Multivariate Stats, Week 6

Noah Silbert

September 29, 2017

To do

0. Singular Value Decomposition, PCA, & R
1. Review of homework
2. Intro to discriminant analysis and classification

Singular Value Decomposition

Singular Value Decomposition (SVD) is a generalization of eigendecomposition. For certain square matrices, eigenvalues and eigenvectors are readily obtained, but when the matrix is large, the computations are expensive.

SVD is less computationally expensive, and it applies to a wider class of matrices. An $m \times n$ matrix, with $m \geq n$, can be written $\mathbf{A} = \mathbf{UDV}^T$, where $\mathbf{U}$ is an orthonormal $m \times n$ matrix, $\mathbf{D}$ is an $n \times n$ diagonal matrix, and $\mathbf{V}$ is a $n \times n$ orthonormal matrix.

A matrix $\mathbf{M}$ is orthonormal if $\mathbf{M}^T\mathbf{M} = \mathbf{I}$.

The elements of $\mathbf{D}$ are the singular values (analogous to eigenvalues), the columns of $\mathbf{U}$ are the left singular vectors, and the columns of $\mathbf{V}$ are the right singular vectors of $\mathbf{A}$.

In R, you can do singular value decomposition with the function `svd()`.

SVD, PCA, and R

There are (at least) two ways to do PCA without very simply in R: `princomp()` and `prcomp()`. These use eigendecomposition and SVD, respectively:

```
df = read.table("T3_8_SONS.DAT")
X = as.matrix(df[,1:2])
colnames(X) = c("L","W")
pca.eig = princomp(X)
pca.svd = prcomp(X)
print(rbind(pca.eig$sdev,pca.svd$sdev))
```

```
##         Comp.1   Comp.2
## [1,] 11.23644 4.172481
## [2,] 11.46814 4.258521
```

```
print(cbind(pca.eig$loadings,pca.svd$rotation))
```

```
##       Comp.1     Comp.2        PC1        PC2
## L -0.8249295  0.5652357 -0.8249295 -0.5652357
## W -0.5652357 -0.8249295 -0.5652357  0.8249295
```

Discriminant analysis and classification

A quick note: the book uses "discriminant analysis" to refer to analyses in which the focus is on the relative importance of different variables in separating groups, and it uses "classification analysis" to refer to analyses in which we are modeling and/or predicting group membership. There's a lot of overlap between these two goals (and types of analyses), so I won't be a stickler about this terminology.

Discriminant analysis

Suppose we have p variables measured for each member of each of two groups $g = 1, 2$, and suppose that the data for the two groups have the same covariance matrix $\mathbf{\Sigma} = \mathbf{\Sigma}_1 = \mathbf{\Sigma}_2$ 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):

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

Note that while the i^{th} observation $\mathbf{x}_{gi}$ is a vector, the linear transformation z_{gi} is a scalar (i.e., a single number).

Discriminant analysis continued

The mean of group g is $\bar{z}_g = \mathbf{a}^T \bar{\mathbf{x}}_g$, and the variance of z_g is $s_z^2 = \mathbf{a}^T \mathbf{S}_{pl} \mathbf{a}$, where $\mathbf{S}_{pl} = \dfrac{\sum_{i=1}^{k} \nu_i \mathbf{S}_i}{\sum_{i=1}^{k} \nu_i}$, $\nu_i = n_i - 1$, is the pooled covariance matrix.

