Multivariate Stats, Week 7

Noah Silbert

October 5, 2017

To do

1. Review of homework
2. Review of and more about discriminant analysis
3. Logistic regression
4. Statistical Decision Theory and classification

Discriminant analysis review

Recall that with two groups $g = 1, 2$ and the assumption that the two groups have the same covariance matrix $\mathbf{\Sigma}$ and different mean vectors $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$, the *discriminant function* is a linear combination of the p variables that best separates the two groups (with respect to the variable created by the linear combination of the variables):

$$z_{gi} = \mathbf{a}^T \mathbf{x}_{gi} = a_1 x_{gi1} + a_2 x_{gi2} + \cdots + a_p x_{gip}$$

The mean of group g is $\bar{z}_g = \mathbf{a}^T \bar{\mathbf{x}}_g$, and the variance of z_g is $s_z^2 = \mathbf{a}^T \mathbf{S}_{pl} \mathbf{a}$, and the (mathematical) goal of discriminant analysis is to find the vector $\mathbf{a}$ that maximizes the squared, normalized distance between the transformed means:

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

Discriminat analysis continued again

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 the following equation:

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

So, the discriminant function maps our two groups' multidimensional data onto a single normalized distance dimension. The coefficients of $\mathbf{a}$ give us information about the relative contribution of each of our original variables to this normalized distance.

We can also use this normalized distance dimension to classify observations with respect to our two groups.

Classification

Suppose we have estimated $\mathbf{a}$, and we want to classify a new observation $\mathbf{x}$. We can do so by transforming $\mathbf{x}$ into $z = \mathbf{a}^T\mathbf{x} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}_{pl}^{-1}\mathbf{x}$ and applying a decision rule to z.

One possible decision rule is to assign z to Group 1 (G_1) if z is closer to $\bar{z}_1$ than it is to $\bar{z}_2$, and assign z to G_2 if it is closer to $\bar{z}_2$ than it is to $\bar{z}_1$.

We can state this rule mathematically as "assign z to G_1 if $z > \frac{1}{2}(\bar{z}_1 + \bar{z}_2)$, otherwise assign z to G_2."

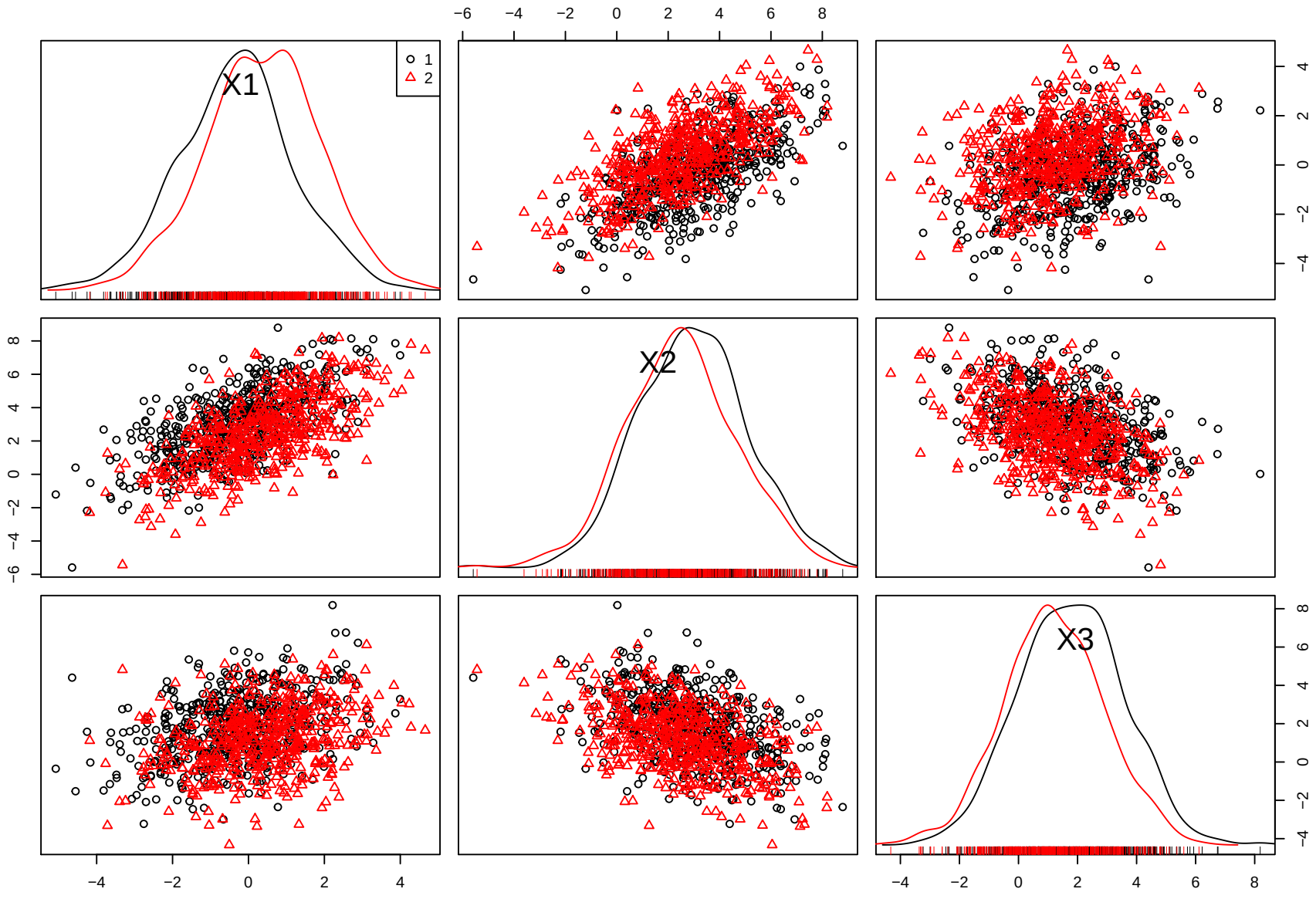
Note that this works because we defined $\mathbf{a} = \mathbf{S}_{pl}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$. If we had defined it as $\mathbf{S}_{pl}^{-1}(\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1)$, then we would assign z to G_1 if $z < \frac{1}{2}(\bar{z}_1 + \bar{z}_2)$.

