Mini-Project 3

STAT-340 Applied Regression Methods

Yerim Oh

Details

Due Date

This assignment is due at 11:59 PM on the 4th of November.

Grading

The problems will be graded for correctness. An emphasis will be placed on full explanations of your thought process. You usually won't need to write more than a few sentences for any given problem, but you should write complete sentences! Understanding and explaining the reasons behind your decisions is more important than making the "correct" decision.

Collaboration

You are allowed to work with others on this assignment, but you must complete and submit your own write up. You should not copy large blocks of code or written text from another student.

Load Packages

The following R code loads packages needed in this assignment.

Problem 1

The data used in this problem were found on the website "data.world". They mention that the data were aggregated from a number of sources including the American Community Survey (census.gov), clinicaltrials.gov, and cancer.gov. The goal of this exercise is to build an Ordinary Least Squares multiple regression model to predict cancer mortality rates (variable `target_deathrate`) by United States counties. Each row in the dataset represents a county and those with missing data are excluded from the analysis.

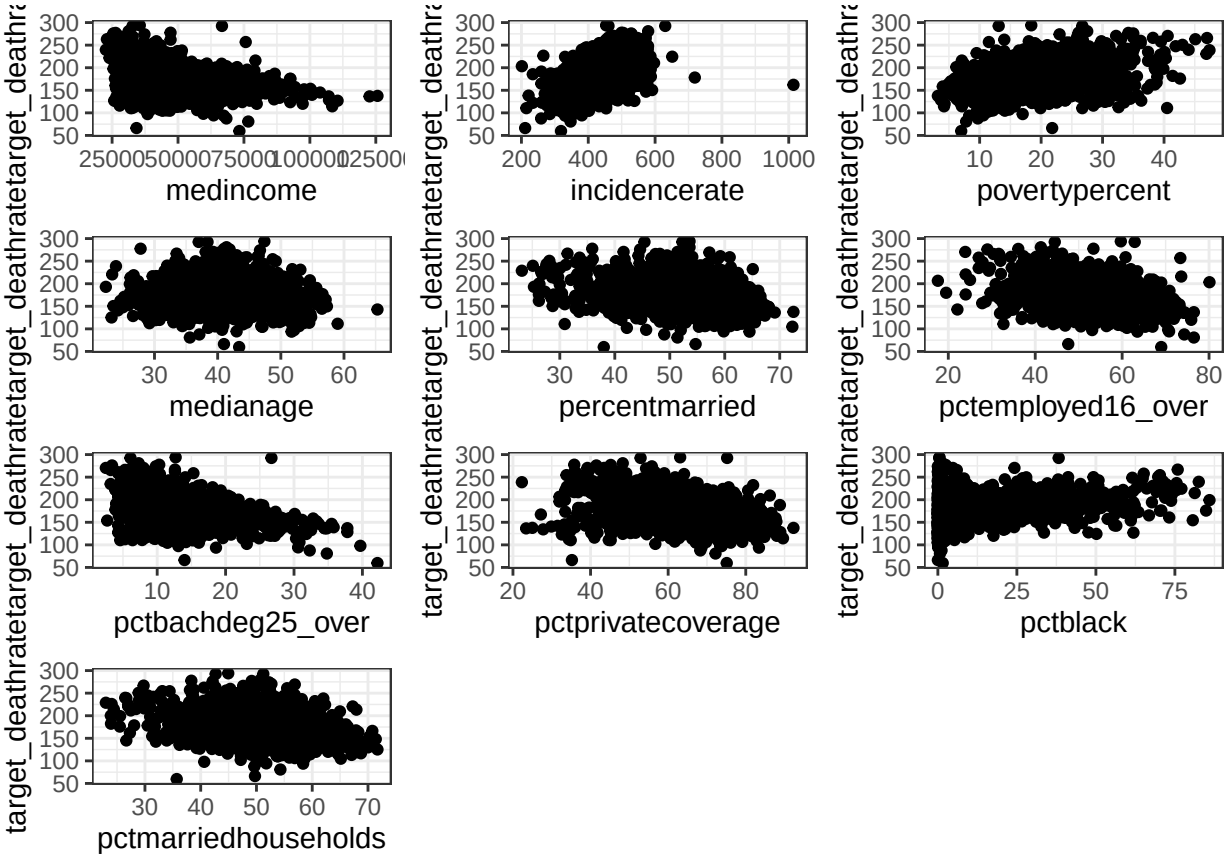
The original file contains many variables, including the name of the county. However, we will use the following variables to conduct the analysis:

- response variable: target_deathrate
- potential explanatory variables: medincome, incidencerate, povertypercent, medianage, percentmarried, pctemployed16_over, pctbachdeg25_over, pctprivatecoverage, pctblack, pctmarriedhouseholds

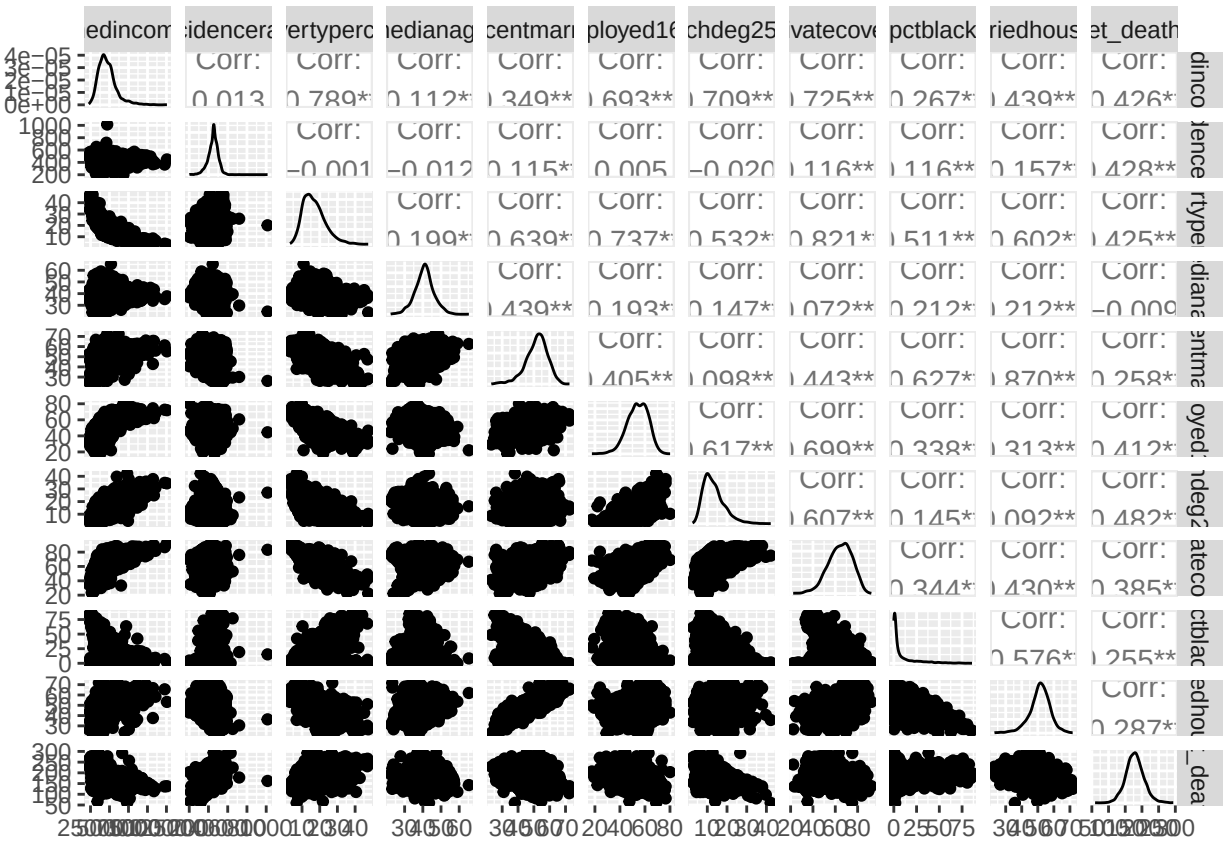
a) Make a pair plot all the variables.

```
p1 <- ggplot(cancer, aes(y = target_deathrate, x = medincome)) + geom_point() + theme_bw()
p2 <- ggplot(cancer, aes(y = target_deathrate, x = incidencerate)) + geom_point() + theme_bw()
p3 <- ggplot(cancer, aes(y = target_deathrate, x = povertypercent)) + geom_point() + theme_bw()
p4 <- ggplot(cancer, aes(y = target_deathrate, x = medianage)) + geom_point() + theme_bw()
p5 <- ggplot(cancer, aes(y = target_deathrate, x = percentmarried)) + geom_point() + theme_bw()
p6 <- ggplot(cancer, aes(y = target_deathrate, x = pctemployed16_over)) + geom_point() + theme_bw()
p7 <- ggplot(cancer, aes(y = target_deathrate, x = pctbachdeg25_over)) + geom_point() + theme_bw()
p8 <- ggplot(cancer, aes(y = target_deathrate, x = pctprivatecoverage)) + geom_point() + theme_bw()
p9 <- ggplot(cancer, aes(y = target_deathrate, x = pctblack)) + geom_point() + theme_bw()
p10 <- ggplot(cancer, aes(y = target_deathrate, x = pctmarriedhouseholds)) + geom_point() + theme_bw()

grid.arrange(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,
             ncol = 3)
```



