データ分析入門 第10回

主成分分析

京都大学 大学院情報学研究科 数理工学専攻/高度情報教育基盤コア

關戸 啓人

多変量解析 — 主成分分析

主成分分析の概要

- 主成分分析は，次元の縮約の観点から，新しい座標を構成するものである
  - 例えば，世界500都市の1時間おきの気温20年分のデータが有るとする
  - データ数は500×175320程度
  - このデータは別の指標から復元可能ではないか？
  - 都市の緯度，経度，人口密度，内陸度，…，などの別の座標を導入することで，全ての気温のデータを保存しなくても良い？
    - このような新しい座標の導入をデータのみから自動的に算出
    - 筋の良い座標の取り方がわかる
    - データ容量，計算量削減

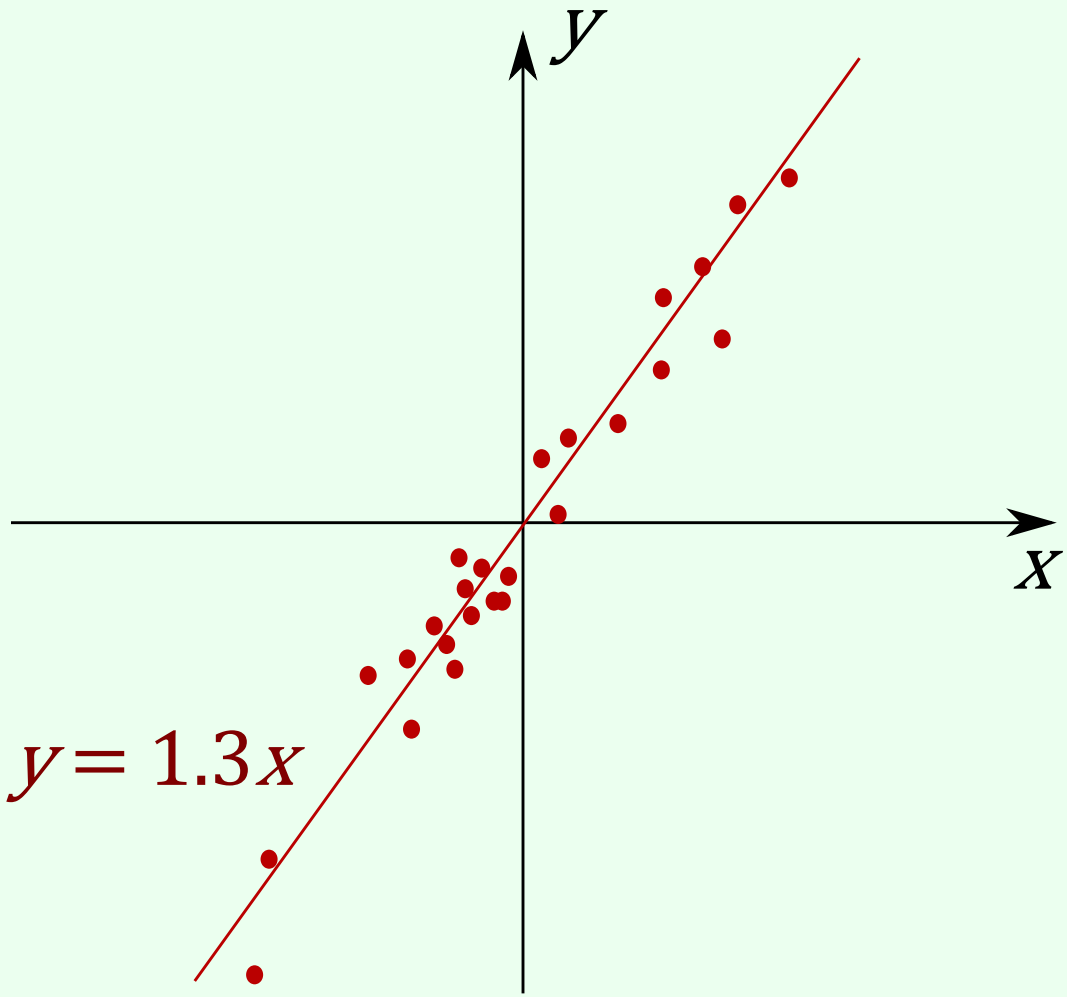
主成分分析の概要

主成分分析を考えるときは，登場する確率変数は全て平均が0になるように，定数を足したり引いたりしていると思おう．最初は，2変数の簡単な例で主成分分析の考え方を述べる．

