Problem Set 3, Fall 2020

Adam Ten Hoeve

01/29/2021

CONTEXT - DOUGHNUTS DATA

As a reminder, I decided to conduct a factorial experiment inspired by the experiment conducted by Lowe (1935) to learn more about how much fat doughnuts absorb in different conditions. Like Lowe, I used four types of fats (fat_type). I also used three types of flour (flour_type): all-purpose flour, whole wheat flour, and gluten-free flour. Again like Lowe, I cooked six identical batches of doughnuts in each flour and fat combination. 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 will need to read your data set into memory and may need to process your data before you begin your analysis. The data are in the CSV file "doughnuts.csv". Please provide your code for doing this in the code chunk below. Once you've done this, display the attributes of your data set using the str() function.

```
# Your code for reading in the data and changing variable types, if needed
doughnuts = read.csv("doughnutsfactorial.csv")
doughnuts$fat_type = as.factor(doughnuts$fat_type)
doughnuts$flour_type = as.factor(doughnuts$flour_type)
# Don't forget to display the data set attributes using the str() function!

str(doughnuts)
```

```
## 'data.frame':    72 obs. of  3 variables:
##  $ fat_type   : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 2 2 2 2 ...
##  $ flour_type : Factor w/ 3 levels "ap","gf","ww": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sim_tot_fat: int  78 71 80 88 62 72 78 75 89 74 ...
```

Question 2 - 10 points

I refitted the model shown in Problem Set 2, Question 6, the code for which is below. Run this code, then answer the questions:

```
doughnuts.reg = lm(sim_tot_fat ~ fat_type, data=doughnuts)

summary(doughnuts.reg)
```

```
##
## Call:
## lm(formula = sim_tot_fat ~ fat_type, data = doughnuts)
##
## 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_type2     11.722      3.577   3.277 0.001654 **
## fat_type3      8.722      3.577   2.438 0.017372 *
## fat_type4    -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
```

Question 1: What is the interpretation of the intercept coefficient?

Answer: The intercept is 66.944. This means that a doughnut of fat_type 1 will, on average, absorb 66.944 grams of fat.

Question 2: What is the interpretation of the coefficient associated with the second dummy vector of the fat type predictor (hint: don't forget to account for the other coefficients in the model)?

Answer: The coefficient for the second level is 11.722. This means that a doughnut of fat_type 2 will absorb 11.722 more grams of fat than a doughnut of fat_type 1. Therefor, a fat_type 2 doughnut will absorb $66.944 + 11.722 = 78.666$ grams of fat, on average.

Question 3: Which of the fat type vector coefficients (if any) are significantly different from zero?

Answer: We can determine which coefficients are significantly different than zero by looking at their p-values. Because the p-values of all the coefficients are tiny (i.e. less than 0.05), we can say that all of the coefficients are significantly different than zero.

Question 3 - 10 points

In Problem Set 2, Question 4, you conducted a two-way factorial ANOVA with an interaction. First, copy this code from your answer to Problem Set 2, Question 4 into the first code chunk and display the results using the summary() function. Next, conduct a regression analysis that is equivalently-specified; that is, it should have sim_tot_fat as the outcome and fat_type, flour_type, 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 questions below.

```
# Your two-way ANOVA code from Problem Set 2, copied and pasted here. Don't forget to end with the summ
anova.model = aov(sim_tot_fat~fat_type*flour_type, data=doughnuts)
summary(anova.model)
```

```
##                     Df Sum Sq Mean Sq F value   Pr(>F)
```

```
## fat_type             3   6967  2322.5  21.976 1.01e-09 ***
## flour_type           2   1063   531.3   5.028  0.00958 **
## fat_type:flour_type  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
```

```r
# Your code for an equivalently-specified regression model. Don't forget the summary() function.
doughnuts.lmod = lm(sim_tot_fat~fat_type*flour_type, data=doughnuts)
summary(doughnuts.lmod)
```

```
##
## Call:
## lm(formula = sim_tot_fat ~ fat_type * flour_type, data = doughnuts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.333  -5.958  -0.250   6.667  21.667
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               75.167      4.197  17.910  < 2e-16 ***
## fat_type2                  7.167      5.935   1.207  0.23199
## fat_type3                  3.667      5.935   0.618  0.53906
## fat_type4                -15.167      5.935  -2.555  0.01316 *
## flour_typegf              -8.833      5.935  -1.488  0.14191
## flour_typeww             -15.833      5.935  -2.668  0.00981 **
## fat_type2:flour_typegf     3.667      8.394   0.437  0.66380
## fat_type3:flour_typegf     2.333      8.394   0.278  0.78198
## fat_type4:flour_typegf    -3.833      8.394  -0.457  0.64954
## fat_type2:flour_typeww    10.000      8.394   1.191  0.23820
## fat_type3:flour_typeww    12.833      8.394   1.529  0.13154
## fat_type4:flour_typeww     8.500      8.394   1.013  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: How many coefficients are associated with the effect of fat type?

Answer: There are 3 coefficients associated with fat_type.

Question 2: How many coefficients are associated with the effect of flour type?

Answer: There are 2 coefficients associated with flour_type.

Question 3: How many coefficients are associated with the effect of the interaction between fat type and flour type?

Answer: There are 6 coefficients for interaction terms.

Question 4: What is the predicted amount of fat absorbed by a doughnut made from gluten-free flour and cooked in fat type 3?

Answer: $75.167 + 3.667 + (-8.833) + 2.333 = 72.334$ grams of fat absorbed.

CONTEXT - FISHERMAN DATA (adapted from Cathy Durso's material)

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)

Age in years (age)

Residence Time in years (restime)

Height in cm (height)

Weight in kg (weight)

Fish meals per week (fishmlwk)

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

Methyl Mercury in mg/g (MeHg)

Total Mercury in mg/g (TotHg)

Question 4 - 5 points

You will need to read this data set into memory and may need to process your data before you begin your analysis. The data are in the CSV file "fishermen_mercury.csv". Please provide your code for doing this in the code chunk below. Once you've done this, display the attributes of your data set using the str() function.

