

「データ解析」（下平英寿）　講義資料８

モデル選択

- 目標： モデル選択について理解する．
  1. 重回帰分析におけるモデル選択（変数選択）
  2. ブートストラップ法の適用

1　重回帰分析におけるモデル選択

1.1　t-検定

- 回帰係数の t-検定によって $\beta_i = 0$ と判定される説明変数 x_i を取り除く．

```
> dat <- read.table("dat0002.txt")
> dim(dat)
[1] 47 10
> names(dat)
 [1] "Zouka"   "Ninzu"   "Kaku"    "Tomo"    "Tandoku" "X65Sai"  "Kfufu"
 [8] "Ktan"    "Konin"   "Rikon"
> fit <- lm(Zouka~.,dat)
> summary(fit)

Call:
lm(formula = Zouka ~ ., data = dat)

Residuals:
      Min        1Q    Median        3Q       Max
-0.076623 -0.020252 -0.002137  0.021601  0.062239

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.886752   1.333787  -5.913 8.23e-07 ***
Ninzu        1.691499   0.192037   8.808 1.30e-10 ***
Kaku         0.024427   0.009567   2.553 0.014932 *
Tomo         0.002694   0.002754   0.978 0.334421
Tandoku      0.055622   0.012759   4.359 9.99e-05 ***
X65Sai      -0.022099   0.009466  -2.335 0.025103 *
Kfufu        0.051202   0.017113   2.992 0.004913 **
Ktan        -0.009525   0.015900  -0.599 0.552805
Konin        0.107444   0.026969   3.984 0.000306 ***
```



```
Rikon        0.108355   0.042978   2.521 0.016136 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03808 on 37 degrees of freedom
Multiple R-Squared: 0.964,Adjusted R-squared: 0.9553
F-statistic: 110.1 on 9 and 37 DF,  p-value: < 2.2e-16
```

- 説明変数は $p = 9$ 個．
- t-検定の p-値が 0.05 以上になる回帰係数は
  1. β_3（Tomo）：p 値=0.33
  2. β_7（Ktan）：p 値=0.55
- これら二つの説明変数は，取り除いても良いと判断する．
- これら二つを取り除いた残りの 7 個の説明変数は，取り除けないと判断する．
- もしひとつだけ取り除くなら，x_7（Ktan）．次に x_3 （Tomo）．
- lm() の更新を行うには，update() が便利．

```
> fit1 <- update(fit,~ .- Ktan) # まず x3 を取り除く（beta3=0 と判断）
> summary(fit1)

Call:
lm(formula = Zouka ~ Ninzu + Kaku + Tomo + Tandoku + X65Sai +
    Kfufu + Konin + Rikon, data = dat)

Residuals:
      Min        1Q    Median        3Q       Max
-0.076112 -0.020354 -0.004012  0.025239  0.062728

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.305754   0.907817  -8.048 9.87e-10 ***
Ninzu        1.659653   0.182968   9.071 4.78e-11 ***
Kaku         0.020128   0.006274   3.208 0.002713 **
Tomo         0.002654   0.002730   0.972 0.337131
Tandoku      0.049836   0.008266   6.029 5.20e-07 ***
X65Sai      -0.026227   0.006435  -4.076 0.000226 ***
Kfufu        0.050457   0.016923   2.982 0.004984 **
Konin        0.108725   0.026656   4.079 0.000223 ***
Rikon        0.100741   0.040708   2.475 0.017912 *
```



```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03776 on 38 degrees of freedom
Multiple R-Squared: 0.9637,Adjusted R-squared: 0.956
F-statistic:   126 on 8 and 38 DF,  p-value: < 2.2e-16

> fit2 <- update(fit1,~ .- Tomo) # さらに x7 を取り除く（beta7=0 と判断）
> summary(fit2)

Call:
lm(formula = Zouka ~ Ninzu + Kaku + Tandoku + X65Sai + Kfufu +
    Konin + Rikon, data = dat)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0744469 -0.0223745 -0.0008725  0.0252404  0.0645647

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.089839   0.879607  -8.060 7.87e-10 ***
Ninzu        1.675926   0.182072   9.205 2.53e-11 ***
Kaku         0.017622   0.005716   3.083 0.003754 **
Tandoku      0.047635   0.007944   5.996 5.24e-07 ***
X65Sai      -0.026829   0.006401  -4.192 0.000154 ***
Kfufu        0.054838   0.016301   3.364 0.001733 **
Konin        0.114131   0.026051   4.381 8.62e-05 ***
Rikon        0.091775   0.039622   2.316 0.025888 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03773 on 39 degrees of freedom
Multiple R-Squared: 0.9628,Adjusted R-squared: 0.9561
F-statistic: 144.1 on 7 and 39 DF,  p-value: < 2.2e-16
```

1.2　説明変数の選択

- p 変数の重回帰モデルを仮定する

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

説明変数の添え字は $i \in \{1, \ldots, p\}$ である．

- 部分集合 $S \subset \{1, \ldots, p\}$ に対して，モデル S を

$$y = \beta_0 + \sum_{i \in S} \beta_i x_i + \epsilon$$

で定義する．言い換えると，モデル S は，

$$\beta_i = 0, \quad i \in \{1, \ldots, p\} \setminus S$$

ということ．モデル S はモデル $\{1, \ldots, p\}$ の部分モデルである．

- たとえば，$p = 5$，$S = \{1, 3\}$ とおく．モデル $\{1, \ldots, 5\}$ は

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \epsilon$$

モデル $\{1, 3\}$ は

$$y = \beta_0 + \beta_1 x_1 + \beta_3 x_3 + \epsilon$$

で言い換えると

$$\beta_2 = \beta_4 = \beta_5 = 0$$

- ここでは適切な S を選択することを考える．これは説明変数の組み合わせを選択しているので，「変数選択」ともいう．部分集合 S を選択すると言う意味で，subset selection ということもある．
- 不必要な変数を減らすことにより，推定精度や予測精度を高めることができる．
- 適切な S を探すアルゴリズム
  1. 総当り法：可能な S の組み合わせは 2^p 個である．たとえば $p = 10$ ならば $2^{10} = 1024$ である．このすべての組み合わせについてモデルの良さをチェックし，最適な S を選ぶ．($S =$ 空集合 を探索範囲から除くこともある．)
  2. 逐次選択法 (stepwise 法)：初期モデルからスタートし，変数を増やしたり減らしたりしながら適切な S を見つけ出す．変数増減法などとも言う．いろいろなバージョンがある．(小さなモデルから始めて増やしていく．大きなモデルから始めて減らしていく．増やすだけ／減らすだけ／両方などのオプションがある)
  3. ただし，必ずしもすべての部分集合を探索範囲に含めるべきではない．たとえば多項式回帰の次数選択では，次数$=0, 1, \ldots, p$ の p 個のモデルだけを調べるべき．
- 良い S かどうかを判定するアルゴリズム．上記いずれの方針で S を探索するにしても，個々の S が良いのか悪いのかを相対的に判断する基準が必要である．
  1. 検定を用いる．(t-検定，F-検定等)
  2. モデルの良さを測る規準を用いる．(修正決定係数，ＡＩＣ等)

