バイオスタティスティクス基礎論
第 3 回　講義テキスト

岩田洋佳
aiwata@mail.ecc.u-tokyo.ac.jp

＜主成分分析＞

農学や生命科学における実験では、同じサンプルについて複数の特徴（characteristics）を計測する場合が少なくありません。例えば、作物の圃場試験では、収量の評価を目的とする場合でも、収量に関連する様々な形質（traits）を同時に調査します。こうして計測された複数の特徴について散布図を描いて眺めることで、何らかの知見を発見できる場合も少なくありません。しかし、計測した特徴の数が多い場合は散布図でデータの変動を把握するのが難しくなります。人間が散布図などを用いて直感的に把握できる次元の数は高々数次元であり、計測した特徴が 10 個以上あるような場合には、データの変動を把握するのは容易ではありません。今回の講義で解説する主成分分析は、多次元データに含まれる変動を、できるだけ情報量を落とさずに低次元データに要約するための方法です。例えば、講義の中で例と示すマーカー遺伝子型データに含まれる変動の要約では、1311 次元のデータを 4 次元程度で要約できることを示します。主成分分析は、多数の変数があるときに、それら変数に含まれている情報を効率よく取り出すのに非常に有効な手法です。

今回の講義では、これまでと同様にイネのデータ（Zhao et al. 2011, Nature Communications 2:467）を用いて説明を進めていきます。なお、今回の講義では品種・系統データ（RiceDiversityLine.csv）と表現型データ（RiceDiversityPheno.csv）だけでなく、マーカー遺伝子型データ（RiceDiversityGeno.csv）も用います。後者のデータは、Zhao ら（2010, PLoS One 5: e10780）が解析に用いた 1,311 SNPs の遺伝子型のデータです。いずれのデータも、Rice Diversity の web ページ http://www.ricediversity.org/からダウンロードしたデータをもとにしています。マーカーデータについては、ソフトウエア fastPHASE（Scheet and Stephens 2006, Am J Hum Genet 78: 629）を用いて欠測値の補間を行ってあります。

まずは 3 種類のデータを読み込んで、それらを結合してみましょう。

```
> line <- read.csv("RiceDiversityLine.csv")
                              # 系統データを line として読み込む
> pheno <- read.csv("RiceDiversityPheno.csv")
                              # 形質データを pheno として読み込む
> geno <- read.csv("RiceDiversityGeno.csv")
                              # 遺伝子型データを geno として読み込む
> line.pheno <- merge(line, pheno, by.x = "NSFTV.ID", by.y = "NSFTVID")
         # line の NSFTV.ID と pheno の NSFTVID が一致するようにデータを結合
> alldata <- merge(line.pheno, geno, by.x = "NSFTV.ID", by.y = "NSFTVID")
         # line と pheno を結合したデータにさらに geno を結合
```

最初に、イネの遺伝資源に含まれる品種・系統にみられる穂の長さ（panicle length）と止め葉の長さ（flag leaf length）の変異を主成分分析で解析してみましょう。まずは、両形質のデータを全データ（alldata）から抜き出します。なお､両形質のデータのうち欠測値を 1 つでも持つサンプルは除いておきます。

```
> mydata <- data.frame(
         panicle.length  = alldata$Panicle.length,
         leaf.length = alldata$Flag.leaf.length
         )
         # alldata の Panicle.length を panicle.length として
         # Flag.leaf.length を leaf.length として抜き出したデータを作成
> mydata
（結果省略）
> missing <- apply(is.na(mydata), 1, sum) > 0
         #  is.na は NA かどうかを T, F で返す関数
         #  返した結果を行毎に和をとり、NA が 1 個以上あるかどうかをチェックしている
         #  Missing は F,T がサンプル数分含まれるベクトル
         #  そのサンプルが 1 個以上 NA をもつ場合は対応する missing の要素が T となる
> mydata <- mydata[!missing, ]
         #  欠測値をもつサンプル（missing が T になっているサンプル）を除く
> mydata
（結果省略）
```

穂の長さと止め葉の長さの変動（variation）を、散布図で確認してみましょう。

```
> plot(mydata)              #  散布図を描く
> lim <- range(mydata)      #  縦横の軸の範囲を合わせるために両形質の値の範囲を調べる
> plot(mydata, xlim = lim, ylim = lim)
                            #  両軸の範囲を合わせた散布図を描く
```

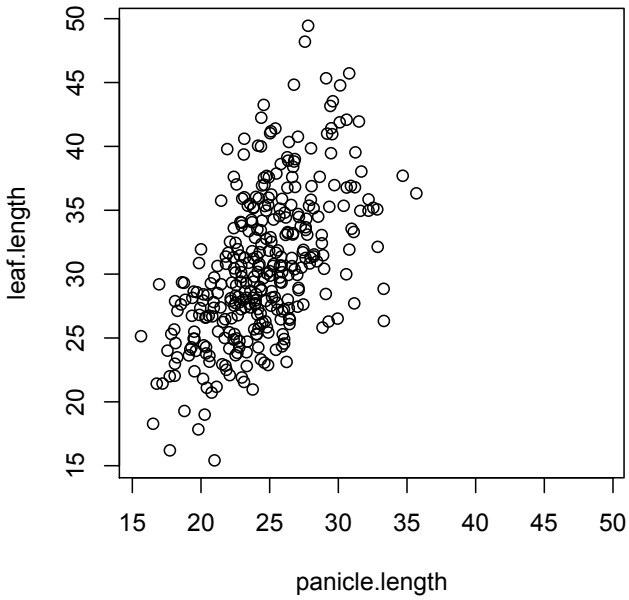


図 1. イネ遺伝資源の 341 品種・系統にみられる穂の長さと止め葉の長さの関係
いずれの長さも cm 単位で計測されている

散布図をみると一方の形質が大きくなると他方も大きくなる傾向がみられます。両形質の分散共分散行列と相関行列を計算して数値で確認してみましょう。

```
> cov(mydata)                          #  共分散行列を計算する
             panicle.length leaf.length
panicle.length      12.67168    11.57718
leaf.length         11.57718    33.41344
> cor(mydata)                          #  相関行列を計算する
             panicle.length leaf.length
panicle.length      1.000000    0.562633
leaf.length         0.562633    1.000000
```

相関も共分散も正の値になっており、両者が共に変動している傾向が数値でも確認できます。

以降の計算や解説を簡単にするために、各形質について平均 0 となるように基準化しておきます（元の変数から平均を引いておく）。

```
> mydata <- sweep(mydata, 2, apply(mydata, 2, mean))
         #  関数 apply を用いて列平均を計算し、それを各列から関数 sweep で引き算する
> summary(mydata)
（結果省略）
> 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)              #  x = 0, y = 0 の線を引く
```

なお、変数を基準化しても分散共分散行列と相関行列で表される変数間の関係は変化しないことに注意しましょう。

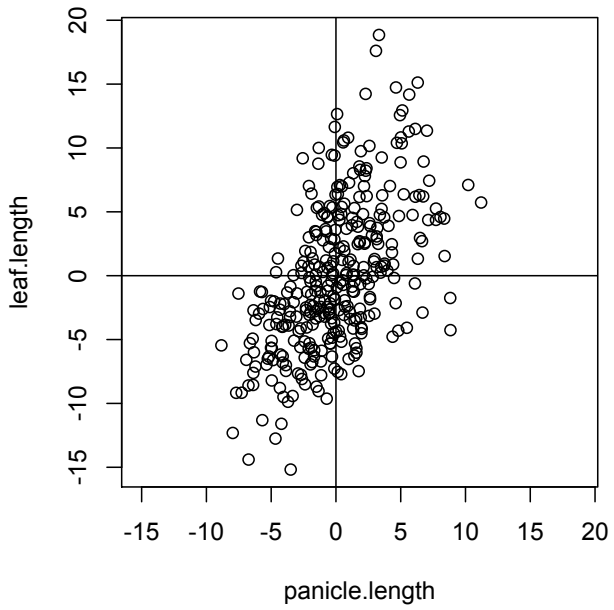


図 2. 平均 0 に基準化された穂の長さと止め葉の長さ

では、主成分分析を行い、得られた主成分得点をプロットしてみましょう。

```
> res <- prcomp(mydata)    #  関数 prcomp を用いて主成分分析を行う
> lim <- range(res$x)      #  res$x として計算された主成分得点を取り出すことができる
> plot(res$x, xlim = lim, ylim = lim)        #  主成分得点の散布図を描く
> abline(h = 0, v = 0)
```

