Principal Component Analysis

Original Data: $X_i=(X_{i1}, X_{i2}, \ldots, X_{ip})'$

linear Combination: $Z_{ij} = \underbrace{a_{j1}X_{i1} + a_{j2}X_{i2} + \cdots + a_{jp}X_{ip}}_{\text{linear Combination of the p variables of ith observation}} = a_j' \cdot X_i$ where $a_j = \begin{pmatrix} a_{j1} \\ \vdots \\ a_{jp} \end{pmatrix}$

⇓

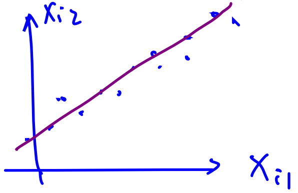
of variables

Principal Component Analysis

- When a very large number of variables is measured on each subject, interpreting the data may be difficult.
- It is often possible to reduce the dimensionality of the data by finding a smaller set of linear combinations of the variables that preserve most of the variability across subjects.
- These linear combinations are called *principal components*.
- Finding the principal components is often an early step in a more complex analysis. Principal component scores can be used to build regression models, to cluster subjects, or to build classification rules.

Why linear combination of variables?

$X_i=(X_{i1}, X_{i2})'$



=> The two variables are highly correlated. => we can use one variable to predict the other variable

⇓

Key of principal component analysis is the correlation among variables, so we work on the covariance/correlation matrix for PCA

⇐ We can do dimension reduction by only keeping one variable, or a linear combination of the two variables

Principal Component Analysis

- Principal components (PCs) are derived from eigenvectors of a covariance matrix (or correlation matrix).

- The data are projected onto hyperplanes of lower dimension defined by the eignevectors.

- Corresponding eigenvalues give the variation in principal component scores.

- PCs do do not require the assumption of multivariate normality.

- If multivariate normality holds, however, some distributional properties of PCs can be established.

Objectives

- **Reduce Dimensionality:** Instead of analyzing variation in a large number, say p, variables as they vary from subject to subject, analyze variation in a much smaller number of principal component scores.

- **Develop Summary Indices:** Find meaningful, or useful, linear combinations of the original variables, such as food quality, consumer satisfaction, or economic indicies.

- ⇒ **Cluster Analysis:** Visually display differences between groups or clusters.

- **Data Screening:** Detect outliers (extreme data vectors) or strong associations among variables

Population Principal Components

- Let the random vector $\underset{\sim}{\mathbf{X}} = (X_1, X_2, \cdots, X_p)'$ have covariance matrix Σ with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$.

- Consider p linear combinations of the variables

$$\begin{array}{rcccl} Y_1 & = & \underset{\sim 1}{\mathbf{a}}{}' \underset{\sim}{\mathbf{X}} & = & a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p \\ Y_2 & = & \underset{\sim 2}{\mathbf{a}}{}' \underset{\sim}{\mathbf{X}} & = & a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_p & = & \underset{\sim p}{\mathbf{a}}{}' \underset{\sim}{\mathbf{X}} & = & a_{p1}X_1 + a_{p2}X_2 + \cdots + a_{pp}X_p \end{array} .$$

- Note that $\text{Var}(Y_i) = \text{Var}(\underset{\sim}{a_i}' \cdot \underset{\sim}{X}) = \underset{\sim}{a_i}' \cdot \text{Var}(\underset{\sim}{X}) \cdot \underset{\sim}{a_i} = \underset{\sim}{a_i}' \cdot \Sigma \cdot \underset{\sim}{a_i}$

$$\begin{aligned} \text{Var}(Y_i) &= \underset{\sim i}{\mathbf{a}}{}' \Sigma \underset{\sim i}{\mathbf{a}}, \qquad i = 1, 2, ..., p \\ \text{Cov}(Y_i, Y_k) &= \underset{\sim i}{\mathbf{a}}{}' \Sigma \underset{\sim k}{\mathbf{a}}, \qquad i, k = 1, 2, ..., p. \end{aligned}$$

$\text{Cov}(Y_i, Y_k) = \text{Cov}(\underset{\sim}{a_i}' \underset{\sim}{X}, \underset{\sim}{a_k}' \underset{\sim}{X}) = \underset{\sim}{a_i}' \cdot \text{Var}(X) \cdot \underset{\sim}{a_k} = \underset{\sim}{a_i}' \cdot \Sigma \underset{\sim}{a_k}$

Population Principal Components

- *Principal Components* are uncorrelated linear combinations of the original variables determined sequentially as follows:

  $\underset{\sim}{a}_1$: 1st PC direction/loadings

  - The first PC is the linear combination $Y_1 = \underset{\sim}{\mathbf{a}}_1' \underset{\sim}{\mathbf{X}}$ that maximizes Var$(Y_1) = \underset{\sim}{\mathbf{a}}_1' \Sigma \underset{\sim}{\mathbf{a}}_1$ subject to $\underset{\sim}{\mathbf{a}}_1' \underset{\sim}{\mathbf{a}}_1 = 1$.

  - The second PC is the linear combination $Y_2 = \underset{\sim}{\mathbf{a}}_2' \underset{\sim}{\mathbf{X}}$ that maximizes Var$(Y_2) = \underset{\sim}{\mathbf{a}}_2' \Sigma \underset{\sim}{\mathbf{a}}_2$ subject to $\underset{\sim}{\mathbf{a}}_2' \underset{\sim}{\mathbf{a}}_2 = 1$ and Cov$(Y_1, Y_2) = \underset{\sim}{\mathbf{a}}_1' \Sigma \underset{\sim}{\mathbf{a}}_2 = 0$.

    $\vdots$

  - The ith PC is the linear combination $Y_i = \underset{\sim}{\mathbf{a}}_i' \underset{\sim}{\mathbf{X}}$ that maximizes Var$(Y_i) = a_i' \Sigma a_i$ subject to $\underset{\sim}{\mathbf{a}}_i' \underset{\sim}{\mathbf{a}}_i = 1$ and Cov$(Y_i, Y_k) = \underset{\sim}{\mathbf{a}}_i' \Sigma \underset{\sim}{\mathbf{a}}_k = 0$ for $k < i$.

    $\vdots$

Population Principal Components

- Let Σ have eigenvalue-eigenvector pairs $(\lambda_i, \underset{\sim i}{\mathbf{e}})$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$. Then, the ith principal component is given by

$$Y_i = \underset{\sim i}{\mathbf{e}}' \underset{\sim}{\mathbf{X}} = e_{i1}X_1 + e_{i2}X_2 + \cdots + e_{ip}X_p, \quad i = 1, ..., p.$$

- Then,

$$\text{Var}(Y_i) = \underset{\sim i}{\mathbf{e}}'\Sigma\underset{\sim i}{\mathbf{e}} = \underset{\sim i}{\mathbf{e}}'\lambda_i\underset{\sim i}{\mathbf{e}} = \lambda_i\underset{\sim i}{\mathbf{e}}'\underset{\sim i}{\mathbf{e}} = \lambda_i, \text{ since } \underset{\sim i}{\mathbf{e}}'\underset{\sim i}{\mathbf{e}} = 1$$

$$\text{Cov}(Y_i, Y_k) = \underset{\sim i}{\mathbf{e}}'\Sigma\underset{\sim k}{\mathbf{e}} = \underset{\sim i}{\mathbf{e}}'\lambda_k\underset{\sim k}{\mathbf{e}} = 0, \text{ since } \underset{\sim i}{\mathbf{e}}'\underset{\sim k}{\mathbf{e}} = 0$$

- Using properties of the trace of Σ we have

$$\sigma_{11}+\sigma_{22}+\cdots+\sigma_{pp} = \sum_i \text{Var}(X_i) = \lambda_1+\lambda_2+\cdots+\lambda_p = \sum_i \text{Var}(Y_i).$$

Population Principal Components

- Because the total population variance, $trace(\Sigma)$, is equal to the sum of the variances of the principal components, $\sum_i \lambda_i$, we say that

  $$\frac{\lambda_k}{\lambda_1 + \lambda_2 + ... + \lambda_p}$$

  is the proportion of the total variance associated with (or explained by) the kth principal component.

- If a large proportion of the total variance (say 80% or 90%) is explained by the first k PCs, then we can ignore the original p variables and restrict attention to the first k PCs without much loss of information about variation among members of the population.

Spectral Decomposition of a Covariance Matrix

Mathematically any covariance (or correlation) matrix can be expressed as

$$\Sigma = E \Lambda E'$$

where

$$E = \begin{bmatrix} e_{11} & e_{21} & \dots & e_{p1} \\ e_{12} & e_{22} & \dots & e_{p2} \\ \vdots & \vdots & & \vdots \\ e_{1p} & e_{2p} & \dots & e_{pp} \end{bmatrix} = \left[\underset{\sim}{\mathbf{e}}_1 \; \underset{\sim}{\mathbf{e}}_2 \; \dots \; \underset{\sim}{\mathbf{e}}_p \right]$$

is the matrix with eigenvectors as the columns and

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \lambda_p \end{bmatrix}$$

is a diagonal matrix of eigenvalues.

Spectral Decomposition of a Covariance Matrix

The spectral decomposition is also expressed as

$$\Sigma = E\Lambda E' = \lambda_1 \underset{\sim 1}{\mathbf{e}} \underset{\sim 1}{\mathbf{e}}^{'} + \lambda_2 \underset{\sim 2}{\mathbf{e}} \underset{\sim 2}{\mathbf{e}}^{'} + \cdots + \lambda_p \underset{\sim p}{\mathbf{e}} \underset{\sim p}{\mathbf{e}}^{'}$$

If the first k eigenvalues account for a large portion of the total variance, the covariance matrix (or correlation matrix) can be well approximated by first k terms in the decomposition, i.e.,

$$\Sigma = E\Lambda E' \approx \lambda_1 \underset{\sim 1}{\mathbf{e}} \underset{\sim 1}{\mathbf{e}}^{'} + \lambda_2 \underset{\sim 2}{\mathbf{e}} \underset{\sim 2}{\mathbf{e}}^{'} + \cdots + \lambda_k \underset{\sim k}{\mathbf{e}} \underset{\sim k}{\mathbf{e}}^{'}$$

It is desirable to have "k' much smaller than the original number of variables p.

Principal Component Scores

- Principal component scores are generally centered at zero.

- The centered score of the k-th principal component for the m-th member of the population is

$$Y_{\mathrm{mk}} = e_{k1}(X_{\mathrm{m1}} - \mu_1) + e_{k2}(X_{\mathrm{m2}} - \mu_2) + \cdots + e_{kp}(X_{\mathrm{mp}} - \mu_p)$$

- The expected value of this centered principal component score is zero.

- The population variance of the scores for the i-th principal component is λ_i, the i-th largest eigenvalue.

- The principal component scores are interpreted by understanding what low and high scores represent. This is determined by looking at the signs and relative sizes of the coefficients.

