Linear Regression Model Selection

Alexander McLain

September 5, 2023

This example will use the bodyfat data. First, we'll read it in.

```
library(printr)
bf_dat <- read.csv("bodyfat2.csv")
bf_df <- data.frame(bf_dat)
head(bf_df)
```

density	bodyfat	age	weight	height	neck	chest	abdomen	hip	thigh	knee	ankle	biceps	forearm	wrist
1.0708	12.3	23	154.25	67.75	36.2	93.1	85.2	94.5	59.0	37.3	21.9	32.0	27.4	17.1
1.0853	6.1	22	173.25	72.25	38.5	93.6	83.0	98.7	58.7	37.3	23.4	30.5	28.9	18.2
1.0414	25.3	22	154.00	66.25	34.0	95.8	87.9	99.2	59.6	38.9	24.0	28.8	25.2	16.6
1.0751	10.4	26	184.75	72.25	37.4	101.8	86.4	101.2	60.1	37.3	22.8	32.4	29.4	18.2
1.0340	28.7	24	184.25	71.25	34.4	97.3	100.0	101.9	63.2	42.2	24.0	32.2	27.7	17.7
1.0502	20.9	24	210.25	74.75	39.0	104.5	94.4	107.8	66.0	42.0	25.6	35.7	30.6	18.8

Best Subset Selection

Here, we'll perform best subset selection by fitting a separate least squares regression for each possible combination of the p predictors. We can perform a best subset search using *regsubsets* (part of the *leaps* library), which identifies the best model for a given number of k predictors, where best is quantified using RSS.

```
library(tidyverse)
library(leaps)
bf_df <- bf_df[,-1]
best_subset <- regsubsets(bodyfat ~ ., bf_df, nvmax = 10)
round(coef(best_subset, id = 8),4)
```

```
## (Intercept)         age      weight        neck     abdomen         hip
##    -22.6564      0.0658     -0.0899     -0.4666      0.9448     -0.1954
##       thigh     forearm       wrist
##      0.3024      0.5157     -1.5367
```

