# STAT545 HW 5

*Yi Yang*

*10/27/2018*

## 1 Problem 1: Exponential family distribution

This continues from parts (a) and (b) Problem 1 from Homework 4 on exponential family distibiutions.

1. Consider a random variable $x$ that can take $D$ values and that is distributed according to the discrete distribution with parameter $\overrightarrow{\pi}$, We will write this as $p(x|\overrightarrow{\pi})$, with $p(x = c|\overrightarrow{\pi}) = \pi_c$ for $c \in \{1, \dots, D\}$.

c) What is the form of the conjugate prior on the parameters of $p(x|\pi)$? You only need to write it upto a multiplicative constant (i.e., you don't have to write the normalization constant). What is its feature vector?

d) If you call the natural parameters of this distribution $\overrightarrow{a} = (a_1, \dots)$ (part (c) will give the the dimensionality of $\overrightarrow{a}$), what are the parameters of the posterior distribution given a set of observations $X = (x_1, \dots, x_N)$? (The point here is that in general the posterior distribution can be very complicated, even for the simple priors (so that we need methods like MCMC). However for conjugate priors, the posterior lies in the smae family as the prior, it just has different parameters)

2. Let $x$ be Poisson distribution with mean $\lambda$. Repeat parts (c) and (d).
3. Let $x$ be a 1-dimensional Gaussian with mean $\mu$ and variance $\sigma^2$. Repeat parts (c), (d) (Note: both $\mu$ and $\sigma^2$ are parameters).
4. Let $x$ follow a geometric distribution with success probability $p$: $(Pr(X = k) = (1-p)^k p$ for $k = 0, 1, 2, \dots)$. Repeat parts (c), (d).

**Solution:**

1. The discrete distribution can be expressed in exponential form:

$$\begin{aligned} p(x|\pi) &= \exp(\sum_{c=1}^{D} \delta(x=c) \cdot \log(\pi_c)) \\ &= \pi_D \exp(\sum_{c=1}^{D-1} \delta(x=c) \log(\frac{\pi_c}{\pi_D})) \\ &= \frac{1}{Z(\theta)} \exp(\theta^\top \phi(x)) \end{aligned}$$

We have $\theta = [\log\frac{\pi_1}{\pi_D}, \dots, \log\frac{\pi_{D-1}}{\pi_D}]^\top$, and $\phi(x) = [\delta(x=1), \delta(x=2), \dots, \delta(x=D-1)]$. Define $\zeta(\theta) = \log Z(\theta)$, we have $p(x|\theta) = \exp(\theta^\top\phi(x) - \zeta(\theta))$, the conjugate prior is given by

$$p(\theta|a, b) \propto \eta(\theta) \exp(\theta^\top a - \zeta(\theta) b)$$

The feature vector is given by $[\theta_1, \theta_2, \dots, \theta_{D-1}, \zeta(\theta)]^\top = [\log\frac{\pi_1}{\pi_D}, \log\frac{\pi_2}{\pi_D}, \dots, \log\frac{\pi_{D-1}}{\pi_D}, -\log\pi_D]^\top$.

The parameters of the posterior distribution is given by

$$(a + \sum_{i=1}^{N} \phi(x_i), b + N) = (a_1 + \sum_{i=1}^{N} \delta(x=1), a_2 + \sum_{i=1}^{N} \delta(x=2), \dots, a_{D-1} + \sum_{i=1}^{N} \delta(x=D-1), b+N)$$

2. The conjugate prior is given by

$$\begin{aligned} p(x|\lambda) &= \frac{\lambda^x \exp(-\lambda)}{x!} = \exp(-\lambda)\frac{1}{x!}\exp((\log\lambda)x) \\ &= \frac{1}{Z(\theta)}h(x)\exp(\theta\cdot\phi(x)) \end{aligned}$$

   We have $\theta = \log\lambda$, $\phi(x) = x$ and $h(x) = \frac{1}{x!}$. With $\zeta(\theta) = logZ(\theta)$, we have $p(x|\theta) = exp(\theta\cdot\phi(x) - \zeta(\theta))$, and the conjugate prior is

$$p(\theta|a,b) \propto \eta(\theta)\exp(\theta\cdot a - \zeta(\theta)b)$$

The feature vector is

$$(\theta, \zeta(\theta)) = (\log\lambda, -\lambda)$$

The parameters of the posterior distribution is

$$(a + \sum_{i=1}^{N}\phi(x_i), b+N) = (a + \sum_{i=1}^{N} x_i, b+N)$$

3. The conjugate prior is given by

$$\begin{aligned} p(x|\mu,\sigma^2) &= \frac{\exp\left(-\frac{\mu^2}{2\sigma^2}\right)}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x\right) \\ &= \frac{1}{Z(\theta)}\exp(\theta^\top\phi(x)) \end{aligned}$$

   We have $\theta = (\frac{1}{2\sigma^2}, \frac{\mu}{\sigma^2})$, $\phi(x) = (x^2, x)$ and $h(x) = 1$. With $\zeta(\theta) = logZ(\theta)$, we have $p(x|\theta) = exp(\theta^\top\phi(x) - \zeta(\theta))$, and the conjugate prior is

$$p(\theta|a,b) \propto \eta(\theta)exp(\theta^\top a - \zeta(\theta)b)$$

The feature vector is

$$(\theta_1, \theta_2, \zeta(\theta)) = (\frac{1}{2\sigma^2}, \frac{\mu}{\sigma^2}, \frac{\mu^2}{\sigma^2} - \frac{1}{2}\log 2\pi\sigma^2)$$

The parameters of the posterior distribution is

$$(a + \sum_{i=1}^{N}\phi(x_i), b+N) = (a_1 + \sum_{i=1}^{N} x_i^2, a_1 + \sum_{i=1}^{N} x_i, b+N)$$

4. For a geometric distribution, we have

$$\begin{aligned} Pr(X=k) &= \exp(X\log(1-p) + \log(p)) = p\exp(X\log(1-p)) \\ &= \frac{1}{Z(\theta)}h(x)\exp(\theta\cdot\phi(x)) \end{aligned}$$

   We have $\theta = log(1-p)$, $\phi(x) = X$ and $h(x) = 1$. With $\zeta(\theta) = logZ(\theta)$, we have $p(x|\theta) = exp(\theta\cdot\phi(x) - \zeta(\theta))$, so that the conjugate prior is

$$p(\theta|a,b) \propto \eta(\theta)\exp(\theta\cdot a - \zeta(\theta)b)$$

   The feature vector is

$$(\theta, \zeta(\theta)) = (\log(1-p), \log p)$$

   The parameters of the posterior distribution is

$$(a + \sum_{i=1}^{N}\phi(x_i), b+N) = (a + \sum_{i=1}^{N} x_i, b+N)$$

## 2 Problem 2: Gradient descent for blind source separation

Let $Y$ be a $3 \times T$ matrix consisting of three independent audio recordings. Each column $\mathbf{y}_t = (y_{1t}, y_{2t}, y_{3t})^\top$ consists of the intensities of each source at time $t$. Even though each audio signal is a time-series, we will model the $y_{ij}$ 's as i.i.d. draws from the hyperbolic secant distribution $p(y) = \frac{1}{\pi \cosh(y)}$.

1. Plot $p(y)$. Suppose a Gaussian with the same mean and variance.

Instead of observing $\mathbf{Y}$, we observe $\mathbf{X} = \mathbf{A} \cdot \mathbf{Y}$, where $\mathbf{A}$ is a $3 \times 3$ 'mixing'-matrix. Given $\mathbf{X}$, we want to recover both $\mathbf{A}$ and $\mathbf{Y}$. Note this problem is ill-posed, i.e., there are infinite collection of valid $(\mathbf{A}, \mathbf{Y})$ pairs. However, our model of $\mathbf{Y}$ (in particular, the source-independence and non-Gaussian assumptions) will allow us to recover a sensible solution.

Assume, $\mathbf{A}$ is invertible, and define $\mathbf{W} = \mathbf{A}^{-1}$, Write $w^s$ for row s of $\mathbf{W}$.

2. Show that $\log p(\mathbf{X}|\mathbf{W}) = T \log |\mathbf{W}| + \sum_{t=1}^{T} \sum_{s=1}^{3} \log p(\mathbf{w}^s \mathbf{x}_t)$. (Hint: see e.g. the last paragraph here: http://sccn.ucsd.edu/wiki/Random_Variables_and_Probability_Density_Functions). We will find the maximum likelihood (ML) estimate of $\mathbf{W}$ by gradient ascent.
3. Show that $\frac{\partial \log p(\mathbf{x}|\mathbf{w})}{\partial w_{ij}} = T a_{ji} + \sum_{t=1}^{T} \frac{\partial \log p(y_{it})}{\partial y_{it}} x_{jt}$ (Hint: look at wikipedia for the derivative of a determinant).

