StatR 502 Homework 5

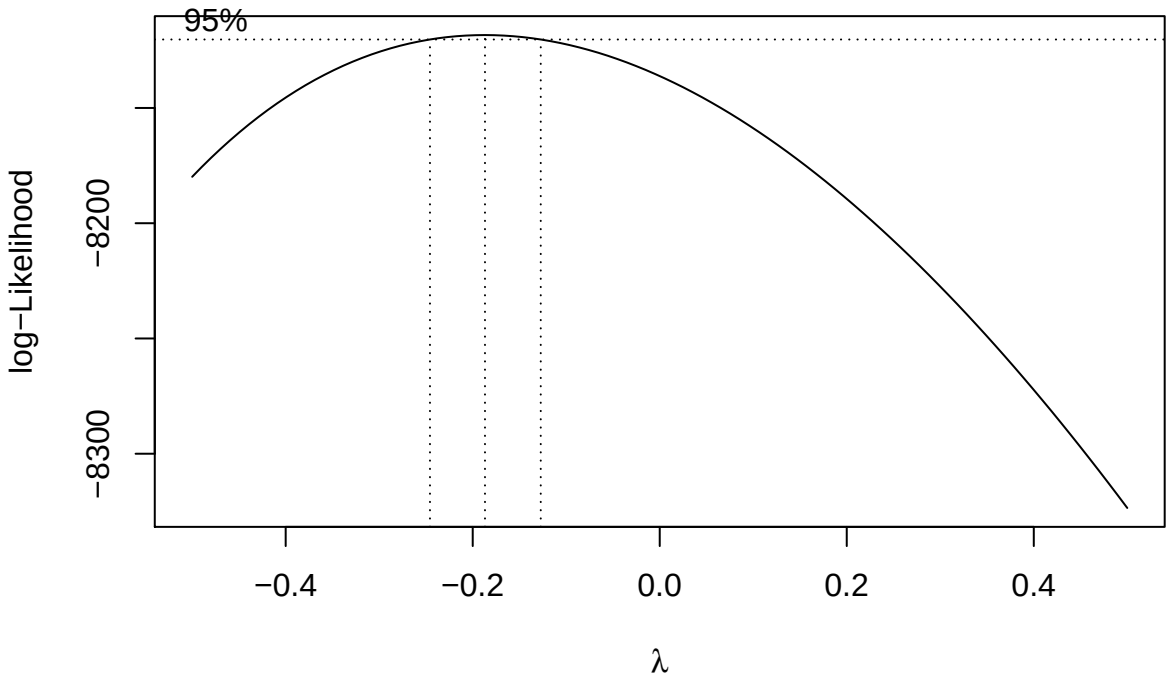
Rebecca Hadi

Due Thursday, Feb. 8, 2018 at 6:30 pm

(1) Abalone Models

(a) Using the abalone data from HW 4, use a Box-Cox test to determine whether or not a transformation of the response variable (number of rings) would be appropriate.

```
library(MASS)
lambda <- boxcox(rings ~ length + width + height + allweight + factor(sex), data = abalone,
                 lambda = seq(-.5, .5, length = 50))
```



