Homework 4

Robert Ackerman
University of North Carolina

July 18, 2014

1 Multivariate Kernel Density Estimator

Let $Z_1, ..., Z_n$ be i.i.d. copies of a random variable $Z \sim f(x)$ where $Z = (Y, X)$.

1.1 What is the kernel density estimator of $f(z_o)$?

$$\hat{f}(x_0, y_0) = \frac{1}{Nh^2} \sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h_x}\right) K\left(\frac{y_i - y_0}{h_y}\right)$$

Where, if we make the simplifying assumption $h_x = h_y$ becomes:

$$\hat{f}(x_0, y_0) = \frac{1}{Nh^2} \sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right) K\left(\frac{y_i - y_0}{h}\right) \tag{1}$$

1.2 What is the bias and variance of this estimator?

Recall,

$$\begin{aligned}
Bias &= \mathbb{E}\left[\hat{f}(x_0, y_0)\right] - f(x_0, y_0) \\
&= \mathbb{E}\left[\frac{1}{Nh^2} \sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right) K\left(\frac{y_i - y_0}{h}\right)\right] - f(x_0, y_0) \\
&= \frac{1}{h^2}\mathbb{E}\left[K\left(\frac{x_1 - x_0}{h}\right) K\left(\frac{y_1 - y_0}{h}\right)\right] - f(x_0, y_0) \text{ (by i.i.d)} \\
&= \frac{1}{h^2} \int\int K\left(\frac{x_1 - x_0}{h}\right) K\left(\frac{y_i 1 - y_0}{h}\right) f(x_1, y_1) \partial x_1 \partial y_1 - f(x_0, y_0) \\
&= \frac{1}{h^2} \int\int K(u)K(v) f(x_0 + hu, y_0 + hv) h\partial u h\partial v - f(x_0, y_0) \\
&\left(\text{Where } u = \frac{x_1 - x_0}{h}, v = \frac{y_1 - y_0}{h}, \implies \partial x_1 = h\partial u, \partial y_1 = h\partial v\right)
\end{aligned}$$

$$= \int \int K(u)K(v)f(x_0 + hu, y_0 + hv)\partial u \partial v - f(x_0, y_0) \tag{2}$$

Now, a taylor series expansion of $f(x_0 + hu, y_0 + hv)$ gives:

$$f(x_0 + hu, y_0 + hv) = \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} \frac{h^{j+k}}{j!k!} \left(\frac{\partial^{j+k} f(x_0, y_0)}{\partial^j u \partial^k v} \right) u^j v^k$$

Simplifying notation:

$$C_{j,k}(x_0, y_0) = \frac{1}{j!k!} \left(\frac{\partial^{j+k} f(x_0, y_0)}{\partial^j u \partial^k v} \right)$$

Yields:

$$= \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} h^{j+k} C_{j,k}(x_0, y_0) u^j v^k \tag{3}$$

Plugging (3) into (2) gives:

$$\begin{aligned}
&\int \int K(u)K(v) \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} h^{j+k} C_{j,k}(x_0, y_0) u^j v^k \partial u \partial v - f(x_0, y_0) \\
=& \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} h^{j+k} C_{j,k}(x_0, y_0) \left[\int K(u) u^j \partial u \right] \left[\int K(v) v^k \partial v \right] - f(x_0, y_0) \\
=& \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} h^{j+k} C_{j,k} \mu_j \mu_k - f(x_0, y_0)
\end{aligned}$$

Where μ_j and μ_k are the j^{th} and k^{th} moments of the x and y kernels.

$$
\begin{aligned}
=&f(x_0,y_0)\int K(u)\partial u\int K(v)\partial v\\
+&h\frac{\partial f(x_0,y_0)}{\partial v}\int K(u)\partial u\int vK(v)\partial v+\dots\\
+&h^{p-1}\frac{\partial^{p-1}f(x_0,y_0)}{\partial^{p-1}v}\int K(u)\partial u\int v^{p-1}K(v)\partial v\\
+&h^{p}\frac{\partial^{p}f(x_0,y_0)}{\partial^{p}v}\int K(u)\partial u\int v^{p}K(v)\partial v+\dots\\
+&h\frac{\partial f(x_0,y_0)}{\partial v}\int uK(u)\partial u\int K(v)\partial v+\dots\\
+&h^{p-1}\frac{\partial^{p-1}f(x_0,y_0)}{\partial^{p-1}u}\int u^{p-1}K(u)\partial u\int K(v)\partial v\\
+&h^{p}\frac{\partial^{p}f(x_0,y_0)}{\partial^{p}u}\int u^{p}K(u)\partial u\int K(v)\partial v\\
+&\sum_{j=p+1}^{\infty}\sum_{k=p+1}^{\infty}h^{j+k}C_{j,k}\mu_j\mu_k-f\left(x_0,y_0\right)
\end{aligned}
$$

