

Properties of matrices 行列の特徴

ryamada

2016年12月22日

- 1 似ている行列 (相似) Matrix similarity
- 2 相似行列に共通する特徴量
 - 2.1 Rでの計算
- 3 Exercise 1
 - 3.1 Exercise 1-1
 - 3.2 Exercise 1-2
- 4 対称行列
- 5 Exercise 2
 - 5.1 Exercise 2-1
- 6 距離行列 分散共分散行列 対称行列 固有値が正 正定値行列
 - 6.1 距離行列
 - 6.2 分散共分散行列
 - 6.3 固有値がすべて正 正定値
- 7 多変量正規分布
 - 7.1 Rでの多変量正規乱数の発生
- 8 Exercise 3
 - 8.1 Exercise 3-1
 - 8.2 Exercise 3-3
- 9 多次元の任意な分布
 - 9.1 Exercise 4
 - 9.2 Exercise 4-1
 - 9.3 Exercise 4-2
 - 9.4 Exercise 4-3
 - 9.5 Exercise 4-4
 - 9.6 Exercise 4-5
 - 9.7 Exercise 4-6
 - 9.8 Exercise 4-7

1 似ている行列 (相似) Matrix similarity

When we use matrices for statistical analyses, we change bases so that “better” axes should grab data sets in simpler or better ways.

The change of bases are achieved with linear transformation. Therefore matrices that are transformed each other by linear transformation should be considered “similar”.

Among the similar matrices, the simplest one tends to represent data structure the “best” way from data interpretation standpoint.

The simplest one among the similar matrices is Jordan normal.

Or the simplest one is diagonal, if diagonalizable, and, in many cases the matrices in statistical analyses are diagonalizable.

統計解析に行列を使うとき、データセットが単純に理解できるような基底の取り直しをすることがよくある。

基底の取り直しは線形変換で行うので、線形変換で互に行きあえる行列同士は「相似(似ている)」関係にあると言う。

相互に相似な関係にある行列の中で、最も単純な行列を代表とすることがあり、それを使うことでデータ解釈が最もわかりやすくなる。

相似な行列の中で最も単純なのは、ジョルダン標準形のそれである。

対角化可能な行列なら、ジョルダン標準形は対角化行列である。実際、統計解析で現れる行列の多くは、対角化可能である。

任意の正方行列 M はジョルダン標準形 J と逆行列を持つ行列 R とを用いて、次のように表せる。

$$M = RJR^{-1}$$

$M' = R'JR'^{-1}$ となると、 M と M' は相似であると言う。

以下では、ジョルダン標準形が対角行列であるようなもの(対角化可能行列)に絞って話を進めるが、特に断らなければ、提示している性質等は対角化可能行列に限るものではない。

2 相似行列に共通する特徴量

相互に相似な行列は見かけは違うが、「本質的に同じ」なので、その「本質」に関する性質は共通である。

- 階数(Rank)
- 行列式(Determinant)
- トレース(跡) (Trace)
- 固有値(Eigen values)

2.1 Rでの計算

Rank, Determinant, Trace, Eigen valuesは次のように計算できる

```
library(Matrix)
d <- 4
M <- matrix(rnorm(d^2), d, d)
eigen(M)[[1]]
```

```
## [1] 0.9466934+0.7043068i 0.9466934-0.7043068i -0.7764073+0.0000000i
## [4] -0.4631181+0.0000000i
```

```
rankMatrix(M)[[1]]
```

```
## [1] 4
```

```
det(M)
```

```
## [1] 0.5006183
```

```
sum(diag(M)) # Trace
```

```
## [1] 0.6538614
```

3 Exercise 1

3.1 Exercise 1-1

逆行列を持つ行列Rを使って、対角行列Jを変換した行列M($M = RJR^{-1}$)がJと相似であることを、固有値が一致していることを確認して確かめよ。

そして、ランク、行列式、トレースが同一であることを確かめよ。

```
d <- 4
J <- diag(rnorm(d))
R <- matrix(rnorm(d^2), d, d)
M <- R %*% J %*% solve(R)
eigen(J)[[1]]
```

```
## [1] 0.6664244 0.4946348 -0.3298071 -2.5987765
```

```
eigen(M)[[1]]
```

```
## [1] -2.5987765 0.6664244 0.4946348 -0.3298071
```

```
rankMatrix(J)[[1]]
```

```
## [1] 4
```

```
rankMatrix(M)[[1]]
```

```
## [1] 4
```

```
det(J)
```

```
## [1] 0.28253
```

```
det(M)
```

```
## [1] 0.28253
```

```
sum(diag(J)) # Trace
```

```
## [1] -1.767524
```

```
sum(diag(M)) # Trace
```

```
## [1] -1.767524
```

3.2 Exercise 1-2

行列が相互に相似かどうかは、(すべての固有値が異なるならば)固有値のセットが等しいことで確認できる。

したがって相似な行列に共通する特徴は、固有値と紐づけて算出できるはずである。

以下を確かめよ。

$$\text{determinant} = \prod_{i=1}^n \lambda_i$$
$$\text{trace} = \sum_{i=1}^n \lambda_i$$

Exercise 1-3

d 個の異なる固有値を持つ行列を作り、それと相似な行列を複数作れ。

それらの固有値のセットが一致すること、トレース・行列式が一致することを確認せよ。

また、トレースの計算は、対角成分の和を取る方法と、固有値の和を取る方法の両方を実施せよ。行列式の計算も`det()`関数を使用する方法と固有値の積を取る方法の両方を実施せよ。

```
## [[1]]
## [1] -0.005874437
##
## [[2]]
## [1] -0.005874437
##
## [[3]]
## [1] -0.005874437
##
## [[4]]
## [1] -0.005874437
##
## [[5]]
## [1] -0.005874437
##
## [[6]]
## [1] -0.005874437
##
## [[7]]
## [1] -0.005874437
##
## [[8]]
## [1] -0.005874437
##
## [[9]]
## [1] -0.005874437
##
## [[10]]
## [1] -0.005874437
```

```
## [[1]]
## [1] -1.1270724  0.2598962  0.1753220  0.3677703
##
## [[2]]
## [1]  0.01864946 -0.08800771 -0.16304809 -0.09167764
##
## [[3]]
## [1] -0.4087273 -0.1333516 -0.1797845  0.3977795
##
## [[4]]
## [1]  0.09169251  0.10586957  0.78498649 -1.30663255
##
## [[5]]
## [1] -0.8054772  0.3793815 -0.2021903  0.3042020
##
## [[6]]
## [1] -0.23311621 -0.07557493  0.26938267 -0.28477549
##
## [[7]]
## [1]  0.3012185  0.4975727 -0.7432951 -0.3795801
##
## [[8]]
## [1] -0.4654869 -0.4615346  0.3414734  0.2614641
##
## [[9]]
## [1]  0.19422537 -0.46111316  0.01595501 -0.07315119
##
## [[10]]
## [1]  0.3756908 -0.5521447 -0.1003050 -0.0473250
```

```
## [[1]]
## [1] -0.94706534  0.33803480  0.18662199  0.09832458
##
## [[2]]
## [1] -0.94706534  0.33803480  0.18662199  0.09832458
##
## [[3]]
## [1] -0.94706534  0.33803480  0.18662199  0.09832458
##
## [[4]]
## [1] -0.94706534  0.33803480  0.18662199  0.09832458
##
## [[5]]
## [1] -0.94706534  0.33803480  0.18662199  0.09832458
##
## [[6]]
## [1] -0.94706534  0.33803480  0.18662199  0.09832458
##
## [[7]]
## [1] -0.94706534  0.33803480  0.18662199  0.09832458
##
## [[8]]
## [1] -0.94706534  0.33803480  0.18662199  0.09832458
##
## [[9]]
## [1] -0.94706534  0.33803480  0.18662199  0.09832458
##
## [[10]]
## [1] -0.94706534  0.33803480  0.18662199  0.09832458
```

4 対称行列

$M = M^T$ である正方行列を対称行列と言う。

対称行列は

$$\begin{aligned} M &= RJR^{-1} = RJR^T \\ R^T &= R^{-1} \\ RR^T &= R^T R = I \end{aligned}$$

と対角化で現れる行列 R が特別な性質(直交行列)を持つ。

確かめてみる。

```
d <- 5
n.iter <- 10
for(i in 1:n.iter){
  M <- matrix(rnorm(d^2), d, d)
  M <- M + t(M) # 対称行列化
  eigen.out <- eigen(M)
  print(round(eigen.out[[2]] %*% t(eigen.out[[2]]), 8))
}
```

[illegible]

```
## [3,] 0 0 1 0 0
## [4,] 0 0 0 1 0
## [5,] 0 0 0 0 1
```

5 Exercise 2

5.1 Exercise 2-1

対角行列と直交行列によって相似な関係にある行列を複数作り、それらが対称行列であることを確かめよ。

なお、直交行列をランダムに発生するには、`GPRotation::Rnandom.Start()`関数を用いるとよい。


```

## [1] "unit?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
## [1] "zero?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
## [1] "unit?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
## [1] "zero?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
## [1] "unit?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
## [1] "zero?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
## [1] "unit?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
## [1] "zero?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
## [1] "unit?"

