Multivariate Analysis and Discrimination

BST 226

Statistical Methods for Bioinformatics

David M. Rocke

Cystic Fibrosis Data Set

- The 'cystfibr' data frame has 25 rows and 10 columns. It contains lung function data for cystic fibrosis patients (7-23 years old)
- We will examine the relationships among the various measures of lung function

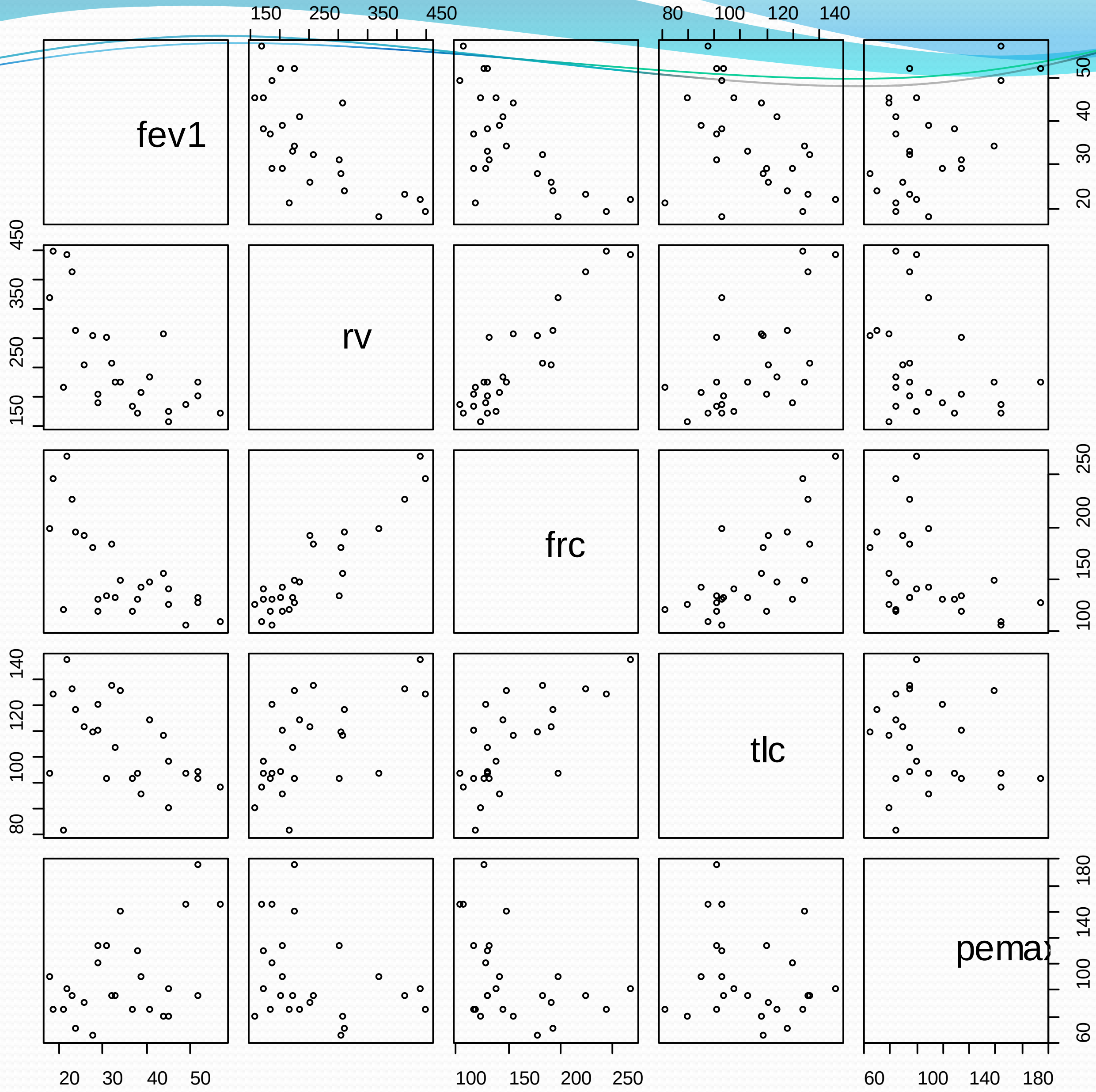
- age: a numeric vector. Age in years.
- sex: a numeric vector code. 0: male, 1:female.
- height: a numeric vector. Height (cm).
- weight: a numeric vector. Weight (kg).
- bmp: a numeric vector. Body mass (% of normal).
- **fev1: a numeric vector. Forced expiratory volume.**
- **rv: a numeric vector. Residual volume.**
- **frc: a numeric vector. Functional residual capacity.**
- **tlc: a numeric vector. Total lung capacity.**
- **pemax: a numeric vector. Maximum expiratory pressure.**

Scatterplot matrices

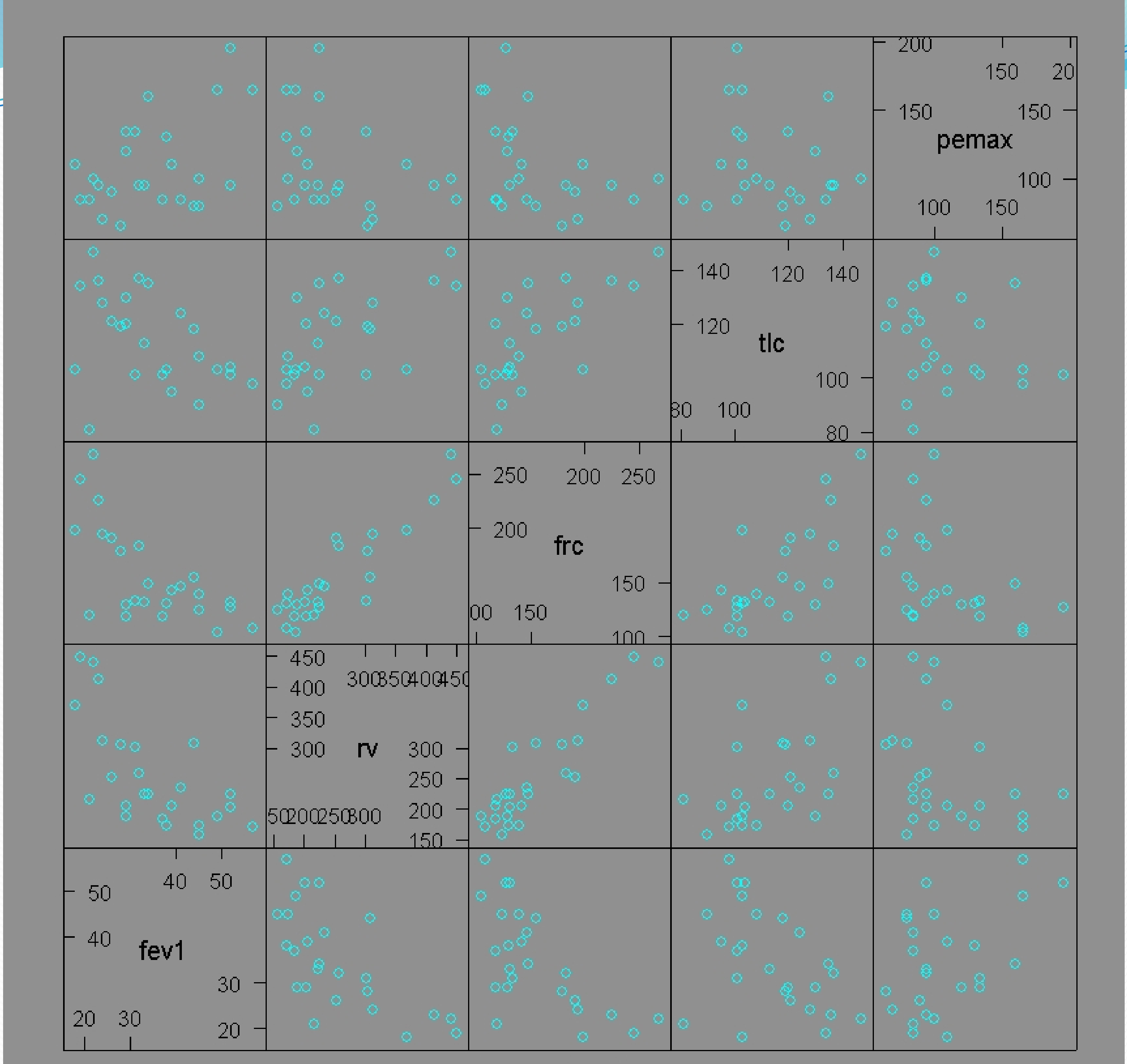
- We have five variables and may wish to study the relationships among them
- We could separately plot the (5)(4)/2 = 10 pairwise scatterplots
- In R we can use the `pairs()` function, or the `splom()` function in the `lattice` package.
- In Stata, we can use `graph matrix`
- Most other statistical packages can do the same

Scatterplot matrices

```
> pairs(lungcap)

> library(lattice)
> splom(lungcap)
```