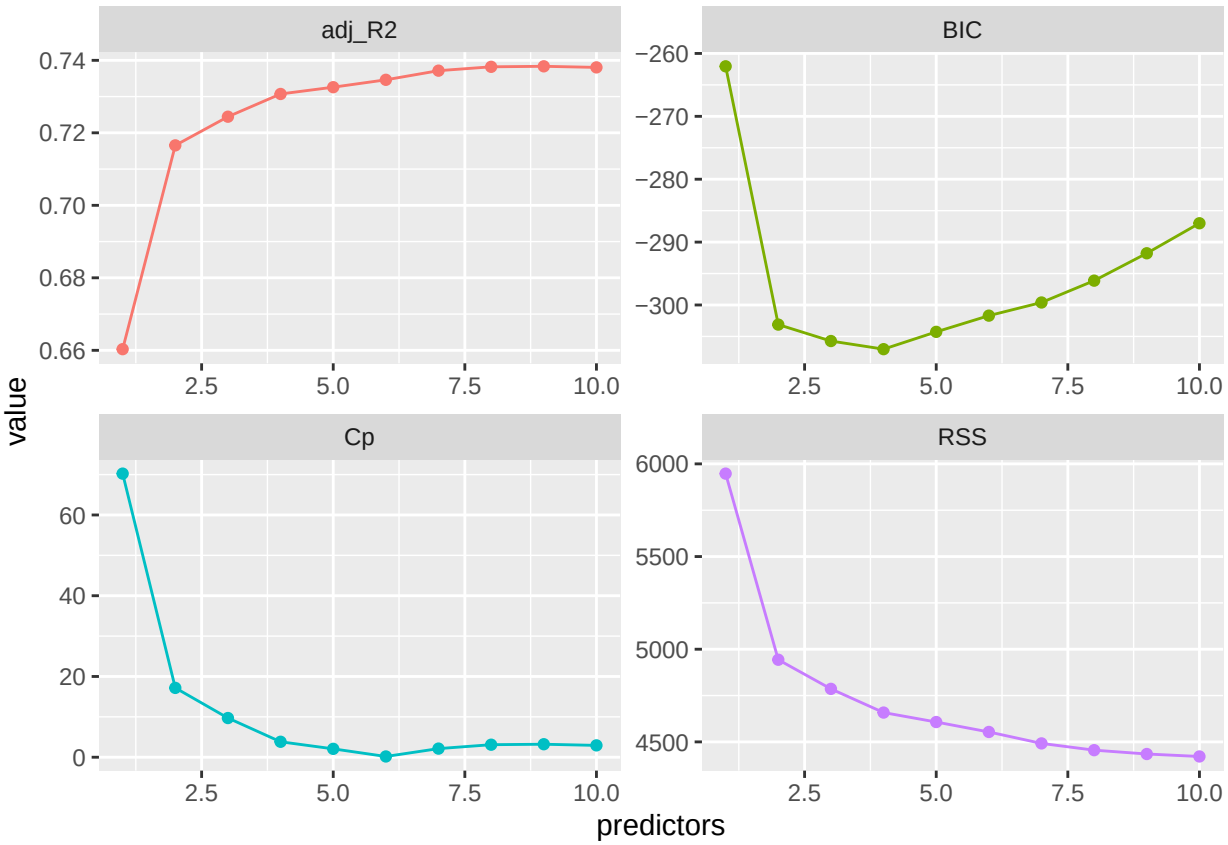
```
results <- summary(best_subset)
results
```

```
## Subset selection object
## Call: regsubsets.formula(bodyfat ~ ., bf_df, nvmax = 10)
## 13 Variables  (and intercept)
##         Forced in Forced out
## age         FALSE      FALSE
## weight      FALSE      FALSE
## height      FALSE      FALSE
```

```
## neck        FALSE      FALSE
## chest       FALSE      FALSE
## abdomen     FALSE      FALSE
## hip         FALSE      FALSE
## thigh       FALSE      FALSE
## knee        FALSE      FALSE
## ankle       FALSE      FALSE
## biceps      FALSE      FALSE
## forearm     FALSE      FALSE
## wrist       FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive
##           age weight height neck chest abdomen hip thigh knee ankle biceps
## 1  ( 1 )  " " " "    " "    " "  " "   "*"     " " " "   " "  " "   " "
## 2  ( 1 )  " " "*"    " "    " "  " "   "*"     " " " "   " "  " "   " "
## 3  ( 1 )  " " "*"    " "    " "  " "   "*"     " " " "   " "  " "   " "
## 4  ( 1 )  " " "*"    " "    " "  " "   "*"     " " " "   " "  " "   " "
## 5  ( 1 )  " " "*"    " "    "*"  " "   "*"     " " " "   " "  " "   " "
## 6  ( 1 )  "*" "*"    " "    " "  " "   "*"     " " "*"   " "  " "   " "
## 7  ( 1 )  "*" "*"    " "    "*"  " "   "*"     " " "*"   " "  " "   " "
## 8  ( 1 )  "*" "*"    " "    "*"  " "   "*"     "*" "*"   " "  " "   " "
## 9  ( 1 )  "*" "*"    " "    "*"  " "   "*"     "*" "*"   " "  " "   "*"
## 10  ( 1 ) "*" "*"    " "    "*"  " "   "*"     "*" "*"   " "  "*"   "*"
##           forearm wrist
## 1  ( 1 )  " "     " "
## 2  ( 1 )  " "     " "
## 3  ( 1 )  " "     "*"
## 4  ( 1 )  "*"     "*"
## 5  ( 1 )  "*"     "*"
## 6  ( 1 )  "*"     "*"
## 7  ( 1 )  "*"     "*"
## 8  ( 1 )  "*"     "*"
## 9  ( 1 )  "*"     "*"
## 10  ( 1 ) "*"     "*"
```

