Exercise_9.5.R

fhernanb

Thu May 04 09:36:48 2017

```
require(MASS) # to load the stepAIC function
```

```
## Loading required package: MASS
```

```
require(MPV) # to load the data
```

```
## Loading required package: MPV
```

```
##
## Attaching package: 'MPV'
```

```
## The following object is masked from 'package:MASS':
##
##     cement
```

```
## The following object is masked from 'package:datasets':
##
##     stackloss
```

```
# Excersice 9.5 from MPV
data(table.b3)
table.b3[22:26,] # Can you see the missing values?
```

```
##        y    x1  x2  x3  x4   x5 x6 x7    x8   x9  x10 x11
## 22 21.47 360.0 180 290 8.4 2.45  2  3 214.2 76.3 4250   1
## 23 16.59 400.0 185  NA 7.6 3.08  4  3 196.0 73.0 3850   1
## 24 31.90  96.9  75  83 9.0 4.30  2  5 165.2 61.8 2275   0
## 25 29.40 140.0  86  NA 8.0 2.92  2  4 176.4 65.4 2150   0
## 26 13.27 460.0 223 366 8.0 3.00  4  3 228.0 79.8 5430   1
```

```
datis <- table.b3[-c(23,25),]

# The full model ---------------------------------------------------------------
full.model <- lm(y ~ ., data = datis)
summary(full.model)
```

```
##
## Call:
## lm(formula = y ~ ., data = datis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3441 -1.6711 -0.4486  1.4906  5.2508
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.339838  30.355375   0.571   0.5749
## x1          -0.075588   0.056347  -1.341   0.1964
## x2          -0.069163   0.087791  -0.788   0.4411
## x3           0.115117   0.088113   1.306   0.2078
## x4           1.494737   3.101464   0.482   0.6357
## x5           5.843495   3.148438   1.856   0.0799 .
## x6           0.317583   1.288967   0.246   0.8082
## x7          -3.205390   3.109185  -1.031   0.3162
## x8           0.180811   0.130301   1.388   0.1822
## x9          -0.397945   0.323456  -1.230   0.2344
## x10         -0.005115   0.005896  -0.868   0.3971
## x11          0.638483   3.021680   0.211   0.8350
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.227 on 18 degrees of freedom
## Multiple R-squared:  0.8355, Adjusted R-squared:  0.7349
## F-statistic:  8.31 on 11 and 18 DF,  p-value: 5.231e-05
```

```
# logLik and AIC for full.model
length(coef(full.model))            # Number of betas
```

```
## [1] 12
```

```
logLik(full.model)                  # logLik with 13 df
```

```
## 'log Lik.' -70.04893 (df=13)
```

```
-2 * logLik(full.model) + 2 * 13    # AIC manually
```

```
## 'log Lik.' 166.0979 (df=13)
```

```
AIC(full.model, k=2)                # AIC automatically
```

```
## [1] 166.0979
```

```
AIC(full.model, k=log(30))               # BIC automatically
```

```
## [1] 184.3134
```

```
# backward selection ------------------------------------------------------
modback <- stepAIC(full.model, trace=TRUE, direction="backward")
```

```
## Start:  AIC=78.96
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11
##
##        Df Sum of Sq    RSS    AIC
## - x11   1     0.465 187.87 77.036
## - x6    1     0.632 188.03 77.063
## - x4    1     2.418 189.82 77.346
## - x2    1     6.462 193.86 77.979
## - x10   1     7.836 195.24 78.190
## - x7    1    11.065 198.47 78.683
## <none>              187.40 78.962
## - x9    1    15.758 203.16 79.384
## - x3    1    17.770 205.17 79.679
## - x1    1    18.736 206.14 79.820
## - x8    1    20.047 207.45 80.011
## - x5    1    35.864 223.26 82.215
##
## Step:  AIC=77.04
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10
##
```

```