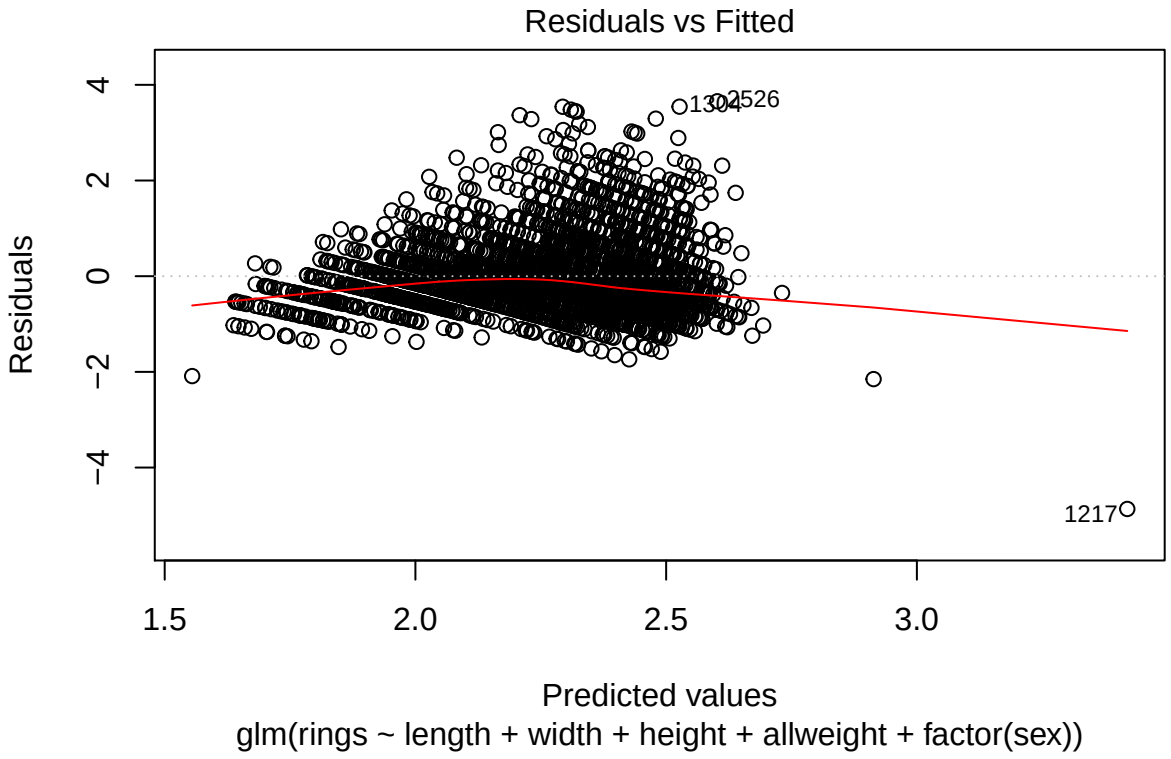
The Box-Cox Plot shows us that the Log Likelihood is maximized around λ = -0.2. This would suggest that a transformation of the response variable is appropriate. Since the λ is negative, the response variable transformation can be transformed into $y^{-0.2}$.

```
#Transformed Model
mod0 <- lm((rings) ~ length + width + height + allweight + factor(sex),
           data = abalone)
mod1 <- lm((rings)^-0.2 ~ length + width + height + allweight + factor(sex),
           data = abalone)
```

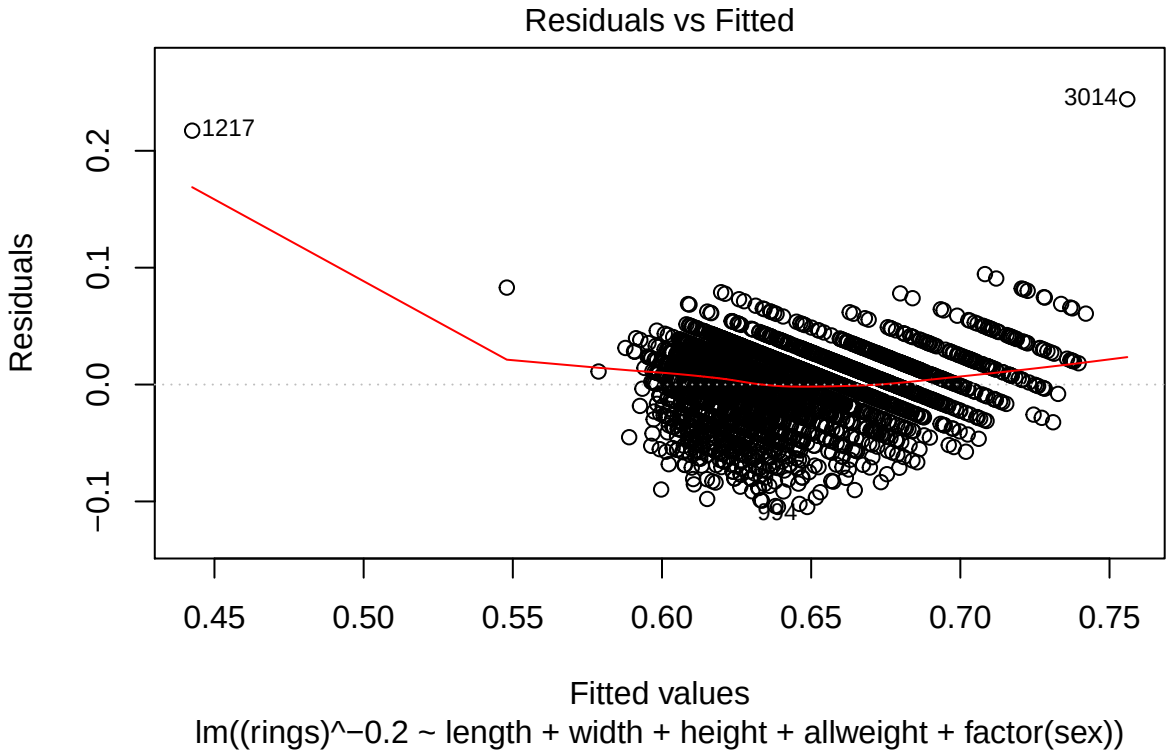
(b) Compare the model fit that results of any transformation you may have done in part (a) with a Poisson GLM modeling the number of rings as counts. Note that AIC/BIC cannot help you compare a Poisson GLM with a Gaussian LM–the different error assumptions make the likelihoods incomparable.

```
pois.mod <- glm(rings ~ length + width + height + allweight + factor(sex),
                data = abalone, family = poisson)

#Compare residual plots
plot(pois.mod,1)
```




```
plot(mod1,1)
```



