Problem Set 3, Fall 2021

Ben Karabinus

```
# Load any packages, if any, that you use as part of your answers here
# For example:
library(MASS)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

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 - 5 points

You may need to process your data before you begin your analysis. Specifically, you will need to make sure that the variable type is set to 'factor' for both of your grouping variables and 'num' for your outcome variable.

```
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
```

Like in Problem Set 1, 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.

```
# create factors for fat_type and flour_type
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.

```
# check the data set structure
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 ...

# additional check of sim_tot_fat variable
class(doughnuts.factorial$sim_tot_fat)

## [1] "integer"

# change sim_tot_fat to numeric data type
doughnuts.factorial$sim_tot_fat <- as.numeric(doughnuts.factorial$sim_tot_fat
)
# ensure changes
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      : num  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 2 - 10 points

Using the code in the code chunk below, I fitted a regression model that has a specification equivalent to a one-way ANOVA. Specifically, I used the doughnuts.factorial data set to re-create the one-way ANOVA with sim_tot_fat as the outcome variable and fat_type_factor as the grouping variable. This is the same model as the first model you estimated in Problem Set 2, Question 4. Run this code before continuing with this question.

```
doughnuts.reg = lm(sim_tot_fat ~ fat_type_factor, data=doughnuts.factorial) #
You may need to change the variable names depending on how you named the vari
ables in Question 1
summary(doughnuts.reg)

##
## 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
```

You will now use the output of this regression model to answer the following four questions. Note that the lm() function dummy coded the fat type variable for you.

Question 1: If you plug in the appropriate zeros and ones into the estimated regression equation to obtain the mean of the *Canola oil* group, which of the four terms shown in the regression output remain (i.e., would not be multiplied by zero)?

Your answer here:

Canola is the reference category in the linear model so the (intercept) would remain. Since Canola is the reference category the intercept is calculated assuming Canola is the fat type and each additional fat type coefficient is calculated as its difference from the intercept.

Question 2: If you plug in the appropriate zeros and ones into the estimated regression equation to obtain the mean of the *Peanut oil* group, which of the four terms shown in the regression output remain (i.e., would not be multiplied by zero)?

Your answer here:

Using the regression equation to calculate the mean fat absorbed for the Peanut oil group would require including the (intercept) and the fat_type_factorPeanut coefficient in the regression equation.

$$\widehat{sim\_tot\_fat} = 66.94 + (1)8.722 + (0)11.722 - (0)13.611$$

Question 3: Based on the output, which fat type had a *lower* mean than that of Canola oil?

Your answer here:

Based on the regression output Sunflower oil had a lower mean fat absorption than Canola oil.

Question 4: Based on the output, which of the three other fat types - Peanut oil Shortening, and Sunflower oil - was significantly different from Canola oil? For full credit, list the names of the fat types (if any) that are significantly different from Canola oil.

Your answer here:

When creating a model using the lm() function in R, factors with 2 or more levels are compared against the reference level in summary output. Keeping this in mind the following fat types are significantly different from canola oil.

- Peanut oil
- Shorening
- Sunflower oil

Question 3 - 10 points

In Problem Set 2, Question 5, you conducted a two-way factorial ANOVA with an interaction. First, copy the code you wrote for Problem Set 2, Question 5 into the code chunk below and display the results using the summary() function. You won't answer any questions about the two-way ANOVA model directly, but you should notice some similarities between the two-way ANOVA with interaction output and your equivalently-specified regression model that may help you answer the questions about the regression model.

```
# create two-way ANOVA with interaction term
fat_flour_int.aov<-aov(sim_tot_fat~fat_type_factor*flour_type_factor,
                        data=doughnuts.factorial)
# ANOVA Summary
summary(fat_flour_int.aov)

##                                   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
```

Next, conduct a regression analysis that is equivalently-specified; that is, it should have sim_tot_fat as the outcome and fat_type_factor, flour_type_factor, and their interaction as predictors. (Hint: much like in the aov() function, interactions are specified in lm() by using * between the two variables that interact). Display the summary of this model using the summary() function. Once you have done this, answer the five questions below based on the regression model output.

```
# create the regression model using lm()
d_model <- lm(sim_tot_fat~fat_type_factor*flour_type_factor,
              data = doughnuts.factorial)

# print model summary
summary(d_model)

## 
## Call:
## lm(formula = sim_tot_fat ~ fat_type_factor * flour_type_factor, 
##     data = doughnuts.factorial)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -28.333  -5.958  -0.250   6.667  21.667 
## 
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                   75.167      4.197  17.910
## fat_type_factorPeanut                          3.667      5.935   0.618
## fat_type_factorShortening                      7.167      5.935   1.207
## fat_type_factorSunflower                     -15.167      5.935  -2.555
## flour_type_factorgf                           -8.833      5.935  -1.488
## flour_type_factorww                          -15.833      5.935  -2.668
## fat_type_factorPeanut:flour_type_factorgf      2.333      8.394   0.278
## fat_type_factorShortening:flour_type_factorgf  3.667      8.394   0.437
## fat_type_factorSunflower:flour_type_factorgf  -3.833      8.394  -0.457
## fat_type_factorPeanut:flour_type_factorww     12.833      8.394   1.529
## fat_type_factorShortening:flour_type_factorww 10.000      8.394   1.191
## fat_type_factorSunflower:flour_type_factorww     8.500      8.394   1.013
##                                                Pr(>|t|)    
## (Intercept)                                     < 2e-16 ***
## fat_type_factorPeanut                           0.53906    
## fat_type_factorShortening                       0.23199    
## fat_type_factorSunflower                        0.01316 *  
## flour_type_factorgf                             0.14191    
## flour_type_factorww                             0.00981 ** 
## fat_type_factorPeanut:flour_type_factorgf       0.78198    
## fat_type_factorShortening:flour_type_factorgf   0.66380    
## fat_type_factorSunflower:flour_type_factorgf    0.64954    
## fat_type_factorPeanut:flour_type_factorww       0.13154    
## fat_type_factorShortening:flour_type_factorww   0.23820    
## fat_type_factorSunflower:flour_type_factorww    0.31529    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.28 on 60 degrees of freedom
## Multiple R-squared:  0.5715, Adjusted R-squared:  0.493
## F-statistic: 7.275 on 11 and 60 DF,  p-value: 1.026e-07
```

Question 1: Not counting the intercept, how many coefficients were estimated in this model?

Your answer here:

Not counting the intercept 11 coefficients were estimated in the model.

Question 2: Not counting the intercept, how many coefficients in this model are associated with the *main effect* of fat type?

Your answer here:

Not counting the intercept 3 coefficients in the model are associated with the main effects of fat type.

Question 3: Not counting the intercept, how many coefficients in this model are associated with the *main effect* of flour type?

Your answer here:

Not counting the intercept 2 coefficients in the model are associated with the main effects of flour type.

Question 4: Not counting the intercept, how many coefficients in this model are associated with the *interaction* between fat type and flour type?

Your answer here:

Not counting the intercept 6 coefficients in the model are associated with interactions between fat type and flour type.

Question 5: What is the predicted amount of fat absorbed by a doughnut made from gluten-free flour and cooked in Sunflower oil? For full credit, you may use any valid method of finding this value (e.g., manually inputting values into an equation, using the predict() function, etc.), but you must show how you obtained it.

```
# create doughnut
d_predict <- data.frame(fat_type_factor = "Sunflower", flour_type_factor = "g
f")
# predict sim_tot_fat
predict(d_model, d_predict)

##        1
## 47.33333
```

Your answer here:

A doughnut made using gluten-free flower and cooked in sunflower oil will have a predicted mean fat absorption of ≈ 47.33 grams.

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

Preamble to Questions 4-5 - do this part before starting these questions!

Before moving on, set the variables you'll use to the proper data types by completing the lines in the code chunk below.

```
# Read in data
fish <- read.csv("fishermen_mercury.csv", header=TRUE, sep=",") # Loads the C
SV file into memory. You may need to adapt this line to work on your computer

# transform categorical variables
fish$fishpart_factor <- as.factor(fish$fishpart)
#fish$fisherman <- as.factor(fish$fisherman)
```

Check your work by running the following code chunk. Be sure that fishmlwk and weight are integer-type or numeric variables (R should type these two appropriately automatically) and fishpart_factor is a factor-type variable before you complete the rest of the problem set.

```
# check the structure of the dataset
str(fish)

