Table 6.1–1 The `poly t` function

Command	Description
`p = poly t(x,y,n)`	Fits a polynomial of degree n to data described by the vectors x and y, where x is the independent variable. Returns a row vector `p` of length $n + 1$ that contains the polynomial coef cients in order of descending powers.

will denote as $w = p_1z + p_2$. Thus, referring to Table 6.1–1, we see that the vector `p` will be $[p_1, p_2]$ if `n` is 1. This polynomial has a different interpretation in each of the three cases:

- **The linear function:** $y = mx + b$. In this case the variables w and z in the polynomial $w = p_1z + p_2$ are the original data variables x and y, and we can nd the linear function that ts the data by typing `p = poly t(x,y,1)`. The rst element p_1 of the vector `p` will be m, and the second element p_2 will be b.
- **The power function:** $y = bx^m$. In this case $\log_{10} y = m \log_{10} x + \log_{10} b$, which has the form $w = p_1z + p_2$, where the polynomial variables w and z are related to the original data variables x and y by $w = \log_{10} y$ and $z = \log_{10} x$. Thus we can nd the power function that ts the data by typing `p = poly t(log10(x), log10(y),1)`. The rst element p_1 of the vector `p` will be m, and the second element p_2 will be $\log_{10} b$. We can nd b from $b = 10^{p_2}$.
- **The exponential function:** $y = b(10)^{mx}$. In this case $\log_{10} y = mx + \log_{10} b$, which has the form $w = p_1z + p_2$, where the polynomial variables w and z are related to the original data variables x and y by $w = \log_{10} y$ and $z = x$. Thus we can nd the exponential function that ts the data by typing `p = poly t(x, log10(y),1)`. The rst element p_1 of the vector `p` will be m, and the second element p_2 will be $\log_{10} b$. We can nd b from $b = 10^{p_2}$.

EXAMPLE 6.1–1 Temperature Dynamics

The temperature of coffee cooling in a porcelain mug at room temperature (68°F) was measured at various times. The data follow.

Time t (sec)	Temperature T (°F)
0	145
620	130
2266	103
3482	90

Develop a model of the coffee's temperature as a function of time, and use the model to estimate how long it took the temperature to reach 120°F.

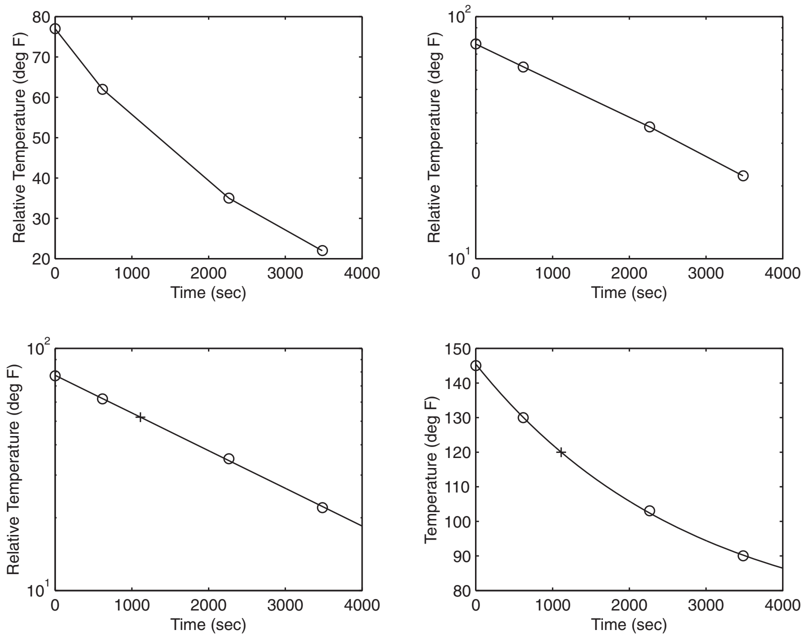

Figure 6.1–3 Temperature of a cooling cup of coffee, plotted on various coordinates.

■ Solution

Because $T(0)$ is nite but nonzero, the power function cannot describe these data, so we do not bother to plot the data on log-log axes. Common sense tells us that the coffee will cool and its temperature will eventually equal the room temperature. So we subtract the room temperature from the data and plot the relative temperature, $T - 68$, versus time. If the relative temperature is a linear function of time, the model is $T - 68 = mt + b$. If the relative temperature is an exponential function of time, the model is $T - 68 = b(10)^{mt}$. Figure 6.1–3 shows the plots used to solve the problem. The following MATLAB script le generates the top two plots. The time data are entered in the array `time`, and the temperature data are entered in `temp`.

```
% Enter the data.
time = [0,620,2266,3482];
temp = [145,130,103,90];
% Subtract the room temperature.
temp = temp - 68;
% Plot the data on rectilinear scales.
subplot(2,2,1)
plot(time,temp,time,temp,'o'),xlabel('Time (sec)'),...
   ylabel('Relative Temperature (deg F)')
%
% Plot the data on semilog scales.
```

```
p = polyfit(time,log10(temp),1);
m = p(1)
b = 10^p(2)
```

```
subplot(2,2,2)
semilogy(time,temp,time,temp,'o'),xlabel('Time (sec)'),...
   ylabel('Relative Temperature (deg F)')
```

