Multivariate Statistics Week 09: Interim Summary

Noah Silbert

October 20, 2017

Random number generation & probability modeling

Generating random numbers is useful for simulating data and exploring how models work. `rnorm` generates univariate normal random variates, and `rmnorm` (from the `mnormt` library) generates multivariate normal random variates.

```
library(mnormt)
# sample size
N = 317
# univariate simulated data
m = 5; s = 4.6
x = rnorm(mean=m,sd=s,n=N)
# multivariate simulated data
mu = c(5,-2,10)
S = matrix(c(4.6,4,2,4,8.4,-3,2,-3,6),nrow=3)
X = rmnorm(mean=mu, varcov=S, n=N)
```

Probability Density Functions for normal variates

Recall that the univariate and multivariate normal PDFs are:

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$f(\mathbf{x}|\boldsymbol{\mu},\Sigma) = \frac{1}{\sqrt{2\pi}^p |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)$$

The R functions `dnorm()` and `dmnorm()` give the PDF values for univariate and multivariate normal data points, respectively. That is, for an input x value or $\mathbf{x}$ vector, these R functions give the values of the mathematical functions above.

dnorm() and dmnorm()

```
x[1]
```

```
## [1] 2.248041
```

```
dnorm(x=x[1],mean=m,sd=s)
```

```
## [1] 0.07251604
```

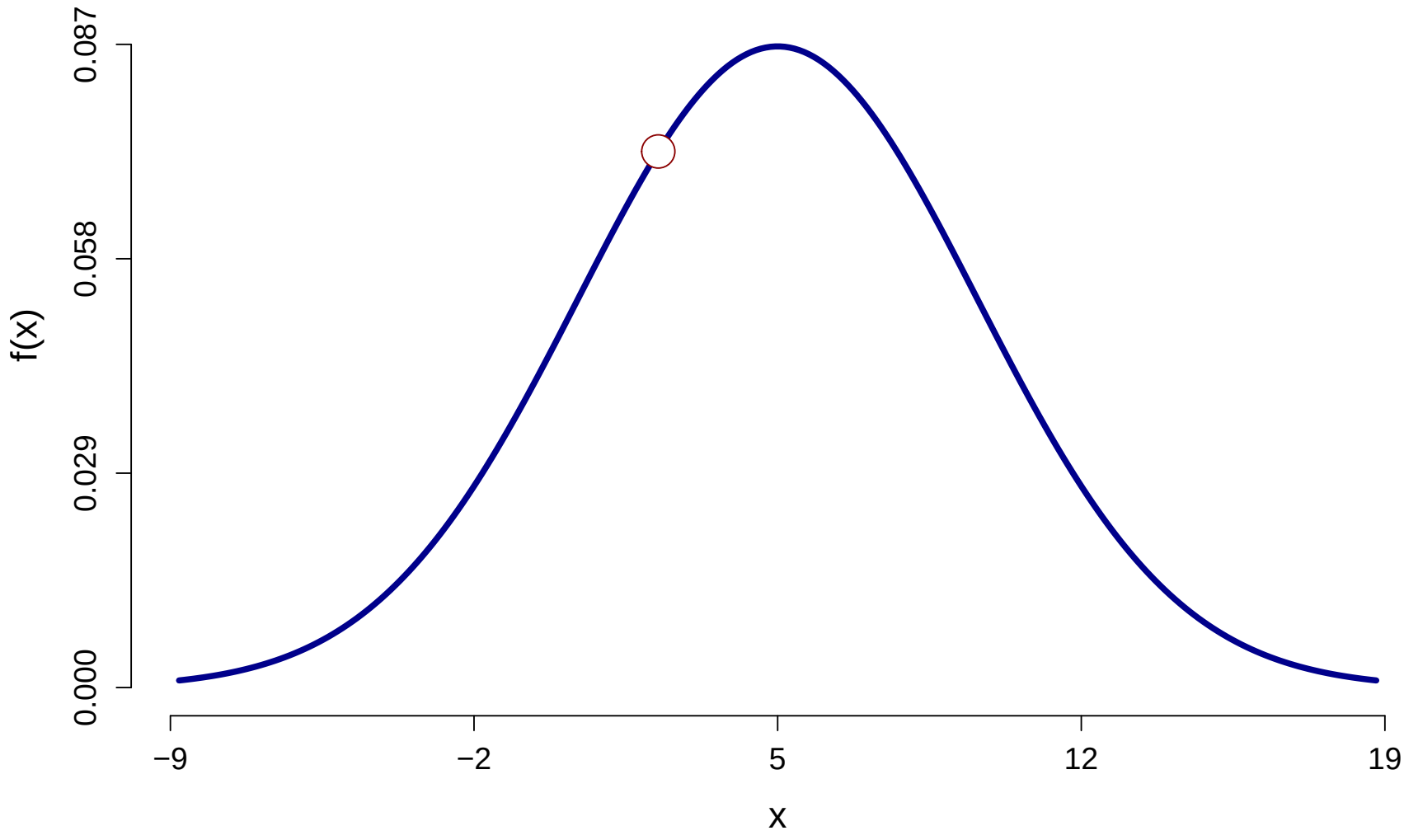
```
X[1,]
```

```
## [1]  5.874952 -0.607674 10.005612
```

