Econ 573 Project Mateo

Juan M. Alvarez

Variable Selection

Best Subset Selection

```
# Loading our regularized data
Gas0 <- read.csv("C:/Users/mateo/nc_gas_scaled.csv", na.strings = "?", stringsAsFactors = T)
# Getting rid of the column with county names.
Gas <- subset(Gas0, select = -county)
Gas <- Gas[, !(names(Gas) %in% c("pop", "child", "other", "prpval"))] # getting rid of aliasing
```

We perform best subset selection on the data with aliasing.

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.4.3
```

```
regfit.full <- regsubsets(gas ~ ., data = Gas, nvmax = 35)
reg.summary <- summary(regfit.full)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(gas ~ ., data = Gas, nvmax = 35)
## 29 Variables  (and intercept)
##        Forced in Forced out
## aiw        FALSE      FALSE
## pop05      FALSE      FALSE
## wis        FALSE      FALSE
## upop       FALSE      FALSE
## rpop       FALSE      FALSE
## mhv        FALSE      FALSE
## bach       FALSE      FALSE
## hs         FALSE      FALSE
## x9gr       FALSE      FALSE
## fpov       FALSE      FALSE
## ppov       FALSE      FALSE
## hisp       FALSE      FALSE
## white      FALSE      FALSE
## black      FALSE      FALSE
## voting     FALSE      FALSE
## mhi        FALSE      FALSE
## awpw       FALSE      FALSE
## aaepw      FALSE      FALSE
## unemp      FALSE      FALSE
## capin      FALSE      FALSE
## crime      FALSE      FALSE
## nomuns     FALSE      FALSE
## munp       FALSE      FALSE
## nmunp      FALSE      FALSE
## shigh      FALSE      FALSE
## noveh      FALSE      FALSE
## commt      FALSE      FALSE
## agmort     FALSE      FALSE
## retax      FALSE      FALSE
## 1 subsets of each size up to 29
## Selection Algorithm: exhaustive
##           aiw pop05 wis upop rpop mhv bach hs  x9gr fpov ppov hisp white black
## 1  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  " "  "*"  " "  " "   " "
## 2  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  " "  " "  " "  " "   " "
## 3  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  "*"  " "  "*"  " "   " "
## 4  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  "*"  " "  " "  " "   " "
## 5  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  "*"  " "  "*"  " "   " "
## 6  ( 1 )  "*" " "   " " "*"  " "  " " " "  " " " "  "*"  " "  " "  " "   " "
## 7  ( 1 )  "*" " "   " " "*"  " "  " " " "  " " " "  "*"  " "  "*"  " "   " "
## 8  ( 1 )  "*" " "   " " "*"  " "  " " " "  " " " "  "*"  " "  "*"  " "   " "
## 9  ( 1 )  "*" " "   " " "*"  " "  " " " "  " " " "  "*"  " "  "*"  " "   " "
## 10  ( 1 ) "*" " "   " " "*"  " "  " " " "  " " " "  "*"  " "  "*"  " "   " "
## 11  ( 1 ) "*" " "   " " "*"  " "  " " " "  " " " "  "*"  " "  "*"  " "   " "
## 12  ( 1 ) "*" " "   " " " "  "*"  " " " "  " " " "  "*"  " "  "*"  "*"   "*"
## 13  ( 1 ) "*" " "   " " " "  "*"  "*" " "  " " " "  "*"  " "  "*"  "*"   "*"
## 14  ( 1 ) "*" "*"   " " " "  "*"  "*" " "  " " " "  "*"  " "  "*"  "*"   "*"
## 15  ( 1 ) "*" "*"   " " "*"  "*"  " " "*"  " " " "  "*"  " "  "*"  "*"   "*"
## 16  ( 1 ) "*" "*"   " " "*"  "*"  " " "*"  "*" " "  "*"  " "  "*"  "*"   "*"
```

```
## 17  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  " "  "*"  "*"   "*"
## 18  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 19  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 20  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 21  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 22  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 23  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 24  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 25  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 26  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 27  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 28  ( 1 ) "*" "*"   "*" "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 29  ( 1 ) "*" "*"   "*" "*"  "*"  "*" "*"  "*" "*"  "*"  "*"  "*"  "*"   "*"
##           voting mhi awpw aaepw unemp capin crime nomuns munp nmunp shigh noveh
## 1  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   " "   " "
## 2  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   " "   " "
## 3  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   " "   " "
## 4  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 5  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 6  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 7  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 8  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 9  ( 1 )  " "    "*" "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 10  ( 1 ) " "    "*" "*"  "*"   " "   "*"   " "   " "    " "  " "   "*"   " "
## 11  ( 1 ) " "    "*" "*"  "*"   " "   "*"   " "   " "    " "  " "   "*"   " "
## 12  ( 1 ) " "    "*" "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 13  ( 1 ) " "    "*" "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 14  ( 1 ) " "    "*" "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 15  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 16  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 17  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 18  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 19  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   " "    " "  " "   "*"   " "
## 20  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   " "
## 21  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   " "
## 22  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 23  ( 1 ) "*"    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 24  ( 1 ) "*"    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 25  ( 1 ) "*"    "*" "*"  "*"   "*"   " "   "*"   "*"    "*"  " "   "*"   "*"
## 26  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  " "   "*"   "*"
## 27  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
## 28  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
## 29  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
##           commt agmort retax
## 1  ( 1 )  " "   " "    " "
## 2  ( 1 )  " "   "*"    "*"
## 3  ( 1 )  " "   " "    "*"
## 4  ( 1 )  " "   " "    " "
## 5  ( 1 )  " "   " "    " "
## 6  ( 1 )  " "   " "    "*"
## 7  ( 1 )  " "   " "    "*"
## 8  ( 1 )  " "   " "    "*"
```

```
## 9  ( 1 )  " "   " "    "*"
## 10  ( 1 ) " "   " "    "*"
## 11  ( 1 ) " "   "*"    "*"
## 12  ( 1 ) " "   "*"    "*"
## 13  ( 1 ) " "   "*"    "*"
## 14  ( 1 ) " "   "*"    "*"
## 15  ( 1 ) " "   " "    "*"
## 16  ( 1 ) " "   " "    "*"
## 17  ( 1 ) " "   " "    "*"
## 18  ( 1 ) " "   " "    "*"
## 19  ( 1 ) " "   " "    "*"
## 20  ( 1 ) " "   " "    "*"
## 21  ( 1 ) " "   "*"    "*"
## 22  ( 1 ) "*"   " "    "*"
## 23  ( 1 ) "*"   " "    "*"
## 24  ( 1 ) "*"   "*"    "*"
## 25  ( 1 ) "*"   "*"    "*"
## 26  ( 1 ) "*"   "*"    "*"
## 27  ( 1 ) "*"   "*"    "*"
## 28  ( 1 ) "*"   "*"    "*"
## 29  ( 1 ) "*"   "*"    "*"
```

We observe that aiw, upop, fpov, hisp, awpw, aaepw, shigh and retax seem to be important as they are included in at least 2/3 of the models. Variables such as pop05, mhv, white, black, and crime are seem somewaht important as they are included in at least 1/2 of the models.

```
names(reg.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
reg.summary$rsq
```

```
##  [1] 0.9364481 0.9580577 0.9696996 0.9753697 0.9768665 0.9782998 0.9797290
##  [8] 0.9806635 0.9809374 0.9812714 0.9814523 0.9818044 0.9820096 0.9821954
## [15] 0.9823426 0.9825302 0.9826326 0.9827053 0.9828033 0.9828747 0.9829272
## [22] 0.9829626 0.9829887 0.9830120 0.9830214 0.9830265 0.9830313 0.9830324
## [29] 0.9830324
```

```
reg.summary$rss
```

```
##  [1] 23533.139 15531.146 11220.184  9120.549  8566.271  8035.546  7506.295
##  [8]  7160.278  7058.826  6935.175  6868.178  6737.802  6661.793  6593.018
## [15]  6538.511  6469.012  6431.112  6404.176  6367.903  6341.445  6322.027
## [22]  6308.903  6299.239  6290.633  6287.136  6285.237  6283.467  6283.058
## [29]  6283.054
```

As expected, the R^2 statistic increases monotonically as more variables are added to the model. Conversely, the RSS decreases monotonically as more variables are added.

```
reg.summary$adjr2
```

```
##  [1] 0.9357996 0.9571929 0.9687527 0.9743326 0.9756360 0.9768998 0.9781867
##  [8] 0.9789635 0.9790312 0.9791670 0.9791338 0.9792946 0.9792902 0.9792628
## [15] 0.9791894 0.9791626 0.9790320 0.9788621 0.9787191 0.9785392 0.9783307
## [22] 0.9780948 0.9778406 0.9775758 0.9772854 0.9769812 0.9766681 0.9763410
## [29] 0.9760030
```

