Module 07 Homework

Diana McSpadden

10/10/2020

Stat 6021: Homework Set 7

Name: H. Diana McSpadden

UID: hdm5s

10.15.2020

Attended Study Group With:

Caprill Wright, Abby Bernhardt, Jing Huang, Katie Barbre, Michael Kastanowski, Brian, Nam, Srinivasa Chivaluri

Attended Study Group With

You will continue to use the birthwt data set from the MASS package for this question.

The data were collected at Baystate Medical Center, Springfield, Mass during 1986.

The data contain information regarding weights of newborn babies as well as a number of potential predictors. Before proceeding, be sure to read the documentation about the data set by typing ?birthwt. The goal of the data set is to relate the birthweight of newborns with the characteristics of their mothers during pregnancy.

```
# attach the data
library(MASS)

attach(birthwt)
head(birthwt,5)
```

```
##    low age lwt race smoke ptl ht ui ftv  bwt
## 85   0  19 182    2     0   0  0  1   0 2523
## 86   0  33 155    3     0   0  0  0   3 2551
## 87   0  20 105    1     1   0  0  0   1 2557
## 88   0  21 108    1     1   0  0  1   2 2594
## 89   0  18 107    1     1   0  0  1   0 2600
```

The birthwt data frame has 189 rows and 10 columns. The data were collected at Baystate Medical Center, Springfield, Mass during 1986.

low: indicator of birth weight less than 2.5 kg.

age: mother's age in years.

lwt: mother's weight in pounds at last menstrual period.

race: mother's race (1 = white, 2 = black, 3 = other).

smoke: smoking status during pregnancy.

ptl: number of previous premature labours.

ht: history of hypertension.

ui: presence of uterine irritability.

ftv: number of physician visits during the first trimester.

bwt: birth weight in grams.

Question 1a) Which of these variables are categorical?

Ensure that R is viewing the categorical variables correctly (i.e. use the is.numeric() function to check). If needed, use the factor() function to force R to treat the necessary variables as categorical.

```
low<-factor(low) #ensure low is a factor

race<-factor(race)#ensure race is a factor
levels(race) <- c("white", "black", "other")
contrasts(race)
```

```
##       black other
## white     0     0
## black     1     0
## other     0     1
```

```
smoke<-factor(smoke) #ensure smoke is a factor
ht<-factor(ht) #ensure ht is a factor
ui<-factor(ui) #ensure ui is a factor

head(birthwt)
```

```
##    low age lwt race smoke ptl ht ui ftv  bwt
## 85   0  19 182    2     0   0  0  1   0 2523
## 86   0  33 155    3     0   0  0  0   3 2551
## 87   0  20 105    1     1   0  0  0   1 2557
## 88   0  21 108    1     1   0  0  1   2 2594
## 89   0  18 107    1     1   0  0  1   0 2600
## 91   0  21 124    3     0   0  0  0   0 2622
```

Answer 1a

The following predictor variables are categorical: * low: indicator of birth weight less than 2.5 kg. * race: mother's race (1 = white, 2 = black, 3 = other). * smoke: smoking status during pregnancy. * ht: history of hypertension. * ui: presence of uterine irritability.

Question 1b) A classmate of yours makes the following suggestion:

We should remove the variable low as a predictor for the birth weight of babies."

Do you agree with your classmate? Briefly explain. Hint: you do not need to do any statistical analysis to answer this question.

Answer 1b Yes, I agree with the classmate. Low is a yes/no (1/0) indicator of whether the baby is low birthweight. It will be perfectly correlated with bwt, and is a categorical response variable: a factor variable of the bwt response variable.

Question 1c) Based on your answer to part 1b, perform all possible regressions …

using the regsubsets() function from the leaps package.

Write down the predictors that lead to a first-order model having the best:

1. adjusted R2,
2. mean-sqaured error,
3. Mallow's Cp,
4. BIC.