1.3 AICを最小にするモデルを選ぶ（逐次選択法）

- 逐次選択法法で最大モデルから変数を減らしたり増やしたりして探索する．
- モデルのよさの規準として，AICを用いる．AIC=Akaike Information Criterion （赤池情報量規準）である．この値が小さいモデルほど，良いモデル．
- これを実行するには，step() 関数を用いる．

```
> dat <- read.table("dat0002.txt")
> fit <- lm(Zouka~.,dat)
> fitstep <- step(fit)
Start:  AIC= -298.44
 Zouka ~ Ninzu + Kaku + Tomo + Tandoku + X65Sai + Kfufu + Ktan +
    Konin + Rikon

          Df Sum of Sq      RSS      AIC
- Ktan     1     0.001    0.054 -299.987
- Tomo     1     0.001    0.055 -299.241
<none>                    0.054 -298.440
- X65Sai   1     0.008    0.062 -293.982
- Rikon    1     0.009    0.063 -292.989
- Kaku     1     0.009    0.063 -292.814
- Kfufu    1     0.013    0.067 -290.257
- Konin    1     0.023    0.077 -283.663
- Tandoku  1     0.028    0.081 -280.959
- Ninzu    1     0.113    0.166 -247.312

Step:  AIC= -299.99
 Zouka ~ Ninzu + Kaku + Tomo + Tandoku + X65Sai + Kfufu + Konin +
    Rikon

          Df Sum of Sq      RSS      AIC
- Tomo     1     0.001    0.056 -300.832
<none>                    0.054 -299.987
- Rikon    1     0.009    0.063 -294.964
- Kfufu    1     0.013    0.067 -292.107
- Kaku     1     0.015    0.069 -290.722
- X65Sai   1     0.024    0.078 -284.943
- Konin    1     0.024    0.078 -284.920
- Tandoku  1     0.052    0.106 -270.442
- Ninzu    1     0.117    0.171 -247.832

Step:  AIC= -300.83
```



```
Zouka ~ Ninzu + Kaku + Tandoku + X65Sai + Kfufu + Konin + Rikon

          Df Sum of Sq      RSS      AIC
<none>                    0.056 -300.832
- Rikon    1     0.008    0.063 -296.774
- Kaku     1     0.014    0.069 -292.582
- Kfufu    1     0.016    0.072 -290.857
- X65Sai   1     0.025    0.081 -285.353
- Konin    1     0.027    0.083 -284.022
- Tandoku  1     0.051    0.107 -272.127
- Ninzu    1     0.121    0.176 -248.570

> extractAIC(fit)
[1]   10.0000 -298.4404
> n <- 47; p <- 10
> n*log(sum(resid(fit)^2)/n) + 2*p # stepAIC() の計算する AIC 値の定義
[1] -298.4404
> extractAIC(fitstep)
[1]    8.0000 -300.8322
> extractAIC(fitstep) - extractAIC(fit) # AIC の変化
[1] -2.000000 -2.391807
> AIC(fit)
[1] -163.0602
> n*(1+log(2*pi*sum(resid(fit)^2)/n)) + 2*(p+1) # AIC() の計算する AIC 値の定義
[1] -163.0602
> AIC(fitstep)
[1] -165.452
> AIC(fitstep) - AIC(fit)  # AIC の変化
[1] -2.391807
```

- まず x_7（Ktan）が取り除かれ，次に x_3(Tomo) が取り除かれた．ここで終了．(どの変数をひとつだけ引いても足しても，これ以上ＡＩＣ値は改善しない.)　つまり局所最適解.
- 結局，t-検定のときと同じ結果になった．(いつもこうなるとは限らない.)
- 一般にパラメタベクトル θ，その次元 $\dim\theta$，対数尤度 $\ell(\theta)$ とかくと，ＡＩＣは次式で定義される

$$\mathrm{AIC} = -2\times\ell(\hat{\theta}) + 2\times\dim\theta$$

- AIC の第１項にある対数尤度 $\ell(\hat{\theta})$ は，一般にモデルの次元を大きくするほど，あてはまりが良くなり，対数尤度が大きくなる．したがって，次元を大きくすれば，AIC の第１項は小さくなる．ところが，次元を大きくすれば AIC の第２項が大きくなる．(モデルの次元＝モデルの複雑さと解釈できることに注意.)　AIC では，モデルの当てはまりのよさと複雑さの両者を考慮して最適なモデルを選ぶための規準である．

- 重回帰モデルの場合，

$$\theta = (\beta_0, \beta_1, \ldots, \beta_p, \sigma), \quad \dim\theta = p+1$$

$$\ell(\hat{\theta}) = -\frac{n}{2}\left\{1 + \log(2\pi\hat{\sigma}^2)\right\}$$

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} e_i^2}{n}$$

であるから，

$$\mathrm{AIC} = n\times\left\{1+\log(2\pi\hat{\sigma}^2)\right\} + 2\times(p+1)$$

である．

- AIC 値は AIC() で計算できる．
- step() では，内部で extractAIC() 関数を呼び出して AIC 値を計算している．extractAIC() は次式を計算する

$$n\log(\hat{\sigma}^2) + 2p$$

つまり

$$\mathrm{AIC} - n\times(1+\log 2\pi) - 2$$

- AIC() と extractAIC() の差は S に依存しないので，モデル選択における AIC 値の変化は，どちらで計算しても同じ．

1.4 AICを最小にするモデルを選ぶ（総当り法）

```
# run0082.R
# AIC最小モデル：総当り法
func0082a <- function(p) { # p個の説明変数のすべての組み合わせを生成
  m <- 2^p # モデルの数
  wh <- matrix(logical(m*p),m)
  for(i in 1:p) {
    k <- 2^(i-1)
    wh[,i] <- c(rep(F,k),rep(T,k))
  }
  na <- apply(wh,1,function(v) paste((1:p)[v],collapse="")) # 名前
  na <- paste("(",na,")",sep="")
  dimnames(wh) <- list(na,1:p)
  wh
}
func0082b <- function(x,y,wh) { # x=説明変数の行列，y=目的変数のベクトル
  x <- as.matrix(x); y <- as.vector(y)
  x <- cbind(1,x) # = X（第１列は１のベクトル）
  wh <- cbind(T,wh) # = モデルに含める説明変数の定義（beta0 は常にモデルに入れる）
  xx <- t(x) %*% x # = X'X をあらかじめ計算しておいたほうが，多少節約になる
  xy <- t(x) %*% y # = X'y をあらかじめ計算しておいたほうが，多少節約になる
  m <- nrow(wh) # モデルの数
  aics <- rep(0,m) # AIC 値を収納するアレイ
  names(aics) <- rownames(wh) # モデルの名前
  n <- nrow(x) # データサイズ
  for(i in 1:m) {
    w <- wh[i,] # モデルを定義する論理型ベクトル
    pw <- sum(w) - 1 # 説明変数の数
    xxw <- xx[w,w,drop=F]; xyw <- xy[w,drop=F] # 部分を取り出す
    bew <- solve(xxw) %*% xyw # 回帰係数
    predw <- as.vector(x[,w] %*% bew) # 予測値
    residw <- y - predw # 残差
    rssw <- sum(residw^2) # RSS
    aicw <- n*(1+log(2*pi*rssw/n)) + 2*(pw+2) # AIC
    aics[i] <- aicw # これだけ保存しておく
  }
  aics
}
```



```
> source("run0082.R")
> dat <- read.table("dat0002.txt")
> x <- dat[,-1]; y <- dat[,1] # y=Zouka, x=残りの９個の説明変数
> wh <- func0082a(p) # まず 512 個のモデルを生成
> dim(wh)
[1] 512   9
> wh[1:10,] # 最初の１０個のモデル
         1     2     3     4     5     6     7     8     9
()    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
(1)    TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
(2)   FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
(12)   TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
(3)   FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
(13)   TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
(23)  FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
(123)  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
(4)   FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
(14)   TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
> aics <- func0082b(x,y,wh) # すべてのモデルの AIC 値を計算
> sort(aics)[1:10] # AIC を小さい順に並べ替え，最初の１０個を表示
  (1245689)  (12345689)  (12456789) (123456789)    (124568)    (1234568)
 -165.4520   -164.6066   -163.8606   -163.0602   -161.3940   -159.5837
```

```
  (1245678)   (1456789)   (1246789)  (12346789)
  -159.4135   -159.3666   -159.0927   -158.6019
> order(aics)[1:10] # ＡＩＣ最小の１０個のモデル番号
 [1] 444 448 508 512 188 192 252 506 492 496
> wh[order(aics)[1:10],] # それらのモデル定義
                1     2     3     4     5     6     7     8     9
(1245689)   TRUE  TRUE FALSE TRUE  TRUE TRUE FALSE TRUE  TRUE
(12345689)  TRUE  TRUE  TRUE TRUE  TRUE TRUE FALSE TRUE  TRUE
(12456789)  TRUE  TRUE FALSE TRUE  TRUE TRUE  TRUE TRUE  TRUE
(123456789) TRUE  TRUE  TRUE TRUE  TRUE TRUE  TRUE TRUE  TRUE
(124568)    TRUE  TRUE FALSE TRUE  TRUE TRUE FALSE TRUE FALSE
(1234568)   TRUE  TRUE  TRUE TRUE  TRUE TRUE FALSE TRUE FALSE
(1245678)   TRUE  TRUE FALSE TRUE  TRUE TRUE  TRUE TRUE FALSE
(1456789)   TRUE FALSE FALSE TRUE  TRUE TRUE  TRUE TRUE  TRUE
(1246789)   TRUE  TRUE FALSE TRUE FALSE TRUE  TRUE TRUE  TRUE
(12346789)  TRUE  TRUE  TRUE TRUE FALSE TRUE  TRUE TRUE  TRUE
```

- x_3(Tomo) と x_7（Ktan）が取り除かれた.（すべての組みあわせ中で，これがＡＩＣ最小）
- 結局，逐次選択法法，t-検定のときと同じ結果になった.（いつもこうなるとは限らない.）