Principal Component Scores

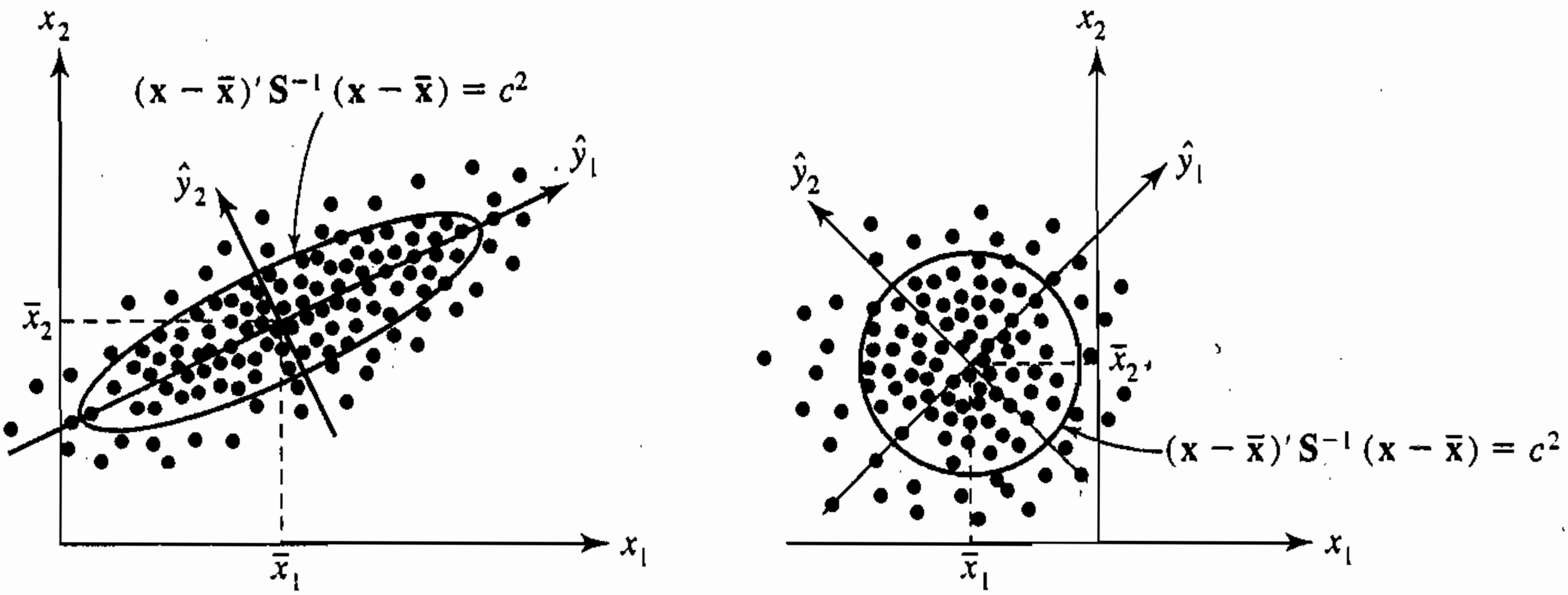
- The correlation between the ith principal component and the kth original variable,

$$\rho_{Y_i,X_k} = \frac{e_{ik}\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}}$$

  is a measure of the contribution of the kth variable to the variation of the ith principal component scores.

- If you extract principal components from standardized variables (eigenvectors of the correlation matrix), the k-th element of the i-th eigenvector, e_{ik}, directly determines how much the k-th standardized variable contributes to the score of the i-th principal component.

Two examples of PCs from MVN data



(a) $\hat{\lambda}_1 > \hat{\lambda}_2$

(b) $\hat{\lambda}_1 \doteq \hat{\lambda}_2$

PCs from Standardized Variables

- When variables are measured on different scales it is useful to standardize the variables before extracting the PCs, i.e., compute z-scores:

$$Z_i = \frac{(X_i - \mu_i)}{\sqrt{\sigma_{ii}}},$$

- Note that $\text{Cov}(\underset{\sim}{\mathbf{Z}}) = \text{Corr}(\underset{\sim}{\mathbf{X}})$, the correlation matrix of the original variables.
- Let $(\lambda_k, \underset{\sim}{\mathbf{e}}_k)$ denote the k-th eigenvalue-eigenvector pair of $\text{Corr}(\underset{\sim}{\mathbf{X}})$. Then, the score of the k-th principal component is

$$Y_k = e_{k1}Z_1 + e_{k2}Z_2 + \cdots + e_{kp}Z_p$$

PCs from Standardized Variables

Proceeding as before,

$$\begin{aligned} \text{trace(correlation matrix)} &= \sum_{i=1}^{p} \lambda_i = \sum_{i=1}^{p} \text{Var}(Z_i) = p \\ \rho_{Y_i, Z_k} &= e_{ik}\sqrt{\lambda_i}, \quad i, k = 1, ..., p. \end{aligned}$$

Further,

$$\left(\begin{array}{c} \text{Proportion of standardized} \\ \text{population variance due} \\ \text{to the } i\text{th principal component} \end{array} \right) = \frac{\lambda_i}{p}, \quad i = 1, ..., p,$$

where the λ_i are eigenvalues of the correlation matrix.

PCs from Standardized Variables

- In general, the PCs extracted from Σ=Cov($\underset{\sim}{\mathbf{Z}}$) and from Corr($\underset{\sim}{\mathbf{X}}$) will not be the same.

- Standardizing variables has important consequences.

- When should one standardize the variables before computing PCs?

PCs from Standardized Variables

- If variables are measured on very different scales (e.g. patient weights in kg vary from 40 to 100, protein concentration in ppm varying between 1 and 10), then the variables with the larger variances will dominate.

- When one variable has a much larger variance than any of the other variables, we will end up with a single PC that is essentially proportional to the dominating variable.

- Consider standardizing the variables when
  - different variables have greatly different variances (this makes all of the variables equally important)
  - you do not want changes in measurement scales to affect the results
  - you want to give more emphasis to describing correlations and less emphasis to describing variances of variables

PCs from Uncorrelated Variables

- If x_1, x_2, ..., x_p are uncorrelated random variables, then Σ is a diagonal matrix with elements $\sigma_{11} = Var(x_1)$, $\sigma_{22} = Var(x_2)$, ..., $\sigma_{pp} = Var(x_p)$. (Suppose $\sigma_{11} \geq \sigma_{22} \geq \cdots \geq \sigma_{pp}$).

- The eigenvalues in this case are $\lambda_i = \sigma_{ii}$ and one choice for the corresponding eigenvector is

$$\underset{\sim}{\mathbf{e}}_i = \left[\begin{array}{cccccc} 0 & ... & 0 & 1 & 0 & ...0 \end{array} \right]'.$$

- Since $\underset{\sim}{\mathbf{e}}_i' \underset{\sim}{\mathbf{X}} = x_i$ we note that the PCs are just the original variables. Thus, we gain nothing by trying to extract the PCs when the x_i's are uncorrelated.

Sample Principal Components

- If $\underset{\sim 1}{\mathbf{X}}, \underset{\sim 2}{\mathbf{X}}, ..., \underset{\sim n}{\mathbf{X}}$ is a random sample of $p-$dimensional vectors from a distribution with mean vector $\underset{\sim}{\boldsymbol{\mu}}$ and covariance matrix Σ, then the sample mean vector, sample covariance matrix and sample correlation matrix are $\underset{\sim}{\bar{\mathbf{X}}}, S,$ and R, respectively.

- Eigenvalue-eigenvector pairs of S are denoted $(\hat{\lambda}_i, \underset{\sim i}{\hat{\mathbf{e}}})$ and the ith sample PC is given by

$$\hat{y}_i = \underset{\sim i}{\hat{\mathbf{e}}}'x = \hat{e}_{i1}x_1 + \hat{e}_{i2}x_2 + \cdots + \hat{e}_{ip}x_p, \quad i = 1, ..., p.$$

- The sample variance of the ith PC is $\hat{\lambda}_i$ and the sample correlation between $(\hat{y}_i, \hat{y}_k)$ is zero for all $i \neq k$.

- The total sample variance $s_{11} + s_{22} + ... + s_{pp}$ is equal to $\hat{\lambda}_1 + ... + \hat{\lambda}_p$ and the relative contribution of the kth variable to the ith sample PC is given by $r_{\hat{y}_i, x_k}$.

Sample Principal Components

- Estimated PC scores are centered by subtracting the sample mean vector from each data vector: $(\underset{\sim}{\mathbf{X}}_i - \underset{\sim}{\bar{\mathbf{X}}})$.

- The estimated scores for the m-th subject on the k-th centered PC is $\widehat{y}_{\mathrm{mk}} = \underset{\sim}{\widehat{\mathrm{e}}}{}'_k(\underset{\sim}{\mathbf{X}}_{\mathrm{m}} - \underset{\sim}{\bar{\mathbf{X}}})$ where $\widehat{e}_k$ is the k-th eigenvector of S.

- If $\widehat{y}_{\mathrm{mk}}$ is the score on the k-th sample PC for the mth subject in the sample, then the sample mean of the scores for the kth sample PC (averaging across subjects in the sample) is zero:

$$\bar{\widehat{y}}_k = \frac{1}{n}\sum_{\mathrm{m}=1}^{n} \underset{\sim}{\widehat{\mathrm{e}}}{}'_k(\underset{\sim}{\mathbf{X}}_{\mathrm{m}} - \underset{\sim}{\bar{\mathbf{X}}}) = \frac{1}{n}\underset{\sim}{\widehat{\mathrm{e}}}{}'_k \sum_{\mathrm{m}=1}^{n}(\underset{\sim}{\mathbf{X}}_{\mathrm{m}} - \underset{\sim}{\bar{\mathbf{X}}}) = 0.$$

- The sample variance of the kth sample PC is $\widehat{\lambda}_k$, the k-th largest eigenvalue of S.

- Sample principal components are uncorrelated.

Carapace Measurements for Female Turtles

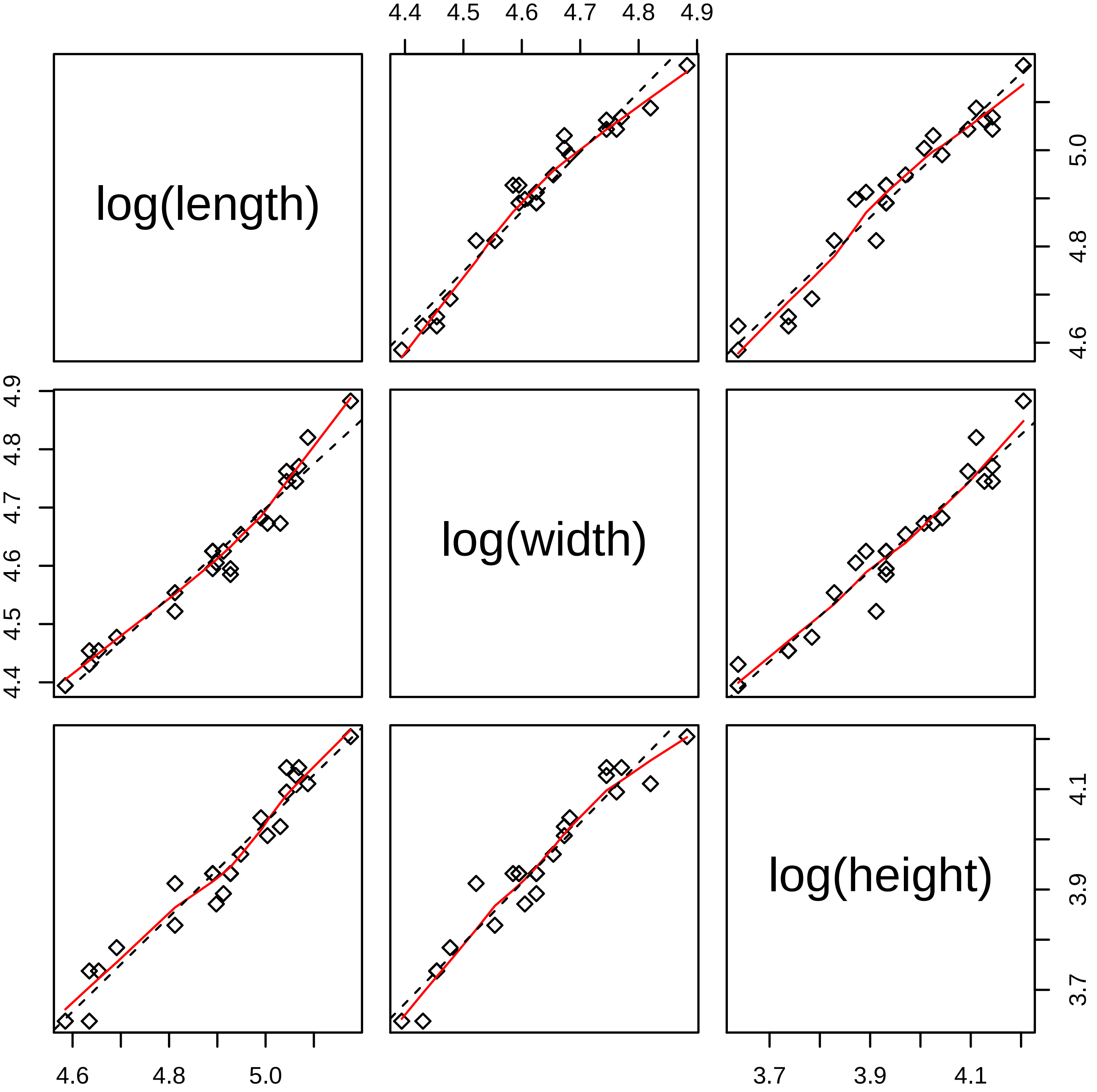
- Data on three dimensions of female turtle carapaces (shells):
  - X_1=log(carapace length)
  - X_2=log(carapace width)
  - X_3=log(carapace height)
- Since the measurements are all on the same scale (mm), the PCs may be extracted from the sample covariance matrix S
- See R and SAS code and output.