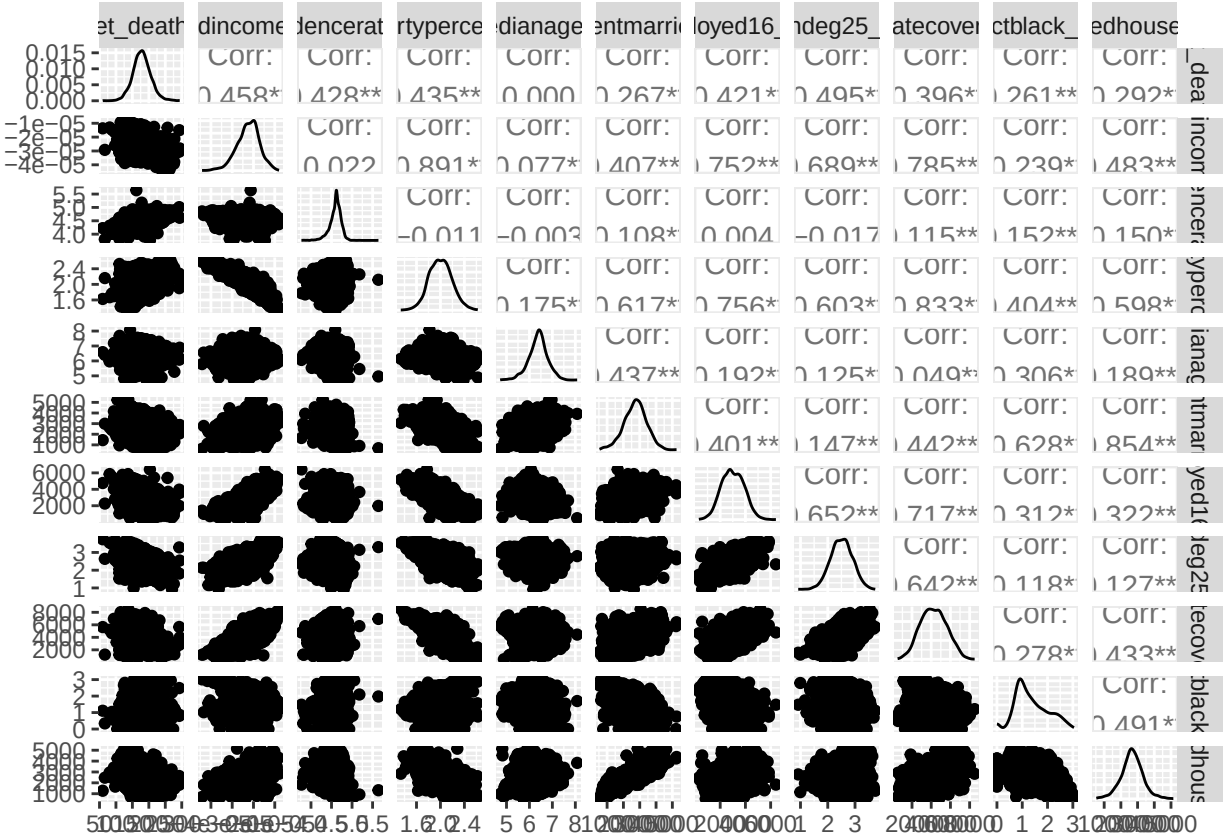

```
ggpairs(cancer)
```


edincom
idencera
ertyperc
edianag
centmar
ployed16
chdeg25
vatecove
pctblack
riedhous
et_death
Corr:
0.013
0.789*
0.112*
0.349**
0.693**
0.709**
0.725**
0.267*
0.439**
0.426*
−0.001
−0.012
0.115*
0.005
−0.020
0.116**
0.116**
0.157*
0.428**
0.199*
0.639*
0.737*
0.532*
0.821*
0.511**
0.602*
0.425**
0.439**
0.193*
0.147*
0.072**
0.212*
0.212**
−0.009
0.405**
0.098**
0.443**
0.627*
0.870**
0.258*
0.617**
0.699**
0.338*
0.313**
0.412*
0.607**
0.145*
0.092**
0.482*
0.344*
0.430**
0.385*
0.576*
0.255**
0.287*

b) Develop a predictive model for the response variable `target_deathrate` by transforming the response and/or quantitative explanatory variables so that the associations between the transformed variables are approximately linear and the variance is approximately constant.

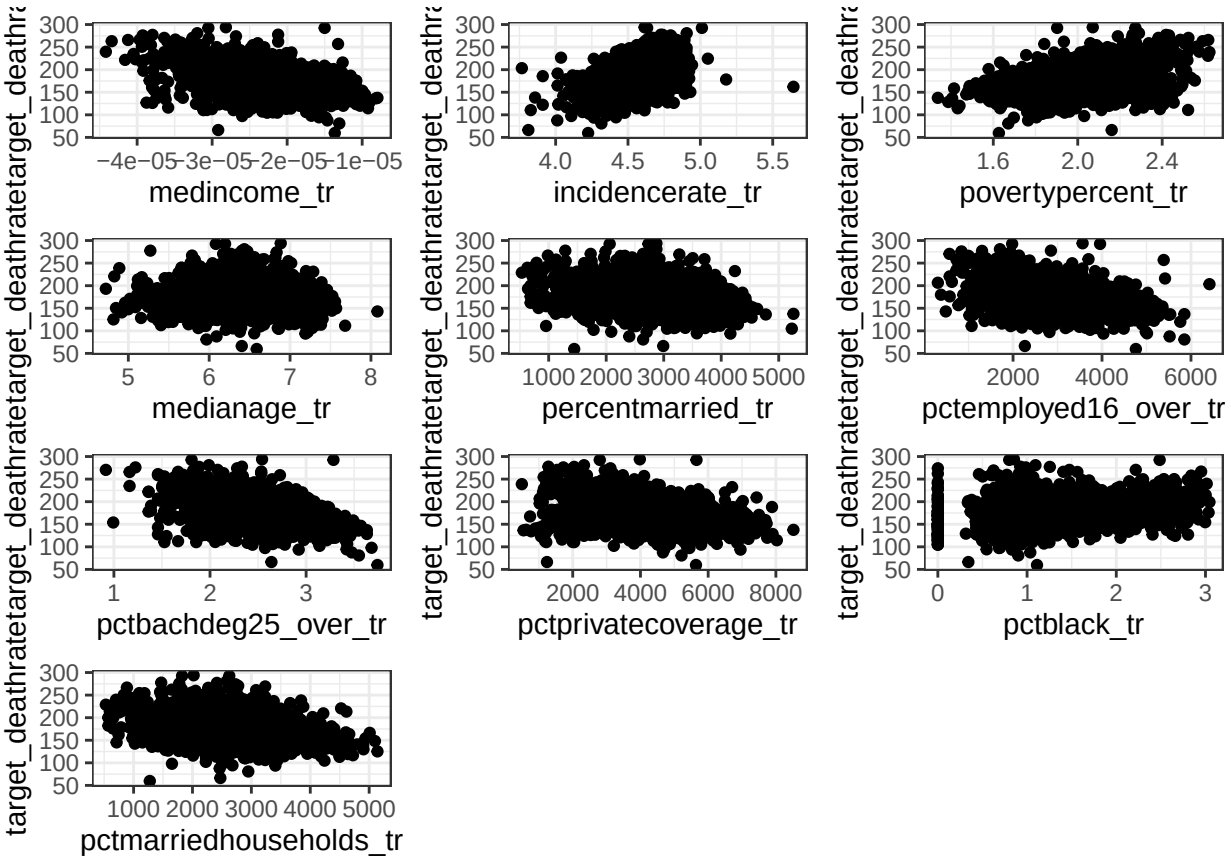
```
cancer_tr <- cancer %>% mutate(medincome_tr = -1/(medincome),
                               incidencerate_tr = (incidencerate)^0.25,
                               povertypercent_tr = (povertypercent)^0.25,
                               medianage_tr = sqrt(medianage),
                               percentmarried_tr = (percentmarried)^2,
                               pctemployed16_over_tr = (pctemployed16_over)^2,
                               pctbachdeg25_over_tr = log(pctbachdeg25_over),
                               pctprivatecoverage_tr = (pctprivatecoverage)^2,
                               pctblack_tr = (pctblack)^0.25,
                               pctmarriedhouseholds_tr = (pctmarriedhouseholds)^2)
cancer_tr <- cancer_tr %>% select(!c(medincome,
                                     incidencerate,
                                     povertypercent,
                                     medianage,
                                     percentmarried,
                                     pctemployed16_over,
                                     pctbachdeg25_over,
                                     pctprivatecoverage,
                                     pctblack,
                                     pctmarriedhouseholds))
```

```
ggpairs(cancer_tr)
```



```
tr1 <- ggplot(cancer_tr, aes(y = target_deathrate, x = medincome_tr)) + geom_point() + theme_bw()
tr2 <- ggplot(cancer_tr, aes(y = target_deathrate, x = incidencerate_tr)) + geom_point() + theme_bw()
tr3 <- ggplot(cancer_tr, aes(y = target_deathrate, x = povertypercent_tr)) + geom_point() + theme_bw()
tr4 <- ggplot(cancer_tr, aes(y = target_deathrate, x = medianage_tr)) + geom_point() + theme_bw()
tr5 <- ggplot(cancer_tr, aes(y = target_deathrate, x = percentmarried_tr)) + geom_point() + theme_bw()
tr6 <- ggplot(cancer_tr, aes(y = target_deathrate, x = pctemployed16_over_tr)) + geom_point() + theme_bw
tr7 <- ggplot(cancer_tr, aes(y = target_deathrate, x = pctbachdeg25_over_tr)) + geom_point() + theme_bw
tr8 <- ggplot(cancer_tr, aes(y = target_deathrate, x = pctprivatecoverage_tr)) + geom_point() + theme_bw
tr9 <- ggplot(cancer_tr, aes(y = target_deathrate, x = pctblack_tr)) + geom_point() + theme_bw()
tr10 <- ggplot(cancer_tr, aes(y = target_deathrate, x = pctmarriedhouseholds_tr)) + geom_point() + theme

grid.arrange(tr1,tr2,tr3,tr4,tr5,tr6,tr7,tr8,tr9,tr10,
             ncol = 3)
```