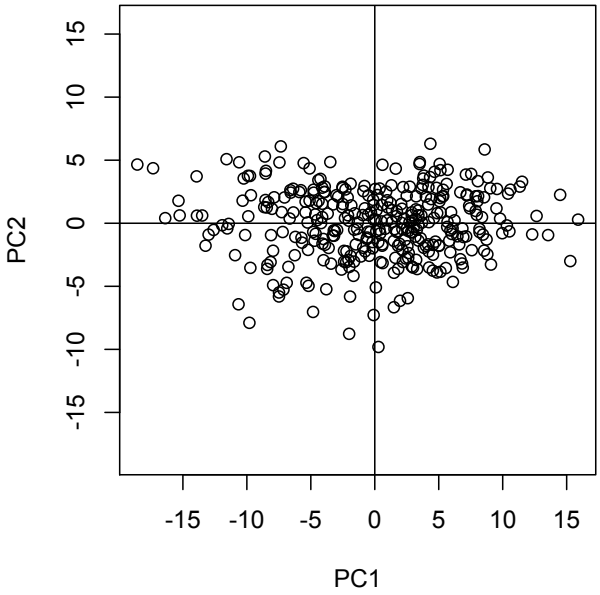


図 3. 横軸が第 1 主成分（PC1）得点、縦軸が第 2 主成分（PC2）得点

主成分得点と元の変数の関係を並列して散布図を描いて確認してみましょう。

```
> op <- par(mfrow = c(1,2))          # 関数 par はグラフ描画のオプション設定用の関数
                                     # mfrow… は 1 行 2 列でグラフを描くという設定
> 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)          # 描画オプションを元に戻す
                   # op は最初の行で描画設定を行ったときの変更前のオプション設定
```

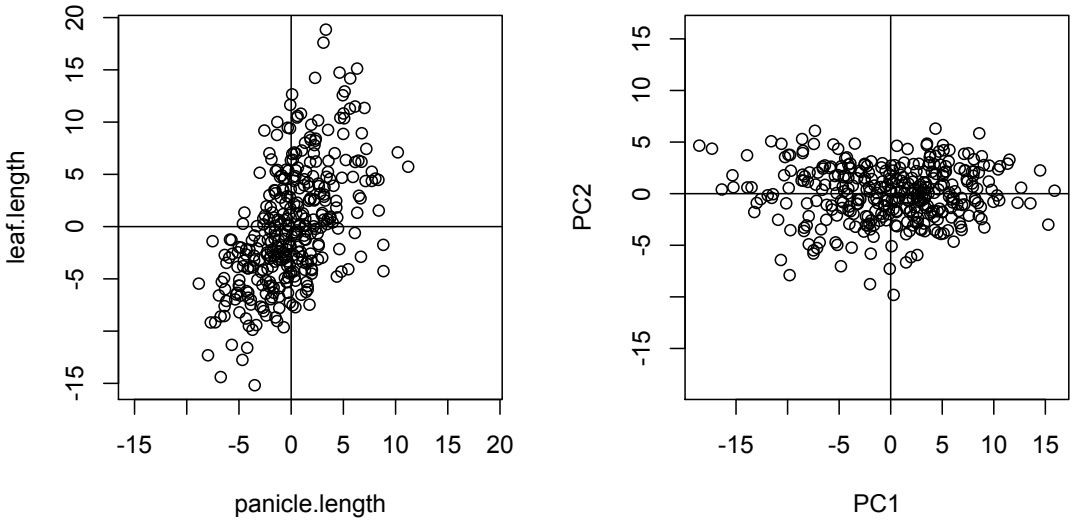


図 4. （左）元の変数、（右）主成分得点

主成分得点と元の変数を並べて見比べると、主成分得点が元の変数を回転（および反転）させたものとなっていることが分かります。主成分分析とは、元の変数の変動を、できるだけ少ない次元の新しい変数で代表させて表現するための方法です。例えば右図の横軸は、穂の長さと止め葉の長さという 2 つの変数の変動を、1 つの新しい変数（すなわち、第 1 主成分）の変動として表現しているものです。第 1 主成分だけで、元の変数のもつ変動の大部分が説明できていることが分かります。なお、第 1 主成分で説明できなかった変動を表すのが第 2 主成分の得点です。

では、各主成分が実際にどの程度の変動を説明しているか確認してみましょう。

```
> summary(res)                          # 主成分分析結果の表示
Importance of components:
                       PC1    PC2
Standard deviation    6.2117 2.7385
Proportion of Variance 0.8373 0.1627
Cumulative Proportion  0.8373 1.0000
```

第 1 主成分は全変動の 83.7%を、また、第 2 主成分は残りの 16.3%を説明することが分かります。すなわち、穂の長さと止め葉の長さの変動の 8 割以上を 1 つの変数（第 1 主成分）で表すことができることが分かります。

もう少し詳しく結果をみてみましょう。

```
> res                        #  主成分分析結果が代入されたオブジェクトの名前を入力する
Standard deviations:
[1] 6.211732 2.738524

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

Standard deviations は、新しい変数である第 1、第 2 主成分の標準偏差を表しています。また、Rotation は、新しい変数である第 1、第 2 主成分の軸の向きを表す単位ベクトルを表しています（なお、後で説明するようにこの単位ベクトルを固有ベクトル（eigenvector）とよびます)。

なお、上述した結果は次のようにすると別々に取り出すこともできます。

```
> res$sdev                           #  各主成分の標準偏差
[1] 6.211732 2.738524
> res$rotation                       #  各主成分の軸の向きを表すベクトル
                   PC1       PC2
panicle.length -0.4078995 -0.9130268
leaf.length    -0.9130268  0.4078995
```

主成分分析の結果を図示してみましょう。

```
> op <- par(mfrow = c(1,2))
> plot(res)                  #  各主成分の分散（固有値）の大きさを表す棒グラフ
> biplot(res)                #  主成分得点と変数間の関係を表すバイプロット
> par(op)
```

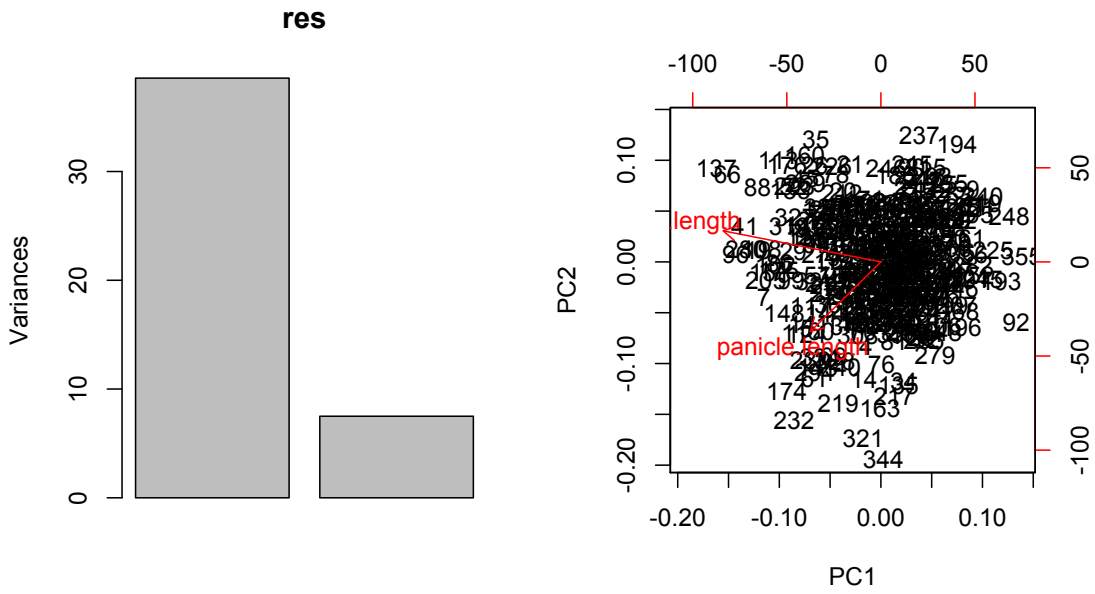


図 5.（左）主成分の分散（固有値）
（右）主成分得点と変数間の関係を表すバイプロット

左図で示されているのは主成分得点の分散で、上で確認した主成分の標準偏差の 2 乗になっています（なお、主成分得点の分散は後で説明するように分散共分散行列の固有値（eigenvalue）となっています)。右図で示されているのは主成分得点と変数間の関係を表すバイプロット（biplot）です。バイプロットをみると、止め葉の長さ（leaf.length）も穂の長さ（panicle.length）も矢印が左側を向いており、横軸を表す第 1 主成分については、両形質が大きいサンプルでは小さな値に、両形質が小さいサンプルでは大きな値になることが分かります。すなわち、第 1 主成分は、「サイズ」を表すような変数であると解釈できます。一方、止め葉の長さ（leaf.length）の矢印は（少し）上側、穂の長さ（panicle.length）の矢印は下側を向いており、縦軸を表す第 2 主成分については、止め葉の長さが大きなサンプルでは大きな値に、穂の長さが大きなサンプルでは小さな値になることが分かります。すなわち、第 2 主成分は止め葉の長さと穂の長さの「比」を表すような変数であると解釈できます。このバイプロットの詳細については後ほど再度解説を行います。

＜主成分分析の定式化＞

ここでは、先ほどの 2 次元データを例に、主成分分析の定式化について概説します。

まず、穂の長さと止め葉の長さの 2 つの変数のもつ変動を、1 つの新しい変数で表すことを考えます。この新しい変数を表す軸の向きを、ここでは仮に $(1/\sqrt{2}, 1/\sqrt{2})$ とします。新しい変数の値は、データ点からこの軸に下ろした垂線の足の位置に相当する値となります。すなわち、以下に描く図の赤線が新しい変数を表す軸、灰色の線分がデータ点から新しい軸に下ろした垂線、緑色の＋が垂線の足になります。この垂線の足の位置が新しい変数の値となります。