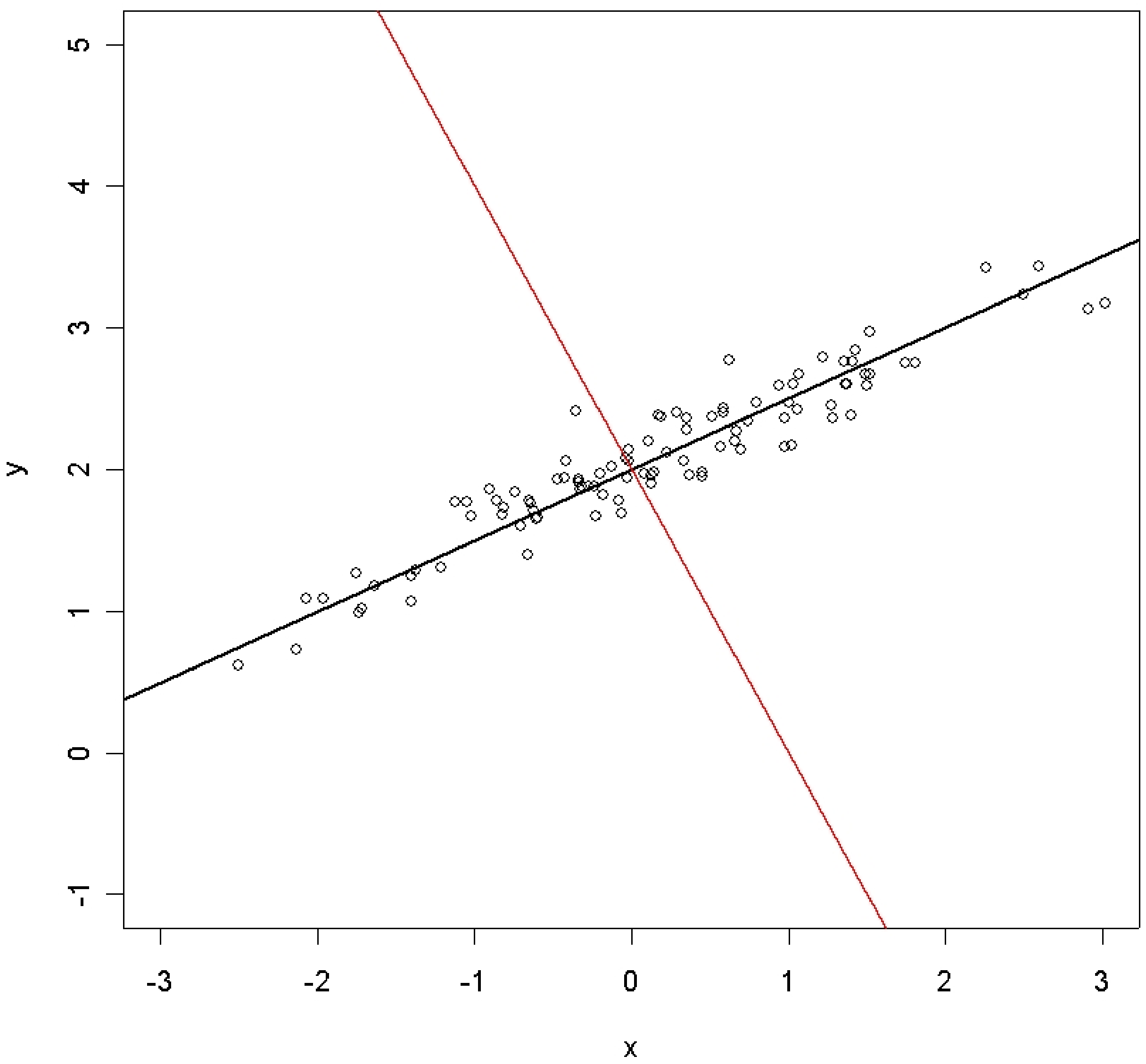
fev1
rv
frc
tlc
pemax
150
250
350
450
80
100
120
140
20
30
40
50
100
150
200
250
60
100
140
180



Scatter Plot Matrix

Principal Components Analysis

- The idea of PCA is to create new variables that are combinations of the original ones.
- If x_1, x_2, …, x_p are the original variables, then a component is $a_1x_1 + a_2x_2 +…+ a_px_p$
- We pick the first PC as the linear combination that has the largest variance
- The second PC is that linear combination orthogonal to the first one that has the largest variance, and so on
- Frequently, we scale the variables first, so that each has mean 0 and variance 1.
- Then the covariance matrix of X is also the correlation matrix.


y
-1
0
1
2
3
4
5
-3
-2
-1
0
1
2
3
x

Formally, if X is an n observations by p variables mean centered matrix and

u is a unit vector of length p $\left(|u|^2 = u^\top u = \sum_{i=1}^{p} u_i^2 = 1 \right)$ then

Xu is an $n \times 1$ vector consisting of the projections of each of the n data points on u

The first principal component of X satisfies

$$u = \underset{\|u\|=1}{\arg\max}\{|Xu|^2\} = \underset{\|u\|=1}{\arg\max}\{u^\top X^\top Xu\} \text{ or}$$

$$u = \underset{u}{\arg\max}\left[\frac{u^\top X^\top Xu}{u^\top u}\right]$$

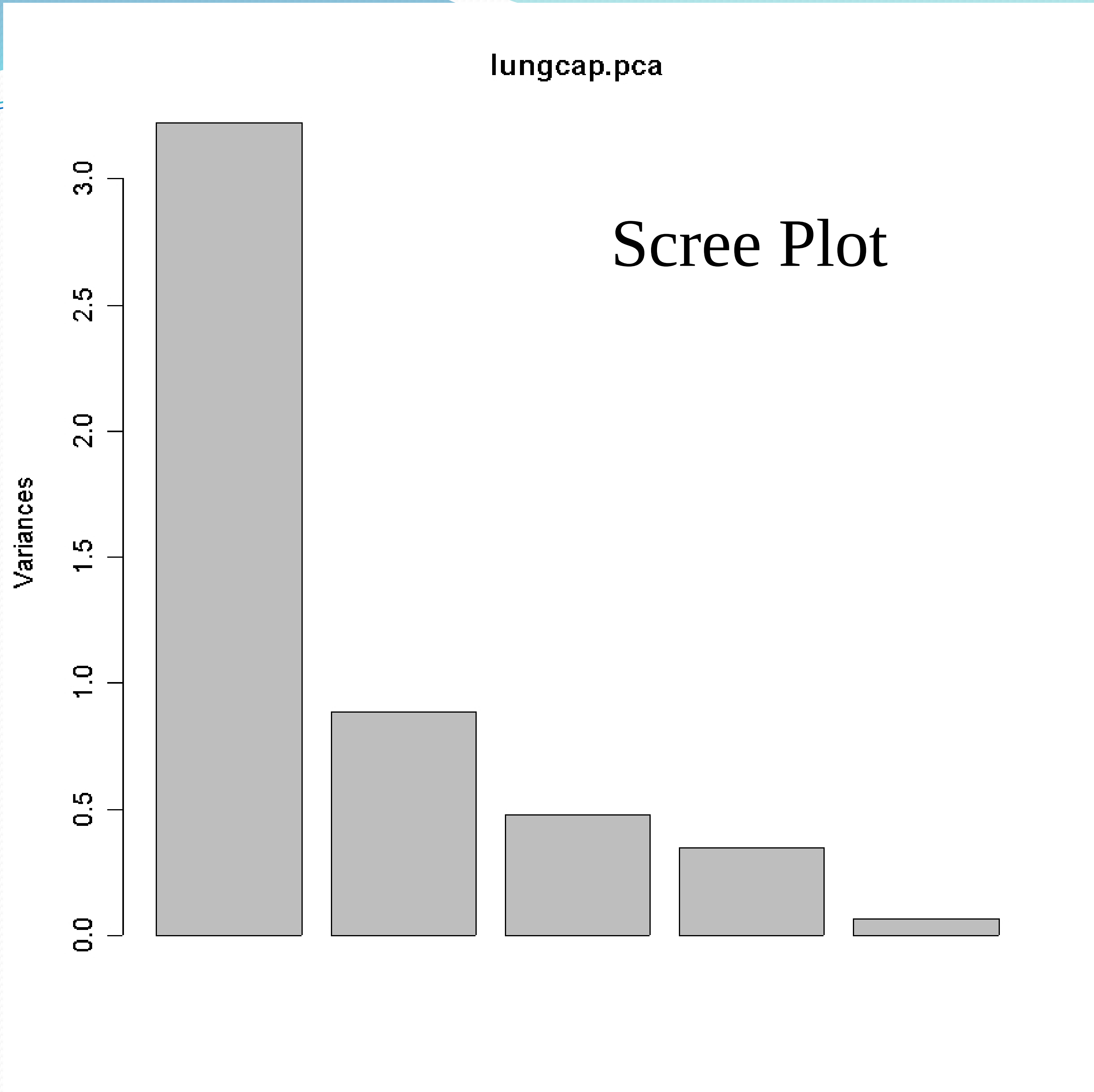
These can all be extracted from the eigenvalue/eigenvector decomposition of $X^\top X$.