It follows that

$$\frac{1}{T} \frac{\partial \log p(\mathbf{X}|\mathbf{W})}{\partial \mathbf{W}} = \mathbf{A}^T + \frac{1}{T} \sum_{t=1}^{T} \frac{\partial \log p(\mathbf{y}_t)}{\partial \mathbf{y}_t} \mathbf{x}_t^\top$$

For our choice of $p(y)$, we have $\frac{\partial \log p(\mathbf{y}_t)}{\partial \mathbf{y}_t} = -\tanh(\mathbf{y}_t) := [-\tanh(y_{1t}), -\tanh(y_{2t}), -\tanh(y_{3t})]^\top$.

The summation over $T$ on the RHS can be written as $-\tanh(\mathbf{Y}) \cdot \mathbf{X}^\top$: this is just one line of R code.

4. Gradient ascent iteratively tries to reach a local maximum of the likelihood according to the rule

$$\mathbf{W}_{new} = \mathbf{W}_{old} + \eta \left( \frac{1}{T} \frac{\partial \log p(\mathbf{X}|\mathbf{W})}{\partial \mathbf{W}} \Big|_{\mathbf{W}_{old}} \right)$$

   The last term is the gradient evaluated at $\mathbf{W}_{old}$. Ping in the epression for the gradient to write the update rule.

5. Write a function that takes a matrix $\mathbf{X}$ as input, and performs gradient ascent to get a Ml_estimate of $\mathbf{W}$. Your function should have an initilization rule and a stopping criteria (which you will describe later). You will also need to set the learning rate $\eta$. You can get better results by using a smaller $\eta$ for later iterations.

On the course webpage are three wav files, mike1.wav, mike2.wav and mike3.wav. Download them, and load them into R using load.wave from the audio package. Hopefully, you will get three 490000-length vectors, which you can store into a matrix $XX$. Directly running your function on this probably won't work (but you can try).

6. Write a $3 \times 3$ covariance matrix of $\mathbf{X}$. Directly running your function on this pari of recordings, plot a scatter plot of values (i.e., for rows $(1, 2)$, $(2, 3)$ and $(1, 3)$).
7. If Cov is the covariance matrix, calculate its square root using sqrtm function from the expm package. Call this sqCov. Define a new witened matrix $\mathbf{X}_{white} = \mathbf{sqCov}^{-1} \cdot \mathbf{X}$. This should have coriance equal to the identity matrix. Verify this.
8. Run your gradient ascent algorithm on the whitened data. How did you initialize $\mathbf{W}$? Describe how you set the learning rate $\eta$ as well as your stopping criteria (I set $\eta$ to about 1). How many iterations did you need? Plot the evolution of the log-likelihood. What is the returned estimate $\hat{\mathbf{W}}$? Your actual demixing matrix for the original data $\mathbf{X}$ is this value multiplied by $\mathbf{sqCov}^{-1}$. Print this out. If you had to restart with different initializations, describe this too.

9. Plot histogram of the raw data, the whitened data (a total of 6 histograms).
10. Your estimate of the latent source is given by $\hat{\mathbf{Y}} = \hat{\mathbf{W}} \cdot \mathbf{sqCov}^{-1} \cdot \mathbf{X}$. Plot a histogram of the marginals for each source. What is the their covariance? Plot the three pairwise scatterplots.

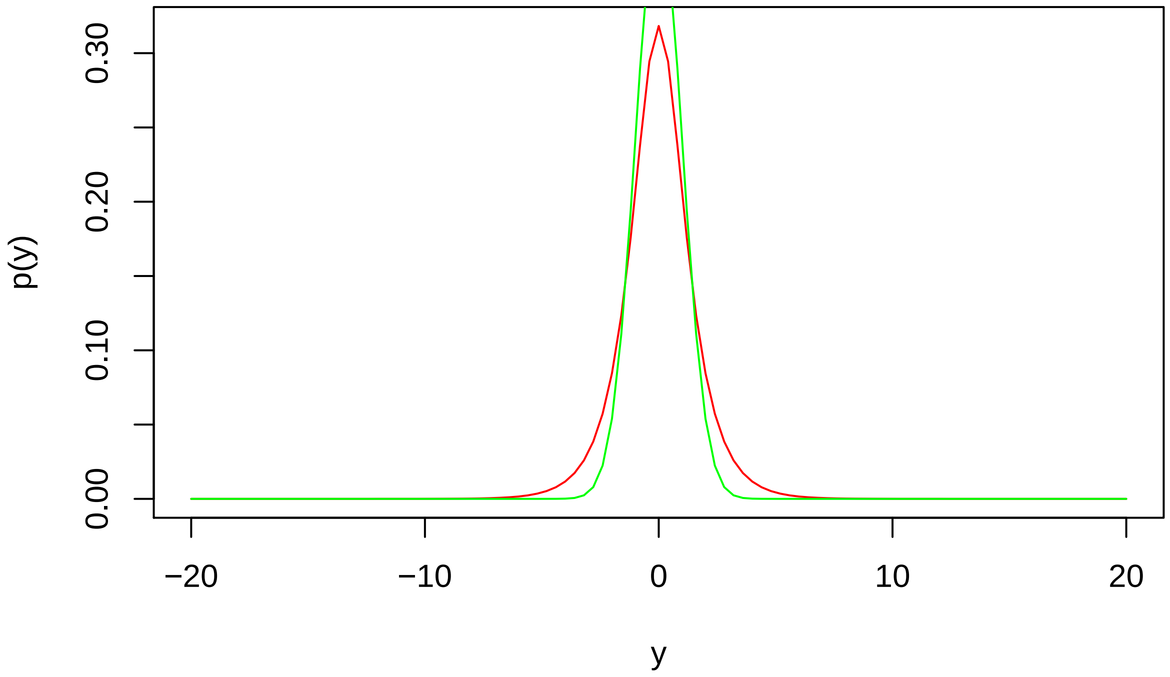
You can verify your answer by listening to the recoverd sources. You can save these using the save.wave command. You should normalize these first. (I divided by twice the maxximum value for each source).

11. Is the maximum likelihood estimate of **W** you obtained unique? If so explain why, if not, explain some sources of degeneracy.

**Solution:**

1.

```
hyper = function(y) { 1 / (pi*cosh(y)) }
curve(hyper, from=-20, to=20, xlab="y", ylab="p(y)",col='red')
x = seq(-20, 20, 0.1)
curve(dnorm(x, mean=0, sd=1), add=T, col='green')
```



2. Proof: For each column $\boldsymbol{y}_t$, where $i = 1, 2, 3$. The linear transformation has rendered a relationship of pdf as follows

$$\boldsymbol{x}_t = A\boldsymbol{y}_t \rightarrow p(\boldsymbol{x}_t) = |\boldsymbol{A}|^{-1} p(\boldsymbol{A}^{-1}\boldsymbol{x}_t) = |\boldsymbol{W}| p(W\boldsymbol{x}_t)$$

$$\begin{aligned}\log p(\boldsymbol{X}|\boldsymbol{W}) &= \log \prod_{t=1}^{T} |\boldsymbol{W}| p(\boldsymbol{W}\boldsymbol{x}_t) \\ &= T \log |\boldsymbol{W}| + \sum_{t=1}^{T} \sum_{s=1}^{3} \log p(\boldsymbol{w}^s \boldsymbol{x}_t)\end{aligned}$$

The maximum likelihood (ML) estimate of $\boldsymbol{W}$ can be approximated by gradient descent.