Comparing the residual plots of the Poisson model (pois.mod) to the transformed linear model (mod1), we see that both seem to have some underlying pattern defining the residuals. However, when we look at the scale, we see that the scale of the residuals for the Box-Cox transformed model ranges between -0.1 and 0.1, which is a much tighter distribution than the Poisson that ranges between -2 and 4. This would imply that

the Box-Cox transformed model has a better fit than the Poisson model.

(c) Starting with whichever model you preferred in part (b), search for "best" models using step wise regression (or other model search strategies such as `leaps`) to find "best" models. Use at least two search methods (e.g., forward, backward, both, using AIC, using BIC) with different starting points. Do you get the same final model with the different methods?

The final model from the backward step AIC is different than the model I get from the forward AIC. In the backward AIC, I started with a model with no interactions and told it to remove terms to improve fit, so the interactions that are part of the forward stepAIC scope were not considered. In the forward stepAIC, some of the interactions improved the model fit and were therefore included.

```
#Backward stepAIC
backward.mod1 <- MASS::stepAIC(mod1, direction = "backward")
```

```
## Start:  AIC=-22444.65
## (rings)^-0.2 ~ length + width + height + allweight + factor(sex)
##
##               Df Sum of Sq    RSS    AIC
## - length       1  0.000009 2.7035 -22447
## <none>                     2.7035 -22445
## - allweight    1  0.059701 2.7632 -22377
## - width        1  0.060810 2.7643 -22376
## - height       1  0.077784 2.7813 -22356
## - factor(sex)  2  0.081748 2.7853 -22354
##
## Step:  AIC=-22446.64
## (rings)^-0.2 ~ width + height + allweight + factor(sex)
##
##               Df Sum of Sq    RSS    AIC
## <none>                     2.7035 -22447
## - allweight    1   0.06177 2.7653 -22377
## - height       1   0.07784 2.7814 -22358
## - factor(sex)  2   0.08218 2.7857 -22356
## - width        1   0.32272 3.0263 -22090
```

```
#Forward stepAIC
forward.mod1 <- MASS::stepAIC(mod1, scope = ~ (length + width + height + allweight + factor(sex))^2, di
```

```
## Start:  AIC=-22444.65
## (rings)^-0.2 ~ length + width + height + allweight + factor(sex)
##
##                          Df Sum of Sq    RSS    AIC
## + length:width            1  0.209418 2.4941 -22699
## + height:factor(sex)      2  0.157779 2.5457 -22632
## + length:factor(sex)      2  0.139243 2.5643 -22609
## + width:factor(sex)       2  0.128214 2.5753 -22595
## + length:height           1  0.111142 2.5924 -22576
## + width:height            1  0.104362 2.5992 -22568
## + length:allweight        1  0.101226 2.6023 -22564
## + allweight:factor(sex)   2  0.096675 2.6069 -22556
## + width:allweight         1  0.091495 2.6120 -22552
## + height:allweight        1  0.024996 2.6785 -22472
## <none>                               2.7035 -22445
##
## Step:  AIC=-22698.8
## (rings)^-0.2 ~ length + width + height + allweight + factor(sex) +
```

```
##     length:width
##
##                          Df Sum of Sq    RSS    AIC
## + height:factor(sex)      2  0.055751 2.4383 -22767
## + allweight:factor(sex)   2  0.035432 2.4587 -22740
## + length:factor(sex)      2  0.034637 2.4595 -22739
## + width:factor(sex)       2  0.028467 2.4656 -22731
## + width:allweight         1  0.008091 2.4860 -22707
## + length:allweight        1  0.005378 2.4887 -22704
## + width:height            1  0.003747 2.4904 -22702
## + length:height           1  0.002430 2.4917 -22700
## <none>                               