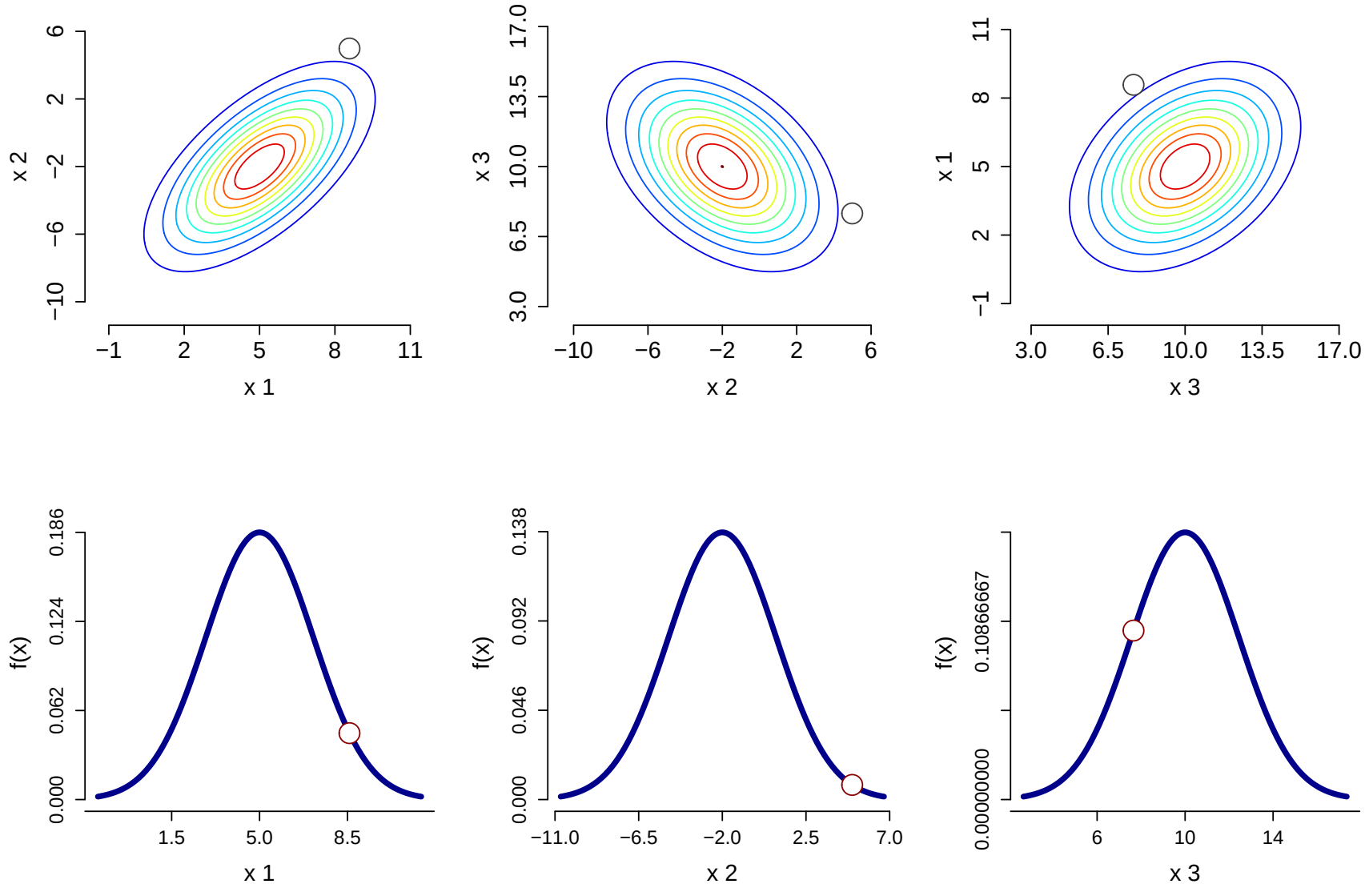
```
dmnorm(x=X[1,],mean=mu,varcov=S)
```

```
## [1] 0.01492386
```

Univariate visualization



Multivariate visualization



Probabilities & normal variates

The functions `pnorm()`, `pmnorm()`, and `sadmvn()` calculate integrals of normal PDFs (i.e., probabilities that these variates are in certain ranges).

For example, $Pr(x < 4|\mu = 5, \sigma = 4.6)$:

```
pnorm(4, mean=m, sd=s)
```

```
## [1] 0.4139517
```

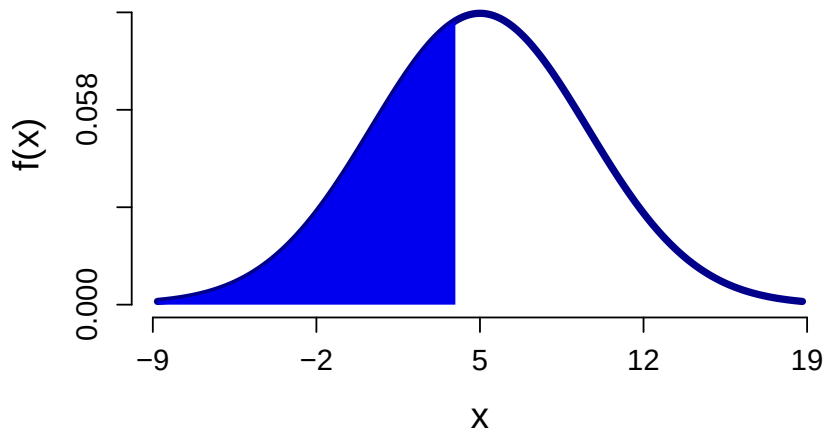
Or $Pr(4 < x < 7|\mu = 5, \sigma = 4.6)$:

```
pnorm(7, mean=m, sd=s) - pnorm(4, mean=m, sd=s)
```