c) Perform best subset selection in order to choose the best model. What is the best model obtained according to Cp and BIC? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained.

```
regfit.full <- regsubsets(target_deathrate ~., data = cancer_tr, nvmax = 10)
```

```
summary(regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(target_deathrate ~ ., data = cancer_tr, nvmax = 10)
## 10 Variables  (and intercept)
##                          Forced in Forced out
## medincome_tr                 FALSE      FALSE
## incidencerate_tr             FALSE      FALSE
## povertypercent_tr            FALSE      FALSE
## medianage_tr                 FALSE      FALSE
## percentmarried_tr            FALSE      FALSE
## pctemployed16_over_tr        FALSE      FALSE
## pctbachdeg25_over_tr         FALSE      FALSE
## pctprivatecoverage_tr        FALSE      FALSE
## pctblack_tr                  FALSE      FALSE
## pctmarriedhouseholds_tr      FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: exhaustive
##           medincome_tr incidencerate_tr povertypercent_tr medianage_tr
## 1  ( 1 )  " "          " "              " "                " "
## 2  ( 1 )  " "          "*"              " "                " "
## 3  ( 1 )  " "          "*"              "*"                " "
## 4  ( 1 )  "*"          "*"              " "                " "
## 5  ( 1 )  "*"          "*"              " "                " "
## 6  ( 1 )  "*"          "*"              " "                " "
## 7  ( 1 )  "*"          "*"              "*"                " "
## 8  ( 1 )  "*"          "*"              "*"                "*"
## 9  ( 1 )  "*"          "*"              " "                "*"
## 10  ( 1 ) "*"          "*"              "*"                "*"
##           percentmarried_tr pctemployed16_over_tr pctbachdeg25_over_tr
## 1  ( 1 )  " "               " "                   "*"
## 2  ( 1 )  " "               " "                   "*"
## 3  ( 1 )  " "               " "                   "*"
## 4  ( 1 )  " "               " "                   "*"
## 5  ( 1 )  " "               " "                   "*"
## 6  ( 1 )  " "               " "                   "*"
## 7  ( 1 )  " "               " "                   "*"
## 8  ( 1 )  " "               " "                   "*"
## 9  ( 1 )  "*"               "*"                   "*"
## 10  ( 1 ) "*"               "*"                   "*"
##           pctprivatecoverage_tr pctblack_tr pctmarriedhouseholds_tr
## 1  ( 1 )  " "                   " "         " "
## 2  ( 1 )  " "                   " "         " "
## 3  ( 1 )  " "                   " "         " "
## 4  ( 1 )  " "                   "*"         " "
## 5  ( 1 )  "*"                   "*"         " "
## 6  ( 1 )  "*"                   "*"         "*"
## 7  ( 1 )  "*"                   "*"         "*"
```

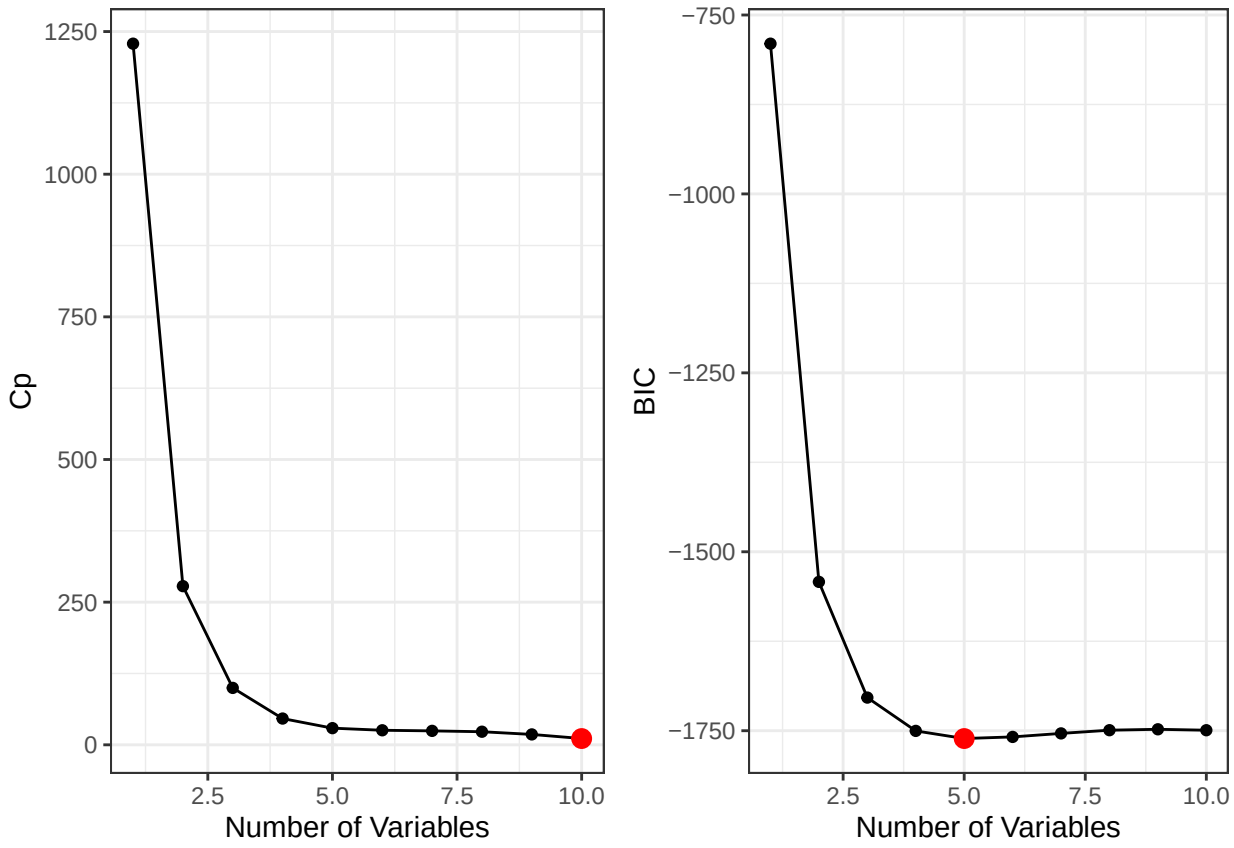
```
## 8  ( 1 )  "*"                  "*"          "*"
## 9  ( 1 )  "*"                  "*"          "*"
## 10  ( 1 ) "*"                  "*"          "*"
```

```
# create a data frame with Cp and BIC
results.full <- data.frame(numb.var = 1:length(summary(regfit.full)$cp),
                           Cp = summary(regfit.full)$cp,
                           BIC = summary(regfit.full)$bic)
results.full
```