```
# Your code for reading in the data and changing variable types, if needed
fishermen = read.csv("fishermen_mercury.csv")
fishermen$fisherman = as.factor(fishermen$fisherman)
fishermen$fishpart = as.factor(fishermen$fishpart)
# Don't forget to display the data set attributes using the str() function!
str(fishermen)
```

```
## 'data.frame':    135 obs. of  9 variables:
##  $ fisherman: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 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 ...
##  $ fishpart : Factor w/ 4 levels "0","1","2","3": 3 2 3 2 2 2 2 3 2 2 ...
##  $ MeHg     : num  4.01 4.03 3.58 10.99 10.52 ...
##  $ TotHg    : num  4.48 4.79 3.86 11.44 10.85 ...
```

Question 5 - 10 points

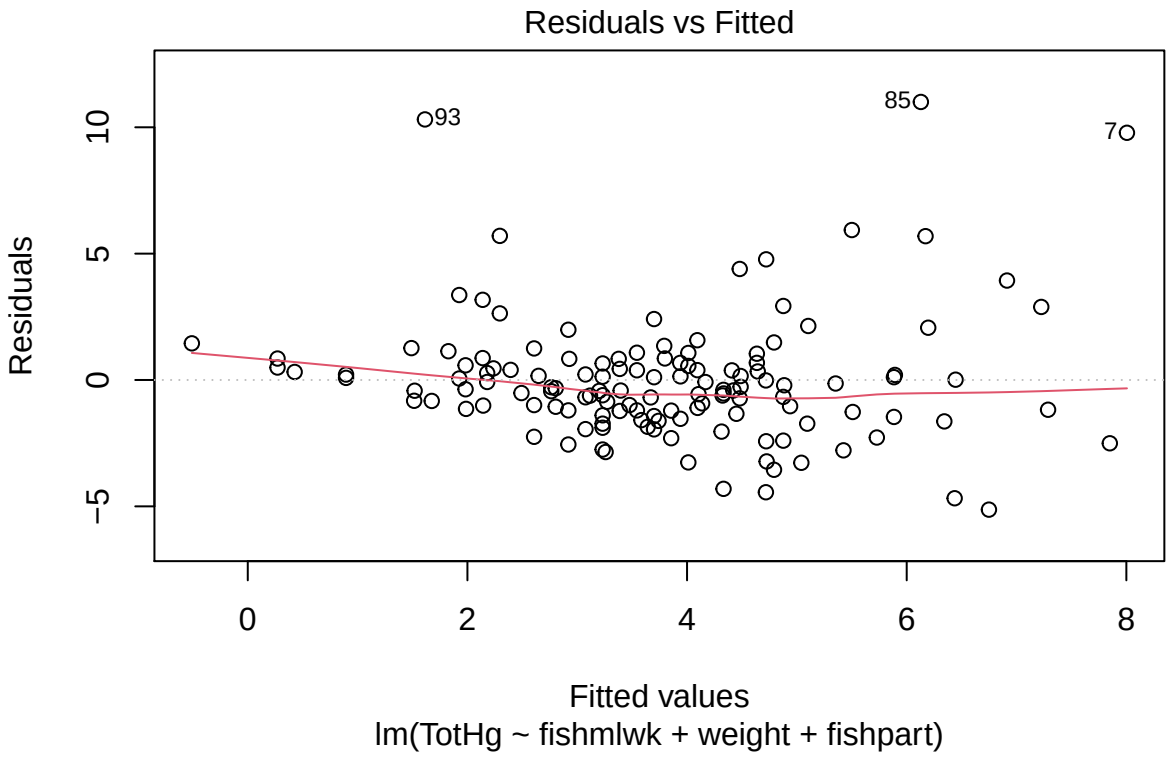
Fit a regression model with "TotHg" as the outcome variable and "fishmlwk" (numeric), "weight" (numeric), and "fishpart" (categorical) as predictor variables. Double-check that these variables are of the proper type before you start your analysis and include any code you used to change these variables in the code chunk in Question 4. Please display the model output using the summary() function.

