Homework2 Min Xu

1. Did you receive any help whatsoever from anyone in solving this assignment? Yes / No. If you answered yes, give full details: (e.g., \Jane explained to me what is asked in Question 3.4").

No

2. Did you give any help whatsoever to anyone in solving this assignment? Yes / No. If you answered yes, give full details: (e.g., \I pointed Joe to section 2.3 to help him with Question 2").

No

1.1.1

- $E[\omega] = E[(\mathrm{X}^T\mathrm{X})^{-1}X^T y] = (\mathrm{X}^T\mathrm{X})^{-1}X^T E[y] = (\mathrm{X}^T\mathrm{X})^{-1}X^T(X\beta + E[\epsilon]) = (\mathrm{X}^T\mathrm{X})^{-1}(\mathrm{X}^T\mathrm{X})\beta + (\mathrm{X}^T\mathrm{X})^{-1}X^T E[\epsilon]$

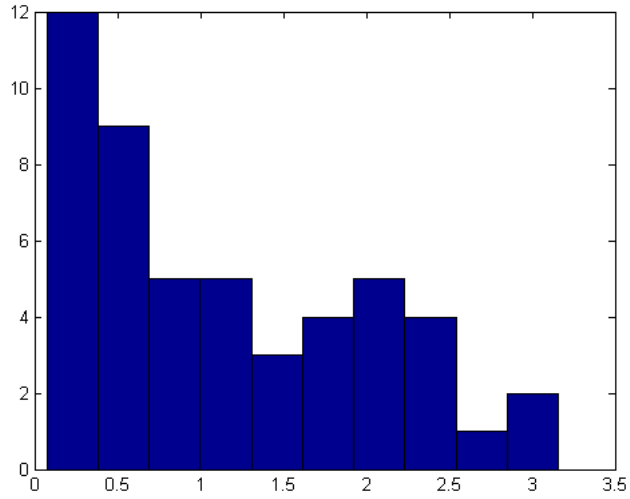
  Since $E[\epsilon] = 0$, $E[\omega] = \beta$

1.1.2

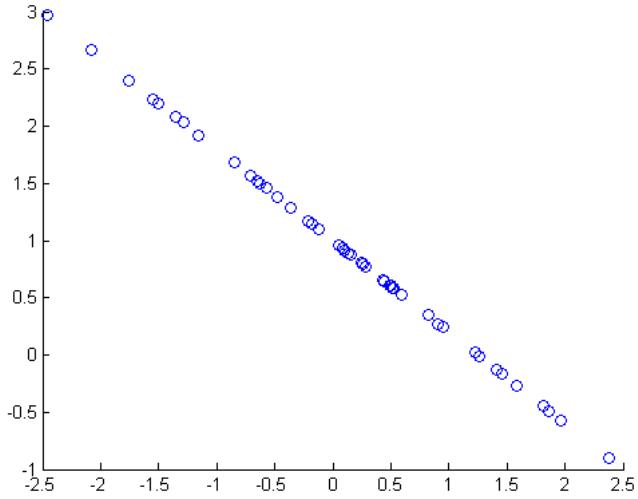
- No, because X won't be full rank and invertible, thus X^TX won't be full rank and invertible. No, the same reason above.

1.2.1

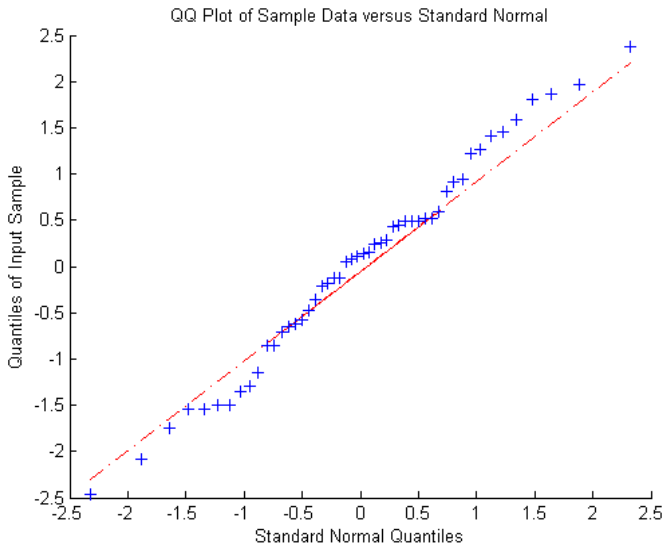
- 22.155
- $\mathrm{E}[\omega] = \begin{bmatrix}0\\1\end{bmatrix}$, $\mathrm{Cov}[\omega] = \begin{bmatrix}1 & -1\\-1 & 1.2\end{bmatrix}$
- empirical mean of $\omega = \begin{bmatrix}0.203\\0.983\end{bmatrix}$, empirical cov of $\omega = \begin{bmatrix}1.260 & -1.008\\-1.008 & 0.806\end{bmatrix}$
- hist() plot:



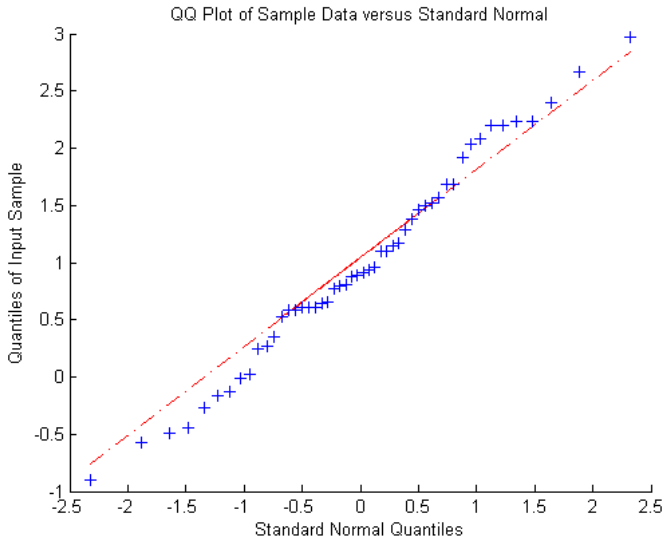
- scatter() plot:



- qqplot() for ω_1

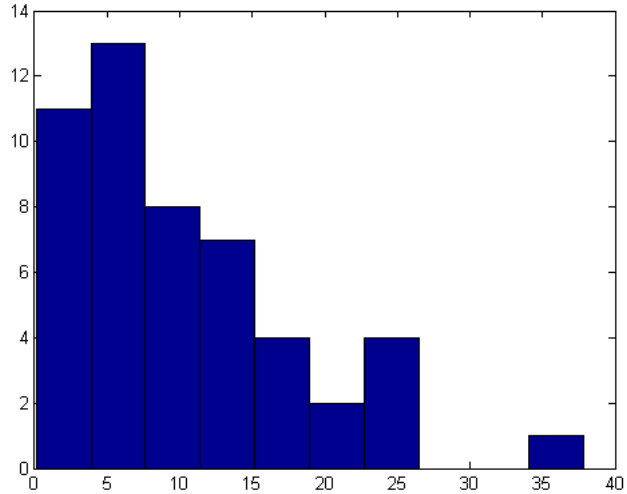


- qqplot() for ω_2

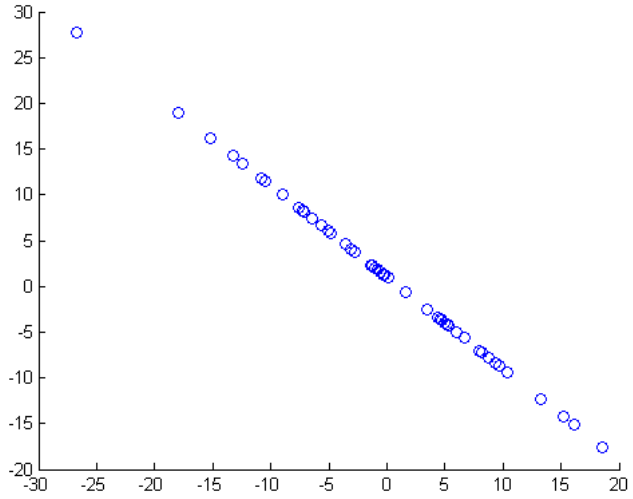


1.2.2

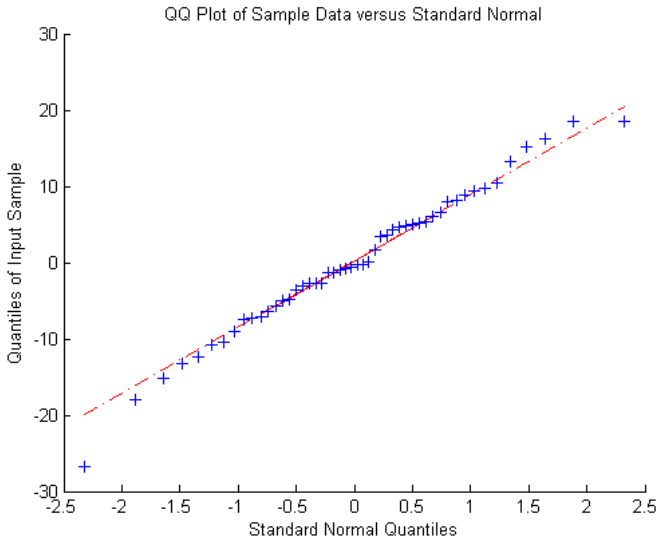
- 7.698e3
- $\mathrm{E}[\omega] = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $\mathrm{Cov}[\omega] = \begin{bmatrix} 