Problem Set 4, Fall 2021

Krista Miller

knitr::opts\_chunk$set(echo = TRUE)  
# Load any packages, if any, that you use as part of your answers here  
# For example:   
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.5

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble 3.1.0 v dplyr 1.0.5  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1  
## v purrr 0.3.4

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(GGally)

## Warning: package 'GGally' was built under R version 4.0.5

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(ggpubr)

## Warning: package 'ggpubr' was built under R version 4.0.5

library(leaps)

## Warning: package 'leaps' was built under R version 4.0.5

library(tidyverse)  
library("dplyr")

CONTEXT: Factorial experiment with doughnuts

Donna is the owner of a boutique doughnut shop. Because many of her customers are conscious of their fat intake but want the flavor of fried doughnuts, she decided to develop a doughnut recipe that minimizes the amount of fat that the doughnuts absorb from the fat in which the doughnuts are fried.

She conducted a factorial experiment that had a similar procedures as Lowe (1935). Like Lowe, she used four types of fats (fat\_type). She also used three types of flour (flour\_type): all-purpose flour, whole wheat flour, and gluten-free flour. For each combination of fat type and flour type, she cooked six identical batches of doughnuts. Each batch contained 24 doughnuts, and the total fat (in grams) absorbed by the doughnuts in each batch was recorded (sim\_tot\_fat).

## Question 1 - Nested model testing (15 points)

As previously noted, ANOVA is a special case of regression, so anything that can be done in the ANOVA framework can be done in the regression framework. However, this property often isn’t obvious when comparing the output of equivalently-specified analyses. For example, the output of the two-way ANOVA with an interaction displays clearly labeled tests of two main effects and one test of the interaction, but the output of the equivalent regression model displays estimates of numerous coefficients that have interpretations different than those used in the ANOVA framework.

In this question, you will use nested model testing to conduct the equivalent tests of main effects and interactions using the regression framework.

Before you start, read in the data and do your data processing.

doughnuts.factorial <- read.csv("doughnutsfactorial.csv", header=TRUE, sep=",") # Loads the CSV file into memory. You may need to adapt this line to work on your computer

As in the previous problem set, please create two new variables in the doughnuts.factorial data set. The first new variable will be called fat\_type\_factor and will contain the same values as in the fat\_type variable but will have a variable type of factor. The second new variable will be called flour\_type\_factor and will contain the same values as in the flour\_type variable but will also have a variable type of factor.

# Complete the lines to properly create the two new variables  
  
doughnuts.factorial$fat\_type\_factor <- as.factor(doughnuts.factorial$fat\_type)  
  
doughnuts.factorial$flour\_type\_factor <- as.factor(doughnuts.factorial$flour\_type)

Check your work by running the following code chunk. Be sure that fat\_type\_factor and flour\_type\_factor are factor-type variables before you complete the rest of the problem set.

str(doughnuts.factorial)

## 'data.frame': 72 obs. of 5 variables:  
## $ fat\_type : chr "Canola" "Canola" "Canola" "Canola" ...  
## $ flour\_type : chr "ap" "ap" "ap" "ap" ...  
## $ sim\_tot\_fat : int 78 71 80 88 62 72 78 75 89 74 ...  
## $ fat\_type\_factor : Factor w/ 4 levels "Canola","Peanut",..: 1 1 1 1 1 1 3 3 3 3 ...  
## $ flour\_type\_factor: Factor w/ 3 levels "ap","gf","ww": 1 1 1 1 1 1 1 1 1 1 ...

# Question 1, Part 1 - Nested model testing of main effects (5 points)

For this part, you will start by fitting three regression models: an intercept-only model, a fat type-only model, and a flour-type only model. For all models, use sim\_tot\_fat as the outcome.

Fit the intercept-only model here:

model.null <- lm(sim\_tot\_fat~1, data=doughnuts.factorial)  
  
# Use the summary() function to display your results!  
   
summary(model.null)

##   
## Call:  
## lm(formula = sim\_tot\_fat ~ 1, data = doughnuts.factorial)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -37.653 -9.903 0.847 9.597 27.347   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 68.653 1.701 40.35 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14.44 on 71 degrees of freedom

Fit the fat type-only model here:

model.fatType <- lm(sim\_tot\_fat~fat\_type\_factor, data=doughnuts.factorial)  
  
# Use the summary() function to display your results!  
   
summary(model.fatType)