##        Df Sum of Sq    RSS    AIC
## - x6    1     0.536 188.40 75.121
## - x4    1     2.363 190.23 75.411
## - x2    1     6.642 194.51 76.078
## - x10   1     7.985 195.85 76.285
## <none>              187.87 77.036
## - x7    1    14.124 201.99 77.211
## - x9    1    16.914 204.78 77.622
## - x3    1    17.815 205.68 77.754
## - x1    1    18.280 206.15 77.822
## - x8    1    20.301 208.17 78.114
## - x5    1    36.370 224.24 80.345
##
## Step:  AIC=75.12
## y ~ x1 + x2 + x3 + x4 + x5 + x7 + x8 + x9 + x10
##
##        Df Sum of Sq    RSS    AIC
## - x4    1     3.451 191.85 73.666
## - x2    1     6.932 195.33 74.205
## - x10   1     9.351 197.75 74.574
## <none>              188.40 75.121
## - x7    1    14.473 202.87 75.342
## - x3    1    17.802 206.20 75.830
## - x9    1    18.146 206.55 75.880
## - x1    1    18.780 207.18 75.972
## - x8    1    21.244 209.65 76.326
## - x5    1    39.332 227.73 78.809
##
## Step:  AIC=73.67
## y ~ x1 + x2 + x3 + x5 + x7 + x8 + x9 + x10
##
##        Df Sum of Sq    RSS    AIC
## - x2    1    10.780 202.63 73.306
## - x7    1    11.113 202.97 73.355
## <none>              191.85 73.666
## - x10   1    14.988 206.84 73.923
## - x1    1    16.602 208.46 74.156
## - x9    1    18.072 209.92 74.366
## - x3    1    21.314 213.17 74.826
## - x8    1    28.835 220.69 75.867
## - x5    1    40.323 232.18 77.389
##
## Step:  AIC=73.31
## y ~ x1 + x3 + x5 + x7 + x8 + x9 + x10
##
##        Df Sum of Sq    RSS    AIC
## - x7    1    10.457 213.09 72.815
## - x3    1    10.595 213.23 72.835
## - x1    1    11.998 214.63 73.032
## - x9    1    12.643 215.28 73.122
## - x10   1    13.887 216.52 73.295
## <none>              202.63 73.306
## - x8    1    27.665 230.30 75.145
## - x5    1    30.191 232.82 75.472
```

```
##
## Step:  AIC=72.82
## y ~ x1 + x3 + x5 + x8 + x9 + x10
##
##        Df Sum of Sq    RSS    AIC
## - x3    1    4.8720 217.96 71.494
## - x9    1    5.2049 218.29 71.539
## - x1    1    5.3212 218.41 71.555
## <none>              213.09 72.815
## - x10   1   18.3677 231.46 73.296
## - x5    1   23.3458 236.44 73.934
## - x8    1   26.0316 239.12 74.273
##
## Step:  AIC=71.49
## y ~ x1 + x5 + x8 + x9 + x10
##
##        Df Sum of Sq    RSS    AIC
## - x1    1     0.765 218.73 69.599
## - x9    1     5.863 223.82 70.290
## <none>              217.96 71.494
## - x10   1    20.291 238.25 72.164
## - x5    1    23.020 240.98 72.506
## - x8    1    31.634 249.59 73.559
##
## Step:  AIC=69.6
## y ~ x5 + x8 + x9 + x10
##
##        Df Sum of Sq    RSS    AIC
## - x9    1     5.097 223.82 68.290
## <none>              218.73 69.599
## - x5    1    40.404 259.13 72.684
## - x8    1    57.407 276.13 74.591
## - x10   1   135.105 353.83 82.029
##
## Step:  AIC=68.29
## y ~ x5 + x8 + x10
##
##        Df Sum of Sq    RSS    AIC
## <none>              223.82 68.290
## - x5    1    36.314 260.14 70.800
## - x8    1    52.960 276.78 72.661
## - x10   1   194.838 418.66 85.076
```

```
modback$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11
##
## Final Model:
## y ~ x5 + x8 + x10
##
##
```

```
##    Step Df   Deviance Resid. Df Resid. Dev      AIC
## 1                            18   187.4007 78.96155
## 2 - x11  1  0.4648362        19   187.8655 77.03587
## 3  - x6  1  0.5356445        20   188.4012 75.12128
## 4  - x4  1  3.4514854        21   191.8526 73.66591
## 5  - x2  1 10.7796848        22   202.6323 73.30587
## 6  - x7  1 10.4571693        23   213.0895 72.81545
## 7  - x3  1  4.8720101        24   217.9615 71.49363
## 8  - x1  1  0.7654631        25   218.7270 69.59881
## 9  - x9  1  5.0970905        26   223.8241 68.28989
```

```r
summary(modback)
```

```
##
## Call:
## lm(formula = y ~ x5 + x8 + x10, data = datis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6101 -1.9868 -0.6613  2.0369  5.8811
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.590404  11.771925   0.390   0.6998
