LAB 5 SOLUTION Stat 230 - MLR Models - Polynomial Regression, Multicollinearity, and Nested F-tests (MLR II)

P.B. Matheson adapted from A.S. Wagaman

Your mission, should you choose to accept it, is to model *diameter* as a function of other measurements of sea snails.

At the end of lab, we will discuss what models you found that you like for predicting diameter. Some steps to help you (items you should be sure to do) are provided. Some example code is provided with generic values, for you to modify.

```
abalone <- read.table("https://pmatheson.people.amherst.edu/stat230/abalone.txt",
    h = T, sep = ",")
```

This data set was stored as a comma-delimited text file (not a .csv), so I read it in as a table, and had to use the sep option to tell it about the commas.

This data set contains information on 4177 abalones (sea snails). The variables included are:
* Sex - M, F, and I (infant) - Note there are THREE levels here
* Length - mm - Longest shell measurement
* Diameter - mm - perpendicular to length
* Height - mm - with meat in shell
* Whole weight - grams - whole abalone
* Shucked weight - grams - weight of meat
* Viscera weight - grams - gut weight
* Shell weight - grams - after being dried
* Rings - number of rings in the shell - 1.5+rings gives the age in years

Note that the descriptions provided do not exactly match the variable names - whoever entered the data shortened the names somewhat. To see the names of variables in the data set, use:

```
names(abalone)
```

```
## [1] "Sex"       "Length"    "Diameter"  "Height"    "Wholewt"   "Shuckedwt"
## [7] "Viscerawt" "Shellwt"   "Rings"
```

The weight variables all have weight abbreviated as wt.

> Take a look at the variables in your data set, basic summaries, etc.

SOLUTION:

```
glimpse(abalone)
```

```
## Rows: 4,177
## Columns: 9
## $ Sex       <chr> "M", "M", "F", "M", "I", "I", "F", "F", "M", "F", "F", "M", ~
## $ Length    <dbl> 0.455, 0.350, 0.530, 0.440, 0.330, 0.425, 0.530, 0.545, 0.47~
## $ Diameter  <dbl> 0.365, 0.265, 0.420, 0.365, 0.255, 0.300, 0.415, 0.425, 0.37~
## $ Height    <dbl> 0.095, 0.090, 0.135, 0.125, 0.080, 0.095, 0.150, 0.125, 0.12~
## $ Wholewt   <dbl> 0.5140, 0.2255, 0.6770, 0.5160, 0.2050, 0.3515, 0.7775, 0.76~
## $ Shuckedwt <dbl> 0.2245, 0.0995, 0.2565, 0.2155, 0.0895, 0.1410, 0.2370, 0.29~
## $ Viscerawt <dbl> 0.1010, 0.0485, 0.1415, 0.1140, 0.0395, 0.0775, 0.1415, 0.14~
## $ Shellwt   <dbl> 0.150, 0.070, 0.210, 0.155, 0.055, 0.120, 0.330, 0.260, 0.16~
## $ Rings     <int> 15, 7, 9, 10, 7, 8, 20, 16, 9, 19, 14, 10, 11, 10, 10, 12, 7~
```

We can verify that we only have one categorical variable. We see that Rings is all integers.

> Obtain the correlation matrix for your data set. Determine if there may be possible issues with multicollinearity for MLR models, and also determine what predictors may be a good choice to fit a SLR predicting diameter width.

SOLUTION:

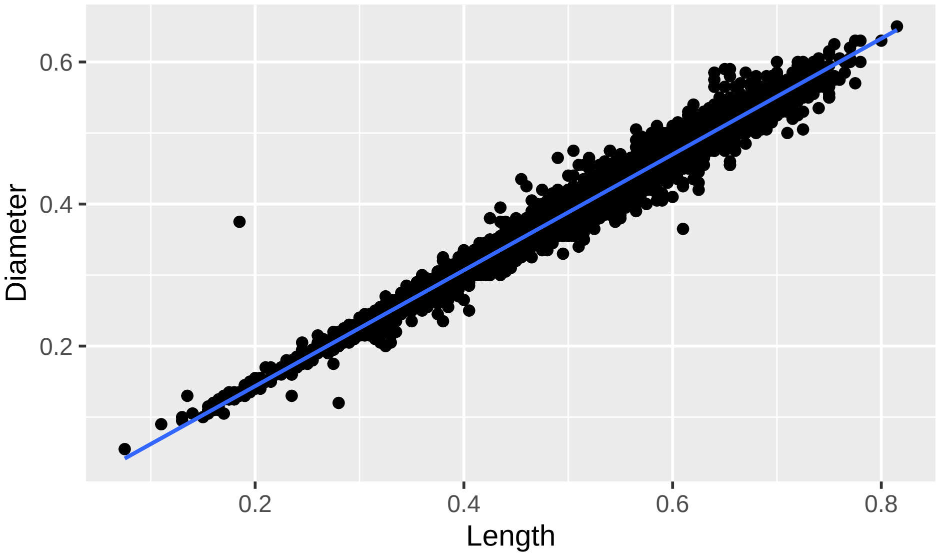
```r
cor(select(abalone, -Sex)) #remove Sex
```

```
##                Length   Diameter     Height    Wholewt  Shuckedwt  Viscerawt
## Length    1.00000000 0.98681158 0.82755361 0.92526117 0.89791366 0.90301770
## Diameter  0.98681158 1.00000000 0.83368369 0.92545210 0.89316248 0.89972443
## Height    0.82755361 0.83368369 1.00000000 0.81922077 0.77497229 0.79831930
## Wholewt   0.92526117 0.92545210 0.81922077 1.00000000 0.96940546 0.96637508
## Shuckedwt 0.89791366 0.89316248 0.77497229 0.96940546 1.00000000 0.93196132
## Viscerawt 0.90301770 0.89972443 0.79831930 0.96637508 0.93196132 1.00000000
## Shellwt   0.89770557 0.90532978 0.81733801 0.95535544 0.88261706 0.90765632
## Rings     0.55671958 0.57465985 0.55746732 0.54038968 0.42088366 0.50381925
##               Shellwt      Rings
## Length    0.89770557 0.55671958
## Diameter  0.90532978 0.57465985
## Height    0.81733801 0.55746732
## Wholewt   0.95535544 0.54038968
## Shuckedwt 0.88261706 0.42088366
## Viscerawt 0.90765632 0.50381925
## Shellwt   1.00000000 0.62757404
## Rings     0.62757404 1.00000000
```

