Application

Yao Lu

```r
#case
setwd("C:/Users/Administrator/Documents/frmselection/data-raw")
dat <- read.table("401kjae.txt")
summary(dat)
```

```
##        V1                V2                V3               V4       
##  Min.   :0.02319   Min.   :0.01104   Min.   :    53   Min.   : 4.00  
##  1st Qu.:0.78029   1st Qu.:0.27008   1st Qu.:   278   1st Qu.: 7.00  
##  Median :0.93671   Median :0.43985   Median :   628   Median : 8.00  
##  Mean   :0.86956   Mean   :0.74633   Mean   :  4621   Mean   :13.14  
##  3rd Qu.:1.00000   3rd Qu.:0.83593   3rd Qu.:  2173   3rd Qu.:17.00  
##  Max.   :1.00000   Max.   :5.00000   Max.   :443040   Max.   :76.00  
##        V5        
##  Min.   :0.0000  
##  1st Qu.:0.0000  
##  Median :0.0000  
##  Mean   :0.4149  
##  3rd Qu.:1.0000  
##  Max.   :1.0000  
```

```r
#The standard deviation of variables
sqrt(diag(var(dat)))
```

```
##           V1           V2           V3           V4           V5 
## 1.668051e-01 8.443782e-01 1.629964e+04 9.629316e+00 4.927518e-01 
```

```r
y <- dat[,1]
x <- cbind(dat[,2], log(dat[,3]), log(dat[,3])^2, dat[,4], (dat[,4])^2, dat[,5])
colnames(x) <- c("mrate", "lemp", "lemp2","age", "age2","sole")
```

```r
#QMLE
library(frm)
```

```
## Warning: package 'frm' was built under R version 3.4.1
```

```r
f <- frm(y, x, linkfrac = "logit", table = FALSE)
f$p
```

```
##     INTERCEPT         mrate          lemp         lemp2           age 
##  5.4293327516  0.5420939070 -1.0384143315  0.0540149530  0.0621111389 
##          age2          sole 
## -0.0007078292  0.1189706257 
```

```r
#The standard deviation of estimated coefficients
sqrt(diag(f$p.var))
```

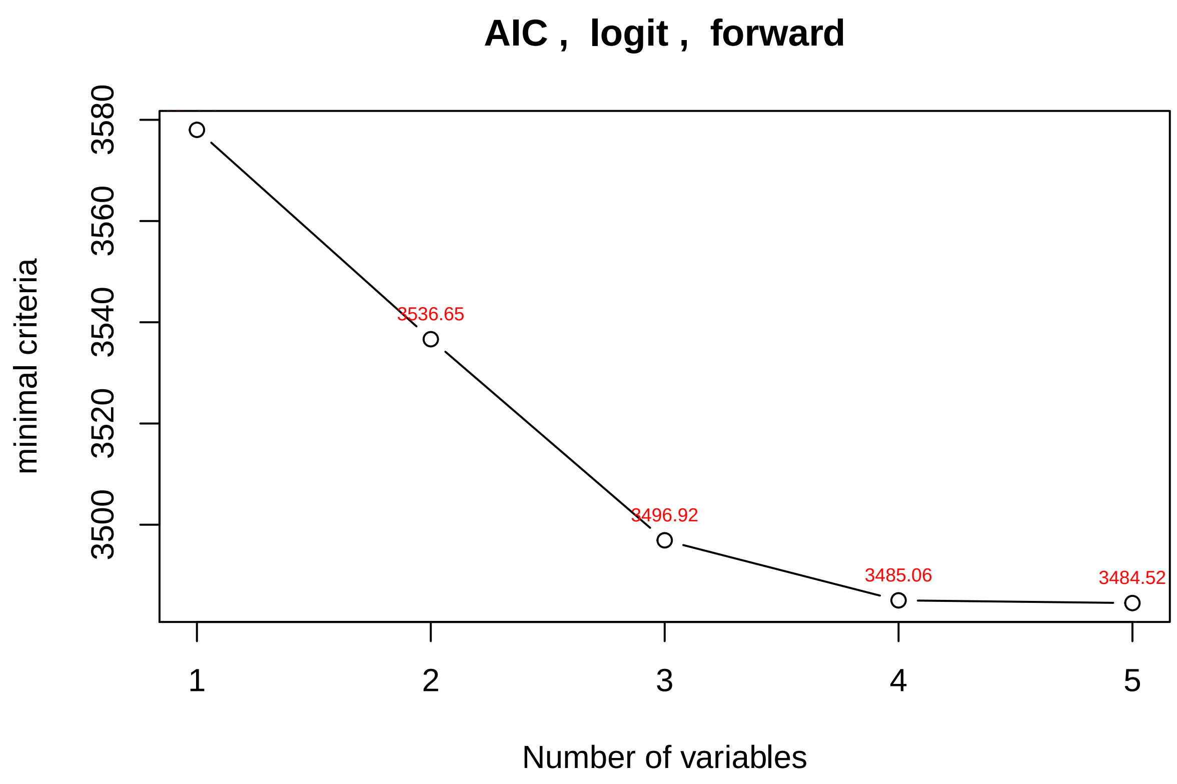
```
##   INTERCEPT       mrate        lemp       lemp2         age        age2 
## 0.422046856 0.078655656 0.110476930 0.007101361 0.007818906 0.000179411 
##        sole 
## 0.050254398 
```

```r
x <- as.data.frame(x, row.names = TRUE)
#OLS
```

```r
ols <- lm(y ~ x$mrate + x$lemp + x$lemp2 + x$age + x$age2 +x$sole)
coef(ols)
```

```
##   (Intercept)       x$mrate        x$lemp       x$lemp2         x$age
##  1.213348e+00  3.406984e-02 -1.012416e-01  5.143599e-03  6.444477e-03
##        x$age2        x$sole
## -7.805808e-05  1.406416e-02
```

```r
#frmselect() to do the model selection
#The four examples
x <- as.matrix(x, row.names = TRUE)
library(frmselection)
frmselect(x,y, criterion = "AIC",linkfrac = "logit", plotit=TRUE)#The default is forward
```



```
## $criterion
## [1] "AIC"
##
## $linkfrac
## [1] "logit"
##
## $method
## [1] "forward"
##
## $criteria
## [1] 3578.014 3536.654 3496.923 3485.057 3484.524
##
## $min_criteria
## [1] 3484.524
##
## $index
## [1] 1 2 4 3 5
##
## $variable
## [1] "mrate" "lemp"  "age"   "lemp2" "age2"
##
## $coefficient
##     INTERCEPT         mrate          lemp         lemp2           age
##  5.6875150627  0.5549711372 -1.0877563559  0.0563963401  0.0622834183
##          age2
## -0.0007116164
```

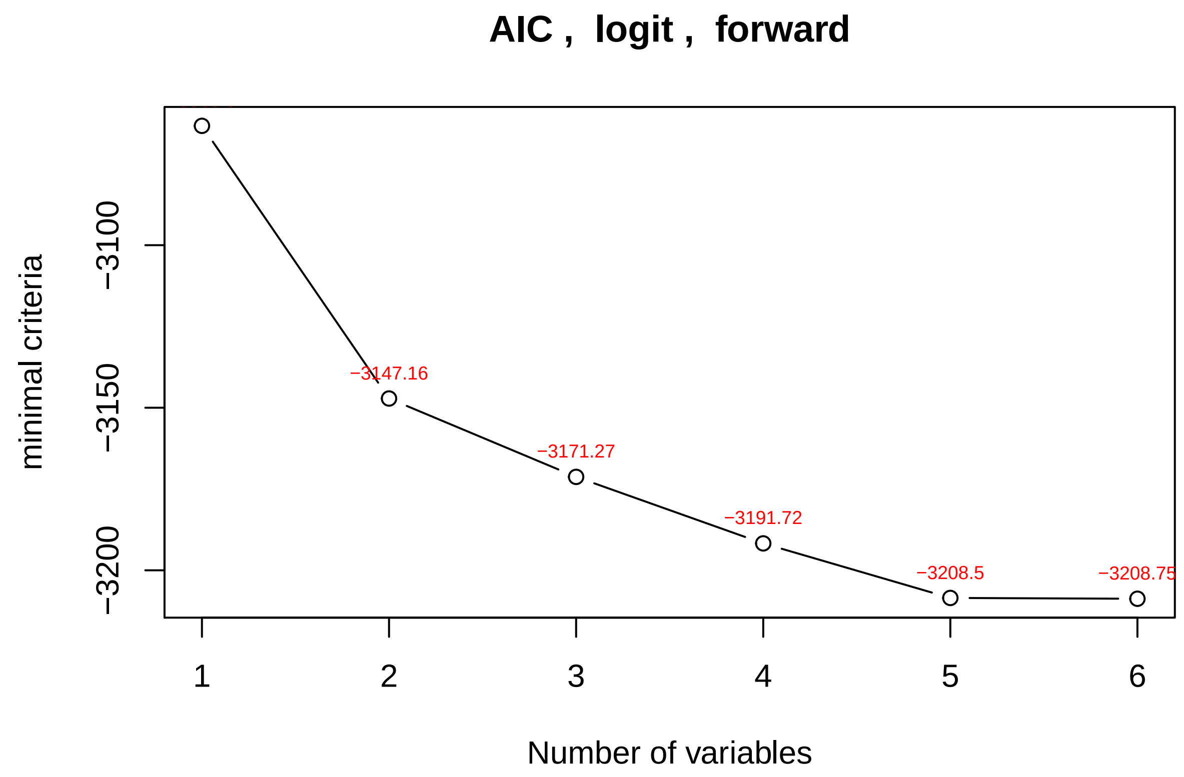
```
#frmselect(x,y, criterion = "AIC",linkfrac = "logit", method = "allsubsets")
#frmselect(x,y, criterion = "AIC",linkfrac = "logit", method = "backward")
#frmselect(x,y, criterion = "AIC",linkfrac = "logit", method = "both")
```

```
#betaselect() to do the model selection
#The four examples
newdat <- data.frame(y, x)
subdat <- newdat[y<1,]
suby <- subdat[,1]
subx <- subdat[,-1]
subx <- as.matrix(subdat[,-1])
nrow(subx)
