データ解析

Ｒによる多変量解析入門

主成分分析

lec20040108 下平英寿 shimo@is.titech.ac.jp

概要

- 主成分分析

  データ行列を低次元に射影

  少数個の合成変量を用いてわかりやすく表現

  「情報圧縮」に相当

- データ行列とプロット，誤差最小の１次元射影
- 主成分分析（主成分得点と主成分負荷，バイプロット）
- 特異値分解
- そのほか

データ行列とプロット

データ行列

$$\boldsymbol{X} = \underbrace{\left.\begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ x_{n1} & \cdots & x_{np} \end{bmatrix}\right\} n}_{p} = \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{x}^{(n)} \end{bmatrix} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p]$$

$\boldsymbol{x}^{(i)}$ は行ベクトル，$\boldsymbol{x}_j$ は列ベクトル

各列の平均を引き去って「中心化」してあるものと仮定して議論を進める

$$\boldsymbol{X} \leftarrow \boldsymbol{X} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n'\boldsymbol{X}$$

```
> ### データ行列
> a <- c("A05201","A06102","A06202","F01503","A06205",
+        "A06301","A06302","A06304","A06601","A06602")
```

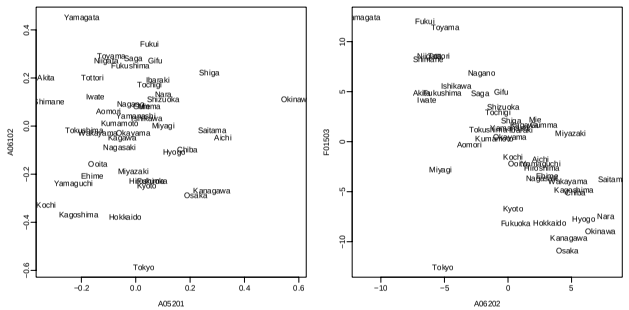


```
> na <- paste(seq(along=a),a,X2000$item[a])
> na
 [1] "1 A05201 Rate of natural increase  "
 [2] "2 A06102 Members per private household  "
 [3] "3 A06202 Ratio of family nuclei households  "
 [4] "4 F01503 Ratio of dual-income households  "
 [5] "5 A06205 Ratio of one-person households  "
 [6] "6 A06301 Ratio of households with members65 years old and over "
 [7] "7 A06302 Ratio of aged-couple households  "
 [8] "8 A06304 Ratio of aged-single person households  "
 [9] "9 A06601 Rate of marriages (per 1,000 persons)  "
[10] "10 A06602 Rate of divorces (per 1,000 persons)  "
> jna <- paste(seq(along=a),a,X2000$jitem[a])
> jna
 [1] "1 A05201 自 然 増 加 率 "
 [2] "2 A06102 一 般 世 帯 の 平 均 人 員 "
 [3] "3 A06202 核 家 族 世 帯 割 合 "
 [4] "4 F01503 共 働 き 世 帯 割 合 "
 [5] "5 A06205 単 独 世 帯 割 合 "
```

```
 [6] "6 A06301 65 歳以上の親族のいる世帯割合  "
 [7] "7 A06302 高齢夫婦のみの世帯の割合  "
 [8] "8 A06304 高 齢 単 身 世 帯 の 割 合 "
 [9] "9 A06601 婚　 姻　 率 （人口千人当たり） "
[10] "10 A06602 離　 婚　 率 （人口千人当たり） "
> x <- X2000$x[,a]
> dim(x)
[1] 47 10
> m0 <- apply(x,2,mean) # 平均ベクトル
> m0
     A05201      A06102      A06202      F01503      A06205      A06301
 0.07957447  2.79680851 57.25978723 34.63319149 24.88893617 36.86638298
     A06302      A06304      A06601      A06602
 8.46042553  6.81085106  5.63787234  1.84404255
> x0 <- sweep(x,2,m0) # 中心化
```

成分のプロット

```
> myplot(x0[,1:2]) # プロット (x1,x2)
> myplot(x0[,3:4]) # プロット (x3,x4)
```



誤差最小の１次元射影

１次元への射影と誤差

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{v}, \quad \|\boldsymbol{v}\| = 1$$

$$y_i = \boldsymbol{x}^{(i)}\boldsymbol{v} = \boldsymbol{v}'(\boldsymbol{x}^{(i)\prime}), \quad i = 1, \ldots, n$$

$$\begin{aligned} \text{誤差} &= \sum_{i=1}^{n} \|\boldsymbol{x}^{(i)} - y_i\boldsymbol{v}'\|^2 \\ &= \sum_{i=1}^{n} \|\boldsymbol{x}^{(i)}(\boldsymbol{I}_p - \boldsymbol{v}\boldsymbol{v}')\|^2 \\ &= \mathrm{tr}(\boldsymbol{X}(\boldsymbol{I}_p - \boldsymbol{v}\boldsymbol{v}')^2\boldsymbol{X}') \\ &= \mathrm{tr}(\boldsymbol{X}\boldsymbol{X}' - \boldsymbol{X}\boldsymbol{v}\boldsymbol{v}'\boldsymbol{X}') \\ &= \mathrm{tr}(\boldsymbol{X}'\boldsymbol{X}) - \boldsymbol{v}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{v} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{p} x_{ij}^2 - \sum_{i=1}^{n} y_i^2 \end{aligned}$$