The correlation matrix reveals some very strong correlations between the quantitative variables in the data set. Given that some of these reach 0.9 on their own, we would be wary of multicollinearity issues. We can see the weights are strongly correlated with each other. Remember that there can be multicollinearity issues that aren't the result of just pairwise associations, but if you see correlations of 0.9 or stronger (in magnitude) between predictors, you'll get that R^2 of 0.8 and VIF of > 5 just from that.

In terms of fitting an SLR to predict diameter, we focus on its correlations with the other variables. Note that a high correlation (in magnitude) doesn't mean we have a linear relationship! We still have to check that. Still, it looks like Length has the strongest relationship with Diameter, so let's see if that relationship is linear.

```r
gf_point(Diameter ~ Length, data = abalone) %>%
   gf_lm()
```



This relationship is linear so we could fit an SLR.

> Plot and fit a SLR predicting diameter using a predictor you think is appropriate. Check your model conditions. If there are issues with the conditions, would transformations of variables be appropriate to help? What transformations?

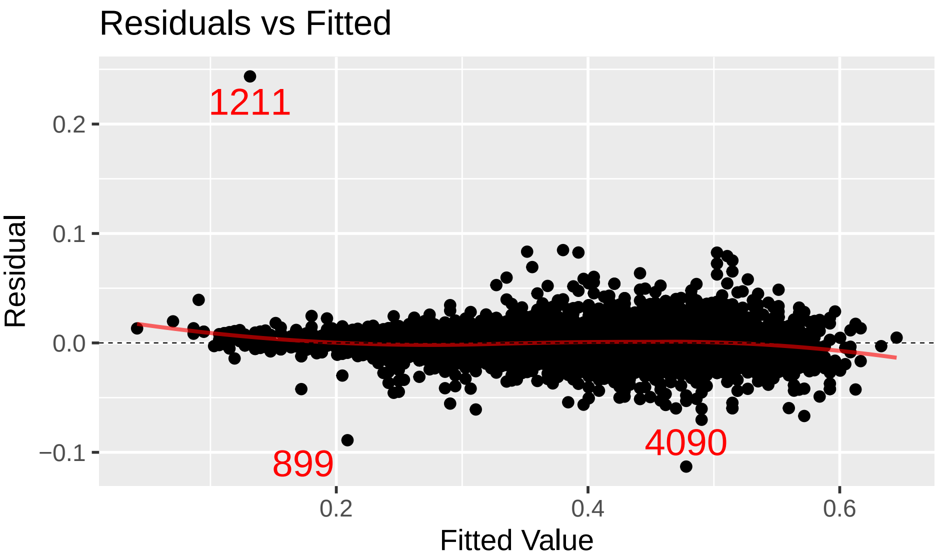
SOLUTION:

```
fm1 <- lm(Diameter ~ Length, data = abalone)
msummary(fm1)
```

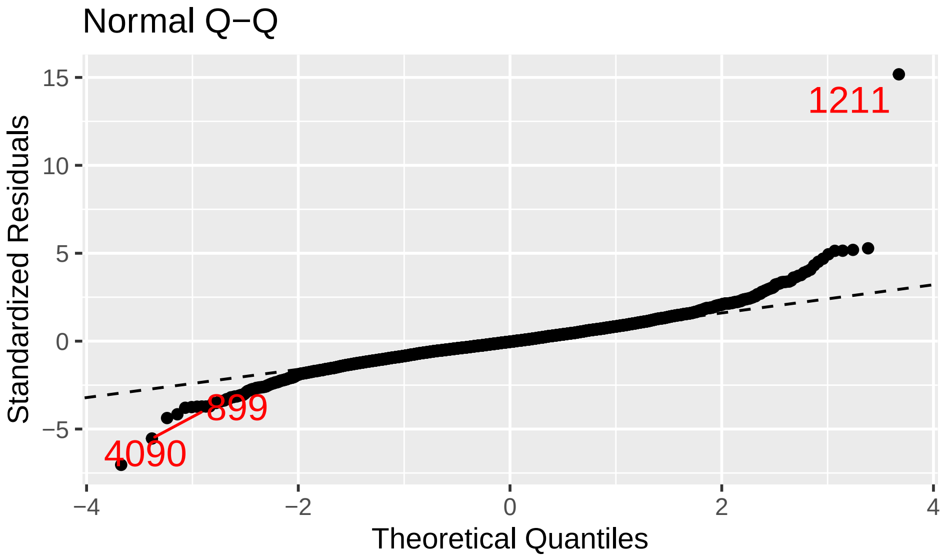
```
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept) -0.0194137  0.0011129 -17.444 < 2.2e-16 ***
## Length       0.8154607  0.0020702 393.902 < 2.2e-16 ***
##
## Residual standard error: 0.016066 on 4175 degrees of freedom
## Multiple R-squared:  0.9738, Adjusted R-squared:  0.97379
## F-statistic: 1.5516e+05 on 1 and 4175 DF,  p-value: < 2.22e-16
```

```
mplot(fm1, which = 1)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
mplot(fm1, which = 2)
```