## x5           2.597240   1.264562   2.054   0.0502 .
## x8           0.217814   0.087817   2.480   0.0199 *
## x10         -0.009485   0.001994  -4.757 6.38e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.934 on 26 degrees of freedom
## Multiple R-squared:  0.8035, Adjusted R-squared:  0.7808
## F-statistic: 35.44 on 3 and 26 DF,  p-value: 2.462e-09
```

```r
# forward selection ----------------------------------------------------
empty.model <- lm(y ~ 1, data = datis)
horizonte <- formula( lm(y ~ ., data = datis) )
horizonte
```

```
## y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11
```

```r
modforw <- stepAIC(empty.model, trace=FALSE, direction="forward",
                   scope=horizonte)
modforw$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## y ~ 1
##
## Final Model:
## y ~ x1 + x4
##
##
##   Step Df  Deviance Resid. Df Resid. Dev       AIC
## 1                          29  1139.1050 111.10402
## 2 + x1  1 866.49528         28   272.6097  70.20532
## 3 + x4  1  18.57161         27   254.0381  70.08861
```

```
summary(modforw)
```

```
## 
## Call:
## lm(formula = y ~ x1 + x4, data = datis)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -6.5011 -2.1243 -0.3884  1.9964  6.9582 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  7.179421  18.787955   0.382    0.705    
## x1          -0.044479   0.005225  -8.513 3.98e-09 ***
## x4           3.077228   2.190294   1.405    0.171    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3.067 on 27 degrees of freedom
## Multiple R-squared:  0.777,  Adjusted R-squared:  0.7605 
## F-statistic: 47.03 on 2 and 27 DF,  p-value: 1.594e-09
```