誤差最小の１次元射影

$$v'X'Xv \to \text{最大}, \quad \text{ただし} \|v\| = 1$$

ラグランジュの未定乗数法

$$f(v, \lambda) = v'X'Xv - \lambda(v'v - 1)$$

$$\frac{\partial f}{\partial v} = 2X'Xv - 2\lambda v, \quad \frac{\partial f}{\partial \lambda} = v'v - 1$$

したがって$X'X$の固有ベクトルをv，その固有値をλ とすれば$v'X'Xv$の極値をとる．

最大固有値とその固有ベクトルをλとvにすれば
　$v'X'Xv$は最大値λをとる
　誤差は最小値 $\sum_{i=1}^{n}\sum_{j=1}^{p} x_{ij}^2 - \lambda$をとる．

分散共分散行列と主成分

$$\Sigma = \frac{1}{n-1}X'X$$

の固有値と規格化した固有ベクトルを

$$\lambda_1 \geq \lambda_2 \geq \cdots \lambda_p \geq 0, \quad v_1, v_2, \ldots, v_p$$

とする．射影$y = Xv$の成分の分散

$$\sigma_y^2 = \frac{1}{n-1}\|y\|^2$$

を最大にするには$v = v_1$とすればよい．最大値は$\sigma_y^2 = \lambda_1$である．

$y_1 = Xv_1$を第１主成分と呼ぶ．$y_2 = Xv_2$を第２主成分と呼ぶ．
同様に，$y_j = Xv_j$を第j主成分 ($j = 1, \ldots, p$)と呼ぶ．

第j主成分の分散$\frac{1}{n-1}\|y_j\|^2$はλ_jである．

主成分分析

主成分分析
principal component analysis (PCA)

主成分 (principal component) $y_j = Xv_j$
v_1に直交するベクトルのうちで「誤差」を最小にするのはv_2
v_1, v_2に直交するベクトルのうちで「誤差」を最小にするのはv_3
$v_1, \ldots, v_{r-1}$に直交するベクトルのうちで「誤差」を最小にするのはv_r

データ行列Xを数個の主成分$y_1, \ldots, y_r$で表現する．
y_jの寄与率と$y_1, \ldots, y_j$の累積寄与率

$$\text{寄与率} = \frac{\lambda_j}{\lambda_1 + \cdots + \lambda_p}, \quad \text{累積寄与率} = \frac{\lambda_1 + \cdots + \lambda_j}{\lambda_1 + \cdots + \lambda_p}$$

分散共分散行列と固有値固有ベクトルの計算

```
> ### 共分散行列と固有ベクトルの計算
> V0 <- (t(x0) %*% x0)/(nrow(x)-1)
> V0[1:3,1:3] # 左上3x3の部分だけ表示

               A05201        A06102     A06202
A05201  0.0324172063 -0.0002253006  0.4004043
A06102 -0.0002253006  0.0486787234 -0.4910355
A06202  0.4004042553 -0.4910354764 20.3899847
> var(x)[1:3,1:3] # これでも同じ
            A05201        A06102     A06202
A05201  0.0324172063 -0.0002253006  0.4004043
A06102 -0.0002253006  0.0486787234 -0.4910355
A06202  0.4004042553 -0.4910354764 20.3899847
> e0 <- eigen(V0,sym=T) #  対称行列の固有値と固有ベクトルの計算
> names(e0)
[1] "values"  "vectors"
> e0$values # 固有値を並べたベクトル
```



```
 [1] 8.732516e+01 1.565085e+01 1.310680e+01 2.861857e+00 2.358101e-01
 [6] 6.444274e-02 4.545187e-02 1.793557e-02 4.103978e-03 2.339494e-04
> names(e0$val) <- paste("PC",seq(ncol(e0$vec))) # 名前をPC1, PC2,...とつける
> dim(e0$vectors) # 固有ベクトルを横に並べた行列
[1] 10 10
> dimnames(e0$vec)[[2]] <- paste("PC",seq(ncol(e0$vec))) # 列の名前を修正
> e0$vec[,1:4] # 最初の４個の固有ベクトル

               PC 1         PC 2         PC 3         PC 4
A05201  0.01136942 -0.024796630 -0.02136824 -0.002535740
A06102 -0.01795377 -0.029224479 -0.01326229 -0.034233876
A06202  0.37199395 -0.490506842  0.56363509  0.356567377
F01503 -0.63202407 -0.251940268 -0.26249289  0.678383071
A06205  0.27540588  0.719576916 -0.15221985  0.445404972
A06301 -0.61839763  0.280689420  0.52690380 -0.286673558
A06302 -0.03000842  0.166503771  0.38267299  0.287893034
A06304  0.01690031  0.263061228  0.39828453  0.218185429
A06601  0.04037257 -0.030253849 -0.08243454  0.018647494
```

```
A06602  0.02518684  0.006534314  0.01083280 -0.008683317
> t(e0$vec[,1:3]) %*% e0$vec[,1:3] # 正規直交

              PC 1          PC 2          PC 3
PC 1  1.000000e-00  6.609312e-17 -8.861135e-17
PC 2  6.609312e-17  1.000000e+00 -1.428347e-16
PC 3 -8.861135e-17 -1.428347e-16  1.000000e+00
```

