Primer of Biostatistics
The 3rd Lecture

Hiroyoshi Iwata
aiwata@mail.ecc.u-tokyo.ac.jp

<Principal component analysis>

Experiments in agriculture and life sciences often measure multiple characteristics of the same samples. For example, in field trials of crops, various traits related to yield are examined simultaneously, even for the purpose of yield assessment. It is often the case that you can discover some kind of knowledge by drawing and viewing scatter plots of the multiple characteristics measured simultaneously. However, if the number of characteristics measured is large, it will be difficult to grasp the variation of data with a scatter plot. The number of dimensions that humans can intuitively grasp using the scatter plot is at most several dimensions, and it is not easy to grasp the variation of data when there are more than 10 measured characteristics. Principal component analysis described in this lecture is a method for summarizing variation contained in multidimensional data into low-dimensional features (i.e., principal component scores) without reducing the amount of information as much as possible. For example, the variation summary included in the marker genotype data shown as an example in this lecture shows that data in 1,311 dimensions can be summarized in four dimensions. Principal component analysis is a very effective method to efficiently extract the information contained in variables when the number of variables is large.

In this lecture, we will use the rice dataset (Zhao et al. 2011, Nature Communications 2: 467) as an example. In this lecture, we will also use marker genotype data (RiceDiversityGeno.csv) as well as line (accession) data (RiceDiversityLine.csv) and phenotypic data (RiceDiversityPheno.csv). The the marker genotype data is the genotypes of 1,311 SNPs that Zhao et al. (2010, PLoS One 5: e10780) used in their analysis. All data are based on the data downloaded from the Rice Diversity web page http://www.ricediversity.org/. As for marker data, missing value was imputed by using the software fastPHASE (Scheet and Stephens 2006, Am J Hum Genet 78: 629).

Load three types of data and combine them.

```
> line <- read.csv("RiceDiversityLine.csv")
                  # accession (line) data
> pheno <- read.csv("RiceDiversityPheno.csv")
                  # phenotypic data
> geno <- read.csv("RiceDiversityGeno.csv")
                  # marker genotype data
> line.pheno <- merge(line, pheno, by.x = "NSFTV.ID", by.y = "NSFTVID")
                  # combine line and pheno
> alldata <- merge(line.pheno, geno, by.x = "NSFTV.ID", by.y = "NSFTVID")
                  # combine line.pheno and geno
```

Analyze the variation of panicle length and flag leaf length in varieties and lines included in rice germplasm by principal component analysis. First, the data of both traits are extracted from all data (alldata). Then, samples which have at least one missing value among the data of both traits are excluded.

```
> mydata <- data.frame(
         panicle.length  = alldata$Panicle.length,
         leaf.length = alldata$Flag.leaf.length
         )
         # Extract Panicle.length and Flag.leaf.length
> mydata
 (omitted)
> missing <- apply(is.na(mydata), 1, sum) > 0
         #  is.na is a function for check whether elements are NA or not
         #  sum of this values becomes the number of NAs in a sample
         #  missing is a vector which contains a vector of T or F
         #  if a sample has missing, the corresponding element becomes T
> mydata <- mydata[!missing, ]
         #  remove samples which have missing data
> mydata
  (omitted)
```

Let's check the variation of panicle length and flag leaf length with a scatter plot.

```
> plot(mydata)              #  draw the scatter plot
> lim <- range(mydata)      #  set range of values in both variables
> plot(mydata, xlim = lim, ylim = lim)
                            #  draw the plot with the same range for x and y
```

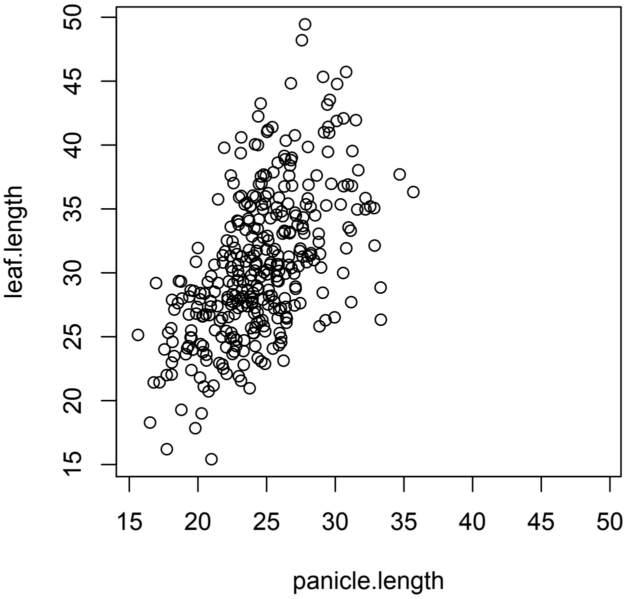

Figure 1. The relationship between panicle length and flag leaf length in 341 accessions of rice germplasm
Both lengths are measured in cm

Look at the scatter plot. When one trait becomes large, the other tends to also become large. Calculate the variance-covariance matrix and correlation matrix of both traits and confirm it numerically.

```
> cov(mydata)                          #  variance-covariance matrix
             panicle.length leaf.length
panicle.length      12.67168    11.57718
leaf.length         11.57718    33.41344
> cor(mydata)                          #  correlation matrix
             panicle.length leaf.length
panicle.length      1.000000    0.562633
leaf.length         0.562633    1.000000
```

Both the correlation and the covariance have positive values. Thus, we can confirm that both tend to vary together.

In order to simplify the following calculations and explanations, each trait is normalized to be zero on average (the average is subtracted from the original variable).

```
> mydata <- sweep(mydata, 2, apply(mydata, 2, mean))
         #  calculate column means and subtract them from the columns
> summary(mydata)
 (omitted)
> cov(mydata)
               panicle.length leaf.length
panicle.length       12.67168    11.57718
leaf.length          11.57718    33.41344
> cor(mydata)
               panicle.length leaf.length
panicle.length       1.000000    0.562633
leaf.length          0.562633    1.000000
> lim <- range(mydata)
> plot(mydata, xlim = lim, ylim = lim)
> abline(h = 0, v = 0)              #  draw the lines of x = 0 and y = 0
```

Note that scaling the variables does not change the relationship between the variables represented by the variance-covariance matrix and the correlation matrix.

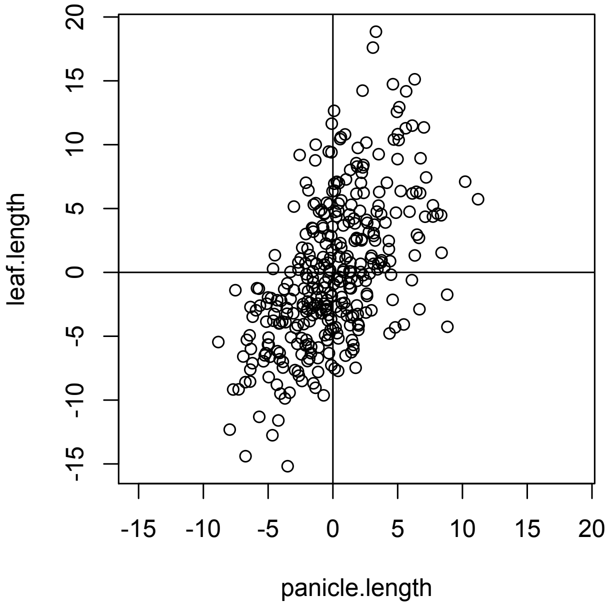

Figure 2. Panicle length and flag leaf length normalized so that their means

are equal to 0

Let's perform principal component analysis and plot the obtained principal component scores.

```
> res <- prcomp(mydata)    #  perform principal component analysis
> lim <- range(res$x)      #  extract principal component scores as res$x
> plot(res$x, xlim = lim, ylim = lim)       #  draw the plot of scores
> abline(h = 0, v = 0)
```

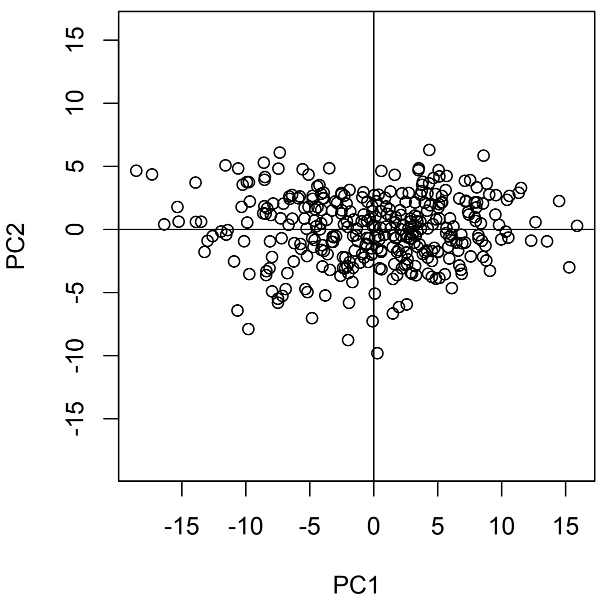

Figure 3. The horizontal axis is the first principal component (PC1) score, and the vertical axis is the second principal component (PC2) score

Let's check the relationship between principal component scores and original variables by drawing a scatter diagram side by side.

```
> op <- par(mfrow = c(1,2)) # par is a function setting graph parameters
          # mfrow = c(1,2) means draw graphs in the 1 row and two culums format
> lim <- range(mydata)
> plot(mydata, xlim = lim, ylim = lim)
> abline(h = 0, v = 0)
> lim <- range(res$x)
> plot(res$x, xlim = lim, ylim = lim)
> abline(h = 0, v = 0)
> par(op)         # reset the graph parameters
          # op is the parameters before we set them in the first line.
```