1.5 AIC を最小にするモデルを選ぶ（総当り法の改良）

- 説明変数の数 p が大きくなると，モデルの総数 2^p は爆発的に増える.
- じつはすべてのモデルで AIC 値を計算しなくても，結果として AIC 最小モデルを見つけられるアルゴリズムがある.「分枝限定法」をモデル探索に適用すればよい.
- 分枝限定法では，探索木を展開する途中で，明らかに最適解が含まれない枝を切り落とす．結果として，総当たり法と同じ結果を出すことができるが，その計算時間はかなり短縮される．計算量の改善率は，どれだけ枝を切り落とせるかということであるが，これは具体的なデータに依存する.
- library(leaps) の regsubsets() を利用する.

```
> library(leaps) # このライブラリは標準ではない.
> dat <- read.table("dat0002.txt")
> rg <- regsubsets(Zouka~.,dat,nvmax=100) # nvmax は説明変数の個数の最大値
> srg <- summary(rg) # 結果を整理
> srg$which # 説明変数の個数=1,...,9で，それぞれベストなモデル
  (Intercept) Ninzu  Kaku  Tomo Tandoku X65Sai Kfufu  Ktan Konin Rikon
1        TRUE FALSE FALSE FALSE   FALSE  FALSE FALSE FALSE  TRUE FALSE
2        TRUE  TRUE FALSE FALSE   FALSE   TRUE FALSE FALSE FALSE FALSE
3        TRUE  TRUE FALSE FALSE   FALSE   TRUE FALSE  TRUE FALSE FALSE
```



```
4        TRUE  TRUE  TRUE FALSE    TRUE  FALSE FALSE FALSE  TRUE FALSE
5        TRUE  TRUE  TRUE FALSE    TRUE   TRUE FALSE FALSE  TRUE FALSE
6        TRUE  TRUE  TRUE FALSE    TRUE   TRUE  TRUE FALSE  TRUE FALSE
7        TRUE  TRUE  TRUE FALSE    TRUE   TRUE  TRUE FALSE  TRUE  TRUE
8        TRUE  TRUE  TRUE  TRUE    TRUE   TRUE  TRUE FALSE  TRUE  TRUE
9        TRUE  TRUE  TRUE  TRUE    TRUE   TRUE  TRUE  TRUE  TRUE  TRUE
> srg$rss # これらの RSS
[1] 0.42450489 0.17547268 0.13754637 0.08168998 0.07258712 0.06316087 0.05552269
[8] 0.05417537 0.05365501
> n <- nrow(dat)
> p <- nrow(srg$which)
> aicrg <- 2*(n/2)*(1+log(2*pi*srg$rss/n)) + 2*(2 + (1:p))  # AIC 値の計算
> aicrg # 説明変数の個数=1,...,9で，それぞれベストなモデルの AIC 値
[1]  -81.8478 -121.3695 -130.8150 -153.3034 -156.8562 -161.3940 -165.4520
[8] -164.6066 -163.0602
> order(aicrg) # AIC 値を小さくする順番に並べ替える
[1] 7 8 9 6 5 4 3 2 1
> srg$which[7,] # AIC 最小モデル
(Intercept)       Ninzu        Kaku        Tomo     Tandoku      X65Sai
       TRUE        TRUE        TRUE       FALSE        TRUE        TRUE
     Kfufu        Ktan       Konin       Rikon
       TRUE       FALSE        TRUE        TRUE
> aicrg[7] # AIC 値
[1] -165.452
```

- 単純な総当り法と当然同じ結果.

1.6 逐次選択法，総当り法，分枝限定法の比較

- AIC 最小化モデルの探索アルゴリズム
  1. 逐次選択法 (stepwise) ＝局所最適解，とても高速
  2. 総当り法＝大域最適解，低速
  3. 分枝限定法＝大域最適解，高速
- この３通りのアルゴリズムを比較する.

```
# run0083.R
# AIC 最小モデル：逐次探索法, 総当り法, 分枝限定法の比較
# x=説明変数の行列，y=目的変数のベクトル
func0083a <- function(x,y) { # 逐次探索法
  dat <- data.frame(x,y) # データフレームにしておく
```



```
  fit <- lm(y~.,dat) # すべての変数を使って回帰
  na0 <- names(coef(fit)) # 変数名
  fitstep <- step(fit,trace=F) # 途中結果を表示しない
  na1 <- names(coef(fitstep)) # 変数名
  wh <- na0 %in% na1 # na0 の各要素が na1 に含まれるかどうかの論理型ベクトル
  names(wh) <- na0 # 変数名を要素の名前につけておく
  wh <- wh[-1] # (Intercept) は取り除く
  list(which=wh,aic=AIC(fitstep))
}
source("run0082.R")
func0083b <- function(x,y) { # 総当り法
  whs <- func0082a(ncol(x)) # 変数のすべての組み合わせを生成
  colnames(whs) <- colnames(x) # 変数名を列の名前につけておく
  aics <- func0082b(x,y,whs) # 総当り法ですべての AIC 値を計算
  i <- order(aics)[1] # ベストモデルの添え字
  list(which=whs[i,],aic=aics[i])
}
library(leaps)
func0083c <- function(x,y) { # 分枝限定法
  dat <- data.frame(x,y) # データフレームにしておく
  rg <- regsubsets(y~.,dat,nvmax=100) # nvmax は説明変数の個数の最大値
  srg <- summary(rg) # 結果を整理
  n <- nrow(x); p <- nrow(srg$which) # データサイズ，説明変数の数
  wh0 <- srg$which[1,]; wh0[-1] <- F # y~1 モデルの定義
  srg$which <- rbind(wh0,srg$which) # y~1 も候補モデルに追加
  srg$rss <- c(sum((y-mean(y))^2),srg$rss) # y~1 も候補モデルに追加
  aicrg <- 2*(n/2)*(1+log(2*pi*srg$rss/n)) + 2*(2 + (0:p))  # AIC 値の計算
  i <- order(aicrg)[1]  # ベストモデルの番号（変数の数）
  list(which=srg$which[i,-1],aic=aicrg[i])
}
func0083 <- function(x,y,iss=1:3) { # 上記３通りを比較
  nas <- c("逐次探索","総当たり","分枝限定")[iss];
  funcs <- c(func0083a,func0083b,func0083c)[iss];
  ans <- list(); # 結果を保存するリスト
  showpat <- function(v) # 変数パターンの表示
    paste(sapply(v,function(x) if(x) "+" else "-"),collapse="")
  showline <- function(ra=NULL) # 結果の表示
    if(is.null(ra)) # 表の一行目
      cat(sprintf("# %8s %8s %16s %8s\n","手法","秒","説明変数","AIC"))
    else # 変数選択結果の表示
      cat(sprintf("# %8s %8g %16s %8g\n",
                  ra$name,ra$time,showpat(ra$which),round(ra$aic,2)))
```



```
  showline(); # 表の一行目
  for(i in 1:length(funcs)) { # 関数を順番に実行
    ti <- system.time(ans[[i]] <- funcs[[i]](x,y)); # 実行し「秒」を取得
    ans[[i]]$name <- nas[i]; ans[[i]]$time <- ti[1];
    showline(ans[[i]]); # 結果の表示
  }
  which <- t(sapply(ans,function(a) a$which)) # 選択した説明変数
  aic <- sapply(ans,function(a) a$aic) # AIC 値
  time <- sapply(ans,function(a) a$time) # 秒
  rownames(which) <- names(aic) <- names(time) <- nas
  list(which=which,aic=aic,time=time)
}
```

```
> source("run0083.R")
> dat <- read.table("dat0002.txt")
> x <- dat[,-1]; y <- dat[,1]
> func0083a(x,y)
$which
  Ninzu    Kaku    Tomo Tandoku  X65Sai   Kfufu    Ktan   Konin   Rikon
   TRUE    TRUE   FALSE    TRUE    TRUE    TRUE   FALSE    TRUE    TRUE

$aic
[1] -165.452

> func0083b(x,y)
$which
  Ninzu    Kaku    Tomo Tandoku  X65Sai   Kfufu    Ktan   Konin   Rikon
   TRUE    TRUE   FALSE    TRUE    TRUE    TRUE   FALSE    TRUE    TRUE

$aic
(1245689)
 -165.452

> func0083c(x,y)
$which
  Ninzu    Kaku    Tomo Tandoku  X65Sai   Kfufu    Ktan   Konin   Rikon
   TRUE    TRUE   FALSE    TRUE    TRUE    TRUE   FALSE    TRUE    TRUE

$aic
[1] -165.452

> func0083(x,y)
```

```
#      手法        秒          説明変数       AIC
# 逐次探索      0.09         ++-+++-++  -165.45
# 総当たり      0.47         ++-+++-++  -165.45
# 分枝限定      0.01         ++-+++-++  -165.45
$which
          Ninzu Kaku  Tomo Tandoku X65Sai Kfufu  Ktan Konin Rikon
逐次探索   TRUE TRUE FALSE    TRUE   TRUE  TRUE FALSE  TRUE  TRUE
総当たり   TRUE TRUE FALSE    TRUE   TRUE  TRUE FALSE  TRUE  TRUE
分枝限定   TRUE TRUE FALSE    TRUE   TRUE  TRUE FALSE  TRUE  TRUE

$aic
逐次探索 総当たり 分枝限定
-165.452 -165.452 -165.452

$time
逐次探索 総当たり 分枝限定
    0.09     0.47     0.01
```

