Empirical Methods in Financial Econometrics:Project 6

Xijie Zhou

October 24, 2019

Exercise 1

A

The function code is below:

Listing 1: ols_beta.m

```
function beta = ols_beta (z,X,Y)
if z ~= 0
    X = [ones(numel(X(:,1)),1) X];
end
beta = inv(X'*X)*X'*Y;
```

When z = 1, it can add a column of 1's to the matrix X. When z = 0, the column will disappear.

B

Mean squared errors for different models when J = 1000			
Stocks	AR(1)	HAR(1)	No Change
DIS	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
PG	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}

Table 1: Mean squared errors for different models when J = 1000

I think AR(1) does best on the MSE criterion, because the MSE of AR(1) is least in all models.

C

Mean squared errors for different models when J = 250			
Stocks	AR(1)	HAR(1)	No Change
DIS	1.7197×10^{-7}	1.9213×10^{-7}	3.5273×10^{-7}
PG	5.9842×10^{-8}	7.1602×10^{-8}	1.1244×10^{-7}

Table 2: Mean squared errors for different models when J = 250

Mean squared errors for different models when J = 500			
Stocks	AR(1)	HAR(1)	No Change
DIS	2.9958×10^{-8}	7.3248×10^{-8}	2.4572×10^{-7}
PG	1.7446×10^{-8}	2.6189×10^{-8}	8.0687×10^{-8}

Table 3: Mean squared errors for different models when J = 500

I think AR(1) is consistently better when evaluated using different window widths, because the MSE of AR(1) is keeping least after using different J's.

D

I think this model wouldn't be a good model out-of-sample. Because at least now, we can find the bigger J is, the least MSE is. As a result, we need to wait so long to get "proper" data, which is wasteful and inefficient.

Exercise 2

A

The function code is below:

Listing 2: y_h.m

```
function y_hat = y_h(N,sigmax2,sigmau2,beta)
x_hat = normrnd(0, sqrt(sigmax2), 1, N);
u_hat = normrnd(0, sqrt(sigmau2), 1, N);
y_hat = x_hat .* beta + u_hat;
end
```

B

The OLS estimator for β is 1.0134.

C

I think it should look like a normal distribution.

D

The figure is below:

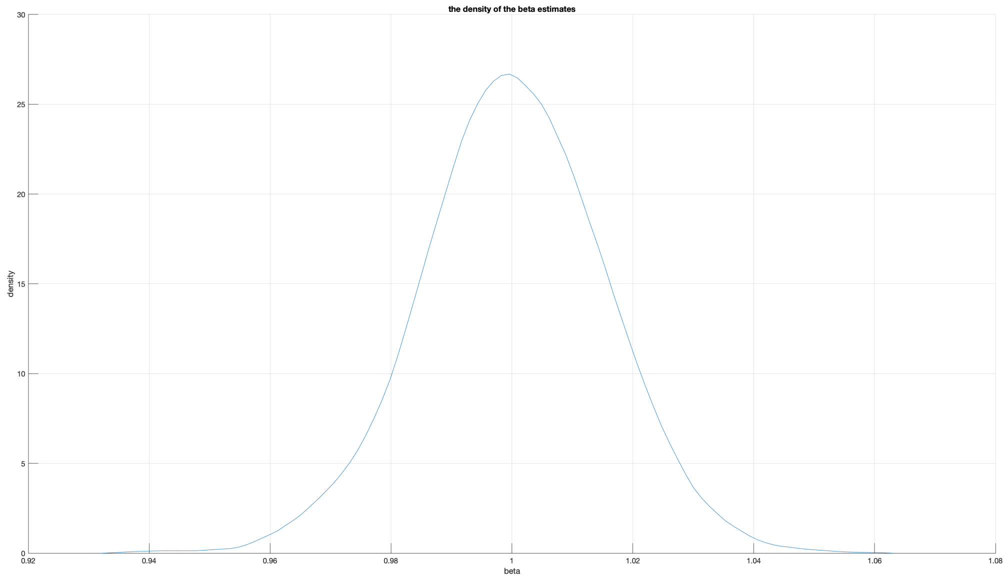

Figure 1: The density of the beta estimates

Interpret: The density of the beta estimates follows a normal distribution. The beta is almost symmetrical with 1 and the width of it is from 0.92 to 1.08. The density of it is from 0 to around 27.