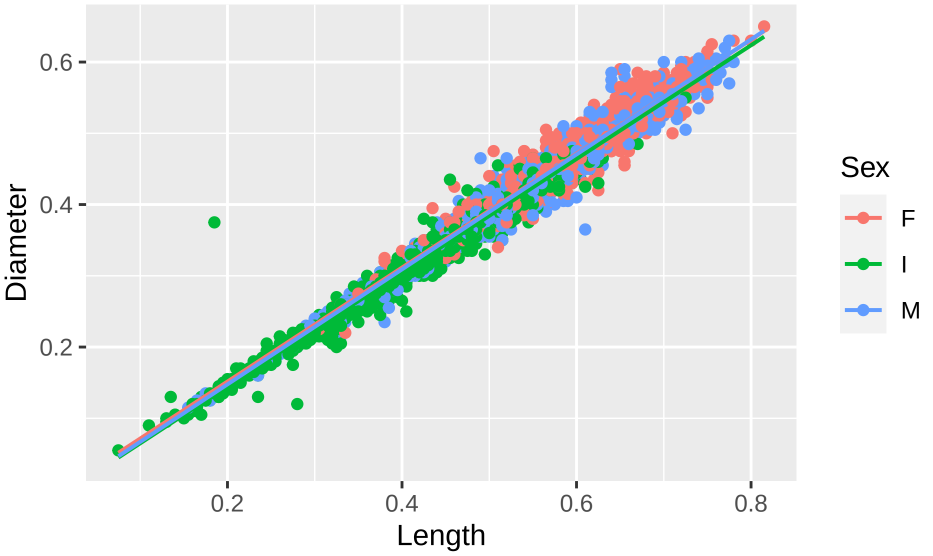


The model is significant, and we have very small residual standard error on the original scale of Diameter (like 1/50th of the scale). However, the residuals versus fitted plot shows some increasing variability, and the QQplot shows the residuals deviating from the expected line for normality in both tails, pretty badly. There is definitely an outlier (observation 1211). A sqrt or log transformation of Diameter (or maybe of Length, but I'd start with Diameter) might help here - we'd have to try and see.

> Add an interaction term between Sex and your chosen SLR predictor to the model (and save it with a new name). Practice interpreting the coefficients in this model.

SOLUTION:

```
gf_point(Diameter ~ Length, color = ~Sex, data = abalone) %>% gf_lm()
```

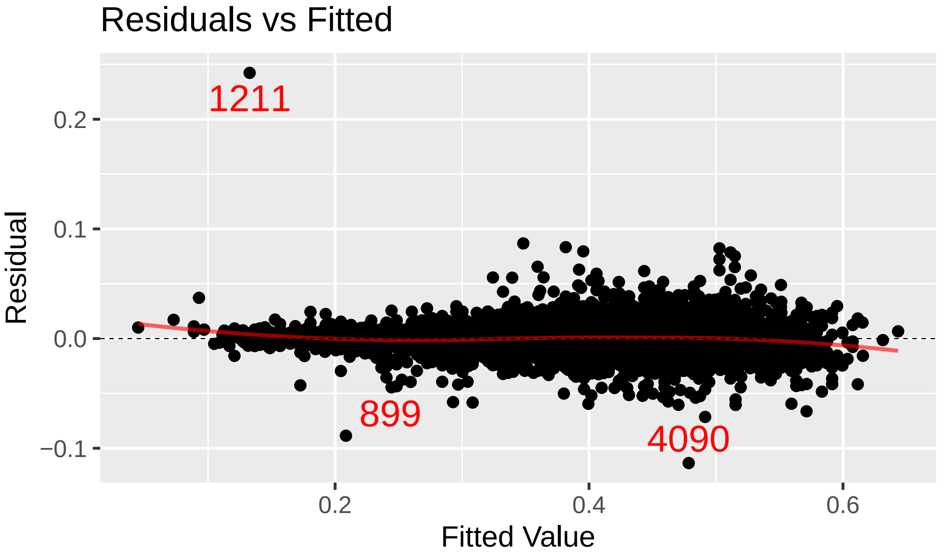


```
#note the following code makes R also include Length and Sex as main effects
fm2 <- lm(Diameter ~ Length*Sex, data = abalone)
msummary(fm2)
```

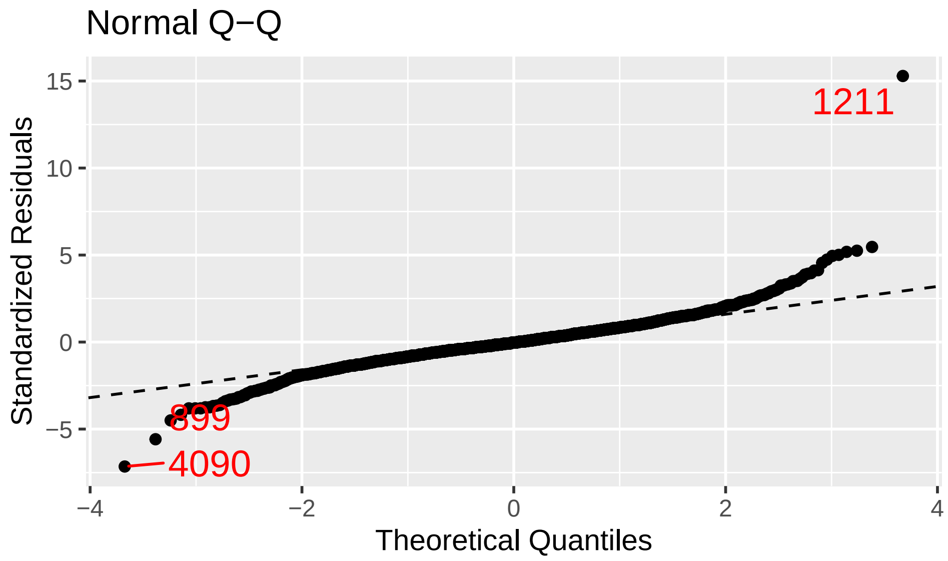
```
##                Estimate Std. Error  t value  Pr(>|t|)
## (Intercept) -0.0083718  0.0029861  -2.8036  0.005077 **
## Length       0.7997052  0.0051004 156.7930 < 2.2e-16 ***
## SexI        -0.0065418  0.0034653  -1.8878  0.059124 .
## SexM        -0.0053529  0.0037440  -1.4297  0.152866
## Length:SexI -0.0015500  0.0064719  -0.2395  0.810727
## Length:SexM  0.0072396  0.0064556   1.1214  0.262162
##
## Residual standard error: 0.015881 on 4171 degrees of freedom
## Multiple R-squared:  0.97442,    Adjusted R-squared:  0.97439
## F-statistic:  31780 on 5 and 4171 DF,  p-value: < 2.22e-16
```

```
mplot(fm2, which = 1)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
mplot(fm2, which = 2)
```