```

```

##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
## [1] "zero?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
## [1] "unit?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
## [1] "zero?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
## [1] "unit?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
## [1] "zero?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
## [1] "unit?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
## [1] "zero?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
## [1] "unit?"
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    1    0    0    0    0

```

```
## [2,] 0 1 0 0 0
## [3,] 0 0 1 0 0
## [4,] 0 0 0 1 0
## [5,] 0 0 0 0 1
## [1] "zero?"
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0 0 0 0 0
## [2,] 0 0 0 0 0
## [3,] 0 0 0 0 0
## [4,] 0 0 0 0 0
## [5,] 0 0 0 0 0
## [1] "unit?"
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 1 0 0 0 0
## [2,] 0 1 0 0 0
## [3,] 0 0 1 0 0
## [4,] 0 0 0 1 0
## [5,] 0 0 0 0 1
## [1] "zero?"
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0 0 0 0 0
## [2,] 0 0 0 0 0
## [3,] 0 0 0 0 0
## [4,] 0 0 0 0 0
## [5,] 0 0 0 0 0
```

6 距離行列 分散共分散行列 対称行列 固有値が正 正定値行列

行列には相似という間柄があり、それは固有値で決まることが分かった。

また、相似な行列の間で保存される性質があることもわかった。

また、 RMR^{-1} という変換が相似な行列を作ることもわかった。

また対象行列の場合 $RMR^{-1} = RMR^T$ と書け、 R が直交行列であることもわかった。

これらは、一般的な行列の話であったが、この節では、統計解析でよく登場する特別な行列を取り上げ、その場合に固有値・相似行列の特徴がどのような性質を持つかを確認する。

6.1 距離行列

n 個の要素があって、それらの要素ペアの距離を行列とすることがある。

要素間の関連を調べるときの基本的なデータ構造であり

距離の場合、自身から自身への距離は0である。自身以外への距離は正である。

AからBへの距離とBからAへの距離は同じなので、距離行列は、次の性質を持つ。

- 対角成分が0
- 対称
- 非対角成分は正

このような行列では、以下のような特徴がある。

```
d <- 5
M <- matrix(abs(rnorm(d^2)), d, d)
diag(M) <- 0
M <- M+t(M) # 対称にする

eigen.out <- eigen(M)
trace <- sum(diag(M)) # zero
trace2 <- sum(eigen.out[[1]])
determinant <- det(M)
determinant2 <- prod(eigen.out[[1]])

eigen.out[[1]] # Negative and positive eigen values
```

```
## [1] 6.793615 1.304973 -1.348410 -3.016367 -3.733812
```

```
round(eigen.out[[2]] %*% t(eigen.out[[2]]), 6) # 固有値ベクトルが作る行列は直交
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 1    0    0    0    0
## [2,] 0    1    0    0    0
## [3,] 0    0    1    0    0
## [4,] 0    0    0    1    0
## [5,] 0    0    0    0    1
```

6.2 分散共分散行列

k 個の変数、 n 個のサンプルがあるとき、 $k \times k$ の分散共分散行列が算出できる。

対角成分は k 個の変数の分散である。トレースは k 個の変数の分散の和である。

k 個の変数の分散の和が固有値の和である。

対称行列である。これは距離行列の場合と同じ。

```
k <- 5
n <- 100
X <- matrix(0, n, k)
X[, 1] <- rnorm(n)
for(i in 2:k){
  X[, i] <- rnorm(n) + X[, i-1]
}
M <- cov(X)
M
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0.8595245 0.9007479 1.012900 1.112291 1.303707
## [2,] 0.9007479 2.1549984 2.135070 2.199831 2.140712
## [3,] 1.0129005 2.1350704 3.014396 3.147072 3.177027
## [4,] 1.1122910 2.1998308 3.147072 4.066352 4.109933
## [5,] 1.3037072 2.1407123 3.177027 4.109933 5.501328
```

```
round(M-t(M), 8) # 対称行列
```

```
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
```

```
det(M)
```

```
## [1] 0.8700331
```

```
sum(diag(M))
```

```
## [1] 15.5966
```

```
eigen.out <- eigen(M)
round(eigen.out[[2]] %*% t(eigen.out[[2]]), 8)
```

```
##      [, 1] [, 2] [, 3] [, 4] [, 5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
```

```
eigen.out[[1]]
```

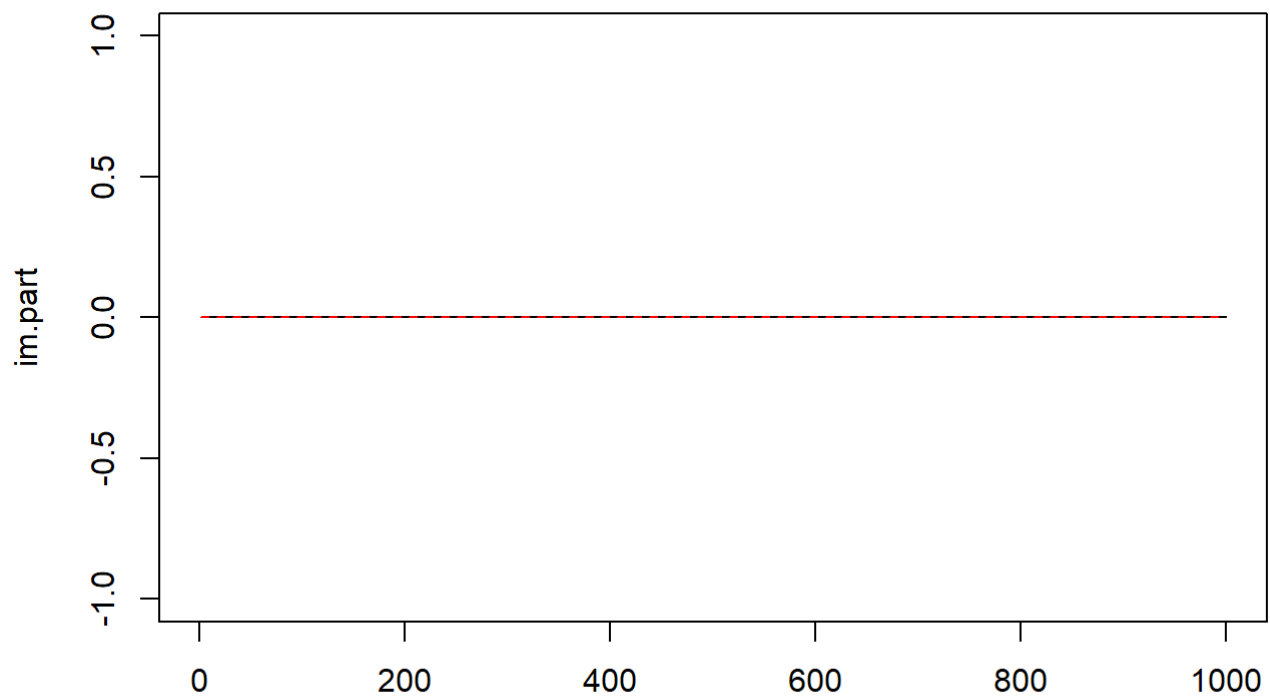
```
## [1] 13.0430299  1.3829623  0.5955992  0.3284382  0.2465694
```