```
# Comparing ---------------------------------------------------------------
coef(modback)
```

```
##  (Intercept)           x5           x8          x10 
##  4.590404189  2.597240342  0.217814237 -0.009485195 
```

```
AIC(modback)
```

```
## [1] 155.4262
```

```
coef(modforw)
```

```
## (Intercept)          x1          x4 
##  7.17942062 -0.04447915  3.07722832 
```

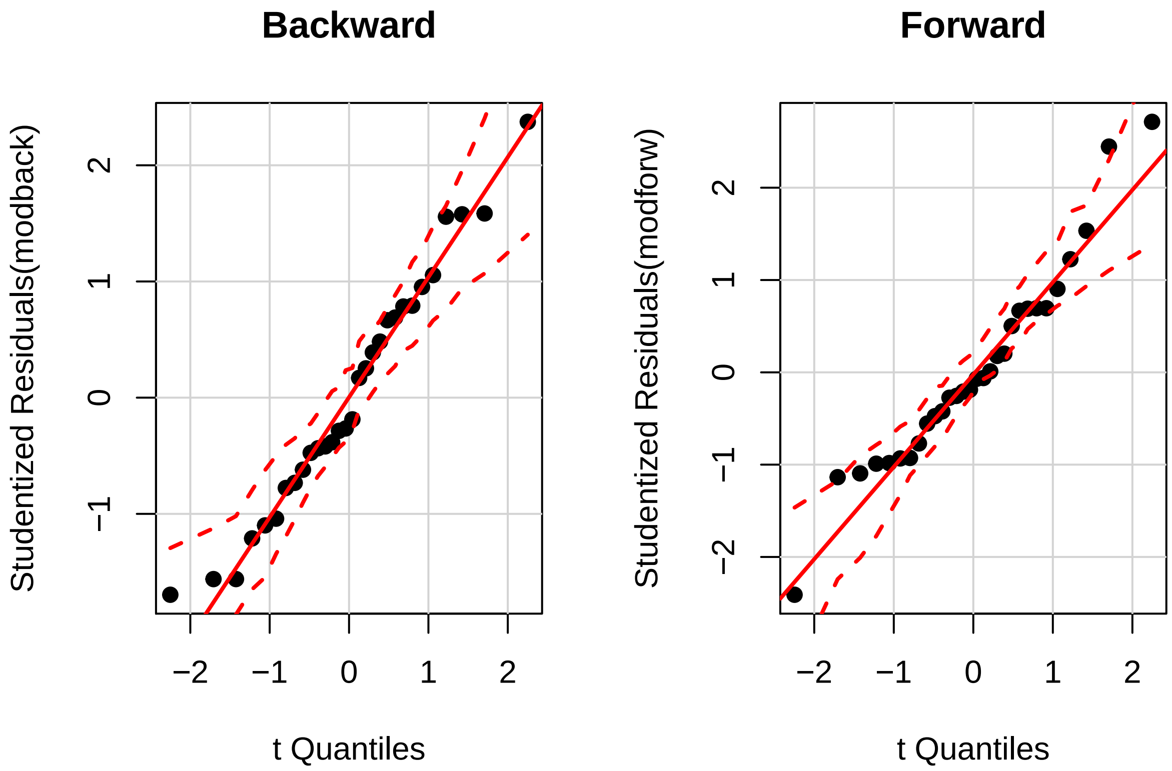
```
AIC(modforw)
```

```
## [1] 157.2249
```

```
# In a graphical way -----------------------------------------------------
par(mfrow=c(1,2))
require(car)
```

```
## Loading required package: car
```

```
qqPlot(modback, main="Backward", pch=19)
qqPlot(modforw, main="Forward", pch=19)
```



```r
library(leaps)
prueba <- regsubsets(y ~ ., data = datis, nbest=2, intercept=T)
summary(prueba)
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ ., data = datis, nbest = 2, intercept = T)
## 11 Variables  (and intercept)
##     Forced in Forced out
## x1      FALSE      FALSE
## x2      FALSE      FALSE
## x3      FALSE      FALSE
## x4      FALSE      FALSE
## x5      FALSE      FALSE
## x6      FALSE      FALSE
## x7      FALSE      FALSE
## x8      FALSE      FALSE
## x9      FALSE      FALSE
## x10     FALSE      FALSE
## x11     FALSE      FALSE
## 2 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          x1  x2  x3  x4  x5  x6  x7  x8  x9  x10 