An example plot and model fit is shown. The diagnostic plots still suggest a transformation might be helpful, but let's focus on understanding the coefficients here, instead of that.

The main effect for Length (the only significant predictor in this model) tells us that for each 1 mm increase in Length for FEMALE (the reference group) sea snails, we expect the diameter to increase on average by 0.7997 mm, with the effects of Sex and its interactions with Length accounted for.

NOTE: We would not keep all these predictors in the model if they are nonsignficant; we would take them out and rerun to interpret the final model that would have just length at this point.

FOR LEARNING PURPOSES, lets interpret the coefficients anyway. The main effect for SexI tells us that for infant sea snails, relative to female sea snails, the diameter would be 0.0065 mm less on average when at comparable lengths, with the effects of Length and it's interactions with Sex accounted for. SexM has a similar interpretation, still relative to FEMALE sea snails (because "F" is the reference level - out of 3 levels (coded as F, I, and M), F is first in alphabetical order which is how R picked).

Finally, we examine the interaction effects. The interaction Length:SexI is the adjustment in the slope for length that we see for infant sea snails relative to female sea snails. So, for infant sea snails, relative to female sea snails, if Length increases by 1 mm there is an associated decrease of 0.0016 mm in diameter on average beyond the increase of 0.7997 mm expected for the female sea snails, accounting for the effects of the other predictors. (You don't need to refer back to the reference level, but here, I thought it useful to see how similar the slopes are).

> Throw in the kitchen sink. Use all predictors to predict diameter - just their main effects, no interaction terms. How well does the model appear to fit?

SOLUTION:

```
fm.full <- lm(Diameter ~ ., data = abalone)
msummary(fm.full)
```

```
##                Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept)  0.00308399  0.00206421   1.4940   0.13524
## SexI        -0.00451820  0.00071216  -6.3444 2.472e-10 ***
## SexM        -0.00104023  0.00057779  -1.8004   0.07188 .
## Length       0.72374214  0.00562993 128.5526 < 2.2e-16 ***
## Height       0.06720591  0.01066498   6.3015 3.253e-10 ***
## Wholewt      0.00173556  0.00512197   0.3388   0.73474
## Shuckedwt    0.00716226  0.00605255   1.1833   0.23674
## Viscerawt   -0.01948487  0.00903848  -2.1558   0.03116 *
## Shellwt      0.05122375  0.00781585   6.5538 6.291e-11 ***
## Rings        0.00053260  0.00010711   4.9725 6.876e-07 ***
##
## Residual standard error: 0.015216 on 4167 degrees of freedom
## Multiple R-squared:  0.97654,    Adjusted R-squared:  0.97649
## F-statistic:  19275 on 9 and 4167 DF,  p-value: < 2.22e-16
```

The full model (fm.full) appears to predict diameter really well. The R-squared is almost 0.98 and the residual SE is 0.015 on a scale of about 0.6, which is really a small fraction. We haven't checked conditions though.

> Use a nested F-test to test between your SLR and kitchen sink models (why wouldn't it work to compare the kitchen sink and the interaction model?). Be sure to explicitly state what hypotheses are being tested (can you identify the subset of variables being tested?), as well as what your conclusion is.

SOLUTION:

```
anova(fm1, fm.full)
```

```
## Analysis of Variance Table
##
## Model 1: Diameter ~ Length
## Model 2: Diameter ~ Sex + Length + Height + Wholewt + Shuckedwt + Viscerawt +
##     Shellwt + Rings
##   Res.Df      RSS Df Sum of Sq       F     Pr(>F)
## 1   4175 1.077661
## 2   4167 0.964726  8  0.112934 60.9756 < 2.22e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interaction model and kitchen sink model are not nested so that is why that comparison wouldn't work.

Comparing these two models, the SLR and kitchen sink models, the null for the nested F-test is that every predictor has a slope of 0 except for Length, whose slope is unspecified. The alternative is that at least one of the added predictors in fm.full has a coefficient which is significantly different from 0.

The F statistic is 60.9756 and the p-value is the smallest R can report (less than 2.2 times 10ˆ-16). This is a significant result. So, we reject the null and conclude that at least one of the main effects aside from Length is significant for predicting Diameter. This means we'd use the full model over the SLR.

> Examine your full model for issues of multicollinearity by computing the VIFs. Explain how you can tell if VIFs are unusual and determine how you should proceed based on your results. Be sure to call the car library as shown.

SOLUTION:

```
car::vif(fm.full)
```

```
##                  GVIF Df GVIF^(1/(2*Df))
## Sex         1.5618659  2       1.1179205
## Length      8.2455616  1       2.8715086
## Height      3.5893413  1       1.8945557
## Wholewt   113.7983790  1      10.6676323
## Shuckedwt  32.5550061  1       5.7056994
## Viscerawt  17.7053369  1       4.2077710
## Shellwt    21.3514408  1       4.6207619
## Rings       2.1511963  1       1.4666957
```

GVIF stands for generalized VIF. This is because the model has a categorical variable in it. The concepts are the same though. We suspect issues with VIFs greater than 5 (including GVIFs). You can ignore the final column - look at the GVIF column. Here, there are clearly issues. The weight variables have very high values, and even length has a VIF near 8. So, we need to remove some variables. We could try removing three of the four weight variables to see if that helps, or maybe all the weight variables. Length is more easily obtained than weight in this circumstance since shelling must be done, so we might try to keep that variable, since we know it was a strong predictor on its down.

> Build two other models that you are interested in considering for predicting diameter using either interaction terms or higher powers. Make these two models nested (one within the other). Is either model nested inside the full model above?

SOLUTION:

```
fm3 <- lm(Diameter ~ Length + Height + Rings + Sex, data = abalone)
fm4 <- lm(Diameter ~ Length*Sex + Height*Sex + Rings*Sex, data = abalone)
```

For my two models, I made one with several pairwise interactions with Sex and the other is just the main effects of those predictors.

We could also fit a model with a higher power:

```
fm5 <- lm(Diameter ~ Length + Height + Rings + I(Rings^2) + Sex, data = abalone)
#adds Rings^2 to the model
```

The fm3 model I have is nested in fm.full. The other two models (fm1 & fm2) are not nested in fm.full. fm3 is nested in both fm4 and fm5, but fm4 and fm5 are not nested witin each other.

> Perform a nested F-test to compare the two models you choose to examine previously. Be sure you can explicitly state what subset of variables is being tested, and what the conclusion is.

SOLUTION:

```
anova(fm3, fm4)
```

```
## Analysis of Variance Table
##
## Model 1: Diameter ~ Length + Height + Rings + Sex
## Model 2: Diameter ~ Length * Sex + Height * Sex + Rings * Sex
##   Res.Df      RSS Df Sum of Sq       F    Pr(>F)
## 1   4171 1.003018
## 2   4165 0.991965  6 0.0110523 7.73426 2.7372e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our null hypothesis is that all of the interaction terms in model 2 (fm4) have slopes of 0. The alternative is that at least one interaction term has a non-zero slope.

The result here (assuming conditions hold) suggests that we want to keep at least one of the interaction terms, so we need to look at the bigger model (fm4).

> Continue to fit models until you find one you are happy (at least somewhat happy) with. At that time, report your final fitted model for predicting diameter.

SOLUTION:

First, we consider if there are other predictors we want to add.

```
cor(select(abalone, Length, Wholewt, Shuckedwt, Viscerawt, Shellwt))
```

```
##               Length    Wholewt  Shuckedwt  Viscerawt    Shellwt
## Length    1.00000000 0.92526117 0.89791366 0.90301770 0.89770557
## Wholewt   0.92526117 1.00000000 0.96940546 0.96637508 0.95535544
## Shuckedwt 0.89791366 0.96940546 1.00000000 0.93196132 0.88261706
## Viscerawt 0.90301770 0.96637508 0.93196132 1.00000000 0.90765632
## Shellwt   0.89770557 0.95535544 0.88261706 0.90765632 1.00000000
```

Length has high enough correlations with each weight related variable for there to be serious multicollinearity just from the pairs (not even linear combos) alone. So, we'll ONLY add Length to the model and not use any weight variables.

Let's create a model 6 instead of our model 4 and try to remove the interaction terms in the model.

```
fm6 <- lm(Diameter ~ Length*Sex + Height*Sex + Rings, data = abalone)
msummary(fm6)
```

```
##                  Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept) -1.3815e-02  2.9619e-03  -4.6644 3.193e-06 ***
## Length       7.7939e-01  5.9715e-03 130.5184 < 2.2e-16 ***
## SexI        -4.8379e-04  3.4054e-03  -0.1421   0.88703
## SexM        -2.4433e-03  3.6452e-03  -0.6703   0.50272
## Height       5.3781e-02  1.2872e-02   4.1781 3.000e-05 ***
## Rings        7.8277e-04  9.3688e-05   8.3551 < 2.2e-16 ***
## Length:SexI -5.7592e-02  1.1013e-02  -5.2297 1.782e-07 ***
## Length:SexM -2.4962e-02  9.1542e-03  -2.7269   0.00642 **
## SexI:Height  1.8578e-01  3.4162e-02   5.4383 5.687e-08 ***
## SexM:Height  1.0237e-01  2.4270e-02   4.2181 2.517e-05 ***
##
## Residual standard error: 0.015433 on 4167 degrees of freedom
## Multiple R-squared:  0.97587,    Adjusted R-squared:  0.97582
## F-statistic:  18724 on 9 and 4167 DF,  p-value: < 2.22e-16
```

```
fm7 <- lm(Diameter ~ Length*Sex + Height + Rings, data = abalone)
anova(fm7, fm6)
```

```
## Analysis of Variance Table
##
## Model 1: Diameter ~ Length * Sex + Height + Rings
## Model 2: Diameter ~ Length * Sex + Height * Sex + Rings
##   Res.Df      RSS Df  Sum of Sq       F     Pr(>F)
## 1   4169 1.001889
## 2   4167 0.992459  2 0.00942948 19.7956 2.7763e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#significant so need Height*Sex

fm8 <- lm(Diameter ~ Length + Height*Sex + Rings, data = abalone)
anova(fm8, fm6)
```

```
## Analysis of Variance Table
##
## Model 1: Diameter ~ Length + Height * Sex + Rings
## Model 2: Diameter ~ Length * Sex + Height * Sex + Rings
##   Res.Df      RSS Df  Sum of Sq       F     Pr(>F)
## 1   4169 0.999187
## 2   4167 0.992459  2 0.00672791 14.1241 7.7033e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#again, need Length*Sex
```

Because Sex has 3 levels, each of these interactions is giving us 2 terms to test, which is why a nested F-test was used. You can see from the individual t-tests in fm6 that those terms are also both significant for both pairs tested.

