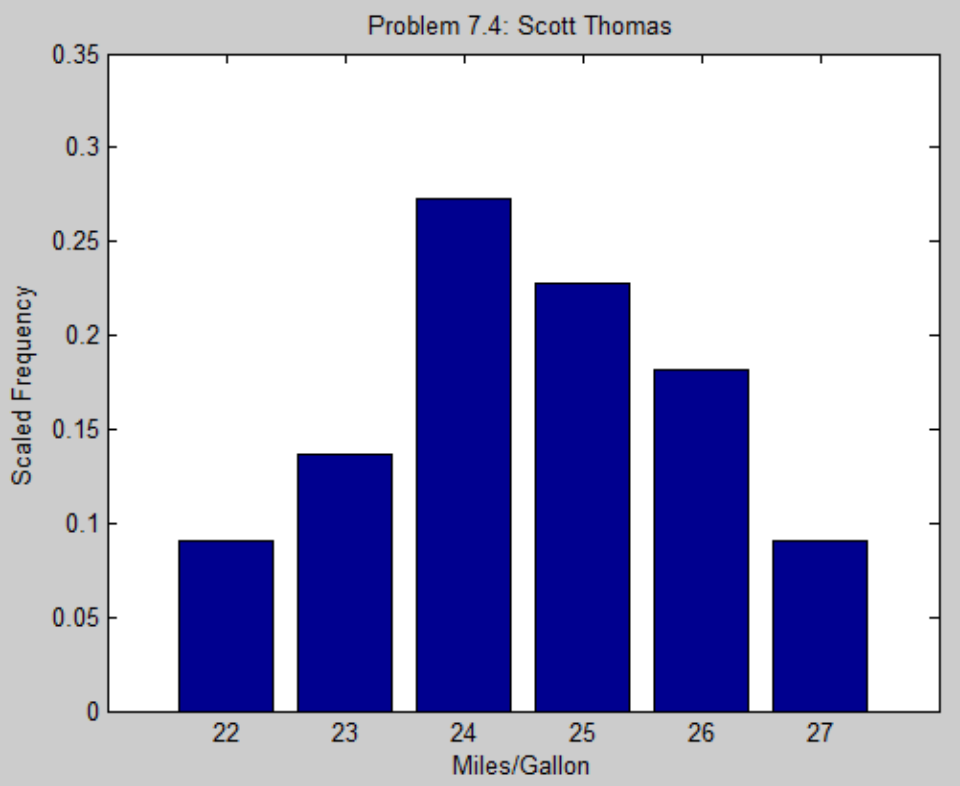
```
mpg = xlsread('prob7_1');
numvalues = length(mpg)
x = 22:27;
[n,xout] = hist(mpg,x)
bar(xout,n/numvalues)
xlabel('Miles/Gallon')
ylabel('Relative Frequency')
title('Problem 7.4: Scott Thomas')
```



Problem 7.4:

Step 4: The **Scaled Frequency** is calculated by dividing the **Absolute Frequency** by the total area of the histogram (Bin Width × ∑ Number of Values). Since the total area under the histogram rectangles is one, the area under the rectangles for a particular range is the probability of occurrence.

```
mpg = xlsread('prob7_1');
x = 22:27;
[n,xout] = hist(mpg,x)
binwidth = x(2) - x(1);
area = binwidth*sum(n);
mpg_scaled = n/area;
bar(x,mpg_scaled)
xlabel('Miles/Gallon')
ylabel('Scaled Frequency')
title('Problem 7.4: Scott Thomas')
```



Problem 7.4:

Step 5: For a **Normal Probability Distribution Curve**, it can be shown that approximately 68 percent of the area lies between the limits of:

$$\mu - \sigma \leq x \leq \mu + \sigma$$

where μ is the **Mean**, σ is the **Standard Deviation**. The Normal Probability Function is:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