Plot the results

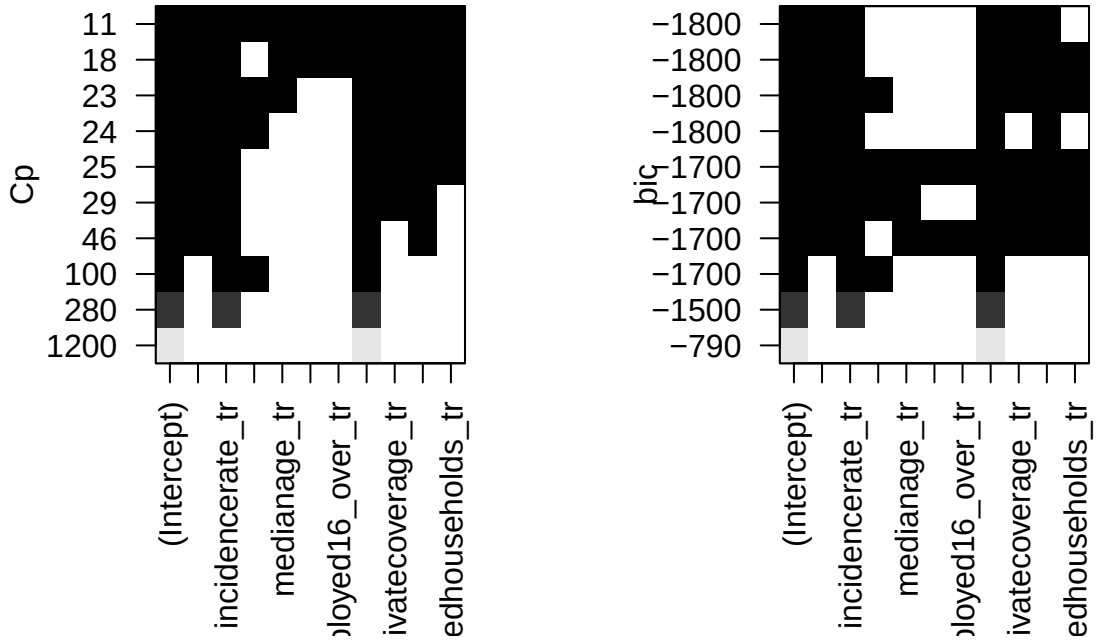
```
##    numb.var         Cp        BIC
## 1         1 1228.72597  -790.0893
## 2         2  278.07692 -1542.0445
## 3         3   99.72608 -1703.7270
## 4         4   46.01549 -1750.2642
## 5         5   29.20656 -1760.9389
## 6         6   25.48142 -1758.6742
## 7         7   24.42276 -1753.7609
## 8         8   22.93072 -1749.2834
## 9         9   18.21137 -1748.0377
## 10       10   11.00000 -1749.3083
```

```
# plot the results
f1 <- ggplot(results.full, aes(x = numb.var, y = Cp)) +
  geom_point() + geom_line() + theme_bw() +
  xlab("Number of Variables") + ylab("Cp") +
  geom_point(data = results.full[which.min(summary(regfit.full)$cp),],
             aes(y = Cp, x = numb.var),
             size = 3, col = "red")
f2 <- ggplot(results.full, aes(x = numb.var, y = BIC)) +
  geom_point() + geom_line() + theme_bw() +
  xlab("Number of Variables") + ylab("BIC") +
  geom_point(data = results.full[which.min(summary(regfit.full)$bic),],
             aes(y = BIC, x = numb.var),
             size = 3, col = "red")
grid.arrange(f1, f2, ncol = 2)
```




```
par(mfrow = c(1,2))
plot(regfit.full, scale = "Cp")
plot(regfit.full, scale = "bic")
```

Variable selection plots



Get coefficients of "Best models" Based on Cp

```
# The best model based on Cp is the one with 10 variables
coef(regfit.full, 10)
```

```
##             (Intercept)            medincome_tr        incidencerate_tr
##           -6.089763e+01           -9.200160e+05            8.100942e+01
##       povertypercent_tr            medianage_tr       percentmarried_tr
##           -2.091645e+01           -6.063544e+00            5.013645e-03
##   pctemployed16_over_tr    pctbachdeg25_over_tr   pctprivatecoverage_tr
##           -3.406660e-03           -2.181231e+01           -2.138683e-03
##             pctblack_tr pctmarriedhouseholds_tr
##            3.811772e+00           -6.649184e-03
```

Based on BIC

```
# The best model based on BIC is the one with 5 variables
coef(regfit.full, 5)
```

```
##           (Intercept)          medincome_tr      incidencerate_tr
##         -1.647427e+02         -8.142235e+05          8.293151e+01
##  pctbachdeg25_over_tr pctprivatecoverage_tr           pctblack_tr
##         -2.105662e+01         -2.080918e-03          4.130221e+00
```

d) Repeat c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results in c)?

Forward Stepwise Selection

```
regfit.fwd <- regsubsets(target_deathrate ~ ., cancer_tr, nvmax = 10, method = "forward")
```

```
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(target_deathrate ~ ., cancer_tr, nvmax = 10,
##     method = "forward")
## 10 Variables  (and intercept)
##                          Forced in Forced out
## medincome_tr                 FALSE      FALSE
## incidencerate_tr             FALSE      FALSE
## povertypercent_tr            FALSE      FALSE
## medianage_tr                 FALSE      FALSE
## percentmarried_tr            FALSE      FALSE
## pctemployed16_over_tr        FALSE      FALSE
## pctbachdeg25_over_tr         FALSE      FALSE
## pctprivatecoverage_tr        FALSE      FALSE
## pctblack_tr                  FALSE      FALSE
## pctmarriedhouseholds_tr      FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: forward
##           medincome_tr incidencerate_tr povertypercent_tr medianage_tr
## 1  ( 1 )  " "          " "              " "               " "
## 2  ( 1 )  " "          "*"              " "               " "
## 3  ( 1 )  " "          "*"              "*"               " "
## 4  ( 1 )  " "          "*"              "*"               " "
## 5  ( 1 )  "*"          "*"              "*"               " "
```

```
## 6  ( 1 )  "*"          "*"              "*"               " "
## 7  ( 1 )  "*"          "*"              "*"               " "
## 8  ( 1 )  "*"          "*"              "*"               "*"
## 9  ( 1 )  "*"          "*"              "*"               "*"
## 10  ( 1 ) "*"          "*"              "*"               "*"
##           percentmarried_tr pctemployed16_over_tr pctbachdeg25_over_tr
## 1  ( 1 )  " "               " "                   "*"
## 2  ( 1 )  " "               " "                   "*"
## 3  ( 1 )  " "               " "                   "*"
## 4  ( 1 )  " "               " "                   "*"
## 5  ( 1 )  " "               " "                   "*"
## 6  ( 1 )  " "               " "                   "*"
## 7  ( 1 )  " "               " "                   "*"
## 8  ( 1 )  " "               " "                   "*"
## 9  ( 1 )  " "               "*"                   "*"
## 10  ( 1 ) "*"               "*"                   "*"
##           pctprivatecoverage_tr pctblack_tr pctmarriedhouseholds_tr
## 1  ( 1 )  " "                   " "         " "
## 2  ( 1 )  " "                   " "         " "
## 3  ( 1 )  " "                   " "         " "
## 4  ( 1 )  " "                   "*"         " "
## 5  ( 1 )  " "                   "*"         " "
## 6  ( 1 )  "*"                   "*"         " "
## 7  ( 1 )  "*"                   "*"         "*"
## 8  ( 1 )  "*"                   "*"         "*"
## 9  ( 1 )  "*"                   "*"         "*"
## 10  ( 1 ) "*"                   "*"         "*"
```