It looks like we can't remove any interaction terms. What else can we check?

```
car::vif(fm6)
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## Length        9.0171500  1       3.0028570
## Sex         927.5159664  2       5.5186182
## Height        5.0826443  1       2.2544721
## Rings         1.5998252  1       1.2648420
## Length:Sex 7032.0657201  2       9.1573693
## Sex:Height 2600.9102830  2       7.1413668
```

We KNOW these VIFs will be high because we have interaction terms in the model. You'd get the same issue with higher powers of predictors. So, because of our model structure, this isn't too concerning.

Now, what about a model with just Length*Sex compared to model 6? Well, we have that above as our model 2.

```
anova(fm2, fm6)
```

```
## Analysis of Variance Table
##
## Model 1: Diameter ~ Length * Sex
## Model 2: Diameter ~ Length * Sex + Height * Sex + Rings
##   Res.Df      RSS Df Sum of Sq       F    Pr(>F)
## 1   4171 1.051955
## 2   4167 0.992459  4 0.0594957 62.4506 < 2.22e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The significant result means we need to use the larger model, fm6.

That said, let's compare their model summaries:

```
msummary(fm2)
```

```
##                Estimate Std. Error  t value  Pr(>|t|)
## (Intercept) -0.0083718  0.0029861  -2.8036  0.005077 **
## Length       0.7997052  0.0051004 156.7930 < 2.2e-16 ***
## SexI        -0.0065418  0.0034653  -1.8878  0.059124 .
## SexM        -0.0053529  0.0037440  -1.4297  0.152866
## Length:SexI -0.0015500  0.0064719  -0.2395  0.810727
## Length:SexM  0.0072396  0.0064556   1.1214  0.262162
##
## Residual standard error: 0.015881 on 4171 degrees of freedom
## Multiple R-squared:  0.97442,    Adjusted R-squared:  0.97439
## F-statistic:  31780 on 5 and 4171 DF,  p-value: < 2.22e-16
```

```
msummary(fm6)
```

```
##                 Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept) -1.3815e-02  2.9619e-03  -4.6644 3.193e-06 ***
## Length       7.7939e-01  5.9715e-03 130.5184 < 2.2e-16 ***
## SexI        -4.8379e-04  3.4054e-03  -0.1421   0.88703
## SexM        -2.4433e-03  3.6452e-03  -0.6703   0.50272
## Height       5.3781e-02  1.2872e-02   4.1781 3.000e-05 ***
## Rings        7.8277e-04  9.3688e-05   8.3551 < 2.2e-16 ***
## Length:SexI -5.7592e-02  1.1013e-02  -5.2297 1.782e-07 ***
## Length:SexM -2.4962e-02  9.1542e-03  -2.7269   0.00642 **
## SexI:Height  1.8578e-01  3.4162e-02   5.4383 5.687e-08 ***
## SexM:Height  1.0237e-01  2.4270e-02   4.2181 2.517e-05 ***
##
## Residual standard error: 0.015433 on 4167 degrees of freedom
## Multiple R-squared:  0.97587,    Adjusted R-squared:  0.97582
## F-statistic:  18724 on 9 and 4167 DF,  p-value: < 2.22e-16
```

We get a TINY improvement in the residual standard error (fourth decimal place), and a correspondingly tiny R^2 improvement in the third decimal place between these two models. And we had to add 4 terms to get that. The simpler model is already doing a really good job. What about just the SLR?

```
anova(fm1, fm2)
```

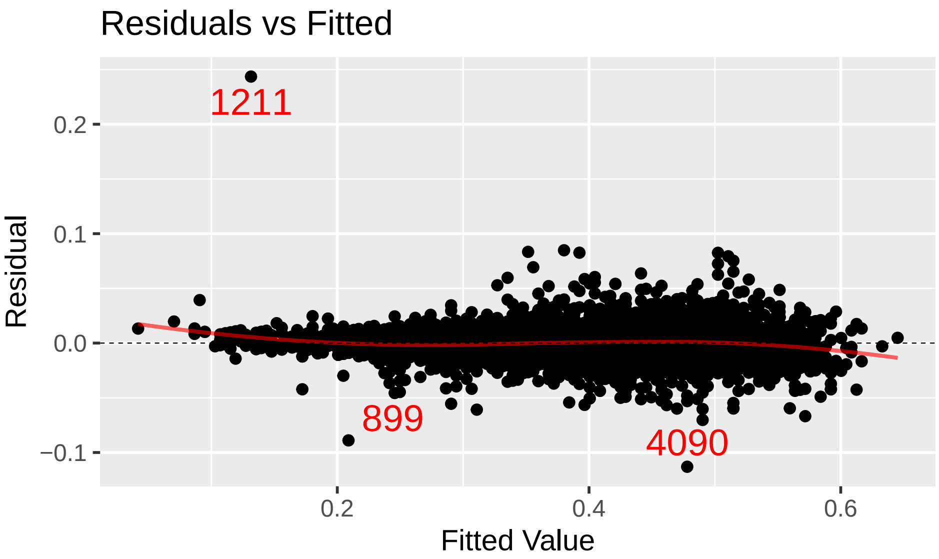
```
## Analysis of Variance Table
##
## Model 1: Diameter ~ Length
## Model 2: Diameter ~ Length * Sex
##   Res.Df     RSS Df Sum of Sq       F     Pr(>F)
## 1   4175 1.07766
## 2   4171 1.05195  4 0.0257061 25.4812 < 2.22e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This test tells us that we should try to keep the interaction term with Sex or the Sex main effect (or both), even though the interactions don't appear to be significant, and the SLR R^2 is not majorly different from what we have for models 2 or 6.