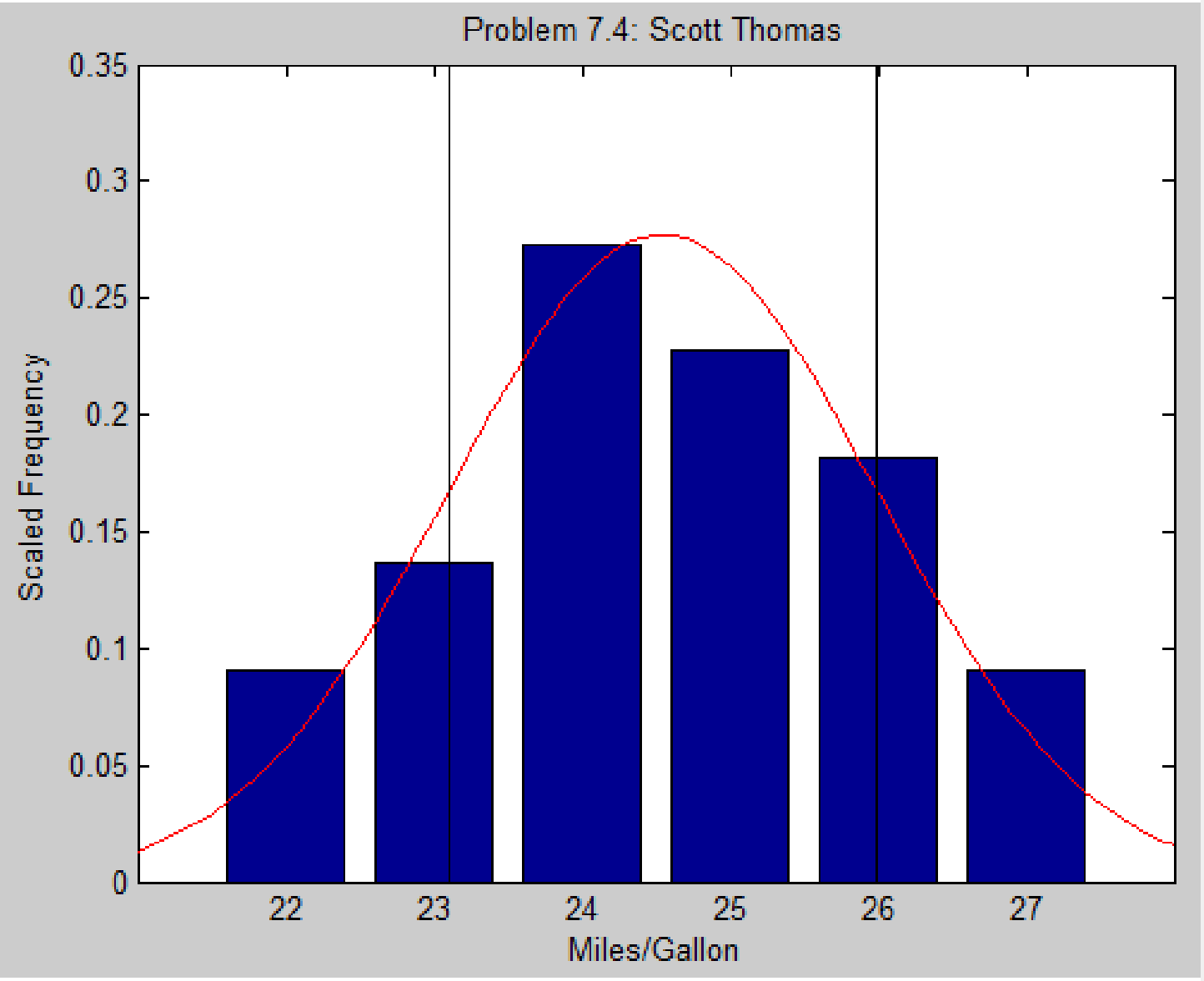
where σ^2 is the **Variance**. Plot this equation onto the bar plot for the **Scaled Frequency Histogram**.