分散共分散行列は、対象行列であって、対角成分はすべて正。

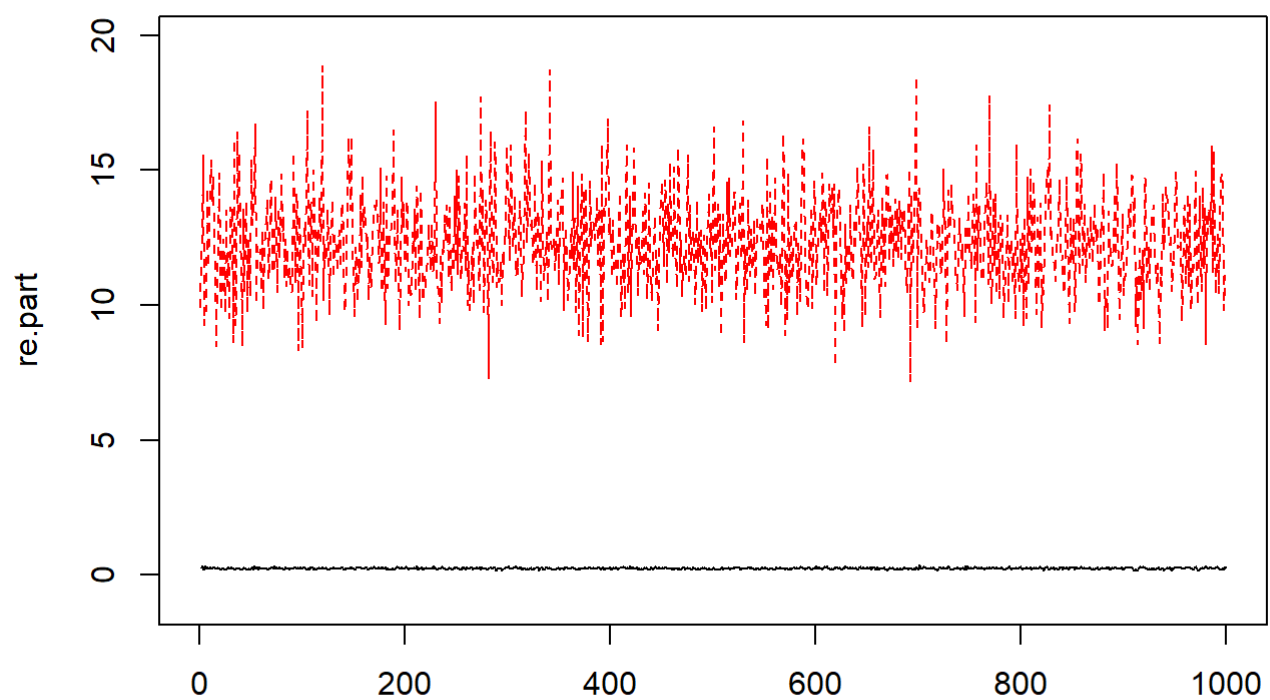
また、以下に示すように、固有値もすべて正。

```
n.iter <- 1000
im.part <- matrix(0,n.iter,2)
re.part <- matrix(0,n.iter,2)
for(i in 1:n.iter){
  d <- sample(2:50,1)
  X <- matrix(0,n,k)
  X[,1] <- rnorm(n)
  for(j in 2:k){
    X[,j] <- rnorm(n) + X[,j-1]
  }
  M <- cov(X)
  M
  eigen.out <- eigen(M)
  im.part[i,] <- range(Im(eigen.out[[1]]))
  re.part[i,] <- range(Re(eigen.out[[1]]))
}

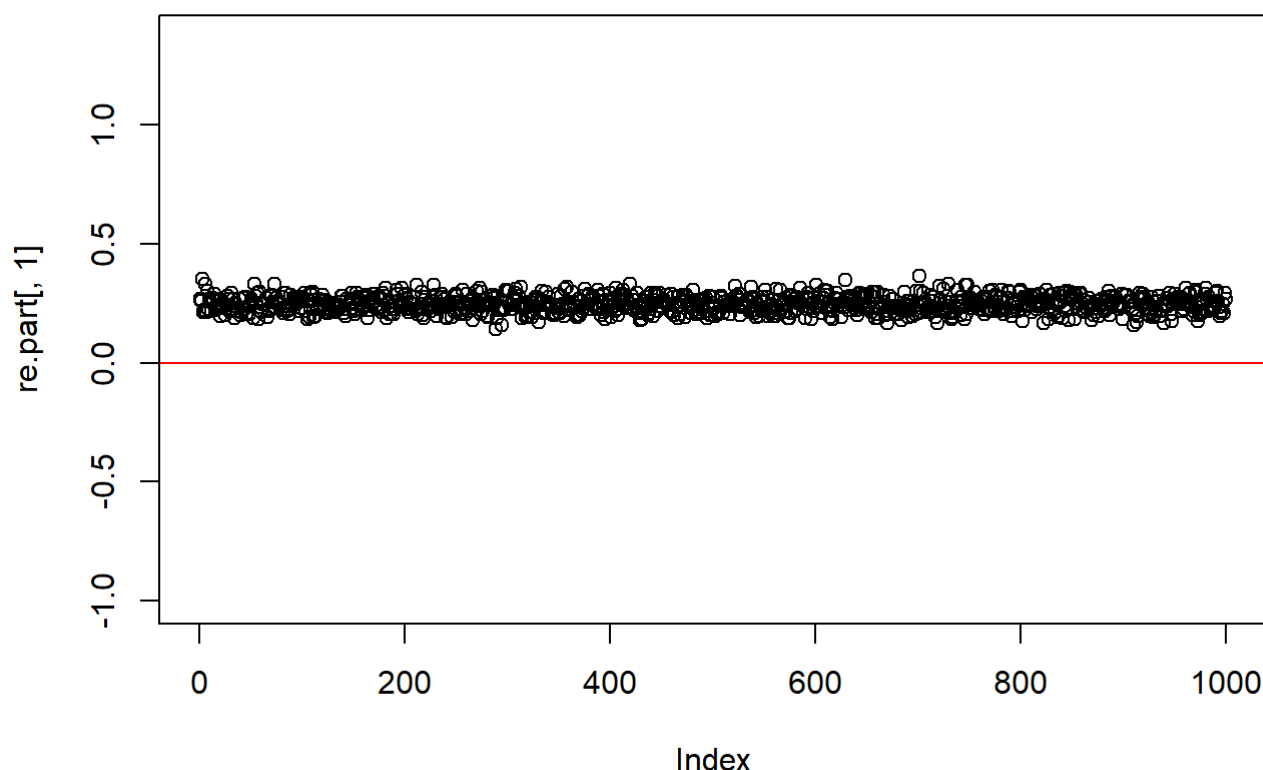
matplot(im.part,type="l")
```



```
matplot(re.part, type="l", ylim=c(-1, max(re.part)+1))
```



```
plot(re.part[, 1], ylim=c(-1, max(re.part[, 1]+1)))
abline(h=0, col=2)
```



```
min(re.part[, 1])
```

```
## [1] 0.1425431
```

6.3 固有値がすべて正 正定値

固有値がすべて正であるような対称行列は正定値行列と呼ばれている。

正定値行列 M は、任意の0でないベクトル v に対して

$$v^T M v > 0$$

となる。

正定値行列の特殊例でもある、単位行列の場合は、以下のように、「ベクトルの長さの二乗」である。

これが0ベクトルなら0となり、そうでなければ正になる、というのは、ベクトルの「長さ」を定義するのに適切な性質を持つ行列を正定値行列と言う、というようにも理解できる。

逆に言えば、統計解析で登場する行列は、この性質を満たすものが多いこととなる。

$$v^T M v = v^T v$$

このような M は、ベクトルの長さの定め方でもあり、 $v^T M v$ の値を決めるという意味では、「計量」を定めるとも言える。

対称行列の固有値分解 $M = RJR^{-1} = RJR^T$ で、 V は回転行列の性質を持つから、

対角成分がすべて正である対角行列と回転行列から作る行列は正定値行列であることになる。

```
library(GPArotation)
d <- 5
#V <- Random.Start(d)
V <- matrix(rnorm(d^2), d, d)
lambdas <- abs(rnorm(d))
M <- V %*% diag(lambdas) %*% solve(V)
eigen(M)[[1]]
```

```
## [1] 1.4071641 0.9562336 0.7730902 0.5352204 0.1090145
```

```
lambdas
```

```
## [1] 0.9562336 1.4071641 0.5352204 0.1090145 0.7730902
```

```
n.vec <- 1000
vecs <- matrix(rnorm(n.vec*d), nrow=d)
```

```
range(diag(t(vecs) %*% M %*% vecs)) # t(v) M vのみを取り出し、そのrangeを確認。すべて正
```

```
## [1] -28.02539 47.73647
```