```

```
## [1] 2711
```

```
betaselect(subx, suby, plotit = TRUE)
```

```
## Warning in min(na.omit(criteria)): no non-missing arguments to min;
## returning Inf
```



```
## $criterion
## [1] "AIC"
##
## $link
## [1] "logit"
##
## $method
## [1] "forward"
##
## $criteria
## [1] -3063.299 -3147.160 -3171.267 -3191.722 -3208.503 -3208.748
##
## $min_criteria
## [1] -3208.748
##
## $index
## [1] 4 2 3 5 1 6
##
## $variable
## [1] "age"   "lemp"  "lemp2" "age2"  "mrate" "sole"
##
## $coefficient
##     (Intercept) x[, index]mrate  x[, index]lemp x[, index]lemp2
##    3.1074721030    0.1348000668   -0.5657128718    0.0301189346
##   x[, index]age  x[, index]age2  x[, index]sole           (phi)
##    0.0544227457   -0.0007579094   -0.0552907457    6.1747102148
```

```r
#betaselect(subx, suby, method = "backward")
#betaselect(subx, suby, method = "both")
#betaselect(subx, suby, method = "allsubsets")
```

```r
#Another way by replacing 1 with 0.9999
y[y==1] <- 0.9999
betaselect(x, y)
```

```
## $criterion
## [1] "AIC"
##
## $link
## [1] "logit"
##
## $method
## [1] "forward"
##
## $criteria
## [1] -23095.15 -23286.16 -23352.28 -23413.18 -23426.17
##
## $min_criteria
## [1] -23426.17
##
## $index
## [1] 1 2 4 6 3
##
## $variable
## [1] "mrate" "lemp"  "age"   "sole"  "lemp2"
##
## $coefficient
##     (Intercept) x[, index]mrate  x[, index]lemp x[, index]lemp2
##      3.53993100      0.28415390     -0.43861267      0.02071822
##   x[, index]age  x[, index]sole           (phi)
##      0.01263720      0.25381206      2.22938411
```

```r
#betaselect(x, y, method = "backward")
#betaselect(x, y, metihod = "both")
#betaselect(x, y, method = "allsubsets")
```

```r
#Testing whether the modified quasi-binomial family frm_bamlss()function works
x <- as.data.frame(x, row.names = TRUE)
y <- dat[,1]
d <- data.frame(y,x)
formula <- as.formula(paste("y ~ ", paste(colnames(x), collapse = "+")))
library(bamlss)
```

```
## Warning: package 'bamlss' was built under R version 3.4.4

## Loading required package: coda

## Warning: package 'coda' was built under R version 3.4.4

## Loading required package: colorspace

## Warning: package 'colorspace' was built under R version 3.4.4

## Loading required package: mgcv
## Loading required package: nlme

## This is mgcv 1.8-17. For overview type 'help("mgcv-package")'.

##
## Attaching package: 'bamlss'

## The following object is masked from 'package:mgcv':
##
##     smooth.construct
```

```
b <- bamlss(formula, data = d, family = frm_bamlss(link = "logit") ,sampler = FALSE,
            multiple = FALSE)
```

```
## AICc -7635.33 logPost 3769.892 logLik 3824.678 edf 7.0000 eps 1.0000 iteration   1
## AICc -8037.17 logPost 3970.810 logLik 4025.596 edf 7.0000 eps 0.2019 iteration   2
## AICc -8116.95 logPost 4010.703 logLik 4065.490 edf 7.0000 eps 0.0643 iteration   3
## AICc -8127.98 logPost 4016.216 logLik 4071.002 edf 7.0000 eps 0.0120 iteration   4
## AICc -8128.43 logPost 4016.443 logLik 4071.230 edf 7.0000 eps 0.0006 iteration   5
## AICc -8128.43 logPost 4016.444 logLik 4071.231 edf 7.0000 eps 0.0000 iteration   6
## AICc -8128.43 logPost 4016.444 logLik 4071.231 edf 7.0000 eps 0.0000 iteration   6
## elapsed time:  0.47sec
```

```
coef(b)
```

```
## pi.p.(Intercept)       pi.p.mrate        pi.p.lemp       pi.p.lemp2
##     5.4293327517     0.5420939069    -1.0384143315     0.0540149530
##         pi.p.age        pi.p.age2        pi.p.sole
##     0.0621111389    -0.0007078292     0.1189706257
```

