# PCA with Matrix 行列でPCA

**ryamada**

**2016年12月24日**



# 1 行列の特徴 Features of matrix

Do you remember what kind of features do square matrices have? 正方行列の特徴を思い出そう。

Their essential features did not change with rotation.

正方行列は回転させても変わらない性質があった。

The sum of eigenvalues (trace) and the product of eigenvalues (determinant) are among the features.

固有値の和(トレース)と固有値の積(行列式)がそれである。

Look back Exercises 4-6 and 4-7. When variance-covariance matrix was rotated, the sum and product of eigenvalues did not change but the individual eigenbvalues changed.

『行列の特徴』のExercises 4-6, 4-7で見たように、分散共分散行列を回転させると、固有値の和と積は変わらないが、固有値の内訳は変わる。

The diagonalization gave you heterogeneous sets of eigen values.

そして、対角化すると、固有値の大小が最も大きくばらつくようにできることを見た。

When the heterogeneity of eigenvalues is the biggest, the largest eigenvalue is the largest realizable and the smallest eigenvalue is the smallest realizable.

最も大きくばらつくとは、最大固有値が、色々な回転の中で最大となり、最小固有値が色々な回転の中で最小になるような内訳の取り方を指す。

# 2 Exercise 1

## 2.1 Exercise 1-1

『行列の特徴』のExercises 4-6, 4-7を再度実施せよ。

Re do Exercises 4-6, 4-7

# 3 PCA

PCAは分散共分散行列を回転させて、固有値のばらつきを大きくして、数少ない固有値で、すべての固有値の和の多くの部分を説明しようとするものである。

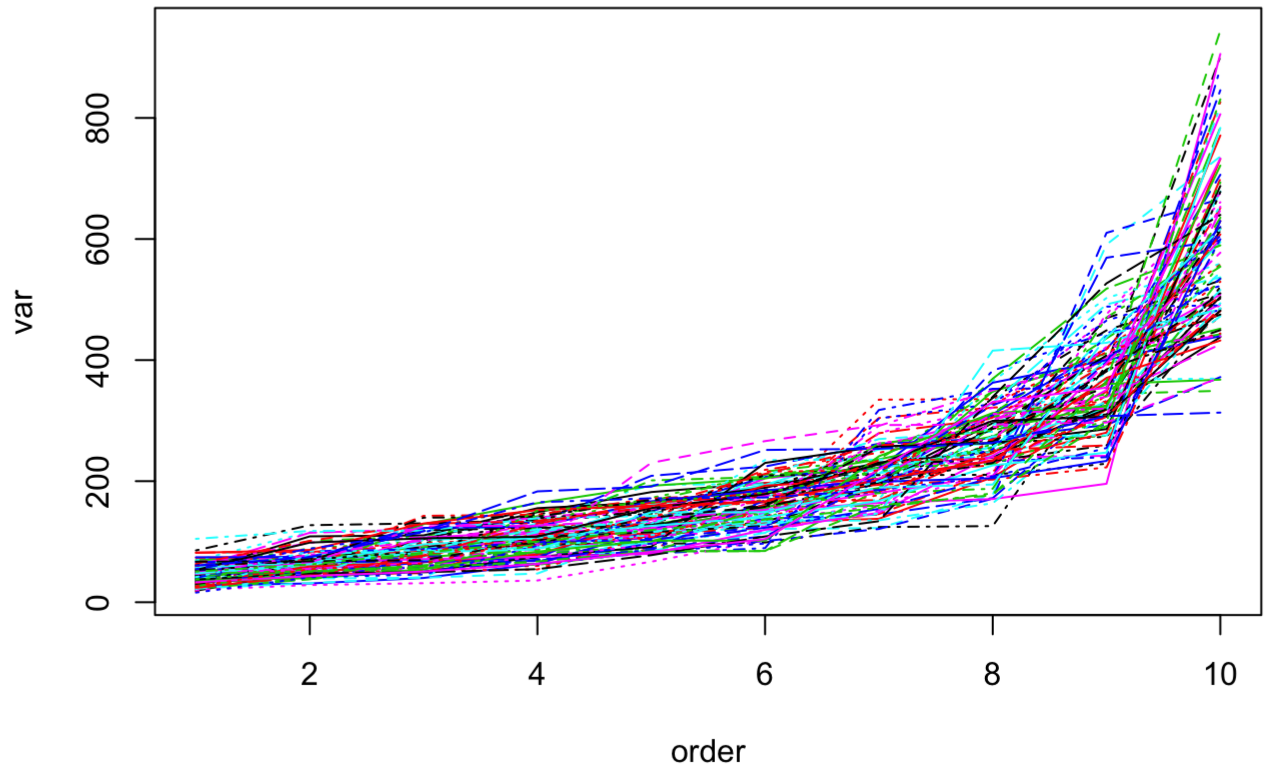
PCA is the procedure to find the rotation that makes the heterogeneity of eigenvalues biggest, that gives you the direction(s) that describes the maximum fraction of the total variation of sample records.

