ch03and10

```
###chapter 03 and 10, model selection###############
#### Example: Indian
fn.data <- "http://statacumen.com/teach/ADA2/ADA2_notes_Ch02_indian.dat"
indian <- read.table(fn.data, header=TRUE)

# Description of variables
#    id = individual id
#   age = age in years           yrmig = years since migration
#    wt = weight in kilos           ht = height in mm
#  chin = chin skin fold in mm    fore = forearm skin fold in mm
#  calf = calf skin fold in mm   pulse = pulse rate-beats/min
# sysbp = systolic bp            diabp = diastolic bp

# Create the "fraction of their life" variable
#   yrage = years since migration divided by age
indian$yrage <- indian$yrmig / indian$age



# correlation matrix and associated p-values testing "H0: rho == 0"
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

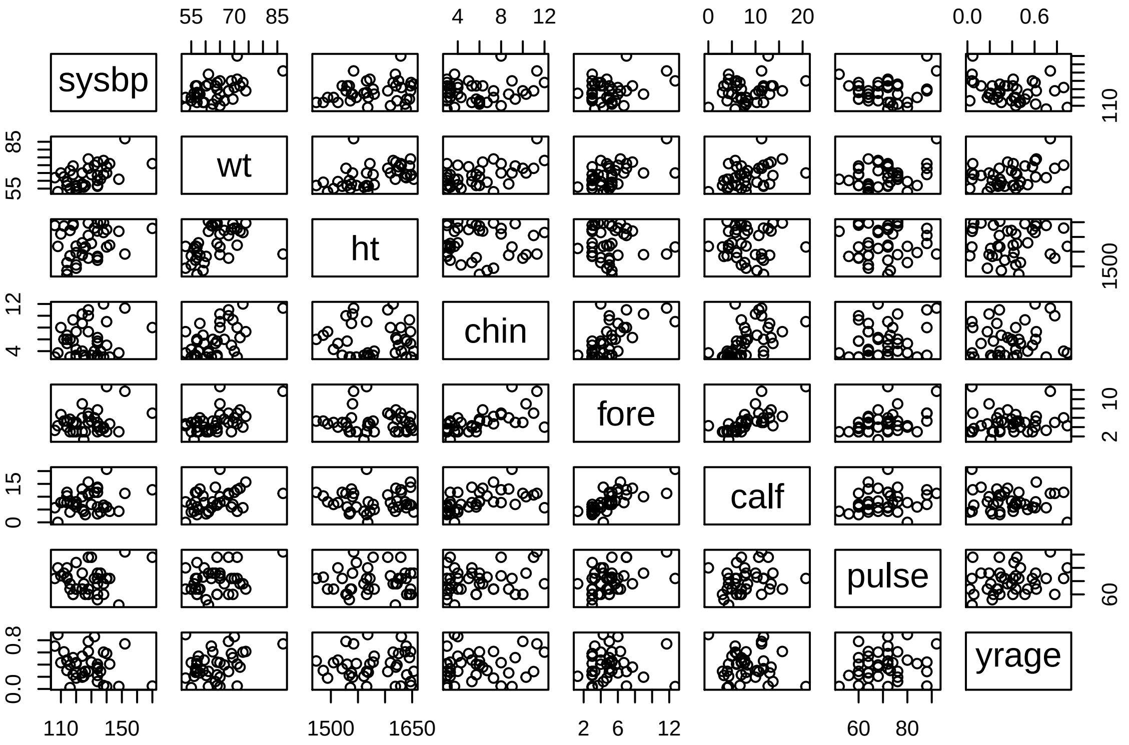
```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

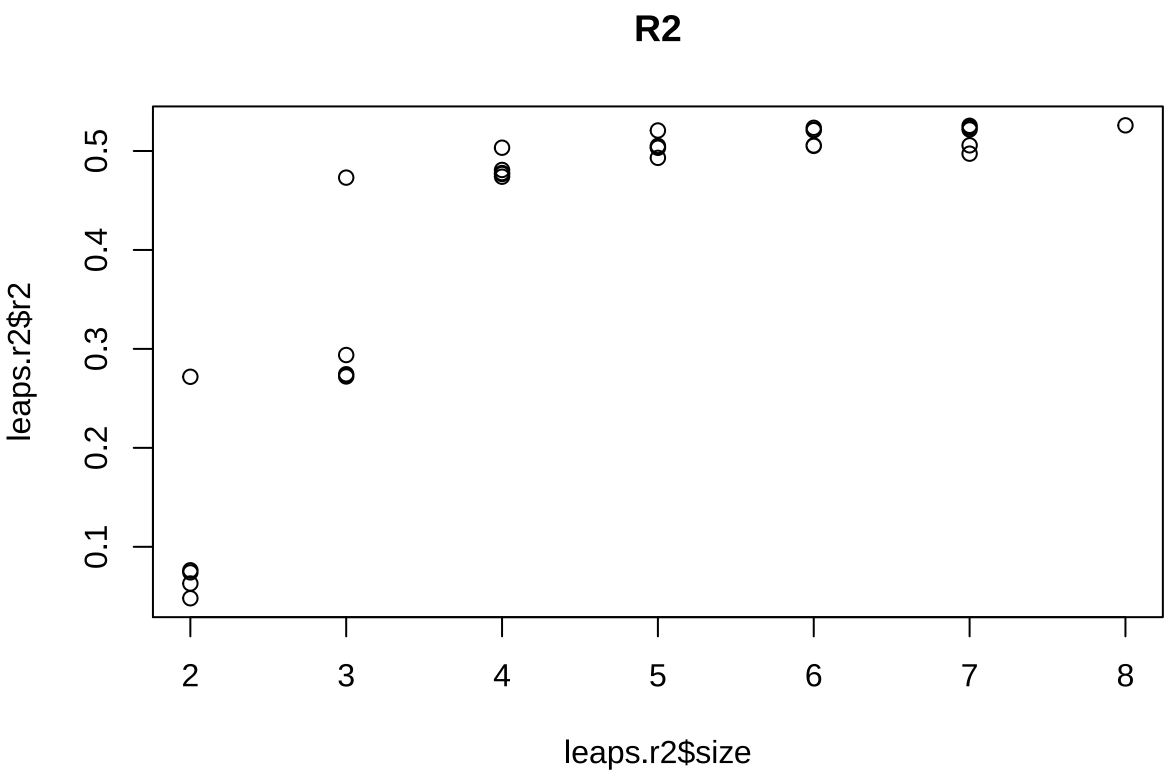
```
i.cor <- rcorr(as.matrix(indian[,c("sysbp", "wt", "ht", "chin"
                                 , "fore", "calf", "pulse", "yrage")]))
# print correlations with the response to 3 significant digits
signif(i.cor$r[1, ], 3)
```

```
##  sysbp     wt     ht   chin   fore   calf  pulse  yrage
##  1.000  0.521  0.219  0.170  0.272  0.251  0.133 -0.276
```

```
x<-signif(i.cor$r[1, ], 3)

# scatterplots
pairs(indian[,c("sysbp", "wt", "ht", "chin", "fore", "calf", "pulse", "yrage")])
```



