Project 1: Regression

Courtney Schoen

Question

Do the elements of an opossum's body, their age, and location determine the overall length of their body?

Loading Data Sets

```
possum <- read_csv("possum (1).csv")
```

```
## Rows: 104 Columns: 14
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): Pop, sex
## dbl (12): case, site, age, hdlngth, skullw, totlngth, taill, footlgth, earco...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Clean Data

```
#giving variables names that are easier to understand
possum_rename <- dplyr::rename(possum,
                 head_length = hdlngth,
                 skull_width =  skullw,
                 total_length = totlngth,
                 tail_length = taill,
                 ear_conch_length = earconch,
                 foot_length = footlgth,
                 chest_girth = chest,
                 belly_girth = belly)

#putting all measurements in cm
possum_in_cm <- possum_rename %>%
    mutate(head_length = head_length / 10,
           skull_width = skull_width / 10,
           foot_length = foot_length / 10,
           ear_conch_length = ear_conch_length / 10,
           eye = eye /10)

#selecting only the data I need
opossum <- possum_in_cm %>%
    select(site, sex, age, head_length, skull_width,
           total_length, tail_length, foot_length,
           ear_conch_length, eye, chest_girth, belly_girth) %>%
  mutate(sex = case_when(
    sex == "f" ~ 0,
    sex == "m" ~ 1
  ))

#checking missing values
sum(is.na(opossum))
```

```
## [1] 3
```

```
#removing rows with missing data
my_opossum <- na.omit(opossum)
#checking if it worked
sum(is.na(my_opossum))
```

```
## [1] 0
```

This is representative of the data in my_opossum after it has been cleaned. However I feel it is most useful in this form as this is how the variables looked as I was working through regression.

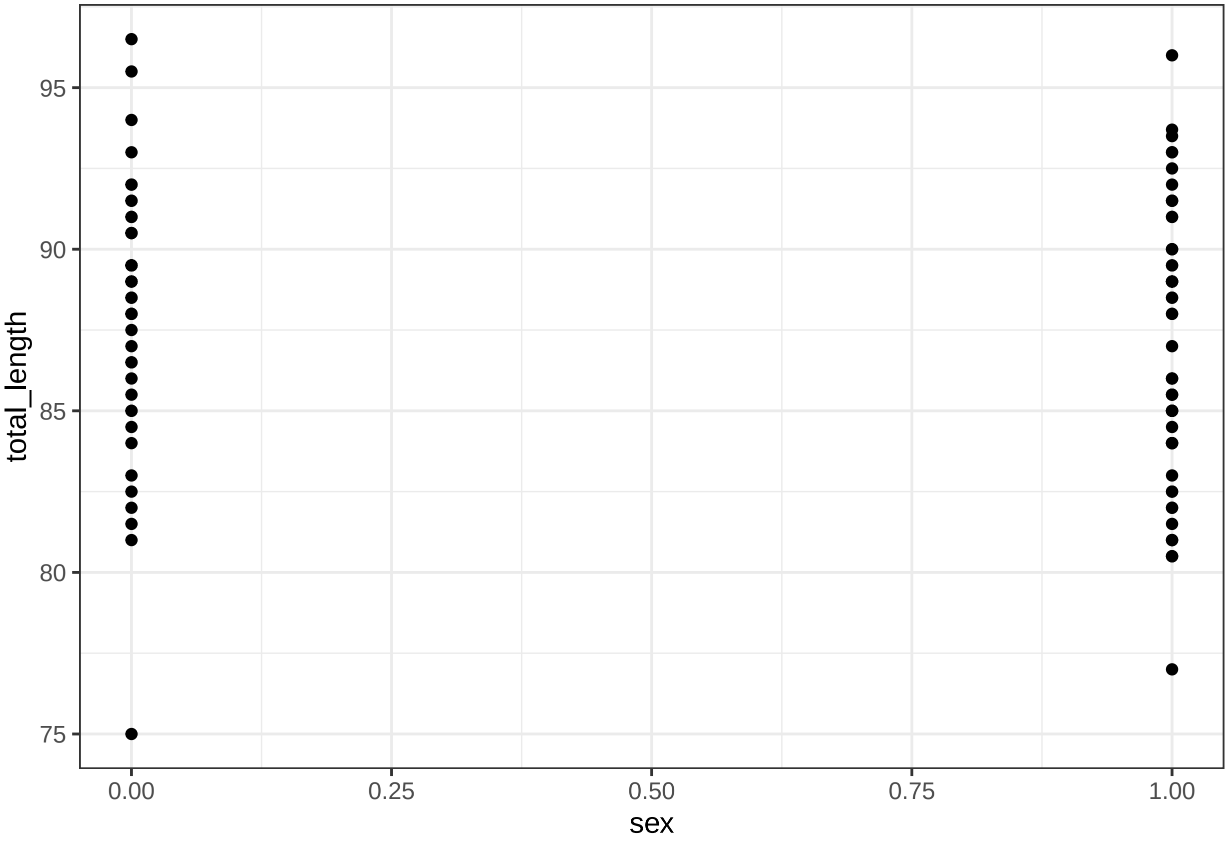
Variable	Description
site	location (1-7)
sex	gender indicator (0 = female, 1 = male)
age	opossum age (1 - 8)
head_length	length of head (in cm)
skull_width	width of skull (in cm)
total_length	total_length of opossum (in cm)
tail_length	length of tail (in cm)
foot_length	length of foot (in cm)
ear_conch_length	Length of interior of ear (in cm)
eye	distance between eyes (in cm)
chest_girth	distance around chest (in cm)
belly_girth	distance around belly (in cm)

Exploratory Analysis

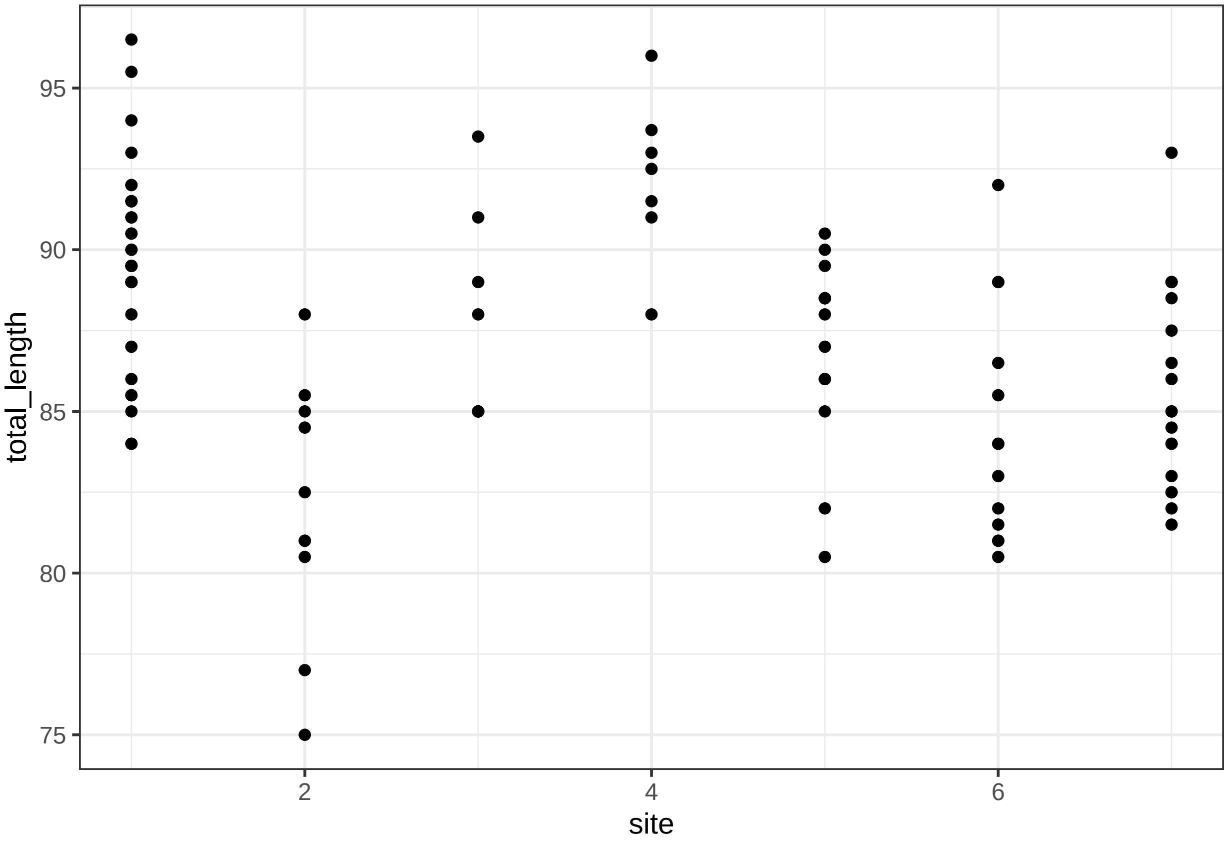
```
summary(my_opossum)
```