R Code

```
#  This code creates scatter plot matrices and principal components
#  for the turtle example considered in the lecture. This code is
#  posted as  turtles.R.  The data are posted as  turtles.dat
#
#  This file has data on both male (coded 2)  and female (coded 1)
#  turtles.  There is  one line of data for each turtle with four
#  numbers on each line.  The first column has the sex code,
#  the next three columns  provide the length, height, and
#  width of the carapace, respectively.

   turtle.all <- read.table(file="turtles.dat",
         header=F, col.names=c("sex", "length", "width", "height"))
  head(turtle.all)
```

```
#  Select the female turtles (coded as 1) and delete
#  the first column.

 turtle.f<-turtle.all[turtle.all[,1]=="1", -1]

#  Compute the number of female turtles

   n<-dim(turtle.f)[1]

#  Compute natural logs of each measurement

   turtle.f <- log(turtle.f)

#  Create a scatter plot matrix. The panel.smooth function
#  passes a smooth curve through each plot.  The abline
#  function fits a straight line to each plot.
```

```
   par(pch=5,fin=c(5,5))
   pairs(turtle.f,labels=c("log(length)",
         "log(width)","log(height)"),
         panel=function(x,y){panel.smooth(x,y)
          abline(lsfit(x,y),lty=2) })

#  Compute principal components from the sample  covariance matrix.
#  This function creates a list with the following components
#      sdev:  standard deviations of the component
#             scores (square roots of eigenvalues
#             of the sample covariance matrix)
#  rotation:  The coefficients needed to compute
#             the scores (elements of eigenvectors)
#         x:  a nxp matrix of scores

   turtlef.pc <- prcomp(turtle.f)
```

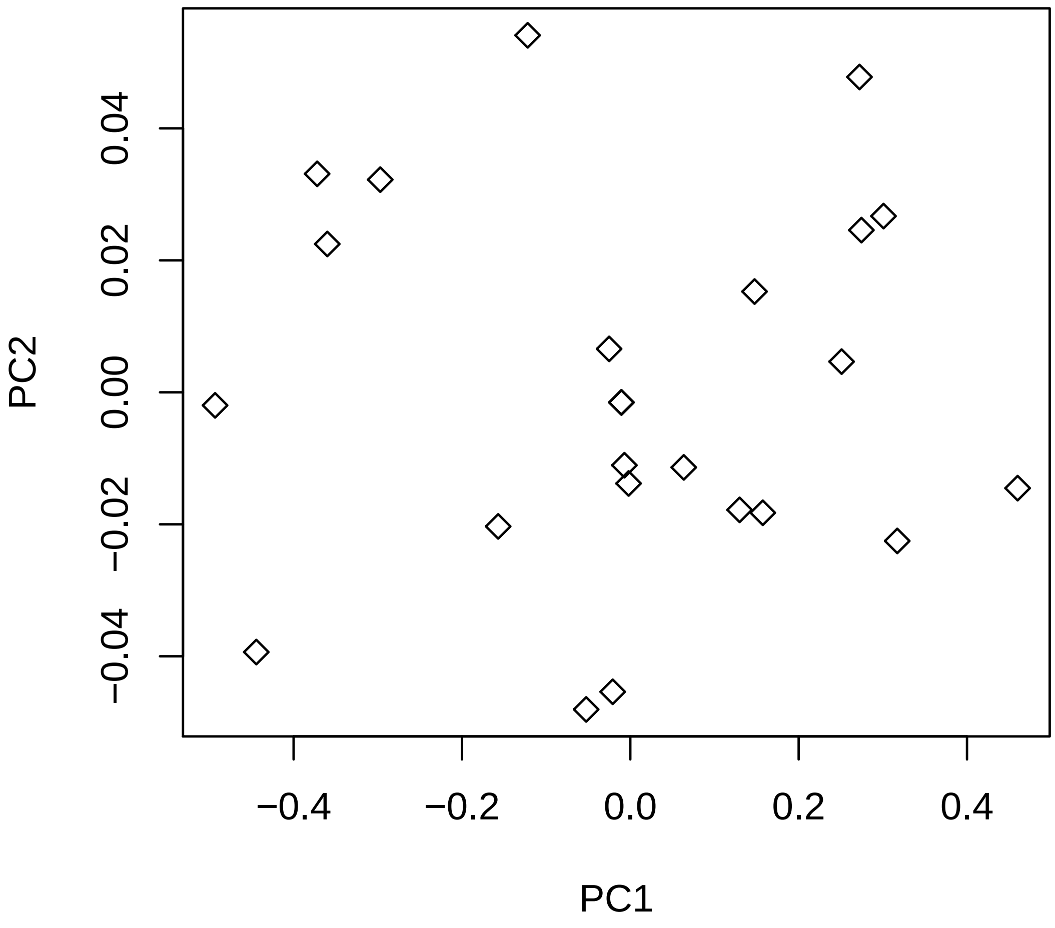

log(length)
log(width)
log(height)
4.4
4.5
4.6
4.7
4.8
4.9
5.0
4.8
4.6
4.9
4.8
4.7
4.6
4.5
4.4
4.1
3.9
3.7
4.6
4.8
5.0
3.7
3.9
4.1

```
turtlef.pc

Standard deviations:
[1] 0.25734377 0.02767404 0.02332817

Rotation:
           PC1        PC2         PC3
[1,] 0.6266648 -0.5525704 -0.54950625
[2,] 0.4878158 -0.2717450  0.82957243
[3,] 0.6077228  0.7879217 -0.09925955
```

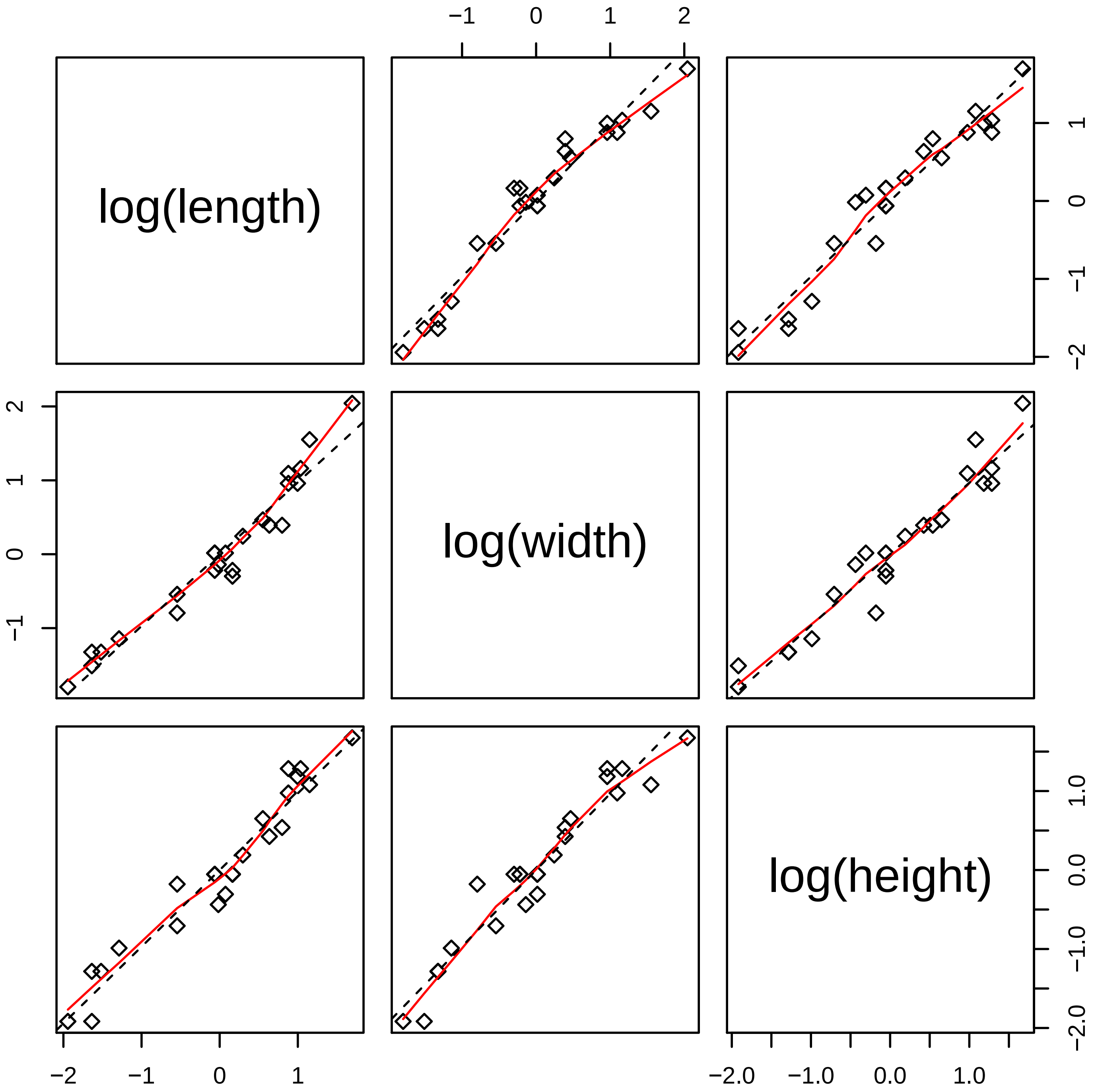
```
#  Compute the proportion of total variance explained
#  by each component

   s <- var(turtlef.pc$x)
   pvar<-round(diag(s)/sum(diag(s)), digits=6)
   cat("proportion of variance: ", pvar, fill=T)

   proportion of variance:  0.980602 0.01134 0.008058

#  Compute the cumulative proportion of total variance
#  explained by each component

   cpvar <- round(cumsum(diag(s))/sum(diag(s)),  digits=6)
   cat("cumulative proportion of variance: ",  cpvar, fill=T)

   cumulative proportion of variance:  0.980602 0.991942 1
```

```
#  Print some component scores
   head(turtlef.pc$x)

          PC1           PC2          PC3
1 -0.07985909 -0.0009417157  0.001818249
2 -0.07374361 -0.0075025878  0.001290866
3 -0.05695205  0.0048932712  0.004402337
4 -0.05554911  0.0034414942  0.002614878
5 -0.04449958  0.0052953012  0.001952550
6 -0.01506775  0.0100318949 -0.003411586

# Compute correlations between component scores
# and the variables
   cor(turtle.f, turtlef.pc$x)

                  PC1        PC2          PC3
log(length) 0.9885057 -0.1135573 -0.099806303
log(width)  0.9834465 -0.1233798  0.132704477
log(height) 0.9947131  0.1026662  0.002346062
```

```
#  Plot component scores

   par(pch=5, fin=c(5,5))
   plot(turtlef.pc$x[,1],turtlef.pc$x[,2], xlab="PC1",ylab="PC2")
```



```
#  To compute principal components from the sample correlation
#  matrix, you must first standardize the data

   turtle.fs <- scale(turtle.f, center=T, scale=T)

#  Plot standardized variables

  pairs(turtle.fs,labels=c("log(length)",
      "log(width)","log(height)"), panel=function(x,y){
          panel.smooth(x,y)
            abline(lsfit(x,y),lty=2) })

#  Compute principal components for the correlation matrix

   turtlef.cor <- var(turtle.fs)
   turtlef.cor
```