Problem 7.4:

```
mu = mean(mpg)
sigma = std(mpg)
xplot= 21:0.1:28;
p = 1/(sigma*sqrt(2*pi))*exp(-(xplot-mu).^2/(2*sigma^2));
plot(xplot,p,'-r')
hold on

lower_limit = mu - sigma
upper_limit = mu + sigma
llplot = [lower_limit,lower_limit];
ulplot = [upper_limit,upper_limit];
yplot = [0 0.35];
plot(llplot,yplot,'-k', ulplot,yplot, '-k')
hold off
```

Problem 7.4:




```
mu =

    24.5455


sigma =

     1.4385


lower_limit =

    23.1070


upper_limit =

    25.9839
```

Problem 7.7:

7.* Data analysis of the breaking strength of a certain fabric shows that it is normally distributed with a mean of 300 lb and a variance of 9.

a. Estimate the percentage of fabric samples that will have a breaking strength no less than 294 lb.

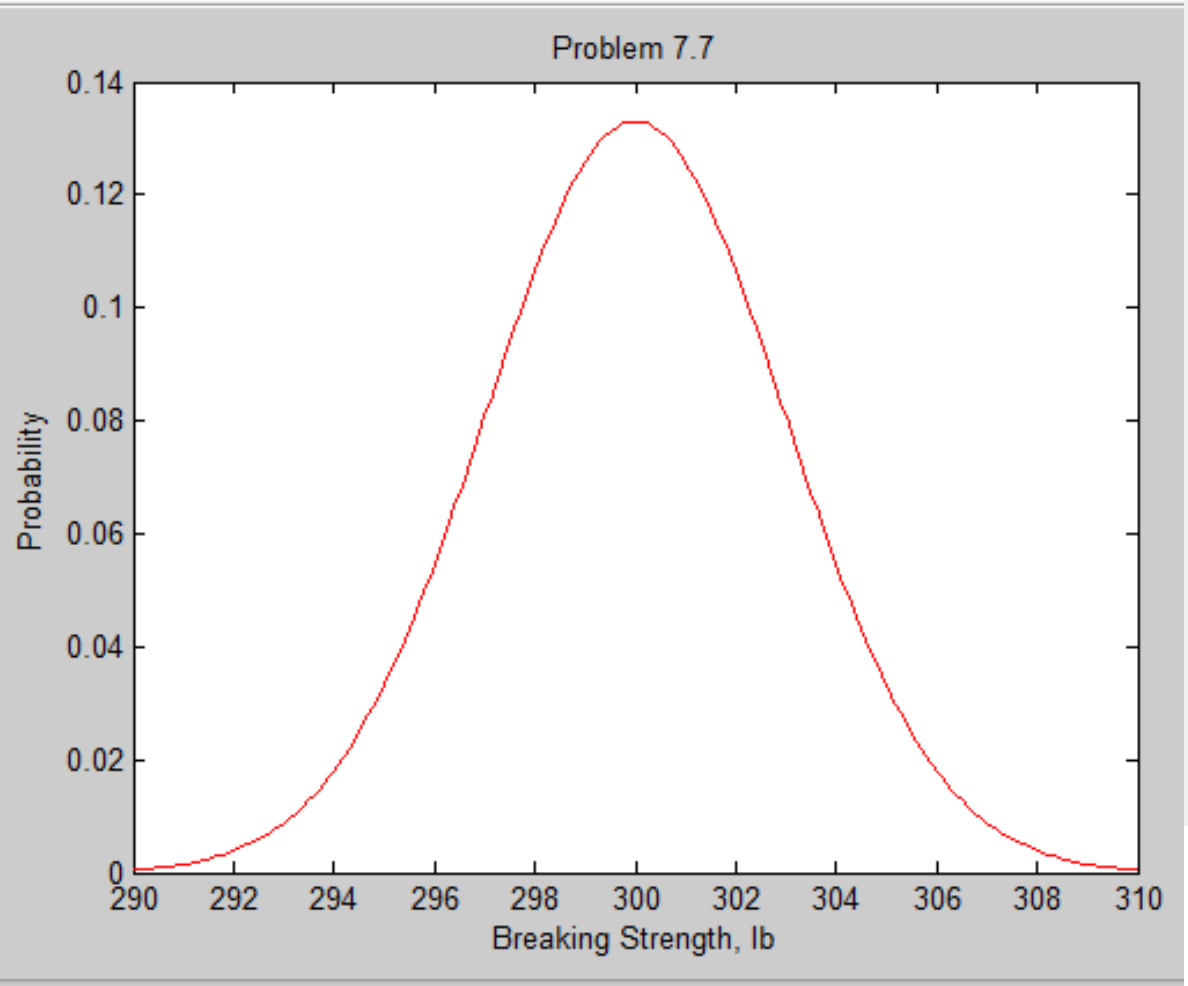
b. Estimate the percentage of fabric samples that will have a breaking strength no less than 297 lb and no greater than 303 lb.

