Homework 04 Solutions

2022-09-22

36-617: Applied Linear Models
Fall 2022
Solutions

```
library(arm)  ## includes lme4, MASS, Matrix
library(ggplot2); theme_set(theme_bw())
library(gridExtra) ## to arrange ggplots...
library(GGally) ## for ggpairs...

library(leaps) ## regsubsets(), summary(), coef()
library(car)   ## subsets(), mmps(), vif(), etc.
```

Problem 1: Sheather, Ch 6, pp. 216–221, #3.

Continuing the analysis of data in "cars04.csv"... Notes:

- The data set is available as "cars04.csv".
- There are some errors in the results for fitting the model (6.37) in the textbook, so I suggest you refit the models for this problem directly from the data in "cars04.csv".

The variables are:

$Y =$ Suggested Retail Price;

$x_1 =$ Engine size;

$x_2 =$ Cylinders;

$x_3 =$ Horse power;

$x_4 =$ Highway mpg;

$x5 =$ Weight;

$x_6 =$ Wheel Base;

$x_7 =$ Hybrid (1=Hybrid; 0=just gasoline-powered)

The first model proposed is

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \epsilon \tag{6.36}$$

After examining Box-Cox transformation proposals and "rounding" to more interpretable values, the second model proposed is

$$\log(Y) = \beta_0 + \beta_1 x_1^{0.25} + \beta_2 \log(x_2) + \beta_3 \log(x_3) + \beta_4 (1/x_4) + \beta_5 x_5 + \beta_6 \log(x_6) + \beta_7 x_7 + \epsilon \tag{6.37}$$

Since there are some problems with the analysis presented in Sheather, we refit all the models and reproduce all the outputs (except that I am just going to accept the transformations suggested in (6.37)).

Problem 1(a)

Decide whether (6.36) is a valid model. Give reasons to support your answer.

We start by refitting the model and reproducing the diagnostic plots in Sheather.

```
cars <- read.csv("cars04.csv")
str(cars)
```

```
## 'data.frame':    234 obs. of  13 variables:
##  $ Vehicle.Name        : chr  "Chevrolet Aveo 4dr" "Chevrolet Aveo LS 4dr hatch" "Chevrolet Cavalier
##  $ Hybrid              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SuggestedRetailPrice: int  11690 12585 14610 14810 16385 13670 15040 13270 13730 15460 ...
##  $ DealerCost          : int  10965 11802 13697 13884 15357 12849 14086 12482 12906 14496 ...
##  $ EngineSize          : num  1.6 1.6 2.2 2.2 2.2 2 2 2 2 2 ...
##  $ Cylinders           : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Horsepower          : int  103 103 140 140 140 132 132 130 110 130 ...
##  $ CityMPG             : int  28 28 26 26 26 29 29 26 27 26 ...
##  $ HighwayMPG          : int  34 34 37 37 37 36 36 33 36 33 ...
##  $ Weight              : int  2370 2348 2617 2676 2617 2581 2626 2612 2606 2606 ...
##  $ WheelBase           : int  98 98 104 104 104 105 105 103 103 103 ...
##  $ Length              : int  167 153 183 183 183 174 174 168 168 168 ...
##  $ Width               : int  66 66 69 68 69 67 67 67 67 67 ...
```

```
names(cars)
```

```
##  [1] "Vehicle.Name"         "Hybrid"               "SuggestedRetailPrice"
##  [4] "DealerCost"           "EngineSize"           "Cylinders"
##  [7] "Horsepower"           "CityMPG"              "HighwayMPG"
## [10] "Weight"               "WheelBase"            "Length"
## [13] "Width"
```

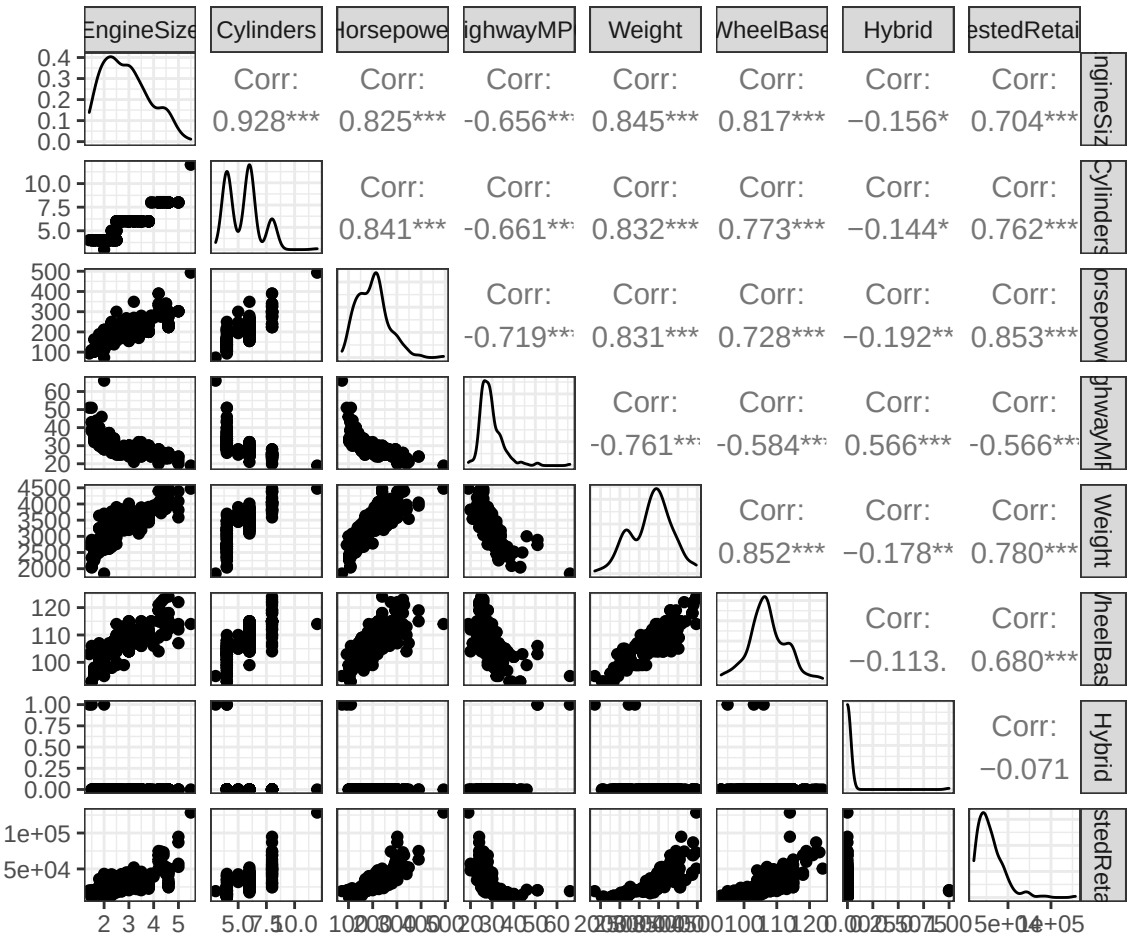
```
cars.red <- cars[,c("EngineSize","Cylinders",
"Horsepower","HighwayMPG","Weight","WheelBase","Hybrid","SuggestedRetailPrice")]
dim(cars.red)
```

```
## [1] 234   8
```

```
names(cars.red)
```

```
## [1] "EngineSize"           "Cylinders"            "Horsepower"
## [4] "HighwayMPG"           "Weight"               "WheelBase"
## [7] "Hybrid"               "SuggestedRetailPrice"
```

```
ggpairs(cars.red)
```

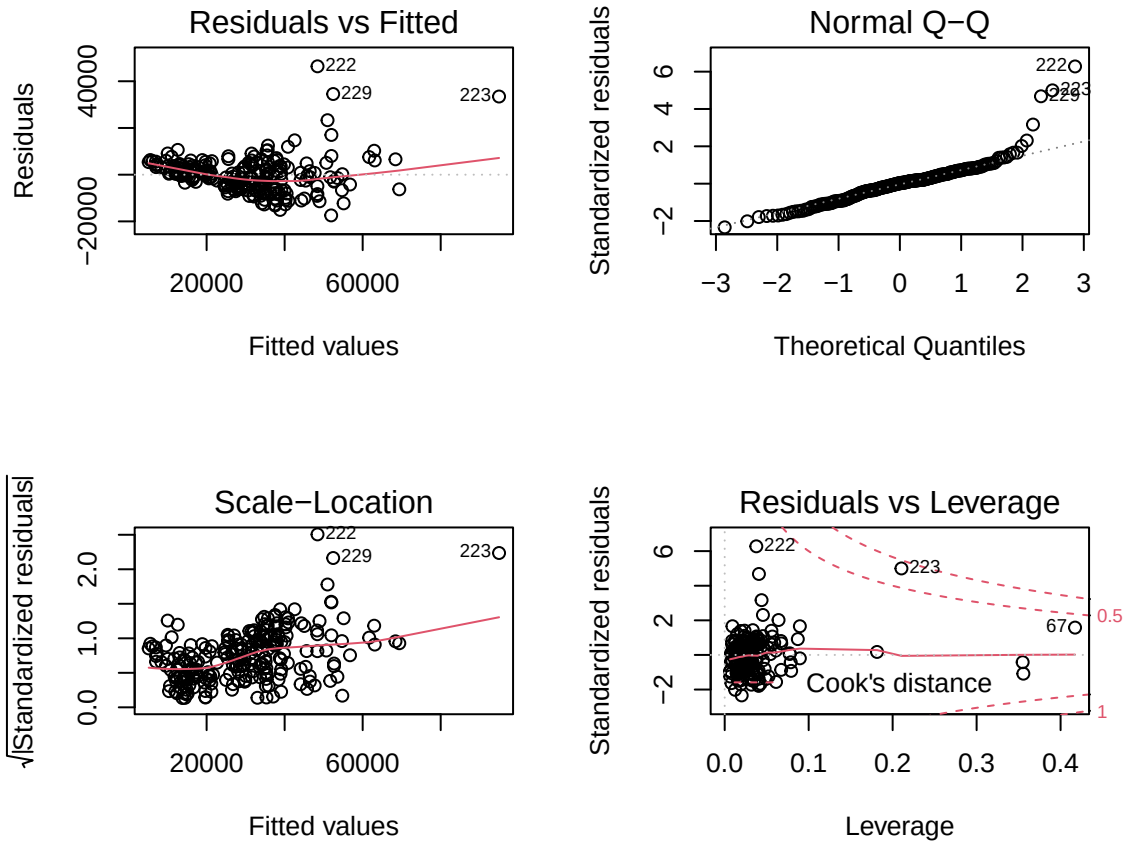



```
  ## Sheather did not include the response variable "SuggestedRetailPrice",
  ## but I think it is good to include since it can help us guess transformations...
summary(lm.6.36 <- lm(SuggestedRetailPrice ~ ., data=cars.red))
```

```
##
## Call:
## lm(formula = SuggestedRetailPrice ~ ., data = cars.red)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -17436  -4134    173   3561  46392
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -68965.793  16180.381  -4.262 2.97e-05 ***
## EngineSize   -6957.457   1600.137  -4.348 2.08e-05 ***
## Cylinders     3564.755    969.633   3.676 0.000296 ***
## Horsepower     179.702     16.411  10.950  < 2e-16 ***
## HighwayMPG     637.939    202.724   3.147 0.001873 **
## Weight          11.911      2.658   4.481 1.18e-05 ***
## WheelBase       47.607    178.070   0.267 0.789444
## Hybrid         431.759   6092.087   0.071 0.943562
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7533 on 226 degrees of freedom
## Multiple R-squared:  0.7819, Adjusted R-squared:  0.7751
## F-statistic: 115.7 on 7 and 226 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm.6.36)
```



Neither the scatter plot matrix nor the `summary()` *printout can tell us whether the model is valid (the scatter plot matrix can help us make guesses about transformations and/or collinearity. The* `summary` *output can tell us whether the fitted model will be useful for answering questions our collaborator may have). Validity of the model depends only on*

- *Linear relationship between* Y *and the* X*'s*
- *Normal errors* ϵ
- *Constant error variance* $\sigma^2 = Var(\epsilon)$

We can use the casewise diagnostic plots for this:

Linear relationship between Y **and the** X**'s:** *The "Residuals vs. Fitted" plot suggests a curved relationship between the residuals* $\hat{e}$ *and the fitted values* $\hat{y}$*. This suggests that the relationship between* Y *and the* X*'s is nonlinear.*

Normal errors ϵ**:** *The "Normal Q-Q" plot of the standardized residuals* r_i *shows quite a bit of right-skew.*

This suggests that the errors ϵ are not normally distributed.

Constant error variance $\sigma = \sqrt{Var(\epsilon)}$: *The red trend line in the "Scale-Location" is increasing from left to right, suggesting that the variance $Var(\epsilon)$ is not constant across the data.*

Since there is evidence of violations of all three basic assumptions of the model, we can say the model is not fully valid.

Note: *The model might still be useful in some way, e.g. for prediction. However it does not satisfy the basic statistical assumptions above, and so any tests or estimates for the β's, for the $\hat{y}$'s for σ should be viewed with some suspicion.*

Problem 1(b)

The plot of residuals against fitted values produces a curved pattern. Describe what, if anything can be learned about model (6.36) from this plot.

The curved pattern in the residuals suggests we should look for transofmrations of X's or Y's to produce a set of raw residuals $\hat{e}$ with less functional dependence on the fitted values $\hat{y}$.

Problem 1(c)

Identify any bad leverage points for model (6.36).

From the "Residuals vs Leverage" plot we can see:

- *Observation #223 seems to be influential, with a Cook's Distance > 0.5.*
- *Observation #67 may also be influention, with a Cook's Distance near 0.5.*
- *Observation #222 does not have high leverage, but its residual is a hugh outlier—more than 6 SD's from the middle of the residual distribution.*
- *Two more observations, not labelled by R, have relatively high leverage (≈ 0.35) but pretty small residuals.*

Note: *The usual cutoff for "high leverage" is*

$$h_{high} = 2 \cdot \frac{p+1}{n} = 2 \cdot \frac{8}{234} \approx 0.07$$

so there are actually 5 or more points with leverage $h_{ii} > h_{high}$, but the points I've identified above are the ones that I might devote some worry to in practice.

Problem 1(d)

Decide whether (6.37) is a valid model.