確率変数Xは体重，Yは身長を表すとし，データ(x_k, y_k)が与えられたとしよう．ただし，データは平均が0であるように，平均体重，平均身長を引いたものが与えられる．プロットして（回帰分析などで）調べてみると，XとYには関係があって，近似的に以下となる（とする）．

$$Y = 1.3X$$

主成分分析の概要



主成分分析の概要

つまり，データ (x_k, y_k) は直線 $y = 1.3x$ の付近に散らばっているのであり，$(\alpha + \varepsilon, 1.3\alpha + \delta)$ で α を適当に取ると，ε や δ は小さいことが多い．そこには，（理論的に説明できるかどうかはわからないが）何らかの力が働いていると考えることができ，その何らかの力は，確率変数

$$Z = \frac{X + 1.3Y}{\sqrt{1^2 + 1.3^2}}$$

で表されるであろう．確率変数 X, Y を直交変換で Z, U に移すとしたら，U は

$$U = \frac{1.3X - Y}{\sqrt{1.3^2 + (-1)^2}}$$

となる．

主成分分析は，このように，確率変数を**直交変換**することである

主成分分析の概要

この例では，Zは体の大きさ，Uは肥満度を表しているように思える．どちらが重要な確率変数かは置いておいて，データの散らばりをより良く説明している確率変数はZである．$z_k = x_k + 1.3y_k$の値を見れば，大体のx_k, y_kの値がわかるという意味である．

このように，直交変換した後の確率変数で，元のデータを1番良く説明している変数を第1主成分，2番目により良く説明している変数を第2主成分，などと呼ぶ．

主成分分析は，次元の縮小に用いられる．直交変換した後の全ての確率変数を用いれば，元のデータは完全に復元できる．しかし，それなりに小さいkについて，第1主成分から第k主成分までのみを用いても，ほぼデータは復元できるようになる．よって，いくつかの主成分のみを考えても支障がなくなるからである．(支障が出ないように次元を減らす)

主成分の定義A

主成分の定義は2種類あるが，1つ目の定義を述べる．1つ目の定義での考え方は，ばらつきとは，分散である．そして，ばらつきをより良く説明する，とは分散が大きいことと考える．

元々の確率変数を $X_1, X_2, \ldots, X_n$ とすると，よって，この定義では，第1主成分 Z_1 は，

$$Z_1 = w_1 X_1 + w_2 X_2 + \cdots + w_n X_n,$$

$$w_1^2 + w_2^2 + \ldots + w_n^2 = 1$$

と書けるものの中で，最も分散が大きいものである．また，第 k 主成分は，上の形で書け，第 $k-1$ 主成分までと直交するものの中で，分散が最大となる確率変数となる．

主成分の定義B

2つ目の定義では，第1主成分 Z_1 を元々の変数との相関係数の2乗和を最大化する確率変数と取る．第 k 主成分は，同様に，第 $k-1$ 主成分までと直交する中で，元々の変数との相関係数の2乗和を最大化するように取る．

本講義では，定義Aを主に用いる．定義Aと定義Bでは，主成分が異なるものになるが，どちらも行列の固有値問題に帰着する．定義Aでは共分散行列，定義Bでは相関行列の固有値問題になる．

主成分の定義B

共分散行列の (i, j) 成分は，X_i と X_j の共分散

$$\mathrm{Cov}(X_i, X_j) = \mathrm{E}[(X_i - \mathrm{E}[X_i])(X_j - \mathrm{E}[X_j])]$$

であり，相関行列の (i, j) 成分は，X_i と X_j の相関係数

$$\frac{X_i \text{と} X_j \text{の共分散}}{(X_i \text{の標準偏差})(X_j \text{の標準偏差})}$$

である．相関係数は絶対値1以下となる．

確率変数 X_1 は分散が大きいが，確率変数 X_2 は分散が小さい，となれば全体の結果は確率変数 X_1 の影響が強くなる．これを防ぐため，各変数を正規化して考えたものが，相関行列を用いたものだと考えることができる．