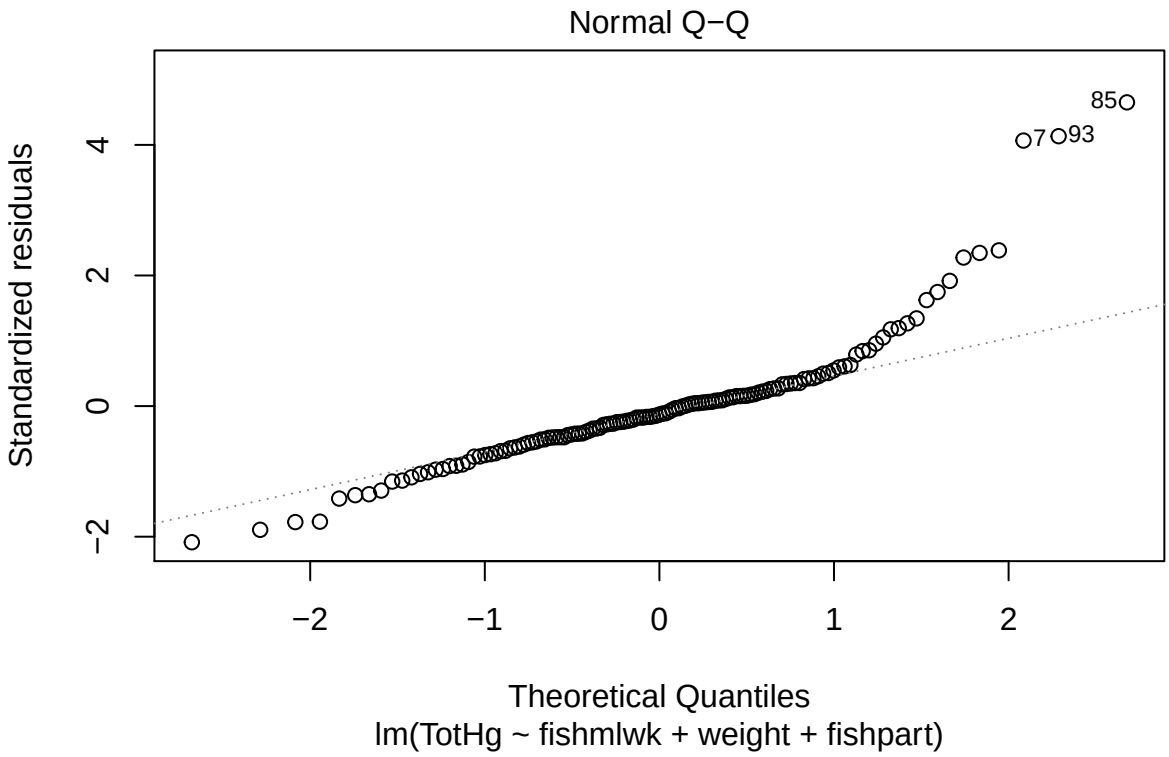
```r
# Code for your regression model and summary() output
fishermen.lmod = lm(TotHg~fishmlwk+weight+fishpart, data=fishermen)
summary(fishermen.lmod)
```

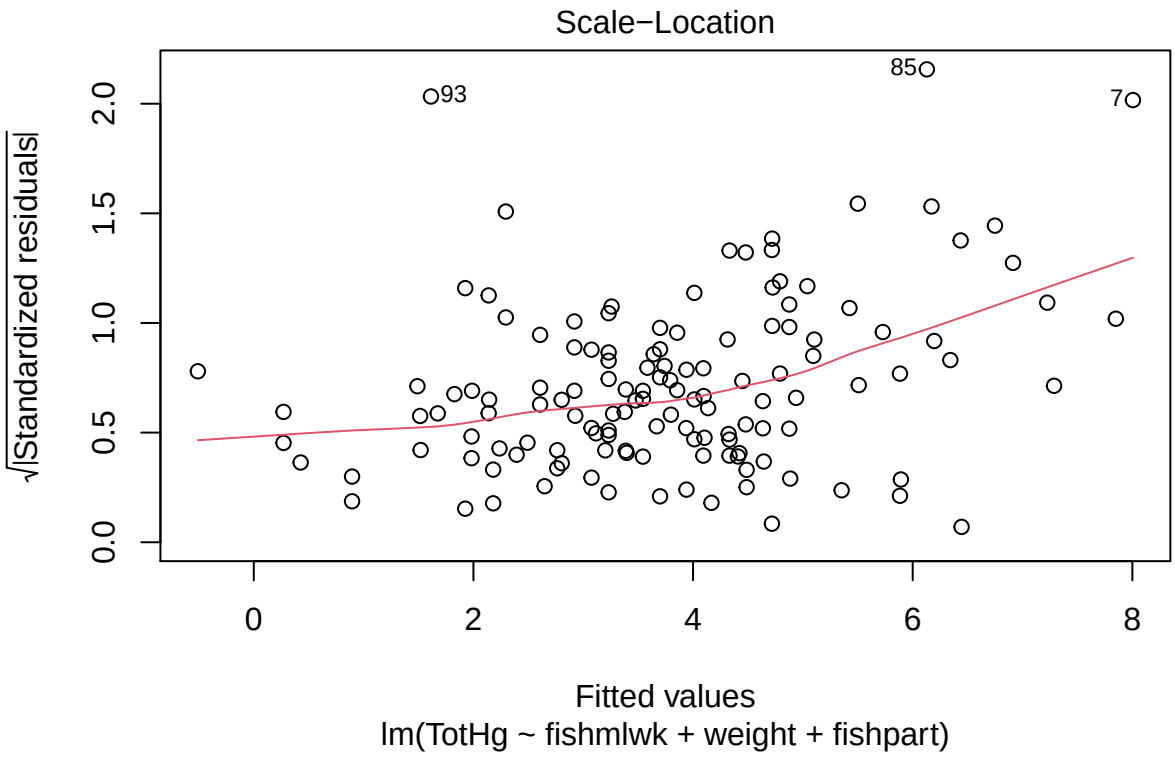
```
##
## Call:
## lm(formula = TotHg ~ fishmlwk + weight + fishpart, data = fishermen)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1298 -1.2455 -0.3262  0.6778 11.0020
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.02782    2.54490  -3.940 0.000133 ***
## fishmlwk      0.12320    0.04440   2.775 0.006347 **
## weight        0.15604    0.03431   4.549 1.23e-05 ***
## fishpart1     2.18255    1.02701   2.125 0.035480 *
## fishpart2     1.47379    0.89973   1.638 0.103854
## fishpart3     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
```

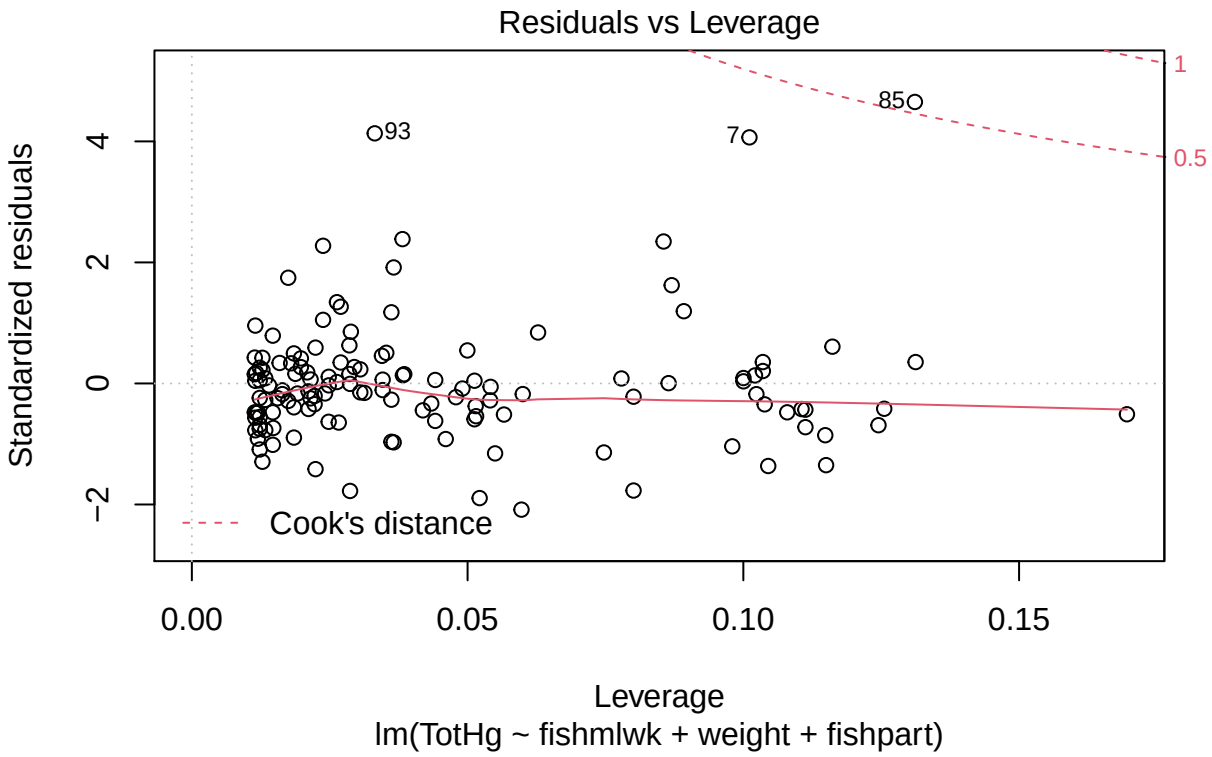
Next, generate the diagnostic plots for this model and comment (1-3 sentences each) on what each plot implies about the model assumptions and/or the presence of data points that are outliers/influential points.

