演習：主成分分析

慶應義塾大学　片山 翔太

秋学期演習の予定

- **主成分分析 (第4回)**
  - 言語データからBoW形式へ
  - 潜在的意味分析
  - 特異値分解と主成分分析
- **線形回帰分析における特徴選択 (第5回)**
  - Best subset selection, Ridge, Lasso
- **パネルデータ解析 (第6回)**
  - Difference-in-difference (DiD) method (差分の差法)
  - Synthetic control method (合成変数法)

潜在的意味分析

背景

- **言語データ考える**

```
Luke, I am your father.
When I was your age, we didn't have Star Destroyers...
Luke, may I have some privacy, please.
No Luke, you're not old enough.
Hey, Dad, look at me! I'm jedi!
Harry Potter was a highly unusual boy in many ways.
Get out of the way, Harry Potter.
Listen to me Harry.
```

#出てくる単語を使って各文章の特徴がつかめそう

各文章の背後に何らかのトピックがありそう...

背景

- **言語データからの情報抽出**
  - 出てくる言語の特徴から何らかの情報を抜き出したい
  - どうやってデータにするか？ → **Bag of Words形式**

I like it. She likes it.

It is fine.

She is fine.

#全文章に出てくる単語

	I	like	it	is	fine	she
文1	1	2	2	0	0	1
文2	0	0	1	1	1	0
文3	0	0	0	1	1	1

#各文章で出てくる単語の個数

テキストデータからBoW形式へ

```
> library(tm)  #パッケージ”tm”をインストールして読み込み
> rawtext <- read.csv("test_sentences.csv",stringsAsFactors=F,
header=F)[,1]
> sentences <- VCorpus(VectorSource(rawtext))
                                    #データを読み込んでコーパスを作成
```

元の文章データへのアクセスはcontentを使う

```
> sentences[[1]]$content
[1] "Luke I am your father."
> sentences[[2]]$content
[1] "When I was your age we didn't have Star Destroyers..."
```

意味のない単語を削除

大文字を小文字へ, 数字を削除, ストップワードを削除, etc

```
> sent.clean <- tm_map(sentences, content_transformer(tolower))
> sent.clean <- tm_map(sent.clean, removeNumbers)
> sent.clean <- tm_map(sent.clean, removeWords, stopwords("en"))
> sent.clean <- tm_map(sent.clean, removePunctuation)
> sent.clean <- tm_map(sent.clean, stemDocument)
> sent.clean <- tm_map(sent.clean, stripWhitespace)
```

```
> sentences[[1]]$content
[1] "Luke I am your father."
> sent.clean[[1]]$content
[1] "luke father"
```

#削除後の文章はこんな感じ

・BoW形式への変換

```
> bow <- DocumentTermMatrix(sent.clean,control=list(wordLengths=c(2,20)))
```

#長さが2~20の単語のみ取得

optional

```
> bow <- bow[,findFreqTerms(bow,2)]
```

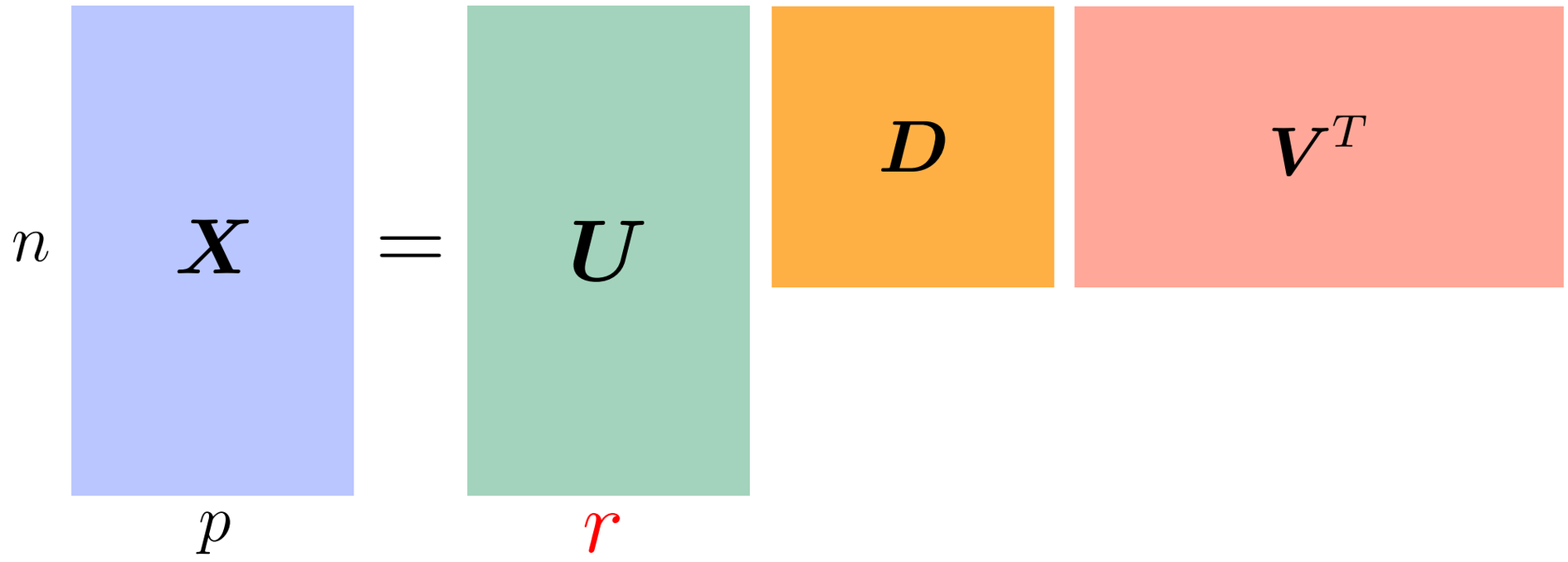
#2つ以上の文章で出てくる単語のみ取得

```
> as.matrix(bow)
    Terms
Docs harri luke potter way
   1     0    1      0   0
   2     0    0      0   0
   3     0    1      0   0
   4     0    1      0   0
   5     0    0      0   0
   6     1    0      1   1
   7     1    0      1   1
   8     1    0      0   0
```

これで言語データが行列形式(BoW)のデータへと変換された

- **行列の特異値分解(SVD)**
  - 任意の実n×p行列**X**は次のように分解が可能

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$$



ただし, $\boldsymbol{D} = \mathrm{diag}(d_1, \ldots, d_r),\ d_j > 0,\ r = \mathrm{rank}(\boldsymbol{X})$

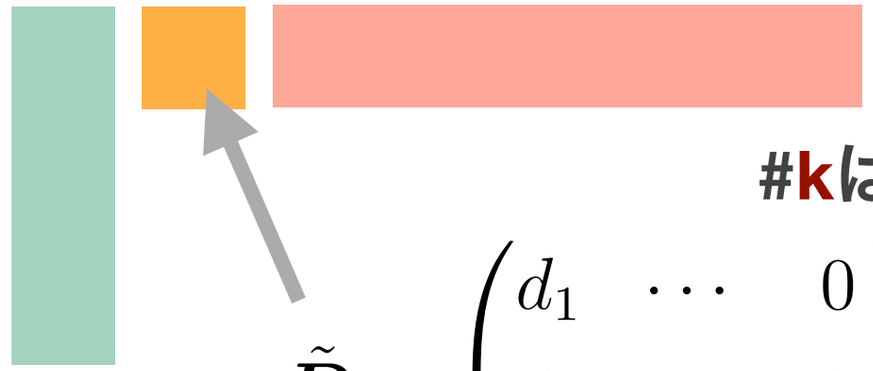
であり, $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ (単位行列)