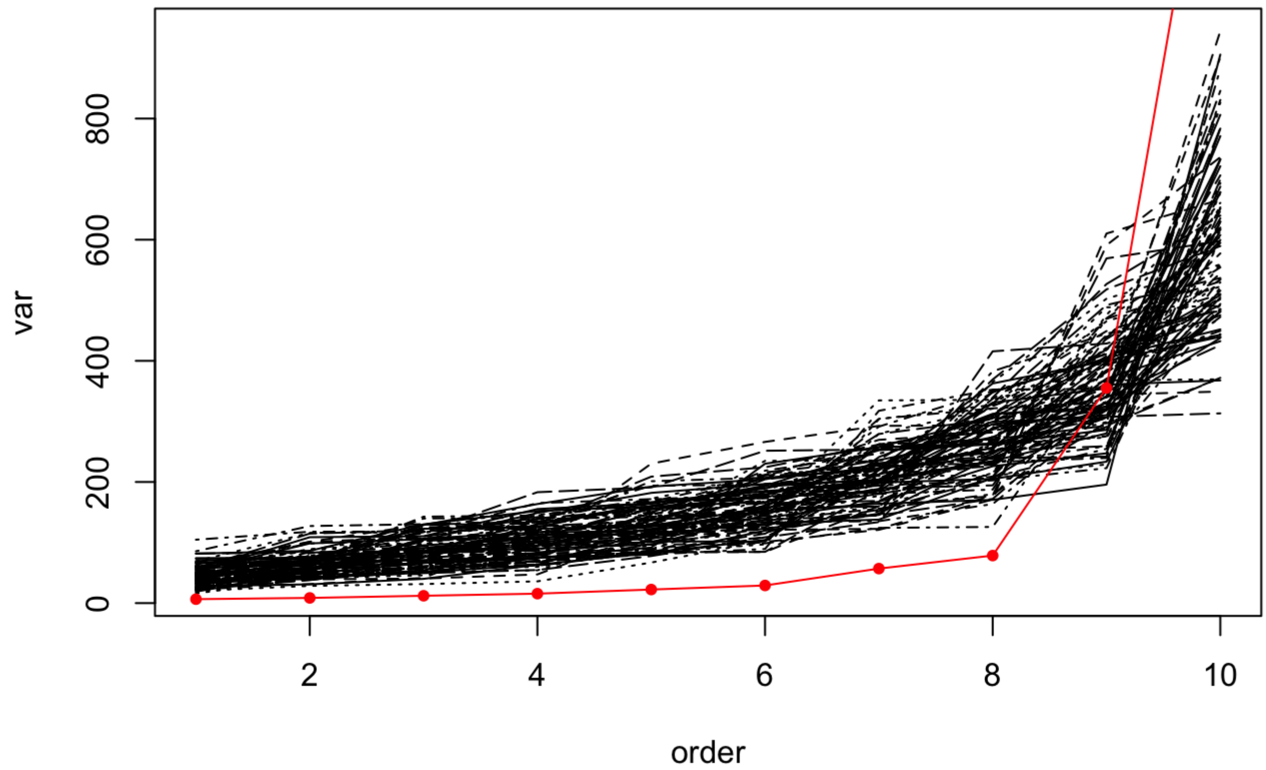
### 3.0.1 Exercise 4-6

Calculating the eigenvalues of variance-covariance matirx x, you can tell the variance of every axes when the records are rotated so that variance-covariance matrix is diagonalized.