1.7 AIC最小モデル：アルゴリズム比較の実行（その１）

- 変数の数を次第に増やしながら，アルゴリズムの比較をする.
- X2000から説明変数２７個，目的変数１個のデータセットを作成
- $p = 2, \ldots, 13$ までは３通りのアルゴリズムを比較
- $p = 14, \ldots, 27$ は，逐次探索と分枝限定法の比較（総当り法は遅すぎる）

```
# run0084.R
# AIC最小モデル：アルゴリズム比較の実行（その１）
source("run0083.R") # アルゴリズム比較
source("run0066.R") # データセットをX2000から取り出す
cat("## X2000からデータセットを作成 \n")
code2 <- c("A05201","A06102","A06202","F01503","A06205",
           "A06301","A06302","A06304","A06601","A06602")
name2 <- c("Zouka","Ninzu","Kaku","Tomo","Tandoku",
           "65Sai","Kfufu","Ktan","Konin","Rikon")
code3 <- c(paste("A0410",3:8,sep=""),"A04307","A04405","A04406")
name3 <- c(paste("Mik",seq(20,45,by=5),sep=""),"Shb60","Rik40","Rik50")
code3 <- c(paste(code3,"01",sep=""),paste(code3,"02",sep=""))
name3 <- c(paste(name3,"M",sep=""),paste(name3,"F",sep=""))
dat <- mygetdat(c(code2,code3),c(name2,name3))
x <- dat[,-1]; y <- dat[,1]
```



```
cat("## 変数を増やしながら変数選択（2-13)\n")
for(p in 2:13) {
  cat("## 変数の数=",p,"\n")
  a <- func0083(x[,1:p],y)
}
cat("## 変数を増やしながら変数選択（14-27)\n")
for(p in 14:27) {
  cat("## 変数の数=",p,"\n")
  a <- func0083(x[,1:p],y,c(1,3))
}
```

```
> source("run0084.R")
### X2000からデータセットを作成
# データサイズ= 47 * 28
# 最初の３行
         Zouka Ninzu  Kaku  Tomo Tandoku 65Sai Kfufu Ktan Konin Rikon Mik20M
Hokkaido  0.04  2.42 60.54 26.54   29.95 30.50  9.90 7.39  5.77  2.40   91.2
Aomori   -0.02  2.86 54.20 34.38   24.08 38.99  7.45 6.61  5.24  1.96   90.0
Iwate    -0.07  2.92 50.87 38.82   24.47 42.42  7.87 6.05  5.14  1.48   88.7
         Mik25M Mik30M Mik35M Mik40M Mik45M Shb60M Rik40M Rik50M Mik20F Mik25F
Hokkaido   64.8   39.0   23.2   16.3   12.2    9.0    4.7    5.5   85.7   52.6
Aomori     64.0   40.3   26.0   19.5   14.7    8.9    4.7    5.8   83.3   48.9
Iwate      63.7   42.1   29.3   22.4   16.8    9.5    3.9    4.6   82.2   48.2
         Mik30F Mik35F Mik40F Mik45F Shb60F Rik40F Rik50F
Hokkaido   28.3   16.5   11.0    7.8   38.4    8.7    8.4
Aomori     24.2   12.5    8.0    5.6   42.9    8.3    7.9
Iwate      24.2   12.6    7.7    5.6   41.5    6.5    6.1
# 変数の意味
             code                                Imi        Tani
Zouka      A05201                        自然増加率         (%)
Ninzu      A06102              一般世帯の平均人員（人:person)
Kaku       A06202                   核家族世帯割合         (%)
Tomo       F01503                   共働き世帯割合         (%)
Tandoku    A06205                   単独世帯割合         (%)
65Sai      A06301 65歳以上の親族のいる世帯割合         (%)
Kfufu      A06302       高齢夫婦のみの世帯の割合         (%)
Ktan       A06304              高齢単身世帯の割合         (%)
Konin      A06601       婚姻率（人口千人当たり）
Rikon      A06602       離婚率（人口千人当たり）
Mik20M   A0410301    未婚者割合［20〜24歳・男］         (%)
Mik25M   A0410401    未婚者割合［25〜29歳・男］         (%)
```



```
Mik30M   A0410501    未婚者割合［30〜34歳・男］         (%)
Mik35M   A0410601    未婚者割合［35〜39歳・男］         (%)
Mik40M   A0410701    未婚者割合［40〜44歳・男］         (%)
Mik45M   A0410801    未婚者割合［45〜49歳・男］         (%)
Shb60M   A0430701    死別者割合［60歳以上・男］         (%)
Rik40M   A0440501    離別者割合［40〜49歳・男］         (%)
Rik50M   A0440601    離別者割合［50〜59歳・男］         (%)
Mik20F   A0410302    未婚者割合［20〜24歳・女］         (%)
Mik25F   A0410402    未婚者割合［25〜29歳・女］         (%)
Mik30F   A0410502    未婚者割合［30〜34歳・女］         (%)
Mik35F   A0410602    未婚者割合［35〜39歳・女］         (%)
Mik40F   A0410702    未婚者割合［40〜44歳・女］         (%)
Mik45F   A0410802    未婚者割合［45〜49歳・女］         (%)
Shb60F   A0430702    死別者割合［60歳以上・女］         (%)
Rik40F   A0440502    離別者割合［40〜49歳・女］         (%)
Rik50F   A0440602    離別者割合［50〜59歳・女］         (%)
## 変数を増やしながら変数選択（2-13)
## 変数の数= 2
#      手法        秒          説明変数       AIC
# 逐次探索      0.02               ++    -38.7
# 総当たり      0.01               ++    -38.7
# 分枝限定      0.01               ++    -38.7
## 変数の数= 3
#      手法        秒          説明変数       AIC
# 逐次探索      0.02              +++   -51.07
# 総当たり         0              +++   -51.07
# 分枝限定      0.01              +++   -51.07
## 変数の数= 4
#      手法        秒          説明変数       AIC
# 逐次探索      0.04             ++-+  -108.61
# 総当たり      0.01             ++-+  -108.61
# 分枝限定      0.01             ++-+  -108.61
## 変数の数= 5
#      手法        秒          説明変数       AIC
# 逐次探索      0.02            +++++  -144.05
# 総当たり      0.03            +++++  -144.05
# 分枝限定      0.01            +++++  -144.05
## 変数の数= 6
#      手法        秒          説明変数       AIC
# 逐次探索      0.05           ++-+++  -145.67
# 総当たり      0.06           ++-+++  -145.67
# 分枝限定      0.01           ++-+++  -145.67
```



```
## 変数の数= 7
#      手法        秒          説明変数       AIC
# 逐次探索      0.08           ++-+++-  -145.67
# 総当たり      0.12           ++-+++-  -145.67
# 分枝限定      0.02           ++-+++-  -145.67
## 変数の数= 8
#      手法        秒          説明変数       AIC
# 逐次探索      0.08          ++-+++-+  -161.39
# 総当たり      0.32          ++-+++-+  -161.39
# 分枝限定      0.02          ++-+++-+  -161.39
## 変数の数= 9
#      手法        秒          説明変数       AIC
# 逐次探索      0.08         ++-+++-++  -165.45
# 総当たり      0.47         ++-+++-++  -165.45
# 分枝限定      0.01         ++-+++-++  -165.45
## 変数の数= 10
#      手法        秒          説明変数       AIC
# 逐次探索      0.13        ++-+++-+-+  -167.35
# 総当たり      0.94        ++-+++-+-+  -167.35
# 分枝限定      0.01        ++-+++-+-+  -167.35
## 変数の数= 11
#      手法        秒          説明変数       AIC
# 逐次探索       0.1       ++-+-++++++  -172.49
# 総当たり      1.92       ++-+--++-++  -173.56
# 分枝限定      0.01       ++-+--++-++  -173.56
## 変数の数= 12
#      手法        秒          説明変数       AIC
# 逐次探索      0.14      +++++-+++--+  -176.99
# 総当たり      3.91      +++++-+++--+  -176.99
# 分枝限定      0.02      +++++-+++--+  -176.99
## 変数の数= 13
#      手法        秒          説明変数       AIC
# 逐次探索      0.21     ++-+--+++--++  -178.86
# 総当たり      7.79     ++-+--+++--++  -178.86
# 分枝限定      0.02     ++-+--+++--++  -178.86
## 変数を増やしながら変数選択（14-27)
## 変数の数= 14
#      手法        秒          説明変数       AIC
# 逐次探索      0.25   ++-+--+++--+-+  -178.45
# 分枝限定      0.02   ++-+--+++--++-  -178.86
## 変数の数= 15
#      手法        秒          説明変数       AIC
```

```
# 逐次探索      0.29  ++-+--+++--+-+-  -178.45
# 分枝限定      0.02  ++-+--+++--+-+--  -178.86
## 変数の数= 16
#      手法        秒          説明変数       AIC
# 逐次探索      0.24 +++++-+++--+--++  -187.17
# 分枝限定      0.03 +++++-+++--+--++  -187.17
## 変数の数= 17
#      手法        秒          説明変数       AIC
# 逐次探索      0.36 ++-++-++---+---++  -193.41
# 分枝限定      0.02 ++-++-++---+---++  -193.41
## 変数の数= 18
#      手法        秒          説明変数       AIC
# 逐次探索      0.36 ++-+-++++---++-+-+  -194.78
# 分枝限定      0.04 ++-+-+++---+---+-+  -196.28
## 変数の数= 19
#      手法        秒          説明変数       AIC
# 逐次探索      0.41 ++-+-++++---++-+-+-  -194.78
# 分枝限定      0.03 ++-+-+++---+---+-+-  -196.28
## 変数の数= 20
#      手法        秒          説明変数       AIC
# 逐次探索      0.39 ++-+-++++-+-+-++-+-+  -195.82
# 分枝限定      0.03 ++-+-+++---+---+-+--  -196.28
## 変数の数= 21
#      手法        秒          説明変数       AIC
# 逐次探索      0.46 ++-+-++++-+-+-++-+-+-  -195.82
# 分枝限定      0.05 ++-+-+++---+---+-+---  -196.28
## 変数の数= 22
#      手法        秒          説明変数       AIC
# 逐次探索      0.51 ++-+-++++-+-+-++-+-+--  -195.82
# 分枝限定      0.05 ++-+-+++---+---+-+----  -196.28
## 変数の数= 23
#      手法        秒          説明変数       AIC
# 逐次探索      0.67 ++-+-++++---+-+-+---++   -195.9
# 分枝限定      0.07 ++-+-+++---+---+-+-----  -196.28
## 変数の数= 24
#      手法        秒          説明変数       AIC
# 逐次探索      0.63 ++-+-++++---++-+-+---++-   -195.9
# 分枝限定      0.09 ++-+-+++---+---+-+------  -196.28
## 変数の数= 25
#      手法        秒          説明変数       AIC
# 逐次探索       0.7 ++-+-++++---++-+-+---++--   -195.9
# 分枝限定      0.13 ++-+-+++---+---+-+-------  -196.28
```



```
## 変数の数= 26
#      手法        秒          説明変数       AIC
# 逐次探索      0.51 ++-+--+++++++--++++-++-+++   -201.9
# 分枝限定       0.2 ++-+--+++--+-+----+-+-++--+  -204.64
## 変数の数= 27
#      手法        秒          説明変数       AIC
# 逐次探索      0.54 ++-++++++++++--+--+-++-++++   -202.6
# 分枝限定      0.37 ++-+--+++--+-+----+-+-++--+-  -204.64
```