主成分

以降，定義Ａで述べる．共分散行列の固有値を大きい順に

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$$

とする．また，λ_kに対応する固有ベクトルを

$$(w_{k,1}, w_{k,2}, \ldots, w_{k,n})^{\mathrm{T}}$$

とする．第k主成分は

$$Z_k = w_{k,1}X_1 + w_{k,2}X_2 + \cdots + w_{k,n}X_n$$

で，その分散はλ_kである．

証明は，例えば，ラグランジュの未定乗数法を用いる．（ここでは略）

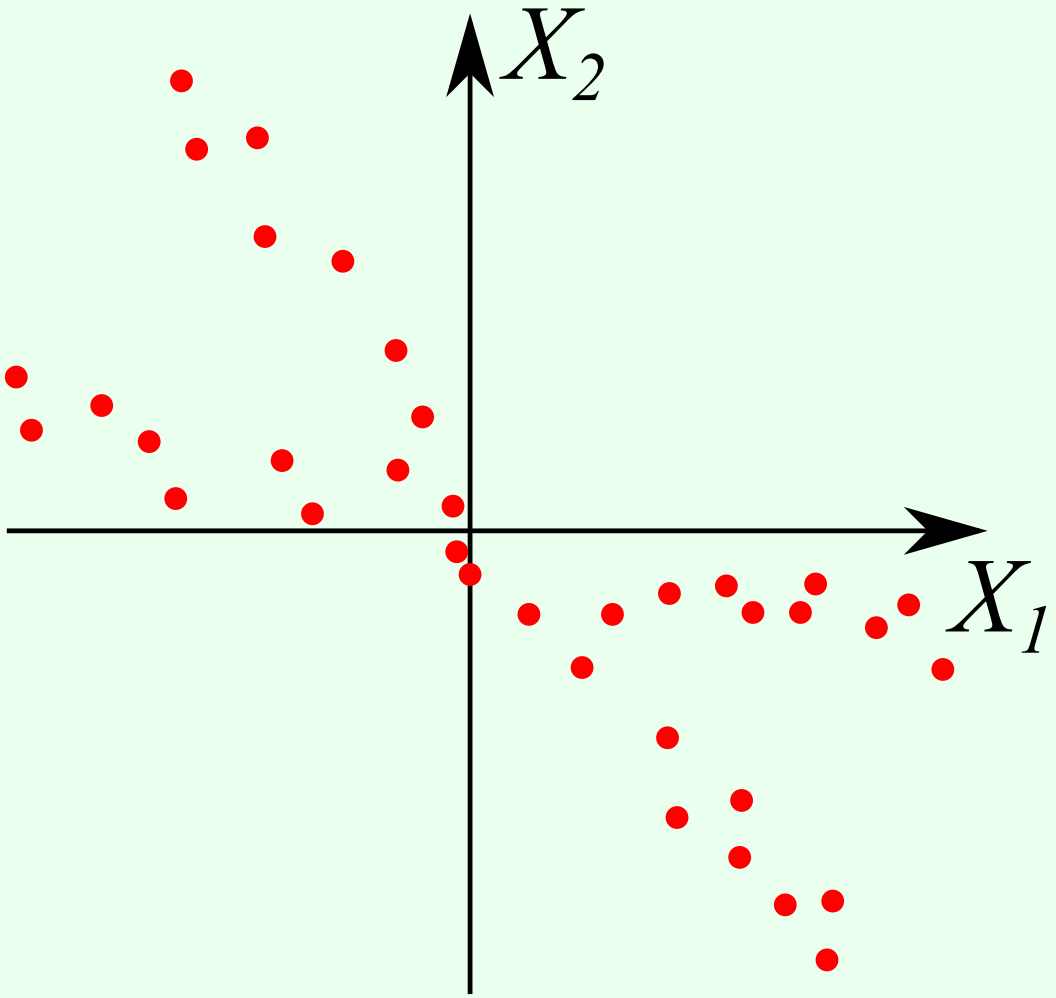
直交変換の意味

第$k-1$主成分まで取り出した時，第k主成分を，第$k-1$主成分までと直交するように選ぶ，というのは，第$k-1$主成分までと，無相関になるように選ぶ，というのと同値．