データ行列の近似

#特にBoW形式の場合は横長になる

サイズの大きな行列X

$\approx$



#kは自分で設定

$$\tilde{\boldsymbol{D}} = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_k \end{pmatrix}, \ k < r$$

近似の精度はkの個数に依存することに注意

Rでやってみよう

Slide3の言語データをBoW形式にしたもの

```
> xx <- read.csv("bow_test.csv",header=T)
```

特異値分解(SVD)を実行

```
> result <- svd(xx); str(result)
List of 3
 $ d: num [1:8] 3.02 2.28 2 1.73 1.57 ..
 $ u: num [1:8, 1:8] 1.60e-17 1.02e-16 1
 $ v: num [1:23, 1:8] 4.17e-16 -2.75e-01
```

```
> ap.xx <- result$u %*% diag(result$d) %*% t(result$v)
> sum(ap.xx - xx)
[1] 4.845766e-15
```

#k=rの場合はちゃんと元の行列を再現している

・kを変えて近似してみる

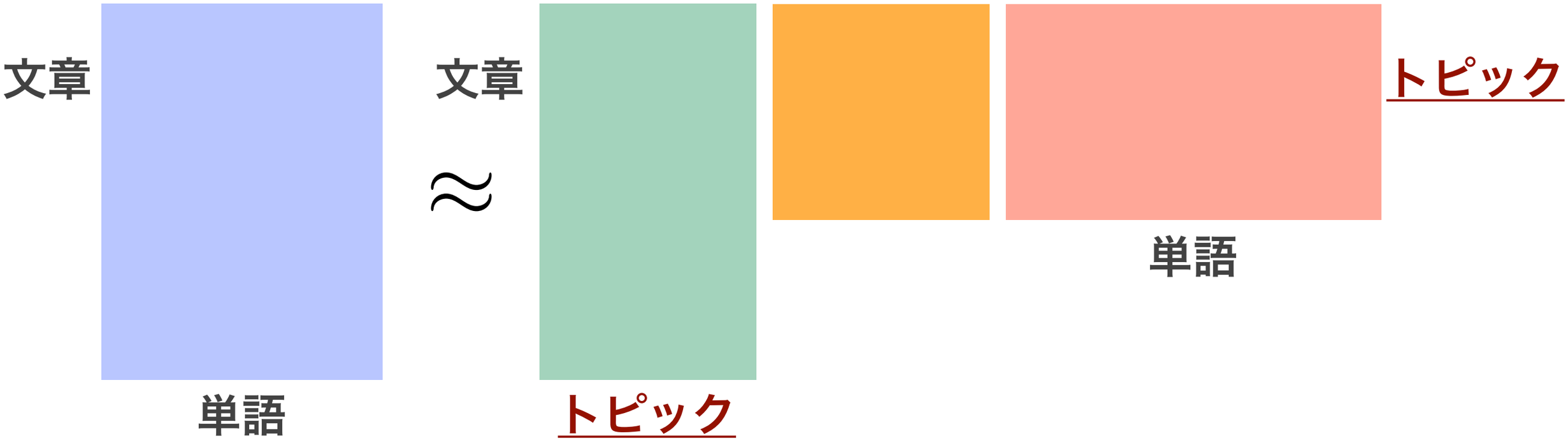
```
> result <- svd(xx)
> U <- as.matrix(result$u)
> D <- diag(result$d)
> V <- result$v
> ap.xx <- U[,1:2] %*% D[1:2,1:2] %*% t(V[,1:2])
> sum(ap.xx - xx)
[1] -7.198575
> ap.xx <- U[,1:5] %*% D[1:5,1:5] %*% t(V[,1:5])
> sum(ap.xx - xx)
[1] -0.1451013
```

#k=2の場合はそんなに良くない

#k=5なら結構良い近似

SVDによる近似の意味

- **BoW形式データのSVD分解**



緑の行列 ➡ 各文章が**どのトピックに属しているか**の情報

赤の行列 ➡ **各トピックの特性** (各単語に対する重み)

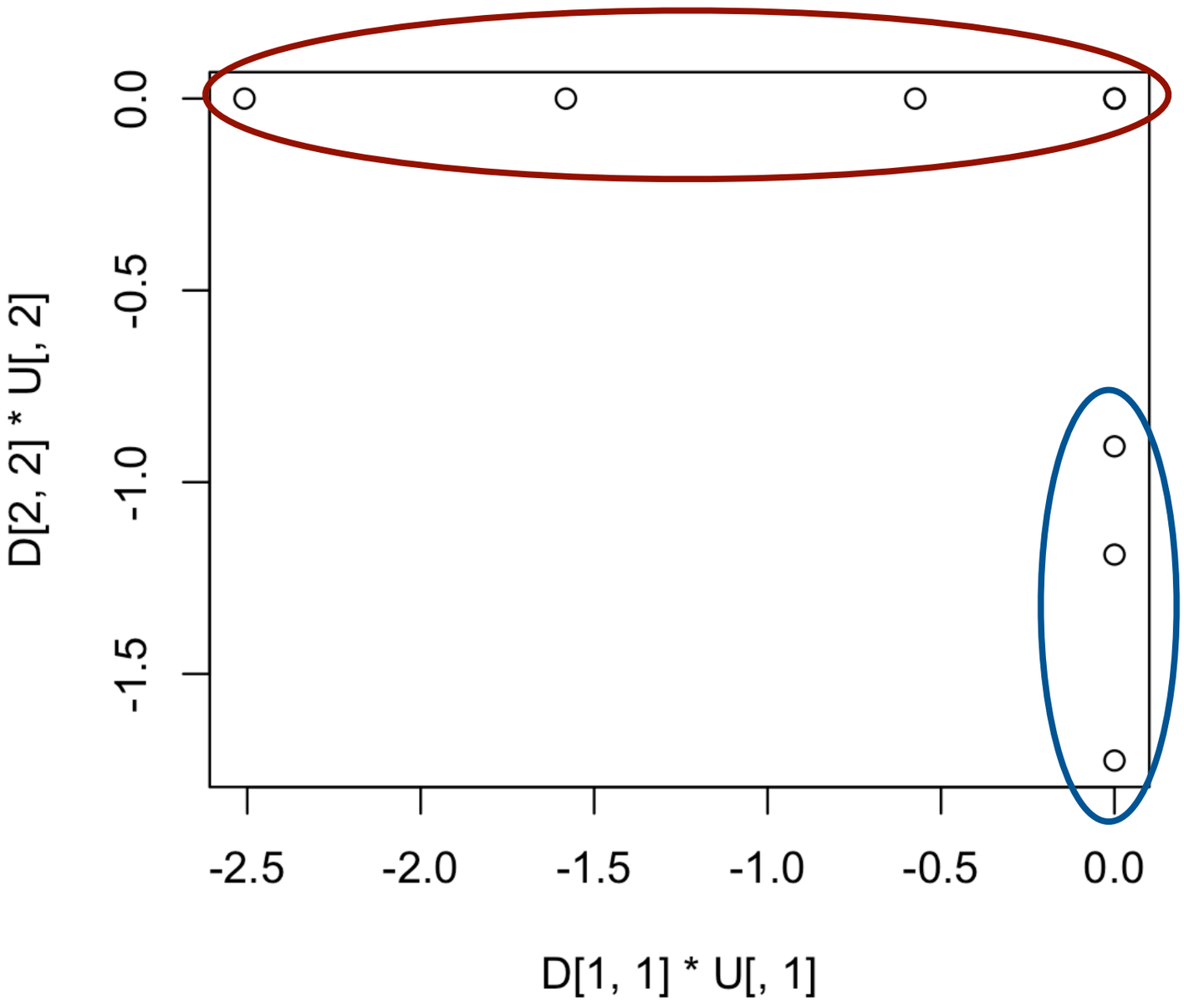
・赤い行列(V)の中身を見てみる

```
> round(V[,1],3)
   age     boy     dad destroy  enough  father     get   harri     hey
 0.000  -0.275   0.000   0.000   0.000   0.000  -0.173  -0.511   0.000
  high    jedi  listen    look    luke    mani     may     old   pleas
-0.275   0.000  -0.063   0.000   0.000  -0.275   0.000   0.000   0.000
potter privaci    star   unusu     way
-0.448   0.000   0.000  -0.275  -0.448
```

```
> round(V[,2],3)
   age     boy     dad destroy  enough  father     get   harri     hey
 0.000   0.000   0.000   0.000  -0.228  -0.174   0.000   0.000   0.000
  high    jedi  listen    look    luke    mani     may     old   pleas
 0.000   0.000   0.000   0.000  -0.733   0.000  -0.331  -0.228  -0.331
potter privaci    star   unusu     way
 0.000  -0.331   0.000   0.000   0.000
```

