M647, Spring 2012, Practice Problems for the Midterm

The midterm for M647 will be Thursday March 8, 7:00-9:00 p.m. in Blocker 122. Class will be canceled for Friday March 9. The exam will consist of two parts: Part 1 will not require MATLAB, while Part 2 will require MATLAB. Students will have to turn in Part 1 before starting Part 2, but for Part 2 students will have access to all M-files we've used this semester, from both lecture and homework. Students will be expected to access data files from the course web site.

This is *not* a practice midterm. In particular, this set of problems is much longer than the midterm will be.

Part 1 Questions

1. Suppose you have data $\{(x_k, y_k, z_k)\}_{k=1}^N$ that appears consistent with the model

$$z = ax + by.$$

Use the method of least squares regression to find a value for b in terms of the data. Your value for a can be specified in terms of your value for b.

2. The Pareto probability density function has the form

$$f(x; a, b) = \begin{cases} \frac{ab^a}{x^{a+1}} & x > b \\ 0 & x < b, \end{cases}$$

where a and b are both taken to be positive parameters.

2a. Suppose X is distributed according to the Pareto distribution, and compute $E[X]$ for $a > 1$. (What happens for $0 < a \leq 1$?)

2b. Given observations $\{x_k\}_{k=1}^N$ find the MLE estimate for a for the Pareto distribution with $b = 1$.

2c. Write down the MLE estimator for a for the Pareto distribution with $b = 1$.

2d. Given observations $\{x_k\}_{k=1}^N$ find MLE estimates for both a and b for the Pareto distribution.

3. For linear regression, we said in class that our MLE estimator for $\vec{p}$ is

$$\hat{p} = (F^{tr}F)^{-1}F^{tr}\vec{Y},$$

where each component Y_k of $\vec{Y}$ is a Gaussian $N(\mu_k, \sigma^2)$ random variable, with $\mu_k = (F\vec{p})_k$. Show that for each $j = 1, 2, \dots, m$,

$$\begin{aligned} E[\hat{p}_j] &= p_j \\ \text{Var}[\hat{p}_j] &= V_{jj}\sigma^2, \end{aligned}$$

where the V_{jj} are diagonal elements of the *curvature* matrix $V = (F^{tr}F)^{-1}$. Derive these two equalities.

4. An object is shot straight upward with initial velocity v. Ignoring air resistance, and assuming gravity is the only force acting on the object, use dimensional analysis to determine the general form (i.e., the form up to a multiplicative constant) for the greatest height the object achieves. Discuss whether or not your result makes sense physically.

5. For a fluid such as oil moving through a pipe, the velocity v at a certain point along the pipe will generally depend on the diameter D of the pipe, the density ρ of the fluid, the viscosity μ of the fluid, and the pressure drop $\frac{dp}{dx}$ at the point. Ignoring viscosity (though see Problem 6), find the dependence of v on the other variables.

6. This problem regards the situation described in Problem 5.

6a. Find a general form for v if viscosity is not ignored.

6b. Suppose you would like to build a large pipeline with $D = 10$ m, but you would first like to run experiments on a table model with $D = .05$ m. If you want to use the same fluid for both your model and your pipeline, how must the pressure drops be related?

Note. We will say quite a bit about viscosity later in the course during our section on PDE, but for this problem we only require its dimensions, $[\mu] = ML^{-1}T^{-1}$.

7. Use the method of dimensional analysis to determine a general form for the period P of a pendulum of length l released from angle θ with angular velocity ω. Ignore air resistance.

Part 2 Questions

On the exam, you will need to send your solution M-files for the questions from Part 2 to my email address phoward@math.tamu.edu.

1. In our reference *A Concrete Approach to Mathematical Modelling*, by Mesterton-Gibbons, the author gives data for economic growth of the Massachusetts ecomony, 1890-1926. In particular, three ratios are given for each year—a labor index L, a capital index K, and an output index Q—and the following model is posed:

$$Q = aL^{\gamma}K^{1-\gamma},$$

where $a > 0$ and it is expected that $0 < \gamma < 1$. The data is defined in *lkq.m*, available on the course web site.

1a. Find regression values for a and γ, and compute the approximate standard deviation s for your fit.

1b. Find 95% confidence intervals for your parameter values from (a).

1c. Compute means μ with error estimates for $(L, K) = (1, 1)$, $(L, K) = (.5, .5)$, and $(L, K) = (5, 10)$, at 95% confidence.

1d. Analyze the standardized residuals for your fit at 68%.

