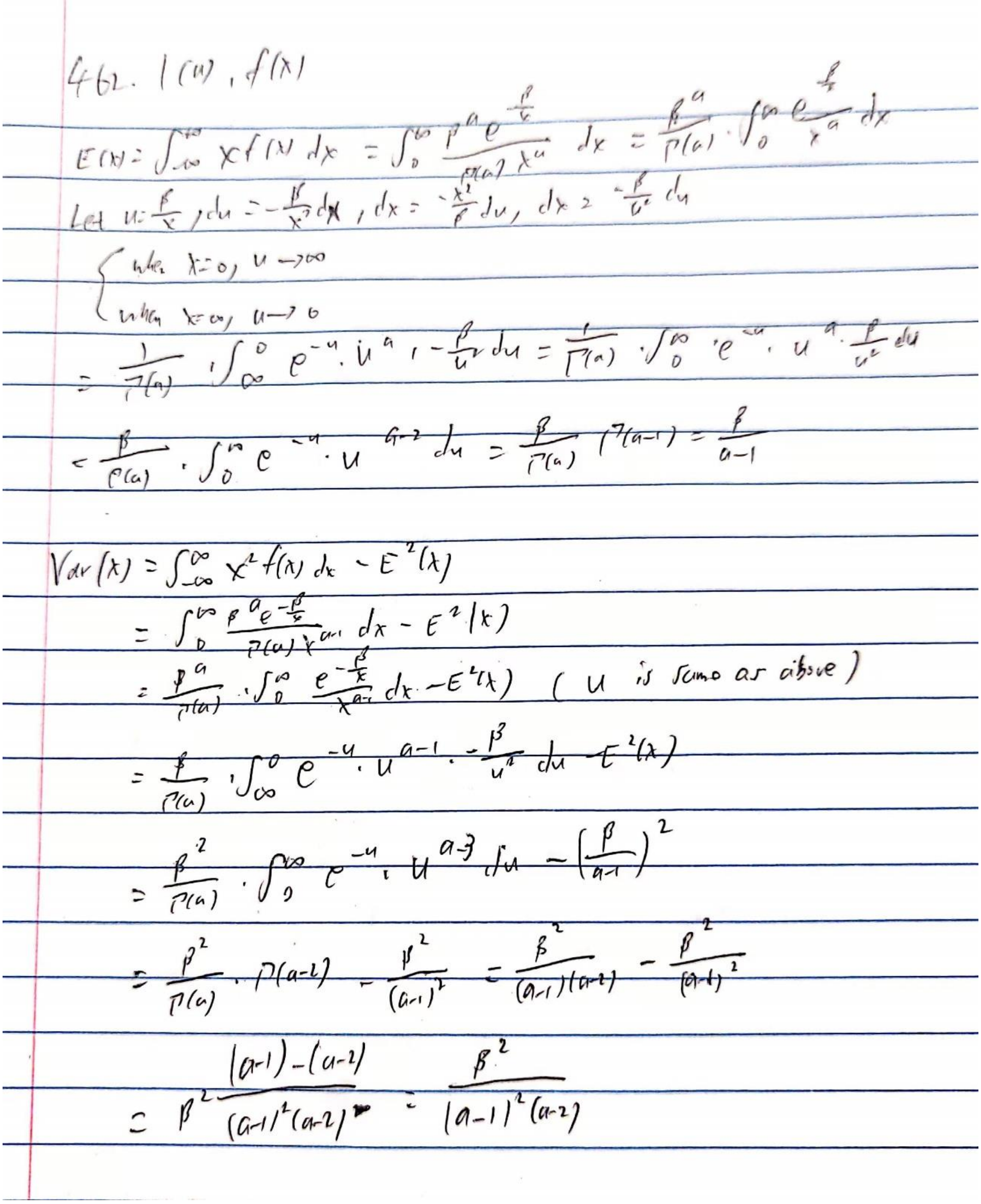
462 1(a)



462. 1(a), f(x)

$E(x) = \int_{-\infty}^{\infty} x f(x)\, dx = \int_0^{\infty} \frac{\beta^a e^{-\frac{\beta}{x}}}{\Gamma(a) x^a} dx = \frac{\beta^a}{\Gamma(a)} \cdot \int_0^{\infty} \frac{e^{\frac{\beta}{x}}}{x^a} dx$

Let $u = \frac{\beta}{x}$, $du = -\frac{\beta}{x^2} dx$, $dx = -\frac{x^2}{\beta} du$, $dx = -\frac{\beta}{u^2} du$

$\begin{cases} \text{when } x = 0,\ u \to \infty \\ \text{when } x = \infty,\ u \to 0 \end{cases}$

$= \frac{1}{\Gamma(a)} \cdot \int_{\infty}^{0} e^{-u} \cdot u^a \cdot -\frac{\beta}{u^2} du = \frac{1}{\Gamma(a)} \cdot \int_0^{\infty} e^{-u} \cdot u^a \cdot \frac{\beta}{u^2} du$

$= \frac{\beta}{\Gamma(a)} \cdot \int_0^{\infty} e^{-u} \cdot u^{a-2} du = \frac{\beta}{\Gamma(a)} \Gamma(a-1) = \frac{\beta}{a-1}$

$Var(x) = \int_{-\infty}^{\infty} x^2 f(x)\, dx - E^2(x)$

$= \int_0^{\infty} \frac{\beta^a e^{-\frac{\beta}{x}}}{\Gamma(a) x^{a-1}} dx - E^2(x)$

$= \frac{\beta^a}{\Gamma(a)} \cdot \int_0^{\infty} \frac{e^{-\frac{\beta}{x}}}{x^{a-2}} dx - E^2(x)$ (u is same as above)

$= \frac{\beta}{\Gamma(a)} \cdot \int_{\infty}^{0} e^{-u} \cdot u^{a-1} \cdot -\frac{\beta^3}{u^2} du - E^2(x)$

$= \frac{\beta^2}{\Gamma(a)} \cdot \int_0^{\infty} e^{-u} \cdot u^{a-3} du - \left(\frac{\beta}{a-1}\right)^2$

$= \frac{\beta^2}{\Gamma(a)} \cdot \Gamma(a-2) - \frac{\beta^2}{(a-1)^2} = \frac{\beta^2}{(a-1)(a-2)} - \frac{\beta^2}{(a-1)^2}$

$= \beta^2 \frac{(a-1)-(a-2)}{(a-1)^2(a-2)} = \frac{\beta^2}{(a-1)^2(a-2)}$

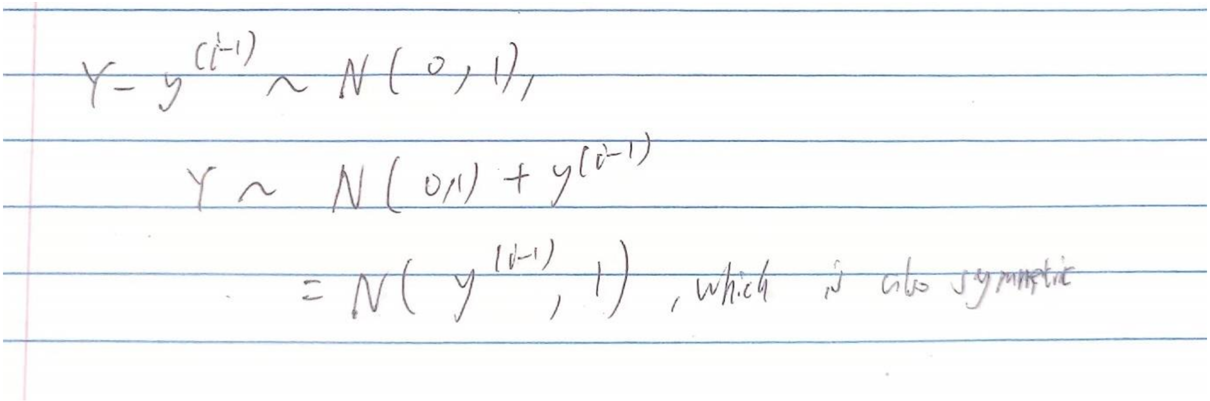
For the given distribution, above steps shown how to find mean and variance.