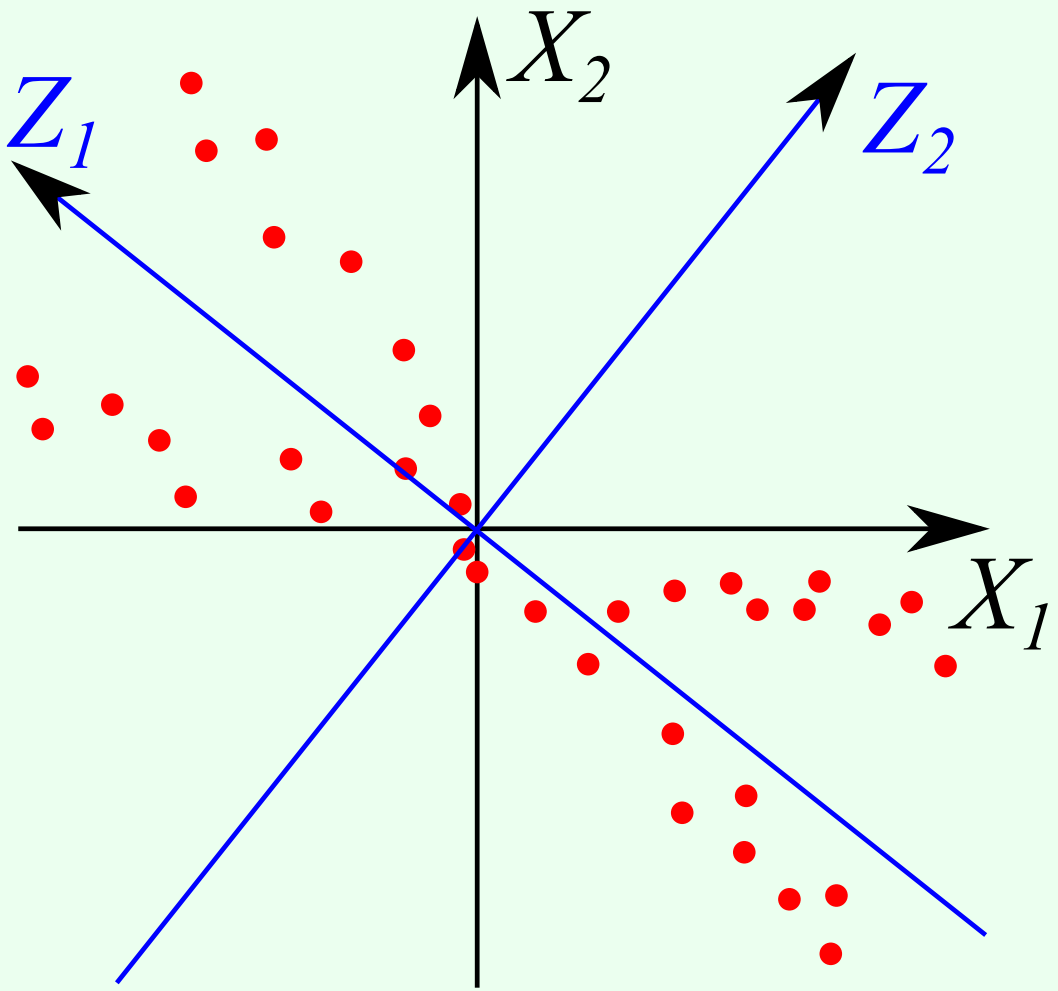
XとYは正の相関がある（相関係数が0より十分に大きい）というのは，Xが大きいと，Yも大きい傾向にあり，Xが小さいとYも小さい傾向がある．