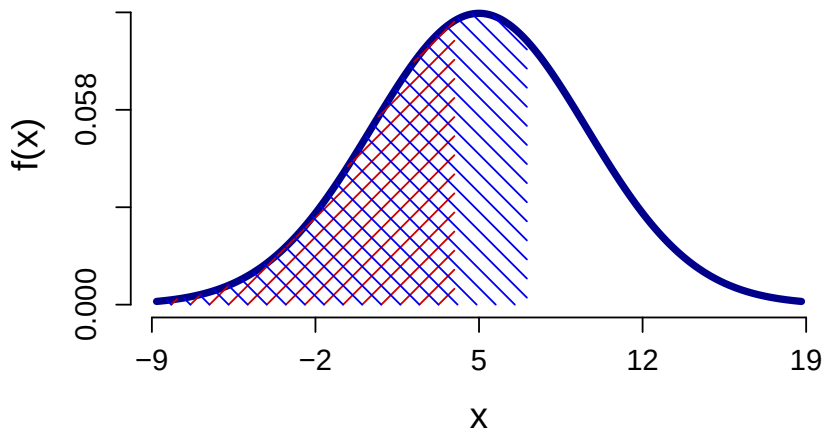
```
## [1] 0.2541882
```

Visualization

$Pr(x < 4|\mu = 5, \sigma = 4.6)$



$Pr(4 < x < 7|\mu = 5, \sigma = 4.6)$



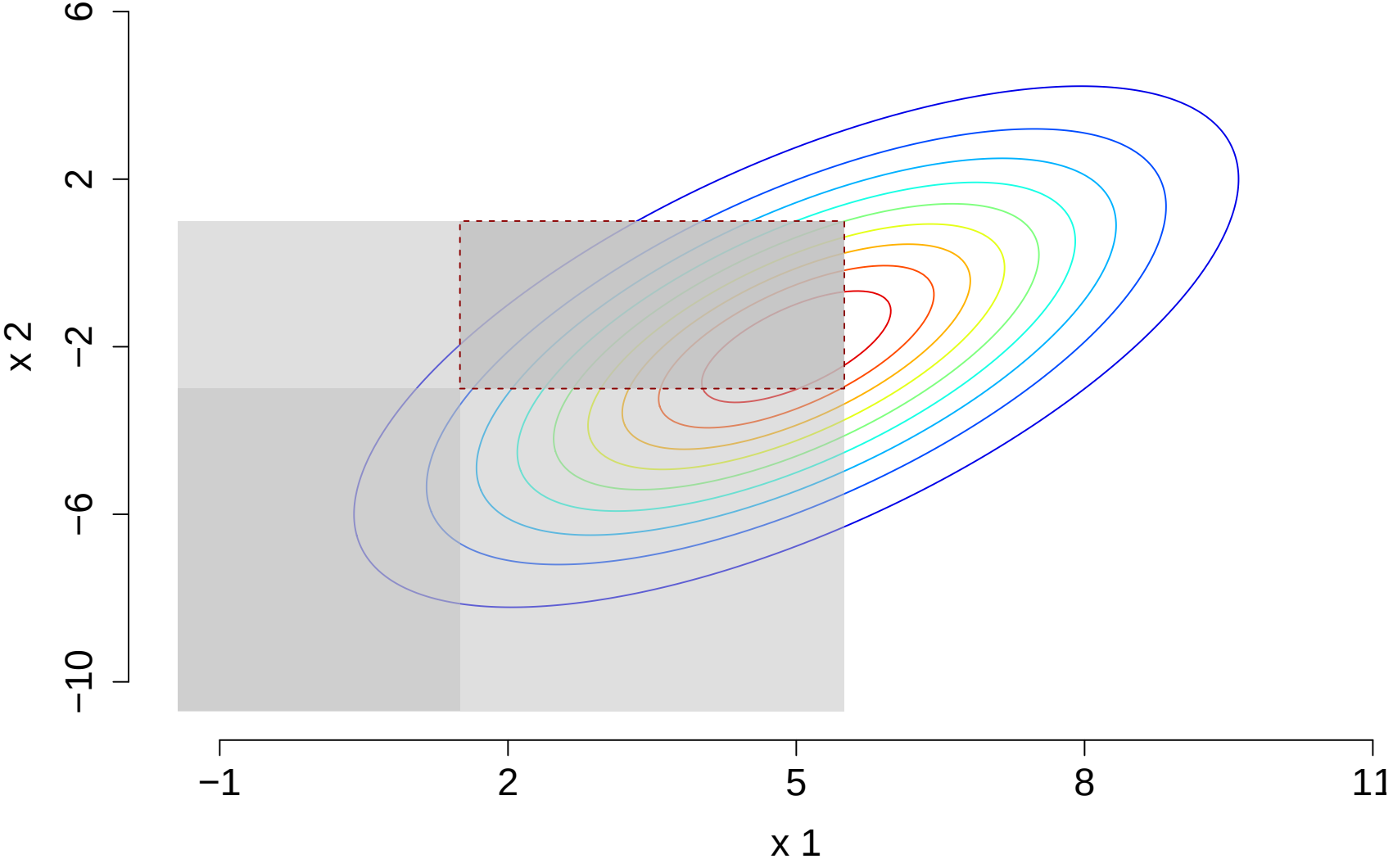
Multivariate probabilities

For the sake of simplicity, we'll focus just on x_1 and x_2 here. We can calculate the probability that a 2D normal variate falls below certain values on each dimension with `pmnorm()`, and we can calculate the probability of 2D normal variates falling in a rectangular region with `sadmvn()`:

```
lower = c(1.5,-3)
upper = c(5.5,1)
pr.low = pmnorm(x=lower,mean=mu[1:2],varcov=S[1:2,1:2])
pr.upp = pmnorm(x=upper,mean=mu[1:2],varcov=S[1:2,1:2])
pr.rct = sadmvn(lower=lower, upper=upper,
                mean=mu[1:2], varcov=S[1:2,1:2])
print(c(pr.low,pr.upp,pr.rct))
```

```
## [1] 0.04563992 0.56606029 0.24741026
```

Visualization



Expected values and means

We can summarize the central tendency of a distribution with the expected value. The expected value of a random variable X is denoted $\mathbb{E}[X]$. For a continuous random variable:

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} xf(x)dx$$

For a discrete random variable:

$$\mathbb{E}[X] = \sum_{i=-\infty}^{\infty} x_i p(x_i)$$

For a particular sample of a univariate random variable $x_i, i \in 1, 2, \ldots, N$, we approximate $p(x)$ with $\frac{1}{N}$, and calculate the (sample) mean as $\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{N} = \frac{1}{N} \sum_{i=1}^{N} x_i$.

Multivariate means

With multivariate data, each observation is a vector, so instead of a single expected value, we get a vector of expected values: one for each dimension of the multivariate random variable.

We simulated 317 observations of univarite and multivariate (3D) normal variables above (`x` and `X`, respectively). We can calculate means in R with `mean()` and `colMeans()`:

```
mean(x)
```

```
## [1] 4.681032
```

```
colMeans(X)
```

```
## [1]  4.899620 -2.135009  9.981729
```

Variance

Variance is the expected value of the squared deviation of the data from the mean:

$$\mathbb{E}[(x-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)dx \qquad \text{continuous}$$

$$= \sum_{i=-\infty}^{\infty} (x_i-\mu)^2 p(x) \qquad \text{discrete}$$

Standard deviation is the square root of the variance.

For a univariate sample, we can estimate the variance as:

$$\mathrm{Var}(x) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2$$

Covariance

With multivariate data, we have variances for each variable and *covariances* between variables. The covariance for two random variables X and Y is the expected value of the cross-product of their deviations:

$$\mathrm{Cov}(X, Y) = \sigma_{xy} = \mathbb{E}[(X - \mathbb{E}(X))(Y - \mathbb{E}(Y))]$$

The sample covariance matrix for a data matrix X can be expressed in linear algebra notation as follows ($\mathbf{I}$ and $\mathbf{J}$ are $N \times N$):

$$\Sigma = \frac{1}{N-1}\mathbf{X}^T\left(\mathbf{I} - \frac{1}{N}\mathbf{J}\right)\mathbf{X}$$

Correlation and covariance are related by:

$$\rho_{xy} = \frac{\sigma_{xy}}{\sqrt{\sigma_{xx}\sigma_{yy}}}$$

Covariance, correlation, and variance in R

We can get sample variances and covariance matrices with `var()` and `cov()`, and we can get a correlation matrix with `cor()`:

```
round(var(x),2)
```

```
## [1] 21.15
```

```
round(cov(X),2)
```

```
##      [,1]  [,2]  [,3]
## [1,] 4.08  3.28  2.07
## [2,] 3.28  7.79 -3.47
## [3,] 2.07 -3.47  6.76
```

Randomness

A data vector is *random* if there is uncertainty about the values that it takes. A particular probabilistic model (e.g., multivariate normal) makes claims about the probability that a random vector takes certain values.

Summary statistics (e.g., means, [co]variances) are functions of random data, so they are also random (typically with known, theoretically derived statistical properties).

Other quantities that are functions of random data are also random. Thus far, we've discussed distances (proximities), principal components, and discriminant analysis. In each case, we've made extensive use of functions of random vectors.