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

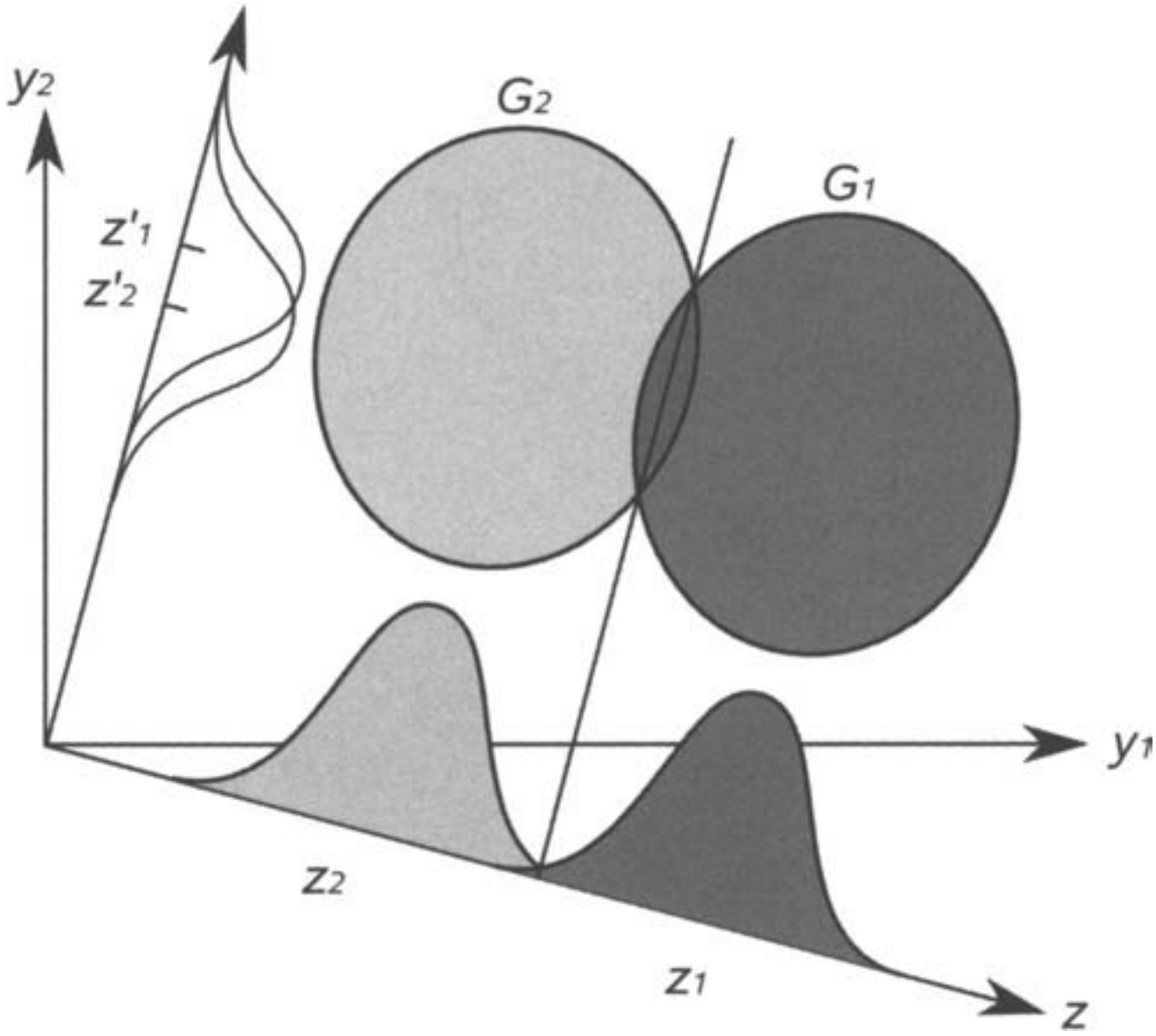
The maximum of this distance occurs when $\mathbf{a} = \mathbf{S}_{pl}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$ (or any scalar multiple of this). If we substitute this expression for $\mathbf{a}$ into the equation above, we see that

$$\begin{aligned}
\frac{(\bar{z}_1 - \bar{z}_2)^2}{s_z^2} &= \frac{\left[\mathbf{a}^T(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)\right]^2}{\mathbf{a}^T\mathbf{S}_{pl}\mathbf{a}} \\
&= \frac{\left[(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}_{pl}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)\right]^2}{(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}_{pl}^{-1}\mathbf{S}_{pl}\mathbf{S}_{pl}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)} \\
&= \frac{\left[(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}_{pl}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)\right]^2}{(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}_{pl}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)} \\
&= (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{S}_{pl}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)
\end{aligned}$$

The point here is that $(\bar{z}_1 - \bar{z}_2)^2/s_z^2$ is just the standardized distance between the mean vectors $\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$.