An illustration with simulated data

First, we'll generate 3D multivariate normal data for two groups:

```
library(mnormt)
mu.1 = c(-0.35,2.9,1.8); mu.2 = c(0.35,2.4,1.3)
S = matrix(c(2.3,2,1,2,4.2,-1.5,1,-1.5,3),nrow=3)
n.1 = 500; n.2 = 500
X.1 = rmnorm(n=n.1,mean=mu.1,varcov=S)
X.2 = rmnorm(n=n.2,mean=mu.2,varcov=S)
X.b = rbind(X.1,X.2)
df = data.frame(X.b)
df["G"] = c(rep(1,n.1),rep(2,n.2))
```

Illustration continued

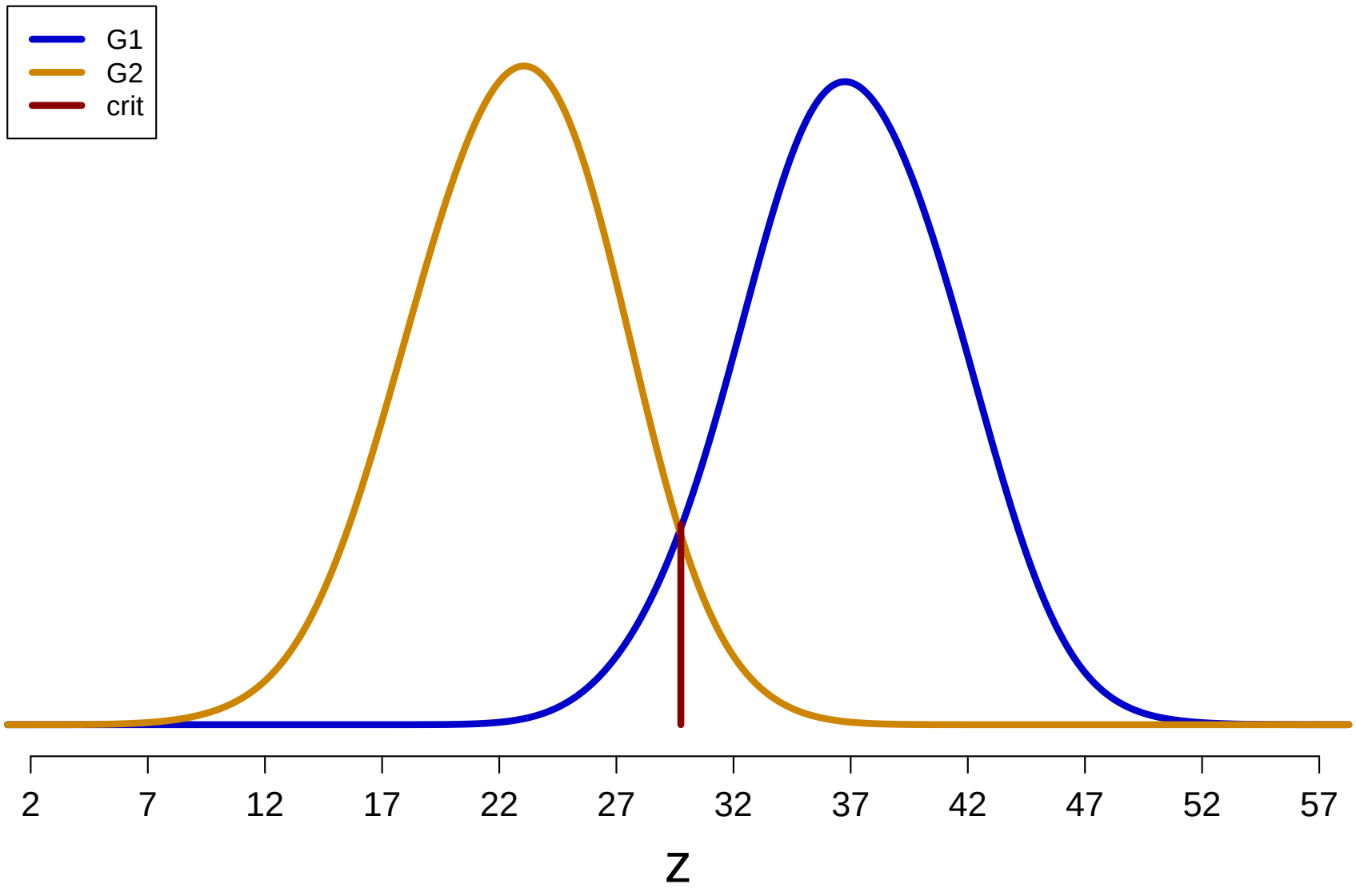


LDA

We can do LDA on this data, transform the data, then get density estimates (smoothed histograms) of the transformed data in order to visualize the transformed data/discriminant function:

```
m.1 = colMeans(X.1); m.2 = colMeans(X.2)
S.p = ((n.1-1)*cov(X.1) + (n.2-1)*cov(X.2))/(n.1 + n.2 - 2)
a = solve(S.p) %*% (m.1 - m.2)
z = X.b %*% a

# get the range and limits of the transformed data
z.rng = max(z) - min(z)
z.lims = c(min(z)-.25*z.rng,max(z)+.25*z.rng)

# density estimates of transformed data
z.1 = z[df$G == 1]; z.2 = z[df$G == 2]
f.z1 = density(z.1,from=z.lims[1],to=z.lims[2], bw=2.5)
f.z2 = density(z.2,from=z.lims[1],to=z.lims[2], bw=2.5)
```

The transformed data



Correct and incorrect classification rates

We can evaluate how well a classifier does by tallying correct and incorrect classifications:

```
# decision criterion = mean of means of z scores
crit = .5*(mean(z.1)+mean(z.2))

# check if z is less than criterion, add 1
df["G.pred"] = (z < crit) + 1

# cross-tabulate known (G) and predicted (G.pred) group assignments
xtabs(~ G + G.pred, data=df)
```

```
##    G.pred
## G     1   2
##   1 483  17
##   2  10 490
```

Unequal priors and misclassification costs