Since this is a symmetric matrix, it can be written as

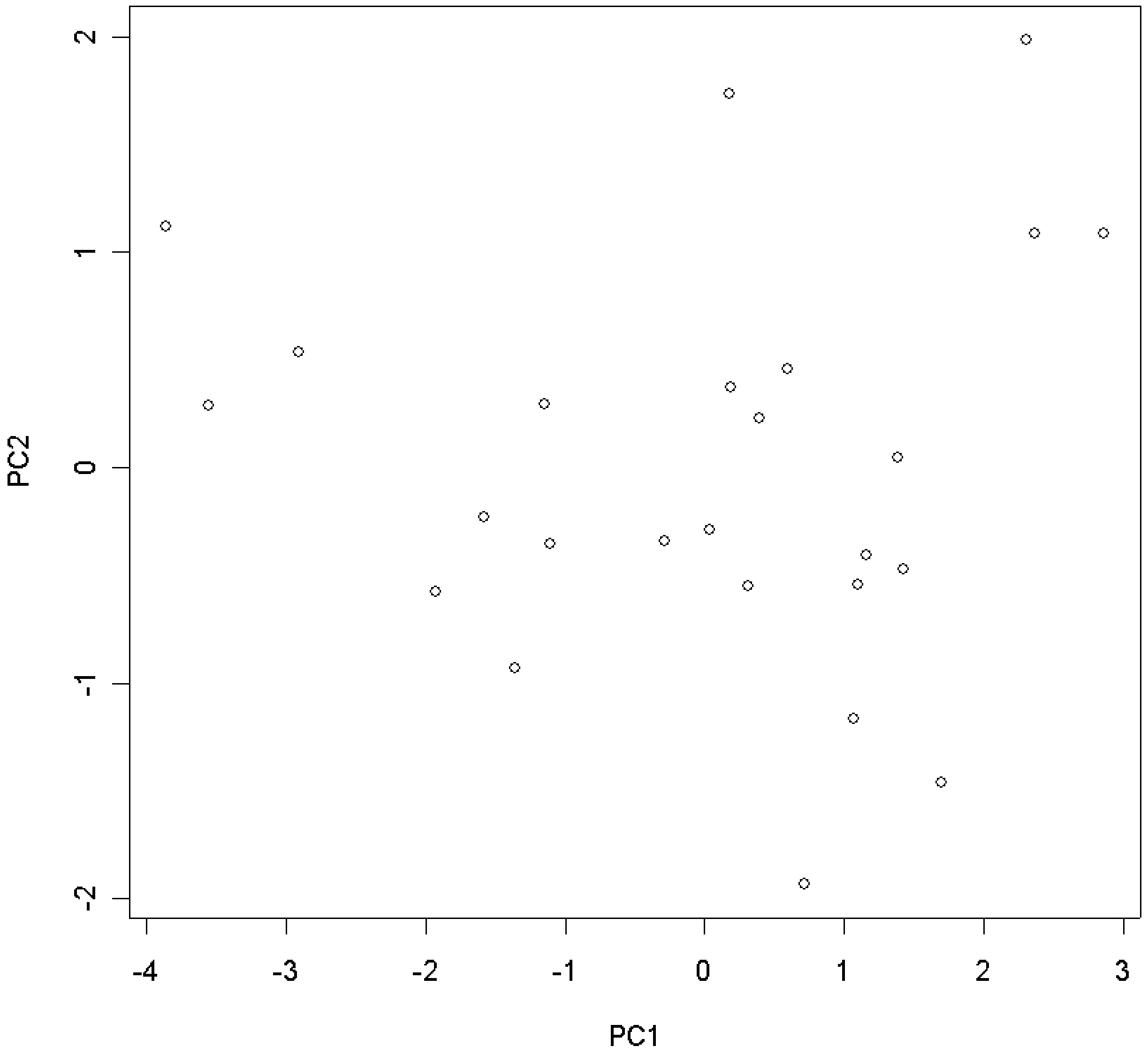
$X^\top X = U^\top \Lambda U$ where U is an orthonormal matrix of eigenvectors (each is of length 1 and they are orthogonal) and Λ is a diagonal matrix of eigenvalues

```
> lungcap.pca <- prcomp(lungcap,scale=T)
> plot(lungcap.pca)
> names(lungcap.pca)
[1] "sdev"     "rotation" "center"   "scale"    "x"
> lungcap.pca$sdev
[1] 1.7955824 0.9414877 0.6919822 0.5873377 0.2562806
> lungcap.pca$center
  fev1     rv    frc    tlc  pemax
 34.72 255.20 155.40 114.00 109.12
> lungcap.pca$scale
    fev1       rv      frc      tlc    pemax
11.19717 86.01696 43.71880 16.96811 33.43691

> plot(lungcap.pca$x[,1:2])
```

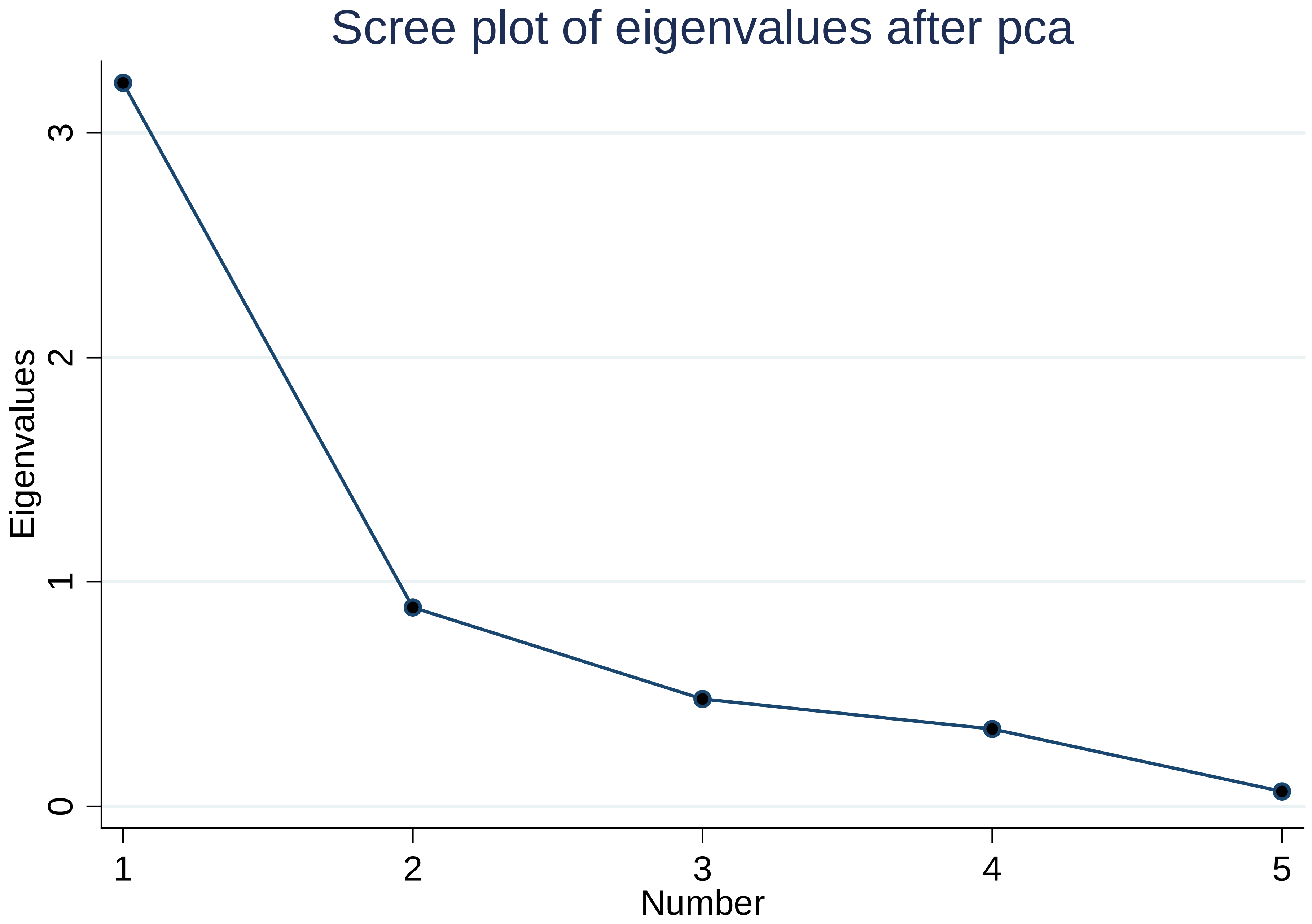
Always use scaling before PCA unless all variables are on the same scale. This is equivalent to PCA on the correlation matrix instead of the covariance matrix