##   
## Call:  
## lm(formula = sim\_tot\_fat ~ fat\_type\_factor, data = doughnuts.factorial)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -35.944 -4.736 -0.167 5.514 21.056   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 66.944 2.529 26.467 < 2e-16 \*\*\*  
## fat\_type\_factorPeanut 8.722 3.577 2.438 0.017372 \*   
## fat\_type\_factorShortening 11.722 3.577 3.277 0.001654 \*\*   
## fat\_type\_factorSunflower -13.611 3.577 -3.805 0.000306 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 10.73 on 68 degrees of freedom  
## Multiple R-squared: 0.4708, Adjusted R-squared: 0.4475   
## F-statistic: 20.17 on 3 and 68 DF, p-value: 1.856e-09

Fit the flour type-only model here:

model.flourType <- lm(sim\_tot\_fat~flour\_type\_factor, data=doughnuts.factorial)  
  
# Use the summary() function to display your results!  
   
summary(model.flourType)

##   
## Call:  
## lm(formula = sim\_tot\_fat ~ flour\_type\_factor, data = doughnuts.factorial)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -35.083 -11.260 0.563 11.385 29.208   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 74.083 2.880 25.723 <2e-16 \*\*\*  
## flour\_type\_factorgf -8.292 4.073 -2.036 0.0456 \*   
## flour\_type\_factorww -8.000 4.073 -1.964 0.0535 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14.11 on 69 degrees of freedom  
## Multiple R-squared: 0.07181, Adjusted R-squared: 0.04491   
## F-statistic: 2.669 on 2 and 69 DF, p-value: 0.07646

Now, conduct two nested model tests to conduct the ANOVA-equivalent tests of main effect.

Compare the intercept-only model to the fat type-only model:

# Code to conduct your nested model test here, be sure your results are displayed in your knitted document  
  
anova(model.null, model.fatType, test='F')

## Analysis of Variance Table  
##   
## Model 1: sim\_tot\_fat ~ 1  
## Model 2: sim\_tot\_fat ~ fat\_type\_factor  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 71 14798.3   
## 2 68 7830.9 3 6967.4 20.167 1.856e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Compare the intercept-only model to the flour type-only model:

# Code to conduct your nested model test here, be sure your results are displayed in your knitted document  
anova(model.null, model.flourType, test='F')

## Analysis of Variance Table  
##   
## Model 1: sim\_tot\_fat ~ 1  
## Model 2: sim\_tot\_fat ~ flour\_type\_factor  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 71 14798   
## 2 69 13736 2 1062.7 2.6692 0.07646 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Question 1, Part 2 - Nested model testing of interaction (5 points)

Unlike the previous part, the “base” model for this comparison is not an intercept-only model. Rather, the base model is a model where the interaction is omitted. In the regression framework, this means that the correct reduced model for this ANOVA-equivalent test is a model that includes fat type and flour type, but no interaction between them.

Fit the reduced model, which will contain just fat type and flour type (no interaction), below

model.main <- aov(sim\_tot\_fat~flour\_type\_factor+fat\_type\_factor, data= doughnuts.factorial) # Complete this line  
  
# Use the summary() function to display your results!  
   
summary(model.main)

## Df Sum Sq Mean Sq F value Pr(>F)   
## flour\_type\_factor 2 1063 531.3 5.181 0.00813 \*\*   
## fat\_type\_factor 3 6967 2322.5 22.647 3.4e-10 \*\*\*  
## Residuals 66 6768 102.5   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Fit the full model, which will contain fat type, flour type, and their interaction

model.interaction <- aov(sim\_tot\_fat~flour\_type\_factor\*fat\_type\_factor, data= doughnuts.factorial)# Complete this line  
  
# Use the summary() function to display your results!  
   
summary(model.interaction)

## Df Sum Sq Mean Sq F value Pr(>F)   
## flour\_type\_factor 2 1063 531.3 5.028 0.00958 \*\*   
## fat\_type\_factor 3 6967 2322.5 21.976 1.01e-09 \*\*\*  
## flour\_type\_factor:fat\_type\_factor 6 427 71.2 0.674 0.67095   
## Residuals 60 6341 105.7   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Now, conduct one nested model test to conduct the ANOVA-equivalent test of the interaction effect.