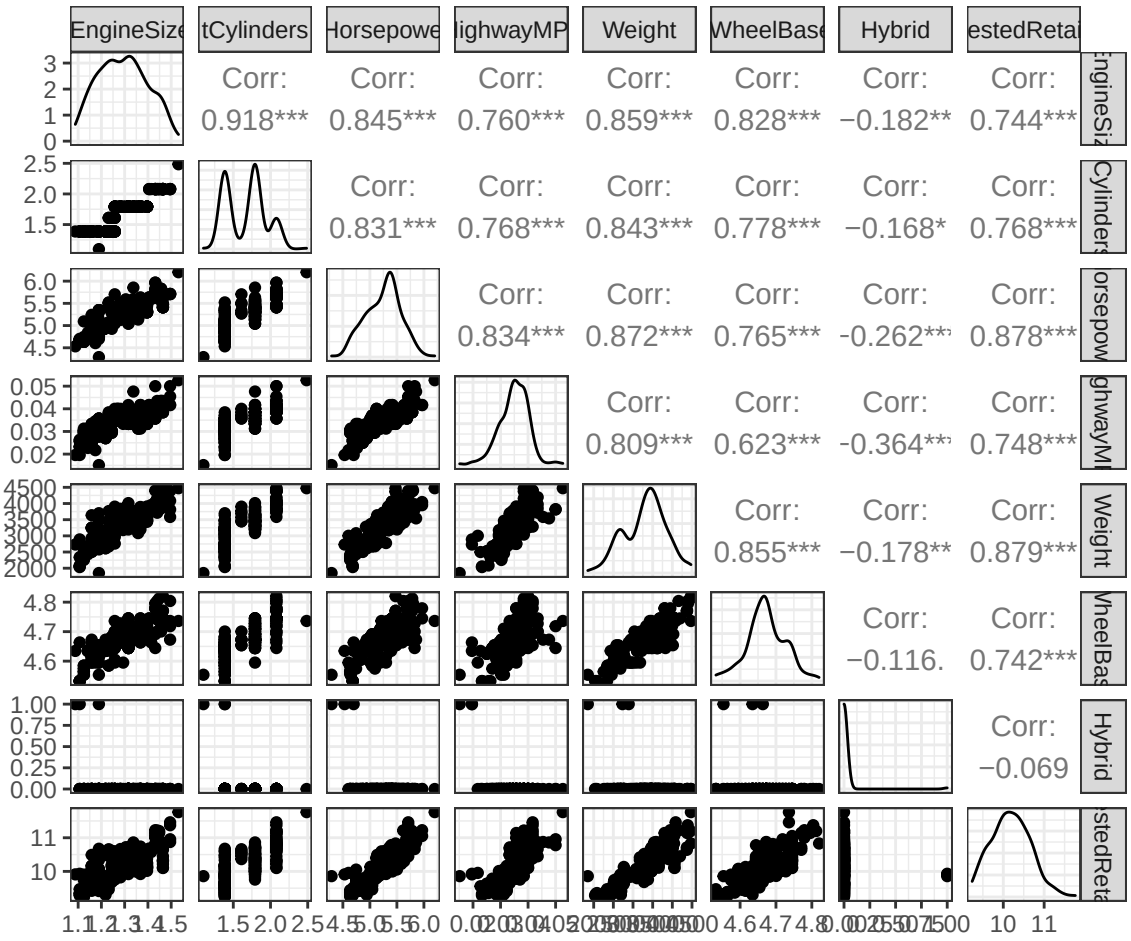
First we refit the model and reproduce the output in Sheather (this is the part that Sheather appaers to have gotten wrong):

```
attach(cars.red) ## so I don't have to type "cars.red$" all the time...
tSuggestedRetailPrice <- log(SuggestedRetailPrice)
tEngineSize <- EngineSize^(0.25)
tCylinders <- log(Cylinders)
tHorsepower <- log(Horsepower)
tHighwayMPG <- 1/HighwayMPG
tWheelBase <- log(WheelBase)
cars.t <- data.frame(tEngineSize,
                     tCylinders, tHorsepower, tHighwayMPG,
                     Weight, tWheelBase, Hybrid,tSuggestedRetailPrice)
```

```
detach()
names(cars.t)
```

```
## [1] "tEngineSize"           "tCylinders"            "tHorsepower"
## [4] "tHighwayMPG"           "Weight"                "tWheelBase"
## [7] "Hybrid"                "tSuggestedRetailPrice"
```

```
ggpairs(cars.t)
```

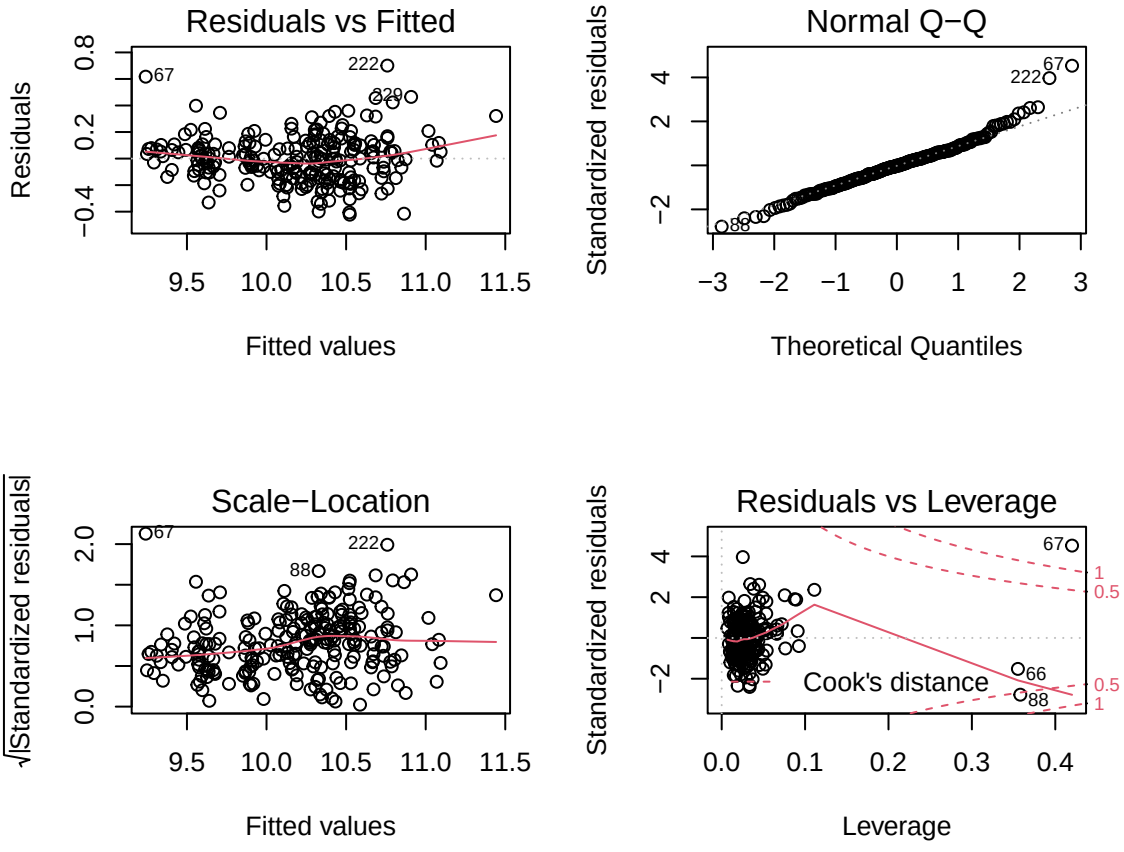



```
summary(lm.6.37 <- lm(tSuggestedRetailPrice ~ ., data=cars.t))
```

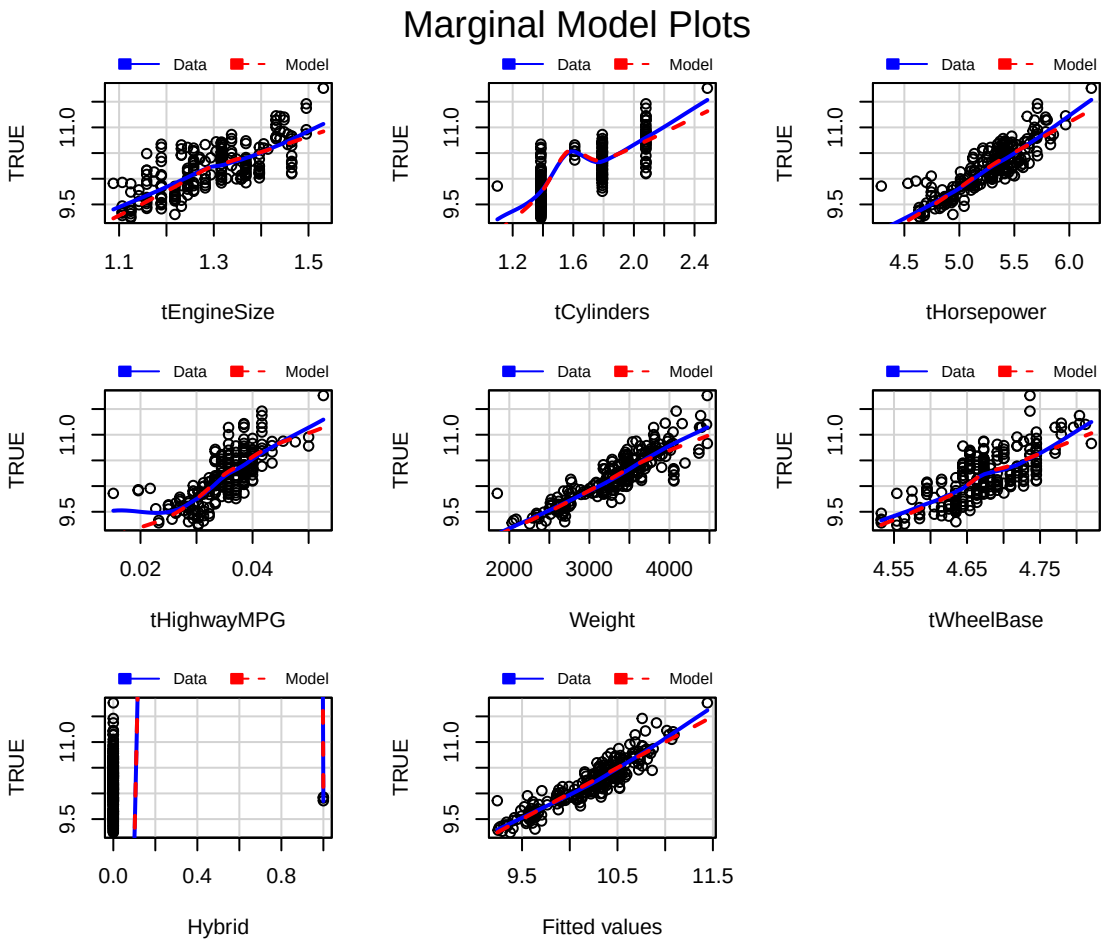
```
##
## Call:
## lm(formula = tSuggestedRetailPrice ~ ., data = cars.t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42288 -0.10983 -0.00203  0.10279  0.70068
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.703e+00  2.010e+00   2.838  0.00496 **
## tEngineSize -1.575e+00  3.332e-01  -4.727 4.01e-06 ***
## tCylinders   2.335e-01  1.204e-01   1.940  0.05359 .
```

```
## tHorsepower  8.992e-01  8.876e-02  10.130  < 2e-16 ***
## tHighwayMPG  8.029e-01  4.758e+00   0.169  0.86614
## Weight       5.043e-04  6.367e-05   7.920 1.07e-13 ***
## tWheelBase  -6.385e-02  4.715e-01  -0.135  0.89240
## Hybrid       6.422e-01  1.150e-01   5.582 6.78e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1789 on 226 degrees of freedom
## Multiple R-squared:  0.8621, Adjusted R-squared:  0.8578
## F-statistic: 201.8 on 7 and 226 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm.6.37)
```




```
mmps(lm.6.37)
```



Now, following the same approach as before, we use the casewise diagnostic plots to check: We can use the casewise diagnostic plots for this:

Linear relationship between Y and the X's: *The red trend-line in the "Residuals vs Fitted" plot is nearly flat, suggesting no relationship between $\hat{e}$ and $\hat{y}$. This suggests that we have an approximately linear relationship between the new Y and the new X's in the transformed model.*

Normal errors ϵ: *Except for two outliers (obs. #222 and #67) the "Normal Q-Q" plot confirms that the standardized residuals r_i are approximately normally distributed, suggesting that the errors ϵ are normally distributed.*

Constant error variance $\sigma = \sqrt{Var(\epsilon)}$: *The red trend line in the "Scale-Location" plot is nearly flat, suggesting that the error variance $Var(\epsilon)$ is constant across the data set.*

Of course no fit will be perfectly valid, but the evidence from the casewise diagnosic plots suggests that (6.37) is a much more valid model than (6.36).

***Note:** We can also see, from the scatterplot matrix and from the marginal model plots, that we seem to have done a good job with transforms. Especially, in the marginal model plots, the estimates of $E[Y|X]$ from the nonparametric regressions (blue) and the linear model (red) line up well, so it doesn't appear that any further transformations are needed.*

Problem 1(e)

To obtain a final model, the analyst wants to simply remove the two insignificant predictors $1/x_4$ (i.e., `tHighwayMPG`) and $\log(x_6)$ (i.e., `tWheelBase`) from (6.37). Perform a partial F-test to see if this is a sensible

strategy.

We will fit the smaller model (without the two variables that the analyst wants to remove) and then compare that with `lm.6.37`*:*

```
lm.smaller <- update(lm.6.37, . ~ . - tHighwayMPG - tWheelBase)
anova(lm.smaller, lm.6.37)
```

```
## Analysis of Variance Table
##
## Model 1: tSuggestedRetailPrice ~ tEngineSize + tCylinders + tHorsepower +
##     Weight + Hybrid
## Model 2: tSuggestedRetailPrice ~ tEngineSize + tCylinders + tHorsepower +
##     tHighwayMPG + Weight + tWheelBase + Hybrid
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1    228 7.2358
## 2    226 7.2337  2 0.0021769 0.034 0.9666
```

With a p-value of 0.97, there reall is no evidence in favor of keeping these two variables in the model, and so `lm.smaller` *seems like a sufficient model:*

```
summary(lm.smaller)
```

```
##
## Call:
## lm(formula = tSuggestedRetailPrice ~ tEngineSize + tCylinders +
##     tHorsepower + Weight + Hybrid, data = cars.t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42224 -0.11001 -0.00099  0.10191  0.70205
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.422e+00  3.291e-01  16.474  < 2e-16 ***
## tEngineSize -1.591e+00  3.157e-01  -5.041 9.45e-07 ***
## tCylinders   2.375e-01  1.186e-01   2.003   0.0463 *
## tHorsepower  9.049e-01  8.305e-02  10.896  < 2e-16 ***
## Weight       5.029e-04  5.203e-05   9.666  < 2e-16 ***
## Hybrid       6.340e-01  1.080e-01   5.870 1.53e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1781 on 228 degrees of freedom
## Multiple R-squared:  0.862,  Adjusted R-squared:  0.859
## F-statistic: 284.9 on 5 and 228 DF,  p-value: < 2.2e-16
```

We probably should go ahead and check casewise diagnostic plots, marginal model plots, etc., to be sure that we still have a valid model and no more transformations are needed, but I will stop here.