```
# The leaps package provides best subsets with other selection criteria.
library(leaps)

# First, fit the full model
lm.indian.full <- lm(sysbp ~ wt + ht + chin + fore + calf + pulse + yrage, data = indian)

# Second, create the design matrix which leap uses as argument
#   using model.matrix(lm.XXX) as input to leaps()


# R^2 -- for each model size, report best subset of size 5
X.indian <- indian[,c(4:9,12)]
leaps.r2 <- leaps(x = X.indian, y = indian$sysbp
                , method = 'r2'
                ,nbest = 5, names =c("wt","ht", "chin", "fore", "calf", "pulse","yrage"))
# plot model R^2 vs size of model
plot(leaps.r2$size, leaps.r2$r2, main = "R2")
```



```
#dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture #notes/plots/chap10plot1.jpg")
#dev.off()
##best five models
leaps.r2$which[order(-leaps.r2$r2)[1:5],]
```

```
##     wt   ht chin  fore  calf pulse yrage
## 7 TRUE TRUE TRUE  TRUE  TRUE  TRUE  TRUE
## 6 TRUE TRUE TRUE  TRUE FALSE  TRUE  TRUE
## 6 TRUE TRUE TRUE  TRUE  TRUE FALSE  TRUE
## 5 TRUE TRUE TRUE  TRUE FALSE FALSE  TRUE
## 6 TRUE TRUE TRUE FALSE  TRUE  TRUE  TRUE
```

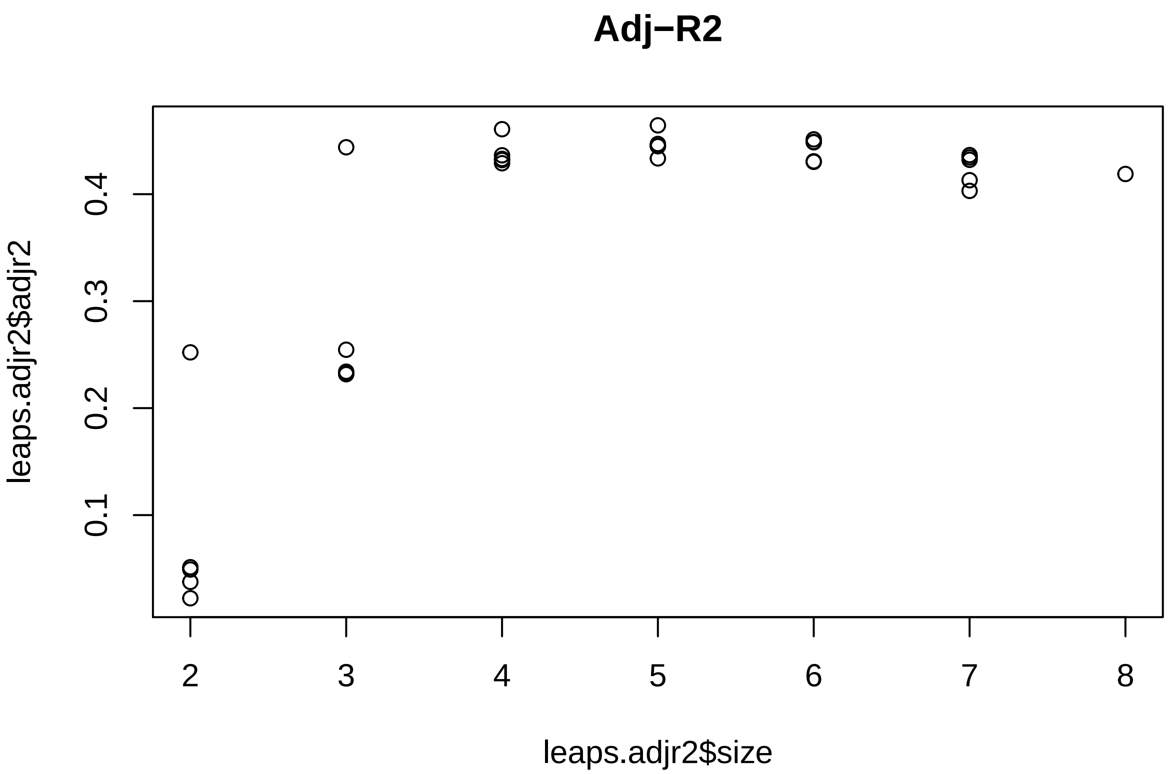
```
leaps.r2$r2[order(-leaps.r2$r2)[1:5]]
```

```
## [1] 0.5259164 0.5254780 0.5236789 0.5234360 0.5217823
```

```
# report the best model (indicate which terms are in the model)
best.model.r2 <- leaps.r2$which[which((leaps.r2$r2 == max(leaps.r2$r2))),]
# these are the variable names for the best model
names(best.model.r2)[best.model.r2]
```

```
## [1] "wt"    "ht"    "chin"  "fore"  "calf"  "pulse" "yrage"
```

```
# adj-R^2 -- for each model size, report best subset of size 5
leaps.adjr2 <- leaps(x = X.indian, y = indian$sysbp
                , method = 'adjr2'
                , nbest = 5, names =c("wt","ht", "chin", "fore", "calf", "pulse","yrage"))
# plot model R^2 vs size of model
plot(leaps.adjr2$size, leaps.adjr2$adjr2, main = "Adj-R2")
```



```
#dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture #notes/plots/chap10plot2.jpg")
#dev.off()

#report the best 5 models according to adjusted R^2
leaps.adjr2$which[order(-leaps.adjr2$adjr2)[1:5],]
```

```
##     wt    ht chin  fore  calf pulse yrage
## 4 TRUE  TRUE TRUE FALSE FALSE FALSE  TRUE
## 3 TRUE FALSE TRUE FALSE FALSE FALSE  TRUE
## 5 TRUE  TRUE TRUE  TRUE FALSE FALSE  TRUE
## 5 TRUE  TRUE TRUE FALSE FALSE  TRUE  TRUE
## 5 TRUE  TRUE TRUE FALSE  TRUE FALSE  TRUE
```

```
leaps.adjr2$adjr2[order(-leaps.adjr2$adjr2)[1:5]]
```

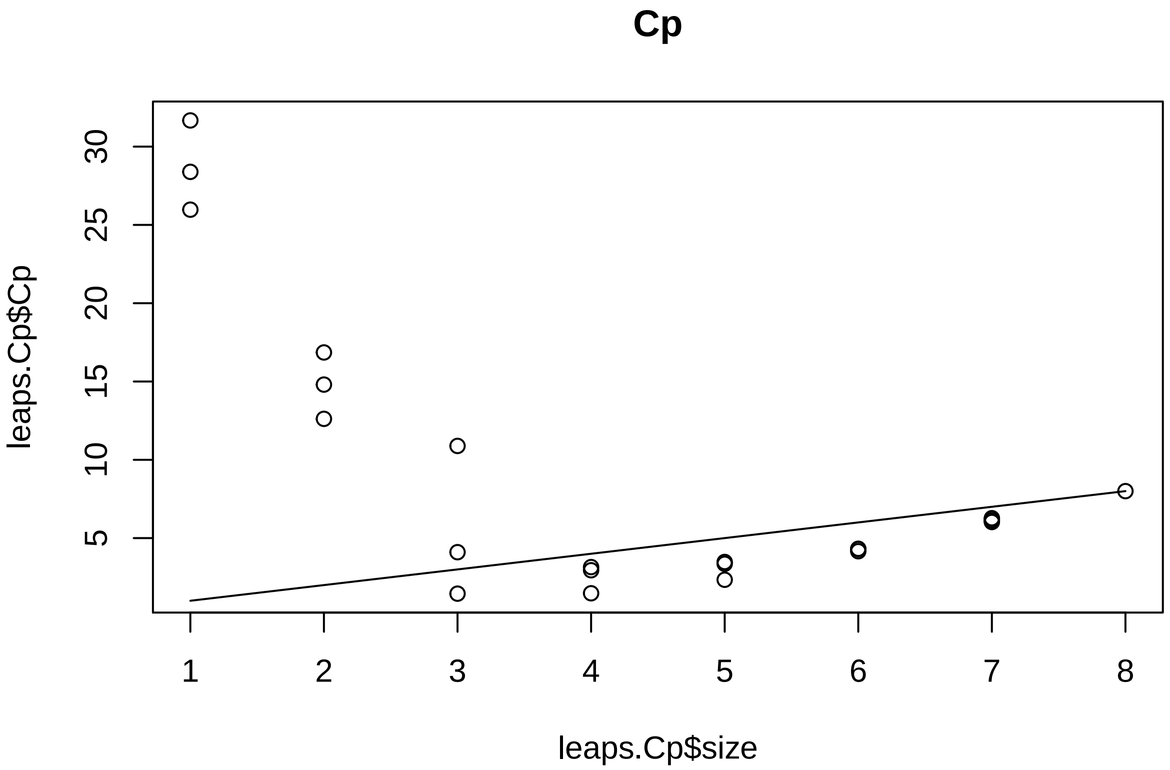
```
## [1] 0.4643276 0.4607546 0.4512293 0.4488217 0.4484703
```

```
# report the best model (indicate which terms are in the model)
best.model.adjr2 <- leaps.adjr2$which[which((leaps.adjr2$adjr2 == max(leaps.adjr2$adjr2))),]

# these are the variable names for the best model
names(best.model.adjr2)[best.model.adjr2]
```

```
## [1] "wt"    "ht"    "chin"  "yrage"
```

```
# Cp -- for each model size, report best subset of size 3
leaps.Cp <- leaps(x = model.matrix(lm.indian.full), y = indian$sysbp
                , method = 'Cp'
                , int = FALSE, nbest = 3, names = colnames(model.matrix(lm.indian.full)))