# Code to conduct your nested model test here, be sure your results are displayed in your knitted document  
anova(model.main, model.interaction, test='F')

## Analysis of Variance Table  
##   
## Model 1: sim\_tot\_fat ~ flour\_type\_factor + fat\_type\_factor  
## Model 2: sim\_tot\_fat ~ flour\_type\_factor \* fat\_type\_factor  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 66 6768.2   
## 2 60 6340.8 6 427.42 0.6741 0.671

# Question 1, Part 3 - Interpreting your results (5 points)

You will answer three questions comparing the results of your nested regression model tests and the ANOVA-style tests.

Question A: Run the code chunk below to see the results of the one-way ANOVA for fat type you conducted in a previous problem set before answering the question.

doughnuts.fat = aov(sim\_tot\_fat ~ fat\_type\_factor, data=doughnuts.factorial)  
  
summary(doughnuts.fat)

## Df Sum Sq Mean Sq F value Pr(>F)   
## fat\_type\_factor 3 6967 2322.5 20.17 1.86e-09 \*\*\*  
## Residuals 68 7831 115.2   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Look at the results of the nested model test you conducted for *fat type* in *Question 1, Part 1*. Does the F-change test statistic and p-value from that nested model test match the F statistic and p-value (within rounding) of the test in the one-way ANOVA?

Your answer here (yes/no): yes

Question B: Run the code chunk below to see the results of the one-way ANOVA for flour type you conducted in a previous problem set before answering the question.

doughnuts.flour = aov(sim\_tot\_fat ~ flour\_type\_factor, data=doughnuts.factorial)  
  
summary(doughnuts.flour )

## Df Sum Sq Mean Sq F value Pr(>F)   
## flour\_type\_factor 2 1063 531.3 2.669 0.0765 .  
## Residuals 69 13736 199.1   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Look at the results of the nested model test you conducted for *flour type* in *Question 1, Part 1*. Does the F-change test statistic and p-value from that nested model test match the F statistic and p-value (within rounding) of the test in the one-way ANOVA?

Your answer here (yes/no): yes

Question C) Run the code chunk below to see the results of the two-way ANOVA with an interaction model you conducted in a previous problem set before answering the question.

doughnuts.fact.2aov = aov(sim\_tot\_fat ~ fat\_type\_factor + flour\_type\_factor + fat\_type\_factor\*flour\_type\_factor, data=doughnuts.factorial)  
  
summary(doughnuts.fact.2aov)

## Df Sum Sq Mean Sq F value Pr(>F)   
## fat\_type\_factor 3 6967 2322.5 21.976 1.01e-09 \*\*\*  
## flour\_type\_factor 2 1063 531.3 5.028 0.00958 \*\*   
## fat\_type\_factor:flour\_type\_factor 6 427 71.2 0.674 0.67095   
## Residuals 60 6341 105.7   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Look at the results of the nested model test you conducted in *Question 1, Part 2*. Does the F-change test statistic and p-value from the nested model test match the F statistic and p-value (within rounding) of the interaction test in the two-way ANOVA with an interaction?

Your answer here (yes/no): yes

CONTEXT - FISHERMAN DATA (many thanks to Dr. Durso for obtaining this data set)

Data Source: N.B. Al-Majed and M.R. Preston (2000). “Factors Influencing the Total Mercury and Methyl Mercury in the Hair of Fishermen in Kuwait,” Environmental Pollution, Vol. 109, pp. 239-250.

<http://users.stat.ufl.edu/~winner/datasets.html>, downloaded on 4/23/2019

Description: Factors related to mercury levels among fishermen and a control group of non-fishermen.

Variables (names of variables in the data set)

Fisherman indicator (“fisherman”), categorical 0 = No 1 = Yes

Age in years (“age”), continuous

Residence Time in years (“restime”), continuous

Height in cm (“height”), continuous

Weight in kg (“weight”), continuous

Fish meals per week (“fishmlwk”), continuous

Parts of fish consumed (“fishpart”), categorical 0 = none 1 = muscle tissue only 2 = muscle tissue and sometimes whole fish 3 = whole fish

Methyl Mercury in mg/g (“MeHg”), continuous

Total Mercury in mg/g (“TotHg”), continuous

# Do this part before starting Questions 2-4!

Before moving on to conducting automated model selection, you’ll need to do some data processing. First, set the variables you’ll use to the proper data types by completing the lines in the code chunk below. The variables you will include as predictors in your automated model selection are fisherman, age, restime, height, weight, fishmlwk, and fishpart

fish <- read.csv("fishermen\_mercury.csv", header=TRUE, sep=",") # Loads the CSV file into memory. You may need to adapt this line to work on your computer  
   
fish$fisherman\_factor <- as.factor(fish$fisherman)  
   
fish$fishpart\_factor <- as.factor(fish$fishpart)