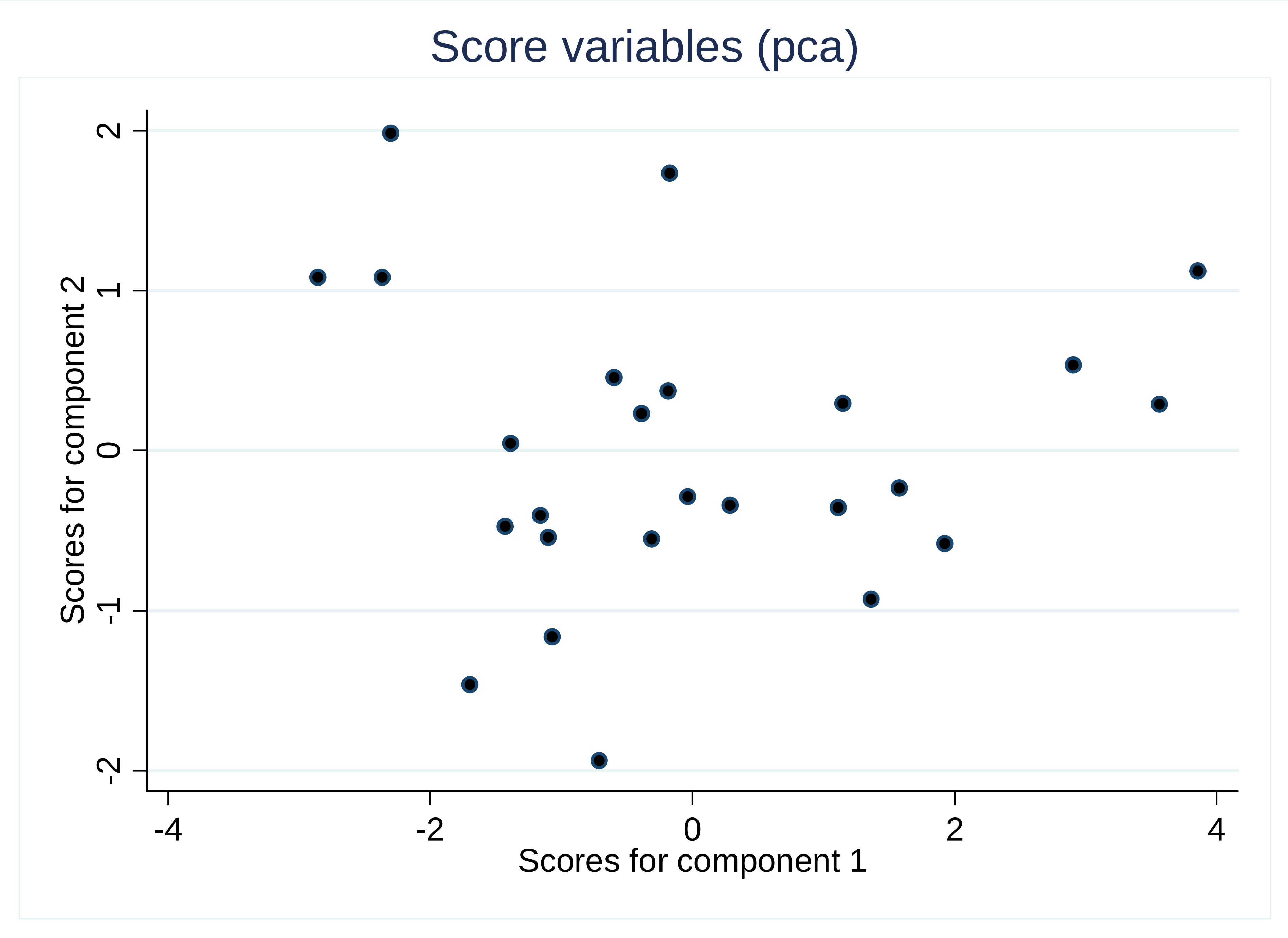


Scree Plot


PC2
2
1
0
-1
-2
-4
-3
-2
-1
0
1
2
3
PC1

```
. pca fev1 rv frc tlc pemax

Principal components/correlation                  Number of obs    =        25
                                                  Number of comp.  =         5
                                                  Trace            =         5
    Rotation: (unrotated = principal)             Rho              =    1.0000

    --------------------------------------------------------------------------
       Component |   Eigenvalue   Difference         Proportion   Cumulative
    -------------+------------------------------------------------------------
           Comp1 |      3.22412      2.33772             0.6448       0.6448
           Comp2 |      .886399       .40756             0.1773       0.8221
           Comp3 |      .478839      .133874             0.0958       0.9179
           Comp4 |      .344966      .279286             0.0690       0.9869
           Comp5 |     .0656798            .             0.0131       1.0000
    --------------------------------------------------------------------------

Principal components (eigenvectors)

    ------------------------------------------------------------------------------
        Variable |    Comp1     Comp2     Comp3     Comp4     Comp5 | Unexplained
    -------------+--------------------------------------------------+-------------
            fev1 |  -0.4525    0.2140    0.5539    0.6641   -0.0397 |           0
              rv |   0.5043    0.1736   -0.2977    0.4993   -0.6145 |           0
             frc |   0.5291    0.1324    0.0073    0.3571    0.7582 |           0
             tlc |   0.4156    0.4525    0.6474   -0.4134   -0.1806 |           0
           pemax |  -0.2970    0.8377   -0.4306   -0.1063    0.1152 |           0
    ------------------------------------------------------------------------------
```


Scree plot of eigenvalues after pca
Eigenvalues
0
1
2
3
1
2
3
4
5
Number


Score variables (pca)
Scores for component 2
2
1
0
-1
-2
-4
-2
0
2
4
Scores for component 1

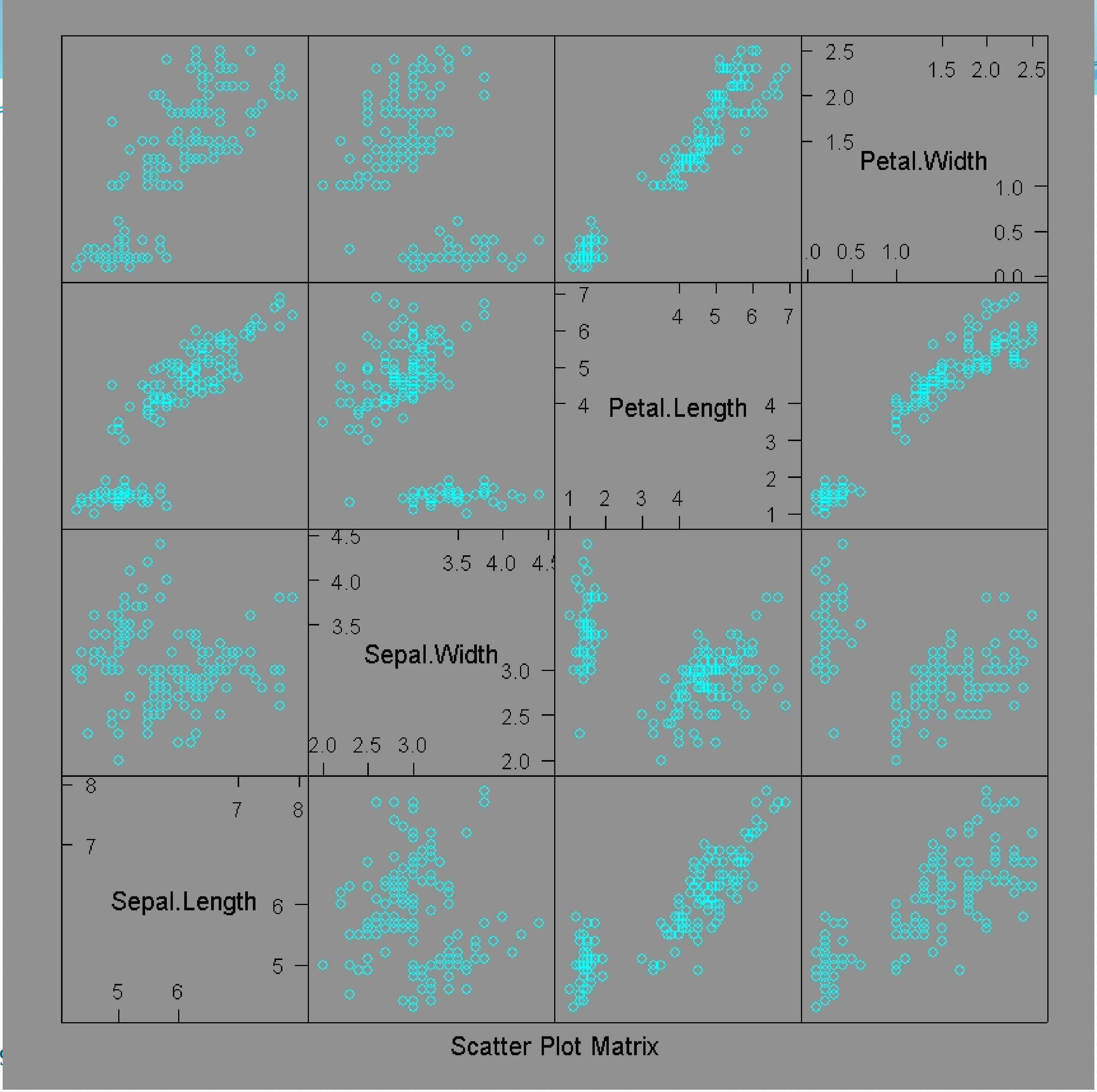
PCA on High Dimensional Data

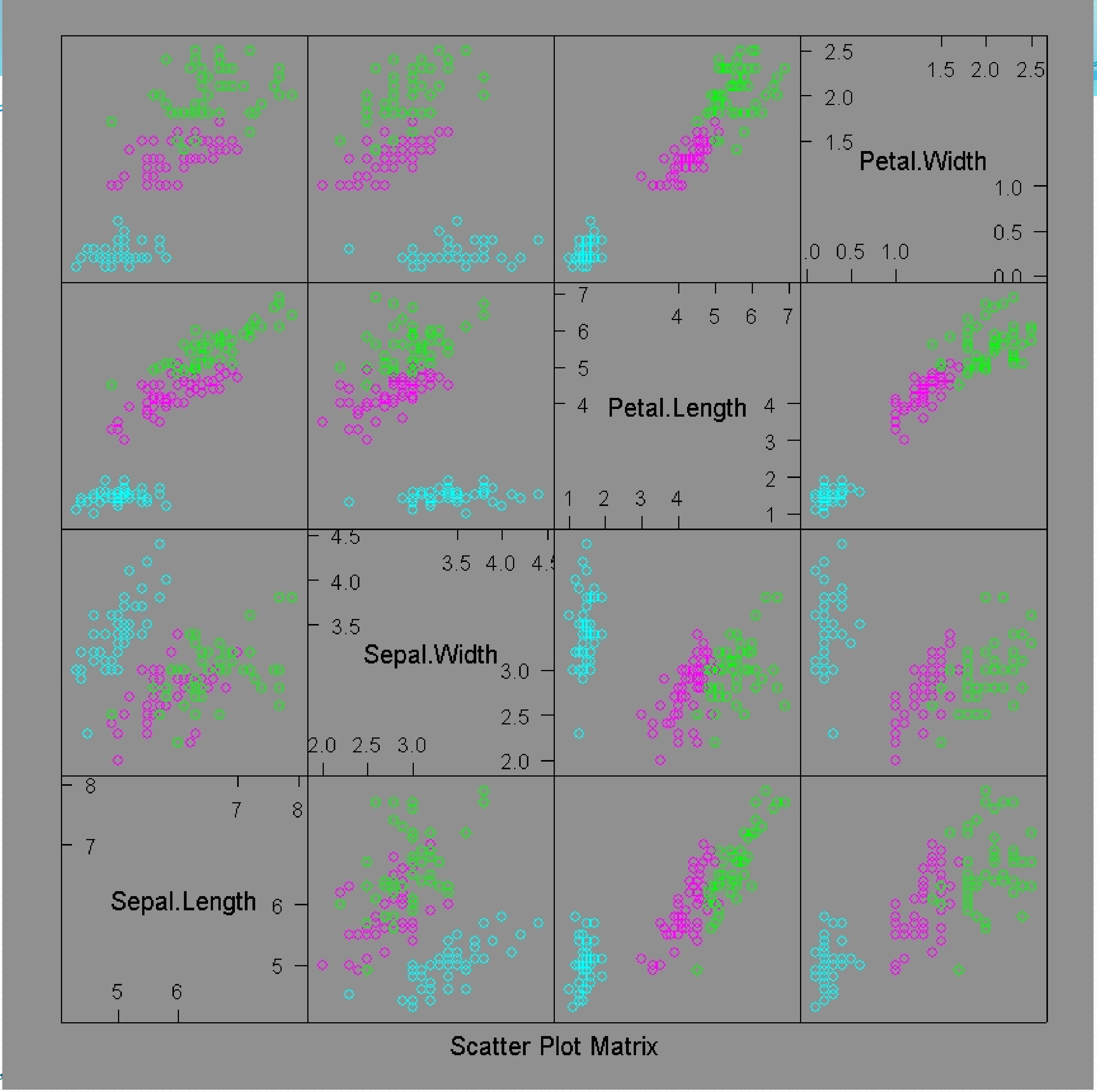
- If p is much larger than n, say p = 50,000 and n = 100, we can still do PCA.
- In general, there are as many principal components as the minimum of n and p.
- More precisely, there are as many principal components as the rank of X, which is the number of non-zero eigenvalues, and which is no greater than min(n, p)
- There is no particular reason why the first PC's are likely to be good for separating groups.
- PCA is *unsupervised learning* (doesn't use the class labels if any), not *supervised learning*.