Work on Question 1c

First, use the regsubsets() function

```
# add the leaps library
library(leaps)

# create all the regsubsets
# set nbest to 5
allreg <- regsubsets(bwt~age+lwt+race+smoke+ptl+ht+ui+ftv, data=birthwt,
nbest=5) # use all variables except low
```

For R2Adj, MSE, Mallows and BIC get the "best" models.

```
##create a "data frame" that stores the predictors in the various models
considered as well as their various criteria
best <- as.data.frame(summary(allreg)$outmat)

best$adjr2 <- summary(allreg)$adjr2
best$p <- as.numeric(substr(rownames(best),1,1))+1
best$mse <- (summary(allreg)$rss)/(dim(birthwt)[1]-best$p)
best$cp <- summary(allreg)$cp
best$bic <- summary(allreg)$bic
```

```
##sort by various criteria
best[order(best$adjr2),] # want largest
```

```
##          age lwt race smoke ptl ht ui ftv      adjr2 p      mse        cp
## 1  ( 5 )                      *           0.01869798 2 521810.8 41.106290
## 1  ( 4 )       *                          0.02933373 2 516155.2 38.655657
## 1  ( 3 )                   *              0.03111683 2 515207.0 38.244805
## 1  ( 2 )            *                     0.03276849 2 514328.7 37.864238
## 1  ( 1 )                               *  0.07569828 2 491500.7 27.972591
## 2  ( 5 )       *                       *  0.09170255 3 482990.3 25.165802
## 2  ( 4 )                   *           *  0.10102669 3 478032.2 23.028874
## 2  ( 3 )            *      *              0.10268198 3 477152.0 22.649510
## 2  ( 2 )                            *   *  0.10270463 3 477139.9 22.644319
## 2  ( 1 )            *                  *  0.10338741 3 476776.9 22.487837
## 3  ( 5 )            *      *       *      0.11671522 4 469689.8 20.344987
## 3  ( 4 )                   *       *   *  0.12706975 4 464183.7 17.984670
## 3  ( 3 )            *              *   *  0.12904641 4 463132.6 17.534089
## 3  ( 2 )       *                   *   *  0.13400871 4 460493.9 16.402930
## 7  ( 5 )   *   *    *      *   *       *    * 0.15416313 8 449776.7 15.639888
## 4  ( 5 )            *      *           *    * 0.15763857 5 447928.6 11.978571
## 4  ( 4 )   *        *      *           *      0.15772122 5 447884.7 11.959833
## 4  ( 3 )            *      *   *       *      0.15980709 5 446775.5 11.486929
## 3  ( 1 )            *      *           *      0.16210659 4 445552.7  9.998012
## 5  ( 5 )       *    *      *   *       *      0.16337433 6 444878.6 11.647312
## 4  ( 2 )       *    *      *           *      0.16642595 5 443255.9  9.986314
## 7  ( 4 )   *        *      *   *   *   *    * 0.17459305 8 438913.0 11.083575
## 6  ( 5 )            *      *   *   *   *    * 0.17884228 7 436653.5  9.147707
## 6  ( 4 )   *        *      *   *   *   *      0.17906949 7 436532.7  9.096753
## 5  ( 4 )            *      *       *   *    * 0.18111774 6 435443.5  7.646422
## 5  ( 3 )   *        *      *       *   *      0.18120963 6 435394.6  7.625702
## 5  ( 2 )            *      *   *   *   *      0.18331011 6 434277.7  7.152073
## 4  ( 1 )            *      *       *   *      0.18555660 5 433083.1  5.649050
## 8  ( 1 )   *   *    *      *   *   *   *    * 0.18841940 9 431560.8  9.000000
## 7  ( 3 )   *   *    *      *       *   *    * 0.19179739 8 429764.6  7.246637