320.5 & -310 \\ -310 & 300 \end{bmatrix}$
- empirical mean of $\omega = \begin{bmatrix} 0.130 \\ 0.870 \end{bmatrix}$, empirical cov of $\omega = \begin{bmatrix} 88.692 & -88.692 \\ -88.692 & 88.692 \end{bmatrix}$
- hist() plot:



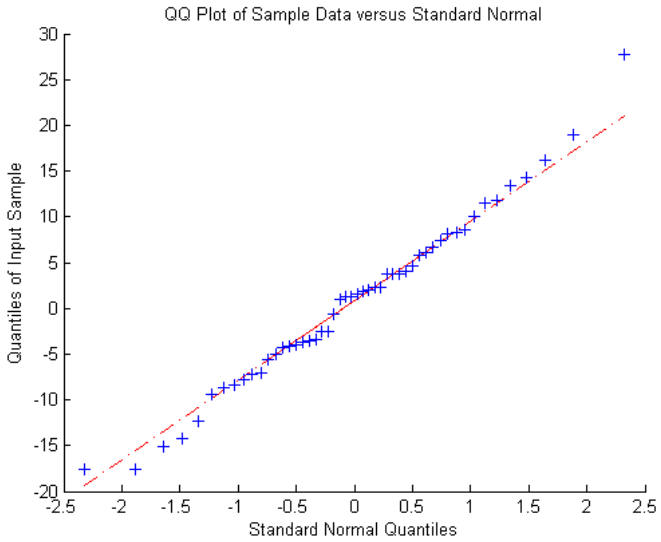
- scatter() plot:



- qqplot() for ω_1

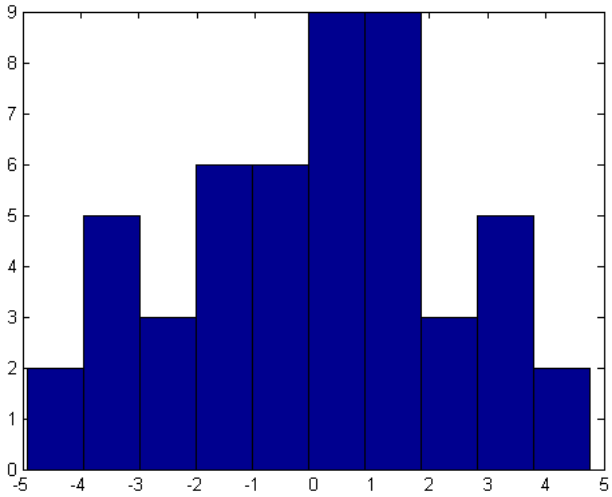


- qqplot() for ω_2

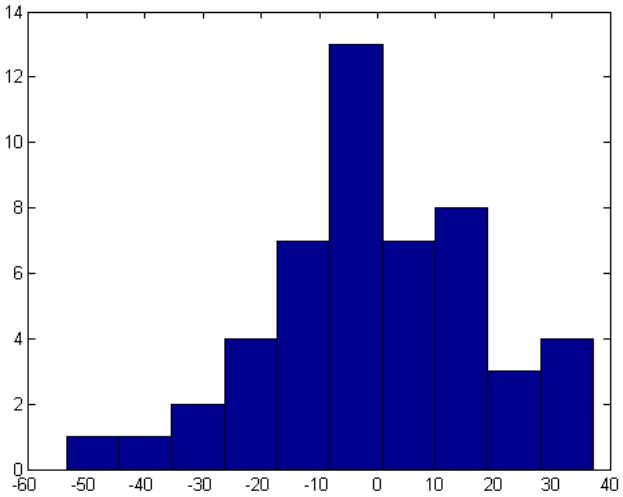


1.2.3

- part 1:



- part 2:



1.2.4

- Yes, since the more samples there are, the closer $\mathrm{E}[\omega]$ and $\mathrm{Cov}[\omega]$ get to empirical estimates
- It is a straight linear line, centered at y = kX, shows that distribution of ω are linearly related to its normal distribution
- Experiment 1, because two features have less collinearity
- No, it will be positively correlated since slope of $\omega_1 - \omega_2$ plot will be positive

2.1

(a) $P(Y = +) = 1/3$, $P(Y = -) = 2/3$,

$H_S(Y) = -P(Y = +)\log_2(P(Y = +)) - P(Y = -)\log_2(P(Y = -)) =$ **0.918**

(b) $P(X_1 = T) = 1/2$, $P(X_1 = F) = 1/2$, $H_S(Y|X_1 = T) = 0.918$, $H_S(Y|X_1 = F) = 0$,

$G(Y, X_1) = H_S(Y) - P(X_1 = T)H_S(Y|X_1 = T) - P(X_1 = F)H_S(Y|X_1 = F) =$ **0.459**

2.2

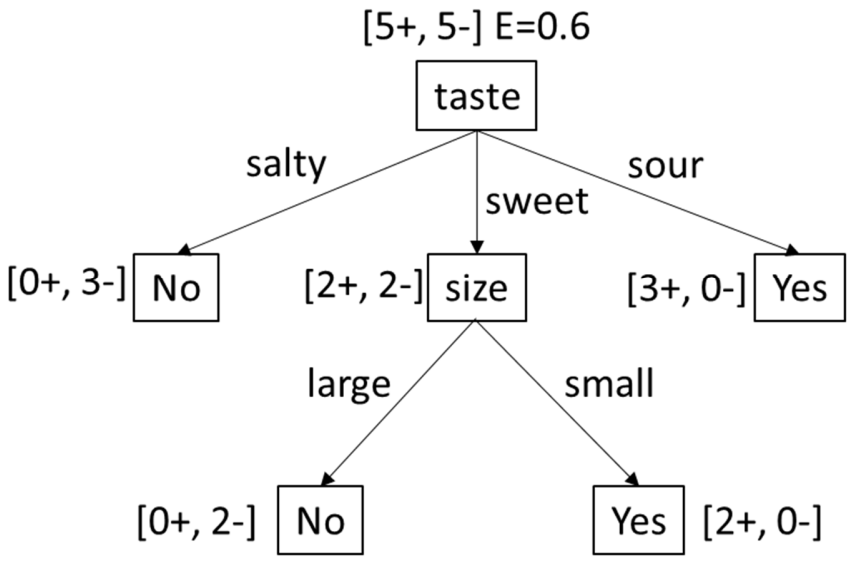
(a) Calculations based on: $G(Y, X) = H(Y) - \sum_v P(X = v)H(Y|X = v)$

G(Y, X = temp) = 0.029

G(Y, X = taste) = 0.6

G(Y, X = size) = 0.278

(b)



2.3

(a) (m-1)/m, the extreme case will be classifying all samples to one label

(b) No, one possibility is that if you switch the root from one feature to the other (e.g., in 2.2(b), switch taste and size), the number of nodes of the decision tree may change, but the classification result remains the same