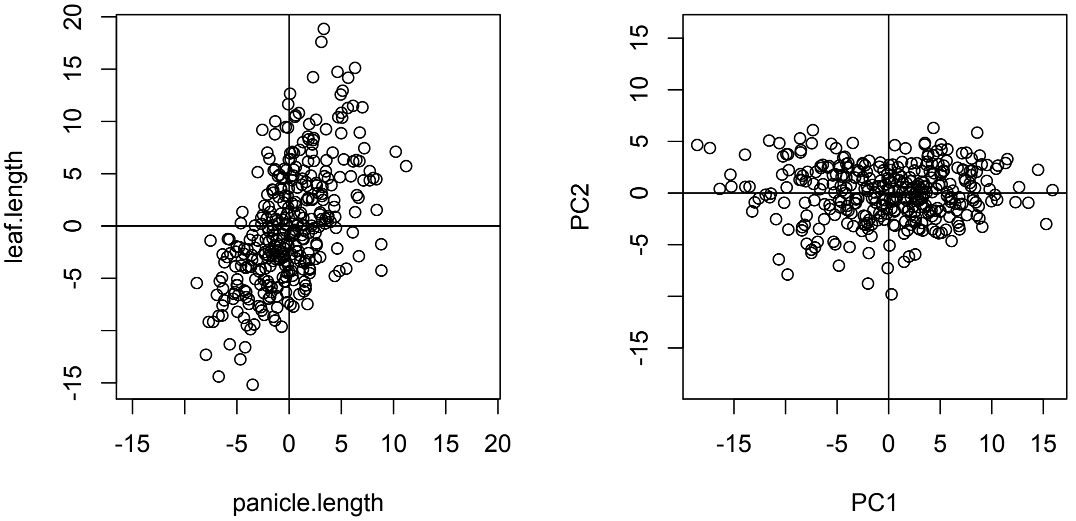

Figure 4. (left) original variable, (right) principal component score

The side-by-side comparison of the principal component scores and the original variables shows that the principal component scores have the rotation (and inversion) of the original variables. Principal component analysis is a method for representing the variation of the original variables with new variables with as few dimensions as possible. For example, the horizontal axis in the right figure expresses the variation of the two variables of panicle length and flag leaf length as the variation of one new variable (i.e,, the first principal component). It can be seen that the first principal component alone can explain most of the variation of the original variables. In addition, the scores of the second principal component represents the variation that could not be explained by the first principal component.

Now let's check how much each principal component actually explains the variations.

```
> summary(res)                    # summary of PCA
Importance of components:
                          PC1    PC2
Standard deviation     6.2117 2.7385
Proportion of Variance 0.8373 0.1627
Cumulative Proportion  0.8373 1.0000
```

It can be seen that the first principal component accounts for 83.7% of the total variation, and the second principal component accounts for the remaining 16.3%. That is, it can be seen that 80% or more of the variation of panicle length and flag leaf length can be represented by one variable (first main component).

Let's look at the results in more detail.

```
> res                    #  the object which has the result of PCA
Standard deviations:
[1] 6.211732 2.738524

Rotation:
                      PC1        PC2
panicle.length -0.4078995 -0.9130268
leaf.length    -0.9130268  0.4078995
```

Standard deviations represent the standard deviation of the first and second principal components which are new variables. In addition, Rotation represents a unit vector representing the orientation of the axes of the first and second principal components (note that these unit vectors are called eigenvectors, as will be described later).

In addition, the results mentioned above can also be retrieved separately as follows.

```
> res$sdev                          #  standard deviation of PC scores
[1] 6.211732 2.738524
> res$rotation                      #  eigenvectors
                   PC1        PC2
panicle.length -0.4078995 -0.9130268
leaf.length    -0.9130268  0.4078995
```

Let's draw graphs for the result of principal component analysis.

```
> op <- par(mfrow = c(1,2))
> plot(res)                #  variance (eigenvalues) of PCs
> biplot(res)
 #  biplot which shows relationship between PCs and original variables
```

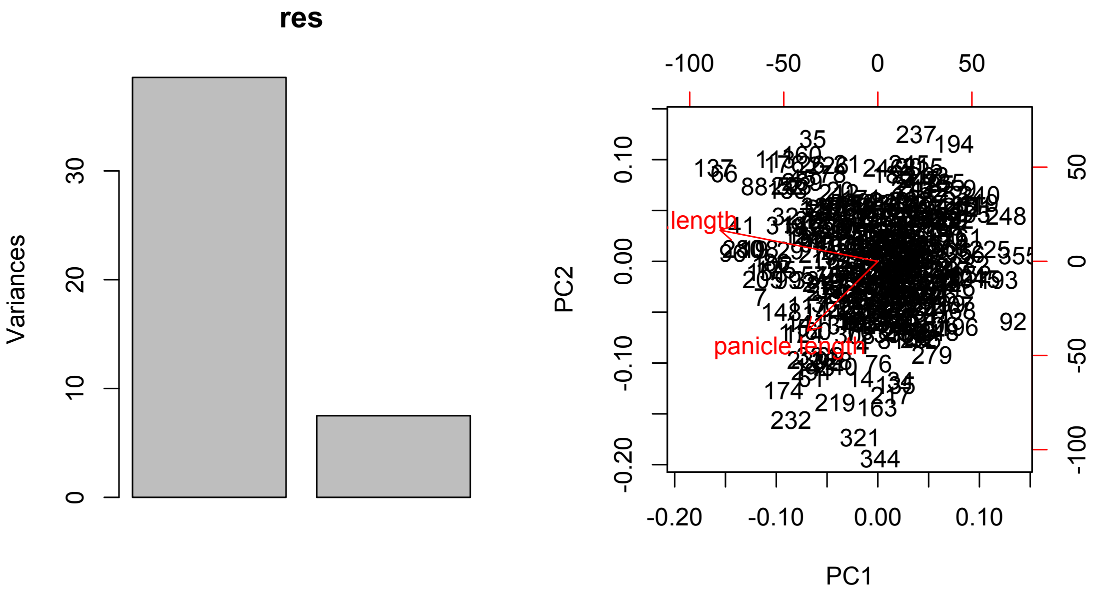

Fig. 5 (left) Principal component variance (eigenvalues)
(Right) Biplot representing the relationship between principal component scores and variables

The left graph showed the variance of the principal component scores, which is the square of the standard deviation of the principal component (note that the variance of the principal component score is the eigenvalue of the variance covariance matrix). The right graph is a biplot that shows the relationship between principal component scores and variables. Looking at the biplot, both the flag leaf length (leaf. Length) and the panicle length (panicle. Length) have arrows pointing to the left, and both traits have large values for the first principal component (the horizontal axis). Samples that have smaller scores for the component have larger values in both traits. That is, the first principal component can be interpreted as a variable that represents "size". On the other hand, the arrow of the flag leaf length is (slightly) upward, the arrow of the panicle length is downward. A sample with larger flag leaf length has a larger value, while a sample with large panicle length has a smaller value. That is, it can be interpreted that the second principal component is a variable that represents the "ratio" of the length of the flag leaf and the panicle length. The details about the biplot will be explained again later.

<Formulation of principal component analysis>
Here, I will outline the formulation of principal component analysis, using the two-dimensional data mentioned earlier as an example.

First, let us consider that the variation of the two variables of panicle length and flag leaf length is represented by one new variable. The orientation of the axis representing this new variable is tentatively assumed as $(1/\sqrt{2}, 1/\sqrt{2})$ here. The value of the new variable corresponds to the position of the foot of the perpendicular line from this data point to this axis. That is, the red line of the figure drawn below is the axis that represents the new variable, the gray line is the perpendicular line from the data point to the new axis, and the green + is the foot of the perpendicular line. The position of the foot of this perpendicular is the value of the new variable.

```
> lim <- range(mydata)
> plot(mydata, xlim = lim, ylim = lim)      # draw a scatter plot
> abline(h = 0, v = 0)                      # draw x and y axes
> u.temp <- c(1 / sqrt(2), 1 / sqrt(2))     # the new variable
> abline(0, u.temp[2] / u.temp[1], col = "red")  # draw new axis
> score.temp <- as.matrix(mydata) %*% u.temp   # calculate the inner product
> x <- score.temp * u.temp[1]               # x coordinate of the foot
> y <- score.temp * u.temp[2]               # y coordinate of the foot
> segments(x, y, mydata$panicle.length, mydata$leaf.length, col = "gray")
                                            # draw segments with gray
> points(x, y, pch = 4, col = "green")      # draw foots with green
```

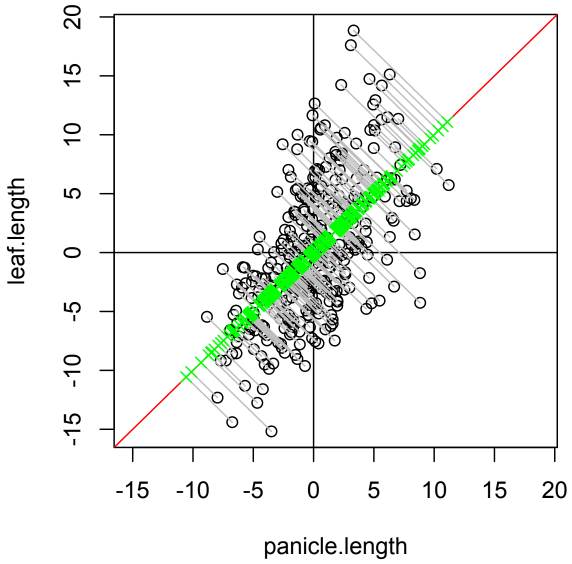

Figure 6. Variation of two variables represented by the temporarily set new variable axis

Now, let's focus on one sample and examine the relationship between the value of the original variable and the value of the new variable in more detail. Here, let's draw a figure to pay attention to one sample that has the longest flag leaf length.

```
> id <- which.max(mydata$leaf.length)
         #  find the order of the sample which has largest leaf length
> arrows(0, 0,                         #  starting points of arrows
         mydata$panicle.length[id], mydata$leaf.length[id],   #  end points
         col = "purple")               #  colors of arrows
> arrows(x[id], y[id],
         mydata$panicle.length[id], mydata$leaf.length[id],
         col = "pink")
> arrows(0, 0, x[id], y[id], col = "blue")
```