We note the following:

$$\int K(u)\partial u=\int K(v)\partial v=1$$

(by definition the integral of a density is one)

$$\int uK(u)\partial u=\int vK(v)\partial v=0$$

(by assumption that our kernel is symmetric around 0)

$$\int u^{p-1}K(u)\partial u=\int v^{p-1}K(v)\partial v=0$$

(since we are using a p^{th} order kernel)

Applying these to the terms in the previous step leaves:

$$
\begin{aligned}
&=f(x_0,y_0)+h^p\frac{\partial^p f(x_0,y_0)}{\partial^p v}\int v^pK(v)\\
&+h^p\frac{\partial^p f(x_0,y_0)}{\partial^p u}\int u^pK(u)\partial u\\
&+\sum_{j=p+1}^{\infty}\sum_{k=p+1}^{\infty}h^{j+k}C_{j,k}\mu_j\mu_k-f\left(x_0,y_0\right)
\end{aligned}
$$

$$
\begin{aligned}
&= h^p \frac{\partial^p f(x_0, y_0)}{\partial^p v} \int v^p K(v) + h^p \frac{\partial^p f(x_0, y_0)}{\partial^p u} \int u^p K(u) \partial u \\
&+ \sum_{j=p+1}^{\infty} \sum_{k=p+1}^{\infty} h^{j+k} C_{j,k} \mu_j \mu_k \\
&= h^p \left[\frac{\partial^p f(x_0, y_0)}{\partial^p v} \int v^p K(v) + \frac{\partial^p f(x_0, y_0)}{\partial^p u} \int u^p K(u) \partial u \right] \\
&+ \sum_{j=p+1}^{\infty} \sum_{k=p+1}^{\infty} h^{j+k} C_{j,k} \mu_j \mu_k \\
&= O(h^p) + O(h^{p+1}) + O(h^{p+2}) \dots
\end{aligned}
$$

Finally, since the first term is the dominant term we have:

$$
Bias = O\left(h^p\right) \tag{4}
$$

Now, we turn to variance:

$$
\begin{aligned}
V\left[\hat{f}(x_0, y_0)\right] &= V\left[\frac{1}{Nh^2} \sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right) K\left(\frac{y_i - y_0}{h}\right)\right] \\
&= \frac{1}{N^2 h^4} V\left[\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right) K\left(\frac{y_i - y_0}{h}\right)\right] \\
&= \frac{1}{N^2 h^4} V\left[K\left(\frac{x_1 - x_0}{h}\right) K\left(\frac{y_1 - y_0}{h}\right)\right] \text{(by i.i.d.)} \\
&= \frac{1}{N^2 h^4}\left[\mathbb{E}\left[K^2\left(\frac{x_1 - x_0}{h}\right) K^2\left(\frac{y_1 - y_0}{h}\right)\right] - \mathbb{E}^2\left[K\left(\frac{x_1 - x_0}{h}\right) K\left(\frac{y_1 - y_0}{h}\right)\right]\right]
\end{aligned}
$$

From (**??**), we note the second term:

$$
\mathbb{E}^2\left[K\left(\frac{x_1 - x_0}{h}\right) K\left(\frac{y_1 - y_0}{h}\right)\right] = \left[O(h^2)\right]^2 = O(h^4)
$$

So, we have:

$$
\begin{aligned}
V\left[\hat{f}(x_0, y_0)\right] &= \frac{1}{N^2 h^4}\left[\mathbb{E}\left[K^2\left(\frac{x_1 - x_0}{h}\right) K^2\left(\frac{y_1 - y_0}{h}\right)\right] - O(h^4)\right] \\
&= \frac{1}{N^2 h^4}\left[\int\int\left[K^2\left(\frac{x_1 - x_0}{h}\right) K^2\left(\frac{y_1 - y_0}{h}\right) f(x_1, y_1) \partial x_1 \partial y_1\right] - O(h^4)\right]
\end{aligned}
$$

Applying a change of variables as before gives:

$$= \frac{1}{N^2h^4}\left[\int\int\left[K^2(u)K^2(v)f(x_{0+hu}, y_0+hv)h\partial u h\partial v\right] - O(h^4)\right]$$

(substituting in (3))

$$= \frac{1}{N^2h^4}\left[\int\int\left[K^2(u)K^2(v)\sum_{j=0}^{\infty}\sum_{k=0}^{\infty}h^{j+k}C_{j,k}\left(x_0,y_0\right)u^jv^kh^2\partial u\partial v\right] - O(h^4)\right]$$

$$= \frac{1}{N^2h^4}\left[\sum_{j=0}^{\infty}\sum_{k=0}^{\infty}h^{j+k}C_{j,k}\left(x_0,y_0\right)\int\int\left[K^2(u)K^2(v)u^jv^kh^2\partial u\partial v\right] - O(h^4)\right]$$

(based on the expansion in the bias derivation)

$$= \frac{1}{N^2h^4}\left[O(h^2) - O(h^4)\right]$$

(Again, the first term is dominant)

$$= \frac{1}{N^2h^4}\left[O(h^2)\right]$$

$$= O\left(\frac{1}{N^2h^4}\times h^2\right) = O\left(\frac{1}{Nh^2}\right)$$

So finally we have,

$$V\left[\hat{f}(x_0,y_0)\right] = O\left(\frac{1}{Nh^2}\right) \tag{5}$$

1.3 What is the optimal bandwidth that minimizes the MSE of this estimator? Explain the curse of dimensionality by comparing the rate of convergence of the estimator of $f(z_0)$ using optimal bandwidth with that done in class for a scalar Z.