7 多変量正規分布

多変量正規分布は、統計解析でよく使われる。

$$N(\mu, \Sigma)$$

とパラメタ表現される。

変数の数 n 個の多変量正規分布の場合、 μ は長さ n のベクトルであり、各変数の平均値を表す。

Σ は $n \times n$ 行列であり、正定値行列であることになっている。

正定値行列は、対角化可能で、 $\Sigma = RJR^{-1}$ と対角化したときに、 R が直交行列 ($R^{-1} = R^T$) であるようなものであった。

これを利用して、多変量正規分布からの乱数発生を試みる。

7.1 Rでの多変量正規乱数の発生

```
library(mvtnorm)
#help(rmvnorm)
sigma <- matrix(c(4, 2, 2, 3), ncol=2)
x <- rmvnorm(n=500, mean=c(1, 2), sigma=sigma)
colMeans(x)
```

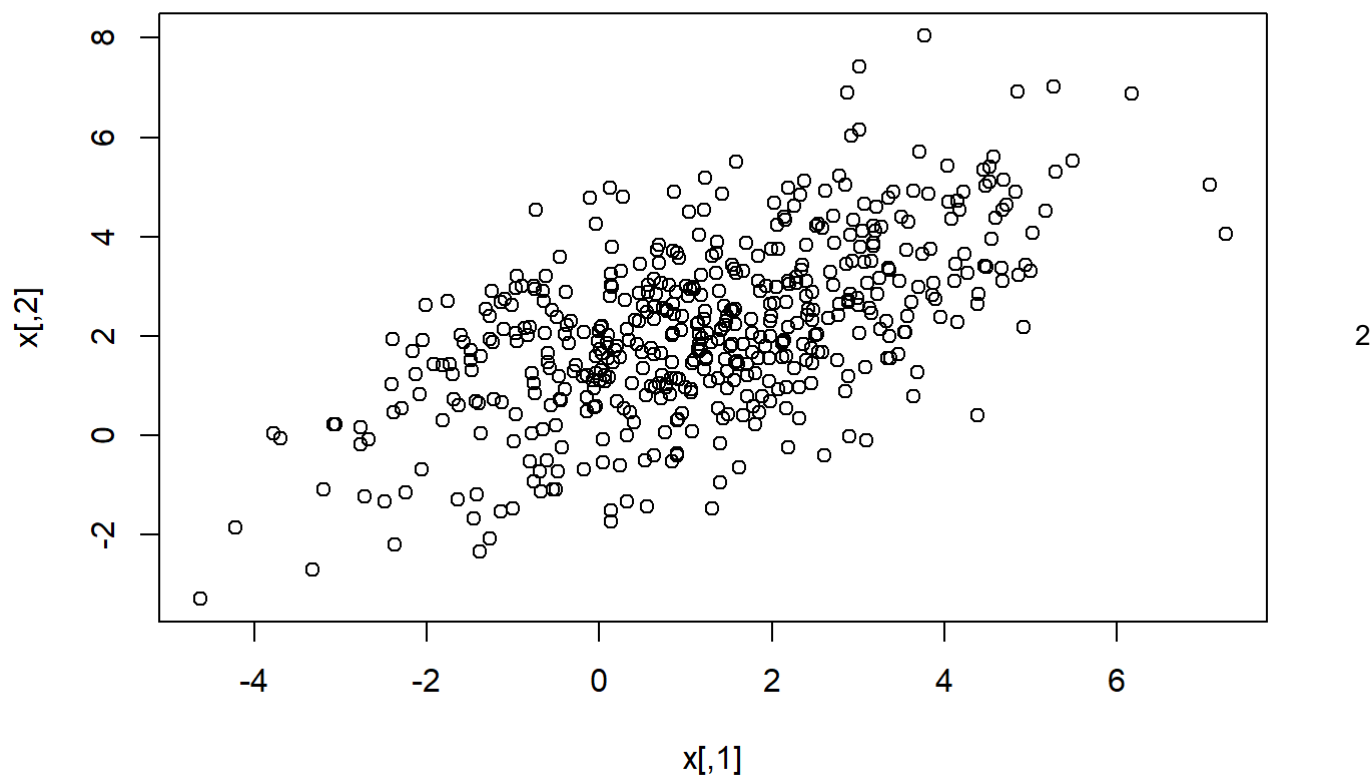
```
## [1] 1.238762 2.153684
```



```
var(x)
```

```
##          [, 1]      [, 2]
## [1, ] 3.715631 1.988485
## [2, ] 1.988485 3.016844
```

```
plot(x)
```



変数標準正規分布は、3変数の平均がともに0で、分散共分散行列が 2×2 の単位行列であるようなものである。

```
d <- 2
m <- rep(0, d)
sigma <- diag(rep(1, d))
m
```

```
## [1] 0 0
```

```
sigma
```

```
##          [, 1] [, 2]
## [1, ]      1      0
## [2, ]      0      1
```

```
n <- 10000  
x <- rmvnorm(n=n, mean=m, sigma=sigma)  
apply(x, 2, mean)
```

```
## [1] -0.017040833  0.004191151
```

```
apply(x, 2, var)
```

```
## [1] 1.0128233 0.9904683
```

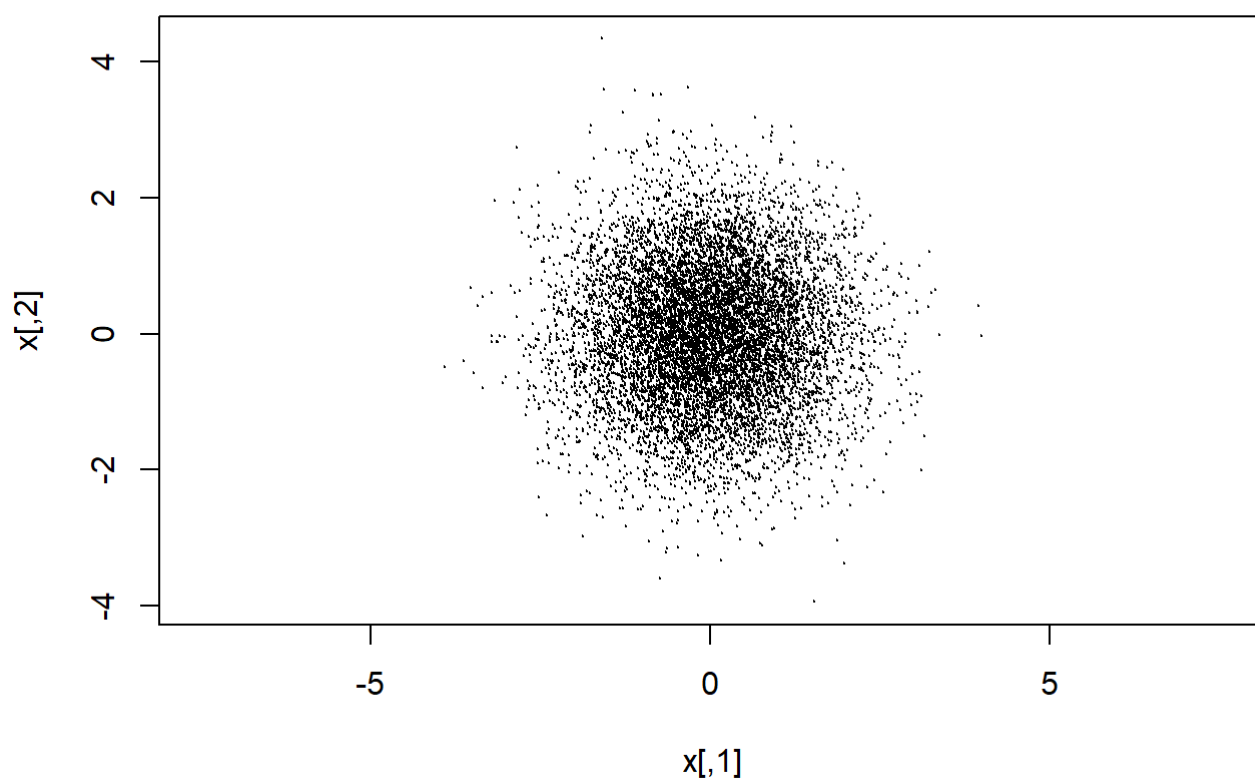
```
cov(x)
```

```
##           [, 1]      [, 2]  
## [1,]  1.012823348 -0.001449913  
## [2,] -0.001449913  0.990468274
```

```
round(cov(x), 8)
```

```
##           [, 1]      [, 2]  
## [1,]  1.01282335 -0.00144991  
## [2,] -0.00144991  0.99046827
```

```
plot(x, asp=TRUE, pch=20, cex=0.1)
```