log(length)
log(width)
log(height)
−1
0
1
2
1
0
−1
−2
2
1
0
−1
1.0
0.0
−1.0
−2.0
−2
−1
0
1
−2.0
−1.0
0.0
1.0

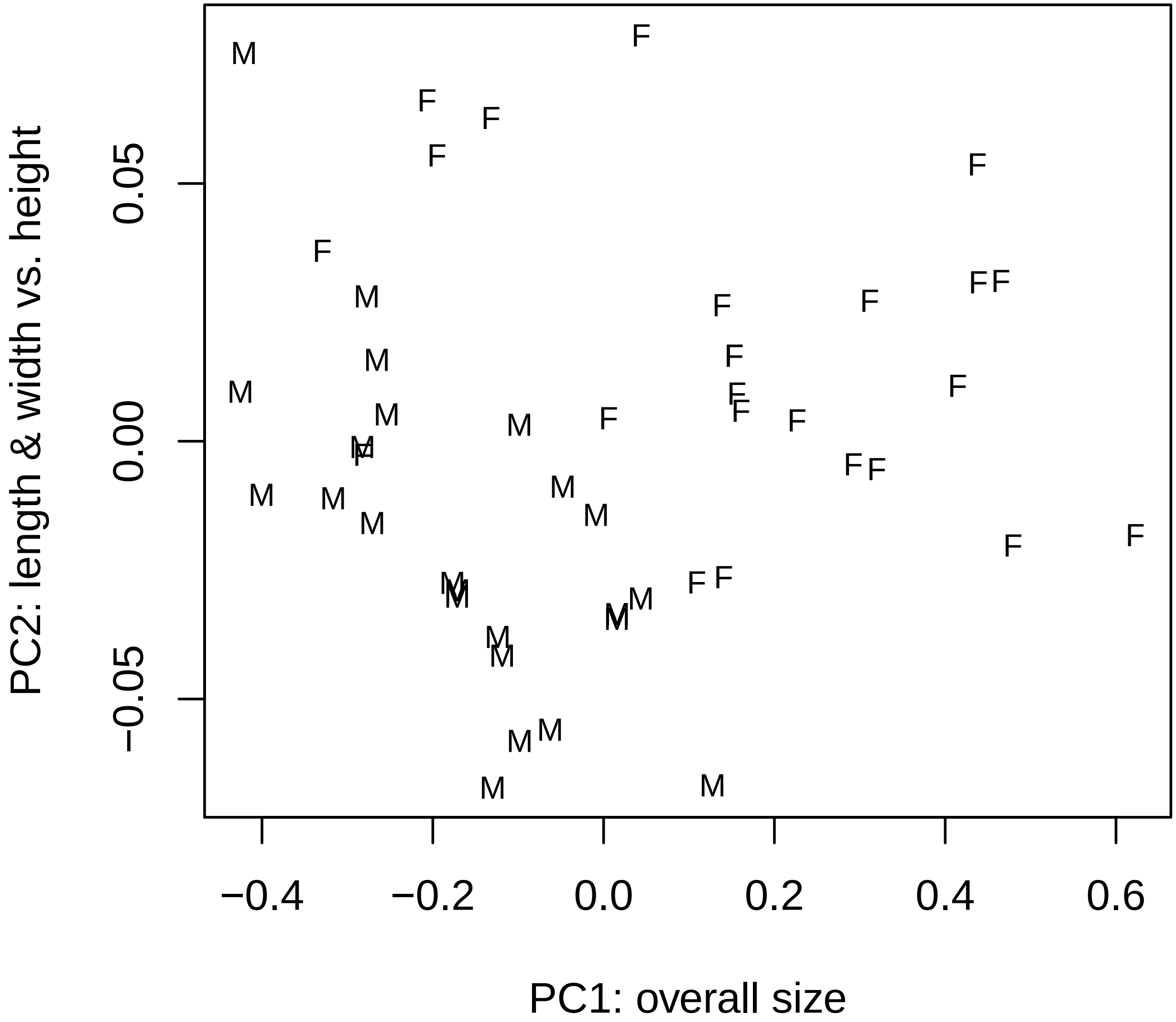
```
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.9726948 0.9709821
[2,] 0.9726948 1.0000000 0.9666505
[3,] 0.9709821 0.9666505 1.0000000

    turtlefs.pc <- prcomp(turtle.fs)
    turtlefs.pc$sdev

[1] 1.7147071 0.1829964 0.1621475

     turtlefs.pc$rotation
           PC1        PC2        PC3
[1,] 0.5780354 -0.1130261 -0.8081461
[2,] 0.5771777 -0.6434538  0.5028252
[3,] 0.5768371  0.7570946  0.3067029
```

```
s <- var(turtlefs.pc$x)
pvar<-round(diag(s)/sum(diag(s)), digits=6)
cat("proportion of variance: ", pvar, fill=T)

proportion of variance:  0.980073 0.011163 0.008764

cpvar <- round(cumsum(diag(s))/sum(diag(s)), digits=6)
cat("cumulative proportion of variance: ", cpvar, fill=T)

cumulative proportion of variance:  0.980073 0.991236 1
```

```
#  Principal components are sometimes useful for showing differences
#  between groups.  We will  illustrate this by displaying component
#  scores computed from the file containing 24 female turtles
#  (coded 1)  and 24  male turtles (coded 2).

#  First establish plotting symbols (M=male  F=female)

   nall <- dim(turtle.all)[1]
   turtle.type <-rep("F",nall)
   turtle.type[turtle.all[ ,1]>=2] <- "M"

#  Compute logs of the measurements

   turtle.a <- log(turtle.all[ , -1])
```

```
#   Compute principal components

    turtlea.pc <- prcomp(turtle.a)

   turtlea.pc$sdev

  [1] 0.26600038 0.03580422 0.02164380

    turtlea.pc$rotation
            PC1        PC2         PC3
 [1,] 0.6018462 -0.5549734 -0.57426963
 [2,] 0.4756146 -0.3285717  0.81598495
 [3,] 0.6415387  0.7642285 -0.06620384
```

```
#  Plot component scores

    par(fin=c(5,5))
    plot(turtlea.pc$x[,1],turtlea.pc$x[,2],
         xlab="PC1: overall size",
         ylab="PC2: length & width vs. height",type="n")
    text(turtlea.pc$x[,1],turtlea.pc$x[,2],
                  labels=turtle.type, cex=0.75)
```


PC2: length & width vs. height
PC1: overall size
0.05
0.00
−0.05
−0.4
−0.2
0.0
0.2
0.4
0.6
M
F

Five Socioeconomic Variables

- Data on socioeconomic variables for $n = 14$ census tracks in Madison, Wisconsin:
  - X_1: population (in thousands)
  - X_2: percentage with professional degrees
  - X_3: percentage employed (over age 16)
  - X_4: government employment (percent)
  - X_5: median home value (in hundreds of thousands of dollars)
- We extracted the PCs using both the covariance matrix S and the correlation matrix R (for illustration).

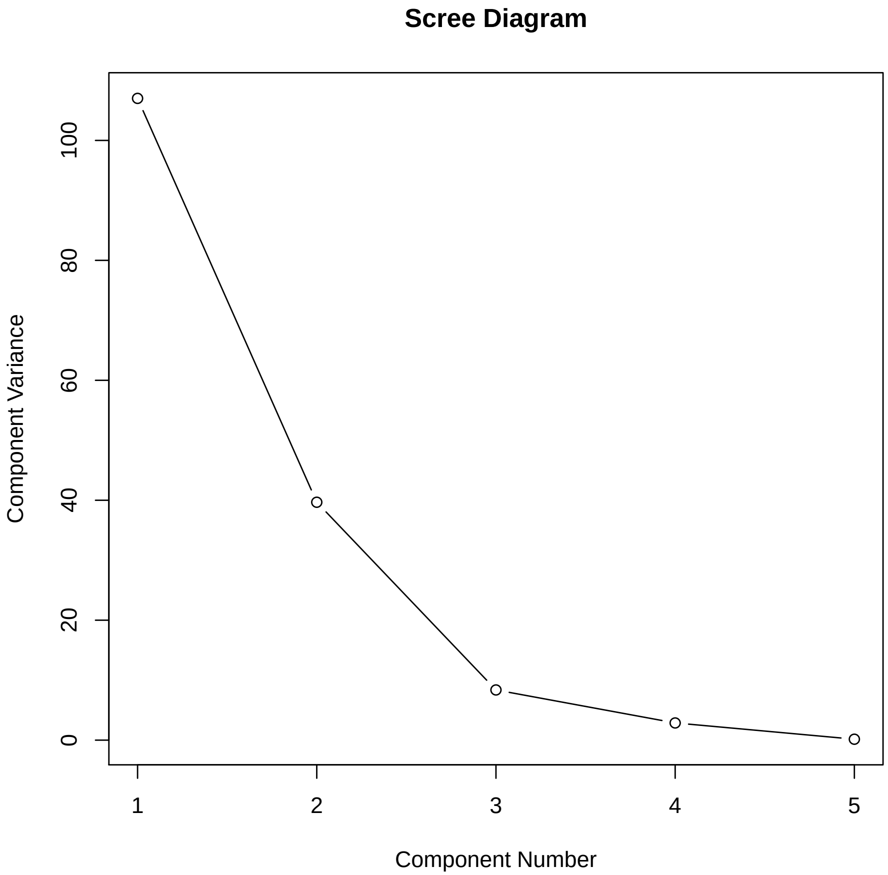
```
#  This code analyzes the Madison data.  It is posted as  madison.R
#  Enter the samples covariance matrix for the five variables

      madison <- matrix( c(3.397, -1.102, 4.306, -2.078, 0.027,
                    -1.102,  9.673,  -1.513,  10.953,  1.203,
                     4.306, -1.513,  55.626, -28.937, -0.044,
                    -2.078, 10.953, -28.937,  89.067,  0.957,
                     0.027,  1.203,  -0.044,   0.957,  0.319 ),
                     ncol=5, byrow="T")

#  Compute principal components from the sample covariance matrix.

     madison.pc <- princomp(covmat=madison)
     summary(madison.pc)
```

```
Importance of components:
                          Comp.1   Comp.2   Comp.3   Comp.4   Comp.5
Standard deviation     10.34481 6.29863 2.893290 1.693488 0.3938254
Proportion of Variance  0.67695 0.25096 0.052954 0.018141 0.0009811
Cumulative Proportion   0.67695 0.92792 0.980877 0.999018 1.0000000
```

```
   print(madison.pc$loadings, cutoff=0.0)
```

```
Loadings:
     Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
[1,]  0.039 -0.071 -0.188  0.977 -0.058
[2,] -0.105 -0.130  0.961  0.171 -0.139
[3,]  0.492 -0.864 -0.046 -0.091  0.005
[4,] -0.863 -0.480 -0.153 -0.030  0.007
[5,] -0.009 -0.015  0.125  0.082  0.989
```

```
# Construct a scree plot

   xpos <- 1:nrow(madison)
   plot(xpos, madison.pc$sdev^2, xlab="Component Number",
    ylab="Component Variance", type="b",  main = "Scree Diagram")
```



Five Socioeconomic Variables

- How many PCs to keep? When using S, we note that we can explain about 93% of the variability with the first two PCs.