3. Proof: The derivative of the determinant of a matrix is given by $\frac{\partial \log |\boldsymbol{W}|}{\partial \boldsymbol{W}} = \boldsymbol{W}^{-\top} = \boldsymbol{A}^{\top}$. That is, $\frac{\partial \log |\boldsymbol{W}|}{\partial w_{ij}} = a_{ji}$.

$$\begin{aligned}\frac{\partial \log p(\boldsymbol{X}|\boldsymbol{W})}{\partial w_{ij}} &= Ta_{ji} + \sum_{t=1}^{T} \frac{\partial \log p(y_{it})}{\partial y_{it}} \frac{y_{it}}{w_{ij}} \\ &= Ta_{ji} + \sum_{t=1}^{T} \frac{\partial \log p(y_{it})}{\partial y_{it}} x_{jt}\end{aligned}$$

4.
$$\begin{aligned}\boldsymbol{W}_{new} &= \boldsymbol{W}_{old} + \eta \left( \frac{1}{T} \frac{\partial \log p(\boldsymbol{X}|\boldsymbol{W})}{\partial \boldsymbol{W}} \Big|_{\boldsymbol{W}_{old}} \right) \\ &= \boldsymbol{W}_{old} + \eta \left( A^\top + \frac{1}{T} \sum_{t=1}^{T} \frac{\partial \log p(\boldsymbol{y}_t)}{\partial \boldsymbol{y}_t} \boldsymbol{x}_t^\top \Big|_{\boldsymbol{W}_{old}} \right)\end{aligned}$$

5. The function is defined as follows. Since $\frac{\partial \log p(y)}{\partial y} = -\tanh(y)$, we have $\log p(y) = \log(e^y + e^{-y})$. Therefore, we replace a formula in the algorithm as follows, $\sum_{t=1}^{T}\sum_{s=1}^{3} \log p(w^s x_t) = \text{sum}\{\log[\exp(\mathbf{Y}) + \exp(-\mathbf{Y})]\}$. The gradient ascent function is implemented as follows, the stopping criteria is set be a tolerance error of $\log p(\mathbf{X}|\mathbf{W})$ is obtained or a maximum number of iteration is obtained.

```r
# gradient ascent algorithm
calculate_log_ML = function(ncols, W, Y) {
    temp1 <- ncols * log( abs(det(W)) )
    temp2 <- sum( log( exp(Y) + exp(-Y) ) )
    return(temp1+temp2)
}
GradientAscent = function(X, eta=0.05, tol=0.01, max.iter=100) {
    set.seed(112)
    ncols <- ncol(X)
    W.old <- matrix(rnorm(9,0,1), nrow=3, ncol=3)
    Y <- W.old %*% X
    log_ML_old <- calculate_log_ML(ncols, W.old, Y)
    iter <- 0
    log_ML <- c(log_ML_old)
    while(iter < max.iter) {
        A <- solve(W.old)
        W.delta <- -tanh(Y) %*% t(X) * 1/ncols + t(A)
        W.new <- W.old + eta * W.delta
        Y <- W.new %*% X

        log_ML_new <- calculate_log_ML(ncols, W.new, Y)
        W.old <- W.new
        if (log_ML_new - log_ML_old < tol) break
        log_ML_old <- log_ML_new
        log_ML <- c(log_ML,log_ML_old )
        iter <- iter + 1
    }

    list(W=W.old, iters=iter, log_ML=log_ML )
}
```

6. The recording is read as follows

```r
library(audio)
mike1 <- load.wave('data/mike1.wav')
mike2 <- load.wave('data/mike2.wav')
mike3 <- load.wave('data/mike3.wav')
```

```r
X <- matrix(nrow=3, ncol=length(mike1))
X[1,] <- mike1
X[2,] <- mike2
X[3,] <- mike3
dim(X)
```

```
## [1]      3 490000
```

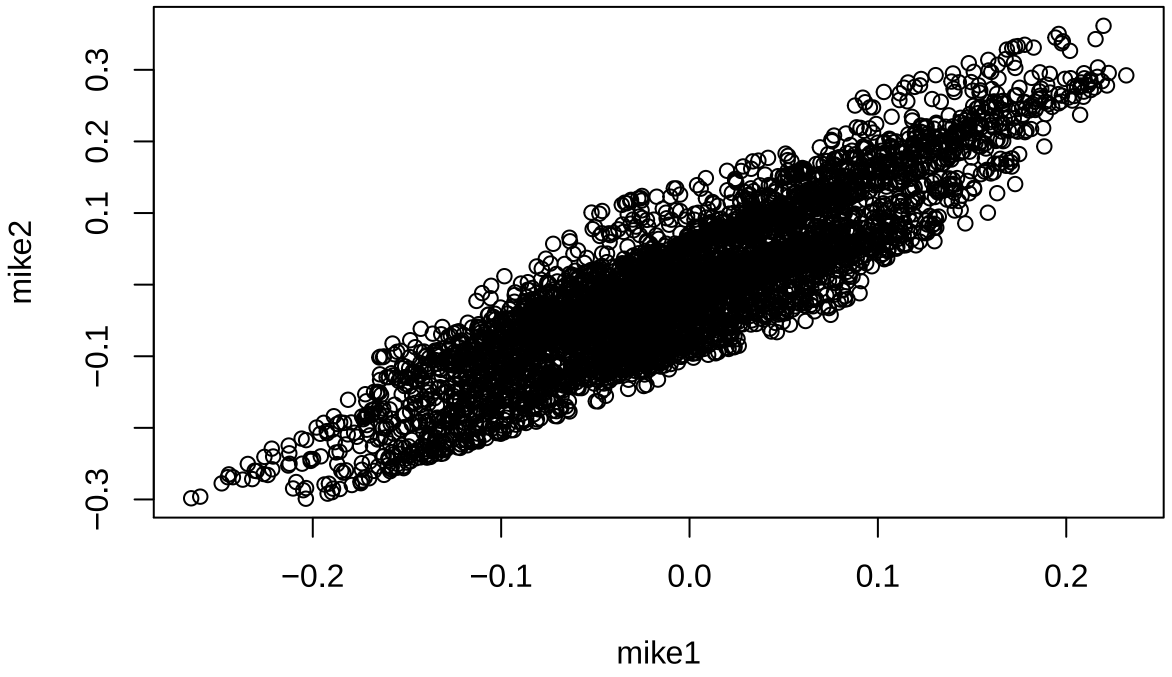
The covariance matrix is given as

```r
X.cov <- matrix(nrow=3, ncol=3)
for (i in seq(1,3)){
  for (j in seq(1,3)){
    X.cov[i,j] <- cov(X[i,],X[j,])
  }
}
X.cov
```

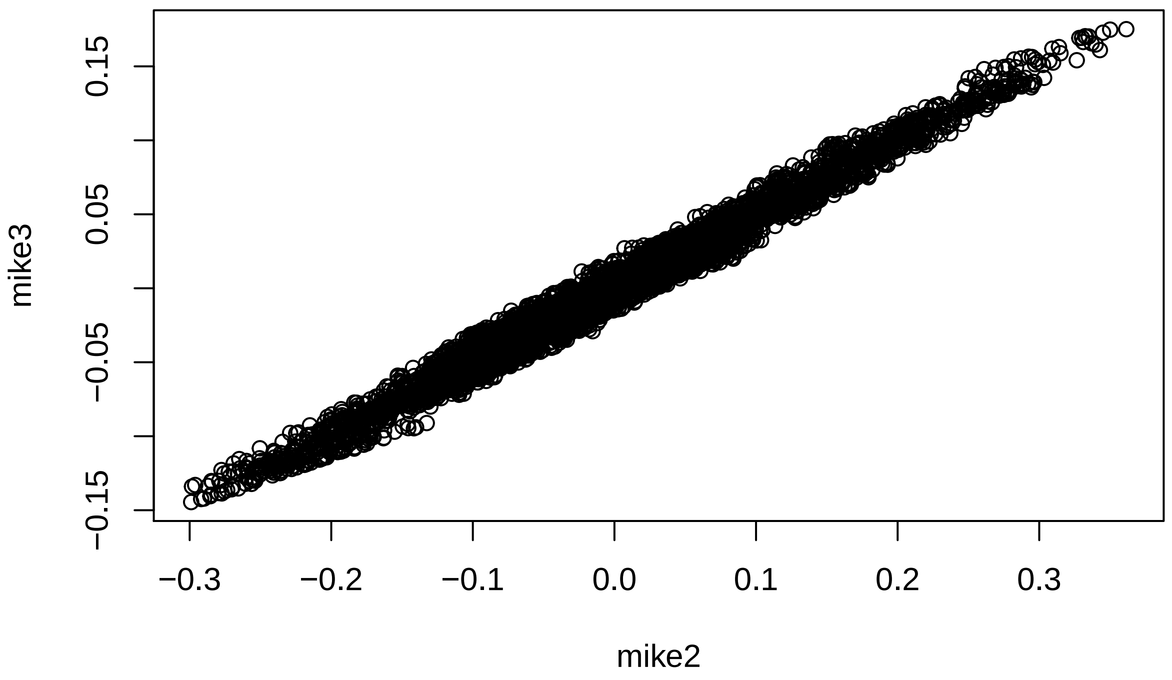
```
##             [,1]        [,2]        [,3]
## [1,] 0.007731833 0.009049568 0.004389554
## [2,] 0.009049568 0.014568385 0.007381079
## [3,] 0.004389554 0.007381079 0.003785355
```