How to apply MH?

I apply the given distribution in random walk MH and independence MH. For random walk MH, I choose to use N(0,1) and unif(-1,1) as posterior distribution. For independence MH, I choose to use logNormal(beta/(alpha-1), (beta^2)/((alpha-1)^2*(alpha-2))) as posterior distribution. Choose the mean of distribution as the starting point.

Random walk

For N(0,1), normal distribution’s pdf is symmetric around its mean.

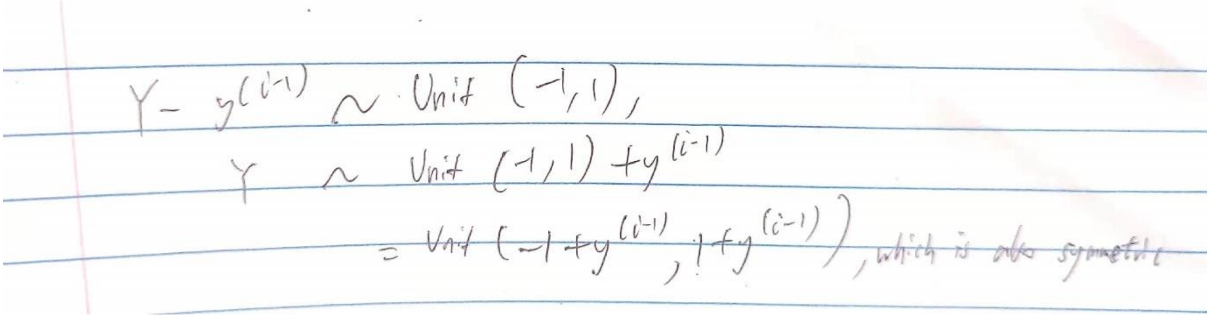


$Y - y^{(i-1)} \sim N(0,1),$

$Y \sim N(0,1) + y^{(i-1)}$

$= N(y^{(i-1)}, 1)$, which is also symmetric

For unif(-1,1), with an positive number k, unif(-k,k)’s pdf is symmetric function,



$Y - y^{(i-1)} \sim \text{Unif}(-1,1),$

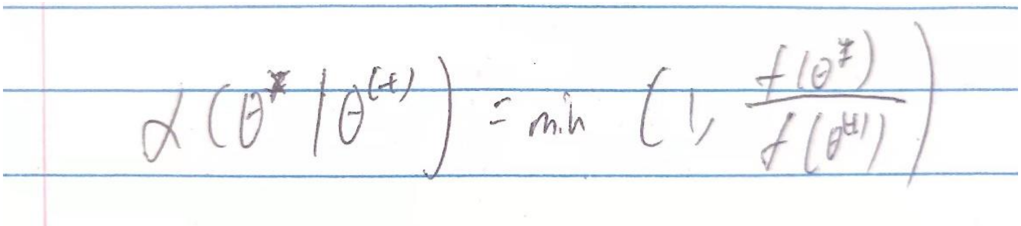
$Y \sim \text{Unif}(-1,1) + y^{(i-1)}$

$= \text{Unif}(-1+y^{(i-1)}, 1+y^{(i-1)})$, which is also symmetric

Then, define a function with number of iterations = n, alpha and beta from given distribution.

At each iteration t in 1:n, do the steps:

1. Generate random y from proposal distribution that written above.
2. Generate probability for moving forward or not. The probability function is



$$\alpha(\theta^* | \theta^{(t)}) = \min\left(1, \frac{f(\theta^*)}{f(\theta^{(t)})}\right)$$

3. Generate random number between 0 and 1. If the random number is >= probability, then stays the same. If the random number is < probability, then moving forward.

Independence

$$X \sim \text{Lognormal}(\mu, \sigma^2) = \ln(x) \sim N(\mu, \sigma^2)$$

(formula for explaning lognormal distriubution)

For independence MH, the posterior distribution is logNormal(beta/(alpha-1), (beta^2)/((alpha-1)^2*(alpha-2))), I choose to use the selected mean and variance for lognormal because in given distribution, E(x) and var(x) are shown below:

$$E(X) = \frac{\beta}{\alpha-1}, \quad Var(X) = \frac{\beta^2}{(\alpha-1)^2(\alpha-2)}$$

, which is the same mean and variance as posterior distribution. Because of same mean and variance between proposal distribution and given distribution, the proposal distribution is independent with the past.

Then, define a function with number of iterations = n, alpha and beta from given distribution.

At each iteration t in 1:n, do the steps:

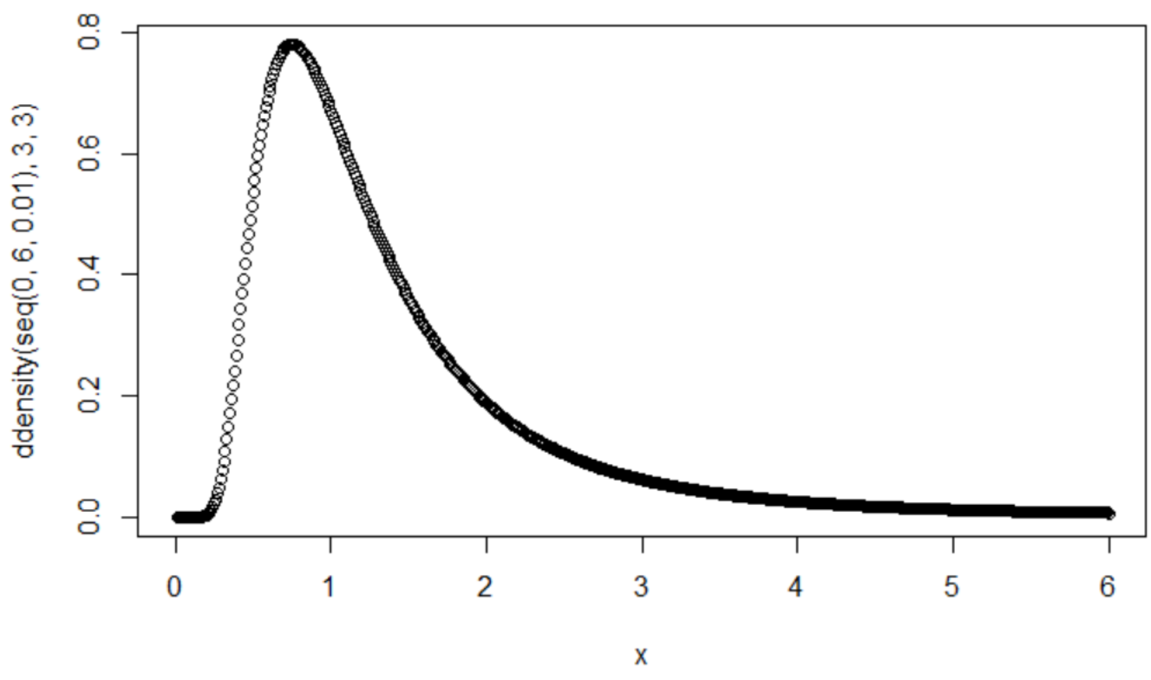
1. Generate random y from proposal distribution that written above.
2. Generate probability for moving forward or not. The probability function is

$$\alpha(y|y^{(i-1)}) = \min\left(1, \frac{f(y)\, q(y)^{(i-1)}}{f(y^{(i-1)})\, q(y)}\right)$$

3. Generate random number between 0 and 1. If the random number is >= probability, then stays the same. If the random number is < probability, then moving forward.

