Maple for Models in
Module 8.3 on "Empirical Models"
File: *EmpiricalModels.mw*

Linear Empirical Model

Norris.dat
National Institute of Standards and Technology (NIST) "study involving calibration of ozone monitors," where *x* is "NIST's measurement of ozone concentration" and *y* is "the customer's measurement" (http://www.itl.nist.gov/div898/strd/lls/data/Norris.shtml)

Some of the data

```
> pts := [[0.2,0.1],[0.4,0.3],[0.3,0.3],[0.3,0.6]]:
```

Alternatively, data in two lists. Form a list of ordered pairs

```
> xLst := [0.2, 0.4, 0.3, 0.3]:
  yLst := [0.1, 0.3, 0.3, 0.6]:
```

Plot the points (Figure 8.3.1)

```
> with(plots):
  lp := listplot(pts,
     style = point, symbolsize = 20,
     labels = ["x", "y"],
     color = blue):
  display(lp);
```

Fit function

```
with(CurveFitting):
```

```
fiteq := LeastSquares( xLst, yLst , x,
    curve = m*x +) ;b
```

Plot function (Figure 8.3.2)

```
> pltfit := plot( fiteq, x = 0.0..0.6,
     color = blue):
  display(lp, pltfit);
```

Plot point at (0.34, 0.365) (Figure 8.3.3)

```
> plotPt := listplot([[0.34, 0.365]],
     style = point, symbolsize = 30,
     labels = ["x", "y"],
     color = red):
  display(lp, pltfit, plotPt);
```

Complete set of data

```
> xLst := [0.2, 337.4, 118.2, 884.6, 10.1, 226.5, 666.3, 996.3, 448.6,
  777.0, 558.2,
         0.4, 0.6, 775.5, 666.9, 338.0, 447.5, 11.6, 556.0, 228.1, 995.8,
  887.6,
         120.2, 0.3, 0.3, 556.8, 339.1, 887.2, 999.0, 779.0, 11.1, 118.3,
  229.2,
         669.1, 448.9, 0.5]:
  yLst := [0.1, 338.8, 118.1, 888.0, 9.2, 228.1, 668.5, 998.5, 449.1,
```

```
778.9, 559.2,
      0.3, 0.1, 778.1, 668.8, 339.3, 448.9, 10.8, 557.7, 228.3, 998.0,
888.8,
      119.6, 0.3, 0.6, 557.6, 339.3, 888.0, 998.5, 778.9, 10.2, 117.6,
228.9,
      668.4, 449.2, 0.2]:
```

Non-Linear One-Term Model

DanWood.dat
The variable *x* is "the absolute emperature of the filament in 1000 degrees Kelvin," while *y* is the "energy radieted from a carbon filament lamp per cm^2 per second."
(http://www.itl.nist.gov/div898/strd/nls/data/daniel_wood.shtml)
Reference:Data and model described inDaniel, C. and F. S. Wood (1980). "Fitting Equations to Data," Second Edition. New York, NY: John Wiley and Sons, pp. 428-431. Originally published in E.S.Keeping, "Introduction to Statistical Inference," Van Nostrand Company, Princeton, NJ, 1962, p. 354.

Plot data

```
> xLst := [1.309,1.471,1.490,1.565,1.611,1.680]:
  yLst := [2.138,3.421,3.597,4.340,4.882,5.660]:
  with(ListTools):
  pairsLst := Transpose([xLst, yLst]):
```

```
> with(plots):
  lp :=listplot(pairsLst, labels = [" x ", " y "],
     color = blue, style = point, symbolsize = 20):
  display(lp);
```

Plotting data with a line through the first and last lines helps us visualize the concavity (Figure 8.3.5).

```
> line :=listplot([[xLst[1], yLst[1]],[xLst[6], yLst[6]]],
        color = gray):
  display(lp, line);
```

Squaring *x* coordinates does not seem quite enough (Figure 8.3.6)

```
> xLstPower := map(x->x^2, xLst):
  lpPower :=listplot(Transpose([xLst2, yLst]),
     labels = [" x ", " y "],
     color = blue, style = point, symbolsize = 20):
  display(lpPower);
```