The scatter plots are given by, let us extract the first 5000 points to plot the scatter plots

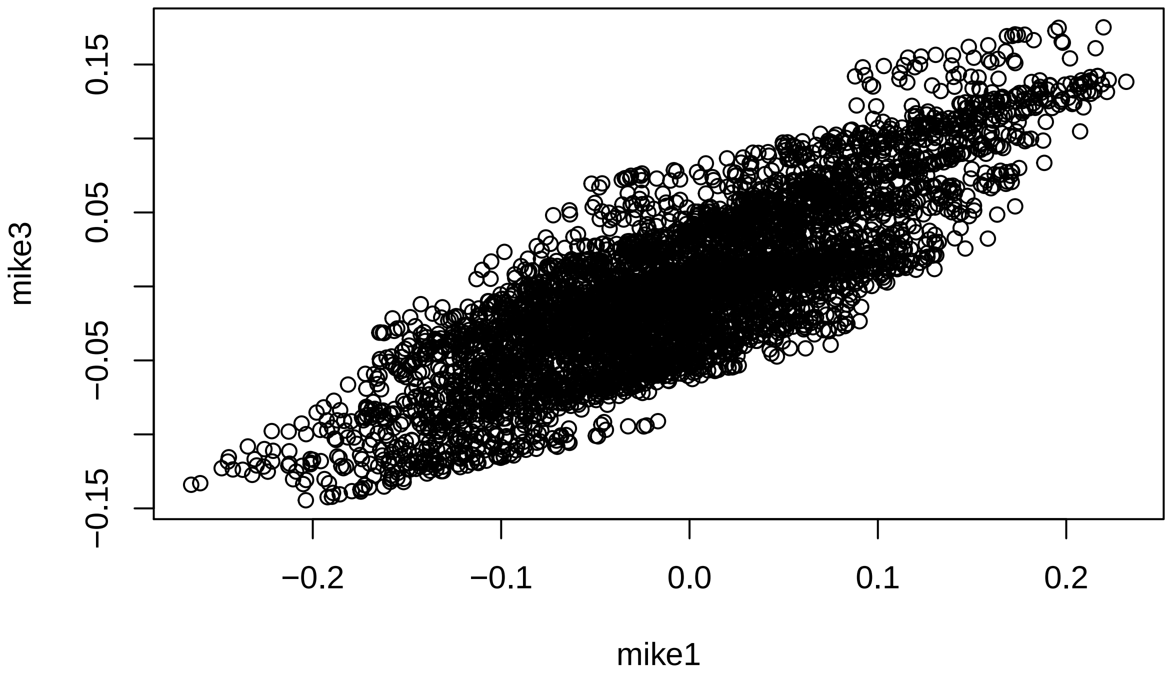
```r
plot(X[1,1:5000], X[2,1:5000], xlab='mike1', ylab='mike2')
```



```r
plot(X[2,1:5000], X[3,1:5000], xlab='mike2', ylab='mike3')
```



```
plot(X[1,1:5000], X[3,1:5000], xlab='mike1', ylab='mike3')
```



7. The square root of the covariance matrix is given by

```
library(expm)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'expm'
```

```
## The following object is masked from 'package:Matrix':
##
```

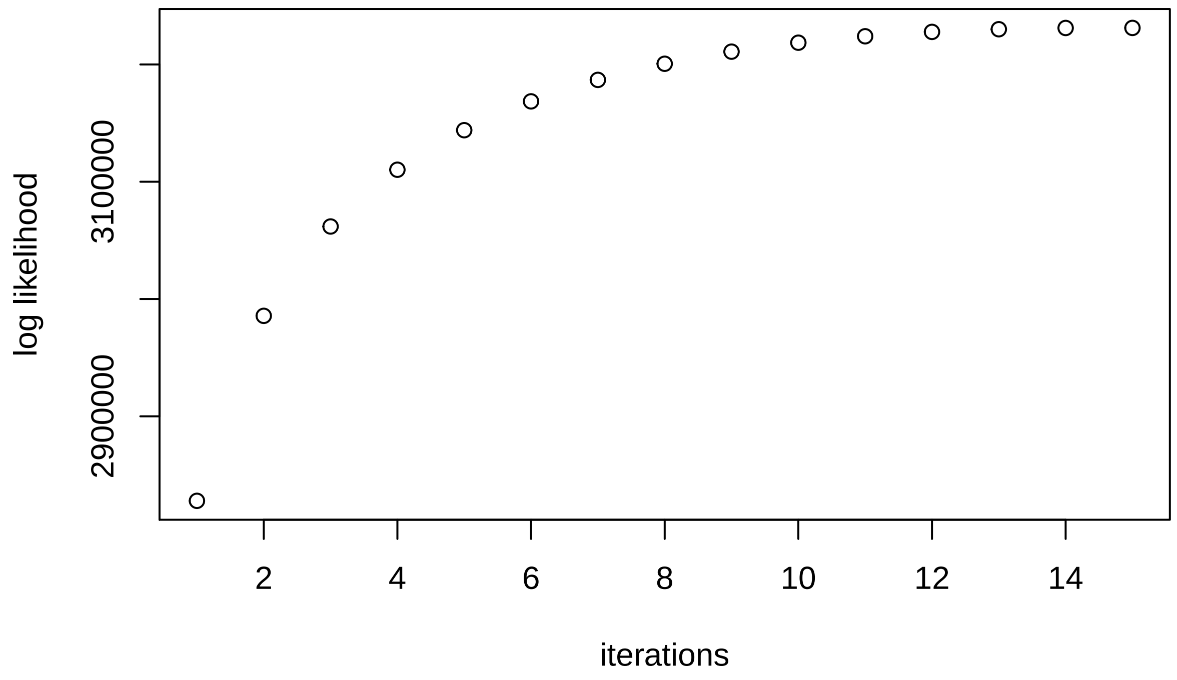
```
##     expm
```

```
X.sqCov <- sqrtm(X.cov)
X_white <- solve(X.sqCov) %*% X
X_white.cov <- matrix(nrow=3, ncol=3)
for (i in seq(1,3)){
  for (j in seq(1,3)){
    X_white.cov[i,j] <- cov(X_white[i,],X_white[j,])
  }
}
X_white.cov
```

```
##               [,1]          [,2]          [,3]
## [1,]  1.000000e+00 -6.165228e-15  5.989866e-15
## [2,] -6.165228e-15  1.000000e+00 -1.535059e-14
## [3,]  5.989866e-15 -1.535059e-14  1.000000e+00
```

8.

```
rslt <- GradientAscent(X_white)
iters <- rslt$iters + 1
plot(seq(1,iters), rslt$log_ML, xlab='iterations', ylab='log likelihood')
```



**W** is initialized with normal distribution with mean 0 and unit standard deviation. Set step rate $\eta = 0.05$, maximum iteration number is 100 and the tolerance to be 0.01, the algorithm convergers after 14 iterations.