トピック1 = ハリーポッター, **トピック2** = ルーク

- **トピック1とトピック2を使ったプロット**
  - UDの第1列と第2列で散布図を描く



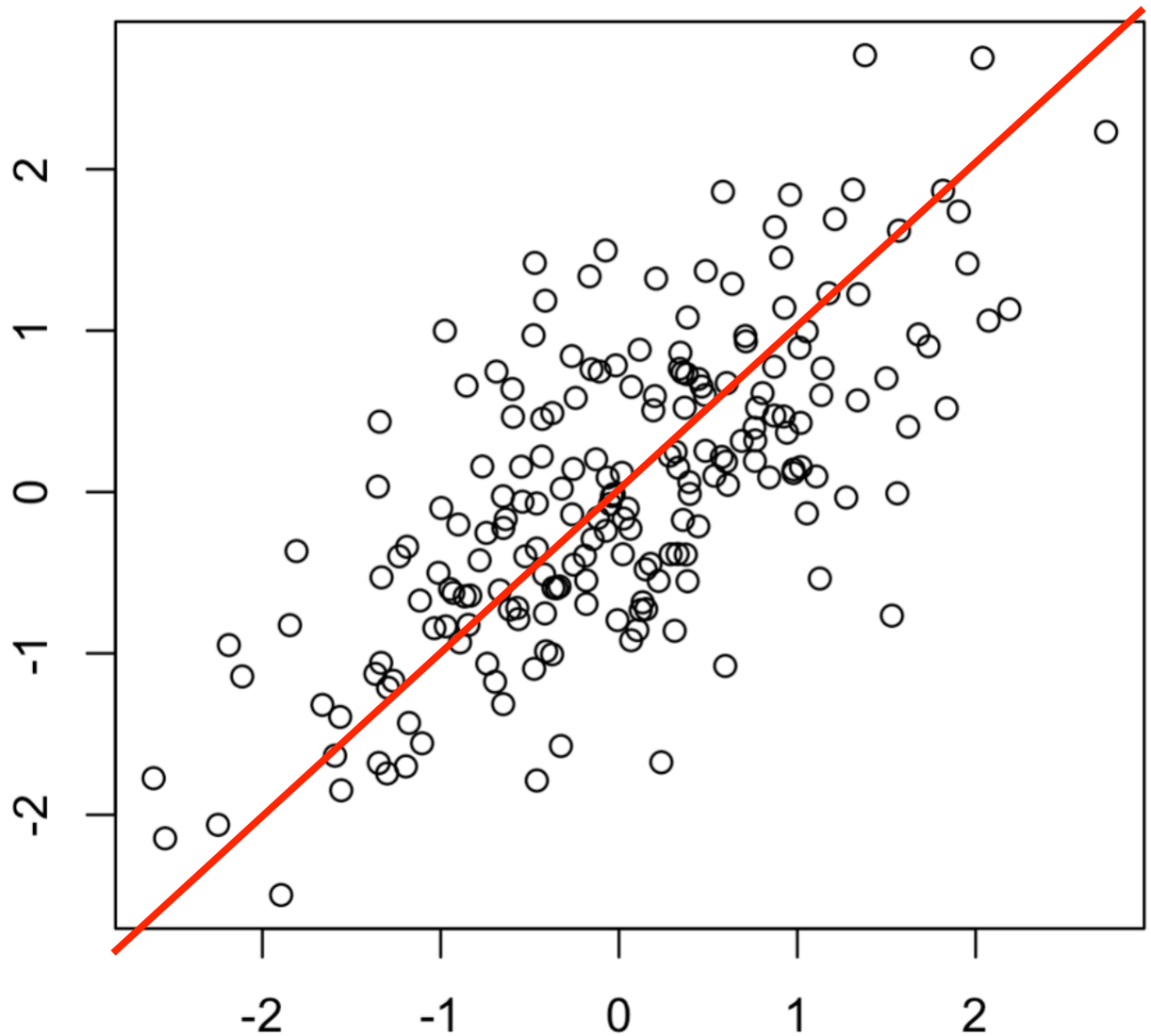
#なんとなく文章(点)が
分類されている

#このプロットには一切の
教師データを使っていない

教師なし学習！

主成分分析

- **主成分分析**
  - 教師なし学習のひとつ，大規模データの視覚化が可能
  - データを理解しやすい変数に削減する(**次元削減**)