XとYは負の相関がある（相関係数が0より十分に小さい）というのは，Xが大きいと，Yは小さい傾向にあり，Xが小さいとYは大きい傾向がある．

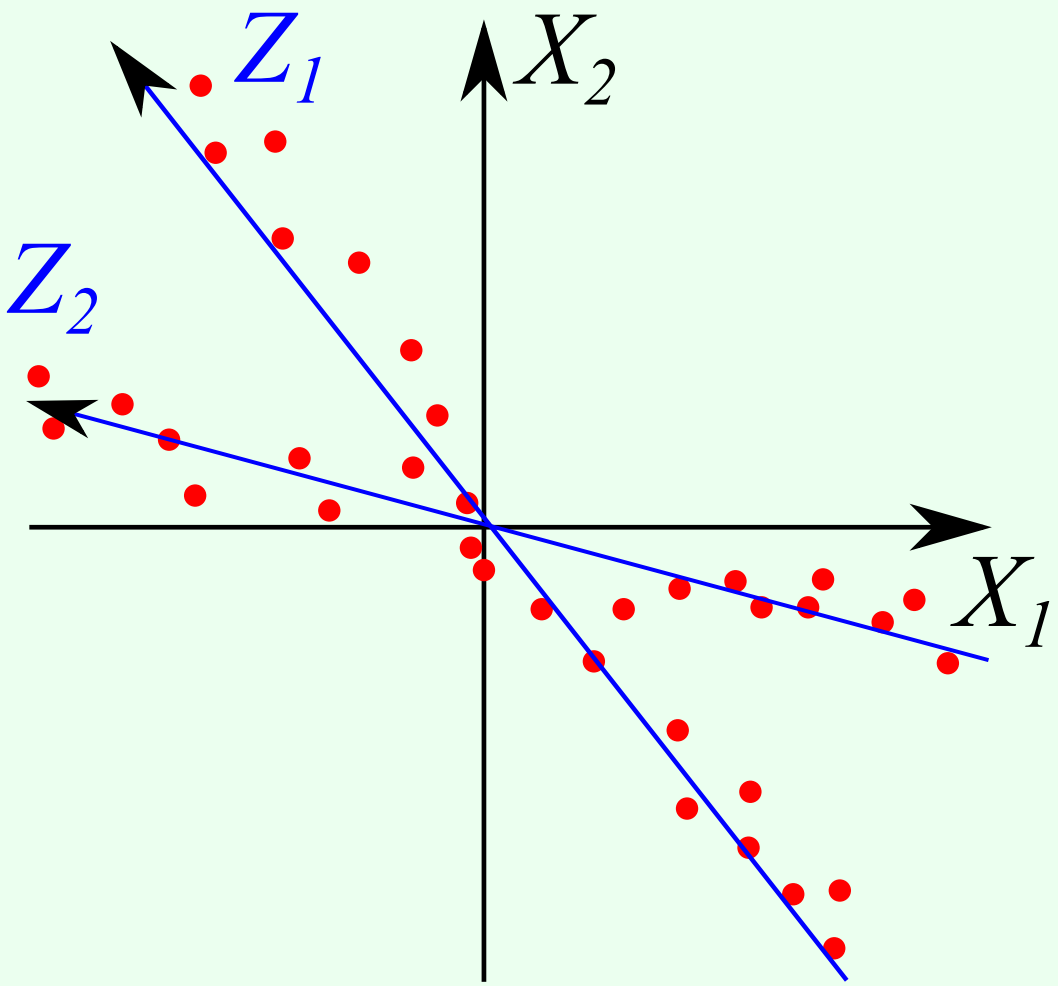
例：データ



例：主成分分析の結果



例：因子分析，独立成分分析などを用いると



例に関する補足

無相関であるということは，便利な面もたくさんある．無相関という制約がゆえに，隠れた要素を発見できないかもしれない．

データについて，個々の構成要素を得ようとする方法として，別の方法で，因子分析がある．因子分析のやり方は，いろいろな定義があり，それぞれ結果も一致しない．

他には，例えば，独立成分分析などもある．独立成分分析では，各確率変数ができるだけ独立になるように定める．これも，いろいろな定義がある．

対して，主成分分析は，少ない主成分でデータを説明する，データの総合的なスコアを定める，ということに特化している．

データと分散

確率変数の数を n，標本数を m として，標本

$$(x_{k,1}, x_{k,2}, \ldots, x_{k,n}),\ 1 \leq k \leq m$$

を考える．この時，X_i の標本平均は

$$\sum_{k=1}^{m} x_{k,i} = 0$$

とする．X_i の不偏分散は

$$\frac{1}{m-1} \sum_{k=1}^{m} x_{k,i}^2$$

であった．また，X_i と X_j の不変共分散は

$$\frac{1}{m-1} \sum_{k=1}^{m} x_{k,i} x_{k,j}$$

である．

データと分散

データ行列を

$$A = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{pmatrix} \in M_{m,n}(\mathbb{R})$$