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。

Plot them on top of the plot of 4-4. その分散を4-4のプロットに重ねてプロットせよ。



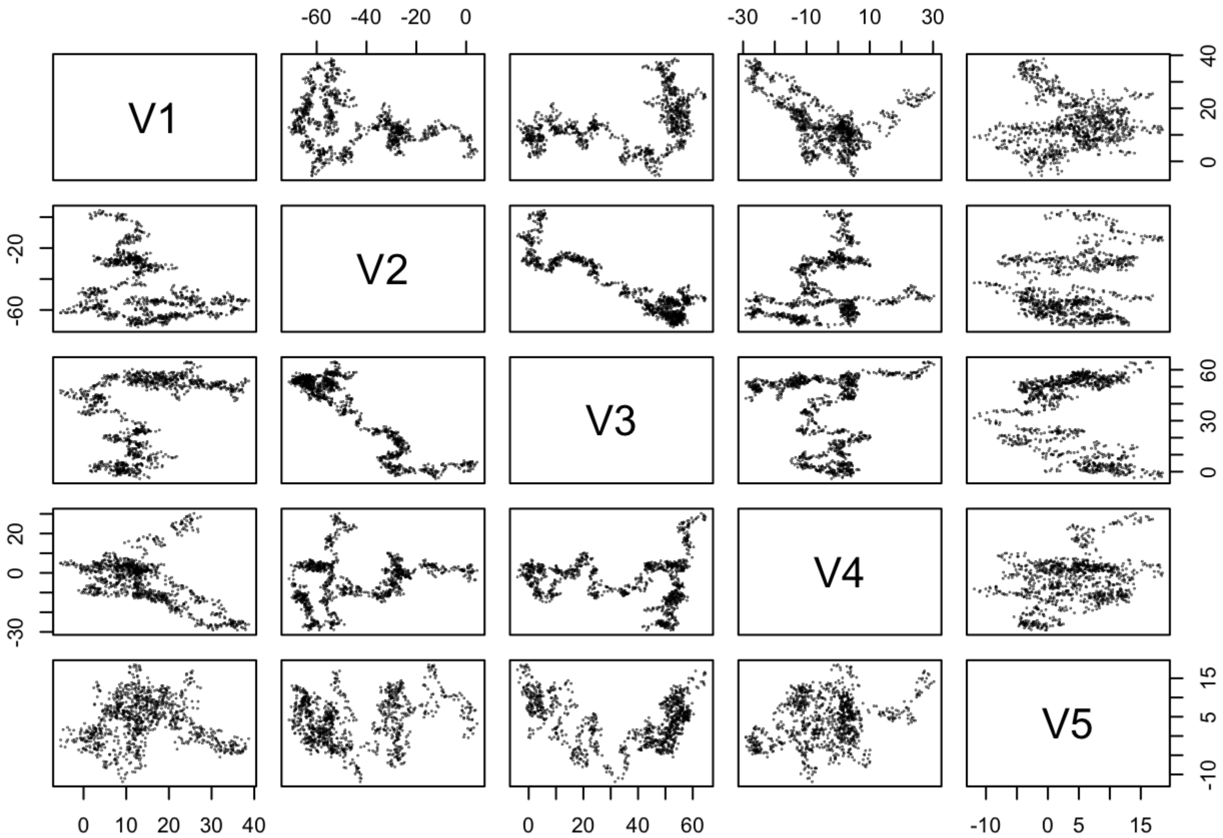

### 3.0.2 Exercise 4-7

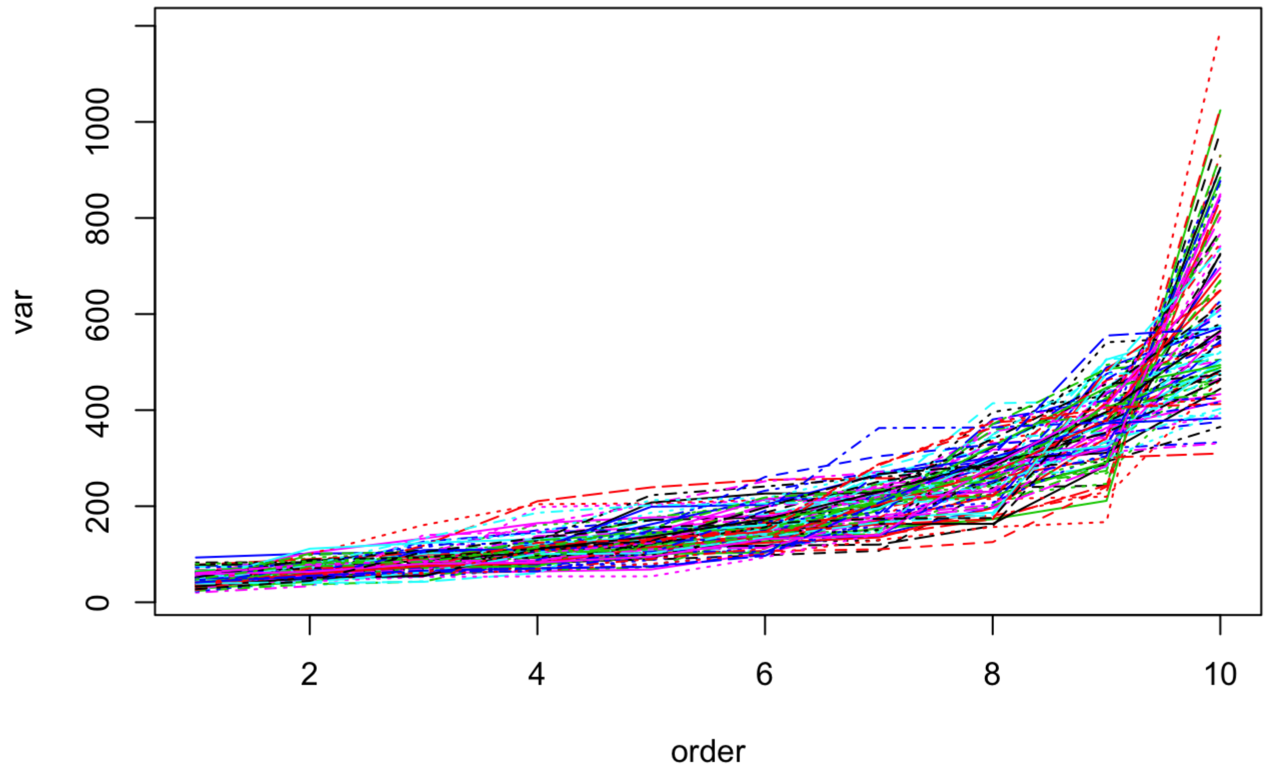
Increase the number of variables of 4-4 and 4-5 and do the same.