```
> lim <- range(mydata)
> plot(mydata, xlim = lim, ylim = lim)       # 散布図を再描画
> abline(h = 0, v = 0)                       # x, y 軸の描画
> u.temp <- c(1 / sqrt(2), 1 / sqrt(2))      # 1/√2, 1/√2 の向きをもつ新変数
> abline(0, u.temp[2] / u.temp[1], col = "red")  # 新変数に対応する軸を描く
> score.temp <- as.matrix(mydata) %*% u.temp   # 新変数の値（内積）を求める
> x <- score.temp * u.temp[1]                # 垂線の足の x 座標を求める
> y <- score.temp * u.temp[2]                # 垂線の足の y 座標を求める
> segments(x, y, mydata$panicle.length, mydata$leaf.length, col = "gray")
                                             # 垂線を灰色の線分で描く
> points(x, y, pch = 4, col = "green")       # 垂線の足を緑色の＋のプロットで描く
```

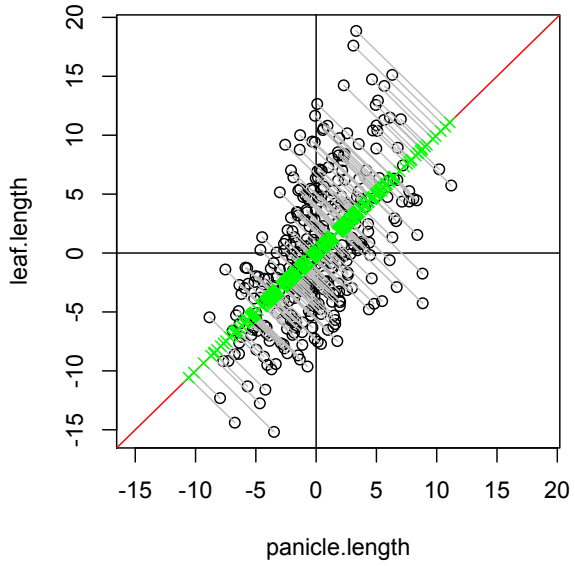


図 6. 仮に設定した新しい変数軸で表された 2 変数の変動

では、１つのサンプルに注目して、元の変数の値と新しい変数の値の関係をもう少し詳しく見てみましょう。ここでは、最も止め葉の長さの長かったサンプルに注目して図を描いてみます。

```
> id <- which.max(mydata$leaf.length)  #  最も止め葉の長いサンプルの順番を調べる
> arrows(0, 0,                         #  矢印の始点（元の変数への矢印）
         mydata$panicle.length[id], mydata$leaf.length[id],   #  矢印の終点
         col = "purple")              #  矢印の色
> arrows(x[id], y[id],
         mydata$panicle.length[id], mydata$leaf.length[id],
         col = "pink")
> arrows(0, 0, x[id], y[id], col = "blue")
```

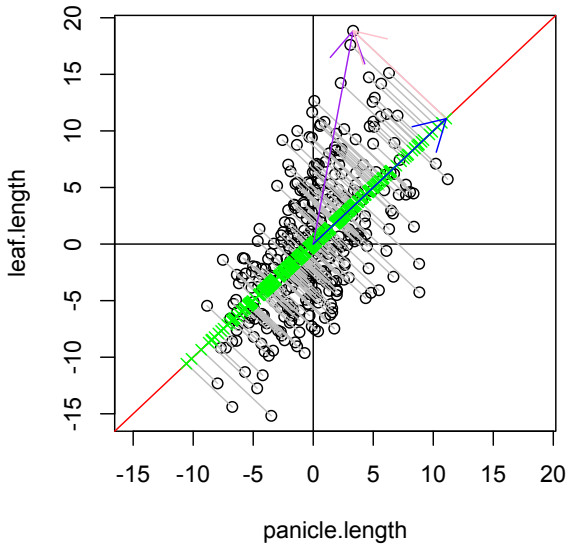


図 7. 元の変数（紫矢印のベクトル）と新しい変数（青矢印のベクトル）

さて、図 7 のように元の変数を新しい変数で表すと、桃色矢印のベクトルで示された情報が失われてしまいます。今、元の変数を表すベクトルを $\mathbf{x}_i$、新しい変数を表すベクトルを $\mathbf{y}_i$、失われてしまう情報を表すベクトルを $\mathbf{e}_i$ とすると、元の変数の変動の 2 乗は、

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

と表せます（ピタゴラスの定理ですね)。すなわち、元の変数の変動の 2 乗は、新しい変数の変動の 2 乗と新しい変数では失われてしまう変動の 2 乗に分割できることになります。したがって失われてしまう情報を最小化しようとすることと、新しい変数の変動を最大化することは同義であることが分かります。

では、新しい変数の変動を最大化するような軸はどのように求めればよいのでしょうか。軸の向きを決めるベクトル$\mathbf{u}_1$とし、新しい変数の変動が最大にするような$\mathbf{u}_1$を求めることを考えます。様々なサイズのベクトルを考えると無限の可能性が生じてしまいますので、ここではベクトルのサイズを 1 とします（単位ベクトル)。すなわち、

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

とします。この条件のもとで新しい変数の値z_{1i}の分散

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

を最大化することを考えます。なお、z_{1i}は垂線の足の位置の値であり、$\mathbf{u}_1$と$\mathbf{x}_i$の内積

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

として表されます。なお、式(1)の$\mathbf{y}_i$とz_{1i}の関係は、

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

となります。

式(2)の条件のもとで式(3)を最大化するには、ラグランジュの未定乗数法を用います。すなわち、

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

を最大化する$\mathbf{u}_1$を求めます。まず、上式をu_{11}およびu_{12}で偏微分します。

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

上式を整理すると

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

上の 2 式を、行列を用いて表すと

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

ここで、左辺の先頭から行列までの部分は、対角成分が分散、非対角成分が共分散の分散共分散行列となっていることに注意しましょう。今、分散共分散行列を $\mathbf{V}$、軸の向きを表すベクトルを $\mathbf{u}_1$ とすると、上式は、

$$\mathbf{V}\mathbf{u}_1 = \lambda\mathbf{u}_1 \qquad (5)$$

と表せます。

式(5)は、$\mathbf{u}_1 = \mathbf{0}$ が自明な解ですが、これは求めようとしている解ではありません。行列 $\mathbf{V}$ に対して、式(5)が成り立つ $\mathbf{u}_1 = \mathbf{0}$ 以外の解を求めることを固有値問題とよびます。また、ベクトル $\mathbf{u}_1$ を $\mathbf{V}$ の固有ベクトル、λ をその固有値とよびます。

ここまでの結果をまとめると、「元の変数の変動を最もよく説明する新しい変数を求める」ということは、結局、「元の変数の分散共分散行列を求め、その固有ベクトルを求める」ことに一致するということになります。

なお、式(5)は次のように書き直すことができます。

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

この式が $\mathbf{u}_1 = \mathbf{0}$ 以外の解をもつためには、上式の係数を表す行列の行列式が 0 でなければいけません。すなわち、

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

この式を固有（または特性）方程式とよびます。

ここでは、変数の数が 2 つの場合を例として説明をしてきましたが、一般に変数の数が m 個ある場合には、$\mathbf{V}$ は m×m の分散共分散行列となります。なお、行列 $\mathbf{A}$ が m×m の対称行列の場合（分散共分散行列は必ず対称行列となります）、$\mathbf{A}$ は m 個の実数の固有値 $\lambda_1,...,\lambda_m$ をもち、対応する固有ベクトル $\mathbf{u}_1,...,\mathbf{u}_m$ は、要素が全て実数で、互いに直交する単位ベクトルとなるように選ぶことができます。今、固有値の大きな順（$\lambda_1 \geq ... \geq \lambda_m$）に固有ベクトルを並び替えると、$\mathbf{u}_1,...,\mathbf{u}_m$ が第 1 ,…, 第 m 主成分の固有ベクトルとなります。

では、上述した計算手順によって主成分分析を行ってみましょう。まずは、分散共分散行列 $\mathbf{V}$ を求めます。

```
> cov <- var(mydata)       #  分散共分散行列を求める。関数 cov を使ってもよい
> cov
             panicle.length leaf.length
panicle.length      12.67168    11.57718
leaf.length         11.57718    33.41344
```

次に、分散共分散行列の固有値分解（eigenvalue decomposition）を行います。固有値分解には関数 eigen を用います。