Fisher’s Iris Data

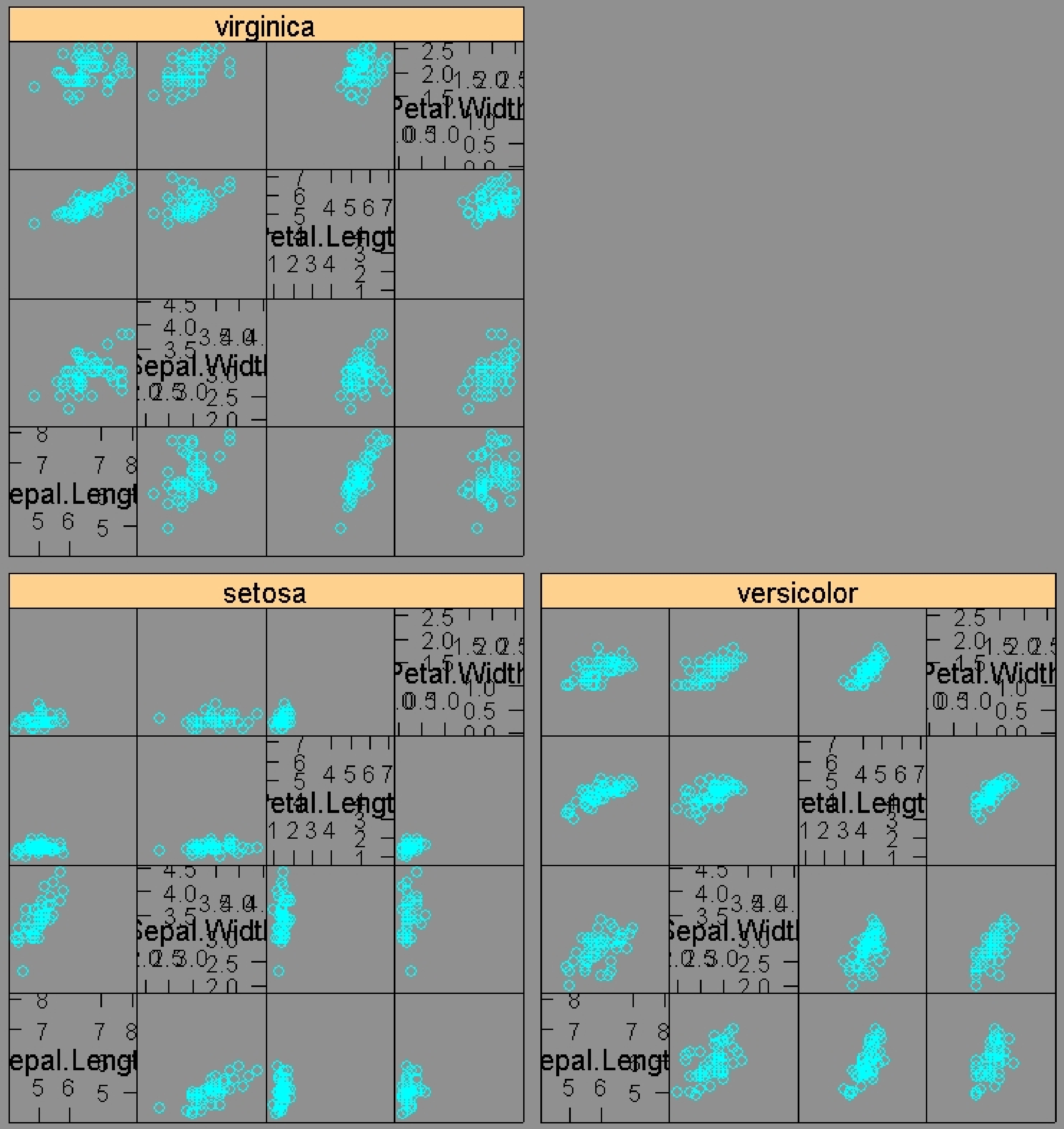
This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are _Iris setosa_, _versicolor_, and _virginica_.

```
> data(iris)
> help(iris)
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
> attach(iris)
> iris.dat <- iris[,1:4]
> splom(iris.dat)
> splom(iris.dat,groups=Species)
> splom(~ iris.dat | Species)
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```


Petal.Width
2.5
2.0
1.5
1.0
0.5
0.0
1.5 2.0 2.5
.0 0.5 1.0
Petal.Length
7
6
5
4
3
2
1
4 5 6 7
1 2 3 4
Sepal.Width
4.5
4.0
3.5
3.0
2.5
2.0
3.5 4.0 4.5
2.0 2.5 3.0
Sepal.Length
8
7
6
5
7 8
5 6
Scatter Plot Matrix

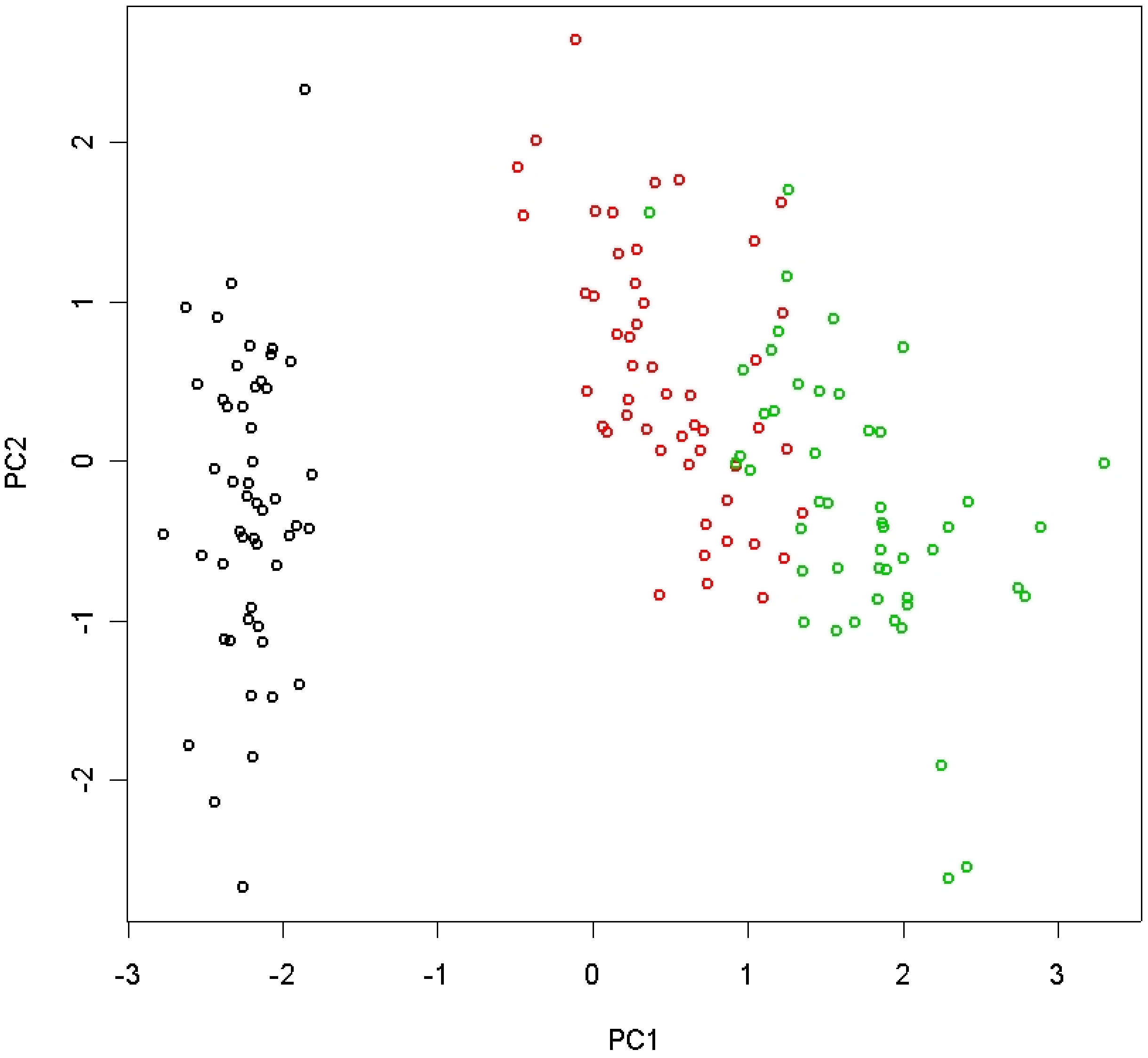


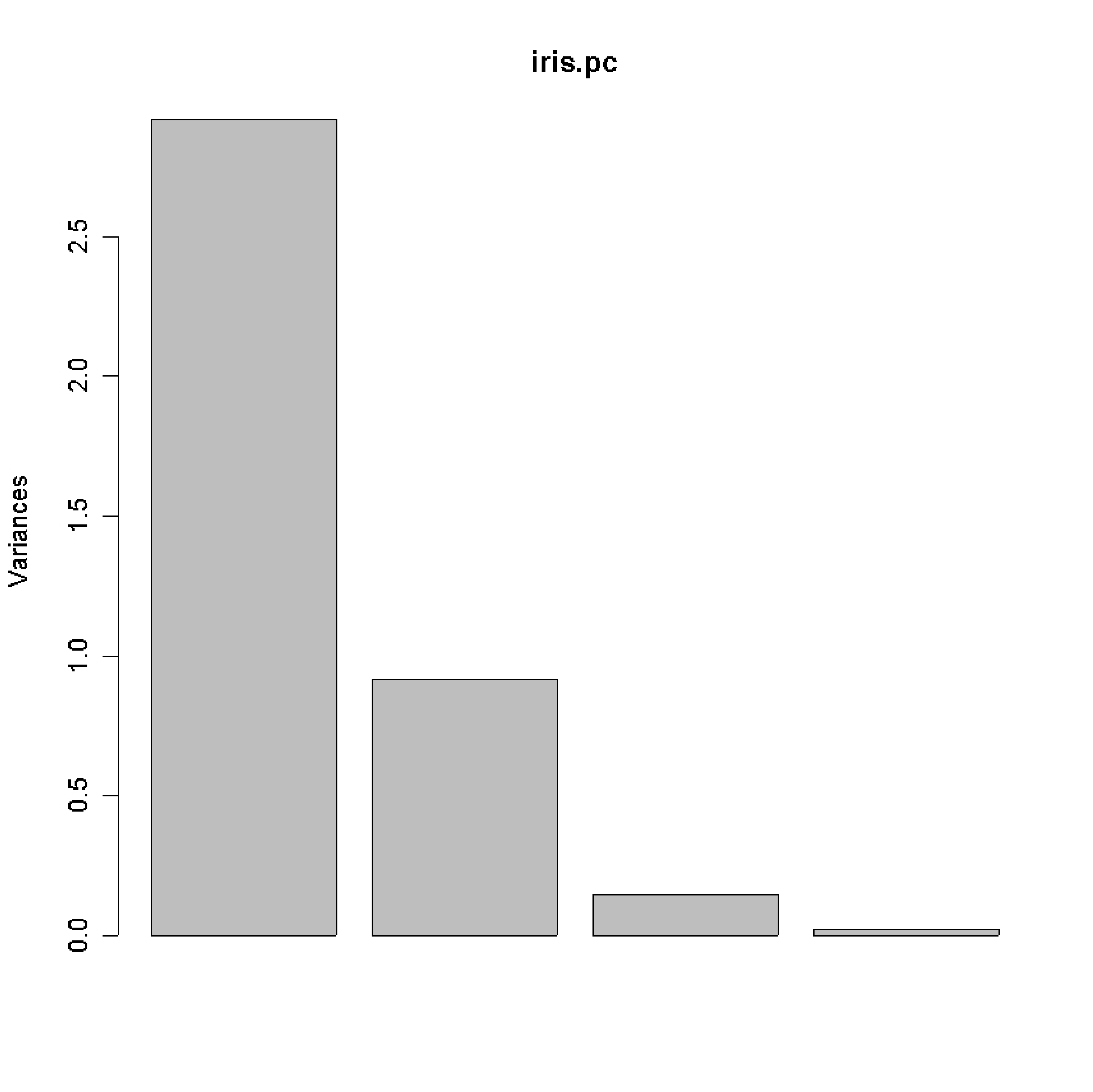
Petal.Width
2.5
2.0
1.5
1.0
0.5
0.0
1.5 2.0 2.5
.0 0.5 1.0
Petal.Length
7
6
5
4
3
2
1
4 5 6 7
1 2 3 4
Sepal.Width
4.5
4.0
3.5
3.0
2.5
2.0
3.5 4.0 4.5
2.0 2.5 3.0
Sepal.Length
8
7
6
5
7 8
5 6
Scatter Plot Matrix



Scatter Plot Matrix