## 'data.frame':    135 obs. of  10 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 ...
##  $ fishpart_factor: Factor w/ 4 levels "0","1","2","3": 3 2 3 2 2 2 2 3 2
2 ...
```

Question 4 - 10 points

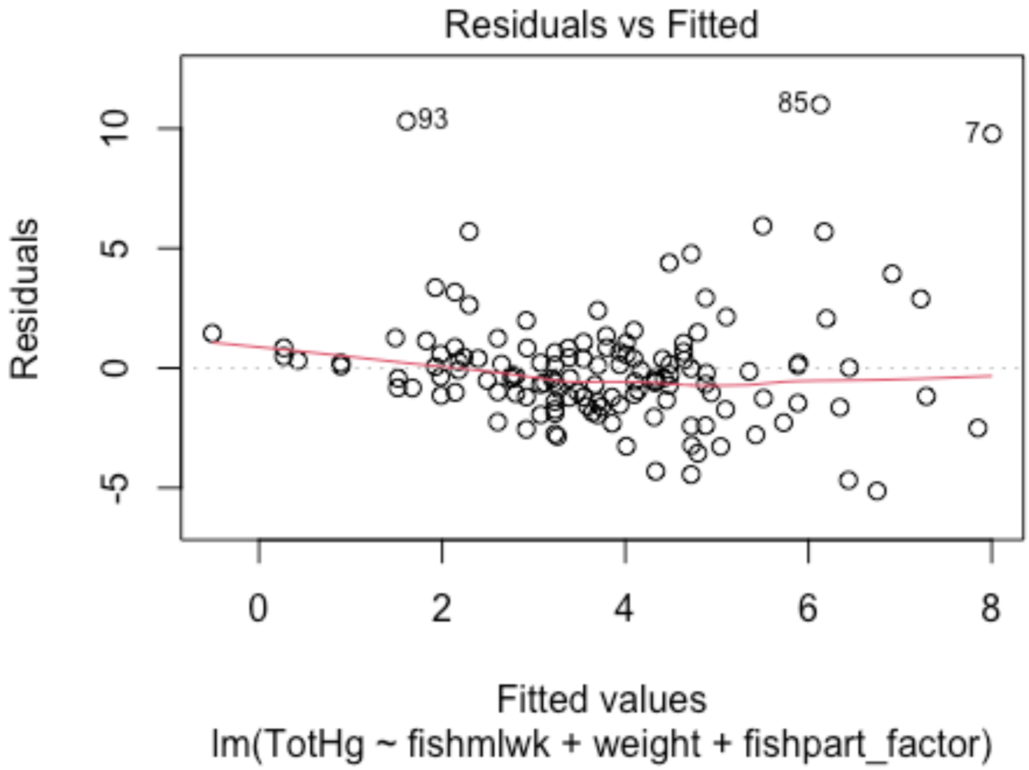
Fit a regression model with "TotHg" as the outcome variable and "fishmlwk", "weight", and "fishpart_factor" as predictor variables. Do not include interaction terms or polynomial terms in this model. Please display the model output using the summary() function.

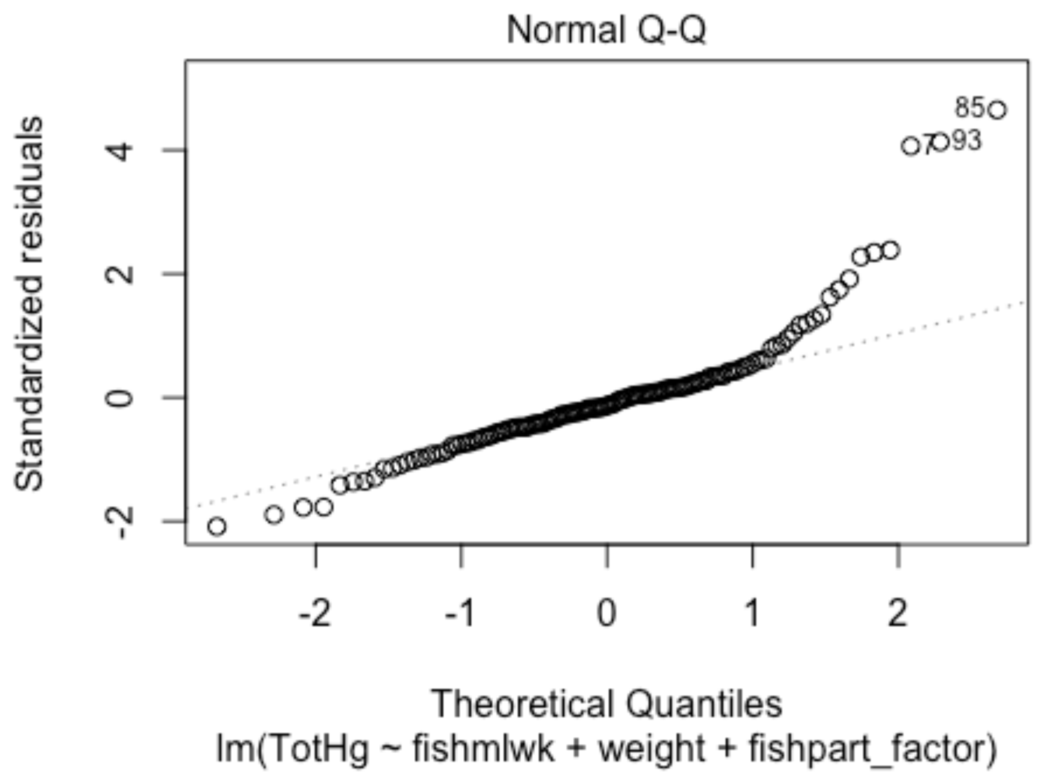
```
# create the regression model
TotHg.reg <- lm(TotHg ~ fishmlwk + weight + fishpart_factor, data = fish)

# print model summary
summary(TotHg.reg)

##
## Call:
## lm(formula = TotHg ~ fishmlwk + weight + fishpart_factor, data = fish)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max 
## -5.1298 -1.2455 -0.3262  0.6778 11.0020 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      -10.02782    2.54490  -3.940 0.000133 ***
## fishmlwk           0.12320    0.04440   2.775 0.006347 ** 
## weight             0.15604    0.03431   4.549 1.23e-05 ***
## fishpart_factor1   2.18255    1.02701   2.125 0.035480 *  
## fishpart_factor2   1.47379    0.89973   1.638 0.103854    
## fishpart_factor3   2.55652    1.22244   2.091 0.038461 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.537 on 129 degrees of freedom
## Multiple R-squared:  0.2822, Adjusted R-squared:  0.2543 
## F-statistic: 10.14 on 5 and 129 DF,  p-value: 3.296e-08
```

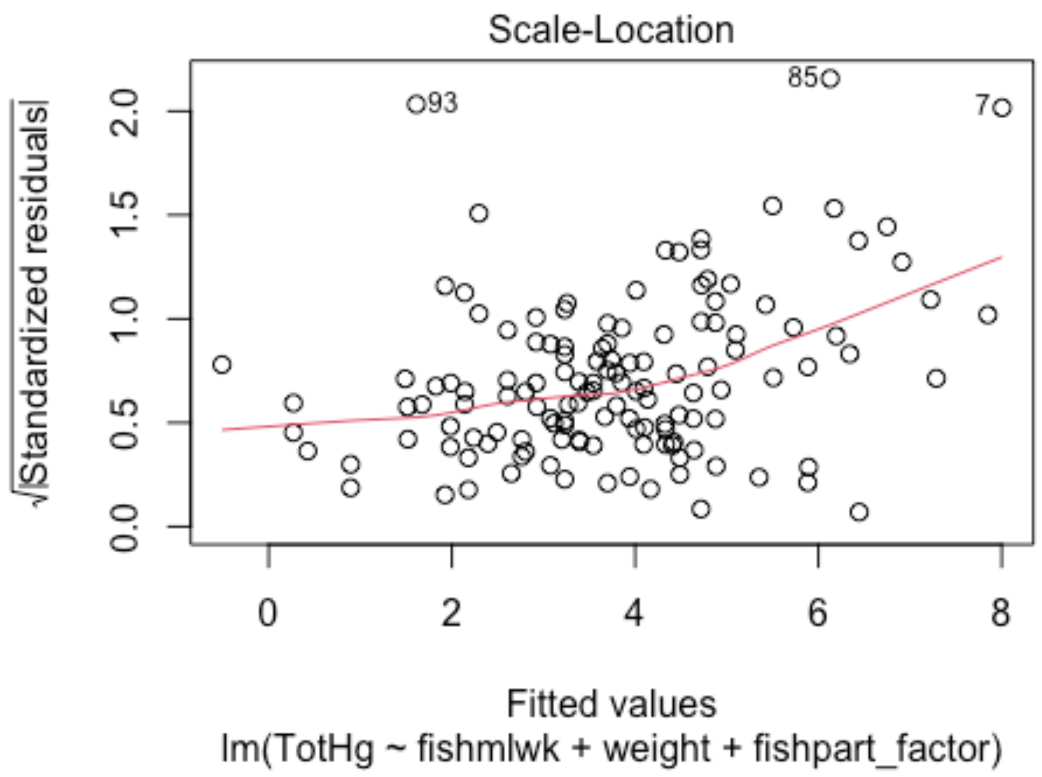
Next, generate the diagnostic plots for this model. Be sure that these are visible in your knitted document. Once you've done so, answer the questions below