# plot model R^2 vs size of model
plot(leaps.Cp$size, leaps.Cp$Cp, main = "Cp")
```

```
  lines(leaps.Cp$size, leaps.Cp$size)        # adds the line for Cp = p
```



```
 # dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture #notes/plots/chap10plot3.jpg")
 # dev.off()

## Get the 5 "best" models according to Cp
leaps.Cp$which[order(leaps.Cp$Cp)[1:5],]
```

```
##   (Intercept)   wt    ht  chin  fore  calf pulse yrage
## 3        TRUE TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
## 4        TRUE TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE
## 5        TRUE TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE
## 4        TRUE TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE
## 4        TRUE TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE
```

```
leaps.Cp$Cp[order(leaps.Cp$Cp)[1:5]]
```

```
## [1] 1.453122 1.477132 2.340175 2.947060 3.150596
```

```
# best subset, returns results sorted by BIC
f.bestsubset <- function(form, dat, nbest = 5){
  library(leaps)
  bs <- regsubsets(form, data=dat, nvmax=30, nbest=nbest, method="exhaustive");
  bs2 <- cbind(summary(bs)$which, (rowSums(summary(bs)$which)-1)
               , summary(bs)$rss, summary(bs)$rsq
               , summary(bs)$adjr2, summary(bs)$cp, summary(bs)$bic);
  cn <- colnames(bs2);
    cn[(dim(bs2)[2]-5):dim(bs2)[2]] <- c("SIZE", "rss", "r2", "adjr2", "cp", "bic");
    colnames(bs2) <- cn;
  ind <- sort.int(summary(bs)$bic, index.return=TRUE); bs2 <- bs2[ind$ix,];
  return(bs2);
}
# perform on our model
i.best <- f.bestsubset(formula(sysbp ~ wt + ht + chin + fore + calf + pulse + yrage)
          , indian)
  op <- options(); # saving old options
  options(width=90) # setting command window output text width wider
i.best
```

```
##   (Intercept) wt ht chin fore calf pulse yrage SIZE      rss         r2      adjr2
## 2           1  1  0    0    0    0     0     1    2 3441.363 0.47310778 0.44383599
## 3           1  1  0    1    0    0     0     1    3 3243.990 0.50332663 0.46075463
## 3           1  1  0    0    1    0     0     1    3 3390.815 0.48084699 0.43634816
## 3           1  1  0    0    0    1     0     1    3 3411.145 0.47773431 0.43296868
## 3           1  1  1    0    0    0     0     1    3 3417.624 0.47674226 0.43189159
## 3           1  1  0    0    0    0     1     1    3 3435.481 0.47400828 0.42892328
## 4           1  1  1    1    0    0     0     1    4 3130.425 0.52071413 0.46432755
## 4           1  1  0    1    0    0     1     1    4 3232.168 0.50513668 0.44691747
## 4           1  1  0    1    1    0     0     1    4 3243.771 0.50336023 0.44493203
## 4           1  1  0    1    0    1     0     1    4 3243.988 0.50332702 0.44489490
## 4           1  1  1    0    1    0     0     1    4 3310.944 0.49307566 0.43343750
## 5           1  1  1    1    1    0     0     1    5 3112.647 0.52343597 0.45122930
## 5           1  1  1    1    0    0     1     1    5 3126.303 0.52134520 0.44882174
## 5           1  1  1    1    0    1     0     1    5 3128.297 0.52103997 0.44847027
## 5           1  1  0    1    1    0     1     1    5 3228.867 0.50564205 0.43073933
## 5           1  1  0    1    0    1     1     1    5 3231.936 0.50517225 