```
> eig <- eigen(cov)                    #  固有値分解
> eig                          #  固有値と固有ベクトルが求まる
$values
[1] 38.585610  7.499513

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

固有値分解を行うと、固有値（eigenvalues）λ_1,λ_n と固有ベクトル（eigenvectors）$\mathbf{u}_1,\mathbf{u}_n$ が求まります。

関数 eigen で得られた結果を関数 prcomp で得られた結果と見比べてみましょう。

```
> 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
```

標準偏差（standard deviations）は固有値の平方根になっています。これは後で述べるように新しい変数の値（主成分得点 principal component scores とよばれる）の分散が固有値に一致するためです。また、Rotation に表されているのは固有ベクトルで、正負の違い以外は両者の結果は一致しています。なお、係数の正負は、軸のどちら側を正の値とするかに依存しますが、それを一意に決めるルールがないために場合によっては逆さまになることがあります。今回の結果では､関数 prcomp を用いた結果と関数 eigen を用いた結果では第 1 主成分得点の正負が逆向きになっています。

では、次に新しい変数の値、すなわち、主成分得点を計算してみましょう。主成分得点は式(4)を用いて計算できます。例えば、1 番目のサンプルの主成分得点は、以下のようにして計算できます。

```
> mydata[1,]                          #  1 番目のサンプルのデータ
  panicle.length leaf.length
1     -3.995677   -2.221425
> eig$vectors[,1]                     #  第 1 主成分の固有ベクトル
[1] 0.4078995 0.9130268
> mydata[1,1] * eig$vectors[1,1] + mydata[1,2] * eig$vectors[2,1]
         #  第 1 主成分得点は、データベクトルと固有ベクトルの内積として計算できる
[1] -3.658055
> res$x[1,1]      #  関数 prcomp で求められた 1 番目のサンプルの第 1 主成分得点
[1] 3.658055
```

全てのサンプルと全ての主成分について一度に主成分得点を計算するには、以下に示すようにします。すなわち、データ行列に対する、固有値ベクトルを列として束ねた行列の積として計算をします。

```
> 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）
```

求めた主成分得点と関数 prcomp を用いて得られた主成分得点は、第 1 主成分得点の正負を除くと一致しています（正負が逆になるのは、先に説明したように正負の決定が任意であることが原因です)。

では、主成分得点の分散と共分散を調べてみましょう。また、この値を固有値と見くらべてみましょう。

```
> 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
```

上の結果から 2 つの重要な点が分かります。 1 つは第 1 主成分と第 2 主成分の共分散が 0 であることです。これから両方で重複して説明されている情報が無いということが分かります（一方が変動すると、他方が変動するという傾向が無い)。もう 1 つは、主成分得点の分散が、主成分の固有値に一致することです。この関係は以下のようにして導くことができます。

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

ここで、z_{ji} は、i 番目のサンプルの第 j 主成分得点です。また、$\mathbf{z}_j = (z_{i1},...,z_{in})^{\mathrm{T}}$ は第 j 主成分の全サンプルの得点からなる列ベクトル、$\mathbf{X} = (\mathbf{x}_1,...,\mathbf{x}_n)^{\mathrm{T}}$ は、i 番目のサンプルの元の m 個の変数の値からなる列ベクトル $\mathbf{x}_1 = (x_{11},...,x_{1m})^{\mathrm{T}}$ を束ねたデータ行列です。なお、この関係から、先に述べたように、関数 prcomp の結果で示される主成分得点の標準偏差が関数 eigen の結果で示される固有値の平方根に一致します。

もうひとつ重要な関係を確認してみましょう。以下に示すように固有値の和（すなわち主成分の分散の和）は元の変数の分散の和に一致します。

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

したがって、全ての主成分の固有値の和に対する第 j 主成分の固有値の割合を計算すると、それはすなわち、元の変数の分散の和のうち第 j 主成分が説明する比率を表すことになります。この比率のことを第 j 主成分の寄与率（contribution）とよびます。また、第 1 主成分から第 j 主成分までの寄与率の和をとったものを第 j 主成分までの累積寄与率（cumulative contribution）とよびます。寄与率と累積寄与率は、後に述べるように有効な主成分数を決定する際に良い基準となります。では、寄与率と累積寄与率を計算してみましょう。

```
> 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）
```

第 1 主成分によって全変動（元の変数の分散の和）の 83.7%が説明されていることが分かります。これは、関数 prcomp の結果を関数 summary で表示させたときに示される結果と同じものです。

＜相関行列に基づく主成分分析＞

ここまでは、分散共分散行列に基づく主成分分析について説明してきました。この方法は、変数の中に計測尺度が異なるものが含まれる場合には適用することができません。なぜなら、計測尺度の異なる変数間では共分散に対する意味付けが難しくなるためです。

例えば、長さと個数を計った 2 つの変数にみられる共分散を考えた場合、長さを m の単位で計測した場合と cm の単位で計測した場合で共分散の大きさが異なってしまいます（後者が 100 倍大きくなる）。したがって、この 2 変数の分散共分散行列に基づく主成分分析の結果は、長さの計測単位に依存して変化してしまいます。

また、例えば、2 つの変数のいずれも長さの計測をしたものであっても、一方が他方に比べて非常に大きいと、共分散の大きさを決めるのは主に大きいほうの変数となってしまい、主成分分析は、主に大きいほうの変数の変動に依存した結果が得られてしまいます。

上のような問題は、共分散の推定値が

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

というかたちで計算されることによります。

では、この問題について、具体的に計算して確認してみましょう。まず、穂の長さ（Panicle.length）と 1 穂穎花数（Florets.per.panicle）のデータを抜き出して、散布図を描いてみましょう。穂の長さは cm で計測された変数、1 穂穎花数は個数として計測された変数です。

```
> mydata <- data.frame(             #  穂の長さと 1 穂穎花数を抜き出す
         panicle.length  = alldata$Panicle.length,
         panicle.florets = alldata$Florets.per.panicle
         )
> missing <- apply(is.na(mydata), 1, sum) > 0    #  欠測がある場合に T になる
> mydata <- mydata[!missing, ]      # 欠測をもつサンプルを除く
> plot(mydata)
```

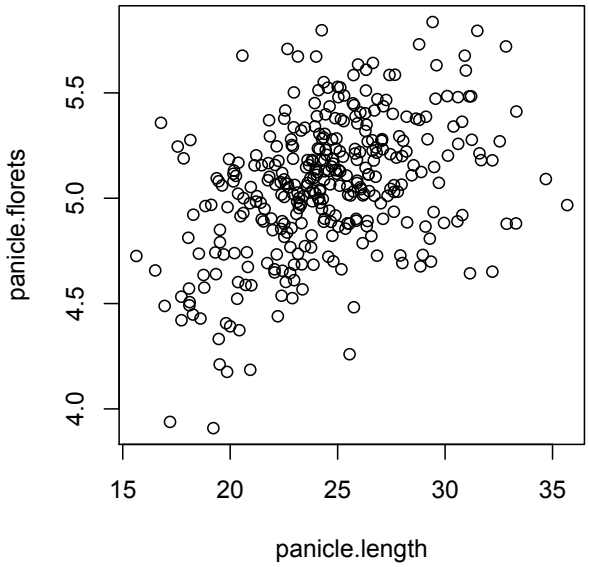


図 8. 穂の長さ（横軸）と 1 穂あたりの穎花数（縦軸）の関係
一方が大きくなると他方が大きくなる関係がある

次に、分散共分散行列に基づく主成分分析を行ってみましょう。注意してほしいことは、次の解析は「間違った解析の例」であるという点です。

```
> 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
```

解析の結果は、固有ベクトルから、第 1 主成分は主に穂の長さを説明する変数であることが分かります。

では、仮に穂の長さが m 単位で計測されているとどうなるでしょうか（次の解析も「間違った解析の例」です）。

```
> mydata$panicle.length <- mydata$panicle.length / 100
                                    #  100 で割って cm 単位から m 単位に変更する
> res.2 <- prcomp(mydata)           #  cm 単位のデータで主成分分析
> res.2
Standard deviations:
[1] 0.32097715 0.03220266

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

先ほどと一転し、第 1 主成分は主に 1 穂穎花数の大きさを説明する変数となっていることが分かります。つまり、計測の尺度が異なることにより、主成分分析の結果が全く変わってしまいます。

このような問題を解決するにはどうすればよいのでしょうか？ 1 つの方法は、変数をそれぞれ平均 0、分散 1 となるように基準化してから主成分分析を行うことです。このように基準化することで各変数の変動の大きさの違いに影響されずに主成分分析を行うことができます。では、実際に計算してみましょう。

```
> mydata.scaled <- scale(mydata)
                  #  関数 scale を用いて計測値を平均 0、分散 1 に基準化する
> var(mydata.scaled)       #  分散共分散行列を表示する（分散が確かに 1 になっている）
              panicle.length panicle.florets
panicle.length      1.0000000      0.4240264
panicle.florets     0.4240264      1.0000000
> res.scaled <- prcomp(mydata.scaled)        #  基準化されたデータで主成分分析
> res.scaled                         #  結果の表示
Standard deviations:
[1] 1.1933258 0.7589292