Check your work by running the following code chunk. Be sure that age, restime, height, weight, and fishmlwk and are either integer-type variables or numeric-type variables (R should type these two appropriately automatically) and that fisherman\_factor and fishpart\_factor are factor-type variables before you complete the rest of the problem set.

str(fish)

## 'data.frame': 135 obs. of 11 variables:  
## $ fisherman : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ age : int 45 38 24 41 43 58 45 46 46 46 ...  
## $ restime : int 6 13 2 2 11 2 6 0 14 5 ...  
## $ height : int 175 173 168 183 175 176 184 170 175 175 ...  
## $ weight : int 70 73 66 80 78 75 85 68 80 75 ...  
## $ fishmlwk : int 14 7 7 7 21 21 21 7 21 7 ...  
## $ fishpart : int 2 1 2 1 1 1 1 2 1 1 ...  
## $ MeHg : num 4.01 4.03 3.58 10.99 10.52 ...  
## $ TotHg : num 4.48 4.79 3.86 11.44 10.85 ...  
## $ fisherman\_factor: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...  
## $ fishpart\_factor : Factor w/ 4 levels "0","1","2","3": 3 2 3 2 2 2 2 3 2 2 ...

Next, transform the outcome variable, TotHg, by taking it’s log (note: do not conduct a Box Cox transformation; just take the log of the outcome variable). Conducting such a transformation isn’t a routine part of automated model selection, but it is an option for improving regression diagnostics and we’ll use it for this particular problem.

fish$logTotHg <- log(fish$TotHg)

Finally, to make some of the later data manipulation easier, the following code chunk creates a new data set that contains only the transformed outcome and the predictors that will be included in the automated model selection. The select() function as used below requires that you have either the dplyr package or the tidyverse packaged loaded into memory. The first argument identifies the data set (fish) from which variables will be obtained, and the remaining arguments are variables from the fish data set that you want to be copied into the new data set.

fish.auto <- select(fish, fisherman\_factor, fishpart\_factor, age, restime, height, weight, fishmlwk, logTotHg)

Upon occasion, using the select() function will cause an error because there is also a base R function named select(). If you encounter an error and you believe that this is the cause, R allows you to add the library name in front of the function to clarify the source of the function. For example: dpylr::select(fish, fisherman\_factor, fishpart\_factor, age, restime, height, weight, fishmlwk, logTotHg). If you encounter an error running the above code chunk, try adding dpylr:: in front of the select() function.

Have one last look at your data structure to check that everything is as expected:

str(fish.auto)

## 'data.frame': 135 obs. of 8 variables:  
## $ fisherman\_factor: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...  
## $ fishpart\_factor : Factor w/ 4 levels "0","1","2","3": 3 2 3 2 2 2 2 3 2 2 ...  
## $ age : int 45 38 24 41 43 58 45 46 46 46 ...  
## $ restime : int 6 13 2 2 11 2 6 0 14 5 ...  
## $ height : int 175 173 168 183 175 176 184 170 175 175 ...  
## $ weight : int 70 73 66 80 78 75 85 68 80 75 ...  
## $ fishmlwk : int 14 7 7 7 21 21 21 7 21 7 ...  
## $ logTotHg : num 1.5 1.57 1.35 2.44 2.38 ...

## Question 2 - Forward selection (10 points)

Use forward selection to find the best set of predictors in the fish.auto data set to predict the log of total mercury (logTotHg). Be sure to include fisherman+\_factor, age, restime, height, weight, fishmlwk, and fishpart\_factor in your pool of potential predictors. Do not include interaction terms or polynomial terms as part of your pool of potential predictors.

Be sure to include trace=1 in your function.