Discuss Result

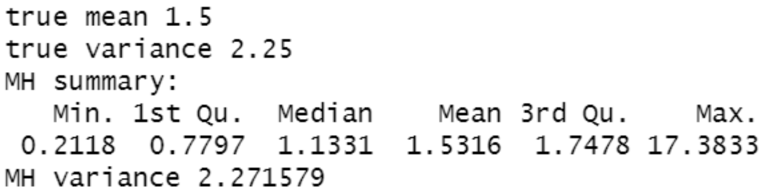
To discuss the result, I plot each step of MH by trace plot to see if it’s a good mixing or not, and histogram to test its distribution. I’m generate the sample for distribution function with given density function, alpha = 3, beta = 3. Choose n = 10000 for number of iterations.



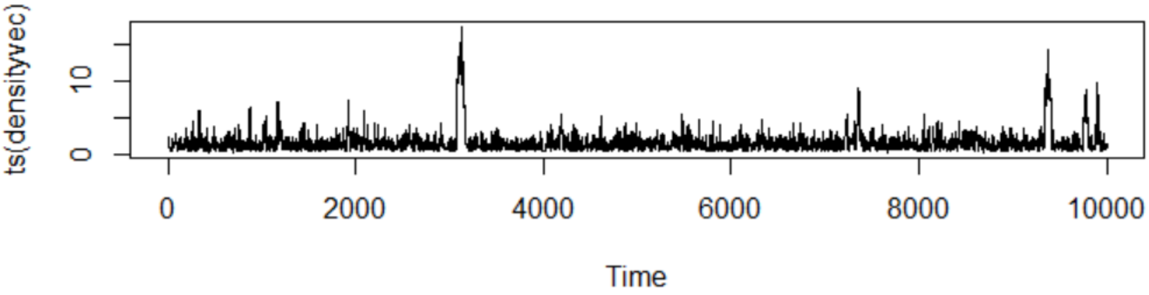
(The given distribution function plot, when alpha = beta = 3)

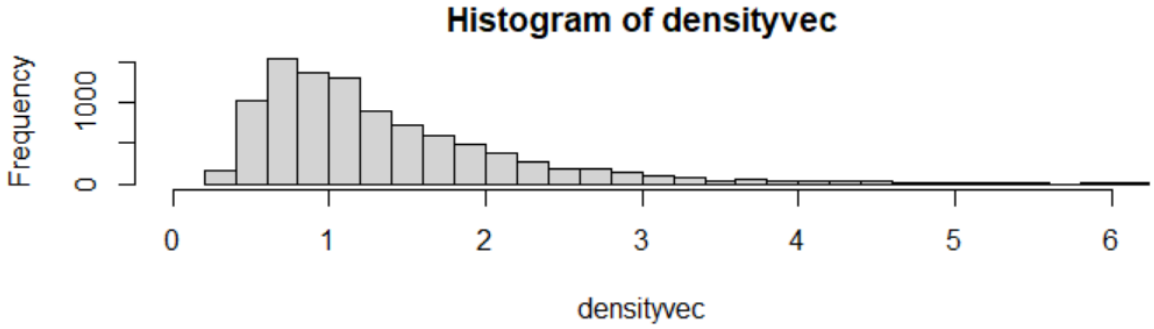
For random walk, N(0,1) as posterior distribution

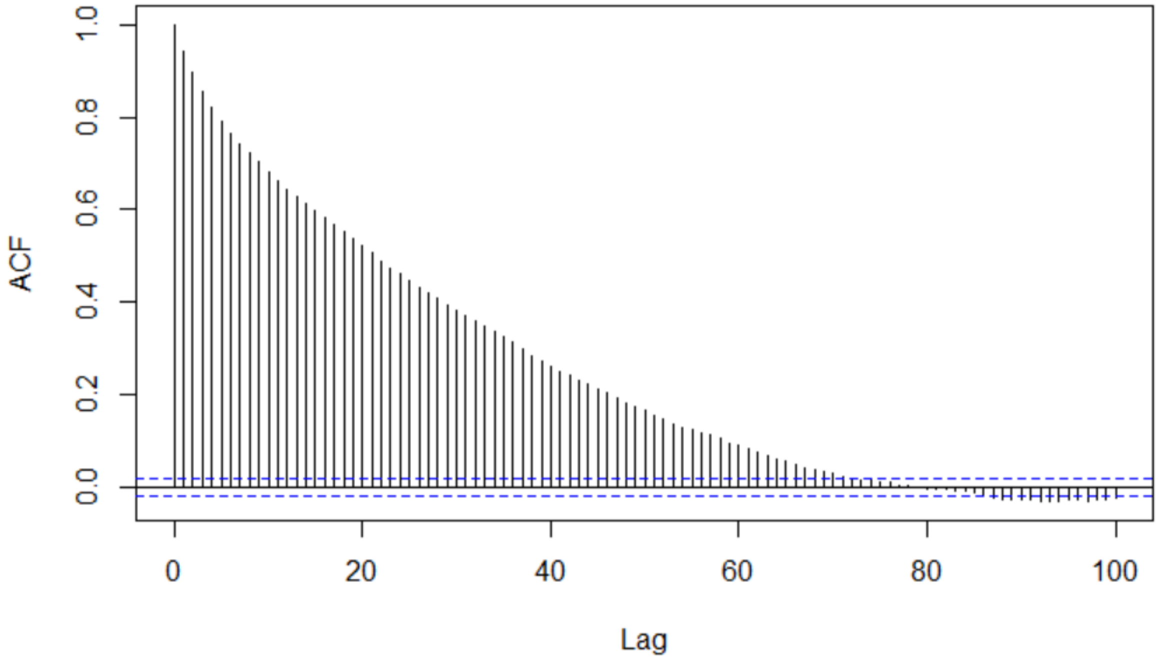
Acceptance rate: 0.545



```
true mean 1.5
true variance 2.25
MH summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.2118  0.7797  1.1331  1.5316  1.7478 17.3833 
MH variance 2.271579
```







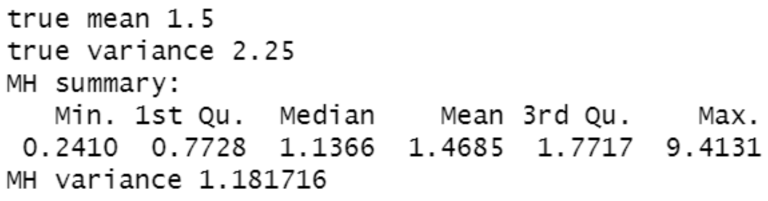
For MCMC output analysis, the acceptance rate is not low(which is > 0.5), sample mean(1.53) and variance(2.27) for this chain are close to true mean(1.5) and variance(2.25).

From the trace plot, we need to decide number of iterations until its convergence, depend on initial value. We may try to use brun-in method to ignore iteration from 1 to 500. From the histogram, it's similar as the true distribution curve.