$$P(x \leq b) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{b-\mu}{\sigma\sqrt{2}}\right)\right]$$

$$P(a \leq x \leq b) = \frac{1}{2}\left[\text{erf}\left(\frac{b-\mu}{\sigma\sqrt{2}}\right) - \text{erf}\left(\frac{a-\mu}{\sigma\sqrt{2}}\right)\right]$$

Problem 7.7:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$



Command Window

```
Problem 7.7: Scott Thomas
Part a:

P1 =

    0.0228


P2 =

    0.9772

Part b:

P3 =

    0.6827
```

Random Number Generation

MATLAB can generate random numbers that are uniformly distributed or normally distributed. These sets of random numbers can be used to analyze outcomes.

x = rand(n): generates uniformly distributed random numbers in the range [0,1]. To generate uniformly distributed random numbers over the interval [a,b]:

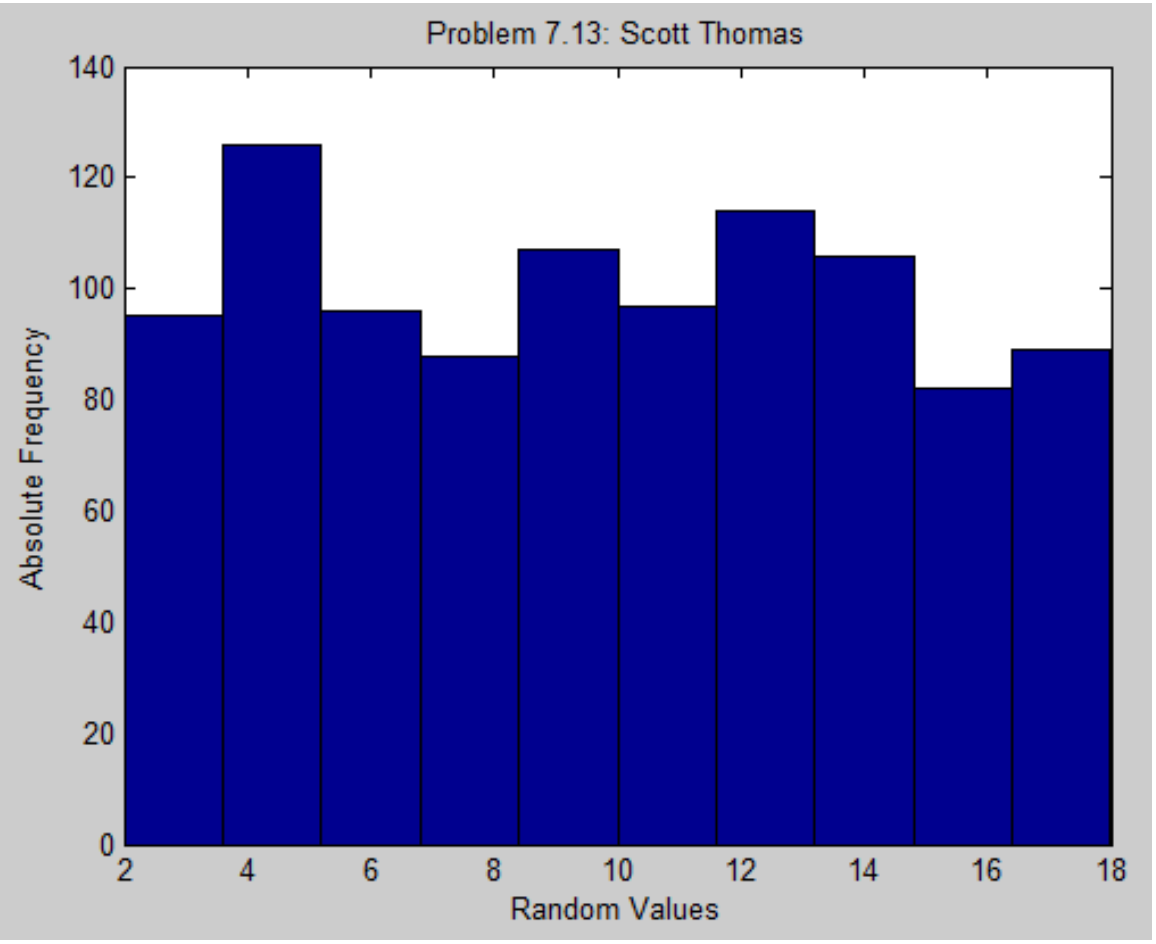
$$y = (b - a)x + a$$

x = randn(n): generates normally distributed random numbers that have a mean of $\mu = 0$ and a standard deviation of $\sigma = 1$. To generate normally distributed random numbers that have a mean of μ and a standard deviation of σ:

$$y = \sigma x + \mu$$

Problem 7.13:

13. Use a random number generator to produce 1000 uniformly distributed numbers with a mean of 10, a minimum of 2, and a maximum of 18. Obtain the mean and the histogram of these numbers, and discuss whether they appear uniformly distributed with the desired mean.



Command Window

```
Problem 7.13: Scott Thomas

Bmean =

   10.1603
```

Problem 7.15:

15. The mean of the sum (or difference) of two independent random variables equals the sum (or difference) of their means, but the variance is always the sum of the two variances. Use random number generation to verify this statement for the case where $z = x + y$, where x and y are independent and normally distributed random variables. The mean and variance of x are $\mu_x = 8$ and $\sigma_x^2 = 2$. The mean and variance of y are $\mu_y = 15$ and $\sigma_y^2 = 4$. Find the mean and variance of z by simulation, and compare the results with the theoretical prediction. Do this for 100, 1000, and 5000 trials.

$$y = \sigma x + \mu$$

Problem 7.15:

Command Window

```
Problem 7.15: Scott Thomas

muz =

    23


variancez =

     6

z100mean =

   23.2078

z1000mean =

   23.0165

z5000mean =

   23.0132

z100variance =

    4.5784

z1000variance =

    5.8030

z5000variance =

    5.6632
```

Interpolation

To estimate values between data points, several types of Interpolation can be used.

Linear Interpolation: The straight line connecting the two points are used for estimates of intermediate values.

Cubic Spline Interpolation: A third-order equation is derived between successive sets of three data points.