8 Exercise 3

8.1 Exercise 3-1

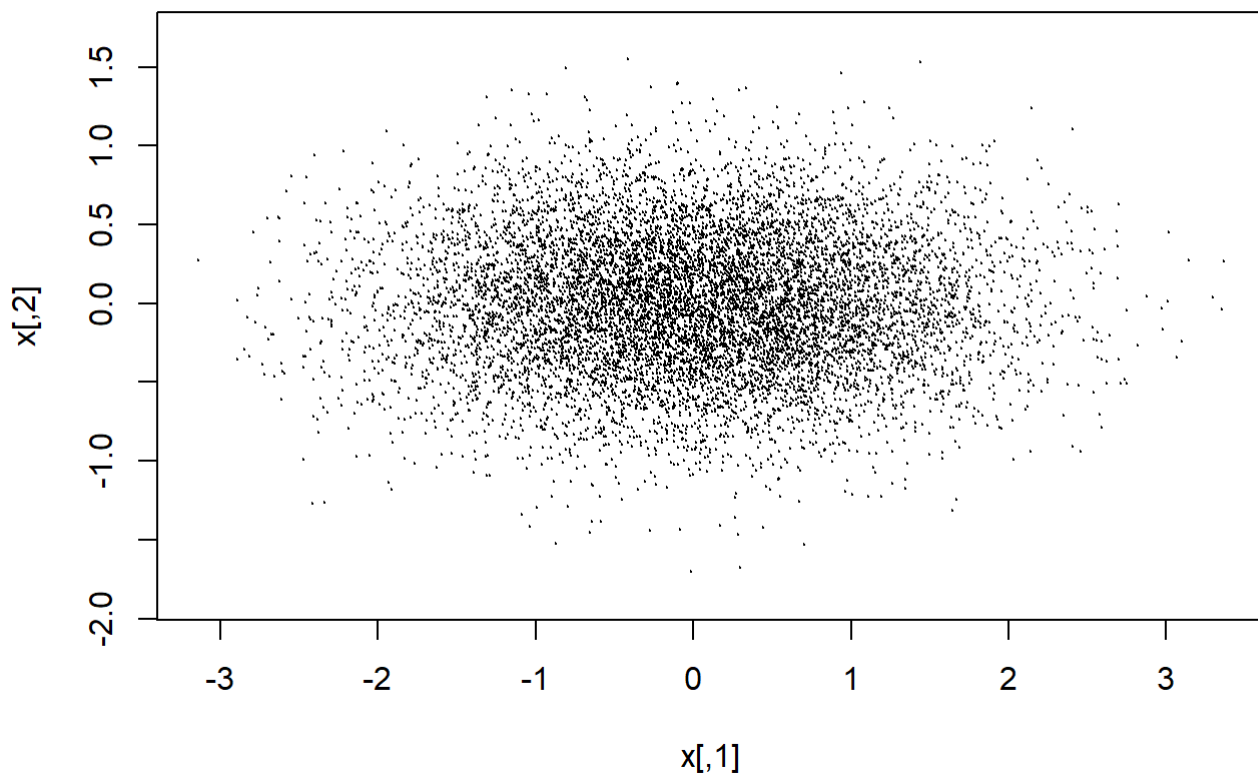
2変量正規分布において、各変数の分散を(1,1)からずらして、乱点を発生せよ。なお、分散共分散行列は対角行列のままとする。

```
## [1] 0.001465525 -0.001657303
```

```
## [1] 0.9028532 0.1855441
```

```
##           [,1]      [,2]  
## [1,] 0.9028532 -0.00139296  
## [2,] -0.00139296 0.18554414
```

```
##           [,1]      [,2]  
## [1,] 0.8862389 0.0000000  
## [2,] 0.0000000 0.1872114
```



```
## Exercise 3-2
```

4-1で用いたSigmaと相似な行列で対称なものは正定値行列であるから、`mvtnorm::rmvnorm()`のsigma引数として適当なはずである。

そのようなsigmaを作り、乱点を発生せよ。

```
R <- Random.Start(d)
sigma3 <- R %*% sigma2 %*% t(R)
x <- rmvnorm(n=n, mean=m, sigma=sigma3)
apply(x, 2, mean)
```

```
## [1] -0.008019015 -0.003858142
```

```
apply(x, 2, var)
```

```
## [1] 0.8749017 0.1860475
```

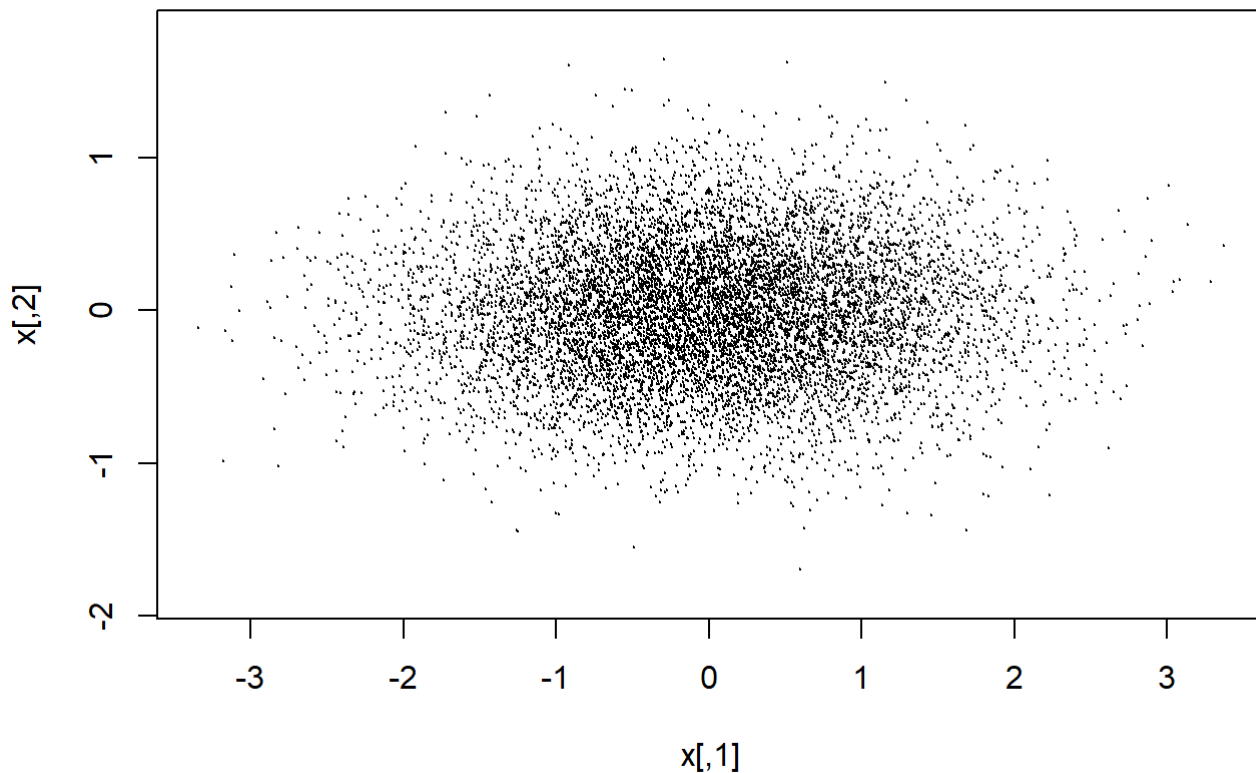
```
round(cov(x), 8)
```

```
##           [, 1]      [, 2]
## [1, ] 0.87490172 0.00907371
## [2, ] 0.00907371 0.18604747
```

```
sigma3
```

```
##           [, 1]      [, 2]
## [1, ] 0.886157366 0.007549287
## [2, ] 0.007549287 0.187292921
```

```
plot(x, asp=TRUE, pch=20, cex=0.1)
```