主成分の計算

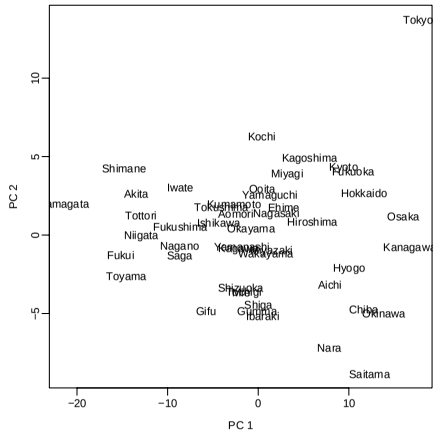
```
> ### 主成分の計算
> y0 <- x0 %*% e0$vec # 主成分
> dim(y0)
[1] 47 10
> y0[,1:2] # 第１主成分と第２主成分

               PC 1         PC 2
Hokkaido   11.65835480  2.688557372
Aomori     -2.50270706  1.370962196
Iwate      -8.60116971  3.051127469
Miyagi      3.18132115  3.937354045
Akita     -13.45275717  2.638906005
Yamagata  -21.48675175  2.002452843
Fukushima  -8.63518224  0.543435206
Ibaraki     0.48904733 -5.140466526
Tochigi    -1.62162510 -3.606495417
Gumma      -0.12489052 -4.839825710
Saitama    12.27243555 -8.838409099
```



```
Chiba      11.63842123 -4.730669153
Tokyo      17.61916595 13.741588298
Kanagawa   16.70577057 -0.760774838
Niigata   -12.92952550  0.006651134
Toyama    -14.55950001 -2.589386345
Ishikawa   -4.39214027  0.808954554
Fukui     -15.20451343 -1.259903172
Yamanashi  -1.83226293 -0.735432315
Nagano     -8.68914543 -0.666043009
Gifu       -5.78347299 -4.838657125
Shizuoka   -1.93751492 -3.347542267
Aichi       7.89827961 -3.148484022
Mie        -1.89766762 -3.641903045
Shiga       0.01741338 -4.419381359
Kyoto       9.41130511  4.370893885
Osaka      16.01430929  1.199400939
Hyogo       9.99959854 -2.078840380
Nara        7.86141793 -7.176390104
Wakayama    0.79757369 -1.151550437
```

```
  Tottori   -12.93442428  1.265461777
  Shimane   -14.81203306  4.261607915
  Okayama    -0.80039248  0.430983015
  Hiroshima   5.98052896  0.861990417
  Yamaguchi   1.26203174  2.572904575
  Tokushima  -4.07558397  1.802271640
  Kagawa     -2.18407031 -0.810311757
  Ehime       2.82506343  1.770194999
  Kochi       0.38829954  6.290711485
  Fukuoka    10.45463116  4.062573532
  Saga       -8.67703456 -1.278133390
  Nagasaki    1.99731300  1.406091730
  Kumamoto   -2.68249847  1.981934796
  Ooita       0.43037858  2.969402221
  Miyazaki    1.42728153 -0.937610879
  Kagoshima   5.63355258  4.948605488
  Okinawa    13.85336913 -4.988807187
> myplot(y0[,1:2]) # 第１主成分と第２主成分のプロット
```




```
> e0$vec[,1:2] # 第１固有値と第２固有値の固有ベクトル

              PC 1         PC 2
  A05201  0.01136942 -0.024796630
  A06102 -0.01795377 -0.029224479
  A06202  0.37199395 -0.490506842
  F01503 -0.63202407 -0.251940268
  A06205  0.27540588  0.719576916
  A06301 -0.61839763  0.280689420
  A06302 -0.03000842  0.166503771
  A06304  0.01690031  0.263061228
  A06601  0.04037257 -0.030253849
  A06602  0.02518684  0.006534314
> var(y0[,1:3]) # 主成分は互いに直交している（無相関）
              PC 1          PC 2          PC 3
PC 1  8.732516e+01 -2.592582e-15  1.565201e-15
PC 2 -2.592582e-15  1.565085e+01 -1.110119e-15
PC 3  1.565201e-15 -1.110119e-15  1.310680e+01
```

```
> diag(var(y0)) # 各主成分の分散はV0の固有値に等しい
       PC 1         PC 2         PC 3         PC 4         PC 5         PC 6
8.732516e+01 1.565085e+01 1.310680e+01 2.861857e+00 2.358101e-01 6.444274e-02
       PC 7         PC 8         PC 9        PC 10
4.545187e-02 1.793557e-02 4.103978e-03 2.339494e-04
> e0$val
       PC 1         PC 2         PC 3         PC 4         PC 5         PC 6
8.732516e+01 1.565085e+01 1.310680e+01 2.861857e+00 2.358101e-01 6.444274e-02
       PC 7         PC 8         PC 9        PC 10
4.545187e-02 1.793557e-02 4.103978e-03 2.339494e-04
> e0$val/sum(e0$val) # 各固有値の全体に対する比を寄与率と呼ぶ
       PC 1         PC 2         PC 3         PC 4         PC 5         PC 6
7.319020e-01 1.311751e-01 1.098526e-01 2.398620e-02 1.976405e-03 5.401166e-04
       PC 7         PC 8         PC 9        PC 10
3.809477e-04 1.503241e-04 3.439684e-05 1.960810e-06
> cumsum(e0$val)/sum(e0$val) # 累積寄与率
 [1] 0.7319020 0.8630771 0.9729296 0.9969158 0.9988923 0.9994324 0.9998133
 [8] 0.9999636 0.9999980 1.0000000
```