Problem 1(f)

The analyst's boss has complained about model (6.37) saying that it fails to take account of the manufacturer of the vehicle (e.g., BMW vs Toyota). Describe how model (6.37) could be expanded in order to estimate the effect of manufacturer on suggested retail price.

This is slightly tricky to do well. The basic idea (and, really, all you have to say to answer the question) is that you want to extract the manufacturer name from the variable `Vehicle.Name` *in the orignal data set* `cars`*,*

and add that to the regressions above, and see what happens...

Doing the extraction is a little tricky but here's a way to proceed, if you are curious:

The `Vehicle.Name` *variable in the original data set is nearly unique for each car in the data set:*

```
dim(cars)
```

```
## [1] 234  13
```

```
length(unique(cars$Vehicle.Name))
```

```
## [1] 232
```

(There are two cars duplicated in the data set,

```
tmp <- table(cars$Vehicle.Name)
tmp[tmp>1]
```

```
##
##       Infiniti G35 4dr Mercedes-Benz C240 4dr
##                      2                      2
```

```
cars[grep(names(tmp[tmp>1])[1],cars$Vehicle.Name),]
```

```
##         Vehicle.Name Hybrid SuggestedRetailPrice DealerCost EngineSize
## 116 Infiniti G35 4dr      0                28495      26157        3.5
## 161 Infiniti G35 4dr      0                32445      29783        3.5
##     Cylinders Horsepower CityMPG HighwayMPG Weight WheelBase Length Width
## 116         6        260      18         26   3336       112    187    69
## 161         6        260      18         26   3677       112    187    69
```

```
cars[grep(names(tmp[tmp>1])[2],cars$Vehicle.Name),]
```

```
##               Vehicle.Name Hybrid SuggestedRetailPrice DealerCost EngineSize
## 169 Mercedes-Benz C240 4dr      0                32280      30071        2.6
## 170 Mercedes-Benz C240 4dr      0                33480      31187        2.6
##     Cylinders Horsepower CityMPG HighwayMPG Weight WheelBase Length Width
## 169         6        168      20         25   3360       107    178    68
## 170         6        168      19         25   3360       107    178    68
```

but this doesn't change the problem.)

With 232 unique vehicle names for 234 observation in the data set, if we just include vehicle name in a regression, it will soak up all of the variability explained by the other variables, and we won't get any useful information out.

We note that the make (i.e. the manufacturer name) of the car always comes first in the vehicle name, followed by a space. So we can make a new variable, `Make`*, that just has the manufacturers' names:*

```
splits <- strsplit(cars$Vehicle.Name," ")
head(splits,10)
```

```
## [[1]]
## [1] "Chevrolet" "Aveo"      "4dr"
##
## [[2]]
## [1] "Chevrolet" "Aveo"      "LS"        "4dr"       "hatch"
##
## [[3]]
## [1] "Chevrolet" "Cavalier"  "2dr"
```

```
##
## [[4]]
## [1] "Chevrolet" "Cavalier"  "4dr"
##
## [[5]]
## [1] "Chevrolet" "Cavalier"  "LS"        "2dr"
##
## [[6]]
## [1] "Dodge" "Neon"  "SE"    "4dr"
##
## [[7]]
## [1] "Dodge" "Neon"  "SXT"   "4dr"
##
## [[8]]
## [1] "Ford"  "Focus" "ZX3"   "2dr"   "hatch"
##
## [[9]]
## [1] "Ford"  "Focus" "LX"    "4dr"
##
## [[10]]
## [1] "Ford"  "Focus" "SE"    "4dr"
```

```
mfr.name <- sapply(splits, function(x) x[1])
head(mfr.name,10)
```

```
##  [1] "Chevrolet" "Chevrolet" "Chevrolet" "Chevrolet" "Chevrolet" "Dodge"
##  [7] "Dodge"     "Ford"      "Ford"      "Ford"
```

```
cars$Make <- mfr.name
length(unique(cars$Make))
```

```
## [1] 33
```

```
table(cars$Make)
```

```
##
##         Acura          Audi           BMW         Buick      Cadillac
##             5            13            13             7             4
##     Chevrolet      Chrvsler      Chrysler         Dodge          Ford
##            13             1            10             6            10
##         Honda       Hyundai      Infiniti        Jaguar           Kia
##            11            10             6             8             7
##         Lexus       Lincoln        Mazda6 Mercedes-Benz       Mercury
##             6             7             1            15             6
##          Mini    Mitsubishi        Nissan    Oldsmobile       Pontiac
##             2             2             7             2             5
##          Saab        Saturn         Scion        Subaru        Suzuki
##             6             6             1             6             5
##        Toyota    Volkswagen         Volvo
##            15             9             9
```

Now we could introduce the `Make` *variable into any of the regressions we fitted above, to see if the car maker affect prediction of the* `SuggestedRetailPrice`*, separately from the other features of the car.*

Problem 2: Sheather, Ch 7, p. 261, #3.

Continuing with analysis of "pgatour2006.csv"... Note:

- The data is available in "pgatour2006.csv".

The variables are:

$Y =$ PrizeMoney;

$x_1 =$ DrivingAccuracy;

$x_2 =$ GIR;

$x_3 =$ PuttingAverage;

$x_4 =$ BirdieConversion;

$x5 =$ SandSaves;

$x_6 =$ Scrambling;

$x_7 =$ PuttsPerRound

and the model we are considering is

$$\log(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \epsilon \tag{7.10}$$

We want to do variable selection to choose a subset of the predictors to model $\log(Y)$.

Problem 2(a)

Identify the optimal model or models based on R^2_{adj}, AIC, AIC$_C$, BIC from the approach based on all possible subsets.

Remember that "all subsets" selection divides the possible models into groups according to the number of predictors, up to the maximum number of predictors (seven, in our case):

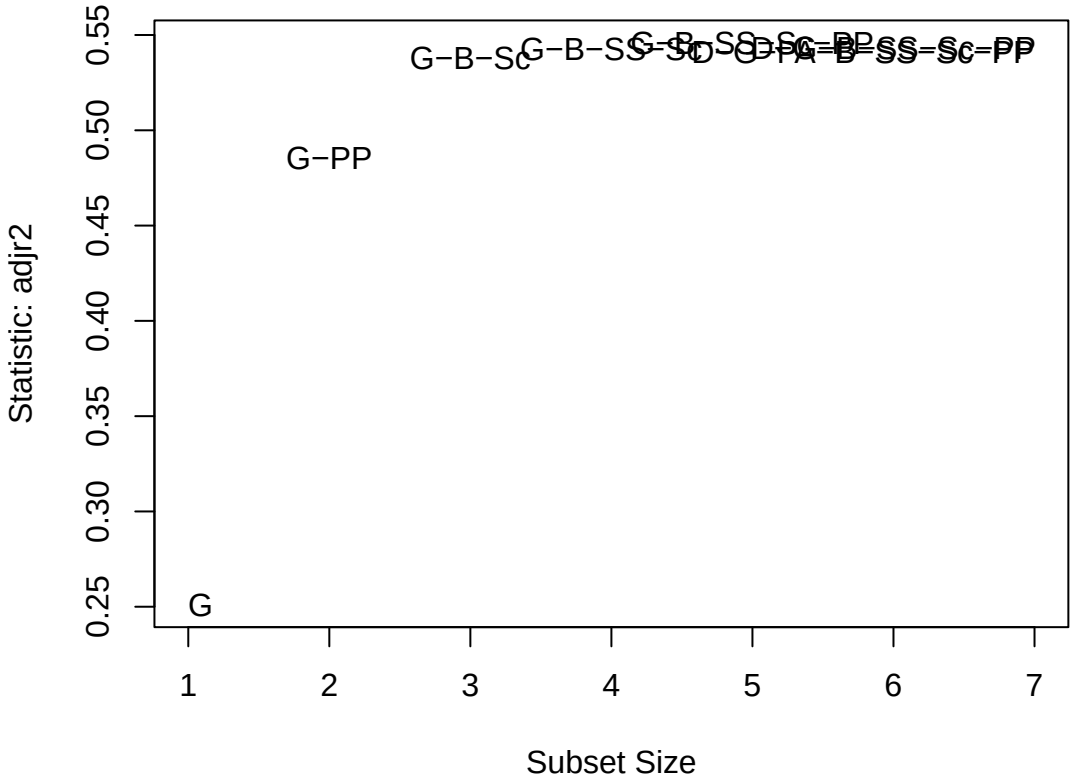
- *First find the 1-predictor model with the smallest RSS.*
- *Then find the 2-predictor model with the smallest RSS.*
- *... and so forth up to the 7-predictor model with the smallest RSS.*

These seven models can then be compared with AIC, BIC, etc.

```
golf <- read.csv("pgatour2006.csv")

golf.red <- golf[,c("PrizeMoney","DrivingAccuracy","GIR","PuttingAverage",
                    "BirdieConversion","SandSaves","Scrambling","PuttsPerRound")]

all.subsets <- regsubsets(log(PrizeMoney) ~ ., data=golf.red)
```

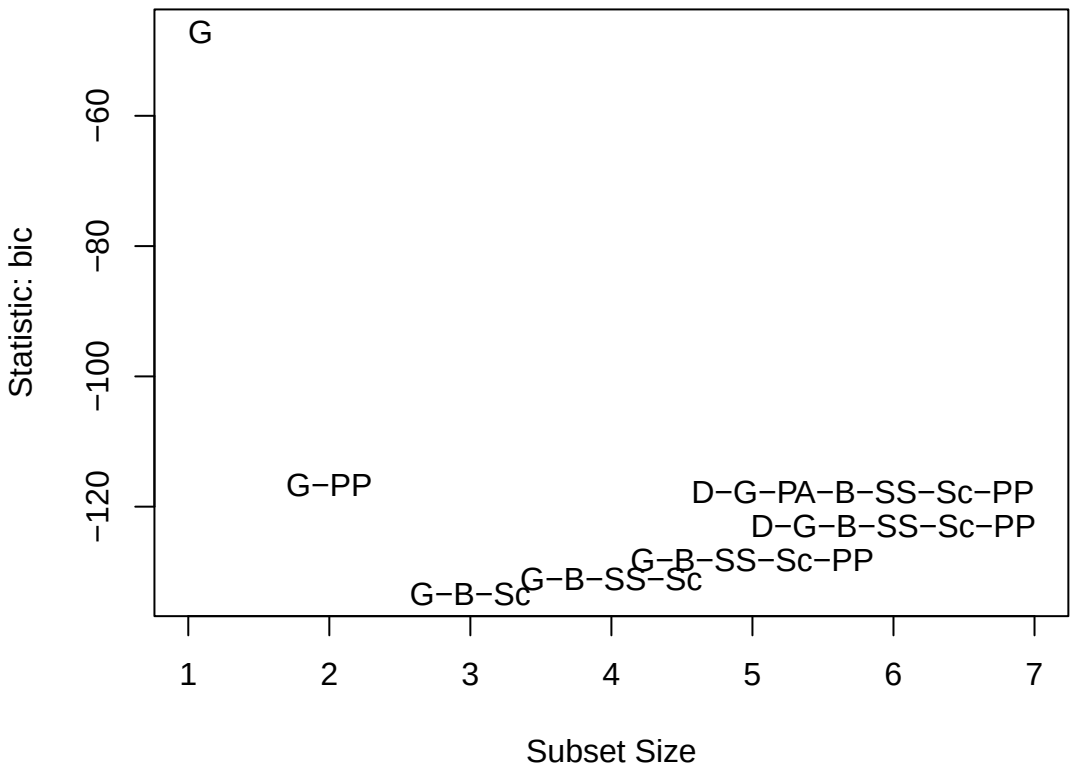
We can get some of the results we want from the "subsets()" function:

```
subsets(all.subsets,statistic="adjr2",legend=FALSE)
```




```
##                   Abbreviation
## DrivingAccuracy              D
## GIR                          G
## PuttingAverage              PA
## BirdieConversion             B
## SandSaves                   SS
## Scrambling                  Sc
## PuttsPerRound               PP
```

```
subsets(all.subsets,statistic="bic",legend=FALSE)
```




```
##                   Abbreviation
## DrivingAccuracy              D
## GIR                          G
## PuttingAverage              PA
## BirdieConversion             B
## SandSaves                   SS
## Scrambling                  Sc
## PuttsPerRound               PP
```

But to completely do the problem we need another strategy, since the "subsets()" function does not know AIC or CAIC (see the "statistic" parameter in Figure 1 below):

```
help(subsets)
## opens a browser window with the help, but I've included part of the
## contents of the browser window in Figure 1 below.
```

So, we have to calculate AIC and CAIC (and in the process we'll also see how to do BIC)

Note that from lecture 08, slide 7, at the bottom, we can write the log-likelihood as

$$\textit{(log-likelihood)} = c_1(n) - c_2(n)\log(RSS) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(RSS)$$

which means we can calculate AIC by hand as

```
AIC = -2(log-liklihood) + 2*(p+2) = [n*log(2pi)] + n*log(RSS) + 2*(p+2)
```