```
> data(iris)
> iris.pc <- prcomp(iris[,1:4],scale=T)
> plot(iris.pc$x[,1:2],col=rep(1:3,each=50))
> names(iris.pc)
[1] "sdev"     "rotation" "center"   "scale"    "x"
> plot(iris.pc)
> iris.pc$sdev
[1] 1.7083611 0.9560494 0.3830886 0.1439265
> iris.pc$rotation
                    PC1         PC2        PC3        PC4
Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```


2
1
0
-1
-2
PC2
-3
-2
-1
0
1
2
3
PC1


iris.pc
Variances
0.0
0.5
1.0
1.5
2.0
2.5

Discriminant Analysis

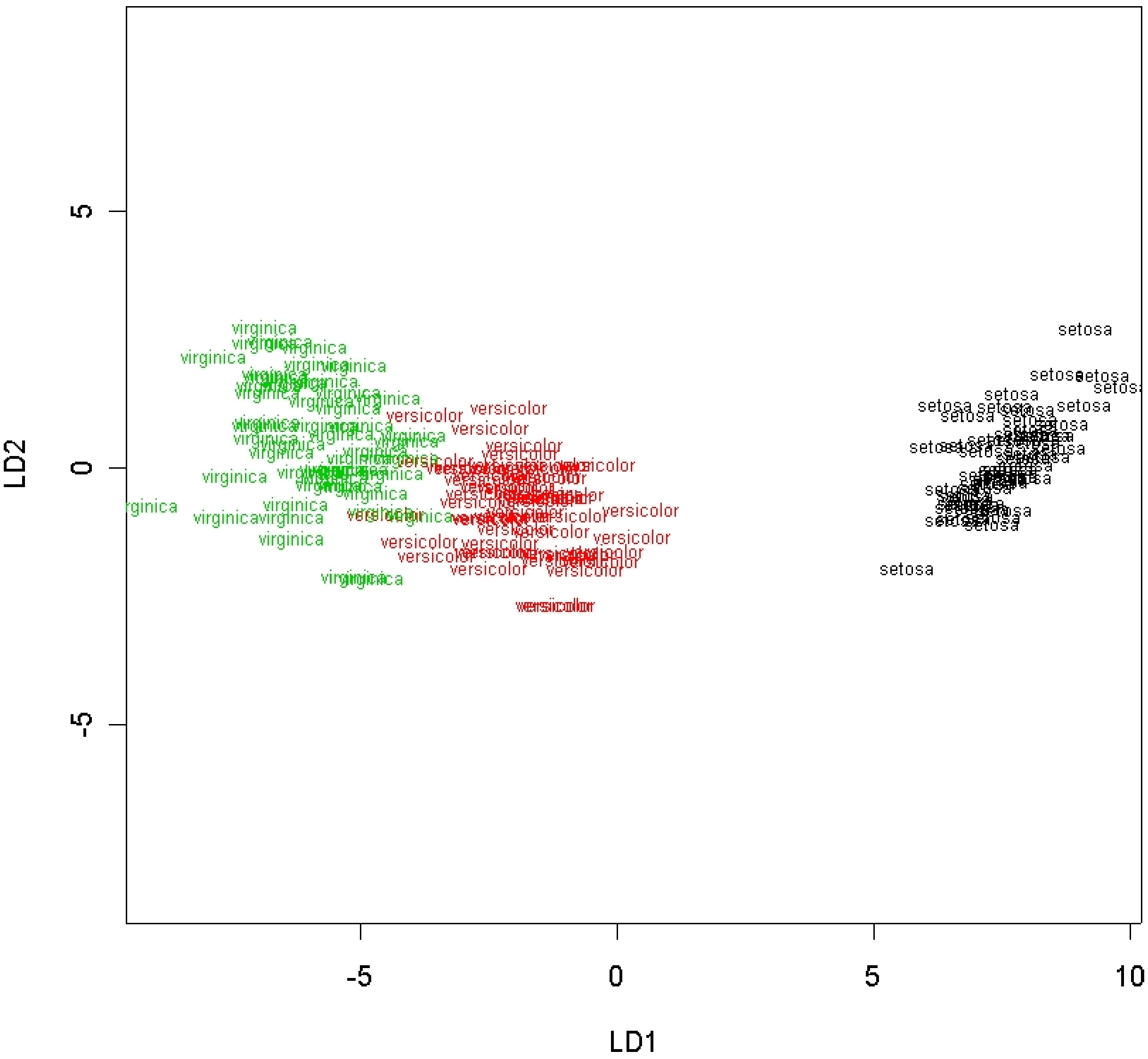
- An alternative to logistic regression for classification is discrimininant analysis
- This comes in two flavors, (Fisher's) Linear Discriminant Analysis or LDA and (Fisher's) Quadratic Discriminant Analysis or QDA
- In each case we model the shape of the groups and provide a dividing line/curve