```
f <- frm::frm(y, x, linkfrac = "logit", table = FALSE)
f$p
```

```
##     INTERCEPT         mrate          lemp         lemp2           age
##  5.4293327516  0.5420939070 -1.0384143315  0.0540149530  0.0621111389
##          age2          sole
## -0.0007078292  0.1189706257
```

```
#probit link
bsub <- bamlss(formula, data = subdat, family = frm_bamlss(link = "probit"),
               sampler = FALSE, multiple = FALSE)
```

```
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 1.0000 iteration   1
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 2.0980 iteration   2
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.6246 iteration   3
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.6289 iteration   4
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.3243 iteration   5
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.2807 iteration   6
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.1738 iteration   7
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.1414 iteration   8
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0955 iteration   9
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0760 iteration  10
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0545 iteration  11
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0434 iteration  12
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0324 iteration  13
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0259 iteration  14
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0199 iteration  15
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0160 iteration  16
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0125 iteration  17
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0100 iteration  18
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0079 iteration  19
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0063 iteration  20
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0050 iteration  21
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0040 iteration  22
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0032 iteration  23
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0025 iteration  24
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0020 iteration  25
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0016 iteration  26
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0013 iteration  27
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0010 iteration  28
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0008 iteration  29
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0006 iteration  30
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0005 iteration  31
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0004 iteration  32
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0003 iteration  33
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0002 iteration  34
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0002 iteration  35
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0001 iteration  36
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0001 iteration  37
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0001 iteration  38
## AICc -1580.38 logPost 742.4243 logLik 797.2111 edf 7.0000 eps 0.0001 iteration  38
## elapsed time:  0.40sec
```

```
coef(bsub)
```

```