To choose the best value of k we can then look at our measures of model fit.

```
# extract and plot results
summ_results <- tibble(predictors = 1:10,
       adj_R2 = results$adjr2,
       Cp = abs(results$cp - 2:11),
       BIC = results$bic,
       RSS = results$rss)
summ_results %>%
  gather(statistic, value, -predictors) %>%
  ggplot(aes(predictors, value, color = statistic)) +
  geom_line(show.legend = F) +
  geom_point(show.legend = F) +
  facet_wrap(~ statistic, scales = "free")
```




```
summ_results
```

predictors	adj_R2	Cp	BIC	RSS
1	0.6603188	70.2433717	-262.0435	5947.463
2	0.7165395	17.1708899	-303.1197	4943.245
3	0.7244466	9.7068799	-305.7338	4786.054
4	0.7307199	3.8244473	-307.0259	4658.236
5	0.7325892	2.0747591	-304.2743	4607.169
6	0.7346244	0.1859836	-301.6966	4553.520
7	0.7371457	2.1347313	-299.6035	4491.849
8	0.7382101	3.1013984	-296.1315	4455.324
9	0.7383504	3.2166221	-291.7763	4434.613
10	0.7380516	2.9318135	-287.0028	4421.330

Automated model selection procedures

Forward and backward

```
library(MASS)
library(printr)
?stepAIC
```

```
Choose a model by AIC in a Stepwise Algorithm

Description:
```

```
     Performs stepwise model selection by AIC.

Usage:

     stepAIC(object, scope, scale = 0,
             direction = c("both", "backward", "forward"),
             trace = 1, keep = NULL, steps = 1000, use.start = FALSE,
             k = 2, ...)

Details:

     The set of models searched is determined by the 'scope' argument.
     The right-hand-side of its 'lower' component is always included in
     the model, and right-hand-side of the model is included in the
     'upper' component.  If 'scope' is a single formula, it specifies
     the 'upper' component, and the 'lower' model is empty.  If 'scope'
     is missing, the initial model is used as the 'upper' model.

     Models specified by 'scope' can be templates to update 'object' as
     used by 'update.formula'.

     There is a potential problem in using 'glm' fits with a variable
     'scale', as in that case the deviance is not simply related to the
     maximized log-likelihood. The 'glm' method for 'extractAIC' makes
     the appropriate adjustment for a 'gaussian' family, but may need
     to be amended for other cases. (The 'binomial' and 'poisson'
     families have fixed 'scale' by default and do not correspond to a
     particular maximum-likelihood problem for variable 'scale'.)

     Where a conventional deviance exists (e.g. for 'lm', 'aov' and
     'glm' fits) this is quoted in the analysis of variance table: it
     is the _unscaled_ deviance.
```

```r
bf_mod <- lm(bodyfat~.,data = bf_df)
back_mf <- stepAIC(bf_mod,direction= "backward")
```

```
## Start:  AIC=749.85
## bodyfat ~ age + weight + height + neck + chest + abdomen + hip +
##     thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - knee     1      0.06 4420.1 747.85
## - chest    1      0.51 4420.6 747.88
## - height   1      1.12 4421.2 747.91
## - ankle    1     11.86 4431.9 748.52
## - biceps   1     20.74 4440.8 749.03
## - hip      1     31.51 4451.6 749.64
## <none>                 4420.1 749.85
## - weight   1     45.10 4465.2 750.41
## - thigh    1     53.61 4473.7 750.89
## - age      1     74.72 4494.8 752.07
## - neck     1     75.66 4495.7 752.13
## - forearm  1     97.11 4517.2 753.33
## - wrist    1    178.85 4598.9 757.84
```

```