```
##       site            sex              age          head_length    
##  Min.   :1.000   Min.   :0.0000   Min.   :1.000   Min.   : 8.250  
##  1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.: 9.070  
##  Median :4.000   Median :1.0000   Median :3.000   Median : 9.290  
##  Mean   :3.673   Mean   :0.5842   Mean   :3.822   Mean   : 9.273  
##  3rd Qu.:6.000   3rd Qu.:1.0000   3rd Qu.:5.000   3rd Qu.: 9.480  
##  Max.   :7.000   Max.   :1.0000   Max.   :9.000   Max.   :10.310  
##   skull_width     total_length    tail_length     foot_length  
##  Min.   :5.000   Min.   :75.00   Min.   :32.00   Min.   :6.03  
##  1st Qu.:5.500   1st Qu.:84.50   1st Qu.:36.00   1st Qu.:6.45  
##  Median :5.640   Median :88.00   Median :37.00   Median :6.79  
##  Mean   :5.696   Mean   :87.27   Mean   :37.05   Mean   :6.84  
##  3rd Qu.:5.810   3rd Qu.:90.00   3rd Qu.:38.00   3rd Qu.:7.25  
##  Max.   :6.860   Max.   :96.50   Max.   :43.00   Max.   :7.79  
##  ear_conch_length      eye         chest_girth     belly_girth   
##  Min.   :4.130    Min.   :1.280   Min.   :22.00   Min.   :25.00  
##  1st Qu.:4.480   1st Qu.:1.440   1st Qu.:25.50   1st Qu.:31.00
##  Median :4.680   Median :1.490   Median :27.00   Median :32.50
##  Mean   :4.813   Mean   :1.505   Mean   :27.06   Mean   :32.64
##  3rd Qu.:5.200   3rd Qu.:1.570   3rd Qu.:28.00   3rd Qu.:34.00
##  Max.   :5.620   Max.   :1.780   Max.   :32.00   Max.   :40.00
```

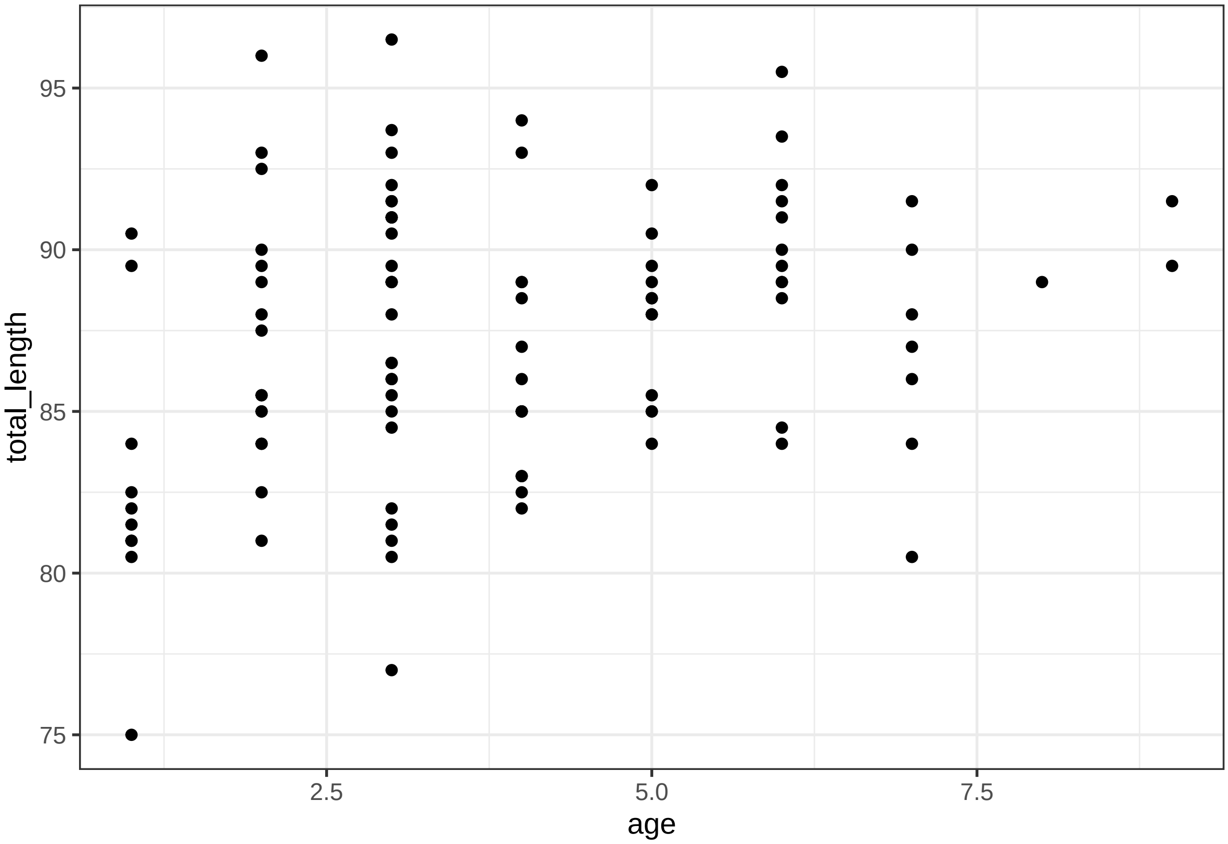
```
#looking at site and sex as they are categorical
ggplot(my_opossum) +
    geom_point(aes(sex, total_length))
```