We can (and often should) also take prior probabilities and mis-classification costs into account.

Let the prior probability of G_1 be p_1 and the prior probability of G_2 be p_2, and let the cost/reward of (mis)classifying a member of group a as a member of group b be u_{ab}. Finally, let the PDF for the two groups be denoted $f(\mathbf{x}|G_1)$ and $f(\mathbf{x}|G_2)$.

Then, we can state a new classification rule that minimizes misclassification penalties and takes prior probabilities into account:

Assign $\mathbf{x}$ to G_1 if $p_1(u_{11} - u_{12})f(\mathbf{x}|G_1) > p_2(u_{22} - u_{21})f(\mathbf{x}|G_2)$, otherwise assign $\mathbf{x}$ to G_2.

We can rearrange terms to express this as a relationship between the likelihood ratio (i.e., the relative heights of the PDFs for the two groups), on the one hand, and the prior probabilities and costs/rewards, on the other:

Assign $\mathbf{x}$ to G_1 if $\dfrac{f(\mathbf{x}|G_1)}{f(\mathbf{x}|G_2)} > \dfrac{p_2(u_{22} - u_{21})}{p_1(u_{11} - u_{12})}$, otherwise assign $\mathbf{x}$ to G_2.

Continued

If our data are multivariate normal (with different mean vectors and a shared covariance matrix), we can apply a log transformation[1], and this rule can be expressed in terms of the mean vectors, pooled covariance matrix, priors, and costs/rewards as follows:

Assign $\mathbf{x}$ to G_1 if

$$(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T \mathbf{S}_{pl}^{-1} \mathbf{x} > \frac{1}{2}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T \mathbf{S}_{pl}^{-1}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) + \ln\left(\frac{p_2(u_{22} - u_{21})}{p_1(u_{11} - u_{12})}\right)$$

. Otherwise, assign $\mathbf{x}$ to G_2.