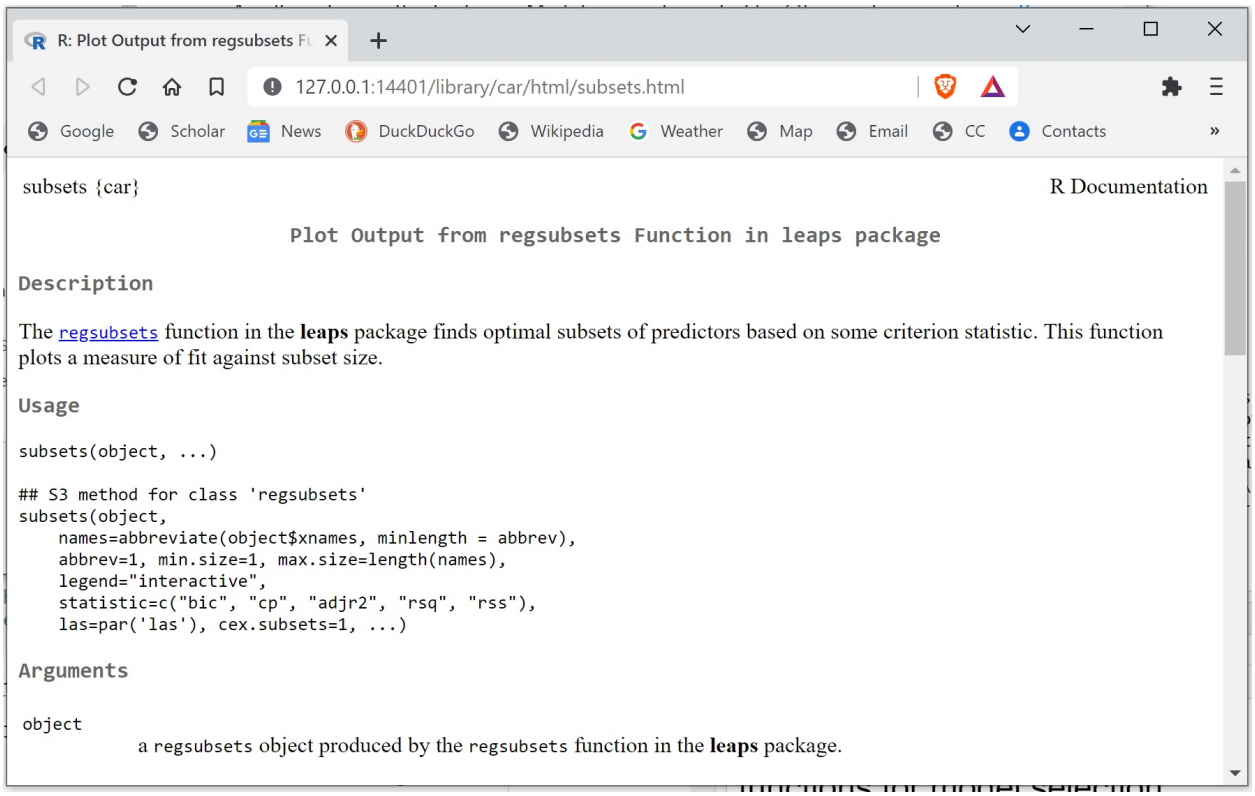


Figure 1: help(subsets)

and since the `[n*log(2pi)]` *term will cancel when we subtract AIC's we can ignore it and just write*

```
AIC = n*log(RSS) + 2*(p+2)
```

Similarly,

```
CAIC = AIC + 2*(p+2)*(p+3)/(n-p-1) = n*log(RSS) + 2*(p+2) + 2*(p+2)*(p+3)/(n-p-1)
```

and

```
BIC = n*log(RSS) + log(n)*(p+2)
```

We can get the RSS for each model from

```
tmp <- summary(all.subsets)
names(tmp)
```

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

```
p <- 1:7 ## number of parameters in each subset model
n <- dim(golf.red)[1]

attach(tmp)
results <- data.frame(which,rss,adjr2,bic=n*log(rss)+log(n)*(p+2),aic=n*log(rss)+2*(p+2),
                      caic=n*log(rss) + 2*(p+2) + 2*(p+2)*(p+1)/(n-p-1))
detach()
```

Note that different authors will use somewhat different definitions of AIC and BIC. The differences are usually just in what is done with the constants $c_1(n)$ and $c_2(n)$, so the value of the criterion changes, but the model that minimizes the criterion does not change.

We can print the "results" data frame and get the minumum by hand:

```
results
```

```
##   X.Intercept. DrivingAccuracy  GIR PuttingAverage BirdieConversion SandSaves
## 1         TRUE           FALSE TRUE          FALSE            FALSE     FALSE
## 2         TRUE           FALSE TRUE          FALSE            FALSE     FALSE
## 3         TRUE           FALSE TRUE          FALSE             TRUE     FALSE
## 4         TRUE           FALSE TRUE          FALSE             TRUE      TRUE
## 5         TRUE           FALSE TRUE          FALSE             TRUE      TRUE
## 6         TRUE            TRUE TRUE          FALSE             TRUE      TRUE
## 7         TRUE            TRUE TRUE           TRUE             TRUE      TRUE
##   Scrambling PuttsPerRound       rss     adjr2      bic      aic     caic
## 1      FALSE         FALSE 139.59511 0.2510765 983.8286 973.9943 974.0561
## 2      FALSE          TRUE  95.35465 0.4857746 914.4027 901.2902 901.4146
## 3       TRUE         FALSE  85.19106 0.5381917 897.5903 881.1997 881.4080
## 4       TRUE         FALSE  83.90549 0.5427792 899.8881 880.2195 880.5336
## 5       TRUE          TRUE  82.90524 0.5458520 902.8157 879.8689 880.3110
## 6       TRUE          TRUE  82.86756 0.5436566 908.0047 881.7798 882.3724
## 7       TRUE          TRUE  82.86555 0.5412404 913.2780 883.7750 884.5410
```

Or we can write a function to identify the row that minimizes each criteron:

```
minimize <- function(res,col) {
  obj <- res[,col]
  k <- (1:length(obj))[obj==min(obj)]
  return(res[k,])
}

maximize <- function (res,col) {
  obj <- res[,col]
  k <- (1:length(obj))[obj==max(obj)]
  return(res[k,])
}

maximize(results,"adjr2")
```

```
##   X.Intercept. DrivingAccuracy  GIR PuttingAverage BirdieConversion SandSaves
## 5         TRUE           FALSE TRUE          FALSE             TRUE      TRUE
##   Scrambling PuttsPerRound      rss    adjr2      bic      aic    caic
## 5       TRUE          TRUE 82.90524 0.545852 902.8157 879.8689 880.311
```

```
minimize(results,"bic")
```

```
##   X.Intercept. DrivingAccuracy  GIR PuttingAverage BirdieConversion SandSaves
## 3         TRUE           FALSE TRUE          FALSE             TRUE     FALSE
##   Scrambling PuttsPerRound      rss     adjr2      bic      aic    caic
## 3       TRUE         FALSE 85.19106 0.5381917 897.5903 881.1997 881.408
```

```
minimize(results,"aic")
```

```
##   X.Intercept. DrivingAccuracy  GIR PuttingAverage BirdieConversion SandSaves
## 5         TRUE           FALSE TRUE          FALSE             TRUE      TRUE
##   Scrambling PuttsPerRound      rss    adjr2      bic      aic    caic
## 5       TRUE          TRUE 82.90524 0.545852 902.8157 879.8689 880.311
```

```
minimize(results,"caic")
```

```
##   X.Intercept. DrivingAccuracy  GIR PuttingAverage BirdieConversion SandSaves
```

```
## 5         TRUE           FALSE TRUE          FALSE             TRUE      TRUE
##   Scrambling PuttsPerRound      rss    adjr2      bic      aic    caic
## 5       TRUE          TRUE 82.90524 0.545852 902.8157 879.8689 880.311
```

Reading the "TRUE" and "FALSE" values as when to include or not include a variable in the model, we see that for "all subsets" selection:

- *The optimal R^2_{adj} model is*
  `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling + SandSaves + PuttPerRound`
- *The optimal BIC model is*
  `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling`
- *The optimal AIC model is*
  `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling + SandSaves + PuttsPerRound`
- *The optimal CAIC model is*
  `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling + SandSaves + PuttsPerRound`

It's a bit hard to read the graph that `subsets()` *gives us for R^2_{adj} but clearly the best BIC model we identified is the same as the one that* `subsets()` *gives us for BIC.*

Problem 2(b)

Identify the optimal model or models based on AIC and BIC from the approach based on backward selection.

We can specify a version of "backward selection" with the "method" parameter in the `regsubsets()` *function. Instead of looking at all models at each subset size, this*

- *First calculates RSS for the largest model, 7 predictors, in our case*
- *Then takes the 6-predictor model that increases RSS the least from the 7-predictor model*
- *Then takes the 5-predictor model that increases RSS the least from the 6-predictor model*
- *. . . and so forth down to the 1-predictor model*

We can then compare these seven models with AIC, BIC, etc.

```
backward <- regsubsets(log(PrizeMoney) ~ ., data=golf.red, method ="backward")

tmp <- summary(backward)
names(tmp)
```

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

```
attach(tmp)
results <- data.frame(which,bic=n*log(rss)+log(n)*(p+2),aic=n*log(rss)+2*(p+2))
detach()

results
```

```
##   X.Intercept. DrivingAccuracy  GIR PuttingAverage BirdieConversion SandSaves
## 1         TRUE           FALSE TRUE          FALSE            FALSE     FALSE
## 2         TRUE           FALSE TRUE          FALSE             TRUE     FALSE
## 3         TRUE           FALSE TRUE          FALSE             TRUE     FALSE
## 4         TRUE           FALSE TRUE          FALSE             TRUE      TRUE
## 5         TRUE           FALSE TRUE          FALSE             TRUE      TRUE
## 6         TRUE            TRUE TRUE          FALSE             TRUE      TRUE
## 7         TRUE            TRUE TRUE           TRUE             TRUE      TRUE
##   Scrambling PuttsPerRound      bic      aic
## 1      FALSE         FALSE 983.8286 973.9943
```

```
## 2      FALSE         FALSE 925.6317 912.5193
## 3       TRUE         FALSE 897.5903 881.1997
## 4       TRUE         FALSE 899.8881 880.2195
## 5       TRUE          TRUE 902.8157 879.8689
## 6       TRUE          TRUE 908.0047 881.7798
## 7       TRUE          TRUE 913.2780 883.7750
```

```
minimize(results,"bic")
```

```
##   X.Intercept. DrivingAccuracy  GIR PuttingAverage BirdieConversion SandSaves
## 3         TRUE           FALSE TRUE          FALSE             TRUE     FALSE
##   Scrambling PuttsPerRound      bic      aic
## 3       TRUE         FALSE 897.5903 881.1997
```

```
minimize(results,"aic")
```

```
##   X.Intercept. DrivingAccuracy  GIR PuttingAverage BirdieConversion SandSaves
## 5         TRUE           FALSE TRUE          FALSE             TRUE      TRUE
##   Scrambling PuttsPerRound      bic      aic
## 5       TRUE          TRUE 902.8157 879.8689
```

We see that for "backward selection"

- *The optimal BIC model is*
  `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling`
- *The optimal AIC model is*
  `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling + SandSaves + PuttsPerRound`

Problem 2(c)

Identify the optimal model or models based on AIC and BIC from the approach based on forward selection.

We can specify a version of "forward selection" with the "method" parameter in the `regsubsets()` *function. Instead of looking at all models at each subset size, this*

- *First finds the 1-predictor model with the smallest RSS*
- *Then takes the 2-predictor model that decreases RSS the most from the 1-predictor model*
- *Then takes the 3-predictor model that decreases RSS the most from the 2-predictor model*
- *. . . and so forth up to the 7-predictor model*

We can then compare these seven models with AIC, BIC, etc.

```
forward <- regsubsets(log(PrizeMoney) ~ ., data=golf.red, method ="forward")

tmp <- summary(forward)
names(tmp)
```

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

```
attach(tmp)
results <- data.frame(which,bic=n*log(rss)+log(n)*(p+2),aic=n*log(rss)+2*(p+2))
detach()

results
```

```
##   X.Intercept. DrivingAccuracy  GIR PuttingAverage BirdieConversion SandSaves
## 1         TRUE           FALSE TRUE          FALSE            FALSE     FALSE
## 2         TRUE           FALSE TRUE          FALSE            FALSE     FALSE
```

```
## 3         TRUE           FALSE TRUE          FALSE             TRUE     FALSE
## 4         TRUE           FALSE TRUE          FALSE             TRUE     FALSE
## 5         TRUE           FALSE TRUE          FALSE             TRUE      TRUE
## 6         TRUE            TRUE TRUE          FALSE             TRUE      TRUE
## 7         TRUE            TRUE TRUE           TRUE             TRUE      TRUE
##   Scrambling PuttsPerRound      bic      aic
## 1      FALSE         FALSE 983.8286 973.9943
## 2      FALSE          TRUE 914.4027 901.2902
## 3      FALSE          TRUE 902.1169 885.7263
## 4       TRUE          TRUE 900.1392 880.4705
## 5       TRUE          TRUE 902.8157 879.8689
## 6       TRUE          TRUE 908.0047 881.7798
## 7       TRUE          TRUE 913.2780 883.7750
```

```
minimize(results,"bic")
```

```
##   X.Intercept. DrivingAccuracy  GIR PuttingAverage BirdieConversion SandSaves
## 4         TRUE           FALSE TRUE          FALSE             TRUE     FALSE
##   Scrambling PuttsPerRound      bic      aic
## 4       TRUE          TRUE 900.1392 880.4705
```

```
minimize(results,"aic")
```

```
##   X.Intercept. DrivingAccuracy  GIR PuttingAverage BirdieConversion SandSaves
## 5         TRUE           FALSE TRUE          FALSE             TRUE      TRUE
##   Scrambling PuttsPerRound      bic      aic
## 5       TRUE          TRUE 902.8157 879.8689
```

We see that for "forward selection"

- *The optimal BIC model is*
  `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling + PuttsPerRound`
- *The optimal AIC model is*
  `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling + SandSaves + PuttsPerRound`

Problem 2(d)

Carefully explain why the models chosen in (a) & (c) are not the same while those in (a) and (b) are the same.

Here are the results that I got from model selection:

Best Models from 2(a) "all subsets":

- *BIC:* `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling`
- *AIC:* `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling + SandSaves + PuttsPerRound`

Best Models from 2(b)"backward":

- *BIC:* `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling`
- *AIC:* `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling + SandSaves + PuttsPerRound`

Best Models from 2(c) "forward":

- *BIC:* `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling + PuttsPerRound`
- *AIC:* `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling + SandSaves + PuttsPerRound`

We know that the "all subsets" models are the best of all possible models on the seven variables we have to work with, whereas the "backward" and "forward" models come from heuristics that may or may not find the best of all possible models on these seven variables.