```
# print regression diagnostic plots (TotHg)
plot(TotHg.reg)
```

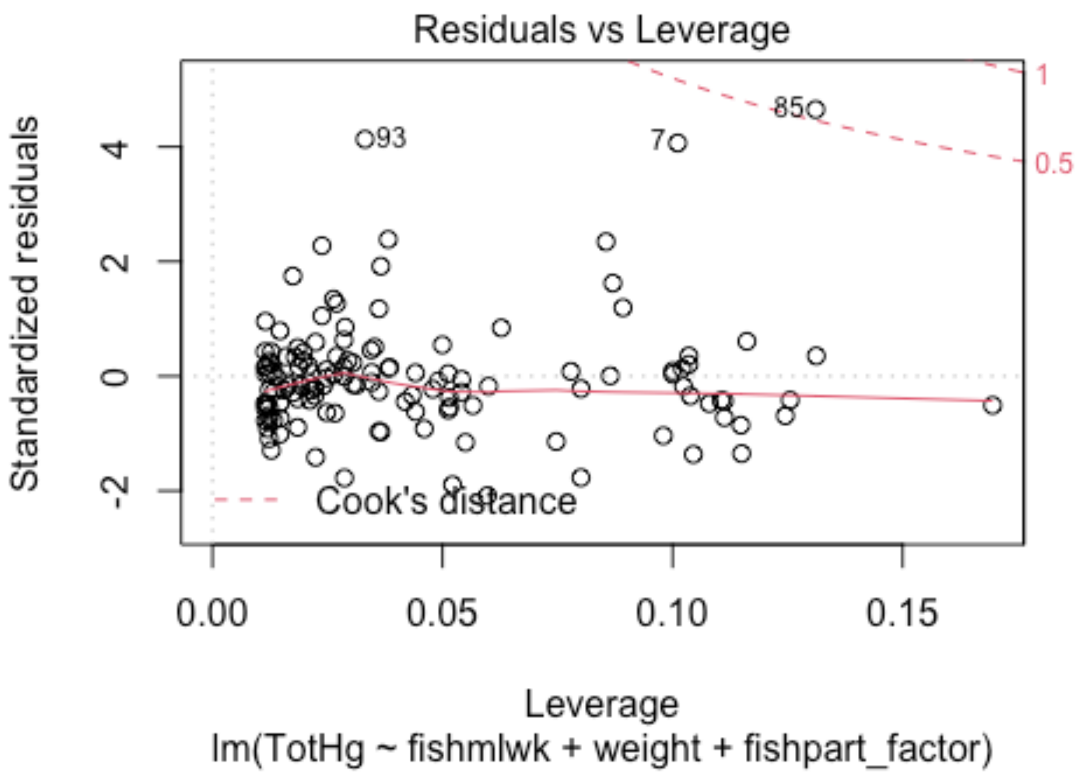

Residuals vs Fitted
Residuals
Fitted values
lm(TotHg ~ fishmlwk + weight + fishpart_factor)
93
85
7
-5
0
5
10
0
2
4
6
8



Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(TotHg ~ fishmlwk + weight + fishpart_factor)
85
7
93



Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
2
4
6
8
93
85
7
Fitted values
lm(TotHg ~ fishmlwk + weight + fishpart_factor)



Question 1: The Residuals vs Fitted plot (the first plot) allows you to assess the assumptions of linearity and equality of residual variance across the range of the fitted values. What specific part of this plot should be examined to evaluate the *linearity* assumption?

Your answer here:

The "Residuals vs Fitted" plot helps evaluate the linearity assumption by producing what is sometimes referred to as a "Loess curve," the solid red line shown on the Residuals vs Fitted plot above. If the assumption of linearity holds the Loess curve should be mostly straight and cut across the graph at around 0 on the y-axis.

Question 2: The Normal Q-Q plot (the second plot) allows you to assess the assumption of normality across the range of the fitted values, and is interpreted similarly to plots you've seen previously. With regard to the assumption of normality, is the left side of the QQ plot more concerning or is the right side more concerning>

Your answer here (left or right):

The right side is more concerning.

Question 3: The Scale-Location plot (the third plot) allows you to assess the assumption of equality of residual variance across the range of the fitted values, and we discussed three shapes that indicate potential violations of this assumption (bowtie, bullhorn, and diamond). Are any of these shapes obviously present in this plot?

Your answer here (yes or no):

Yes

Question 4: The Residuals vs Leverage plot (the four plot) helps you identify potentially influential points. If you use the Cook's D guideline of 0.5, which point/s would you identify as being influential? Please list the observation number/s of any point/s you would consider removing based on this guideline.

Your answer here:

Using the Cook's D guideline of 0.5 I would consider removing observation number 85.

Question 5 - 5 points

If we follow a Cook's D guideline of 0.05, we may want to remove this potentially influential point. We can remove the point and see how the point's removal changes the model estimates and diagnostic plots.

First, remove observation 85 from the fish data set. Save the data set with the removed point into a data set called "fish.remove"

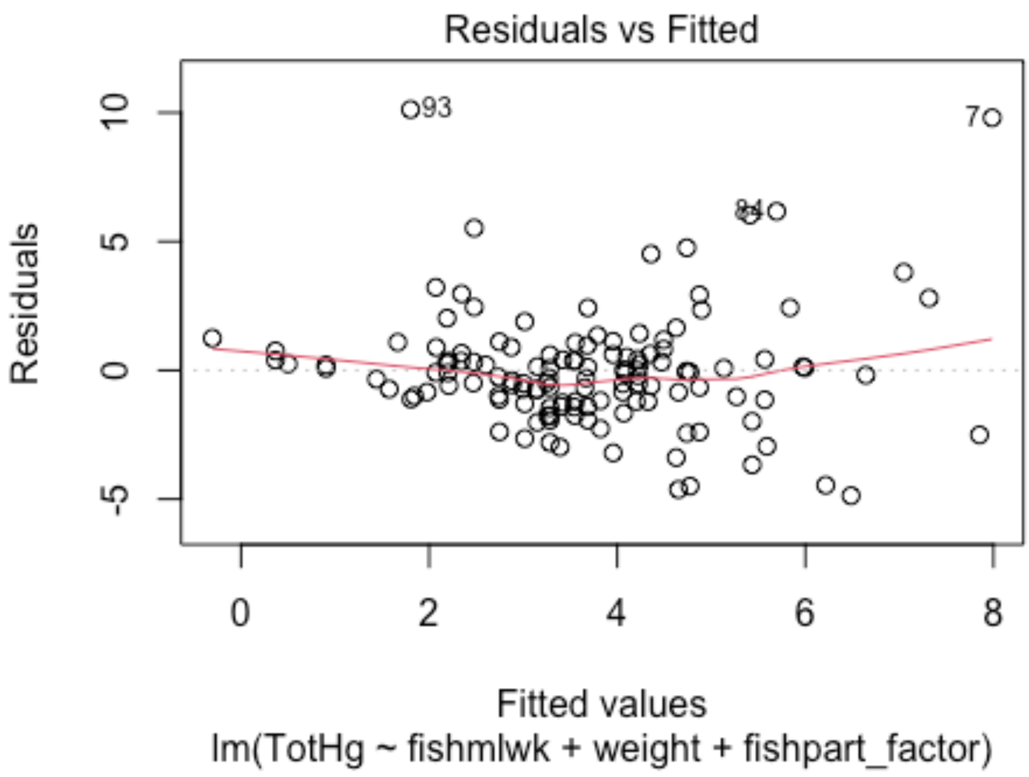
```
# remove observation 85
fish.remove <- fish[-c(85),]
```

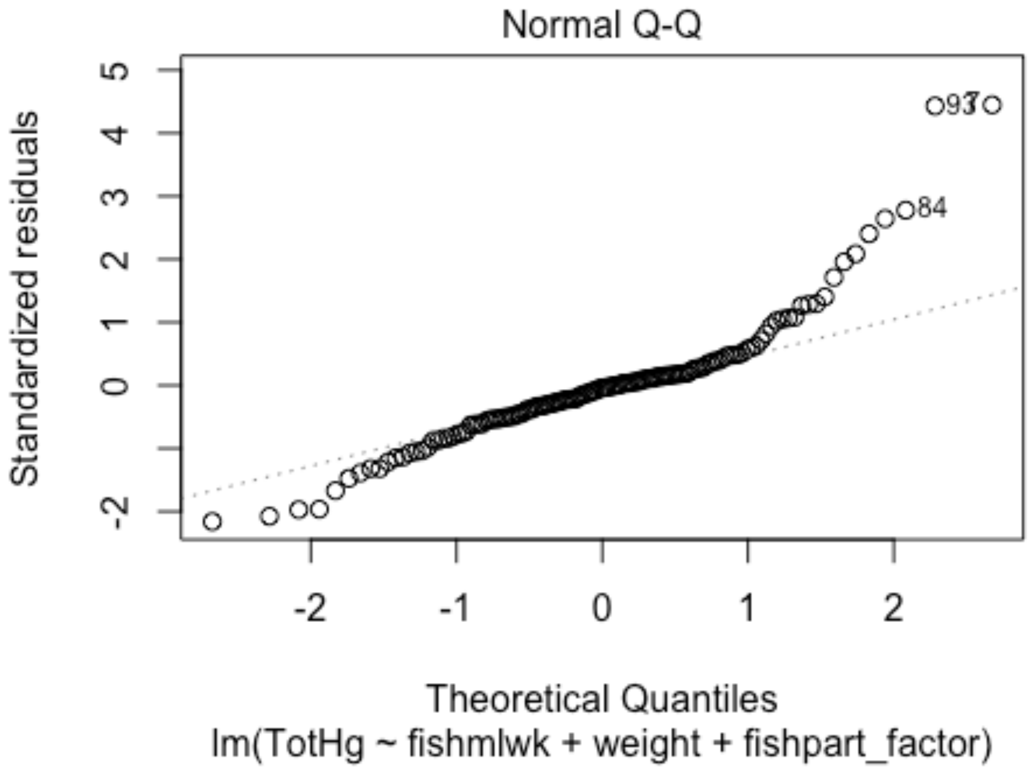
Next, re-fit the *same* model as you fitted in Question 4 and save it as a model object called "fish.reg.remove". This time, use the fish.remove data set.

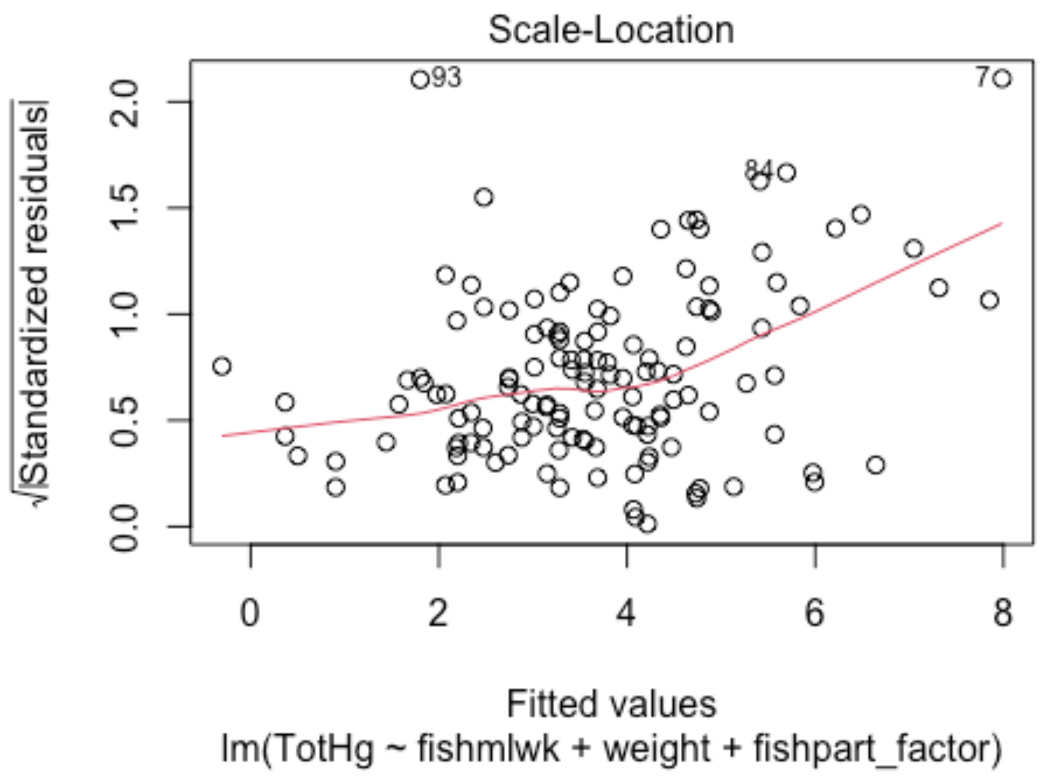
```
# create the new linear model
fish.reg.remove <- lm(TotHg ~ fishmlwk + weight + fishpart_factor, data = fis
h.remove)