とすると，不偏共分散行列は

$$A^{\mathrm{T}} A/(m-1)$$

と書ける．したがって，主成分を求めるには，

共分散行列の（大きい方から数個の）固有値と固有ベクトルを求める

または

データ行列の（大きい方から数個の）特異値と右特異ベクトルを求める

を行うことによって，求めることができる．

データと分散

元々の確率変数での分散の和と，主成分での分散の和は等しい．つまり

$$\sum_{i=1}^{n}\sum_{k=1}^{m} x_{k,i}^2 = \sum_{i=1}^{n}\sum_{k=1}^{m} z_{k,i}^2 = \mathrm{tr}(A^{\mathrm{T}}A)$$ (直交変換だから)

となる．主成分 Z_i の寄与率を

$$\sum_{k=1}^{m} z_{k,i}^2 \Big/ \sum_{i=1}^{n}\sum_{k=1}^{m} x_{k,i}^2$$

主成分 $Z_1, Z_2, \ldots, Z_s$ の累積寄与率は

$$\sum_{i=1}^{s}\sum_{k=1}^{m} z_{k,i}^2 \Big/ \sum_{i=1}^{n}\sum_{k=1}^{m} x_{k,i}^2$$

で定義される．

累積寄与率が，ある程度大きくなるように，使用する主成分の数を決めることが多い．

（補足）無相関

主成分 Z_i と Z_j の不偏共分散は（$i \neq j$）

$$\frac{1}{m-1}\sum_{k=1}^{m}(w_{i,1}x_{k,1}+w_{i,2}x_{k,2}+\cdots+w_{i,n}x_{k,n})(w_{j,1}x_{k,1}+\cdots+w_{j,n}x_{k,n})$$

$$=\frac{1}{m-1}w_i^{\mathrm{T}}A^{\mathrm{T}}Aw_j$$

$$=\frac{1}{m-1}\lambda_j w_i^{\mathrm{T}}w_j=0$$

となる．ここで，$w_k=(w_{k,1},w_{k,2},\ldots,w_{k,n})^{\mathrm{T}}$．

$A=UDV^{\mathrm{T}}$ とすると，主成分からなるデータ行列は $AV=UD$ で

$\mathrm{Cov}(AV)=\mathrm{Cov}(UD)=(UD)^{\mathrm{T}}UD=D^{\mathrm{T}}U^{\mathrm{T}}UD=D^{\mathrm{T}}D$ （対角行列）

ここでは，分散を最大化する方針で主成分を定義したが，「直交変換である，かつ，無相関にする」という方針でも同じ結果が得られる．

（補足）残差最小化

- 第k主成分のみを用いてデータ行列を復元した時
  - 復元されたデータ行列を$\tilde{X} = (\tilde{x}_{i,j})$とすると
  - $\sum (x_{i,j} - \tilde{x}_{i,j})^2$が最小化されている
- 主成分分析はデータ行列をできるだけ良く近似するように次元の縮小を行っている
- （これを定義と思っても主成分分析が得られる）

主成分分析の概略：まとめ

- 主成分分析とは，数学的には確率変数の直交変換
  - $X_1, X_2, \ldots, X_n$ から $Z_1, Z_2, \ldots, Z_n$ に変換
  - $Z_1, Z_2, \ldots, Z_n$ は無相関，分散は $V[Z_1] \geq V[Z_2] \geq \cdots \geq V[Z_n]$
- 主成分分析は，次元の縮約に用いられる
  - データのばらつきを説明するには，大きい方からいくつかの主成分だけで十分かもしれない．累積寄与率を参考にする
- 主成分は，データの裏に隠れた要素，要因を表しているかも
  - 希望するものが得られているかはわからない，説明がつくとも限らない
  - そういうものの解析をしたいのなら，因子分析，独立成分分析なども視野にいれる
- 計算方法は，共分散行列の固有値分解，または，データ行列の特異値分解
  - 実際には，固有値，あるいは，特異値の大きい方から数個だけ必要