```
max(reg.summary$adjr2)
```

```
## [1] 0.9792946
```

```
which.max(reg.summary$adjr2)
```

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

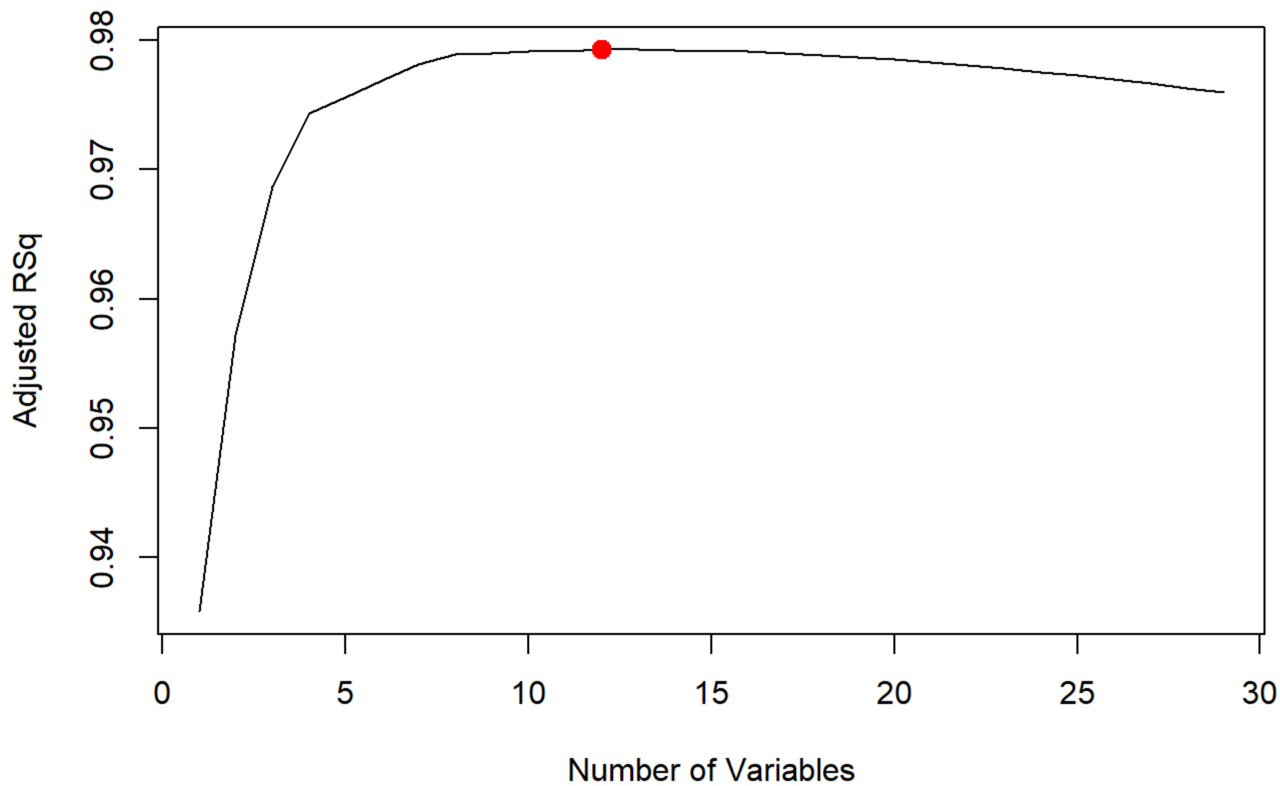
However, we see that the maximum value for adjusted R^2 is 0.9792946 which is achieved by the model with 12 variables, and after that it decreases as we add more variables.

```
coef(regfit.full, 12)
```

```
## (Intercept)         aiw        rpop        fpov        hisp       white
##   58.040000    2.664815    3.471451   26.013634  -22.481056  -16.726071
##       black         mhi        awpw       aaepw       shigh      agmort
##  -11.764041    2.153835  -49.780272  102.339100    5.770164  -17.362629
##       retax
##   40.834075
```

We continue to compare different statistics such as the Cp and the BIC to compare goodness of fit.

```
plot(reg.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
points(12, reg.summary$adjr2[12], col = "red", cex = 2, pch = 20)
```




```
reg.summary$cp
```

```
##  [1] 166.1845614  79.0337245  33.0049589  11.6127541   7.4375075   3.5246577
##  [7]  -0.3717773  -2.2267735  -1.3570651  -0.7346656   0.5189129   1.0663865
## [13]   2.2195630   3.4533341   4.8460677   6.0717682   7.6495227   9.3494254
## [19]  10.9453147  12.6505442  14.4341996  16.2879823  18.1803196  20.0844360
## [25]  22.0454792  24.0243244  26.0046045  28.0000404  30.0000000
```

```
min(reg.summary$cp)
```

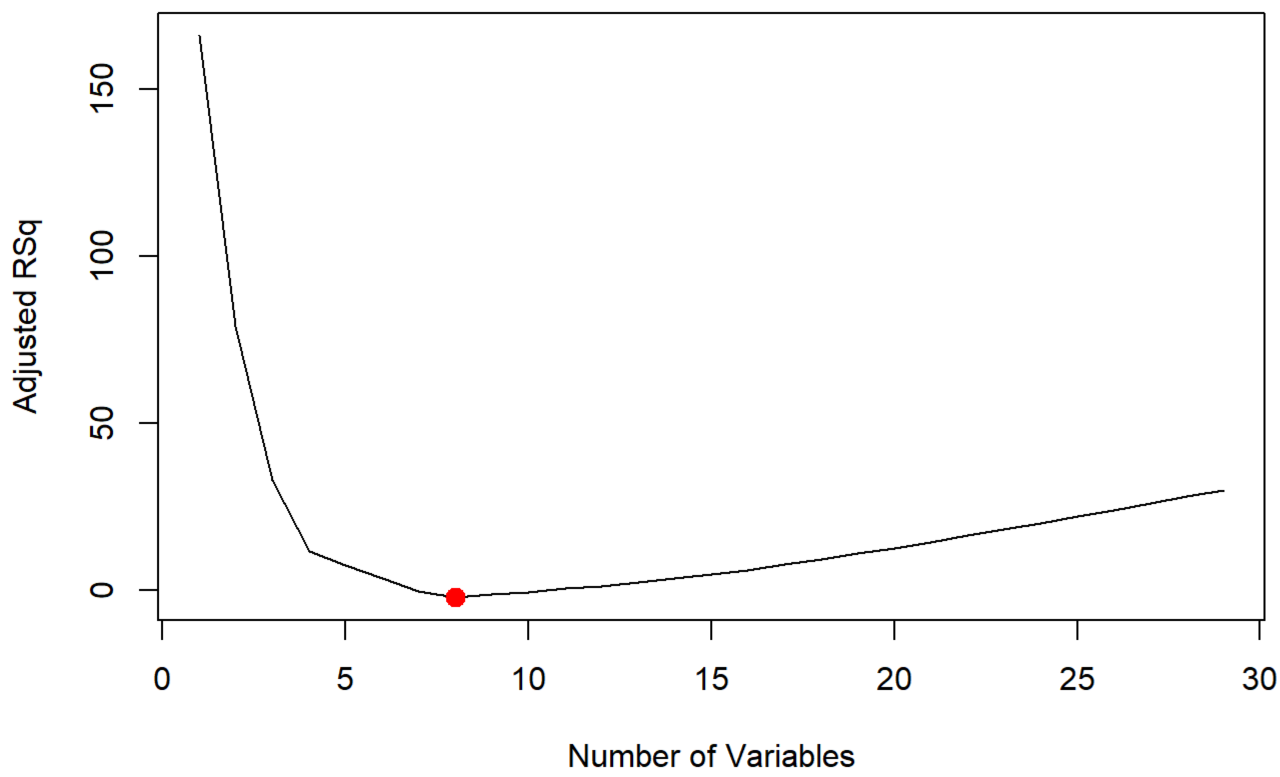
```
## [1] -2.226774
```

```
which.min(reg.summary$cp)
```

```
## [1] 8
```

We see that the model with the least Cp value is the one with 8 variables.

```
plot(reg.summary$cp, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
points(8, reg.summary$cp[8], col = "red", cex = 2, pch = 20)
```




```
reg.summary$bic
```

```
##  [1] -266.3795 -303.3305 -331.2387 -347.3519 -349.0165 -350.8071 -353.0152
##  [8] -353.1293 -349.9512 -347.1133 -343.4788 -340.7902 -337.3195 -333.7521
## [15] -329.9771 -326.4405 -322.4230 -318.2375 -314.2003 -310.0115 -305.7130
## [22] -301.3157 -296.8638 -292.3953 -287.8458 -283.2708 -278.6938 -274.0952
## [29] -269.4900
```

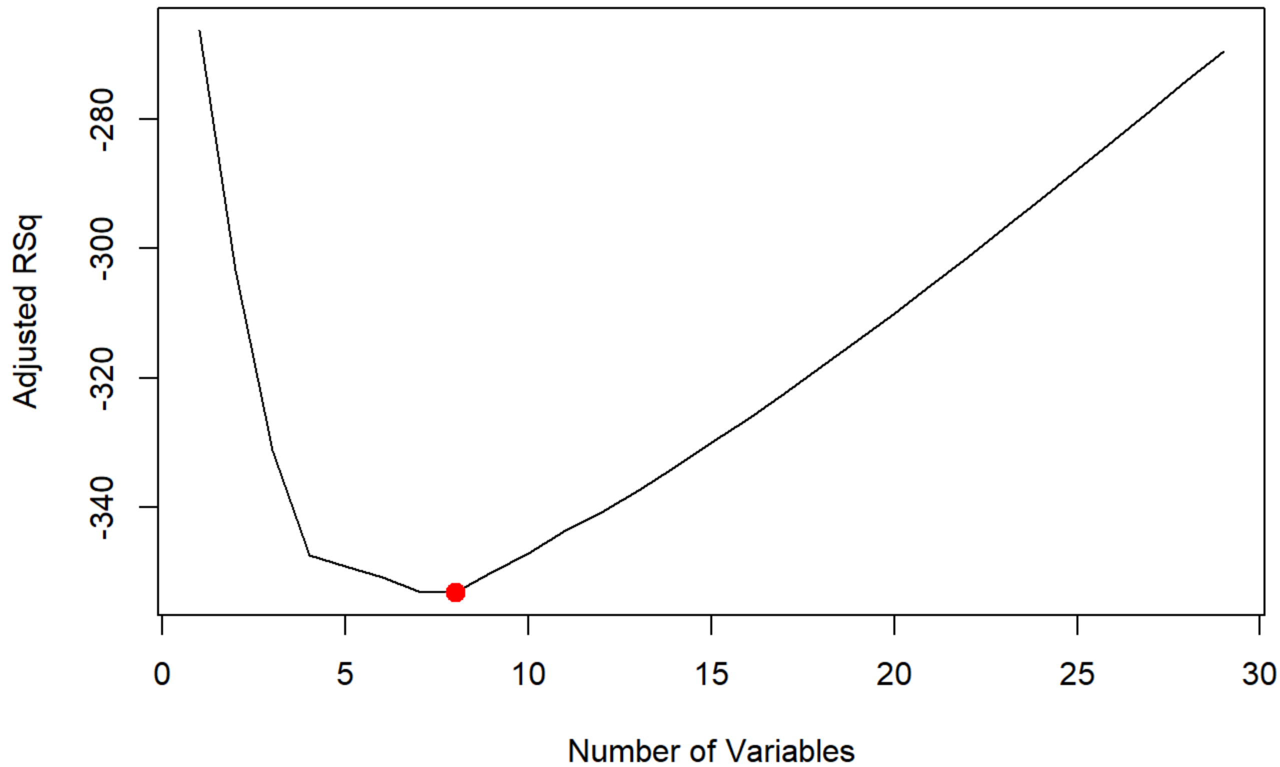
```
min(reg.summary$bic)
```