## pi.p.(Intercept)       pi.p.mrate        pi.p.lemp       pi.p.lemp2
##     2.1795009401     0.0389236239    -0.4087798643     0.0218251310
##         pi.p.age        pi.p.age2        pi.p.sole
##     0.0384463457    -0.0005703433    -0.0982876292
```

```
f1 <- frm(suby, subx, linkfrac = "probit", table = FALSE)
f1$p
```

```
##     INTERCEPT         mrate          lemp         lemp2           age
##  2.1718763071  0.0366391391 -0.4063760349  0.0217071868  0.0381034837
##          age2          sole
## -0.0005654725 -0.0960132316
```

```
#Lasso Procedure
#Using quasi-binomial family
dat <- read.table("401kjae.txt")
y <- dat[,1]
x <- cbind(dat[,2], log(dat[,3]), log(dat[,3])^2, dat[,4], (dat[,4])^2, dat[,5])
colnames(x) <- c("mrate", "lemp", "lemp2","age", "age2","sole")
x <- as.data.frame(x, row.names = TRUE)
x$sole <- as.factor(x$sole)
d <- data.frame(y,x)
xname <- colnames(x)
ind <- sapply(x, is.factor)
ind <- which(ind) #find the index of the factor variables
f1 <- as.formula(paste(" ~ ", paste(xname[-ind], collapse = "+")))
f2 <- as.formula(paste(" ~ ", paste(xname[ind], collapse = "+")))
formula <- y ~ la(f1) + la(f2) #form the formula with two kinds of variables
b <- bamlss(formula, data = d, family = frm_bamlss(link = "logit"), sampler = FALSE,
            optimizer = lasso, nlambda = 100, upper = 1e+08, lower = 1e+03,
            multiple = FALSE)
```

```
## AICc -8072.81 edf 1.0000 lambda 100000 iteration   1
## AICc -8072.81 edf 1.0000 lambda 890215 iteration   2
## AICc -8072.81 edf 1.0000 lambda 792482 iteration   3
## AICc -8072.81 edf 1.0000 lambda 705480 iteration   4
## AICc -8072.81 edf 1.0001 lambda 628029 iteration   5
## AICc -8072.81 edf 1.0001 lambda 559081 iteration   6
## AICc -8072.81 edf 1.0001 lambda 497702 iteration   7
## AICc -8072.81 edf 1.0001 lambda 443062 iteration   8
## AICc -8072.81 edf 1.0001 lambda 394420 iteration   9
## AICc -8072.81 edf 1.0001 lambda 351119 iteration  10
## AICc -8072.81 edf 1.0001 lambda 312571 iteration  11
## AICc -8072.81 edf 1.0001 lambda 278255 iteration  12
## AICc -8072.81 edf 1.0001 lambda 247707 iteration  13
## AICc -8072.81 edf 1.0001 lambda 220513 iteration  14
## AICc -8072.81 edf 1.0002 lambda 196304 iteration  15
## AICc -8072.82 edf 1.0002 lambda 174752 iteration  16
## AICc -8072.82 edf 1.0002 lambda 155567 iteration  17
## AICc -8072.82 edf 1.0002 lambda 138488 iteration  18
## AICc -8072.82 edf 1.0003 lambda 123284 iteration  19
## AICc -8072.82 edf 1.0003 lambda 109749 iteration  20
## AICc -8072.82 edf 1.0003 lambda 977009 iteration  21
## AICc -8072.82 edf 1.0004 lambda 869749 iteration  22
## AICc -8072.82 edf 1.0004 lambda 774263 iteration  23
## AICc -8072.82 edf 1.0005 lambda 689261 iteration  24
## AICc -8072.82 edf 1.0005 lambda 613590 iteration  25
## AICc -8072.82 edf 1.0006 lambda 546227 iteration  26
## AICc -8072.82 edf 1.0007 lambda 486260 iteration  27
## AICc -8072.82 edf 1.0007 lambda 432876 iteration  28
## AICc -8072.82 edf 1.0008 lambda 385352 iteration  29
## AICc -8072.83 edf 1.0009 lambda 343046 iteration  30
## AICc -8072.83 edf 1.0011 lambda 305385 iteration  31
## AICc -8072.83 edf 1.0012 lambda 271858 iteration  32
## AICc -8072.83 edf 1.0013 lambda 242012 iteration  33
## AICc -8072.83 edf 1.0015 lambda 215443 iteration  34
## AICc -8072.84 edf 1.0017 lambda 191791 iteration  35
## AICc -8072.84 edf 1.0019 lambda 170735 iteration  36
## AICc -8072.84 edf 1.0021 lambda 151991 