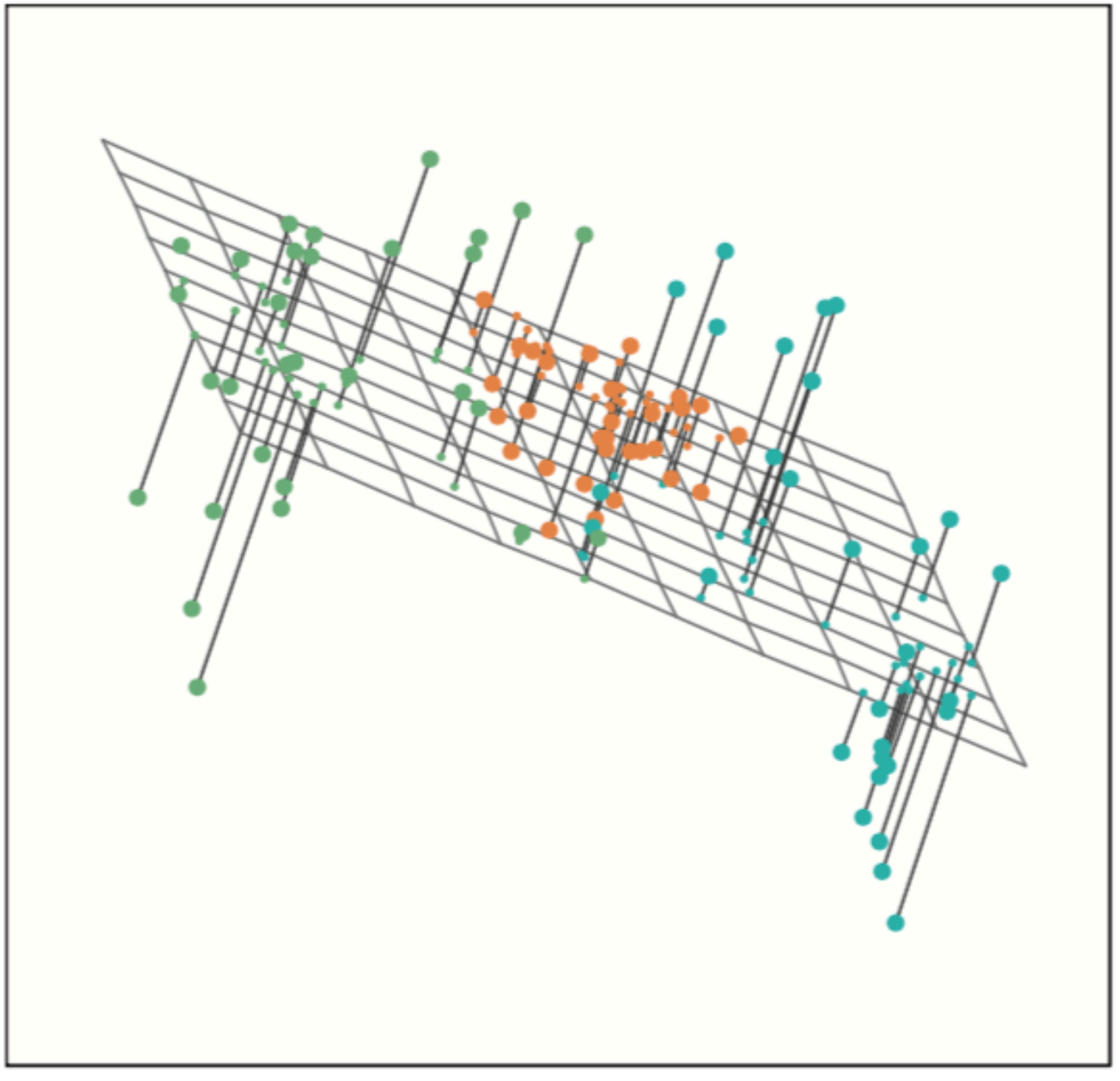
#2次元のデータだけど

赤い軸(1次元)に各点を落とせそう

#次元の大きなデータを低次元へ

解釈が簡単になる

・3次元を2次元に落とす場合のイメージ



- **複数の変数を組み合わせて新しい変数を作成**

Originalデータ

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{2n} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_n^T \end{pmatrix}$$

Ex. $\boldsymbol{x}_1^T = (x_{11}, x_{12}, \cdots, x_{1p})$

Newデータ (主成分スコア)

#ベクトルで書けば $z_{i1} = \boldsymbol{\phi}_1^T \boldsymbol{x}_i$

$$z_{i1} = \phi_{11} x_{i1} + \phi_{12} x_{i2} + \cdots + \phi_{1p} x_{ip}, \quad i = 1, 2, \ldots, n$$

$$z_{i2} = \phi_{21} x_{i1} + \phi_{22} x_{i2} + \cdots + \phi_{2p} x_{ip}, \quad i = 1, 2, \ldots, n$$

$$\vdots$$

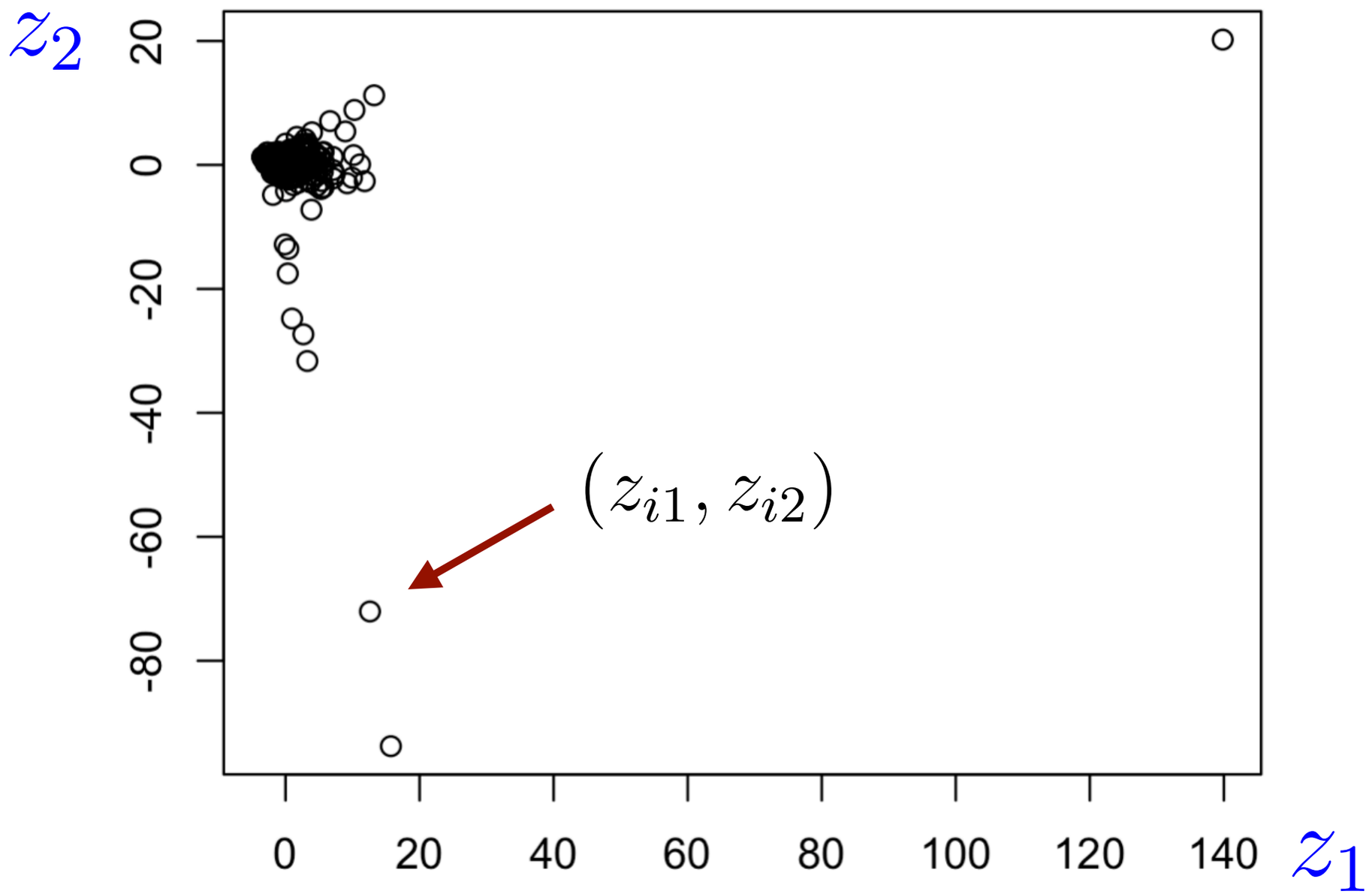
$$z_{ik} = \phi_{k1} x_{i1} + \phi_{k2} x_{i2} + \cdots + \phi_{kp} x_{ip}, \quad i = 1, 2, \ldots, n$$