- One way to describe the way LDA and QDA work is to think of the data as having for each group an elliptical distribution
- We allocate new cases to the group for which they have the highest likelihoods
- This provides a linear cut-point if the ellipses are assumed to have the same shape and a quadratic one if they may be different

```
> library(MASS)
> iris.lda <- lda(iris[,1:4],iris[,5])
> iris.lda
Call:
lda(iris[, 1:4], iris[, 5])

Prior probabilities of groups:
    setosa versicolor  virginica 
 0.3333333  0.3333333  0.3333333 

Group means:
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa            5.006       3.428        1.462       0.246
versicolor        5.936       2.770        4.260       1.326
virginica         6.588       2.974        5.552       2.026

Coefficients of linear discriminants:
                    LD1         LD2
Sepal.Length  0.8293776  0.02410215
Sepal.Width   1.5344731  2.16452123
Petal.Length -2.2012117 -0.93192121
Petal.Width  -2.8104603  2.83918785

Proportion of trace:
   LD1    LD2 
0.9912 0.0088 
```

```
> plot(iris.lda,col=rep(1:3,each=50))
> iris.pred <- predict(iris.lda)
> names(iris.pred)
[1] "class"     "posterior" "x"
> iris.pred$class[71:80]
 [1] virginica  versicolor versicolor versicolor versicolor
     versicolor versicolor
 [8] versicolor versicolor versicolor
Levels: setosa versicolor virginica
> iris.pred$posterior[71:80,]
         setosa versicolor    virginica
71 7.408118e-28  0.2532282 7.467718e-01
72 9.399292e-17  0.9999907 9.345291e-06
73 7.674672e-29  0.8155328 1.844672e-01
74 2.683018e-22  0.9995723 4.277469e-04
75 7.813875e-18  0.9999758 2.421458e-05
76 2.073207e-18  0.9999171 8.290530e-05
77 6.357538e-23  0.9982541 1.745936e-03
78 5.639473e-27  0.6892131 3.107869e-01
79 3.773528e-23  0.9925169 7.483138e-03
80 9.555338e-12  1.0000000 1.910624e-08

> sum(iris.pred$class != iris$Species)
[1] 3
```

This is an error rate of 3/150 = 2%


LD2
5
0
-5
-5
0
5
10
LD1
virginica
versicolor
setosa

```
iris.cv <- function(ncv,ntrials)
{
  nwrong <- 0
  n <- 0
  for (i in 1:ntrials)
  {
    test <- sample(150,ncv)
    test.ir <- data.frame(iris[test,1:4])
    train.ir <- data.frame(iris[-test,1:4])
    lda.ir <- lda(train.ir,iris[-test,5])
    lda.pred <- predict(lda.ir,test.ir)
    nwrong <- nwrong + sum(lda.pred$class != iris[test,5])
    n <- n + ncv
  }
  print(paste("total number classified = ",n,sep=""))
  print(paste("total number wrong = ",nwrong,sep=""))
  print(paste("percent wrong = ",100*nwrong/n,"%",sep=""))
}
> iris.cv(10,1000)
[1] "total number classified = 10000"
[1] "total number wrong = 213"
[1] "percent wrong = 2.13%"
```

Lymphoma Data Set

- Alizadeh et al. Nature (2000) "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling"
- We will analyze a subset of the data consisting of 61 arrays on patients with
  - 45 Diffuse large B-cell lymphoma (DLBCL/DL)
  - 10 Chronic lymphocytic leukaemia (CLL/CL)
  - 6 Follicular leukaemia (FL)

Data Available

- The original Nature paper
- The expression data in the form of unbackground corrected log ratios. A common reference was always on Cy3 with the sample on Cy5 (array.data.txt and array.data.zip). 9216 by 61
- A file with array codes and disease status for each of the 61 arrays, ArrayID.txt

Identify Differentially Expressed Genes

- We will assume that the log ratios are on a reasonable enough scale that we can use them as is
- For each gene, we can run a one-way ANOVA and find the p-value, obtaining 9,216 of them. We can use apply() or genediff() from LMGene
- Adjust p-values with p.adjust or padjust
- Identify genes with small adjusted p-values

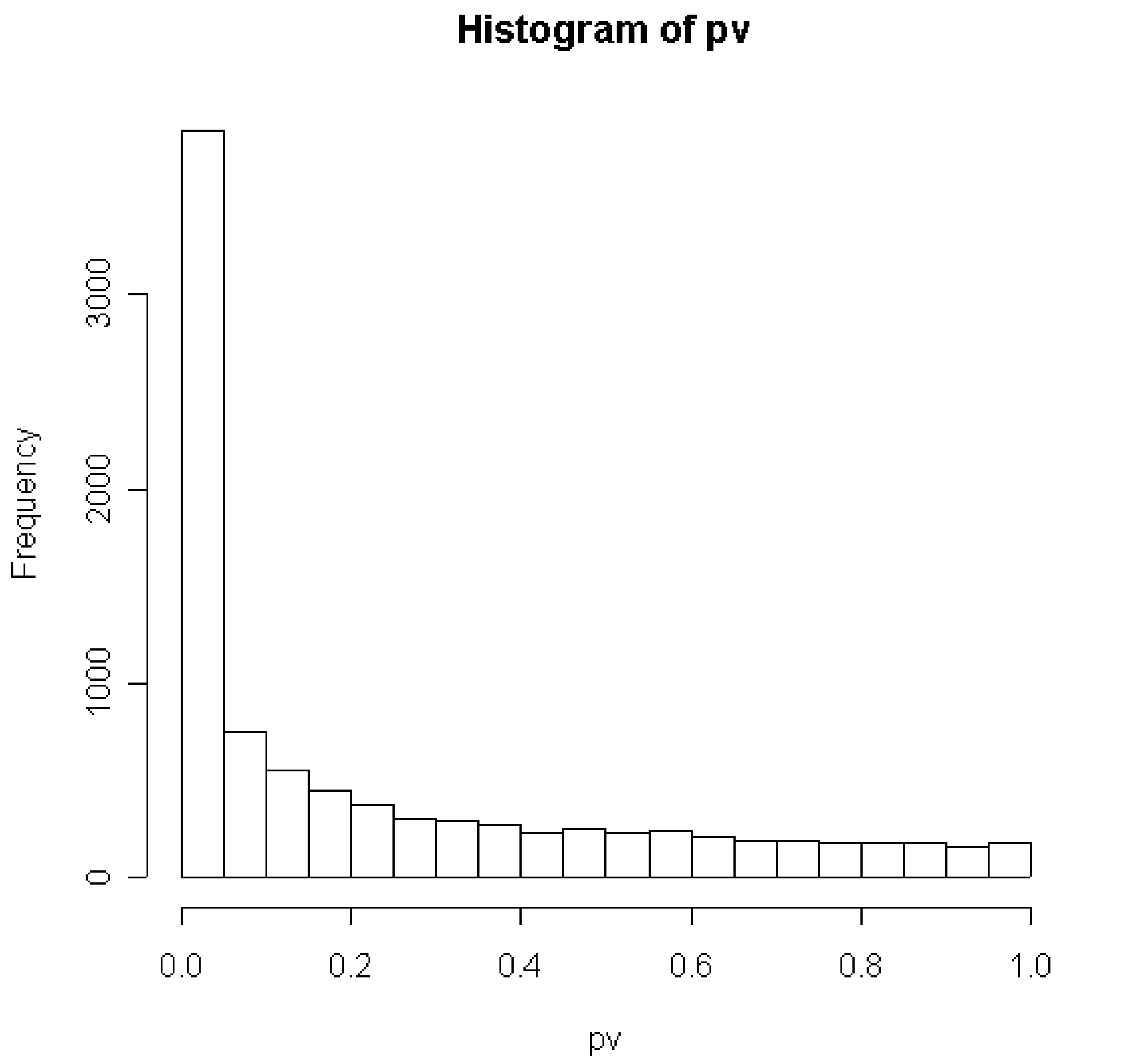
Develop Classifier

- Reduce dimension with ANOVA gene selection or with PCA. (We could also use stepwise logistic regression.)
- Use logistic regression or LDA.
- Evaluate the four possibilities and their sub-possibilities with cross validation. With 61 arrays one could reasonable omit 10% or 6 at random.

