Experimental data analysis
Lecture 3: Confidence intervals

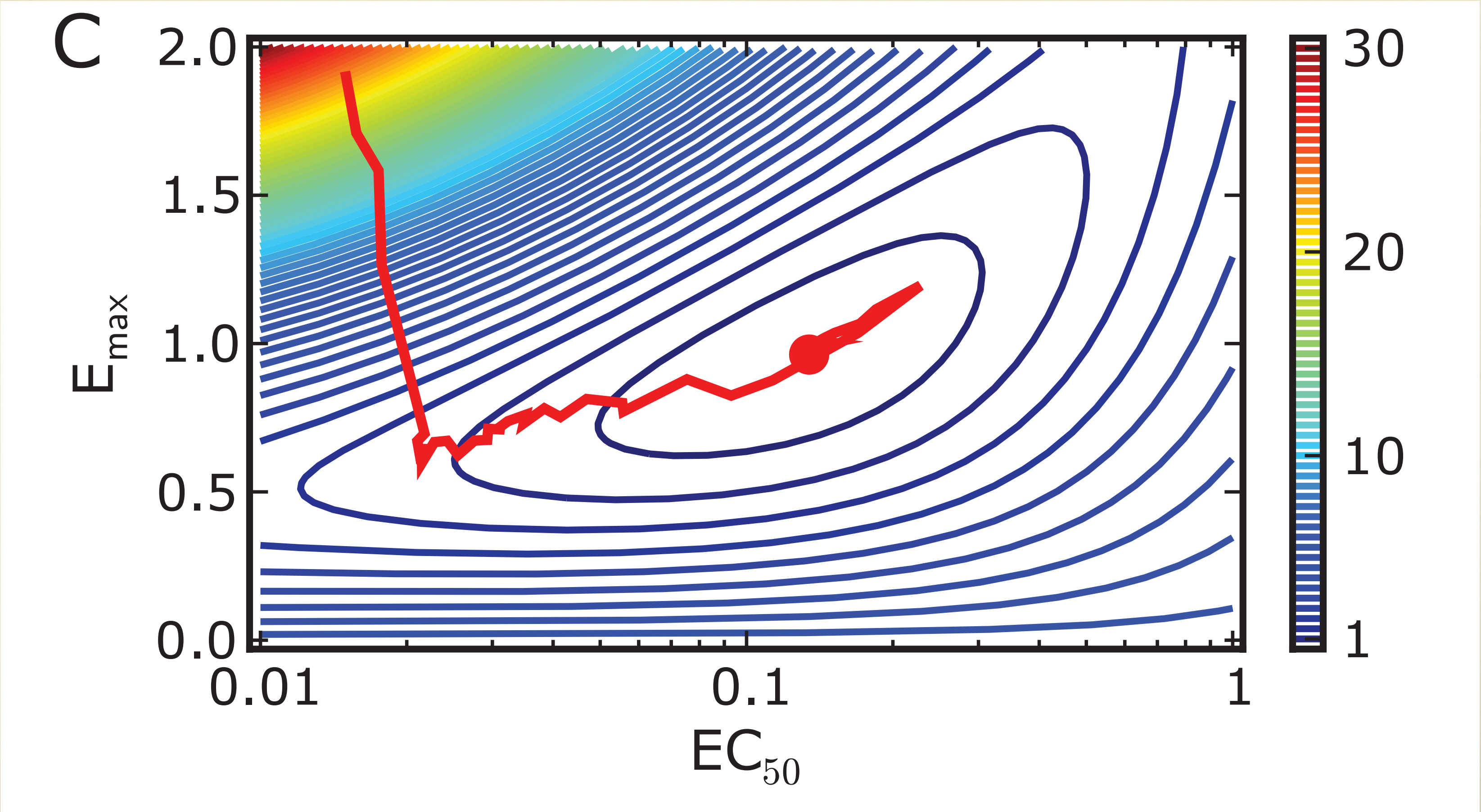
Dodo Das

Review of lecture 2

- Nonlinear regression - Iterative likelihood maximization

- Levenberg-Marquardt algorithm (Hybrid of steepest descent and Gauss-Newton)

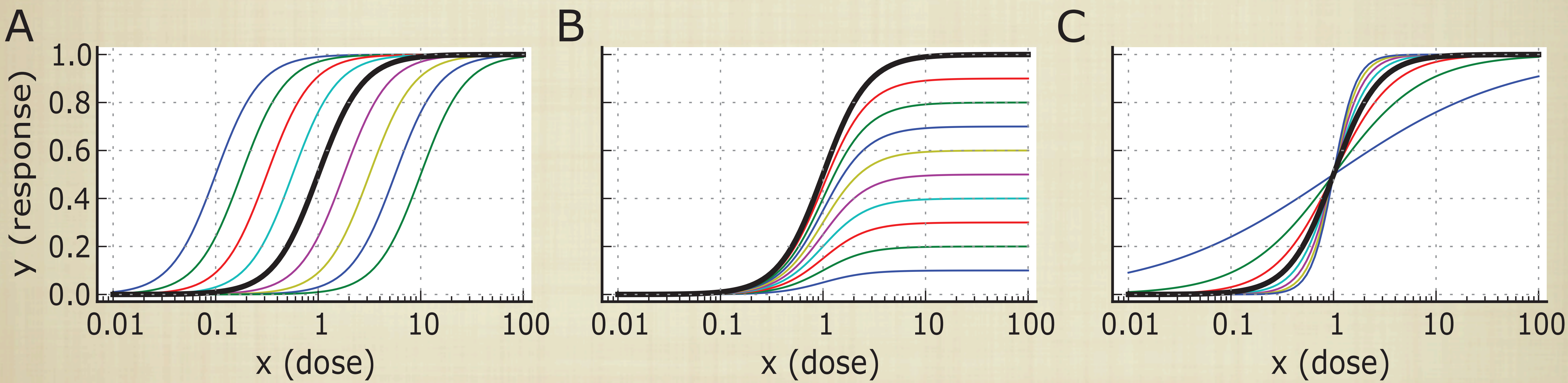
- Stochastic optimization - MCMC, Simulated annealing.