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

解析の結果、第 1 主成分は両変数がともに大きくなることを説明する変数、第 2 主成分は一方が大きくなったときに他方が小さくなることを説明する変数であることが分かります。

なお、このように基準化された変数間で計算される分散共分散行列は、基準化前の変数間で計算される相関行列に一致します。したがって、別の言い方をす

ると、分散共分散行列の固有値分解ではなく、相関行列の固有値分解を行えば同じ結果が得られます。これを、関数 eigen を用いて確認してみましょう。

```
> 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）
```

なお、関数 prcomp では、scale = T というオプションを指定すると相関行列に基づく主成分分析を行います。

```
> 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）
```

実は、2 変数の相関行列に基づく主成分分析ではいつも同じ固有ベクトルが計算されます。また、2 変数間の相関を r とすると固有値はいつも $1+r$、$1-r$ となります。以下に示した式を眺めればその仕組みを理解することができるでしょう。

2 変数間の相関行列を $\mathbf{R}=\begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}$ とすると、固有方程式を 0 とする λ は、

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

の解として求められる。すなわち、

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

固有値がλ_1のとき、固有ベクトルは、$\mathbf{Ru}_1 = \lambda_1 \mathbf{u}_1$を満たす。
すなわち、

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

を満たす。これを解くと、

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

固有値λ_2に対する固有ベクトル$\mathbf{u}_2$についても同様に求めることができ、

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

となる。

＜多変量データへの適用＞

ここまで2つの変数の例をもとに主成分分析の解説を行ってきました。しかし、実際に主成分分析を利用する場面ではもっと多数の変数からなるデータを解析する場合がほとんどです。ここでは、7つの変数からなるデータを解析しながら、主成分数の決定の仕方と、主成分の意味の解釈の仕方について説明します。

まず、7つの変数（止め葉の長さ Flag.leaf.length、止め葉の幅 Flag.leaf.width、草丈 Plant.height、穂の数 Panicle.number、穂の長さ Panicle.length、種子の長さ Seed.length、種子の幅 Seed.width）を抽出します。

```
> 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, ]
```

相関行列に基づく主成分分析を行ってみましょう。

```
> 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
```

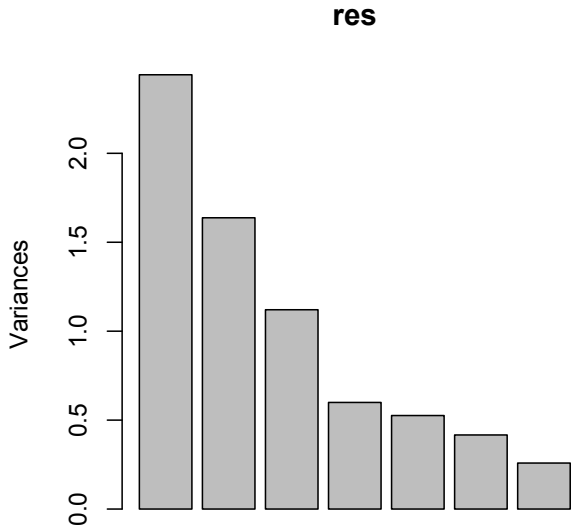


図 9. 主成分得点の分散（固有値）を表す棒グラフ
相関行列に基づく主成分分析では固有値の和は変数の数に一致する

7 つの変数からなるデータでは 7 つの主成分が計算されるが、では上位何主成分まででデータを要約すればよいのでしょうか。有効な主成分数を決めるための方法として様々なものが提案されていますが、ここでは、簡単なルールを紹介します。

1. 累積寄与率が、ある定められた割合を超える主成分数を採用する。定められた割合として 70%～90%の値が使われることが多い。
2. 寄与率が、元の変数 1 つあたりの平均説明力を超える主成分を採用する。変数の数が q の場合、寄与率が 1/q を超える主成分を採用する。
3. 相関行列の場合、上のルールでは“固有値が 1 を超える”主成分が採用される。しかし、この基準は厳しすぎる場合が多い。0.7 程度が適当であるという報告もある。
4. 固有値のグラフで、急な変化からなだらかな変化に変わる点を採用する主成分とする。

1 番目のルールに基づき定められた割合を 80%とすると、累積寄与率が 82.8%となる上位 4 主成分が選ばれます。次に、2 番目のルールに基づくと、寄与率が 1/7 = 14.3%を超える上位 3 主成分が選ばれます。これは、3 番目のルールでも同じです（ただし、固有値 0.7 以上とすると上位 5 主成分が選ばれます)。最後に 4 番目のルールでは、第 4 主成分までは固有値が急に減少し、それ以降は減少がなだらかになります。したがって、上位 4 主成分が選ばれます。以上を併せると、上位 3 または 4 主成分が適切な主成分数と考えられます。

では、上位 4 主成分までの散布図を描いてみましょう。なお、遺伝構造との関係を見るために分集団 Sub.population 毎に色づけしてみましょう。

```
subpop <- alldata$Sub.population[!missing]
          #  分集団データを抜き出す。欠測をもつサンプル分を抜いておくのを忘れずに
> op <- par(mfrow = c(1,2))          #  グラフを 1 行 2 列の配置にする
> plot(res$x[,1:2], col = as.numeric(subpop))      #  第 1、第 2 主成分の散布図
> plot(res$x[,3:4], col = as.numeric(subpop))      #  第 3、第 4 主成分の散布図
> par(op)
> require(rgl)              #  OpenGL を用いた 3 次元グラフ描画のためのパッケージ
> plot3d(res$x, col = as.numeric(subpop))
                                    #  第 1～3 主成分の散布図
```

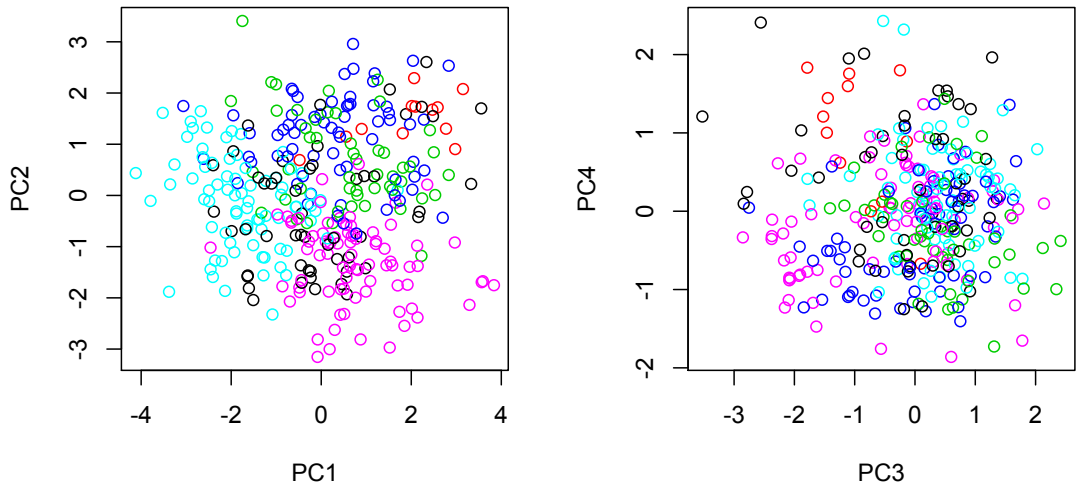


図 10. 第 1 ～第 4 主成分得点の散布図
主成分得点と遺伝的背景の違いとの間に関係があることが分かる

散布図を描いてみると、同色の点がある程度かたまって散布されることが分かります。これは、主成分得点と遺伝的背景との間に何らかの関連があることを示唆しています。

では、これら上位 4 主成分は元の変数のどのような変動を捉えた主成分なのでしょうか。まずは、固有ベクトルを見てみましょう。

```
> 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
```

固有ベクトルを見ると第 1 主成分は穂の数（panicle.number）と種子の幅（seed.width）以外は正の値になっており、種子の幅を除いた「サイズ」を説明する変数だと解釈できます。第 2 主成分は穂の数（panicle.number）に比較的大きな重みが与えられているのが分かります。

こうして数値をもとに主成分の意味を解釈していくのはなかなか難しいものです。そこで、グラフで視覚化して眺めてみましょう。まずはバイプロットを描いてみましょう。

```
> op <- par(mfrow = c(1,2))          #  グラフを 1 行 2 列で配置
> biplot(res, choices = 1:2)         #  第 1, 2 主成分でバイプロット
> biplot(res, choices = 3:4)         #  第 3, 4 主成分でバイプロット
> par(op)                            #  グラフの設定を元に戻す
```