2.4941 -22699
## + height:allweight        1  0.001508 2.4926 -22699
##
## Step:  AIC=-22766.62
## (rings)^-0.2 ~ length + width + height + allweight + factor(sex) +
##     length:width + height:factor(sex)
##
##                          Df Sum of Sq    RSS    AIC
## + width:allweight         1 0.0105854 2.4278 -22778
## + allweight:factor(sex)   2 0.0081183 2.4302 -22773
## + width:height            1 0.0065435 2.4318 -22773
## + length:allweight        1 0.0064438 2.4319 -22773
## + length:factor(sex)      2 0.0051999 2.4331 -22769
## + length:height           1 0.0036619 2.4347 -22769
## + width:factor(sex)       2 0.0046210 2.4337 -22769
## + height:allweight        1 0.0025832 2.4358 -22768
## <none>                               2.4383 -22767
##
## Step:  AIC=-22778.44
## (rings)^-0.2 ~ length + width + height + allweight + factor(sex) +
##     length:width + height:factor(sex) + width:allweight
##
##                          Df Sum of Sq    RSS    AIC
## + allweight:factor(sex)   2 0.0120582 2.4157 -22790
## + width:height            1 0.0040346 2.4237 -22782
## + length:factor(sex)      2 0.0043511 2.4234 -22780
## + width:factor(sex)       2 0.0042577 2.4235 -22780
## + length:height           1 0.0026493 2.4251 -22780
## <none>                               2.4278 -22778
## + length:allweight        1 0.0010843 2.4267 -22778
## + height:allweight        1 0.0001153 2.4276 -22777
##
## Step:  AIC=-22790.26
## (rings)^-0.2 ~ length + width + height + allweight + factor(sex) +
##     length:width + height:factor(sex) + width:allweight + allweight:factor(sex)
##
##                         Df Sum of Sq    RSS    AIC
## + height:allweight       1 0.0037884 2.4119 -22793
## + width:factor(sex)      2 0.0052800 2.4104 -22793
## <none>                               2.4157 -22790
## + length:allweight       1 0.0004986 2.4152 -22789
## + width:height           1 0.0002948 2.4154 -22789
## + length:factor(sex)     2 0.0017111 2.4140 -22788
```

```
## + length:height       1 0.0000398 2.4157 -22788
##
## Step:  AIC=-22793.25
## (rings)^-0.2 ~ length + width + height + allweight + factor(sex) +
##     length:width + height:factor(sex) + width:allweight + allweight:factor(sex) +
##     height:allweight
##
##                       Df Sum of Sq    RSS    AIC
## + width:height         1 0.0187449 2.3932 -22816
## + length:height        1 0.0174474 2.3945 -22814
## + width:factor(sex)    2 0.0038123 2.4081 -22794
## <none>                             2.4119 -22793
## + length:allweight     1 0.0002238 2.4117 -22792
## + length:factor(sex)   2 0.0009905 2.4109 -22791
##
## Step:  AIC=-22816.04
## (rings)^-0.2 ~ length + width + height + allweight + factor(sex) +
##     length:width + height:factor(sex) + width:allweight + allweight:factor(sex) +
##     height:allweight + width:height
##
##                       Df Sum of Sq    RSS    AIC
## + length:allweight     1 0.0031815 2.3900 -22818
## + length:height        1 0.0016884 2.3915 -22816
## <none>                             2.3932 -22816
## + width:factor(sex)    2 0.0016510 2.3915 -22814
## + length:factor(sex)   2 0.0012416 2.3919 -22814
##
## Step:  AIC=-22818.26
## (rings)^-0.2 ~ length + width + height + allweight + factor(sex) +
##     length:width + height:factor(sex) + width:allweight + allweight:factor(sex) +
##     height:allweight + width:height + length:allweight
##
##                       Df  Sum of Sq    RSS    AIC
## <none>                               2.3900 -22818
## + width:factor(sex)    2 0.00267522 2.3873 -22818
## + length:height        1 0.00014845 2.3898 -22816
## + length:factor(sex)   2 0.00082853 2.3892 -22815
```