Demo 3: Nonlinear regression in MATLAB

- Objective: Using a Hill function to model dose-response data.

$$y = f(x;\, y_{\max}, EC_{50}, n) = \frac{y_{\max}}{1 + \left(EC_{50}/x\right)^n}$$



Demo 3: Nonlinear regression in MATLAB

Need the statistics toolbox which contains the function nlinfit

i) Write a MATLAB function to compute the Hill function for a given set of parameter values.

File Edit Text Go Cell Tools Debug Desktop Window Help

```
function response = Hill( par, dose )
%% Return the predicted response for a Hill function
ymax = par(1);
EC50 = 10^par(2);
n = par(3);

response = ymax./(1 + (EC50./dose).^n);
end
```

Demo 3: Nonlinear regression in MATLAB

ii) Choose some model parameters to simulate data [or collect data from experiments]

```
%% Nonlinear regression. Generate data for a dose response curve by adding
%% some normally distributed noise to a Hill function. Use nlinfit to
%% perform nonlinear regression

%% Specify model parameters. The Hill function is given by
%% y(x) = ymax/(1 + (EC50/dose)^n)
ymax = 1;
Lec50 = 0;
n = 2;
modelpar = [ymax Lec50 n];

%% Generate some data
dose = (logspace(-1, 1, 15))';
yTrue = Hill(modelpar, dose);

%% Add normal error to simulate experimental noise
NoiseStd = 0.1; % The standard deviation of the noise
err = NoiseStd*randn(length(yTrue), 1);
yExpt = yTrue + err;
```

Demo 3: Nonlinear regression in MATLAB

iii) Pick an initial guess and call nlinfit

```matlab
%% Call nlinfit to perform nonlinear regression. Add an option to see the
%% SSR value after each iteration
options = statset('Display', 'iter');
betaGuess = [1, 1, 1] % Initial guess for parameter values
betaHat = nlinfit(dose, yExpt, @Hill, betaGuess, options)

%% Not all initial guesses work. This one fails to find the minimum.
betaGuess = [0.1, 1, 4]
nlinfit(dose, yExpt, @Hill, betaGuess, options)

%% Plot data and fit
semilogx(dose, yExpt, 'ro', 'MarkerSize', 8)
hold on
doseFit = (logspace(-1, 1, 201))';
yFit = Hill(betaHat, doseFit);
plot(doseFit, yFit, 'b-', 'LineWidth', 2)
plot(doseFit, Hill(modelpar, doseFit), 'b--', 'LineWidth', 1)
xlim([0.09, 11])
ylim([-0.05, 1.15])
xlabel('dose')
ylabel('response')
grid on
legend('Data', 'Nonlinear fit', 'True model', 'Location', 'NW')
```