R言語

R言語の特徴

- フリーウェア，無償
- オープンソフトウェア，個人が開発，改良できる
- プログラミング言語で，自由にロジックが記述可能，汎用性が高い
- 行列計算，ベクトル計算は速い．統計処理だけなら速い
- 統計処理するには，関数などを書かねばならず，とっつきにくい（Rコマンダーなどの GUI もある）
- 行列計算以外の処理などは速くない

その他との比較

- Excel
  - 直感的な操作が可能
  - 有償（そこまで高価ではなく，インストールされているPCも多い）
  - 限られた処理しかできない（VBAで多少補える，また有償でアドインが販売されていることもある）
- 有償ソフトウェア（SASなど）
  - 性能は様々．基本的に良い．汎用性は様々
  - 有償で，一般的にかなり高価
- C言語など
  - 無償，汎用性が高い．うまくプログラミングすれば高速
  - 開発コストがとても高い

変数，代入，演算

変数は代入されることによって定義される．名前は，アルファベット，数字からなり，最初の文字は数字であってはならない．名前は，大文字，小文字は区別される．また，以下の名前を使用することはできない：`break`，`else`，`for`，`function`，`if`，`in`，`next`，`repeat`，`return`，`while`，`TRUE`，`FALSE`．

代入するには，`<-`を用いる（他にも，`->`，`=`，`assign`があるが，通常`<-`だけで十分である）．例えば，

```
x <- 1
```

とすると，変数`x`に`1`が代入される．

演算は，`+`が足し算，`-`が引き算，`*`が掛け算，`/`が割り算，`**`または`^`がべき乗を表す．例えば，

```
2/3
```

は`0.6666667`となり，

```
2**4
```

は`16`となる．

ベクトル

ベクトル(2,4,7)を変数yに代入するには，

```
y <- c(2, 4, 7)
```

と書く．ベクトル同士の足し算などもでき，

```
c(2, 4, 7) + c(5, 4, 3)
```

とすると，ベクトル(7,8,10)となり，

```
c(2, 4, 7) * c(5, 4, 3)
```

とすると，ベクトル(10,16,21)となる．(各要素ごとに掛け算された)

長さが違うベクトル同士の演算は，長さが短いベクトルが周期的に長さが何倍かに拡張されて行われる．例えば，

```
c(1,2,3,4,5,6) + c(100,1000)
```

は(101,1002,103,1004,105,1006)となるが，

```
c(1,2,3,4,5,6,7) + c(100,1000)
```

はエラーになる．(「長いオブジェクトの長さが短いオブジェクトの長さの倍数になっていません」といわれる)

ベクトル

スカラー変数は，長さ1のベクトルと思って計算される．例えば，

```
c(1, 2, 3, 4) * 2
```

は(2,4,6,8)に，

```
c(1, 2, 3, 4) + 2
```

は(3,4,5,6)に，

```
c(1, 2, 3, 4) ** 2
```

は(1,4,9,16)となる．

`c(2:6)`は`c(2,3,4,5,6)`と同じです．ベクトル`x`の長さを取得するには`length(x)`を用います．ベクトル`x`の`k`番目の要素は`x[k]`で参照できます（最初の要素は1番目の要素です）．2番目の要素から，4番目の要素を取り出すには，`x[2:4]`で取り出せます．ベクトル`x`とベクトル`y`をつなげて，長いベクトルを作るには，`append(x,y)`とします．ベクトル`x`の中で，3より大きい要素だけを取り出すには`x[x>3]`と書きます．これは，要素数5のベクトル`x`の中で，2番目と4番目の要素だけ取り出すとき，`x[c(FALSE,TRUE,FALSE,TRUE,FALSE)]`と書くことができ，それの省略形になります．