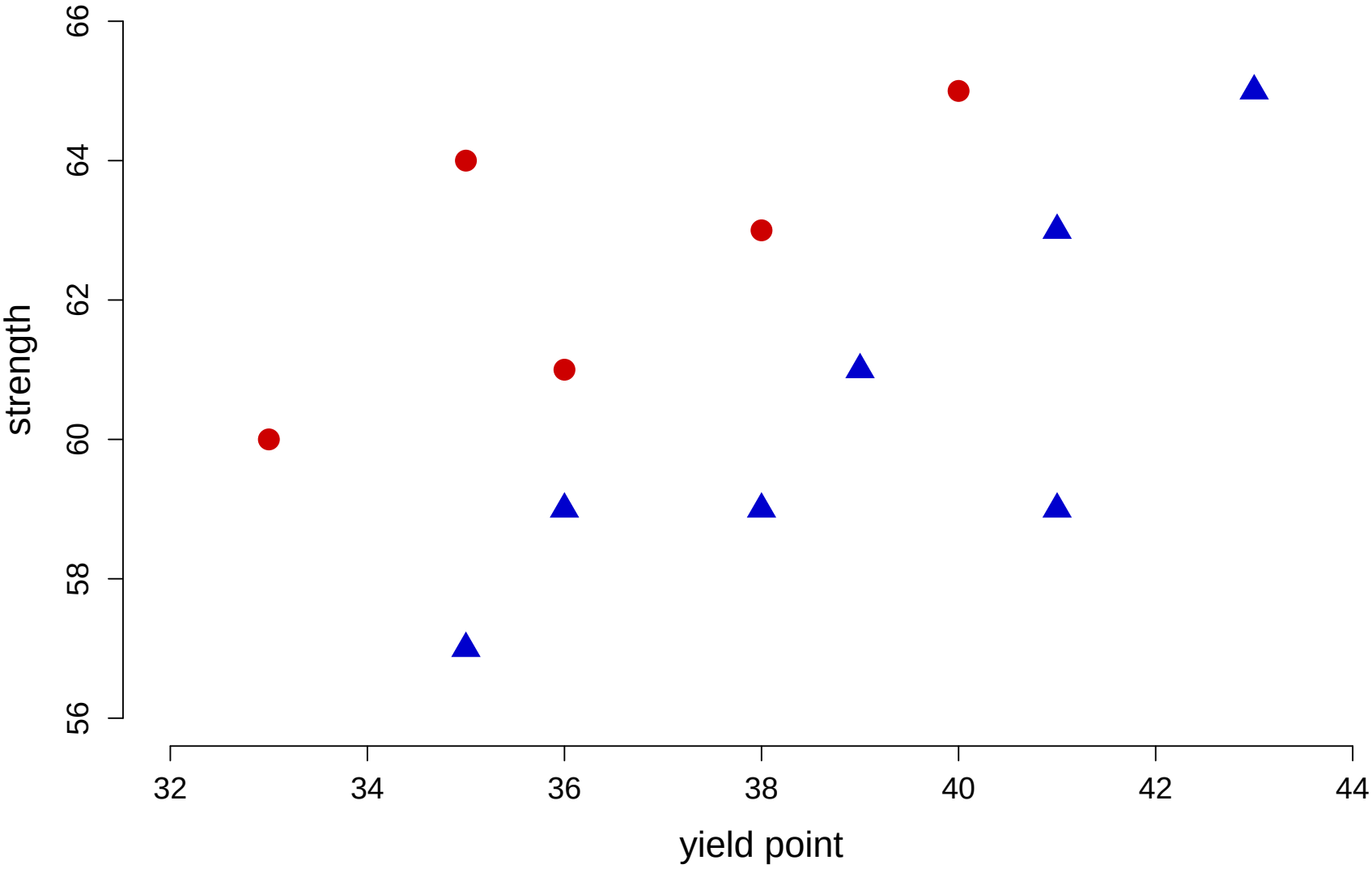
Discriminant analysis illustrated

Here's an illustration of discriminant analysis with two groups in two dimensions:



Discriminant analysis, continued again with math!