```
ggplot(my_opossum) +
    geom_point(aes(site, total_length))  #boxplot
```



```
ggplot(my_opossum) +
    geom_point(aes(age, total_length))
```



```
#there does appear to be different distributions of age at different sites so this may still be an okay

cor(my_opossum$age, my_opossum$tail_length)
```

```
## [1] 0.1202054
```

```
cor(my_opossum$head_length, my_opossum$tail_length)
```

```
## [1] 0.275155
```

```
cor(my_opossum$chest_girth, my_opossum$belly_girth)
```

```
## [1] 0.6097571
```

```
cor(my_opossum$skull_width, my_opossum$eye)
```

```
## [1] 0.3143186
```

```
cor(my_opossum$tail_length, my_opossum$foot_length)
```

```
## [1] -0.1145598
```

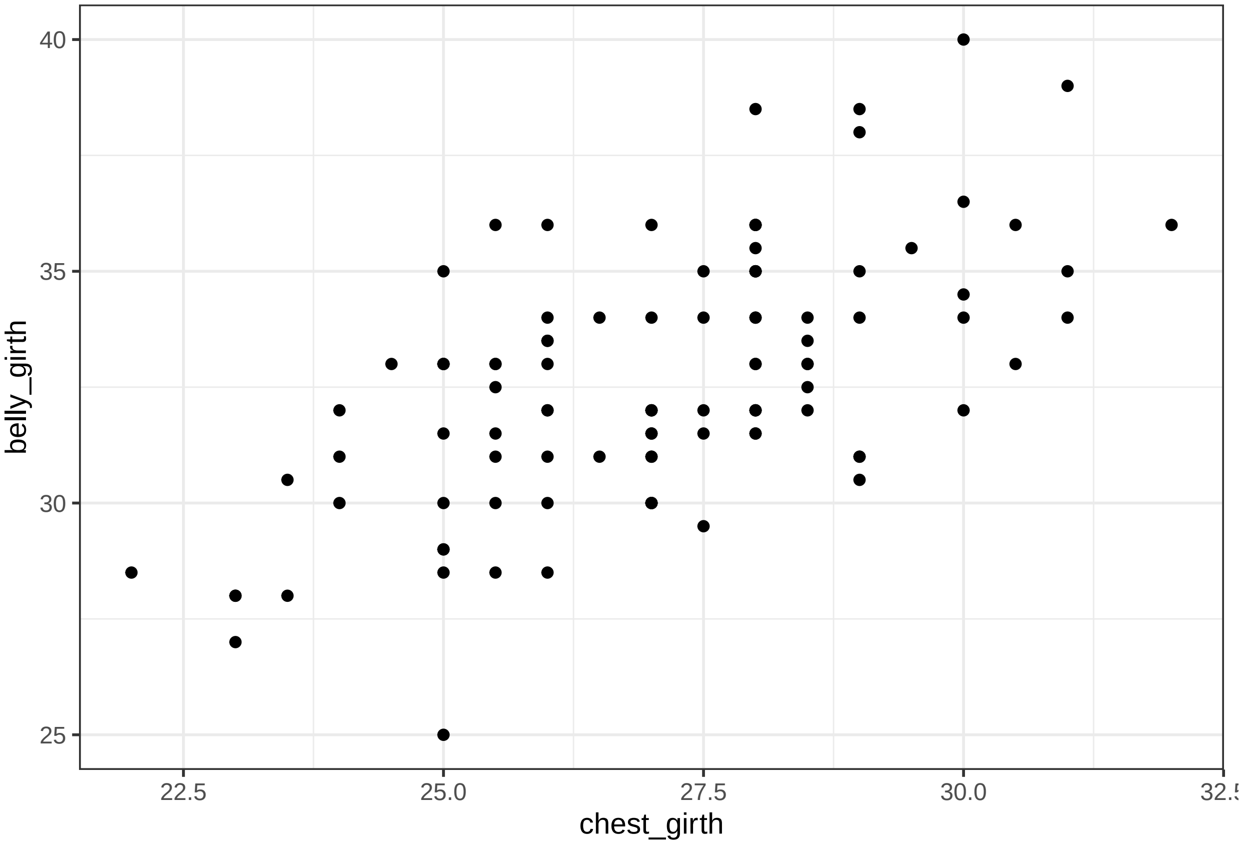
```
cor(my_opossum$head_length, my_opossum$foot_length)
```

```
## [1] 0.4159449
```

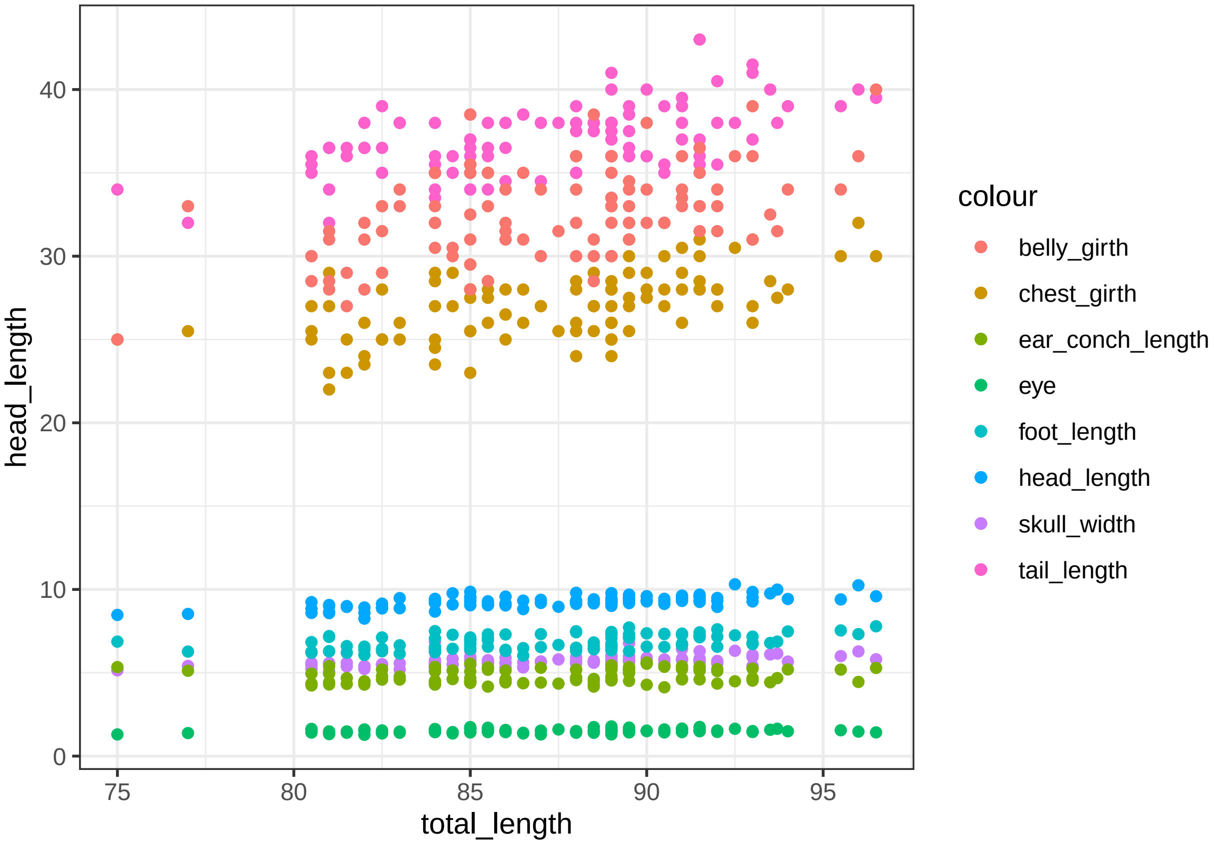
```
cor(my_opossum$skull_width, my_opossum$ear_conch_length)
```