Recall,

$$\begin{aligned}
MSE(\hat{\theta}) &= \mathbb{E}[(\hat{\theta}-\theta)^2] = \mathbb{E}[((\hat{\theta}-\mathbb{E}[\hat{\theta}]) + (\mathbb{E}[\hat{\theta}]-\theta))^2] \\
&= \mathbb{E}[(\hat{\theta}-\mathbb{E}[\hat{\theta}])^2 + (\mathbb{E}[\hat{\theta}]-\theta)^2 + 2(\hat{\theta}-\mathbb{E}[\hat{\theta}])(\mathbb{E}[\hat{\theta}]-\theta)] \\
&\text{(the last term} = 0 \text{ by L.I.E.)} \\
&= \mathbb{E}[(\hat{\theta}-\mathbb{E}[\hat{\theta}])^2 + (\mathbb{E}[\hat{\theta}]-\theta)^2] \\
&= \left(\mathbb{E}[\hat{\theta}-\theta)\right)^2 + \mathbb{E}\left[(\hat{\theta}-\mathbb{E}(\hat{\theta}))^2\right] \\
&= Bias^2 + Variance \\
&\approx c_1h^{2p} + c_2\frac{1}{Nh^2}
\end{aligned}$$

Then to find the optimal bandwidth we take the F.O.C. of MSE w.r.t. h:

$$\begin{aligned}
\frac{\partial MSE}{\partial h} &= 2pc_1 h^{2p-1} - 2c_2 \frac{1}{Nh^3} \\
0 &= 2pc_1 h^{2p-1} - 2c_2 \frac{1}{Nh^3} \\
2c_2 \frac{1}{Nh^3} &= 2pc_1 h^{2p-1} \\
c_2 \frac{1}{Nh^3} &= pc_1 h^{2p-1} \\
c_2 \frac{1}{N} &= pc_1 h^{2p-1} h^3 \\
c_2 \frac{1}{Npc_1} &= h^{2p-1} h^3 \\
c_2 \frac{1}{Npc_1} &= h^{2p-1+3} \\
c_2 \frac{1}{Npc_1} &= h^{2p+2}
\end{aligned}$$

$\implies h^* = O\left(N^{-\frac{1}{2p+2}}\right)$

In class we derived, $h^* = O\left(N^{-\frac{1}{2p+1}}\right)$. Since optimal h depends on sample size means increasing the dimension means that we need to increase our sample size, which is the curse of dimensionality.

1.4 We are often interested in finding $f(z)$ for the entire support of Z. Assume that this support is compact. Find out the optimal bandwidth by minimizing IMSE instead of MSE.

$$IMSE = \int_z MSE[\hat{f}(z)\partial z \tag{6}$$

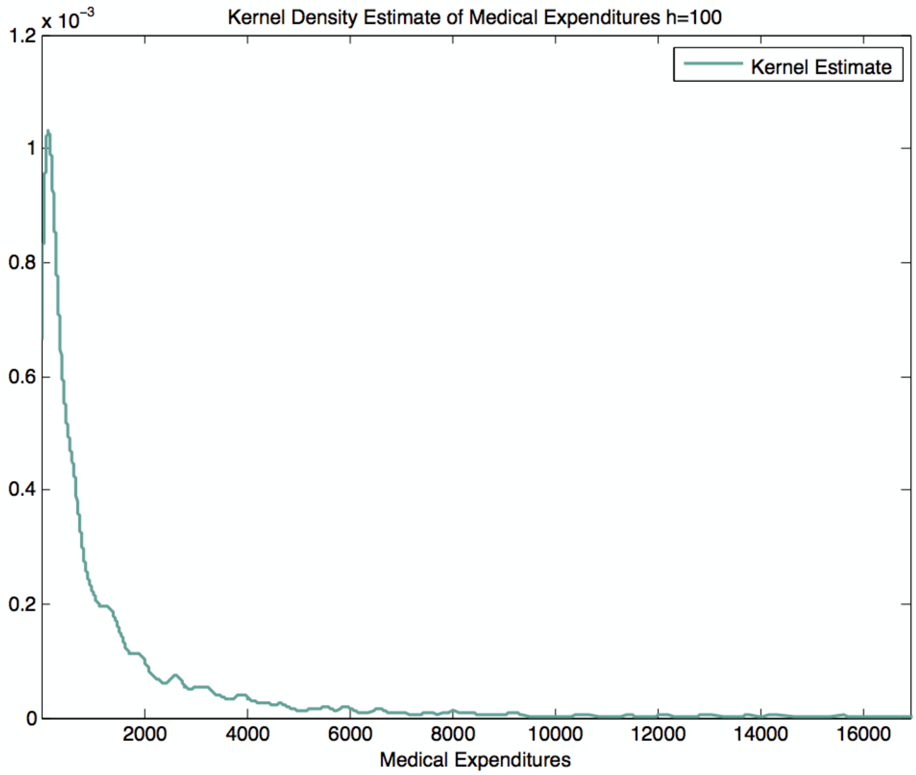
To find the optimal h, take the F.O.C. of (6) w.r.t. h and set it equal to zero:

$$
\begin{aligned}
\frac{\partial IMSE}{\partial h} &= \frac{\partial}{\partial h}\int_z MSE[\hat{f}(z)\partial z = 0 \\
&= \int_z \frac{\partial}{\partial h} MSE[\hat{f}(z)\partial z = 0 \\
&\text{(Using what we found in the previous section)} = 0 \\
&= \int_z \left(2pc_1 h^{2p-1} - 2c_2\frac{1}{Nh^3}\right) = 0 \\
2h^{2p-1}\int_z pc_1 &= \frac{2}{Nh^3}\int_z c_2 \\
h^{2p+2} &= \frac{1}{N}\left(\int_z pc_1\right)^{-1}\int_z c_2 \\
h^* &= \left[\frac{1}{N}\left(\int_z pc_1\right)^{-1}\int_z c_2\right]^{\frac{1}{2-+2}} \\
h^* &= O\left(N^{-\frac{1}{2p+2}}\right)
\end{aligned}
$$

2 Problem 9-3 of Cameron and Trivedi *Microeconometrics* (page 335)

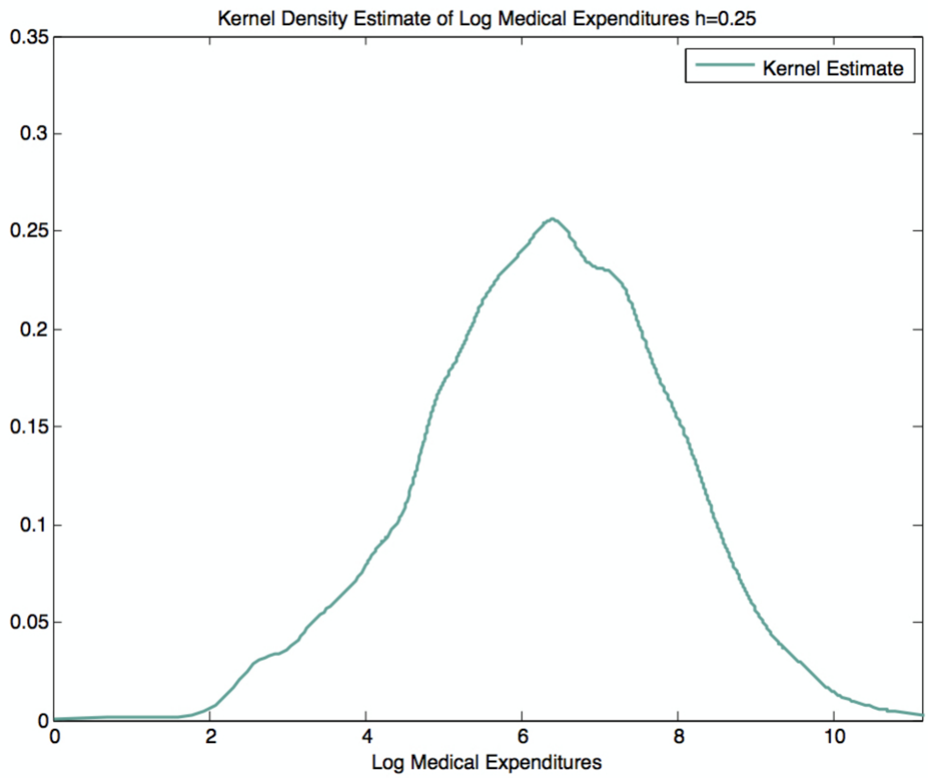
Use the Section 4.6.4 data on health expenditure. Use a kernel density estimate with Gaussian kernel.

2.1 Obtain the kernel density estimate for health expenditure, choosing a suitable bandwidth by eyeballing and trial and error. State the bandwidth chosen.