```
## [1] -353.1293
```

```
which.min(reg.summary$bic)
```

```
## [1] 8
```

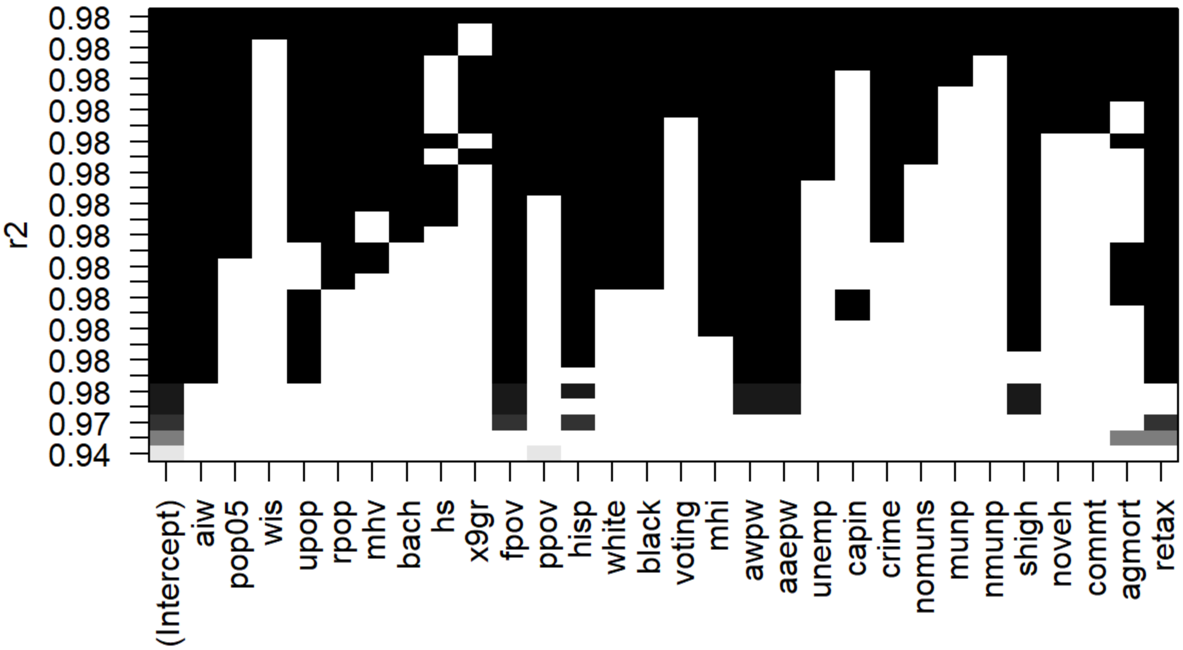
```
plot(reg.summary$bic, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
points(8, reg.summary$bic[8], col = "red", cex = 2, pch = 20)
```



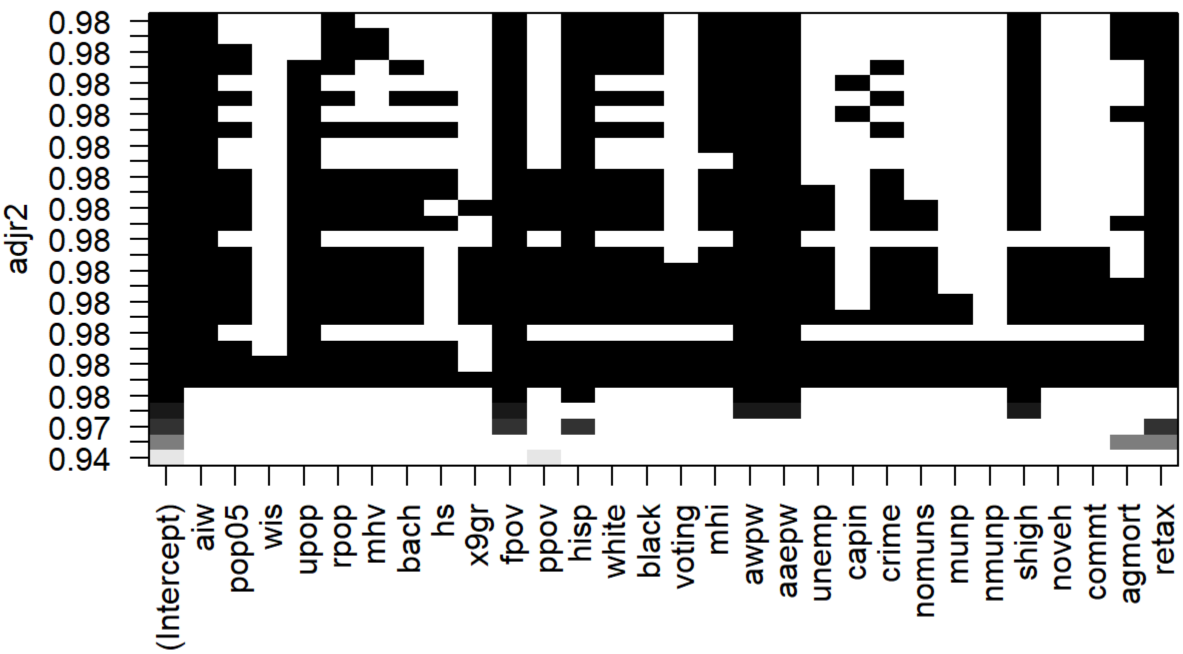

```
coef(regfit.full, 8)
```

```
## (Intercept)         aiw        upop        fpov        hisp        awpw
##   58.040000    2.777718  -36.589771   28.686229  -16.302460  -55.564941
##       aaepw       shigh       retax
##  101.231029    4.678508   33.592998
```

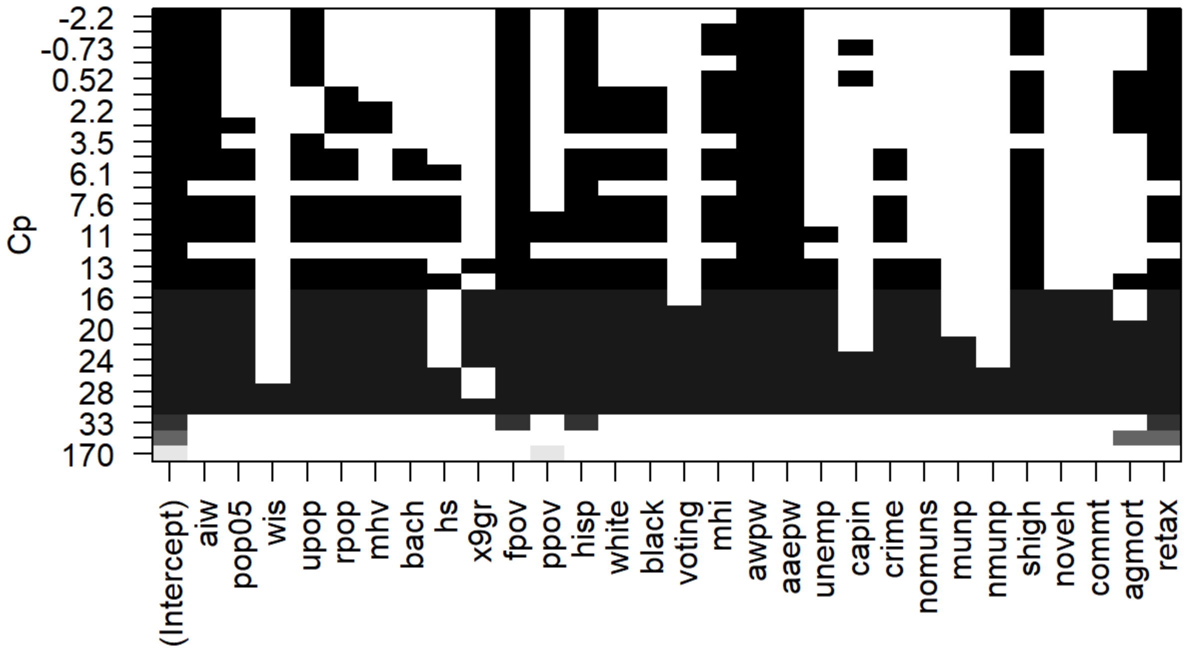
```
plot(regfit.full, scale = "r2")
```



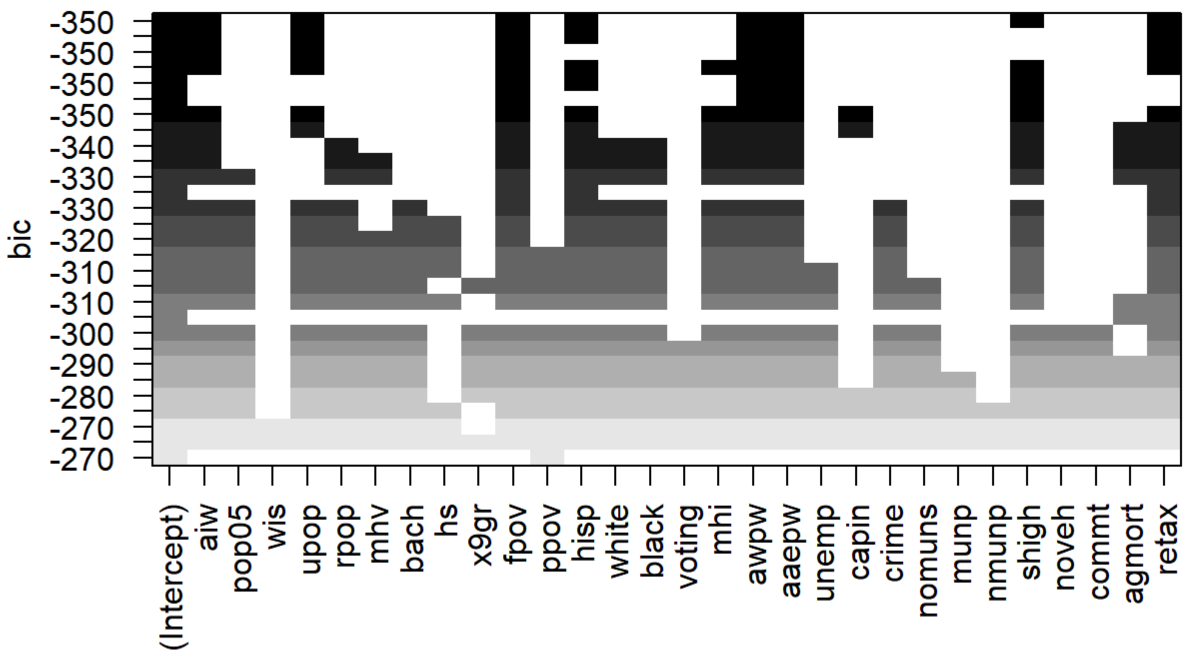