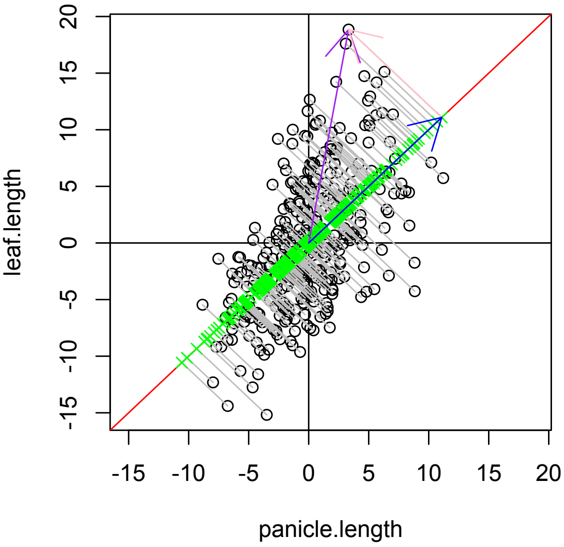

Figure 7. Original variable (vector of purple arrows) and new variable (vector of blue arrows)

Now, if the original variable is represented by a new variable as shown in Figure 7, the information indicated by the pink arrow vector will be lost. Now, assuming that the vector representing the original variable as $\mathbf{x}_i$, the vector representing the new variable as $\mathbf{y}_i$, and the vector representing the lost information as $\mathbf{e}_i$, the square of the variation of the original variable is

$$\begin{aligned} |\mathbf{x}_i|^2 &= |\mathbf{y}_i + \mathbf{e}_i|^2 \\ &= (\mathbf{y}_i + \mathbf{e}_i)^{\mathrm{T}}(\mathbf{y}_i + \mathbf{e}_i) \\ &= \mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i + \mathbf{e}_i^{\mathrm{T}}\mathbf{y}_i + \mathbf{y}_i^{\mathrm{T}}\mathbf{e}_i + \mathbf{e}_i^{\mathrm{T}}\mathbf{e}_i \\ &= |\mathbf{y}_i|^2 + |\mathbf{e}_i|^2 + 2\mathbf{e}_i^{\mathrm{T}}\mathbf{y}_i \\ &= |\mathbf{y}_i|^2 + |\mathbf{e}_i|^2 \end{aligned} \tag{1}$$

That is, the square of the variation of the original variable can be divided into the square of the variation of the new variable and the square of the variation that is lost in the new variable. This means that the minimization of lost information is synonymous with the maximization of the variability of new variables.

How can we find an axis that maximizes the variability of the new variables? Consider a vector, $\mathbf{u}_1$, that determines the orientation of the axis, and seek that maximizes the variation of the new variable. Since there are infinite possibilities when we consider vectors of various sizes, let the size of vector size as 1 (unit vector) here. That is,

$$|\mathbf{u}_1| = \mathbf{u}_1^{\mathrm{T}}\mathbf{u}_1 = u_{11}^2 + u_{12}^2 = 1 \tag{2}$$

The variance of the value of the new variable z_{1i} under this condition

$$\frac{1}{n-1}\sum_{i=1}^{n} z_{1i}^2 \tag{3}$$

We will maximize this value. z_{1i} is the position of the foot of the perpendicular, and the inner product of $\mathbf{u}_1$ and $\mathbf{x}_i$

$$z_{1i} = \mathbf{x}_i^{\mathrm{T}}\mathbf{u}_1 = u_{11}x_{1i} + u_{12}x_{2i} \tag{4}$$

Note that the relationship with in equation (1) is

$$\mathbf{y}_i = z_{1i}\mathbf{u}_1$$

To maximize Equation (3) under the condition of Equation (2), use the method of Lagrange's undetermined multipliers. That is,

$$L(\mathbf{u}_1, \lambda) = \frac{1}{n-1}\sum_{i=1}^{n}(u_{11}x_{1i} + u_{12}x_{2i})^2 - \lambda(u_{11}^2 + u_{12}^2 - 1)$$

First, partially differentiate the above equation with u_{11} and u_{12}.

$$\frac{\partial L}{\partial u_{11}} = \frac{1}{n-1}\sum_{i=1}^{n} 2(u_{11}x_{1i} + u_{12}x_{2i})x_{1i} - 2\lambda u_{11} = 0$$
$$\frac{\partial L}{\partial u_{12}} = \frac{1}{n-1}\sum_{i=1}^{n} 2(u_{11}x_{1i} + u_{12}x_{2i})x_{2i} - 2\lambda u_{12} = 0$$

The above formula can be arranged as

$$\frac{1}{n-1}\left(u_{11}\sum_{i=1}^{n} x_{1i}^2 + u_{12}\sum_{i=1}^{n} x_{1i}x_{2i}\right) = \lambda u_{11}$$
$$\frac{1}{n-1}\left(u_{11}\sum_{i=1}^{n} x_{1i}x_{2i} + u_{12}\sum_{i=1}^{n} x_{2i}^2\right) = \lambda u_{12}$$

If the above two expressions are expressed using a matrix

$$\frac{1}{n-1}\begin{pmatrix} \sum_{i=1}^{n} x_{1i}^2 & \sum_{i=1}^{n} x_{1i}x_{2i} \\ \sum_{i=1}^{n} x_{1i}x_{2i} & \sum_{i=1}^{n} x_{2i}^2 \end{pmatrix}\begin{pmatrix} u_{11} \\ u_{12} \end{pmatrix} = \lambda \begin{pmatrix} u_{11} \\ u_{12} \end{pmatrix}$$

Here, note that in the portion from the top of the left side to the matrix, the diagonal components represent variances, while the nondiagonal component are covariances. Now, let the variance-covariance matrix be $\mathbf{V}$ and the vector representing the orientation of the axis be $\mathbf{u}_1$

$$\mathbf{V}\mathbf{u}_1 = \lambda \mathbf{u}_1 \tag{5}$$

For the equation (5), $\mathbf{u}_1 = \mathbf{0}$ is a self-evident solution, but is not the solution we are going to solve. Finding a solution for the matrix $\mathbf{V}$, except for which Eq. (5) holds, is called the eigenvalue problem. We call the vector $\mathbf{u}_1$ as an eigenvector of $\mathbf{V}$ and λ as its eigenvalues.

To summarize the results, "to find a new variable that best describes the variation of the original variable" eventually corresponds to "to find the variance-covariance matrix of the original variable and find its first eigenvector".

Equation (5) can be rewritten as follows.

$$(\mathbf{V} - \lambda \mathbf{I})\mathbf{u}_1 = \mathbf{0}$$

The determinant of the matrix representing the coefficients of the above equation must be 0 in order for this equation to have a solution other than $\mathbf{u}_1 = \mathbf{0}$. That is,

$$|\mathbf{V} - \lambda \mathbf{I}| = 0$$

This equation is called an eigen (or characteristic) equation.

Here, although the case where the number of variables is two has been described as an example, in general, if there are *m* variables, then $\mathbf{V}$ is an *m* × *m* variance-covariance matrix. If the matrix $\mathbf{V}$ is an *m* × *m* symmetric matrix (the variance-covariance matrix is always a symmetric matrix), $\mathbf{V}$ has *m* real eigenvalues $\lambda_1,\ldots,\lambda_m$ and the corresponding eigenvectors

$\mathbf{u}_1,...,\mathbf{u}_m$, which are unit vectors in which all elements are real numbers and orthogonal to each other. Now, if eigenvectors are rearranged in descending order of eigenvalues ($\lambda_1 \geq ... \geq \lambda_m$), $\mathbf{u}_1,...,\mathbf{u}_m$ becomes eigenvectors of the first, ..., m-th principal components.

Let's perform principal component analysis according to the calculation procedure mentioned above. First, calculate the variance-covariance matrix $\mathbf{V}$.

```
> cov <- var(mydata)      #  calculate the covariance matrix
> cov
             panicle.length leaf.length
panicle.length      12.67168    11.57718
leaf.length        11.57718    33.41344
```

Next, perform eigenvalue decomposition of the variance-covariance matrix. We use the function eigen for eigenvalue decomposition.

```
> eig <- eigen(cov)                 #  eigenvalue decomposition
> eig                       #  eigenvalues and eigenvectors are obtained
$values
[1] 38.585610  7.499513

$vectors
          [,1]       [,2]
[1,] 0.4078995 -0.9130268
[2,] 0.9130268  0.4078995
```

Eigenvalue decomposition yields eigenvalues (eigenvalues), λ_1, λ_n, and eigenvectors (eigenvectors), $\mathbf{u}_1, \mathbf{u}_n$.

```
> res <- prcomp(mydata)
> res
Standard deviations:
[1] 6.211732 2.738524

Rotation:
                     PC1        PC2
panicle.length -0.4078995 -0.9130268
leaf.length    -0.9130268  0.4078995
> sqrt(eig$values)
[1] 6.211732 2.738524
```

Standard deviations are the square root of the eigenvalues. This is because the variance of the new variables (called principal component scores) is same as the eigenvalue, as described later. In addition, rotation is represented by eigenvectors, and both results are identical except for the difference between positive and negative. The sign of the coefficient depends on which side of the axis is a positive value, but in some cases it may be upside down because there is no rule to uniquely determine it. In the present result, the result of using the function prcomp and the result of using the function eigen are opposite in the positive / negative of the first principal component score.