主成分得点と主成分負荷

主成分得点

主成分を標準偏差で割って分散を１に標準化する．

$$z_j = \frac{y_j}{\sqrt{\lambda_j}}, \quad j = 1, \ldots, p$$

$$Z = [z_1, \ldots, z_p] = \begin{bmatrix} z^{(1)} \\ \vdots \\ z^{(n)} \end{bmatrix}$$

各個体$x^{(i)}$，$i = 1, \ldots, n$に対応して$z^{(i)}$を主成分得点と呼ぶ．

主成分負荷

$$B = \frac{1}{n-1} X'Z \quad \text{とおくと} \quad X = ZB'$$

$$B = [b_1, \ldots, b_p] = \begin{bmatrix} b^{(1)} \\ \vdots \\ b^{(p)} \end{bmatrix}$$

つまり$x_j = Z(b^{(j)\prime})$と考えると，x_jを目的変数，$z_1, \ldots, z_p$を説明変数とした回帰分析の回帰係数が$b^{(j)}$．

各変量x_j，$j = 1, \ldots, p$に対応して$b^{(j)}$を主成分負荷と呼ぶ．

主成分得点と主成分負荷の計算

```
> ### 主成分得点と主成分負荷
> z0 <- y0 %*% diag(1/sqrt(e0$val)) # 主成分得点
> dimnames(z0)[[2]] <- paste("PC",seq(ncol(z0))) # 列の名前を修正
> var(z0[,1:3]) # 主成分の分散を１に規格化したもの
              PC 1          PC 2          PC 3
PC 1  1.000000e+00 -8.060336e-17  5.023568e-17
PC 2 -8.060336e-17  1.000000e+00 -7.757496e-17
PC 3  5.023568e-17 -7.757496e-17  1.000000e-00
> b0 <- t(x0) %*% z0 /(nrow(x0)-1) # 主成分負荷
> x00 <- z0 %*% t(b0) # z0を説明変数，x0を目的変数とした重回帰
> sum( (x00 - x0)^2 ) # 元のx0に一致する
[1] 3.399127e-25
> z0[,1:2]

                   PC 1         PC 2
  Hokkaido    1.247577636  0.679595361
  Aomori     -0.267818352  0.346542558
```



```
  Iwate     -0.920423779  0.771243380
  Miyagi     0.340437839  0.995257744
  Akita     -1.439599265  0.667044824
  Yamagata  -2.299328801  0.506166496
  Fukushima -0.924063509  0.137365878
  Ibaraki    0.052333672 -1.299372385
  Tochigi   -0.173532479 -0.911625536
  Gumma     -0.013364718 -1.223378432
  Saitama    1.313291318 -2.234113316
  Chiba      1.245444516 -1.195786575
  Tokyo      1.885452776  3.473505815
  Kanagawa   1.787708997 -0.192303522
  Niigata   -1.383607476  0.001681229
  Toyama    -1.558033438 -0.654527579
  Ishikawa  -0.470009369  0.204482065
  Fukui     -1.627057269 -0.318469808
  Yamanashi -0.196073142 -0.185897610
  Nagano    -0.929838190 -0.168357850
  Gifu      -0.618897922 -1.223083045
```

```
Shizuoka  -0.207336312 -0.846169109
Aichi      0.845206479 -0.795852511
Mie       -0.203072194 -0.920575637
Shiga      0.001863431 -1.117101351
Kyoto      1.007117556  1.104845015
Osaka      1.713714711  0.303176463
Hyogo      1.070071698 -0.525475221
Nara       0.841261856 -1.813999387
Wakayama   0.085349530 -0.291081136
Tottori   -1.384131702  0.319874875
Shimane   -1.585057370  1.077220446
Okayama   -0.085651172  0.108940974
Hiroshima  0.639985171  0.217888112
Yamaguchi  0.135051867  0.650361429
Tokushima -0.436134215  0.455566044
Kagawa    -0.233720566 -0.204825129
Ehime      0.302314179  0.447457928
Kochi      0.041552503  1.590123532
Fukuoka    1.118765405  1.026909880
```