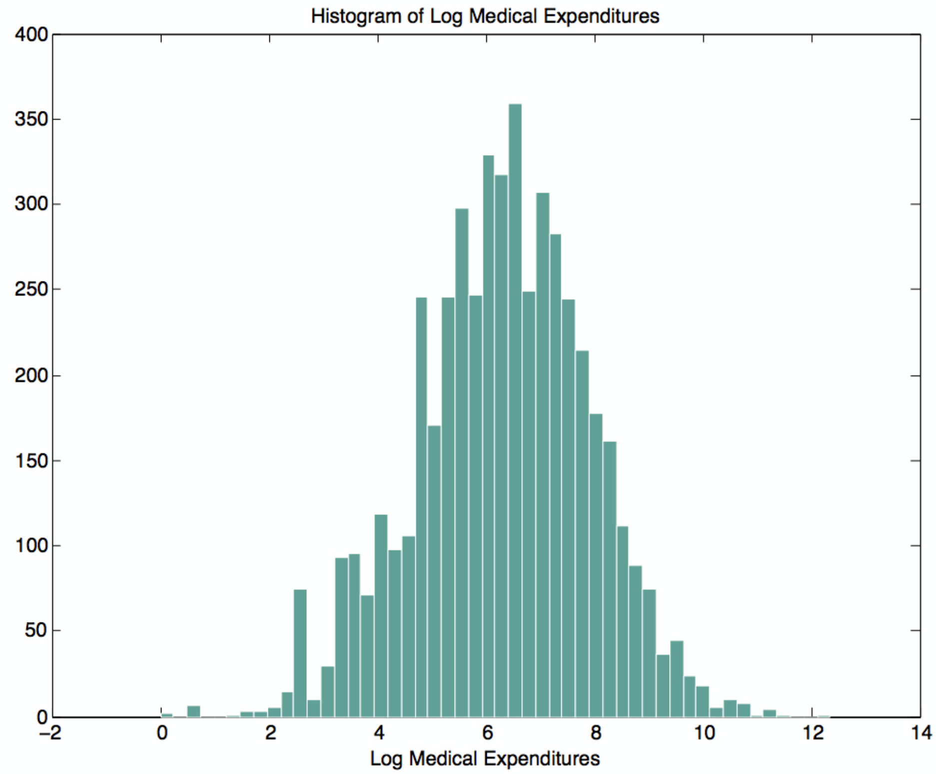
As noted in the figure title, this estimate is using a bandwidth of $h = 100$.

2.2 Obtain the kernel density estimate for natural logarithm of health expenditure, choosing a suitable bandwidth by eyeballing and trial and error. State the bandwidth chosen.



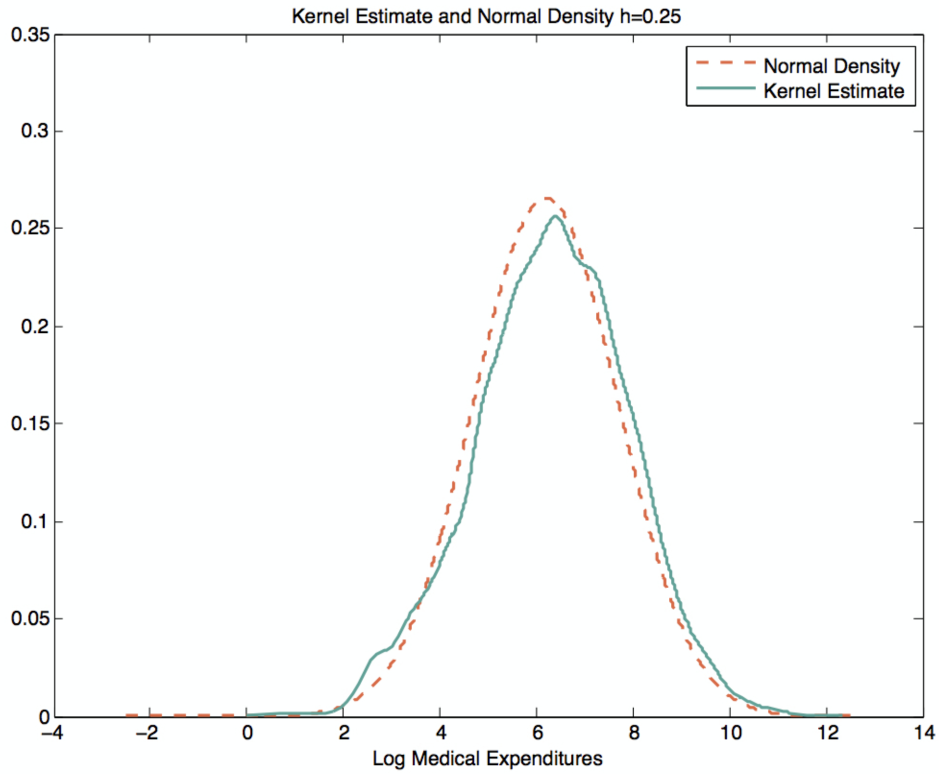
As noted in the figure title, this estimate is using a bandwidth of $h = 0.25$.

2.3 Compare your answer in part (b) to an appropriate histogram.



The kernel estimate, and the histogram estimate yield similar results. The shape of the estimates for the density of log medical care expenditures are roughly the same, but the kernel density estimate is obviously smoother.

2.4 If possible superimpose a fitted normal density on the same graph as the kernel density estimate from part (b). Do health expenditures appear to be log-normally distributed?



The kernel estimated density is very close to the normal density, so log medical expenditures do appear to be log-normally distributed.