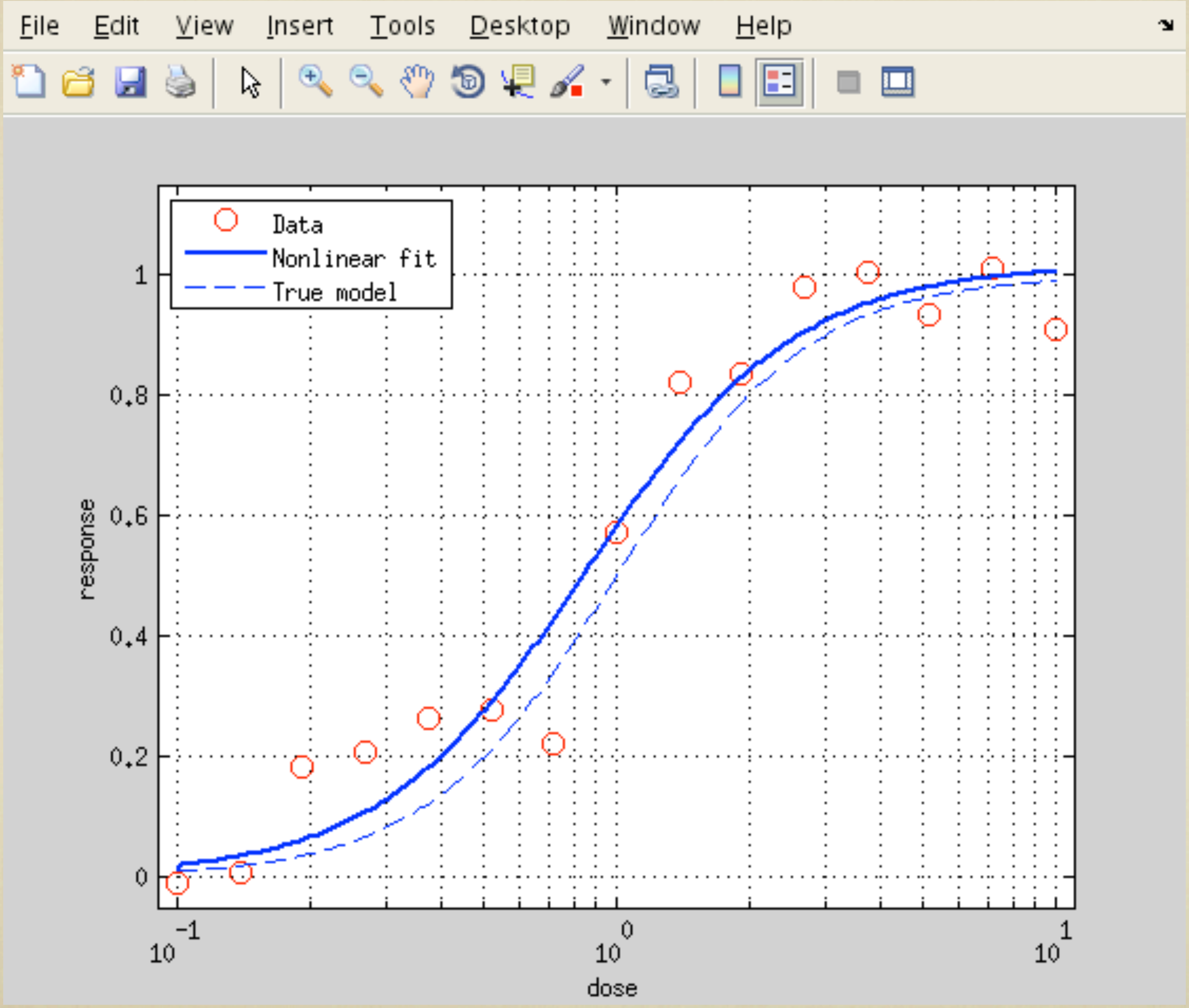
Demo 3: Nonlinear regression in MATLAB

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
betaGuess =

     1     1     1


                                            Norm of           Norm of
  Iteration              SSE               Gradient            Step
 -----------------------------------------------------------
          0          3.36869
          1          2.41072             1.09421            1.13207
          2          1.29038              2.7598           0.717239
          3         0.737374             4.97887            1.03529
          4         0.315402             1.19888            1.09758
          5         0.151169            0.253951           0.412226
          6         0.115426           0.0941071           0.336512
          7         0.108014           0.0424598           0.247683
          8         0.107063          0.00288839          0.0328246
          9         0.106997         0.000564608          0.0306803
         10         0.106991         1.90122e-05         0.00157004
         11          0.10699         5.20445e-06         0.00286241
         12          0.10699         1.79044e-07        0.000198578
         13          0.10699         4.70273e-08        0.000264638
         14          0.10699         2.26291e-09        3.04801e-05
Iterations terminated: relative change in SSE less than OPTIONS.TolFun

betaHat =

    1.0164   -0.0697    1.8357

>> |
```

Demo 3: Nonlinear regression in MATLAB



Demo 3: Nonlinear regression in MATLAB

Convergence can be sensitive to the initial guess.

```
%% Not all initial guesses work. This one fails to find the minimum.
betaGuess = [0.1, 1, 4]
nlinfit(dose, yExpt, @Hill, betaGuess, options)
```

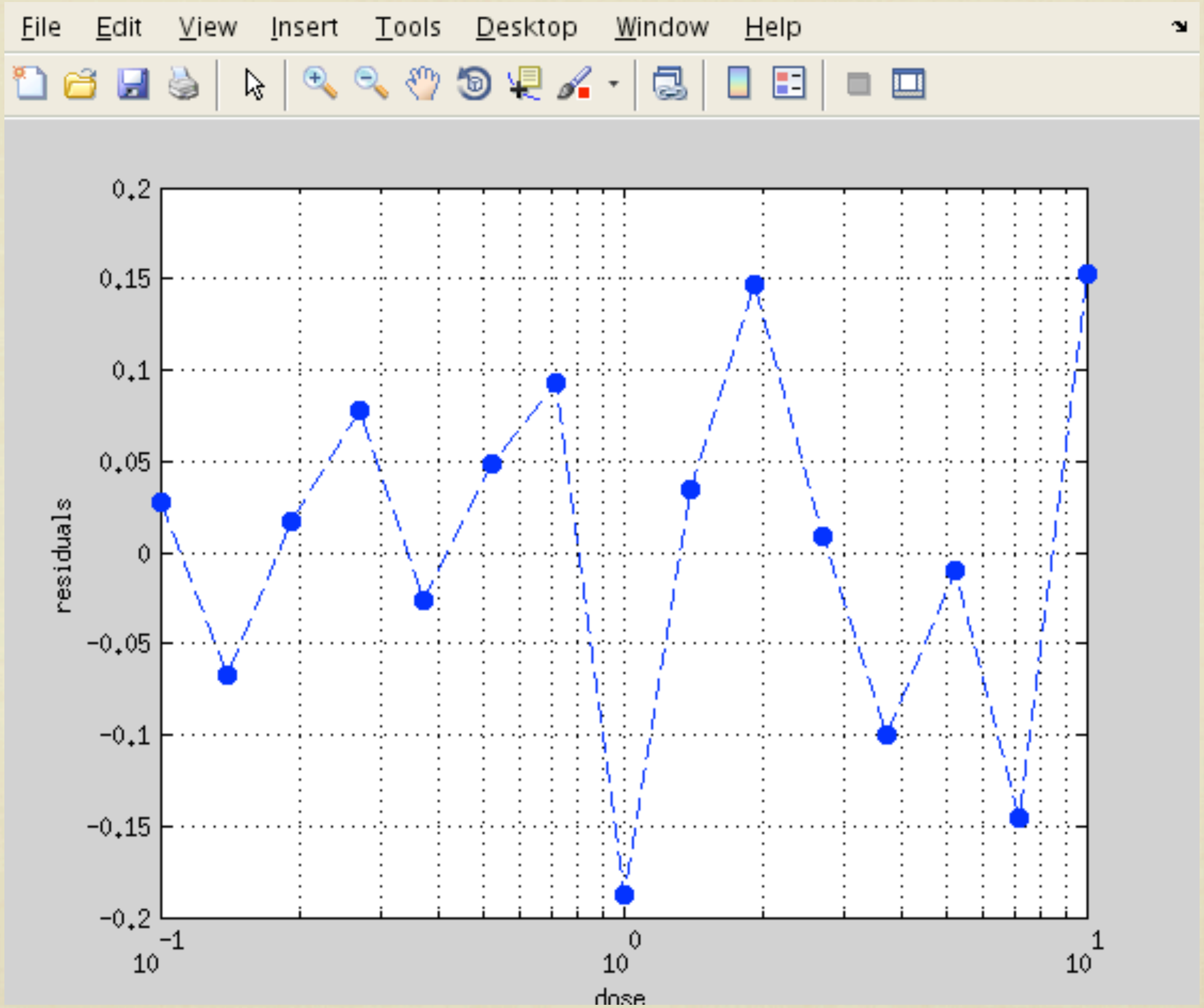
Demo 3: Nonlinear regression in MATLAB

Convergence can be sensitive to the initial guess.

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

beta Guess =

    0.1000    1.0000    4.0000


                                         Norm of          Norm of
   Iteration             SSE            Gradient           Step
  -----------------------------------------------------------
           0          6.52615
           1          4.77415            1.74131          20.0928
           2           4.5339            4.89515          26.4957
           3          3.82728            2.56718          251.139
Warning: Rank deficient, rank = 1,  tol =    1.5114e-14.
> In nlinfit>LMfit at 294
  In nlinfit at 166
           4          3.62008           0.299641         0.115246
Warning: Rank deficient, rank = 1,  tol =    1.4483e-14.
> In nlinfit>LMfit at 294
  In nlinfit at 166
           5          3.61836         0.00296674        0.0114105
Warning: Rank deficient, rank = 1,  tol =    1.4418e-14.
> In nlinfit>LMfit at 294
  In nlinfit at 166
           6          3.61836        2.96377e-06      0.000113991
Warning: Rank deficient, rank = 1,  tol =    1.4411e-14.
> In nlinfit>LMfit at 294
  In nlinfit at 166
           7          3.61836        2.96349e-10       1.1398e-07
Iterations terminated: relative norm of the current step is less than OPTIONS.TolX
Warning: The Jacobian at the solution is ill-conditioned, and some
model parameters may not be estimated well (they are not identifiable).
Use caution in making predictions.
> In nlinfit at 223

ans =

    0.4849    0.7968 -293.7222

fx >>
```