The crucial point of this equation is that differences in prior probabilities and/or costs and rewards for (mis)classification just shift the decision criterion up or down. If $p_2 > p_1$, then z has to be larger in order to be assigned to G_1 than it would need to be otherwise. Similarly if $u_{22} - u_{21} > u_{11} - u_{12}$, and vice versa.

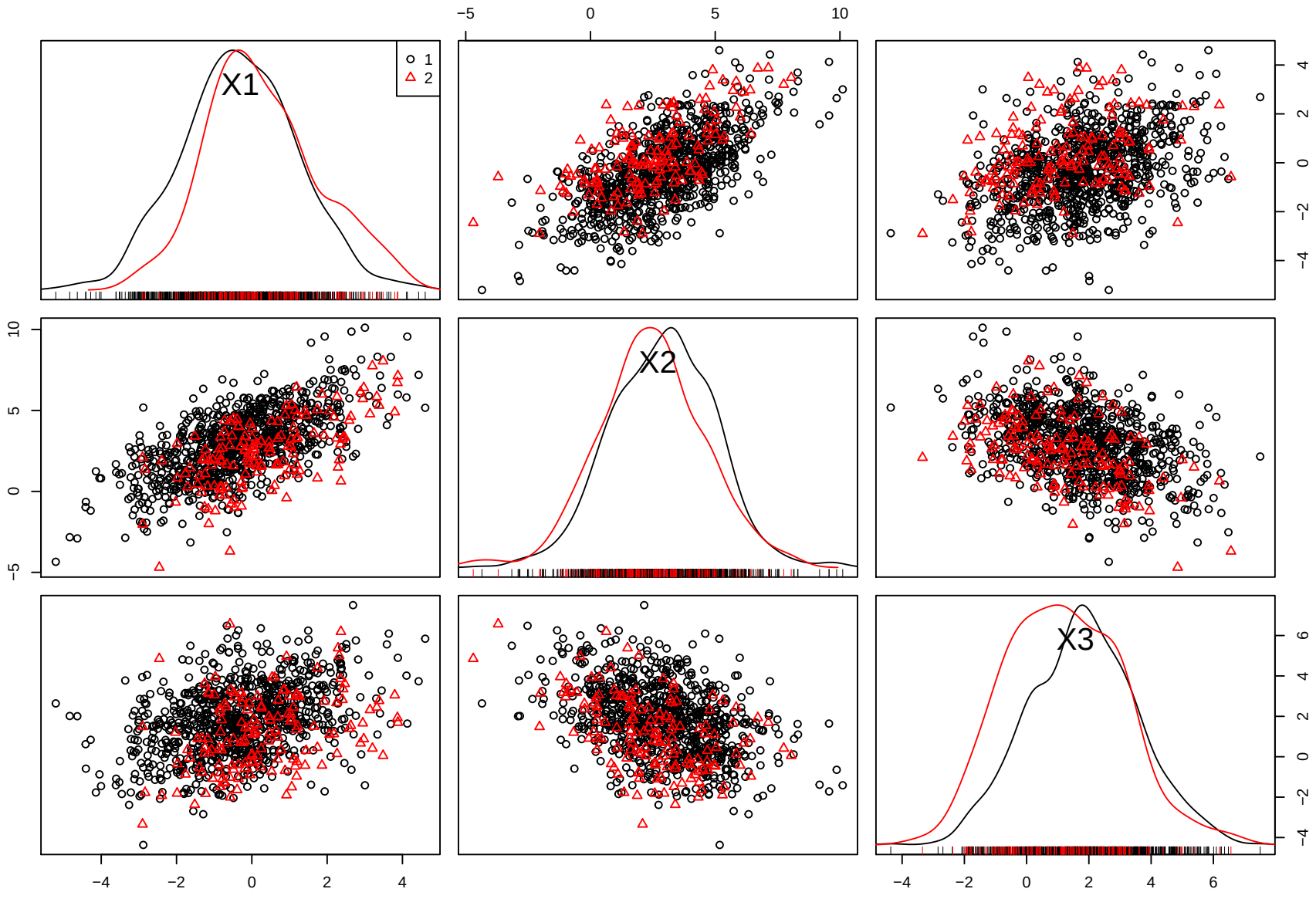
[1] This is a monotonic transformation, so the rule produces the same results.