(c) Yes, as long as we have a large enough decision tree, the classification boundary can be close to a non-linear boundary, thus can perfectly classify any linearly separable dataset

3.1.1

- $\mathrm{R} \leq \sqrt{100} = 10, k \leq m = \left(\frac{R}{\gamma}\right)^2 = 100$, so upper bound on R is 10, max number of mistakes is 100

3.1.2

- No, for XOR function, it requires: $1 \times w_0 + 0 \times w_1 > 1$, $0 \times w_0 + 1 \times w_1 > 1$, where we have $w_1 > 1$ and $w_2 > 1$. But it also requires $1 \times w_0 + 1 \times w_1 < -1$, which cannot be possible

3.2.1 $h_1 = 0.6225$

3.2.2 $h_2 = 0.5$

3.2.3 $o_1 = 0.3778$

3.2.4 $o_2 = 0.628$

3.2.5 $\delta_{O1} = 0.1463$

3.2.6 $\delta_{O2} = 0.0869$

3.2.7 $\delta_{h1} = -0.0015$

3.2.8 $w_{h1,O1} = -0.3909$

3.2.9 $w_{h1,O2} = 0.6054$

3.2.10 $w_{i1,h1} = 0.4999$

Matlab Code

1. 2

```
%X = [-1 -1.5; 0 0.5; 1 0.5; 2 1.5] %4 by 2
X = [-1 -1; 0 0; 1 1; 2 2.1] %4 by 2
beta = [0 1]' %2 by 1
condNum = cond(X' * X)
covw = inv(X' * X)
datasetNumber = 50;
sampleNumber = 4;
eps = normrnd(0, 1, datasetNumber ,1);
eps = repmat(eps, 1, sampleNumber)'; %4 by 50
y = repmat(X * beta, 1, datasetNumber)  + eps; %4 by 50
w = inv(X' * X) * X' * y; %2 by 50
w = w';
beta = repmat(beta, 1, datasetNumber)'; %50 by 2
for i = 1:datasetNumber
    error(i) = norm(w(i, :) - beta(i, :));
end
mean(w)
cov(w)
figure(1)
hist(error, 10);
figure(2)
scatter(w(:,1), w(:,2));
figure(3)
qqplot(w(:,1));
figure(4)
qqplot(w(:,2));

Xtest = [2 0];
ytest = Xtest * w';
figure(5)
hist(ytest, 10)
```

2.2

```
H_Y = -1/2*log2(1/2) - 1/2*log2(1/2);
P_temp_hot = 1/2;
P_temp_cold = 1/2;
P_taste_salty = 3/10;
P_taste_sweet = 2/5;
P_taste_sour = 3/10;
P_size_large = 1/2;
P_size_small = 1/2;

H_Y_given_hot = -2/5*log2(2/5)-3/5*log2(3/5);
H_Y_given_cold = -2/5*log2(2/5)-3/5*log2(3/5);
H_Y_temp = H_Y - P_temp_hot * H_Y_given_hot - ...
           P_temp_cold * H_Y_given_cold

H_Y_given_salty = 0;
H_Y_given_sweet = 1;
H_Y_given_sour = 0;
H_Y_taste = H_Y - P_taste_salty * H_Y_given_salty - ...
           P_taste_sweet * H_Y_given_sweet - ...
           P_taste_sour * H_Y_given_sour
```

```
H_Y_given_large = -1/5*log2(1/5) - 4/5*log2(4/5);
H_Y_given_small = -1/5*log2(1/5) - 4/5*log2(4/5);
H_Y_size = H_Y - P_size_large * H_Y_given_large - ...
            P_size_small * H_Y_given_small
```

3.2

```
i = [1; 0];
wih = [0.5; -0.5];
h = 1 ./ (1 + exp(-i .* wih))
who = [-0.4 -0.5; 0.6 0.3];
o = 1./ (1+ exp(-who * h))
t = [1; 1];
deltao = o .* (1 - o) .* (t - o)
deltah = h .* (1 - h) .* (who' * deltao)
ita = 0.1;
who = who + ita * deltao * h'
wih = wih + ita * deltah .* i
```