E

The figure is below:

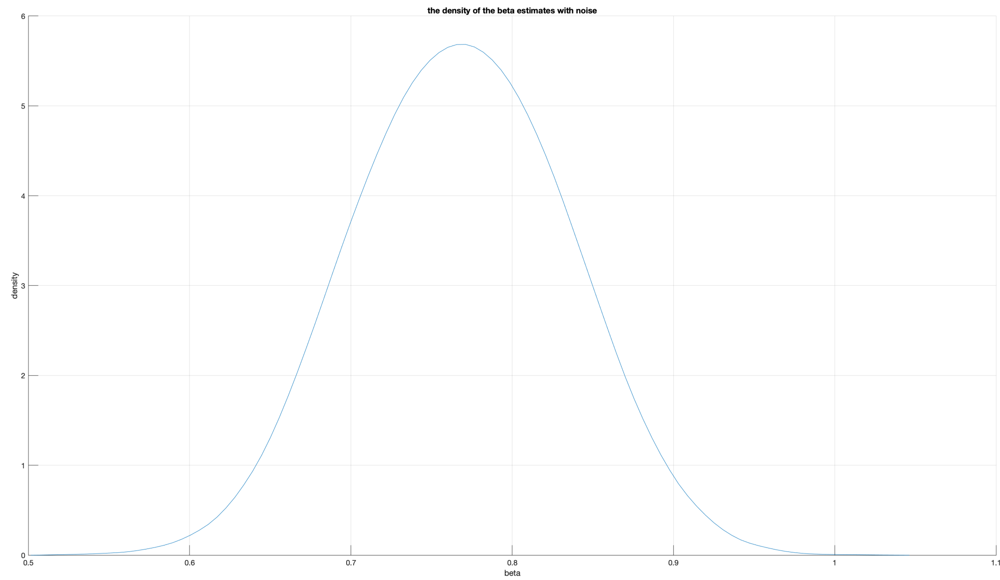

Figure 2: The density of the beta estimates with noise when $\sigma_n^2 = 0.30\sigma_n^2$

Interpret: The density of the beta estimates follows a normal distribution. The beta is almost symmetrical with 0.78 and the width of it is from 0.5 to 1.1. The density of it is from 0 to around 5.8. Compared to the figure 1, it moved to the left and the density of it decreased.

F

The figure is below:

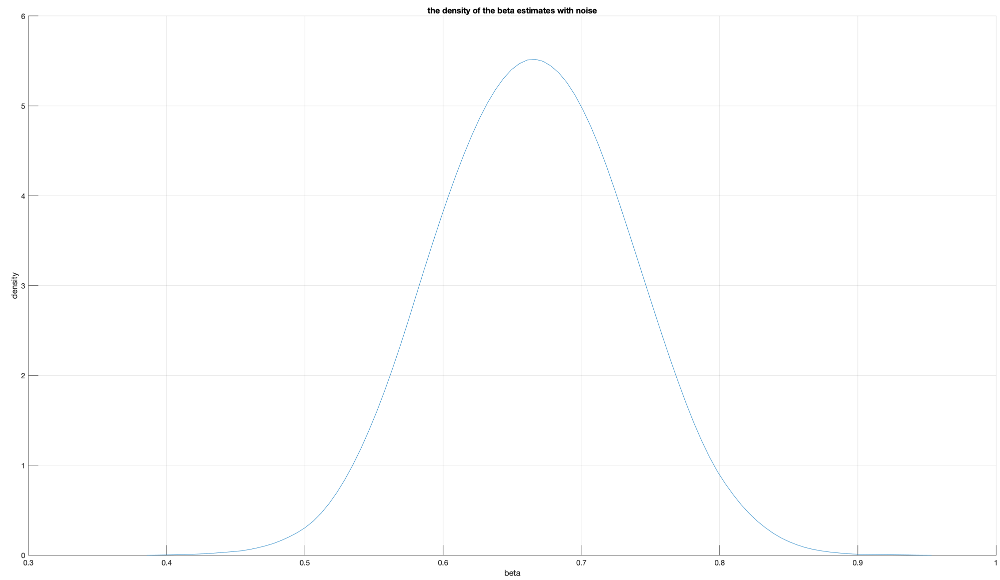

Figure 3: The density of the beta estimates with noise when $\sigma_n^2 = 0.50\sigma_n^2$

Interpret: The density of the beta estimates follows a normal distribution. The beta is almost symmetrical with 0.67 and the width of it is from 0.3 to 1. The density of it is from 0 to around 5.5. Compared to the figure 2, it moved to the left and the density of it decreased.

Based on the three figures above, we can conclude that the smaller J is, the small beta and its density are.