```
## [1] 0.02529349
```

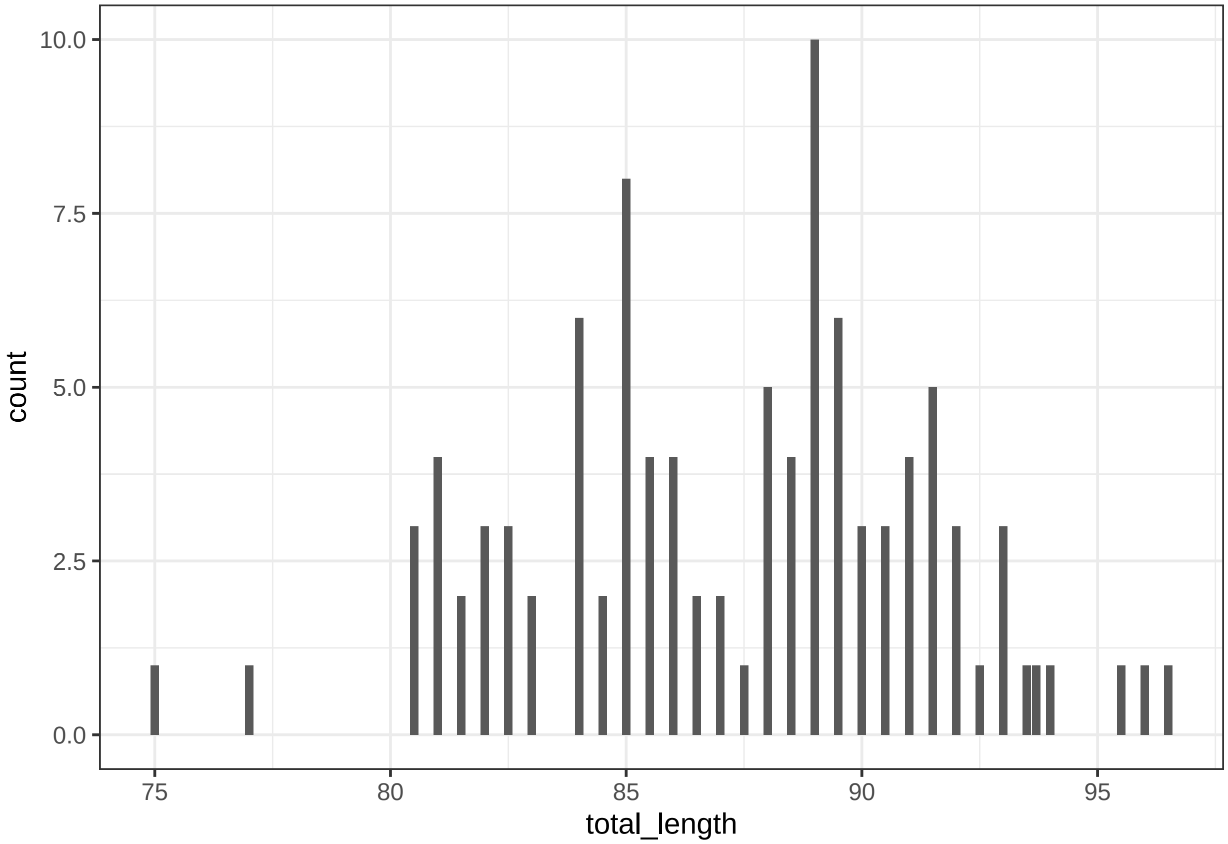
```
ggplot(my_opossum)+
    geom_point(aes(chest_girth, belly_girth))
```



```
ggplot(my_opossum, aes(x = total_length)) +
    geom_point(aes(y = head_length, colour = "head_length")) +
    geom_point(aes(y = skull_width, colour = "skull_width")) +
    geom_point(aes(y = tail_length, colour = "tail_length")) +
    geom_point(aes(y = foot_length, colour = "foot_length")) +
    geom_point(aes(y = ear_conch_length, colour = "ear_conch_length")) +
    geom_point(aes(y = eye, colour = "eye")) +
    geom_point(aes(y = chest_girth, colour = "chest_girth")) +
    geom_point(aes(y = belly_girth, colour = "belly_girth"))
```



```
ggplot(my_opossum) +
    geom_bar(aes(total_length))
```



Comment:

In the summary I am able to see an overall idea of the data. I can see the minimum and maximum value, as well as the type of variables I am dealing with. We see that there are three variables appearing to be categorical. Site is represented by numbers, but they are only whole numbers representing locations. Sex is represented by characters and only has two options. Age is numerical but is not continuous. So it is not necessarily categorical, but also not really qualitative either.

Next I wanted to take a look at the categorical variables and see if they appear to have any predictive power over total_length. When looking at total_length and sex, we that females can tend to be slightly larger so it could be a good predictor. When looking at total_length and site, there does appear to be different distributions of total_length at different site. So, site could be a good predictor. In terms of age, total_length does seem to increase overall as age increases, so it may be a good predictor.

Next I wanted to see if there is any correlation between predictors as I may not need them all if there is. We do see some higher correlations between total_length and head_length, total_lone higher correlation of 0.6 between chest_girth and belly_girth. This correlation does not appear to be too high to throw one of those predictors out but we will investigate it more.

To further explore the predictors with high correlation, I graphed them against one another. They do still appear pretty linearly related but are not a direct response of one another. So, they could be beneficial.

Next I wanted to see how all of the qualitative variables behave in relation to total_length. To do this, I graphed all of the predictors vs the response. I can see that for every response, as total_length increases, response also increases.

Last I wanted to look at the distribution of the response. We see that total_length has an overall normal shape but is slightly skewed to the right and so this will need to be taken into account.

Creating Train and Test