# print the model summary
summary(fish.reg.remove)

##
## Call:
## lm(formula = TotHg ~ fishmlwk + weight + fishpart_factor, data = fish.remo
ve)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8667 -1.1706 -0.0931  0.6257 10.1226
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -8.49716    2.35016  -3.616  0.00043 ***
## fishmlwk          0.13599    0.04074   3.338  0.00111 **
## weight            0.13430    0.03171   4.235 4.32e-05 ***
## fishpart_factor1  2.21262    0.94061   2.352  0.02018 *
## fishpart_factor2  1.42901    0.82407   1.734  0.08531 .
## fishpart_factor3  1.14095    1.15375   0.989  0.32458
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.324 on 128 degrees of freedom
## Multiple R-squared:  0.2927, Adjusted R-squared:  0.2651
## F-statistic: 10.59 on 5 and 128 DF,  p-value: 1.572e-08
```

Finally, display the diagnostic plots of the re-fitted model.

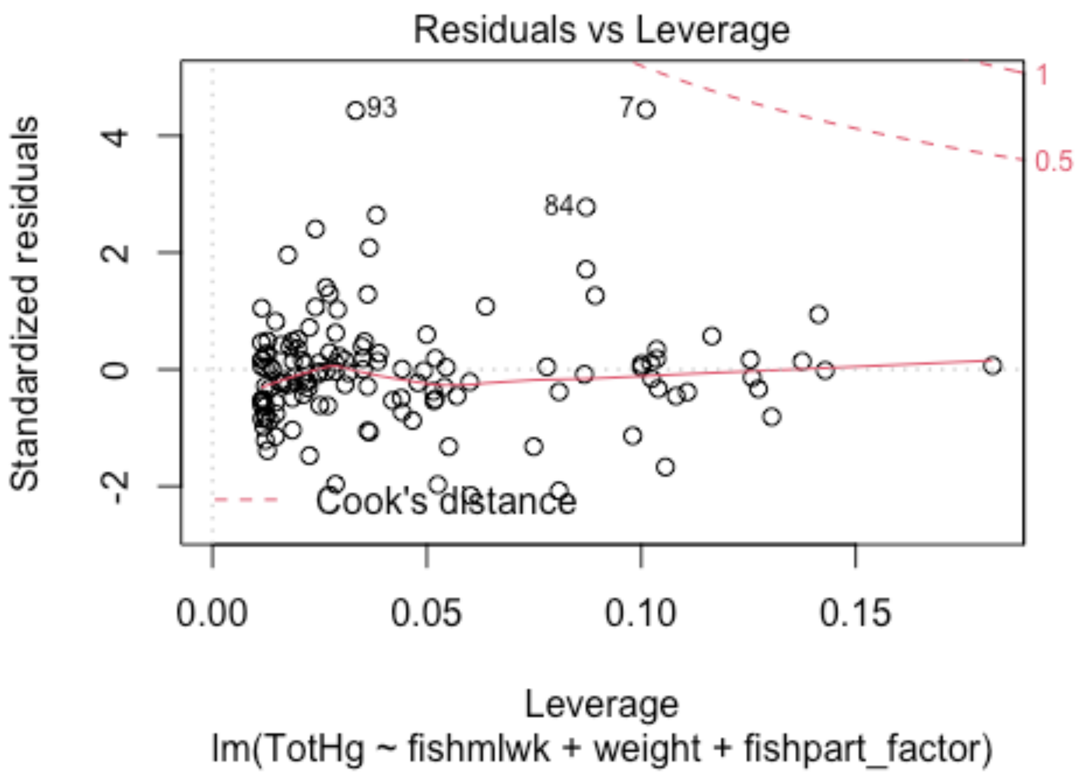
```
# print diagnostic plots (fish.reg.remove)
 plot(fish.reg.remove)
```


Residuals vs Fitted
Residuals
10
5
0
-5
93
7
84
0
2
4
6
8
Fitted values
lm(TotHg ~ fishmlwk + weight + fishpart_factor)


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(TotHg ~ fishmlwk + weight + fishpart_factor)
97
7
84
-2
0
1
2
3
4
5
-2
-1
0
1
2



Scale-Location
√|Standardized residuals|
Fitted values
lm(TotHg ~ fishmlwk + weight + fishpart_factor)
93
7
84
0.0
0.5
1.0
1.5
2.0
0
2
4
6
8



Have a look at both the model coefficients and the diagnostic plots and compare them to the coefficients and plots of the original model in Question 4. Once you've done that, answer the questions below:

A) Look at the p-values associated with fishmlwk, weight, fishpart_factor1, fishpart_factor2, and fishpart_factor3 in both models. Using an alpha level of 0.05, would your conclusion about any of the coefficients be different for any of them? That is, would your conclusion switch from reject -> fail to reject or from fail to reject -> reject for any of the coefficients?

Your answer here (yes/no):

Yes

B) If you answered 'yes' to the previous question, which coefficient/s would your conclusion change between models and how would the conclusion change? Be sure to name the coefficient explicitly (i.e., fishmlwk/weight/fishpart_factor1/fishpart_factor2/fishpart_factor3) and state how the decisions would differ (reject in first model -> fail to reject in second model or from fail to reject in first model -> reject in second model)

Your answer here:

Based on a significance level of $\alpha = 0.05$ using the first model I would reject the null hypothesis (statistically insignificant predictor) for the following coefficients:

- (Intercept)
- fishmlwk
- weight
- fishpart_factor1
- fishpart_factor3

Based on a significance level of $\alpha = 0.05$ using the second model I would reject the null hypothesis (statistically insignificant predictor) for the following coefficients:

- (Intercept)
- fishmlwk
- weight
- fishpart_factor1

Aside: fishpart_factor2 is noteworthy with an $\alpha \approx 0.1$.

The change in my conclusion between the two models is that fishpart_factor3 is not a statistically significant predictor in the second model.

Question 6 - 10 points

There are several reasons to transform variables, one of which we will explore in this question. If the diagnostic plots indicate that the residuals are not normally distributed across the range of fitted values, one can apply a nonlinear transformation to the outcome variable to change the shape of the distribution of the residuals. A common method for doing this is the Box-Cox transformation. Dr. Cathy Durso offered the following explanation of this approach:

“The Box-Cox transformations are a parametrized family of power transformations designed to be applied to the outcome variable to improve the Normality of residuals of a linear model. For $\lambda \neq 0$, the transformation maps y to $\frac{y^\lambda - 1}{\lambda}$ while for $\lambda = 0$, the tranformation maps y to $\ln y$.

For each value of λ in the range of the argument “lambda”, the “boxcox” function in the “MASS” package fits the linear model it is given as an argument but with the Box-Cox transformation applied to the outcome variable, assumed to be positive. The function “boxcox” computes the log likelihood of the residuals under the assumption of Normality. This is plotted against the λ’s and the λ’s and the corresponding log likelihoods are returned. In typical use, a value of λ close to maximizing the log likelihood is chosen and regression performed with this transformation applied to the outcome variable.”

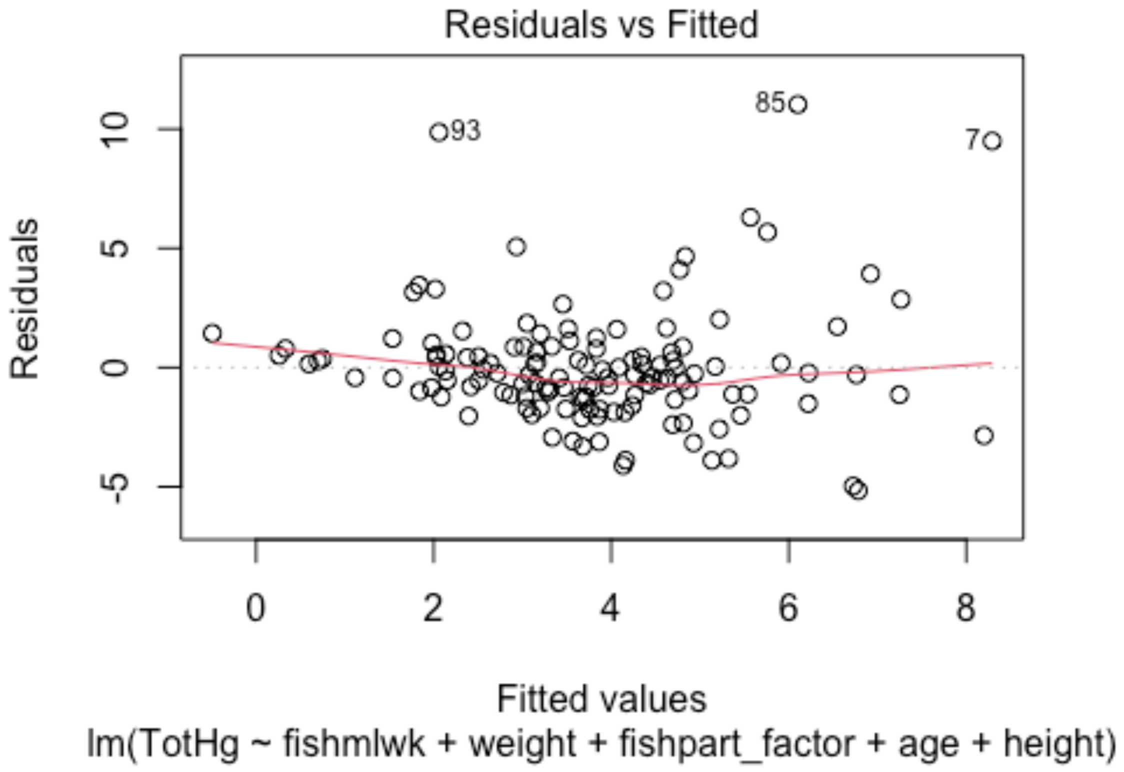
In this problem, you will walk through the steps of conducting a Box-Cox transformation.

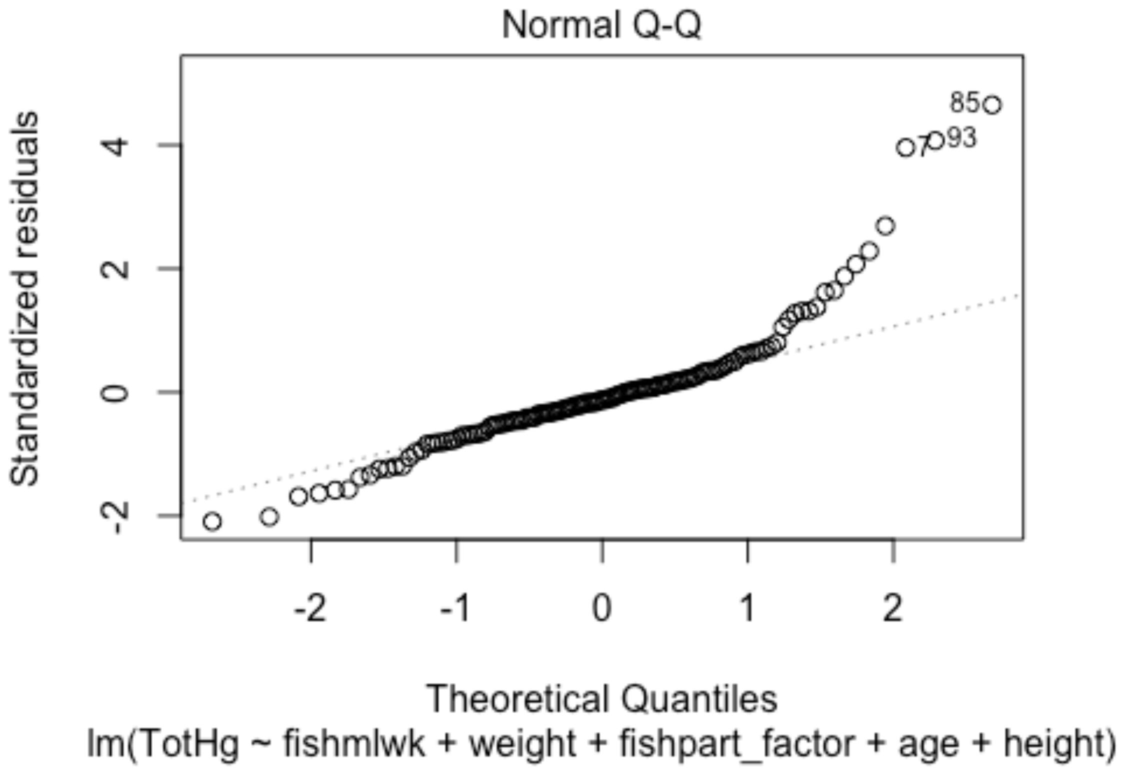
Fitting the base model

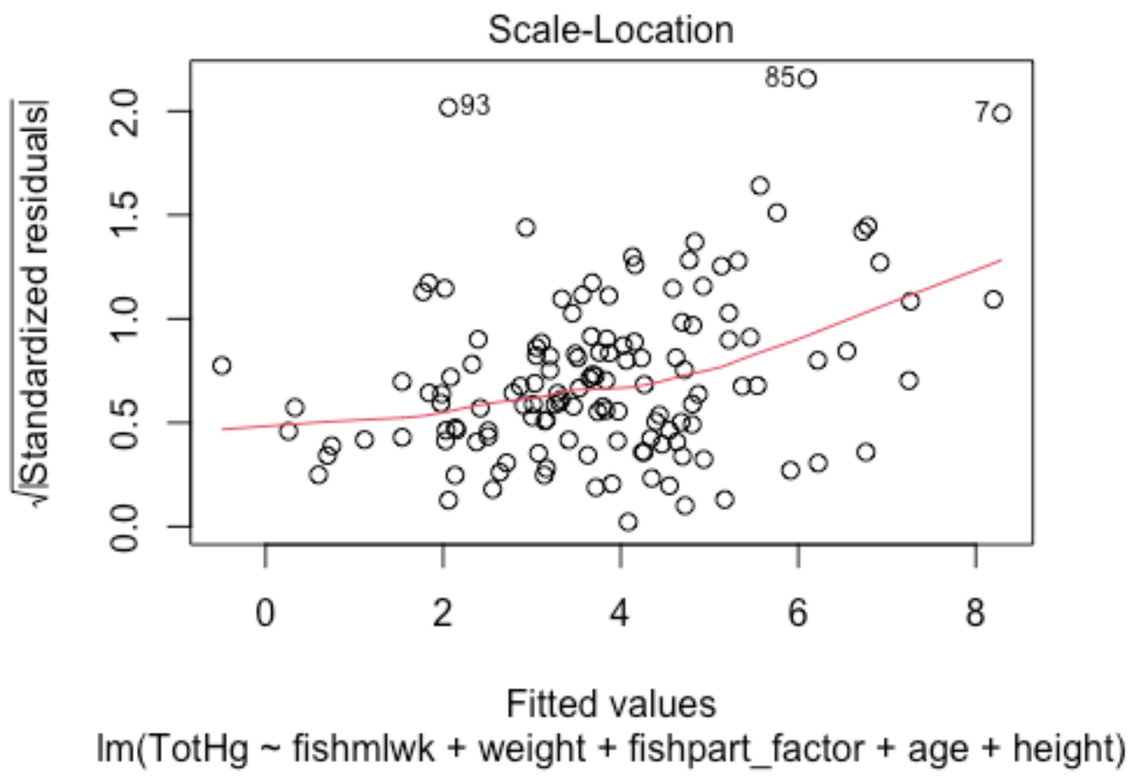
You start the process by fitting the model and examining the diagnostic plots to determine if there is non-normality in the model residuals. This time, the model contains fishmlwk, weight, fishpart_factor, age, and height.

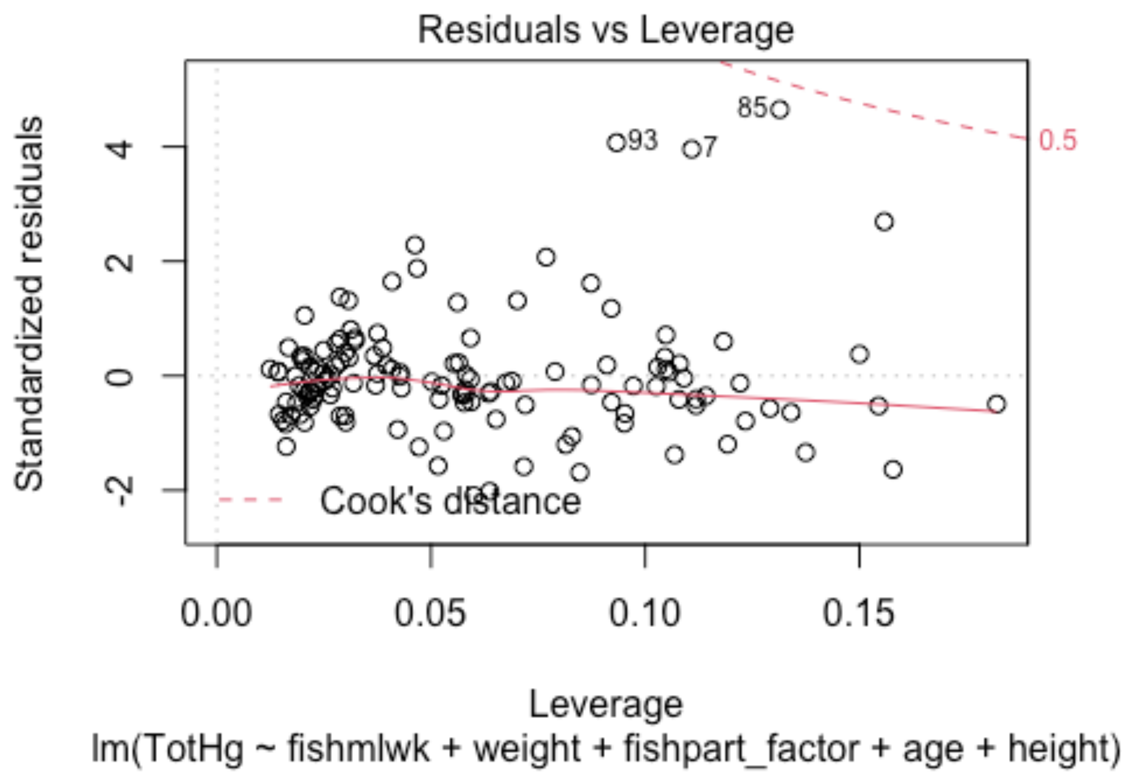
Run the code chunk below and examine the output. You do not need to modify anything (except maybe the very first line; see the comment). Once you've done that, continue onto the next section.