```
plot(regfit.full, scale = "adjr2")
```




```
plot(regfit.full, scale = "Cp")
```




```
plot(regfit.full, scale = "bic")
```



plots

of the variables included (black sqaures) according to the optimal model associated with that statistic.

Conclusion, Choose the one with 8 variables.

Forward and Backward Stepwise Selection

Forward

```
regfit.fwd <- regsubsets(gas ~ ., data = Gas, nvmax = 35, method = "forward")
fwd.summary <- summary(regfit.fwd)
fwd.summary
```

```
## Subset selection object
## Call: regsubsets.formula(gas ~ ., data = Gas, nvmax = 35, method = "forward")
## 29 Variables  (and intercept)
##        Forced in Forced out
## aiw        FALSE      FALSE
## pop05      FALSE      FALSE
## wis        FALSE      FALSE
## upop       FALSE      FALSE
## rpop       FALSE      FALSE
## mhv        FALSE      FALSE
## bach       FALSE      FALSE
## hs         FALSE      FALSE
## x9gr       FALSE      FALSE
## fpov       FALSE      FALSE
## ppov       FALSE      FALSE
## hisp       FALSE      FALSE
## white      FALSE      FALSE
## black      FALSE      FALSE
## voting     FALSE      FALSE
## mhi        FALSE      FALSE
## awpw       FALSE      FALSE
## aaepw      FALSE      FALSE
## unemp      FALSE      FALSE
## capin      FALSE      FALSE
## crime      FALSE      FALSE
## nomuns     FALSE      FALSE
## munp       FALSE      FALSE
## nmunp      FALSE      FALSE
## shigh      FALSE      FALSE
## noveh      FALSE      FALSE
## commt      FALSE      FALSE
## agmort     FALSE      FALSE
## retax      FALSE      FALSE
## 1 subsets of each size up to 29
## Selection Algorithm: forward
##           aiw pop05 wis upop rpop mhv bach hs  x9gr fpov ppov hisp white black
## 1  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  " "  "*"  " "  " "   " "  
## 2  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  " "  "*"  " "  " "   " "  
## 3  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  " "  "*"  " "  "*"   " "  
## 4  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  " "  "*"  "*"  "*"   " "  
## 5  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  " "  "*"  "*"  "*"   " "  
## 6  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  "*"  "*"  "*"  "*"   " "  
## 7  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  "*"  "*"  "*"  "*"   " "  
## 8  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  "*"  "*"  "*"  "*"   " "  
## 9  ( 1 )  " " " "   " " " "  "*"  " " " "  " " " "  "*"  "*"  "*"  "*"   " "  
## 10  ( 1 ) "*" " "   " " " "  "*"  " " " "  " " " "  "*"  "*"  "*"  "*"   " "  
## 11  ( 1 ) "*" " "   " " " "  "*"  " " "*"  " " " "  "*"  "*"  "*"  "*"   " "  
## 12  ( 1 ) "*" " "   " " " "  "*"  " " "*"  " " " "  "*"  "*"  "*"  "*"   "*"  
## 13  ( 1 ) "*" " "   " " " "  "*"  " " "*"  " " " "  "*"  "*"  "*"  "*"   "*"  
## 14  ( 1 ) "*" "*"   " " " "  "*"  " " "*"  " " " "  "*"  "*"  "*"  "*"   "*"  
## 15  ( 1 ) "*" "*"   " " "*"  "*"  " " "*"  " " " "  "*"  "*"  "*"  "*"   "*"  
## 16  ( 1 ) "*" "*"   " " "*"  "*"  " " "*"  " " " "  "*"  "*"  "*"  "*"   "*"  
```

```
## 17  ( 1 ) "*" "*"    " " "*"  "*"  " " "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 18  ( 1 ) "*" "*"    " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 19  ( 1 ) "*" "*"    " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 20  ( 1 ) "*" "*"    " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 21  ( 1 ) "*" "*"    " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 22  ( 1 ) "*" "*"    " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 23  ( 1 ) "*" "*"    " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 24  ( 1 ) "*" "*"    " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 25  ( 1 ) "*" "*"    " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 26  ( 1 ) "*" "*"    " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 27  ( 1 ) "*" "*"    " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 28  ( 1 ) "*" "*"    "*" "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 29  ( 1 ) "*" "*"    "*" "*"  "*"  "*" "*"  "*" "*"  "*"  "*"  "*"  "*"   "*"
##           voting mhi awpw aaepw unemp capin crime nomuns munp nmunp shigh noveh
## 1  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   " "   " "
## 2  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   "*"   " "
## 3  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   "*"   " "
## 4  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   "*"   " "
## 5  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   "*"   " "
## 6  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   "*"   " "
## 7  ( 1 )  " "    " " " "  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 8  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 9  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 10  ( 1 ) " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 11  ( 1 ) " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 12  ( 1 ) " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 13  ( 1 ) " "    " " "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 14  ( 1 ) " "    " " "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 15  ( 1 ) " "    " " "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 16  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 17  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 18  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 19  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   " "    " "  " "   "*"   " "
## 20  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   " "
## 21  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   " "
## 22  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 23  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 24  ( 1 ) "*"    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 25  ( 1 ) "*"    "*" "*"  "*"   "*"   " "   "*"   "*"    "*"  " "   "*"   "*"
## 26  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  " "   "*"   "*"
## 27  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
## 28  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
## 29  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
##           commt agmort retax
## 1  ( 1 )  " "   " "    " "
## 2  ( 1 )  " "   " "    " "
## 3  ( 1 )  " "   " "    " "
## 4  ( 1 )  " "   " "    " "
## 5  ( 1 )  " "   " "    "*"
## 6  ( 1 )  " "   " "    "*"
## 7  ( 1 )  " "   " "    "*"
## 8  ( 1 )  " "   " "    "*"
```

```
## 9  ( 1 )  " "   " "    "*"
## 10  ( 1 ) " "   " "    "*"
## 11  ( 1 ) " "   " "    "*"
## 12  ( 1 ) " "   " "    "*"
## 13  ( 1 ) " "   " "    "*"
## 14  ( 1 ) " "   " "    "*"
## 15  ( 1 ) " "   " "    "*"
## 16  ( 1 ) " "   " "    "*"
## 17  ( 1 ) " "   " "    "*"
## 18  ( 1 ) " "   " "    "*"
## 19  ( 1 ) " "   " "    "*"
## 20  ( 1 ) " "   " "    "*"
## 21  ( 1 ) " "   "*"    "*"
## 22  ( 1 ) " "   "*"    "*"
## 23  ( 1 ) "*"   "*"    "*"
## 24  ( 1 ) "*"   "*"    "*"
## 25  ( 1 ) "*"   "*"    "*"
## 26  ( 1 ) "*"   "*"    "*"
## 27  ( 1 ) "*"   "*"    "*"
## 28  ( 1 ) "*"   "*"    "*"
## 29  ( 1 ) "*"   "*"    "*"
```

```
names(fwd.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
# Comparing R^2 and RSS of all the models
fwd.summary$rsq
```

```
##  [1] 0.9364481 0.9579632 0.9690250 0.9716441 0.9744659 0.9753383 0.9765282
##  [8] 0.9781978 0.9794840 0.9802261 0.9809222 0.9812755 0.9815761 0.9818637
## [15] 0.9821274 0.9823628 0.9825559 0.9827053 0.9828033 0.9828747 0.9829272
## [22] 0.9829558 0.9829864 0.9830117 0.9830213 0.9830265 0.9830313 0.9830324
## [29] 0.9830324
```

```
fwd.summary$rss
```

```
##  [1] 23533.139 15566.142 11469.971 10500.117  9455.209  9132.156  8691.574
##  [8]  8073.309  7597.030  7322.240  7064.480  6933.650  6822.348  6715.839
## [15]  6618.171  6531.015  6459.507  6404.176  6367.903  6341.477  6322.027
## [22]  6311.428  6300.110  6290.722  6287.160  6285.244  6283.467  6283.058
## [29]  6283.054
```

```
which.min(fwd.summary$rss)
```

```
## [1] 29
```

We compare statistic for model fit

```
# Comparing Adjusted R^2
fwd.summary$adjr2
```

```
##  [1] 0.9357996 0.9570964 0.9680570 0.9704502 0.9731077 0.9737473 0.9747423
##  [8] 0.9762811 0.9774324 0.9780043 0.9785374 0.9786928 0.9787910 0.9788765
## [15] 0.9789359 0.9789629 0.9789395 0.9788621 0.9787191 0.9785391 0.9783307
## [22] 0.9780860 0.9778375 0.9775755 0.9772853 0.9769812 0.9766681 0.9763410
## [29] 0.9760030
```

```
max(fwd.summary$adjr2)
```

```
## [1] 0.9789629
```

```
which.max(fwd.summary$adjr2)
```

```
## [1] 16
```

Model with 16 variables has the highest R^2.