Now let's calculate the value of the new variable, i.e., the principal component score. Principal component scores can be calculated using equation (4). For example, the principal component score of the first sample can be calculated as follows:

```
> mydata[1,]                          #  the data of the first sample
  panicle.length leaf.length
1      -3.995677   -2.221425
> eig$vectors[,1]                     #  the eigen vector of the first PC
[1] 0.4078995 0.9130268
> mydata[1,1] * eig$vectors[1,1] + mydata[1,2] * eig$vectors[2,1]
[1] -3.658055
> res$x[1,1]      #  compare with the result of function prcomp
[1] 3.658055
```

To calculate principal component scores for all samples and all principal components at once: That is, it is calculated as the product of the matrix of eigenvectors and the data matrix.

```
> score <- as.matrix(mydata) %*% eig$vectors
> head(score)
       [,1]      [,2]
1 -3.658055  2.7420420
（以下省略）
> head(res$x)
        PC1       PC2
1  3.658055  2.7420420
（以下省略）
```

The obtained principal component scores and the principal component scores obtained using the function prcomp are identical except for the positive / negative of the first principal component scores (the positive / negative being reversed is the positive / negative as described above).

Let's examine the variance and covariance of principal component scores. Compares the elements of this matrix with the eigenvalues.

```
> var(score)
             [,1]         [,2]
[1,] 3.858561e+01 6.763202e-16
[2,] 6.763202e-16 7.499513e+00
> eig$values
[1] 38.585610  7.499513
```

The above results show two important points. One is that the covariance of the first principal component and the second principal component is zero. It turns out that there is no redundancy in both components. The other is that the variance of the principal component scores matches the eigenvalues of the principal components. This relationship can be derived as follows:

$$
\begin{aligned}
&\frac{1}{n-1}\sum_{i=1}^{n} z_{ji}^2 \\
&= \frac{1}{n-1}\mathbf{z}_j^{\mathrm{T}}\mathbf{z}_j = \frac{1}{n-1}(\mathbf{X}\mathbf{u}_j)^{\mathrm{T}}(\mathbf{X}\mathbf{u}_j) = \frac{1}{n-1}\mathbf{u}_j^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{u}_j \\
&= \mathbf{u}_j^{\mathrm{T}}\mathbf{V}\mathbf{u}_j \qquad \left(\because \frac{1}{n-1}\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{V}\right) \\
&= \lambda_j \mathbf{u}_j^{\mathrm{T}}\mathbf{u}_j \qquad \left(\because \mathbf{V}\mathbf{u}_j = \lambda_j \mathbf{u}_j\right) \\
&= \lambda_j \qquad \left(\because \mathbf{u}_j^{\mathrm{T}}\mathbf{u}_j = 1\right)
\end{aligned}
$$

where z_{ji} is the jth principal component score of the ith sample. $\mathbf{z}_j = \left(z_{i1},\ldots,z_{in}\right)^{\mathrm{T}}$ is a column vector consisting of the scores of all samples of the j-th principal component, and $\mathbf{X} = \left(\mathbf{x}_1,\ldots,\mathbf{x}_n\right)^{\mathrm{T}}$ is a data matrix bundling column vectors $\mathbf{x}_1 = \left(x_{11},\ldots,x_{1m}\right)^{\mathrm{T}}$ consisting of the original m variables of the i-th sample. Because of this relationship, as mentioned earlier, the standard deviation of the principal component scores in the result of the function prcomp coincides with the square root of the eigenvalue in the result of the function eigen.

Let's check another important relationship. The sum of the eigenvalues (ie, the sum of the variances of the principal components) matches the sum of the variances of the original variables, as shown below.

```
> sum(eig$values)
[1] 46.08512
> sum(diag(cov))
[1] 46.08512
```

Therefore, calculating the ratio of the eigenvalue of the jth principal component to the sum of the eigenvalues of all principal components, is same as calculating the ratio of the variance of jth principal component to the sum of the variances of the original variables. This ratio is called the contribution of the j-th principal component. Also, the sum of contributions from the first principal component to the j-th principal component is called the cumulative contribution of the j-th principal component. Contribution and cumulative contribution provide a good basis for determining the number of effective (necessary) components, as discussed later. Now let's calculate the contribution and the cumulative contribution of principal coponents.

```
> eig$values / sum(eig$values)
[1] 0.8372682 0.1627318
> cumsum(eig$values) / sum(eig$values)
[1] 0.8372682 1.0000000
> summary(res)
（結果は省略）
```

You can see that the first principal component accounts for 83.7% of the total variation (sum of the variance of the original variables). This is the same as the result shown when displaying the result of the function prcomp with the function summary.

<Principal Component Analysis Based on Correlation Matrix>

So far, we have discussed principal component analysis based on the variance-covariance matrix. This method cannot be applied when the variables include different measurement scales. This is because it is difficult to give meaning to covariance between variables with different measurement scales.

For example, when considering the covariance between two variables of length and number, the size of the covariance is different when measuring the length in units of *meter* and when measuring in units of *centi-meter*. (The latter will be 100 times larger). Therefore, the result of principal component analysis based on the variance-covariance matrix of these two variables changes depending on the unit of measure of length.

Also, for example, even when both of two variables are measured in length, if one is very large compared to the other, the larger variable mainly determines the magnitude of covariance. In principle component analysis, results are obtained that depend mainly on the variation in the larger variable.

The problem above is mainly come from the estimate of covariance has the following form:

$$\sum_i^n (x_i - \bar{x})(y_i - \bar{y}) \Big/ (n-1)$$

Now let's calculate and check the problem specifically. First, let's draw a scatter diagram by extracting the data of panicle length (Panicle.length) and the number of florets in a single panicle (Florets.per.panicle). The panicle length is a variable measured in centi-meter (cm), and the number of florets in a single panicle is a variable measured as a number.

```
> mydata <- data.frame(             #  extract two traits
        panicle.length  = alldata$Panicle.length,
        panicle.florets = alldata$Florets.per.panicle
        )
> missing <- apply(is.na(mydata), 1, sum) > 0
        #  if a sample missing, the corresponding element becomes T
> mydata <- mydata[!missing, ]      # remove samples with missing data
> plot(mydata)
```

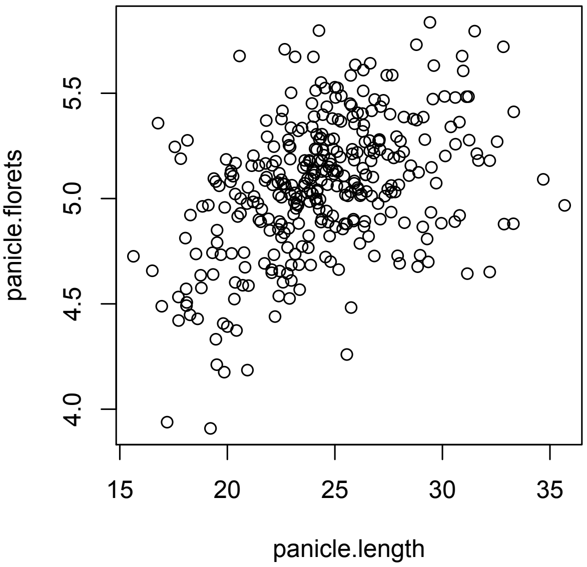

Figure 8. Relationship between panicle length (horizontal axis) and florets number per panicle (vertical axis)

There is a relation that one becomes larger and the other becomes larger

Next, let's do principal component analysis based on the variance-covariance matrix. The point to note is that the next analysis is an "incorrect analysis example".

```
> res <- prcomp(mydata)
> res
Standard deviations:
[1] 3.5623427 0.2901551

Rotation:
                       PC1         PC2
panicle.length  0.99926174 -0.03841834
panicle.florets 0.03841834  0.99926174
```

The analysis results show that the first principal component is a variable that mainly explains panicle length from eigenvectors.

Let’s see what happens if the panicle length is measured in meters (the next analysis is also an "incorrect analysis example").

```
> mydata$panicle.length <- mydata$panicle.length / 100
                                    #  convert from the cm unit to the m unit
> res.2 <- prcomp(mydata)           #  PCA with the m unit
> res.2
Standard deviations:
[1] 0.32097715 0.03220266

Rotation:
                   PC1        PC2
panicle.length  0.04750446 -0.99887103
panicle.florets 0.99887103  0.04750446
```

It turns out that the first principal component is a variable that mainly explains the variation in the number of florets. In other words, the result of the principal component analysis changes completely because the measurement scale is different.

How can we solve this problem? One way is to perform the principal component analysis after scaling the variables to mean 0 and variance 1 respectively. By performing normalization in this way, principal component analysis can be performed without being influenced by the difference in magnitude of variation in each variable. Let's actually calculate it.

```
> mydata.scaled <- scale(mydata)
                  #  make variables have their mean 0 and their variance 1
> var(mydata.scaled)      #  covariance becomes 1
             panicle.length panicle.florets
panicle.length      1.0000000      0.4240264
panicle.florets     0.4240264      1.0000000
> res.scaled <- prcomp(mydata.scaled)       #  PCA with scaled data
> res.scaled
Standard deviations:
[1] 1.1933258 0.7589292

Rotation:
                  PC1       PC2
panicle.length  0.7071068 -0.7071068
panicle.florets 0.7071068  0.7071068
```

As a result of analysis, it can be seen that the first principal component is a variable that explains that both variables become large, and the second principal component is a variable that explains that the other becomes smaller when one becomes larger.

Note that the variance-covariance matrix calculated between variables scaled in this way matches the correlation matrix calculated between the variables before scaling. Thus, in other words, instead of the eigenvalue decomposition of the variance-covariance matrix, the eigenvalue decomposition of the correlation matrix produces the same result. Let's confirm this using the function eigen.

```
> eigen(cov(mydata.scaled))
$values
[1] 1.4240264 0.5759736

$vectors
          [,1]       [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068

> eigen(cor(mydata))
（結果を省略）
```

The prcomp function performs principal component analysis based on the correlation matrix when the option scale = T is specified.

```
> res.scaled.2 <- prcomp(mydata, scale = T)
> res.scaled.2
Standard deviations:
[1] 1.1933258 0.7589292

Rotation:
                     PC1        PC2
panicle.length  0.7071068 -0.7071068
panicle.florets 0.7071068  0.7071068
> res.scaled
（結果は省略）
```

In fact, principal component analysis based on a correlation matrix of two variables always calculates the same eigenvector. Also, if the correlation between two variables is r, the eigenvalues are always $1 + r$ and $1 - r$. You can understand the mechanism by looking at the formula shown below.