変数の次元をpからkへ削減

データを2次元に表現する場合

$$z_{i1} = \phi_{11} x_{i1} + \phi_{12} x_{i2} + \cdots + \phi_{1p} x_{ip}, \quad i = 1, 2, \ldots, n$$

$$z_{i2} = \phi_{21} x_{i1} + \phi_{22} x_{i2} + \cdots + \phi_{2p} x_{ip}, \quad i = 1, 2, \ldots, n$$



Rで手を動かしながらやってみる

- **民力データ from データサイエンスコンソーシアム**

http://datascience.jp/tutorial.html

#行の名前を1列目に指定

```
> Ppower <- read.csv("Ppower.csv",header=T,row.names=1)
> head(Ppower)
           population households office income nationaltax
Hokkaido      5650573    2522295 270504 145293       1379098
Aomori        1479358     551806  74341  32498        242730
Iwate         1405060     488354  72456  34152        224269
Miyagi        2350026     856527 115297  61092        735121
Akita         1173722     410308  65300  27294        186373
Yamagata      1225990     387732  70523  29851        202270
```

47都道府県×24変数

24変数を合成して新しい変数を作成していく

新しい変数の作り方

- **Step1**
  - データを中心化する

$$x_{ij} - \bar{x}_j, \quad \bar{x}_j = \frac{1}{n}\sum_{k=1}^{n} x_{kj}$$

  - (Option) 基準化を行うこともある

$$\frac{x_{ij} - \bar{x}_j}{s_j}, \quad s_j^2 = \frac{1}{n-1}\sum_{k=1}^{n}(x_{kj} - \bar{x}_j)^2$$

- これ以降データ行列**X**は中心化されているとする

Rで中心化

```
> xx <- scale(Ppower,scale=T)  #スケールが大きいので基準化も行う
> round(apply(xx,2,mean),5)
      population       households           office           income
               0                0                0                0
     nationaltax        localtax   agriculturalout     forestryprod
               0                0                0                0
         landing          factory industryamount         employer
               0                0                0                0
     storesales     powerconsum          deposit     publicexpense
               0                0                0                0
          houses            cars educationalamount       booksales
               0                0                0                0
   newspapersales               tv            phone         posting
               0                0                0                0
```

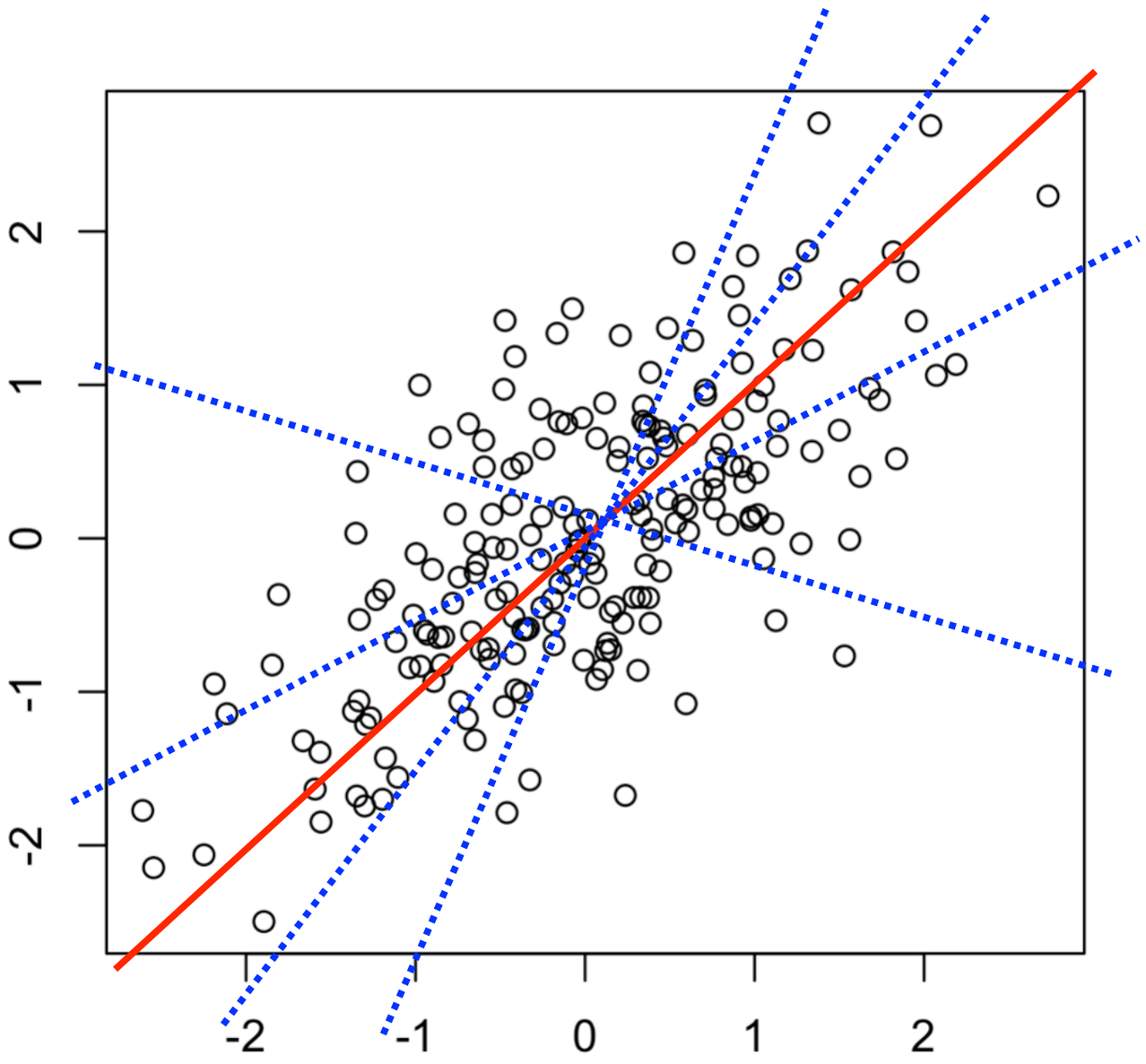
- **Step2**
  - 分散を最大にする係数**ϕ_1**を求める
  - 注意：中心化されていればzの平均は0

$$\bar{z}_1 = \frac{1}{n}\sum_{i=1}^{n} z_{1i} = \phi_{11}\bar{x}_1 + \phi_{12}\bar{x}_2 + \cdots + \phi_{1p}\bar{x}_p = 0$$

分散 $\frac{1}{n}\sum_{i=1}^{n} z_{i1}^2 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{\phi}_1^T \boldsymbol{x}_i)^2$ **これを最大化**

ただし $\|\boldsymbol{\phi}_1\|_2^2 = \boldsymbol{\phi}_1^T \boldsymbol{\phi}_1 = 1$ としておく

#この条件がないと分散がいくらでも大きくなる



赤い実線上で最もばらついているように見える

データを**最も解釈しやすい**方向になっている

- **Step3**
  - Step2で求めたϕ_1に直交するϕ_2の中で分散最大なものを探す

$$\frac{1}{n}\sum_{i=1}^{n} z_{i2}^2 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{\phi}_2^T \boldsymbol{x}_i)^2$$ **を最大化**