```
#Comparing BIC
fwd.summary$bic
```

```
##  [1] -266.3795 -303.1054 -329.0368 -333.2663 -339.1432 -338.0144 -338.3540
##  [8] -341.1279 -342.6033 -341.6823 -340.6608 -337.9249 -334.9380 -331.9063
## [15] -328.7661 -325.4866 -321.9824 -318.2375 -314.2003 -310.0110 -305.7130
## [22] -301.2757 -296.8500 -292.3939 -287.8454 -283.2707 -278.6938 -274.0952
## [29] -269.4900
```

```
min(fwd.summary$bic)
```

```
## [1] -342.6033
```

```
which.min(fwd.summary$bic)
```

```
## [1] 9
```

```
#Comparing cp
fwd.summary$cp
```

```
##  [1] 166.184561  79.423613  35.787858  26.982632  17.341231  15.742069
##  [7]  12.833518   7.945369   4.639112   3.577660   2.705934   3.248338
## [13]   4.008313   4.821696   5.733567   6.762558   7.965880   9.349425
## [19]  10.945315  12.650898  14.434200  16.316120  18.190027  20.085435
## [25]  22.045744  24.024405  26.004605  28.000040  30.000000
```

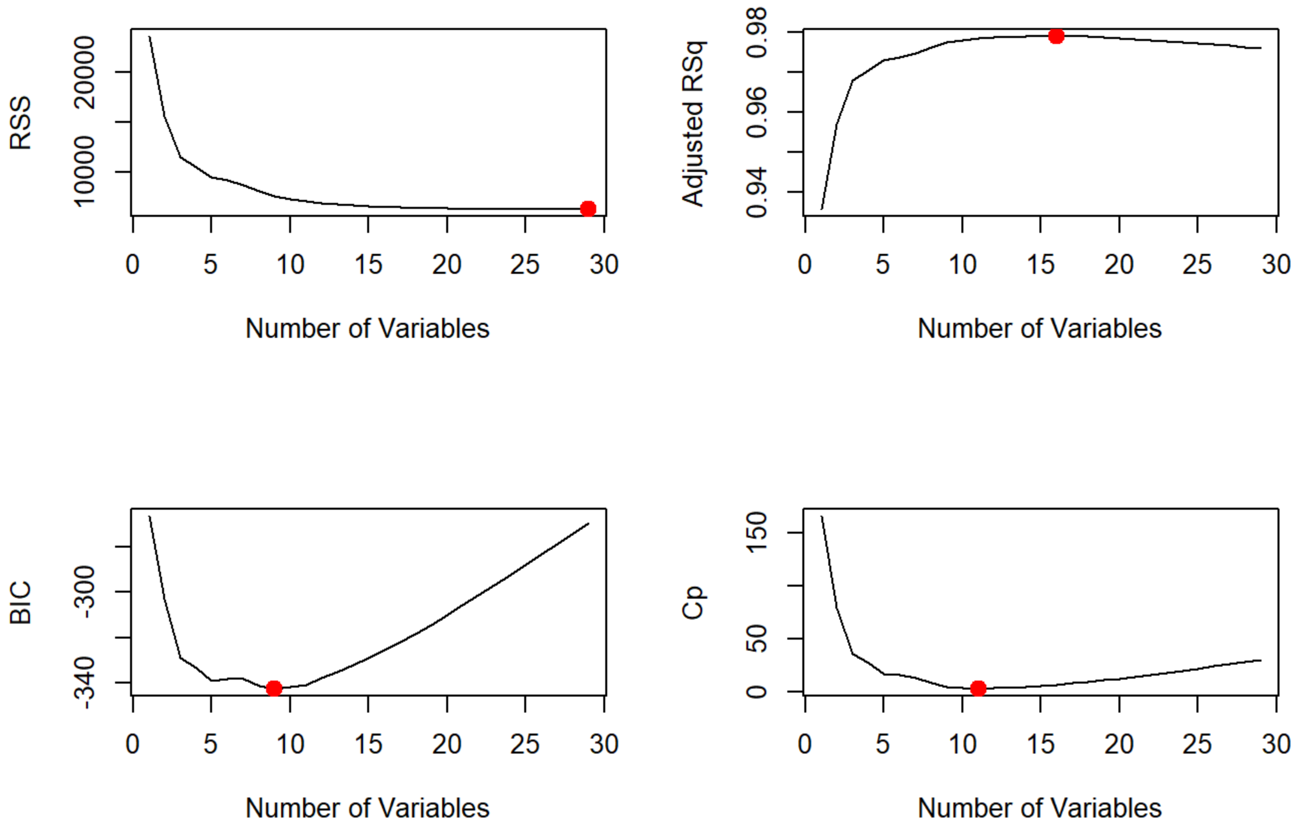
```
min(fwd.summary$cp)
```

```
## [1] 2.705934
```

```
which.min(fwd.summary$cp)
```

```
## [1] 11
```

```
par(mfrow = c(2, 2))
plot(fwd.summary$rss, xlab = "Number of Variables", ylab = "RSS", type = "l")
points(29, fwd.summary$rss[29], col = "red", cex = 2, pch = 20)
plot(fwd.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
points(16, fwd.summary$adjr2[16], col = "red", cex = 2, pch = 20)
plot(fwd.summary$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")
points(9, fwd.summary$bic[9], col = "red", cex = 2, pch = 20)
plot(fwd.summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
points(11, fwd.summary$cp[11], col = "red", cex = 2, pch = 20)
```




```
coef(regfit.fwd, 9)
```

```
## (Intercept)        rpop        fpov        ppov        hisp       white
##  58.0400000   5.8451299  22.2317020   0.7837597 -21.8263236  -9.6383518
##        awpw       aaepw       shigh       retax
## -53.4243254  89.0844766   4.5687761  26.7599318
```

```
coef(regfit.fwd, 11)
```

```
## (Intercept)         aiw        rpop        bach        fpov        ppov
##   58.040000    2.632752    4.994427  -16.791480   14.695717    5.325588
##        hisp       white        awpw       aaepw       shigh       retax
##  -22.853877   -9.565758  -48.673544   93.139118    4.643085   37.420460
```

```
coef(regfit.fwd, 16)
```

```
## (Intercept)         aiw       pop05        upop        rpop        bach
##   58.040000    2.867611    5.294658   49.066136   10.265916  -35.029930
##        fpov        ppov        hisp       white       black         mhi
##   18.586287    3.426006  -27.414276  -37.840496  -18.885415    1.468990
##        awpw       aaepw       crime       shigh       retax
##  -48.218290  103.205967   -4.340587    4.755376   39.969560
```

Backward

```
regfit.bwd <- regsubsets(gas ~ ., data = Gas, nvmax = 35, method = "backward")
bwd.summary <- summary(regfit.bwd)
bwd.summary
```

```
## Subset selection object
## Call: regsubsets.formula(gas ~ ., data = Gas, nvmax = 35, method = "backward")
## 29 Variables  (and intercept)
##        Forced in Forced out
## aiw        FALSE      FALSE
## pop05      FALSE      FALSE
## wis        FALSE      FALSE
## upop       FALSE      FALSE
## rpop       FALSE      FALSE
## mhv        FALSE      FALSE
## bach       FALSE      FALSE
## hs         FALSE      FALSE
## x9gr       FALSE      FALSE
## fpov       FALSE      FALSE
## ppov       FALSE      FALSE
## hisp       FALSE      FALSE
## white      FALSE      FALSE
## black      FALSE      FALSE
## voting     FALSE      FALSE
## mhi        FALSE      FALSE
## awpw       FALSE      FALSE
## aaepw      FALSE      FALSE
## unemp      FALSE      FALSE
## capin      FALSE      FALSE
## crime      FALSE      FALSE
## nomuns     FALSE      FALSE
## munp       FALSE      FALSE
## nmunp      FALSE      FALSE
## shigh      FALSE      FALSE
## noveh      FALSE      FALSE
## commt      FALSE      FALSE
## agmort     FALSE      FALSE
## retax      FALSE      FALSE
## 1 subsets of each size up to 29
## Selection Algorithm: backward
##           aiw pop05 wis upop rpop mhv bach hs  x9gr fpov ppov hisp white black
## 1  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  " "  " "  " "  " "   " "
## 2  ( 1 )  " " " "   " " " "  " "  " " " "  " " " "  " "  " "  " "  " "   " "
## 3  ( 1 )  " " " "   " " " "  "*"  " " " "  " " " "  " "  " "  " "  " "   " "
## 4  ( 1 )  " " " "   " " " "  "*"  " " " "  " " " "  "*"  " "  " "  " "   " "
## 5  ( 1 )  " " " "   " " " "  "*"  " " " "  " " " "  "*"  " "  "*"  " "   " "
## 6  ( 1 )  " " " "   " " " "  "*"  " " " "  " " " "  "*"  " "  "*"  " "   " "
## 7  ( 1 )  "*" " "   " " " "  "*"  " " " "  " " " "  "*"  " "  "*"  " "   " "
## 8  ( 1 )  "*" " "   " " " "  "*"  " " " "  " " " "  "*"  " "  "*"  "*"   " "
## 9  ( 1 )  "*" " "   " " " "  "*"  " " " "  " " " "  "*"  " "  "*"  "*"   "*"
## 10  ( 1 ) "*" " "   " " " "  "*"  " " "*"  " " " "  "*"  " "  "*"  "*"   "*"
## 11  ( 1 ) "*" " "   " " " "  "*"  " " "*"  " " " "  "*"  " "  "*"  "*"   "*"
## 12  ( 1 ) "*" "*"   " " " "  "*"  " " "*"  " " " "  "*"  " "  "*"  "*"   "*"
## 13  ( 1 ) "*" "*"   " " " "  "*"  " " "*"  " " " "  "*"  " "  "*"  "*"   "*"
## 14  ( 1 ) "*" "*"   " " "*"  "*"  " " "*"  " " " "  "*"  " "  "*"  "*"   "*"
## 15  ( 1 ) "*" "*"   " " "*"  "*"  " " "*"  " " " "  "*"  " "  "*"  "*"   "*"
## 16  ( 1 ) "*" "*"   " " "*"  "*"  " " "*"  "*" " "  "*"  " "  "*"  "*"   "*"
```