Exercise 3

A

Mean squared errors for different models when J = 1000					
Stocks	ARQ(1)	HARQ(1)	AR(1)	HAR(1)	No Change
DIS	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
PG	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}

Table 4: Mean squared errors for different models when J = 1000

B

Based table 4, for DIS, ARQ(1) has the smallest MSE; for PG, AR(1) has the smallest MSE. However, when we consider the table 5 and table 6 below which changed J, for DIS, AR(1) has the smallest MSE; for PG, AR(1) has the smallest MSE. To conclude, I don't think there is a model that is consistently better for both of my stocks.

Mean squared errors for different models with when J = 250					
Stocks	ARQ(1)	HARQ(1)	AR(1)	HAR(1)	No Change
DIS	3.3352×10^{-6}	3.1005×10^{-6}	1.7197×10^{-7}	1.9213×10^{-7}	3.5273×10^{-7}
PG	1.4489×10^{-5}	1.7150×10^{-5}	5.9842×10^{-8}	7.1602×10^{-8}	1.1244×10^{-7}

Table 5: Mean squared errors for different models when J = 250

Mean squared errors for different models when J = 500					
Stocks	ARQ(1)	HARQ(1)	AR(1)	HAR(1)	No Change
DIS	3.1877×10^{-8}	4.3796×10^{-8}	2.9958×10^{-8}	7.3248×10^{-8}	2.4572×10^{-7}
PG	4.1382×10^{-6}	4.5341×10^{-6}	1.7446×10^{-8}	2.6189×10^{-8}	8.0687×10^{-8}

Table 6: Mean squared errors for different models when J = 500

C

I don't think there is a model that is consistently better. If I must put one of them in practice, I would choose, because

Mean squared errors for different models when J = 1000					
Stocks	ARQ(1)	HARQ(1)	AR(1)	HAR(1)	No Change
AAPL	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
AXP	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
BA	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
BAC	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
BLK	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
C	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
CAT	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
CSCO	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
CVX	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
GNTX	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
GE	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
GOOG	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
GS	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
HD	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
INTC	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
IBM	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
JPM	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
JNJ	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
KO	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
MS	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
MET	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
MSFT	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
MMC	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
MCD	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
MMM	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
MRK	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
NKE	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
PNC	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
PFE	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
STT	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
SPY	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
TSLA	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
UTX	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
UNH	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
VZ	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}
WMT	4.7274×10^{-7}	4.0670×10^{-7}	1.2443×10^{-8}	2.3202×10^{-8}	8.5995×10^{-8}
XOM	1.0494×10^{-8}	2.5557×10^{-8}	1.1389×10^{-8}	6.1813×10^{-8}	2.8138×10^{-7}

Table 7: Mean squared errors for different models when J = 1000

Code

The code is below:

Listing 3: ex_1.m

```
% question A
% create function

```

```
% question B
[dates, prices] = load_stock('DIS.csv');
[dates_return,deltax] = log_returns(dates, prices);
[n,T] = size(deltax);
rv = realized_var(deltax);
J = 1000;

AR1_E = ar_e(rv,J,T);
HAR1_E = har_e(rv,J,T);
NC_E = nc_e(rv,J,T);

ms_AR1_E = mean(AR1_E.^2);
ms_HAR1_E = mean(HAR1_E.^2);
ms_NC_E = mean(NC_E.^2);

% question C
J_1 = 250;
AR1_E_1 = ar_e(rv,J_1,T);
HAR1_E_1 = har_e(rv,J_1,T);
NC_E_1 = nc_e(rv,J_1,T);

ms_AR1_E_1 = mean(AR1_E_1.^2);
ms_HAR1_E_1 = mean(HAR1_E_1.^2);
ms_NC_E_1 = mean(NC_E_1.^2);

J_2 = 500;
AR1_E_2 = ar_e(rv,J_2,T);
HAR1_E_2 = har_e(rv,J_2,T);
NC_E_2 = nc_e(rv,J_2,T);

ms_AR1_E_2 = mean(AR1_E_2.^2);
ms_HAR1_E_2 = mean(HAR1_E_2.^2);
ms_NC_E_2 = mean(NC_E_2.^2);
% question D
```