Random distances

The Euclidean distance between two points (vectors) $\mathbf{x}$ and $\mathbf{y}$ is:

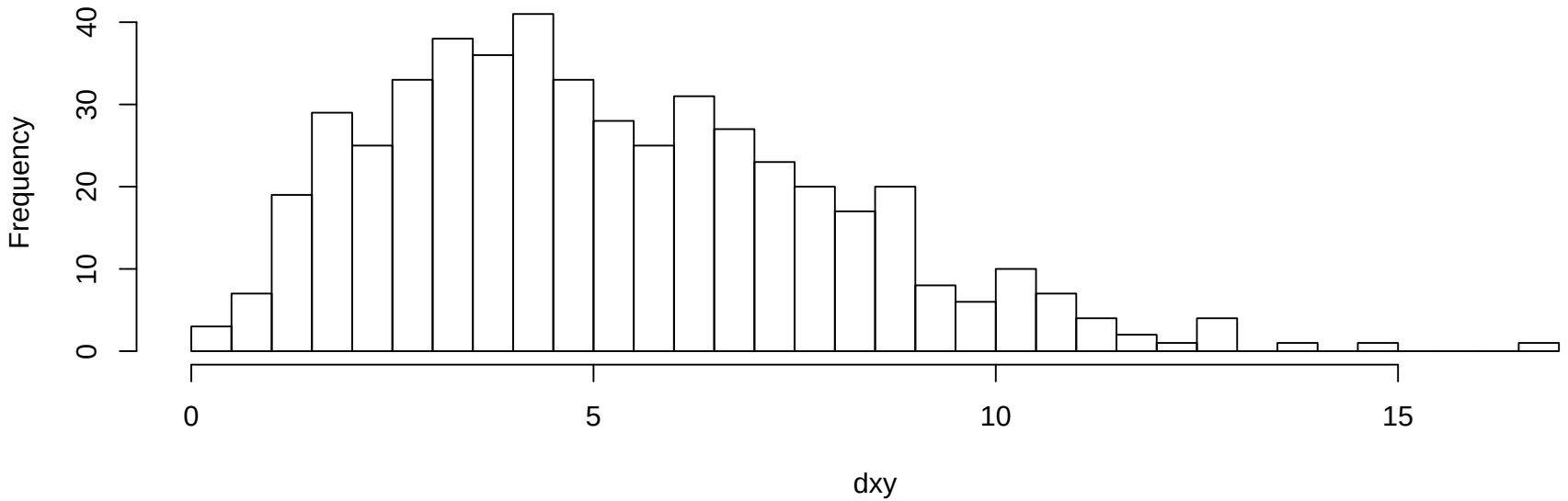
$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y})}$$
$$= \sqrt{\sum_{i=1}^{P} (x_i - y_i)^2}$$

If the vectors $\mathbf{x}$ and $\mathbf{y}$ are random, then the distance between the vectors is also random.

The distribution of random distances can be worked out theoretically (under various assumptions about how the random vectors are distributed), or, more conveniently, we can easily simulate random data, calculate (random) distances between simulated points, and visualize the result.

Simulating and visualizing random distances

```
n.d = 500
dxy = vector(length=n.d)
for(i in 1:n.d){
  xx = rmnorm(mean=mu, varcov=S, n=1)
  yy = rmnorm(mean=mu, varcov=S, n=1)
  dxy[i] = sqrt(t(xx-yy) %*% (xx-yy))
}
hist(dxy,breaks=50,main="")
```



Statistical distance and normal random variables

If $\mathbf{x}$ and $\mathbf{y}$ are identically distributed, and if we can estimate the covariance matrix governing them, we can use the *statistical* distance between them:

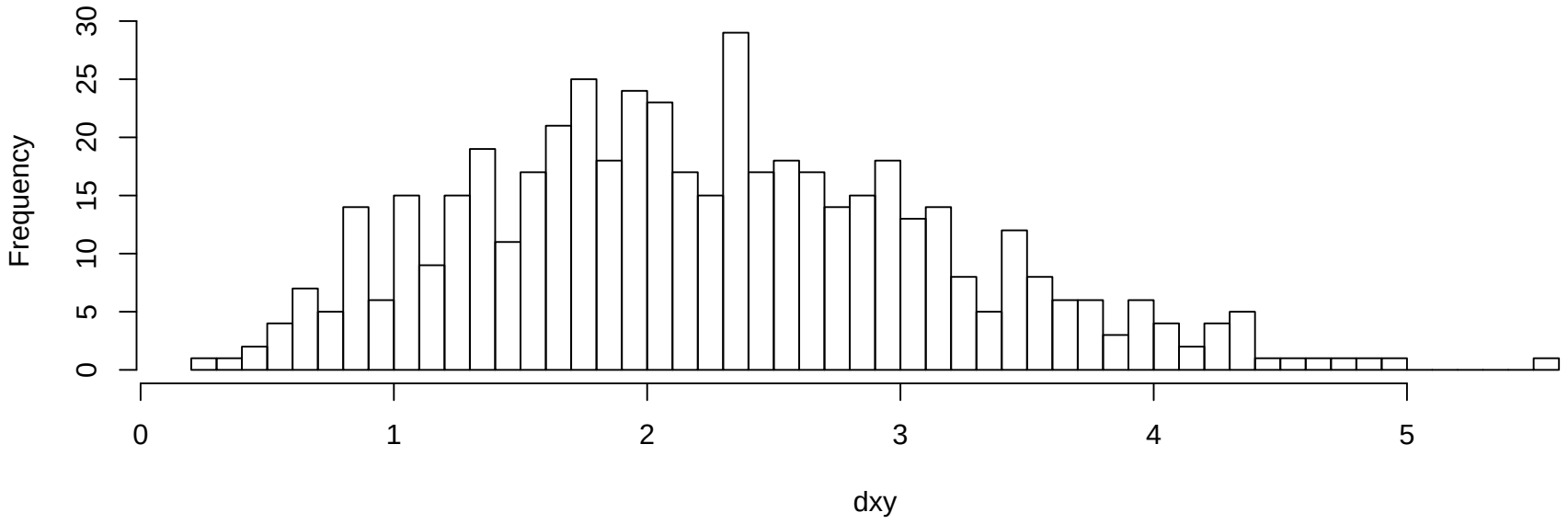
$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{y})}$$

The statistical distance looks a lot like the exponentiated term in the multivariate normal pdf:

$$f(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{2\pi}^p |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right)$$

Simulating and visualizing statistical distances

```
dxy = vector(length=n.d)
for(i in 1:n.d){
  xx = rmnorm(mean=mu, varcov=S, n=1)
  yy = rmnorm(mean=mu, varcov=S, n=1)
  dxy[i] = sqrt(t(xx-yy) %*% solve(S) %*% (xx-yy))
}
hist(dxy,breaks=50,main="")
```



MDS & Clustering

We used multidimensional scaling and a number of clustering algorithms to "smooth" some of the randomness out of data.

MDS takes a set of (random) proximities and finds a set of coordinates in relatively low-dimensional space such that the proximities in the low-dimensional space correspond closely to the original proximities.

Hierarchical clustering algorithms take a set of (random) proximities and merge individual observations into clusters. Different algorithms use different rules for calculating inter-cluster distances.

K-means clustering iteratively lumps observations together into clusters based on (random) proximities.

In each case, some of the randomness in the original proximities is neglected in order to (try to) obtain a simpler, easy to interpret representation of the data.

PCA

Linear combinations of random variables are also random. One implication of this is that principal components inherit randomness from the variables they are based on.

The linear combination is defined by the elements of the eigenvectors of the covariance or correlation matrix:

```
Y = matrix(0,nrow=nrow(X),ncol=ncol(X))
for(i in 1:ncol(Y)){
  m.t = mean(X[,i]); s.t = sd(X[,i])
  Y[,i] = (X[,i]-m.t)/s.t
}
R = cor(X)
e = eigen(R)
```

The principal components are the dimensions along which *statistical distance* is defined (i.e., $\mathbf{S}^{-1}$ adjusts distances to take covariances into account).