ただし $\|\boldsymbol{\phi}_2\|_2^2 = \boldsymbol{\phi}_2^T\boldsymbol{\phi}_2 = 1, \quad \boldsymbol{\phi}_1^T\boldsymbol{\phi}_2 = 0$

- **Step4**
  - これを求めたい変数の数だけ繰り返す(max. p)

$$\frac{1}{n}\sum_{i=1}^{n} z_{i\ell}^2 = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{\phi}_\ell^T \boldsymbol{x}_i)^2$$ **を最大化**

ただし $\|\boldsymbol{\phi}_\ell\|_2^2 = 1, \quad \boldsymbol{\phi}_1^T\boldsymbol{\phi}_\ell = \boldsymbol{\phi}_2^T\boldsymbol{\phi}_\ell = \cdots = \boldsymbol{\phi}_{\ell-1}^T\boldsymbol{\phi}_\ell = 0$

最大化問題の解

- **標本共分散行列の固有ベクトル**で最大化される

$$\boldsymbol{\Sigma} = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T$$

#Xは中心化されている
ことに注意

固有値：$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$　　**#固有値は全て正**

固有ベクトル：$\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_p$　　**#固有ベクトルはp次元**

固有ベクトルは正規直交するように取れる

$$\boldsymbol{v}_i^T\boldsymbol{v}_j = 0 \ \ (i \neq j), \quad \boldsymbol{v}_i^T\boldsymbol{v}_i = \|\boldsymbol{v}_i\|^2 = 1$$

- **Step2〜4の実際**
  - 標本共分散行列を計算
  - Step2 : 最大固有値に対応する固有ベクトルを計算 $\boldsymbol{\phi}_1 = \boldsymbol{v}_1$
  - Step3 : 次に大きな固有値に対する固有ベクトルを計算
  - Step4 : 以下求めたい変数の数まで求める

$$\boldsymbol{\phi}_2 = \boldsymbol{v}_2, \quad \boldsymbol{\phi}_3 = \boldsymbol{v}_3, \cdots, \boldsymbol{\phi}_\ell = \boldsymbol{v}_\ell$$

- **Step 5 : 対応する主成分スコアzを計算**

$$\begin{aligned} z_{i1} &= \boldsymbol{v}_1^T \boldsymbol{x}_i, \quad i = 1, \ldots, n \\ z_{i2} &= \boldsymbol{v}_2^T \boldsymbol{x}_i, \quad i = 1, \ldots, n \\ &\vdots \\ z_{i\ell} &= \boldsymbol{v}_\ell^T \boldsymbol{x}_i, \quad i = 1, \ldots, n \end{aligned}$$

Rでstep2~step4

標本共分散行列の計算

```
> sigma <- t(xx)%*%xx/47
```

$$\boldsymbol{\Sigma} = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T$$

固有値・固有ベクトル

```
> result <- eigen(sigma)
> evalue <- result$values
> evector <- result$vectors
```

$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$

evector $= V = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_p)$

#固有ベクトルは正規直交 $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}$

チェックしてみよう

```
##round(t(evector) %*% evector,5)
##round(evector %*% t(evector),5)
```

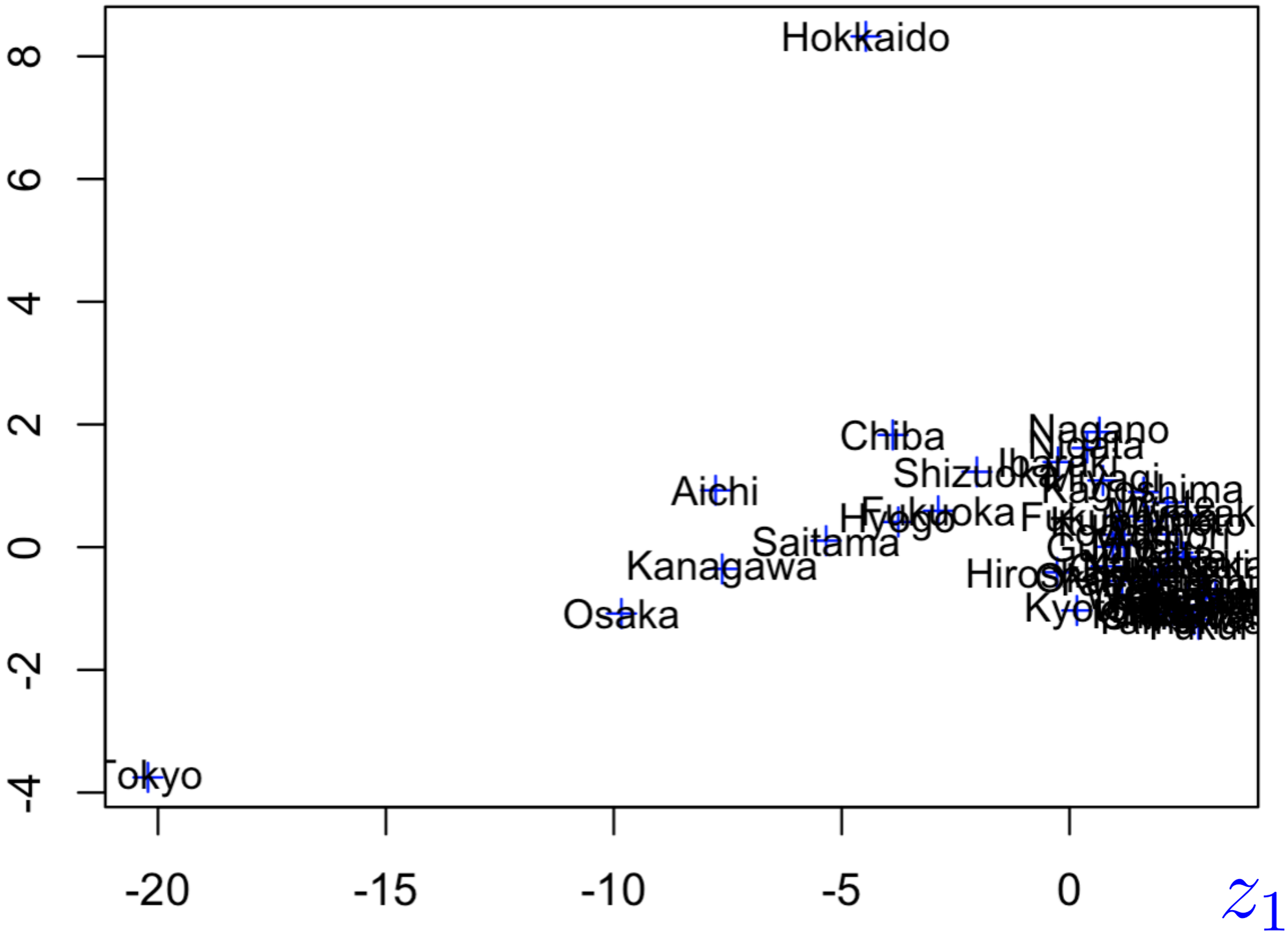
Rでstep5

第2主成分スコアまで計算してプロット

```
> z1 <- xx %*% evector[,1]
> z2 <- xx %*% evector[,2]
```

```
> plot(z1,z2)
> text(z1,z2,row.names(xx),cex=0.8,pos=3)
```

z_2


8
6
4
2
0
-2
-4
Hokkaido
Chiba
Nagano
Shizuoka
Aichi
Saitama
Kanagawa
Osaka
okyo
-20
-15
-10
-5
0

z_1

主成分スコアの性質

- **主成分スコアは平均0で互いに相関なし**

$$\frac{1}{n}\sum_{i=1}^{n} z_{i1} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{v}_1^T \boldsymbol{x}_i = \boldsymbol{v}_1^T \left(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i\right) = 0$$