Assuming that the correlation matrix between two variables is

$\mathbf{R} = \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}$, λ making the eigen (characteristic) equation equal to 0 is

$$|\mathbf{R} - \lambda \mathbf{I}| = 0 \Leftrightarrow (1-\lambda)^2 - r^2 = 0$$

It is obtained as a solution of

$$\lambda_1 = 1 + r, \quad \lambda_2 = 1 - r$$

When the eigenvalue is λ_1, the eigenvectors satisfy $\mathbf{R}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$.

That is,

$$u_{11} + r u_{12} = (1+r) u_{11}$$
$$r u_{11} + u_{12} = (1+r) u_{12}$$

If you solve them

$$u_{11} = u_{12} = \frac{1}{\sqrt{2}} \approx 0.71$$

Similarly, the eigenvectors $\mathbf{u}_2$ for the eigenvalue λ_2 can be obtained,

$$u_{11} = -\frac{1}{\sqrt{2}}, \quad u_{12} = \frac{1}{\sqrt{2}}.$$

<Application to multivariate data>

So far, I have explained the principle component analysis based on two examples of variables. However, in most cases where principal component analysis is actually used, data consisting of a large number of variables is often analyzed. Here, while analyzing data consisting of seven variables, we will explain how to determine the number of principal components and how to interpret the meaning of principal components.

First, extract seven variables (leaf.length, leaf.width, plant.height, panicle.number, panicle.length, seed.length, seed.width).

```
> plot(res)
> mydata <- data.frame(
        leaf.length = alldata$Flag.leaf.length,
        leaf.width  = alldata$Flag.leaf.width,
        plant.height = alldata$Plant.height,
        panicle.number = alldata$Panicle.number,
        panicle.length = alldata$Panicle.length,
        seed.length = alldata$Seed.length,
        seed.width = alldata$Seed.width
        )
> missing <- apply(is.na(mydata), 1, sum) > 0
> mydata <- mydata[!missing, ]
```

Let's perform principal component analysis based on the correlation matrix.

```
> res <- prcomp(mydata, scale = T)  #  分散１に基準化して主成分分析
> summary(res)
Importance of components:
                          PC1    PC2    PC3     PC4    PC5     PC6     PC7
Standard deviation     1.5626 1.2797 1.0585 0.77419 0.7251 0.64540 0.50854
Proportion of Variance 0.3488 0.2339 0.1601 0.08562 0.0751 0.05951 0.03694
Cumulative Proportion  0.3488 0.5827 0.7428 0.82844 0.9035 0.96306 1.00000
> plot(res)                 #  主成分得点の分散（固有値）の棒グラフを表示
```

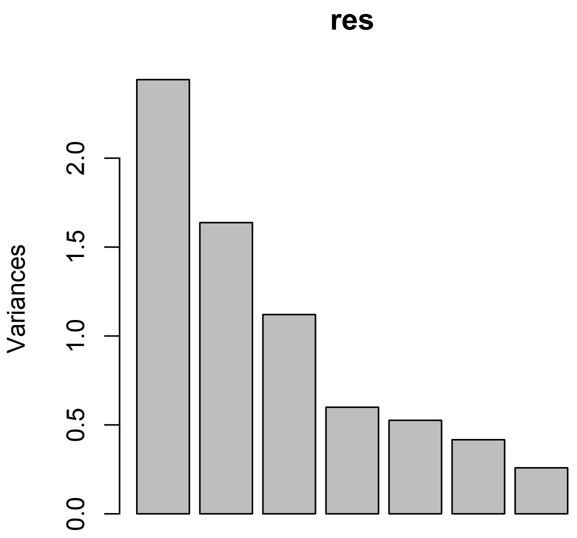

Figure 9. Bar chart showing variance (eigenvalue) of principal component scores. Sum of eigenvalues equals to the number of variables in principal component analysis based on correlation matrix

While seven principal components are calculated for data consisting of seven variables, how many principal components should be used to summarize the data? Although various methods have been proposed as methods for determining the number of effective principal components, here I present some simple rules.

1. Adopt the number of principal components whose cumulative contribution exceeds a specified percentage. 70% to 90% are often used as a pre-defined percentage.
2. Adopt a principal component whose contribution exceeds the average explanatory power per original variable. When the number of variables is q, a principal component whose contribution exceeds 1 / q is adopted.
3. In the case of the correlation matrix, the above rule adopts the principal component whose "eigen value exceeds 1". However, this standard is often too strict. There is also a report that about 0.7 is appropriate.

4. In the graph of eigen values, use as the number of components the point that changes from abrupt change to gentle change.

Assuming that the percentage determined based on the first rule is 80%, the first four principal components with a cumulative contribution of 82.8% are selected. Next, based on the second rule, the first three principal components whose contribution rate exceeds 1/7 = 14.3% are selected. This is the same with the third rule (However, if the eigenvalue is 0.7 or more, the first five principal components are selected). Finally, in the fourth rule, the eigenvalue decreases rapidly until the fourth principal component, and then decreases gradually. Therefore, the first four principal components are chosen. Combining the above, the first three or four principal components are considered to be the appropriate number of principal components.

Let's draw a scatter plot of the first four principal components. In addition, let's color each subpopulation (Sub.population) in order to see the relationship with genetic structure.

```
subpop <- alldata$Sub.population[!missing]
  #  extract subpopulation information (make sure to remove missing data)
> op <- par(mfrow = c(1,2))        #  arrange graphs in 1 row and 2 columns
> plot(res$x[,1:2], col = as.numeric(subpop))     #  plot of PC1 and 2
> plot(res$x[,3:4], col = as.numeric(subpop))     #  plot of PC3 and 4
> par(op)
> df <- data.frame(subpop = subpop, res$x[,1:3])
> plot_ly(data = df, x = ~PC1, y = ~PC2, z = ~PC3, color = ~subpop, type =
"scatter3d", mode = "markers")
```

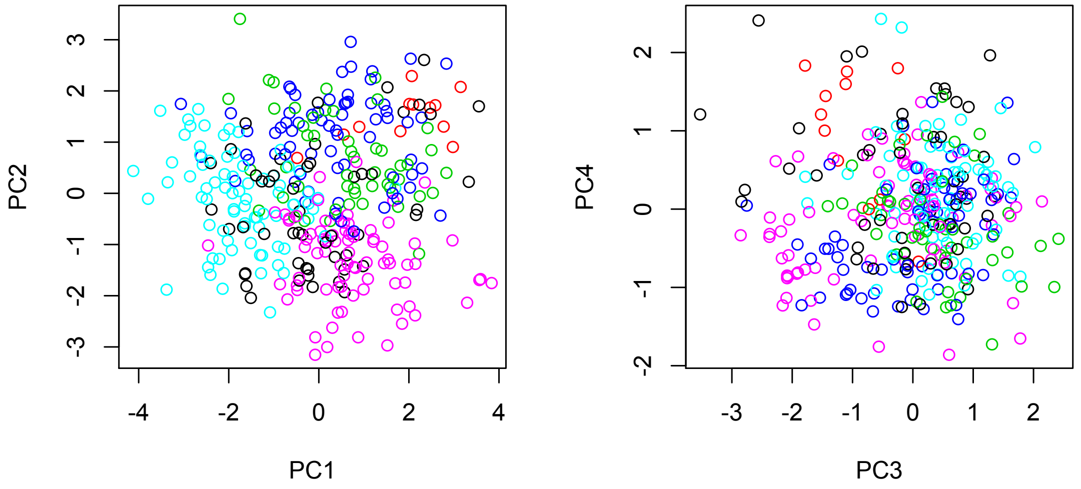

Fig. 10. Scatter plot of the scores of the first to fourth principal components. It can be seen that there is a relationship between principal component scores and genetic background

When you draw a scatter plot, you can see that the dots of the same color are plotted closely. This suggests that there is some relationship between principal component scores and genetic background.

What kind of variation do the first four principal components capture? First, let's look at their eigenvectors.

```
> res$rotation[,1:4]
                       PC1        PC2         PC3         PC4
leaf.length     0.46468878 -0.0863622  0.33735685 -0.28604795
leaf.width      0.26873998 -0.5447022  0.19011509 -0.36565484
plant.height    0.43550572  0.2369710  0.35223063  0.55790981
panicle.number -0.03342277  0.6902669  0.07073948 -0.15465539
panicle.length  0.56777431  0.1140531  0.01542783  0.07533158
seed.length     0.27961838 -0.2343565 -0.67403236  0.42404985
seed.width     -0.34714081 -0.3086850  0.51615742  0.51361303
```

Looking at the eigenvectors, the first principal component is a positive value except for the number of panicles (panicle.number) and the width of the seed (seed.width), and it is a variable that explains the “size” excluding the width of the seed. The second principal component shows that the number of panicles (panicle.number) has a relatively large weight.