PCA continued

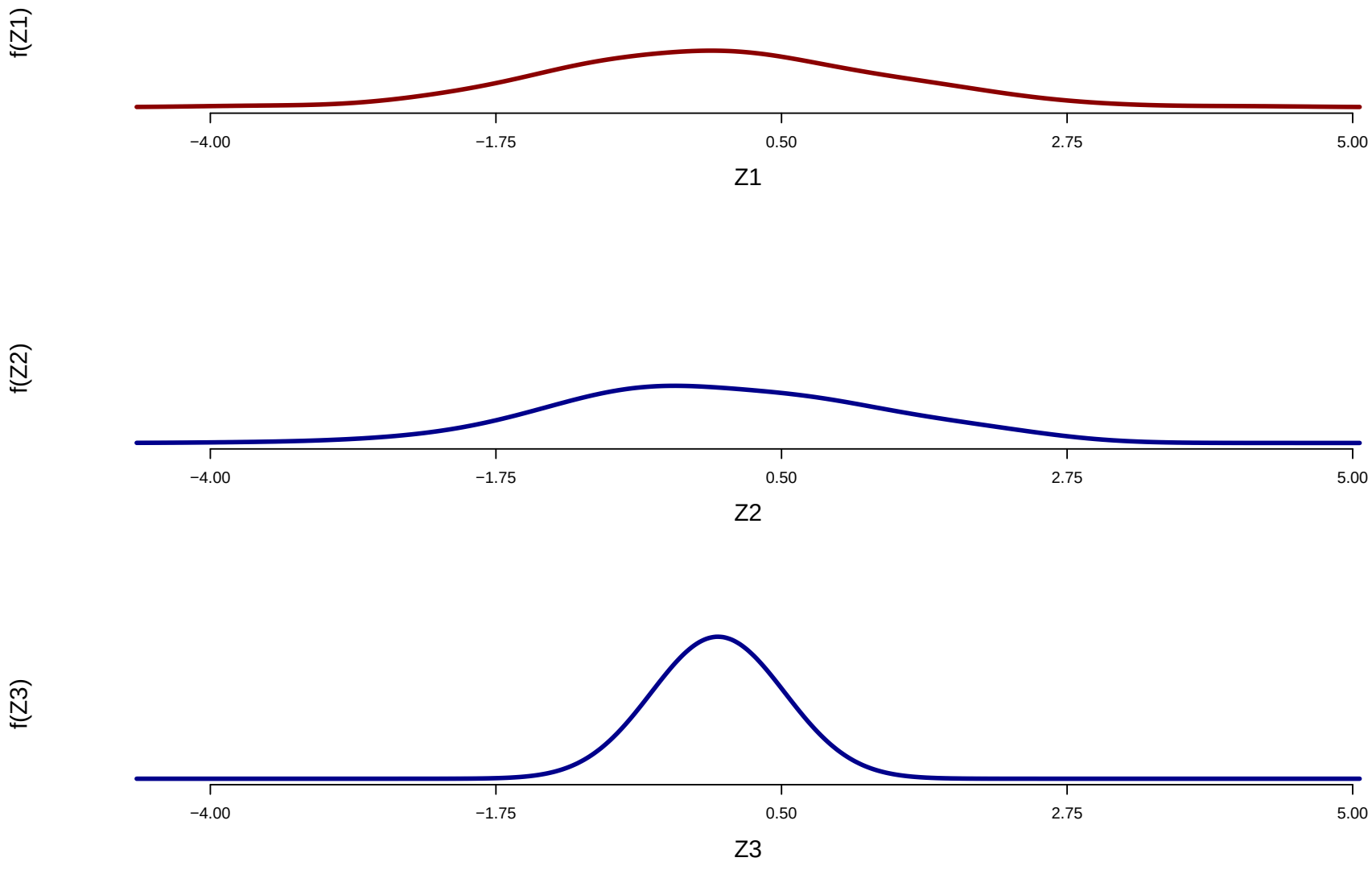
We can easily visualize the randomness inherited by the principal components:

```
# extract eigenvalues, see how many components are useful
lam = e$values
lam/sum(lam)
```

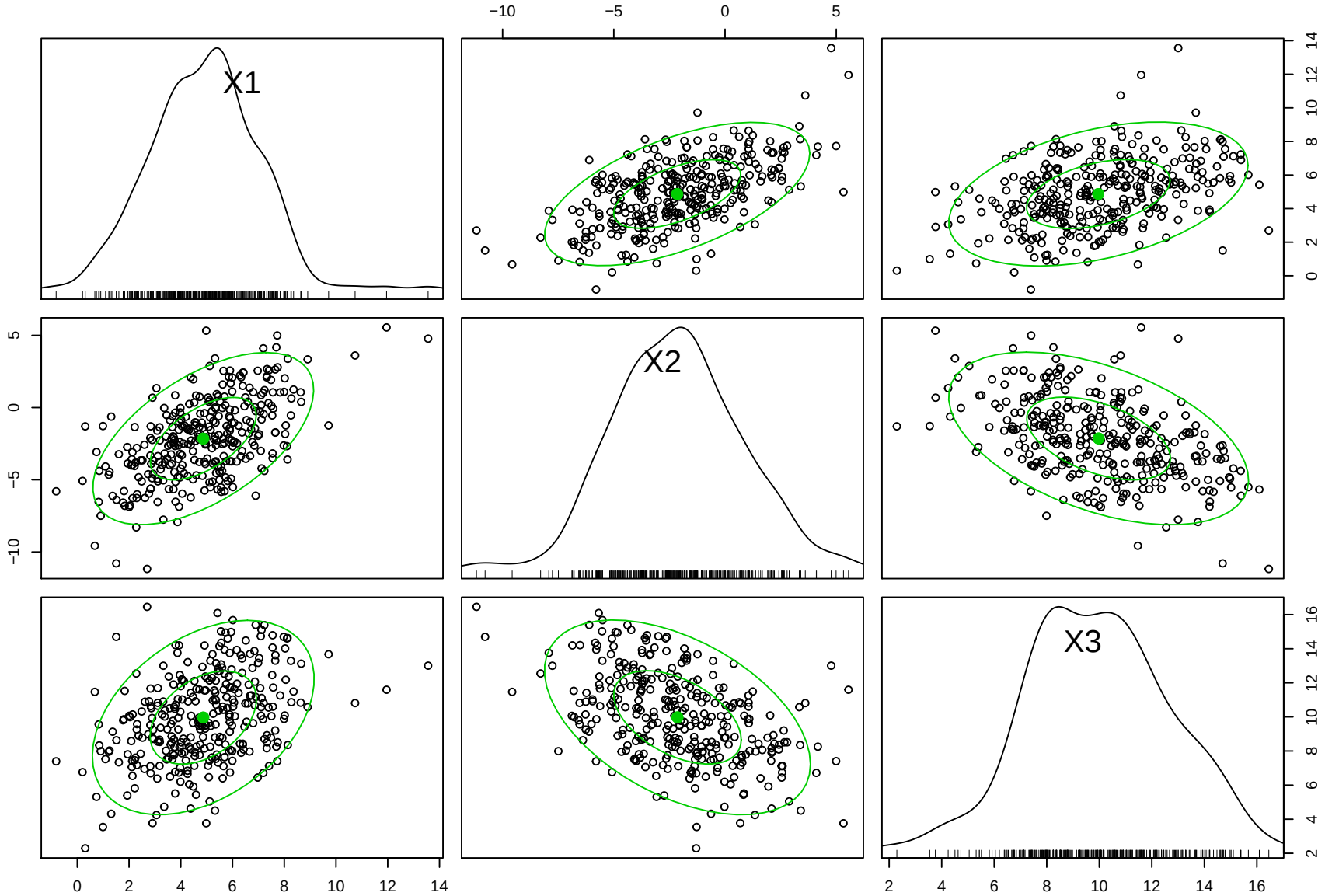
```
## [1] 0.531519982 0.459757806 0.008722212
```

```
# extract eigenvectors, transform data
V = e$vectors
Z = Y %*% V
```