8.2 Exercise 3-3

分散共分散行列を RJR^{-1} にて変換すると、乱点分布が回転し、楕円の軸が斜めになった。

実際、分散共分散行列の変換に用いた直交行列 R を利用して、乱点を変換すると、楕円の軸を正すことができる。

以下のコードを読み、変換した分散共分散行列に従う 2 変量正規分布の軸が補正されていることを、コードの各行の処理にコメントを加えることで確認せよ。

```
R <- Random.Start(d)
sigma3 <- R %*% sigma2 %*% t(R)
x <- rmvnorm(n=n, mean=m, sigma=sigma3)
apply(x, 2, mean)
```

```
## [1] -0.001739469 -0.013244057
```

```
apply(x, 2, var)
```

```
## [1] 0.1910216 0.9028340
```

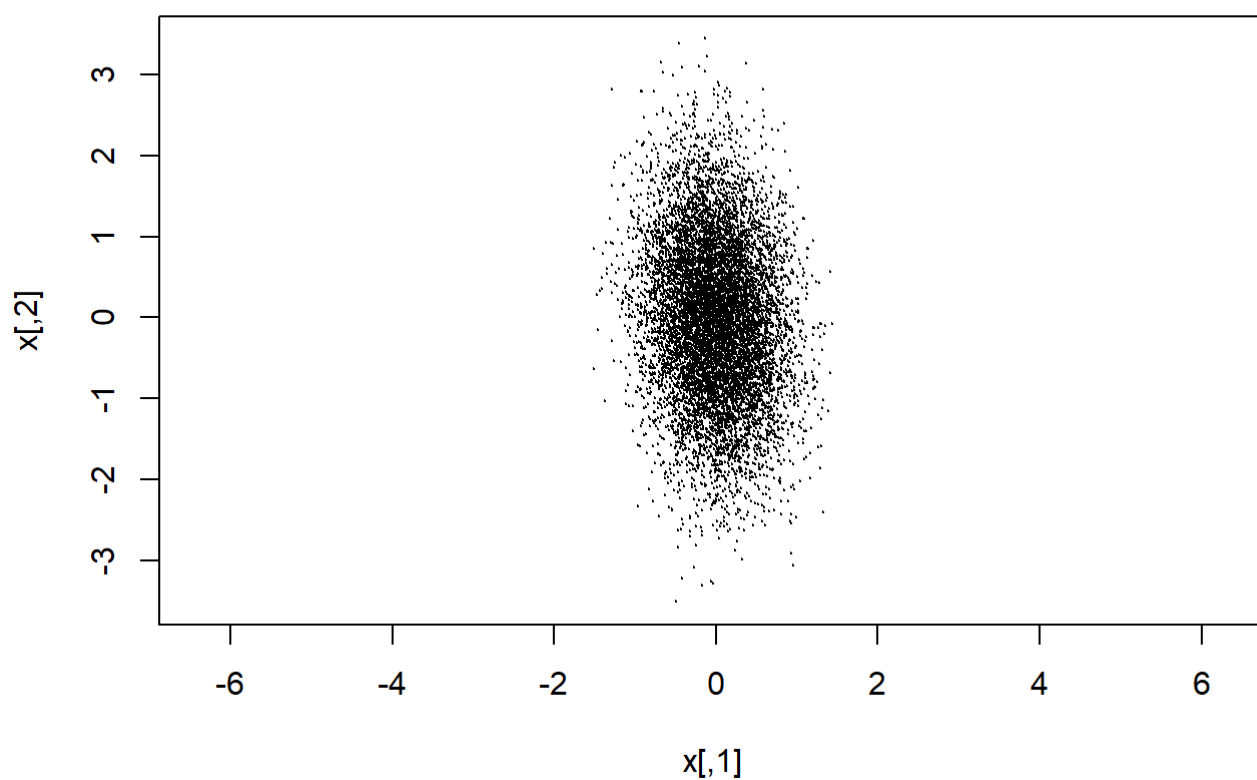
```
cov(x)
```

```
##           [, 1]      [, 2]
## [1, ]  0.19102158 -0.06315869
## [2, ] -0.06315869  0.90283404
```

```
round(cov(x), 8)
```

```
##           [, 1]      [, 2]
## [1, ]  0.19102158 -0.06315869
## [2, ] -0.06315869  0.90283404
```

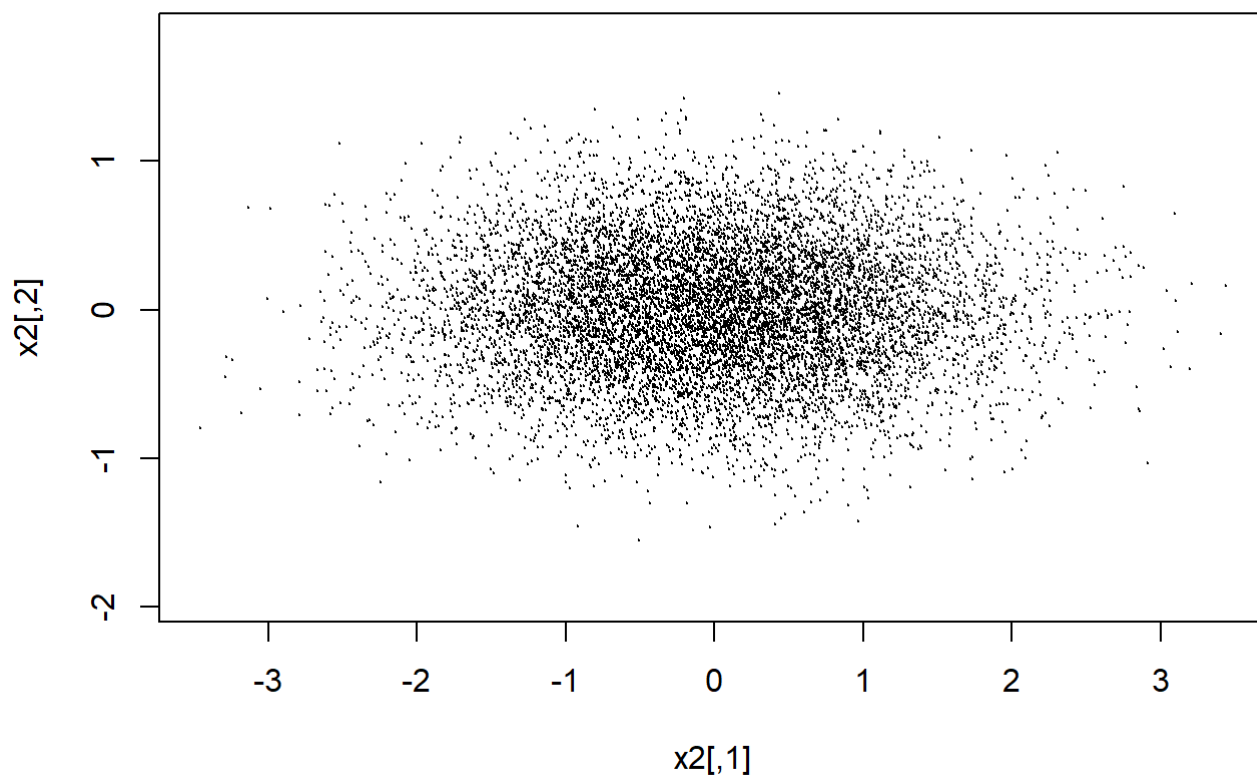
```
plot(x, asp=TRUE, pch=20, cex=0.1)
```



```
x2 <- t(t(R) %*% t(x))
cov(x2)
```

```
##           [, 1]      [, 2]
## [1, ]  0.9083943834 -0.0004266295
## [2, ] -0.0004266295  0.1854612334
```

```
plot(x2, asp=TRUE, pch=20, cex=0.1)
```