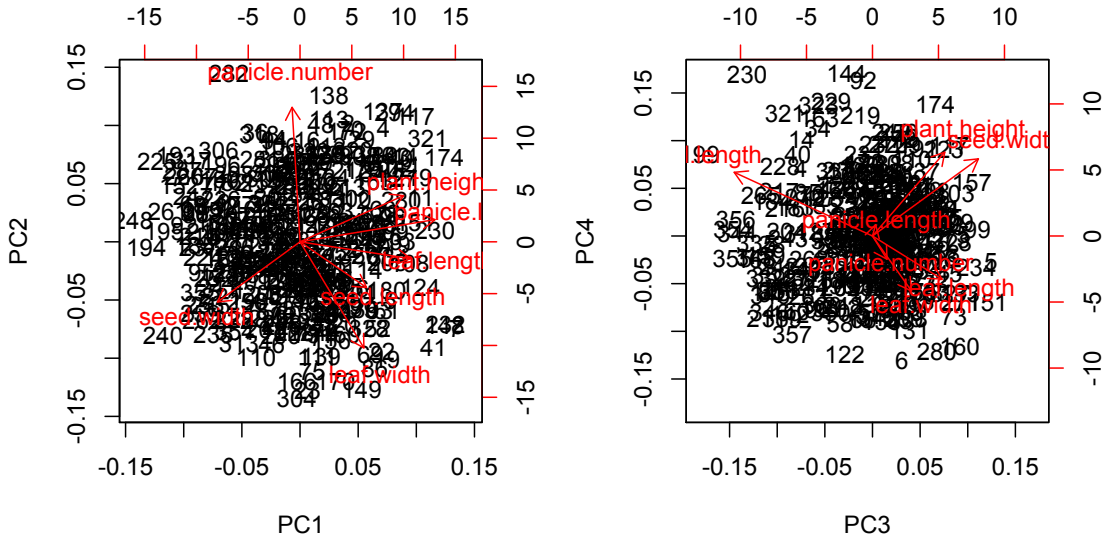


図 11. 第 1〜4 主成分のバイプロットの結果

変数名がついている矢印の大きさと向きが、各主成分とその変数の関係の強さを表しています。例えば、左図から、第 1 主成分が大きな値をとっているサンプル（例えば、174, 230 など）では、草丈（plant.height）、穂の長さ（panicle.length）、止め葉の長さ（panicle.length）などの長さを計測した変数が大きな値をとっていると読み取ることができます。また、逆に長さを計測した変数のうち種子の幅だけは他の変数と逆向きで、種子の幅（seed.width）が大きいものほど第 1 主成分が小さな値をとっていると考えられます。また、穂の数（panicle.number）は、第 1 主成分方向には向いておらず、この主成分にはほとんど関与していないことが分かります。第 3、第 4 主成分についても同様に読み取っていくことができます。

元の変数と主成分の関係を表す統計量として因子負荷量（factor loadings）とよばれるものがあります。因子負荷量とは、元の変数の値と主成分得点の間の相関係数です。この相関係数の絶対値が 1 に近いものは両者の関係が強いことを示しており、0 に近ければ関係が弱いまたは無いことを示しています。

では、因子負荷量を計算してみましょう。

```
> factor.loadings <- cor(mydata, res$x[,1:4]) # 元の変数と主成分得点の相関を計算
> 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
```

では、この結果を図示してみましょう。因子負荷量は半径 1 の円内に収まるので次のようなグラフを描くことができます。

```
> theta <- 2 * pi * (0:100 / 100)  #  0〜2πまで100刻みの値をthetaに代入
> x <- cos(theta)                  #  xはthetaの余弦
> y <- sin(theta)                  #  yはthetaの正弦
> op <- par(mfrow = c(1,2))        #  グラフを1行2列で配置
> plot(factor.loadings[,1:2], xlim = c(-1,1), ylim = c(-1,1), pch = 4)
                                   #  第1、第2主成分の因子負荷量を散布
> text(factor.loadings[,1:2], rownames(factor.loadings), col = "red")
                                   #  変数名を赤色で示す
> lines(x, y, col = "gray")        #  半径1の円を灰色で描く
> abline(v = 0, h = 0)             #  x軸とy軸を描く
> plot(factor.loadings[,3:4], xlim = c(-1,1), ylim = c(-1,1), pch = 4)
                                   #  第3、第4主成分の因子負荷量を散布
> text(factor.loadings[,3:4], rownames(factor.loadings), col = "red")

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

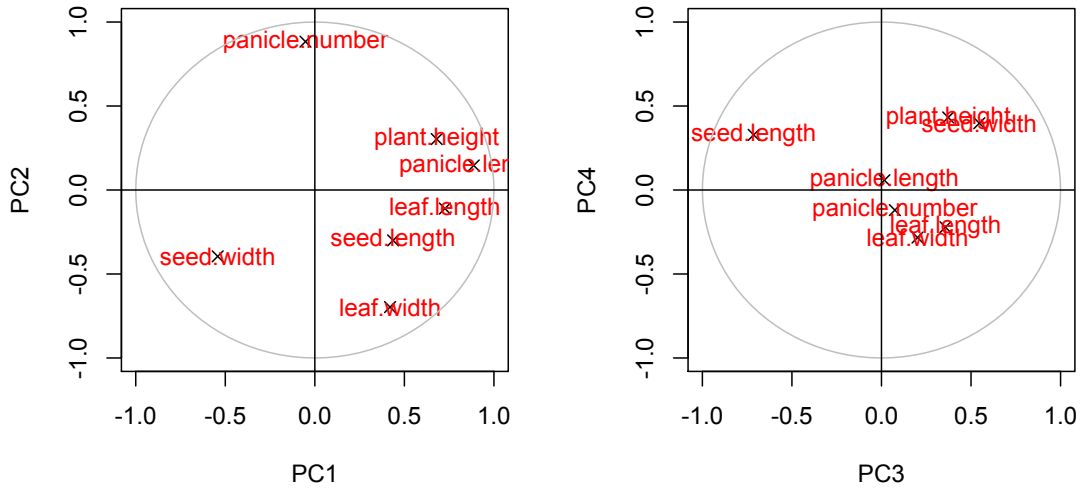


図 12. 第 1〜4 主成分の因子負荷量
円に近い位置に散布された変数ほどそれら主成分との関係が強い
逆に原点近くに散布された変数はそれら主成分との関係が弱い

なお、相関行列に基づく主成分分析では次のようにして因子負荷量を計算することもできる。

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

最後に、マーカー遺伝子型データの主成分分析を行ってみましょう。まず、マーカー遺伝子型データを抜き出します。

```
> mydata <- alldata[, 50:ncol(alldata)]
        # alldata の 50 列目から最後までがマーカーデータ。関数 ncol は列数を返す
> dim(mydata)                          #  関数 dim はデータの行数と列数を返す
[1]  374 1311
> head(mydata)[,1:10]
                  #  関数 head を使って最初の 10 マーカーの 6 サンプル分のデータを表示
（結果は省略）
```

このデータは、変数の数がとても多い（1311 変数）データです。変数の数がサンプルの数よりも多いのも特徴です。では、このデータを用いて分散共分散行列に基づく主成分分析を行ってみましょう。

```
> res.pca <- prcomp(mydata)          #  分散共分散行列に基づく主成分分析
> summary(res.pca)                   #  結果の表示
（結果は省略）
> plot(res.pca)                      #  主成分得点の分散（固有値）の棒グラフの描画
```

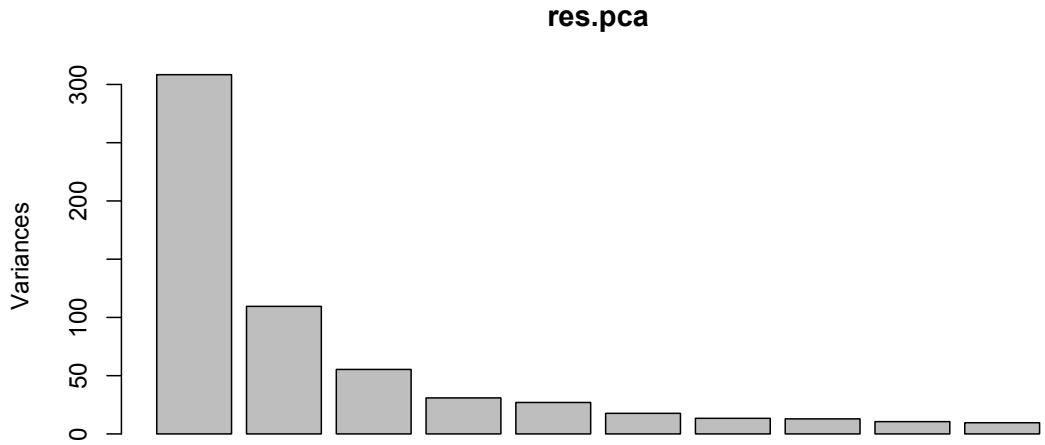


図 13. 主成分得点の分散（固有値）の棒グラフ

先ほど示した 4 番目のルールに従うと、上位 4 主成分を用いるのがよいと判断できます（それ以外のルールでは主成分の数が多くなりすぎます）。

では、上位 4 主成分の散布図を描いてみましょう。