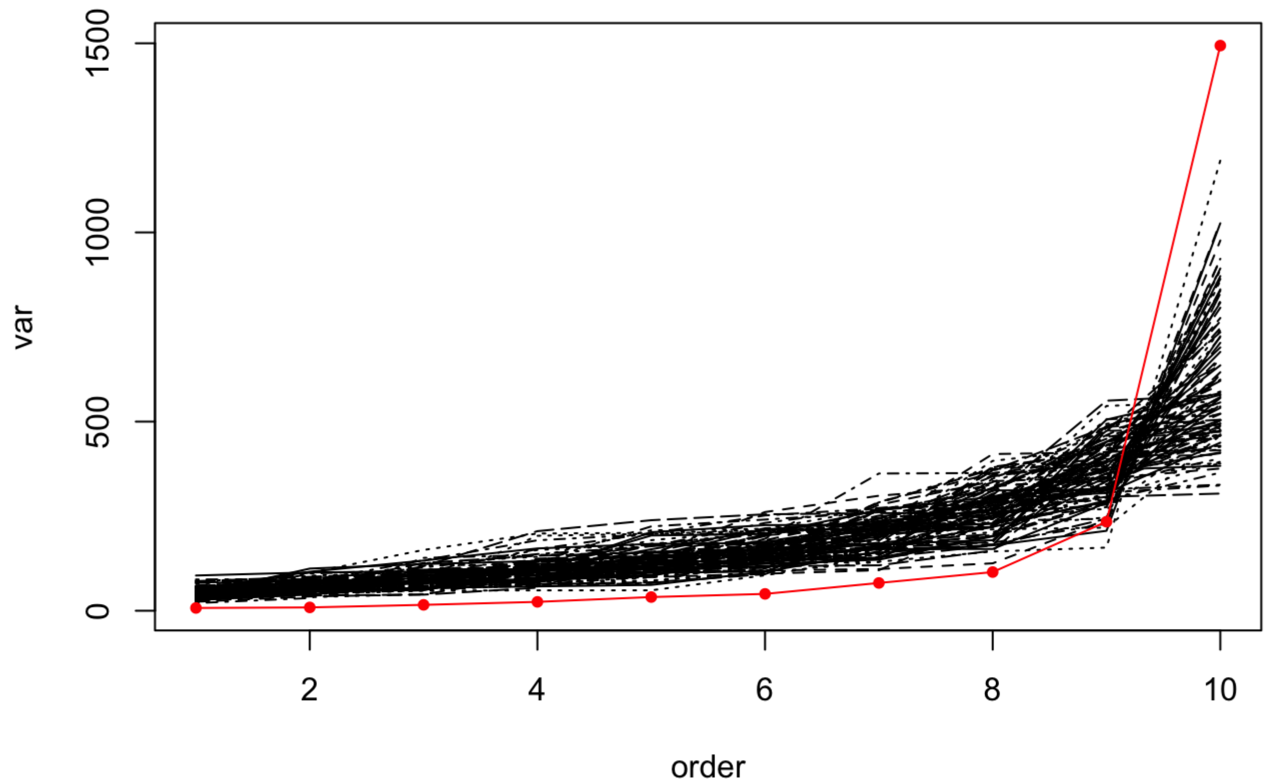
4-4,4-5と同様のプロットを変数の数を増やして実施せよ。

```
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)
```




var
0
200
400
600
800
1000
2
4
6
8
10
order


var
0
500
1000
1500
2
4
6
8
10
order