Cubing x coordinates does not seem quite enough (Figure 8.3.7)

```
> xLstPower := map(x->x^3, xLst):
  lpPower :=listplot(Transpose([xLstPower, yLst]),
     labels = [" x ", " y "],
     color = blue, style = point, symbolsize = 20):
  display(lpPower);
```

Raising x coordinates to the fourth power seems too much (Figure 8.3.8)

```
> xLstPower := map(x->x^4, xLst):
  lpPower :=listplot(Transpose([xLstPower, yLst]),
     labels = [" x ", " y "],
     color = blue, style = point, symbolsize = 20):
  display(lp);
```

Raising *x* coordinates to the power 3.5 seems better (Figure 8.3.9)

```
> xLstPower := map(x->x^3.5, xLst):
  lpPower :=listplot(Transpose([xLstPower, yLst]),
     labels = [" x ", " y "],
     color = blue, style = point, symbolsize = 20):
  display(lpPower);
```

Plot linear regression line with points (Figure 8.3.10)

$$fitData := y = 0.9881859455\, z - 0.3931312687$$

```
> fitData := LeastSquares( xLstPower, yLst , z,
      curve = m*z +b);
```

$$fitData := -0.3931312543 + 0.988185942356932245\, z$$

```
> plFit := plot( fitData , x = 0.0..7.5,
     color = blue):
  plots[display](lpPower, plFit);
```

Always, eventually plot actual and predicted data together (Figures 8.3.11 and 8.3.12)

```
> plFitSubst := eval(fitData, [z = x^3.5]);
```

$$plFitSubst := -0.3931312543 + 0.988185942356932245\, x^{3.5}$$

```
> plData := plot(plFitSubst, x = 0.0..1.7,
     color = blue):
  plots[display](lp, plData);
```

```
> plData := plot(plFitSubst, x = 1.3..1.7,
     color = blue):
  plots[display](lp, plData);
```

Solving for y in a One-Term Model

Misra1a.dat
The data is from NIST dental research by D. Misra (1978) in monomolecular adsorption, where x represents pressure and y volume.

(http://www.itl.nist.gov/div898/strd/nls/data/Misra1a.shtml)

Plot data (Figure 8.3.13)

```
> xLst:=[77.6,114.9,141.1,190.8,239.9,289.0,332.8,378.4,434.8,477.3,536.8,
  593.1,
        689.1,760.0]:
  yLst:=[10.07,14.73,17.94,23.93,29.61,35.18,40.02,44.82,50.76,55.05,61.01
  ,66.40,
        75.47,81.78]:
```

```
> pts := Transpose([xLst, yLst]):
```

```
> lp :=listplot(pts, labels = [" x ", " y "],
     color = blue, style = point, symbolsize = 20):
  display(lp);
```

Raising y coordinates to the power 6/5 seems linear (Figure 8.3.14)

```
> yLstPower := map(y->y^(6/5), yLst):
  lpPower :=listplot(Transpose([xLst, yLstPower]),
     labels = [" x ", " y "],
     color = blue, style = point, symbolsize = 20):
  display(lpPower);
```

```
> fitData := LeastSquares( xLst, yLstPower , x,
      curve = m*x +b);
```

$$fitData := -5.486294310 + 0.267868537772353642\, x$$

```
> plFit := plot( fitData , x = 0..800,
     color = blue):
  plots[display](lpPower, plFit);
```

Always, eventually plot actual and predicted data together. Solve for *y*. (Figure 8.3.15)

```
> plFitSubst := fitData^(5/6);
```

$$plFitSubst := (-5.486294310 + 0.267868537772353642\, x)^{(5/6)}$$

```
> plData := plot(plFitSubst, x = 77..770,
     color = blue):
  plots[display](lp, plData);
```

Multi-term Models
Filip.dat
The data is from NIST research.
(http://www.itl.nist.gov/div898/strd/nls/data/filip.shtml)