# Code for conducting a forward selection, don't forget to set trace = 1 as part of the step() function  
lm.null<- lm(logTotHg~1, data= fish.auto)  
  
  
fwd.model<- step(lm.null, direction='forward', scope= ~ fisherman\_factor + age + restime + height+ weight + fishmlwk + fishpart\_factor, trace=1)

## Start: AIC=-35.75  
## logTotHg ~ 1  
##   
## Df Sum of Sq RSS AIC  
## + weight 1 14.4439 87.622 -54.353  
## + fishpart\_factor 3 16.2863 85.779 -53.222  
## + height 1 2.8525 99.213 -37.580  
## + fisherman\_factor 1 2.7102 99.356 -37.387  
## + fishmlwk 1 2.1889 99.877 -36.680  
## <none> 102.066 -35.754  
## + age 1 0.9863 101.079 -35.065  
## + restime 1 0.9818 101.084 -35.059  
##   
## Step: AIC=-54.35  
## logTotHg ~ weight  
##   
## Df Sum of Sq RSS AIC  
## + fishpart\_factor 3 11.0779 76.544 -66.600  
## + fisherman\_factor 1 3.9920 83.630 -58.648  
## + fishmlwk 1 1.7671 85.855 -55.103  
## <none> 87.622 -54.353  
## + age 1 0.6720 86.950 -53.392  
## + restime 1 0.3533 87.269 -52.898  
## + height 1 0.3217 87.300 -52.849  
##   
## Step: AIC=-66.6  
## logTotHg ~ weight + fishpart\_factor  
##   
## Df Sum of Sq RSS AIC  
## <none> 76.544 -66.600  
## + fisherman\_factor 1 0.255106 76.289 -65.051  
## + height 1 0.253720 76.290 -65.048  
## + age 1 0.028570 76.515 -64.651  
## + fishmlwk 1 0.016532 76.527 -64.629  
## + restime 1 0.001631 76.542 -64.603

Display the model selected using forward selection by using the summary() function.

# Display the model selected by forward selection using the summary() function!  
  
summary(fwd.model)

##   
## Call:  
## lm(formula = logTotHg ~ weight + fishpart\_factor, data = fish.auto)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.4462 -0.2406 0.0432 0.4148 1.8856   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.94571 0.76915 -3.830 0.000199 \*\*\*  
## weight 0.04106 0.01037 3.960 0.000123 \*\*\*  
## fishpart\_factor1 1.18211 0.28983 4.079 7.85e-05 \*\*\*  
## fishpart\_factor2 0.99304 0.25679 3.867 0.000173 \*\*\*  
## fishpart\_factor3 1.26124 0.35505 3.552 0.000533 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7673 on 130 degrees of freedom  
## Multiple R-squared: 0.2501, Adjusted R-squared: 0.227   
## F-statistic: 10.84 on 4 and 130 DF, p-value: 1.3e-07

fwd.model$anova

## Step Df Deviance Resid. Df Resid. Dev AIC  
## 1 NA NA 134 102.06580 -35.75371  
## 2 + weight -1 14.44389 133 87.62191 -54.35290  
## 3 + fishpart\_factor -3 11.07791 130 76.54400 -66.60023

fwd.model$coefficients

## (Intercept) weight fishpart\_factor1 fishpart\_factor2   
## -2.94571489 0.04106001 1.18211214 0.99304025   
## fishpart\_factor3   
## 1.26123941

## Question 3 - Backward selection (10 points)

Use backward selection to find the best set of predictors in the fish.auto data set to predict the log of total mercury (logTotHg). Be sure to include fisherman+\_factor, age, restime, height, weight, fishmlwk, and fishpart\_factor in your pool of potential predictors. Do not include interaction terms or polynomial terms as part of your pool of potential predictors.

Be sure to include trace=1 in your function.

# Code for conducting a backward selection, don't forget to set trace = 1 as part of the step() function  
lm.full<- lm(logTotHg ~ ., data=fish.auto)  
bkwd.model<- step(lm.full, direction ="backward", trace=1)