#データ行列は中心化されている

$$\frac{1}{n}\sum_{i=1}^{n} z_{i1} z_{i2} = \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{v}_1^T \boldsymbol{x}_i)\,(\boldsymbol{v}_2^T \boldsymbol{x}_i) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{v}_1^T (\boldsymbol{x}_i \boldsymbol{x}_i^T) \boldsymbol{v}_2$$
$$= \boldsymbol{v}_1^T \boldsymbol{\Sigma} \boldsymbol{v}_2 = \lambda_2 \boldsymbol{v}_1^T \boldsymbol{v}_2 = 0$$

#$\boldsymbol{v}_2$は$\boldsymbol{\Sigma}$の固有ベクトル

Rで確認

```
> Z <- xx %*% evector[,1:5]   #例えば第5主成分スコアまで求める

> round(apply(Z,2,mean),5)
[1] 0 0 0 0 0

> round(t(Z)%*%Z/47,5)
          [,1]    [,2]    [,3]    [,4]    [,5]
[1,] 18.53123 0.00000 0.00000 0.00000 0.00000
[2,]  0.00000 2.51227 0.00000 0.00000 0.00000
[3,]  0.00000 0.00000 1.16363 0.00000 0.00000
[4,]  0.00000 0.00000 0.00000 0.63024 0.00000
[5,]  0.00000 0.00000 0.00000 0.00000 0.26602
```

#対角成分には固有ベクトルが並ぶ (式で確認してみよう)

```
> round(evalue,5)[1:5]
[1] 18.53123  2.51227  1.16363  0.63024  0.26602
```

主成分スコアを解釈する

$$z_{i1} = \phi_{11}x_{i1} + \phi_{12}x_{i2} + \cdots + \phi_{1p}x_{ip}, \quad i = 1, 2, \ldots, n$$

$$z_{i2} = \phi_{21}x_{i1} + \phi_{22}x_{i2} + \cdots + \phi_{2p}x_{ip}, \quad i = 1, 2, \ldots, n$$

```
> round(evector[,1][1:9],3)
```

ϕ_1

```
   population     households         office         income     nationaltax
       -0.225         -0.227         -0.228         -0.229         -0.207
    localtax agriculturalout forestryprod        landing
       -0.212         -0.023         0.017         -0.043
```

```
> round(evector[,2][1:9],3)
```

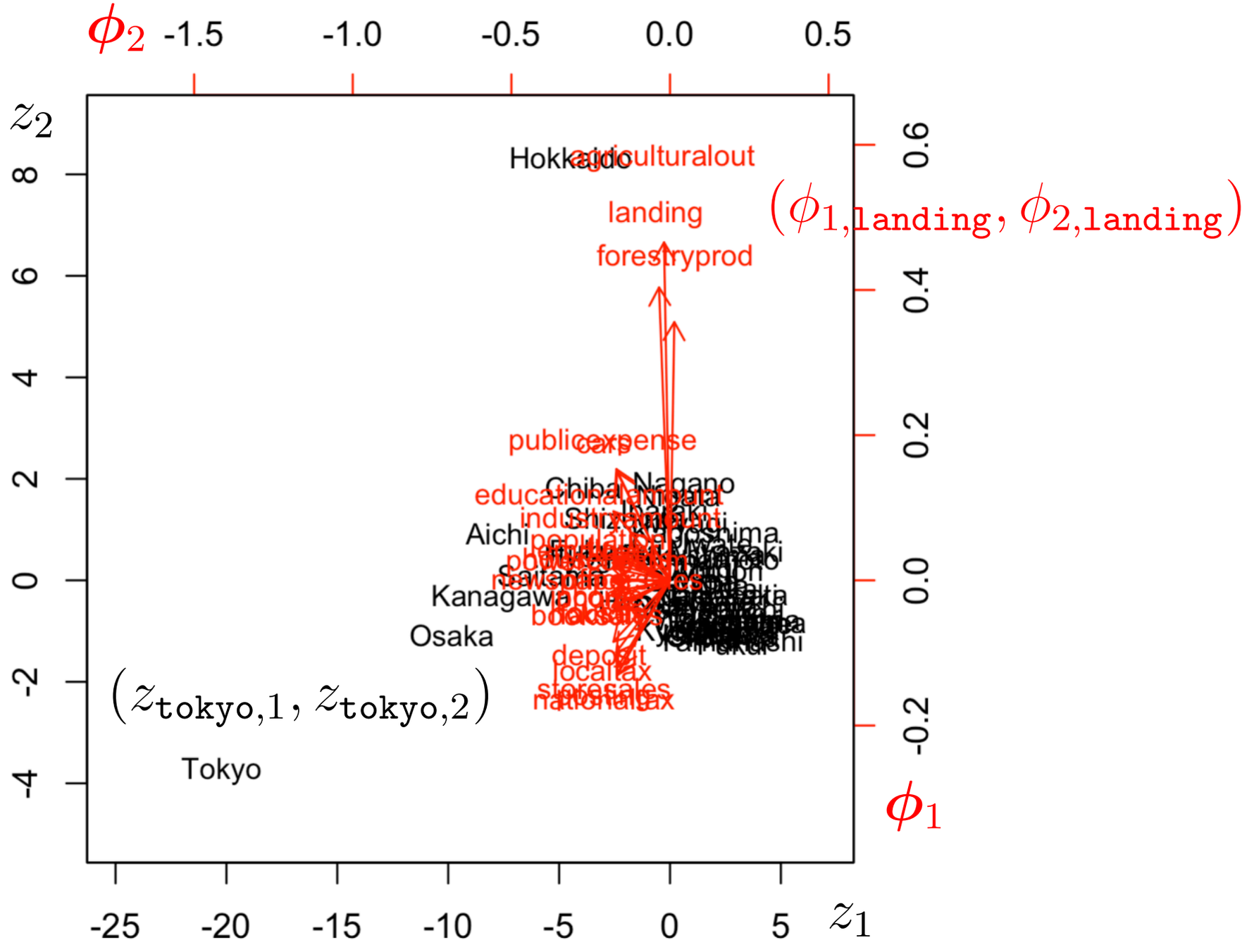
ϕ_2

```
   population     households         office         income     nationaltax
        0.053          0.030          0.002         -0.020         -0.163
    localtax agriculturalout forestryprod        landing
       -0.124          0.582         0.445          0.504
```

#赤い部分が第2主成分スコアに効いていそう

固有ベクトルと主成分スコアのプロット(biplot)

```
> biplot(Z[,1:2],evector[,1:2],ylim=c(-5,9),xlim=c(-25,7),cex=0.85)
```