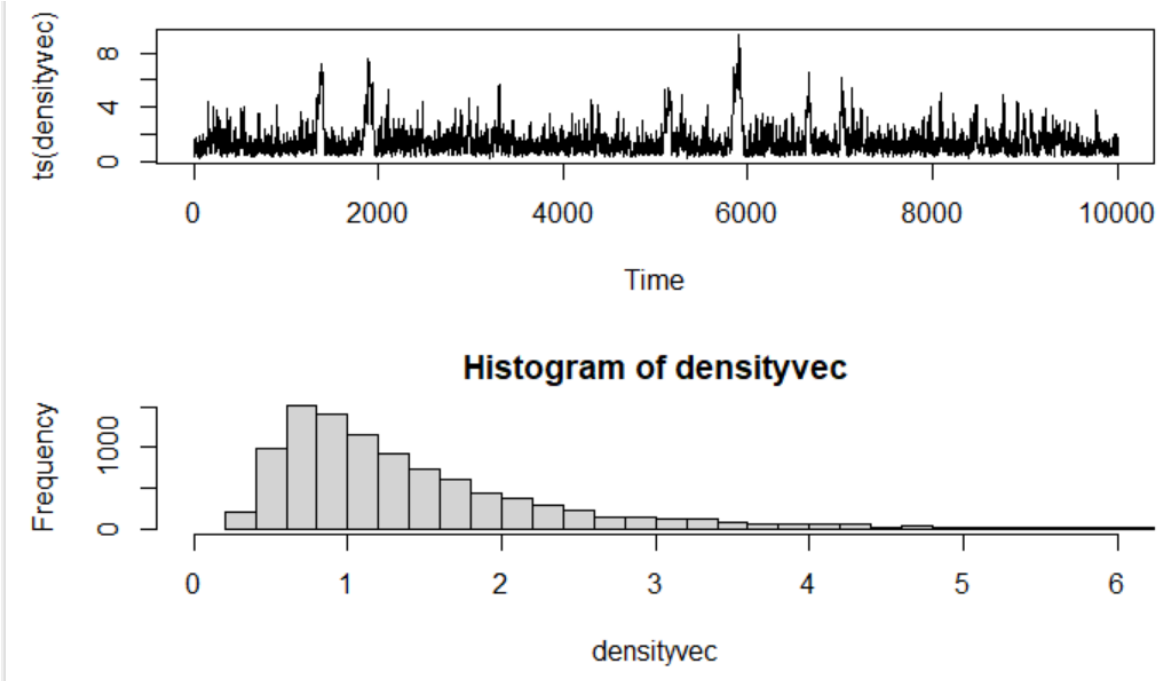
From the auto-correlation graph, it is measuring the degree of dependence between successive draws in the chain. We can see if lag increases, correlation decreases. To make the correlation lower in the beginning part, I can choose to use brun-in to ignore first 70 simulations.

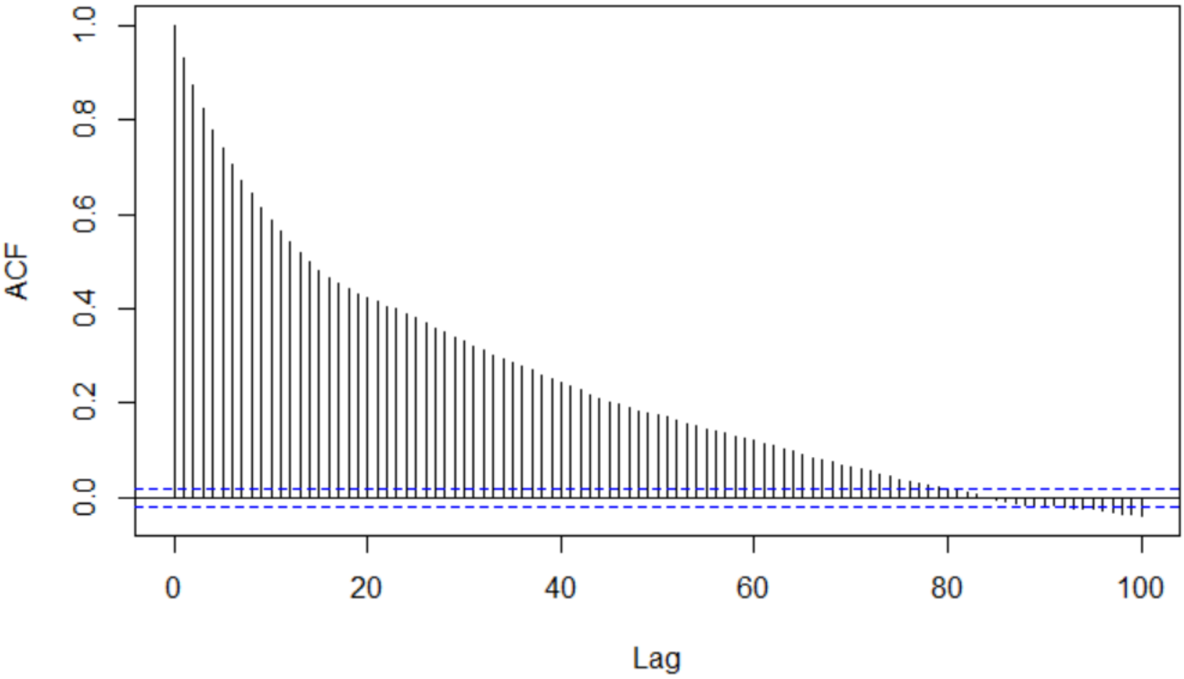
For random walk, unif(-1,1) as posterior distribution

Acceptance rate: 0.661



```
true mean 1.5
true variance 2.25
MH summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.2410  0.7728  1.1366  1.4685  1.7717  9.4131 
MH variance 1.181716
```





For MCMC output analysis, the acceptance rate is not low (which is > 0.5), sample mean(1.47) close to true mean(1.5), but variance(1.18) for this chain is smaller than true variance(2.25).

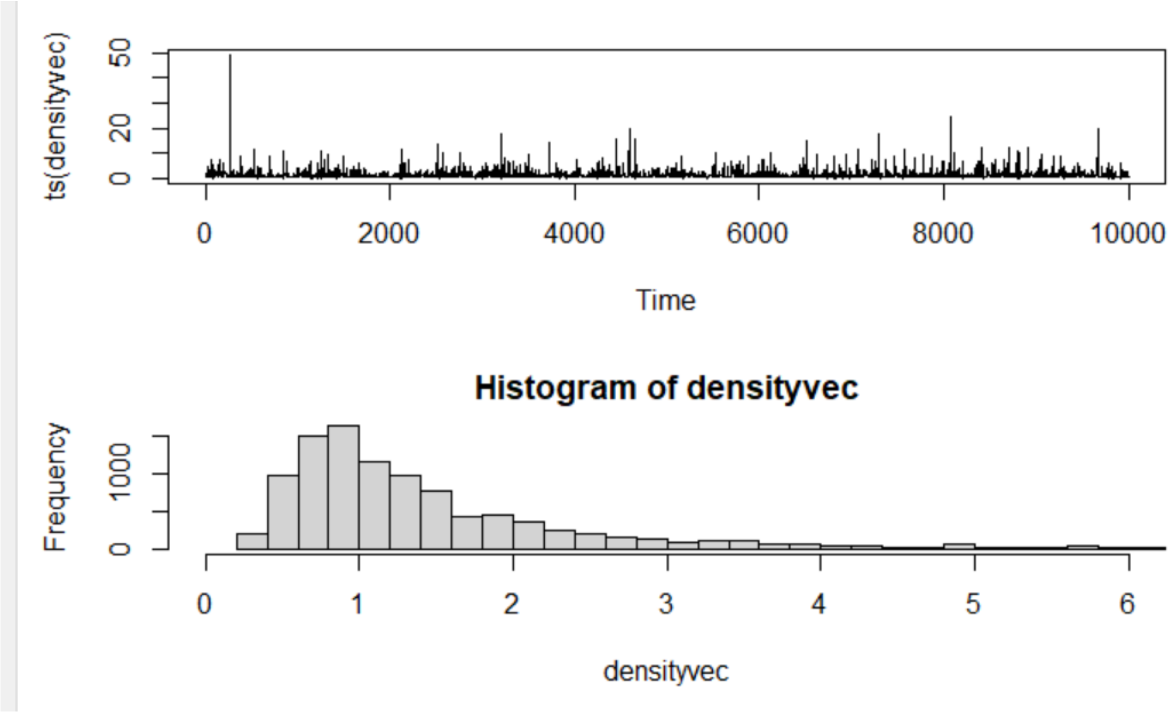
From the trace plot, we may try to use brun-in method to ignore iteration from 1 to 200. From the histogram, it's similar as the true distribution curve.