- *The BIC model for "backward" is the same as for "all subsets" because the "backward" selection method stumbled on the best overall model for BIC (look at line 3 of the "results" data frame in parts (a) and (b) to verify this).*
- *The BIC model for "forward" selection is different because the "forward" heuristic didn't identify the best overall model for BIC (look at line 3 of the "results" output in part (a), vs line 4 of the "results" output in part (c) to verify this. Note also that the "line 3" model in part (c) is* `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + PuttsPerRound` *instead of the BIC-optimal model* `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling`.
- *The AIC models are the same in each case, because both the "forward" and the "backward" heuristics stumbled on the best overall model for AIC (look at line 5 of the "results" data frame in parts (a), (b) and (c) to verify this).*

Problem 2(e)

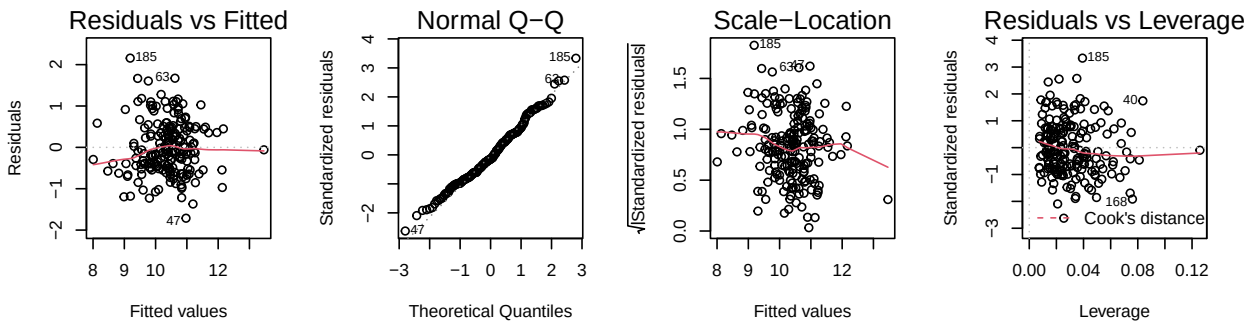
Recommend a final model. Give detailed reasons to support your choice.

There are really only two models to consider, from the model selection above. They are:

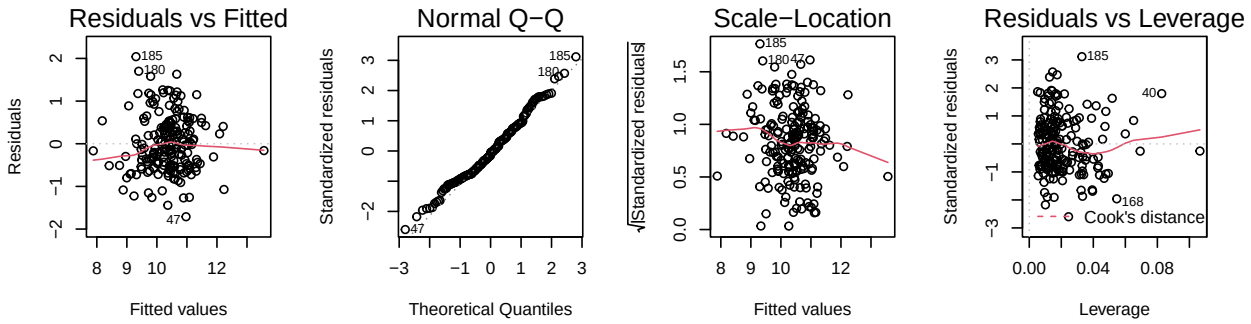
```
lm.AIC <- lm(log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling +
               SandSaves + PuttsPerRound, data=golf.red)

lm.BIC <- lm(log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling,data=golf.red)
```

Looking at the casewise diagnostic plots,

```
par(mfrow=c(1,4))
plot(lm.AIC)
```




```
plot(lm.BIC)
```



we see that both models seem to satisfy the assumptions of linear regression equally well.

Linearity: *There is no real trend in the Residuals vs Fitted plot for either model, so no transforms seem to be needed for either model.*

Normality: *The Normal Q-Q plots for both models look about the same.*

Constant Variance: *Except for "edge effects" the trend lines in both Scale-Location plots are pretty horizontal, supporting the idea of constant variance for both models.*

In addition the Residuals vs Leverage plots look about the same for both models.

Now let's look at summaries and, since the models are nested, a partial F-test to see whether the additional two variables in `lm.AIC` *are provide an improvement over* `lm.BIC`*.*

```
summary(lm.AIC)
```

```
##
## Call:
## lm(formula = log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling +
##     SandSaves + PuttsPerRound, data = golf.red)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71291 -0.48168 -0.09097  0.44843  2.15763
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.583181   7.158721  -0.081   0.9352
## GIR               0.197022   0.028711   6.862 9.31e-11 ***
## BirdieConversion  0.162752   0.032672   4.981 1.41e-06 ***
## Scrambling        0.049635   0.024738   2.006   0.0462 *
## SandSaves         0.015524   0.009743   1.593   0.1127
## PuttsPerRound    -0.349738   0.230995  -1.514   0.1317
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6606 on 190 degrees of freedom
## Multiple R-squared:  0.5575, Adjusted R-squared:  0.5459
## F-statistic: 47.88 on 5 and 190 DF,  p-value: < 2.2e-16
```

```
summary(lm.BIC)
```

```
##
## Call:
```

```
## lm(formula = log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling,
##     data = golf.red)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71081 -0.50717 -0.06683  0.41975  2.04147
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -11.08314    1.45712  -7.606 1.23e-12 ***
## GIR                0.15658    0.01787   8.761 1.01e-15 ***
## BirdieConversion   0.20625    0.02164   9.531  < 2e-16 ***
## Scrambling         0.09178    0.01539   5.965 1.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6661 on 192 degrees of freedom
## Multiple R-squared:  0.5453, Adjusted R-squared:  0.5382
## F-statistic: 76.75 on 3 and 192 DF,  p-value: < 2.2e-16
```

```
anova(lm.BIC,lm.AIC)
```

```
## Analysis of Variance Table
##
## Model 1: log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling
## Model 2: log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling + SandSaves +
##     PuttsPerRound
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    192 85.191
## 2    190 82.905  2    2.2858 2.6193 0.07548 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both the t-statistics in `summary(lm.AIC)` *and the partial F-test suggest that neither* `SandSaves` *nor* `PuttsPerRound` *is needed.*

Conclusion: *The smaller model,* `lm.BIC`*,* `log(PrizeMoney) ~ 1 + GIR + BirdieConversion + Scrambling`*, seems to be the preferable model.*

Important Note: *Because we are repeatedly using the data, for variable selection, and to conduct t-tests and an F-test, the results of the tests may be misleading because of capitalization on chance. The usual result of testing "significance'' after model selection on the same data is that the results are more significant than they should be.*

In this case, we are seeing non-significance for the two variables `SandSaves` *and* `PuttsPerRound`*, despite this tendency toward inflated significance. So these two variables probably really are non-significant predictors.*

Problem 2(f)

Interpret the regression coefficients in the final model. Is it necessary to be cautious about taking these results to literally?

The final model is `lm.BIC`*.*

Since we are considering `log(PrizeMoney)` *we can interpret coefficient estimates in* `summary(lm.BIC)` *in terms of percent change in prize money:*

- *An increase of 1 unit in* `GIR` *(1 extra percent of the time the player was able to hit the greens in regulation) is associated with a* $100 \times 0.15658\% \approx 16\%$ *increase in* `PrizeMoney`.
- *An increase of 1 unit in* `BirdieConversion` *(1 extra percent of the time the player makes a "birdie" or better after hitting the green in regulation) is associated with a* $100 \times 0.20625\% \approx 21\%$ *increase in* `PrizeMoney`.
- *An increase of 1 unit in* `Scrambling` *(1 extra percent of the time the player misses the green in regulation but still makes "par") is associated with a* $100 \times 0.09178\% \approx 9\%$ *increase in* `PrizeMoney`.

So it looks like all three of `GIR`, `BirdieConversion` *and* `Scrambling` *are importantly associated with a golfer's prize money. We don't know if increases in* `BirdieConversion` *and* `Scrambling` *cause a player to earn more prize money, or if there is a third lurking variable (like natural talent) that influences all three variables,* `BirdieConversion`, `Scrambling` *and* `PrizeMoney`.

Important Note: *All three variables have small SE's and are highly significant, but just like with testing after variable selection, calculating SE's after variable selection usually results in SE's that are too small, and so predictors look more significant than they actually are.*

In this case, all the p-values are very very small, so we might guess that these variable probably really are significant predictors.

Problem 3: [Based on Gelman & Hill (2009), p. 51, #5]

The subfolder `beauty` in the hw04 folder in the "Files" area for our course on canvas contains data from Hamermesh and Parker (2005) on student evaluations of instructors' beauty and teaching quality for several courses at the University of Texas. The teaching evaluations were conducted at the end of the semester, and the beauty judgments were made later, by six students who had not attended the classes and were not aware of the course evaluations. Various documents in the folder give background and some variable definitions (some variables are defined in the ".log" file there, others' definitions you will have to deduce from pdf's in the subfolder).

Problem 3(a)

Fit a regression model predicting `courseevaluation` (average student evaluations) from `btystdave` (the average of 6 standardized beauty ratings for each instructor) and `female`. Then fit the same model with the interaction between `btystdave` and `female` added in.

i Graph each fitted model on a scatter plot of `courseevaluation` vs `btystdave`. Indicate clearly in the graph what the various parameters in the model represent geometrically.

ii Display the four standard diagnostic plots in R and comment on their features, for each model. Comment on whether the fit seems adequate from the evidence in these plots, for either model. In case there are problems with the fit, indicate what they are and how you might improve things.

iii Produce summaries of the two fitted models; comment on the coefficient estimates and their standard errors, and on R^2, for each model Use a partial F test to determine whether the interaction should be kept. *Your comments should include not only technical points ("B" in the "ABA^{-1}" metaphor for applied statistics from the course syllabus), but also what it means for understanding how factors may influence course evaluations ("A^{-1}").*

Part (i):

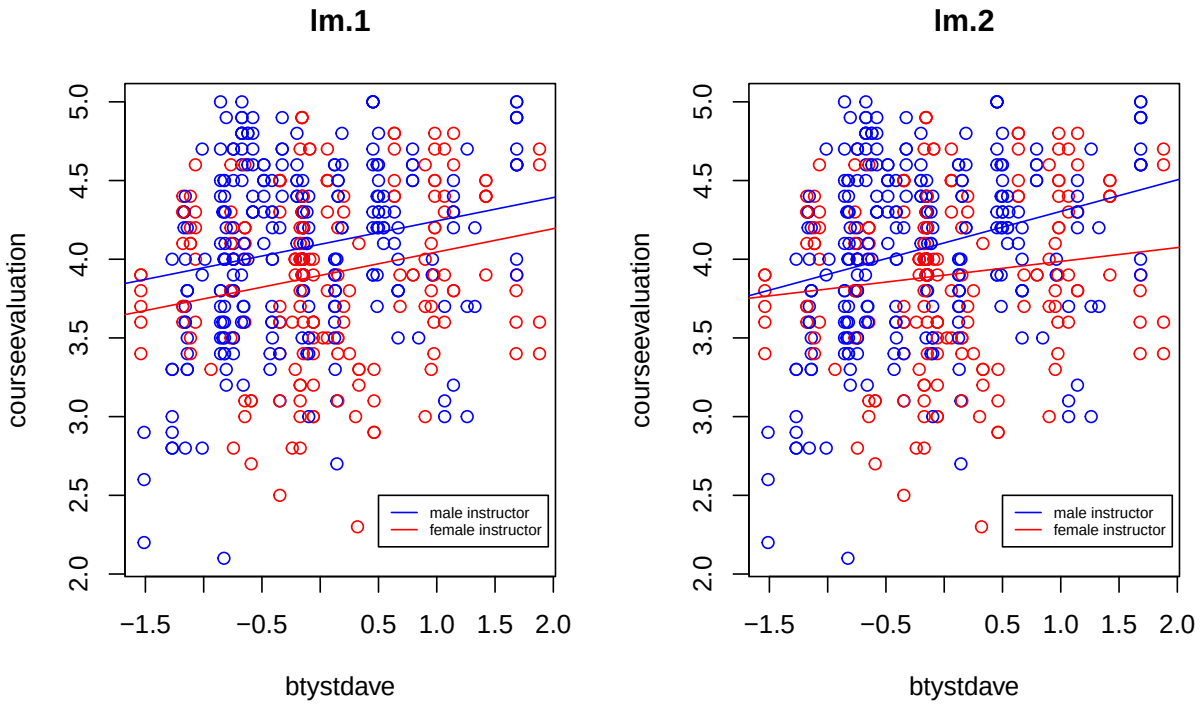
```
beauty <- read.csv("ProfEvaltnsBeautyPublic.csv")
## str(beauty) ## too long to include here...
lm.1 <- lm(courseevaluation ~ btystdave + female,data=beauty)
print(beta.lm.1 <- coef(lm.1))
```

```
## (Intercept)   btystdave      female
##   4.0947104   0.1485876  -0.1978096
```

```
lm.2 <- lm(courseevaluation ~ btystdave * female,data=beauty)
print(beta.lm.2 <- coef(lm.2))
```

```
##      (Intercept)        btystdave           female btystdave:female
##        4.1036435        0.2002743       -0.2050501       -0.1126579
```

```
par(mfrow=c(1,2))
plot(courseevaluation ~ btystdave, data=beauty,main="lm.1",
     col=c("blue","red")[beauty$female+1])
abline(beta.lm.1[1],beta.lm.1[2],col="blue")
abline(beta.lm.1[1]+beta.lm.1[3],beta.lm.1[2],col="red")
legend(0.5,2.5,lty=1,col=c("blue","red"),
       legend=c("male instructor","female instructor"),cex=0.55)
plot(courseevaluation ~ btystdave, data=beauty,main="lm.2",
     col=c("blue","red")[beauty$female+1])
abline(beta.lm.2[1],beta.lm.2[2],col="blue")
abline(beta.lm.2[1]+beta.lm.2[3],beta.lm.2[2]+beta.lm.2[4],col="red")
legend(0.5,2.5,lty=1,col=c("blue","red"),
       legend=c("male instructor","female instructor"),cex=0.55)
```



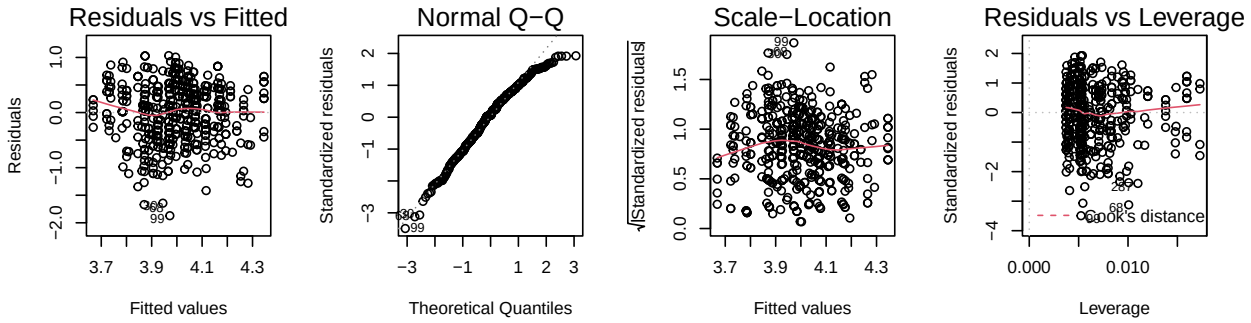
For the additive model (`lm.1`*, the plot on the left), the slope of the line relating* `courseevaluation` *to* `btystdave` *is 0.15, for both male and female instructors. The intercept for male instructors is 4.09, and for female instructors it's 3.9. In this model, male instructors get about a 0.2 boost in course evaluations, vs. female instructors.*

For the interactive model (`lm.2`*, the plot on the right), the slope of the line relating* `courseevaluation` *to* `btystdave` *is 0.2 for male instructors and 0.09 for female instructors. The intercept for male instructors is 4.1 whereas for female instructors it is 3.9. Thus according to this model, evaluations for courses with male*

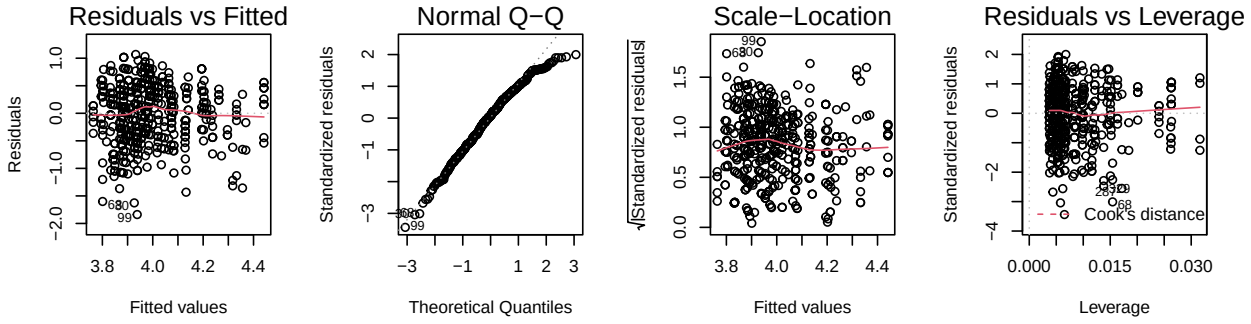
instructors benefit by about 0.21 to start with, and the slope as a function of `btystave` *is about 0.11 higher for male instructors, than for female instructors.*

Part (ii):