```
> subpop <- alldata$Sub.population          #  分集団の情報
> op <- par(mfrow = c(1,2))                 #  グラフを 1 行 2 列で表示
> plot(res.pca$x[,1:2], col = as.numeric(subpop))    #  第 1、2 主成分の散布図
> plot(res.pca$x[,3:4], col = as.numeric(subpop))    #  第 3、4 主成分の散布図
> legend(-10, 20, levels(subpop), col = 1:nlevels(subpop), pch = 1, cex = 0.5)
                                            #  凡例をつける
> par(op)
```

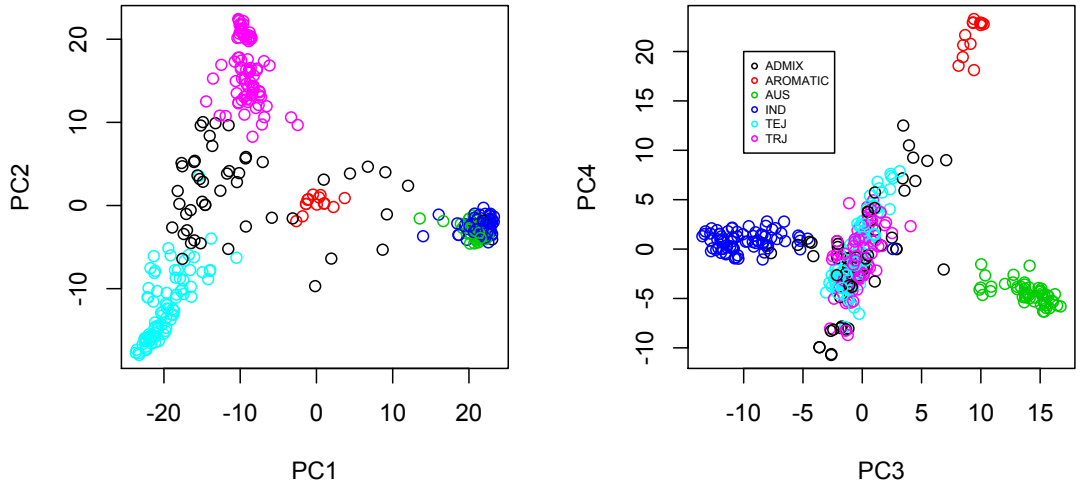


図 13. マーカー遺伝子型データをもとにした主成分分析の結果
第 1〜第 4 主成分の得点をもとに品種を 5 群（＋1 群）に分けることができる

3 次元の散布図も描いてみましょう。

```
> plot3d(res.pca$x[,1:3], type = "s", radius = 0.5, col = as.numeric(subpop))
> plot3d(res.pca$x[,2:4], type = "s", radius = 0.5, col = as.numeric(subpop))
```

最後に、alldata に含まれる PC1-4 と今回計算した第 1〜第 4 主成分の比較をしてみましょう。

```
> 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
```

計算方法が少し異なるため完全に一致はしていませんが、alldata に含まれる PC1-4（1〜4 行目）と今回計算した第 1〜第 4 主成分（1〜4 列目）が、ほぼ同じ情報を提供していることが分かります。

＜多次元尺度構成法＞

サンプルのもつ特性を直接計測することができないものの、サンプル間の特性の違いを評価することはできる場合があります。言い換えると、サンプルのもつ特性を多次元空間内の点としては位置付けることはできないものの、互いの距離関係は計測できる場合があります。

例えば、集団遺伝学では、遺伝マーカーの多型をもとに集団間の遺伝距離が計算されます。この場合、集団間の距離は分かるものの、その集団の遺伝的特性が多変量データとして計測されているわけではありません。別の例としては、ある対象に対して人間が感じる印象の相同性などが挙げられます。例えば、多数の品種のパンジーの花の写真を 100 名の被験者に見せ、任意の数のグループに分類してもらうという試験をしたとします。その結果、ある 2 品種が常に同じグループに分類された場合は、両者は被験者に似ていると判断されたことを意味します。逆に、ある 2 品種常に異なるグループに分類された場合には、両者は似ていないと判断されたことを意味します。このようなデータを用いて、全ての品種組合せについて、被験者 100 人中何人が別のグループに分類したかを集計すれば、その値を品種間の距離とすることができます。この場合も、各品種の花を人間の印象という多次元空間の中に位置付けることはできないものの、品種間の花の印象の違いを距離として計測することはできます。

ここでは、このように距離として計測されたデータをもとに、サンプルのもつ変動を低次元の変数で要約する方法について概説します。このような方法には様々なものがありますが、ここでは古典的多次元尺度構成法（classical multidimensional scaling）を紹介します。

ここでは、マーカー遺伝子型データをもとに、イネ品種・系統間の距離を計算し、この距離行列をもとに多次元尺度構成法による解析を行ってみます。

まずは、マーカー遺伝子型データを抜き出して、それをもとに距離行列を計算してみましょう。

```
> 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
```

では、計算された距離行列をもとに多次元尺度構成法による解析を行ってみましょう。

```
> 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
```

主成分分析と同様に固有値が計算されます。また、固有値をもとに寄与率、累積寄与率が計算できます。

```
> 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
```

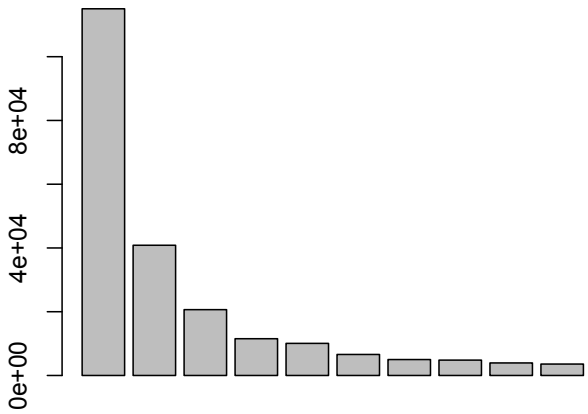


図 14. 多次元尺度構成法で計算された上位 10 固有値

図 14 の固有値の棒グラフをもとに主成分分析のルールに従うと、データに含まれる変動は 4 つの次元で表すのがよいと考えられます。

では、4 次元空間での布置を表す散布図を描いてみましょう。

```
> subpop <- alldata$Sub.population            #  分集団の情報を抜き出す
> op <- par(mfrow = c(1,2))                   #  グラフを 1 行 2 列で表す
> plot(res.mds$points[,1:2], col = as.numeric(subpop))
                    #  cmdscale 関数で求められる座標値は points として保存されている
                    #  第 1 軸、第 2 軸での散布図
> plot(res.mds$points[,3:4], col = as.numeric(subpop))
                    #  第 3 軸、第 4 軸での散布図
> legend(5, -10, levels(subpop), col = 1:nlevels(subpop), pch = 1, cex = 0.5)
                                                #  凡例をつける
> par(op)
```

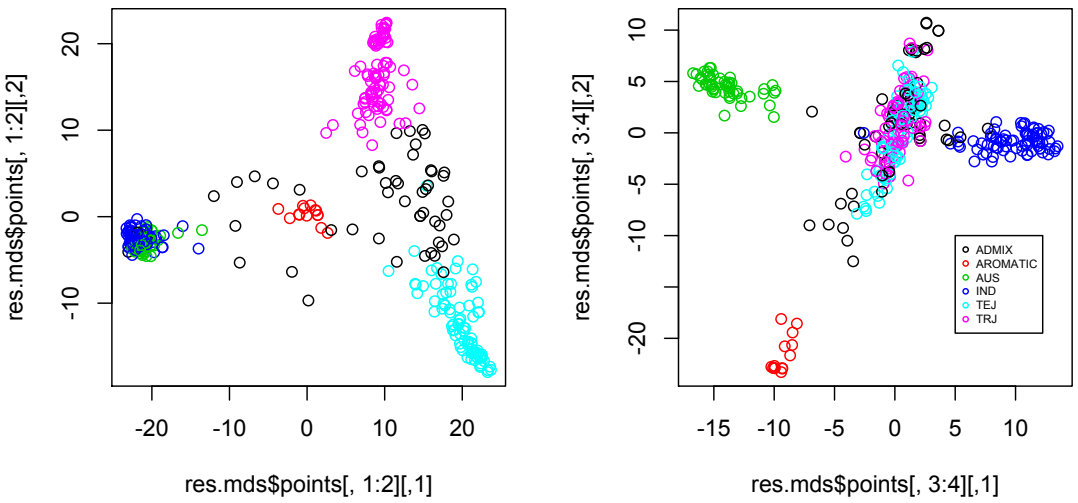


図 15. 多次元尺度構成法で求められた品種・系統の 4 次元空間での布置

主成分分析と同じく 3 次元の散布図も描いてみましょう。