x11
## 1  ( 1 ) "*" " " " " " " " " " " " " " " " " " " " "
## 1  ( 2 ) " " " " " " " " " " " " " " " " " " "*" " "
## 2  ( 1 ) "*" " " " " "*" " " " " " " " " " " " " " "
## 2  ( 2 ) "*" " " " " " " " " "*" " " " " " " " " " "
## 3  ( 1 ) " " " " " " " " "*" " " " " "*" " " "*" " "
## 3  ( 2 ) " " " " " " " " " " " " "*" "*" " " "*" " "
## 4  ( 1 ) " " " " " " " " "*" " " " " "*" "*" "*" " "
## 4  ( 2 ) " " " " " " "*" "*" " " " " "*" " " "*" " "
## 5  ( 1 ) " " " " " " " " "*" " " "*" "*" "*" "*" " "
## 5  ( 2 ) " " " " " " " " "*" " " " " "*" "*" "*" "*"
## 6  ( 1 ) " " " " " " "*" "*" " " "*" "*" "*" "*" " "
## 6  ( 2 ) " " "*" "*" " " "*" " " " " "*" "*" "*" " "
## 7  ( 1 ) "*" " " "*" " " "*" " " "*" "*" "*" "*" " "
## 7  ( 2 ) "*" " " "*" "*" "*" " " "*" "*" "*" " " " "
## 8  ( 1 ) "*" "*" "*" " " "*" " " "*" "*" "*" "*" " "
## 8  ( 2 ) "*" " " "*" "*" "*" " " "*" "*" "*" "*" " "
```

```
summary(prueba)
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ ., data = datis, nbest = 2, intercept = T)
## 11 Variables  (and intercept)
##     Forced in Forced out
## x1      FALSE      FALSE
## x2      FALSE      FALSE
## x3      FALSE      FALSE
## x4      FALSE      FALSE
## x5      FALSE      FALSE
## x6      FALSE      FALSE
## x7      FALSE      FALSE
## x8      FALSE      FALSE
## x9      FALSE      FALSE
## x10     FALSE      FALSE
## x11     FALSE      FALSE
## 2 subsets of each size up to 8
## Selection Algorithm: exhaustive
##           x1  x2  x3  x4  x5  x6  x7  x8  x9  x10 x11
## 