```
#splitting data into testing and training
split <- initial_split(my_opossum, prop = .8)

#creating data sets of just train data and just test data
opposum_train <- training(split)
opposum_test <- testing(split)

#splitting train data again to help fit
folds <- vfold_cv(opposum_train, v = 10)
```

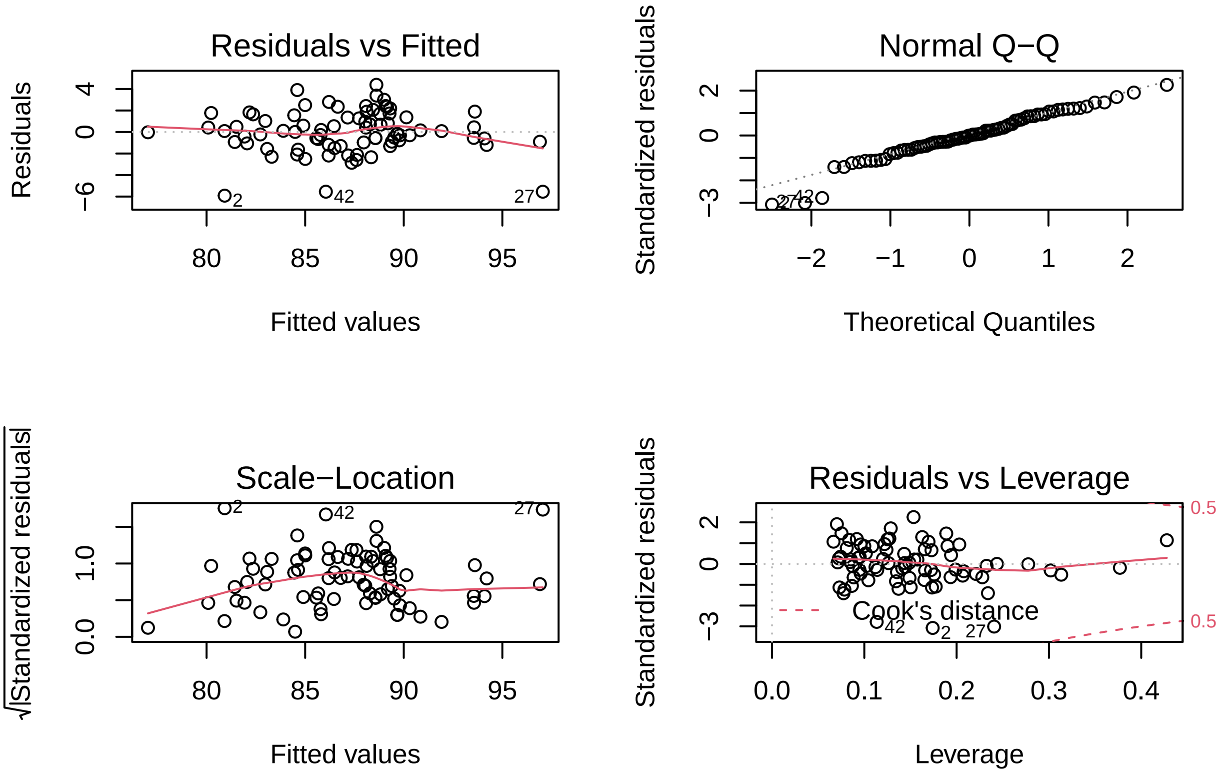
Comment: Here we split the data into two different sets train and test. In test is 80% of the data and that will be used for fitting and working with the data. In train will be 20% of the data in which we will be able to check our model using separate data in order to get the most accurate results. I chose 80% and 20% due to the fact that I have a smaller data set. After splitting the data initially, we then split train again into 10 sections. This aids the model in best fitting the data.

Multiple Linear Regression

Analyzing Full Fit of Model

```
#fitting full model
opossum_fit <- lm(total_length ~ ., data = opposum_train)

#checking assumptions of linear model
par(mfrow = c(2,2))
plot(opossum_fit)
```



```r
#predicting total_length using full model and test data
full_preds <- predict(opossum_fit, data.frame(opposum_test))
full_preds
```

```
##        1        2        3        4        5        6        7        8 
## 88.49050 89.28239 94.03696 92.05937 95.48363 94.01936 87.39203 84.57318 
##        9       10       11       12       13       14       15       16 
## 86.12687 89.33712 94.00550 91.45221 90.36205 86.76537 81.43769 85.27419 
##       17       18       19       20       21 
## 83.67624 84.03727 86.67117 84.68484 87.69013
```

```r
#rmse of full model
(opposum_fit_rmse <- sqrt(mean((opposum_test$total_length - full_preds)^2)))
```