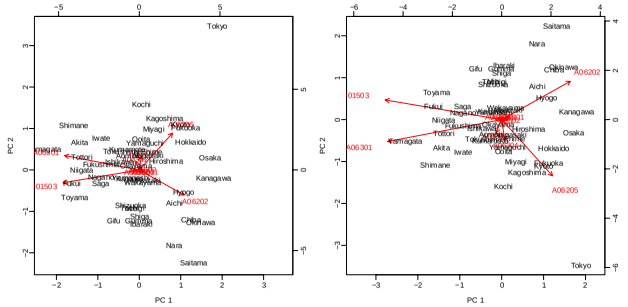
```
Saga      -0.928542187 -0.323077920
Nagasaki   0.213735392  0.355422364
Kumamoto  -0.287058094  0.500980082
Ooita      0.046055443  0.750585424
Miyazaki   0.152735488 -0.237002941
Kagoshima  0.602854719  1.250875049
Okinawa    1.482469338 -1.261036963
> b0[,1:2]

              PC 1        PC 2
A05201   0.1062448 -0.09809832
A06102  -0.1677743 -0.11561540
A06202   3.4762065 -1.94050145
F01503  -5.9061341 -0.99670466
A06205   2.5736110  2.84672900
A06301  -5.7787978  1.11043961
A06302  -0.2804225  0.65870806
A06304   0.1579299  1.04070046
A06601   0.3772734 -0.11968770
A06602   0.2353658  0.02585050
```

バイプロット

バイプロット

```
> mybiplot(z0,b0) # biplot
> mybiplot(z0,b0,scale=c(1,-1)) # PC2軸の符号を反転
```



バイプロットの性質

実際には２次元だけが表示されているが仮にp次元で表示したとすると

$$X = ZB'$$

$$x_{ij} = \boldsymbol{z}^{(i)}\boldsymbol{b}^{(j)\prime}, \quad i = 1, \ldots, n,\ j = 1, \ldots, p$$

- **個体iを表す主成分得点$\boldsymbol{z}^{(i)}$**
- **変量jを表す主成分負荷$\boldsymbol{b}^{(j)}$**
  これらの内積がデータ行列のx_{ij}成分．

特異値分解と主成分分析

特異値分解による主成分分析

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}' \\ &= d_1\boldsymbol{u}_1\boldsymbol{v}_1' + \cdots + d_p\boldsymbol{u}_p\boldsymbol{v}_p' \end{aligned}$$

$$\underset{n\times p}{\boldsymbol{U}} = \underset{n\times p}{[\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p]}, \quad \underset{p\times p}{\boldsymbol{V}} = \underset{p\times p}{[\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p]}$$

$$\boldsymbol{D} = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_p \end{bmatrix}, \quad d_1 \geq \cdots \geq d_p \geq 0$$

分散共分散行列は

$$\Sigma = \frac{1}{n-1}\boldsymbol{X}'\boldsymbol{X} = \frac{1}{n-1}\boldsymbol{V}\boldsymbol{D}^2\boldsymbol{V}'$$

固有ベクトルは$\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p$，固有値は

$$\lambda_1 = \frac{1}{n-1}d_1^2, \ldots, \lambda_p = \frac{1}{n-1}d_p^2$$

主成分$\boldsymbol{y}_j = \boldsymbol{X}\boldsymbol{v}_j,\ j = 1, \ldots, p$をならべて

$$\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p] = \boldsymbol{X}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{D}$$

主成分得点をならべた行列は，$\Lambda = \frac{1}{n-1}\boldsymbol{D}^2$とおいて

$$\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_p] = \boldsymbol{Y}\Lambda^{-1/2} = \sqrt{n-1}\,\boldsymbol{U}$$

主成分負荷をならべた行列は

$$\boldsymbol{B} = \frac{1}{n-1}\boldsymbol{X}'\boldsymbol{Z} = \frac{1}{\sqrt{n-1}}\boldsymbol{V}\boldsymbol{D}$$

特異値分解による主成分分析の計算

```
> ### 特異値分解による計算
> s0 <- mysvd(x0) # 特異値分解(svd)を呼び出してdimnames等をしているだけ
> names(s0) # 要素はd=特異値，u=左固有ベクトルを並べた行列，v=右...
[1] "d" "u" "v"
>  length(s0$d)
[1] 10
>  dim(s0$u)
[1] 47 10
>  dim(s0$v)
[1] 10 10
> n <- nrow(x0)  # n=47としておく
> sum( (s0$u %*% diag(s0$d) %*% t(s0$v) - x0)^2 )
[1] 9.6082e-28
> s0$d^2/(n-1) # e0$eigに等しい
        PC 1         PC 2         PC 3         PC 4         PC 5         PC 6
8.732516e+01 1.565085e+01 1.310680e+01 2.861857e+00 2.358101e-01 6.444274e-02
        PC 7         PC 8         PC 9        PC 10
```