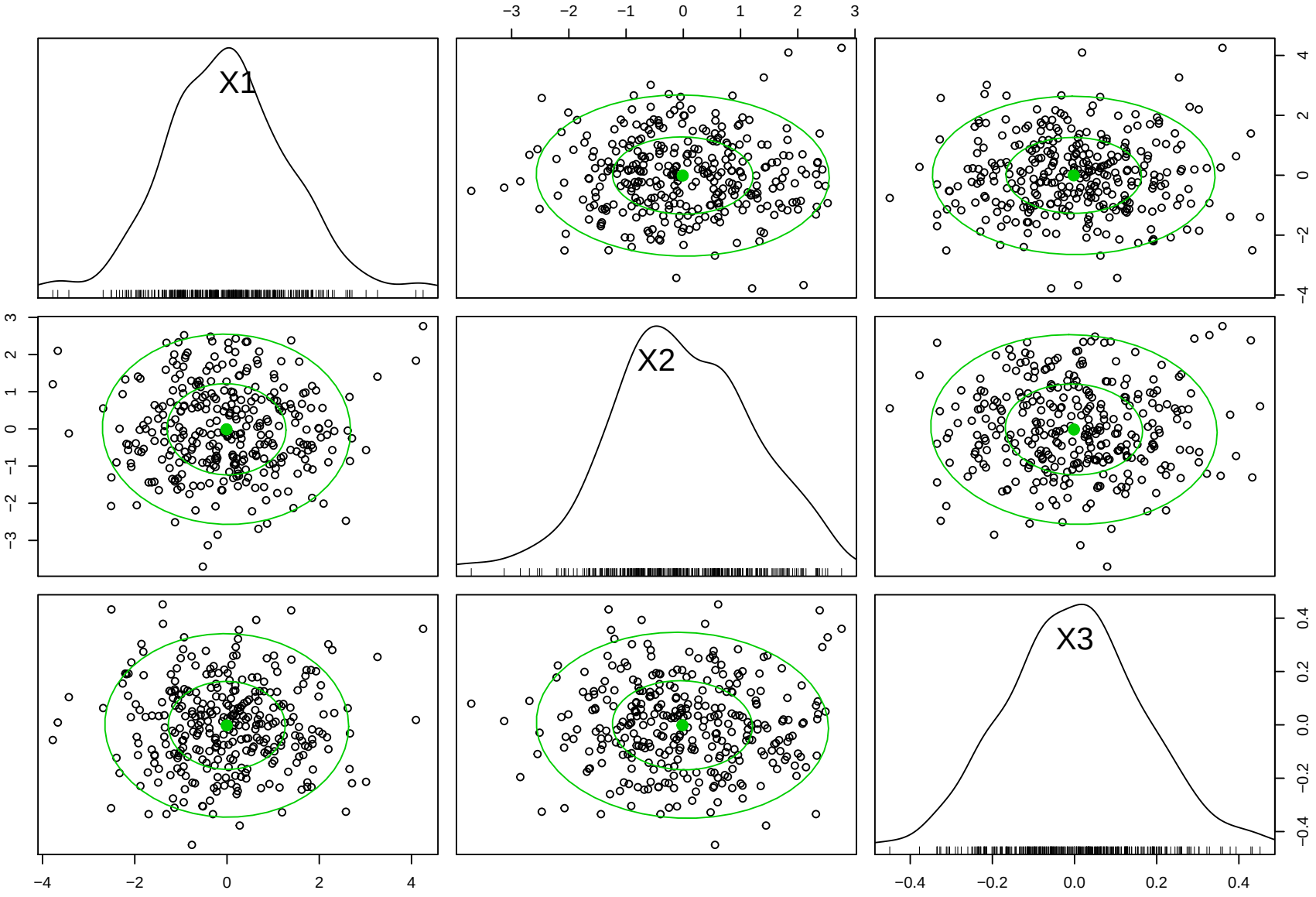
PCA continued continued



Original data



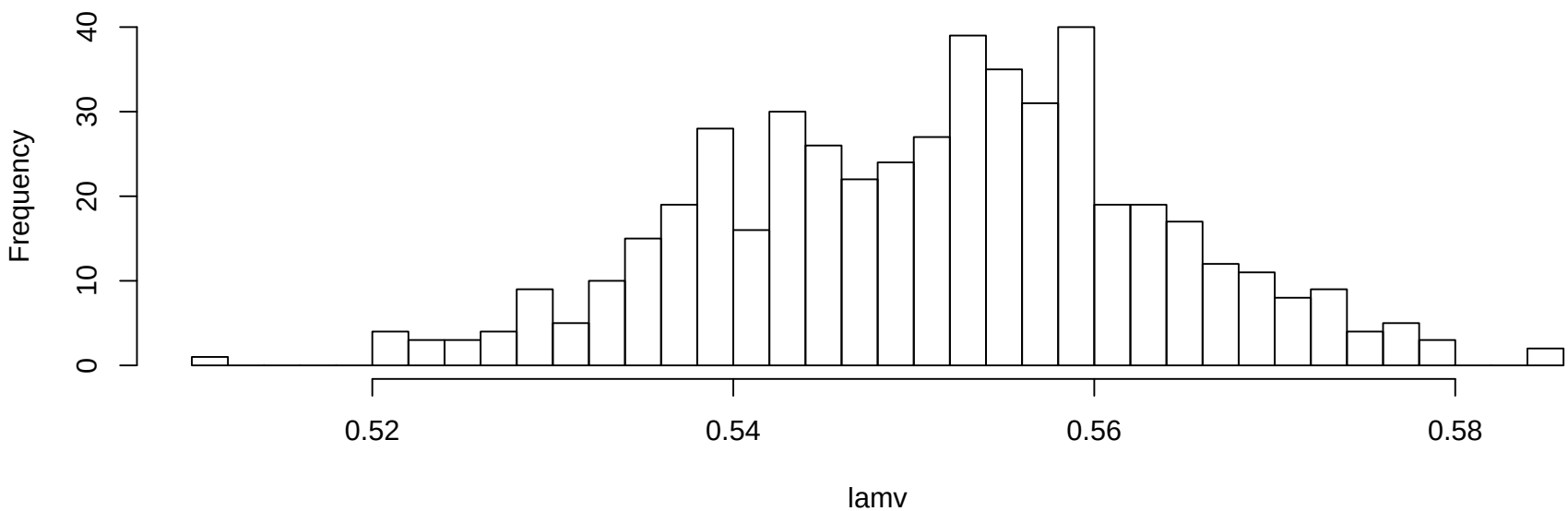
Principal Components



Randomness all the way down

Note that the eigenvalues and eigenvectors *also* inherit randomness from the data:

```
lamv = vector(length=500)
for(i in 1:500){
  Xt = rmnorm(mean=mu,varcov=S,n=N)
  lt = eigen(cor(Xt))$values
  lamv[i] = lt[1]/sum(lt)
}
hist(lamv,breaks=50,main="")
```