- 逐次探索と分枝限定法で選ばれるモデルは，ほとんど同じことが多い．
- 総当り法ではpを一つ増やすたびに探索範囲が２倍大きくなり，実行速度もこれにほぼ比例している．
- 逐次探索（step()）と分枝限定法（library(leaps)）はどちらも十分早い．

1.8 AIC最小モデル：アルゴリズム比較の実行（その２）

```
# run0086.R
# AIC最小モデル：アルゴリズム比較の実行（その２）
source("run0083.R") # アルゴリズム比較
cat("### 乱数でデータセットを生成\n")
n <- 100; pm <- 40 # サンプルサイズと変数の数
x <- matrix(runif(n*pm,-1,1),n) # xは，まず一様分布 [-1,1]
x <- x + runif(n,-0.5,0.5) # 各列に同じ乱数を足して，変数間に相関をつける
cat("# n=",n,", p=",pm,"\n")
be <- c(rep(1,5),rep(0.2,5),rep(0,pm-10)) # 真の回帰係数
y <- x %*% be + 1*rnorm(n) # yの作成
cat("# beta\n")
print(be)
cat("# xの相関行列一部\n")
print(round(cor(x,x),3)[1:10,1:10])
cat("# xとyの相関\n")
print(round(cor(y,x),3))
ans <- list() # 一時結果をしまうアレイ
for(p in c(2:pm)) {
  cat("## 変数の数=",p,"\n")
  ans[[as.character(p)]] <- func0083(x[,1:p],y,c(1,3))
}
kekka <- list(aic = sapply(ans,function(v) v$aic),
              time = sapply(ans,function(v) v$time))
kekka$p <- as.numeric(dimnames(kekka$time)[[2]])
cat("## 結果\n")
```



```
print(kekka)
matplot(kekka$p,t(kekka$time),log="y",type="b",xlab="p",ylab="time(sec)")
legend(min(kekka$p),max(kekka$time)*0.9,
       c("1. step","2. leaps"),lty=1:2,col=1:2,bty="n")
dev.copy2eps(file="run0086-t.eps")
```

```
> source("run0086.R")
### 乱数でデータセットを生成
# n= 100 , p= 40
# beta
 [1] 1.0 1.0 1.0 1.0 1.0 0.2 0.2 0.2 0.2 0.2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[20] 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
[39] 0.0 0.0
# xの相関行列一部
       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
 [1,] 1.000 0.386 0.296 0.230 0.251 0.253 0.141 0.314 0.226 0.223
 [2,] 0.386 1.000 0.269 0.291 0.319 0.169 0.097 0.252 0.021 0.150
 [3,] 0.296 0.269 1.000 0.315 0.260 0.281 0.327 0.287 0.265 0.186
 [4,] 0.230 0.291 0.315 1.000 0.337 0.340 0.140 0.194 0.144 0.388
 [5,] 0.251 0.319 0.260 0.337 1.000 0.194 0.052 0.126 0.198 0.415
 [6,] 0.253 0.169 0.281 0.340 0.194 1.000 0.212 0.212 0.069 0.107
 [7,] 0.141 0.097 0.327 0.140 0.052 0.212 1.000 0.135 0.147 0.110
 [8,] 0.314 0.252 0.287 0.194 0.126 0.212 0.135 1.000 0.180 0.087
 [9,] 0.226 0.021 0.265 0.144 0.198 0.069 0.147 0.180 1.000 0.243
[10,] 0.223 0.150 0.186 0.388 0.415 0.107 0.110 0.087 0.243 1.000
# xとyの相関
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
[1,] 0.621 0.575 0.641 0.608 0.622 0.462 0.236 0.402 0.308 0.465 0.494 0.398
     [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
[1,] 0.101 0.415 0.383 0.378 0.391 0.238 0.451 0.356 0.335 0.257 0.399 0.379
     [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
[1,] 0.432 0.346 0.418 0.228 0.224 0.364 0.477 0.422 0.333  0.42   0.4 0.284
     [,37] [,38] [,39] [,40]
[1,] 0.349 0.527 0.408  0.33
## 変数の数= 2
#      手法        秒          説明変数       AIC
# 逐次探索      0.02              ++    403.5
# 分枝限定      0.01              ++    403.5
## 変数の数= 3
#      手法        秒          説明変数       AIC
# 逐次探索      0.02             +++   359.63
# 分枝限定      0.02             +++   359.63
```



```
## 変数の数= 4
#      手法        秒          説明変数       AIC
# 逐次探索      0.11            ++++   322.06
# 分枝限定      0.01            ++++   322.06
## 変数の数= 5
#      手法        秒          説明変数       AIC
# 逐次探索      0.03           +++++   278.13
# 分枝限定      0.02           +++++   278.13
## 変数の数= 6
#      手法        秒          説明変数       AIC
# 逐次探索      0.03          ++++++   269.11
# 分枝限定      0.01          ++++++   269.11
## 変数の数= 7
#      手法        秒          説明変数       AIC
# 逐次探索      0.06         +++++++-  269.11
# 分枝限定      0.02         +++++++-  269.11
## 変数の数= 8
#      手法        秒          説明変数       AIC
# 逐次探索      0.06         +++++++-+  267.31
# 分枝限定      0.02         +++++++-+  267.31
## 変数の数= 9
#      手法        秒          説明変数       AIC
# 逐次探索      0.07        ++++++-++  267.09
# 分枝限定      0.02        ++++++-++  267.09
## 変数の数= 10
#      手法        秒          説明変数       AIC
# 逐次探索       0.1       ++++++-+-+  260.68
# 分枝限定      0.02       ++++++-+-+  260.68
## 変数の数= 11
#      手法        秒          説明変数       AIC
# 逐次探索      0.14      ++++++-+-+-  260.68
# 分枝限定      0.02      ++++++-+-+-  260.68
## 変数の数= 12
#      手法        秒          説明変数       AIC
# 逐次探索      0.18     ++++++-+-+--  260.68
# 分枝限定      0.02     ++++++-+-+--  260.68
## 変数の数= 13
#      手法        秒          説明変数       AIC
# 逐次探索      0.31    ++++++-+-+---  260.68
# 分枝限定      0.02    ++++++-+-+---  260.68
## 変数の数= 14
#      手法        秒          説明変数       AIC
```

```
# 逐次探索       0.2   +++++++-+++----+   257.84
# 分枝限定       0.02  +++++++-+++----+   257.84
## 変数の数= 15
#       手法         秒          説明変数       AIC
# 逐次探索       0.25  +++++++-+++----+-   257.84
# 分枝限定       0.03  +++++++-+++----+-   257.84
## 変数の数= 16
#       手法         秒          説明変数       AIC
# 逐次探索       0.31 +++++++-+++----+--   257.84
# 分枝限定       0.02 +++++++-+++----+--   257.84
## 変数の数= 17
#       手法         秒          説明変数       AIC
# 逐次探索       0.34 +++++++-+++----+--+   257.36
# 分枝限定       0.02 +++++++-+++----+--+   257.36
## 変数の数= 18
#       手法         秒          説明変数       AIC
# 逐次探索        0.4 +++++++-+++----+--+-   257.36
# 分枝限定       0.03 +++++++-+++----+--+-   257.36
## 変数の数= 19
#       手法         秒          説明変数       AIC
# 逐次探索       0.47 +++++++-+++----+--+--   257.36
# 分枝限定       0.02 +++++++-+++----+--+--   257.36
## 変数の数= 20
#       手法         秒          説明変数       AIC
# 逐次探索       0.54 +++++++-+++----+--+---   257.36
# 分枝限定       0.03 +++++++-+++----+--+---   257.36
## 変数の数= 21
#       手法         秒          説明変数       AIC
# 逐次探索       0.68 +++++++-+++----+--+----   257.36
# 分枝限定       0.03 +++++++-+++----+--+----   257.36
## 変数の数= 22
#       手法         秒          説明変数       AIC
# 逐次探索       0.65 +++++++-+++----+--+-----   257.36
# 分枝限定       0.03 +++++++-+++----+--+-----   257.36
## 変数の数= 23
#       手法         秒          説明変数       AIC
# 逐次探索       0.75 +++++++-+++----+--+------   257.36
# 分枝限定       0.03 +++++++-+++----+--+------   257.36
## 変数の数= 24
#       手法         秒          説明変数       AIC
# 逐次探索       0.85 +++++++-+++----+--+-------   257.36
# 分枝限定       0.04 +++++++-+++----+--+-------   257.36
```