3 Gregory and Veall (1985), Formulating Wald Tests of Nonlinear Restrictions, *Econometrica*, 53, 1465-1468.

Suppose that you have n observations $(y_t, X_{1t}, X_{2t}) : t = 1, ..., n$ from the linear regression model

$$y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \epsilon_t. \tag{7}$$

You want to test $H_g : g(\beta_1, \beta_2) = 0$ against $K_g : g(\beta_1, \beta_2) \neq 0$ where $g(\beta_1, beta_2) = \beta_1 - 1/\beta_2$. Note that the same test can alternatively be done by testing $H_h : h(\beta_1, \beta_2) = 0$ against $K_h : h(\beta_1, \beta_2) \neq 0$ where $h(\beta_1, \beta_2) = \beta_1\beta_2 - 1$.

Consider the following DGP based on (**??**):

$$\beta_0 = 1, \beta_1 = 10, \beta_2 = 0.1,$$

$$\begin{bmatrix} X_{1,t} \\ Y_{2,t} \end{bmatrix} = \begin{bmatrix} 0.6 & 0.3 \\ 0.3 & 0.6 \end{bmatrix} \begin{bmatrix} X_{1,t-1} \\ Y_{2,t-1} \end{bmatrix} + \begin{bmatrix} V_{1,t} \\ V_{2,t} \end{bmatrix}$$

Where $(\epsilon_t, V_{1,t}, V_{2,t}) \overset{i.i.d.}{\sim} N(0, I_3)$

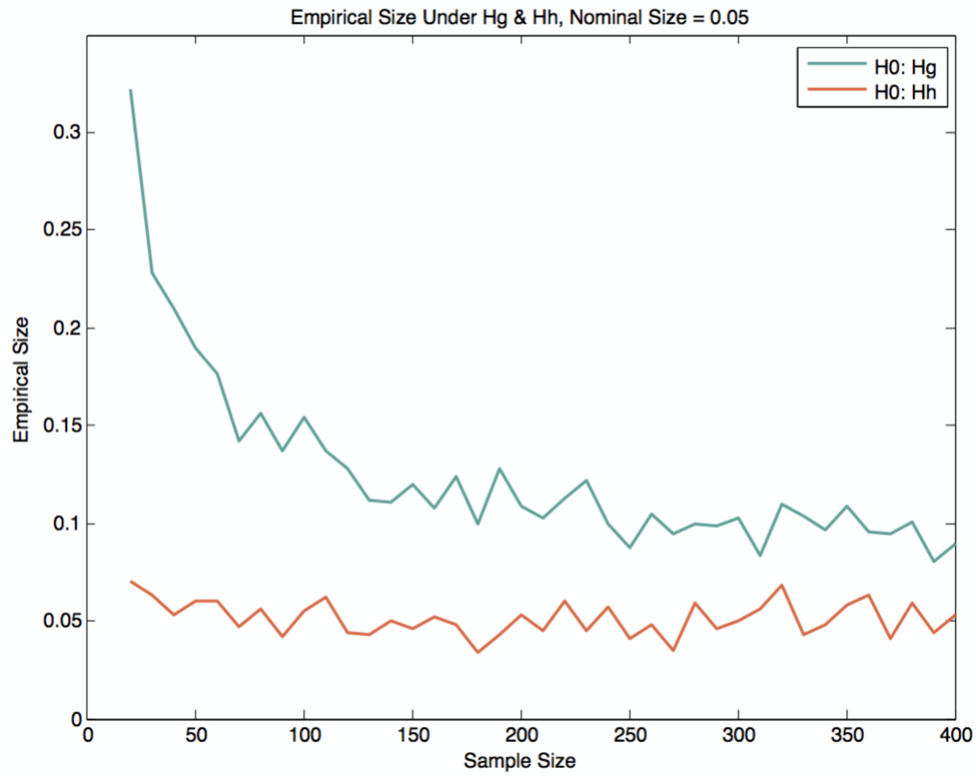
3.1 Draw $n = 20$ observations from the model and perform the Wald tests for H_g and H_h. Keeping everything else fixed, repeat the experiment 1000 times by drawing a new series of $\{\epsilon_t\}$ each time. Perform the two Wald tests every time. Report the empirical size of the two tests.

Part (a) Results	
	Empirical Size
H_g	0.3220
H_h	0.0700

Empirical size is calculated based on the percentage of times the null is rejected in our 1000 experiments. Obviously, with small sample size the form of the null hypothesis can have a substantial impact on the empirical size of your results.

3.2 Redo part (a) for each sample size $n \in \{20, 30, 40, \ldots, 400\}$ and draw a figure as follows. The x-axis has a grid for n, while the y-axis has the empirical size. You plot two lines: the empirical size based on the g-type test and that based on the h-type test. Hint: when $n = 400$, the empirical size of both tests should be fairly close to nominal size, 5%

.



As we increase sample size, the discrepancy between the two forms of the null hypothesis begins to decrease. At $n = 400$ both are getting close to nominal size 0.05, although there is still a difference and the form of the null still matters for the empirical size of our test.