0.43019835
## 1           1  1  0    0    0    0     0     0    1 4756.056 0.27182072 0.25214020
## 6           1  1  1    1    1    0     1     1    6 3099.310 0.52547798 0.43650510
## 6           1  1  1    1    1    1     0     1    6 3111.060 0.52367894 0.43436875
## 6           1  1  1    1    0    1     1     1    6 3123.448 0.52178233 0.43211651
## 2           1  1  0    1    0    0     0     0    2 4612.426 0.29381129 0.25457859
## 6           1  1  0    1    1    1     1     1    6 3228.324 0.50572524 0.41304872
## 2           1  1  0    0    0    1     0     0    2 4739.383 0.27437355 0.23406097
## 2           1  1  0    0    0    0     1     0    2 4749.950 0.27275566 0.23235320
## 2           1  1  1    0    0    0     0     0    2 4754.044 0.27212880 0.23169151
## 6           1  1  1    0    1    1     1     1    6 3283.293 0.49730910 0.40305455
## 7           1  1  1    1    1    1     1     1    7 3096.446 0.52591643 0.41886531
## 1           1  0  0    0    0    0     0     1    1 6033.372 0.07625642 0.05129038
## 1           1  0  0    0    1    0     0     0    1 6047.218 0.07413652 0.04911319
## 1           1  0  0    0    0    1     0     0    1 6120.639 0.06289527 0.03756811
## 1           1  0  1    0    0    0     0     0    1 6217.854 0.04801119 0.02228176
##          cp         bic
## 2  1.453122 -13.9989263
## 3  1.477132 -12.6388375
## 3  2.947060 -10.9124614
## 3  3.150596 -10.6793279
## 3  3.215466 -10.6053171
## 3  3.394238 -10.4020763
## 4  2.340175 -10.3650555
## 4  3.358774  -9.1176654
## 4  3.474934  -8.9779148
## 4  3.477106  -8.9753065
## 4  4.147436  -8.1785389
## 5  4.162196  -6.9236046
## 5  4.298910  -6.7528787
## 5  4.318869  -6.7280169
## 5  5.325728  -5.4939516
## 5  5.356448  -5.4569068
## 1 12.615145  -5.0439888
## 6  6.028670  -3.4275113
## 6  6.146308  -3.2799319
## 6  6.270326  -3.1249499
## 2 13.177196  -2.5763538
## 6  7.320289  -1.8369533
## 2 14.448217  -1.5173924
## 2 14.554009  -1.4305331
## 2 14.595000  -1.3969306
## 6  7.870614  -1.1784808
## 7  8.000000   0.1999979
## 1 25.402961   4.2336138
## 1 25.541579   4.3230122
## 1 26.276637   4.7936744
## 1 27.249897   5.4082455
```

```
options(op); # reset (all) initial options

## step() function specification
## The first two arguments of step(object, scope, ...) are
# object = a fitted model object.
# scope  = a formula giving the terms to be considered for adding or dropping
## default is AIC
# for BIC, include k = log(nrow( [data.frame name] ))
# test="F" includes additional information
#           for parameter estimate tests that we're familiar with


#################stepwise regression###############

#forward selection
# start with an empty model (just the intercept 1)
lm.indian.empty <- lm(sysbp ~ 1, data = indian)
# Forward selection, BIC with F-tests
lm.indian.forward.red.AIC <- step(lm.indian.empty
                                 , sysbp ~ wt + ht + chin + fore + calf + pulse + yrage
                                 , direction = "forward", test = "F")
```

```
## Start:  AIC=201.71
## sysbp ~ 1
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## + wt     1   1775.38 4756.1 191.34 13.8117 0.0006654 ***