Illustration with unequal prior probabilities

Suppose $p_1 = 0.85$ and $p_2 = 0.15$. We can simulate a new data set:

```
n.1 = 850; n.2 = 150
p.1 = n.1/(n.1 + n.2); p.2 = 1-p.1
X.1 = rmnorm(n=n.1,mean=mu.1,varcov=S)
X.2 = rmnorm(n=n.2,mean=mu.2,varcov=S)
X.b = rbind(X.1,X.2)
df = data.frame(X.b)
df["G"] = c(rep(1,n.1),rep(2,n.2))
```

Illustration continued

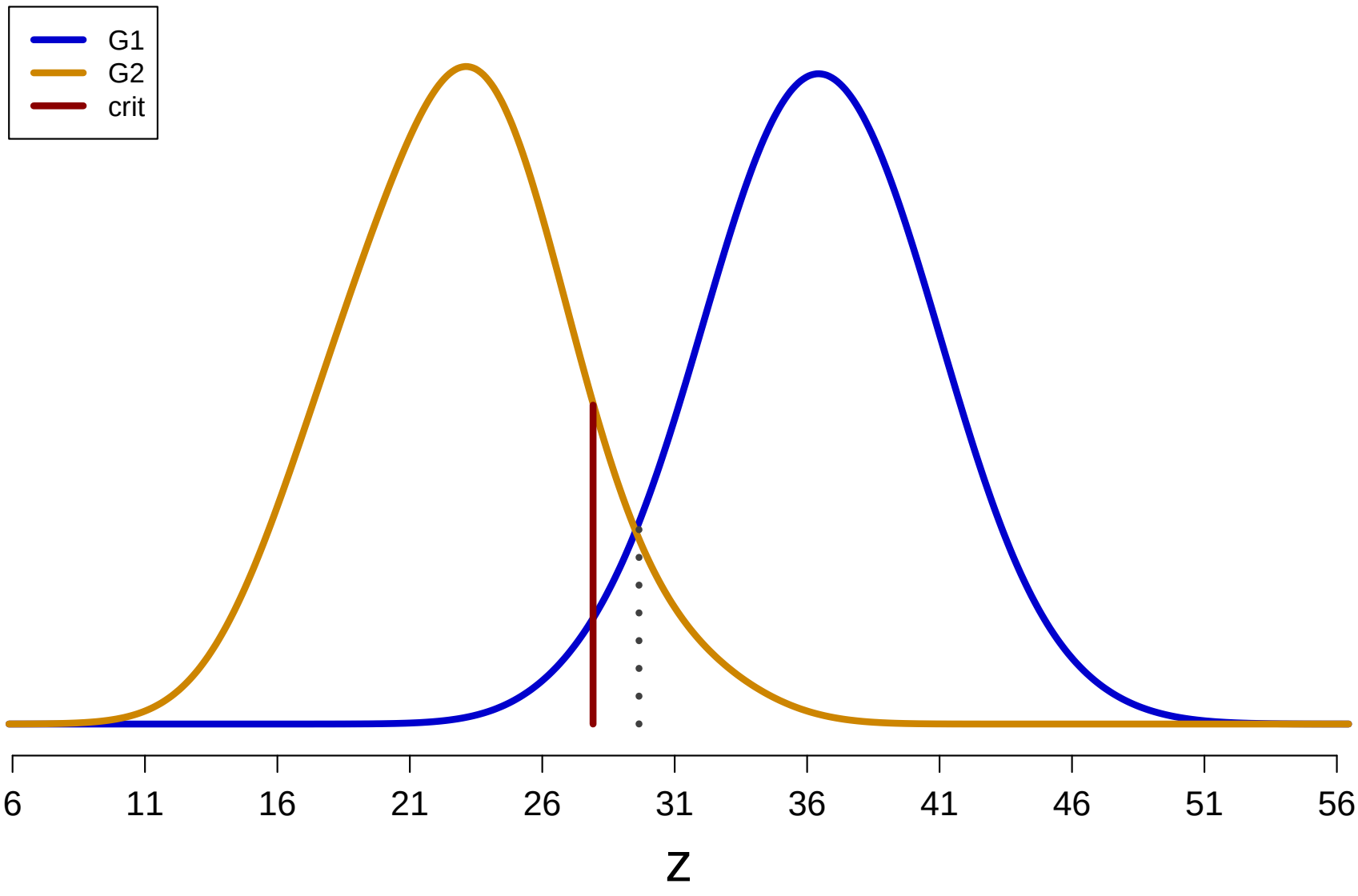


LDA

We can do LDA on the new data, transform it, and get density estimates (smoothed histograms) of the transformed data in order to visualize the transformed data/discriminant function:

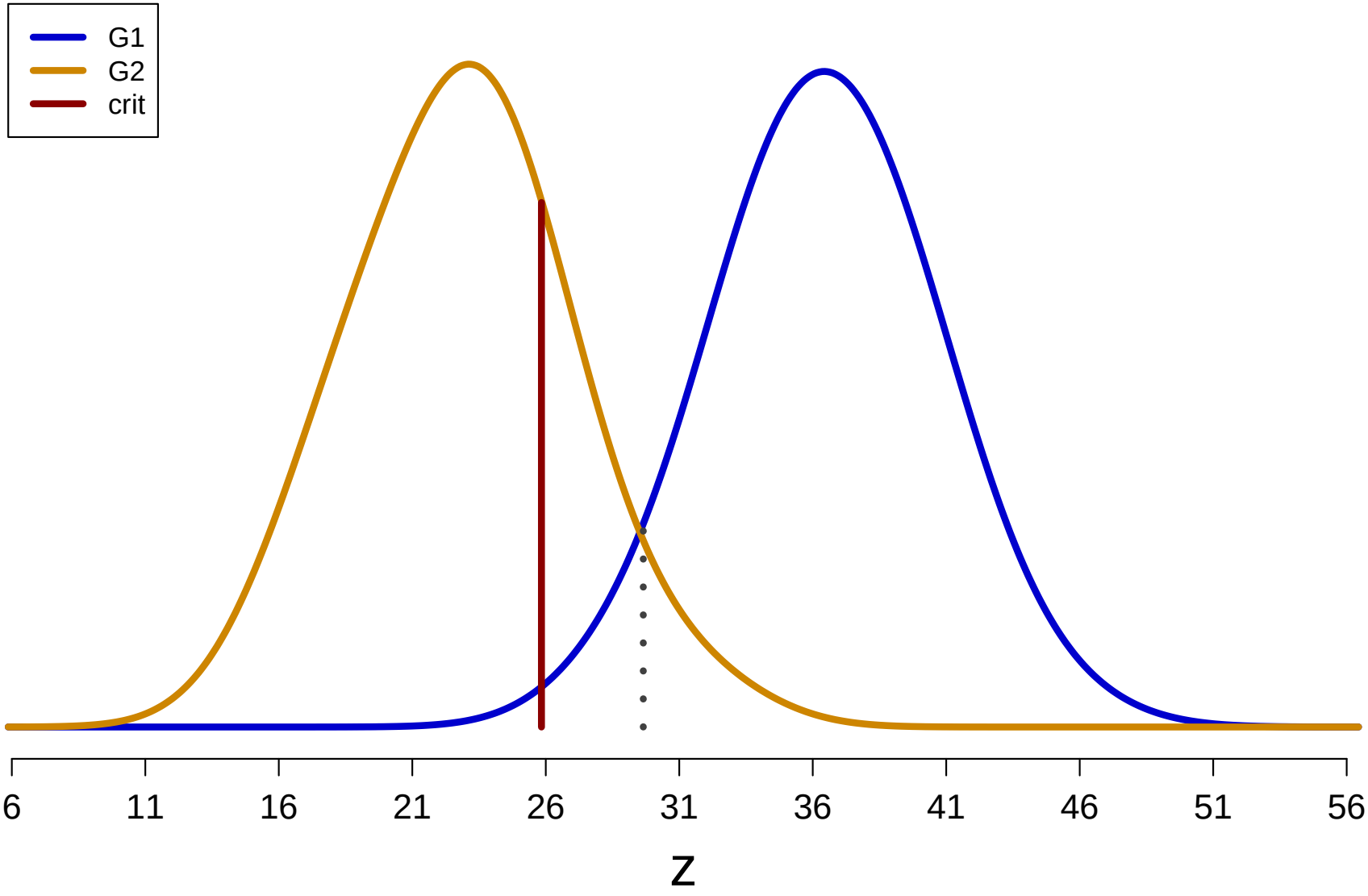
```
m.1 = colMeans(X.1); m.2 = colMeans(X.2)
S.p = ((n.1-1)*cov(X.1) + (n.2-1)*cov(X.2))/(n.1 + n.2 - 2)
a = solve(S.p) %*% (m.1 - m.2)
z = X.b %*% a

# get the range and limits of the transformed data
z.rng = max(z) - min(z)
z.lims = c(min(z)-.25*z.rng,max(z)+.25*z.rng)

# density estimates of transformed data
z.1 = z[df$G == 1]; z.2 = z[df$G == 2]
f.z1 = density(z.1,from=z.lims[1],to=z.lims[2], bw=2.5)
f.z2 = density(z.2,from=z.lims[1],to=z.lims[2], bw=2.5)
```

The transformed data



With costs and rewards

Suppose $u_{22} - u_{21} = 2$ and $u_{11} - u_{12} = 16$ (i.e., it's 8 times better to correctly classify members of G_1 than members of G_2).