```r
# Code to produce diagnostic plots
plot(fishermen.lmod)
```


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


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(TotHg ~ fishmlwk + weight + fishpart)
85
7
93
−2
0
2
4
−2
−1
0
1
2


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



Comments about residual plot:

Answer here: For the most part, the residual plot looks good. Most of the residuals are near 0 with not obvious structure and follow a near linear curve. There appears to be some "trumpeting," as the variance of residuals is larger at higher fitted values, but it is not too extreme. We can also see that a few data points (at indices 93, 85 and 7) have larger residuals than the rest of the data, so those may be influential points that we should examine closer. Overall, I would say the linearity and homoskedasiticy assumptions are being maintained.

Comments about QQ plot:

Answer here: The residuals within the lowest quantiles fall under the theoretical line and the residuals in the highest quantiles fall well above the theoretical line. This means our distribution of residuals has heavier tales than the theoretical normal distribution, particularly on the right side, than what we would expect from a normal distribution. These exteme tails exist even when we consider our three potentially influential points from the residual plot. Therefor, our data likely violates the normality assumption of linear regression.

Comments about standardized residual plot:

Answer here: This plot shows us a similar result as the first residual plot. The "trumpeting" is more noticeable in this plot than the initial plot, so we may reconsider the validity of the homoskedasticity assumption. And we can still see the three outliers. They're persistent.

Comments about leverage vs. residuals plot:

Answer here: The leverage vs. residuals plot tells us more about our pesky outliers. We can see that point 85 falls outside of the 0.5 Cook's boundary and point 7 comes close to that boundary. Point 93 has a high residual, but not a large leverage, so it's likely not influential. We also see that there are other points in the data that have high leverages, but have small residuals, so they are unlikely to be influential.

Based on what you saw in these plots, are there any observations that you would consider removing? If so, which one/s?

Answer here: Point 85 is the outlier that is most likely to be an influential point, so it could be argued that it should be removed. However, I generally think removing data points just because they don't fit with the rest of the data to be bad practice. Otherwise, we should remove all the data that doesn't fit with what we think our model should be. Because our conclusions about our assumptions wouldn't change if we removed 85 or other points, I would not remove them.