## + yrage  1    498.06 6033.4 200.62  3.0544 0.0888139 .
## + fore   1    484.22 6047.2 200.71  2.9627 0.0935587 .
## + calf   1    410.80 6120.6 201.18  2.4833 0.1235725
## <none>               6531.4 201.71
## + ht     1    313.58 6217.9 201.79  1.8660 0.1801796
## + chin   1    189.19 6342.2 202.57  1.1037 0.3002710
## + pulse  1    114.77 6416.7 203.02  0.6618 0.4211339
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Step:  AIC=191.34
## sysbp ~ wt
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## + yrage  1   1314.69 3441.4 180.72 13.7530 0.0006991 ***
## <none>               4756.1 191.34
## + chin   1    143.63 4612.4 192.15  1.1210 0.2967490
## + calf   1     16.67 4739.4 193.20  0.1267 0.7240063
## + pulse  1      6.11 4749.9 193.29  0.0463 0.8308792
## + ht     1      2.01 4754.0 193.32  0.0152 0.9024460
## + fore   1      1.16 4754.9 193.33  0.0088 0.9257371
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=180.72
## sysbp ~ wt + yrage
##
##         Df Sum of Sq    RSS    AIC F value Pr(>F)
## + chin   1   197.372 3244.0 180.42  2.1295 0.1534
## <none>               3441.4 180.72
## + fore   1    50.548 3390.8 182.15  0.5218 0.4749
## + calf   1    30.218 3411.1 182.38  0.3101 0.5812
## + ht     1    23.738 3417.6 182.45  0.2431 0.6251
## + pulse  1     5.882 3435.5 182.66  0.0599 0.8081
##
## Step:  AIC=180.42
## sysbp ~ wt + yrage + chin
##
##         Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>               3244.0 180.42
## + ht     1   113.565 3130.4 181.03  1.2334 0.2745
## + pulse  1    11.822 3232.2 182.28  0.1244 0.7265
## + fore   1     0.219 3243.8 182.42  0.0023 0.9620
## + calf   1     0.003 3244.0 182.42  0.0000 0.9959
```

```
summary(lm.indian.forward.red.AIC)
```

```
##
## Call:
## lm(formula = sysbp ~ wt + yrage + chin, data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.6382  -6.6316   0.4521   6.3593  24.2086
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  52.9092    15.0895   3.506 0.001266 **
## wt            1.4407     0.2766   5.209 8.51e-06 ***
## yrage       -27.3522     7.1185  -3.842 0.000491 ***
## chin         -1.0135     0.6945  -1.459 0.153407
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 9.627 on 35 degrees of freedom
## Multiple R-squared:  0.5033, Adjusted R-squared:  0.4608
## F-statistic: 11.82 on 3 and 35 DF,  p-value: 1.684e-05
```

```
#backward selection
# start with a full model
lm.indian.full <- lm(sysbp ~ wt + ht + chin + fore + calf + pulse + yrage, data = indian)
summary(lm.indian.full)
```

```
##
## Call:
## lm(formula = sysbp ~ wt + ht + chin + fore + calf + pulse + yrage,
##     data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.3993  -5.7916  -0.6907   6.9453  23.5771
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 106.45766   53.91303   1.975 0.057277 .
## wt            1.71095    0.38659   4.426 0.000111 ***
## ht           -0.04533    0.03945  -1.149 0.259329
## chin         -1.15725    0.84612  -1.368 0.181239
## fore         -0.70183    1.34986  -0.520 0.606806
## calf          0.10357    0.61170   0.169 0.866643
## pulse         0.07485    0.19570   0.383 0.704699
## yrage       -29.31810    7.86839  -3.726 0.000777 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.994 on 31 degrees of freedom
## Multiple R-squared:  0.5259, Adjusted R-squared:  0.4189
## F-statistic: 4.913 on 7 and 31 DF,  p-value: 0.0008079
```

```
# model reduction using update() and subtracting (removing) model terms
lm.indian2.red <- lm.indian.full;
# remove calf
lm.indian2.red <- update(lm.indian2.red, ~ . - calf );
summary(lm.indian2.red)
```

```
##
## Call:
## lm(formula = sysbp ~ wt + ht + chin + fore + pulse + yrage, data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6993  -5.3152  -0.7725   7.2966  23.7240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 106.13739   53.05581   2.000 0.053993 .
## wt            1.70900    0.38051   4.491 8.65e-05 ***
## ht           -0.04478    0.03871  -1.157 0.256008
## chin         -1.14165    0.82823  -1.378 0.177635
## fore         -0.56731    1.07462  -0.528 0.601197
## pulse         0.07103    0.19142   0.371 0.713018
## yrage       -29.54000    7.63983  -3.867 0.000509 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.841 on 32 degrees of freedom
## Multiple R-squared:  0.5255, Adjusted R-squared:  0.4365
## F-statistic: 5.906 on 6 and 32 DF,  p-value: 0.0003103
```

```r
# remove pulse
lm.indian2.red <- update(lm.indian2.red, ~ . - pulse);
summary(lm.indian2.red);
```

```
##
## Call:
## lm(formula = sysbp ~ wt + ht + chin + fore + yrage, data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6147  -5.9803  -0.2065   6.6755  24.9269
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 110.27872   51.18665   2.154 0.038601 *
## wt            1.71825    0.37470   4.586 6.22e-05 ***
## ht           -0.04504    0.03820  -1.179 0.246810
## chin         -1.17716    0.81187  -1.450 0.156514
## fore         -0.43385    0.99933  -0.434 0.667013
## yrage       -28.98171    7.39172  -3.921 0.000421 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.712 on 33 degrees of freedom
## Multiple R-squared:  0.5234, Adjusted R-squared:  0.4512
## F-statistic: 7.249 on 5 and 33 DF,  p-value: 0.0001124
```

```r
# remove fore
lm.indian2.red <- update(lm.indian2.red, ~ . - fore ); summary(lm.indian2.red);
```

```
##
## Call:
## lm(formula = sysbp ~ wt + ht + chin + yrage, data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1030  -6.3484   0.2834   6.7766  24.8883
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 104.52292   48.84627   2.140 0.039629 *