2. The general single species population model is

$$\frac{dy}{dt} = \frac{r}{a}y(1 - (\frac{y}{K})^a); \quad y(0) = y_0,$$

which can be solved exactly as

$$y(t) = \frac{Ky_0}{\left(y_0^a + (K^a - y_0^a)e^{-rt}\right)^{1/a}}.$$

It's easy to see that if $a = 1$ this is simply the logistic model, and it's straightforward to show, using L'Hospital's rule, that the Gompertz model is obtained in the limit as $a \to 0$. Fit this model to the U. S. population data in *uspop.m* (available on the course web site), and use your results to argue that the Gompertz model is the best model from this family for fitting U.S. population growth. Certainly one of the things you'll want to compute for this problem is the standard deviation s associated with your fit.

Note. Physically a should be a positive parameter. Use MATLAB's documentation on *lsqcurvefit* to find out how to incorporate a lower bound on your parameter values.

3. When an object such as a marble falls through a viscous fluid such as oil its terminal velocity depends on the marble's radius r, the gravitational acceleration $g = 9.81\ m/s^{-2}$, the fluid viscosity μ, and the density difference $\Delta\rho = \rho_m - \rho_f$, where ρ_m denotes marble density and ρ_f denotes fluid density.

3a. Find a general relationship for the dependence of v on the variables r, g, μ, and $\Delta\rho$.

3b. Use the data $\{(r_k, g_k, \mu_k, d\rho_k, v_k)\}_{k=1}^{10}$ in the MATLAB M-file *marbledata.m* (available on the course web site) to complete your model from Part (a).

3c. Use your model to predict v for the case $r = .0065$, $\mu = 15kg{\cdot}m^{-1}s^{-1}$, $\Delta\rho = 672.8kg{\cdot}m^{-3}$. This experiment was carried out during the undergraduate modeling class Fall 2009 (Dial soap, marble, tennis ball can), and we timed $v = .0097m/s$.

Solutions

Part 1 Solutions

1. We proceed by minimizing the error

$$E(a, b) = \sum_{k=1}^{N} (z_k - ax_k - by_k)^2.$$

We have

$$\frac{\partial E}{\partial a} = \sum_{k=1}^{N} 2(z_k - ax_k - by_k)(-x_k) = 0$$

$$\frac{\partial E}{\partial b} = \sum_{k=1}^{N} 2(z_k - ax_k - by_k)(-y_k) = 0,$$

which gives two equations for the two unknowns a and b

$$a\sum_{k=1}^{N} x_k^2 + b\sum_{k=1}^{N} x_k y_k = \sum_{k=1}^{N} x_k z_k$$
$$a\sum_{k=1}^{N} x_k y_k + b\sum_{k=1}^{N} y_k^2 = \sum_{k=1}^{N} y_k z_k.$$

The problem specifies that we should eliminate a and solve for b, so we multiply the second equation by

$$\frac{\sum_{k=1}^{N} x_k^2}{\sum_{k=1}^{N} x_k y_k}$$

and subtract the result from the first equation. We find

$$b\Big(\sum_{k=1}^{N} x_k y_k - \frac{\sum_{k=1}^{N} x_k^2 \sum_{k=1}^{N} y_k^2}{\sum_{k=1}^{N} x_k y_k}\Big) = \sum_{k=1}^{N} x_k z_k - \frac{\sum_{k=1}^{N} x_k^2 \sum_{k=1}^{N} y_k z_k}{\sum_{k=1}^{N} x_k y_k},$$

so that

$$b = \frac{\sum_{k=1}^{N} x_k z_k - \frac{\sum_{k=1}^{N} x_k^2 \sum_{k=1}^{N} y_k z_k}{\sum_{k=1}^{N} x_k y_k}}{\sum_{k=1}^{N} x_k y_k - \frac{\sum_{k=1}^{N} x_k^2 \sum_{k=1}^{N} y_k^2}{\sum_{k=1}^{N} x_k y_k}}.$$

We use b to compute a,

$$a = \frac{\sum_{k=1}^{N} y_k z_k - b\sum_{k=1}^{N} y_k^2}{\sum_{k=1}^{N} x_k y_k}.$$

Alternatively, we could write the design matrix

$$F = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_N & y_N \end{pmatrix},$$

and compute

$$\begin{pmatrix} a \\ b \end{pmatrix} = (F^{tr}F)^{-1}F^{tr} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{pmatrix}.$$