行列

行列を作るには，

`matrix(ベクトル, nrow=行数, ncol=列数, byrow=TRUE)`

とすれば，ベクトルの各要素を，(1,1) 成分，(1,2) 成分，と埋めていった行列が作られる．
`byrow=FALSE` **とすると，列について連続になるように埋めていく．**

行列についても，演算は各要素ごとに行われる．行列積には `%*%` **を用いる．**

行列 `x` **の2行目を取り出してベクトルにするときは** `x[2,]`**，2列目を取り出してベクトルにするときは** `x[,2]` **とする．**

行列 `x` **の各行の平均を求めるときは** `apply(mean, 1, x)`**，各列の平均を求めるときは** `apply(mean, 2, x)` **とする．**

行列

例

```
> x <- matrix(c(1,2,3,4),nrow=2,ncol=2,byrow=TRUE)
> x
     [,1] [,2]
[1,]    1    2
[2,]    3    4
> y <- matrix(c(2,3,4,5),nrow=2,ncol=2,byrow=TRUE)
> x * y
     [,1] [,2]
[1,]    2    6
[2,]   12   20
> x %*% y
     [,1] [,2]
[1,]   10   13
[2,]   22   29
```

read.table

read.tableはデータを表として読み込む関数である．もっともシンプルな形は，

```
x <- read.table("hoge.txt")
```

とすれば，hoge.txtに記述されている内容が，列に対してはスペース区切りで表（データフレーム型）として，xに代入される．オプションもいろいろ有り，例えば，

```
x <- read.table("hoge.csv", header=TRUE, sep=",")
```

の意味は，header=TRUEは1行目は列のラベルが書かれているという意味であり，sep=","は列の区切り文字がコンマという意味である．

また，csvファイルを読み込むときには，read.csvという関数が用意されている．

補足：リストは，どんな型でも良い変数の配列で，データフレーム型はリストの2次元版で，どんな方でも良い変数の行列を表す型である．

条件分岐

条件分岐の書き方は以下のとおりです．

```
if(条件文){
  処理
}else if(条件文){
  処理
}else if(条件文){
  処理
}else{
  処理
}
```

forループ

forを用いてループ処理するには，以下のようにします．

```
for(変数名 in ベクトル){
  処理 (変数にベクトルの各々の要素が代入されてこの処理がされる)
}
```

例えば，変数iについて，1から100までループを回すには，

```
for(i in c(1:100)){
  処理
}
```

と書くことができます．

関数

関数（プログラムの塊）を作るには，以下のようにします．

```
関数名 <- function(引数){
  処理
  return(関数が返す値)
}
```

例えば，BMIを求める関数は，

```
BMI <- function(h, w){
  return( w/((h/100)^2) )
}
```

と定義し，

```
BMI(170,60)
```

と呼び出すことができます．(20.76125が得られます)

回帰分析

Rで回帰分析（未知パラメータが線形の場合）を行うには，`lm()`関数を用いる．（余談だが，非線形回帰分析の場合は，`nls()`関数が用意されている）

最も簡単な`lm()`関数の使い方は，従属変数Yに対応するデータの入ったベクトルを`y`，説明変数X_kに対応するデータが入ったベクトルを`x_k`として，

$$Y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_n X_n + \varepsilon$$

というモデルで回帰分析する場合は，`lm(y ~ x_1 + x_2 + ... + x_n)`と打てば良い．

回帰分析した詳細な結果は`lm()`関数が返す．それは，`summary()`関数や，`plot()`関数に渡すことで見ることができる．

主成分分析

Rで主成分分析するには，`prcomp`関数を用いる．確率変数$X_1, X_2, \ldots, X_n$に対応するデータのベクトルを，`x1, x2, ..., xn`とすると，

```
prcomp(~x1+x2+...+xn)
```

で主成分分析できる．また，定義B（相関係数行列を用いる方法）で主成分分析する場合は

```
prcomp(~x1+x2+...+xn, scale=TRUE)
```

とする．データフレーム型`x`に含まれる全てのデータを用いて主成分分析する場合は

```
prcomp(x)
```

とも書ける．

主成分分析

寄与率，累積寄与率は`summary`で確認できる．

```
res <- prcomp(data)

summary(res)
```

といった感じである．