3.3 Now let us see if bootstrap can help us obtain a better approximation of nominal size when n is as small 20. To do residual-based bootstrap, take the number of bootstrap replication $B = 999$ (this is entirely under your control). Follow the algorithm described below for H_g.

. . .

Repeat the Monte Carlo experiment 1000 times to obtain the empirical size of the test based on bootstrap critical value. Now do the same for the null hypothesis H_h. Compare your results with that in part (a).

Part (c) Results

	Empirical Size	Empirical Size with Bootstrap
H_g	0.3220	0.209
H_h	0.0700	0.008

Again, empirical size is calculated based on the percentage of times the null is rejected in our 1000 experiments. Bootstrap gets empirical size closer to nominal size for H_g, but it actually makes it worse for H_h.

My code for producing these results is contained in the following section.

4 Codes for Problem 3 (c)

My codes for part (c) are not easily separated from my code for all of Question 3, so I have included that main file in its entirety. I have also included the three short function codes that fmincon in the main code calls upon for the NLS Objective function, and the H_g and H_h constraints.

4.1 Question 3 Main File

```
% Robert Ackerman
% ECON 870: Advanced Econometrics
% Fall 2013, UNC Chapel Hill
% Instructor: Prof. Saraswata Chaudhuri
% TA: Kaiji Motegi

% Homework 4
% Problem 3
% November 23, 2013

% Gregory, Allan W. and Veall, Michael R. (1985) "Formulating Wald Tests of
% Nonlinear Restrictions". Econometrica Vol. 53, No. 6(Nov., 1985), pp.
% 1465-1468).

%% Step 1: Prelminary Settings
clear all; clc;
tic;
reps=1000;
nparam = 3;
nvec = (20:10:400);
empsizeg = zeros(size(nvec));
empsizeh = zeros(size(nvec));
%% (a) n=20 & (b) Looping for Different n

for k=1:1
    n=nvec(k);
% Fix seed for v1.
defaultStream = RandStream.getGlobalStream;
defaultStream.reset(0);

% DGP [X1,t X2,t]'= [0.6 0.3: 0.3 0.6][X1,t-1 X2,t-1]'+[V1,t V2,t]'
v1 = randn(n,1);

% Fix seed for v2.
defaultStream = RandStream.getGlobalStream;
defaultStream.reset(1);
v2 = randn(n,1);

% X
x = zeros(n,3);
x(:,1) = 1;
x(1,2) = 0 + v1(1);
x(1,3) = 0 + v2(1);
for i=2:n
    x(i,2) = x(i-1,2)*0.6 + x(i-1,3)*0.3 + v1(i);
    x(i,3) = x(i-1,2)*0.3 + x(i-1,3)*0.6 + v2(i);
end


% Model & Wald Test
wg = zeros(reps,1);
wh = zeros(reps,1);
sizeg = zeros(reps,1);
sizeh = zeros(reps,1);
pvalg = zeros(reps,1);
```

```
pvalh = zeros(reps,1);
wald = zeros(reps,2);
for i=1:1000
    eps = randn(n,1);
    y = 1 + 10*x(:,2) + 0.1*x(:,3) + eps;
    b = ((x'*x) \ eye(3)) *x'*y;
    resid = y-x*b;
    b_var = (x'*x\eye(3))*(x'*((1/(n-nparam)).*resid'*resid)*x)*(x'*x
\eye(3));
    g = (b(2) - 1/b(3));
    G = [0 1 1/(b(3)^2)];
    wg(i) = (b(2) - 1/b(3))^2/((b_var(2,2) - 2*b(3)^-2*b_var(2,3) +  +
b(3)^-4*b_var(3,3)));
    pvalg(i) = 1 - chi2cdf(wg(i),1);
        if pvalg(i) < 0.05
        sizeg(i) = 1;
        else
        sizeg(i) = 0;
        end
    h = (b(2)*b(3)-1);
    H = [0 b(3) b(2)];
    wh(i) = ((b(2)*b(3)-1)^2)/(b(3)^2*b_var(2,2) + 2*b(2)*b(3)*b_var(2,3) +
b(2)^2*b_var(3,3));
    pvalh(i) = 1 - chi2cdf(wh(i),1);
        if pvalh(i) < 0.05
        sizeh(i) = 1;
        else
        sizeh(i) = 0;
        end
   wald(i,:) = [wg(i), wh(i)];
end
empsizeg(k) = mean(sizeg);
empsizeh(k) = mean(sizeh);
wald_size_n = zeros(length(nvec),2);
crit = chi2inv(1-0.05,1);
wald_size(1) = sum(wald(:,1)>crit)/reps;
wald_size(2) = sum(wald(:,2)>crit)/reps;
wald_size_n(n,:) = wald_size;
end

% Plot
figure(1)
q = plot(nvec,empsizeg, 'LineWidth', 1.5);
ylim([0 0.35]);
set(q,'Color',[40/255 163/255 151/255]);
hold on
p = plot(nvec, empsizeh, 'LineWidth', 1.5);
hold on
xlabel('Sample Size');
ylabel('Empirical Size');
title('Empirical Size Under Hg & Hh, Nominal Size = 0.05')
legend('H0: Hg', 'H0: Hh')
set(p,'Color',[245/255 106/255 67/255])
```

```
second = toc;
%% (c) Bootstrap

% Settings
run = 1;

if run == 1;
B = 999;
eps_bs = zeros(n,B);
y_bs = zeros(n,B);
b_bs = zeros(nparam,B);
resid_bs = zeros(n,B);
SE_bs = zeros(nparam,B);
t_bs = zeros(1,B);
t_q = zeros(1,1);
size_bs = zeros(1,2);
for null = 1:2

data = [y x];
b0_0=1;
b1_0=10;
b2_0=0.1;
b_0 = [b0_0 b1_0 b2_0];
Obj = @(b_0)NLSObjHW4(b_0,data);
% for hg/hh nulls
    if null == 1
        con = @(b_0)NLS_Cons_g(b_0);
    else
        con = @(b_0)NLS_Cons_h(b_0);
    end
% fmincon for nls estimates
options = optimoptions('fmincon','Display','off');
b_nls = (fmincon(Obj,b_0,[],[],[],[],[],[],con,options))';
resid_nls = y - x*b_nls;

% generate bs samples
    for s = 1:B
            eps_bs(:,s) = datasample(resid_nls,n);
            y_bs(:,s) = x*b_nls + eps_bs(:,s);

            b_bs(:,s) = x \ y_bs(:,s);
            resid_bs(:,s) = y_bs(:,s) - x*b_bs(:,s);
            b_bs_var = (x'*x \ eye(3))*(x'*((1/(n-
nparam)).*resid_bs(:,s)'*resid_bs(:,s))*x)*(x'*x \ eye(3));
            t_bs(s) = (b_bs(:,s) - b_nls)'*(b_bs_var \ eye(3))*(b_bs(:,s) -
b_nls);
    end

        % sort
        t_bs = sort(t_bs);
        t_q(1,1) = quantile(t_bs,0.95);

        % test original t-statistic from part a using t_quant
        size_bs(1,null) = sum(wald(:,null)>t_q(1,1))/reps;
end
```

```
disp(['Empirical Size = ' num2str(wald_size)]);
disp(['Empirical Size with Bootstrap = ' num2str(size_bs)]);



end
```