Demo 3: Nonlinear regression in MATLAB

Compute and plot residuals



Parameter confidence intervals

- Question: What does a 95% confidence interval mean?

- eg: Say, a best fit parameter estimate is $\hat{a} = 1$, and we have estimated the 95% CI to be [0.5, 1.5]. How can we interpret this result?

Parameter confidence intervals

- Question: What does a 95% confidence interval mean?

- eg: Say, a best fit parameter estimate is $\hat{a} = 1$, and we have estimated the 95% CI to be [0.5, 1.5]. How can we interpret this result?

- If we repeat our experiment and the fitting procedure many times, 95% of the times the true (but unknown) parameter value will lie within this CI.

Computing parameter confidence intervals

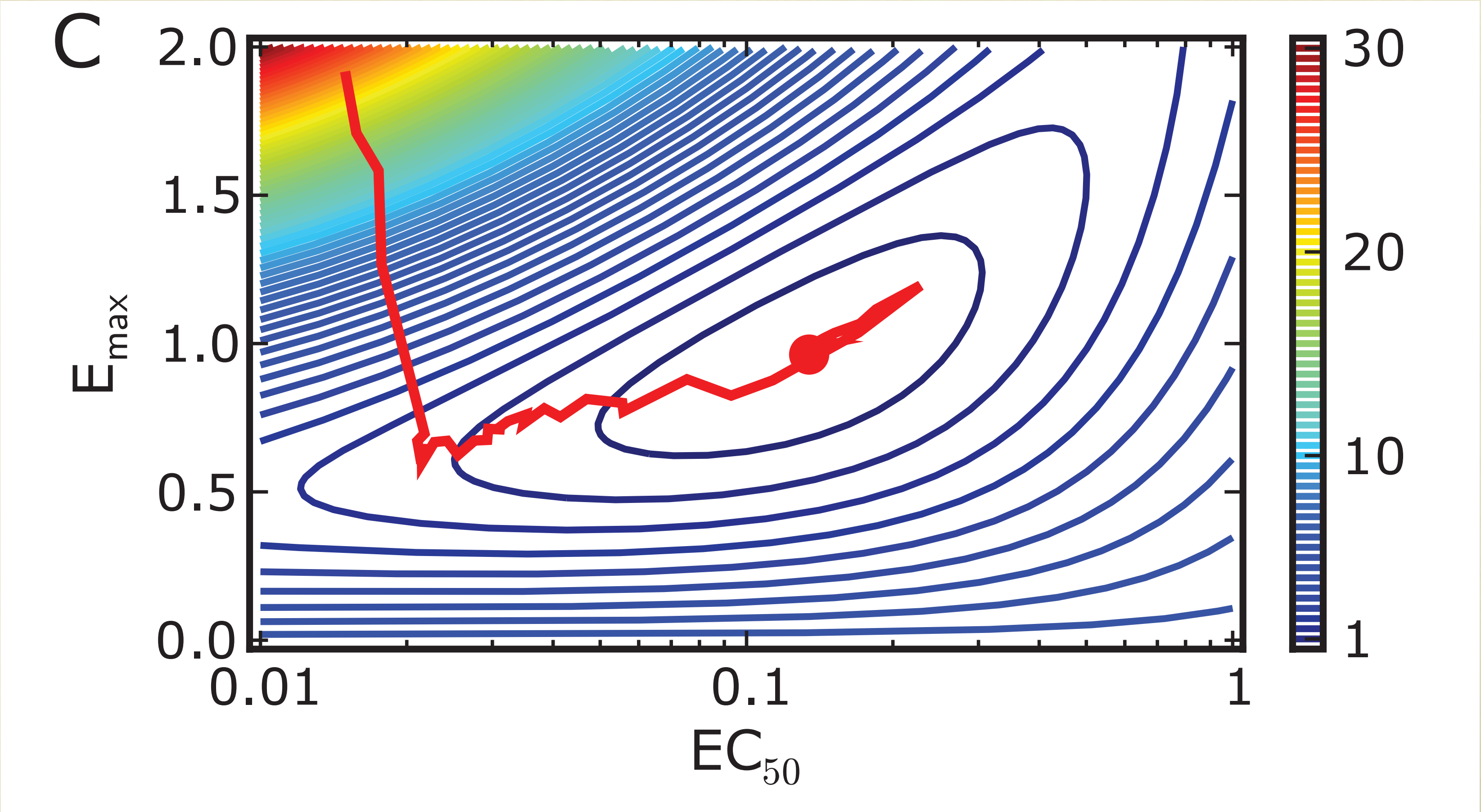
- Two approaches:

1. **Asymptotic confidence intervals**: Based on an analytical approximation.

2. **Bootstrap confidence intervals**: Computational technique based on resampling the errors.

Asymptotic confidence intervals

- Use a local approximation of the SSR landscape to estimate the curvature (covariance matrix).
- Assuming that the errors are normally distributed, this covariance matrix can be used to compute a confidence interval around each parameter.



Demo 4: Computing asymptotic CIs in MATLAB

Use the nlparci command in the Statistics toolbox

```
%% Estimate asymptotic confidence intervals for a nonlinear regression
%% fit using nlparci (Statistics toolbox)

%% Fit simulated dose-response data to a Hill function, given by
%% y(x) = ymax/(1 + (EC50/dose)^n)

%% Load data
DoseResponse = load('DoseResponseData.dat', '-ascii');
dose = DoseResponse(:,1);
yExpt = DoseResponse(:,2);

%% Call nlinfit to perform nonlinear regression.
betaGuess = [1, 1, 1]; % Initial guess for parameter values
[betaHat, res, Jac, Cov, mse] = nlinfit(dose, yExpt, @Hill, betaGuess);

%% Supply output to nlparci to estimate asymptotic 95% CIs.
asympCI = nlparci(betaHat, res, 'covar', Cov);

%% Print parameter estimates and asymptotic CIs
[betaHat' asympCI]
```

Demo 4: Computing asymptotic CIs in MATLAB

Use the nlparci command in the Statistics toolbox

```
%% Estimate asymptotic confidence intervals for a nonlinear regression
%% fit using nlparci (Statistics toolbox)

%% Fit simulated dose-response data to a Hill function, given by
%% y(x) = ymax/(1 + (EC50/dose)^n)

%% Load data
DoseResponse = load('DoseResponseData.dat', '-ascii');
dose = DoseResponse(:,1);
yExpt = DoseResponse(:,2);

%% Call nlinfit to perform nonlinear regression.
betaGuess = [1, 1, 1]; % Initial guess for parameter values
[betaHat, res, Jac, Cov, mse] = nlinfit(dose, yExpt, @Hill, betaGuess);

%% Supply output to nlparci to estimate asymptotic 95% CIs.
asympCI = nlparci(betaHat, res, 'covar', Cov);

%% Print parameter estimates and asymptotic CIs
[betaHat' asympCI]
```

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
ans =

    1.0268    0.9040    1.1497
    0.0058   -0.0930    0.1045
    2.8238    1.2890    4.3585

>>
```

Bootstrapping: The principle

- A computational approach that addresses the following question:
- Given a limited number of observations, how can we estimate some quantity, eg: mean, median etc. for the population from which the observations are drawn?
- If we 'resample' from the observations, we can, in some sense, simulate the population distribution.

Bootstrapping in practice for nonlinear regression.

1. Use the nonlinear least squares regression to determine the best-fit estimates of the model parameters, and the predicted model response $(y_i)_{\text{predicted}}$ at each value of the independent variable x_i.

2. Calculate the residuals $\epsilon_i = (y_i)_{\text{observed}} - (y_i)_{\text{predicted}}$ at each of the N data points.

3. Resample the residuals **with replacement** to generate a new set of residuals $\{\epsilon_i^*\}$. What this means is that we generate a new set of N residuals where each of N values is one of the original residuals chosen with equal probability. Typically, some of the original residuals will be chosen more than once, while some will not be chosen at all. For example, say we have three data points and we calculate the residuals to be 0.1, -0.2 and 0.3. Then some possible sets of resampled residuals are: {-0.2, 0.1, -0.2}, {0.1, 0.3, -0.2}, {0.3, -0.2, -0.2}, and so on.

4. Add the resampled residuals to the predicted response to generate a **bootstrap data set**, $\{x_i, y_i^*\} = \{x_i, (y_i)_{\text{predicted}} + \epsilon_i^*\}$

5. Treat the bootstrap dataset as an independent replicate experiment, and fit it to the model to calculate new estimates of model parameters.

6. Repeat steps 3-5 many times - typically 500 to 1000 times - each time generating a new bootstrap data set, and fitting it to the model. Store the resulting best-fit parameter estimates. These independent estimates constitute a sample from the bootstrap distribution of the model parameters.

7. For each parameter, calculate the standard deviation of the bootstrap sample. This standard deviation is the estimated bootstrap standard error for that parameter.

8. To calculate the 95% bootstrap CIs, compute the 97.5^{th} and the 2.5^{th} percentile values of each parameter from the bootstrap distributions. (There are other prescriptions for calculating bootstrap CIs, but this one is the simplest.)

Demo 5: Estimating bootstrap CIs in MATLAB

```matlab
%% Construct bootstrap samples for a nonlinear regression and estimate
%% bootstrap confidence intervals.