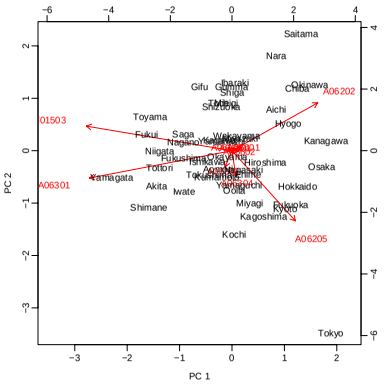
```
4.545187e-02 1.793557e-02 4.103978e-03 2.339494e-04
> s0$v[,1:2] # 符号を除き，e0$vecに等しい
              PC 1         PC 2
A05201   0.01136942  0.024796630
A06102 -0.01795377  0.029224479
A06202   0.37199395  0.490506842
F01503 -0.63202407  0.251940268
A06205   0.27540588 -0.719576916
A06301 -0.61839763 -0.280689420
A06302 -0.03000842 -0.166503771
A06304   0.01690031 -0.263061228
A06601   0.04037257  0.030253849
A06602   0.02518684 -0.006534314
> (s0$u %*% diag(s0$d))[1:10,1:2] # 符号を除き，y0に等しい

                  [,1]       [,2]
  Hokkaido   11.6583548 -2.6885574
  Aomori     -2.5027071 -1.3709622
  Iwate      -8.6011697 -3.0511275
  Miyagi      3.1813211 -3.9373540
  Akita     -13.4527572 -2.6389060
  Yamagata  -21.4867518 -2.0024528
  Fukushima  -8.6351822 -0.5434352
  Ibaraki     0.4890473  5.1404665
  Tochigi    -1.6216251  3.6064954
  Gumma      -0.1248905  4.8398257
> (sqrt(n-1)*s0$u)[1:10,1:2] # 符号を除き，z0に等しい
                PC 1       PC 2
Hokkaido   1.24757764 -0.6795954
Aomori    -0.26781835 -0.3465426
Iwate     -0.92042378 -0.7712434
Miyagi     0.34043784 -0.9952577
Akita     -1.43959926 -0.6670448
Yamagata  -2.29932880 -0.5061665
Fukushima -0.92406351 -0.1373659
Ibaraki    0.05233367  1.2993724
Tochigi   -0.17353248  0.9116255
Gumma     -0.01336472  1.2233784
```

```
> (s0$v %*% diag(s0$d)/ sqrt(n-1)) [,1:2] # 符合を除き，b0に等しい

                [,1]       [,2]
  A05201   0.1062448  0.09809832
  A06102 -0.1677743  0.11561540
  A06202   3.4762065  1.94050145
  F01503 -5.9061341  0.99670466
  A06205   2.5736110 -2.84672900
  A06301 -5.7787978 -1.11043961
  A06302 -0.2804225 -0.65870806
  A06304   0.1579299 -1.04070046
  A06601   0.3772734  0.11968770
  A06602   0.2353658 -0.02585050
> cumsum(s0$d^2)/sum(s0$d^2) # 累積寄与率
 [1] 0.7319020 0.8630771 0.9729296 0.9969158 0.9988923 0.9994324 0.9998133
 [8] 0.9999636 0.9999980 1.0000000
```

```
> mybiplot(sqrt(n-1)*s0$u,s0$v %*% diag(s0$d)/ sqrt(n-1)) # biplot
```



データ行列の標準化

データ行列の標準化

単位の異なる変量を同等に扱うのは不自然．例えばデータ行列のある列だけ長さの単位をメートルからセンチに変えれば分散は10000倍になり，固有ベクトルがその列の方向に大幅に引っ張られることになる．

データ行列の各列を分散が１になるようにあらかじめ標準化しておく．

$$\sigma_{x_1}^2 = \frac{1}{n-1}\|\boldsymbol{x}_1\|^2, \ldots, \sigma_{x_p}^2 = \frac{1}{n-1}\|\boldsymbol{x}_p\|^2$$

として

$$\boldsymbol{x}_j \leftarrow \frac{1}{\sigma_{x_j}}\boldsymbol{x}_j, \ldots, j = 1, \ldots, p$$