```
> plot3d(res.mds$points[,1:3], type = "s", radius = 0.5,
                                    col = as.numeric(subpop))
> plot3d(res.mds$points[,2:4], type = "s", radius = 0.5,
                                    col = as.numeric(subpop))
```

多次元尺度構成法の結果を眺めると主成分分析の結果に酷似しているように見えます。実際にどのくらい一致しているのか多次元尺度構成法で求められた座標値と主成分得点の間の相関を計算してみましょう。

```
> 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
```

上位 4 主成分（行）と多次元尺度の上位 4 次元は互いに相関が 1 または−1 であることが分かります。この結果を少し違った目から見ると、元の変数の値が与えられなくても、その値に基づいたユークリッド距離行列さえあれば主成分分析と同じ解析ができると解釈することができます。

距離行列さえ与えられれば、主成分分析と同じように、少ない変数でサンプルのもつ変動を代表できるということは、多次元尺度法の大変有用な面の一つです。少し具体的な例を用いて、この点について説明します。第 2 回の講義において遺伝的背景を表す主成分得点（PC1-4）と草丈（Plant.height）の間に強い関連があることを示しました。多次元尺度構成法を用いれば、「品種・系統間の距離関係」から、この主成分得点と同じような変動をもつ変数を計算ができ、それをもとに、遺伝的関係が草丈に影響しているかどうかを確認できます。言い換えると、サンプル間の距離関係が、サンプルに見られる別の特徴の変異に関連しているのかどうかを、多次元尺度構成法と別の解析（例えば、重回帰分析）を組合せて用いることにより確認することができるのです。

最後に上述した点について具体的な計算を行って確認してみましょう。

```
> mydata <- data.frame(    #  草丈のデータと多次元尺度構成法で求められた座標値を結合
          plant.height = alldata$Plant.height,
          res.mds$points[,1:4]
          )
> mydata <- na.omit(mydata)          #  欠測値の除去
> model <- lm(plant.height ~ ., data = mydata)
                    #  重回帰分析。草丈を多次元尺度構成法で求められた座標値に回帰する
> anova(model)                       #  分散分析
（結果は省略）
> plot(mydata$plant.height, predict(model)) #  観察値とあてはめ値の対散布
```

＜多次元尺度構成法の定式化＞

ここでは、q 次元空間内に分布する n 個のサンプルの配置を「サンプル間のユークリッド距離」から逆に求める問題を考えてみましょう。

今、i 番目のサンプルの q 次元空間内での配置が列ベクトルで表される座標

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

で表せるとします。このとき、n 個のサンプルの列ベクトルを束ねて転置した n ×q の行列 $\mathbf{X} = (\mathbf{x}_1,...,\mathbf{x}_n)^{\mathrm{T}}$ を用いると、n 個のサンプルの列ベクトルの内積要素とする行列は、

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

と表せます。ここで、内積行列 B の(i, j)要素は、i 番目の j 番目のサンプルの内積、すなわち、

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

です。

このとき、i 番目のサンプルと j 番目のサンプル間のユークリッド距離 d_{ij} は、

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

と表せます。

なお、x の重心は原点にあるとすると、

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

となるため、b_{ij} を i や j について和をとると

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

となります。したがって、以下が成り立ちます。

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

以上の式を用いると、以下の関係が成り立ちます。

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

ここで、

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

です。

式(6)は、距離行列 **D**=$\{d_{ij}\}$が与えられれば、そこから逆に内積行列 **B** を計算できることを意味しています。計算の手順は、まず、距離行列の要素をそれぞれ 2 乗して d_{ij}^2 を計算し、次に、その行平均 $d_{i\cdot}^2$、列平均 $d_{\cdot j}^2$、総平均 $d_{\cdot\cdot}^2$ を計算します。最後に、式(6)にしたがって b_{ij} を計算すれば内積行列 **B** が得られます。

内積行列 **B** が得られれば、$\mathbf{B} = \mathbf{X}\mathbf{X}^{\mathrm{T}}$ を満たす **X** を求めればよいことになります。**X** を求めるには、以下に示すように、行列 **B** のスペクトル分解を行います。

行列 **B** は対称行列で、q 個の固有値と固有ベクトルをもちます。すなわち、

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

今、これらの式を横に束ねると、

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

と表せます。$\mathbf{U} = (\mathbf{u}_1,...,\mathbf{u}_q)$，$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1,...,\lambda_q)$（$\mathrm{diag}(\lambda_1,...,\lambda_q)$ は $\lambda_1,...,\lambda_q$ を対角要

素とする対角行列）とすると、上式は

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

と表せます。なお、固有ベクトルは、単位ベクトル（$\mathbf{u}_i^{\mathrm{T}}\mathbf{u}_i = 1$）で、互いに直交している（$\mathbf{u}_i^{\mathrm{T}}\mathbf{u}_j = 0$）ので、行列 **U** は、

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

を満たします。すなわち、行列 **U** の逆行列は単に **U** の転置行列となっているのです。

式(7)、(8)より、行列 **B** は、

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

と表すことができます。

今、$\mathbf{u}_i^* = \lambda_i^{1/2}\mathbf{u}_i$ とすると、上式は、

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

と表せます。ここで、$\mathbf{U}^* = (\mathbf{u}_1^*,...,\mathbf{u}_q^*)$ です。

したがって、$\mathbf{B} = \mathbf{XX}^{\mathrm{T}}$ を満たす **X** は、

$$\mathbf{X} = \mathbf{U\Lambda}^{1/2} \tag{7}$$

として求められます。ここで、$\mathbf{\Lambda}^{1/2} = \mathrm{diag}(\lambda_1^{1/2},\ldots,\lambda_q^{1/2})$ です。

このように、古典的多次元尺度構成法は、結局、距離行列 **D** から求められる内積行列 **B** の固有値問題となります。そして、その固有値問題を解くことで、n 個のサンプルの q 次元空間での布置を求めることができるのです。

では、上述した手順で関数 cmdscale を使わずに古典的多次元尺度構成法を用いた解析を行ってみましょう。

まず、データの準備をします。先ほどと同じくマーカー遺伝子型データを用います。関数 dist を使ってユークリッド距離を計算し距離行列 **D** を準備します。

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

まず、距離行列 **D** の各要素を 2 乗します。次に、2 乗した要素の行平均、列平均、総平均を求めます。行平均、列平均は関数 apply を使うと簡単に計算できます。最後に、式(6)にしたがって内積行列 **B** を計算します。R のコードは以下のとおりです。

```
> D2 <- as.matrix(D^2)                  #  D の各要素を 2 乗した行列を D2 とする
> D2i. <- apply(D2, 1, mean)            #  関数 apply で行方向（1）に平均（mean）を計算
> D2.j <- apply(D2, 2, mean)            #  関数 apply で列方向（2）に平均（mean）を計算
> D2.. <- mean(D2)                      #  総平均の計算
> B <- - 0.5 * (sweep(sweep(D2, 1, D2i.), 2, D2.j) + D2..)
                              #  式(6)のとおり
                              #  関数 sweep は行（1）または列（2）方向に引き算をする
```

内積行列 **B** の固有値分解を行います。また、式(7)にしたがって座標値を計算します。

```
> eig <- eigen(B)                       #  固有値分解
> eval <- eig$values[1:10]              #  固有値、上位 10 個を採用
> evec <- eig$vectors[,1:10]            #  固有ベクトル、やはり上位 10 個を採用
> points <- evec * rep(sqrt(eval), each = nrow(evec))
                                        #  式（7）に従った計算
```

関数 cmdscale を用いて計算した結果と比較してみましょう。結果が一致していることが分かるでしょう。

```
> head(points)
（結果は省略）
> head(res.mds$points)
（結果は省略）
```

最後に図を描いてみましょう。図 15 と同じ図が描かれるはずです。確認してみましょう

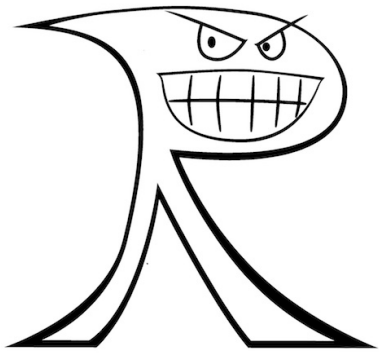
```
> subpop <- alldata$Sub.population
> op <- par(mfrow = c(1,2))
> plot(points[,1:2], col = as.numeric(subpop))
> plot(points[,3:4], col = as.numeric(subpop))
> legend(5, -10, levels(subpop), col = 1:nlevels(subpop), pch = 1, cex = 0.5)
> par(op)
```

＜レポート課題＞

主成分分析を用いて、イネの表現型データの解析を行ってみて下さい（講義で解析したのとは異なる変数の組合せで解析してみてください)。

提出方法：

- レポートは「pdf ファイル」として作成し、メール添付で提出する。
- メールは、「report@iu.a.u-tokyo.ac.jp 宛」に送る。
- レポートの最初に、「所属、学生番号、名前を忘れず」に。
- 提出期限は、5 月 6 日



K.W.