```
## [1] 2.18608
```

```r
opposum_fit_rmse
```

```
## [1] 2.18608
```

```r
#adj rsq of full model
summary(opossum_fit)
```

```
## 
## Call:
## lm(formula = total_length ~ ., data = opposum_train)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -5.9167 -0.9914 -0.0092  1.4134  4.3879 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      -5.12345   10.03724  -0.510  0.61139    
## site             -0.70846    0.22098  -3.206  0.00205 ** 
## sex              -0.59400    0.53441  -1.112  0.27027    
## age               0.02325    0.14891   0.156  0.87637    
## head_length       4.79008    1.14387   4.188 8.30e-05 ***
## skull_width      -0.43761    1.14409  -0.382  0.70329    
## tail_length       1.28951    0.15243   8.460 3.21e-12 ***
## foot_length       1.60229    1.20968   1.325  0.18975    
## ear_conch_length -1.22524    1.21935  -1.005  0.31854    
## eye              -1.38929    2.73866  -0.507  0.61360    
## chest_girth       0.12931    0.22236   0.582  0.56280    
## belly_girth      -0.02477    0.12562  -0.197  0.84424    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.116 on 68 degrees of freedom
## Multiple R-squared:  0.7898, Adjusted R-squared:  0.7558 
## F-statistic: 23.23 on 11 and 68 DF,  p-value: < 2.2e-16
```

Comments: Before doing any type of regression, I decided to create a linear model using all possible predictors. With this model I checked assumptions, made predictions, and found rmse. By doing this, I created a sort of baseline by which to base the rest of my analysis and to see if selecting only the best predictors actually improves the models ability to predict total_length. In terms of assumptions everything looks pretty good. We see the residuals are rather linear and variation is pretty constant. We also see the data is rather normal. While this is not perfect it is close enough for me to feel comfortable moving forward with linear regression without altering my response.

Here we have an rmse of 2.285 which seems rather low considering we have not yet selected the best predictors. We also see that the adj Rsq is 73% and so 73% of the variance is explained by the relationship between the predictors and response.

Analyzing Stepwise Regression Fit of Model

```
#did selection on train data model
ols_step_forward_aic(opossum_fit)
```

```
##
##                           Selection Summary
## ---------------------------------------------------------------------
## Variable         AIC       Sum Sq       RSS       R-Sq      Adj. R-Sq
## ---------------------------------------------------------------------
## head_length    419.580     624.572    823.775    0.43123      0.42394
## tail_length    390.975     886.440    561.906    0.61204      0.60196
## site           350.049    1119.773    328.574    0.77314      0.76418
## sex            349.846    1128.697    319.650    0.77930      0.76753
## ---------------------------------------------------------------------
```

```
ols_step_backward_aic(opossum_fit)
```

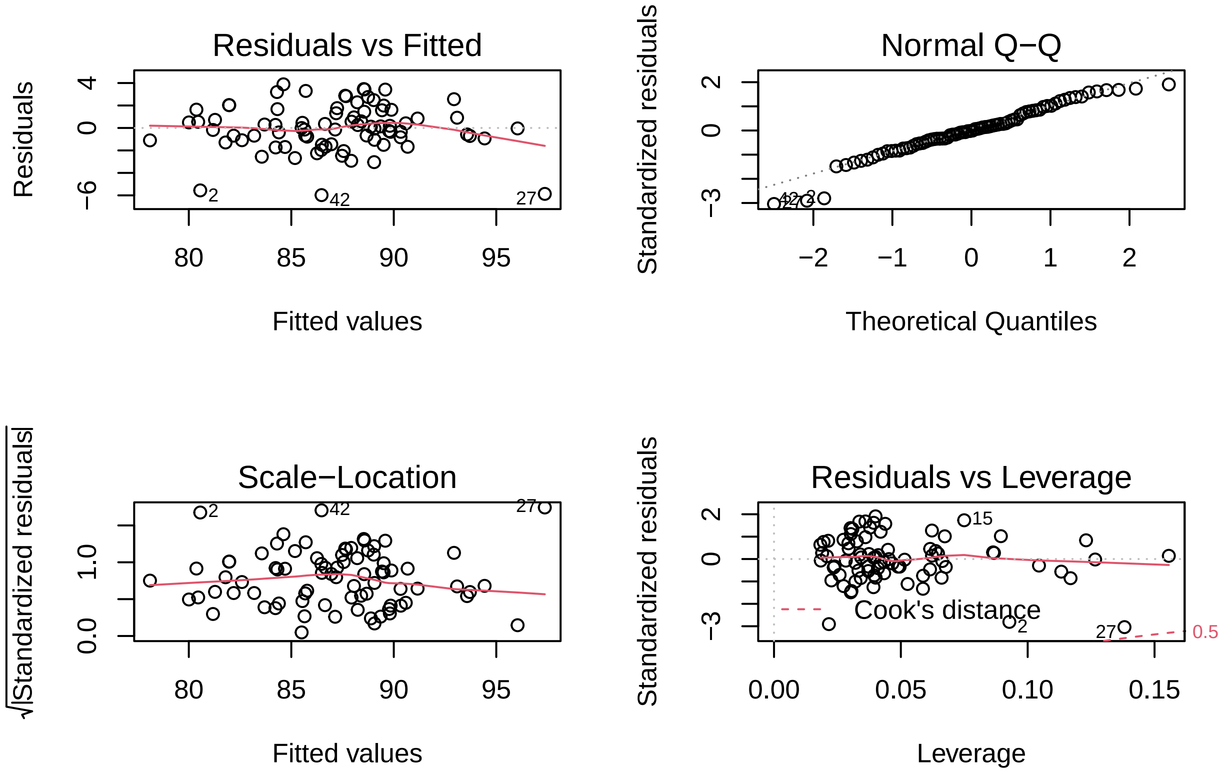
```
##
##
##                         Backward Elimination Summary
## --------------------------------------------------------------------------
## Variable              AIC        RSS       Sum Sq      R-Sq      Adj. R-Sq
## --------------------------------------------------------------------------
## Full Model          359.932    304.388    1143.959    0.78984      0.75584
## age                 357.961    304.497    1143.850    0.78976      0.75929
## belly_girth         356.001    304.648    1143.699    0.78966      0.76261
## skull_width         354.163    305.268    1143.079    0.78923      0.76548
## chest_girth         352.462    306.408    1141.938    0.78844      0.76787
## eye                 350.969    308.359    1139.988    0.78710      0.76960
## ear_conch_length    350.055    312.573    1135.774    0.78419      0.76960
## foot_length         349.846    319.650    1128.697    0.77930      0.76753
## --------------------------------------------------------------------------
```