Here,

$$F^{tr}F = \begin{pmatrix} \sum_{k=1}^{N} x_k^2 & \sum_{k=1}^{N} x_k y_k \\ \sum_{k=1}^{N} x_k y_k & \sum_{k=1}^{N} y_k^2 \end{pmatrix},$$

so that

$$(F^{tr}F)^{-1} = \frac{1}{\sum_{k=1}^{N} x_k^2 \sum_{k=1}^{N} y_k^2 - (\sum_{k=1}^{N} x_k y_k)^2} \begin{pmatrix} \sum_{k=1}^{N} y_k^2 & -\sum_{k=1}^{N} x_k y_k \\ -\sum_{k=1}^{N} x_k y_k & \sum_{k=1}^{N} x_k^2 \end{pmatrix},$$

and

$$F^{tr}\vec{z} = \begin{pmatrix} \sum_{k=1}^N x_k z_k \\ \sum_{k=1}^N y_k z_k \end{pmatrix}.$$

This gives

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \frac{\sum_{k=1}^N y_k^2 \sum_{k=1}^N x_k z_k - \sum_{k=1}^N x_k y_k \sum_{k=1}^N y_k z_k}{\sum_{k=1}^N x_k^2 \sum_{k=1}^N y_k^2 - (\sum_{k=1}^N x_k y_k)^2} \\ \frac{-\sum_{k=1}^N x_k y_k \sum_{k=1}^N x_k z_k + \sum_{k=1}^N x_k^2 \sum_{k=1}^N y_k z_k}{\sum_{k=1}^N x_k^2 \sum_{k=1}^N y_k^2 - (\sum_{k=1}^N x_k y_k)^2} \end{pmatrix}.$$

2. For Part (a),

$$E[X] = \int_b^\infty ab^a x^{-a} dx = ab^a \frac{x^{-a+1}}{-a+1}\Big|_b^\infty = -ab^a \frac{b^{-a+1}}{-a+1} = \frac{ab}{a-1},$$

where in the evaluation we assumed $a > 1$. If $0 < a \le 1$, the expected value is ∞.

For Part (b), the likelihood function is

$$L(a) = \prod_{k=1}^N \frac{a}{x_k^{a+1}} = a^N \prod_{k=1}^N x_k^{-1-a},$$

and the log-likelihood function is

$$L^*(a) = N \ln a - (a+1) \sum_{k=1}^N \ln x_k.$$

We compute

$$\frac{dL^*(a)}{da} = \frac{N}{a} - \sum_{k=1}^N \ln x_k = 0 \Rightarrow a = \frac{N}{\sum_{k=1}^N \ln x_k}.$$

For Part (c), the MLE estimator for a is

$$\hat{a} = \frac{N}{\sum_{k=1}^N \ln X_k}.$$

For Part (d), the likelihood function is

$$L(a,b) = \prod_{k=1}^N \frac{ab^a}{x_k^{a+1}} = a^N b^{aN} \prod_{k=1}^N x_k^{-a-1},$$

and the log-likelihood function is

$$L^*(a,b) = N \ln a + aN \ln b - (a+1) \sum_{k=1}^N \ln x_k.$$

In order to maximize this, we compute

$$\frac{\partial L^*}{\partial a}(a,b) = \frac{N}{a} + N\ln b - \sum_{k=1}^{N} \ln x_k$$
$$\frac{\partial L^*}{\partial b} = \frac{aN}{b}.$$

We cannot set the b-derivative to 0, and so we observe instead that since $\frac{aN}{b} > 0$ L^* is monotonic in b; in particular, we must take b as large as possible. We observe, however, that b cannot exceed any data values (the probability of getting an observed data value can't be 0), so we take

$$b = \min_{k\in\{1,2,\ldots,N\}} x_k =: x_{(1)}.$$

Upon substitution of $b = x_{(1)}$ into the first equation, we find

$$a = \frac{N}{\sum_{k=1}^{N} \ln x_k - N \ln x_{(1)}}.$$

3. We verified the first of these relations in class,

$$E[\hat{p}] = (F^{tr}F)^{-1}F^{tr}E[\vec{Y}] = (F^{tr}F)^{-1}F^{tr}F\vec{p} = \vec{p}.$$

For the second, we have at least two options.

Method 1. We compute

$$\begin{aligned}\mathrm{Cov}(\hat{p}) &= \mathrm{Cov}(VF^{tr}\vec{Y}) = VF^{tr}\mathrm{Cov}(\vec{Y})(VF^{tr})^{tr} \\ &= VF^{tr}\sigma^2 IFV^{tr} = \sigma^2 VV^{-1}V^{tr} = \sigma^2 V^{tr}.\end{aligned}$$

The claim is immediate since $\mathrm{Cov}(\hat{p})_{jj} = \mathrm{Var}[\hat{p}_j]$.