## 7  ( 2 )   *   *    *      *   *   *   *      0.19241200 8 429437.7  7.109564
## 7  ( 1 )       *    *      *   *   *   *    * 0.19289983 8 429178.3  7.000767
## 6  ( 3 )   *   *    *      *       *   *      0.19579424 7 427639.2  5.346166
## 6  ( 2 )       *    *      *       *   *    * 0.19619735 7 427424.9  5.255766
## 6  ( 1 )       *    *      *   *   *   *      0.19681427 7 427096.8  5.117420
## 5  ( 1 )       *    *      *       *   *      0.20008380 6 425358.2  3.369843
##                  bic
## 1  ( 5 )   5.9081173
## 1  ( 4 )   3.8484756
## 1  ( 3 )   3.5009666
## 1  ( 2 )   3.1785020
## 1  ( 1 )  -5.4019683
## 2  ( 5 )  -4.4748241
## 2  ( 4 )  -6.4250332
## 2  ( 3 )  -6.7733623
```

```
## 2  ( 2 )  -6.7781329
## 2  ( 1 )  -6.9220048
## 3  ( 5 )  -5.5296306
## 3  ( 4 )  -7.7583197
## 3  ( 3 )  -8.1867767
## 3  ( 2 )  -9.2666937
## 7  ( 5 )   3.1183476
## 4  ( 5 ) -10.2781524
## 4  ( 4 ) -10.2966967
## 4  ( 3 ) -10.7653277
## 3  ( 1 ) -15.5006606
## 5  ( 5 )  -7.3577086
## 4  ( 2 ) -12.2601243
## 7  ( 4 )  -1.5026987
## 6  ( 5 )  -6.6786122
## 6  ( 4 )  -6.7309160
## 5  ( 4 ) -11.4091949
## 5  ( 3 ) -11.4304043
## 5  ( 2 ) -11.9158783
## 4  ( 1 ) -16.6482522
## 8  ( 1 )  -0.5007912
## 7  ( 3 )  -5.4837468
## 7  ( 2 )  -5.6275314
## 7  ( 1 )  -5.7417324
## 6  ( 3 ) -10.6211600
## 6  ( 2 ) -10.7159220
## 6  ( 1 ) -10.8610345
## 5  ( 1 ) -15.8380956
```

```
best[order(best$mse),] # want smallest
```

```
##          age lwt race smoke ptl ht ui ftv      adjr2 p      mse        cp
## 5  ( 1 )       *    *     *       *  *     0.20008380 6 425358.2  3.369843
## 6  ( 1 )       *    *     *   *   *  *     0.19681427 7 427096.8  5.117420
## 6  ( 2 )       *    *     *       *  *   * 0.19619735 7 427424.9  5.255766
## 6  ( 3 )   *   *    *     *       *  *     0.19579424 7 427639.2  5.346166
## 7  ( 1 )       *    *     *   *   *  *   * 0.19289983 8 429178.3  7.000767
## 7  ( 2 )   *   *    *     *   *   *  *     0.19241200 8 429437.7  7.109564
## 7  ( 3 )   *   *    *     *       *  *   * 0.19179739 8 429764.6  7.246637
## 8  ( 1 )   *   *    *     *   *   *  *   * 0.18841940 9 431560.8  9.000000
## 4  ( 1 )           *     *       *  *     0.18555660 5 433083.1  5.649050
## 5  ( 2 )           *     *   *   *  *     0.18331011 6 434277.7  7.152073
## 5  ( 3 )   *       *     *       *  *     0.18120963 6 435394.6  7.625702
## 5  ( 4 )           *     *       *  *   * 0.18111774 6 435443.5  7.646422
## 6  ( 4 )   *       *     *   *   *  *     0.17906949 7 436532.7  9.096753
## 6  ( 5 )           *     *   *   *  *   * 0.17884228 7 436653.5  9.147707
## 7  ( 4 )   *       *     *   *   *  *   * 0.17459305 8 438913.0 11.083575
## 4  ( 2 )       *    *     *           *     0.16642595 5 443255.9  9.986314
## 5  ( 5 )       *    *     *   *       *     0.16337433 6 444878.6 11.647312
## 3  ( 1 )           *     *           *     0.16210659 4 445552.7  9.998012
```

```