## wt            1.64631    0.33203   4.958 1.94e-05 ***
## ht           -0.03957    0.03563  -1.111 0.274530
## chin         -1.31083    0.74220  -1.766 0.086348 .
## yrage       -28.32580    7.14879  -3.962 0.000361 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.595 on 34 degrees of freedom
## Multiple R-squared:  0.5207, Adjusted R-squared:  0.4643
## F-statistic: 9.235 on 4 and 34 DF,  p-value: 3.661e-05
```

```
# remove ht
lm.indian2.red <- update(lm.indian2.red, ~ . - ht); summary(lm.indian2.red);
```

```
##
## Call:
## lm(formula = sysbp ~ wt + chin + yrage, data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.6382  -6.6316   0.4521   6.3593  24.2086
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  52.9092    15.0895   3.506 0.001266 **
## wt            1.4407     0.2766   5.209 8.51e-06 ***
## chin         -1.0135     0.6945  -1.459 0.153407
## yrage       -27.3522     7.1185  -3.842 0.000491 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.627 on 35 degrees of freedom
## Multiple R-squared:  0.5033, Adjusted R-squared:  0.4608
## F-statistic: 11.82 on 3 and 35 DF,  p-value: 1.684e-05
```

```
# remove chin
lm.indian2.red <- update(lm.indian2.red, ~ . - chin ); summary(lm.indian2.red);
```

```
##
## Call:
## lm(formula = sysbp ~ wt + yrage, data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.4330  -7.3070   0.8963   5.7275  23.9819
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  60.8959    14.2809   4.264 0.000138 ***
## wt            1.2169     0.2337   5.207 7.97e-06 ***
## yrage       -26.7672     7.2178  -3.708 0.000699 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.777 on 36 degrees of freedom
## Multiple R-squared:  0.4731, Adjusted R-squared:  0.4438
## F-statistic: 16.16 on 2 and 36 DF,  p-value: 9.795e-06
```

```
lm.indian2.final <- lm.indian2.red

# Backward selection, BIC with F-tests
```

```r
#BIC won't show
lm.indian.backward.red.BIC <- step(lm.indian.full, direction = "backward", test = "F",
                                   k = log(nrow(indian)))
```

```
## Start:  AIC=199.91
## sysbp ~ wt + ht + chin + fore + calf + pulse + yrage
## 
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)    
## - calf   1      2.86 3099.3 196.28  0.0287 0.8666427    
## - pulse  1     14.61 3111.1 196.43  0.1463 0.7046990    
## - fore   1     27.00 3123.4 196.59  0.2703 0.6068061    
## - ht     1    131.88 3228.3 197.88  1.3203 0.2593289    
## - chin   1    186.85 3283.3 198.53  1.8706 0.1812390    
## <none>               3096.4 199.91                      
## - yrage  1   1386.76 4483.2 210.68 13.8835 0.0007773 ***
## - wt     1   1956.49 5052.9 215.35 19.5874 0.0001105 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Step:  AIC=196.28
## sysbp ~ wt + ht + chin + fore + pulse + yrage
## 
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)    
## - pulse  1     13.34 3112.6 192.79  0.1377 0.7130185    
## - fore   1     26.99 3126.3 192.96  0.2787 0.6011969    
## - ht     1    129.56 3228.9 194.22  1.3377 0.2560083    
## - chin   1    184.03 3283.3 194.87  1.9000 0.1776352    
## <none>               3099.3 196.28                      
## - yrage  1   1448.00 4547.3 207.57 14.9504 0.0005087 ***
## - wt     1   1953.77 5053.1 211.69 20.1724 8.655e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Step:  AIC=192.79
## sysbp ~ wt + ht + chin + fore + yrage
## 
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)    
## - fore   1     17.78 3130.4 189.35  0.1885  0.667013    
## - ht     1    131.12 3243.8 190.73  1.3902  0.246810    
## - chin   1    198.30 3310.9 191.53  2.1023  0.156514    
## <none>               3112.6 192.79                      
## - yrage  1   1450.02 4562.7 204.04 15.3730  0.000421 ***
## - wt     1   1983.51 5096.2 208.35 21.0290 6.219e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Step:  AIC=189.35
## sysbp ~ wt + ht + chin + yrage
## 
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)    
## - ht     1    113.57 3244.0 187.07  1.2334 0.2745301    
## - chin   1    287.20 3417.6 189.11  3.1193 0.0863479 .  
## <none>               3130.4 189.35                      
## - yrage  1   1445.52 4575.9 200.49 15.7000 0.0003607 ***
## - wt     1   2263.64 5394.1 206.90 24.5857 1.945e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=187.07
## sysbp ~ wt + chin + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - chin   1    197.37 3441.4 185.71  2.1295 0.1534065
## <none>               3244.0 187.07
## - yrage  1   1368.44 4612.4 197.14 14.7643 0.0004912 ***
## - wt     1   2515.33 5759.3 205.80 27.1384 8.512e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=185.71
## sysbp ~ wt + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>               3441.4 185.71
## - yrage  1    1314.7 4756.1 194.67  13.753 0.0006991 ***
## - wt     1    2592.0 6033.4 203.95  27.115 7.966e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm.indian.backward.red.BIC)
```

```
##
## Call:
## lm(formula = sysbp ~ wt + yrage, data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.4330  -7.3070   0.8963   5.7275  23.9819
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  60.8959    14.2809   4.264 0.000138 ***
## wt            1.2169     0.2337   5.207 7.97e-06 ***
## yrage       -26.7672     7.2178  -3.708 0.000699 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.777 on 36 degrees of freedom
## Multiple R-squared:  0.4731, Adjusted R-squared:  0.4438
## F-statistic: 16.16 on 2 and 36 DF,  p-value: 9.795e-06
```

```
lm.indian.red.AIC <- step(lm.indian.full, direction="backward", test="F")  #default AIC
```

```
## Start:  AIC=186.6
## sysbp ~ wt + ht + chin + fore + calf + pulse + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - calf   1      2.86 3099.3 184.64  0.0287 0.8666427
## - pulse  1     14.61 3111.1 184.79  0.1463 0.7046990
## - fore   1     27.00 3123.4 184.94  0.2703 0.6068061
## - ht     1    131.88 3228.3 186.23  1.3203 0.2593289
## <none>              3096.4 186.60
## - chin   1    186.85 3283.3 186.89  1.8706 0.1812390
## - yrage  1   1386.76 4483.2 199.04 13.8835 0.0007773 ***
## - wt     1   1956.49 5052.9 203.70 19.5874 0.0001105 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=184.64
## sysbp ~ wt + ht + chin + fore + pulse + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - pulse  1     13.34 3112.6 182.81  0.1377 0.7130185
## - fore   1     26.99 3126.3 182.98  0.2787 0.6011969
## - ht     1    129.56 3228.9 184.24  1.3377 0.2560083
## <none>              3099.3 184.64
## - chin   1    184.03 3283.3 184.89  1.9000 0.1776352
## - yrage  1   1448.00 4547.3 197.59 14.9504 0.0005087 ***
## - wt     1   1953.77 5053.1 201.70 20.1724 8.655e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=182.81
## sysbp ~ wt + ht + chin + fore + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - fore   1     17.78 3130.4 181.03  0.1885  0.667013
## - ht     1    131.12 3243.8 182.42  1.3902  0.246810
## <none>              3112.6 182.81
## - chin   1    198.30 3310.9 183.22  2.1023  0.156514
## - yrage  1   1450.02 4562.7 195.72 15.3730  0.000421 ***
## - wt     1   1983.51 5096.2 200.03 21.0290 6.219e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=181.03
## sysbp ~ wt + ht + chin + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - ht     1    113.57 3244.0 180.42  1.2334 0.2745301