The data form a straight line on the semilog plot only (the top right plot). Thus the data can be described with the exponential function $T = 68 + b(10)^{mt}$. Using the `poly t` command, the following lines can be added to the script le.

```
% Fit a straight line to the transformed data.
p = poly t(time,log10(temp),1);
m = p(1)
b = 10^p(2)
```

The computed values are $m = -1.5557 \times 10^{-4}$ and $b = 77.4469$. Thus our derived model is $T = 68 + b(10)^{mt}$. To estimate how long it will take for the coffee to cool to 120°F, we must solve the equation $120 = 68 + b(10)^{mt}$ for t. The solution is $t = [\log_{10}(120-68)-\log_{10}(b)]/m$. The MATLAB command for this calculation is shown in the following script le, which is a continuation of the previous script and produces the bottom two subplots shown in Figure 6.1–3.

```
% Compute the time to reach 120 degrees.
t_120 = (log10(120-68)-log10(b))/m
% Show the derived curve and estimated point on semilog scales.
t = 0:10:4000;
T = 68+b*10.^(m*t);
subplot(2,2,3)
semilogy(t,T-68,time,temp,'o',t_120,120-68,'+'),
xlabel('Time (sec)'),...
   ylabel('Relative Temperature (deg F)')
%
% Show the derived curve and estimated point on rectilinear scales.
subplot(2,2,4)
plot(t,T,time,temp+68,'o',t_120,120,'+'),xlabel('Time (sec)'),...
   ylabel('Temperature (deg F)')
```

The computed value of `t_120` is 1112. Thus the time to reach 120° F is 1112 sec. The plot of the model, along with the data and the estimated point (1112, 120) marked with a + sign, is shown in the bottom two subplots in Figure 6.1–3. Because the graph of our model lies near the data points, we can treat its prediction of 1112 sec with some con dence.

EXAMPLE 6.1–2 Hydraulic Resistance

A 15-cup coffee pot (see Figure 6.1–4) was placed under a water faucet and lled to the 15-cup line. With the outlet valve open, the faucet's ow rate was adjusted until the water level remained constant at 15 cups, and the time for one cup to ow out of the pot was

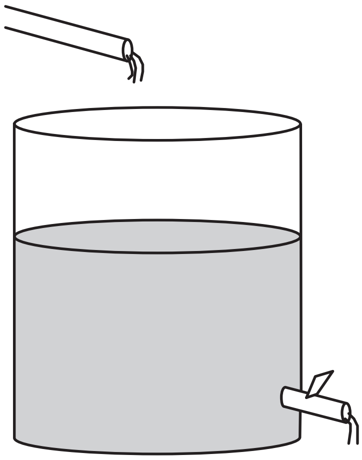

Figure 6.1–4 An experiment to verify Torricelli's principle.

measured. This experiment was repeated with the pot filled to the various levels shown in the following table:

Liquid volume V (cups)	Time to fill 1 cup t (sec)
15	6
12	7
9	8
6	9

(a) Use the preceding data to obtain a relation between the flow rate and the number of cups in the pot. (b) The manufacturer wants to make a 36-cup pot using the same outlet valve but is concerned that a cup will fill too quickly, causing spills. Extrapolate the relation developed in part (a) and predict how long it will take to fill one cup when the pot contains 36 cups.

■ Solution

(a) Torricelli's principle in hydraulics states that $f = rV^{1/2}$, where f is the flow rate through the outlet valve in cups per second, V is the volume of liquid in the pot in cups, and r is a constant whose value is to be found. We see that this relation is a power function where the exponent is 0.5. Thus if we plot $\log_{10}(f)$ versus $\log_{10}(V)$, we should obtain a straight line. The values for f are obtained from the reciprocals of the given data for t. That is, $f = 1/t$ cups per second.