## - abdomen  1   2083.46 6503.5 845.17
##
## Step:  AIC=747.85
## bodyfat ~ age + weight + height + neck + chest + abdomen + hip +
##     thigh + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - chest    1      0.52 4420.6 745.88
## - height   1      1.06 4421.2 745.91
## - ankle    1     12.59 4432.7 746.57
## - biceps   1     20.68 4440.8 747.03
## - hip      1     31.47 4451.6 747.64
## <none>                 4420.1 747.85
## - weight   1     45.26 4465.4 748.42
## - thigh    1     60.46 4480.6 749.28
## - neck     1     77.09 4497.2 750.21
## - age      1     80.99 4501.1 750.43
## - forearm  1     98.18 4518.3 751.39
## - wrist    1    179.35 4599.5 755.88
## - abdomen  1   2083.40 6503.5 843.17
##
## Step:  AIC=745.88
## bodyfat ~ age + weight + height + neck + abdomen + hip + thigh +
##     ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - height   1      0.68 4421.3 743.92
## - ankle    1     12.90 4433.5 744.62
## - biceps   1     20.44 4441.1 745.04
## - hip      1     31.11 4451.8 745.65
## <none>                 4420.6 745.88
## - weight   1     64.84 4485.5 747.55
## - thigh    1     65.82 4486.5 747.61
## - neck     1     76.90 4497.5 748.23
## - age      1     80.68 4501.3 748.44
## - forearm  1     97.89 4518.5 749.40
## - wrist    1    178.96 4599.6 753.88
## - abdomen  1   2350.68 6771.3 851.34
##
## Step:  AIC=743.92
## bodyfat ~ age + weight + neck + abdomen + hip + thigh + ankle +
##     biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - ankle    1      13.3 4434.6 742.68
## - biceps   1      22.4 4443.7 743.19
## - hip      1      30.4 4451.8 743.65
## <none>                 4421.3 743.92
## - thigh    1      68.8 4490.1 745.81
## - neck     1      77.1 4498.4 746.27
## - age      1      81.3 4502.6 746.51
## - forearm  1      98.1 4519.4 747.45
## - weight   1     119.6 4540.9 748.65
## - wrist    1     181.3 4602.6 752.05
```

```
## - abdomen  1    3178.5 7599.9 878.43
##
## Step:  AIC=742.68
## bodyfat ~ age + weight + neck + abdomen + hip + thigh + biceps +
##     forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - biceps   1      20.7 4455.3 741.85
## - hip      1      31.7 4466.4 742.47
## <none>                 4434.6 742.68
## - thigh    1      72.3 4506.9 744.75
## - age      1      77.6 4512.2 745.05
## - neck     1      87.3 4521.9 745.59
## - forearm  1      97.4 4532.0 746.15
## - weight   1     107.2 4541.8 746.69
## - wrist    1     168.0 4602.6 750.05
## - abdomen  1    3182.0 7616.7 876.98
##
## Step:  AIC=741.85
## bodyfat ~ age + weight + neck + abdomen + hip + thigh + forearm +
##     wrist
##
##           Df Sum of Sq    RSS    AIC
## <none>                 4455.3 741.85
## - hip      1      36.5 4491.8 741.91
## - neck     1      79.1 4534.4 744.29
## - age      1      83.8 4539.1 744.55
## - weight   1      93.0 4548.3 745.05
## - thigh    1     100.7 4556.0 745.48
## - forearm  1     140.5 4595.8 747.67
## - wrist    1     166.8 4622.2 749.12
## - abdomen  1    3163.0 7618.3 875.04
```

```
back_mf$coefficients
```

```
##  (Intercept)          age       weight         neck      abdomen          hip
## -22.65637291   0.06577964  -0.08985290  -0.46655783   0.94481514  -0.19543492
##        thigh      forearm        wrist
##   0.30239157   0.51572117  -1.53665172
```

```
for_mf <- stepAIC(bf_mod,direction= "forward",trace = 0)
for_mf$coefficients
```

```
##  (Intercept)          age       weight       height         neck        chest
## -21.35323494   0.06457350  -0.09638287  -0.04393895  -0.47546758  -0.01718468
##      abdomen          hip        thigh         knee        ankle       biceps
##   0.95499717  -0.18858604   0.24834935   0.01394629   0.17788488   0.18230087
##      forearm        wrist
##   0.45573774  -1.65449992
```