In this way, it is difficult to interpret the meaning of the main component based on the numerical values. To visualize it, draw a biplot graph.

```
> op <- par(mfrow = c(1,2))        #  arrange graphs with 1 row by 2 columns
> biplot(res, choices = 1:2)       #  biplot between PC 1 and 2
> biplot(res, choices = 3:4)       #  biplot between PC 3 and 4
> par(op)                          #  reset graph parameters
```

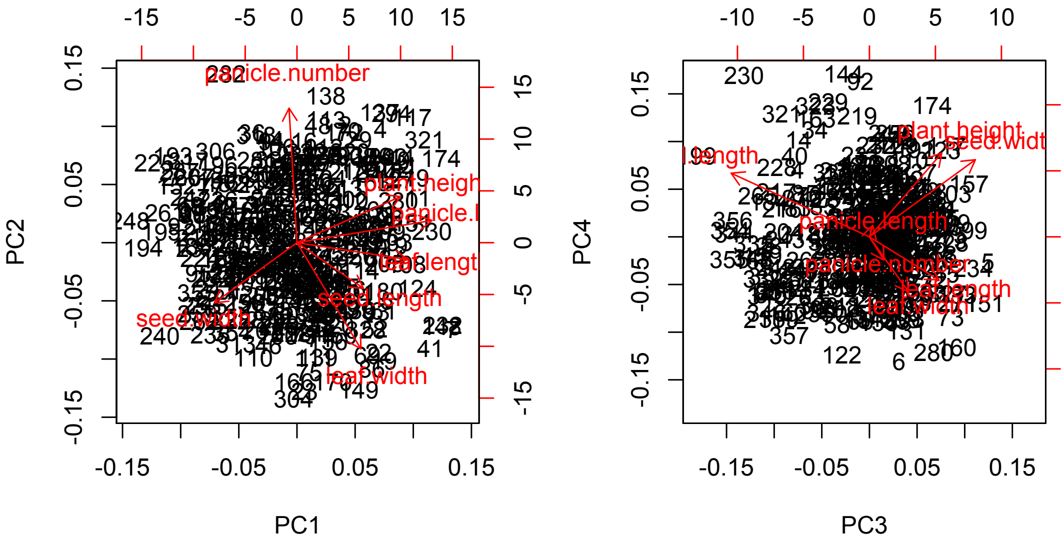

Fig. 11. Results of biplot of the first to fourth principal components

The size and orientation of an arrow with a variable name indicates the relationship between each principal component and that variable. For example, from the left figure, samples which have large scores of the first principal component (for example, 174, 230, etc.) have larger values in plant height, panicle length, length of flag leaf. Among the variables whose length was measured, only the width of the seed has the opposite direction to the other variables, suggesting that the first principal component takes a smaller score when the seed width is larger. It can be seen that the number of panicles is not directed to the first principal component direction, suggesting that it is not involved in the first principal component. The third and fourth principal components can be interpreted in the same way.

There is a statistic called as factor loading, which represents the relation between the original variable and the principal component. Factor loading is the correlation coefficient between the value of the original variable and the principal component score. When the absolute value of this correlation coefficient is close to 1, it indicates that the relationship is strong, while, when it is close to 0, it indicates that the relationship is weak or absent.

Now let's calculate the factor loadings of the variables.

```
> factor.loadings <- cor(mydata, res$x[,1:4])
  # correlation between original variables and principal component scores
> factor.loadings
                       PC1        PC2         PC3         PC4
leaf.length     0.72611038 -0.1105166  0.35710695 -0.22145439
leaf.width      0.41992598 -0.6970483  0.20124513 -0.28308496
plant.height    0.68050970  0.3032487  0.37285150  0.43192611
panicle.number -0.05222553  0.8833255  0.07488083 -0.11973208
panicle.length  0.88718910  0.1459523  0.01633103  0.05832068
seed.length     0.43692427 -0.2999030 -0.71349267  0.32829357
seed.width     -0.54243303 -0.3950201  0.54637516  0.39763215
```

Let's draw graphs for this result. The factor loading can be drawn in the following graphs because it fits in a circle of radius 1.

```
> theta <- 2 * pi * (0:100 / 100)
> x <- cos(theta)
> y <- sin(theta)
> op <- par(mfrow = c(1,2))
> plot(factor.loadings[,1:2], xlim = c(-1,1), ylim = c(-1,1), pch = 4)
> text(factor.loadings[,1:2], rownames(factor.loadings), col = "red")
> lines(x, y, col = "gray")
> abline(v = 0, h = 0)
> plot(factor.loadings[,3:4], xlim = c(-1,1), ylim = c(-1,1), pch = 4)
> text(factor.loadings[,3:4], rownames(factor.loadings), col = "red")

> lines(x, y, col = "gray")
> abline(v = 0, h = 0)
> par(op)
```

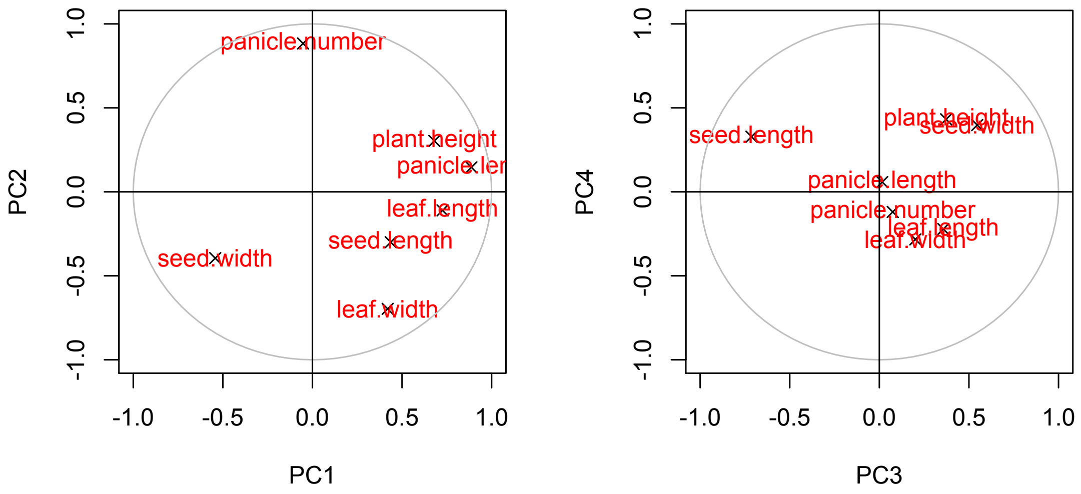

Figure 12. Factor loadings of the first to fourth principal components. The variables scattered closer to the circle have a stronger relationship with their principal components. Conversely, variables scattered near the origin have a weak relationship with their principal components

In the principal component analysis based on the correlation matrix, the factor loading can also be calculated as follows.

```
> factor.loadings <- t(res$sdev * t(res$rotation))[,1:4]
> factor.loadings
 (omitted)
```

Finally, let's perform principal component analysis of marker genotype data. First, we extract marker genotype data.

```
> mydata <- alldata[, 50:ncol(alldata)]
> dim(mydata)
[1]  374 1311
> head(mydata)[,1:10]
 (omitted)
```

This data is data with a very large number of variables (1311 variables). Another feature is that the number of variables is greater than the number of samples. Let's use this data to perform principal component analysis based on the variance-covariance matrix.

```
> res.pca <- prcomp(mydata) #  PCA based on variance covariance matrix
> summary(res.pca)                #  show result
（結果は省略）
> plot(res.pca)                   #  draw graphs
```

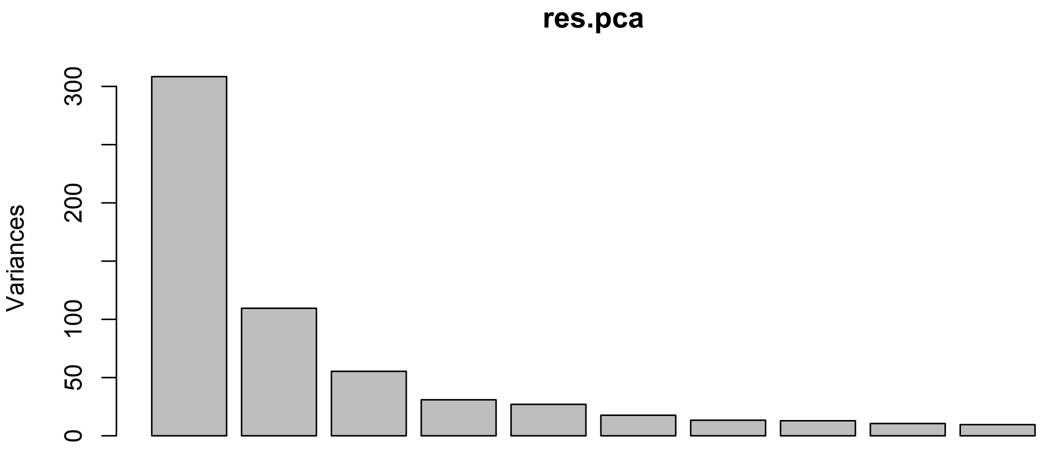

Figure 13. Bar chart of variance(eigenvalue) of principal component scores

According to the fourth rule shown above, it can be judged that it is better to use the first four principal components (other rules cause the number of principal components to be too large).

Let's draw a scatter plot of the first four principal components.

```
> subpop <- alldata$Sub.population          #  information about subpop
> op <- par(mfrow = c(1,2))
> plot(res.pca$x[,1:2], col = as.numeric(subpop))
> plot(res.pca$x[,3:4], col = as.numeric(subpop))
> legend(-10, 20, levels(subpop), col = 1:nlevels(subpop), pch = 1, cex =
0.5)
                                            #  add legend
```

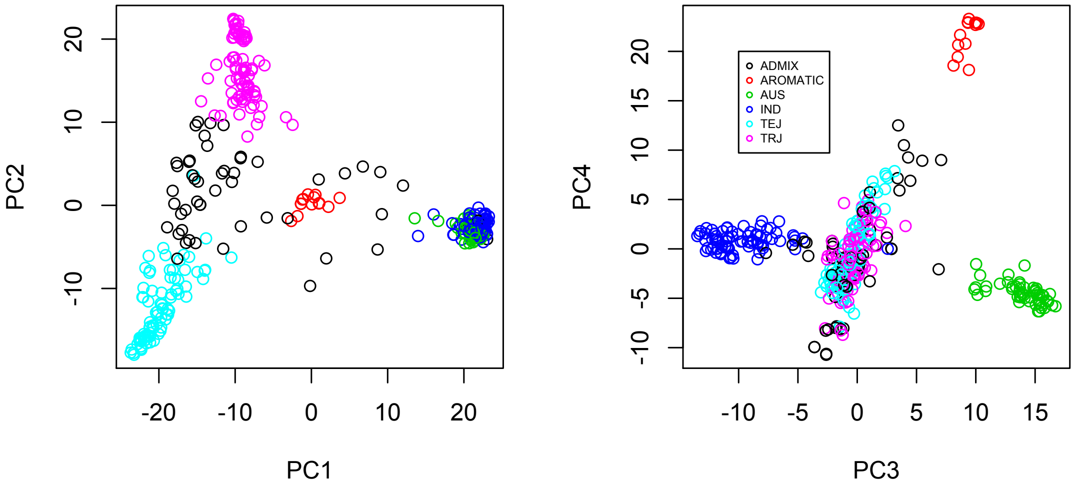

Figure 13. Results of principal component analysis based on marker genotype data

Varieties can be divided into five groups (+1 group) based on the scores of the first to fourth principal components