The estimated $\hat{\mathbf{W}}$ is given by

```
W_hat <- rslt$W
W_hat
```

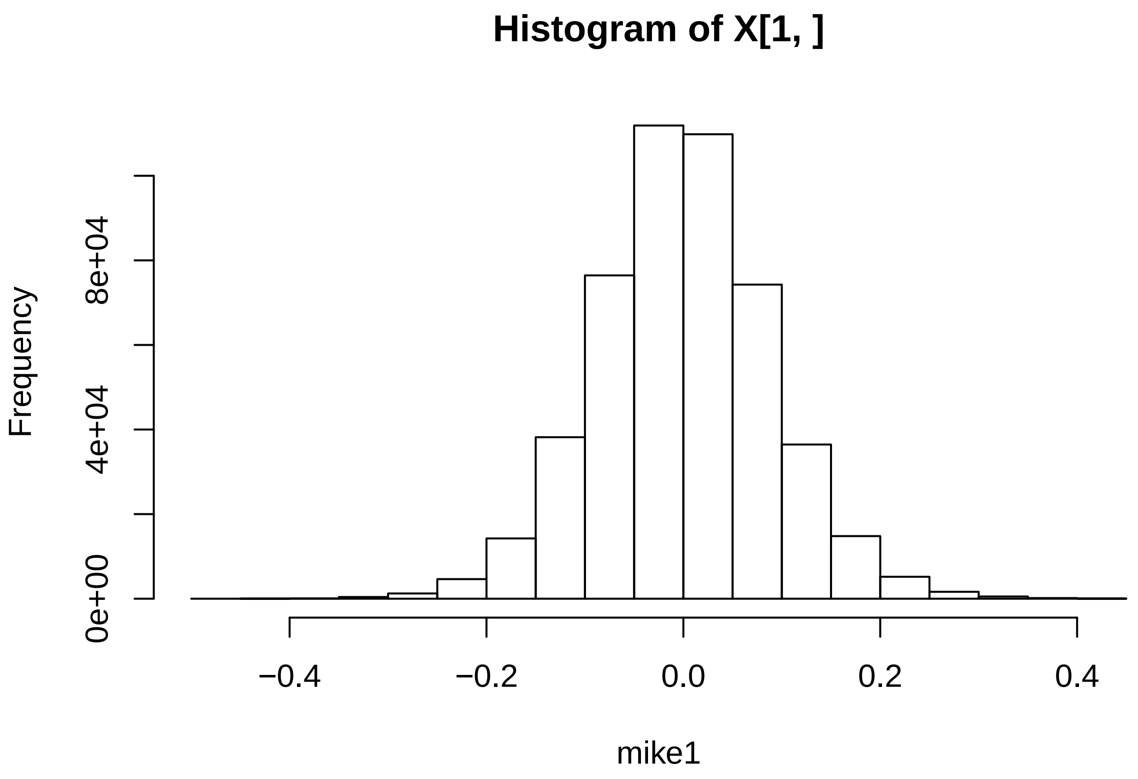
```
##            [,1]       [,2]       [,3]
## [1,] -0.3459975 -1.3934814  1.4781263
## [2,]  2.2143175 -1.0712147  0.2591116
## [3,] -0.6184984 -0.9589408 -0.4750663
```

```
W_demixing <- W_hat %*% solve(X.sqCov)
W_demixing
```

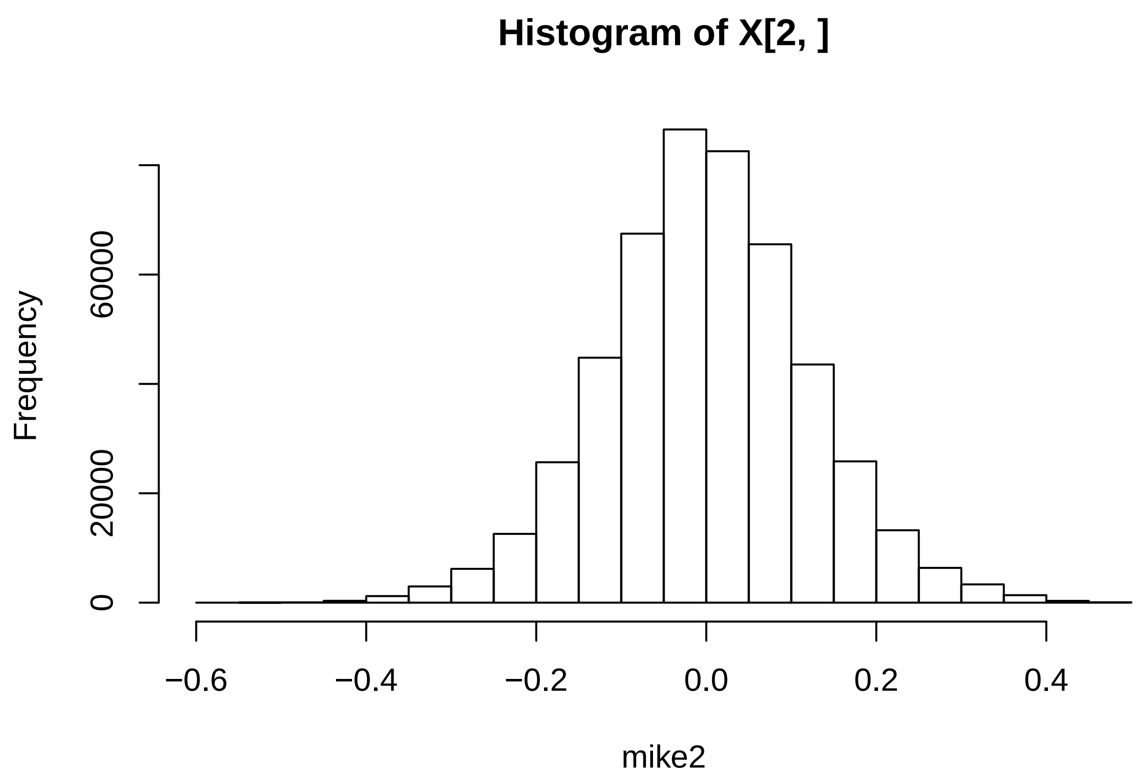
```
##             [,1]        [,2]       [,3]
## [1,] 29.600162 -210.458652 367.983770
## [2,] 65.502483 -121.931136 161.755026
## [3,] -3.462895   -7.160505  -1.679914
```

9.

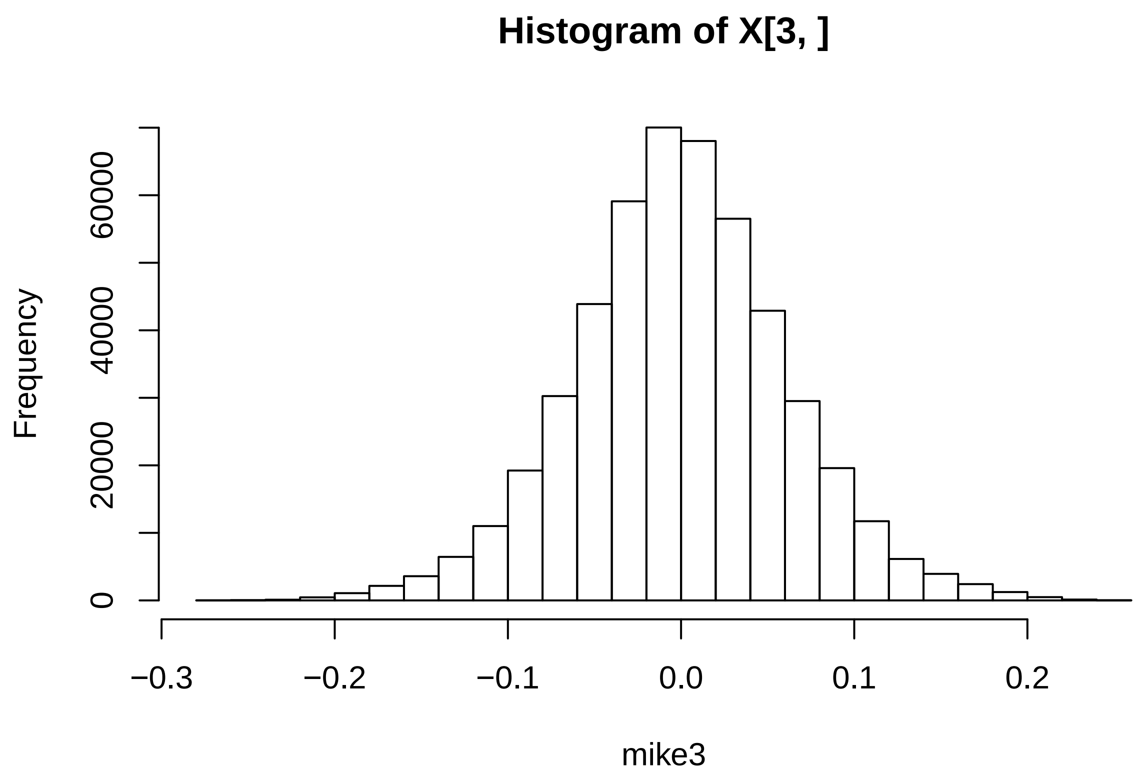
```
# raw data
hist(X[1,], xlab='mike1')
```