iteration  37
## AICc -8072.84 edf 1.0024 lambda 135304 iteration  38
## AICc -8072.85 edf 1.0027 lambda 120450 iteration  39
## AICc -8072.85 edf 1.0030 lambda 107226 iteration  40
## AICc -8072.86 edf 1.0034 lambda 954548 iteration  41
## AICc -8072.86 edf 1.0038 lambda 849753 iteration  42
## AICc -8072.87 edf 1.0043 lambda 756463 iteration  43
## AICc -8072.88 edf 1.0048 lambda 673415 iteration  44
## AICc -8072.88 edf 1.0054 lambda 599484 iteration  45
## AICc -8072.89 edf 1.0060 lambda 533669 iteration  46
## AICc -8072.90 edf 1.0068 lambda 475081 iteration  47
## AICc -8072.91 edf 1.0076 lambda 422924 iteration  48
## AICc -8072.93 edf 1.0085 lambda 376493 iteration  49
## AICc -8072.94 edf 1.0096 lambda 335160 iteration  50
## AICc -8072.96 edf 1.0108 lambda 298364 iteration  51
## AICc -8072.97 edf 1.0121 lambda 265608 iteration  52
## AICc -8072.99 edf 1.0136 lambda 236448 iteration  53
## AICc -8073.02 edf 1.0152 lambda 210490 iteration  54
## AICc -8073.04 edf 1.0171 lambda 187381 iteration  55
## AICc -8073.07 edf 1.0192 lambda 166810 iteration  56
## AICc -8073.10 edf 1.0215 lambda 148496 iteration  57
## AICc -8073.14 edf 1.0242 lambda 132194 iteration  58
## AICc -8073.17 edf 1.0271 lambda 117681 iteration  59
## AICc -8073.22 edf 1.0304 lambda 104761 iteration  60
## AICc -8073.27 edf 1.0341 lambda 93260. iteration  61
## AICc -8073.32 edf 1.0383 lambda 83021. iteration  62
## AICc -8073.39 edf 1.0430 lambda 73907. iteration  63
## AICc -8073.46 edf 1.0482 lambda 65793. iteration  64
## AICc -8073.53 edf 1.0540 lambda 58570. iteration  65
## AICc -8073.62 edf 1.0605 lambda 52140. iteration  66
## AICc -8073.72 edf 1.0678 lambda 46415. iteration  67
## AICc -8073.83 edf 1.0760 lambda 41320. iteration  68
## AICc -8073.95 edf 1.0851 lambda 36783. iteration  69
## AICc -8074.09 edf 1.0953 lambda 32745. iteration  70
## AICc -8074.24 edf 1.1066 lambda 29150. iteration  71
## AICc -8074.41 edf 1.1192 lambda 25950. iteration  72
## AICc -8074.60 edf 1.1333 lambda 23101. iteration  73
## AICc -8074.81 edf 1.1489 lambda 20565. iteration  74
## AICc -8075.04 edf 1.1663 lambda 18307. iteration  75
## AICc -8075.30 edf 1.1856 lambda 16297. iteration  76
## AICc -8075.58 edf 1.2069 lambda 14508. iteration  77
## AICc -8075.90 edf 1.2305 lambda 12915. iteration  78
## AICc -8076.25 edf 1.2566 lambda 11497. iteration  79
## AICc -8076.64 edf 1.2853 lambda 10235. iteration  80
## AICc -8077.06 edf 1.3169 lambda 9111.6 iteration  81
## AICc -8077.53 edf 1.3516 lambda 8111.3 iteration  82
## AICc -8078.04 edf 1.3896 lambda 7220.8 iteration  83
## AICc -8078.60 edf 1.4310 lambda 6428.0 iteration  84
## AICc -8079.21 edf 1.4761 lambda 5722.3 iteration  85
## AICc -8079.87 edf 1.5250 lambda 5094.1 iteration  86
## AICc -8080.58 edf 1.5780 lambda 4534.8 iteration  87
## AICc -8081.35 edf 1.6350 lambda 4037.0 iteration  88
## AICc -8082.17 edf 1.6963 lambda 3593.8 iteration  89
## AICc -8083.05 edf 1.7618 lambda 3199.2 iteration  90
## AICc -8083.99 edf 1.8316 lambda 2848.0 iteration  91
## AICc -8084.98 edf 1.9056 lambda 2535.3 iteration  92
## AICc -8086.02 edf 1.9837 lambda 2257.0 iteration  93
## AICc -8087.12 edf 2.0657 lambda 2009.2 iteration  94
## AICc -8088.26 edf 2.1515 lambda 1788.6 iteration  95
## AICc -8089.44 edf 2.2407 lambda 1592.2 iteration  96
## AICc -8090.66 edf 2.3329 lambda 1417.4 iteration  97
## AICc -8091.91 edf 2.4281 lambda 1261.8 iteration  98
## AICc -8093.18 edf 2.5257 lambda 1123.3 iteration  99
## AICc -8094.47 edf 2.6252 lambda 1000.0 iteration 100
## elapsed time:  4.36sec
```