- Thus, reducing the dataset from five variables to two PCs appears reasonable.

- The number of PCs retained will depend on the relative sizes of the eigenvalues of the covariance, or correlation, matrix, which depend on relative sizes of variances of the original traits and correlation patterns.

- Scree plots are sometimes useful.

- Interpretation is important.

Interpretation of Principal Components

Interpretation is important. In this example, when using S, the first two components focus on variation in X_3 and X_4 because those variables have much larger variances than the other variables.

1. First PC is a contrast between the percentage of the population employed in government jobs (X_4) and the percentage of adults who are employed (X_3). Component scores are large for tracts with relatively high government employment and relatively low adult employment rate.

2. Second PC is weighted sum of variables 3 and 4, with the larger weight on the adult employment percentage. This component has large scores for tracts with relatively high adult employment rates (X_3) and relative high percentages of government employment (X_4)

Interpretation of PCs

- Typically, we will look at both the size and the sign of the coefficients (the $\widehat{e}_{ik}$) and the contributions of each variable (the $r_{\widehat{y}_i,x_k}$) in order to interpret the meaning of the PC.
- The PC scores constitute a 'new' data set.
- We can explore PC scores just like we would explore directly observed variables before moving on to further analysis.

```
 # Compute correlations between component scores and variables

    corrvpc <- diag(1/sqrt(diag(madison)))%*%madison.pc$loadings
                                       %*% diag(madison.pc$sdev)
    corrvpc

           [,1]       [,2]        [,3]         [,4]          [,5]
[1,]  0.2182617 -0.2431805 -0.29493226  0.897824209 -0.0123169944
[2,] -0.3503079 -0.2627714  0.89399293  0.093297761 -0.0175439522
[3,]  0.6829143 -0.7299897 -0.01776570 -0.020674932  0.0002626469
[4,] -0.9460442 -0.3205722 -0.04696102 -0.005327544  0.0002795145
[5,] -0.1670758 -0.1642604  0.64027886  0.244800854  0.6893618236
```

Principal Components for a Correlation Matrix

- Now consider principal components computed from R. The data are the standardized observations $Z_{\mathrm{m}j}$

- The variances of the standardized observations are the same for all of the p attributes measured on each subject

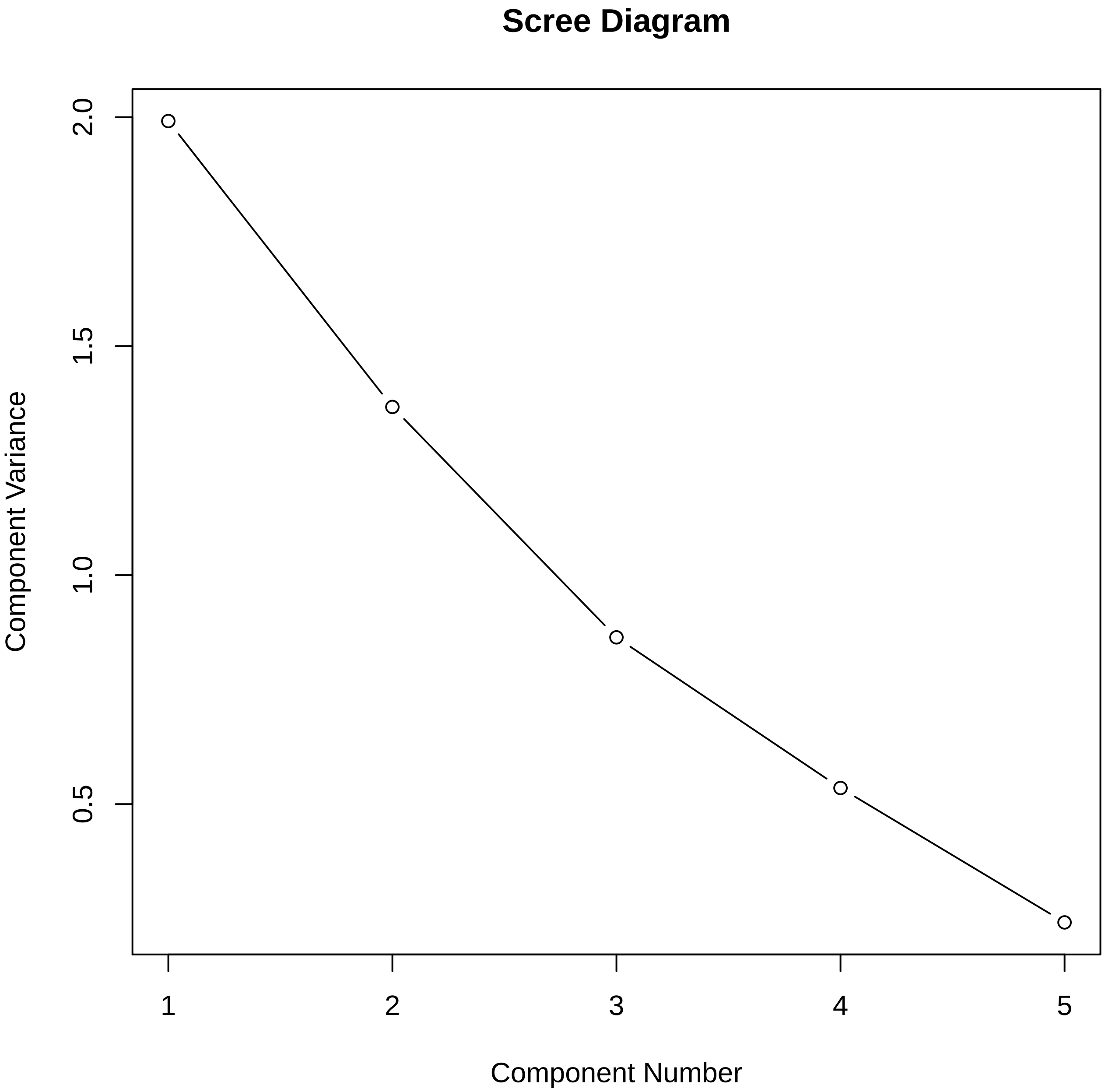
- More attention is paid to correlation patterns

- Note that the covariance matrix for the standardized variables is the correlation matrix, so we can simply analyze the sample correlation matrix.

```
 # Compute principal components from the correlation matrix

    madison.pc2 <- princomp(covmat=madison, cor="T")
    summary(madison.pc2)

Importance of components:
                         Comp.1   Comp.2   Comp.3   Comp.4    Comp.5
Standard deviation     1.411255 1.169311 0.929612 0.731571 0.4916219
Proportion of Variance 0.398328 0.273457 0.172835 0.107039 0.0483384
Cumulative Proportion  0.398328 0.671786 0.844622 0.951661 1.0000000

    print(madison.pc2$loadings, cutoff=0.0)
```

```
Loadings:
     Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
[1,]  0.263  0.463  0.784  0.217  0.235
[2,] -0.593  0.326 -0.164 -0.145  0.703
[3,]  0.326  0.605 -0.225 -0.663 -0.195
[4,] -0.479 -0.252  0.551 -0.571 -0.277
[5,] -0.493  0.500 -0.069  0.408 -0.580
```

```
    plot(xpos, madison.pc2$sdev^2, xlab="Component Number",
           ylab="Component Variance", type="b",
           main = "Scree Diagram")
```


Scree Diagram
Component Variance
2.0
1.5
1.0
0.5
1
2
3
4
5
Component Number

```
 # Compute correlations between component scores
 # and variables

    corrvpc2 <- madison.pc2$loadings %*% diag(madison.pc2$sdev)
    corrvpc2
           [,1]       [,2]        [,3]       [,4]        [,5]
[1,]  0.3708866  0.5411703  0.72873045  0.1587591  0.11543539
[2,] -0.8372003  0.3810940 -0.15230738 -0.1062467  0.34551009
[3,]  0.4599652  0.7074977 -0.20884665 -0.4848610 -0.09577054
[4,] -0.6763850 -0.2950501  0.51205831 -0.4179413 -0.13627212
[5,] -0.6957855  0.5844511 -0.06413275  0.2982262 -0.28503955
```

Test for equal eigenvalues for Σ

- Test the null hypothesis $H_0 : \lambda_{q+1} = \lambda_{q+2} = \cdots = \lambda_{q+r}$ that the r smallest population eigenvalues are equal. (p=q+r)

- This may correspond to a situation in which the first q principal components account for essentially all of the correlations among the measured attributes and most of the variances. What they do not capture is small random variation with essentially no correlation pattern.