Question 6 - 10 points

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. This process is partially carried out below. The regression is for "TotHg" regressed on all the explanatory variables in the data set restricted to cases in which "fisherman" equals 1. Please complete the process as requested.

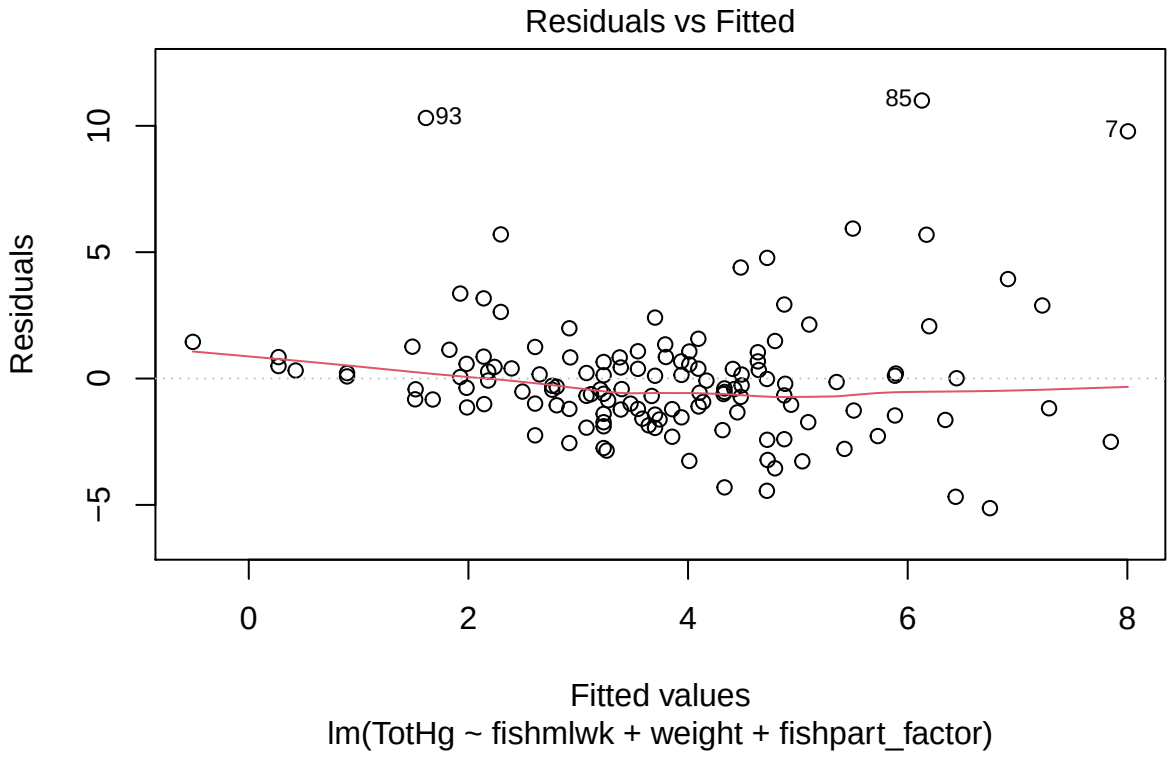
```
fish <- read.csv("fishermen_mercury.csv", header=TRUE, sep=",")  # You may need to change the file path
fish$fishpart_factor <- as.factor(fish$fishpart)
fish.reg = lm(TotHg ~ fishmlwk + weight + fishpart_factor, data=fish)
summary(fish.reg)
```

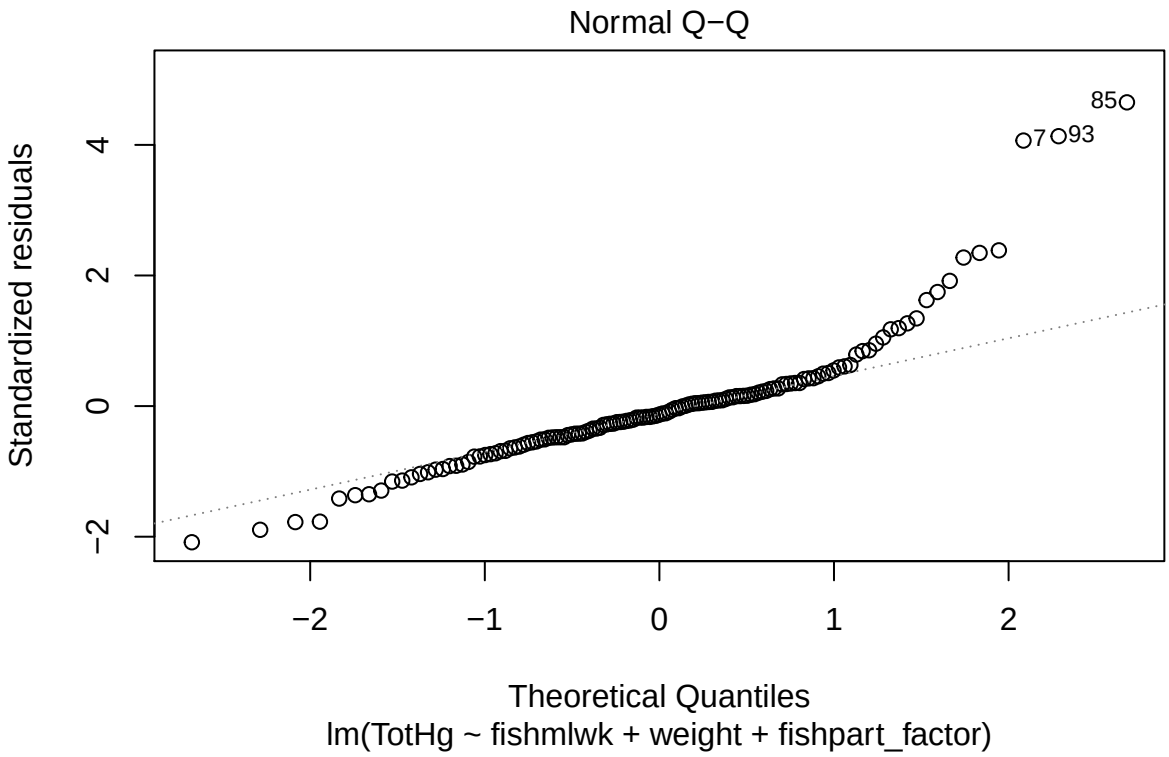
Fit the base model (run this code and examine the output)

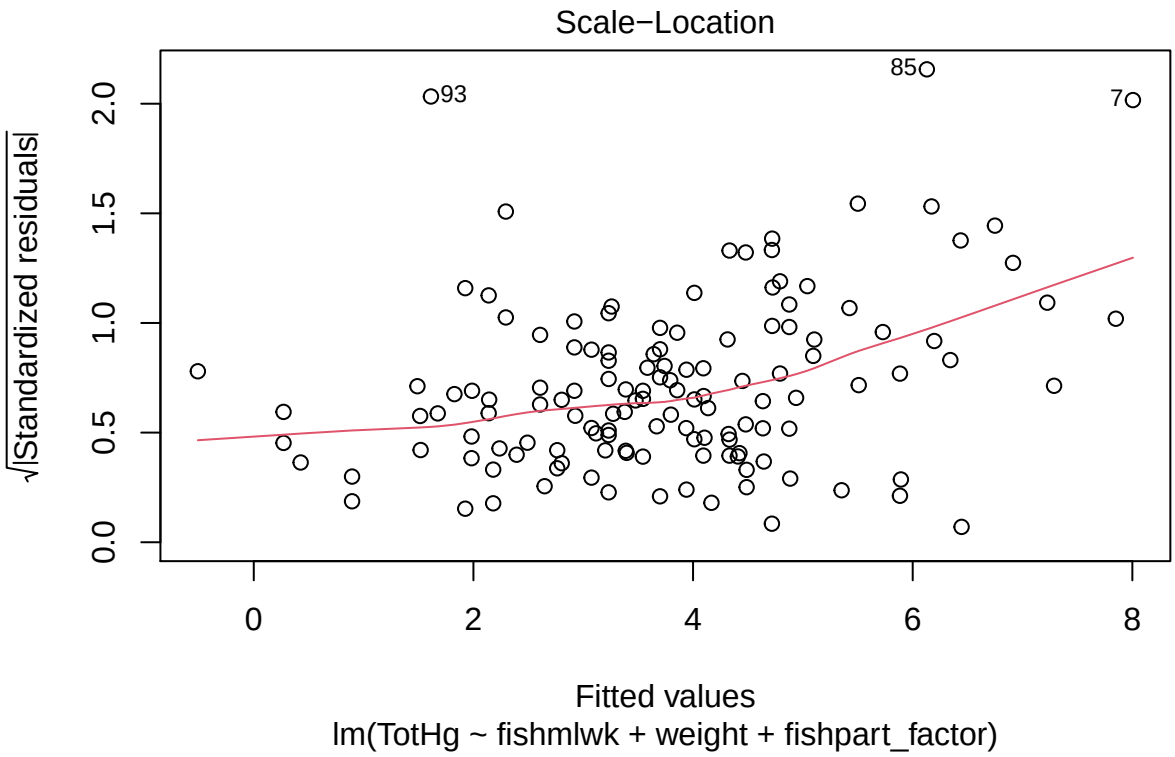
```
##
## Call:
## lm(formula = TotHg ~ fishmlwk + weight + fishpart_factor, data = fish)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1298 -1.2455 -0.3262  0.6778 11.0020
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -10.02782    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
```

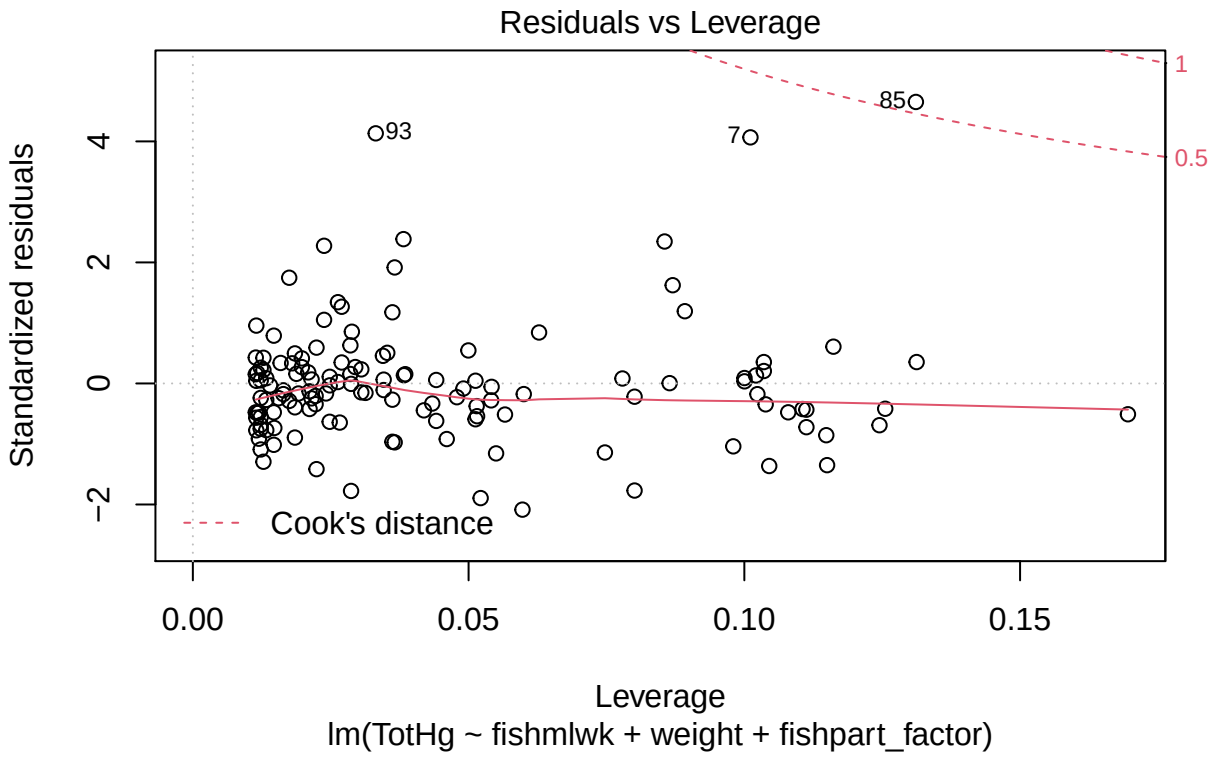
```
##
## Residual standard error: 2.537 on 129 degrees of freedom
## Multiple R-squared:  0.2822, Adjusted R-squared:  0.2543
## F-statistic: 10.14 on 5 and 129 DF,  p-value: 3.296e-08
```