Method 2. Essentially rederiving the identity from class used for Method 1,

$$\begin{aligned}\mathrm{Var}[\hat{p}_j] &= \sigma^2 \sum_{k=1}^{N}((VF^{tr})^2_{jk} = \sigma^2 \sum_{k=1}^{N}(\sum_{l=1}^{m} V_{jl}F^{tr}_{lk})^2 \\ &= \sigma^2 \sum_{k=1}^{N}\sum_{l=1}^{m} V_{jl}F^{tr}_{lk} \sum_{p=1}^{m} V_{jp}F^{tr}_{pk} = \sigma^2 \sum_{l=1}^{m}\sum_{p=1}^{m} V_{jl}V_{jp} \sum_{k=1}^{N} F^{tr}_{lk}F_{kp} \\ &= \sigma^2 \sum_{l=1}^{m}\sum_{p=1}^{m} V_{jl}V_{jp}(F^{tr}F)_{lp} = \sigma^2 \sum_{p=1}^{m} V_{jp}(V(F^{tr}F))_{jp}.\end{aligned}$$

Now, $V(F^{tr}F) = I$, and the claim follows immediately.

4. In this case it's clear that the variables of dependence are g and v so that

$$h = h(g,v) = kg^a v^b,$$

with dimensions

$$L = L^a T^{-2a} L^b T^{-b}.$$

The dimension equations are

$$\begin{aligned} L &: 1 = a + b \\ T &: 0 = -2a - b. \end{aligned}$$

Adding these equations, we find $a = -1$, which gives $b = 2$. We can conclude

$$h = k\frac{v^2}{g}.$$

This makes sense because greater initial speeds will cause the object to rise higher while greater gravitational force will reduce the maximum height.

5. we assume $v = v(D, \rho, \frac{dp}{dx}) = kD^a \rho^b (\frac{dp}{dx})^c$, with dimensions

$$LT^{-1} = L^a M^b L^{-3b} M^c L^{-2c} T^{-2c}.$$

This gives the dimension equations

$$\begin{aligned} L &: 1 = a - 3b - 2c \\ T &: -1 = -2c \\ M &: 0 = b + c, \end{aligned}$$

which imply $c = \frac{1}{2}$, $b = -\frac{1}{2}$, and $a = \frac{1}{2}$. We conclude

$$v = k\sqrt{\frac{D\frac{dp}{dx}}{\rho}}.$$

6. In this case we must use dimensionless products, so we write

$$\pi = \pi(D, \rho, \frac{dp}{dx}, \mu, v) = D^a \rho^b (\frac{dp}{dx})^c \mu^d v^e,$$

with dimensions

$$1 = L^a M^b L^{-3b} M^c L^{-2c} T^{-2c} M^d L^{-d} T^{-d} L^e T^{-e}.$$

The dimension equations are

$$\begin{aligned} L &: 0 = a - 3b - 2c - d + e \\ M &: 0 = b + c + d \\ T &: 0 = -2c - d - e. \end{aligned}$$

In this case, we have two degrees of freedom. For π_1, we set $e = 0$ and $d = 1$ to get the reduced system

$$\begin{aligned} 0 &= a - 3b - 2c - 1 \\ 0 &= b + c + 1 \\ 0 &= -2c - 1, \end{aligned}$$

from which we have $c = -\frac{1}{2}$, $b = -\frac{1}{2}$, and $a = -\frac{3}{2}$. We conclude

$$\pi_1 = \frac{\mu}{\sqrt{\rho(\frac{dp}{dx})D^3}}.$$

For π_2, we take $e = 1$ and in this case $c = 0$ (not necessary, but gives Reynold's number) to get the reduced system

$$\begin{aligned}
0 &= a - 3b - d + 1 \\
0 &= b + d \\
0 &= -d - 1,
\end{aligned}$$

from which we have $d = -1$, $b = 1$, and $a = 1$. We conclude

$$\pi_2 = \frac{v\rho D}{\mu},$$

which is Reynold's number. According to Buckingham's Theorem and the Implicit Function Theorem, there will generally be a function ϕ so that

$$\pi_2 = \phi(\pi_1).$$

This means

$$\frac{v\rho D}{\mu} = \phi(\frac{\mu}{\sqrt{\rho(\frac{dp}{dx})D^3}}),$$

and so

$$v = \frac{\mu}{\rho D}\phi(\frac{\mu}{\sqrt{\rho(\frac{dp}{dx})D^3}}).$$

For (b), we require

$$\frac{\mu}{\sqrt{\rho(\frac{dp}{dx})D^3}} = \frac{\mu_e}{\sqrt{\rho_e(\frac{dp}{dx})_e D_e^3}},$$

and since $\rho = \rho_e$ and $\mu = \mu_e$ this gives

$$\frac{(\frac{dp}{dx})_e}{(\frac{dp}{dx})} = \frac{D^3}{D_e^3} = \frac{10^3}{.05^3} = 8 \times 10^6.$$