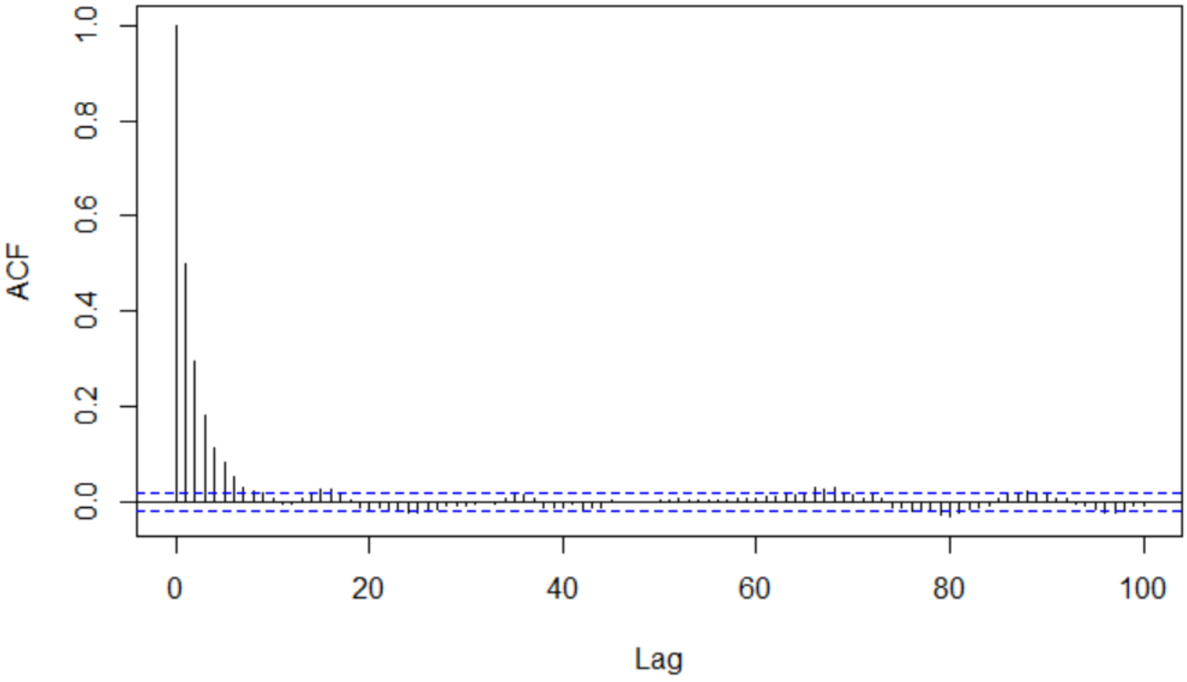
From the auto-correlation graph, it is measuring the degree of dependence between successive draws in the chain. We can see if lag increases, correlation decreases. To make the correlation lower in the beginning part, I can choose to use brun-in to ignore first 80 simulations.

For independence, logNormal(beta/(alpha-1), (beta^2)/((alpha-1)^2*(alpha-2))) = lognormal(1.5,2.25) as posterior distribution, since alpha = beta = 3.

Acceptance rate: 0.66

```
true mean 1.5
true variance 2.25
MH summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.2043  0.7763  1.1014  1.5011  1.7316 49.1975 
MH variance 2.074239
```



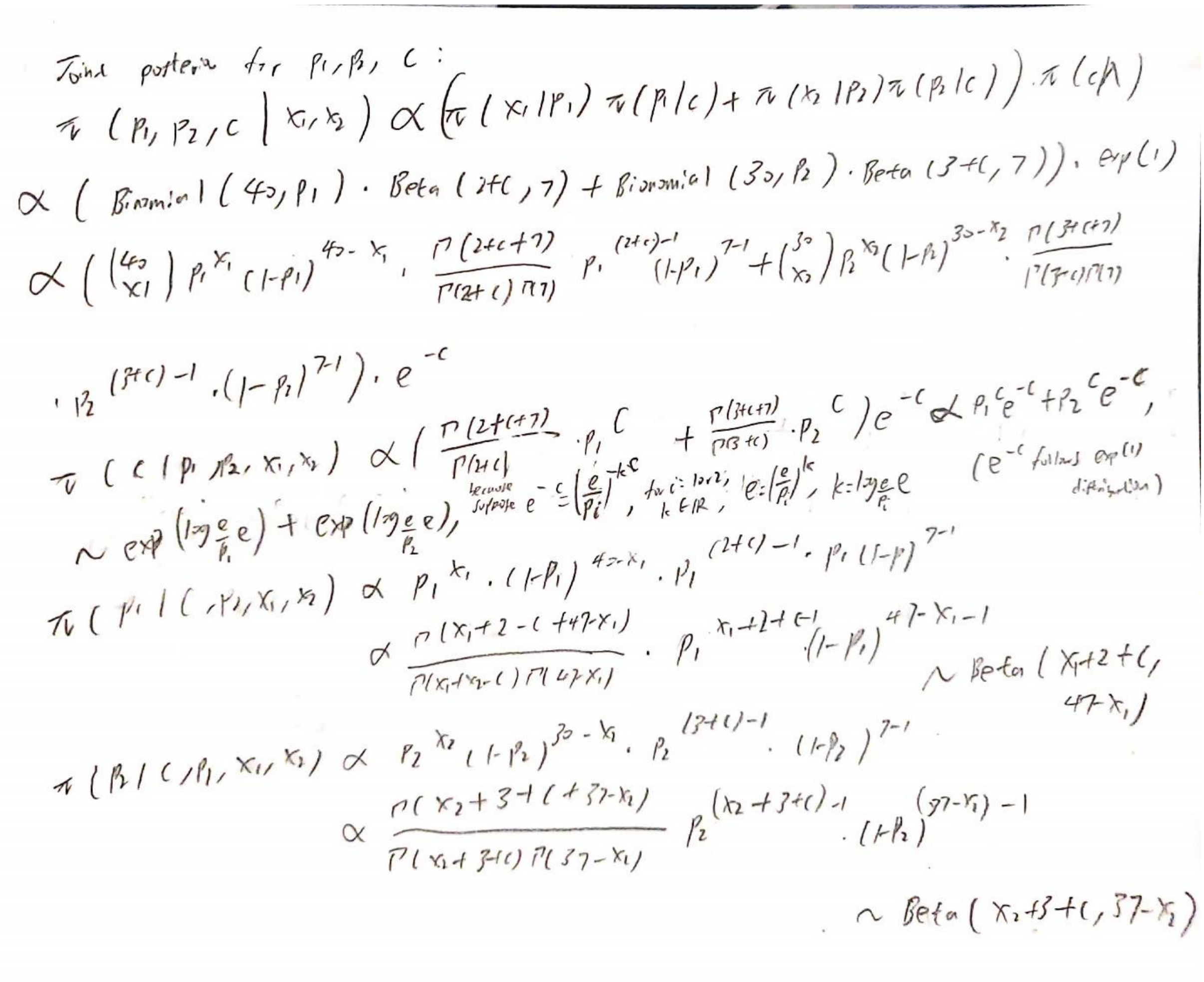


For MCMC output analysis, the acceptance rate is not low (which is > 0.5), sample mean(1.50) close to true mean(1.5), sample variance(2.07) is close to true variance(2.25).

From the trace plot, we need to decide number of iterations until its convergence, depend on initial value. We may try to use brun-in method to ignore iteration from 1 to 2000. From the histogram, it's similar as the true distribution curve.

From the auto-correlation graph, it is measuring the degree of dependence between successive draws in the chain. We can see if lag increases, correlation decreases. To make the correlation lower in the beginning part, I can choose to use brun-in to ignore first 7 simulations.