データ行列の標準化の計算

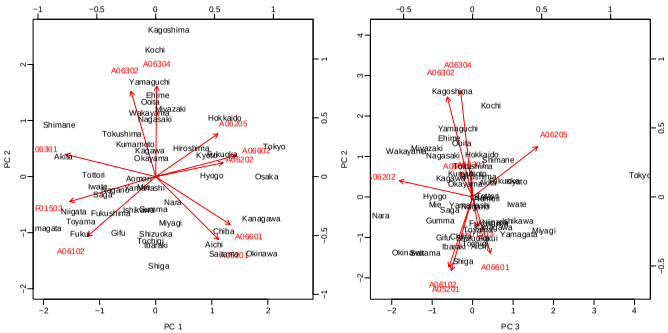
```
> ### データ行列の標準化
> v0 <- apply(x0,2,var) # 分散のベクトル
> v0
     A05201      A06102      A06202      F01503      A06205      A06301
 0.03241721  0.04867872 20.38998474 38.09756568 15.61066623 38.51346272
     A06302      A06304      A06601      A06602
 2.81379547  3.43402969  0.29180842  0.08022895
> x1 <- sweep(x0,2,sqrt(v0),"/") # 標準化
> var(x1[,1:3]) # 各成分の分散が１
             A05201       A06102      A06202
A05201   1.000000000 -0.005671593  0.4924957
A06102 -0.005671593  1.000000000 -0.4928728
A06202   0.492495698 -0.492872775  1.0000000
> cor(x0[,1:3]) # これでも同じ，つまり相関行列
             A05201       A06102      A06202
A05201   1.000000000 -0.005671593  0.4924957
A06102 -0.005671593  1.000000000 -0.4928728
A06202   0.492495698 -0.492872775  1.0000000
```


標準化してから主成分分析の計算

```
> ## 特異値分解による主成分分析
> n <- nrow(x1)
> s1 <- mysvd(x1)
> mybiplot(sqrt(n-1)*s1$u,s1$v%*%diag(s1$d)/ sqrt(n-1)) # biplot
> cumsum(s1$d^2)/sum(s1$d^2) # 累積寄与率
 [1] 0.5082010 0.8324454 0.9296597 0.9592818 0.9776542 0.9882034 0.9951517
 [8] 0.9992321 0.9997006 1.0000000
> z1 <- sqrt(n-1)*s1$u # 主成分得点
> b1 <- s1$v %*% diag(s1$d)/ sqrt(n-1) # 主成分負荷
> dimnames(b1)[[2]] <- paste("PC",seq(ncol(b1))) # 行の名前を修正
> sum( (z1 %*% t(b1) - x1)^2 ) # 確認
[1] 1.760365e-28
```



```
> mybiplot(z1,b1) # biplot (1-2)
> mybiplot(z1,b1,choi=3:2,scale=c(-1,1)) # biplot (3-2)
```



```
 [1] "1 A05201 自 然 増 加 率 "
 [2] "2 A06102 一 般 世 帯 の 平 均 人 員 "
 [3] "3 A06202 核 家 族 世 帯 割 合 "
 [4] "4 F01503 共 働 き 世 帯 割 合 "
 [5] "5 A06205 単 独 世 帯 割 合 "
 [6] "6 A06301 65歳以上の親族のいる世帯割合 "
 [7] "7 A06302 高齢夫婦のみの世帯の割合 "
 [8] "8 A06304 高 齢 単 身 世 帯 の 割 合 "
 [9] "9 A06601 婚 姻 率 （人口千人当たり） "
[10] "10 A06602 離 婚 率 （人口千人当たり） "
```

- **PC1 都市型 vs 農村型？**
  ＋離婚, ＋核家族, ＋単独, ＋結婚, ＋自然増加
  －65歳以上, －共働き, －人数
- **PC2 高齢化?**
  ＋高齢単身, ＋高齢夫婦, －人数, －自然増加, －結婚
- **PC3 核家族?**
  ＋核家族, －単独

そのほか

もうひとつのバイプロット

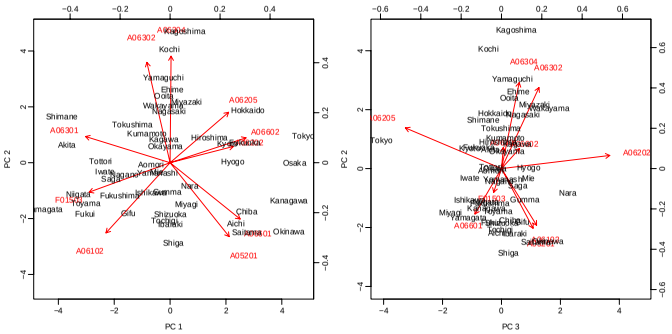
バイプロットではZ, Bを表示するほかにY, Vを表示する方法もある．

$$X = ZB' = YV'$$

```
> ## もうひとつのbiplot
> y1 <- s1$u %*% diag(s1$d) # 主成分
> dimnames(y1)[[2]] <- paste("PC",seq(ncol(y1))) # 列の名前を修正
> v1 <- s1$v # 固有ベクトル
> sum( (y1 %*% t(v1) - x1)^2 ) # 確認
[1] 1.684082e-28
> s1$d/sqrt(n-1) # この比だけz1とy1は異なる.
      PC 1       PC 2       PC 3       PC 4       PC 5       PC 6       PC 7
2.25433141 1.80067886 0.98597313 0.54426153 0.42863015 0.32479623 0.26359482
      PC 8       PC 9      PC 10
0.20200083 0.06844848 0.05471515
```



```
> mybiplot(y1,v1)
> mybiplot(y1,v1,choi=3:2)
```