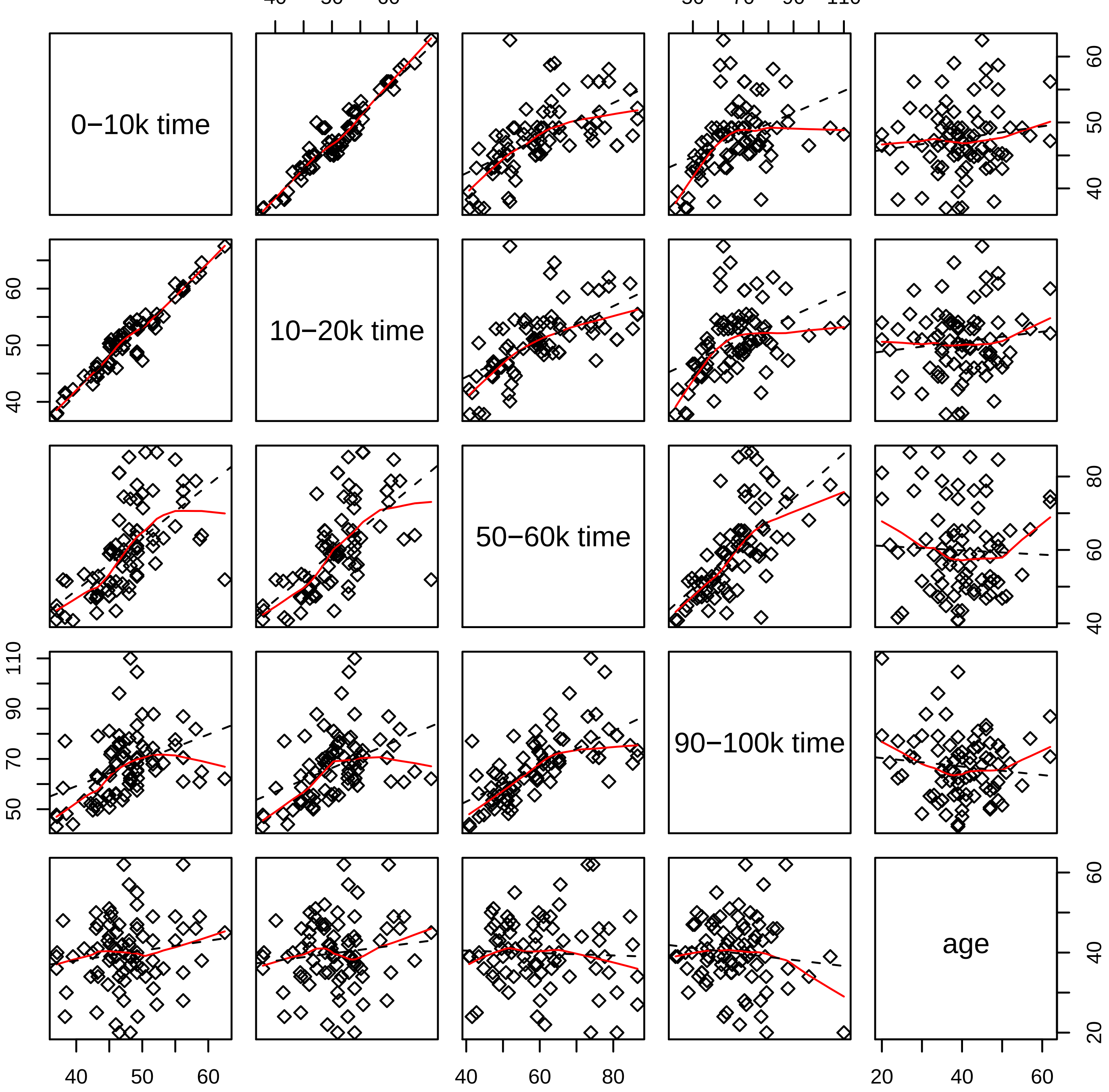
- Large sample chi-square test rejects the null hypothesis if

$$X^2 = (v)(r)ln\left[\frac{1}{r}\sum_{i=q+1}^{q+r}\hat{\lambda}_i\right] - v\sum_{i=q+1}^{q+r} ln(\hat{\lambda}_i) > \chi^2_{r(r+1)/2-1}$$

where v=(df for S)-(2p+5)/6

```
#  This code creates scatter plot matrices and
#  principal components for the 100k road race
#  data (Everitt 1994). This code is posted as
#  race100k.R.  The data are posted as race100k.dat
#
#  There is one line of data for each of 80
#  racers with eleven numbers on each line.
#  The first ten columns give the times (minutes)
#  to complete successive 10k segments of the race.
#  The last column has the racer’s age (in years).

   race.mat <- matrix(scan("race100k.dat"),
               ncol=11,byrow=T)
```

```
#  First compute the number of columns in the matrix

       p1<-dim(race.mat)[2]

#  Compute sample size and the number of section times

    n<-dim(race.mat)[1]
    p<-p1-1

#  Use the pairs function to create a scatter plot matrix.
#  Note that the columns to be included in the plot are
#  put into the "choose" list. The panel.smooth function
#  uses locally weighted regression to pass a smooth curve
#  through each plot.  The abline function uses least squares
#  to fit a straight line to each plot.  This helps you to
#  see if most of the marginal association between two
#  variables on can be described by a straight line. Recall
#  that principal components are computed from variances
#  and covariances (or correlations), which can only account
#  for straight line relationships.

   par(pch=5,fin=c(5,5))
   choose<-c(1,2,6,10,11)

   pairs(race.mat[ ,choose],labels=c("0-10k time",
      "10-20k time","50-60k time", "90-100k time","age"),
       panel=function(x,y){panel.smooth(x,y)
       abline(lsfit(x,y),lty=2) })
```


0−10k time
10−20k time
50−60k time
90−100k time
age
40
50
60
50
70
90
110
60
50
40
80
60
40
60
40
20
60
50
40
110
90
70
50
40
50
60
40
60
80
20
40
60

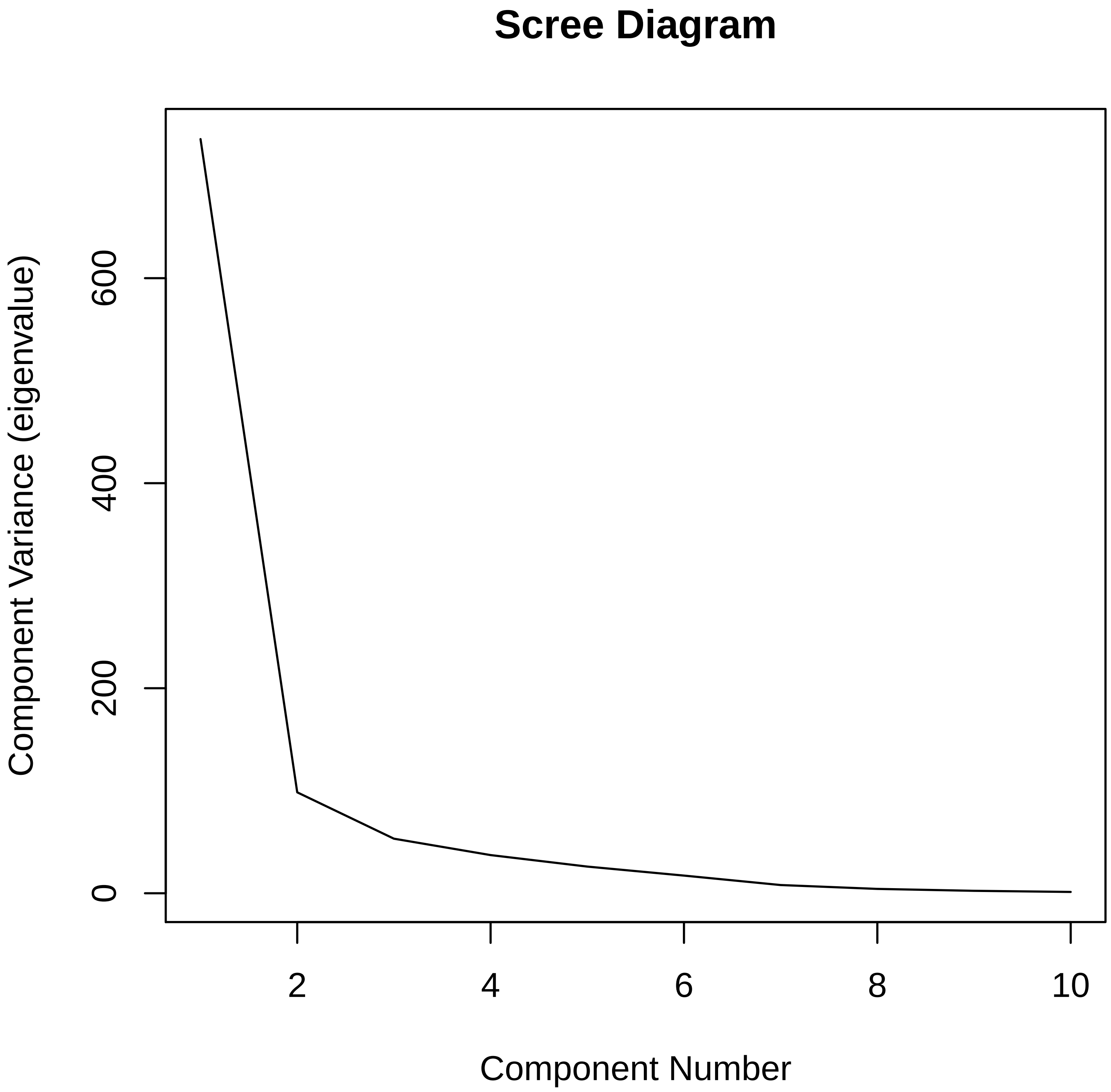
```
#  Compute principal components from the covariance matrix.
#  This function creates a list with the following components
#      sdev:  standard deviations of the component scores (
#             square roots of eigenvalues of the covariance
#             matrix)
#  rotation:  The coefficients needed to compute the scores
#             (elements of eigenvectors)
#         x:  a nxp matrix of scores


   race.pc <- prcomp(race.mat[ ,-p1])

#  Print the results

   race.pc$sdev
 [1] 27.123463  9.923923  7.297834  6.102917  5.102212
 [6]  4.151834  2.834300  2.060942  1.547235  1.135819
```