1(b)



Joint posteria for p_1, p_2, c:

$\pi(p_1, p_2, c \mid x_1, x_2) \propto \left(\pi(x_1 \mid p_1)\,\pi(p_1 \mid c) + \pi(x_2 \mid p_2)\,\pi(p_2 \mid c)\right) \cdot \pi(c \mid \lambda)$

$\propto \left(\text{Binomial}(40, p_1) \cdot \text{Beta}(2+c, 7) + \text{Binomial}(30, p_2) \cdot \text{Beta}(3+c, 7)\right) \cdot \exp(1)$

$\propto \left(\binom{40}{x_1} p_1^{x_1} (1-p_1)^{40-x_1} \cdot \frac{\Gamma(2+c+7)}{\Gamma(2+c)\Gamma(7)} p_1^{(2+c)-1} (1-p_1)^{7-1} + \binom{30}{x_2} p_2^{x_2} (1-p_2)^{30-x_2} \cdot \frac{\Gamma(3+c+7)}{\Gamma(3+c)\Gamma(7)}\right.$

$\left. \cdot\, p_2^{(3+c)-1} \cdot (1-p_2)^{7-1}\right) \cdot e^{-c}$

$\pi(c \mid p_1, p_2, x_1, x_2) \propto \left(\frac{\Gamma(2+c+7)}{\Gamma(2+c)} \cdot p_1^{c} + \frac{\Gamma(3+c+7)}{\Gamma(3+c)} \cdot p_2^{c}\right) e^{-c} \propto p_1^{c} e^{-c} + p_2^{c} e^{-c},$ (e^{-c} follows exp(1) distribution)

because suppose $e^{-c} = \left(\frac{e}{p_i}\right)^{-kc}$, for $i = 1$ or 2, $k \in \mathbb{R}$, $e = \left(\frac{e}{p_i}\right)^{k}$, $k = \log_{\frac{e}{p_i}} e$

$\sim \exp(\log_{\frac{e}{p_1}} e) + \exp(\log_{\frac{e}{p_2}} e),$

$\pi(p_1 \mid c, p_2, x_1, x_2) \propto p_1^{x_1} \cdot (1-p_1)^{40-x_1} \cdot p_1^{(2+c)-1} \cdot p_1 (1-p)^{7-1}$

$\propto \frac{\Gamma(x_1+2-c+47-x_1)}{\Gamma(x_1+x_2-c)\Gamma(47-x_1)} \cdot p_1^{x_1+2+c-1} (1-p_1)^{47-x_1-1}$

$\sim \text{Beta}(x_1+2+c, 47-x_1)$

$\pi(p_2 \mid c, p_1, x_1, x_2) \propto p_2^{x_2} (1-p_2)^{30-x_2} \cdot p_2^{(3+c)-1} \cdot (1-p_2)^{7-1}$

$\propto \frac{\Gamma(x_2+3+c+37-x_2)}{\Gamma(x_2+3+c)\Gamma(37-x_2)} p_2^{(x_2+3+c)-1} \cdot (1-p_2)^{(37-x_2)-1}$

$\sim \text{Beta}(x_2+3+c, 37-x_2)$

By the conditional distribution formulas above, apply gibbs sampling:

1. Choose initial value of c. I choose c = 1
2. Based on initial value c, draw new p1 and p2 from its conditional distribution
3. Based on p1 and p2 in step(2), draw c from its conditional distribution
4. Brun-in and Thinning
5. Repeat the steps 2-4 until obtain 10000 simulations.

Then, discuss the result. Posterior means for c is 4.367, for p1 is 0.286, for p2 is 0.343. Posterior standard deviation for c is 3.144, p1 is 0.073, p2 is 0.082. The true p1 should be 9/30 = 0.3, true p2 should be 0.2 since p is probability of success in binomial distribution. From gibbs sampling solution, p1's mean is close to 0.3, but p2's mean does not. The variance is not very large for both p1 and p2, and gibbs sampling for c has a relatively larger standard deviation.

Gibbs sampling works poor if distribution has strong correlated component, so sometimes it's difficult to get the best distribution for gibbs sampling. (*A.M. Johansen, 2010*) It's hard to determine if the sampling distribution in this question is the best one. By above discussion with basic properties of the solution, it show the mean and variance from the estimation of c, p1 and p2.

2(a)

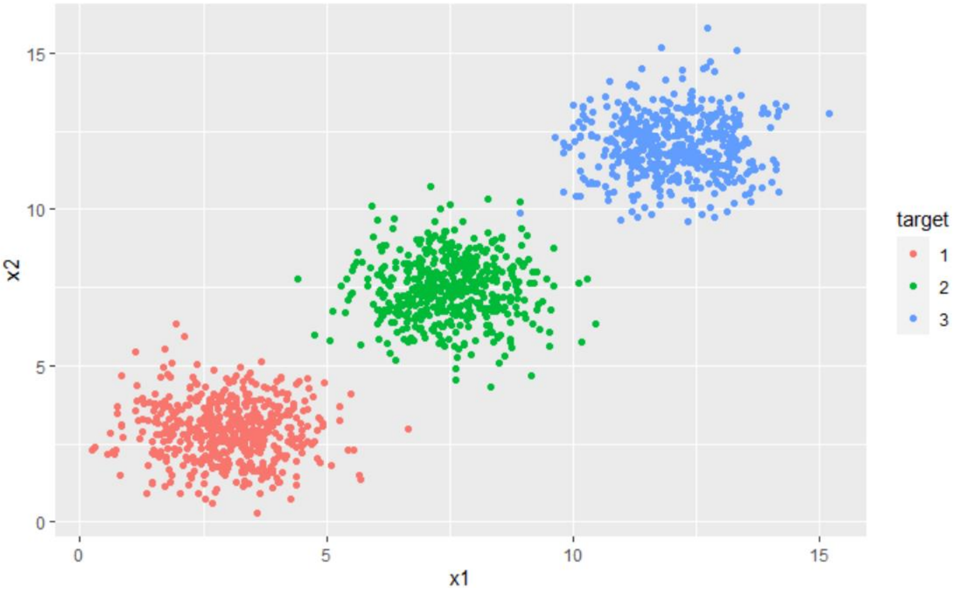
To draw a graph similar as Figure 4.2, First I generate data for this graph. Randomly generate 500 points for each class, from distribution below:

Class 1: x1~Normal(3,1); x1~Normal(3,1)

Class 2: x2~Normal(7.5,1); x1~Normal(7.5,1)

Class 3: x1~Normal(12,1); x1~Normal(12,1)

The scatter plot is shown below:



Then, generate Discriminant functions for each class, use the following formula:

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log \pi_k$$

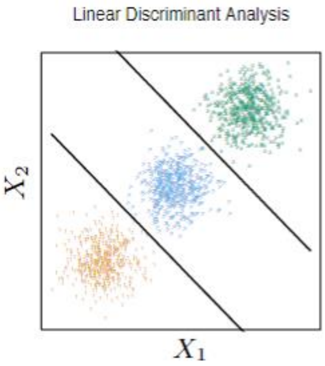
(class slide unit 4 page 41)

Σ is covariance matrix, μk is mean of x when in class k, πk is prior probability of class k presents in the data.

discriment function of class 1: δ1(x) = 0.14706229 * x1 + 0.06090421 * x2 - 1.407257

discriment function of class 2: δ2(x) = 0.3057810 * x1 + 0.2198876 * x2 - 3.065252

discriment function of class 3: δ3(x) = 0.4643660 * x1 + 0.3816839 * x2 - 6.192982



(class slide 4.2)

To generate two lines similar as given plot, I need to find the boundaries between class1 and class 2, between class 2 and class 3. LDA predict value to class k when δk(x) is the maximum among all δ(x).

For the boundary between class 1 and class 2:

when δ1(x) = δ2(x) > δ3(x):

First look at δ1(x) = δ2(x)

0.14706229 * x1 + 0.06090421 * x2 - 1.407257 = 0.3057810 * x1 + 0.2198876 * x2 - 3.065252

0.1587187 x1 + 0.1589834 x2 = 1.657995,

x2 = -0.998335 x1 + 10.42873, which is the function when δ1(x) = δ2(x)

base on the condition, δ2(x) > δ3(x) and x2 = -0.998335 x1 + 10.42873,

0.3057810 * x1 + 0.2198876 * x2 - 3.065252 > 0.4643660 * x1 + 0.3816839 * x2 - 6.192982,

0.158585 x1 + 0.1617963 x2 < 3.12773, when apply x2 = -0.998335 x1 + 10.42873:

-0.002941909 x1 < 1.4404, x1 > -489.6141

so, when x1 > -489.6141, the line segment between class 1 and class 2 is x2 = -0.998335 x1 + 10.42873.

