Function Discovery – Fitting Parameters
for 2-parmaeter functions that are monotonic for positive x values

Table 1: Basic Two-Parameter Models

1	2	3	4	5	6
Model type	**Equation**	**Linerized form**	**Plot type (Matlab cmd.)**	**Fit**	**Coefficients**
linear	$y = mx + b$	$y = mx + b$	linear (plot)	x vs y	b = B(1), m = B(2)
power	$y = bx^m$	$\log(y) = \log(b) + m\log(x)$	log-log (loglog)	$\log_{10}$(x) vs $\log_{10}$(y)	b = 10^B(1) m = B(2),
exponential	$y = be^{mx}$	$\ln(y) = \ln(b) + mx$	semilog (semilogy)	x vs ln(y)	b = e^B(1) m = B(2),
	$y = b10^{mx}$	$\log(y) = \log(b) + mx$		x vs $\log_{10}$(y)	b = 10^B(1) m = B(2),

Fitting the parameters:

Model: Use graphs to decide if one of the two parameter models is appropriate. Write down the equation of the likely model (column 2) and its linear transform (column 3)

Setup: Use transformed variables from column 5 above to:
Create the design matrix, X (column of ones + a column of transformed x variable)
Create the response matrix, Y (column of transformed y-data)

Fit: Fit the linearized form in MATLAB.

- Determine parameter vector B = X\Y
- B will be a column vector where the first element is the intercept and the second is the slope in the linearized equations in column 3 above

Predict: Now calculate predicted values. For a linear model simply use the resulting fitted values in the linear equation (column 2.) or use the matrix equation:

```
>> Yhat = X*B
```

For a power or exponential model you must deal with the fact the parameters are transformed. For these two models you have two options:

1) Fit the linearized form (column 3) and then ***untransform*** the resulting predicted values; y = $10^{\log(y)}$. For example, for a power law fit or a base-10 exponential fit:

```
>> logYhat = X*B;
>> Yhat = 10.^logYhat
```

   Be careful to use a dot operator for the power in this second equation.

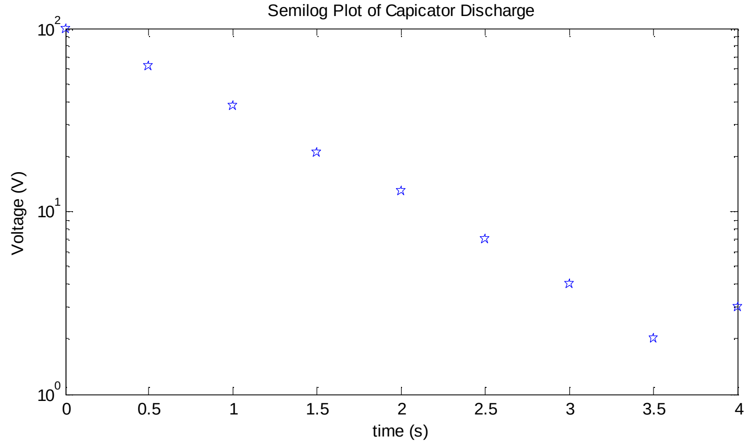
2) ***untransform*** the log(b) and then use it in the original equation to get the predicted values. e.g. >> b = $10^{log(B(1))}$; m = B(2);

```
>> Yhat = m * X(:,2) + b;
```

Plot: Check your fit by plotting both the data and the fitted values together on a single graph (experimental data should be points, the fitted equation a line).

E.G., Fitting the capacitor discharge data

Model: Last lecture we found the data was straight on a semilog plot implying an exponential model.



This implies either a base-ten or a base-e exponential model.
For the base-ten model the equations are:

$V = b10^{mt}$ or $\log(V) = \log(b) + mt$

Setup:
1. Design Matrix: >> X = [ones(length(t),1), t]
2. Response Vector >> Y= log10(V)

Fit: >> B = X \ Y

Predict:

```
>> logVhat = X*B            or      >> b = 10^B(1),   m = B(2)
>> Vhat = 10.^logVhat               >> Vhat = b*10.^(m.*t)
```

Plot: either on <u>linear</u> or semilog plot a
For the linear plot the commands would be:

```
>> plot(t, V, 's', t, Vhat)
>> xlabel 'time (s.)', ylabel 'Voltage'
>> legend('Experimental data', 'Fit: V = (95.8)10^{-0.43*t}')
```

Notice we are using the original untransformed variables.
For a semilog plot simply replace the ***plot*** function with ***semilogy***

Note: you can show super scripts in the legend command by using the carrot (^), the character following the carrot symbol will be superscripted.

e.g.

```
>> legend('Data', 'Fit: y =  23*x^-^3^.^2')     or
>> legend('Data', 'Fit: y =  23*x^{-3.2}')
```

will display the following legend labels