Listing 4: ex_2.m

```
% question A
N = 100;
sigma_x2 = 25.2;
sigma_u2 = 0.5;
beta = 1;
seed = 100;
rng(seed);
x_hat = normrnd(0, sqrt(sigma_x2), 1, N);
u_hat = normrnd(0, sqrt(sigma_u2), 1, N);
y_hat = x_hat.*beta + u_hat;
% y_hat = y_h(N,sigma_x2,sigma_u2,beta);

% question B
beta_hat = ols_beta (0,x_hat.',y_hat.');

% question C
rep = 1000;
beta_hat_sim = zeros(1,rep);
for i = 1:rep
    seed = i;
    rng(seed)
    x_hat = normrnd(0, sqrt(sigma_x2), 1, N);
    u_hat = normrnd(0, sqrt(sigma_u2), 1, N);
    y_hat = x_hat .* beta + u_hat;

    beta_hat_sim(:,i) = ols_beta (0,x_hat.',y_hat.');
end

% question D
f = figure;
set(f,'units','normalized','outerposition',[0 0 1 1]);
[fb1,xi1] = ksdensity(beta_hat_sim,'kernel','epanechnikov','Bandwidth'
    ,0.005);
plot(xi1, fb1);
title('the density of the beta estimates');
```

```
box off; grid on;
xlabel('beta');
ylabel('density');
print(f,'-dpng','-r200','figures/2D');
close(f);

% question E
sigma_eta2 = 0.3*sigma_x2;
eta = normrnd(0, sqrt(sigma_eta2), 1, N);
x_hat_star = x_hat + eta;

beta_hat_star = ols_beta(0,x_hat_star.' , y_hat.' );
beta_hat_star_sim = zeros(1,rep);

for i = 1:rep
    seed = i;
    rng(seed)
    x_hat = normrnd(0, sqrt(sigma_x2), 1, N);
    u_hat = normrnd(0, sqrt(sigma_u2), 1, N);
    eta = normrnd(0, sqrt(sigma_eta2), 1, N);
    x_hat_star = x_hat + eta;

    y_hat = x_hat .* beta + u_hat;

    beta_hat_star_sim(:,i) = ols_beta (0,x_hat_star.',y_hat.');
end

f = figure;
set(f,'units','normalized','outerposition',[0 0 1 1]);
[fb2,xi2] = ksdensity(beta_hat_star_sim,'kernel','epanechnikov','Bandwidth
    ',0.005);
plot(xi2, fb2);
title('the density of the beta estimates with noise');
box off; grid on;
xlabel('beta');
ylabel('density');
```

```
print(f,'-dpng','-r200','figures/2E');
close(f);

% question F
sigma_eta2 = 0.5*sigma_x2;
eta = normrnd(0, sqrt(sigma_eta2), 1, N);
x_hat_star = x_hat + eta;

beta_hat_star = ols_beta(0,x_hat_star.' , y_hat.' );
beta_hat_star_sim = zeros(1,rep);

for i = 1:rep
    seed = i;
    rng(seed)
    x_hat = normrnd(0, sqrt(sigma_x2), 1, N);
    u_hat = normrnd(0, sqrt(sigma_u2), 1, N);
    eta = normrnd(0, sqrt(sigma_eta2), 1, N);
    x_hat_star = x_hat + eta;

    y_hat = x_hat .* beta + u_hat;

    beta_hat_star_sim(:,i) = ols_beta (0,x_hat_star.',y_hat.');
end

f = figure;
set(f,'units','normalized','outerposition',[0 0 1 1]);
[fb2,xi2] = ksdensity(beta_hat_star_sim,'kernel','epanechnikov','Bandwidth
    ',0.005);
plot(xi2, fb2);
title('the density of the beta estimates with noise');
box off; grid on;
xlabel('beta');
ylabel('density');
print(f,'-dpng','-r200','figures/2F');
close(f);
```

Listing 5: ex_3.m

```
% question A
[dates, prices] = load_stock('DIS.csv');
[dates_return,deltax] = log_returns(dates, prices);
[n,T] = size(deltax);
deltan = 1/n;
rv = realized_var(deltax);

tau = tau_f(deltax);
BV = bipower_var(deltax);
alpha = 5;
cutoff = alpha*deltan^0.49*sqrt(tau*BV);
rc = deltax;
rc(abs(deltax)>cutoff)=0;

QIV = 1/(3*deltan)*sum((rc).^4);

J = 1000;

ARQ1_E = arq_e(rv,QIV,J,T);
HARQ1_E = harq_e(rv,QIV,J,T);

ms_ARQ1_E = mean(ARQ1_E.^2);
ms_HARQ1_E = mean(HARQ1_E.^2);

% question B
J_1 = 250;
ARQ1_E_1 = arq_e(rv,QIV,J_1,T);
HARQ1_E_1 = harq_e(rv,QIV,J_1,T);

ms_ARQ1_E_1 = mean(ARQ1_E_1.^2);
ms_HARQ1_E_1 = mean(HARQ1_E_1.^2);

J_2 = 500;
ARQ1_E_2 = arq_e(rv,QIV,J_2,T);
HARQ1_E_2 = harq_e(rv,QIV,J_2,T);
```

```

ms_ARQ1_E_2 = mean(ARQ1_E_2.^2);
ms_HARQ1_E_2 = mean(HARQ1_E_2.^2);
```