1  ( 1 ) "*" " " " " " " " " " " " " " " " " " " " "
## 1  ( 2 ) " " " " " " " " " " " " " " " " " " "*" " "
## 2  ( 1 ) "*" " " " " "*" " " " " " " " " " " " " " "
## 2  ( 2 ) "*" " " " " " " " " "*" " " " " " " " " " "
## 3  ( 1 ) " " " " " " " " "*" " " " " "*" " " "*" " "
## 3  ( 2 ) " " " " " " " " " " " " "*" "*" " " "*" " "
## 4  ( 1 ) " " " " " " " " "*" " " " " "*" "*" "*" " "
## 4  ( 2 ) " " " " " " "*" "*" " " " " "*" " " "*" " "
## 5  ( 1 ) " " " " " " " " "*" " " "*" "*" "*" "*" " "
## 5  ( 2 ) " " " " " " " " "*" " " " " "*" "*" "*" "*"
## 6  ( 1 ) " " " " " " "*" "*" " " "*" "*" "*" "*" " "
## 6  ( 2 ) " " "*" "*" " " "*" " " " " "*" "*" "*" " "
## 7  ( 1 ) "*" " " "*" " " "*" " " "*" "*" "*" "*" " "
## 7  ( 2 ) "*" " " "*" "*" "*" " " "*" "*" "*" " " " "
## 8  ( 1 ) "*" "*" "*" " " "*" " " "*" "*" "*" "*" " "
## 8  ( 2 ) "*" " " "*" "*" "*" " " "*" "*" "*" "*" " "
```

```
do.call(cbind,(summary(prueba)[c("rsq","rss","adjr2","cp","bic")]))
```

```
##             rsq      rss     adjr2         cp       bic
##  [1,] 0.7606808 272.6097 0.7521337  0.1844047 -36.09631
##  [2,] 0.7274044 310.5150 0.7176688  3.8252404 -32.19058
##  [3,] 0.7769845 254.0381 0.7604648  0.4005851 -34.81182
##  [4,] 0.7741259 257.2944 0.7573944  0.7133507 -34.42973
##  [5,] 0.8035088 223.8241 0.7808368 -0.5015020 -35.20935
##  [6,] 0.7886898 240.7045 0.7643079  1.1198744 -33.02806
##  [7,] 0.8079835 218.7270 0.7772608  1.0089180 -32.49923
##  [8,] 0.8059055 221.0940 0.7748504  1.2362726 -32.17632
##  [9,] 