## Start: AIC=-57.62  
## logTotHg ~ fisherman\_factor + fishpart\_factor + age + restime +   
## height + weight + fishmlwk  
##   
## Df Sum of Sq RSS AIC  
## - restime 1 0.0098 75.981 -59.598  
## - age 1 0.0141 75.985 -59.590  
## - fishmlwk 1 0.0323 76.003 -59.558  
## - height 1 0.2509 76.222 -59.170  
## - fisherman\_factor 1 0.2666 76.237 -59.142  
## <none> 75.971 -57.615  
## - fishpart\_factor 3 7.1200 83.091 -51.521  
## - weight 1 7.9943 83.965 -46.108  
##   
## Step: AIC=-59.6  
## logTotHg ~ fisherman\_factor + fishpart\_factor + age + height +   
## weight + fishmlwk  
##   
## Df Sum of Sq RSS AIC  
## - age 1 0.0061 75.987 -61.587  
## - fishmlwk 1 0.0299 76.010 -61.545  
## - fisherman\_factor 1 0.2569 76.237 -61.142  
## - height 1 0.2653 76.246 -61.127  
## <none> 75.981 -59.598  
## - fishpart\_factor 3 7.1148 83.095 -53.514  
## - weight 1 8.0450 84.025 -48.011  
##   
## Step: AIC=-61.59  
## logTotHg ~ fisherman\_factor + fishpart\_factor + height + weight +   
## fishmlwk  
##   
## Df Sum of Sq RSS AIC  
## - fishmlwk 1 0.0280 76.015 -63.537  
## - height 1 0.2632 76.250 -63.120  
## - fisherman\_factor 1 0.2767 76.263 -63.096  
## <none> 75.987 -61.587  
## - fishpart\_factor 3 7.2091 83.196 -55.351  
## - weight 1 8.0458 84.032 -50.000  
##   
## Step: AIC=-63.54  
## logTotHg ~ fisherman\_factor + fishpart\_factor + height + weight  
##   
## Df Sum of Sq RSS AIC  
## - height 1 0.2743 76.289 -65.051  
## - fisherman\_factor 1 0.2756 76.290 -65.048  
## <none> 76.015 -63.537  
## - fishpart\_factor 3 7.2012 83.216 -57.318  
## - weight 1 8.0193 84.034 -51.997  
##   
## Step: AIC=-65.05  
## logTotHg ~ fisherman\_factor + fishpart\_factor + weight  
##   
## Df Sum of Sq RSS AIC  
## - fisherman\_factor 1 0.2551 76.544 -66.600  
## <none> 76.289 -65.051  
## - fishpart\_factor 3 7.3410 83.630 -58.648  
## - weight 1 9.4869 85.776 -51.228  
##   
## Step: AIC=-66.6  
## logTotHg ~ fishpart\_factor + weight  
##   
## Df Sum of Sq RSS AIC  
## <none> 76.544 -66.600  
## - fishpart\_factor 3 11.0779 87.622 -54.353  
## - weight 1 9.2355 85.779 -53.222

Display the model selected using forward selection by using the summary() function.

# Display the model selected by backward selection using the summary() function!  
summary(bkwd.model)

##   
## Call:  
## lm(formula = logTotHg ~ fishpart\_factor + weight, data = fish.auto)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.4462 -0.2406 0.0432 0.4148 1.8856   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.94571 0.76915 -3.830 0.000199 \*\*\*  
## fishpart\_factor1 1.18211 0.28983 4.079 7.85e-05 \*\*\*  
## fishpart\_factor2 0.99304 0.25679 3.867 0.000173 \*\*\*  
## fishpart\_factor3 1.26124 0.35505 3.552 0.000533 \*\*\*  
## weight 0.04106 0.01037 3.960 0.000123 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7673 on 130 degrees of freedom  
## Multiple R-squared: 0.2501, Adjusted R-squared: 0.227   
## F-statistic: 10.84 on 4 and 130 DF, p-value: 1.3e-07

bkwd.model$anova

## Step Df Deviance Resid. Df Resid. Dev AIC  
## 1 NA NA 125 75.97071 -57.61514  
## 2 - restime 1 0.009804131 126 75.98051 -59.59772  
## 3 - age 1 0.006130961 127 75.98664 -61.58683  
## 4 - fishmlwk 1 0.027987609 128 76.01463 -63.53711  
## 5 - height 1 0.274262914 129 76.28889 -65.05091  
## 6 - fisherman\_factor 1 0.255106092 130 76.54400 -66.60023

bkwd.model$coefficients

## (Intercept) fishpart\_factor1 fishpart\_factor2 fishpart\_factor3   
## -2.94571489 1.18211214 0.99304025 1.26123941   
## weight   
## 0.04106001

## Question 4 - Best subsets selection (10 points)

Use best subsets selection to find the best set of predictors in the fish.auto data set to predict the log of total mercury (logTotHg). Be sure to include fisherman\_factor, age, restime, height, weight, fishmlwk, and fishpart\_factor in your pool of potential predictors. Do not include interaction terms or polynomial terms as part of your pool of potential predictors.

For this problem, choose the best model based on BIC.