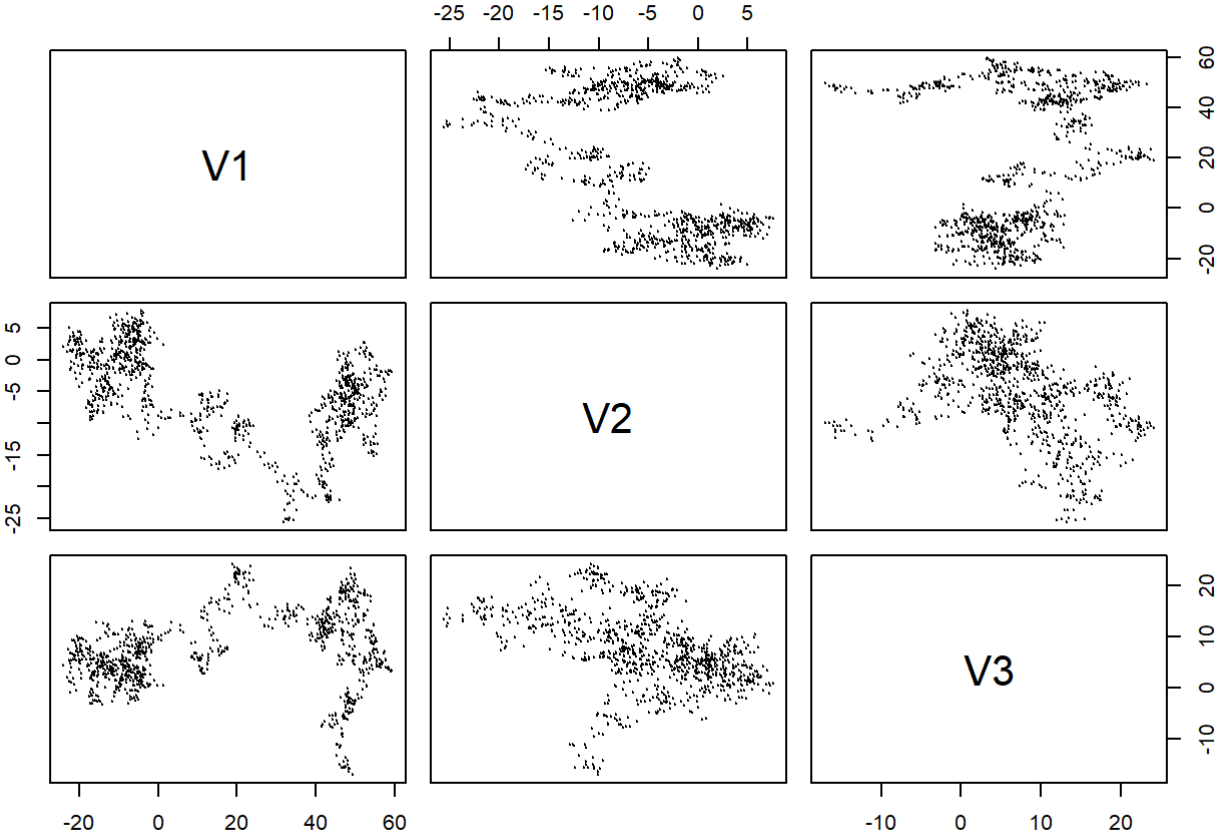
9 多次元の任意な分布

9.1 Exercise 4

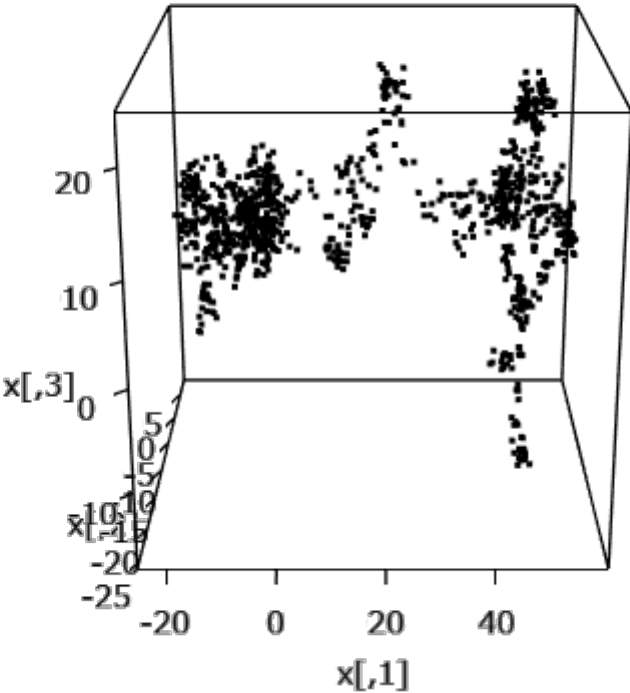
次のようにして生成される、多変量の観測データ x があったとする。

```
d <- 3
n <- 1000
x <- matrix(rnorm(d*n), ncol=d)
x <- apply(x, 2, cumsum)

plot(as.data.frame(x), pch=20, cex=0.1)
```



```
plot3d(x)
```



9.2 Exercise 4-1

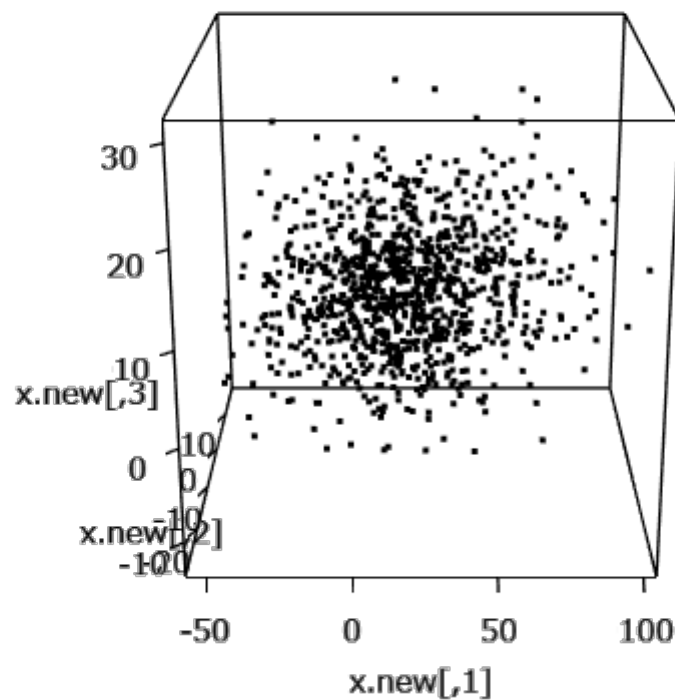
xの各変数の標本平均を求めよ。xの分散共分散行列を求めよ。

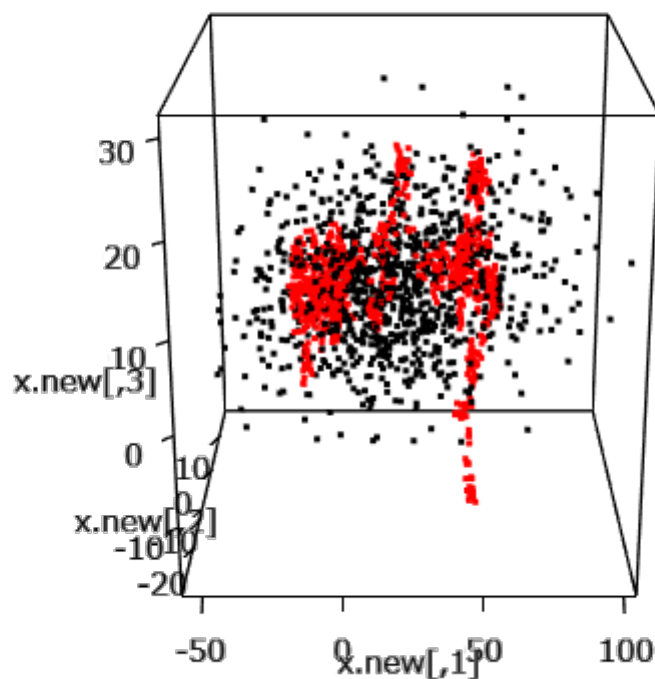
```
## [1] 16.903359 -5.214786 7.335000
```

```
##           [,1]      [,2]      [,3]  
## [1,] 756.97332 -95.89246 46.65691  
## [2,] -95.89246 47.51852 -20.85526  
## [3,] 46.65691 -20.85526 53.57443
```