```
par(mfrow=c(1,4))
plot(lm.1)
```




```
plot(lm.2)
```



There is very little difference between the two sets of casewise diagnostic plots. In both models, there is no functional pattern in the raw residual plot, slightly short tails in the residuals in the Normal Q-Q plots, approximately constant variance in the Scale-Location plots, and very similar, and unconcerning residuals vs. leverage plots.

So, the fits of both models seem adequate (except perhaps for short tails in the Normal Q-Q plots, which don't concern me very much), and I would not do any transformations, etc. to improve either fit.

About the only difference I see is that the dsitribution of fitted values appear to be slightly higher for `lm.2` *than* `lm.1` *(e.g. see the x-axis of the residuals vs fitted plots).*

Part (iii):

```
summary(lm.1)
```

```
##
## Call:
## lm(formula = courseevaluation ~ btystdave + female, data = beauty)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.87196 -0.36913  0.03493  0.39919  1.03237
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.09471    0.03328  123.03  < 2e-16 ***
## btystdave    0.14859    0.03195    4.65 4.34e-06 ***
## female      -0.19781    0.05098   -3.88  0.00012 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5373 on 460 degrees of freedom
## Multiple R-squared:  0.0663, Adjusted R-squared:  0.06224
## F-statistic: 16.33 on 2 and 460 DF,  p-value: 1.407e-07
```

```
summary(lm.2)
```

```
##
## Call:
## lm(formula = courseevaluation ~ btystdave * female, data = beauty)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.83820 -0.37387  0.04551  0.39876  1.06764
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.10364    0.03359 122.158  < 2e-16 ***
## btystdave          0.20027    0.04333   4.622 4.95e-06 ***
## female            -0.20505    0.05103  -4.018 6.85e-05 ***
## btystdave:female  -0.11266    0.06398  -1.761   0.0789 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5361 on 459 degrees of freedom
## Multiple R-squared:  0.07256,    Adjusted R-squared:  0.0665
## F-statistic: 11.97 on 3 and 459 DF,  p-value: 1.471e-07
```

```
anova(lm.1,lm.2)
```

```
## Analysis of Variance Table
##
## Model 1: courseevaluation ~ btystdave + female
## Model 2: courseevaluation ~ btystdave * female
##   Res.Df    RSS Df Sum of Sq     F  Pr(>F)
## 1    460 132.81
## 2    459 131.92  1   0.89124 3.101 0.07891 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The most noticable thing about both models is that their R^2's are pitiful: 0.066 for `lm.1` *and 0.073 for* `lm.2`*. So, even though we have some significant predictors in both models, they are not doing a very good job of accounting for the variation in* `courseevaluation`*.*

In both models, `btystave` *and* `female` *are highly significant predictors, and in* `lm.2`*, the interaction term* `btystdave:female` *is just barely non-significant (p-value = 0.0789).*

Because we are only adding one more variable, the interaction term, the F-test statistic is just the square of the t-statistic for the interaction in the `summary()` *output, and hence there is no new information: the p-value*

is again 0.0789. So, we would probably prefer the simpler model `lm.1`*.*

In practical terms, model `lm.1` *says that a course with a male instructor starts off with a course evaluation of about 4.09, but this drops to about 3.9 if the course has a female instructor. The increase in course evaluation per unit increase in beauty rating is the same for both male and female instructors, about 0.15.*

The standard errors for the coefficients are all quite small (but perhaps a bit smaller than they should be since we used the data twice: once to select a model and once to estimate coefficients), and generally they are reassuring that the effects quoted in the last paragraph are "real".

Problem 3(b)

Now let's look at *all* of the variables in the data set. Should any of the variables in the data set be transformed before being used in a regression model? List each variable that is not a dummy variable, and for each of these,

- Say whether the variable should be transformed (yes or no)
- If yes, indicate what transformation you would make
- Justify these two answers, using both evidence from the data and other considerations

Note: being able to communicate with a client or collaborator matters, so there may be instances where either (a) a transformation might help, but you decide against it since it would be difficult to explain to a client/collaborator, or (b) an automatic method like Box-Cox might suggest one power, but you pick a simpler power "nearby'' because it is easier to explain to a collaborator/client.

Since we're not going to consider `profnumber`*,* `multipleclass` *and* `class1` *through* `class30` *in part (c), I will eliminate them now before we consider transformations.*

I will use Box-Cox to suggest transformations, and then choose more interpretable transformations based on Box-Cox.

```
names(beauty)
```

```
##  [1] "tenured"           "profnumber"        "minority"
##  [4] "age"               "beautyf2upper"     "beautyflowerdiv"
##  [7] "beautyfupperdiv"   "beautym2upper"     "beautymlowerdiv"
## [10] "beautymupperdiv"   "btystdave"         "btystdf2u"
## [13] "btystdfl"          "btystdfu"          "btystdm2u"
## [16] "btystdml"          "btystdmu"          "class1"
## [19] "class2"            "class3"            "class4"
## [22] "class5"            "class6"            "class7"
## [25] "class8"            "class9"            "class10"
## [28] "class11"           "class12"           "class13"
## [31] "class14"           "class15"           "class16"
## [34] "class17"           "class18"           "class19"
## [37] "class20"           "class21"           "class22"
## [40] "class23"           "class24"           "class25"
## [43] "class26"           "class27"           "class28"
## [46] "class29"           "class30"           "courseevaluation"
## [49] "didevaluation"     "female"            "formal"
## [52] "fulldept"          "lower"             "multipleclass"
## [55] "nonenglish"        "onecredit"         "percentevaluating"
## [58] "profevaluation"    "students"          "tenuretrack"
## [61] "blkandwhite"       "btystdvariance"    "btystdavepos"
## [64] "btystdaveneg"
```

```
prof.loc <- grep("profnumber",names(beauty))
multiclass.loc <- grep("multipleclass",names(beauty))
class.locs <- grep("class",names(beauty))
beauty.red <- beauty[,-c(prof.loc,multiclass.loc,class.locs)]
```

From the `str()` *command, our background knowledge of college, and some examination of the papers and variable glossary in the* `beauty` *subdirectory, we find*

tenured *dummy: instructor tenured? 0=no, 1=yes*
minority *dummy: instructor minority? 0=no, 1=yes*
age *continuous: 36 59 51 40 31 62 33 51 33 47 ...*
beautyf2upper *continuous: 6 2 5 4 9 5 5 6 5 6 ...*
beautyflowerdiv *continuous: 5 4 5 2 7 6 4 4 3 5 ...*
beautyfupperdiv *continuous: 7 4 2 5 9 6 4 6 7 7 ...*
beautym2upper *continuous: 6 3 3 2 6 6 4 3 5 6 ...*
beautymlowerdiv *continuous: 2 2 2 3 7 5 4 2 5 3 ...*
beautymupperdiv *continuous: 4 3 3 3 6 5 4 3 3 6 ...*
btystdave *continuous: 0.202 -0.826 -0.66 -0.766 1.421 ...*
btystdf2u *continuous: 0.289 -1.619 -0.188 -0.665 1.721 ...*
btystdfl *continuous: 0.458 -0.0735 0.458 -1.1365 1.521 ...*
btystdfu *continuous: 0.8758 -0.577 -1.5456 -0.0927 1.8444 ...*
btystdm2u *continuous: 0.682 -1.132 -1.132 -1.736 0.682 ...*
btystdml *continuous: -0.9 -0.9 -0.9 -0.313 2.038 ...*
btystdmu *continuous: -0.195 -0.655 -0.655 -0.655 0.723 ...*
courseevaluation *continuous: 4.3 4.5 3.7 4.3 4.4 4.2 4 3.4 4.5 3.9 ...*
didevaluation *continuous: 24 17 55 40 42 182 33 25 48 16 ...*
female *dummy: instructor female? 0=no, 1=yes*
formal *dummy: web pic of instructor wears tie/jacket (dress)? 0=no, 1=yes*
fulldept *dummy: everyone in dept has a web pic? 0=no, 1=yes*
lower *dummy: lower division (freshman/sophomre)? 0=no, 1=yes*
nonenglish *dummy: instructor non-native English speaker? 0=no, 1=yes*
onecredit *dummy: a one-credit course? 0=no, 1=yes*
percentevaluating *continuous: 55.8 85 100 87 87.5 ...*
profevaluation *continuous: 4.7 4.6 4.1 4.5 4.8 4.4 4.4 3.4 4.8 4 ...*
students *continuous: 43 20 55 46 48 282 41 41 60 19 ...*
tenuretrack *dummy: is instructor in tenure-track? 0=no, 1=yes*
blkandwhite *dummy: (don't know what this is!)*
btystdvariance *continuous: 2.13 1.39 2.54 1.76 1.69 ...*
btystdavepos *continuous: 0.202 0 0 0 1.421 ...*
btystdaveneg *continuous: 0 -0.826 -0.66 -0.766 0 ...*

There are a few relationships we can see or guess immediately:

```
attach(beauty.red)
sum(abs(btystdave - (btystdf2u+btystdfl+btystdfu+btystdm2u+btystdml+btystdmu)/6))
```

```
## [1] 4.693333e-05
```

```
sum(abs(btystdave - (btystdavepos+btystdaveneg)))
```

```
## [1] 0.0004629
```

```
detach()
```

So,up to rounding error, we see that

- `btystdave` *is in fact* `(btystdf2u+btystdfl+btystdfu+btystdm2u+btystdml+btystdmu)/6`
- `btystdave` *is also* `btystdavepos+btystdaveneg`

So we will keep `btystdave` *and eliminate the variables that sum up to it. (There is no loss in eliminating* `btystdavepos` *and* `btystdaveneg`*, since these are just the "positive part" and "negative part" of* `btystdave`*. There is a little loss in eliminating* `btystdf2u`*,* `btystdfl`*,* `btystdfu`*,* `btystdm2u`*,* `btystdml` *and* `btystdmu`*, but hopefully not too much since we will keep the unstandardized versions.)*

```
btystd.locs <- grep("btystd",names(beauty.red))[-c(1,8)]
## we are keeping btystdave, which comes first in this list,
## and btystdvariance, which is eighth...
beauty.red <- beauty.red[,-btystd.locs]
names(beauty.red)
```

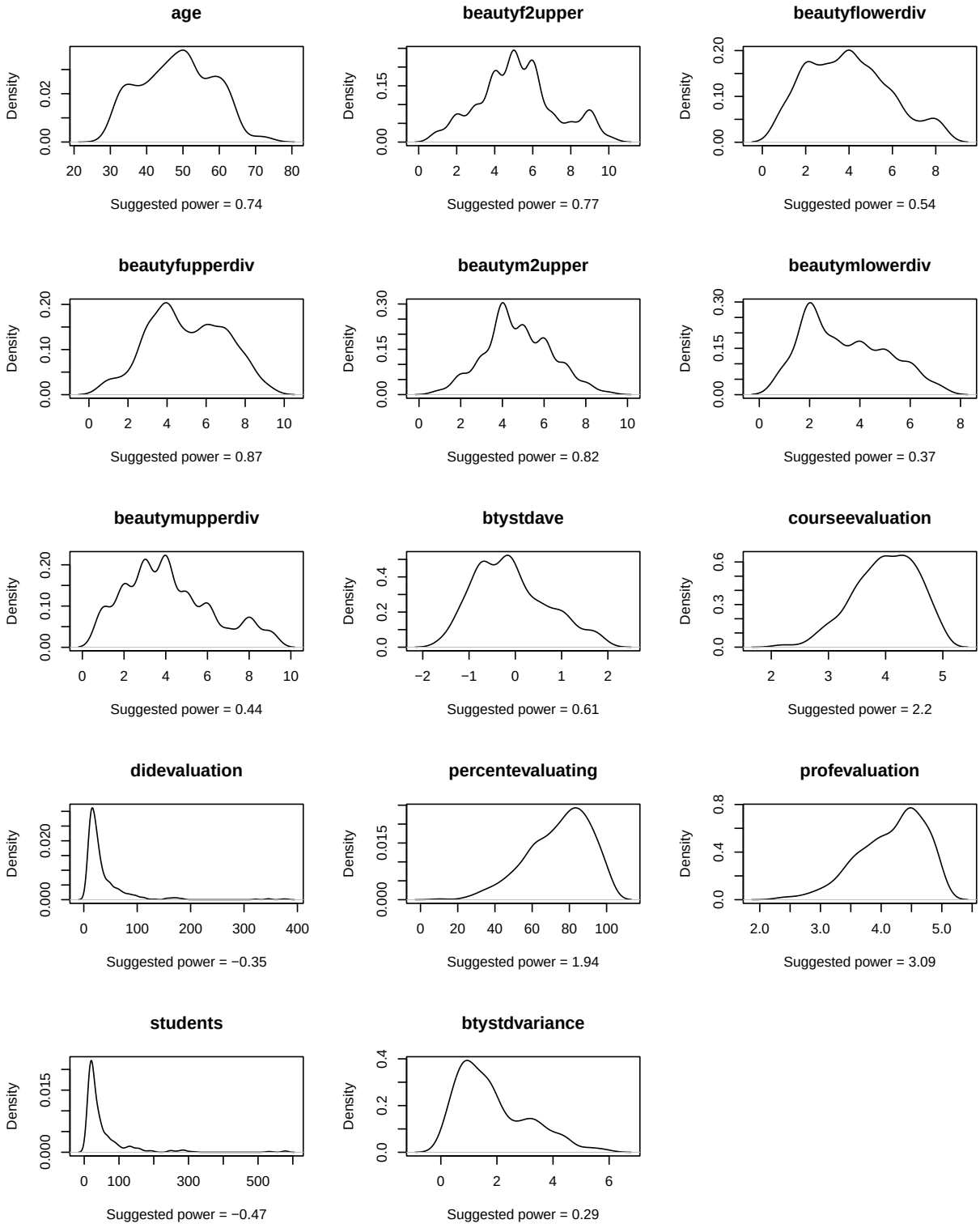
```
##  [1] "tenured"           "minority"          "age"
##  [4] "beautyf2upper"     "beautyflowerdiv"   "beautyfupperdiv"
##  [7] "beautym2upper"     "beautymlowerdiv"   "beautymupperdiv"
## [10] "btystdave"         "courseevaluation"  "didevaluation"
## [13] "female"            "formal"            "fulldept"
## [16] "lower"             "nonenglish"        "onecredit"
## [19] "percentevaluating" "profevaluation"    "students"
## [22] "tenuretrack"       "blkandwhite"       "btystdvariance"
```