4.2 Question 3 NLS Objective File

```
function Obj = NLSObjHW4(b_0, data)
% PURPOSE: compute (the average of) the sum of squared residuals,
%          its first derivatives wrt. theta, hessian, and the asymptotic
%          covariance matrix
%---------------------------------------------------------------------
% USAGE: Obj = NLSobjH4(beta, data)
% where: b_0    = parameters  (1 x 3) [1 b1 b2] is the truth
%        data   = data matrix (n x 4) [y 1 x1 x2]
%---------------------------------------------------------------------
% RETURNS: Obj
%---------------------------------------------------------------------
% Reference:    Gregory, Allan W. and Veall, Michael R. (1985) "Formulating
%               Wald Tests of Nonlinear Restrictions". Econometrica Vol. 53,
%               No. 6(Nov., 1985)
% ---------------------------------------------------------------------
% Written by Robert Ackerman, UNC Chapel Hill.
% November 24, 2013.
% data nx4 [y 1 x_1 x2)
% beta 1 x 3 [1 b1 b2]
% ---------------------------------------------------------------------
Obj = (data(:,1) - data(:,2:end)*b_0')'*(data(:,1) - data(:,2:end)*b_0');
```

4.3 Question 3 H_g Constraint File

```
function [c, ceq] = NLS_Cons_g(b_0)
% PURPOSE: restriction for Hg Null
%---------------------------------------------------------------------------
% Reference:    Gregory, Allan W. and Veall, Michael R. (1985) "Formulating
%               Wald Tests of Nonlinear Restrictions". Econometrica Vol. 53,
%               No. 6(Nov., 1985)
% --------------------------------------------------------------------------
% Written by Robert Ackerman, UNC Chapel Hill.
% November 24, 2013.
% --------------------------------------------------------------------------
c = [];
ceq = b_0(2) - 1/b_0(3) ;
```

4.4 Question 3 H_h Constraint File

```
function [c, ceq] = NLS_Cons_h(b_0)
% PURPOSE: restriction for Hh Null
%-----------------------------------------------------------------------
% Reference:    Gregory, Allan W. and Veall, Michael R. (1985) "Formulating
%               Wald Tests of Nonlinear Restrictions". Econometrica Vol. 53,
%               No. 6(Nov., 1985)
% ----------------------------------------------------------------------
% Written by Robert Ackerman, UNC Chapel Hill.
% November 24, 2013.
% ----------------------------------------------------------------------
c=[];
ceq = b_0(2)*b_0(3) - 1;
```