## 4  ( 3 )            *     *   *     *     0.15980709 5 446775.5 11.486929
## 4  ( 4 )   *        *     *         *     0.15772122 5 447884.7 11.959833
## 4  ( 5 )            *     *         *   * 0.15763857 5 447928.6 11.978571
## 7  ( 5 )   *   *    *     *   *     *   * 0.15416313 8 449776.7 15.639888
## 3  ( 2 )       *                  *  *     0.13400871 4 460493.9 16.402930
## 3  ( 3 )            *             *  *     0.12904641 4 463132.6 17.534089
## 3  ( 4 )                  *       *  *     0.12706975 4 464183.7 17.984670
## 3  ( 5 )            *     *       *        0.11671522 4 469689.8 20.344987
## 2  ( 1 )            *                *     0.10338741 3 476776.9 22.487837
## 2  ( 2 )                           *  *     0.10270463 3 477139.9 22.644319
## 2  ( 3 )            *     *                0.10268198 3 477152.0 22.649510
## 2  ( 4 )                  *         *     0.10102669 3 478032.2 23.028874
## 2  ( 5 )       *                    *     0.09170255 3 482990.3 25.165802
## 1  ( 1 )                            *     0.07569828 2 491500.7 27.972591
## 1  ( 2 )            *                     0.03276849 2 514328.7 37.864238
## 1  ( 3 )                  *               0.03111683 2 515207.0 38.244805
## 1  ( 4 )       *                          0.02933373 2 516155.2 38.655657
## 1  ( 5 )                        *         0.01869798 2 521810.8 41.106290
##                   bic
## 5  ( 1 ) -15.8380956
## 6  ( 1 ) -10.8610345
## 6  ( 2 ) -10.7159220
## 6  ( 3 ) -10.6211600
## 7  ( 1 )  -5.7417324
## 7  ( 2 )  -5.6275314
## 7  ( 3 )  -5.4837468
## 8  ( 1 )  -0.5007912
## 4  ( 1 ) -16.6482522
## 5  ( 2 ) -11.9158783
## 5  ( 3 ) -11.4304043
## 5  ( 4 ) -11.4091949
## 6  ( 4 )  -6.7309160
## 6  ( 5 )  -6.6786122
## 7  ( 4 )  -1.5026987
## 4  ( 2 ) -12.2601243
## 5  ( 5 )  -7.3577086
## 3  ( 1 ) -15.5006606
## 4  ( 3 ) -10.7653277
## 4  ( 4 ) -10.2966967
## 4  ( 5 ) -10.2781524
## 7  ( 5 )   3.1183476
## 3  ( 2 )  -9.2666937
## 3  ( 3 )  -8.1867767
## 3  ( 4 )  -7.7583197
## 3  ( 5 )  -5.5296306
## 2  ( 1 )  -6.9220048
## 2  ( 2 )  -6.7781329
## 2  ( 3 )  -6.7733623
## 2  ( 4 )  -6.4250332
## 2  ( 5 )  -4.4748241
```

```
## 1  ( 1 )  -5.4019683
## 1  ( 2 )   3.1785020
## 1  ( 3 )   3.5009666
## 1  ( 4 )   3.8484756
## 1  ( 5 )   5.9081173
```

```
best[order(best$cp),] # want smallest
```

```
##          age lwt race smoke ptl ht ui ftv      adjr2 p      mse        cp
## 5  ( 1 )       *    *     *       *  *     0.20008380 6 425358.2  3.369843
## 6  ( 1 )       *    *     *   *   *  *     0.19681427 7 427096.8  5.117420
## 6  ( 2 )       *    *     *       *  *   * 0.19619735 7 427424.9  5.255766
## 6  ( 3 )   *   *    *     *       *  *     0.19579424 7 427639.2  5.346166
## 4  ( 1 )           *     *       *  *     0.18555660 5 433083.1  5.649050
## 7  ( 1 )       *    *     *   *   *  *   * 0.19289983 8 429178.3  7.000767
## 7  ( 2 )   *   *    *     *   *   *  *     0.19241200 8 429437.7  7.109564
## 5  ( 2 )           *     *   *   *  *     0.18331011 6 434277.7  7.152073
## 7  ( 3 )   *   *    *     *       *  *   * 0.19179739 8 429764.6  7.246637
## 5  ( 3 )   *       *     *       *  *     0.18120963 6 435394.6  7.625702
## 5  ( 4 )           *     *       *  *   * 0.18111774 6 435443.5  7.646422
## 8  ( 1 )   *   *    *     *   *   *  *   * 0.18841940 9 431560.8  9.000000
## 6  ( 4 )   *       *     *   *   *  *     0.17906949 7 436532.7  9.096753
## 6  ( 5 )           *     *   *   *  *   * 0.17884228 7 436653.5  9.147707
## 4  ( 2 )       *   *     *           *     0.16642595 5 443255.9  9.986314
## 3  ( 1 )           *     *           *     0.16210659 4 445552.7  9.998012
## 7  ( 4 )   *       *     *   *   *  *   * 0.17459305 8 438913.0 11.083575
## 4  ( 3 )           *     *   *       *     0.15980709 5 446775.5 11.486929
## 5  ( 5 )       *   *     *   *       *     0.16337433 6 444878.6 11.647312
## 4  ( 4 )   *       *     *           *     0.15772122 5 447884.7 11.959833
## 4  ( 5 )           *     *           *   * 0.15763857 5 447928.6 11.978571
## 7  ( 5 )   *   *   *     *   *       *   * 0.15416313 8 449776.7 15.639888
## 3  ( 2 )       *                   *  *     0.13400871 4 460493.9 16.402930
## 3  ( 3 )           *               *  *     0.12904641 4 463132.6 17.534089
## 3  ( 4 )               *           *  *     0.12706975 4 464183.7 17.984670
## 3  ( 5 )           *   *           *        0.11671522 4 469689.8 20.344987
## 2  ( 1 )           *                   *     0.10338741 3 476776.9 22.487837
## 2  ( 2 )                               *  *     0.10270463 3 477139.9 22.644319
## 2  ( 3 )           *   *                    0.10268198 3 477152.0 22.649510
## 2  ( 4 )               *               *     0.10102669 3 478032.2 23.028874
## 2  ( 5 )       *                       *     0.09170255 3 482990.3 25.165802
## 1  ( 1 )                               *     0.07569828 2 491500.7 27.972591
## 1  ( 2 )           *                        0.03276849 2 514328.7 37.864238
## 1  ( 3 )               *                    0.03111683 2 515207.0 38.244805
## 1  ( 4 )       *                            0.02933373 2 516155.2 38.655657
## 1  ( 5 )                       *            0.01869798 2 521810.8 41.106290
##                  bic
## 5  ( 1 ) -15.8380956
## 6  ( 1 ) -10.8610345
## 6  ( 2 ) -10.7159220
## 6  ( 3 ) -10.6211600
```

```
## 4  ( 1 ) -16.6482522
## 7  ( 1 )  -5.7417324
## 7  ( 2 )  -5.6275314
## 5  ( 2 ) -11.9158783
## 7  ( 3 )  -5.4837468
## 5  ( 3 ) -11.4304043
## 5  ( 4 ) -11.4091949
## 8  ( 1 )  -0.5007912
## 6  ( 4 )  -6.7309160
## 6  ( 5 )  -6.6786122
## 4  ( 2 ) -12.2601243
## 3  ( 1 ) -15.5006606
## 7  ( 4 )  -1.5026987
## 4  ( 3 ) -10.7653277
## 5  ( 5 )  -7.3577086
## 4  ( 4 ) -10.2966967
## 4  ( 5 ) -10.2781524
## 7  ( 5 )   3.1183476
## 3  ( 2 )  -9.2666937
## 3  ( 3 )  -8.1867767
## 3  ( 4 )  -7.7583197
## 3  ( 5 )  -5.5296306
## 2  ( 1 )  -6.9220048
## 2  ( 2 )  -6.7781329
## 2  ( 3 )  -6.7733623
## 2  ( 4 )  -6.4250332
## 2  ( 5 )  -4.4748241
## 1  ( 1 )  -5.4019683
## 1  ( 2 )   3.1785020
## 1  ( 3 )   3.5009666
## 1  ( 4 )   3.8484756
## 1  ( 5 )   5.9081173
```

```
best[order(best$bic),] # want smallest
```

```
##          age lwt race smoke ptl ht ui ftv      adjr2 p      mse        cp
## 4  ( 1 )            *     *       *  *     0.18555660 5 433083.1  5.649050
## 5  ( 1 )       *    *     *       *  *     0.20008380 6 425358.2  3.369843
## 3  ( 1 )            *     *          *     0.16210659 4 445552.7  9.998012
## 4  ( 2 )       *    *     *          *     0.16642595 5 443255.9  9.986314
## 5  ( 2 )            *     *   *  *  *     0.18331011 6 434277.7  7.152073
## 5  ( 3 )   *        *     *       *  *     0.18120963 6 435394.6  7.625702
## 5  ( 4 )            *     *       *  *   * 0.18111774 6 435443.5  7.646422
## 6  ( 1 )       *    *     *   *   *  *     0.19681427 7 427096.8  5.117420
## 4  ( 3 )            *     *   *      *     0.15980709 5 446775.5 11.486929
## 6  ( 2 )       *    *     *       *  *   * 0.19619735 7 427424.9  5.255766
## 6  ( 3 )   *   *    *     *       *  *     0.19579424 7 427639.2  5.346166
## 4  ( 4 )   *        *     *          *     0.15772122 5 447884.7 11.959833
## 4  ( 5 )            *     *          *   * 0.15763857 5 447928.6 11.978571
## 3  ( 2 )       *                    *  *     0.13400871 4 460493.9 16.402930
```

```
## 3  ( 3 )             *             *  *     0.12904641 4 463132.6 17.534089
## 3  ( 4 )                   *       *  *     0.12706975 4 464183.7 17.984670
## 5  ( 5 )       *     *     *   *     *     0.16337433 6 444878.6 11.647312
## 2  ( 1 )             *                 *     0.10338741 3 476776.9 22.487837
## 2  ( 2 )                           *  *     0.10270463 3 477139.9 22.644319
## 2  ( 3 )             *     *               0.10268198 3 477152.0 22.649510
## 6  ( 4 )   *         *     *   *  *  *     0.17906949 7 436532.7  9.096753
## 6  ( 5 )             *     *   *  *  *   * 0.17884228 7 436653.5  9.147707
## 2  ( 4 )                   *             *  0.10102669 3 478032.2 23.028874
## 7  ( 1 )       *     *     *   *  *  *   * 0.19289983 8 429178.3  7.000767
## 7  ( 2 )   *   *     *     *   *  *  *     0.19241200 8 429437.7  7.109564
## 3  ( 5 )             *     *      *        0.11671522 4 469689.8 20.344987
## 7  ( 3 )   *   *     *     *      *  *   * 0.19179739 8 429764.6  7.246637
## 1  ( 1 )                                 *  0.07569828 2 491500.7 27.972591
## 2  ( 5 )       *                         *  0.09170255 3 482990.3 25.165802
## 7  ( 4 )   *         *     *   *  *  *   * 0.17459305 8 438913.0 11.083575
## 8  ( 1 )   *   *     *     *   *  *  *   * 0.18841940 9 431560.8  9.000000
## 7  ( 5 )   *   *     *     *   *     *   * 0.15416313 8 449776.7 15.639888
## 1  ( 2 )             *                     0.03276849 2 514328.7 37.864238
## 1  ( 3 )                   *               0.03111683 2 515207.0 38.244805
## 1  ( 4 )       *                           0.02933373 2 516155.2 38.655657
## 1  ( 5 )                         *         0.01869798 2 521810.8 41.106290
##                    bic
## 4  ( 1 ) -16.6482522
## 5  ( 1 ) -15.8380956
## 3  ( 1 ) -15.5006606
## 4  ( 2 ) -12.2601243
## 5  ( 2 ) -11.9158783
## 5  ( 3 ) -11.4304043
## 5  ( 4 ) -11.4091949
## 6  ( 1 ) -10.8610345
## 4  ( 3 ) -10.7653277
## 6  ( 2 ) -10.7159220
## 6  ( 3 ) -10.6211600
## 4  ( 4 ) -10.2966967
## 4  ( 5 ) -10.2781524
## 3  ( 2 )  -9.2666937
## 3  ( 3 )  -8.1867767
## 3  ( 4 )  -7.7583197
## 5  ( 5 )  -7.3577086
## 2  ( 1 )  -6.9220048
## 2  ( 2 )  -6.7781329
## 2  ( 3 )  -6.7733623
## 6  ( 4 )  -6.7309160
## 6  ( 5 )  -6.6786122
## 2  ( 4 )  -6.4250332
## 7  ( 1 )  -5.7417324
## 7  ( 2 )  -5.6275314
## 3  ( 5 )  -5.5296306
## 7  ( 3 )  -5.4837468
```

```
## 1  ( 1 )  -5.4019683
## 2  ( 5 )  -4.4748241
## 7  ( 4 )  -1.5026987
## 8  ( 1 )  -0.5007912
## 7  ( 5 )   3.1183476
## 1  ( 2 )   3.1785020
## 1  ( 3 )   3.5009666
## 1  ( 4 )   3.8484756
## 1  ( 5 )   5.9081173
```