7. The period P depends now on four variables, θ, ω, g, and l. We look for dimensionless products

$$\pi = \pi(\theta, \omega, g, l, P) = \theta^a \omega^b g^c l^d P^e,$$

which gives (keeping in mind that θ has no dimension and ω is an angular velocity)

$$1 = T^{-b} L^c T^{-2c} L^d T^e,$$

from which we obtain two equations:

$$\begin{aligned} L: &\quad 0 = c + d \\ T: &\quad 0 = -b - 2c + e. \end{aligned}$$

We have two equations and four variables (notice that a does not appear in either equation), and since there is no redundancy we have two degrees of freedom. (Clearly, you can also check the row reduced echelon form of A. Also, it might be more in the spirit of our calculations to say we have three degrees of freedom, since we will also be making choices on a.) For π_1, we use $\pi_1 = \theta$. For π_2, we choose $e = 0$ and $d = 1$ to give $c = -1$ and $b = 2$, so that $\pi_1 = \frac{l\omega^2}{g}$. For π_3, we choose $e = 1$ and $d = 0$ to give $c = 0$ and $b = 1$, so that $\pi_3 = \omega P$. Buckingham's Theorem and the Implicit Function Theorem suggest that there exists some function ϕ so that

$$\pi_3 = \phi(\pi_1, \pi_2),$$

or

$$P = \frac{1}{\omega}\phi(\theta, \frac{l\omega^2}{g}).$$

Note. We might prefer the choice

$$\pi_3 = \sqrt{\frac{g}{l}}P,$$

which would give

$$P = \sqrt{\frac{l}{g}}\phi(\theta, \frac{l\omega^2}{g}).$$

One reason this is preferable is that we easily recover the case $\omega = 0$.

Part 2 Solutions

1. We begin by transforming this relation to a linear form

$$\ln(\frac{Q}{K}) = \ln a + \gamma \ln(\frac{L}{K}).$$

Plotting $\ln(Q/K)$ versus $\ln(L/K)$, and fitting a regression line, we obtain the initial guess

$$a = 1.0077, \quad \gamma = .7435.$$

The remainder of the problem is solved with the MATLAB M-file *lkqfit.m.*

```
%LKQFIT: MATLAB script M-file for analyzing the labor-capital-output
%data in lkq.
%
%Define the data
lkq;
%Carry out a linear fit
```

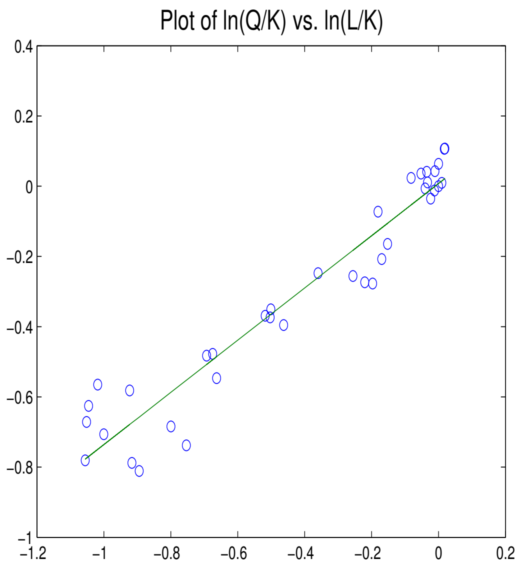

Figure 1: Figure for the labor-capital-output data.

```
p = polyfit(log(L./K),log(Q./K),1);
plot(log(L./K),log(Q./K),'o',log(L./K),p(1)*log(L./K)+p(2))
title('Plot of ln(Q/K) vs. ln(L/K)','FontSize',15)
%
%initial guess at parameter values
gam = p(1)
a = exp(p(2))
%
%nonlinear fit
clear p
E=@(p) norm(Q-p(1)*L.^p(2).*K.^(1-p(2)))^2;
[p,error]=fminsearch(E,[a gam])
q = length(Q)-length(p);
s = sqrt(error/q)
%
%Confidence intervals
Q_a = L.^p(2).*K.^(1-p(2));
Q_gam = p(1)*L.^p(2).*log(L).*K.^(1-p(2))-p(1)*L.^p(2).*K.^(1-p(2)).*log(K);
F = [Q_a' Q_gam'];
delQ = Q-p(1)*L.^p(2).*K.^(1-p(2));
delp = F\delQ'
lsd = sqrt(norm(delQ'-F*delp).^2/q)
pnew = p'+delp
%95% confidence intervals
l = stut(q,.95);
V = inv(F'*F);
```

```
error = l*sqrt(lsd^2*diag(V))
```