Correct and incorrect classification rates

```
# adjusted decision criteria, classification
crit.1 = .5*(mean(z.1)+mean(z.2)) + log(p.2/p.1)
crit.2 = .5*(mean(z.1)+mean(z.2)) + log((p.2*2)/(p.1*16))

df["G.pr.1"] = (z < crit.1) + 1; df["G.pr.2"] = (z < crit.2) + 1

# cross-tabulate known (G) and predicted (G.pr.X) group assignments
xtabs(~ G + G.pr.1, data=df)
```

```
##    G.pr.1
## G     1   2
##   1 842   8
##   2  10 140
```

```
xtabs(~ G + G.pr.2, data=df)
```

```
##    G.pr.2
## G     1   2
##   1 849   1
##   2  26 124
```

Distributional assumptions and optimality

If the data are multivarite normal with a shared covariance matrix, then the decision rules that we've discussed are (theoretically) optimal - costs will be minimized, rewards maximized, and prior probabilities taken into account - which is to say that no other rule can do better. Also, the larger the sample, the more closely these rules will approach optimality.

If the data are not multivariate normal, this general approach can still do well, and it is still useful, but there aren't any guarantees about optimality.

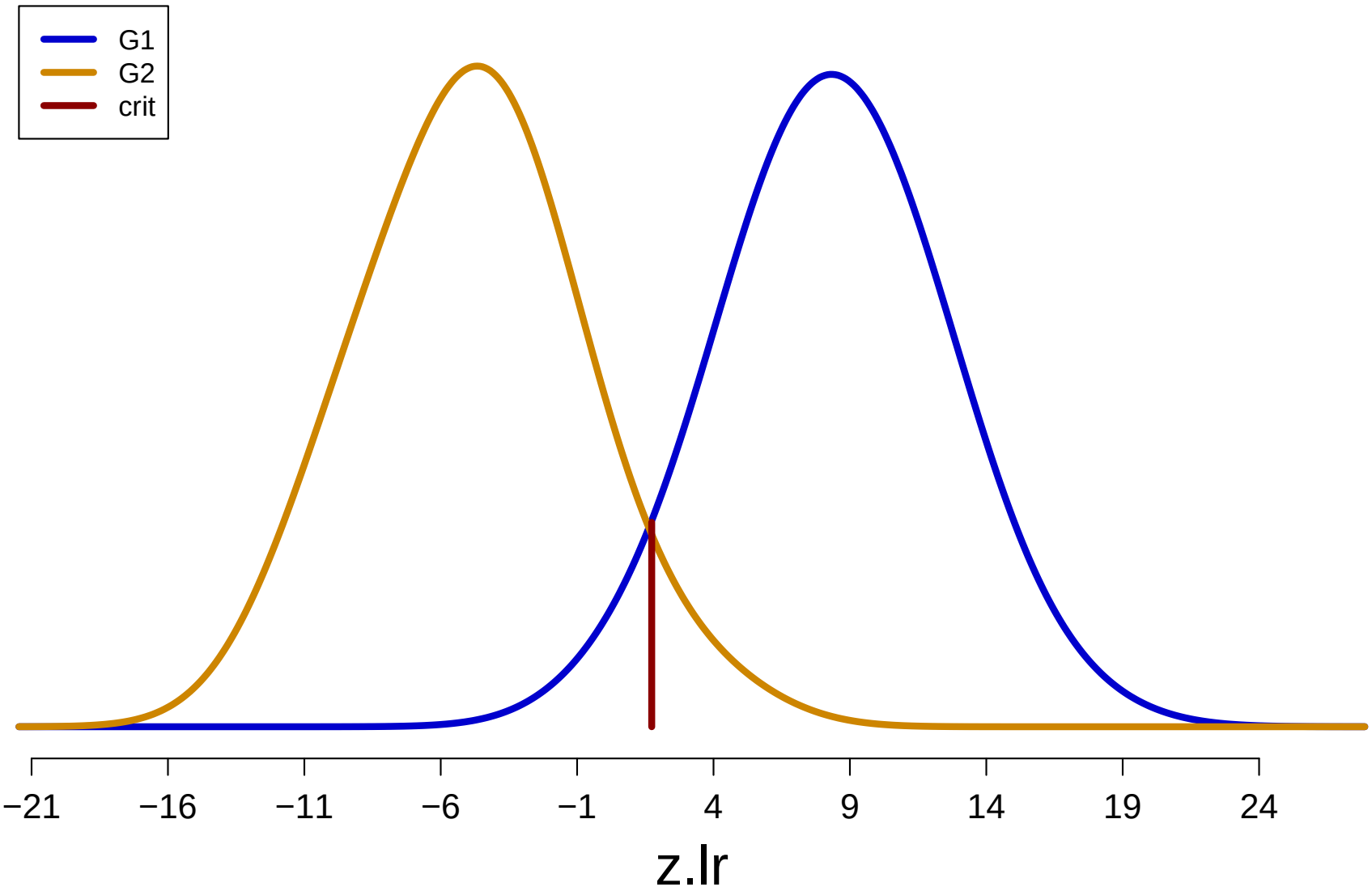
Logistic regression often produces very similar results without relying on assumptions of normality. A logistic regression model can be expressed in terms of the log odds of belonging to G_1 vs G_2:

$$\log\left(\frac{p_1}{p_2}\right) = \log\left(\frac{p_1}{1-p_1}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_p x_p$$

Note the similarity between this equation and the decision rules we discussed above.