(d) Use one of your search methods from (c) on a subset of the data excluding the big outlier. How does it change your results? Do you think it's worth omitting the outlier, or would you prefer another strategy?

```
library(broom)
library(dplyr)
library(ggplot2)

#Model output from stepAIC
step.mod <- lm((rings)^-0.2 ~ width + height + allweight + factor(sex), data = abalone)

##Augment model to find outlier
step.mod.aug <- augment(step.mod, abalone)

#add transformed rings variable to step.mod.aug
step.mod.aug$rings_transform <- (step.mod.aug$rings)^-0.2
```

```r
#Create subset of data without large outlier
step.mod.aug.filter <-  step.mod.aug %>%
                        filter(.cooksd != max(.cooksd))

#Refit mod 1 on this subset of data
mod1.filter <- lm((rings)^-0.2 ~ length + width + height + allweight + factor(sex), data = step.mod.aug

#Perform backward stepAIC on this model with filtered data
backward.mod1.filter <- MASS::stepAIC(mod1.filter, direction = "backward")
```

```
## Start:  AIC=-22560.43
## (rings)^-0.2 ~ length + width + height + allweight + factor(sex)
##
##                Df Sum of Sq    RSS    AIC
## - length        1  0.000024 2.6002 -22562
## <none>                      2.6001 -22560
## - width         1  0.041461 2.6416 -22512
## - factor(sex)   2  0.071535 2.6717 -22478
## - allweight     1  0.094896 2.6950 -22449
## - height        1  0.181037 2.7812 -22349
##
## Step:  AIC=-22562.4
## (rings)^-0.2 ~ width + height + allweight + factor(sex)
##
##                Df Sum of Sq    RSS    AIC
## <none>                      2.6002 -22562
## - factor(sex)   2  0.072277 2.6724 -22479
## - allweight     1  0.098842 2.6990 -22446
## - height        1  0.181074 2.7812 -22351
## - width         1  0.187632 2.7878 -22343
```

```r
step.mod.filter <- lm((rings)^-0.2 ~ width + height + allweight + factor(sex), data = step.mod.aug.filte

#Compare models using AIC
AIC(step.mod, step.mod.filter)
```

```
##                  df       AIC
## step.mod          7 -13428.71
## step.mod.filter   7 -13547.30
```

```r
#Augment filtered model
step.mod.filter.aug  <- augment(step.mod.filter, step.mod.aug.filter)

#Check the max cooksd for this new model
max(step.mod.filter.aug$.cooksd)
```

```
## [1] 0.125106
```

Using the backward stepAIC model on both the filtered and unfiltered data set yields the same predictors. However, the coefficients in the filtered model are differently. Comparing the AIC between the two, the filtered model yields a lower AIC, implying better fit. In this case, I prefer the approach of removing this outlier. Checking the cooksd in the model, there are still values that would fail the $4/n$ test, but not to the extreme extent of the outlier in the unfiltered model (that was 24.05), where as the max Cook's Distance in the filtered model is 0.13.