Forward-Stagewise Regression

```
library(lars)
?lars
```

```
Fits Least Angle Regression, Lasso and Infinitesimal Forward Stagewise
```

```
regression models

Description:

     These are all variants of Lasso, and provide the entire sequence
     of coefficients and fits, starting from zero, to the least squares
     fit.

Usage:

     lars(x, y, type = c("lasso", "lar", "forward.stagewise", "stepwise"),
         trace = FALSE, normalize = TRUE, intercept = TRUE, Gram, eps = 1e-12,
          max.steps, use.Gram = TRUE)

Details:

     LARS is described in detail in Efron, Hastie, Johnstone and
     Tibshirani (2002). With the "lasso" option, it computes the
     complete lasso solution simultaneously for ALL values of the
     shrinkage parameter in the same computational cost as a least
     squares fit. A "stepwise" option has recently been added to LARS.
```

```r
ForStag_mf <- lars(as.matrix(bf_df[,-1]),bf_df$bodyfat,type="forward.stagewise",trace = TRUE)
```

```
Forward Stagewise sequence
Computing X'X .....
LARS Step 1 :    Variable 6     added
LARS Step 2 :    Variable 3     added
LARS Step 3 :    Variable 1     added
LARS Step 4 :    Variable 13    added
LARS Step 5 :    Variable 4     added
LARS Step 6 :    Variable 12    added
LARS Step 7 :    Variable 7     added
NNLS Step:   Variable 3     dropped
NNLS Step:   Variable 1     dropped
LARS Step 8 :    Variable 11    added
LARS Step 9 :    Variable 8     added
LARS Step 10 :   Variable 2     added
NNLS Step:   Variable 13    dropped
LARS Step 11 :   Variable 1     added
LARS Step 12 :   Variable 10    added
LARS Step 13 :   Variable 3     added
LARS Step 14 :   Variable 13    added
LARS Step 15 :   Variable 5     added
LARS Step 16 :   Variable 9     added
NNLS Step:   Variable 4     dropped
LARS Step 17 :   Variable 4     added
NNLS Step:   Variable 1     dropped
LARS Step 18 :   Variable 1     added
Computing residuals, RSS etc .....
```

```r
print(ForStag_mf)
```

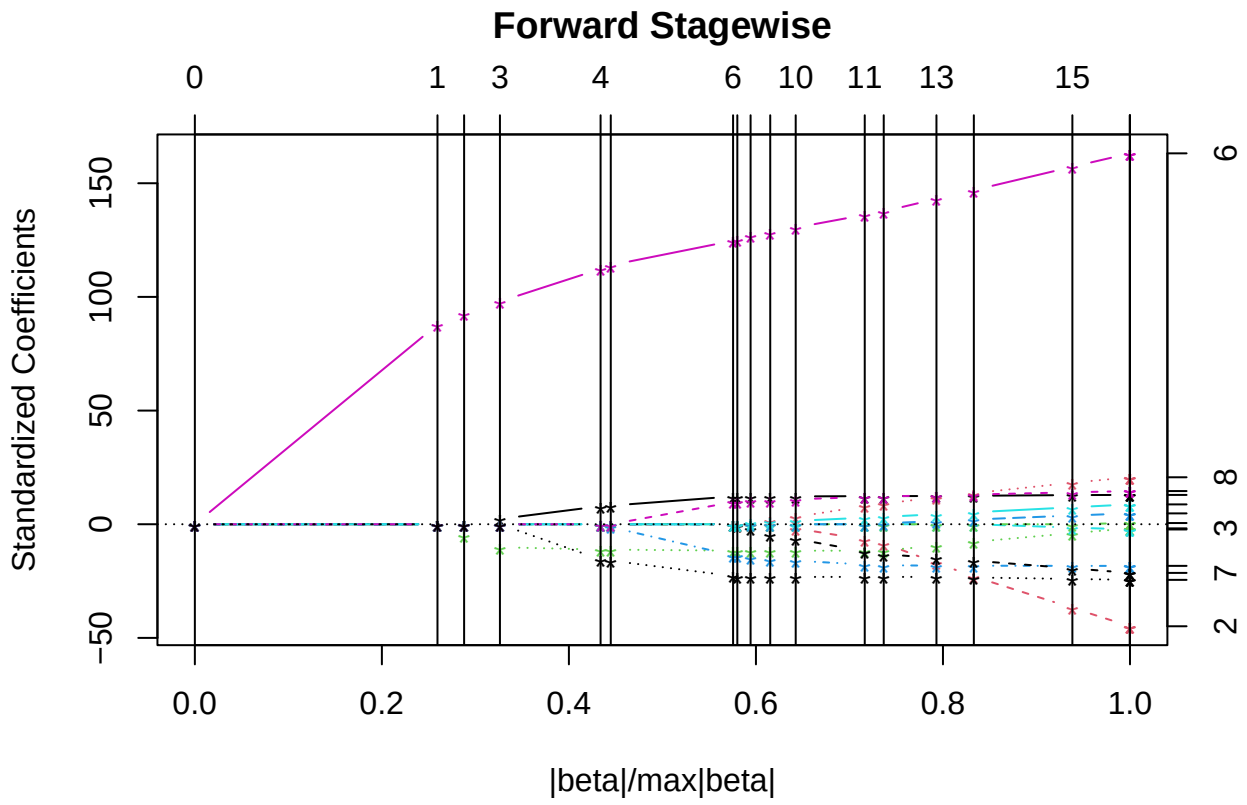
```
Call:
```

```
lars(x = as.matrix(bf_df[, -1]), y = bf_df$bodyfat, type = "forward.stagewise",
    trace = TRUE)
R-squared: 0.749
Sequence of Forward Stagewise moves:
     abdomen height age wrist neck forearm hip height age biceps thigh weight
Var        6      3   1    13    4      12   7     -3  -1     11     8      2
Step       1      2   3     4    5       6   7      7   7      8     9     10
     wrist age ankle height wrist chest knee neck neck age age
Var    -13   1    10      3    13     5    9   -4    4  -1   1
Step    10  11    12     13    14    15   16   16   17  17  18
```

```
round(coef(ForStag_mf),2)
```