Here's the data from Example 8.2 from the book. The data are 2D (yield point and ultimate strength of steel produced at two rolling temperatures):



R code for discriminant analysis

We need the mean vectors and the pooled covariance matrix to estimate **a**:

```
# read in data
dd = read.table("T8_1_STEEL.DAT")

# name the columns
colnames(dd) = c("group","yield point","strength")

# extract separate data sets for each group
g1 = dd[dd$group %in% 1,2:3]
g2 = dd[dd$group %in% 2,2:3]

# sample size for each group
n1 = nrow(g1); n2 = nrow(g2)
```

More R code for discriminant analysis

```r
# mean vectors for each group
mu1 = colMeans(g1); mu2 = colMeans(g2)

# pooled covariance matrix
W1 = (n1-1)*cov(g1)
W2 = (n2-1)*cov(g2)
Spl = (1/(n1+n2-2))*(W1+W2)

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

```
##                    [,1]
## yield point -1.633377
## strength     1.819779
```

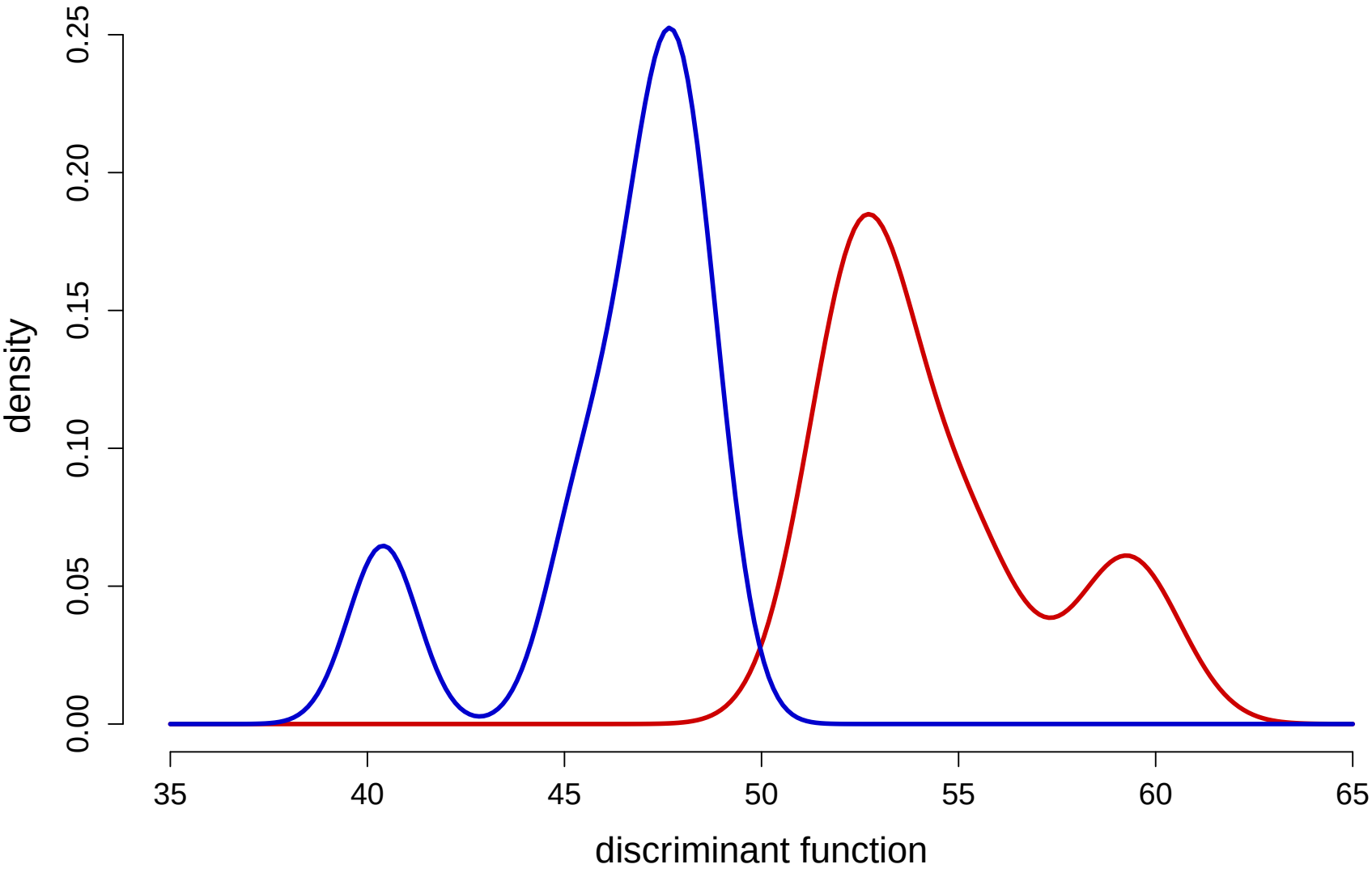
Interpreting and using the transformation vector

The values of the elements of **a** provide information about the relative importance of each variable in separating the groups. In this case, the values are similar for both variables. In many cases, the elements of **a** will vary substantially.

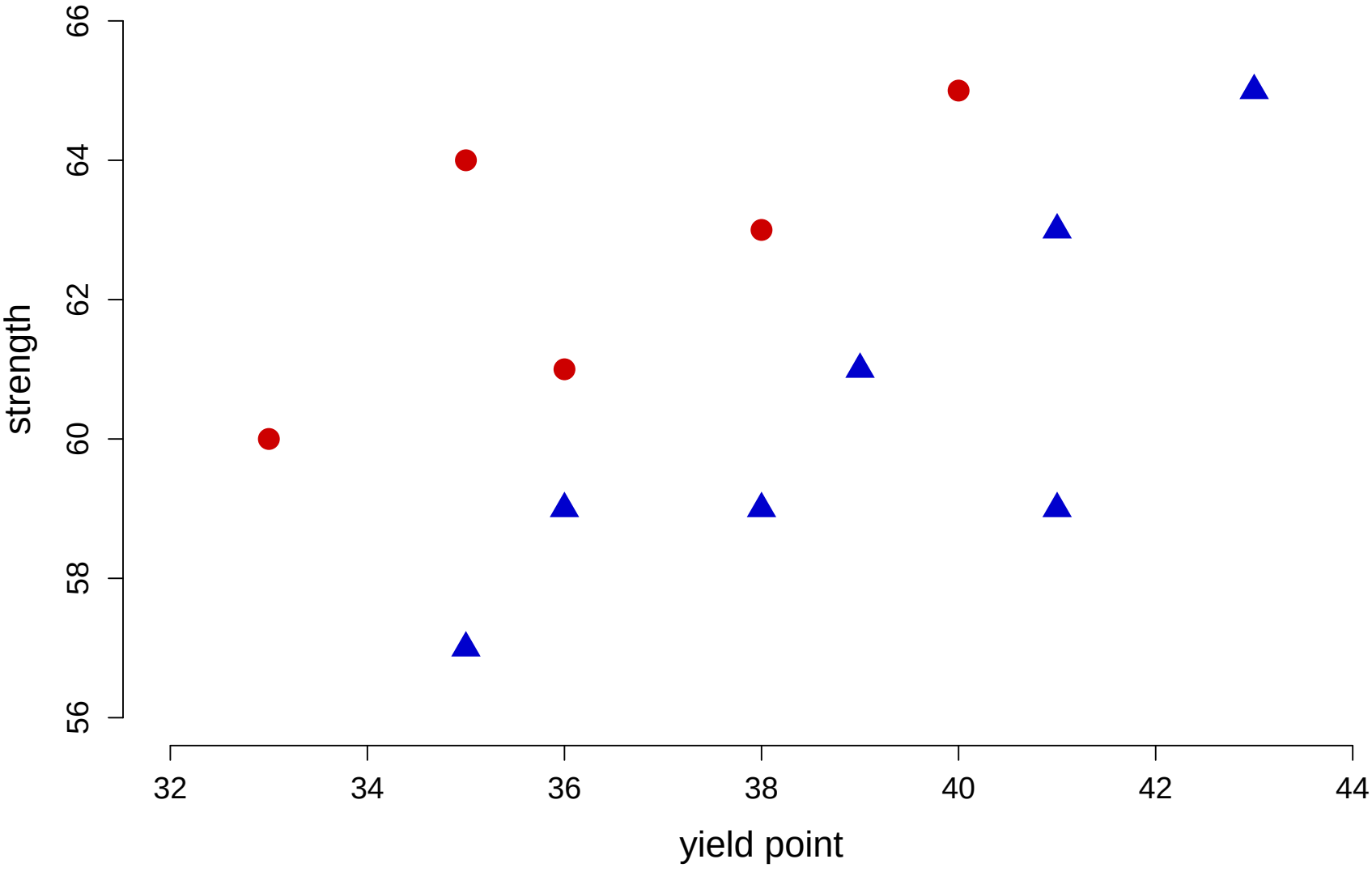
In addition, we can use **a** to transform our original data, projecting it onto the dimension along which the groups are maximally separated.

