

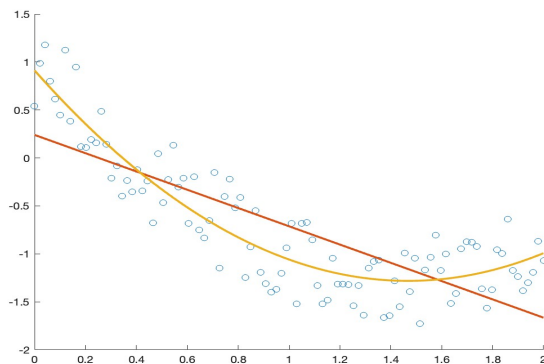
Linear Algebra II

lsq.m

```

1 T = csvread("data.csv");
2
3 % Get the data out of the table.
4 X = T(:,1);
5 Y = T(:,2);
6 f = linspace(0, 2, 100);
7
8 % Construct the linear.
9 P = [ones(size(X)) X];
10 p = inv(P'*P)*(P'*Y);
11
12 function y = l(x, p)
13     y = p(2)*x + p(1);
14 end
15
16 % Construct the quadratic.
17 A = [ones(size(X)) X X.^2];
18 b = inv(A'*A)*(A'*Y);
19
20 function y = q(x, b)
21     y = b(3)*x.^2 + b(2)*x + b(1);
22 end
23
24 % Find the normal matrix (A^T)A.
25 B = A.'*A;
26
27 scatter(X, Y);
28 hold on
29 plot(f, l(f, p), "LineWidth", 2);
30 plot(f, q(f, b), "LineWidth", 2);
31 hold off

```

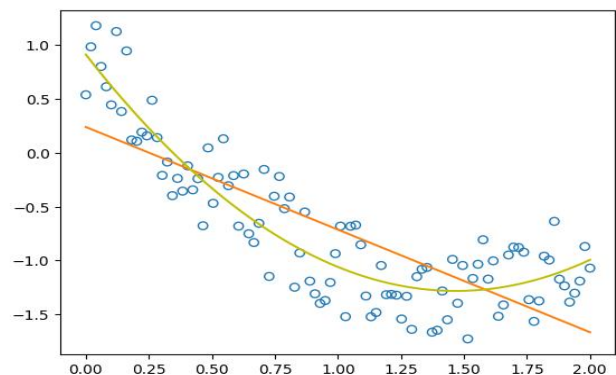


lsq.py

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from numpy.linalg import inv
4
5 D = np.loadtxt("data.csv", delimiter=
6     ",")
7 X, Y = D[:,0], D[:,1]
8 f = np.linspace(0, 2, 100)
9
10 # Linear approximation.
11 P = np.array([np.ones(len(D)), X]).T
12 p = inv(P.T @ P) @ (P.T @ Y)
13
14 def l(x, p):
15     return p[1]*x + p[0]
16
17 # Quadratic approximation.
18 A = np.array([
19     np.ones(len(D)), X, X**2
20 ]).T
21 b = inv(A.T @ A) @ (A.T @ Y)
22
23 def q(x, b):
24     return b[2]*x**2 + b[1]*x + b[0]
25
26 # Find the normal matrix.
27 B = A.T @ A
28
29 plt.scatter(X, Y, facecolors="none",
30     edgecolors="tab:blue")
31 plt.plot(X, l(X, p), c="tab:orange")
32 plt.plot(X, q(X, b), c="y")
33 plt.show()

```



We construct a system of equations called the *normal equations*: given our matrix A , the expression $(A^T A)^{-1}(A^T y)$ computes the orthogonal projection from y onto the column span of Ax . Equivalently, we are minimizing $\|Ab - y\|^2$ — this minimizer is the vector b of squared differences between

each entry of y and the closest point in Ax . Note: we're minimizing something quadratic, so it has a unique solution given by

$$\begin{aligned} Ab &= y, \\ A^\top Ab &= A^\top y, \\ b &= (A^\top A)^{-1}(A^\top y). \end{aligned}$$

We can construct A with $n + 1$ columns to get a degree- n polynomial fit. In this case, the data above were generated by setting an objective function $\tilde{f}(x) = x^2 - 3x + 1$ and constructing a noised dataset

$$D = \left\{ (x_i, \tilde{f}(x_i) \pm \mathcal{N}(0, 1/4) : x_i = 2i/20, i \in \{0, \dots, 20\} \right\},$$

where $\mathcal{N}(0, 1/4)$ is a normal distribution with standard deviation $\sigma = 1/4$.