age	weight	height	neck	chest	abdomen	hip	thigh	knee	ankle	biceps	forearm	wrist
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	-0.12	0.00	0.00	0.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.01	0.00	-0.24	0.00	0.00	0.57	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.04	0.00	-0.27	0.00	0.00	0.66	0.00	0.00	0.00	0.00	0.00	0.00	-1.05
0.04	0.00	-0.27	-0.03	0.00	0.67	0.00	0.00	0.00	0.00	0.00	0.00	-1.08
0.06	0.00	-0.28	-0.36	0.00	0.73	0.00	0.00	0.00	0.00	0.00	0.30	-1.54
0.06	0.00	-0.28	-0.36	0.00	0.73	0.00	0.00	0.00	0.00	0.00	0.31	-1.55
0.06	0.00	-0.28	-0.38	0.00	0.74	-0.02	0.00	0.00	0.00	0.01	0.32	-1.56
0.06	0.00	-0.28	-0.40	0.00	0.75	-0.04	0.02	0.00	0.00	0.02	0.33	-1.57
0.06	0.00	-0.28	-0.42	0.00	0.76	-0.06	0.04	0.00	0.00	0.03	0.34	-1.57
0.06	-0.01	-0.28	-0.46	0.00	0.80	-0.11	0.10	0.00	0.00	0.06	0.37	-1.57
0.06	-0.02	-0.28	-0.47	0.00	0.81	-0.12	0.11	0.00	0.02	0.07	0.38	-1.57
0.06	-0.04	-0.23	-0.47	0.00	0.84	-0.13	0.14	0.00	0.05	0.09	0.40	-1.57
0.06	-0.05	-0.19	-0.47	0.00	0.86	-0.14	0.16	0.00	0.08	0.11	0.41	-1.58
0.06	-0.08	-0.09	-0.48	-0.01	0.92	-0.17	0.22	0.00	0.14	0.16	0.44	-1.63
0.06	-0.10	-0.04	-0.48	-0.02	0.95	-0.19	0.25	0.01	0.18	0.18	0.46	-1.65
0.06	-0.10	-0.04	-0.48	-0.02	0.95	-0.19	0.25	0.01	0.18	0.18	0.46	-1.65
0.06	-0.10	-0.04	-0.48	-0.02	0.95	-0.19	0.25	0.01	0.18	0.18	0.46	-1.65

```
plot(ForStag_mf)
```