%% Fit simulated dose-response data to a Hill function, given by
%% y(x) = ymax/(1 + (EC50/dose)^n)

%% Load data
DoseResponse = load('DoseResponseData.dat', '-ascii');
dose = DoseResponse(:,1);
yExpt = DoseResponse(:,2);

%% Call nlinfit to perform nonlinear regression.
betaGuess = [1, 1, 1]; % Initial guess for parameter values
[betaHat, residuals, Jac, Cov, mse] = nlinfit(dose, yExpt, @Hill, betaGuess

%% Resample the residuals using the bootstrp function
nboot = 200; % Number of bootstrap replicates
[~, bootIndices] = bootstrp(nboot, [], residuals);
bootIndices(:,1:3) % Print the first 3 columns of bootIndices
bootResiduals = residuals(bootIndices);
bootResiduals(:, 1:3) % Print the first 3 columns of bootResiduals

%% Generate bootstrap datasets by adding the resampled residuals
%%to the best fit curve
yMod = Hill(betaHat, dose);
yBoot = repmat(yMod, 1, nboot) + bootResiduals;

%% Fit each of the bootstrap datasets to the Hill function to
%% build up the bootstrap distributions of parameter estimates
betaBoot = zeros(nboot, 3);
for i=1:nboot
    betaBoot(i,:) = nlinfit(dose, yBoot(:,i), @Hill, betaGuess);
end

% Estimate 95% CIs
bootCI = prctile(betaBoot,[2.5 97.5]);
```

Demo 5: Estimating bootstrap CIs in MATLAB

Command Window

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

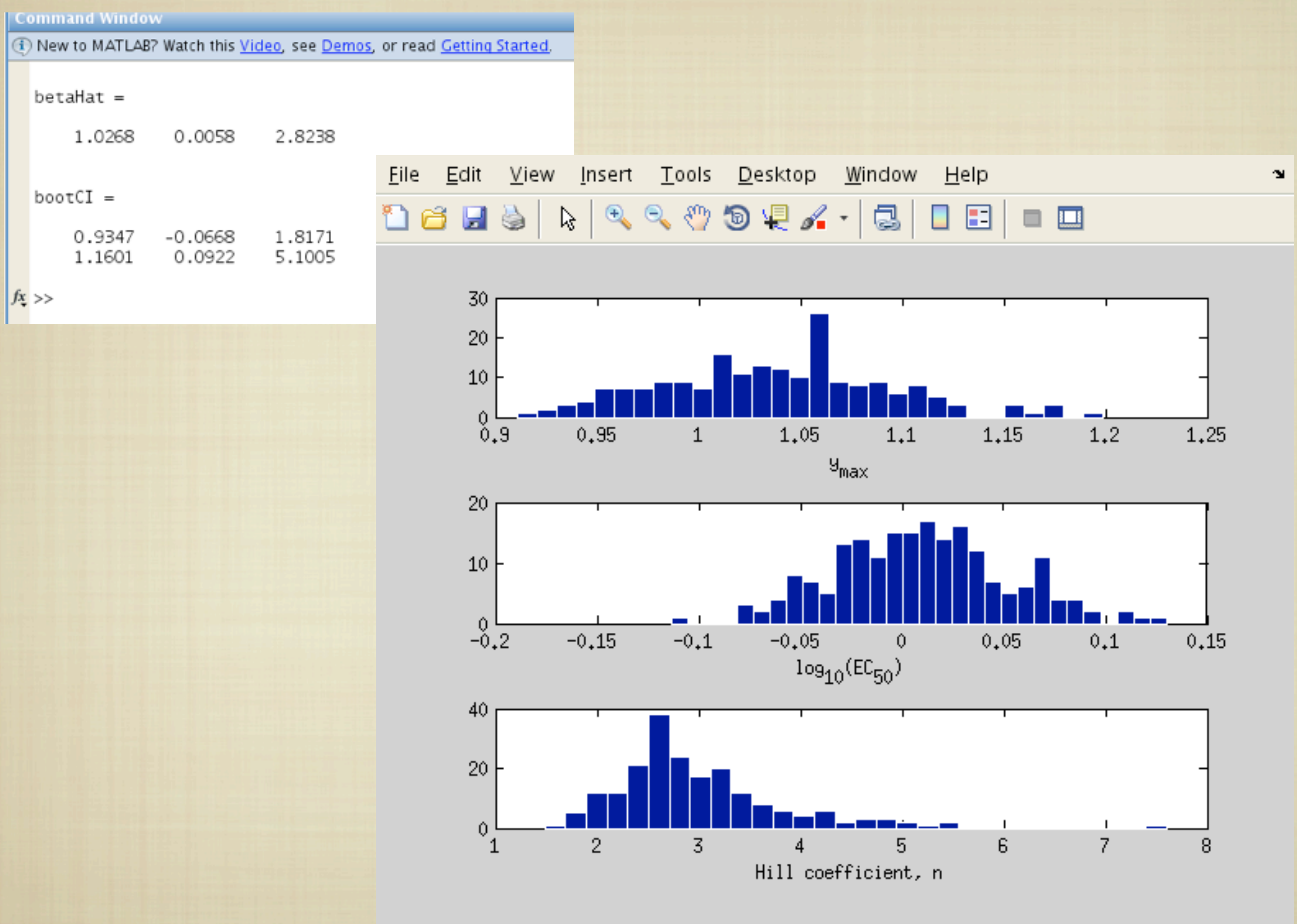
```
%% Resample the residuals using the bootstrp function
nboot = 200; % Number of bootstrap replicates
[~, bootIndices] = bootstrp(nboot, [], residuals);
bootIndices(:,1:3) % Print the first 3 columns of bootIndices
bootResiduals = residuals(bootIndices);
bootResiduals(:, 1:3) % Print the first 3 columns of bootResiduals