```
## 17  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  " "  "*"  "*"   "*"
## 18  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 19  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 20  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 21  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 22  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 23  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 24  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 25  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 26  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 27  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 28  ( 1 ) "*" "*"   "*" "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 29  ( 1 ) "*" "*"   "*" "*"  "*"  "*" "*"  "*" "*"  "*"  "*"  "*"  "*"   "*"
##           voting mhi awpw aaepw unemp capin crime nomuns munp nmunp shigh noveh
## 1  ( 1 )  " "    " " " "  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 2  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 3  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 4  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 5  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 6  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 7  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 8  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 9  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 10  ( 1 ) " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 11  ( 1 ) " "    " " "*"  "*"   " "   " "   "*"   " "    " "  " "   " "   " "
## 12  ( 1 ) " "    " " "*"  "*"   " "   " "   "*"   " "    " "  " "   " "   " "
## 13  ( 1 ) " "    " " "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 14  ( 1 ) " "    " " "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 15  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 16  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 17  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 18  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 19  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   " "    " "  " "   "*"   " "
## 20  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   " "
## 21  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 22  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 23  ( 1 ) "*"    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 24  ( 1 ) "*"    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 25  ( 1 ) "*"    "*" "*"  "*"   "*"   " "   "*"   "*"    "*"  " "   "*"   "*"
## 26  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  " "   "*"   "*"
## 27  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
## 28  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
## 29  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
##           commt agmort retax
## 1  ( 1 )  " "   " "    " "
## 2  ( 1 )  " "   " "    " "
## 3  ( 1 )  " "   " "    " "
## 4  ( 1 )  " "   " "    " "
## 5  ( 1 )  " "   " "    " "
## 6  ( 1 )  " "   " "    "*"
## 7  ( 1 )  " "   " "    "*"
## 8  ( 1 )  " "   " "    "*"
```

```
## 9  ( 1 )  " "   " "    "*"
## 10  ( 1 ) " "   " "    "*"
## 11  ( 1 ) " "   " "    "*"
## 12  ( 1 ) " "   " "    "*"
## 13  ( 1 ) " "   " "    "*"
## 14  ( 1 ) " "   " "    "*"
## 15  ( 1 ) " "   " "    "*"
## 16  ( 1 ) " "   " "    "*"
## 17  ( 1 ) " "   " "    "*"
## 18  ( 1 ) " "   " "    "*"
## 19  ( 1 ) " "   " "    "*"
## 20  ( 1 ) " "   " "    "*"
## 21  ( 1 ) " "   " "    "*"
## 22  ( 1 ) "*"   " "    "*"
## 23  ( 1 ) "*"   " "    "*"
## 24  ( 1 ) "*"   "*"    "*"
## 25  ( 1 ) "*"   "*"    "*"
## 26  ( 1 ) "*"   "*"    "*"
## 27  ( 1 ) "*"   "*"    "*"
## 28  ( 1 ) "*"   "*"    "*"
## 29  ( 1 ) "*"   "*"    "*"
```

```
names(bwd.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
# Comparing R^2 and RSS of all the models
bwd.summary$rsq
```

```
##  [1] 0.8141979 0.9338945 0.9636414 0.9735368 0.9764182 0.9780930 0.9790615
##  [8] 0.9796836 0.9804946 0.9806990 0.9811384 0.9815475 0.9818624 0.9821159
## [15] 0.9823426 0.9825302 0.9826326 0.9827053 0.9828033 0.9828747 0.9829210
## [22] 0.9829623 0.9829883 0.9830117 0.9830213 0.9830265 0.9830313 0.9830324
## [29] 0.9830324
```

```
bwd.summary$rss
```

```
##  [1] 68802.104 24478.736 13463.529  9799.255  8732.306  8112.105  7753.468
##  [8]  7523.106  7222.794  7147.132  6984.399  6832.903  6716.324  6622.430
## [15]  6538.511  6469.012  6431.112  6404.176  6367.903  6341.477  6324.302
## [22]  6309.037  6299.379  6290.722  6287.160  6285.244  6283.467  6283.058
## [29]  6283.054
```

```
min(bwd.summary$rss)
```

```
## [1] 6283.054
```

```
which.min(bwd.summary$rss)
```

```
## [1] 29
```

We compare statistic for model fit

```
# Comparing Adjusted R^2
bwd.summary$adjr2
```

```
##  [1] 0.8123020 0.9325315 0.9625051 0.9724226 0.9751638 0.9766797 0.9774684
##  [8] 0.9778976 0.9785441 0.9785303 0.9787807 0.9790024 0.9791206 0.9791703
## [15] 0.9791894 0.9791626 0.9790320 0.9788621 0.9787191 0.9785391 0.9783229
## [22] 0.9780943 0.9778401 0.9775755 0.9772853 0.9769812 0.9766681 0.9763410
## [29] 0.9760030
```

```
max(bwd.summary$adjr2)
```

```
## [1] 0.9791894
```

```
which.max(bwd.summary$adjr2)
```

```
## [1] 15
```

Model with 15 variables has the highest R^2.

```
#Comparing BIC
bwd.summary$bic
```

```
##  [1] -159.0970 -257.8348 -313.0116 -340.1743 -347.0968 -349.8588 -349.7754
##  [8] -348.1863 -347.6549 -344.1028 -341.8008 -339.3886 -336.5043 -333.3070
## [15] -329.9771 -326.4405 -322.4230 -318.2375 -314.2003 -310.0110 -305.6770
## [22] -301.3136 -296.8616 -292.3939 -287.8454 -283.2707 -278.6938 -274.0952
## [29] -269.4900
```

```
min(bwd.summary$bic)
```

```
## [1] -349.8588
```

```
which.min(bwd.summary$bic)
```

```
## [1] 6
```

```
#Comparing Cp
bwd.summary$cp
```

```
##  [1] 670.5296812 178.7195289  57.9982372  19.1742756   9.2873121   4.3776032
##  [7]   2.3819949   1.8155182   0.4697164   1.6267569   1.8137413   2.1259141
## [13]   2.8270987   3.7810165   4.8460677   6.0717682   7.6495227   9.3494254
## [19]  10.9453147  12.6508982  14.4595535  16.2894758  18.1818830  20.0854352
## [25]  22.0457439  24.0244054  26.0046045  28.0000404  30.0000000
```

```
min(bwd.summary$cp)
```

```
## [1] 0.4697164
```

```
which.min(bwd.summary$cp)
```

```
## [1] 9
```

```
coef(regfit.bwd, 6)
```

```
## (Intercept)        rpop        fpov        hisp        awpw       aaepw
##   58.040000    8.628709   26.459580  -24.857370  -45.201744   81.096390
##       retax
##   18.965003
```

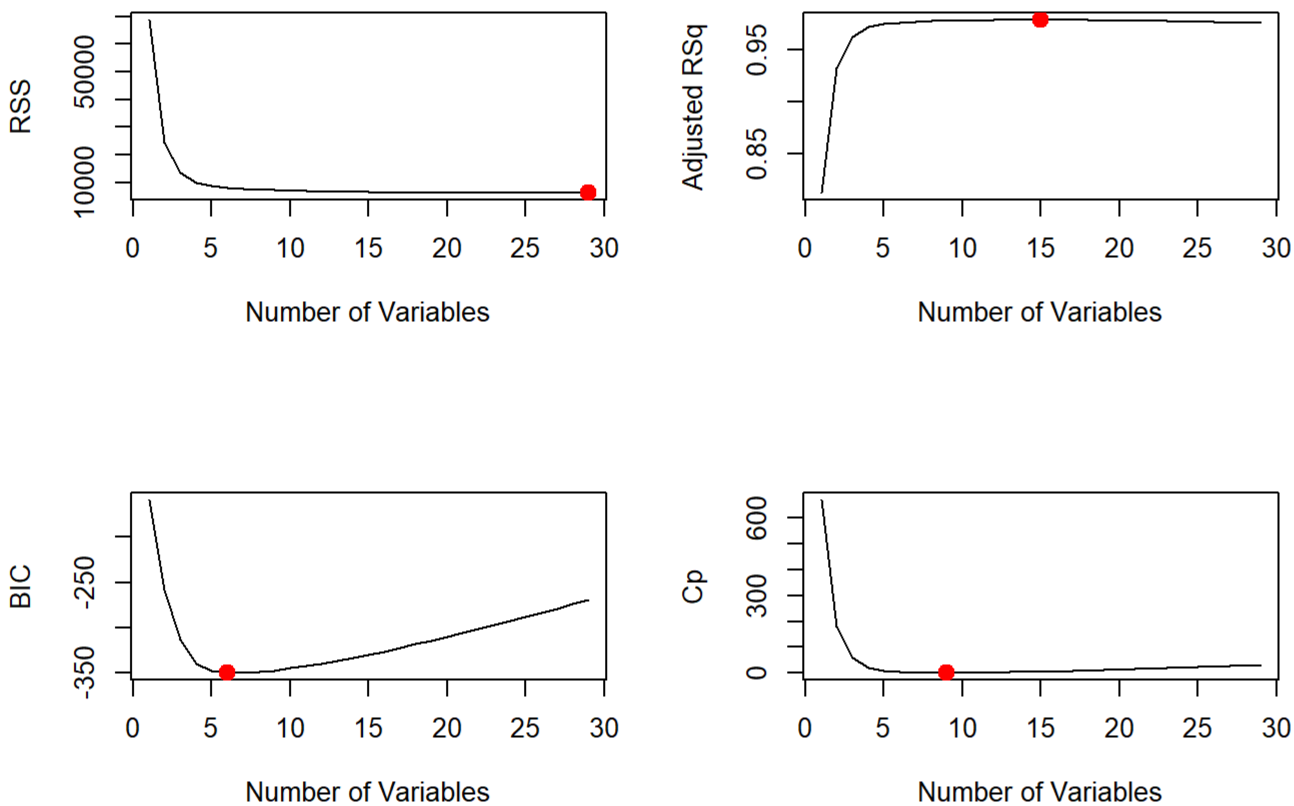
```
coef(regfit.bwd, 9)
```