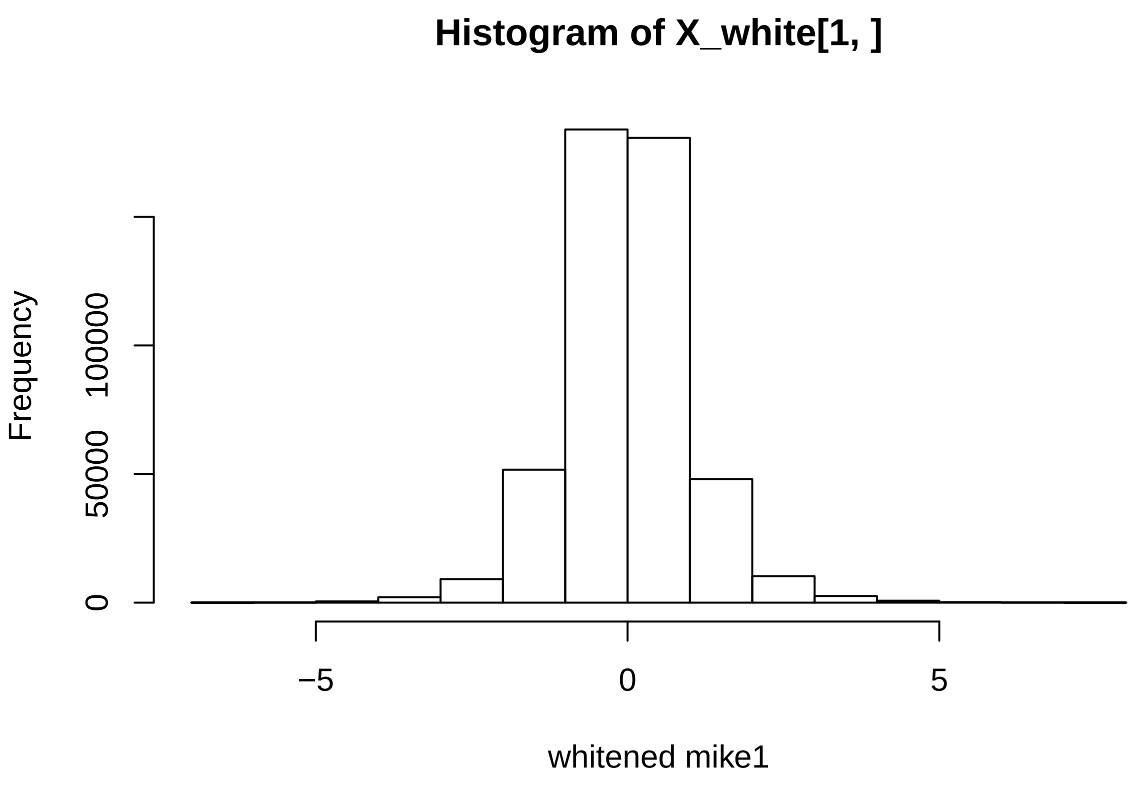
```
hist(X[2,], xlab='mike2')
```



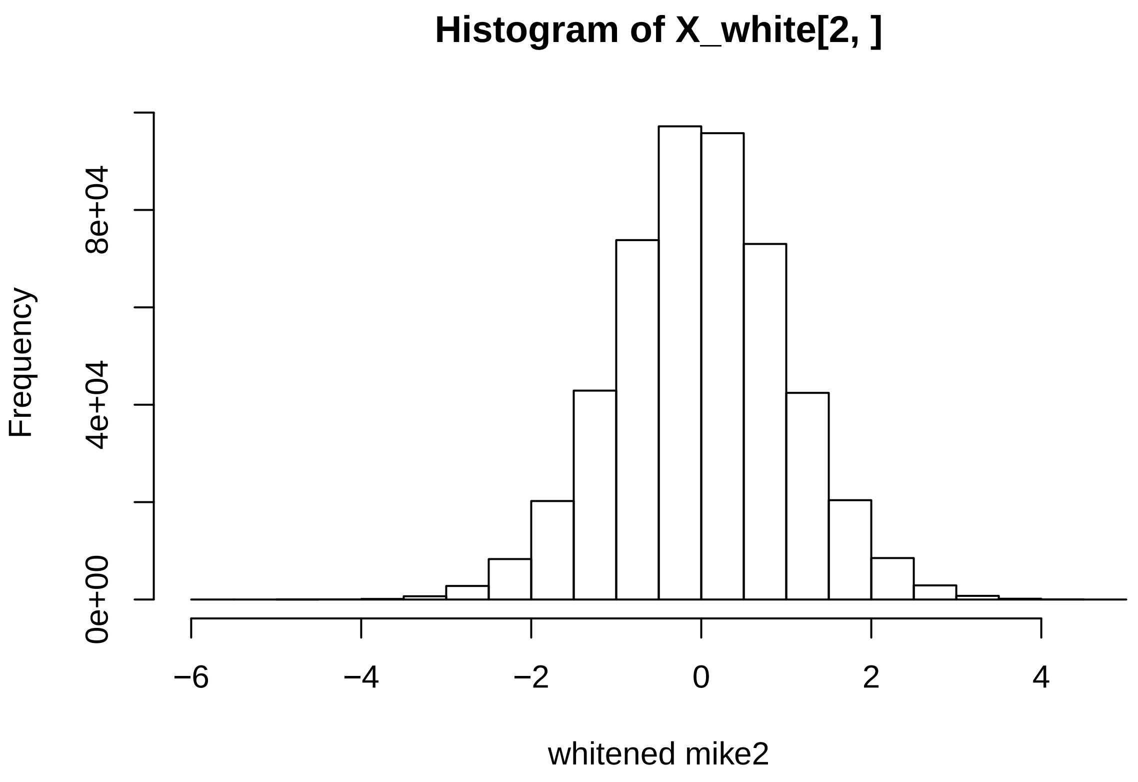
```r
hist(X[3,], xlab='mike3')
```



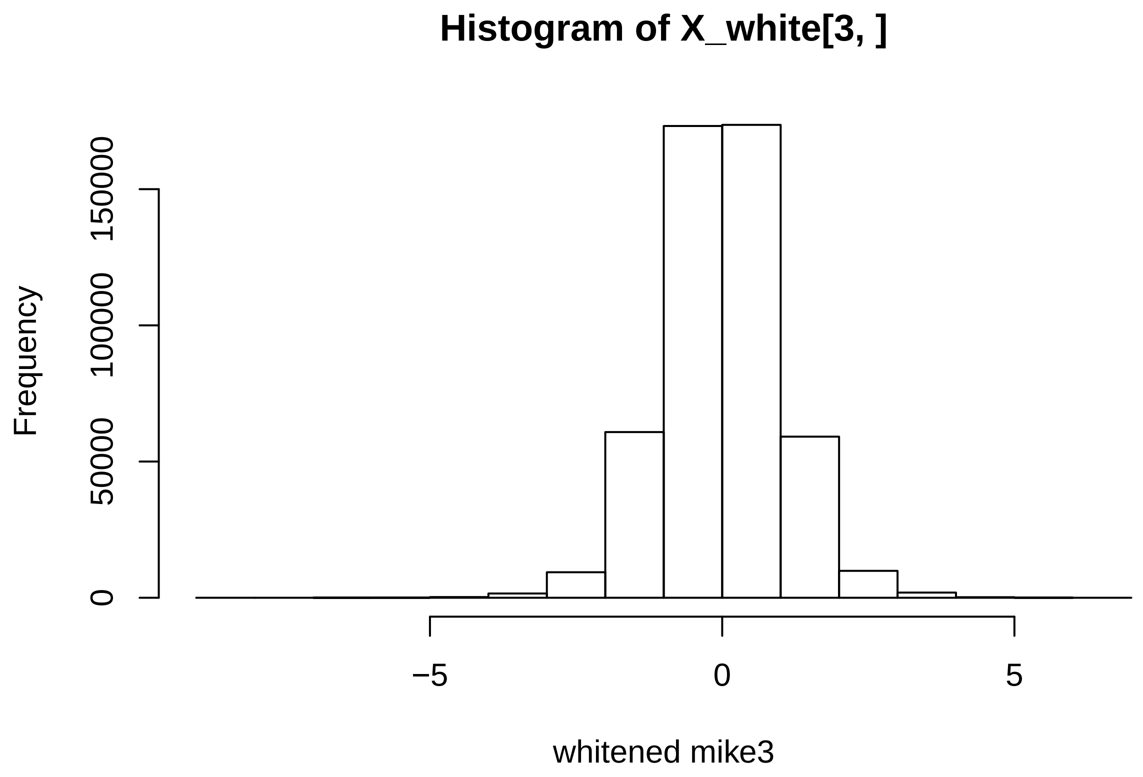
```r
# whitened data
hist(X_white[1,], xlab='whitened mike1')
```



```
hist(X_white[2,], xlab='whitened mike2')
```