ans =

    13    13    11
    12     9    10
     6     4    15
     6    14     6
    15    13     2
    13     9    14
     2     1    12
    10     8    13
    12    12    13
    12    14    15
     8     2     5
    12     8     4
     9     1     3
    12     3     1
    10     6     4


ans =

   -0.0097   -0.0097    0.0089
   -0.1002    0.0342    0.1473
    0.0486    0.0772    0.1527
    0.0486   -0.1453    0.0486
    0.1527   -0.0097   -0.0670
   -0.0097    0.0342   -0.1453
   -0.0670    0.0276   -0.1002
    0.1473   -0.1874   -0.0097
   -0.1002   -0.1002   -0.0097
   -0.1002   -0.1453    0.1527
   -0.1874   -0.0670   -0.0260
   -0.1002   -0.1874    0.0772
    0.0342    0.0276    0.0171
   -0.1002    0.0171    0.0276
    0.1473    0.0486    0.0772

fx >>
```

Demo 5: Estimating bootstrap CIs in MATLAB

File Edit Text Go Cell Tools Debug Desktop Window Help

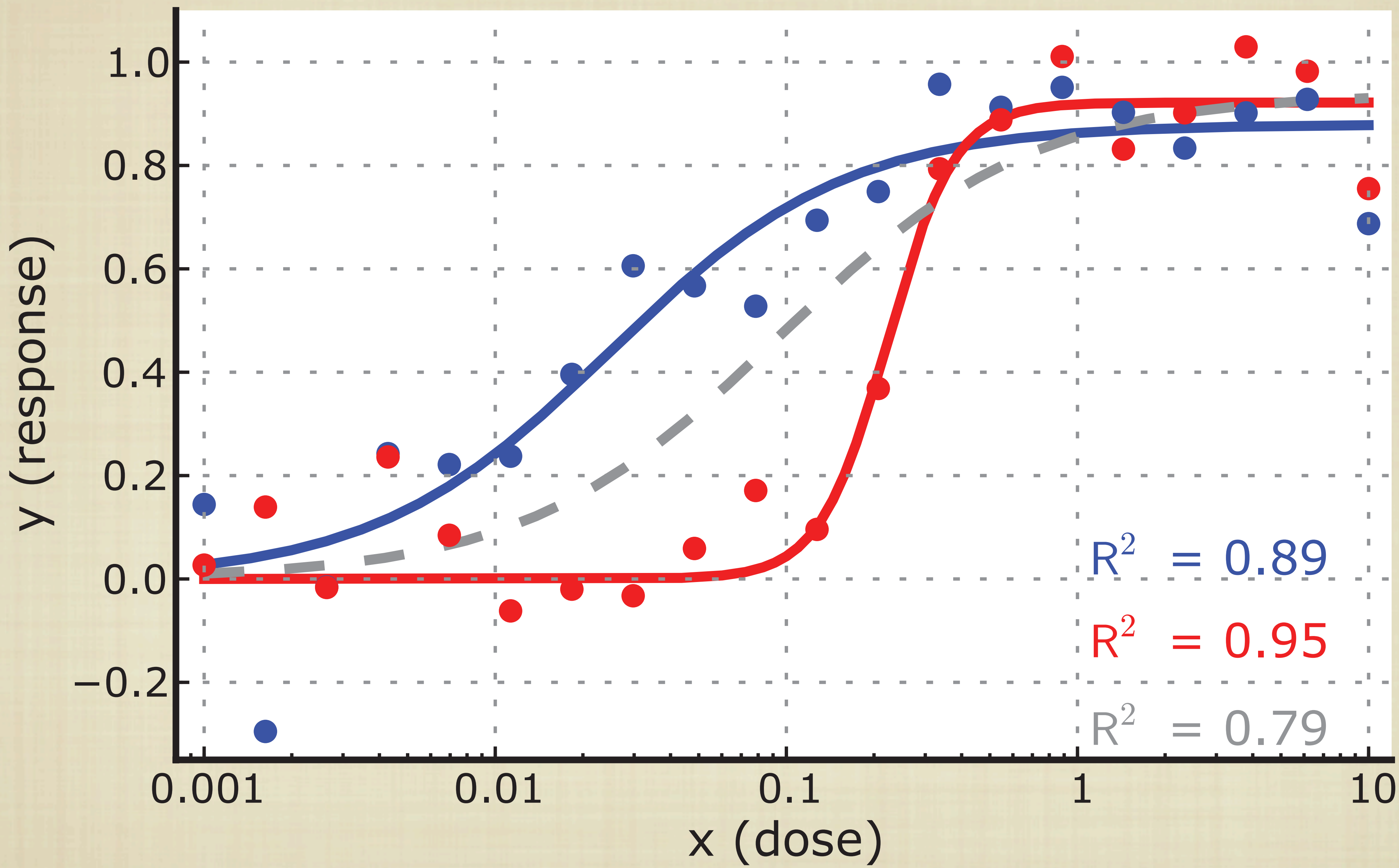
```matlab
%% Construct bootstrap samples for a nonlinear regression and estimate
%% bootstrap confidence intervals.