```
## 変数の数= 25
#       手法         秒          説明変数       AIC
# 逐次探索       0.99 +++++++-+++----+----------+   256.75
# 分枝限定       0.04 +++++++-+++----+----------+   256.75
## 変数の数= 26
#       手法         秒          説明変数       AIC
# 逐次探索       0.99 +++++++-+++----+----------+-   256.75
# 分枝限定       0.05 +++++++-+++----+----------+-   256.75
## 変数の数= 27
#       手法         秒          説明変数       AIC
# 逐次探索       1.16 +++++++-+++----+----------+-+   256.57
# 分枝限定       0.06 +++++++-+++----+----------+-+   256.57
## 変数の数= 28
#       手法         秒          説明変数       AIC
# 逐次探索       1.12 +++++++-+++----+----------+-+-   256.57
# 分枝限定       0.08 +++++++-+++----+----------+-+-   256.57
## 変数の数= 29
#       手法         秒          説明変数       AIC
# 逐次探索       1.25 +++++++-+++----+----------+-+--   256.57
# 分枝限定       0.13 +++++++-+++----+----------+-+--   256.57
## 変数の数= 30
#       手法         秒          説明変数       AIC
# 逐次探索        1.4 +++++++-+++----+----------+-+---   256.57
# 分枝限定       0.15 +++++++-+++----+----------+-+---   256.57
## 変数の数= 31
#       手法         秒          説明変数       AIC
# 逐次探索       1.54 +++++++-+++----+----------+-+----   256.57
# 分枝限定       0.23 +++++++-+++----+----------+-+----   256.57
## 変数の数= 32
#       手法         秒          説明変数       AIC
# 逐次探索       1.53 +++++++-+++----+----------+-+-----   256.57
# 分枝限定       0.27 +++++++-+++----+----------+-+-----   256.57
## 変数の数= 33
#       手法         秒          説明変数       AIC
# 逐次探索       1.75 +++++++-+++----+----------+-+------   256.57
# 分枝限定        0.4 +++++++-+++----+----------+-+------   256.57
## 変数の数= 34
#       手法         秒          説明変数       AIC
# 逐次探索       1.88 +++++++-+++----+----------+-+-------   256.57
# 分枝限定       0.88 +++++++-+++----+----------+-+-------   256.57
## 変数の数= 35
#       手法         秒          説明変数       AIC
```

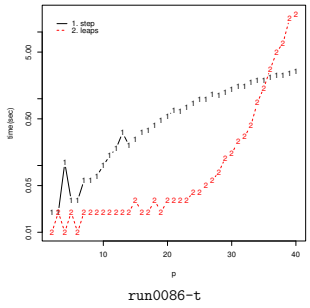


```
# 逐次探索       1.92 +++++++-+++----+----------+-+--------   256.57
# 分枝限定       1.45 +++++++-+++----+----------+-+--------   256.57
## 変数の数= 36
#       手法         秒          説明変数       AIC
# 逐次探索       2.12 +++++++-+++----+--------------------++   258.15
# 分枝限定       2.77 +++++++-+++----+----------+-+---------   256.57
## 変数の数= 37
#       手法         秒          説明変数       AIC
# 逐次探索       2.26 +++++++-+++----+--------------------++-   258.15
# 分枝限定          5 +++++++-+++----+----------+-+----------   256.57
## 変数の数= 38
#       手法         秒          説明変数       AIC
# 逐次探索       2.26 +++++++-+++----+----------+-+----+--+--+   255.84
# 分枝限定       6.84 +++++++-+++----+----------+-+----+--+--+   255.84
## 変数の数= 39
#       手法         秒          説明変数       AIC
# 逐次探索       2.42 +++++++-+++----+--+---------++---------++   254.56
# 分枝限定      15.77 +++++++-+++----+--+--------------------++   254.42
## 変数の数= 40
#       手法         秒          説明変数       AIC
# 逐次探索       2.55 +++++++-+++----+--+---------++---------++-   254.56
# 分枝限定      18.65 +++++++-+++----+--+--------------------++-   254.42
## 結果
$aic
                  2        3        4        5        6        7        8
逐次探索 403.4956 359.6254 322.0608 278.1266 269.1076 269.1076 267.3104
分枝限定 403.4956 359.6254 322.0608 278.1266 269.1076 269.1076 267.3104
                  9       10       11       12       13       14       15
逐次探索 267.0883 260.6802 260.6802 260.6802 260.6802 257.8380 257.8380
分枝限定 267.0883 260.6802 260.6802 260.6802 260.6802 257.8380 257.8380
                 16       17       18       19       20       21       22
逐次探索 257.8380 257.3616 257.3616 257.3616 257.3616 257.3616 257.3616
分枝限定 257.8380 257.3616 257.3616 257.3616 257.3616 257.3616 257.3616
                 23       24       25       26       27       28       29
逐次探索 257.3616 257.3616 256.7461 256.7461 256.5724 256.5724 256.5724
分枝限定 257.3616 257.3616 256.7461 256.7461 256.5724 256.5724 256.5724
                 30       31       32       33       34       35       36
逐次探索 256.5724 256.5724 256.5724 256.5724 256.5724 256.5724 258.1535
分枝限定 256.5724 256.5724 256.5724 256.5724 256.5724 256.5724 256.5724
                 37       38       39       40
逐次探索 258.1535 255.8431 254.5649 254.5649
分枝限定 256.5724 255.8431 254.4218 254.4218
```



```
$time
                2     3     4     5     6     7     8     9    10    11    12    13    14    15
逐次探索 0.02 0.02 0.11 0.03 0.03 0.06 0.06 0.07 0.10 0.14 0.18 0.31 0.20 0.25
分枝限定 0.01 0.02 0.01 0.02 0.01 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.03
               16    17    18    19    20    21    22    23    24    25    26    27    28    29
逐次探索 0.31 0.34 0.40 0.47 0.54 0.68 0.65 0.75 0.85 0.99 0.99 1.16 1.12 1.25
分枝限定 0.02 0.02 0.03 0.02 0.03 0.03 0.03 0.03 0.04 0.04 0.05 0.06 0.08 0.13
               30    31    32    33    34    35    36    37    38    39    40
逐次探索 1.40 1.54 1.53 1.75 1.88 1.92 2.12 2.26 2.26  2.42  2.55
分枝限定 0.15 0.23 0.27 0.40 0.88 1.45 2.77 5.00 6.84 15.77 18.65

$p
 [1]  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
[26] 27 28 29 30 31 32 33 34 35 36 37 38 39 40
```



run0086-t

- 逐次探索と分枝限定法で選ばれるモデルは，ほとんど同じことが多い．
- step() の実装が遅いため，逐次探索はかなり遅いが，変数の数が増えても実行速度が極端に遅くなることはない．
- library(leaps) の実装が優れているため，分枝限定法はかなり早いが，変数の数が３０を越えたあたりから急速に遅くなっている．