## <none>              3130.4 181.03
## - chin   1    287.20 3417.6 182.45  3.1193 0.0863479 .
## - yrage  1   1445.52 4575.9 193.84 15.7000 0.0003607 ***
## - wt     1   2263.64 5394.1 200.25 24.5857 1.945e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=180.42
## sysbp ~ wt + chin + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>              3244.0 180.42
## - chin   1    197.37 3441.4 180.72  2.1295 0.1534065
## - yrage  1   1368.44 4612.4 192.15 14.7643 0.0004912 ***
## - wt     1   2515.33 5759.3 200.81 27.1384 8.512e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm.indian.red.AIC)
```

```
## 
## Call:
## lm(formula = sysbp ~ wt + chin + yrage, data = indian)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -16.6382  -6.6316   0.4521   6.3593  24.2086 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  52.9092    15.0895   3.506 0.001266 ** 
## wt            1.4407     0.2766   5.209 8.51e-06 ***
## chin         -1.0135     0.6945  -1.459 0.153407    
## yrage       -27.3522     7.1185  -3.842 0.000491 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 9.627 on 35 degrees of freedom
## Multiple R-squared:  0.5033, Adjusted R-squared:  0.4608 
## F-statistic: 11.82 on 3 and 35 DF,  p-value: 1.684e-05
```

```
# Stepwise (both) selection, BIC with F-tests, starting with intermediate model
# (this is a purposefully chosen "opposite" model,
#    from the forward and backward methods this model
#    includes all the variables dropped and none kept)
lm.indian.intermediate <- lm(sysbp ~ ht + fore + calf + pulse, data = indian)
# option: trace = 0 does not print each step of the selection
lm.indian.both.red.BIC <- step(lm.indian.intermediate
                               , sysbp ~ wt + ht + chin + fore + calf + pulse + yrage
                               , direction = "both", test = "F", k = log(nrow(indian)), trace = 0)
# the anova object provides a summary of the selection steps in order
lm.indian.both.red.BIC$anova
```

```
##      Step Df    Deviance Resid. Df Resid. Dev      AIC
## 1         NA          NA        34   5651.131 212.3837
## 2 - pulse  1    2.874432        35   5654.005 208.7400
## 3  - calf  1   21.843631        36   5675.849 205.2268
## 4    + wt -1  925.198114        35   4750.651 201.9508
## 5 + yrage -1 1439.707117        34   3310.944 191.5335
## 6    - ht  1   79.870793        35   3390.815 188.7995
## 7  - fore  1   50.548149        36   3441.363 185.7131
```

```
summary(lm.indian.both.red.BIC)
```

```
## 
## Call:
## lm(formula = sysbp ~ wt + yrage, data = indian)
## 
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -18.4330  -7.3070   0.8963   5.7275  23.9819
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  60.8959    14.2809   4.264 0.000138 ***
## wt            1.2169     0.2337   5.207 7.97e-06 ***
## yrage       -26.7672     7.2178  -3.708 0.000699 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.777 on 36 degrees of freedom
## Multiple R-squared:  0.4731, Adjusted R-squared:  0.4438
## F-statistic: 16.16 on 2 and 36 DF,  p-value: 9.795e-06
```

```
##stepwise selection starting from an empty model
lm.indian.step.red.AIC <- step(lm.indian.empty
                                  , sysbp ~ wt + ht + chin + fore + calf + pulse + yrage
                                  , direction = "both", test = "F")
```

```
## Start:  AIC=201.71
## sysbp ~ 1
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## + wt     1   1775.38 4756.1 191.34 13.8117 0.0006654 ***