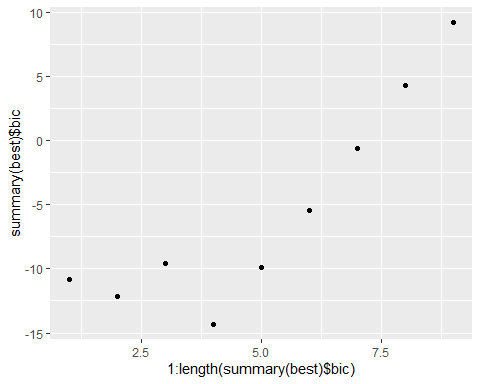
# Code for conducting a best subsets selection  
  
best<- regsubsets(logTotHg~., data=fish.auto, method= "exhaustive", nvmax=9, nbest=1)  
  
subsets<- summary(best)$which  
summary(best)

## Subset selection object  
## Call: regsubsets.formula(logTotHg ~ ., data = fish.auto, method = "exhaustive",   
## nvmax = 9, nbest = 1)  
## 9 Variables (and intercept)  
## Forced in Forced out  
## fisherman\_factor1 FALSE FALSE  
## fishpart\_factor1 FALSE FALSE  
## fishpart\_factor2 FALSE FALSE  
## fishpart\_factor3 FALSE FALSE  
## age FALSE FALSE  
## restime FALSE FALSE  
## height FALSE FALSE  
## weight FALSE FALSE  
## fishmlwk FALSE FALSE  
## 1 subsets of each size up to 9  
## Selection Algorithm: exhaustive  
## fisherman\_factor1 fishpart\_factor1 fishpart\_factor2 fishpart\_factor3  
## 1 ( 1 ) " " " " " " " "   
## 2 ( 1 ) "\*" " " " " " "   
## 3 ( 1 ) "\*" "\*" " " " "   
## 4 ( 1 ) " " "\*" "\*" "\*"   
## 5 ( 1 ) "\*" "\*" "\*" "\*"   
## 6 ( 1 ) "\*" "\*" "\*" "\*"   
## 7 ( 1 ) "\*" "\*" "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*"   
## age restime height weight fishmlwk  
## 1 ( 1 ) " " " " " " "\*" " "   
## 2 ( 1 ) " " " " " " "\*" " "   
## 3 ( 1 ) " " " " " " "\*" " "   
## 4 ( 1 ) " " " " " " "\*" " "   
## 5 ( 1 ) " " " " " " "\*" " "   
## 6 ( 1 ) " " " " "\*" "\*" " "   
## 7 ( 1 ) " " " " "\*" "\*" "\*"   
## 8 ( 1 ) "\*" " " "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*" "\*"

subsets

## (Intercept) fisherman\_factor1 fishpart\_factor1 fishpart\_factor2  
## 1 TRUE FALSE FALSE FALSE  
## 2 TRUE TRUE FALSE FALSE  
## 3 TRUE TRUE TRUE FALSE  
## 4 TRUE FALSE TRUE TRUE  
## 5 TRUE TRUE TRUE TRUE  
## 6 TRUE TRUE TRUE TRUE  
## 7 TRUE TRUE TRUE TRUE  
## 8 TRUE TRUE TRUE TRUE  
## 9 TRUE TRUE TRUE TRUE  
## fishpart\_factor3 age restime height weight fishmlwk  
## 1 FALSE FALSE FALSE FALSE TRUE FALSE  
## 2 FALSE FALSE FALSE FALSE TRUE FALSE  
## 3 FALSE FALSE FALSE FALSE TRUE FALSE  
## 4 TRUE FALSE FALSE FALSE TRUE FALSE  
## 5 TRUE FALSE FALSE FALSE TRUE FALSE  
## 6 TRUE FALSE FALSE TRUE TRUE FALSE  
## 7 TRUE FALSE FALSE TRUE TRUE TRUE  
## 8 TRUE TRUE FALSE TRUE TRUE TRUE  
## 9 TRUE TRUE TRUE TRUE TRUE TRUE

qplot(1:length(summary(best)$bic),summary(best)$bic) # Visualization of BICs



best.subset.bic<-which(summary(best)$bic==min(summary(best)$bic))[1] # Min function searches BICs and shows model number  
best.subset.bic

## [1] 4

varnames<-attr(subsets,"dimnames")[[2]]  
varnames.best <- varnames[subsets[best.subset.bic,]]   
varnames.best

## [1] "(Intercept)" "fishpart\_factor1" "fishpart\_factor2" "fishpart\_factor3"  
## [5] "weight"

Display the model selected using best subsets selection with BIC values. Note: you do not need to “scrape” from the results of using regsubsets() to earn full credit here - you just need to display the model selected using that process, which you can specify manually if you so choose.