A diary session associated with this file is given below.

```
>>lkqfit
gam =
0.7435
a =
1.0077
p =
1.0036 0.7357
error =
0.9342
s =
0.1634
delp =
1.0e-04 *
0.2854
0.4620
lsd =
0.1634
pnew =
1.0037
0.7357
error =
0.0679
0.0920
```

We conclude that $s = .1634$, and we have the following parameter estimates:

$$a = 1.0037 \pm .0679$$
$$\gamma = .7357 \pm .0920.$$

For (c) we use lkqfit2.m.

```
function lkqfit2(x)
%LKQFIT: MATLAB script M-file for analyzing the labor-capital-output
%data in lkq. Computes values \mu_x along with error estimates.
%Input should be a vector [L,K]
%
%Define the data
lkq;
%Carry out a linear fit
p = polyfit(log(L./K),log(Q./K),1);
```

```
plot(log(L./K),log(Q./K),'o',log(L./K),p(1)*log(L./K)+p(2))
title('Plot of ln(Q/K) vs. ln(L/K)','FontSize',15)
%
%initial guess at parameter values
gam = p(1)
a = exp(p(2))
%
%nonlinear fit
clear p
E=@(p) norm(Q-p(1)*L.^p(2).*K.^(1-p(2)))^2;
[p,error]=fminsearch(E,[a gam])
q = length(Q)-length(p);
s = sqrt(error/q)
%
%Confidence intervals
Qf = @(y1,y2) p(1).*y1.^p(2).*y2.^(1-p(2));
Q_a = @(y1,y2) y1.^p(2).*y2.^(1-p(2));
Q_gam = @(y1,y2) p(1)*y1.^p(2).*log(y1).*y2.^(1-p(2))-p(1)*y1.^p(2).*y2.^(1-
p(2)).*log(y2);
F = [Q_a(L,K)' Q_gam(L,K)'];
delQ = Q-Qf(L,K);
delp = F\delQ'
mu1 = [Q_a(x(1),x(2)) Q_gam(x(1),x(2))]*delp;
%Prediction
mu_x = Qf(x(1),x(2))+mu1
%For 95% confidence interval
l = stut(q,.95);
fx = [Q_a(x(1),x(2)) Q_gam(x(1),x(2))];
FL = (F'*F)\F';
a = FL'*fx';
c = norm(a)^2;
%Linear standard deviation
lsd = sqrt(norm(delQ' - F*delp)^2/q);
error = l*sqrt(lsd*c)
```

The implementation is given below.

```
>>lkqfit2([1 1])
gam =
0.7435
a =
1.0077
p =
1.0036 0.7357
error =
```

0.9342
s =
0.1634
delp =
1.0e-04 *
0.2854
0.4620
mu_x =
1.0037
error =
0.1681
>>lkqfit2([.5 .5])
gam =
0.7435
a =
1.0077
p =
1.0036 0.7357
error =
0.9342
s =
0.1634
delp =
1.0e-04 *
0.2854
0.4620
mu_x =
0.5018
error =
0.0841
>>lkqfit2([5 10])
gam =
0.7435
a =
1.0077
p =
1.0036 0.7357
error =
0.9342
s =
0.1634
delp =
1.0e-04 *
0.2854
0.4620

mu_x =
6.0271
error =
0.4924

We see

$$\mu_{(1,1)} = 1.0037 \pm .1681$$
$$\mu_{(.5,.5)} = .5018 \pm .0841$$
$$\mu_{(5,10)} = 6.0271 \pm .4924.$$