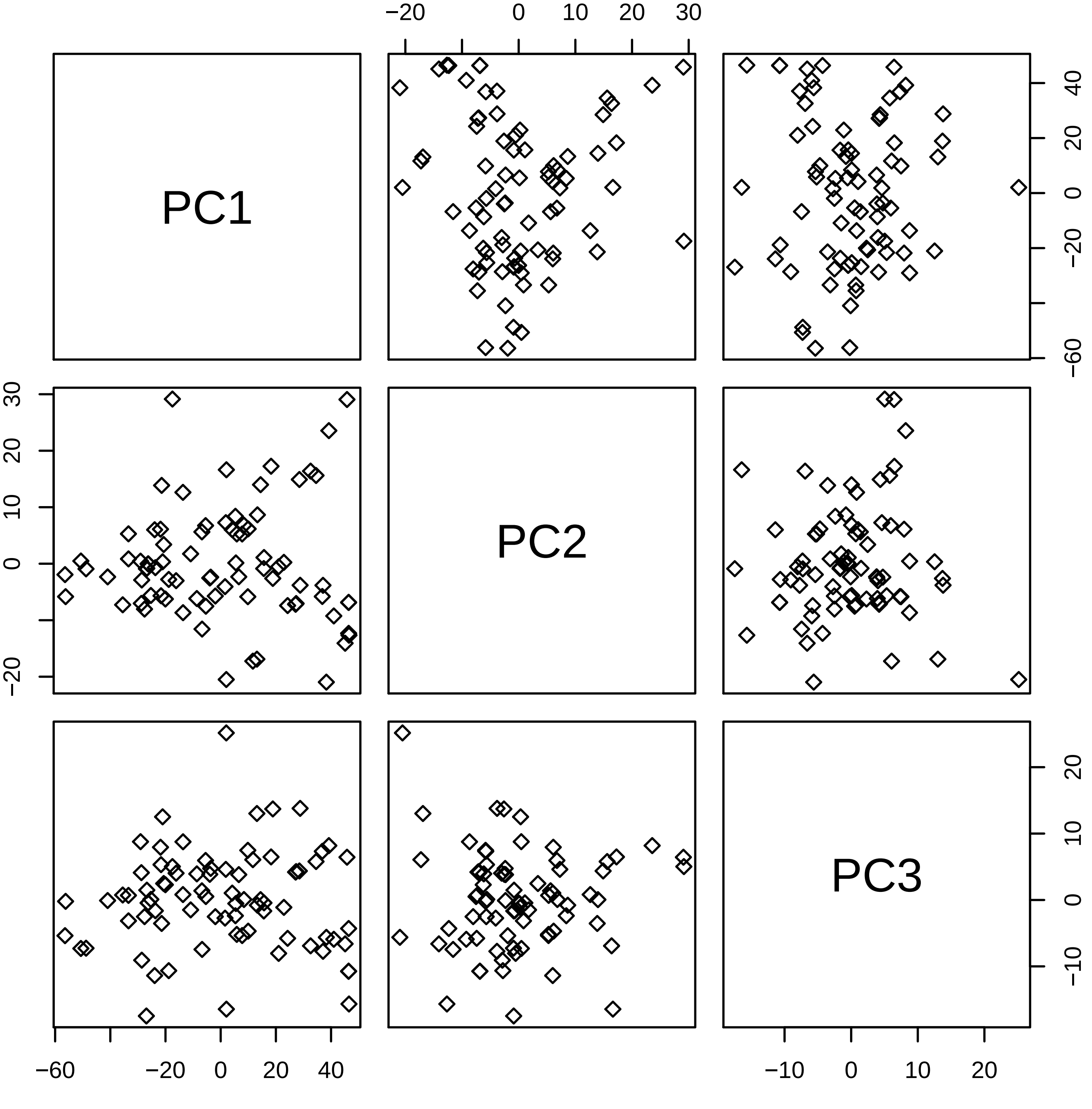
```
   race.pc$rotation
        PC1      PC2      PC3      PC4      PC5      PC6
 [1,] 0.1288  -0.2106   0.3615  -0.0335   0.1473  -0.2058
 [2,] 0.1520  -0.2491   0.4168  -0.0708   0.2238  -0.1309
 [3,] 0.1992  -0.3143   0.3411  -0.0539   0.2470   0.0526
 [4,] 0.2397  -0.3300   0.2027  -0.0066   0.0047   0.1439
 [5,] 0.3144  -0.3021  -0.1351   0.1107  -0.3564   0.2846
 [6,] 0.4223  -0.2147  -0.2223  -0.0868  -0.3730   0.2916
 [7,] 0.3359   0.0496  -0.1936  -0.6016  -0.1897  -0.6436
 [8,] 0.4067   0.0086  -0.5380   0.1290   0.7198   0.0348
 [9,] 0.3990   0.2675   0.1492   0.7175  -0.2098  -0.4142
[10,] 0.3854   0.6888   0.3482  -0.2789   0.0545   0.4051
```

```
          PC7     PC8     PC9    PC10
 [1,]  0.4324 -0.2802  0.0389  0.6900
 [2,]  0.3256 -0.2294  0.0463 -0.7128
 [3,] -0.3435  0.4576 -0.5868  0.0829
 [4,] -0.4479  0.1045  0.7451  0.0708
 [5,] -0.2450 -0.6462 -0.3061 -0.0054
 [6,]  0.5390  0.4494  0.0379 -0.0229
 [7,] -0.1844 -0.0219 -0.0193 -0.0190
 [8,]  0.0293 -0.0817  0.0366  0.0180
 [9,] -0.0461  0.1132 -0.0027 -0.0398
[10,] -0.0300 -0.0945 -0.0025  0.0320
```

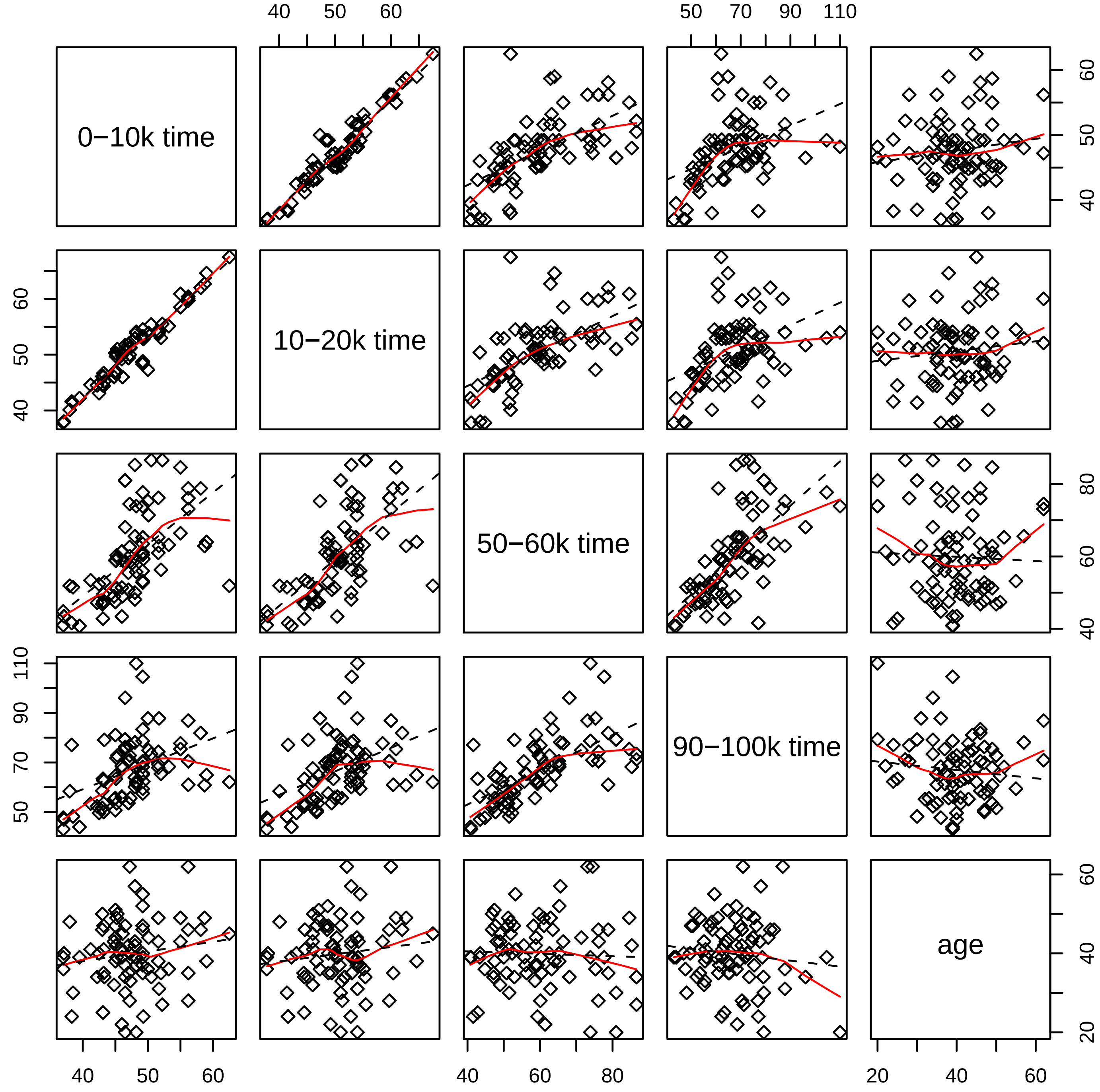
```
#  compute proportion of total variance explained by
#  each component

     summary(race.pc)

Importance of components:
                           PC1    PC2     PC3     PC4     PC5
Standard deviation     27.1235 9.9239 7.29783 6.10292 5.10221
Proportion of Variance  0.7477 0.1001 0.05413 0.03785 0.02646
Cumulative Proportion   0.7477 0.8478 0.90194 0.93980 0.96625
                            PC6     PC7     PC8     PC9    PC10
Standard deviation      4.15183 2.83430 2.06094 1.54723 1.13582
Proportion of Variance  0.01752 0.00816 0.00432 0.00243 0.00131
Cumulative Proportion   0.98377 0.99194 0.99626 0.99869 1.00000
```

```
# produce a scree plot

  plot(race.pc$sdev^2, xlab="Component Number",
        ylab="Component Variance (eigenvalue)",
        main="Scree Diagram", type="l")


 #  plot component scores

    par(pch=5, fin=c(5,5))
    pairs(race.pc$x[,c(1,2,3)],labels=c("PC1","PC2","PC3"))
```


Scree Diagram
Component Variance (eigenvalue)
0
200
400
600
2
4
6
8
10
Component Number


PC1
PC2
PC3
−20
0
10
20
30
40
20
0
−20
−60
30
20
10
0
−20
20
10
0
−10
−60
−20
0
20
40
−10
0
10
20

```
#  To compute principal components from a correlation matrix,
#  you must first standardize the data

#    race.s <- scale(race.mat, center=T, scale=T)

#  Plot standardized data

      choose<-c(1,2,5,10,11)

    pairs(race.s[ ,choose],labels=c("0-10k time",
       "10-20k time","50-60k time", "90-100k time", "age"),
        panel=function(x,y){panel.smooth(x,y)
           abline(lsfit(x,y),lty=2) })
```


0−10k time
10−20k time
50−60k time
90−100k time
age
40
50
60
50
70
90
110
60
50
40
60
50
40
80
60
40
110
90
70
50
60
40
20
40
50
60
40
60
80
20
40
60

```
#  Compute principal components from the correlation matrix

    race.cor <- var(race.s)
    cat("correlation matrix for 10k splits:", fill=T)
```

```
          [,1]     [,2]      [,3]      [,4]       [,5]       [,6]
 [1,] 1.000000 0.951060 0.8445873 0.7858559  0.6205345  0.6178917
 [2,] 0.951060 1.000000 0.8903106 0.8261249  0.6414426  0.6327654
 [3,] 0.844587 0.890310 1.0000000 0.9210859  0.7559463  0.7250990
 [4,] 0.785856 0.826124 0.9210859 1.0000000  0.8869090  0.8418564
 [5,] 0.620534 0.641442 0.7559463 0.8869090  1.0000000  0.9364148
 [6,] 0.617891 0.632765 0.7250990 0.8418564  0.9364148  1.0000000
 [7,] 0.531396 0.540931 0.6050262 0.6906541  0.7541974  0.8395763
 [8,] 0.477372 0.505452 0.6199820 0.6982151  0.7857814  0.8403225
 [9,] 0.542343 0.533807 0.5835764 0.6673532  0.7413497  0.7725735
[10,] 0.414260 0.438128 0.4672533 0.5085771  0.5417422  0.6559189
[11,] 0.149172 0.127104 0.0121828 0.0468020 -0.0160752 -0.0424197
```

```
             [,7]       [,8]       [,9]      [,10]       [,11]
 [1,]  0.53139648  0.4773723  0.5423438  0.4142609  0.14917250
 [2,]  0.54093190  0.5054520  0.5338073  0.4381283  0.12710409
 [3,]  0.60502621  0.6199821  0.5835765  0.4672533  0.01218286
 [4,]  0.69065419  0.6982152  0.6673533  0.5085772  0.04680206
 [5,]  0.75419742  0.7857815  0.7413497  0.5417422 -0.01607529
 [6,]  0.83957633  0.8403225  0.7725735  0.6559189 -0.04241971
 [7,]  1.00000000  0.7796014  0.6972448  0.7191956 -0.04059097
 [8,]  0.77960144  1.0000000  0.7637562  0.6634709 -0.20674428
 [9,]  0.69724482  0.7637562  1.0000000  0.7797619 -0.12320048
[10,]  0.71919560  0.6634709  0.7797619  1.0000000 -0.11289354
[11,] -0.04059097 -0.2067443 -0.1232005 -0.1128935  1.00000000
```

```
races.pc <- prcomp(race.s[ , -11])

  cat("standard deviations of component scores:", fill=T)
    standard deviations of component scores:

    races.pc$sdev

[1] 2.6912189  1.1331038  0.7439637  0.5451001  0.4536530
[6] 0.4279130  0.3300239  0.2204875  0.1984028  0.1923427
```

```
cat("component coefficients", fill=T)
     component coefficients
```