```r
coefi <- lasso.coef(b)
coefi
```

```
## pi.s.la(f1).mrate  pi.s.la(f1).lemp pi.s.la(f1).lemp2   pi.s.la(f1).age
##      0.1354350367     -0.0547313090     -0.0030766660      0.0086310708
##  pi.s.la(f1).age2 pi.s.la(f1).tau21 pi.s.la(f2).sole1 pi.s.la(f2).tau21
##      0.0001524879      0.0010000000      0.0789572173      0.0010000000
##  pi.p.(Intercept)
##      1.8888246661
```

```
lasso.select <- function(x, coefficent, threshold = 1e-3){
  if(!is.numeric(threshold)){
    stop("Threshold should be a number!")
  }
  n <- length(coefficent) #n is the number of the coefficients
  ncommon <- length(x[,-ind]) #the number of the common variables
  nfactor <- length(levels(x[,ind]))  #find the number of levels of factor variables
  #nfactor <- length(unique(x[,ind]))
  #Extract the penalized parameters of common and factor variables separately.
  tau.common <- coefficent[ncommon+1]
  tau.factor <- coefficent[ncommon+nfactor+1]
  #The relation between tau and lambda is an inverse relationship
  lambda.common <- 1/tau.common
  names(lambda.common) <- "mu.s.la(f1).lambda"
  lambda.factor <- 1/tau.factor
  names(lambda.factor) <- "mu.s.la(f2).lambda"
  index.tau <- c(ncommon+1, ncommon+nfactor+1)
  coefi.new <- coefficent[-index.tau] #the new coeffients doesn't contain the penalized parameters
  lasso.index <- NULL
  #Users can set this threshold as whatever they like. The default value is 1e-3
  for(i in 1:length(coefi.new)){
    if(abs(coefi.new[i]) < threshold){
      coefi.new[i] = 0
      lasso.index <- c(lasso.index, i)
    }
  }
  return(list(lambda.common = lambda.common, lambda.factor = lambda.factor,
              lasso.index = lasso.index, modified.coefficients = coefi.new))
}
lasso.select(x, coefi)
```

```
## $lambda.common
## mu.s.la(f1).lambda
##               1000
##
## $lambda.factor
## mu.s.la(f2).lambda
##               1000
##
## $lasso.index
## [1] 5
##
## $modified.coefficients
## pi.s.la(f1).mrate  pi.s.la(f1).lemp pi.s.la(f1).lemp2   pi.s.la(f1).age
##       0.135435037      -0.054731309      -0.003076666       0.008631071
##  pi.s.la(f1).age2 pi.s.la(f2).sole1  pi.p.(Intercept)
##       0.000000000       0.078957217       1.888824666
```

```
#Using beta family
subdat <- d[y<1,]
```

```r
suby <- subdat[,1]
subx <- subdat[,-1]
xname <- colnames(subx)
ind1 <- sapply(subx, is.factor)
ind1 <- which(ind1) #find the index of the factor variables
f11 <- as.formula(paste(" ~ ", paste(xname[-ind], collapse = "+")))
f21 <- as.formula(paste(" ~ ", paste(xname[ind], collapse = "+")))
formula1 <- y ~ la(f11) + la(f21) #form the formula with two kinds of variables
b1 <- bamlss(formula1, data = subdat, family = "beta", sampler = FALSE,
            optimizer = lasso, nlambda = 10, upper = 10e+8, lower = 10e+3,
            multiple = TRUE)
```

```