For (d), we use *lkqfit3.m*.

```
%LKQFIT3: MATLAB script M-file for analyzing the labor-capital-output
%data in lkq. Analyzes standardized residuals.
%
%Define the data
lkq;
%Carry out a linear fit
p = polyfit(log(L./K),log(Q./K),1);
plot(log(L./K),log(Q./K),'o',log(L./K),p(1)*log(L./K)+p(2))
title('Plot of ln(Q/K) vs. ln(L/K)','FontSize',15)
%
%initial guess at parameter values
gam = p(1)
a = exp(p(2))
%
%nonlinear fit
clear p
E=@(p) norm(Q-p(1)*L.^p(2).*K.^(1-p(2)))^2;
[p,error]=fminsearch(E,[a gam])
q = length(Q)-length(p);
s = sqrt(error/q)
%
Qf = @(y1,y2) p(1).*y1.^p(2).*y2.^(1-p(2));
Q_a = @(y1,y2) y1.^p(2).*y2.^(1-p(2));
Q_gam = @(y1,y2) p(1)*y1.^p(2).*log(y1).*y2.^(1-p(2))-p(1)*y1.^p(2).*y2.^(1-
p(2)).*log(y2);
F = [Q_a(L,K)' Q_gam(L,K)'];
delQ = Q-Qf(L,K);
delp = F\delQ'
M=F*((F'*F)\F');
s = sqrt(norm(delQ'-F*delp)^2/q);
r = (delQ'-F*delp)./(s*sqrt(1-diag(M)));
```

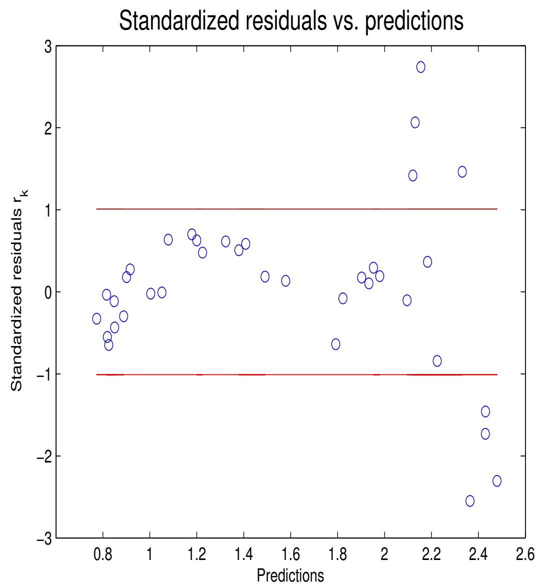

Figure 2: Figure for Problem 2.

```
l = stut(q,.68);
%Predictions
pr = Qf(L,K)'+F*delp;
plot(pr,r,'o',pr,-l*ones([1 length(pr)]),'r',pr,l*ones([1 length(pr)]),'r');
title('Standardized residuals vs. predictions','FontSize',15)
xlabel('Predictions')
ylabel('Standardized residuals r_k')
%
dex=0;
for j=1:length(r)
if abs(r(j)) <= l
dex=dex+1;
end
end
dex
ratio = dex/length(r)
```

The figure this creates is given as Figure 2. The ratio is .7838. The ratio isn't all that much of a concern, but the scattering of the data does not look sufficiently random.

2. For this model, there doesn't seem to be a reasonable way to get approximate parameter values directly, so we'll begin with the parameter values obtained in class for the logistic model. These are $r = .0208$, $K = 486.8046$, and $y0 = 8.2241$. We also take $a = 1$. We use the M-file *ussinglespecies.m*.

```
%USSINGLESPECIES: MATLAB script M-file that carries out a
```

```
%nonlinear regression for U.S. population data modeled
%by the general single-species population model
%Define data
uspop
%Define GSSM solution (p(1)=r, p(2)=K, p(3)=y0, p(4)=a)
y = @(p,t) p(2)*p(3)./(p(3)^p(4)+(p(2)^p(4)-p(3)^p(4))*exp(-p(1)*t)).^(1/p(4));
%Start with logistic fit values
p0 = [.0208 486.8046 8.2241 1];
%options=optimset('MaxFunEvals',1e6)
[p error]=lsqcurvefit(y,p0,decades,pops,[0 0 0 0],[])
sd=sqrt(error/(length(decades)-4))
%
%Plot model along with data
modelpops = y(p,decades);
figure
plot(decades,pops,'o',decades,modelpops)
title('General Single Species Population Fit for U.S. Data','FontSize',15)
```

Running this, we find

$$\begin{aligned} r &= .0061 \\ K &= 1.3928 \times 10^3 \\ y_0 &= 4.3762 \\ a &= 3.5591 \times 10^{-6}, \end{aligned}$$

and the sample standard deviation is

$$s = 3.1753.$$

Also, MATLAB's termination text in this case read *lsqcurvefit stopped because the size of the current step is less than the default value of the step size tolerance.* This suggests that MATLAB is simply trying to make a smaller; i.e. MATLAB is doing the best it can to get the Gompertz model.