Plot data (Figure 8.3.16)

```
> xLst := [-6.860120914, -4.324130045, -4.358625055, -4.358426747,
  -6.955852379, -6.661145254, -6.355462942, -6.118102026, -7.115148017,
  -6.815308569, -6.519993057, -6.204119983, -5.853871964, -6.109523091,
  -5.79832982, -5.482672118, -5.171791386, -4.851705903, -4.517126416,
  -4.143573228, -3.709075441, -3.499489089, -6.300769497, -5.953504836,
  -5.642065153, -5.031376979, -4.680685696, -4.329846955, -3.928486195,
  -8.56735134, -8.363211311, -8.107682739, -7.823908741, -7.522878745,
  -7.218819279, -6.920818754, -6.628932138, -6.323946875, -5.991399828,
  -8.781464495, -8.663140179, -8.473531488, -8.247337057, -7.971428747,
  -7.676129393, -7.352812702, -7.072065318, -6.774174009, -6.478861916,
  -6.159517513, -6.835647144, -6.53165267, -6.224098421, -5.910094889,
  -5.598599459, -5.290645224, -4.974284616, -4.64454848, -4.290560426,
  -3.885055584, -3.408378962, -3.13200249, -8.726767166, -8.66695597,
  -8.511026475, -8.165388579, -7.886056648, -7.588043762, -7.283412422,
  -6.995678626, -6.691862621, -6.392544977, -6.067374056, -6.684029655,
  -6.378719832, -6.065855188, -5.752272167, -5.132414673, -4.811352704,
  -4.098269308, -3.66174277, -3.2644011]:

  yLst := [0.8116, 0.9072, 0.9052, 0.9039, 0.8053, 0.8377, 0.8667, 0.8809,
  0.7975, 0.8162, 0.8515, 0.8766, 0.8885, 0.8859, 0.8959, 0.8913, 0.8959,
  0.8971, 0.9021, 0.909 , 0.9139, 0.9199, 0.8692, 0.8872, 0.89   , 0.891 ,
  0.8977, 0.9035, 0.9078, 0.7675, 0.7705, 0.7713, 0.7736, 0.7775, 0.7841,
  0.7971, 0.8329, 0.8641, 0.8804, 0.7668, 0.7633, 0.7678, 0.7697, 0.77   ,
  0.7749, 0.7796, 0.7897, 0.8131, 0.8498, 0.8741, 0.8061, 0.846 , 0.8751,
  0.8856, 0.8919, 0.8934, 0.894 , 0.8957, 0.9047, 0.9129, 0.9209, 0.9219,
  0.7739, 0.7681, 0.7665, 0.7703, 0.7702, 0.7761, 0.7809, 0.7961, 0.8253,
```

```
  0.8602, 0.8809, 0.8301, 0.8664, 0.8834, 0.8898, 0.8964, 0.8963, 0.9074,
  0.9119, 0.9228]:
```

```
> pts := Transpose([xLst, yLst]):
```

```
> lp :=listplot(pts, labels = [" x ", " y "],
     color = blue, style = point, symbolsize = 20):
  display(lp);
```

Model with fourth degree polynomial (Figure 8.3.17)

```
> fitData := LeastSquares( xLst, yLst , x,
      curve =  g*x^4 + h*x^3 + i*x^2 + j*x + k);
```

$$fitData := 2.644405721 + 1.37440581328720900\,x + 0.00217486862304163515\,x^4 + 0.0492439091396152800\,x^3 + 0.397096879588375139\,x^2$$

```
> plFit := plot( fitData , x = -9..-3,
     color = blue):
  plots[display](lp, plFit);
```

Model with tenth degree polynomial (Figure 8.3.18)

```
> fitData := LeastSquares( xLst, yLst , x,
      curve = a*x^10 + b*x^9 + c*x^8 + d*x^7 + e*x^6 +
          f*x^5 + g*x^4 + h*x^3 + i*x^2 + j*x + k   );
```

$$fitData := -1404.015682 - 2654.51434683313710\,x - 340.148546674300292\,x^4 - 1081.69104004903056\,x^3 - 2219.75672827237850\,x^2 - 0.0000387384868271839331\,x^{10} - 0.00237226569405587498\,x^9 - 0.0644141591435777317\,x^8 - 1.02066450846299616\,x^7 - 10.4461702229166935\,x^6 - 72.1268693018197950\,x^5$$

```
> plFit := plot( fitData , x = -9..-3,
     color = blue):
  plots[display](lp, plFit);
```

Plot of tenth degree polynomial model inside and outside range of data (Figure 8.3.19)

```
> plFit := plot( fitData , x = -10..-2,
     color = blue):
  plots[display](plFit);
```

```
>
```