%% Fit simulated dose-response data to a Hill function, given by
%% y(x) = ymax/(1 + (EC50/dose)^n)

%% Load data
DoseResponse = load('DoseResponseData.dat', '-ascii');
dose = DoseResponse(:,1);
yExpt = DoseResponse(:,2);

%% Call nlinfit to perform nonlinear regression.
betaGuess = [1, 1, 1]; % Initial guess for parameter values
[betaHat, residuals, Jac, Cov, mse] = nlinfit(dose, yExpt, @Hill, betaGuess

%% Resample the residuals using the bootstrp function
nboot = 200; % Number of bootstrap replicates
[~, bootIndices] = bootstrp(nboot, [], residuals);
bootIndices(:,1:3) % Print the first 3 columns of bootIndices
bootResiduals = residuals(bootIndices);
bootResiduals(:, 1:3) % Print the first 3 columns of bootResiduals

%% Generate bootstrap datasets by adding the resampled residuals
%%to the best fit curve
yMod = Hill(betaHat, dose);
yBoot = repmat(yMod, 1, nboot) + bootResiduals;

%% Fit each of the bootstrap datasets to the Hill function to
%% build up the bootstrap distributions of parameter estimates
betaBoot = zeros(nboot, 3);
for i=1:nboot
    betaBoot(i,:) = nlinfit(dose, yBoot(:,i), @Hill, betaGuess);
end

% Estimate 95% CIs
bootCI = prctile(betaBoot,[2.5 97.5]);
```

Demo 5: Estimating bootstrap CIs in MATLAB



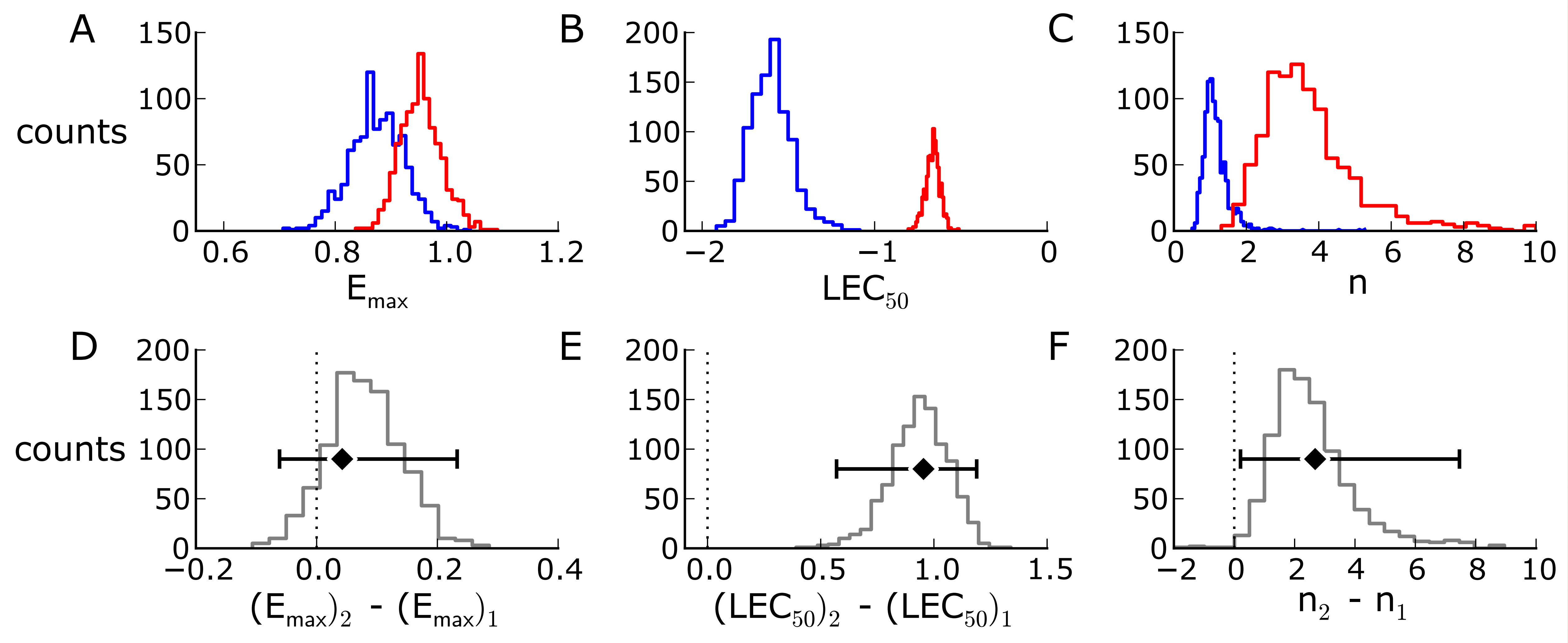
Comparing fits from two different experiments

A



- Which paramters are different between the two datasets?

Bootstrap-based hypothesis testing



- E_{max} is not (the 95% CI of the difference distribution crosses 0), but EC_{50} and n are different.

Friday

- How to pick the best model from a set of proposed models?
- Bias-variance tradeoff
- F-test
- Akaike's information criterion (AIC)