寄与率

- **主成分はいくつ用意したらいいのか？**
  - 累積寄与率が一つの指標
  - 固有値の総和に占める主成分数に対応する固有値の和

$$\frac{\lambda_1 + \lambda_2}{\lambda_1 + \lambda_2 + \cdots + \lambda_p}$$

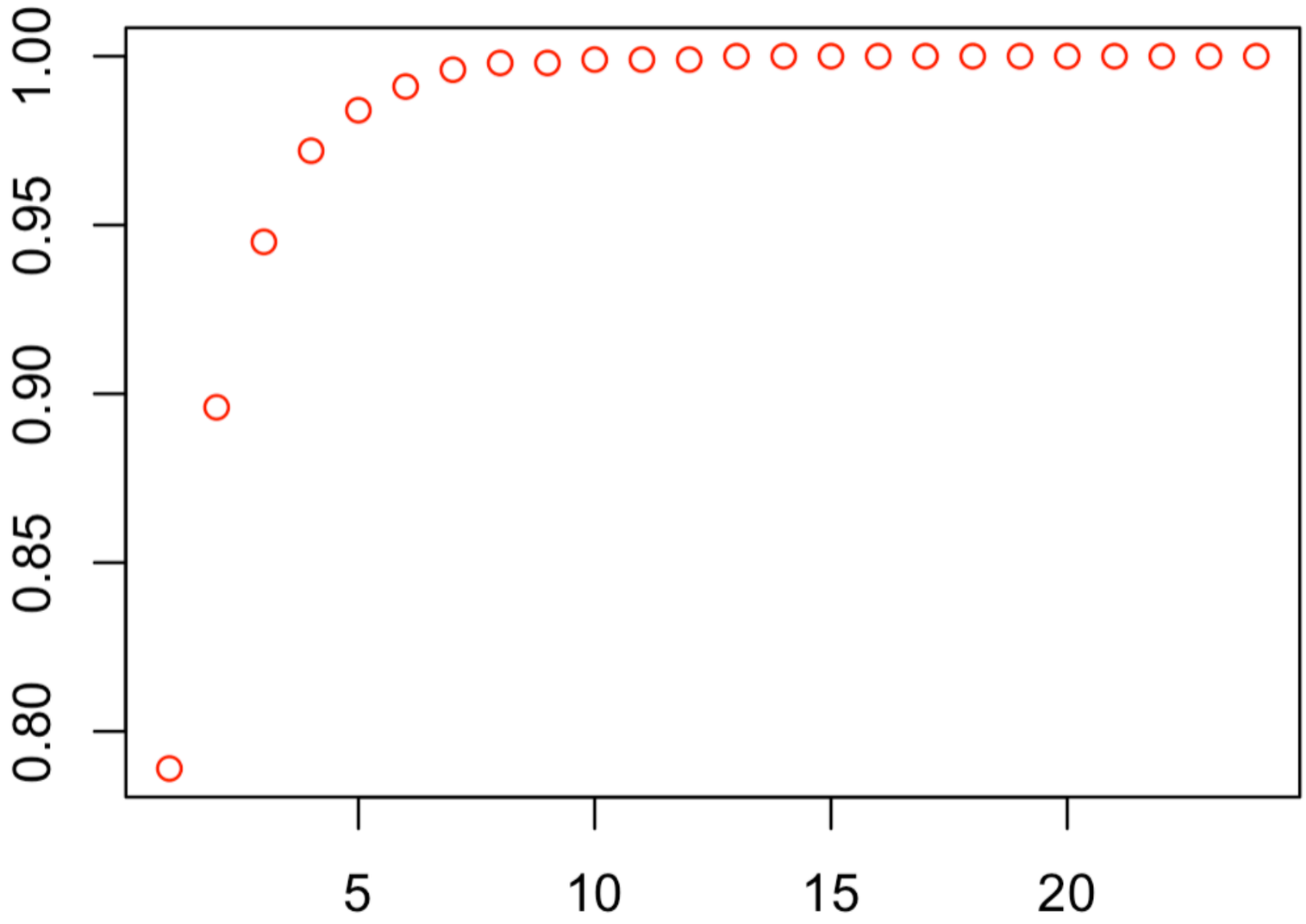
#例えば第2主成分まで使う場合

累積寄与率が十分大きければOK　**#一般的な目安は0.8~0.9**

```
> round(cumsum(evalue)/sum(evalue),3)
 [1] 0.789 0.896 0.945 0.972 0.984 0.991 0.996 0.998 0.998
[10] 0.999 0.999 0.999 1.000 1.000 1.000 1.000 1.000 1.000 1.000
[19] 1.000 1.000 1.000 1.000 1.000 1.000
```

#ベクトルの累積和を求める関数cumsum

- **プロットする場合もある**
  - x軸に主成分数, y軸に累積寄与率



なぜ累積寄与率で評価？ → **SVDで説明可能！**

SVDとの関連

固有値・固有ベクトルから主成分スコアへ

- **標本共分散行列の固有値と固有ベクトル**

$$\boldsymbol{\Sigma}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i, \quad i = 1, 2, \ldots, p$$

where $\boldsymbol{\Sigma} = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} = \tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{X}}, \quad \tilde{\boldsymbol{X}} = \boldsymbol{X}/\sqrt{n}$

非0の固有値 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$ $r = \mathrm{rank}\boldsymbol{X}$

非0固有値の分だけ以下の量が計算可能

$$\boldsymbol{u}_i = \frac{\tilde{\boldsymbol{X}}\boldsymbol{v}_i}{\sqrt{\lambda_i}}, \quad i = 1, 2, \ldots, r$$

#scalingされた主成分スコア

ノルムが1になっている

・行列形式で記述する　#簡単のためr=pを仮定

$$\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r)$$

$$\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r)$$

$$\tilde{\boldsymbol{D}} = \begin{pmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\lambda_r} \end{pmatrix}$$

$$\boldsymbol{U} = \left(\frac{\tilde{\boldsymbol{X}}\boldsymbol{v}_1}{\sqrt{\lambda_1}}, \ldots, \frac{\tilde{\boldsymbol{X}}\boldsymbol{v}_r}{\sqrt{\lambda_r}} \right) = \tilde{\boldsymbol{X}} \left(\boldsymbol{v}_1/\sqrt{\lambda_1}, \ldots, \boldsymbol{v}_r/\sqrt{\lambda_r} \right)$$

$$= \tilde{\boldsymbol{X}}\boldsymbol{V} \begin{pmatrix} 1/\sqrt{\lambda_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1/\sqrt{\lambda_r} \end{pmatrix} = \tilde{\boldsymbol{X}}\boldsymbol{V}\tilde{\boldsymbol{D}}^{-1}$$

$$\tilde{\boldsymbol{X}} = \boldsymbol{U}\tilde{\boldsymbol{D}}\boldsymbol{V}^T$$

where $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}$ is used

・SVD分解

$$\tilde{X} = U \quad \tilde{D} \quad V^T$$

$\tilde{X}$：中心化かつ1/√n倍されたデータ行列

U：各列がscalingされた主成分スコア

$\tilde{D}$：共分散行列の固有値の平方根

V^T：各行が共分散行列の固有ベクトル

第2主成分まで使う ➡ 2×2の対角行列DでXを近似

累積寄与率はXの近似の度合いを評価している！

Appendix

- **標本共分散行列の固有値と固有ベクトル**

$$\boldsymbol{\Sigma}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i, \quad i = 1, 2, \ldots, p$$

行列形式で記述する：

$$\boldsymbol{\Sigma V} = \boldsymbol{V\Lambda}$$

$$\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_p) \qquad \boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_p \end{pmatrix}$$

#Vは直交行列 $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}$

補足：対角化とスペクトル分解

対角化

$$\boldsymbol{V}^T\boldsymbol{\Sigma}\boldsymbol{V} = \boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_p \end{pmatrix}$$

#共分散行列を対角行列にする変換V

スペクトル分解

$$\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T = \sum_{i=1}^{p} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T$$

#p個の行列の和に分解