Let's also draw a 3D scatter plot.

```
> df <- data.frame(subpop = subpop, res.pca$x[,1:4])
> plot_ly(data = df, x = ~PC1, y = ~PC2, z = ~PC3, color = ~subpop, type =
"scatter3d", mode = "markers")
> plot_ly(data = df, x = ~PC2, y = ~PC3, z = ~PC4, color = ~subpop, type =
"scatter3d", mode = "markers")
```

Finally, let's compare PC1-4 included in alldata with the first to fourth principal components calculated this time.

```
> cor(alldata[,c("PC1","PC2","PC3","PC4")], res.pca$x[,1:4])
             PC1         PC2         PC3         PC4
PC1  0.988907541 -0.11988231 -0.03045304 -0.03589106
PC2  0.006737731 -0.07579808  0.96846220 -0.18191250
PC3 -0.129282100 -0.97046613 -0.08514082 -0.03141488
PC4  0.012470575 -0.02915991  0.16366284  0.87422607
```

Although there is not a perfect match because the calculation method is a little different, PC1-4 (rows 1-4) contained in alldata and the first to fourth principal components (rows 1-4) calculated this time provide almost the same information.

<Multidimensional scaling method>

Although it is not possible to directly measure the characteristics of samples, it may be possible to assess differences in characteristics between samples. In other words, although the characteristics of samples cannot be measured as points in multi-dimensional space, it may be possible to measure distances among samples.

For example, in population genetics, genetic distance between populations is calculated based on polymorphisms of genetic markers. In this case, although the distance between the populations is known, the genetic characteristics of the populations are not measured as multivariate data.

Another example is the homology of human impressions to certain objects. For example, let's say that you show photographs of flowers of many varieties of pansy to 100 people, and ask the people to classify the flowers into any number of groups. As a result, when the flowers of two varieties are classified into the same group by many people, it means that flowers are similar between two varieties. Conversely, if the flowers of two varieties are classified into different groups by many people, it means that they were judged as unsimilar. Using such data, it is possible to set the value as the distance between varieties by totaling how many of people classified them into different groups. Again, in this case, although we do not try to characterize the flower of each variety in the multidimensional space of human impression, it is possible to measure the difference in the human impression among the varieties as the distance.

Here, based on the data measured as distance, I will outline the method of summarizing the variation of samples with low-dimensional variables. There are a variety of such methods. Here I will introduce classical multidimensional scaling.

Here, the distances among rice accessions (lines and varieties) are calculated based on marker genotype data, and analysis by multidimensional scaling, which is performed based on the distance matrix.

First, let's extract marker genotype data and calculate the distance matrix based on them.

```
> mydata <- alldata[, 50:ncol(alldata)]    #  50列〜最終列までがマーカーデータ
> D <- dist(mydata)                      #  関数distは距離を計算するための関数
                                         #  様々な定義に基づく距離が計算できるが
                                         #  何も指定しないとユークリッド距離を計算する
```

Let's perform the multidimensional scaling method based on the calculated distance matrix.

```
> res.mds <- cmdscale(D, k = 10, eig = T)
                     #  cmdscaleは古典的多次元尺度構成法のための関数
                     #  Dは距離行列。k = 10は計算する次元数。
                     #  eig = Tは計算した固有値を出力するためのオプション
```

Eigenvalues are calculated as in principal component analysis. In addition, the contribution and the cumulative contribution can be calculated based on the eigenvalues.

```
> res.mds$eig[1:10]                      #  固有値を表示
 [1] 115029.059  40849.407  20648.953  11530.683  10069.314   6591.745
 [7]   4996.271   4819.066   3932.298   3581.676
> res.mds$eig[1:10] / sum(res.mds$eig)
                                         #  固有値を全固有値の和で割る（寄与率）
 [1] 0.310125555 0.110132562 0.055670871 0.031087446 0.027147501 0.017771758
 [7] 0.013470260 0.012992505 0.010601722 0.009656423
> cumsum(res.mds$eig[1:10]) / sum(res.mds$eig)
                               #  固有値の累積和を全固有値の和で割る（累積寄与率）
 [1] 0.3101256 0.4202581 0.4759290 0.5070164 0.5341639 0.5519357 0.5654060
 [8] 0.5783985 0.5890002 0.5986566
> barplot(res.mds$eig[1:10])             #  固有値の棒グラフ
```

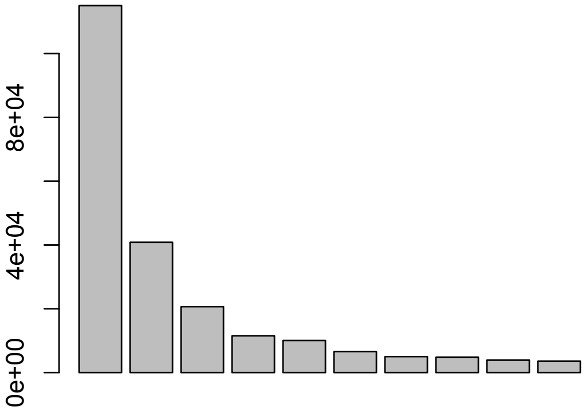

Figure 14. Top 10 eigenvalues calculated by multidimensional scaling

According to the principal component analysis rule based on the bar chart of the eigenvalues in Figure 14, it is considered that the variation contained in the data should be represented by four dimensions.

Now let's draw a scatter plot that shows the placement in 4D space.

```
> subpop <- alldata$Sub.population          #  extract subpopulation info.
> op <- par(mfrow = c(1,2))        #  draw graphs in 1 row x 2 columns
> plot(res.mds$points[,1:2], col = as.numeric(subpop))
                  #  coordinates are stored as points
                  #  plot the 1st and 2nd axes
> plot(res.mds$points[,3:4], col = as.numeric(subpop))
                  #  plot the 3rd and 4th axes
> legend(5, -10, levels(subpop), col = 1:nlevels(subpop), pch = 1, cex = 0.5)
                                                #  add a legend
> par(op)
```

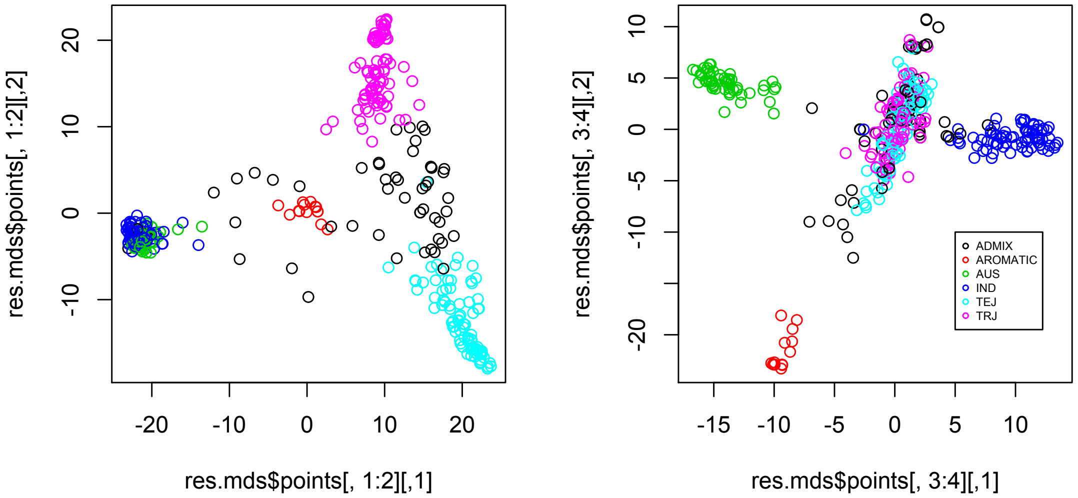

Figure 15. The placement of accessions (varieties and lines) in four-dimensional space determined by multidimensional scaling

Let's draw a three-dimensional scatter plot as well as principal component analysis.

```
> df <- data.frame(subpop = subpop, res.mds$points[,1:4])
> plot_ly(data = df, x = ~X1, y = ~X2, z = ~X3, color = ~subpop, type =
"scatter3d", mode = "markers")
> plot_ly(data = df, x = ~X2, y = ~X3, z = ~X4, color = ~subpop, type =
"scatter3d", mode = "markers")
```

Looking at the results of multidimensional scaling seems to be very similar to the results of principal component analysis. Let's calculate the correlation between the coordinate values and the principal component scores obtained by multidimensional scaling method.

```
> cor(res.pca$x[,1:4], res.mds$points[,1:4])
             [,1]          [,2]          [,3]          [,4]
PC1 -1.000000e+00  6.392724e-15 -3.856842e-16  1.819451e-16
PC2  1.946384e-14  1.000000e+00  7.024318e-16 -2.342063e-16
PC3 -4.646783e-16  8.465118e-16 -1.000000e+00 -1.033884e-15
PC4 -4.482051e-16 -4.494095e-16  8.594246e-16 -1.000000e+00
```

You can see that the first four principal components (rows) and the first four dimensions of the multidimensional scaling are either correlated (-1 or 1) with each other. If you look at this result from a slightly different point of view, even if the values of the original variable are not given, the Euclidean distance matrix based on the values can be used to perform the same analysis as principal component analysis.