Repeat the step above for class 2 and class 3:

when δ2(x) = δ3(x) > δ1(x):

First look at δ1(x) = δ2(x)

0.3057810 * x1 + 0.2198876 * x2 - 3.065252 = 0.4643660 * x1 + 0.3816839 * x2 - 6.192982,

0.158585 x1 + 0.1617963 x2 = 3.12773,

line segment: x2 = -0.9801522 x1 + 19.33128

base on this condition, δ3(x) > δ1(x) and x2 = -0.9801522 x1 + 19.33128,

0.4643660 * x1 + 0.3816839 * x2 - 6.192982 > 0.14706229 * x1 + 0.06090421 * x2 - 1.407257

0.3173037x1 + 0.3207797x2 > 4.785725

apply x2 = -0.9801522 x1 + 19.33128:

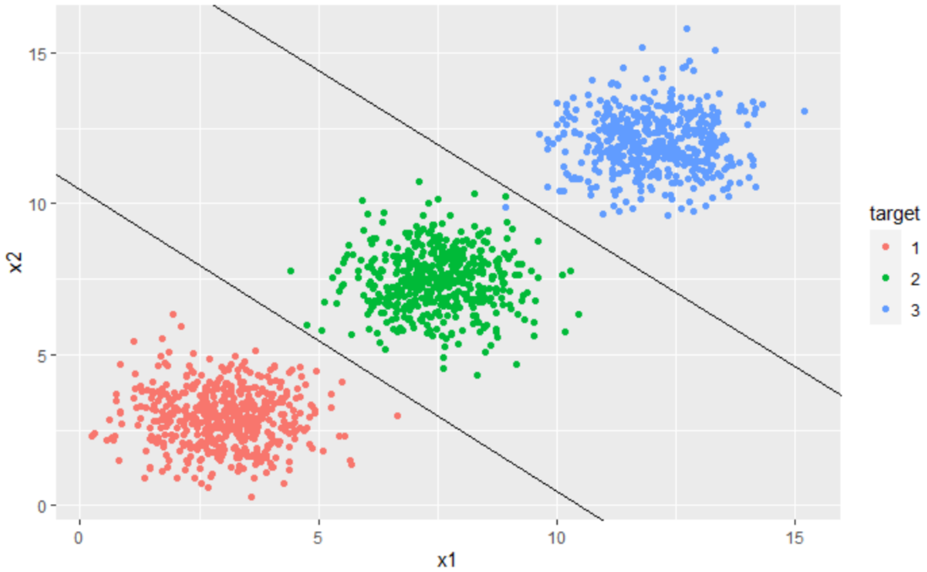
0.3173037x1 + 0.3207797(-0.9801522 x1 + 19.33128) > 4.785725

0.002890771 x1 > -1.415357, x1 > -489.6123

so, when x1 > -489.6123, the line segment between class 2 and class 3 is x2 = -0.9801522 x1 + 19.33128

Now draw these two lines: x2 = -0.998335 x1 + 10.42873 (x1 > -489.6141),

x2 = -0.9801522 x1 + 19.33128 (x1 > -489.6123),



The above steps shows how these two lines obtained. The line x2 = -0.998335 x1 + 10.42873 (x1 > -489.6141) (the between red and green group) is the line segment of class 1 and class 2. The point on this line predicts to both class 1 and class 2, because δmax(x) = δ1(x) = δ2(x) > δ3(x).

The line x2 = -0.9801522 x1 + 19.33128 (x1 > -489.6123) (the between green and blue group) is the line segment of class 2 and class 3. The point on this line predicts to both class 2 and class 3, because δmax(x) = δ2(x) = δ3(x) > δ1(x).

2(b)

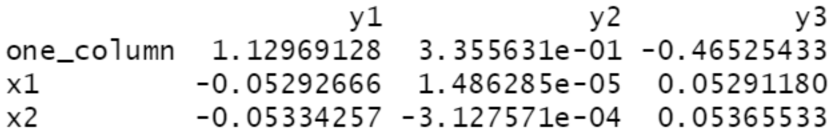
I'm using Linear Regression using an Indicator Matrix that discribed in 4.2

According to the data in 2(a), first generate LSE by following formula:

$$\hat{\beta} = X(X^{\intercal}X)^{-1}X^{\intercal}Y$$,

Y is indicator response matrix that generated by previous target variable.

LSE:



```
                    y1            y2          y3
one_column  1.12969128  3.355631e-01 -0.46525433
x1         -0.05292666  1.486285e-05  0.05291180
x2         -0.05334257 -3.127571e-04  0.05365533
```

For a new observation with the input $X^{*} = c(x1^{*}, x2^{*})$

$$f(X) = t(\hat{B}) * X^{*} = c(f_1(x), f_2(x), f_3(x))$$

$$f_1(x) = 1.1296913 - 5.292666\text{e-}02 * x1 - 0.0533425704 * x2$$

$$f_2(x) = 0.3355631 + 1.486285\text{e-}05 * x1 - 0.0003127571 * x2$$

$$f_3(x) = -0.4652543 + 5.291180\text{e-}02 * x1 + 0.0536553276 * x2$$

(by coding in R)

classify to class k when fk^(x) is the maximum value.

when we classify x as class 1 rather than class2 and class 3,

f1^(x) > f2^(x) and f1^(x) > f3^(x),

1.1296913 - 5.292666e-02 * x1 - 0.0533425704 * x2 > 0.3355631 + 1.486285e-05 * x1 - 0.0003127571 * x2 and

1.1296913 - 5.292666e-02 * x1 - 0.0533425704 * x2 > -0.4652543 + 5.291180e-02* x1 + 0.0536553276 * x2,

The result are

x2 < 14.97513 - 0.9983351 x1 when f1^(x) > f2^(x) and

x2 < 14.90633 - 0.9891643 x1 when f1^(x) > f3^(x)

For x1 > 7.502072, x2 < 14.97513 - 0.9983351* x1 since it satisfies both conditions above.

Otherwise, x2 < 14.90633 - 0.9891643 x1, f1^(x) is the maximum, then we classify to class 1.

when we classify x as class 2 rather than class1 and class 3,

f2^(x) > f1^(x) and f2^(x) > f3^(x)

from above, x2 < 14.97513 - 0.9983351 x1 when f1^(x) > f2^(x). So, when x2 > 14.97513 - 0.9983351 x1 when f2^(x) > f1^(x)

for f2^(x) > f3^(x), 0.3355631 + 1.486285e-05 * x1 - 0.0003127571 * x2 > -0.4652543 + 5.291180e-02* x1 + 0.0536553276 * x2

x2 < 14.83872 - 0.9801523 x1 when f2^(x) > f3^(x)

Also, x2 > 14.97513 - 0.9983351 x1 when f2^(x) > f1^(x)

So, when 14.97513 - 0.9983351 x1 < x2 < 14.83872 - 0.9801523 x1, f2^(x) is the maximum, then we classify to class 2.