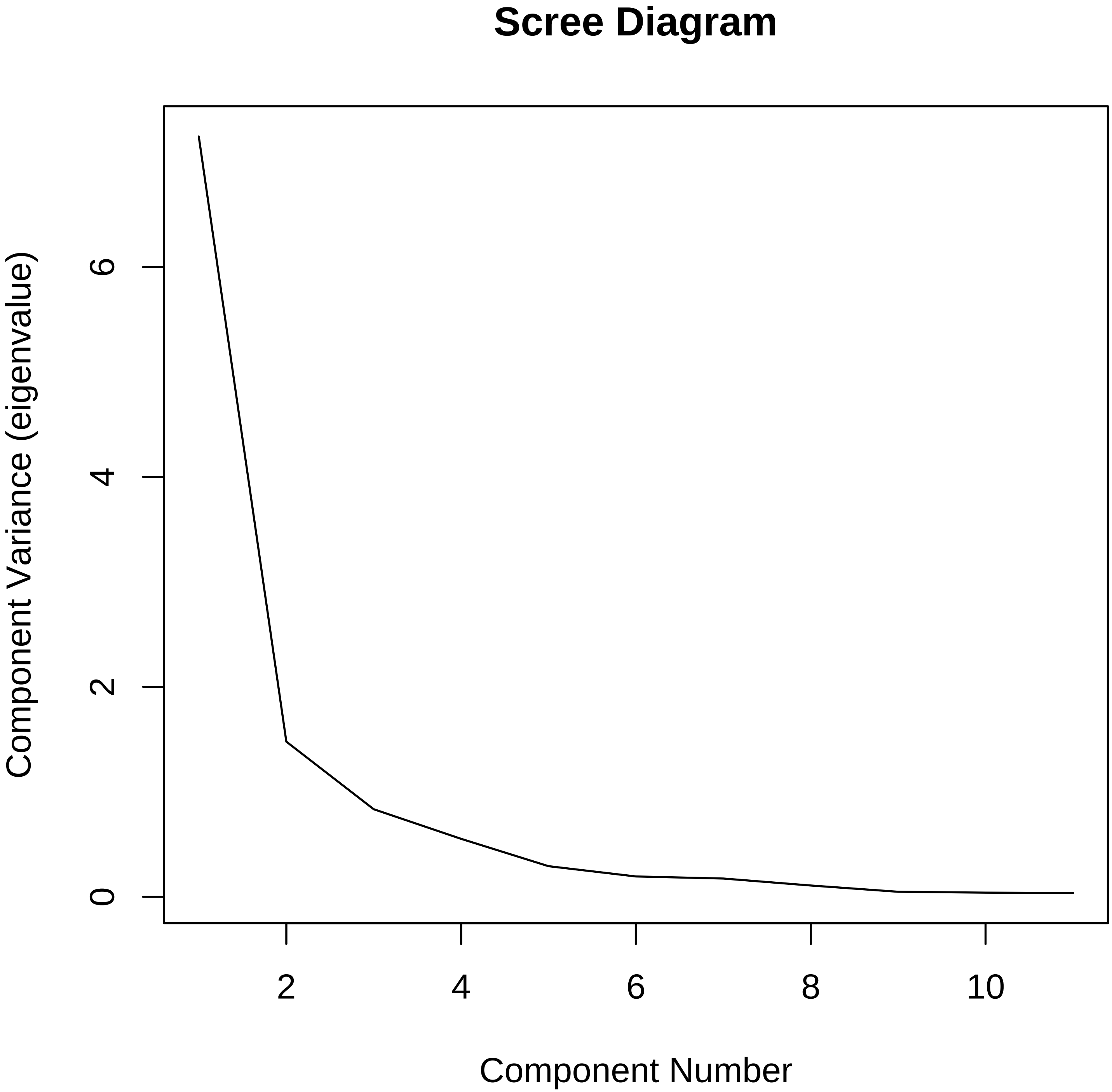
```
races.pc$rotation
```

```
           PC1         PC2         PC3         PC4         PC5         PC6
 [1,] 0.2965040 -0.44952127 -0.28625812  0.05744847 -0.08599593  0.44988879
 [2,] 0.3043491 -0.45106943 -0.25184368  0.07852721 -0.11800900  0.10202337
 [3,] 0.3254657 -0.34156406  0.02755111 -0.02047378 -0.08587480 -0.52159002
 [4,] 0.3444046 -0.20076954  0.22908847 -0.09719636  0.19840389 -0.33413665
 [5,] 0.3377513  0.06180954  0.44697744 -0.18724110  0.33529354  0.06478330
 [6,] 0.3453897  0.15930436  0.29325406  0.04696715  0.20163475  0.13513108
 [7,] 0.3126879  0.27106131  0.02897102  0.73462130  0.10965959  0.25099876
 [8,] 0.3122922  0.30183006  0.20434988 -0.03770352 -0.85328771 -0.06100872
 [9,] 0.3083362  0.28967055 -0.26728079 -0.62881001  0.06842474  0.35254484
[10,] 0.2667831  0.39976197 -0.63369272  0.08312435  0.18453881 -0.43591315
```

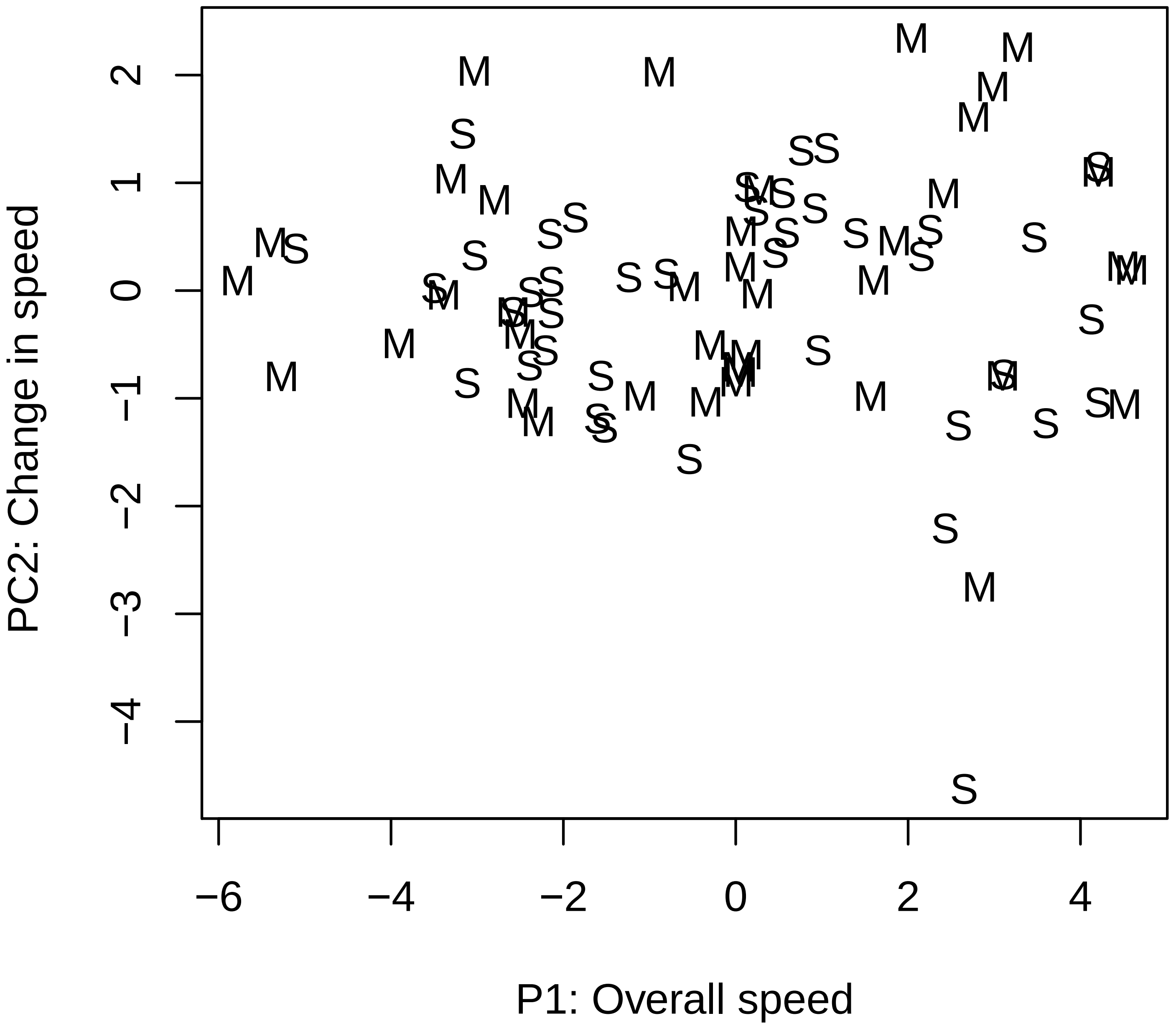
```
             PC7         PC8          PC9         PC10
 [1,] -0.1923123  0.01485076 -0.612826069 -0.047078428
 [2,] -0.1635240 -0.18379664  0.741251560  0.002483905
 [3,]  0.3603175  0.54150034 -0.057189353 -0.265471742
 [4,]  0.1820329 -0.51310747 -0.186770540  0.544828894
 [5,] -0.2252704 -0.24525627  0.006760302 -0.647837829
 [6,] -0.4346583  0.55907502  0.104542699  0.442476362
 [7,]  0.4530905 -0.05705153  0.037490586 -0.057160164
 [8,] -0.1212751 -0.13448246 -0.058637963 -0.011755653
 [9,]  0.4491343  0.08420582  0.107575891  0.069455481
[10,] -0.3345390 -0.09137728 -0.097185763 -0.082188533
```

```
# Compute contributions to the total variance

 summary(races.pc)

Importance of components:
                          PC1    PC2     PC3     PC4     PC5
Standard deviation     2.6912 1.1331 0.74396 0.54510 0.45365
Proportion of Variance 0.7243 0.1284 0.05535 0.02971 0.02058
Cumulative Proportion  0.7243 0.8527 0.90801 0.93772 0.95830
                          PC6    PC7     PC8     PC9    PC10
Standard deviation     0.42791 0.33002 0.22049 0.19840 0.1923
Proportion of Variance 0.01831 0.01089 0.00486 0.00394 0.0037
Cumulative Proportion  0.97661 0.98750 0.99236 0.99630 1.0000
```

```
# Produce a scree plot

 plot(race.pc$sdev^2, xlab="Component Number",
    ylab="Component Variance (eigenvalue)",
    main="Scree Diagram", type="l")
```


Scree Diagram
Component Variance (eigenvalue)
0
2
4
6
2
4
6
8
10
Component Number

```
#  Use the principal component scores from the raw data
#  to look for differences among mature (age < 40) and
#  senior (age > 40) runners.  Mature runners will be
#  indicated by "M" and senior runners will be indicated
#  by "S".

   race.type <-rep("M",n)
   race.type[race.mat[ ,p1]>=40] <- "S"

#  Plot component scores

    par(fin=c(5,5))
    plot(races.pc$x[,1],races.pc$x[,2],
          xlab="P1: Overall Time",
          ylab="PC2: Change in Pace ",type="n")
   text(races.pc$x[,1],races.pc$x[,2],labels=race.type)
```


PC2: Change in speed
P1: Overall speed
2
1
0
−1
−2
−3
−4
−6
−4
−2
0
2
4
M
S