```
X1 = as.matrix(g1); X2 = as.matrix(g2)
z1 = X1 %*% a
z2 = X2 %*% a
```

Transformed example data



The original data again



Discriminant analysis for 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) **H** and **E**:

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

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

With two groups, we wanted to maximize $(\bar{z}_1 - \bar{z}_2)^2 / s_z^2$. With more than two groups, we want to maximize λ such that:

$$\lambda = \frac{\mathbf{a}^T \mathbf{H} \mathbf{a}}{\mathbf{a}^T \mathbf{E} \mathbf{a}}$$

More than two groups continued

We can rewrite this and manipulate algebraically to get:

$$\begin{aligned}
\mathbf{a}^T\mathbf{H}\mathbf{a} &= \lambda\mathbf{a}^T\mathbf{E}\mathbf{a} \\
\mathbf{a}^T\mathbf{H}\mathbf{a} - \lambda\mathbf{a}^T\mathbf{E}\mathbf{a} &= 0 \\
\mathbf{a}^T\left(\mathbf{H}\mathbf{a} - \lambda\mathbf{E}\mathbf{a}\right) &= 0 \\
\mathbf{a}^T\left(\mathbf{H} - \lambda\mathbf{E}\right)\mathbf{a} &= 0 \\
\mathbf{a}^T\left(\mathbf{E}^{-1}\mathbf{H} - \lambda\mathbf{I}\right)\mathbf{a} &= 0
\end{aligned}$$

The point of which is that λ_i (the i^{th} eigenvalue of $\mathbf{E}^{-1}\mathbf{H}$) gives us a measure of how effectively the i^{th} discriminant function separates the groups, and $\mathbf{a}_i$ (the i^{th} eigenvector of $\mathbf{E}^{-1}\mathbf{H}$) gives us our discriminant function coefficients, which we can use to transform our data and figure out which variables are doing the most work in separating the groups on the i^{th} dimension.

Simple LDA in R

There is a convenient function `lda()` in the `MASS` library that performs linear discriminant analysis without requiring you to calculate complicated matrices on your own. The first two input arguments for `lda()` are either a formula and a data frame or a matrix and a grouping factor. You can also optionally specify prior probabilities for each group, along with various other parameters.

`lda()` returns an object containing a number of things, including the mean vectors for the groups, the coefficients that transform the data to project it onto the discriminant functions, the prior probabilities used in the analysis, and the singular values (i.e., the ratio of between and within group SDs on the discriminant functions), along with a few other variables.

We'll look at an R script to see how you can use this function to carry out LDA and visualize the results.