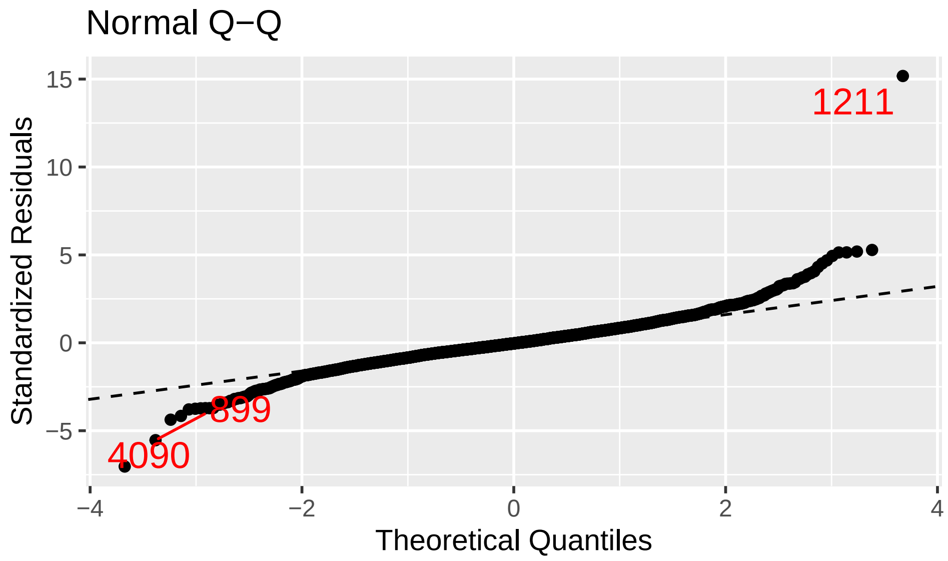
So, which model do we pick? Let's check conditions.

```
mplot(fm1, which = 1) #actually shows some increasing variability
```

```
## `geom_smooth()` using formula 'y ~ x'
```

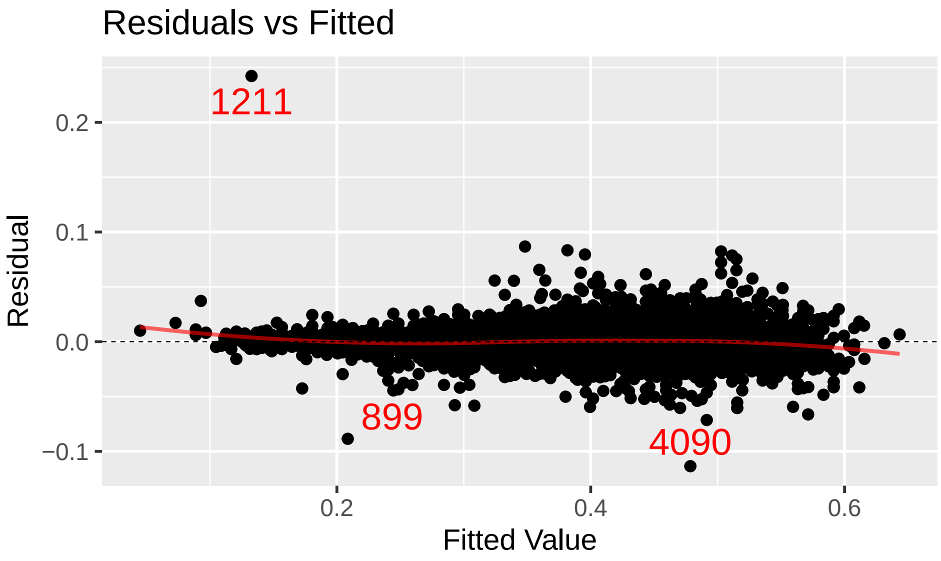


```
mplot(fm1, which = 2) #potential issues in tails, one clear outlying point
```

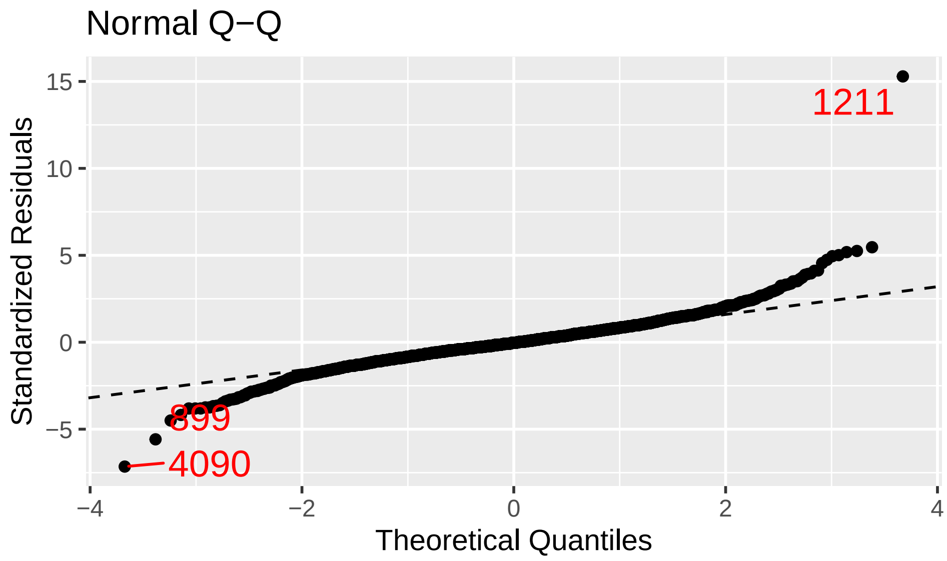


```
mplot(fm2, which = 1) #again, increasing variability
```

```
## `geom_smooth()` using formula 'y ~ x'
```