```
fish.demo <- read.csv("fishermen_mercury.csv", header=TRUE, sep=",")  # You m
ay need to change the file path on this line

fish.demo$fishpart_factor <- as.factor(fish$fishpart)

fish.reg.demo = lm(TotHg ~ fishmlwk + weight + fishpart_factor + age + height
, data=fish.demo)

summary(fish.reg.demo)

##
## Call:
## lm(formula = TotHg ~ fishmlwk + weight + fishpart_factor + age +
##     height, data = fish.demo)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1645 -1.2147 -0.3064  0.7235 11.0284
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -15.85401    5.95019  -2.664  0.00871 **
## fishmlwk           0.12113    0.04555   2.659  0.00883 **
## weight             0.14693    0.03574   4.112 7.01e-05 ***
## fishpart_factor1   2.05185    1.04238   1.968  0.05120 .
## fishpart_factor2   1.47024    0.90256   1.629  0.10580
## fishpart_factor3   2.40495    1.23425   1.949  0.05356 .
## age                0.01657    0.02916   0.568  0.57075
## height             0.03432    0.03416   1.005  0.31701
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.544 on 127 degrees of freedom
## Multiple R-squared:  0.2894, Adjusted R-squared:  0.2503
## F-statistic:  7.39 on 7 and 127 DF,  p-value: 1.938e-07

plot(fish.reg.demo)
```


Residuals vs Fitted
Residuals
Fitted values
lm(TotHg ~ fishmlwk + weight + fishpart_factor + age + height)
93
85
7
10
5
0
-5
0
2
4
6
8


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(TotHg ~ fishmlwk + weight + fishpart_factor + age + height)
85
7
93
-2
0
2
4
-2
-1
0
1
2

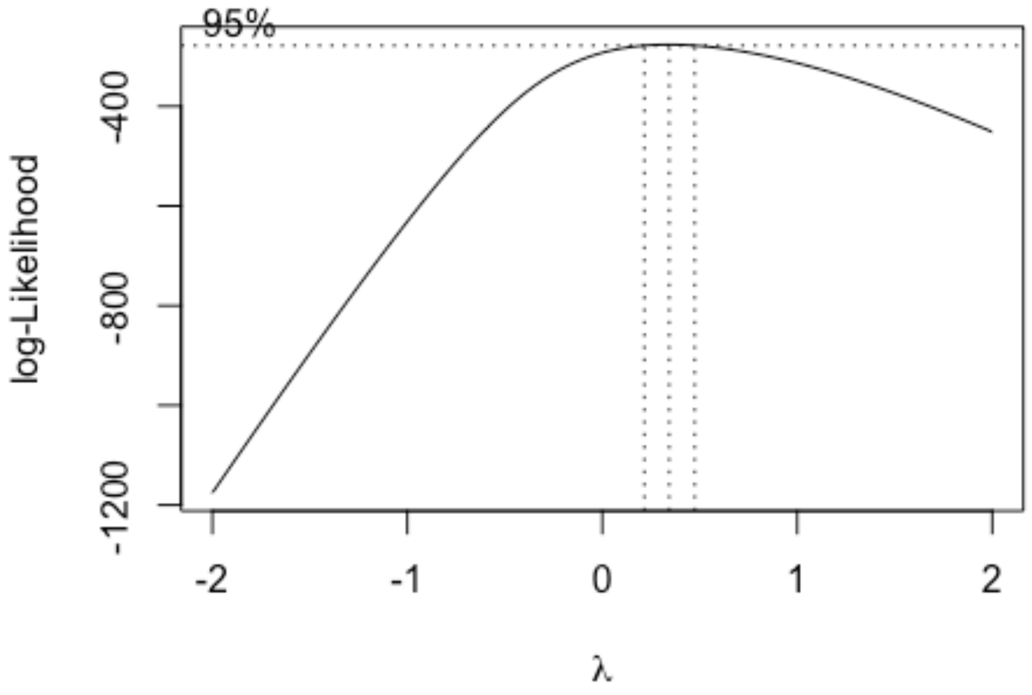

Scale-Location
$\sqrt{|\text{Standardized residuals}|}$
0.0
0.5
1.0
1.5
2.0
93
85
7
0
2
4
6
8
Fitted values
lm(TotHg ~ fishmlwk + weight + fishpart_factor + age + height)



Apply the boxcox() function from the MASS package (loaded at the beginning of this document) to your fitted model

Run the code below and examine what it produces before moving on to the next part. You do not need to modify anything in this code chunk.