Given a distance matrix, as with principal component analysis, being able to represent sample variation with fewer variables is one of the most useful aspects of multidimensional scaling. I will explain this point with a slightly specific example. In the second lecture, it was shown that there is a strong relationship between principal component scores (PC1-4) representing genetic background and plant height (Plant.height). Using the multidimensional scaling method, it is possible to calculate a variable with the same variation as the principal component score from "the relationship between accessions (varieties and lines)". Based on that, You can check whether the genetic relationship affects the plant height. In other words, whether the distance relationship samples samples is related to the variation in another characteristic found in samples by combining multidimensional scaling and another analysis (eg, multiple regression analysis).

Finally, let's make a concrete calculation and check the above-mentioned points.

```
> mydata <- data.frame(   #  combine plant height with the MSD result
        plant.height = alldata$Plant.height,
        res.mds$points[,1:4]
        )
> mydata <- na.omit(mydata)       #  remove missing data
> model <- lm(plant.height ~ ., data = mydata)
                 #  multiple regression analysis
> anova(model)                    #  analysis of variance
 (omitted)
> plot(mydata$plant.height, predict(model))
                 #  biplot of observations and the expectations of the model
```

<Formulation of multidimensional scaling method>
Here, let's consider the problem of finding the arrangement of n samples distributed in q-dimensional space from the "Euclidean distance between samples".

Now, the coordinates where the arrangement of the ith sample in q-dimensional space is represented by a column vector

$$\mathbf{x}_i = (x_{i1}, \ldots, x_{iq})^{\mathrm{T}}$$

At this time, using an n × q matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^{\mathrm{T}}$ obtained by bundling and transposing column vectors of n samples, a matrix that has an inner product of column vectors of n samples as tis element can be expressed as

$$\mathbf{B} = \mathbf{X}\mathbf{X}^{\mathrm{T}}$$

Here, the (i, j) element of the inner product matrix B is the inner product of the i-th j-th sample. That is

$$b_{ij} = \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j = \sum\nolimits_{k=1}^{q} x_{ik} x_{jk} \, .$$

At this time, the Euclidean distance d_{ij} between the i-th sample and the j-th sample can be expressed as

$$\begin{aligned} d_{ij}^2 &= \sum\nolimits_{k=1}^{q} (x_{ik} - x_{jk})^2 \\ &= \sum\nolimits_{k=1}^{q} x_{ik}^2 + \sum\nolimits_{k=1}^{q} x_{jk}^2 - 2\sum\nolimits_{k=1}^{q} x_{ik} x_{jk} \\ &= b_{ii} + b_{jj} - 2b_{ij} \end{aligned}$$

If the center of gravity of x is located at the origin, then

$$\sum\nolimits_{i=1}^{n} x_{ik} = 0$$

If b_{ij} is summed over i and j,

$$\sum\nolimits_{i=1}^{n} b_{ij} = \sum\nolimits_{i=1}^{n} \sum\nolimits_{k=1}^{q} x_{ik} x_{jk} = \sum\nolimits_{k=1}^{q} x_{jk} \sum\nolimits_{i=1}^{n} x_{ik} = 0 .$$

Therefore, the followings hold.

$$\sum_{i=1}^{n} d_{ij}^2 = \sum_{i=1}^{n} (b_{ii} + b_{jj} - 2b_{ij}) = \sum_{i=1}^{n} b_{ii} + nb_{jj}$$

$$\sum_{j=1}^{n} d_{ij}^2 = \sum_{j=1}^{n} (b_{ii} + b_{jj} - 2b_{ij}) = \sum_{j=1}^{n} b_{jj} + nb_{ii} = \sum_{i=1}^{n} b_{ii} + nb_{ii}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} (b_{ii} + b_{jj} - 2b_{ij}) = n\sum_{i=1}^{n} b_{ii} + n\sum_{j=1}^{n} b_{jj} = 2n\sum_{i=1}^{n} b_{ii}$$

Using the above equation, the following relationship holds.

$$b_{ij} = -\frac{1}{2}(d_{ij}^2 - d_{i.}^2 - d_{.j}^2 + d_{..}^2) \tag{6}$$

here,

$$d_{i.}^2 = \frac{1}{n}\sum_{j=1}^{n} d_{ij}^2 = \frac{1}{n}\sum_{i=1}^{n} b_{ii} + b_{ii}$$

$$d_{i.}^2 = \frac{1}{n}\sum_{i=1}^{n} d_{ij}^2 = \frac{1}{n}\sum_{i=1}^{n} b_{ii} + b_{jj}$$

$$d_{..}^2 = \frac{1}{n^2}\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2 = \frac{2}{n}\sum_{i=1}^{n} b_{ii}$$

Equation (6) means that if the distance matrix **D** = $\{d_{ij}\}$ is given, the inner product matrix **B** can be calculated reversely from **D**. The calculation procedure is as follows. First calculate the squares d_{ij}^2 of each element of the distance matrix, and then calculate its row mean $d_{i.}^2$, column mean $d_{.j}^2$, and total mean $d_{..}^2$. Finally, the inner product matrix **B** is obtained by computing b_{ij} according to equation (6).

Once the inner product matrix **B** is obtained, it suffices to find **X** that satisfies $\mathbf{B} = \mathbf{X}\mathbf{X}^{\mathrm{T}}$. To find **X**, perform a spectral decomposition of matrix **B**, as shown below.

Suppose the matrix **B** is a symmetric matrix with q eigenvalues and eigenvectors. That is,

$$\mathbf{B}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1, \quad \cdots, \quad \mathbf{B}\mathbf{u}_q = \lambda_q \mathbf{u}_q .$$

Now, bundling these expressions, we obtain

$$\mathbf{B}(\mathbf{u}_1,\ldots,\mathbf{u}_q) = (\mathbf{u}_1,\ldots,\mathbf{u}_q)\begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_q \end{bmatrix}$$

Let $\mathbf{U} = (\mathbf{u}_1,\ldots,\mathbf{u}_q)$ and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1,\ldots,\lambda_q)$, where $\mathrm{diag}(\lambda_1,\ldots,\lambda_q)$ be a diagonal matrix with diagonal elements $\lambda_1,\ldots,\lambda_q$, then the above equation

$$\mathbf{BU} = \mathbf{U\Lambda} \tag{7}$$

Since the eigenvectors are unit vectors ($\mathbf{u}_i^{\mathrm{T}}\mathbf{u}_i = 1$) and mutually orthogonal ($\mathbf{u}_i^{\mathrm{T}}\mathbf{u}_j = 0$), the matrix U is

$$\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I} \Leftrightarrow \mathbf{U}^{\mathrm{T}} = \mathbf{U}^{-1} \Leftrightarrow \mathbf{UU}^{\mathrm{T}} = \mathbf{I} \tag{8}$$

That is, the inverse of the matrix **U** is simply the transpose of **U**.

From equations (7) and (8), the matrix **B** is

$$\mathbf{B} = \mathbf{U\Lambda U}^{\mathrm{T}}$$

Setting $\mathbf{u}_i^* = \lambda_i^{1/2}\mathbf{u}_i$ the above equation is

$$\mathbf{B} = \mathbf{U}^*\mathbf{U}^{*\mathrm{T}}$$

where $\mathbf{U}^* = (\mathbf{u}_1^*,\ldots,\mathbf{u}_q^*)$.

Therefore, X that satisfies $\mathbf{B} = \mathbf{XX}^{\mathrm{T}}$ is

$$\mathbf{X} = \mathbf{U\Lambda}^{1/2} \tag{9}$$

where $\mathbf{\Lambda}^{1/2} = \mathrm{diag}(\lambda_1^{1/2},\ldots,\lambda_q^{1/2})$.

Thus, the classical multidimensional scaling method eventually becomes the eigenvalue problem of the inner product matrix **B** obtained from the distance matrix **D**. And by solving the eigenvalue problem, we can find the placement of *n* samples in the *q*-dimensional space.

Now let's analyze using the classical multidimensional scaling method without using the function cmdscale in the procedure described above.

First, prepare the data. As before, we use marker genotype data. Calculate the Euclidean distance using the function dist and prepare the distance matrix **D**.

```
> mydata <- alldata[, 50:ncol(alldata)]
> D <- dist(mydata)
```

First, each element of the distance matrix **D** is squared. Next, find the row mean, column mean, and total mean of the squared elements. Row average and column average can be easily calculated using the function apply. Finally, calculate the inner product matrix **B** according to equation (6). The code of R is as follows.

```
> D2 <- as.matrix(D^2)
         #  D2 is the matrix of squares of elemets of the matrix D
> D2i. <- apply(D2, 1, mean)       # row  mean
> D2.j <- apply(D2, 2, mean)       # column mean
> D2.. <- mean(D2)                 #  overall mean
> B <- - 0.5 * (sweep(sweep(D2, 1, D2i.), 2, D2.j) + D2..)
                          #  same as the equation 6
```

Performs eigenvalue decomposition of inner product matrix **B**. Also, calculate coordinate values according to equation (7).

```
> eig <- eigen(B)                  #  eigenvalue decomposition
> eval <- eig$values[1:10]         #  use the first 10 components
> evec <- eig$vectors[,1:10]       #  eigenvectors
> points <- evec * rep(sqrt(eval), each = nrow(evec))
                                   #  calculation following Eq. 7
```

Let's compare it with the result calculated using the function cmdscale. You will see that the results are in agreement.

```
> head(points)
 (omitted)
> head(res.mds$points)
 (omitted)
```

Let's draw a picture last. You will see the same picture as Figure 15. Let's check it.

```
> subpop <- alldata$Sub.population
> op <- par(mfrow = c(1,2))
> plot(points[,1:2], col = as.numeric(subpop))
> plot(points[,3:4], col = as.numeric(subpop))
> legend(5, -10, levels(subpop), col = 1:nlevels(subpop), pch = 1, cex = 0.5)
> par(op)
```

<Report assignment>

Analyze the phenotypic data of rice using principal component analysis (try analyzing with a combination of variables different from those analyzed in the lecture).

Submission method:

・ Create a report as a pdf file and submit it as an email attachment.

・ Send an e-mail to "report@iu.a.u-tokyo.ac.jp".

・ At the beginning of the report, do not forget to write your affiliation, student number, and name.

・ The deadline for submission is June 7