## + yrage  1    498.06 6033.4 200.62  3.0544 0.0888139 .
## + fore   1    484.22 6047.2 200.71  2.9627 0.0935587 .
## + calf   1    410.80 6120.6 201.18  2.4833 0.1235725
## <none>               6531.4 201.71
## + ht     1    313.58 6217.9 201.79  1.8660 0.1801796
## + chin   1    189.19 6342.2 202.57  1.1037 0.3002710
## + pulse  1    114.77 6416.7 203.02  0.6618 0.4211339
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=191.34
## sysbp ~ wt
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## + yrage  1   1314.69 3441.4 180.72 13.7530 0.0006991 ***
## <none>               4756.1 191.34
## + chin   1    143.63 4612.4 192.15  1.1210 0.2967490
## + calf   1     16.67 4739.4 193.20  0.1267 0.7240063
## + pulse  1      6.11 4749.9 193.29  0.0463 0.8308792
## + ht     1      2.01 4754.0 193.32  0.0152 0.9024460
## + fore   1      1.16 4754.9 193.33  0.0088 0.9257371
## - wt     1   1775.38 6531.4 201.71 13.8117 0.0006654 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=180.72
## sysbp ~ wt + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## + chin   1    197.37 3244.0 180.42  2.1295 0.1534065
## <none>               3441.4 180.72
## + fore   1     50.55 3390.8 182.15  0.5218 0.4748916
## + calf   1     30.22 3411.1 182.38  0.3101 0.5811927
## + ht     1     23.74 3417.6 182.45  0.2431 0.6250515
## + pulse  1      5.88 3435.5 182.66  0.0599 0.8080509
## - yrage  1   1314.69 4756.1 191.34 13.7530 0.0006991 ***
## - wt     1   2592.01 6033.4 200.62 27.1149 7.966e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=180.42
## sysbp ~ wt + yrage + chin
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>               3244.0 180.42
## - chin   1    197.37 3441.4 180.72  2.1295 0.1534065
## + ht     1    113.57 3130.4 181.03  1.2334 0.2745301
## + pulse  1     11.82 3232.2 182.28  0.1244 0.7265284
## + fore   1      0.22 3243.8 182.42  0.0023 0.9620247
## + calf   1      0.00 3244.0 182.42  0.0000 0.9959042
## - yrage  1   1368.44 4612.4 192.15 14.7643 0.0004912 ***
## - wt     1   2515.33 5759.3 200.81 27.1384 8.512e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm.indian.step.red.AIC)
```

```
##
## Call:
## lm(formula = sysbp ~ wt + yrage + chin, data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.6382  -6.6316   0.4521   6.3593  24.2086
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  52.9092    15.0895   3.506 0.001266 **
## wt            1.4407     0.2766   5.209 8.51e-06 ***
## yrage       -27.3522     7.1185  -3.842 0.000491 ***
## chin         -1.0135     0.6945  -1.459 0.153407
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.627 on 35 degrees of freedom
## Multiple R-squared:  0.5033, Adjusted R-squared:  0.4608
## F-statistic: 11.82 on 3 and 35 DF,  p-value: 1.684e-05
```

```
##stepwise selection starting from full model
lm.indian.bothfull.red.AIC <- step(lm.indian.full, direction="both", test="F")  #default AIC
```

```
## Start:  AIC=186.6
## sysbp ~ wt + ht + chin + fore + calf + pulse + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - calf   1      2.86 3099.3 184.64  0.0287 0.8666427
## - pulse  1     14.61 3111.1 184.79  0.1463 0.7046990
## - fore   1     27.00 3123.4 184.94  0.2703 0.6068061
## - ht     1    131.88 3228.3 186.23  1.3203 0.2593289
## <none>               3096.4 186.60
## - chin   1    186.85 3283.3 186.89  1.8706 0.1812390
## - yrage  1   1386.76 4483.2 199.04 13.8835 0.0007773 ***
## - wt     1   1956.49 5052.9 203.70 19.5874 0.0001105 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=184.64
## sysbp ~ wt + ht + chin + fore + pulse + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - pulse  1     13.34 3112.6 182.81  0.1377 0.7130185
## - fore   1     26.99 3126.3 182.98  0.2787 0.6011969
## - ht     1    129.56 3228.9 184.24  1.3377 0.2560083
## <none>               3099.3 184.64
## - chin   1    184.03 3283.3 184.89  1.9000 0.1776352
## + calf   1      2.86 3096.4 186.60  0.0287 0.8666427
## - yrage  1   1448.00 4547.3 197.59 14.9504 0.0005087 ***
## - wt     1   1953.77 5053.1 201.70 20.1724 8.655e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=182.81
## sysbp ~ wt + ht + chin + fore + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - fore   1     17.78 3130.4 181.03  0.1885  0.667013
## - ht     1    131.12 3243.8 182.42  1.3902  0.246810
## <none>               3112.6 182.81
## - chin   1    198.30 3310.9 183.22  2.1023  0.156514
## + pulse  1     13.34 3099.3 184.64  0.1377  0.713018
## + calf   1      1.59 3111.1 184.79  0.0163  0.899136
## - yrage  1   1450.02 4562.7 195.72 15.3730  0.000421 ***
## - wt     1   1983.51 5096.2 200.03 21.0290 6.219e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=181.03
## sysbp ~ wt + ht + chin + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## - ht     1    113.57 3244.0 180.42  1.2334 0.2745301
## <none>               3130.4 181.03
## - chin   1    287.20 3417.6 182.45  3.1193 0.0863479 .
## + fore   1     17.78 3112.6 182.81  0.1885 0.6670134
## + pulse  1      4.12 3126.3 182.98  0.0435 0.8360534
## + calf   1      2.13 3128.3 183.00  0.0225 0.8818075
## - yrage  1   1445.52 4575.9 193.84 15.7000 0.0003607 ***
## - wt     1   2263.64 5394.1 200.25 24.5857 1.945e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=180.42
## sysbp ~ wt + chin + yrage
##
##         Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>               3244.0 180.42
## - chin   1    197.37 3441.4 180.72  2.1295 0.1534065
## + ht     1    113.57 3130.4 181.03  1.2334 0.2745301
## + pulse  1     11.82 3232.2 182.28  0.1244 0.7265284
## + fore   1      0.22 3243.8 182.42  0.0023 0.9620247
## + calf   1      0.00 3244.0 182.42  0.0000 0.9959042
## - yrage  1   1368.44 4612.4 192.15 14.7643 0.0004912 ***
## - wt     1   2515.33 5759.3 200.81 27.1384 8.512e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm.indian.bothfull.red.AIC)
```

```
##
## Call:
## lm(formula = sysbp ~ wt + chin + yrage, data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.6382  -6.6316   0.4521   6.3593  24.2086
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  52.9092    15.0895   3.506 0.001266 **
## wt            1.4407     0.2766   5.209 8.51e-06 ***
## chin         -1.0135     0.6945  -1.459 0.153407
## yrage       -27.3522     7.1185  -3.842 0.000491 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.627 on 35 degrees of freedom
## Multiple R-squared:  0.5033, Adjusted R-squared:  0.4608
## F-statistic: 11.82 on 3 and 35 DF,  p-value: 1.684e-05
```