Differential Expression

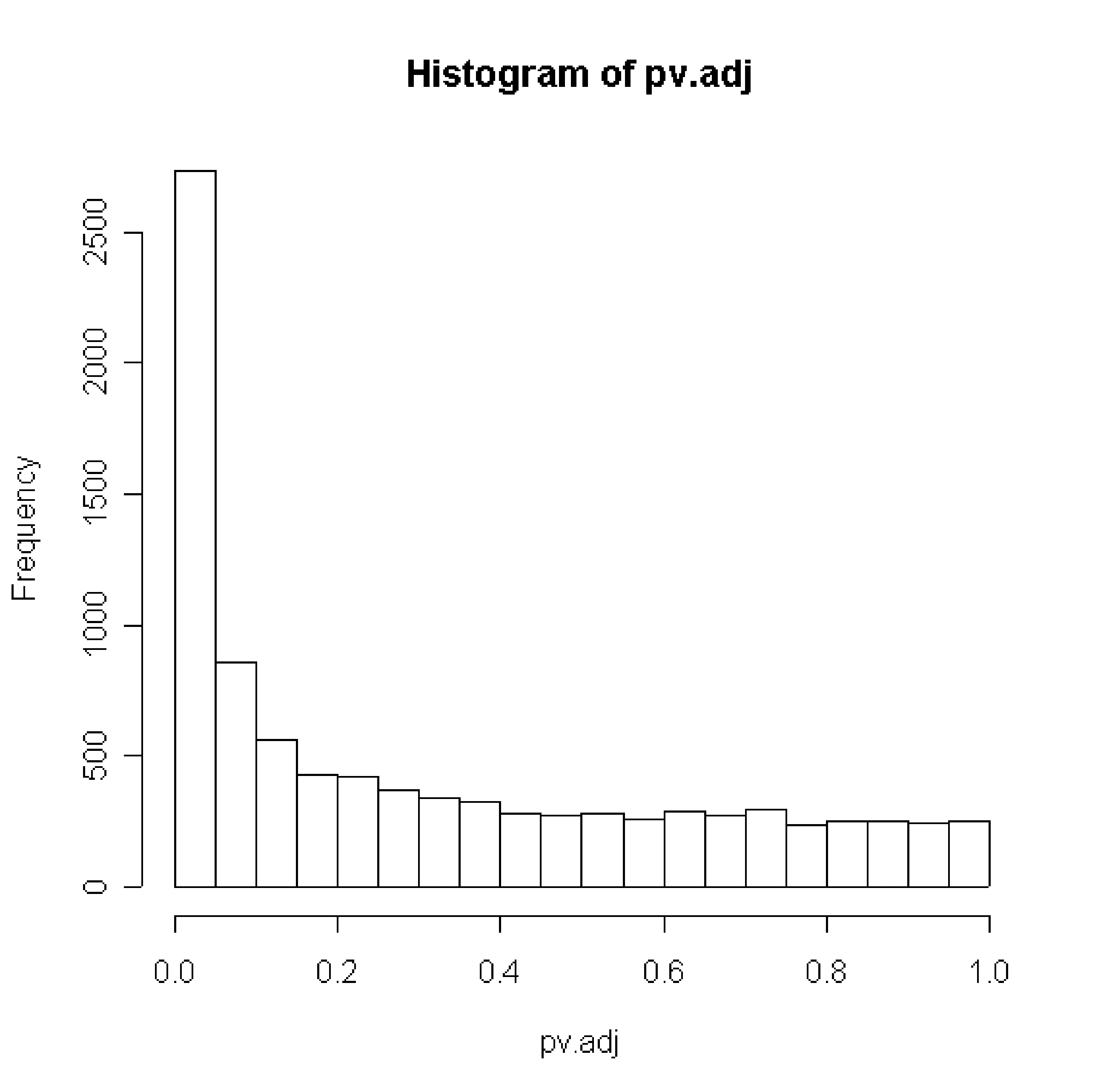
- We can locate genes that are differentially expressed; that is, genes whose expression differs systematically by the type of lymphoma.
- To do this, we could use lymphoma type to predict expression, and see when this is statistically significant.
- For one gene at a time, this is ANOVA.

- It is almost equivalent to locate genes whose expression can be used to predict lymphoma type, this being the reverse process.
- If there is significant association in one direction there should logically be significant association in the other
- This will not be true exactly, but is true approximately
- We can also easily do the latter analysis using the expression of more than one gene using logistic regression, LDA, and QDA


Histogram of pv
Frequency
0
1000
2000
3000
0.0
0.2
0.4
0.6
0.8
1.0
pv

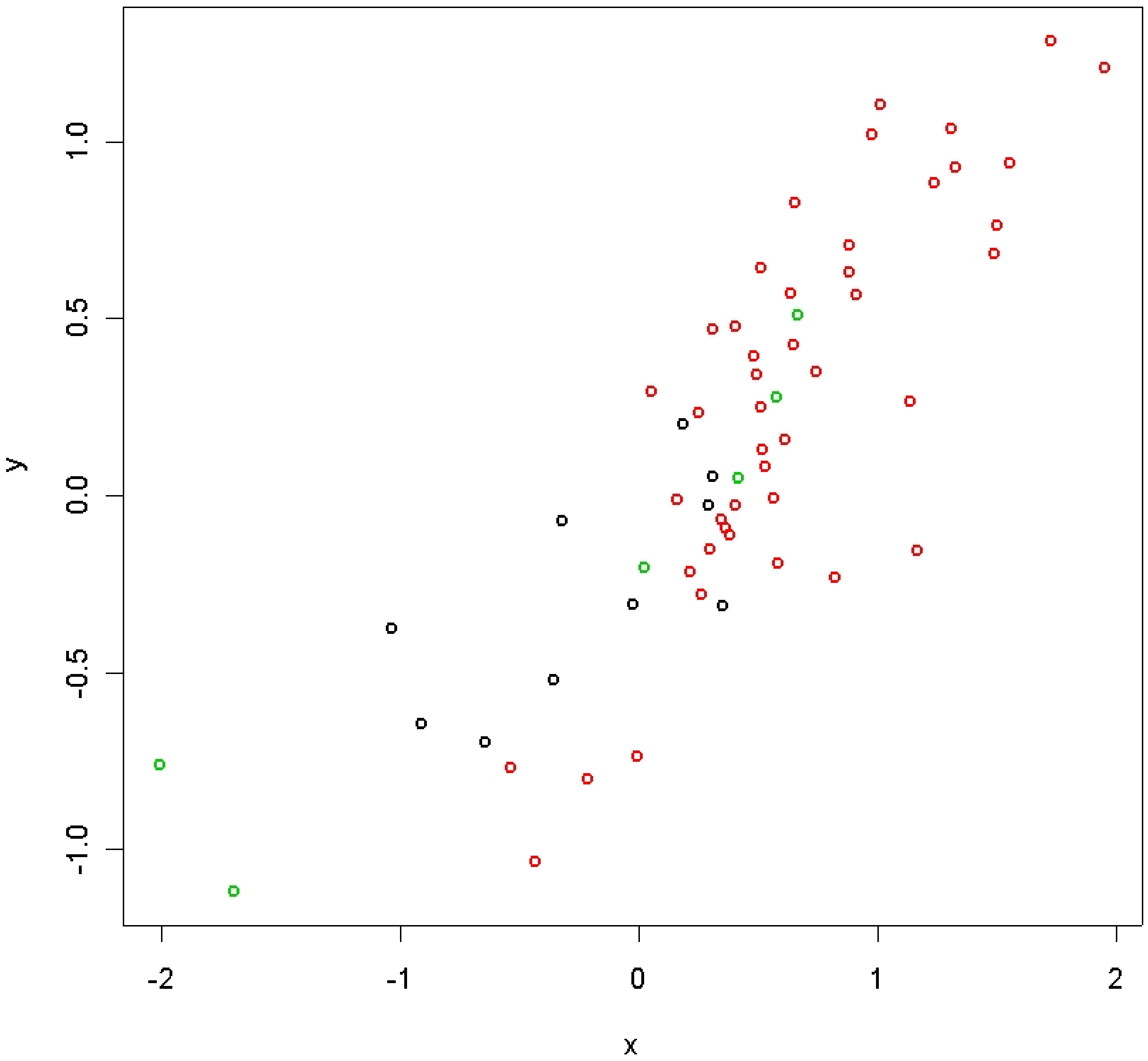
Significant Genes

- There are 3845 out of 9261 genes that have significant p-values from the ANOVA of less than 0.05, compared to 463 expected by chance
- There are 2733 genes with FDR adjusted p-values less than 0.05
- There are only 184 genes with Bonferroni adjusted p-values less than 0.05


Histogram of pv.adj
Frequency
0
500
1000
1500
2000
2500
0.0
0.2
0.4
0.6
0.8
1.0
pv.adj

Logistic Regression

- We will use logistic regression to distinguish DLBCL from CLL and DLBCL from FL
- We will do this first by choosing the variables with the smallest overall p-values in the ANOVA
- We will then evaluate the results within sample and by cross validation


1.0
0.5
0.0
-0.5
-1.0
y
-2
-1
0
1
2
x

Within Sample Errors

Number of Variables	DL/CL Errors	DL/FL Errors
1	7	4
2	7	4
3	5	5
4	0	3
5	0	2
6	0	0

Evaluation of performance

- Within sample evaluation of performance like this is unreliable
- This is especially true if we are selecting predictors from a very large set
- One useful yardstick is the performance of random classifiers

Number of Variables	Percent Errors
1	24.0%
2	24.7%
3	23.3%
4	24.7%
5	28.7%
6	24.7%
7	26.3%
8	23.6%
9	25.7%
10	24.3%

Left is CV performance of best k variables

Random = 25.4%

Conclusion

- Logistic regression on the variables with the smallest p-values does not work very well
- This cannot be identified by looking at the within sample statistics
- Cross validation is a requirement to assess performance of classifiers

```
alkfos {ISwR}     R Documentation

Alkaline phosphatase data

Repeated measurements of alkaline phosphatase in a
randomized trial of Tamoxifen treatment of breast cancer
patients.

Format

A data frame with 43 observations on the following 8
  variables.

grp a numeric vector, group code (1=placebo, 2=Tamoxifen).
c0  a numeric vector, concentration at baseline.
c3  a numeric vector, concentration after 3 months.
c6  a numeric vector, concentration after 6 months.
c9  a numeric vector, concentration after 9 months.
c12 a numeric vector, concentration after 12 months.
c18 a numeric vector, concentration after 18 months.
c24 a numeric vector, concentration after 24 months.
```

Exercises (for later)

- In the ISwR data set alkfos, do a PCA of the placebo and Tamoxifen groups separately, then together. Plot the first two principal components of the whole group with color coding for the treatment and control subjects.
- Conduct a linear discriminant analysis of the two groups using the 7 variables. How well can you predict the treatment? Is this the usual kind of analysis you would see?
- Use logistic regression to predict the group based on the measurements. Compare the in-sample error rates.
- Use cross-validation with repeated training subsamples of 38/43 and test sets of size 5/43. What can you now conclude about the two methods?