There is a masking problem when classify class 2, since two boundaries is too close to each other.

when we classify x as class 3 rather than class1 and class 2,

f3^(x) > f1^(x) and f3^(x) > f2^(x)

from above, x2 < 14.90633 - 0.9891643 x1 when f1^(x) > f3^(x),

x2 < 14.83872 - 0.9801523 x1 when f2^(x) > f3^(x)

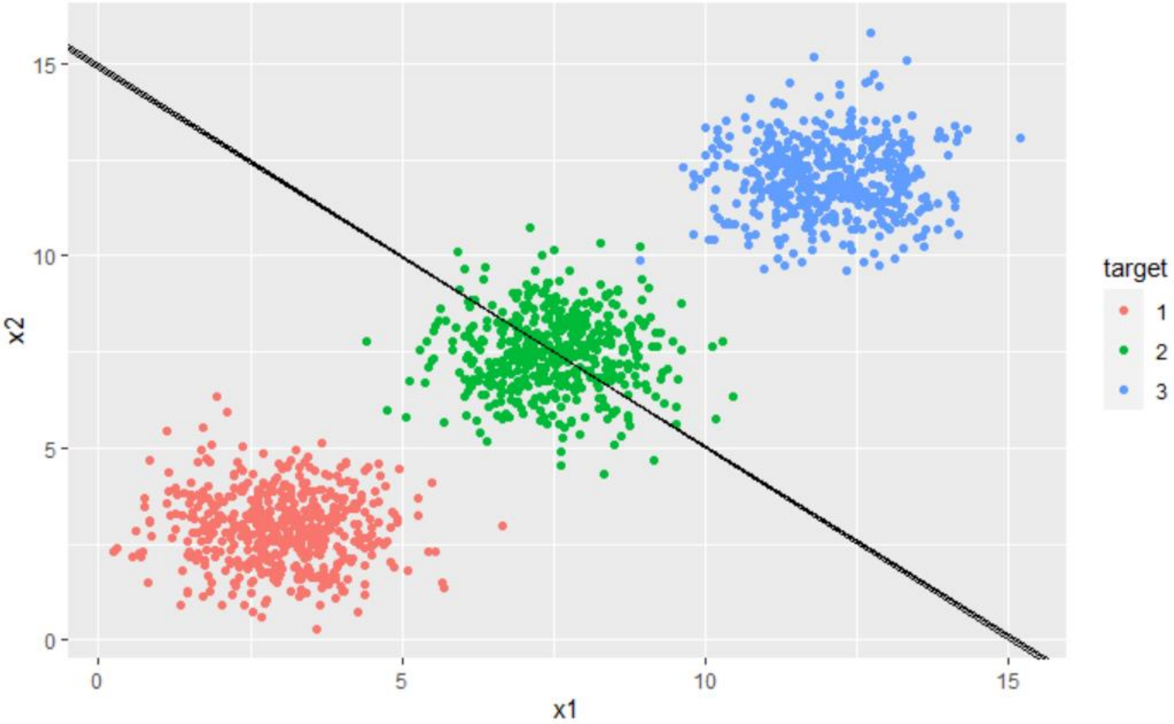
so, x2 > 14.90633 - 0.9891643 x1 when f3^(x) > f1^(x),

x2 > 14.83872 - 0.9801523 x1 when f3^(x) > f2^(x)

when x1 < 7.502219, x2 > 14.83872 - 0.9801523 x1 since it satisfies both of conditions above.

else x2 > 14.90633 - 0.9891643 x1

Draw the graph by above boundaries:



These lines are very close to each other. Obviously, I observed there is a masking problem for class 2, because it looks like a thick line separate the prediction into two classes. The reason for masking problem is:

1. f^k(x) can be negative or greater than 1, which causes classes can be masked by other classes. For example, when x1 = x2 = 0, f^1(x) = 1.1296913 > 0. This may cause problem especially if make predictions outside training data.
2. As described in *C. Zhang, H. Fu* (2006, section 3.1) theorem 1, at least 1 class in linear regression of an indicator response matrix will be masked when satisfy following assumptions:

   Sample classes centroid on the straight line: my centroid for 3 classes theoretically should be (3,3), (7.5,7.5), (12,12) because of mean of distribution when generated the data, which is on the straight line and on distinct points.

   Sample sizes are equal: each sample has 500 points

   Dimension >= 1 and *class size >= 2* are satisfied by the question description.

   All assumptions satisfy in this question. By applying this theorem, at least 1 class will be masked.

Question 3

My setting for this simulation study:

The response variable Y has 3 classes, class 1,2,3. Among these 3 classes, for each simulation, class size for class 1 is 300, for class 2 is 500, for class 3 is 700 when training the data by each model. Choose class size for class 1 is 60, for class 2 is 100, for class 3 is 140 when testing the data by each model.

For p+q predictors, I choose p = q = 3. p predictors include x1, x2 and x3. q predictors include x4, x5 and x6.

For class 1, x1~norm(3,1), x2~uniform(-1,1), x3 ~ gamma(1,1)

For class 2, x1~norm(7,1), x2~uniform(1,3), x3 ~ gamma(3,3)

For class 3, x1~norm(11,1), x2~uniform(3,5), x3 ~ gamma(5,5), which they all have different distributions for each class.

x4 ~ norm(3,3), x5 ~ beta(4,4), x6 = x4 * x5, which they all have same distributions for each class.

x6 is dependent on x4 and x5.

Simulation can decrease the risk to find a biased estimation of classification error, because the function to generate random number by distribution in R (such as rnorm, rbiom), each time randomly generate different numbers according to this distribution. By simulating the result with more times, the classification error will approximate closer to the true result by data with these distributions.

By applying simulation steps to check which methods are more accurate in classifying a given data set, I use 500 simulations to repeat the step below:

Generate given dataset by distribution above, then training the models by training dataset with 1500 rows of data. After that, predict the result by testing dataset with 300 rows of data, calculate classification error for each model and append the value into a vector.

After 500 simulations, I get 3 lists with classification error from 3 models in 500 simulations, then take the average to get the classification error from simulation.

Classification error for linear regression: 0.17532

Classification error for linear discriminant analysis: 0.002673333

Classification error for multiple logistic regression: 0.002666667

The result shows linear discriminant analysis and multiple logistic regression has similar performance to classify the data, both of them has a very low classification error rate. However, linear regression has a relatively high classification error which is 0.17532.

From the result, I know linear discriminant analysis and multiple logistic regression has great performance on the given distribution. Compare with linear regression with indicator matrix, LDA avoids the masking problem (*Hastie et al., 1994*). It has linear boundaries because boundaries appears when δi = δj = max(δ), shows linear boundaries that similar as steps in question 2. Logistic regression is more robust, safer than LDA, which has fewer assumptions. (*Trevor H.,* *Robert T., Jerome F., chapter 4.4.5*)

$$\log \frac{\Pr(G=k|X=x)}{\Pr(G=K|X=x)} = \beta_{k0} + \beta_k^T x.$$

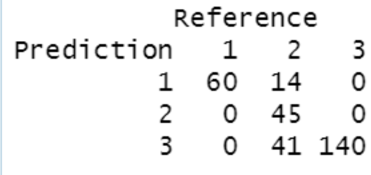
(same form for logistic regression and LDA)

The models have the exact same form, but LDA has one more assumption, a common covariance matrix. This theoretical result shows similarity in the simulation, since these two model's results are similar.

Also see the symptom by this dataset. The dataset does not have many overlapping between each label, which is linear separable to apply linear methods like LDA and logistic regression.

Since LDA and logistic regression in the dataset has a very small error(<0.003), there are still question from this simulation result for linear regression. Why linear regression has poor performance?

In question 2, I mentioned linear regression for classification may have masking problem because of f^k(x) can be negative or greater than 1.



```
          Reference
Prediction   1   2   3
         1  60  14   0
         2   0  45   0
         3   0  41 140
```

(example True/False Table provided by one of the simulation, R caret package)

By this true/False table, there is still some masking problem with class = 2, which shows linear regression is not suitable to use in current classification.

Citation:

A.M. Johansen, 2010, Markov Chain Monte Carlo, International Encyclopedia of Education (Third Edition), Pages 245-252

Zhang, C., Fu, H., 2006, Masking effects on linear regression in multi-class classification, Statistics & Probability Letters, Volume 76, Issue 16, Pages 1800-1807

Hastie, T., Tibshirani, R. and Buja, A. (1994). Flexible discriminant analysis by optimal scoring, Journal of the American Statistical Association 89: 1255–1270.

Trevor H., Robert T., Jerome F., The Elements of Statistical Learning, Data Mining, Inference, and Prediction, Second Edition

Appendix:

Codes are in 462_final_Tao_Shan.RMD and 462_final_Tao_Shan.R. RMD file shows the code clearer than .R file.