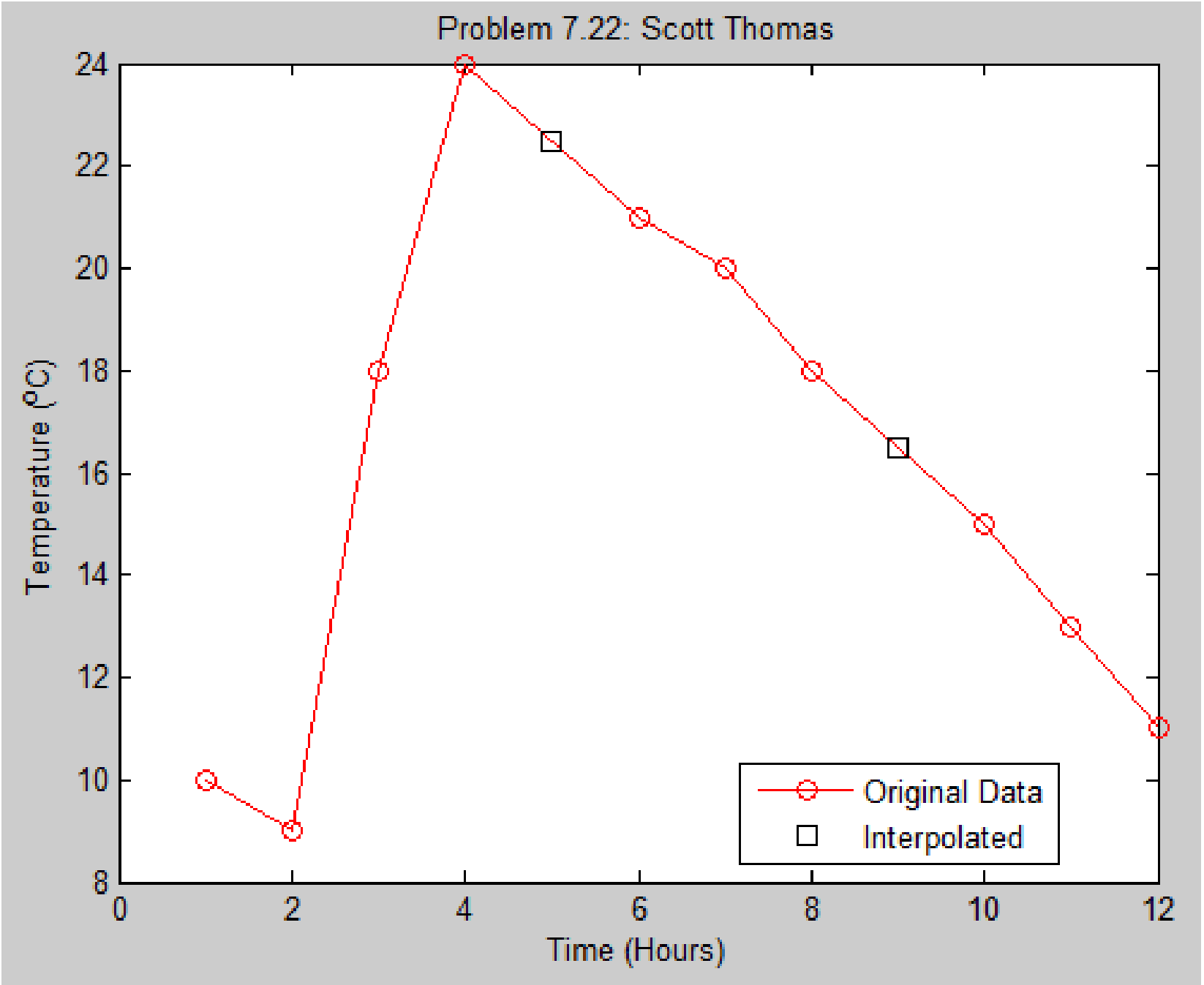
Piecewise Continuous Hermite Interpolation Polynomials (PCHIP): Less Overshoot than Cubic Spline.

Problem 7.22:

22.* Interpolation is useful when one or more data points are missing. This situation often occurs with environmental measurements, such as temperature, because of the difficulty of making measurements around the clock. The following table of temperature versus time data is missing readings at 5 and 9 hours. Use linear interpolation with MATLAB to estimate the temperature at those times.

Time (hours, P.M.)	1	2	3	4	5	6	7	8	9	10	11	12
Temperature (°C)	10	9	18	24	?	21	20	18	?	15	13	11

Problem 7.22:

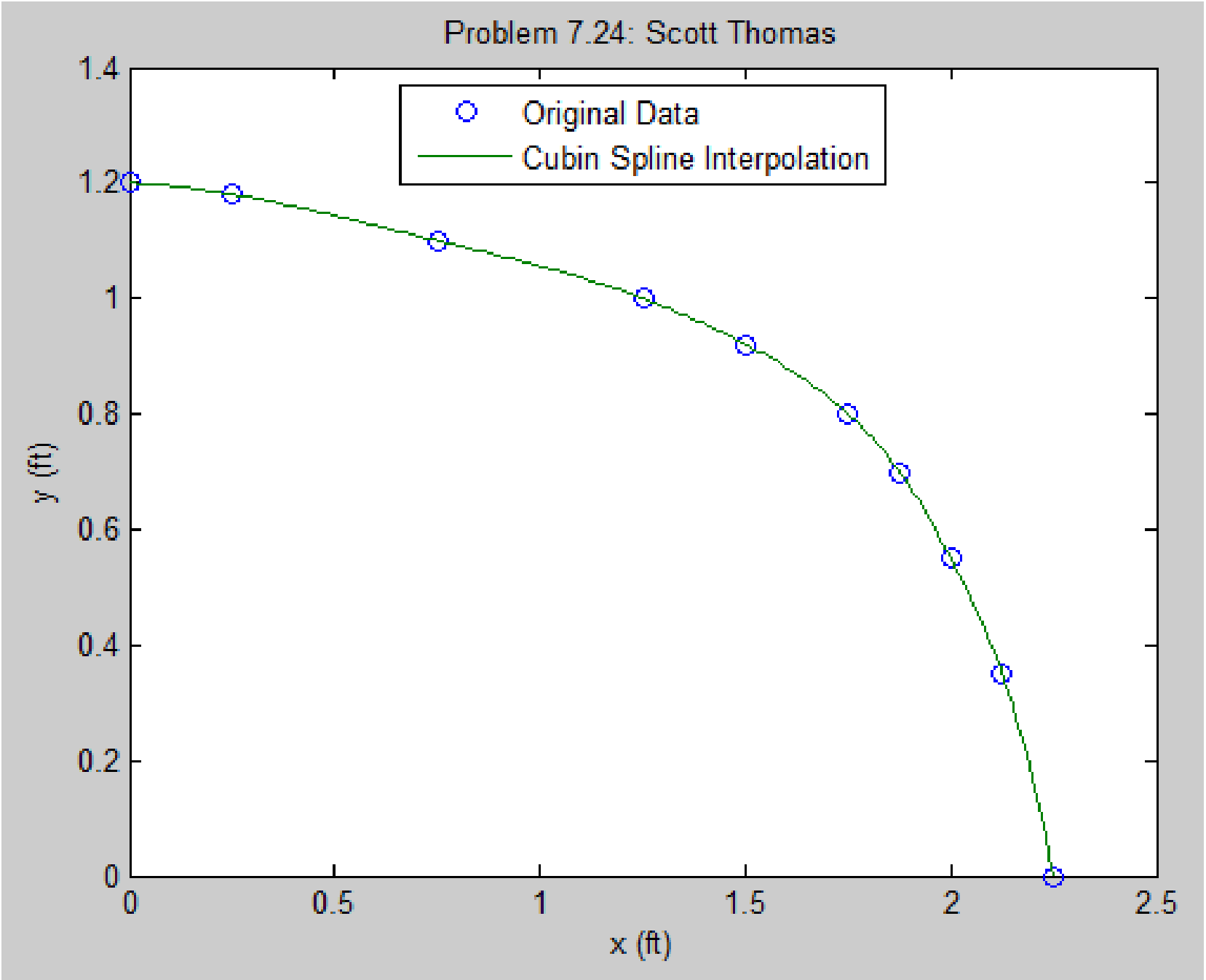

Problem 7.22: Scott Thomas
Temperature (°C)
Time (Hours)
24
22
20
18
16
14
12
10
8
0
2
4
6
8
10
12
Original Data
Interpolated

Problem 7.24:

24. Computer-controlled machines are used to cut and to form metal and other materials when manufacturing products. These machines often use cubic splines to specify the path to be cut or the contour of the part to be shaped. The following coodinates specify the shape of a certain car's front fender. Fit a series of cubic splines to the coordinates, and plot the splines along with the coordinate points.

x (ft)	0	0.25	0.75	1.25	1.5	1.75	1.875	2	2.125	2.25
y (ft)	1.2	1.18	1.1	1	0.92	0.8	0.7	0.55	0.35	0

Problem 7.24:


Problem 7.24: Scott Thomas
Original Data
Cubin Spline Interpolation
y (ft)
x (ft)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
0.5
1
1.5
2
2.5