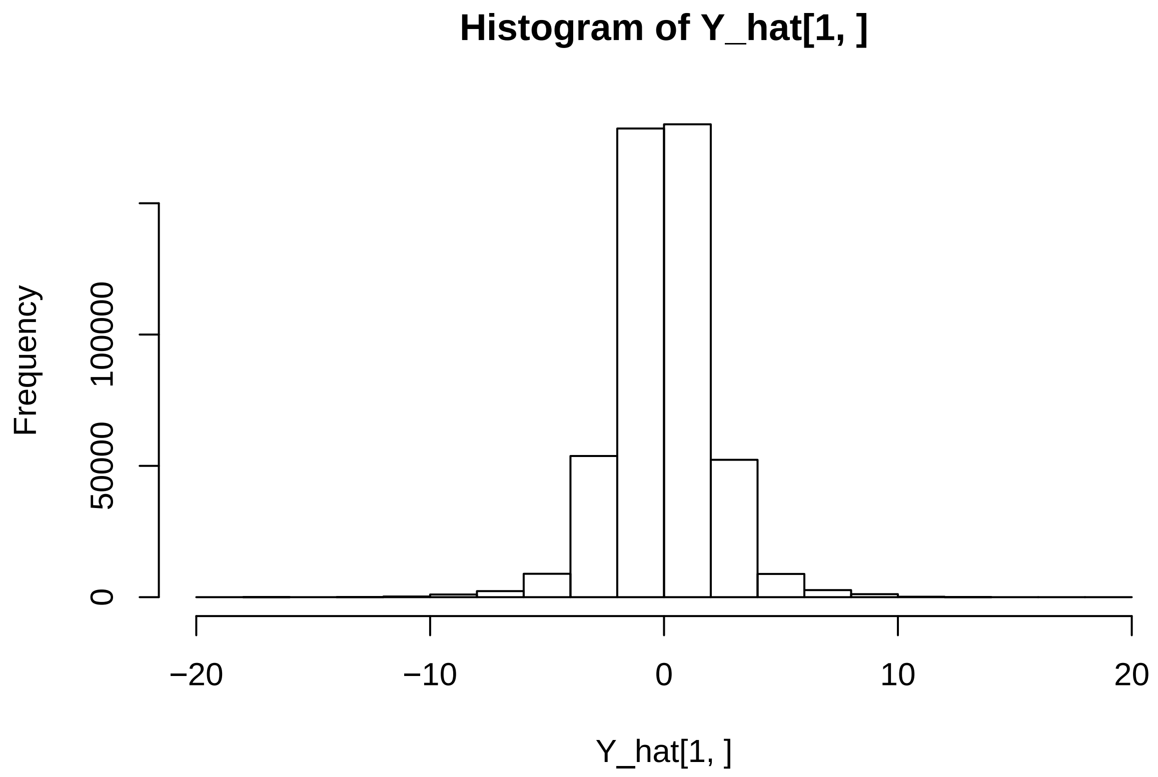


```
hist(X_white[3,], xlab='whitened mike3')
```

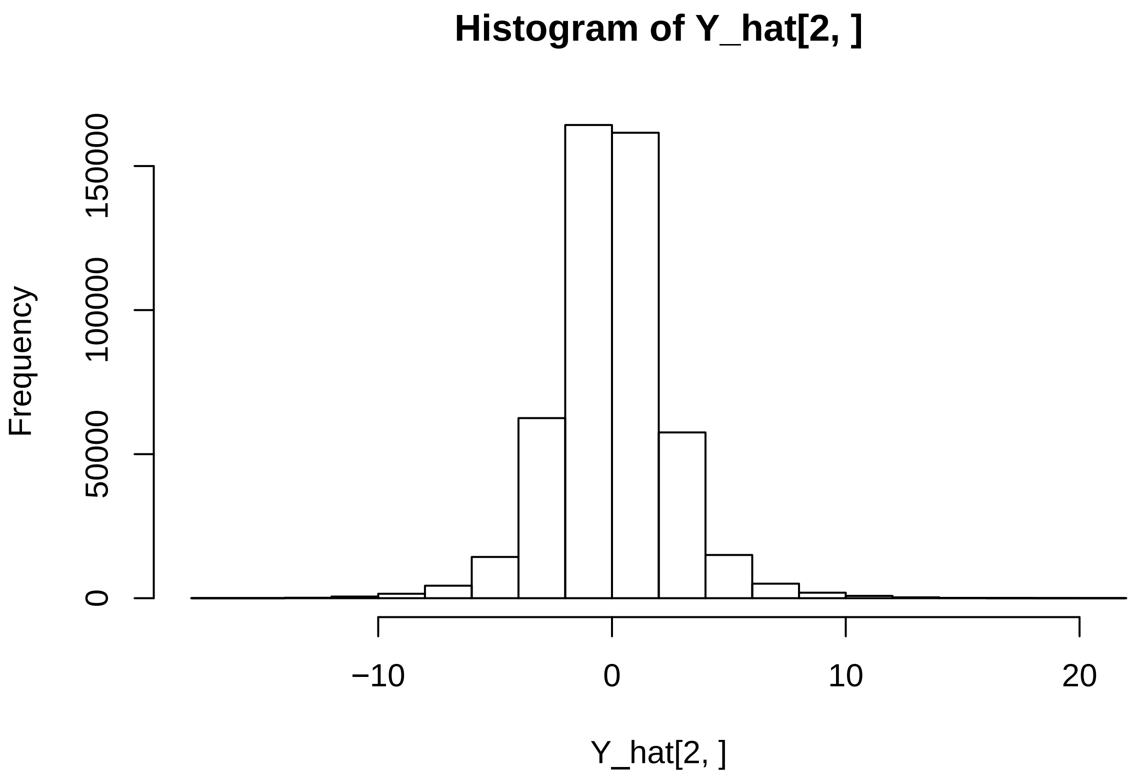


10.

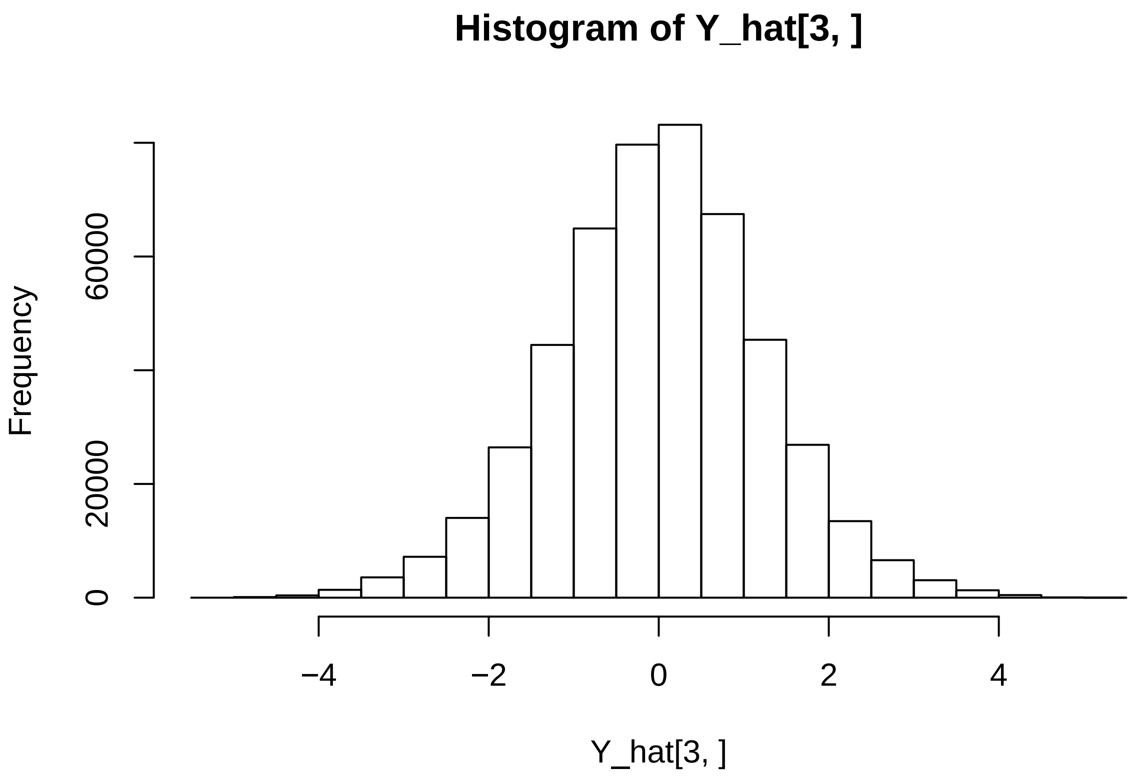
```
# histogram of marginals
Y_hat <- W_demixing %*% X
hist(Y_hat[1,])
```



