Statistics and Data Analysis in MATLAB
Kendrick Kay, kendrick.kay@wustl.edu

MATLAB In-Class Exercises 2 (covering Statistics Lectures 3 and 4)

Exercise 1. Construct an input space consisting of the polynomials x^0, x^1, x^2, x^3, and x^4 evaluated over the range [–1,1]. Then generate a weighted sum of these polynomials using random weights (e.g. `randn`). Repeat this a total of 20 times and visualize the results.

Exercise 2. Generate some random data using `rand(1,100).^4`. We want to find a value that summarizes the central tendency of these data, and suppose we have chosen the error metric of the sum of the absolute differences between the value and the data points. Plot the error surface, i.e. a function that illustrates the relationship between candidate values and the error associated with these values. For the candidate values, use 100 equally spaced values between 0 and 1 (e.g. `linspace(0,1,100)`.

Exercise 3. Generate some random data using `x = rand(1,1000); y = x+0.1*randn(1,1000);`. Using ordinary least-squares (OLS), fit a line to these data and visualize the results.

Exercise 4. Consider the model $y = a^x + b$. Using $a = 1.3$ and $b = 1$, generate some simulated data. Then use nonlinear optimization to fit the model to the data. For a visualization of the search process, download the file `outputfcnplot.m` and pass the options structure `optimset('Display','iter','OutputFcn' @(a,b,c)outputfcnplot(a,b,c,1,y))` to the optimizer (e.g. `lsqnonlin.m`). (Note that the options structure code assumes that `y` is defined in the workspace and refers to the simulated *y*-values.)

Exercise 5. Implement one-dimensional nearest-neighbor regression. To do this, first write a function that conforms to the declaration given below. Then, generate and visualize some data (*x*- and *y*-values). Finally, calculate the prediction of the nearest-neighbor regression model for the range of *x*-values shown in the plot and visualize the results.

```
function yy = nnregress(xx,x,y)

% function yy = nnregress(xx,x,y)
%
% <xx> is a vector of x-coordinates to evaluate at
% <x> is a vector of x-data
% <y> is a vector of y-data
%
% return a vector with the predictions of a
% nearest-neighbor regression model evaluated
% at the values in <xx>.
```