# Display the model selected by best subsets selection   
  
bestsubsets.model = lm(logTotHg~weight+fishpart\_factor, data=fish)   
summary(bestsubsets.model)

##   
## Call:  
## lm(formula = logTotHg ~ weight + fishpart\_factor, data = fish)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.4462 -0.2406 0.0432 0.4148 1.8856   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.94571 0.76915 -3.830 0.000199 \*\*\*  
## weight 0.04106 0.01037 3.960 0.000123 \*\*\*  
## fishpart\_factor1 1.18211 0.28983 4.079 7.85e-05 \*\*\*  
## fishpart\_factor2 0.99304 0.25679 3.867 0.000173 \*\*\*  
## fishpart\_factor3 1.26124 0.35505 3.552 0.000533 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7673 on 130 degrees of freedom  
## Multiple R-squared: 0.2501, Adjusted R-squared: 0.227   
## F-statistic: 10.84 on 4 and 130 DF, p-value: 1.3e-07

To compare the results of the best subsets selection with the results of forward and backward selection, you’ll need to convert the model BIC values to AIC values. There is an example of how to do this in the async (3.3 Best Subsets Selection).

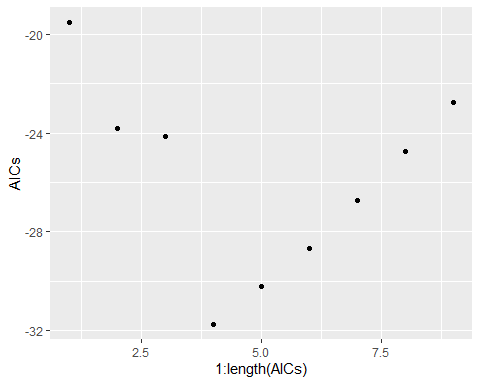
# Convert BIC values to AIC values here  
BICs<-summary(best)$bic  
BICs

## [1] -10.788648 -12.178387 -9.609891 -14.320145 -9.865550 -5.446482 -0.590922  
## [8] 4.303460 9.191314

qs<-1:length(BICs)+2  
n<-nrow(fish.auto)  
AICs<-BICs-log(n)\*qs+2\*qs

Once this is done, determine the best model using best subsets using the AIC values

# Put any code you use to re-determine the best model using best subsets selection with AIC values  
  
qplot(1:length(AICs),AICs)



Display the model selected using best subsets selection with AIC values. Again, you don’t have to “scrape” from the results of using regsubsets().

# Display the model selected by best subsets selection   
bestsubsets.model = lm(logTotHg~weight+fishpart\_factor, data=fish)   
summary(bestsubsets.model)

##   
## Call:  
## lm(formula = logTotHg ~ weight + fishpart\_factor, data = fish)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.4462 -0.2406 0.0432 0.4148 1.8856   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.94571 0.76915 -3.830 0.000199 \*\*\*  
## weight 0.04106 0.01037 3.960 0.000123 \*\*\*  
## fishpart\_factor1 1.18211 0.28983 4.079 7.85e-05 \*\*\*  
## fishpart\_factor2 0.99304 0.25679 3.867 0.000173 \*\*\*  
## fishpart\_factor3 1.26124 0.35505 3.552 0.000533 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7673 on 130 degrees of freedom  
## Multiple R-squared: 0.2501, Adjusted R-squared: 0.227   
## F-statistic: 10.84 on 4 and 130 DF, p-value: 1.3e-07

## Question 5 - 5 points

Question A: Which predictors were included in the model you chose using forward selection?

Your answer here: weight, fishpart\_factor1, fishpart\_factor2, fishpart\_factor3

Question B: Which predictors were included in the model you chose using backward selection?

Your answer here: fishpart\_factor1, fishpart\_factor2, fishpart\_factor3, weight

Question C: Which predictors were included in the model you chose using best subsets selection (AIC)?

Your answer here: fishpart\_factor1, fishpart\_factor2, fishpart\_factor3, weight

Question D: Which predictors were included in the model you chose using best subsets selection (BIC)?

Your answer here:fishpart\_factor1, fishpart\_factor2 , fishpart\_factor3 , weight