1.9 ブートストラップ法の適用

- ブートストラップ法でデータセットを多数個生成する
- 個々のデータセットに対して，分枝限定法によるモデル選択（変数選択）を実行する．

- 選択されたモデル（変数の組み合わせ）を記録し，その頻度を示す.
- 各説明変数についても，その選ばれた頻度を示す.

```
# run0085.R
# AIC最小モデル：ブートストラップ法の適用
source("run0083.R")
# b回のブートストラップ法
func0085a <- function(x,y,b,func=func0083c,...) {
  ans <- list(); n <- nrow(x)
  for(j in 1:b) {
    i <- sample(1:n,replace=T) # リサンプリングする添え字
    ans[[j]] <- func(x[i,],y[i],...) # funcを呼び出す
  }
  ans
}
# ブートストラップ法の結果をまとめる
func0085b <- function(ans) {
  whs <- t(sapply(ans,function(v) v$which)) # "which"が縦にb個並んだ行列
  pattern <- wh2pa(whs) # 「パターン」がb個並んだベクトル
  tab <- rev(sort(table(pattern))) # 頻度の大きい順に並べたテーブル
  tab
}
# ブートストラップ法の結果に対してAIC値を計算する
func0085c <- function(x,y,tab,aic.dif=T,wh0=NULL,func=func0083c) {
  calcaic <- function(wh,x,y) AIC(lm(y~.,data.frame(x[,wh],y)))
  whs <- pa2wh(names(tab),colnames(x)) # "which"の行列，ただし頻度順
  aic <- apply(whs,1,calcaic,x,y) # AIC値
  if(aic.dif&&is.null(wh0))
    wh0 <- func(x,y)$which # オリジナルデータのAIC最小モデル
  if(!is.null(wh0)) aic <- aic - calcaic(wh0,x,y) # wh0モデルからの差
  list(pattern=names(tab),frequency=as.numeric(tab),
       which=whs,aic=aic)
}
# 表の作成
func0085d <- function(a) {
  cat("# 各変数の選択頻度\n")
  print(apply(a$which * a$frequency,2,sum))
  cat("# 各パターンの頻度とAIC値\n")
  out <- data.frame(a$pattern,a$frequency,a$aic)
  names(out) <- c("パターン","頻度","AIC")
  print(out)
```



```
}
# "which"(論理型)をパターン(文字列)に変換
wh2pa <- function(whs) {
  wh2pa1 <- function(wh) # "which"1個だけ変換
    paste(sapply(wh,function(x) if(x) "+" else "-"),collapse="")
  if(is.matrix(whs)) apply(whs,1,wh2pa1) # "which"が縦に並んだ行列
  else wh2pa1(whs)
}
# パターン(文字列)を"which"(論理型)に変換
pa2wh <- function(pa,col.names=NULL) {
  pa2wh1 <- function(ch) { # パターン1個だけ変換
    p <- length(ch); wh <- logical(p)
    for(i in 1:p) wh[i] <- ch[i] == "+"
    wh
  }
  chs <- strsplit(pa,"") # 「文字列のベクタ」を「文字のベクタのリスト」へ変換
  whs <- t(sapply(chs,pa2wh1)) # "which"が縦に並んだ行列
  dimnames(whs)[[2]] <- col.names
  whs
}
```

```
> source("run0085.R")
> dat <- read.table("dat0002.txt")
> x <- dat[,-1]; y <- dat[,1]
> system.time(ans <- func0085a(x,y,1000))
[1] 15.93  0.06 15.97  0.00  0.00
> length(ans)
[1] 1000
> ans[[1]]
$which
  Ninzu    Kaku    Tomo Tandoku  X65Sai   Kfufu    Ktan   Konin   Rikon
   TRUE    TRUE   FALSE    TRUE    TRUE    TRUE   FALSE    TRUE    TRUE

$aic
[1] -184.3093

> ans[[2]]
$which
  Ninzu    Kaku    Tomo Tandoku  X65Sai   Kfufu    Ktan   Konin   Rikon
   TRUE    TRUE   FALSE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE

$aic
```



```
[1] -182.6504

> tab <- func0085b(ans)
> length(tab)
[1] 47
> tab[1]
++-+++-++
      227
> tab[2]
++++++-++
      164
> a <- func0085c(x,y,tab)
> a
$pattern
 [1] "++-+++-++" "++++++-++" "++-+++-+-" "++-+-++++" "++-++++++" "++++++-+-"
 [7] "+++++++++" "+--+++-++" "+--+++-+-" "+--++++-" "+-++++-++" "++++-++++"
[13] "++-++++-" "+--++++++" "+++++--++" "++++++++-" "+-++++-+-" "+-+++++++"
[19] "++++-+++" "+-++++++-" "++-++--+-" "+++++-+++" "++-+--++-" "++++---+-"
[25] "++--++++-" "++-+---+-" "++-+---++" "+---++++-" "+-+-++++-" "++--++-++"
[31] "++-+-+++-" "++++--++-" "+++++--+-" "++++++-++" "++++++--+" "+---+-++-"
[37] "+-+-+-+++" "+-++++-+" "++--+++++" "++-+--+++" "++-+++--+" "++-++++-+"
[43] "+++-++++-" "+++-++++" "++++-+-+-" "++++-+-++" "+++++++-+"

$frequency
 [1] 227 164 125  66  59  49  46  42  31  26  15  15  13  12  12  12  10   9   7
[20]   5   5   5   4   4   3   3   3   2   2   2   2   2   2   2   2   1   1   1
[39]   1   1   1   1   1   1   1   1   1

$which
      Ninzu  Kaku  Tomo Tandoku X65Sai Kfufu  Ktan Konin Rikon
 [1,]  TRUE  TRUE FALSE    TRUE   TRUE  TRUE FALSE  TRUE  TRUE
 [2,]  TRUE  TRUE  TRUE    TRUE   TRUE  TRUE FALSE  TRUE  TRUE
 [3,]  TRUE  TRUE FALSE    TRUE   TRUE  TRUE FALSE  TRUE FALSE
 [4,]  TRUE  TRUE FALSE    TRUE  FALSE  TRUE  TRUE  TRUE  TRUE
 [5,]  TRUE  TRUE FALSE    TRUE   TRUE  TRUE  TRUE  TRUE  TRUE
 [6,]  TRUE  TRUE  TRUE    TRUE   TRUE  TRUE FALSE  TRUE FALSE
 [7,]  TRUE  TRUE  TRUE    TRUE   TRUE  TRUE  TRUE  TRUE  TRUE
 [8,]  TRUE FALSE FALSE    TRUE   TRUE  TRUE FALSE  TRUE  TRUE
 [9,]  TRUE FALSE FALSE    TRUE   TRUE  TRUE FALSE  TRUE FALSE
[10,]  TRUE FALSE FALSE    TRUE   TRUE  TRUE  TRUE  TRUE FALSE
[11,]  TRUE FALSE  TRUE    TRUE   TRUE  TRUE FALSE  TRUE  TRUE
[12,]  TRUE  TRUE  TRUE    TRUE  FALSE  TRUE  TRUE  TRUE  TRUE
```



```
[13,]  TRUE  TRUE FALSE    TRUE   TRUE  TRUE  TRUE  TRUE FALSE
[14,]  TRUE FALSE FALSE    TRUE   TRUE  TRUE  TRUE  TRUE  TRUE
[15,]  TRUE  TRUE  TRUE    TRUE   TRUE FALSE FALSE  TRUE  TRUE
[16,]  TRUE  TRUE  TRUE    TRUE   TRUE  TRUE  TRUE  TRUE FALSE
[17,]  TRUE FALSE  TRUE    TRUE   TRUE  TRUE FALSE  TRUE FALSE
[18,]  TRUE FALSE  TRUE    TRUE   TRUE  TRUE  TRUE  TRUE  TRUE
[19,]  TRUE  TRUE  TRUE    TRUE   TRUE FALSE  TRUE  TRUE  TRUE
[20,]  TRUE FALSE  TRUE    TRUE   TRUE  TRUE  TRUE  TRUE FALSE
[21,]  TRUE  TRUE FALSE    TRUE   TRUE FALSE FALSE  TRUE FALSE
[22,]  TRUE  TRUE  TRUE    TRUE  FALSE FALSE  TRUE  TRUE  TRUE
[23,]  TRUE  TRUE FALSE    TRUE  FALSE FALSE  TRUE  TRUE FALSE
[24,]  TRUE  TRUE  TRUE    TRUE  FALSE FALSE FALSE  TRUE FALSE
[25,]  TRUE  TRUE FALSE   FALSE   TRUE  TRUE  TRUE  TRUE FALSE
[26,]  TRUE  TRUE FALSE    TRUE  FALSE FALSE FALSE  TRUE FALSE
[27,]  TRUE  TRUE FALSE    TRUE  FALSE FALSE FALSE  TRUE  TRUE
[28,]  TRUE FALSE FALSE   FALSE   TRUE  TRUE  TRUE  TRUE FALSE
[29,]  TRUE FALSE  TRUE   FALSE   TRUE  TRUE  TRUE  TRUE FALSE
[30,]  TRUE  TRUE FALSE   FALSE   TRUE  TRUE FALSE  TRUE  TRUE
[31,]  TRUE  TRUE FALSE    TRUE  FALSE  TRUE  TRUE  TRUE FALSE
[32,]  TRUE  TRUE  TRUE    TRUE  FALSE FALSE  TRUE  TRUE FALSE
[33,]  TRUE  TRUE  TRUE    TRUE   TRUE FALSE FALSE  TRUE FALSE
[34,]  TRUE  TRUE  TRUE    TRUE   TRUE FALSE  TRUE  TRUE FALSE
[35,]  TRUE  TRUE  TRUE    TRUE   TRUE  TRUE FALSE FALSE  TRUE
[36,]  TRUE FALSE FALSE   FALSE   TRUE FALSE  TRUE  TRUE FALSE
[37,]  TRUE FALSE  TRUE   FALSE   TRUE FALSE  TRUE  TRUE  TRUE
[38,]  TRUE FALSE  TRUE    TRUE   TRUE  TRUE  TRUE FALSE  TRUE
[39,]  TRUE  TRUE FALSE   FALSE   TRUE  TRUE  TRUE  TRUE  TRUE
[40,]  TRUE  TRUE FALSE    TRUE  FALSE FALSE  TRUE  TRUE  TRUE
[41,]  TRUE  TRUE FALSE    TRUE   TRUE  TRUE FALSE FALSE  TRUE
[42,]  TRUE  TRUE FALSE    TRUE   TRUE  TRUE  TRUE FALSE  TRUE
[43,]  TRUE  TRUE  TRUE   FALSE   TRUE  TRUE  TRUE  TRUE FALSE
[44,]  TRUE  TRUE  TRUE   FALSE   TRUE  TRUE  TRUE  TRUE  TRUE
[45,]  TRUE  TRUE  TRUE    TRUE  FALSE  TRUE FALSE  TRUE FALSE
[46,]  TRUE  TRUE  TRUE    TRUE  FALSE  TRUE FALSE  TRUE  TRUE
[47,]  TRUE  TRUE  TRUE    TRUE   TRUE  TRUE  TRUE FALSE  TRUE

$aic
 [1]  0.0000000  0.8454263  4.0579833  6.3592955  1.5913845  5.8682820
 [7]  2.3918069  8.2500678 24.4535258 10.9012160 10.1098997  6.8501058
[13]  6.0385023  6.0853627  8.7250363  7.8427703 21.9456102  8.0184982
[19] 10.5755090 12.7599506  8.5958008  9.3826127  9.5872208 13.6529519
[25] 18.9339744 12.1485618 13.6408833 19.9050885 21.7950236 28.7048540
```

```
[31]  9.4447417 10.2437282  8.7511848 10.7488674 15.9124285 19.1158503
[37] 22.1981696 21.4046686 17.8858319 10.4506458 16.8101154 18.1330104
[43] 20.8758806 19.8735371 14.7566814 15.8891300 17.1692363

> func0085d(a)
# 各変数の選択頻度
  Ninzu    Kaku    Tomo Tandoku  X65Sai   Kfufu    Ktan   Konin   Rikon
   1000     843     370     986     893     948     301     994     695
# 各パターンの頻度と AIC 値
    パターン 頻度         AIC
1  ++-+++-++  227  0.0000000
2  +++++-++  164  0.8454263
3  ++-+++-+-  125  4.0579833
4  ++-+-++++   66  6.3592955
5  ++-++++++   59  1.5913845
6  ++++++-+-   49  5.8682820
7  +++++++++   46  2.3918069
8  +--+++-++   42  8.2500678
9  +--+++-+-   31 24.4535258
10 +--+++++-   26 10.9012160
11 +-+++++-++   15 10.1098997
12 ++++-++++   15  6.8501058
13 ++-++++-   13  6.0385023
14 +--++++++   12  6.0853627
15 +++++--++   12  8.7250363
16 +++++++-   12  7.8427703
17 +-++++-+-   10 21.9456102
18 +-++++++    9  8.0184982
19 +++++-+++    7 10.5755090
20 +-++++++-    5 12.7599506
21 ++-++--+-    5  8.5958008
22 ++++--+++    5  9.3826127
23 ++-+--++-    4  9.5872208
24 ++++---+-    4 13.6529519
25 ++--++++-    3 18.9339744
26 ++-+---+-    3 12.1485618
27 ++-+---++    3 13.6408833
28 +---++++-    2 19.9050885
29 +-+-++++-    2 21.7950236
30 ++--++-++    2 28.7048540
31 ++-+-+++-    2  9.4447417
32 ++++--++-    2 10.2437282
```



```
33 +++++--+-    2  8.7511848
34 +++++-++-    2 10.7488674
35 ++++++--+    2 15.9124285
36 +---+-++-    1 19.1158503
37 +-+-+-+++    1 22.1981696
38 +-++++-+    1 21.4046686
39 ++--+++++    1 17.8858319
40 ++-+--+++    1 10.4506458
41 ++-+++--+    1 16.8101154
42 ++-++++-+    1 18.1330104
43 +++-++++-    1 20.8758806
44 +++-+++++    1 19.8735371
45 ++++-+-+-    1 14.7566814
46 ++++-+-++    1 15.8891300
47 ++++++++    1 17.1692363
```