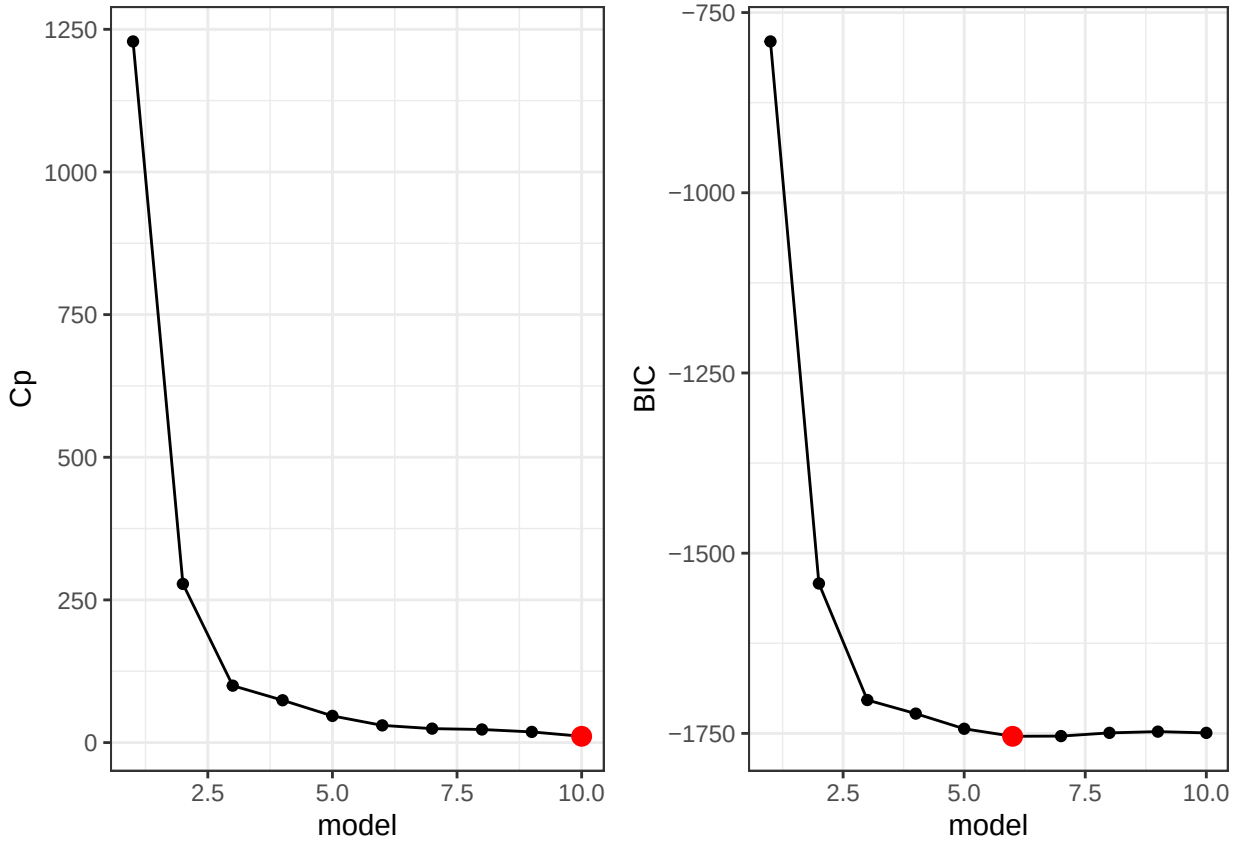
```
# create a data frame with Cp and BIC
results.fwd <- data.frame(model = seq(1,10),
                          Cp = summary(regfit.fwd)$cp,
                          BIC = summary(regfit.fwd)$bic)
results.fwd
```

Plot the results

```
##    model         Cp        BIC
## 1      1 1228.72597  -790.0893
## 2      2  278.07692 -1542.0445
## 3      3   99.72608 -1703.7270
## 4      4   74.18268 -1722.5804
## 5      5   46.71794 -1743.5846
## 6      6   30.14345 -1754.0345
## 7      7   24.42276 -1753.7609
## 8      8   22.93072 -1749.2834
## 9      9   18.69180 -1747.5571
## 10    10   11.00000 -1749.3083
```

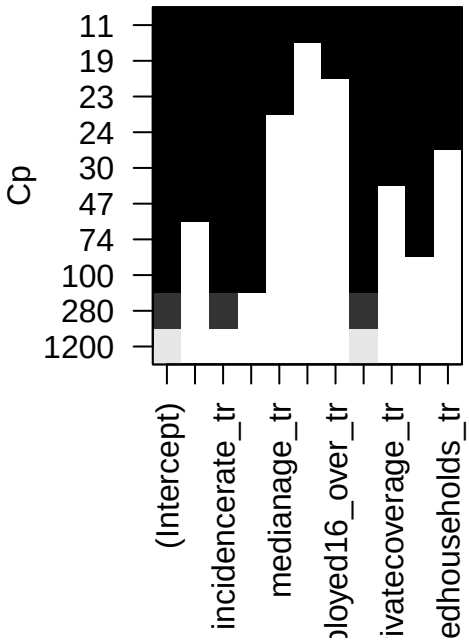
```
# plot the results
fw1 <- ggplot(results.fwd, aes(x = model, y = Cp)) +
  geom_point() + geom_line() + theme_bw() +
  geom_point(data = results.fwd[which.min(summary(regfit.fwd)$cp),],
             aes(y = Cp, x = model),
             size = 3, col = "red")
```

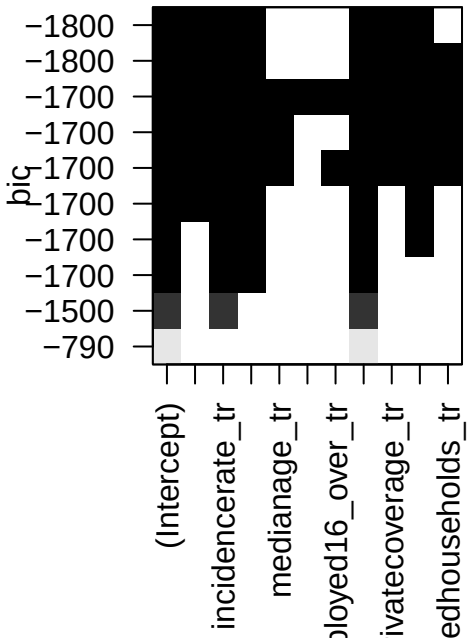
```
fw2 <- ggplot(results.fwd, aes(x = model, y = BIC)) +
  geom_point() + geom_line() + theme_bw() +
  geom_point(data = results.fwd[which.min(summary(regfit.fwd)$bic),],
             aes(y = BIC, x = model),
             size = 3, col = "red")
grid.arrange(fw1, fw2, ncol = 2)
```




```
par(mfrow = c(1,2))
plot(regfit.fwd, scale = "Cp")
plot(regfit.fwd, scale = "bic")
```

Variable selection plots





Get coefficients of "Best models" Based on Cp

```
# The best model based on Cp is the one with 10 variables
coef(regfit.fwd, 10)
```

```
##             (Intercept)            medincome_tr        incidencerate_tr
##           -6.089763e+01           -9.200160e+05            8.100942e+01
##       povertypercent_tr            medianage_tr       percentmarried_tr
##           -2.091645e+01           -6.063544e+00            5.013645e-03
##   pctemployed16_over_tr    pctbachdeg25_over_tr   pctprivatecoverage_tr
##           -3.406660e-03           -2.181231e+01           -2.138683e-03
##             pctblack_tr pctmarriedhouseholds_tr
##            3.811772e+00           -6.649184e-03
```

Based on BIC

```
# The best model based on BIC is the one with 6 variables
coef(regfit.fwd, 6)
```

```
##           (Intercept)          medincome_tr      incidencerate_tr
##         -1.567845e+02         -9.451259e+05          8.306869e+01
##     povertypercent_tr  pctbachdeg25_over_tr pctprivatecoverage_tr
##         -5.628873e+00         -2.086490e+01         -2.345944e-03
##           pctblack_tr
##          4.404680e+00
```

Backward Stepwise Selection

```
regfit.bwd <- regsubsets(target_deathrate ~ ., cancer_tr, nvmax = 10, method = "backward")
```

```
summary(regfit.bwd)
```

```
## Subset selection object
## Call: regsubsets.formula(target_deathrate ~ ., cancer_tr, nvmax = 10,
##     method = "backward")
```