Discriminant analysis

Discriminant analysis with two groups finds a vector $\mathbf{a}$ such that the normalized distance between transformed group means is maximized:

$$\frac{(\bar{z}_1 - \bar{z}_2)^2}{s_z^2} = \frac{\left[\mathbf{a}^T (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)\right]^2}{\mathbf{a}^T \mathbf{S}_{pl} \mathbf{a}}$$

This distance is maximized when $\mathbf{a} = \mathbf{S}_{pl}^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$. If we substitute this expression for $\mathbf{a}$ into the equation above, we get:

$$\frac{(\bar{z}_1 - \bar{z}_2)^2}{s_z^2} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T \mathbf{S}_{pl}^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$$

More than two groups

To extend to $k > 2$ groups, we can use "between group" and "within group" scatter matrices (i.e., non-scaled covariance matrices) $\mathbf{H}$ and $\mathbf{E}$:

$$\mathbf{H} = n \sum_{i=1}^{k} (\bar{\mathbf{x}}_{i\cdot} - \bar{\mathbf{x}}_{\cdot\cdot})(\bar{\mathbf{x}}_{i\cdot} - \bar{\mathbf{x}}_{\cdot\cdot})^T$$

$$\mathbf{E} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \bar{\mathbf{x}}_{i\cdot})(\mathbf{x}_{ij} - \bar{\mathbf{x}}_{i\cdot})^T$$

It's more opaque than the two group case, but with more than two groups, we maximize group separation by finding the eigenvalues and eigenvectors of $\mathbf{E}^{-1}\mathbf{H}$.

Randomness in LDA

As with PCA, the linear transformations in LDA produce new random variables (i.e., the discriminant functions inherit randomness from the data).

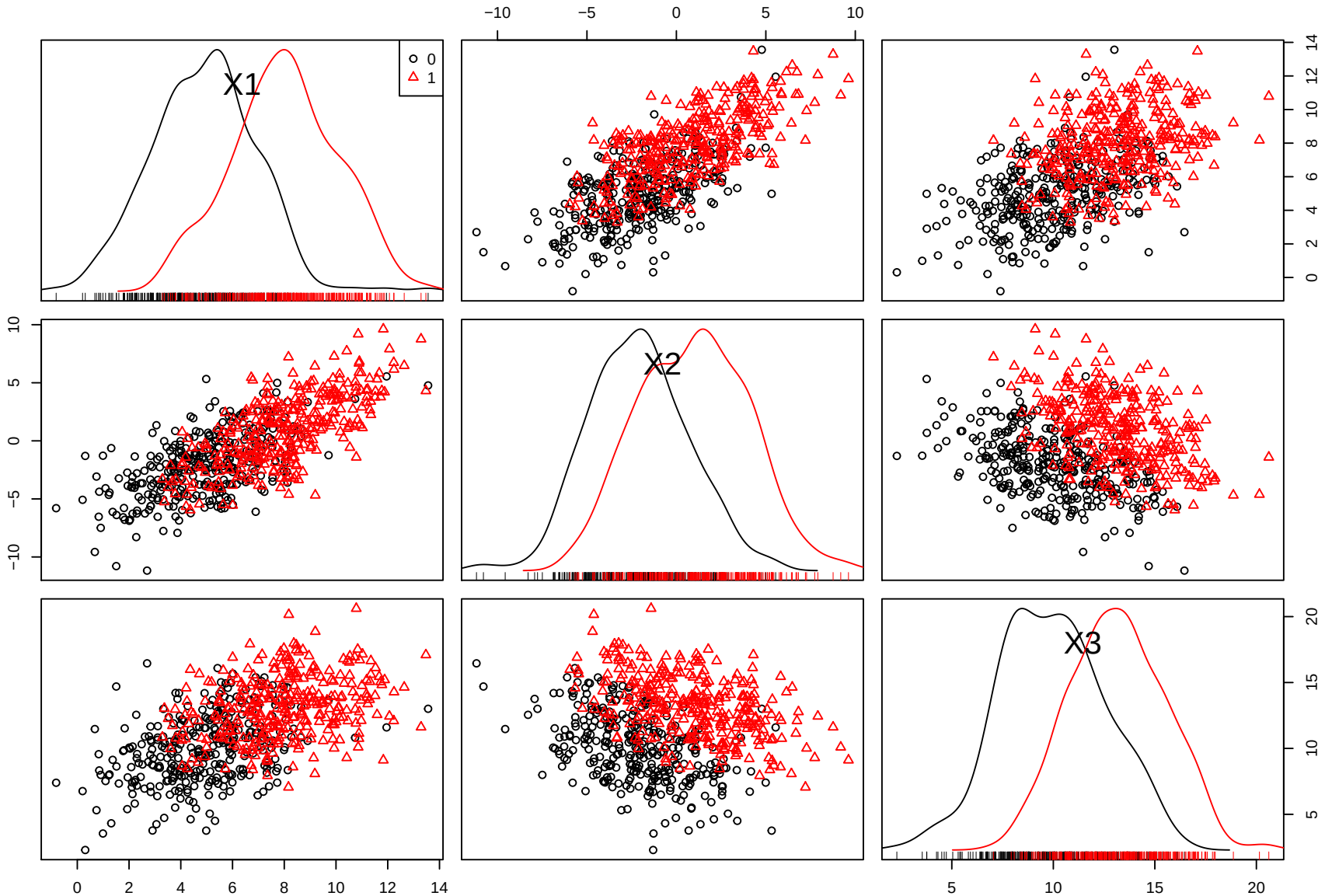
```
# more data for a second group
Y = rmnorm(mean=mu+3,varcov=S,n=N)

# combine X and Y into a single matrix
XY = rbind(X,Y)

# Group indicator
G = c(rep(0,N),rep(1,N))
```