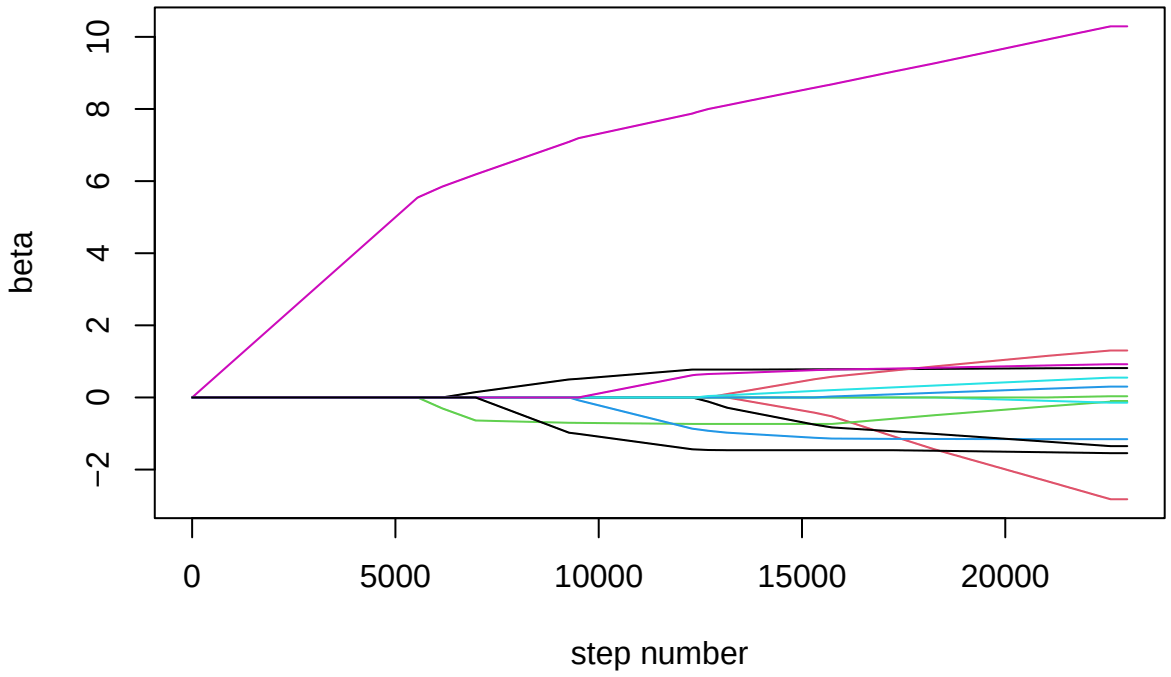


Also, we can complete this algorithm by ourselves

___code courtesy of Mark H. Hansen (http://www.stat.ucla.edu/~cocteau/)___

```
y <- bf_df$bodyfat
M <- as.matrix(bf_df[,-1])
y <- y-mean(y)
M <- M-matrix(apply(M,2,mean),ncol=ncol(M),nrow=nrow(M),byrow=T)
M <- M/matrix(apply(M,2,sd),ncol=ncol(M),nrow=nrow(M),byrow=T)
beta <- matrix(0,ncol=ncol(M),nrow=1)
r <- y
eps <- 0.001
lots <- 23000
for(i in 1:lots){
  co <- t(M)%*%r
  j <- (1:ncol(M))[abs(co)==max(abs(co))][1]
  delta <- eps*sign(co[j])
  b <- beta[nrow(beta),]
  b[j] <- b[j] + delta
  beta <- rbind(beta,b)
  r <- r - delta*M[,j]
  }

matplot(beta,type="l",lty=1,xlab="step number",ylab="beta",main="stagewise")
```




```
beta_mat <- rbind(beta[c(1,seq(5000,20000,5000),lots),])
colnames(beta_mat) <- colnames(M)
rownames(beta_mat) <- c(1,seq(5000,20000,5000),lots)
round(beta_mat, 2)
```

	age	weight	height	neck	chest	abdomen	hip	thigh	knee	ankle	biceps	forearm	wrist
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.00
5000	0.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.0	0.00	0.00	0.00
10000	0.56	0.00	-0.71	-0.22	0.00	7.31	0.00	0.00	0.00	0.0	0.00	0.11	-1.09
15000	0.78	-0.36	-0.74	-1.10	0.00	8.52	-0.69	0.45	0.00	0.0	0.16	0.75	-1.46
20000	0.80	-1.99	-0.34	-1.15	-0.06	9.68	-1.15	1.04	0.00	0.2	0.42	0.86	-1.50
23000	0.82	-2.82	-0.12	-1.16	-0.15	10.29	-1.35	1.30	0.03	0.3	0.55	0.92	-1.54