```
#perform BoxCox
lambda<-boxcox(fish.reg.demo)
```



```
#print lambda
lambda

## $x
##   [1] -2.00000000 -1.95959596 -1.91919192 -1.87878788 -1.83838384 -1.79797
980
##   [7] -1.75757576 -1.71717172 -1.67676768 -1.63636364 -1.59595960 -1.55555
556
##  [13] -1.51515152 -1.47474747 -1.43434343 -1.39393939 -1.35353535 -1.31313
131
##  [19] -1.27272727 -1.23232323 -1.19191919 -1.15151515 -1.11111111 -1.07070
707
##  [25] -1.03030303 -0.98989899 -0.94949495 -0.90909091 -0.86868687 -0.82828
283
##  [31] -0.78787879 -0.74747475 -0.70707071 -0.66666667 -0.62626263 -0.58585
859
##  [37] -0.54545455 -0.50505051 -0.46464646 -0.42424242 -0.38383838 -0.34343
434
##  [43] -0.30303030 -0.26262626 -0.22222222 -0.18181818 -0.14141414 -0.10101
010
##  [49] -0.06060606 -0.02020202  0.02020202  0.06060606  0.10101010  0.14141
414
##  [55]  0.18181818  0.22222222  0.26262626  0.30303030  0.34343434  0.38383
```

```
838
##  [61]  0.42424242  0.46464646  0.50505051  0.54545455  0.58585859  0.62626
263
##  [67]  0.66666667  0.70707071  0.74747475  0.78787879  0.82828283  0.86868
687
##  [73]  0.90909091  0.94949495  0.98989899  1.03030303  1.07070707  1.11111
111
##  [79]  1.15151515  1.19191919  1.23232323  1.27272727  1.31313131  1.35353
535
##  [85]  1.39393939  1.43434343  1.47474747  1.51515152  1.55555556  1.59595
960
##  [91]  1.63636364  1.67676768  1.71717172  1.75757576  1.79797980  1.83838
384
##  [97]  1.87878788  1.91919192  1.95959596  2.00000000
##
## $y
##   [1] -1175.7102 -1152.6920 -1129.7300 -1106.8269 -1083.9854 -1061.2081
##   [7] -1038.4978 -1015.8577  -993.2910  -970.8015  -948.3930  -926.0697
##  [13]  -903.8363  -881.6978  -859.6596  -837.7280  -815.9093  -794.2112
##  [19]  -772.6417  -751.2100  -729.9264  -708.8020  -687.8502  -667.0849
##  [25]  -646.5230  -626.1830  -606.0856  -586.2554  -566.7183  -547.5054
##  [31]  -528.6507  -510.1913  -492.1705  -474.6325  -457.6274  -441.2073
##  [37]  -425.4252  -410.3351  -395.9903  -382.4373  -369.7198  -357.8718
##  [43]  -346.9144  -336.8635  -327.7150  -319.4583  -312.0715  -305.5199
##  [49]  -299.7666  -294.7669  -290.4741  -286.8416  -283.8230  -281.3735
##  [55]  -279.4523  -278.0204  -277.0428  -276.4881  -276.3269  -276.5344
##  [61]  -277.0869  -277.9640  -279.1474  -280.6198  -282.3667  -284.3739
##  [67]  -286.6292  -289.1210  -291.8386  -294.7725  -297.9133  -301.2527
##  [73]  -304.7826  -308.4955  -312.3844  -316.4424  -320.6632  -325.0408
##  [79]  -329.5692  -334.2431  -339.0569  -344.0058  -349.0848  -354.2892
##  [85]  -359.6145  -365.0564  -370.6109  -376.2738  -382.0413  -387.9099
##  [91]  -393.8759  -399.9360  -406.0869  -412.3255  -418.6487  -425.0536
##  [97]  -431.5376  -438.0979  -444.7320  -451.4372
```

Obtain the λ corresponding to the maximum profile log likelihood

The code below pulls the lambda for which the log likelihood is maximized. You do not need to modify anything in this chunk. Run this code and examine what it produces, then answer the two questions before moving onto the next section.

```
ll.best<-which(lambda[[2]]==max(lambda[[2]]))

ll.best

## [1] 59

lambda.best<-lambda[[1]][ll.best]

lambda.best

## [1] 0.3434343
```

Question A: What is the *number* of the lambda that corresponds to the maximum profile likelihood?

Your answer here:

59

Question B: What is the *value* of the lambda that corresponds to the maximum profile likelihood?

Your answer here:

0.3434343

Apply the transformation to the output variable and re-fit the model.

Now that you have the value of the lambda that corresponds to the maximum profile likelihood, you can now apply the Box-Cox transformation to your model.

Transforming the outcome variable

First, use lambda.best to transform the outcome variable. Recall from Dr. Durso's overview that the formula is $\frac{y^{\lambda}-1}{\lambda}$. Complete the line in the code chunk to compute and save the transformed variable as a new variable ("TotHg.BC") in the fish data set, then answer the question below to check that you applied the transformation correctly.

```
# BoxCox transformation on the outcome variable
fish.demo$TotHg.BC <- (fish.demo$TotHg^lambda.best-1)/(lambda.best)
```

Question C: Look at the first observation in the fish.demo data set. The TotHg value for this observation is 4.484. What is the value of TotHg.BC (i.e., the transformed value) for this same observation?

Your answer here:

1.96307822

Re-fitting the regression model using the transformed outcome

Be sure to have completed the previous section before starting this one.

Next, re-fit the regression model that was originally fitted. Keep the predictors the same - fishmlwk, weight, fishpart_factor, age, and height - but use the transformed outcome (TotHg.BC) instead of the original outcome (TotHg). Display the output for this model.

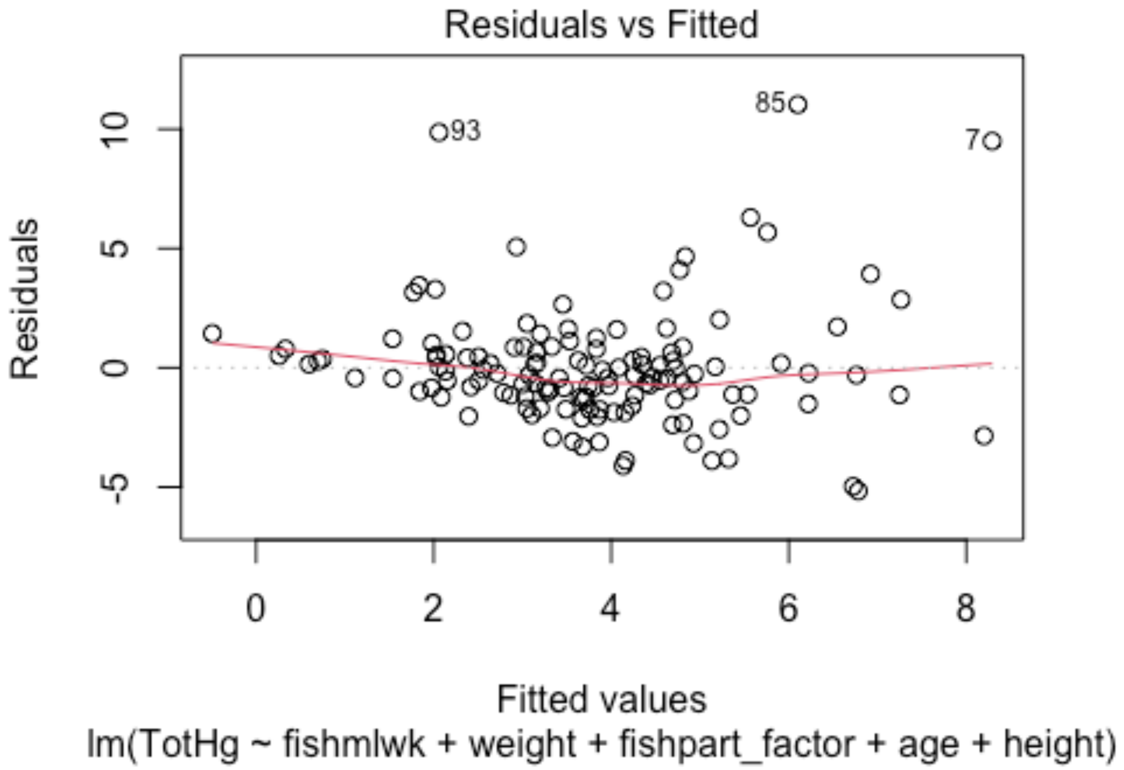
```
# fit the new regression model following box-cox transformation
fish.reg.BC <- lm(TotHg.BC ~ fishmlwk + weight + fishpart_factor + age + heig
ht, data=fish.demo)
```

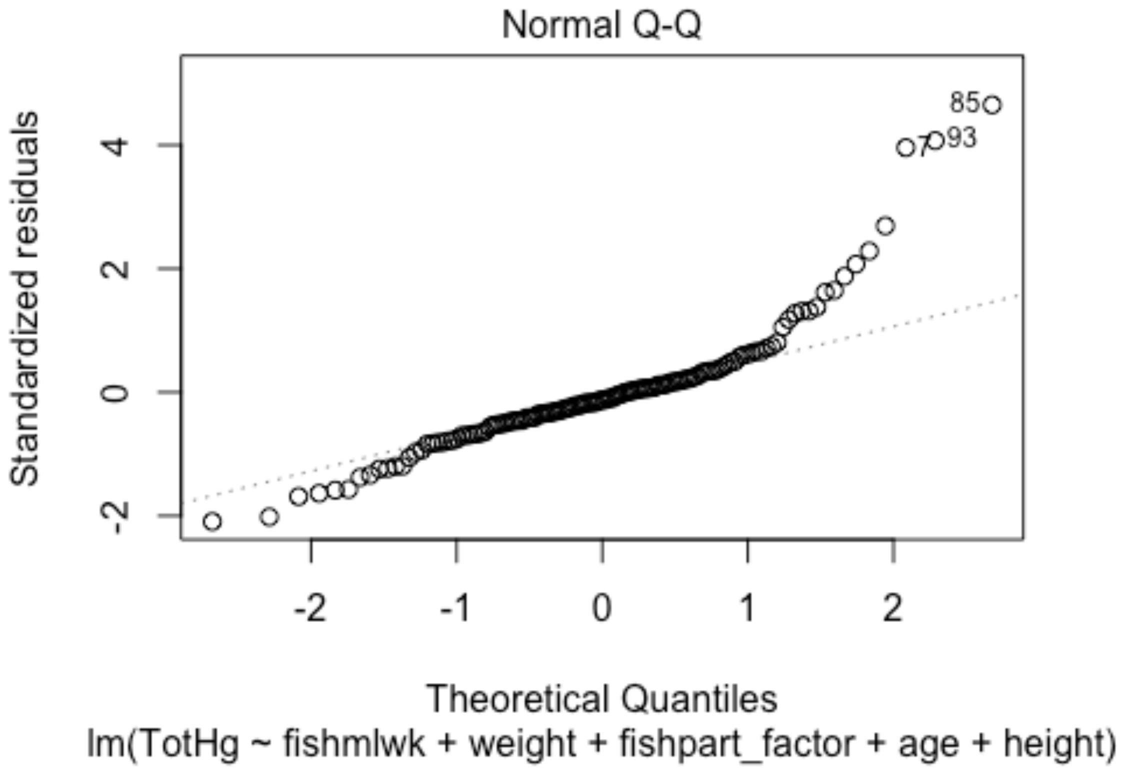
```
# print the model summary
summary(fish.reg.BC)