2. We begin by looking for dimensionless products

$$\pi = \pi(r, g, \mu, \Delta\rho, v) = r^a g^b \mu^c \Delta\rho^d v^e,$$

with dimensions

$$1 = L^a L^b T^{-2b} M^c L^{-c} T^{-c} M^d L^{-3d} L^e T^{-e},$$

and dimension equations

$$\begin{aligned} L: 0 &= a + b - c - 3d + e \\ T: 0 &= -2b - c - e \\ M: 0 &= c + d. \end{aligned}$$

In matrix form this is

$$\begin{pmatrix} 1 & 1 & -1 & -3 & 1 \\ 0 & -2 & -1 & 0 & -1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

In row reduced echelon form,

$$\begin{pmatrix} 1 & 0 & 0 & -\frac{3}{2} & \frac{1}{2} \\ 0 & 1 & 0 & -\frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

We have

$$\begin{aligned} a &= \frac{3}{2}d - \frac{1}{2}e \\ b &= \frac{1}{2}d - \frac{1}{2}e \\ c &= -d. \end{aligned}$$

For π_1, we choose $e = 0$ and $d = 1$, which gives $a = \frac{3}{2}$, $b = \frac{1}{2}$, and $c = -1$. We conclude

$$\pi_1 = \frac{r^{3/2} g^{1/2} \Delta\rho}{\mu}.$$

For π_2, we choose $e = 1$ and $d = 0$, which gives $a = -\frac{1}{2}$, $b = -\frac{1}{2}$, and $c = 0$. We conclude

$$\pi_2 = \frac{v}{\sqrt{rg}}.$$

Buckingham's Theorem and the Implicit Function Theorem suggest there exists a function ϕ so that

$$\pi_2 = \phi(\pi_1),$$

or equivalently

$$\frac{v}{\sqrt{rg}} = \phi(\frac{r^{3/2} g^{1/2} \Delta\rho}{\mu}) \Rightarrow v = \sqrt{rg}\phi(\frac{r^{3/2} g^{1/2} \Delta\rho}{\mu}).$$

The fit for Part (b) is carried out in the MATLAB M-file *marblefit.m*.

```
%MARBLEFIT: MATLAB script M-file that fits the data in
%marbledata to a choice of dimensionless products
%
%define data
marbledata;
```

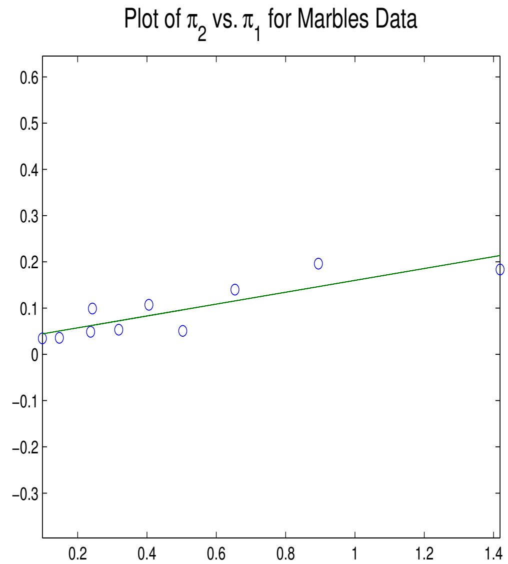

Figure 3: Figure for the marbles data.

```
pi1 = delrho.*r.^(3/2)*sqrt(g)./mu;
pi2 = v./sqrt(r*g);
p = polyfit(pi1,pi2,1)
plot(pi1,pi2,'o',pi1,p(1)*pi1+p(2))
axis equal
title('Plot of \pi_2 vs. \pi_1 for Marbles Data','FontSize',15)
%
%model
v = @(r,g,mu,delrho) p(1)*r^2*g*delrho/mu+p(2)*sqrt(r*g);
%prediction
v(.0065,9.81,15,672.8)
```

The plot this creates appears in Figure 3.

We find that

$$\phi(\pi_1) = .1283\pi_1 + .0317,$$

and so

$$v = .1283(\frac{r^2 g\Delta\rho}{\mu}) + .0317\sqrt{rg}.$$

For Part (c), we find $v = .0104$.

By the way, it's easy to verify, using Newtonian mechanics, that the theoretical expression for v is

$$v = \frac{2r^2 g\Delta\rho}{9\mu}.$$

This would correspond with $p_1 = \frac{2}{9}$ and $p_2 = 0$ in our fit.