The variables we want to consider for transformation are all of the non-dummies that are left:

```
trans.names <- names(beauty.red)[c(3:12,19:21,24)]

powers <- NULL
for (i in trans.names) {
  x <- beauty.red[,i]
  if (min(x)==0) {
    x <- x + 0.01*max(x)
    } else if (min(x)<0) {
      x <- x - min(x)*1.01
      }
  ## forces each variable to be strictly positive, for Box-Cox
  powers <- c(powers,powerTransform(x)$lambda)
}
names(powers) <- trans.names
powers <- round(powers,2)
powers
```

```
##              age     beautyf2upper   beautyflowerdiv   beautyfupperdiv
##             0.74              0.77              0.54              0.87
##     beautym2upper   beautymlowerdiv   beautymupperdiv         btystdave
##             0.82              0.37              0.44              0.61
##  courseevaluation     didevaluation percentevaluating    profevaluation
##             2.20             -0.35              1.94              3.09
##          students    btystdvariance
##            -0.47              0.29
```

```
par(mfrow=c(5,3))
for (i in trans.names) {
  plot(density(beauty.red[,i]),main=i,xlab=paste("Suggested power =",powers[i]))
}
```

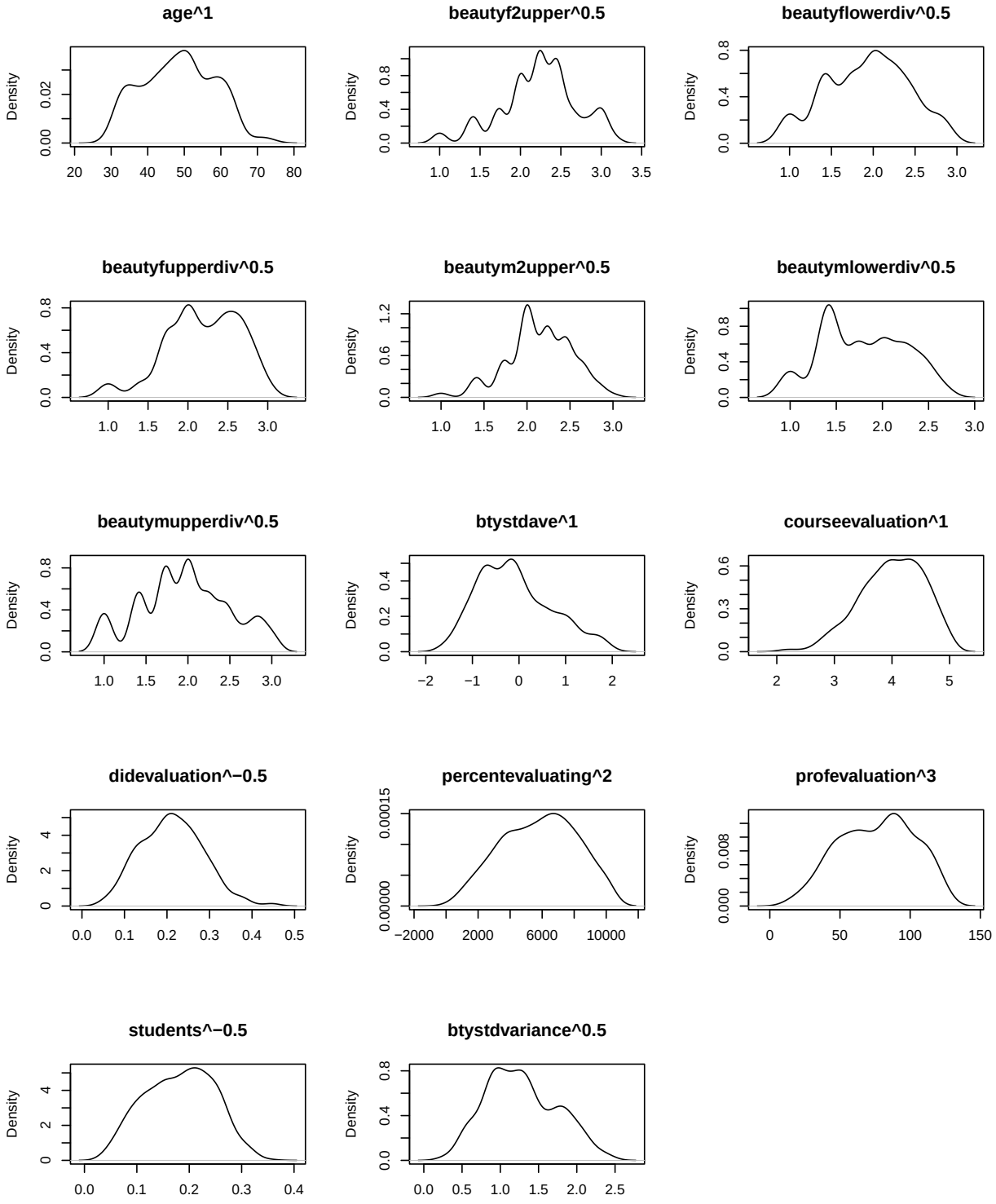


I'm going to round these suggested powers as follows:

```
bjpowers <- powers
bjpowers["age"] <- 1 ## already fairly symmetric
bjpowers["beautyf2upper"]   <- 0.5  ## right skewed, sqrt() easier to talk about
```

```
bjpowers["beautyflowerdiv"] <- 0.5  ## right skewed, sqrt() easier to talk about
bjpowers["beautyfupperdiv"] <- 0.5  ## right skewed, sqrt() easier to talk about
bjpowers["beautym2upper"]   <- 0.5  ## right skewed, sqrt() easier to talk about
bjpowers["beautymlowerdiv"] <- 0.5  ## right skewed, sqrt() easier to talk about
bjpowers["beautymupperdiv"] <- 0.5  ## right skewed, sqrt() easier to talk about
bjpowers["btystdave"] <- 1 ## some right skew but reasonably bounded between -2 and +2
bjpowers["courseevaluation"] <- 1   ## even though 2.2 is suggested, I'm leaving this
                                    ## alone, to make the linear regression coefficients
                                    ## more interpretable
bjpowers["didevaluation"]   <- -0.5 ## I may try a log here too.
bjpowers["percentevaluating"] <- 2  ## close, perhaps more interpretable
bjpowers["profevaluation"]   <- 3   ## I may try 2 later..
bjpowers["students"]        <- -0.5 ## I may try a log here too.
bjpowers["btystdvariance"]  <- 0.5  ## 0.29 is called for but hard to talk about
```

```
par(mfrow=c(5,3))
for (i in trans.names) {
  plot(density(beauty.red[,i]^bjpowers[i]),main=paste(i,bjpowers[i],sep="^"),xlab="")
}
```



We'll go with these transformations, though you can see in my comments above that some others might be better...

Problem 3(c)

Fit the model that regresses `courseevaluation` onto all other variables, except for `profnumber`, `multipleclass`, and the 30 `class` variables (`class1` through `class30`). Use the transformations you recommended in part (b). Make a table indicating

- The t-statistics for each variable
- The VIFs for each variable

in your model.

```
beauty.trans <- beauty.red
for (i in trans.names) {
  beauty.trans[,i] <- beauty.red[,i]^bjpowers[i]
  names(beauty.trans)[grep(i,names(beauty.trans))] <- paste("t",i,sep=".")
}
names(beauty.trans)[grep("t.btystdave",names(beauty.trans))] <- "btystdave"
## didn't transform btystdave...
names(beauty.trans)[grep("t.courseevaluation",names(beauty.trans))] <- "courseevaluation"
## didn't transform courseevaluation...
names(beauty.trans)[grep("t.age",names(beauty.trans))] <- "age"
## didn't transform age...
names(beauty.trans)
```

```
##  [1] "tenured"             "minority"            "age"
##  [4] "t.beautyf2upper"     "t.beautyflowerdiv"   "t.beautyfupperdiv"
##  [7] "t.beautym2upper"     "t.beautymlowerdiv"   "t.beautymupperdiv"
## [10] "btystdave"           "courseevaluation"    "t.didevaluation"
## [13] "female"              "formal"              "fulldept"
## [16] "lower"               "nonenglish"          "onecredit"
## [19] "t.percentevaluating" "t.profevaluation"    "t.students"
## [22] "tenuretrack"         "blkandwhite"         "t.btystdvariance"
```

```
lm.1 <- lm(courseevaluation ~ ., data=beauty.trans)
tab <- cbind(summary(lm.1)$coef, vif=c(NA,vif(lm.1)))

round(tab,4)
```

```
##                     Estimate Std. Error t value Pr(>|t|)      vif
## (Intercept)           2.8629     0.5349  5.3523   0.0000       NA
## tenured               0.0326     0.0307  1.0598   0.2898   2.7083
## minority             -0.0194     0.0324 -0.5994   0.5492   1.4469
## age                   0.0014     0.0014  0.9948   0.3204   2.1224
## t.beautyf2upper      -0.0285     0.0533 -0.5351   0.5928   7.0688
## t.beautyflowerdiv    -0.0282     0.0534 -0.5279   0.5978   7.8156
## t.beautyfupperdiv     0.0131     0.0481  0.2726   0.7853   5.6271
## t.beautym2upper      -0.0432     0.0633 -0.6825   0.4953   6.5295
## t.beautymlowerdiv    -0.0592     0.0576 -1.0282   0.3044   7.7566
## t.beautymupperdiv    -0.0298     0.0499 -0.5967   0.5510   8.0266
## btystdave             0.0996     0.1190  0.8367   0.4032 101.6065
## t.didevaluation      -0.7943     0.6439 -1.2335   0.2181  25.6187
## female               -0.0331     0.0233 -1.4186   0.1567   1.5344
## formal                0.0152     0.0287  0.5285   0.5974   1.3193
## fulldept             -0.0038     0.0364 -0.1055   0.9160   1.4497
## lower                 0.0051     0.0236  0.2163   0.8289   1.4452
## nonenglish           -0.0592     0.0464 -1.2765   0.2025   1.4141
```

```
## onecredit             0.0655     0.0497  1.3189   0.1879   1.5669
## t.percentevaluating   0.0000     0.0000 -0.3371   0.7362   5.2175
## t.profevaluation      0.0187     0.0004 46.5582   0.0000   1.3235
## t.students            1.2703     0.8063  1.5755   0.1159  31.9203
## tenuretrack          -0.0400     0.0340 -1.1783   0.2393   2.2934
## blkandwhite           0.0174     0.0313  0.5550   0.5792   1.5860
## t.btystdvariance     -0.0167     0.0235 -0.7090   0.4787   1.3595
```

Problem 3(d)

On the basis of this table, and what you know about the definitions of the variables, would you eliminate any variables in your model? Why or why not?

- *Since none of them are individually significant, all have moderately large vif's, and they are approximately summarized by* `btystdave`*, I would eliminate the six students' ratings* `t.beautyf2upper, t.beautyflowerdiv, t.beautyfupperdiv, t.beautym2upper, t.beautymlowerdiv` *and* `t.beautymupperdiv`*.*
- *Even though* `t.profevaluation` *is highly significant with a low vif, I would probably remove it. In my experience, evalation of the instructor and evaluation of the course are highly correlated, and so this variable is probably soaking up variation in course evaluation that I'd like to see explained by other variables in the model.*
- *The variables* `t.didevaluation, t.percentevaluating`*, and* `t.students` *are interesting: In untransformed form there is clear (nonlinear) relationship among these variables:* `percentevaluating = 100*didevaluation/students`*. Since* `t.percentevaluating` *has almost no practical effect on course evaluation, I will eliminate that one, and see what the effect is.*
- *I would not get rid of* `btystave` *even though it has a sky-high vif and is only marginally significant: I'd hope that removing the other variables above would allow us to see the effect of* `btystdave` *on course evaluation.*

In a "first pass", this is all the farther I'd go; removing these variables is going to strongly affect the t statistics and the vifs. If I refit the model without these variables, I get

```
lm.2 <- update(lm.1, . ~ . - t.beautyf2upper - t.beautyflowerdiv - t.beautyfupperdiv -
                 t.beautym2upper - t.beautymlowerdiv - t.beautymupperdiv -
                 t.profevaluation - t.percentevaluating)
tab <- cbind(summary(lm.2)$coef, vif=c(NA,vif(lm.2)))

round(tab,4)
```

```
##                 Estimate Std. Error t value Pr(>|t|)    vif
## (Intercept)       4.1781     0.2056 20.3167   0.0000     NA
## tenured           0.0702     0.0735  0.9552   0.3400 2.5303
## minority         -0.1584     0.0780 -2.0313   0.0428 1.3673
## age              -0.0073     0.0032 -2.2880   0.0226 1.8399
## btystdave         0.0915     0.0338  2.7080   0.0070 1.3384
## t.didevaluation  -1.5430     0.8365 -1.8446   0.0658 7.0660
## female           -0.1825     0.0528 -3.4591   0.0006 1.2819
## formal            0.1426     0.0691  2.0635   0.0396 1.2507
## fulldept          0.1997     0.0842  2.3717   0.0181 1.2669
## lower             0.0181     0.0568  0.3189   0.7500 1.3633
## nonenglish       -0.3116     0.1110 -2.8065   0.0052 1.3229
## onecredit         0.5339     0.1155  4.6245   0.0000 1.3827
## t.students        2.6575     0.9420  2.8210   0.0050 7.1209
## tenuretrack      -0.1461     0.0818 -1.7874   0.0746 2.1684
```

```
## blkandwhite        0.2023     0.0717  2.8225   0.0050 1.3589
## t.btystdvariance  -0.0437     0.0528 -0.8281   0.4081 1.1185
```