Visualization

```
scatterplotMatrix(XY,smoother=F,reg.line=F,groups=G)
```



LDA

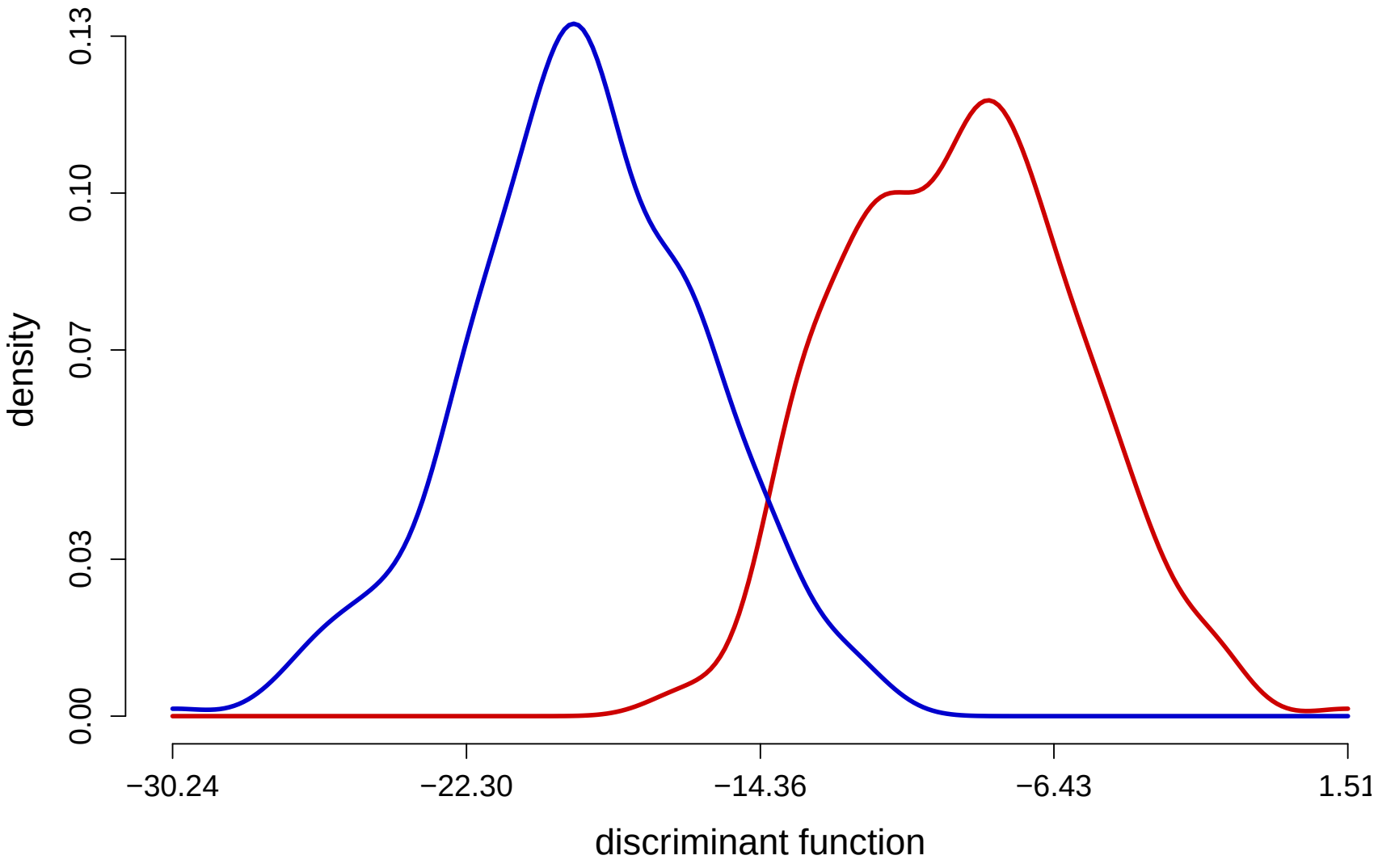
```
# mean vectors for each group
mu1 = colMeans(X); mu2 = colMeans(Y)

# pooled covariance matrix
W1 = (N-1)*cov(X)
W2 = (N-1)*cov(Y)
Spl = (1/(2*N-2))*(W1+W2)

# transformation vector a
a = solve(Spl) %*% (mu1-mu2)

z1 = X %*% a
z2 = Y %*% a
```

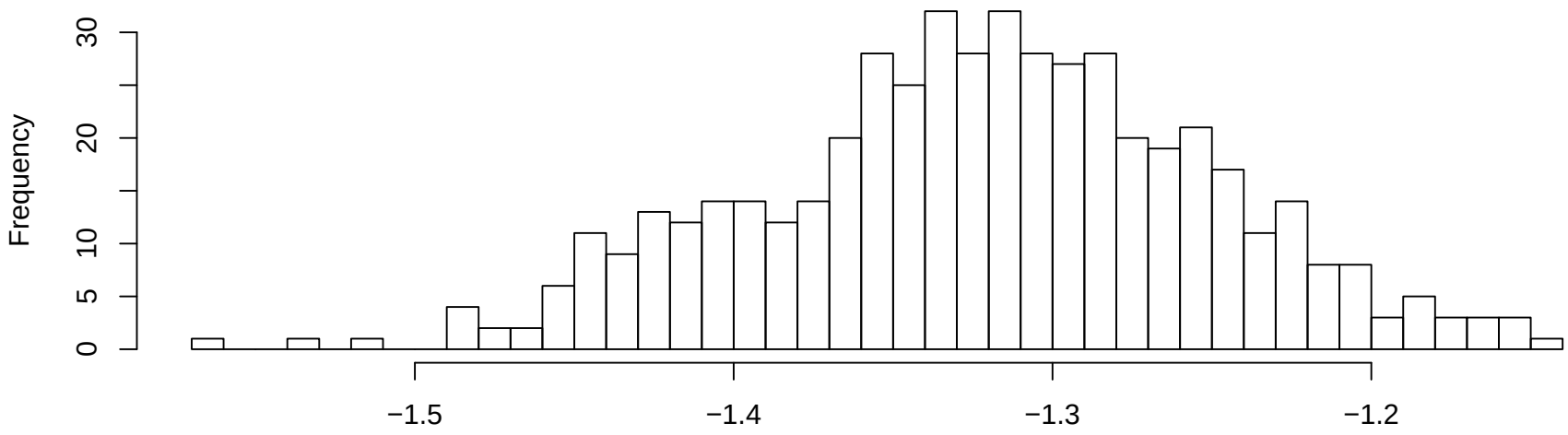
Transformed data



Still more randomness

As with PCA, LDA coefficients also vary from sample to sample:

```
library(MASS)
cfm = matrix(0,nrow=500,ncol=3)
for(i in 1:500){
  Xt = rmnorm(mean=mu,varcov=S,n=N)
  Yt = rmnorm(mean=mu+3,varcov=S,n=N)
  XYt = rbind(Xt,Yt)
  cfm[i,] = lda(XYt,grouping=G)$scaling
}
hist(cfm[,1],breaks=50,main="")
```



Conclusion

- Covariance, distance, and statistical distance are central to everything we've talked about so far: MDS, Clustering, PCA, LDA, and logistic regression.
- Variation propogates through models - randomness is inherited from data by coefficients, parameters, estimates, etc.
- Statistical programming gives us powerful tools for understanding and modeling variation and randomness.