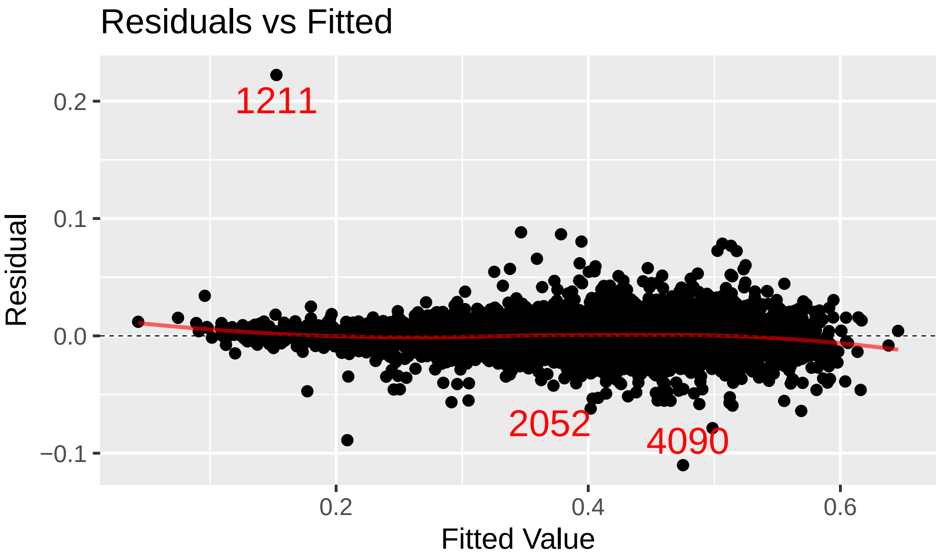


```
mplot(fm2, which = 2) #similar to SLR issues
```

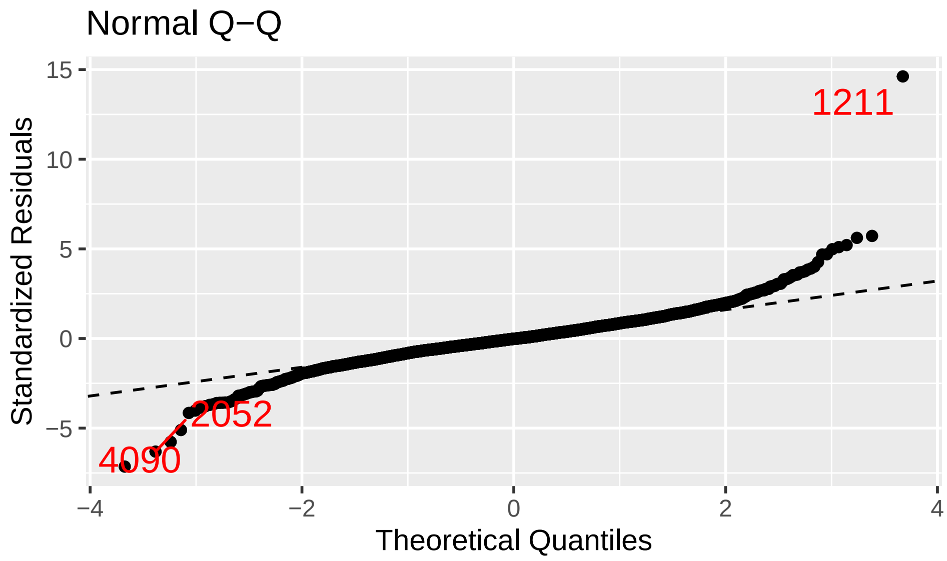


```
mplot(fm6, which = 1) #again, increasing variability
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
mplot(fm6, which = 2) #similar to issues above
```



What happens if we sqrt the response to try to deal with the variability issue? That turns out to be too much of a transformation (overdoes it). What if we take out observation 1211? That improves some features, but not others.

Honestly here, I would probably just use the SLR model. Quick transformations don't seem to be of much use - we could explore more, and it doesn't seem to really improve the model a ton if we use the larger models, so I'll go with the simple one.

> For your fitted model, interpret the coefficients of at least 2 predictors (including at least one interaction term if you have one).

SOLUTION:

My model choice only had one predictor, so use the example with fm2 up above to practice this.

> Check your model conditions. Report on any violations.

SOLUTION:

```
#done above
```

> Check for multicollinearity issues.

SOLUTION:

```
#not valid for SLR
```

> Obtain a predicted value of diameter for a sea snail that has the following characteristics, as well as a 99% confidence interval for the mean diameter at this set of characteristics:

```
newdata <- data.frame(Sex = "M", Length = 0.56, Height = 0.175, Wholewt = 1.01, Shuckedwt = 0.384, Visc
```

SOLUTION:

```
predict(fm1, newdata, int = "confidence", level = 0.99)
```

```
##          fit        lwr        upr
## 1 0.43724428 0.43657549 0.43791308
```

The fitted value (predicted value) is 0.4372. The 99% CI for the mean diameter for snails like this one (same characeristics) is (0.4366, 0.4379).