0.8112433 215.0137 0.7719190  2.6522501 -29.61171
## [10,] 0.8091718 217.3733 0.7694159  2.8789005 -29.28427
## [11,] 0.8145684 211.2261 0.7661949  4.2884510 -26.74369
## [12,] 0.8142238 211.6186 0.7657605  4.3261490 -26.68800
## [13,] 0.8221127 202.6323 0.7655122  5.4630135 -24.58857
## [14,] 0.8219563 202.8105 0.7653060  5.4801255 -24.56221
## [15,] 0.8315760 191.8526 0.7674144  6.4276153 -22.82734
## [16,] 0.8285201 195.3336 0.7631944  6.7619658 -22.28790
```

```
# To have x8 and x3 in the model
prueba <- regsubsets(y ~ ., data = datis, nbest=1, intercept=T,
                     force.in=c("x8", "x3"))
summary(prueba)
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ ., data = datis, nbest = 1, intercept = T,
##     force.in = c("x8", "x3"))
## 11 Variables  (and intercept)
##     Forced in Forced out
## x3      FALSE      FALSE
## x8      FALSE      FALSE
## x1       TRUE      FALSE
## x2      FALSE      FALSE
## x4      FALSE      FALSE
## x5      FALSE      FALSE
## x6      FALSE      FALSE
## x7       TRUE      FALSE
## x9      FALSE      FALSE
## x10     FALSE      FALSE
## x11     FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          x3  x8  x1  x2  x4  x5  x6  x7  x9  x10 x11
## 3  ( 1 ) "*" "*" " " " " " " " " " " " " " " "*" " "
## 4  ( 1 ) "*" "*" " " " " " " "*" " " " " " " "*" " "
## 5  ( 1 ) "*" "*" " " "*" " " "*" " " " " " " "*" " "
## 6  ( 1 ) "*" "*" " " "*" " " "*" " " " " "*" "*" " "
## 7  ( 1 ) "*" "*" "*" " " " " "*" " " "*" "*" "*" " "
## 8  ( 1 ) "*" "*" "*" "*" " " "*" " " "*" "*" "*" " "
```