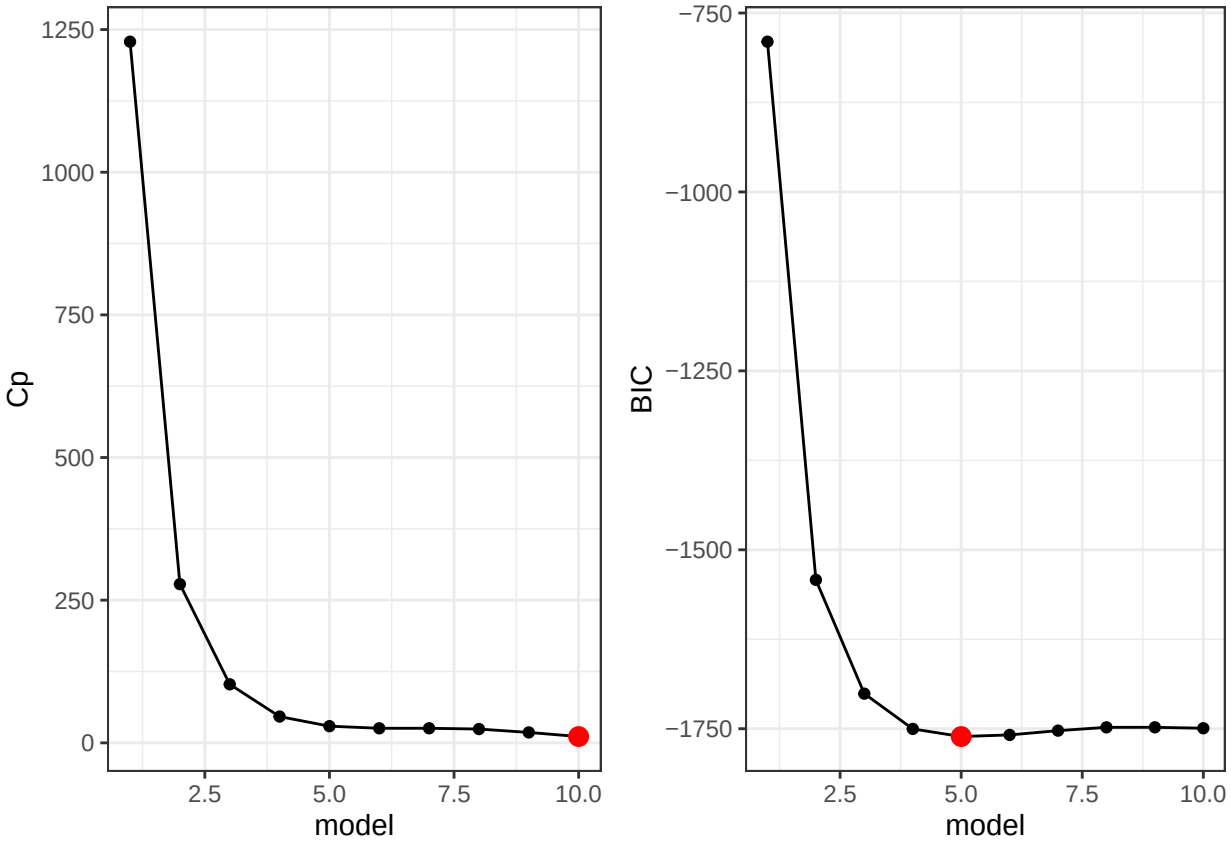
```
## 10 Variables  (and intercept)
##                          Forced in Forced out
## medincome_tr                 FALSE      FALSE
## incidencerate_tr             FALSE      FALSE
## povertypercent_tr            FALSE      FALSE
## medianage_tr                 FALSE      FALSE
## percentmarried_tr            FALSE      FALSE
## pctemployed16_over_tr        FALSE      FALSE
## pctbachdeg25_over_tr         FALSE      FALSE
## pctprivatecoverage_tr        FALSE      FALSE
## pctblack_tr                  FALSE      FALSE
## pctmarriedhouseholds_tr      FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: backward
##           medincome_tr incidencerate_tr povertypercent_tr medianage_tr
## 1  ( 1 )  " "          " "              " "               " "
## 2  ( 1 )  " "          "*"              " "               " "
## 3  ( 1 )  "*"          "*"              " "               " "
## 4  ( 1 )  "*"          "*"              " "               " "
## 5  ( 1 )  "*"          "*"              " "               " "
## 6  ( 1 )  "*"          "*"              " "               " "
## 7  ( 1 )  "*"          "*"              " "               " "
## 8  ( 1 )  "*"          "*"              " "               "*"
## 9  ( 1 )  "*"          "*"              " "               "*"
## 10  ( 1 ) "*"          "*"              "*"               "*"
##           percentmarried_tr pctemployed16_over_tr pctbachdeg25_over_tr
## 1  ( 1 )  " "               " "                   "*"
## 2  ( 1 )  " "               " "                   "*"
## 3  ( 1 )  " "               " "                   "*"
## 4  ( 1 )  " "               " "                   "*"
## 5  ( 1 )  " "               " "                   "*"
## 6  ( 1 )  " "               " "                   "*"
## 7  ( 1 )  "*"               " "                   "*"
## 8  ( 1 )  "*"               " "                   "*"
## 9  ( 1 )  "*"               "*"                   "*"
## 10  ( 1 ) "*"               "*"                   "*"
##           pctprivatecoverage_tr pctblack_tr pctmarriedhouseholds_tr
## 1  ( 1 )  " "                   " "         " "
## 2  ( 1 )  " "                   " "         " "
## 3  ( 1 )  " "                   " "         " "
## 4  ( 1 )  " "                   "*"         " "
## 5  ( 1 )  "*"                   "*"         " "
## 6  ( 1 )  "*"                   "*"         "*"
## 7  ( 1 )  "*"                   "*"         "*"
## 8  ( 1 )  "*"                   "*"         "*"
## 9  ( 1 )  "*"                   "*"         "*"
## 10  ( 1 ) "*"                   "*"         "*"
```

```r
# create a data frame with Cp and BIC
results.bwd <- data.frame(model = seq(1,10),
                          Cp = summary(regfit.bwd)$cp,
                          BIC = summary(regfit.bwd)$bic)
results.bwd
```

Plot the results

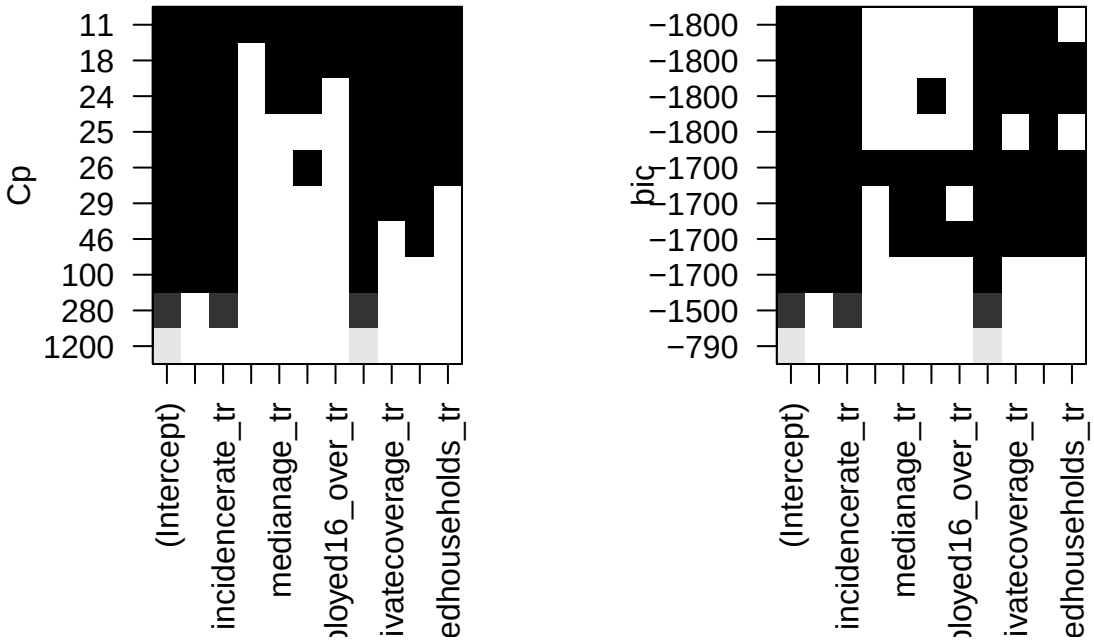
```
##    model         Cp        BIC
## 1      1 1228.72597  -790.0893
## 2      2  278.07692 -1542.0445
## 3      3  102.52061 -1701.0203
## 4      4   46.01549 -1750.2642
## 5      5   29.20656 -1760.9389
## 6      6   25.48142 -1758.6742
## 7      7   25.50299 -1752.6840
## 8      8   24.12538 -1748.0911
## 9      9   18.21137 -1748.0377
## 10    10   11.00000 -1749.3083
```

```
# plot the results
bw1 <- ggplot(results.bwd, aes(x = model, y = Cp)) +
  geom_point() + geom_line() + theme_bw() +
  geom_point(data = results.bwd[which.min(summary(regfit.bwd)$cp),],
             aes(y = Cp, x = model),
             size = 3, col = "red")
bw2 <- ggplot(results.bwd, aes(x = model, y = BIC)) +
  geom_point() + geom_line() + theme_bw() +
  geom_point(data = results.bwd[which.min(summary(regfit.bwd)$bic),],
             aes(y = BIC, x = model),
             size = 3, col = "red")
grid.arrange(bw1, bw2, ncol = 2)
```