```
plot(fish.reg)
```



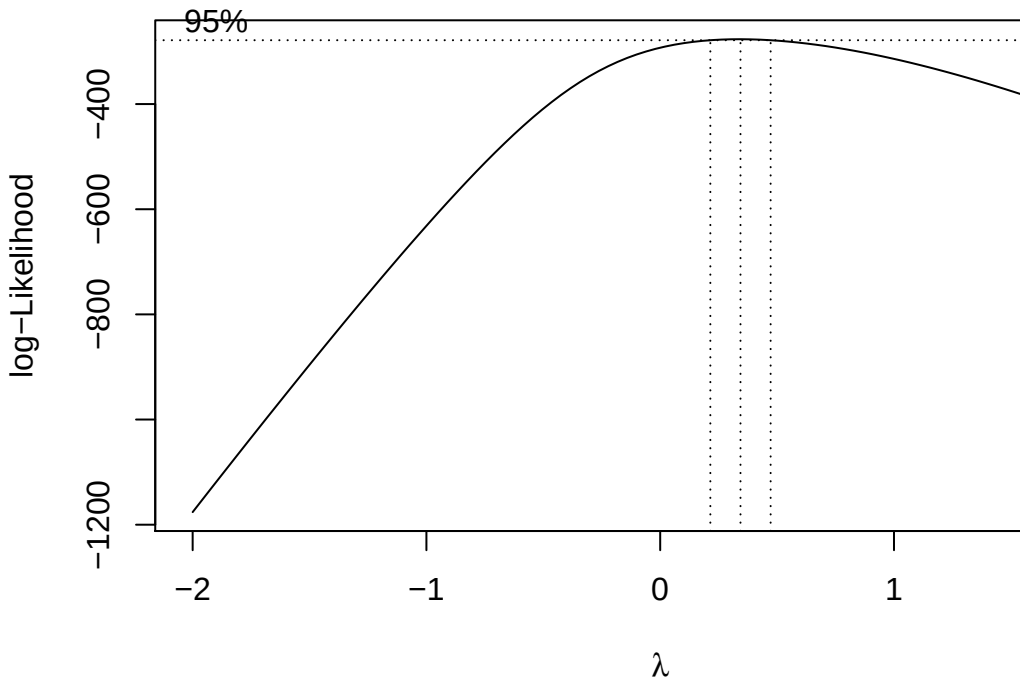

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(TotHg ~ fishmlwk + weight + fishpart_factor)
85
7
93
−2
0
2
4
−2
−1
0
1
2


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