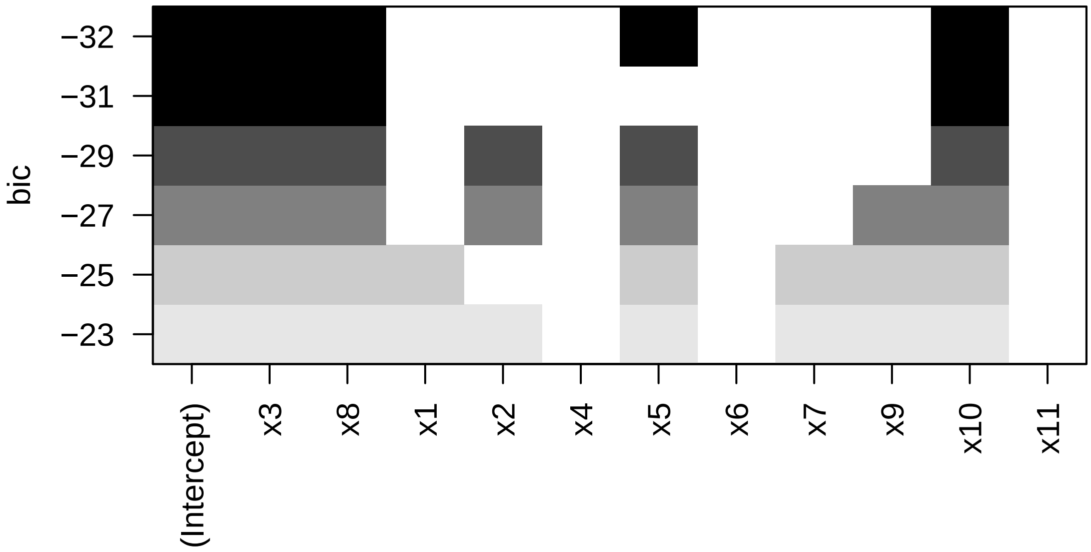
```
do.call(cbind,(summary(prueba)[c("rsq","rss","adjr2","cp","bic")]))
```

```
##            rsq      rss     adjr2       cp       bic
## [1,] 0.7752010 256.0697 0.7492627 2.595716 -31.17167
## [2,] 0.8049501 222.1823 0.7737422 1.340803 -32.02901
## [3,] 0.8088112 217.7841 0.7689803 2.918351 -29.22764
## [4,] 0.8142238 211.6186 0.7657605 4.326149 -26.68800
## [5,] 0.8221127 202.6323 0.7655122 5.463013 -24.58857
## [6,] 0.8315760 191.8526 0.7674144 6.427615 -22.82734
```

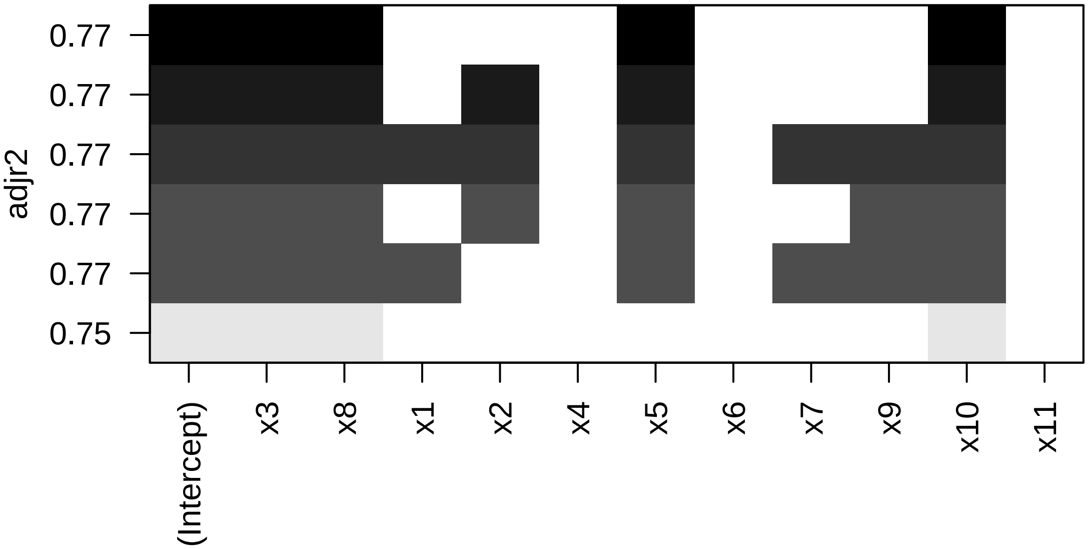
```
# Coefficients for the first 3 models
coef(prueba, 1:3)
```

```
## [[1]]
##  (Intercept)           x3           x8          x10
## 17.860760992 -0.014720296  0.187416711 -0.008456783
##
## [[2]]
## (Intercept)          x3          x8          x5         x10
##  1.88960198  0.01092218  0.23700836  2.93057655 -0.01069613
##
## [[3]]
## (Intercept)          x3          x8          x2          x5         x10
## -0.55381414  0.04401768  0.24735136 -0.04262137  3.72400540 -0.01161920
```

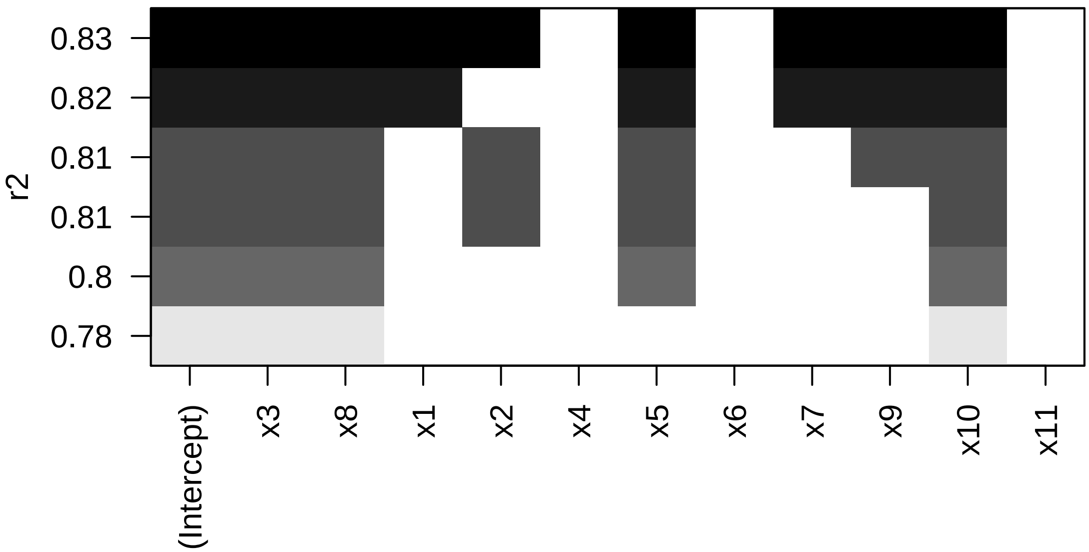
```
# plot a table of models showing variables in each model.
# models are ordered by the selection statistic.
par(mfrow=c(1,1))
plot(prueba, scale="bic")
```



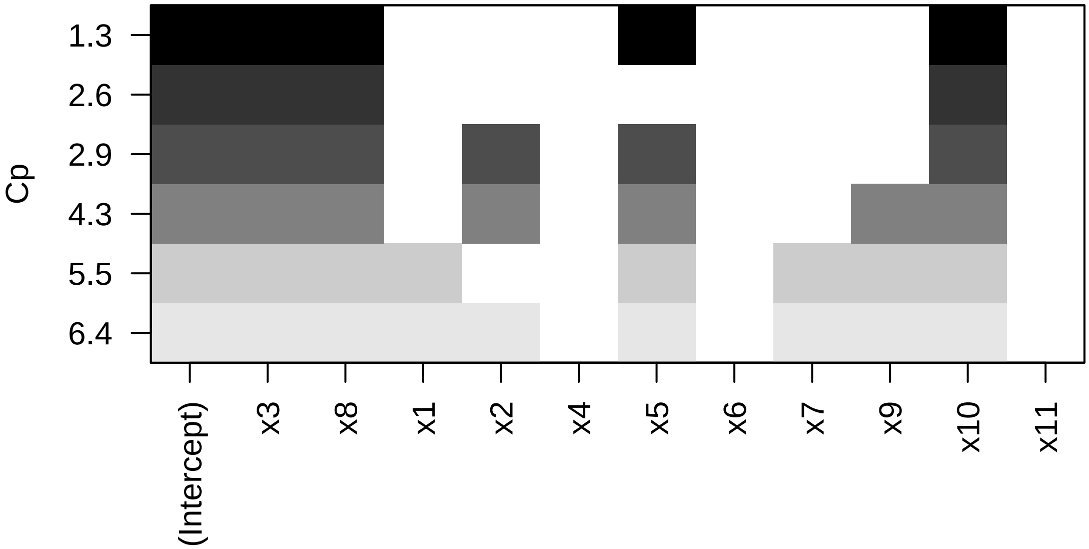
```
plot(prueba, scale="adjr2")
```



```
plot(prueba, scale="r2")
```



```
plot(prueba, scale="Cp")
```


Cp
1.3
2.6
2.9
4.3
5.5
6.4
(Intercept)
x3
x8
x1
x2
x4
x5
x6
x7
x9
x10
x11