9.3 Exercise 4-2

4-1 で求めた平均値ベクトルと分散共分散行列とに基づく、多変量正規乱数を発生し、ペアワイズにプロットせよ。





9.4 Exercise 4-3

回転行列を用いて、 x を回転し(軸を取り直し)、その標本平均と分散共分散行列とを求め、4-1のそれと比較せよ。

各変数の平均も分散も異なることを確かめよ。

```
## [1] "mean x"
```

```
## [1] 16.903359 -5.214786  7.335000
```

```
## [1] "mean x rotated"
```

```
## [1] -18.333692  5.367590  1.335846
```

```
## [1] "cov x"
```

```
##           [,1]      [,2]      [,3]
## [1,] 756.97332 -95.89246  46.65691
## [2,] -95.89246  47.51852 -20.85526
## [3,]  46.65691 -20.85526  53.57443
```

```
## [1] "cov x rotated"
```

```
##           [, 1]      [, 2]      [, 3]
## [1,]  574.0049 -299.23442 123.47419
## [2,] -299.2344  202.79375 -86.50309
## [3,]  123.4742 -86.50309  81.26759
```

9.5 Exercise 4-4

分散は異なるが、分散の和は変わっていないことを、分散共分散行列のトレースを計算することで確かめよ。
同様に、分散共分散行列の固有値の和としてトレースを計算することでも確かめよ。

```
## [1] 858.0663
```

```
## [1] 858.0663
```

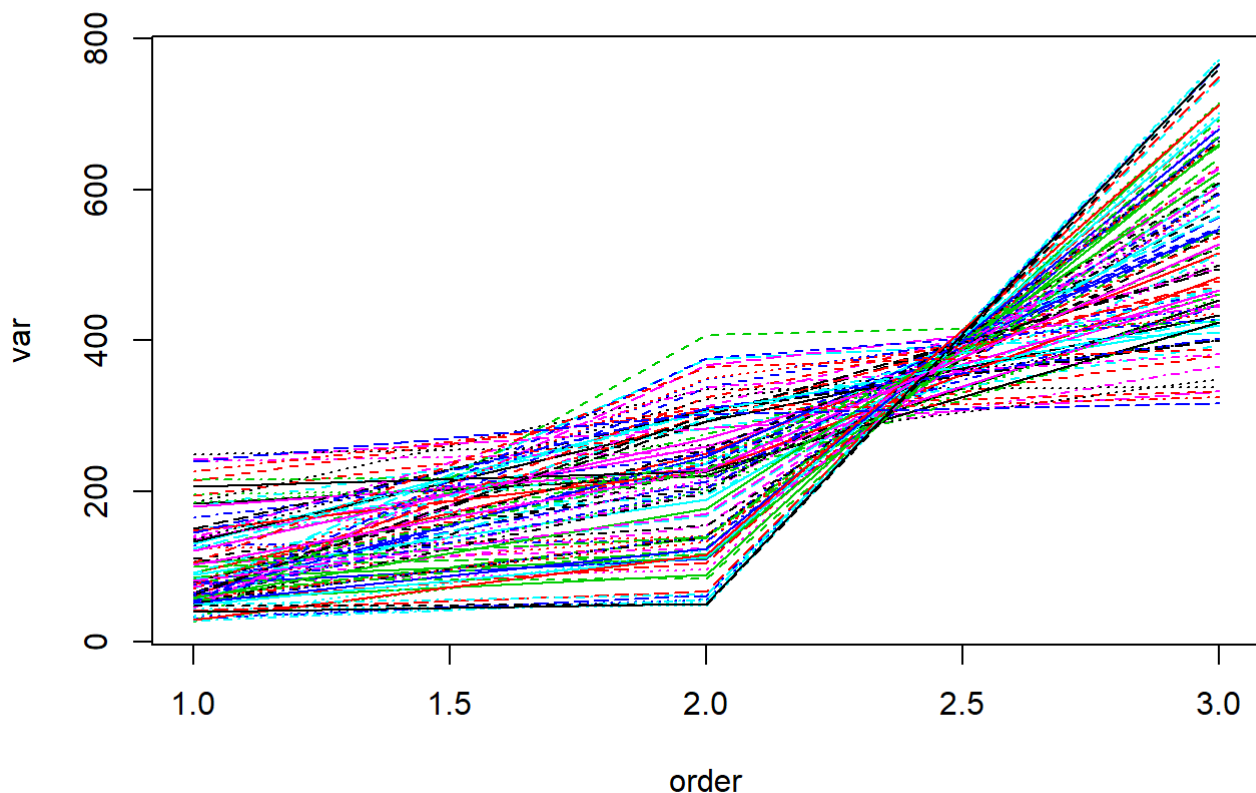
```
## [1] 858.0663
```

```
## [1] 858.0663
```

9.6 Exercise 4-5

データセットxの軸の取り方を回転することによって変えても、各軸の分散の和は変わらない。分散の和は変わらないが、分散の総和の分配具合は変わる。

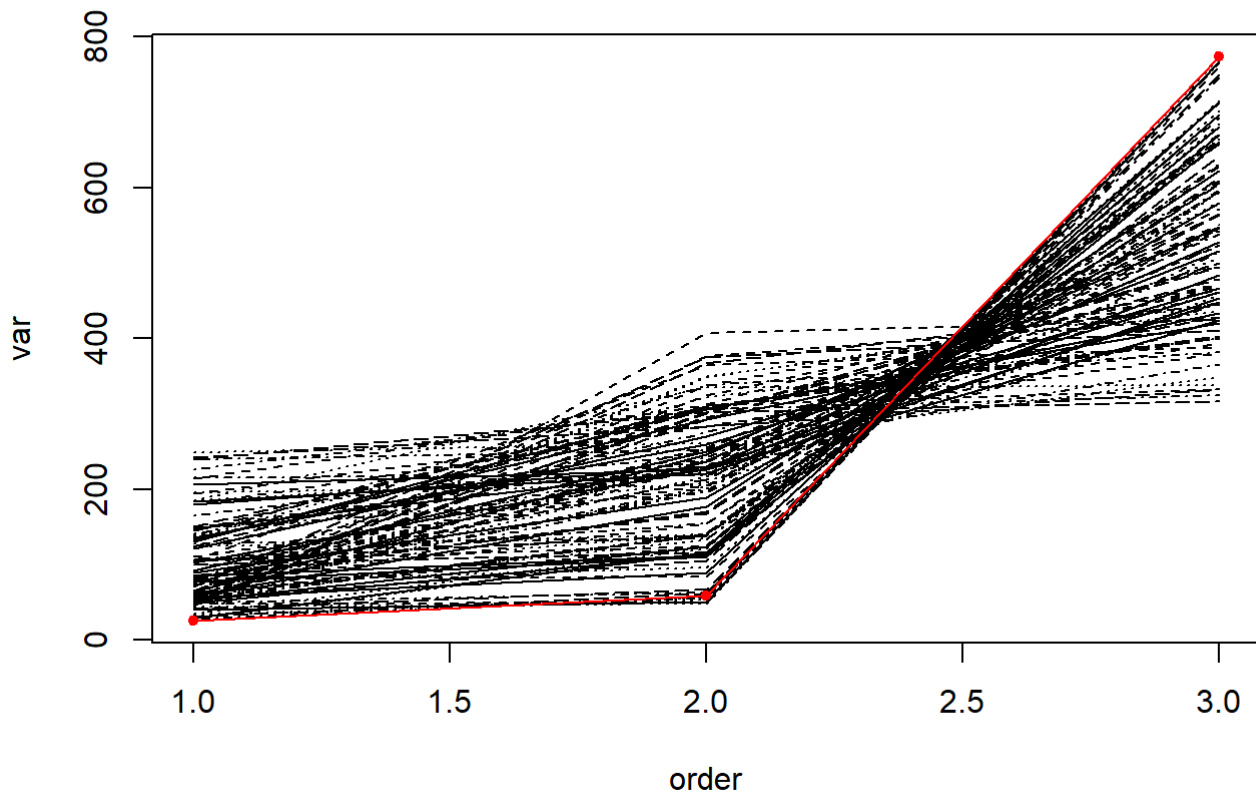
そのことを、多数の回転行列をランダムに発生させ、データセットを回転した上で各軸方向の分散を計算し、その分散の値を大小順序でソートしてプロットし、分配具合が変わることを具体的に確認せよ。



9.7 Exercise 4-6

x の分散共分散行列の固有値を求めることで、分散共分散行列を対角化するような回転をしたときの、各軸の分散を求めることができる。

その分散を4-4のプロットに重ねてプロットせよ。



9.8 Exercise 4-7

4-4,4-5と同様のプロットを変数の数を増やして実施せよ。

```
d <- 10
n <- 1000
x <- matrix(rnorm(d*n), ncol=d)
x <- apply(x, 2, cumsum)

plot(as.data.frame(x[, 1:5]), pch=20, cex=0.1)
```