Things are starting to come into focus:

- `bytstdave` *is a significant predictor now*
- *Most of the predictors are significant or very nearly significant; the exceptions are* `tenured`, `lower` *and* `t.btystdvariance`. *(I would probably try to remove these in a further round of variable selection...)*
- *The variables* `t.didevaluation` *and* `t.students` *still have somewhat high vifs, but both are significant or nearly so. I might try to take one of these out (I think they are just collinear with each other: their correlation is 0.92), or I might try to leave them in and see if the rather large effects they have are "real".*
- *The other variables have coefficients with signs that we might expect:*
  - *minority instructors have somewhat lower course evaluations ($\hat{\beta}_{minority} = -0.1584$)*
  - *each year older an instructor is, the course evaluation lowers a bit ($\hat{\beta}_{age} = -0.0073$)*
  - *increasing beauty rating has a somewhat positive effect on course rating ($\hat{\beta}_{btystdave} = -0.0915$)*
  - *keeping in mind that* `t.didevaluation = 1/sqrt(didevaluation)`, *the fewer students that evaluate the class, the lower the course rating ($\hat{\beta}_{t.didevaluation} = -1.5430$)*
  - *similarly, since* `t.students = 1/sqrt(students)`, *the fewer students in the class, the higher the course rating ($\hat{\beta}_{t.students} = 2.6575$)*
  - *female instructors take a hit a bit larger than minority instructors in course evaluation ($\hat{\beta}_{female} = -0.1825$)*
  - *If the instructor dresses formally for their web photo, or if the department faculty all have web photos, course evaluation increases enough to essentially offset the effect of being female or minority ($\hat{\beta}_{formal} = 0.1426$, $\hat{\beta}_{fulldept} = 0.1997$)*
  - *If the instructor is not a native speaker of English, the course evaluation takes a hit that's about twice as big as the minority instructor hit ($\hat{\beta}_{nonenglish} = -0.3116$)*
  - *One credit course get over a half-point advantage in course evaluation ($\hat{\beta}_{onecredit} = 0.5339$)*
  - *We have no idea what* `blkandwhite` *is, but it has a positive effect on course evaluation! ($\hat{\beta}_{onecredit} = 0.2023$)*

  Of course, all the interpretations above sound like we are saying each predictor "causes" a change in course evaluation, but we really don't know about lurking variables, common causes, etc., so we can't be sure of "causes" here.

Problem 3(e)

Why might the methods used in parts (c) and (d) not be adequate for deciding which variables to keep, and which ones to eliminate, in a regression model?

Essentially, the question is, "what could go wrong with selecting variables on the basis of t-statistics and vif's?"

There are a few potential difficulties with this approach:

1. *Removing variables one-at-a-time can change other coefficients and t-statistics, so if you remove the variables in one order based on t-statistics, and I remove them in another order, we could end up with very different models.*

2. *High vif's do not mean that the model is invalid; they just mean that the estimated coefficients will be hard to interpret. Removing variables on the basis of vif's can remove important predictor variables from a model*

3. *One-variable-at-a-time approaches are examples of "greedy" algorithms: since they do not consider all variables at each step of variable selection, they can miss the very best models (forward selection with BIC in Problem #2 is an example of this).*

In the case of this problem, these dificulties are offset by "subject matter knowledge": since we know a lot about college or university life, we can make better guesses about what order to remove variables in, what groups of variables to remove all at once, etc.

Just for kicks, I tried "all subsets" variable selection on the model fitted in part (c) here (except that I took `profevaluation` *out, because of its close relationship with* `courseevaluation`*). This approach uses no subject matter knowledge (none of our knowledge about university life!), and replaces it with just mathematical optimality:*

```
all.subsets <- regsubsets(courseevaluation ~ . - t.profevaluation, data=beauty.trans,
                          nvmax=dim(beauty.trans)-1)  # -1 for t.profevaluation
tmp <- summary(all.subsets)
attach(tmp)
p <- 1:dim(tmp$which)[1]
n <- dim(beauty.trans)[1]
results <- data.frame(which, bic=n*log(rss)+log(n)*(p+2), aic=n*log(rss)+2*(p+2))
detach()
```

```
oldwidth <- options()$width
options(width=200)
results
```

```
##    X.Intercept. tenured minority   age t.beautyf2upper t.beautyflowerdiv t.beautyfupperdiv t.beautym2upper t.beautymlowerdiv t.beautymupperdiv btystdave t.didevaluation female formal fulldept lower
## 1          TRUE   FALSE    FALSE FALSE           FALSE             FALSE            FALSE           FALSE             FALSE             FALSE     FALSE           FALSE  FALSE  FALSE    FALSE FALSE
## 2          TRUE   FALSE    FALSE FALSE           FALSE             FALSE             TRUE           FALSE             FALSE             FALSE     FALSE           FALSE  FALSE  FALSE    FALSE FALSE
## 3          TRUE   FALSE    FALSE FALSE           FALSE             FALSE             TRUE           FALSE             FALSE             FALSE     FALSE           FALSE   TRUE  FALSE    FALSE FALSE
## 4          TRUE   FALSE    FALSE FALSE           FALSE             FALSE             TRUE           FALSE             FALSE             FALSE     FALSE           FALSE   TRUE  FALSE    FALSE FALSE
## 5          TRUE   FALSE     TRUE FALSE           FALSE             FALSE             TRUE           FALSE             FALSE             FALSE     FALSE           FALSE   TRUE  FALSE    FALSE FALSE
## 6          TRUE   FALSE    FALSE FALSE           FALSE             FALSE             TRUE           FALSE             FALSE             FALSE     FALSE           FALSE   TRUE  FALSE    FALSE FALSE
## 7          TRUE   FALSE     TRUE FALSE           FALSE             FALSE             TRUE           FALSE             FALSE             FALSE     FALSE           FALSE   TRUE  FALSE    FALSE FALSE
## 8          TRUE   FALSE     TRUE FALSE           FALSE             FALSE             TRUE            TRUE             FALSE             FALSE     FALSE           FALSE   TRUE  FALSE    FALSE FALSE
## 9          TRUE   FALSE     TRUE FALSE            TRUE             FALSE             TRUE            TRUE             FALSE             FALSE     FALSE           FALSE   TRUE  FALSE    FALSE FALSE
## 10         TRUE   FALSE     TRUE FALSE            TRUE             FALSE             TRUE            TRUE             FALSE             FALSE     FALSE            TRUE   TRUE  FALSE    FALSE FALSE
## 11         TRUE   FALSE     TRUE FALSE            TRUE             FALSE             TRUE            TRUE             FALSE             FALSE     FALSE            TRUE   TRUE  FALSE    FALSE FALSE
## 12         TRUE   FALSE     TRUE FALSE            TRUE             FALSE             TRUE            TRUE             FALSE             FALSE     FALSE            TRUE   TRUE  FALSE     TRUE FALSE
## 13         TRUE    TRUE     TRUE  TRUE            TRUE             FALSE             TRUE            TRUE             FALSE             FALSE     FALSE            TRUE   TRUE  FALSE    FALSE FALSE
## 14         TRUE    TRUE     TRUE  TRUE            TRUE             FALSE             TRUE            TRUE             FALSE             FALSE     FALSE            TRUE   TRUE   TRUE    FALSE FALSE
## 15         TRUE    TRUE     TRUE  TRUE            TRUE              TRUE             TRUE           FALSE             FALSE             FALSE      TRUE            TRUE   TRUE   TRUE    FALSE FALSE
## 16         TRUE    TRUE     TRUE  TRUE            TRUE              TRUE             TRUE           FALSE             FALSE              TRUE      TRUE            TRUE   TRUE   TRUE    FALSE FALSE
## 17         TRUE    TRUE     TRUE  TRUE            TRUE              TRUE             TRUE           FALSE             FALSE              TRUE      TRUE            TRUE   TRUE   TRUE     TRUE FALSE
## 18         TRUE    TRUE     TRUE  TRUE            TRUE              TRUE             TRUE           FALSE             FALSE              TRUE      TRUE            TRUE   TRUE   TRUE     TRUE  TRUE
## 19         TRUE    TRUE     TRUE  TRUE            TRUE              TRUE             TRUE           FALSE              TRUE              TRUE      TRUE            TRUE   TRUE   TRUE     TRUE  TRUE
## 20         TRUE    TRUE     TRUE  TRUE            TRUE              TRUE             TRUE           FALSE              TRUE              TRUE      TRUE            TRUE   TRUE   TRUE     TRUE  TRUE
## 21         TRUE    TRUE     TRUE  TRUE            TRUE              TRUE             TRUE           FALSE              TRUE              TRUE      TRUE            TRUE   TRUE   TRUE     TRUE  TRUE
## 22         TRUE    TRUE     TRUE  TRUE            TRUE              TRUE             TRUE            TRUE              TRUE              TRUE      TRUE            TRUE   TRUE   TRUE     TRUE  TRUE
##    nonenglish onecredit t.percentevaluating t.students tenuretrack blkandwhite t.btystdvariance      bic      aic
## 1       FALSE      TRUE               FALSE      FALSE       FALSE       FALSE            FALSE 2287.381 2274.968
## 2       FALSE      TRUE               FALSE      FALSE       FALSE       FALSE            FALSE 2268.701 2252.150
## 3       FALSE      TRUE               FALSE      FALSE       FALSE       FALSE            FALSE 2254.834 2234.145
## 4       FALSE      TRUE                TRUE      FALSE       FALSE       FALSE            FALSE 2245.415 2220.588
## 5       FALSE      TRUE                TRUE      FALSE       FALSE       FALSE            FALSE 2235.815 2206.851
## 6        TRUE      TRUE                TRUE      FALSE       FALSE        TRUE            FALSE 2229.402 2196.300
## 7        TRUE      TRUE                TRUE      FALSE       FALSE        TRUE            FALSE 2229.766 2192.526
## 8        TRUE      TRUE                TRUE      FALSE       FALSE        TRUE            FALSE 2233.035 2191.658
## 9        TRUE      TRUE                TRUE      FALSE       FALSE        TRUE            FALSE 2236.075 2190.560
## 10       TRUE      TRUE                TRUE      FALSE       FALSE        TRUE            FALSE 2238.882 2189.229
## 11       TRUE      TRUE                TRUE      FALSE        TRUE        TRUE            FALSE 2243.019 2189.229
## 12       TRUE      TRUE                TRUE      FALSE        TRUE        TRUE            FALSE 2247.324 2189.396
## 13       TRUE      TRUE                TRUE      FALSE        TRUE        TRUE            FALSE 2251.593 2189.528
## 14       TRUE      TRUE                TRUE      FALSE        TRUE        TRUE            FALSE 2255.547 2189.344
## 15       TRUE      TRUE                TRUE      FALSE        TRUE        TRUE            FALSE 2259.621 2189.280
## 16       TRUE      TRUE                TRUE      FALSE        TRUE        TRUE            FALSE 2264.658 2190.179
## 17       TRUE      TRUE                TRUE      FALSE        TRUE        TRUE            FALSE 2270.045 2191.428
## 18       TRUE      TRUE                TRUE      FALSE        TRUE        TRUE            FALSE 2275.607 2192.852
## 19       TRUE      TRUE                TRUE      FALSE        TRUE        TRUE            FALSE 2281.385 2194.493
## 20       TRUE      TRUE                TRUE       TRUE        TRUE        TRUE            FALSE 2287.338 2196.308
## 21       TRUE      TRUE                TRUE       TRUE        TRUE        TRUE             TRUE 2293.425 2198.257
## 22       TRUE      TRUE                TRUE       TRUE        TRUE        TRUE             TRUE 2299.528 2200.222
```

```
minimize(results,"bic")
```

```
##   X.Intercept. tenured minority   age t.beautyf2upper t.beautyflowerdiv t.beautyfupperdiv t.beautym2upper t.beautymlowerdiv t.beautymupperdiv btystdave t.didevaluation female formal fulldept lower
## 6         TRUE   FALSE    FALSE FALSE           FALSE             FALSE             TRUE           FALSE             FALSE             FALSE     FALSE           FALSE   TRUE  FALSE    FALSE FALSE
##   nonenglish onecredit t.percentevaluating t.students tenuretrack blkandwhite t.btystdvariance      bic    aic
## 6       TRUE      TRUE                TRUE      FALSE       FALSE        TRUE            FALSE 2229.402 2196.3
```

```
minimize(results,"aic")
```

```
##    X.Intercept. tenured minority   age t.beautyf2upper t.beautyflowerdiv t.beautyfupperdiv t.beautym2upper t.beautymlowerdiv t.beautymupperdiv btystdave t.didevaluation female formal fulldept lower
## 11         TRUE   FALSE     TRUE FALSE            TRUE             FALSE             TRUE            TRUE             FALSE             FALSE     FALSE            TRUE   TRUE  FALSE    FALSE FALSE
```

```
##    nonenglish onecredit t.percentevaluating t.students tenuretrack blkandwhite t.btystdvariance      bic      aic
## 11       TRUE       TRUE                TRUE       FALSE        TRUE        TRUE             FALSE 2243.019 2189.229
options(width=oldwidth)
```

The best BIC model here is

```
courseevaluation ~ 1 +  t.beautyfupperdiv +  female + nonenglish + onecredit +
    t.percentevaluating + blkandwhite
```

and the best AIC model is

```
courseevaluation ~ 1 + minority + t.beautyf2upper + t.beautyfupperdiv + t.beautym2upper +
     t.didevaluation + female + nonenglish + onecredit + t.percentevaluating +
     blkandwhite
```

This turns out to be a nice illustration of how mathematical optimality may not be what makes the most sense substantively:

- *Each model tries to make a new "beauty" variable instead of taking* `btystdave`*. There isn't really much substantive sense to the beauty variables that the models pick out.*
- *The other variables in these models are a subset of the model from part (d). They are fine, but they also seem to be missing significant predictors with intepretable coefficients.*

If I were advising the university about what (besides teaching quality) affects course evaluation, I would not want to miss some of the variables that the optimal AIC and BIC models miss!