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

Variable selection plots



Get coefficients of "Best models" Based on Cp

```
# The best model based on Cp is the one with 10 variables
coef(regfit.bwd, 10)
```

```
##            (Intercept)            medincome_tr       incidencerate_tr
##           -6.089763e+01           -9.200160e+05            8.100942e+01
##       povertypercent_tr            medianage_tr       percentmarried_tr
##           -2.091645e+01           -6.063544e+00            5.013645e-03
##   pctemployed16_over_tr    pctbachdeg25_over_tr   pctprivatecoverage_tr
##           -3.406660e-03           -2.181231e+01           -2.138683e-03
##             pctblack_tr pctmarriedhouseholds_tr
##            3.811772e+00           -6.649184e-03
```

Based on BIC

```
# The best model based on BIC is the one with 5 variables
coef(regfit.bwd, 5)
```

```
##           (Intercept)          medincome_tr      incidencerate_tr
##         -1.647427e+02         -8.142235e+05          8.293151e+01
##  pctbachdeg25_over_tr pctprivatecoverage_tr           pctblack_tr
##         -2.105662e+01         -2.080918e-03          4.130221e+00
```

The models selected with the Cp criterion are identical. Best subset selection, forward stepwise selection, and backward stepwise selection all had 10 variables, which is all of the explanatory variables in the data.

The models selected with the BIC criterion are also fairly similar for all of the selection method. The best

models of the best subset selection and backward stepwise selection have the identical variables, `medincome_tr`, `incidencerate_tr`, `pctbachdeg25_over_tr`, `pctprivatecoverage_tr`, and `pctblack_tr`. The backward stepwise selection have one more variable added, `povertypercent_tr`.

e) Repeat d), using forward stepwise selection and direct estimation of the test MSE (CV 5-fold). How does your answer compare to the results in c) and d)?

```
# summary of the forward stepwise selection
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(target_deathrate ~ ., cancer_tr, nvmax = 10,
##     method = "forward")
## 10 Variables  (and intercept)
##                          Forced in Forced out
## medincome_tr                 FALSE      FALSE
## incidencerate_tr             FALSE      FALSE
## povertypercent_tr            FALSE      FALSE
## medianage_tr                 FALSE      FALSE
## percentmarried_tr            FALSE      FALSE
## pctemployed16_over_tr        FALSE      FALSE
## pctbachdeg25_over_tr         FALSE      FALSE
## pctprivatecoverage_tr        FALSE      FALSE
## pctblack_tr                  FALSE      FALSE
## pctmarriedhouseholds_tr      FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: forward
##           medincome_tr incidencerate_tr povertypercent_tr medianage_tr
## 1  ( 1 )  " "          " "              " "               " "
## 2  ( 1 )  " "          "*"              " "               " "
## 3  ( 1 )  " "          "*"              "*"               " "
## 4  ( 1 )  " "          "*"              "*"               " "
## 5  ( 1 )  "*"          "*"              "*"               " "
## 6  ( 1 )  "*"          "*"              "*"               " "
## 7  ( 1 )  "*"          "*"              "*"               " "
## 8  ( 1 )  "*"          "*"              "*"               "*"
## 9  ( 1 )  "*"          "*"              "*"               "*"
## 10  ( 1 ) "*"          "*"              "*"               "*"
##           percentmarried_tr pctemployed16_over_tr pctbachdeg25_over_tr
## 1  ( 1 )  " "               " "                   "*"
## 2  ( 1 )  " "               " "                   "*"
## 3  ( 1 )  " "               " "                   "*"
## 4  ( 1 )  " "               " "                   "*"
## 5  ( 1 )  " "               " "                   "*"
## 6  ( 1 )  " "               " "                   "*"
## 7  ( 1 )  " "               " "                   "*"
## 8  ( 1 )  " "               " "                   "*"
## 9  ( 1 )  " "               "*"                   "*"
## 10  ( 1 ) "*"               "*"                   "*"
##           pctprivatecoverage_tr pctblack_tr pctmarriedhouseholds_tr
## 1  ( 1 )  " "                   " "         " "
## 2  ( 1 )  " "                   " "         " "
## 3  ( 1 )  " "                   " "         " "
## 4  ( 1 )  " "                   "*"         " "
```

```
## 5  ( 1 )  " "                  "*"          " "
## 6  ( 1 )  "*"                  "*"          " "
## 7  ( 1 )  "*"                  "*"          "*"
## 8  ( 1 )  "*"                  "*"          "*"
## 9  ( 1 )  "*"                  "*"          "*"
## 10  ( 1 ) "*"                  "*"          "*"
```

Step 1: Split into training and test sets, obtain validation folds

```
# Set seed for reproducibility
set.seed(323)

# Generate partition of the 5 folds
# The result is a list of length 5 with indices of observations to include in each fold.
num_crossval_folds <- 5
cross_fold_inds <- caret::createFolds(
  y = cancer_tr$target_deathrate,    # response variable as a vector
  k = num_crossval_folds # number of folds for CV
)
```

Step 2: Get performance for each fold, using the other folds put together as a training set.

```
# Object to store the results
results_mse <- expand.grid(
  coef_num = seq_len(10),
  fold_num = seq_len(num_crossval_folds),
  test_mse = NA
)
# For loops:
#    10 models from the forward stepwise selection (outside loop)
#    5 model fits for the 5 folds (inside loop)

for(coef_num in seq_len(10)) { # models
  for(fold_num in seq_len(num_crossval_folds)) { # folds

    # Index where to store results
    results_index <- which(
      results_mse$coef_num == coef_num &
      results_mse$fold_num == fold_num
    )

    # Training and testing sets (depends on the fold)
    cancer_train <- cancer_tr %>% slice(-cross_fold_inds[[fold_num]])
    cancer_test  <- cancer_tr %>% slice(cross_fold_inds[[fold_num]])

    # Fit the model
    coefs <- coef(regfit.fwd, coef_num)
    nams <- names(coefs)
    nams <- nams[!nams %in% "(Intercept)"] # exclude the intercept
    form <- as.formula(paste("target_deathrate",
                             paste(nams, collapse = " + "),
                             sep = " ~ "))
    fit <- lm(form, data = cancer_train)
```