## AICc -2940.64 edf 2.0000 lambda 100000,100000 iteration  1
## AICc -2940.65 edf 2.0001 lambda 278255,100000 iteration  2
## AICc -2940.67 edf 2.0003 lambda 774263,100000 iteration  3
## AICc -2940.75 edf 2.0010 lambda 215443,100000 iteration  4
## AICc -2941.03 edf 2.0037 lambda 599484,100000 iteration  5
## AICc -2942.05 edf 2.0134 lambda 166810,100000 iteration  6
## AICc -2945.65 edf 2.0476 lambda 464158,100000 iteration  7
## AICc -2957.77 edf 2.1653 lambda 129154,100000 iteration  8
## AICc -2993.04 edf 2.5319 lambda 35938.,100000 iteration  9
## AICc -3061.40 edf 3.4022 lambda 10000.,100000 iteration 10
## AICc -2940.64 edf 2.0000 lambda 100000,278255 iteration 11
## AICc -2940.65 edf 2.0001 lambda 278255,278255 iteration 12
## AICc -2940.67 edf 2.0003 lambda 774263,278255 iteration 13
## AICc -2940.75 edf 2.0010 lambda 215443,278255 iteration 14
## AICc -2941.03 edf 2.0037 lambda 599484,278255 iteration 15
## AICc -2942.05 edf 2.0134 lambda 166810,278255 iteration 16
## AICc -2945.65 edf 2.0476 lambda 464158,278255 iteration 17
## AICc -2957.77 edf 2.1653 lambda 129154,278255 iteration 18
## AICc -2993.04 edf 2.5319 lambda 35938.,278255 iteration 19
## AICc -3061.40 edf 3.4022 lambda 10000.,278255 iteration 20
## AICc -2940.64 edf 2.0000 lambda 100000,774263 iteration 21
## AICc -2940.65 edf 2.0001 lambda 278255,774263 iteration 22
## AICc -2940.67 edf 2.0003 lambda 774263,774263 iteration 23
## AICc -2940.75 edf 2.0010 lambda 215443,774263 iteration 24
## AICc -2941.03 edf 2.0037 lambda 599484,774263 iteration 25
## AICc -2942.05 edf 2.0134 lambda 166810,774263 iteration 26
## AICc -2945.65 edf 2.0476 lambda 464158,774263 iteration 27
## AICc -2957.77 edf 2.1653 lambda 129154,774263 iteration 28
## AICc -2993.04 edf 2.5319 lambda 35938.,774263 iteration 29
## AICc -3061.40 edf 3.4022 lambda 10000.,774263 iteration 30
## AICc -2940.64 edf 2.0000 lambda 100000,215443 iteration 31
## AICc -2940.65 edf 2.0001 lambda 278255,215443 iteration 32
## AICc -2940.67 edf 2.0003 lambda 774263,215443 iteration 33
## AICc -2940.75 edf 2.0010 lambda 215443,215443 iteration 34
## AICc -2941.03 edf 2.0037 lambda 599484,215443 iteration 35
## AICc -2942.05 edf 2.0134 lambda 166810,215443 iteration 36
## AICc -2945.65 edf 2.0476 lambda 464158,215443 iteration 37
## AICc -2957.77 edf 2.1653 lambda 129154,215443 iteration 38
## AICc -2993.04 edf 2.5319 lambda 35938.,215443 iteration 39
## AICc -3061.40 edf 3.4022 lambda 10000.,215443 iteration 40
## AICc -2940.64 edf 2.0000 lambda 100000,599484 iteration 41
## AICc -2940.65 edf 2.0001 lambda 278255,599484 iteration 42
## AICc -2940.67 edf 2.0003 lambda 774263,599484 iteration 43
## AICc -2940.75 edf 2.0010 lambda 215443,599484 iteration 44
## AICc -2941.03 edf 2.0037 lambda 599484,599484 iteration 45
## AICc -2942.05 edf 2.0134 lambda 166810,599484 iteration 46
## AICc -2945.65 edf 2.0476 lambda 464158,599484 iteration 47
## AICc -2957.77 edf 2.1653 lambda 129154,599484 iteration 48
## AICc -2993.04 edf 2.5319 lambda 35938.,599484 iteration 49
## AICc -3061.40 edf 3.4022 lambda 10000.,599484 iteration 50
## AICc -2940.64 edf 2.0000 lambda 100000,166810 iteration 51
## AICc -2940.65 edf 2.0001 lambda 278255,166810 iteration 52
## AICc -2940.67 edf 2.0003 lambda 774263,166810 iteration 53
## AICc -2940.75 edf 2.0010 lambda 215443,166810 iteration 54
## AICc -2941.03 edf 2.0037 lambda 599484,166810 iteration 55
## AICc -2942.05 edf 2.0134 lambda 166810,166810 iteration 56
## AICc -2945.65 edf 2.0476 lambda 464158,166810 iteration 57
## AICc -2957.77 edf 2.1653 lambda 129154,166810 iteration 58
## AICc -2993.04 edf 2.5319 lambda 35938.,166810 iteration 59
## AICc -3061.40 edf 3.4022 lambda 10000.,166810 iteration 60
## AICc -2940.64 edf 2.0000 lambda 100000,464158 iteration 61
## AICc -2940.65 edf 2.0001 lambda 278255,464158 iteration 62
## AICc -2940.67 edf 2.0003 lambda 774263,464158 iteration 