Answer 1c

Best R2Adj model uses the following predictors: lwt, race, smoke, ht, ui.

Best MSE model uses the following predictors: lwt, race, smoke, ht, ui.

Best Cp model uses the following predictors: lwt, race, smoke, ht, ui.

Best BIC model uses the following predictors: race, smoke, ht, ui.

Question d) Based on your answer to part 1b, use backward selection to find the best model according to AIC.

Start with the first-order model with all the predictors. What is the regression equation selected?

```
# only the intercept
regnull <- lm(bwt~1, data=birthwt) # intercept only/empty model == ENDING
POINT for backward

#modelAdd <- lm(bwt~age+race)
#summary(modelAdd)

# model with all predictors (except low)
regfull <- lm(bwt~age+race+lwt+smoke+ptl+ht+ui+ftv, data=birthwt) # full
model == END POINT

step(regfull, scope=list(lower=regnull, upper=regfull), direction="backward")
```

```
## Start:  AIC=2461.08
## bwt ~ age + race + lwt + smoke + ptl + ht + ui + ftv
##
##         Df Sum of Sq      RSS    AIC
## - age    1       331 77681278 2459.1
## - ftv    1     47283 77728230 2459.2
## - ptl    1    106439 77787385 2459.3
## <none>               77680946 2461.1
## - lwt    1   1762311 79443257 2463.3
## - ht     1   3728637 81409584 2467.9
## - race   1   4599967 82280914 2470.0
```

```
## - smoke  1   4796844 82477791 2470.4
## - ui     1   5732239 83413186 2472.5
##
## Step:  AIC=2459.09
## bwt ~ race + lwt + smoke + ptl + ht + ui + ftv
##
##         Df Sum of Sq      RSS    AIC
## - ftv    1     50343 77731620 2457.2
## - ptl    1    110047 77791325 2457.3
## <none>               77681278 2459.1
## - lwt    1   1789656 79470934 2461.4
## - ht     1   3731126 81412404 2465.9
## - race   1   4707970 82389248 2468.2
## - smoke  1   4843734 82525012 2468.5
## - ui     1   5749594 83430871 2470.6
##
## Step:  AIC=2457.21
## bwt ~ race + lwt + smoke + ptl + ht + ui
##
##         Df Sum of Sq      RSS    AIC
## - ptl    1    108936 77840556 2455.5
## <none>               77731620 2457.2
## - lwt    1   1741198 79472818 2459.4
## - ht     1   3681167 81412788 2463.9
## - race   1   4660187 82391807 2466.2
## - smoke  1   4810582 82542203 2466.6
## - ui     1   5716074 83447695 2468.6
##
## Step:  AIC=2455.47
## bwt ~ race + lwt + smoke + ht + ui
##
##         Df Sum of Sq      RSS    AIC
## <none>               77840556 2455.5
## - lwt    1   1846738 79687294 2457.9
## - ht     1   3718531 81559088 2462.3
## - race   1   4727071 82567628 2464.6
## - smoke  1   5237430 83077987 2465.8
## - ui     1   6302771 84143327 2468.2

##
## Call:
## lm(formula = bwt ~ race + lwt + smoke + ht + ui, data = birthwt)
##
## Coefficients:
## (Intercept)         race          lwt        smoke           ht
ui
##    3104.438     -187.849        3.434     -366.135     -595.820     -
523.419
```

```
modelSelected <- lm(bwt~race+lwt+smoke+ht+ui)
summary(modelSelected)

##
## Call:
## lm(formula = bwt ~ race + lwt + smoke + ht + ui)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1842.14  -433.19    67.09   459.21  1631.03
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2837.264    243.676  11.644  < 2e-16 ***
## raceblack   -475.058    145.603  -3.263 0.001318 **
## raceother   -348.150    112.361  -3.099 0.002254 **
## lwt            4.242      1.675   2.532 0.012198 *
## smoke1      -356.321    103.444  -3.445 0.000710 ***
## ht1         -585.193    199.644  -2.931 0.003810 **
## ui1         -525.524    134.675  -3.902 0.000134 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 645.9 on 182 degrees of freedom
## Multiple R-squared:  0.2404, Adjusted R-squared:  0.2154
## F-statistic:   9.6 on 6 and 182 DF,  p-value: 3.601e-09
```

Answer 1d

bwt ~ race + lwt + smoke + ht + ui

All numbers are in grams

E(bwt) = 2837.264 - (475.058 IF race is black) - (348.150 IF race is other) + (4.242 * lwt) - (356.321 IF smoke == 1) - (585.193 IF ht == 1) - (525.524 IF UI == 1)

Question 2) (No R required) The data for this question are 36 monthly observations ...

on variables affecting sales of a product.

The objective is to determine an efficient model for predicting and explaining market share sales, Share, which is the average monthly market share for the product, in percent.

The predictors are average monthly price in dollars, price, amount of advertising exposure based on gross Nielson rating, nielsen, whether a discount price was in effect, discount (1 if discount, 0 otherwise), whether a package promotion was in effect, promo (1 if promotion, 0 otherwise), and time in months, time.

Work on Question 2

Knowns: 1. y/response variable = Share 2. Regressor/predictor variables: 1. price - avg monthly price in dollars 2. nielsen - amount of advertising exposure based on gross Nielson rating 3. discount: categorical 1 if discount, 0 otherwise 4. promo: categorical, 1 if under promo, 0 otherwise 5. time: time in months

Question 2a) The output below is obtained after using the step() function using forward selection, …

starting with a model with just the intercept term. What is the model selected based on forward selection?