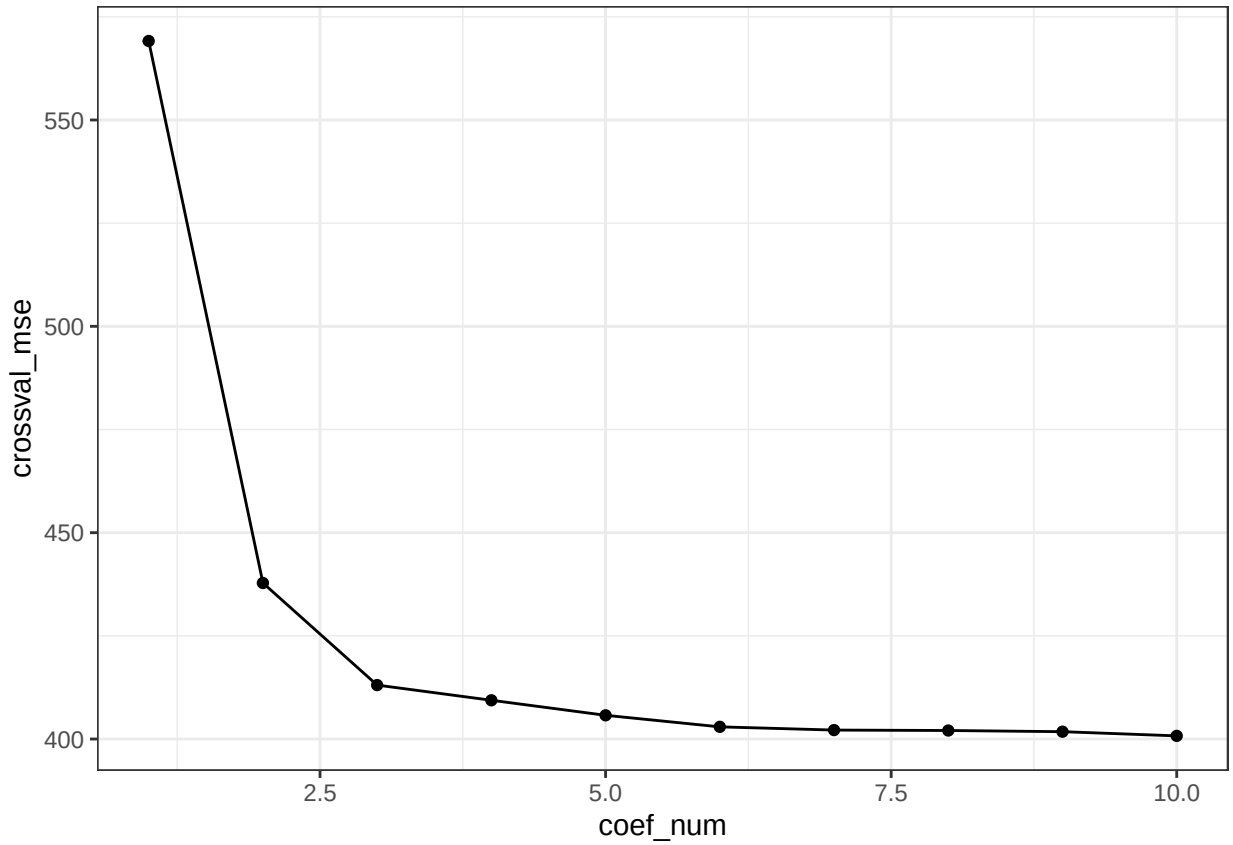
```
    # Get estimated test MSE
    pred = predict(fit, cancer_test)
    results_mse$test_mse[results_index] = mean((cancer_test$target_deathrate-pred)^2)
  }
}
head(results_mse)
```

```
##   coef_num fold_num test_mse
## 1        1        1 651.3894
## 2        2        1 504.5714
## 3        3        1 468.6005
## 4        4        1 462.2894
## 5        5        1 454.5661
## 6        6        1 452.0399
```

```
# summarize the results from cross validation
# need to take the average mse for the k folds
summarized_crossval_mse_results <- results_mse %>%
  group_by(coef_num) %>%
  summarize(
    crossval_mse = mean(test_mse)
  )
summarized_crossval_mse_results
```

```
## # A tibble: 10 x 2
##    coef_num crossval_mse
##       <int>        <dbl>
##  1        1         569.
##  2        2         438.
##  3        3         413.
##  4        4         409.
##  5        5         406.
##  6        6         403.
##  7        7         402.
##  8        8         402.
##  9        9         402.
## 10       10         401.
```

```
# plot the MSE test
ggplot(summarized_crossval_mse_results, aes(x= coef_num, y  = crossval_mse)) +
  geom_point() + geom_line() + theme_bw()
```



These results suggest that the model with all the coefficient variables have the lowest MSE.

However, in real life, we would prefer the simpler model if they have a similar MSE's.

```
# The best model based on MSE test is the one with 10 variables
coef(regfit.fwd, 10)
```

Get coefficients of "Best models"

```
##             (Intercept)            medincome_tr        incidencerate_tr
##           -6.089763e+01           -9.200160e+05            8.100942e+01
##       povertypercent_tr            medianage_tr       percentmarried_tr
##           -2.091645e+01           -6.063544e+00            5.013645e-03
##   pctemployed16_over_tr    pctbachdeg25_over_tr   pctprivatecoverage_tr
##           -3.406660e-03           -2.181231e+01           -2.138683e-03
##            pctblack_tr pctmarriedhouseholds_tr
##            3.811772e+00           -6.649184e-03
```

According to the best models based on Cp criterion of all the selection methods, all of the explanatory variables are included. Also, the MSE of the test data was the lowest for the model with 10 coefficients when we used forward stepwise selection and direct estimation of the test MSE (CV 5-fold). Therefore, we can conclude that all of the variables in the data are important to fit a model.

f) Now fit a lasso model to the data. Use cross-validation to select the optimal value of λ. Create plots of the cross-validation error as a function of λ. Report the resulting coefficient estimates, and discuss the results obtained.

Step 1: Split into training and test sets

```
# Set seed for reproducibility
set.seed(323)

# Divide into training and test sets
train_inds <- caret::createDataPartition(
y = cancer_tr$target_deathrate, # response variable as a vector
p = 0.75 # approx. proportion of data used for training
)
# Create the training and test data sets
cancer_train2 <- cancer_tr %>% slice(train_inds[[1]])
cancer_test2 <- cancer_tr %>% slice(-train_inds[[1]])
```

Step 2: Create the x matrix and y vector associated with the train and test data

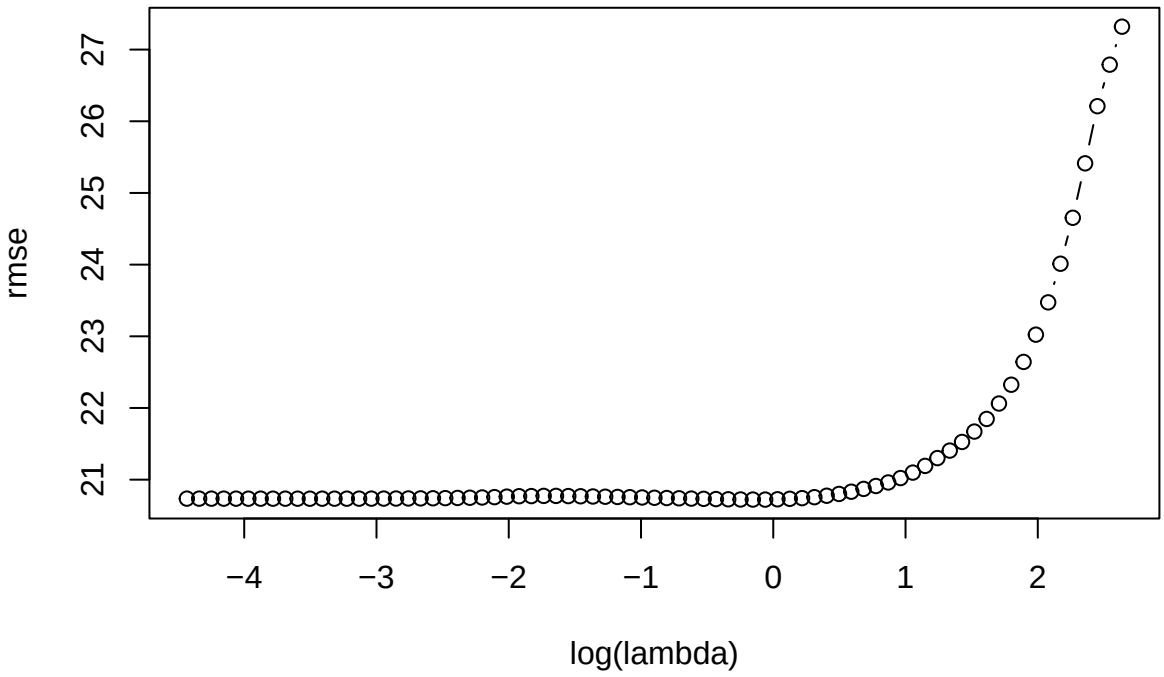
```
x_train <- model.matrix(target_deathrate ~ ., data = cancer_train2)
x_test <- model.matrix(target_deathrate ~ ., data = cancer_test2)
y_train <- cancer_train2$target_deathrate
y_test <- cancer_test2$target_deathrate
```

Step 3: Fit the lasso model on the train data and estimate the rmse

```
lasso.tr <- glmnet(x_train, y_train)
pred_test <- predict(lasso.tr, x_test)
rmse <- sqrt(apply((y_test- pred_test)^2, 2, mean))
```

Step 4: Plot the rmse agains the log lambda and find the best choice of lambda

```
plot(log(lasso.tr$lambda), rmse, type = "b", xlab = "log(lambda)")
```




```
lambda.best <- lasso.tr$lambda[order(rmse)][1]
lambda.best
```

```
## [1] 0.9406151
```

Step 5: Use your choice of lambda to fit the model with all the data

```
x = model.matrix(target_deathrate ~ ., data = cancer_tr)
y = cancer_tr$target_deathrate
lasso.fit.best <- glmnet(x, y, alpha = 1, lambda = lambda.best)
coef(lasso.fit.best)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                      s0
## (Intercept)             -1.234096e+02
## (Intercept)              .
## medincome_tr            -6.625489e+05
## incidencerate_tr         7.550278e+01
## povertypercent_tr        .
## medianage_tr             .
## percentmarried_tr        .
## pctemployed16_over_tr   -5.526360e-04
## pctbachdeg25_over_tr    -2.058876e+01
## pctprivatecoverage_tr   -1.478275e-03
## pctblack_tr              2.628748e+00
## pctmarriedhouseholds_tr -1.428595e-03
```