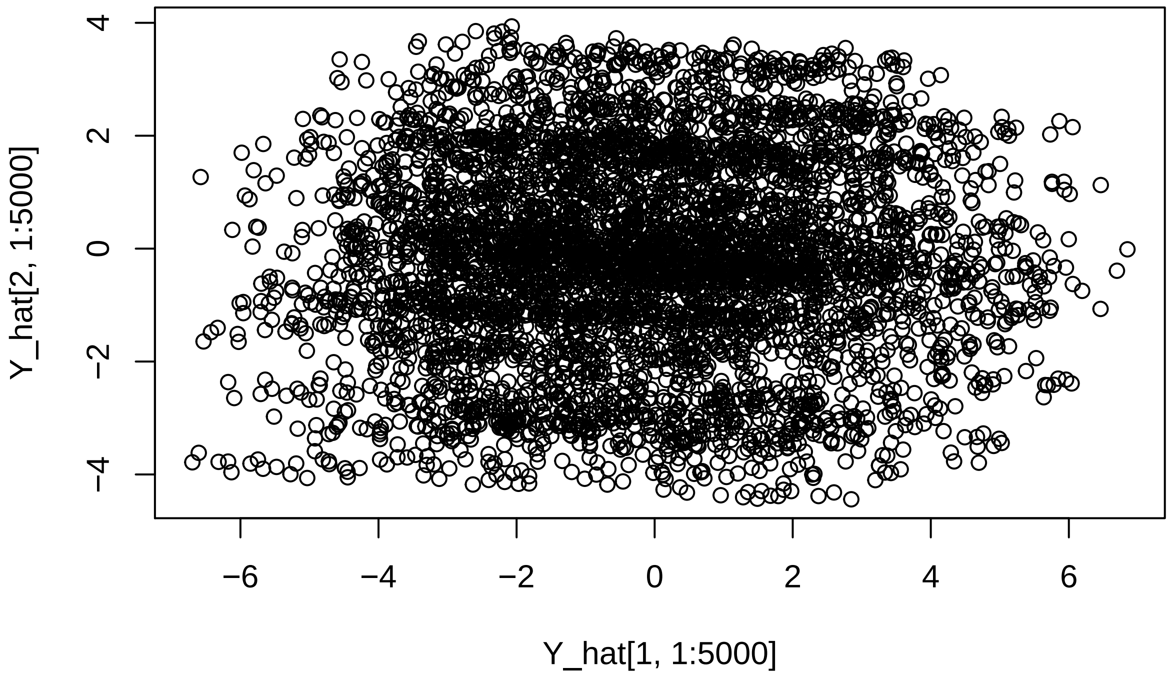
```r
hist(Y_hat[2,])
```



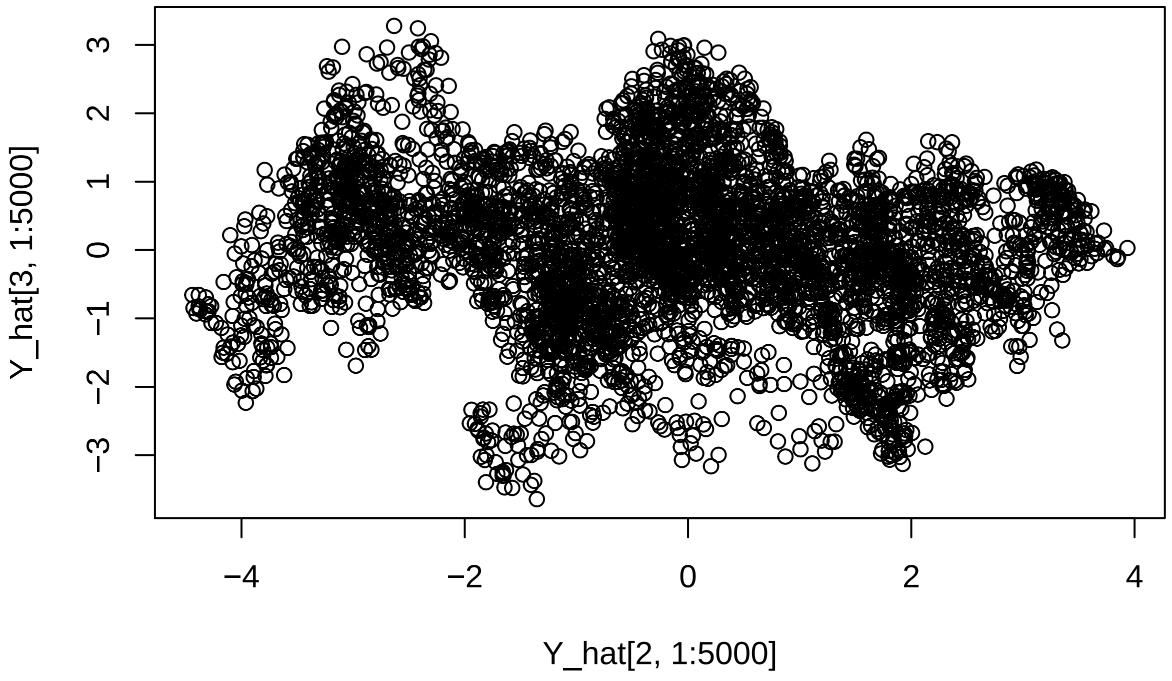
```r
hist(Y_hat[3,])
```



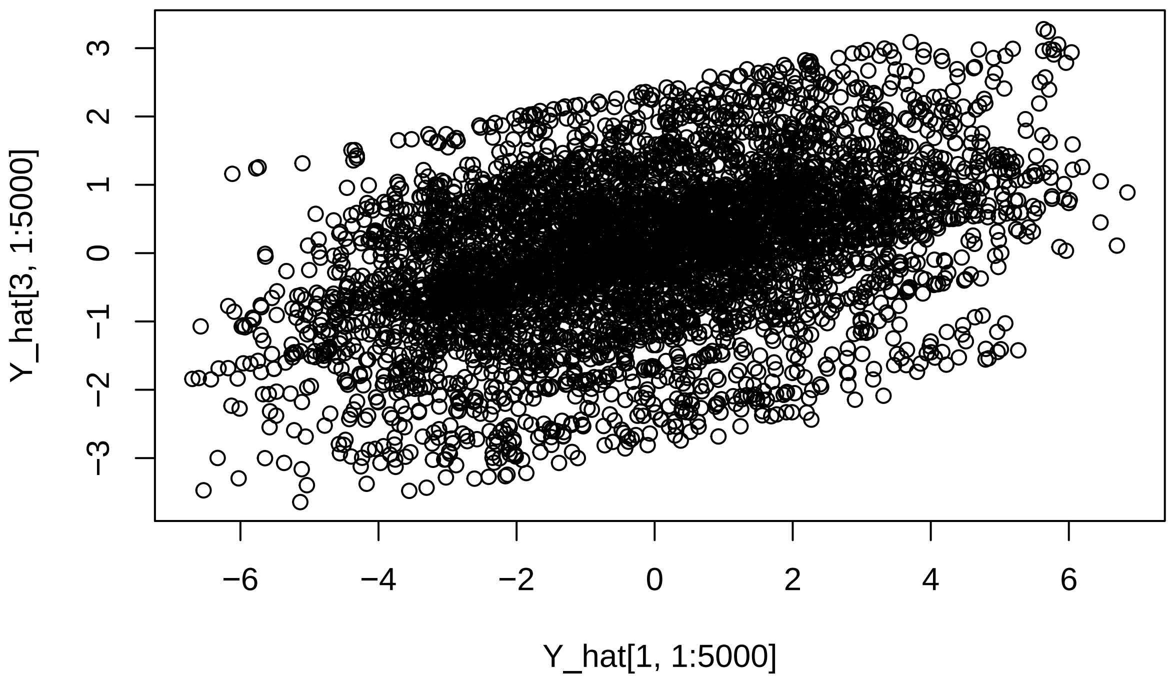
```r
# Use the first 5000 points to form the scatter plots
plot(Y_hat[1,1:5000], Y_hat[2,1:5000])
```



```r
plot(Y_hat[2,1:5000], Y_hat[3,1:5000])
```



```r
plot(Y_hat[1,1:5000], Y_hat[3,1:5000])
```



11. The estimate of $\mathbf{W}$ is not unique since the gradient ascent algorihtm can only converge to a local maxima. A different inital value for $\mathbf{W}$ will converge to a different local maxima.