Answer 2a

Share ~ discount + promo + price

Question 2b) Your client asks you to explain …

what each step in the output shown above means. Explain the forward selection procedure to your client, for this output.

Answer 2b

We start with a prediction based on approximately the average Share value, or what we call the intercept-only model. Essentially, we start without using price, nielsen, discount, promo, or time to help predict Share - so the average Share is the best we could do in that case.

Then we determine the best **single** predictor variable to add to our model by comparing the predictive power of each single predictor model. To compare the predictive power of each of the resulting models we use the statistical criteria: AIC. AIC is a measure of the candidate model's ability to decrease unexplained error. Since we want our model to explain as much of the difference in Share as possible based on the models predictors, we want the smallest AIC value.

The comparison of all your single predictor models resulted in a model with Share based on discount: (Share)~discount, because when comparing AIC's for all the single predictor models, the discount model had the lowest AIC value (-128.137 vs approximately -94 for each of the other candidate models). At this point we made note of the lowest AIC for the selected single predictor models (AIC = -128.137).

We repeated this process for each of the **two** predictor models that use discount as the first predictor, and the model using discount and promo resulted in the lowest AIC. At this point we made note of the lowest AIC for the selected two predictor models (AIC = -129.69) and confirmed that this model has a lower AIC than the single predictor model.

We repeated this process for each of the **three** predictor models that use discount and promo as the first two predictors, and the lowest AIC model used discount, promo and price. At this point we made note of the lowest AIC for the selected three predictor models

(AIC = -132.94) and confirmed that this model has a lower AIC than the two predictor model.

When we repeated the examination of all **four** predictor models that use discount, promo, and price as the first three predictors we did not identify a model with a lower AIC than the selected three predictor model. This allowed us to **stop examining** additional predictor variables.

If you are interested the formula for AIC is: (n * ln(SSE / n)) + 2p, where n is the number of observations in the dataset, SSE is the sum squared residual error of the model, and p is the number of parameters the model includes. As predictor variables are added we expect SSE to decrease, but p increases. Our model selection method based on AIC attempts the find the model that balances these two effects.

Question 2c) Your client asks if he should go ahead and use the models selected in part 2a.

What advice do you have for your client?

Answer 2c

I advise that we also investigate which model is identified with backwards selection, and with stepwise selection which are other methods to identify candidate models. After we have identified those models, I recommend investigating the PRESS statistic for each model, and talking with subject matter experts about the ability to collect and maintain the predictor variables in any of the identified models. Again, this is a balancing act to identify the "best" model that will also be maintainable and makes sense to you (the client).

Question 3) (No R required) Your client asks you to compare and contrast between R2 and the adjusted R2, …

specifically: name one advantage of R2 over the adjusted R2, and name one advantage of the adjusted R2 over R2.

Answer 3

Benefit of **R2**: Comparing R2 is useful when comparing models with identical numbers of variables, and can also be useful when examining a plot of R2p vs. number of parameters an analyst can use judgment to determine the number of regressors for the final model by examining where the curve of the plot becomes apparent.

Benefit of **R2-Adjusted**: R2Adjusted takes into consideration the effect of differing numbers of parameters and the change in degrees of freedom (n - p). R2Adjusted can be used to compare R2Adjusted values of models with different numbers of parameters and if the model with an additional parameter's R2Adjusted value exceeds the R2Adjusted value of the models with fewer predictors, then we know that the additional parameter is significant.

Question 4) Include the function your group wrote to compute the PRESS statistic

(Question 2 in Guided Question Set).

```
#This is a very literal version of the formula
calculatePRESS <- function(theModel) {

  sumPRESS = 0
  hatDiagonals <- lm.influence(theModel)$hat

  for (i in 1:length(hatDiagonals)){
    sumPRESS <- sumPRESS + ((theModel$residuals[i]) / (1 -
hatDiagonals[i]))^2
  }

  return(sumPRESS)
}

# This used the built-in array functions in R
press <- function(theModel) {

  pr <- theModel$residuals / (1-lm.influence(theModel)$hat)

  return (sum(pr^2))
}
```

Question 5)

Please remember to complete the Module 7 Guided Question Set Participation Self-and Peer-Evaluation Questions via Test & Quizzes on Collab.

Answer Question 5: Will do.