##
## Call:
## lm(formula = TotHg.BC ~ fishmlwk + weight + fishpart_factor +
##     age + height, data = fish.demo)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4276 -0.4156  0.0023  0.4627  3.0815
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -5.877885   2.292472  -2.564  0.01151 *
## fishmlwk          0.027332   0.017548   1.558  0.12184
## weight            0.056810   0.013769   4.126 6.63e-05 ***
## fishpart_factor1  1.345115   0.401604   3.349  0.00107 **
## fishpart_factor2  1.125900   0.347738   3.238  0.00154 **
## fishpart_factor3  1.444877   0.475530   3.038  0.00289 **
## age               0.003221   0.011233   0.287  0.77477
## height            0.009951   0.013163   0.756  0.45103
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9803 on 127 degrees of freedom
## Multiple R-squared:  0.3012, Adjusted R-squared:  0.2627
## F-statistic:  7.82 on 7 and 127 DF,  p-value: 7.369e-08
```

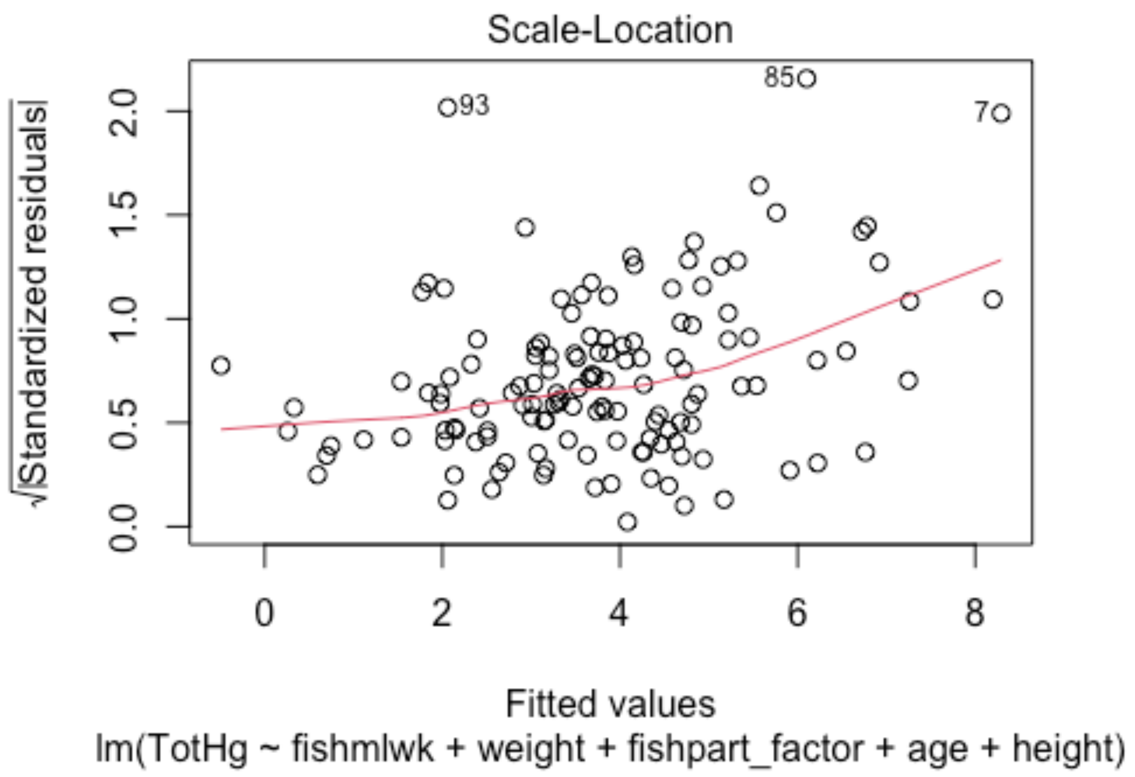
Display diagnostic plots for model with transformed outcome

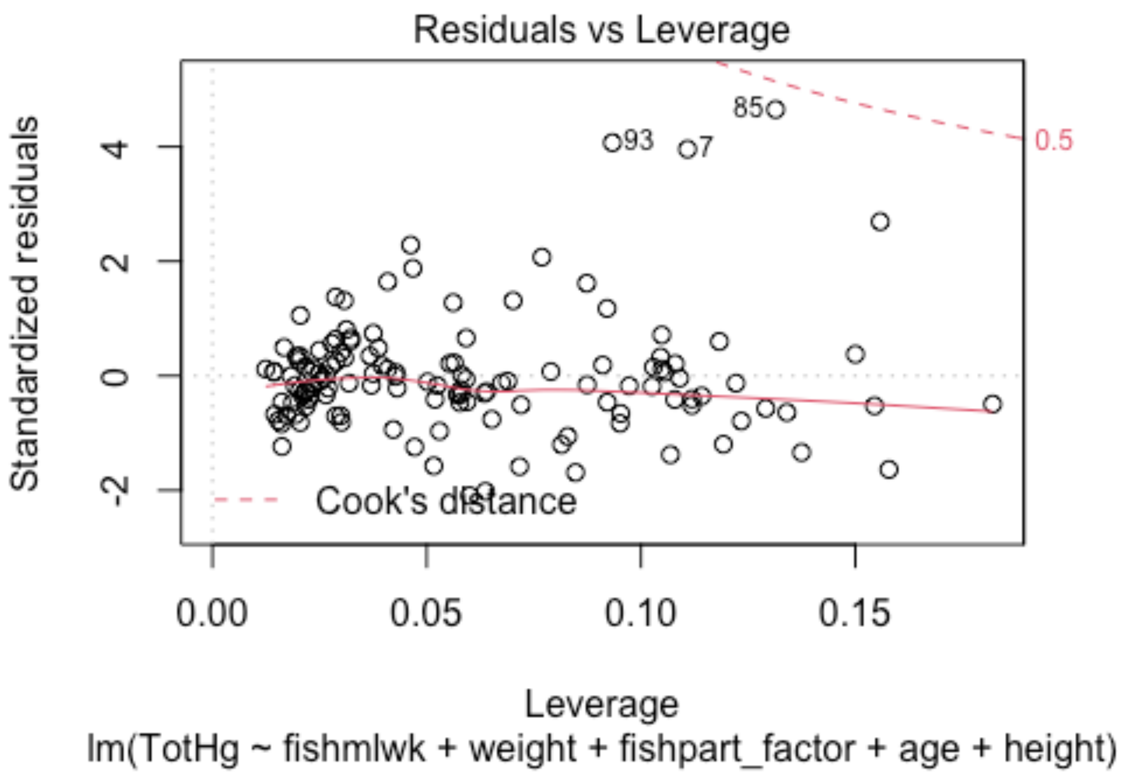
For your convenience, I’ve included code to produce the plots from the original model. Run this chunk (no modifications needed) and continue.