```
## (Intercept)         aiw        rpop        fpov        hisp       white
##   58.040000    2.727824    8.231274   29.163282  -23.144421  -25.829484
##       black        awpw       aaepw       retax
##  -11.589488  -58.152140  103.755897   39.997657
```

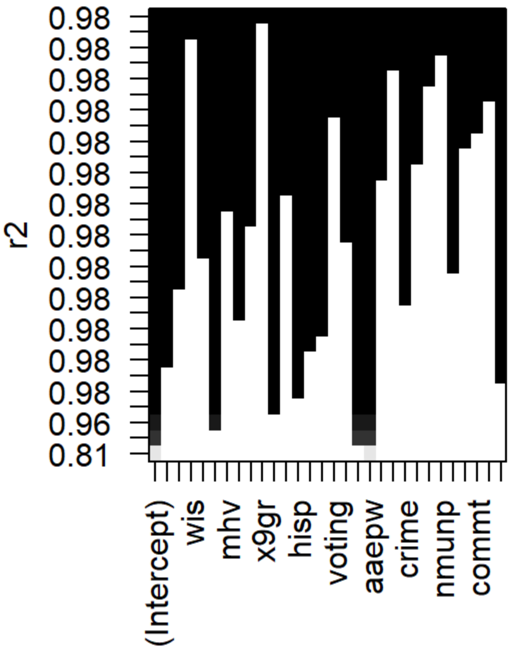
```
coef(regfit.bwd, 15)
```

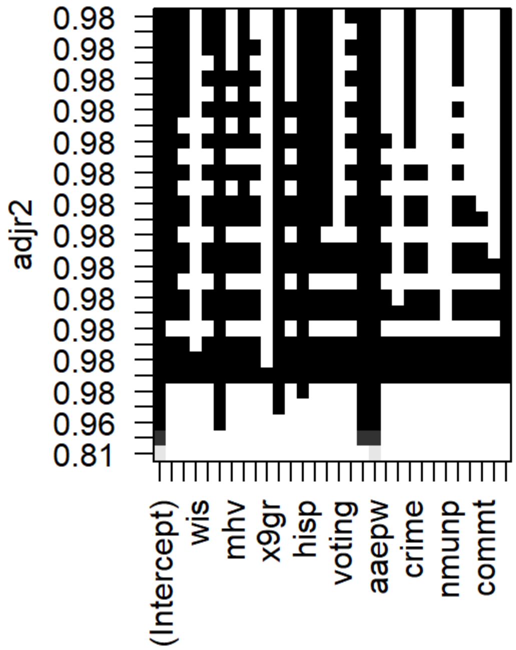
```
## (Intercept)         aiw       pop05        upop        rpop        bach
##   58.040000    2.833348    5.744490   47.256227   10.175278  -34.234257
##        fpov        hisp       white       black         mhi        awpw
##   21.317611  -27.104064  -37.553163  -18.561154    1.437631  -51.413873
##       aaepw       crime       shigh       retax
##  107.265876   -4.438005    4.681464   39.675633
```

```
par(mfrow = c(2, 2))
plot(bwd.summary$rss, xlab = "Number of Variables", ylab = "RSS", type = "l")
points(29, bwd.summary$rss[29], col = "red", cex = 2, pch = 20)
plot(bwd.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
points(15, bwd.summary$adjr2[15], col = "red", cex = 2, pch = 20)
plot(bwd.summary$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")
points(6, bwd.summary$bic[6], col = "red", cex = 2, pch = 20)
plot(bwd.summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
points(9, bwd.summary$cp[9], col = "red", cex = 2, pch = 20)
```

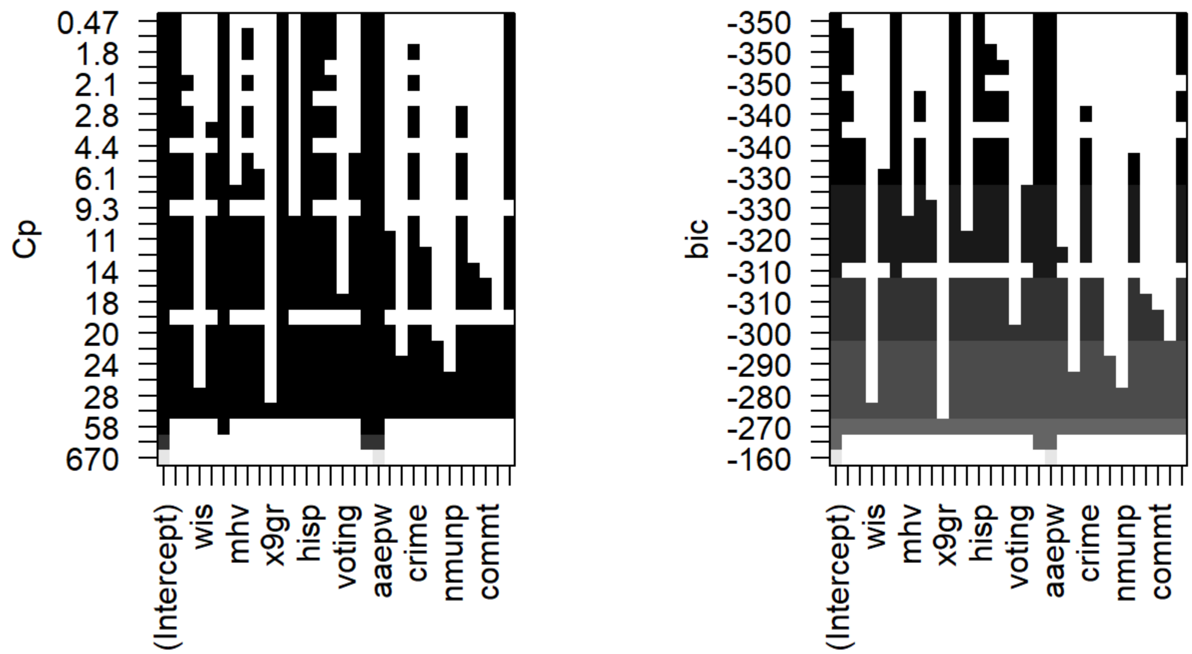



```
par(mfrow = c(1, 2))
plot(regfit.bwd, scale = "r2")
plot(regfit.bwd, scale = "adjr2")
```






```
plot(regfit.bwd, scale = "Cp")
plot(regfit.bwd, scale = "bic")
```



##

Choosing Among Models Using the Validation-Set Approach and Cross-Validation

Validation-Set Approach

Problem: very dependant on what observations are in the training and the test sets.

```
#Splitting the sample
set.seed(6)
train <- sample(c(TRUE, FALSE), nrow(Gas),
replace = TRUE)
test <- (!train)
```

```
regfit.best <- regsubsets(gas ~ .,
data = Gas[train, ], nvmax = 29)
```

```
test.mat <- model.matrix(gas ~ ., data = Gas[test, ])
```

```
val.errors <- rep(NA, 29)

for (i in 1:29) {
  coefi <- coef(regfit.best, id = i)
  pred <- test.mat[, names(coefi)] %*% coefi
  val.errors[i] <- mean((Gas$gas[test] - pred)^2)
}
val.errors
```

```
##  [1]     388.5659     397.4526     196.2208    3499.8788    3744.3575
##  [6]    4184.8812    3638.9586    5415.2362    4378.4629    5394.7007
## [11]   37227.2052   58497.8003   50589.5525  147969.2565   53177.0395
## [16] 2051890.9341 1556438.8959 1732724.4063 2487022.0762 1020688.6573
## [21]  876281.3608  660821.4592  827382.2421  527231.4140  375629.9163
## [26]  243784.6938  254686.4232  241344.3244  213057.5648
```

```
which.min(val.errors)
```

```
## [1] 3
```

```
predict.regsubsets <- function(object, newdata , id, ...) {
form <- as.formula(object$call[[2]])
mat <- model.matrix(form, newdata)
coefi <- coef(object, id = id)
xvars <- names(coefi)
mat[, xvars] %*% coefi
}
```

```
regfit.best <- regsubsets(gas ~ ., data = Gas, nvmax = 29)
coef(regfit.best, 3)
```

```
## (Intercept)        fpov        hisp       retax
##    58.04000    39.10220   -29.03307    50.24728
```

Cross-Validation

```
k <- 10
n <- nrow(Gas)
set.seed(1)
folds <- sample(rep(1:k, length = n))
cv.errors <- matrix(NA, k, 29, dimnames = list(NULL, paste(1:29)))
```

NoW, we write a for loop that perfoms cross-validation for each size.

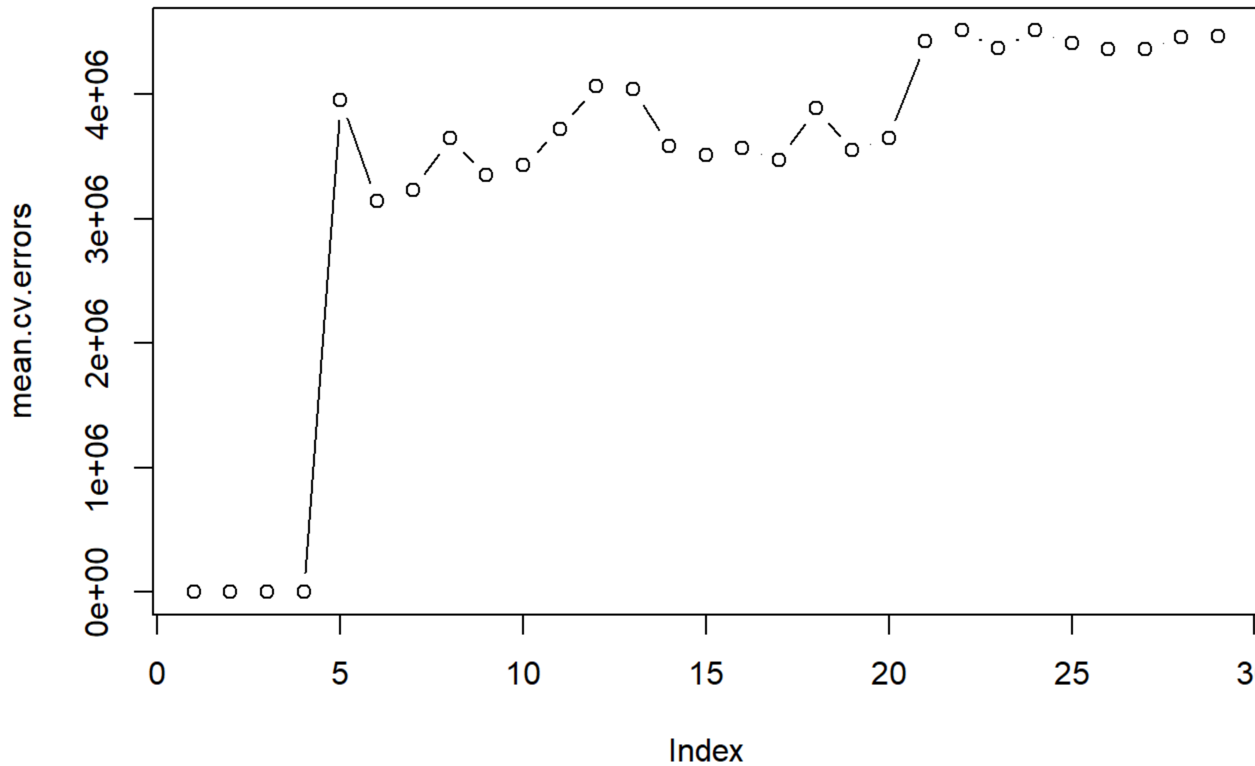
```
for (j in 1:k) {
best.fit <- regsubsets(gas ~ .,
data = Gas[folds != j, ], nvmax = 29)
for (i in 1:29) {
pred <- predict(best.fit, Gas[folds == j, ], id = i)
cv.errors[j, i] <-
mean((Gas$gas[folds == j] - pred)^2)
}
}
```

Averaging over the colums of the matrix in order to obtain a vector for shich the ith element is the cross-validation error for the i-variable model.