- 10 個の説明変数について，それぞれ何回選ばれたかを示している．Tomo と Ktan は選択頻度が 37%，30%程度で低い．
- ＡＩＣ最小になったモデルの頻度が示されている．表示した AIC 値は，AIC 最小モデルとの AIC 値の差．
- 「パターン」は，頻度順に並べ替えている．AIC 最小モデルでも 23%程度の回数しか選ばれない．
- 頻度が５%以上になるパターンは５個ある．これらは同等に良いと考えても良い．６番目のパターンも 4.9%であり，実際には，最初の 10 個程度のパターンはどれも良い可能性があると考えられるだろう．
- さらに，良い「パターン」に共通に含まれる部分パターンを探索してその頻度を数えるなどしても良い．

2　課題

2.1　課題 8-1

X2000 データセットおよび japan.pref データセットから自由に変量を１０個以上選び，重回帰分析を実行する．回帰係数の t-検定による変数選択 (有意水準 5%) と，AIC を最小化法による変数選択をおこない，これらを比較せよ．特に以下の点について報告すること．

- 選んだ変数の「コード」と「意味」
- すべての変数を使った場合および AIC を最小モデルについて：推定した回帰係数，その標準誤差，回帰係数の t 統計量，回帰係数の p-値，決定係数，AIC 値

2.2　課題 8-2

ベクタ x, ベクタ y, 最大次数 pm を入力とし，次数 p=1,2,...,pm までの多項式回帰を実行してその結果を返す関数を作成せよ．この関数の実行例も示せ．ただし，作成する関数は各次数 p に対して以下の値を返すものとする．

- 推定した回帰係数 $\hat{\beta}_0, \ldots, \hat{\beta}_p$
- $\hat{\beta}_p$ の t-検定の p-値，または，次数 $p-1$ から次数 p へ変更したときの RSS の改善をみる F-検定の p-値（どちらの p-値で計算しても，値は同じ）．
- AIC 値

2.3　課題 8-3 ＊

課題 8-1 の分析にブートストラップ法を適用し，各説明変数が選択された頻度を示せ．