```
plot(fish.reg.demo) # Original model diagnostic plots
```


Residuals vs Fitted
Residuals
Fitted values
lm(TotHg ~ fishmlwk + weight + fishpart_factor + age + height)
93
85
7
-5
0
5
10
0
2
4
6
8

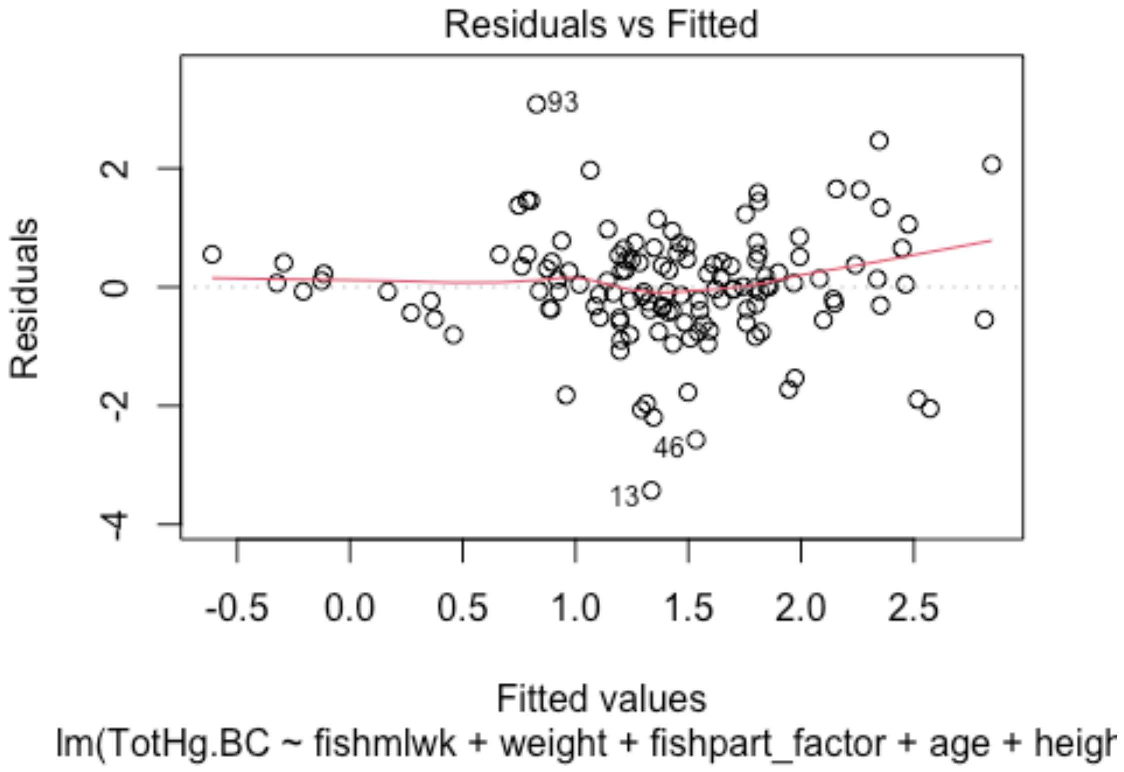

Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(TotHg ~ fishmlwk + weight + fishpart_factor + age + height)
85
7
93
-2
-1
0
1
2
4

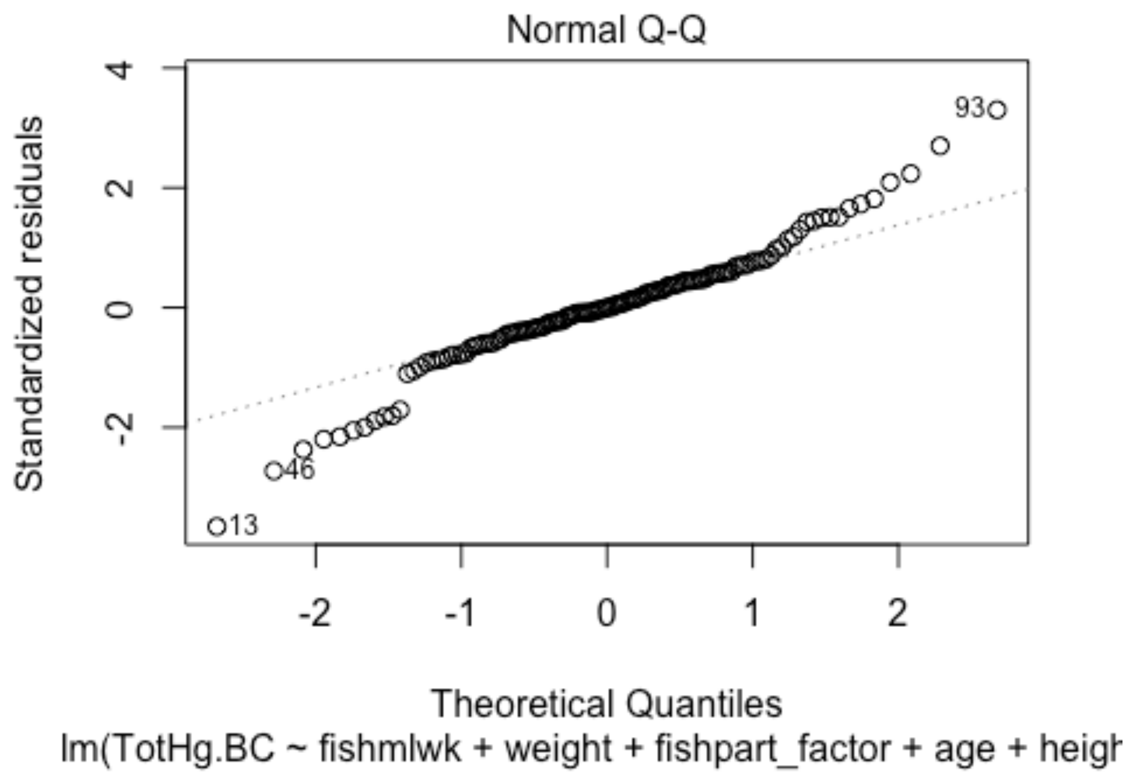

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
93
85
7
0
2
4
6
8
Fitted values
lm(TotHg ~ fishmlwk + weight + fishpart_factor + age + height)

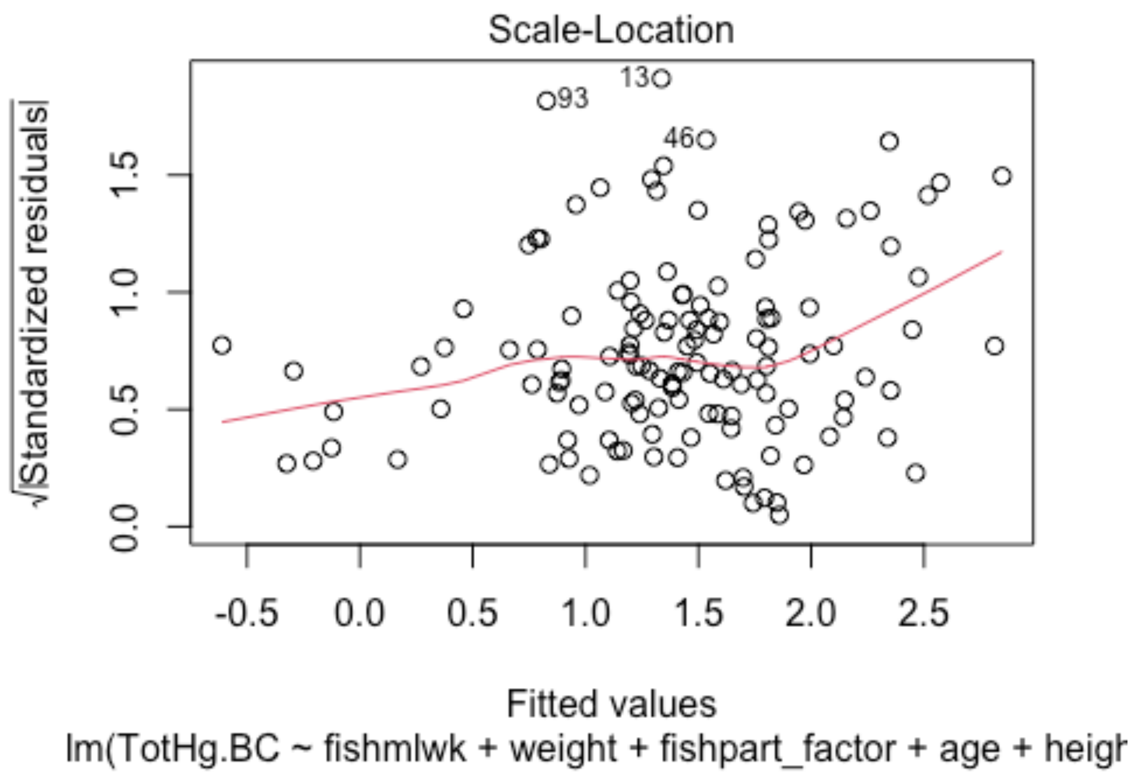


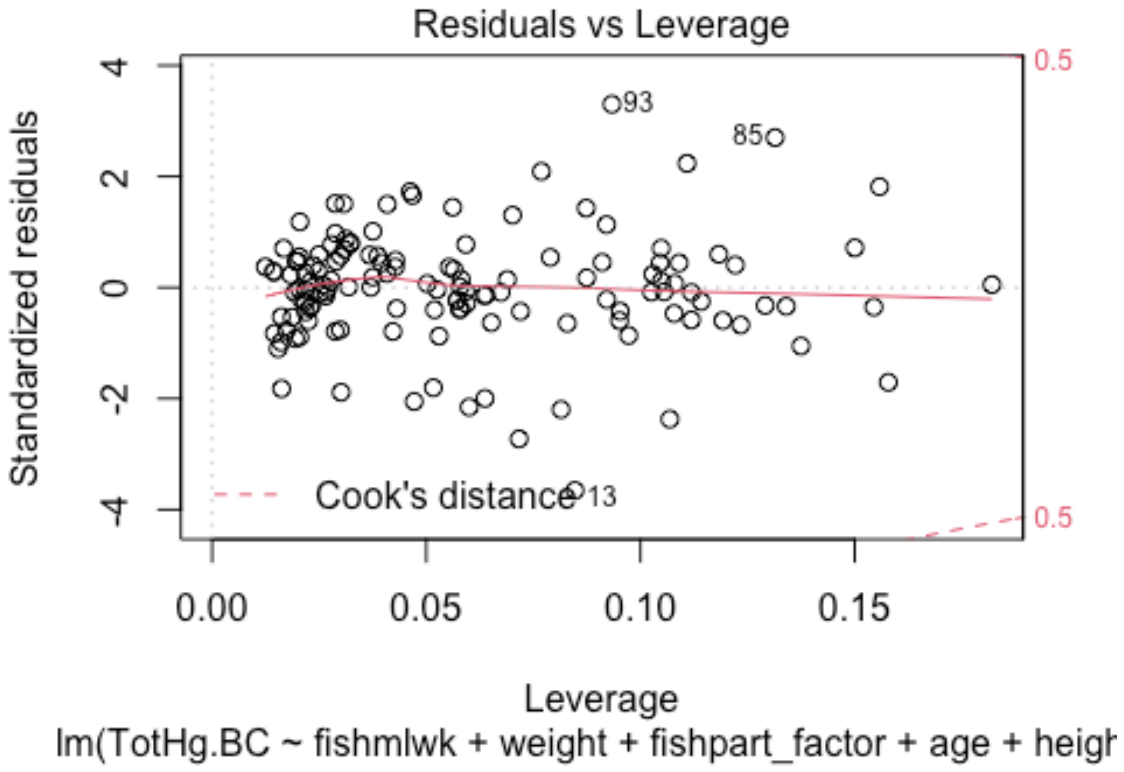
Now, generate the diagnostic plots of the re-fitted model that used the transformed outcome.

```
# Diagnostic plot's for the transformed model
plot(fish.reg.BC)
```


Residuals vs Fitted
93
46
13
Residuals
Fitted values
lm(TotHg.BC ~ fishmlwk + weight + fishpart_factor + age + heigh
2
0
-2
-4
-0.5
0.0
0.5
1.0
1.5
2.0
2.5


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(TotHg.BC ~ fishmlwk + weight + fishpart_factor + age + heigh
93
46
13
4
2
0
-2
-2
-1
0
1
2


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
-0.5
0.0
0.5
1.0
1.5
2.0
2.5
93
13
46
Fitted values
lm(TotHg.BC ~ fishmlwk + weight + fishpart_factor + age + heigh



Look at both of the QQ plots and answer the two questions below.

Question 4: In the regression model using the original outcome variable, was the most concerning side of the QQ plot the left side or the right side?

Your answer here (left or right):

The right side was the most concerning side of the QQ plot.

Question 5: Is the QQ plot of the model that used the transformed outcome different than the one from the model with the original outcome?

Your answer here (yes or no):

Yes