```
mean.cv.errors <- apply(cv.errors, 2, mean)
mean.cv.errors
```

```
##            1            2            3            4            5            6
##     310.0271     291.5978     182.2572     141.3547 3949305.1798 3143307.2172
##            7            8            9           10           11           12
## 3232425.4917 3648188.5825 3346572.7159 3427298.8613 3722187.5226 4063666.9713
##           13           14           15           16           17           18
## 4039000.9262 3581578.6305 3513778.8447 3566832.7333 3472277.6675 3887385.7199
##           19           20           21           22           23           24
## 3548949.3863 3643264.4669 4427279.2385 4510755.2470 4367880.9200 4510969.1036
##           25           26           27           28           29
## 4409376.9785 4358594.8232 4362852.4778 4461869.0553 4464166.2779
```

```
par(mfrow = c(1, 1))
plot(mean.cv.errors, type = "b")
```



Performing best subest selection on the full datta set in order to obaain the 4-variable model.

```
reg.best <- regsubsets(gas ~ ., data = Gas,
nvmax = 29)
coef(reg.best, 4)
```

```
## (Intercept)        fpov        awpw       aaepw       shigh
##    58.04000    20.32452   -78.06650   109.48076    10.74001
```

So the best model has 4 variables: fpov, awpw, aaepw, and shigh.

```
reg.best.summary <- summary(reg.best)
reg.best.summary
```

```
## Subset selection object
## Call: regsubsets.formula(gas ~ ., data = Gas, nvmax = 29)
## 29 Variables  (and intercept)
##        Forced in Forced out
## aiw        FALSE      FALSE
## pop05      FALSE      FALSE
## wis        FALSE      FALSE
## upop       FALSE      FALSE
## rpop       FALSE      FALSE
## mhv        FALSE      FALSE
## bach       FALSE      FALSE
## hs         FALSE      FALSE
## x9gr       FALSE      FALSE
## fpov       FALSE      FALSE
## ppov       FALSE      FALSE
## hisp       FALSE      FALSE
## white      FALSE      FALSE
## black      FALSE      FALSE
## voting     FALSE      FALSE
## mhi        FALSE      FALSE
## awpw       FALSE      FALSE
## aaepw      FALSE      FALSE
## unemp      FALSE      FALSE
## capin      FALSE      FALSE
## crime      FALSE      FALSE
## nomuns     FALSE      FALSE
## munp       FALSE      FALSE
## nmunp      FALSE      FALSE
## shigh      FALSE      FALSE
## noveh      FALSE      FALSE
## commt      FALSE      FALSE
## agmort     FALSE      FALSE
## retax      FALSE      FALSE
## 1 subsets of each size up to 29
## Selection Algorithm: exhaustive
##           aiw pop05 wis upop rpop mhv bach hs   x9gr fpov ppov hisp white black
## 1  ( 1 )  " " " "   " " " "  " "  " " " "  " "  " "  " "  "*"  " "  " "   " "  
## 2  ( 1 )  " " " "   " " " "  " "  " " " "  " "  " "  " "  " "  " "  " "   " "  
## 3  ( 1 )  " " " "   " " " "  " "  " " " "  " "  " "  "*"  " "  "*"  " "   " "  
## 4  ( 1 )  " " " "   " " " "  " "  " " " "  " "  " "  "*"  " "  " "  " "   " "  
## 5  ( 1 )  " " " "   " " " "  " "  " " " "  " "  " "  "*"  " "  "*"  " "   " "  
## 6  ( 1 )  "*" " "   " " "*"  " "  " " " "  " "  " "  "*"  " "  " "  " "   " "  
## 7  ( 1 )  "*" " "   " " "*"  " "  " " " "  " "  " "  "*"  " "  "*"  " "   " "  
## 8  ( 1 )  "*" " "   " " "*"  " "  " " " "  " "  " "  "*"  " "  "*"  " "   " "  
## 9  ( 1 )  "*" " "   " " "*"  " "  " " " "  " "  " "  "*"  " "  "*"  " "   " "  
## 10  ( 1 ) "*" " "   " " "*"  " "  " " " "  " "  " "  "*"  " "  "*"  " "   " "  
## 11  ( 1 ) "*" " "   " " "*"  " "  " " " "  " "  " "  "*"  " "  "*"  " "   " "  
## 12  ( 1 ) "*" " "   " " " "  "*"  " " " "  " "  " "  "*"  " "  "*"  "*"   "*"  
## 13  ( 1 ) "*" " "   " " " "  "*"  "*" " "  " "  " "  "*"  " "  "*"  "*"   "*"  
## 14  ( 1 ) "*" "*"   " " " "  "*"  "*" " "  " "  " "  "*"  " "  "*"  "*"   "*"  
## 15  ( 1 ) "*" "*"   " " "*"  "*"  " " "*"  " "  " "  "*"  " "  "*"  "*"   "*"  
## 16  ( 1 ) "*" "*"   " " "*"  "*"  " " "*"  "*"  " "  "*"  " "  "*"  "*"   "*"  
```

```
## 17  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  " "  "*"  "*"   "*"
## 18  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 19  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 20  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 21  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 22  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 23  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 24  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 25  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 26  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  " " "*"  "*"  "*"  "*"  "*"   "*"
## 27  ( 1 ) "*" "*"   " " "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 28  ( 1 ) "*" "*"   "*" "*"  "*"  "*" "*"  "*" " "  "*"  "*"  "*"  "*"   "*"
## 29  ( 1 ) "*" "*"   "*" "*"  "*"  "*" "*"  "*" "*"  "*"  "*"  "*"  "*"   "*"
##           voting mhi awpw aaepw unemp capin crime nomuns munp nmunp shigh noveh
## 1  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   " "   " "
## 2  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   " "   " "
## 3  ( 1 )  " "    " " " "  " "   " "   " "   " "   " "    " "  " "   " "   " "
## 4  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 5  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 6  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 7  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   " "   " "
## 8  ( 1 )  " "    " " "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 9  ( 1 )  " "    "*" "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 10  ( 1 ) " "    "*" "*"  "*"   " "   "*"   " "   " "    " "  " "   "*"   " "
## 11  ( 1 ) " "    "*" "*"  "*"   " "   "*"   " "   " "    " "  " "   "*"   " "
## 12  ( 1 ) " "    "*" "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 13  ( 1 ) " "    "*" "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 14  ( 1 ) " "    "*" "*"  "*"   " "   " "   " "   " "    " "  " "   "*"   " "
## 15  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 16  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 17  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 18  ( 1 ) " "    "*" "*"  "*"   " "   " "   "*"   " "    " "  " "   "*"   " "
## 19  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   " "    " "  " "   "*"   " "
## 20  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   " "
## 21  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   " "
## 22  ( 1 ) " "    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 23  ( 1 ) "*"    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 24  ( 1 ) "*"    "*" "*"  "*"   "*"   " "   "*"   "*"    " "  " "   "*"   "*"
## 25  ( 1 ) "*"    "*" "*"  "*"   "*"   " "   "*"   "*"    "*"  " "   "*"   "*"
## 26  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  " "   "*"   "*"
## 27  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
## 28  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
## 29  ( 1 ) "*"    "*" "*"  "*"   "*"   "*"   "*"   "*"    "*"  "*"   "*"   "*"
##           commt agmort retax
## 1  ( 1 )  " "   " "    " "
## 2  ( 1 )  " "   "*"    "*"
## 3  ( 1 )  " "   " "    "*"
## 4  ( 1 )  " "   " "    " "
## 5  ( 1 )  " "   " "    " "
## 6  ( 1 )  " "   " "    "*"
## 7  ( 1 )  " "   " "    "*"
## 8  ( 1 )  " "   " "    "*"
```

```
## 9  ( 1 )  " "   " "    "*"
## 10  ( 1 ) " "   " "    "*"
## 11  ( 1 ) " "   "*"    "*"
## 12  ( 1 ) " "   "*"    "*"
## 13  ( 1 ) " "   "*"    "*"
## 14  ( 1 ) " "   "*"    "*"
## 15  ( 1 ) " "   " "    "*"
## 16  ( 1 ) " "   " "    "*"
## 17  ( 1 ) " "   " "    "*"
## 18  ( 1 ) " "   " "    "*"
## 19  ( 1 ) " "   " "    "*"
## 20  ( 1 ) " "   " "    "*"
## 21  ( 1 ) " "   "*"    "*"
## 22  ( 1 ) "*"   " "    "*"
## 23  ( 1 ) "*"   " "    "*"
## 24  ( 1 ) "*"   "*"    "*"
## 25  ( 1 ) "*"   "*"    "*"
## 26  ( 1 ) "*"   "*"    "*"
## 27  ( 1 ) "*"   "*"    "*"
## 28  ( 1 ) "*"   "*"    "*"
## 29  ( 1 ) "*"   "*"    "*"
```

```
reg.best.summary$rss[4]
```

```
## [1] 9120.549
```

```
reg.best.summary$rsq[4]
```

```
## [1] 0.9753697
```

```
reg.best.summary$adjr2[4]
```

```
## [1] 0.9743326
```

```
reg.best.summary$bic[4]
```

```
## [1] -347.3519
```

```
reg.best.summary$cp[4]
```

```
## [1] 11.61275
```