```
library(MASS)
lambda<-boxcox(fish.reg)
```

Run "boxcox" on the base model with default values for the remaining arguments (run this



code and examine the output)

```
ll.best<-which(lambda[[2]]==max(lambda[[2]]))
lambda.best<-lambda[[1]][ll.best]
lambda.best
```

Extract the λ corresponding to the maximum profile log likelihood (run this code and examine the output)

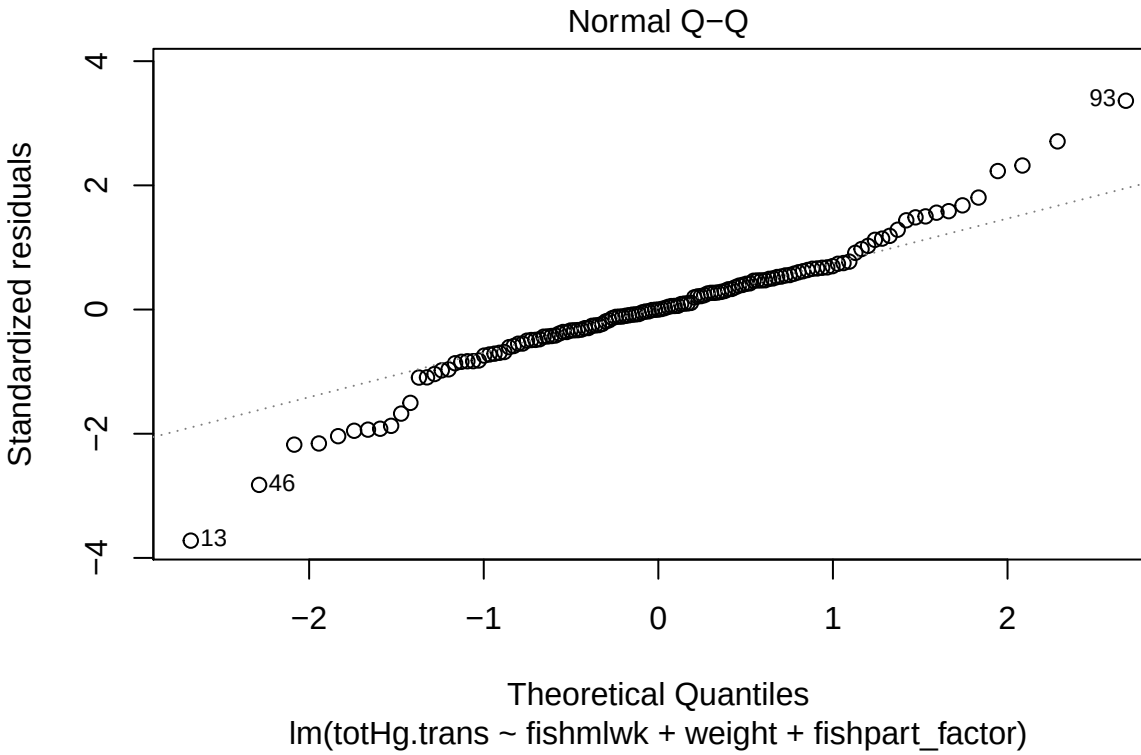
```
## [1] 0.3434343
```

```
# Code for creating transformed outcome variable
y.boxcox = (fish$TotHg^lambda.best - 1) / lambda.best
fish$totHg.trans = y.boxcox

# Code for re-fitting regression model
fish.lmod.boxcox = lm(totHg.trans~fishmlwk+weight+fishpart_factor, data=fish)


# Display diagnostic plots for model with transformed outcome
plot(fish.lmod.boxcox, 2)
```

Refit the model using the transformation of the outcome variable corresponding to this λ and answer the question.



Question: Compare the QQ plot for the Box-Cox transformed model and the QQ plot for the model in the first step of this procedure. Did the Box-Cox transformation noticably improve the normality of the outcome?

Answer: Not really. The right tail is noticably closer to the theoretical normal line, but the left tail is noticably further away. Overall, I would say that the transformed response still does not meet the assumption of normally distributed residuals.