r次元への射影と誤差

$$y_j = Xv_j, \quad V_r = [v_1, \ldots, v_r], \quad V_r'V_r = I_r$$

$$y_{ij} = x^{(i)}v_j, \quad i = 1, \ldots, n,\ j = 1, \ldots, r$$

$$\begin{aligned}
\text{誤差} &= \sum_{i=1}^{n} \|x^{(i)} - \sum_{j=1}^{r} y_{ij}v_j'\|^2 \\
&= \sum_{i=1}^{n} \|x^{(i)}(I_p - V_rV_r')\|^2 \\
&= \mathrm{tr}(X(I_p - V_rV_r')^2X') \\
&= \mathrm{tr}(XX' - XV_rV_r'X') \\
&= \mathrm{tr}(X'X) - \mathrm{tr}(V_r'X'XV_r) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{p} x_{ij}^2 - \sum_{i=1}^{n}\sum_{j=1}^{r} y_{ij}^2
\end{aligned}$$

誤差最小のr次元射影

$$\mathrm{tr}(V_r'X'XV_r) \to \text{最大}, \quad \text{ただし} V_r'V_r = I_r$$

ラグランジュの未定乗数を$r \times r$対称行列Λとして，

$$\begin{aligned}
f(V_r, \Lambda) &= \sum_{i=1}^{r} v_i'X'Xv_i - \sum_{i=1}^{r} \lambda_{ii}(v_i'v_i - 1) - 2\sum_{i=1}^{r}\sum_{j>i}^{r} \lambda_{ij}v_i'v_j \\
&= \mathrm{tr}\left(V_r'X'XV_r - \Lambda(V_r'V_r - I_r)\right)
\end{aligned}$$

$$\frac{\partial f}{\partial v_i} = 2X'Xv_i - 2\sum_{j=1}^{r}\lambda_{ij}v_j, \qquad \frac{\partial f}{\partial V_r} = 2X'XV_r - 2V_r\Lambda$$

ところでΛの固有ベクトルを並べた$r \times r$直交行列Q を用いると

$$Q'\Lambda Q = \mathrm{diag}(\lambda_1, \ldots, \lambda_r)$$

そこで$V_r \leftarrow V_rQ$と置き換えると

$$X'Xv_i = \lambda_i v_i, \quad i = 1, \ldots, r$$

$X'X$の固有ベクトルの任意の組み合わせを$v_1, \ldots, v_r$に用いれば

$$\text{誤差} = \mathrm{tr}(X'X) - (\lambda_1 + \cdots + \lambda_r)$$

は極値を取る．$\lambda_1, \ldots, \lambda_r$は対応する固有値である．誤差最小にするには固有値の大きいほうからr個の固有ベクトルを$v_1, \ldots, v_r$として用いる．

以上の議論より，最初のr個の主成分を用いる主成分分析は誤差最小のr次元射影を求めていることが分かる．

演習問題

5-1．中心化や標準化のされていない生のデータ行列xを入力として主成分分析を行う関数mypca2を作成する．中心化や標準化もmypca2で行い，出力は特異値分解の結果u, v, dと，主成分y，主成分得点z，主成分負荷b，分散共分散行列の固有値を並べたベクトルlambdaとする．

```
mypca2 <- function(x, cor=T) {  # cor=Tのとき標準化を行う(デフォルト)
 # まずxの中心化
 # 次にcor=Tのときのみxの標準化
 s <- mysvd(x)
 # ここでy,z,b,lambdaの計算
 # 最後に出力
 list(u=s$u,v=s$v,d=s$d,y=y,z=z,b=b,lambda=lambda)
}
```

5-2．実数パラメタαと任意の正則行列Aをつかって次のようにZ_αとB_αを定義する

$$Z_\alpha = ZA^\alpha, \qquad B_\alpha = BA^{-\alpha}$$

これでも$X = Z_\alpha B_\alpha'$なので「バイプロットの性質」を満たしている．特に

$$A = \frac{1}{\sqrt{n-1}}D$$

として，mypca2の出力から(Z_α, B_α)のバイプロットを作図する関数mybiplot2を作成する．なお$\alpha = 0$が(Z, B)，$\alpha = 1$が(Y, V)に対応する．

```
mybiplot2 <- function(p,alpha=0,...) {
 # p$u, p$v, p$dとalphaからzaとbaを計算
 za <- ????
 ba <- ????
 mybiplot(za,ba,...)
}
p <- mypca2(x) # デフォルトでは中心化と標準化を行う.
mybiplot2(p) # デフォルトではalpha=0
mybiplot2(p,1) # alpha=1の作図
mybiplot2(p,0.5) # alpha=0.5の作図
```

5-3．以上で作成したmypca2とmybiplot2を用いて，主成分分析の数値例を示せ．また主成分の解釈も行え．