The MATLAB script file follows. The resulting plots appear in Figure 6.1–5. The volume data are entered in the array `cups`, and the time data are entered in `meas_times`.

```
cups = [6,9,12,15];
meas_times = [9,8,7,6];
meas_flow = 1./meas_times;
%
% Fit a straight line to the transformed data.
p = polyfit(log10(cups),log10(meas_flow),1);
coeffs = [p(1),10^p(2)];
m = coeffs(1)
b = coeffs(2)
```

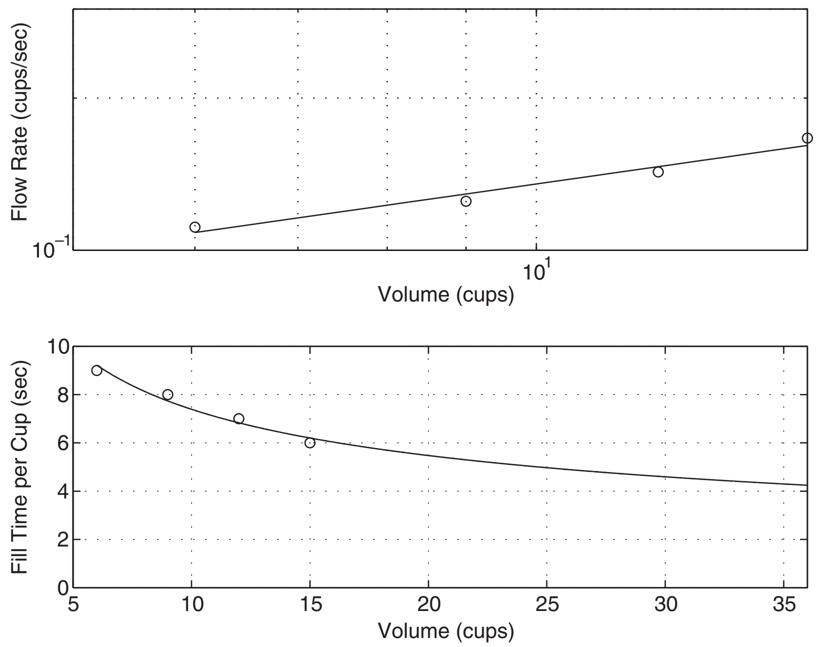

Figure 6.1–5 Flow rate and ll time for a cof fee pot.

```
% Data for the problem.
cups = [6,9,12,15];
meas_times = [9,8,7,6];
meas_ ow = 1./meas_times;
%
% Fit a straight line to the transformed data.
p = poly t(log10(cups),log10(meas_ ow),1);
coeffs = [p(1),10^p(2)];
m = coeffs(1)
b = coeffs(2)
%
% Plot the data and the  tted line on a loglog plot to see
% how well the line  ts the data.
x = 6:0.01:40;
y = b*x.^m;
subplot(2,1,1)
loglog(x,y,cups,meas_ ow,'o'),grid,xlabel('Volume (cups)'),...
   ylabel('Flow Rate (cups/sec)'),axis([5 15 0.1 0.3])
```

The computed values are $m = 0.433$ and $b = 0.0499$, and our derived relation is $f = 0.0499V^{0.433}$. Because the exponent is 0.433, not 0.5, our model does not agree exactly with Torricelli's principle, but it is close. Note that the rst plot in Figure 6.1–5 shows that the data points do not lie exactly on the tted straight line. In this application it is dif cult to measure the time to ll one cup with an accuracy greater than an integer second, so this inaccuracy could have caused our result to disagree with that predicted by Torricelli.

(b) Note that the ll time is $1/f$, the reciprocal of the ow rate. The remainder of the MATLAB script uses the derived ow rate relation $f = 0.0499V^{0.433}$ to plot the extrapolated ll-time curve $1/f$ versus t.

```
% Plot the  ll time curve extrapolated to 36 cups.
subplot(2,1,2)
plot(x,1./y,cups,meas_times,'o'),grid,xlabel('Volume(cups)'),...
   ylabel('Fill Time per Cup (sec)'),axis([5 36 0 10])
%
% Compute the  ll time for V = 36 cups.
 ll_time = 1/(b*36^m)
```

The predicted ll time for 1 cup is 4.2 sec. The manufacturer must now decide if this time is suf cient for the user to avoid over lling. (In fact, the manufacturer did construct a 36-cup pot, and the ll time is approximately 4 sec, which agrees with our prediction.)

6.2 Regression

In Section 6.1 we used the MATLAB function `poly t` to perform regression analysis with functions that are linear or could be converted to linear form by a logarithmic or other transformation. The `poly t` function is based on the least-squares method, which is also called *regression*. We now show how to use this function to develop polynomial and other types of functions.

The Least-Squares Method

Suppose we have the three data points given in the following table, and we need to determine the coef cients of the straight line $y = mx + b$ that best t the following data in the least-squares sense.

x	y
0	2
5	6
10	11

According to the least-squares criterion, the line that gives the best t is the one that minimizes J, the sum of the squares of the vertical differences between