An example

```
df["G.lr"] = -df["G"] + 2
lr.fit = glm(G.lr ~ X1 + X2 + X3, data=df, family=binomial)
z.lr = log(lr.fit$fitted.values/(1-lr.fit$fitted.values))
```



Correct and incorrect classification rates

Note that the elements of the `fitted.values` component of the fitted logistic regression model are probabilities. The model was expressed in terms of log odds above, but if we solve for p_1, we get:

$$p_1 = \frac{1}{1 + \exp\left(-\left(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p\right)\right)}$$

We can use either the log odds or the probability estimates to see how well the model classifies:

```
# compare fitted values to criterion
#df["G.pr.lr"] = -(lr.fit$fitted.values > .85) + 2
df["G.pr.lr"] = -(z.lr > log(p.1/p.2)) + 2

# cross-tabulate known (G) and predicted (G.pr.X) group assignments
xtabs(~ G + G.pr.lr, data=df)
```

```
##    G.pr.lr
## G      1   2
##   1 818  32
##   2   7 143
```

More than two groups

Discriminant analysis can also be used to classify observations (data vectors) with more than two groups. In this case, the decision rule is to assign $\mathbf{x}$ to the group i for which the squared statistical distance between $\mathbf{x}$ and $\bar{\mathbf{x}}_i$ is smallest:

$$D_i^2\mathbf{x} = (\mathbf{x} - \bar{\mathbf{x}}_i)^T \mathbf{S}_{pl}^{-1} (\mathbf{x} - \bar{\mathbf{x}}_i)$$

If we expand the right side of this equation, drop constant terms, and multiply by $-\frac{1}{2}$ (to express the rule in a way that makes it easy to incorporate multivariate normality and prior probabilities), we get the equivalent rule to assign $\mathbf{x}$ to group i for which $L_i(\mathbf{x})$ is maximized:

$$L_i(\mathbf{x}) = \bar{\mathbf{x}}_i^T \mathbf{S}_{pl}^{-1}\mathbf{x} - \frac{1}{2}\bar{\mathbf{x}}_i^T \mathbf{S}_{pl}^{-1}\bar{\mathbf{x}}_i$$

With prior probabilities, multinomial logistic regression

If the prior probabilities of the groups are not equal, a good rule is to assign $\mathbf{x}$ to group i for which $p_i f(\mathbf{x}|G_i)$ is maximized. If the data are multivariate normal, this can be expressed as:

$$L_i'(\mathbf{x}) = \ln(p_i) + \bar{\mathbf{x}}_i^T \mathbf{S}_{pl}^{-1} \mathbf{x} - \frac{1}{2} \bar{\mathbf{x}}_i^T \mathbf{S}_{pl}^{-1} \bar{\mathbf{x}}_i$$

There is an analogous multi-group version of logistic regression called multinomial logistic regression. In a multinomial logistic regression model, the probability of $\mathbf{x}$ belonging to group i is:

$$p_i = \frac{\exp\left(-\left(\beta_{0i} + \beta_{1i} x_1 + \cdots + \beta_{pi} x_p\right)\right)}{\sum_{k=1}^{G} \exp\left(-\left(\beta_{0k} + \beta_{1k} x_1 + \cdots + \beta_{pk} x_p\right)\right)}$$

Quadratic discriminant analysis

Finally, if the covariance matrices for the groups are not equal, the discriminant functions are not linear. Rather, they are quadratic.

In this case, we can use a rule similar to the multi-group (linear) discriminant analysis discussed above, assigning to group i such that the following is minimized (note the group-specific covariance matrix, rather than the pooled covariance matrix):

$$D_i^2\mathbf{x} = (\mathbf{x} - \bar{\mathbf{x}}_i)^T \mathbf{S}_i^{-1} (\mathbf{x} - \bar{\mathbf{x}}_i)$$

If we assume multivariate normality and incorporate prior probabilities, we assign to group i such that $p_i f(\mathbf{x}|G_i)$ is maximized, which is equivalent to maximizing $Q_i\mathbf{x}$:

$$Q_i\mathbf{x} = \ln(p_i) - \frac{1}{2}|\mathbf{S}_i| - \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_i)^T \mathbf{S}_i^{-1} (\mathbf{x} - \bar{\mathbf{x}}_i)$$