63
## AICc -2940.75 edf 2.0010 lambda 215443,464158 iteration 64
## AICc -2941.03 edf 2.0037 lambda 599484,464158 iteration 65
## AICc -2942.05 edf 2.0134 lambda 166810,464158 iteration 66
## AICc -2945.65 edf 2.0476 lambda 464158,464158 iteration 67
## AICc -2957.77 edf 2.1653 lambda 129154,464158 iteration 68
## AICc -2993.04 edf 2.5319 lambda 35938.,464158 iteration 69
## AICc -3061.40 edf 3.4022 lambda 10000.,464158 iteration 70
## AICc -2940.64 edf 2.0000 lambda 100000,129154 iteration 71
## AICc -2940.65 edf 2.0001 lambda 278255,129154 iteration 72
## AICc -2940.67 edf 2.0003 lambda 774263,129154 iteration 73
## AICc -2940.75 edf 2.0010 lambda 215443,129154 iteration 74
## AICc -2941.03 edf 2.0037 lambda 599484,129154 iteration 75
## AICc -2942.05 edf 2.0134 lambda 166810,129154 iteration 76
## AICc -2945.65 edf 2.0476 lambda 464158,129154 iteration 77
## AICc -2957.77 edf 2.1653 lambda 129154,129154 iteration 78
## AICc -2993.04 edf 2.5319 lambda 35938.,129154 iteration 79
## AICc -3061.40 edf 3.4022 lambda 10000.,129154 iteration 80
## AICc -2940.64 edf 2.0000 lambda 100000,35938. iteration 81
## AICc -2940.65 edf 2.0001 lambda 278255,35938. iteration 82
## AICc -2940.67 edf 2.0003 lambda 774263,35938. iteration 83
## AICc -2940.75 edf 2.0010 lambda 215443,35938. iteration 84
## AICc -2941.03 edf 2.0037 lambda 599484,35938. iteration 85
## AICc -2942.05 edf 2.0134 lambda 166810,35938. iteration 86
## AICc -2945.65 edf 2.0476 lambda 464158,35938. iteration 87
## AICc -2957.77 edf 2.1653 lambda 129154,35938. iteration 88
## AICc -2993.04 edf 2.5319 lambda 35938.,35938. iteration 89
## AICc -3061.40 edf 3.4022 lambda 10000.,35938. iteration 90
## AICc -2940.64 edf 2.0000 lambda 100000,10000. iteration 91
## AICc -2940.65 edf 2.0001 lambda 278255,10000. iteration 92
## AICc -2940.67 edf 2.0003 lambda 774263,10000. iteration 93
## AICc -2940.75 edf 2.0010 lambda 215443,10000. iteration 94
## AICc -2941.03 edf 2.0037 lambda 599484,10000. iteration 95
## AICc -2942.05 edf 2.0134 lambda 166810,10000. iteration 96
## AICc -2945.65 edf 2.0476 lambda 464158,10000. iteration 97
## AICc -2957.77 edf 2.1653 lambda 129154,10000. iteration 98
## AICc -2993.04 edf 2.5319 lambda 35938.,10000. iteration 99
## AICc -3061.40 edf 3.4022 lambda 10000.,10000. iteration 100
## elapsed time: 14.35sec
```

```
coefi1 <- lasso.coef(b1)
coefi1
```

```
##   mu.s.la(f11).mrate    mu.s.la(f11).lemp   mu.s.la(f11).lemp2
##         3.983148e-02        -1.657104e-02        -8.878360e-04
##     mu.s.la(f11).age    mu.s.la(f11).age2   mu.s.la(f11).tau21
##         5.042586e-03         9.115312e-05         1.000000e-04
##   mu.s.la(f21).sole1   mu.s.la(f21).tau21     mu.p.(Intercept)
##         7.207647e-03         1.000000e-04         1.206183e+00
## sigma2.p.(Intercept)
##        -1.750043e+00
```

```
#Using beta family
lasso.select(subx, coefi1)
```

```
## $lambda.common
## mu.s.la(f1).lambda
##              10000
##
## $lambda.factor
## mu.s.la(f2).lambda
##              10000
##
## $lasso.index
## [1] 3 5
##
## $modified.coefficients
##   mu.s.la(f11).mrate    mu.s.la(f11).lemp   mu.s.la(f11).lemp2
##          0.039831475         -0.016571042          0.000000000
##     mu.s.la(f11).age    mu.s.la(f11).age2   mu.s.la(f21).sole1
##          0.005042586          0.000000000          0.007207647
##     mu.p.(Intercept) sigma2.p.(Intercept)
##          1.206183194         -1.750043148
```