```
ols_step_both_aic(opossum_fit)
```

```
##
##
##                                  Stepwise Summary
## ---------------------------------------------------------------------------------
## Variable        Method        AIC        RSS        Sum Sq      R-Sq      Adj. R-Sq
## ---------------------------------------------------------------------------------
## head_length    addition    419.580    823.775     624.572    0.43123      0.42394
## tail_length    addition    390.975    561.906     886.440    0.61204      0.60196
## site           addition    350.049    328.574    1119.773    0.77314      0.76418
## sex            addition    349.846    319.650    1128.697    0.77930      0.76753
## ---------------------------------------------------------------------------------
```

```
#new model based on predictors selected, still using train data
selected_fit <- lm(total_length ~
                       head_length + tail_length + site,
                   data = opposum_train)

#checking assumptions of linear model
par(mfrow = c(2,2))
plot(selected_fit)
```



```
#predicting total_length using my chosen model and the test data
selected_preds <- predict(selected_fit, data.frame(opposum_test))
selected_preds
```

```
##        1        2        3        4        5        6        7        8
## 88.85872 89.45934 92.58830 91.96241 94.57797 94.77817 86.83139 84.41604
##        9       10       11       12       13       14       15       16
## 86.23889 89.20054 93.58842 91.98677 90.66065 87.24811 81.52936 85.13356
##       17       18       19       20       21
## 83.47710 84.10299 86.90587 85.00392 86.84343
```

```
#finding rmse of selected model
(selected_fit_rmse <- sqrt(mean((opposum_test$total_length - selected_preds)^2)))
```

```
## [1] 2.083357
```

```
selected_fit_rmse
```

```
## [1] 2.083357
```

```
#finding adj rsq of selected model
summary(selected_fit)
```

```
## 
## Call:
## lm(formula = total_length ~ head_length + tail_length + site, 
##     data = opposum_train)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -5.9768 -1.0958 -0.0231  1.4747  3.8841 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -6.9568     6.7705  -1.028    0.307    
## head_length   5.0052     0.7385   6.778 2.29e-09 ***
## tail_length   1.3767     0.1336  10.301 4.48e-16 ***
## site         -0.8428     0.1147  -7.346 1.94e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.079 on 76 degrees of freedom
## Multiple R-squared:  0.7731, Adjusted R-squared:  0.7642 
## F-statistic: 86.34 on 3 and 76 DF,  p-value: < 2.2e-16
```

Comments: For my multiple linear regression model I used stepwise selection in order to determine what the best predictors of total_length are. All three types of stepwise selection agreed on the predictors chosen so I felt comfortable moving forward. The predictors that were chosen were head_length, tail_length, and site. I then created a model using these variables to predict total_length. In this model I used the testing data. By doing this, I allow my model to be fit and tested on two different sets of data. This creates a better model and more accurate results We see that linearity, constant variance, and normality are well met.

Next I predicted total_length based upon this model using the test data. I see values that make sense in terms of total_length. They are all reasonable and are possible when considering the real world context of the data. As a result of the prediction we see that rmse is 1.98 and adj Rsq is 73%. So we see that 73% of the variance is explained by the relationship between the predictors and response.

Comparing Full Model and Selected Model

Comments: When looking at the full model vs. the model selected using stepwise regression, we do see that the selected model appears to be doing a little bit of a better job. Rmse went from 2.285 to 1.98. Considering we want a low rmse this is a very good sign and shows us that the stepwise regression did a rather good job of selecting predictors for total_length. In terms of adj Rsq we see no change. This is not necessarily good or bad. Overall, the same amount of variance between the predictors and the response is being explained. This leads me to believe that the predictors not used in the selected model are having very little effect on predicting total_length. Regardless, we can still conclude that the multiple linear regression model did a good job at creating a model to predict total_length.

Penalized Regression

Exploring LASSO Regression

```r
#loading and formatting testing data
opossum_recipe <- recipe(total_length ~ ., data = opposum_test) %>%
    step_center(all_predictors()) %>%
    step_scale(all_predictors())

#creating a LASSO regression model with the ability to tune penalty
```

```
opossum_model <- linear_reg(mixture = 1, penalty = tune()) %>%
    set_engine("glmnet")

#putting the recipe (data) and model (method of fitting data) into the workflow
opossum_work <- workflow() %>%
    add_recipe(opossum_recipe) %>%
    add_model(opossum_model)
```

Comment: For penalized regression, I started by creating a recipe using the test data and all possible predictors of total_length. Within this, I also centered and scaled my predictors to make them as normal as possible. As a whole, this is loading in the data that I want to use to create my regression model.

Next I created I a model for linear regression. Here I am telling the model that I want to do a LASSO regression. I also set my penalty as tune. This allows me to later look at a variety of penalty values and select one most beneficial to my data and model.

I then created a workflow by putting together my recipe and model. Overall what this is doing is combining my data with the method by which I would like to fit the data.

Tuning Parameters

```
#creating grid of possible penalty values to test
grid <- grid_regular(penalty(), levels = 50)

#combining the model, recipe, possible penalties to put into the folds and test
tuned_grid <- tune_grid(opossum_work,
          resamples = folds,
          grid = grid)

#determining the best penalty for the model according to rmse
best_fit <- tuned_grid %>%
  select_best(metric = "rmse")
best_fit
```

```
## # A tibble: 1 x 2
##   penalty .config
##     <dbl> <chr>
## 1  0.0954 Preprocessor1_Model45
```

```
#inserting the best penalty into the workflow (model and recipe)
opossum_work_final <- opossum_work %>%
  finalize_workflow(best_fit)

#training model based on optimal metrics (penalty)
final_fit <- opossum_work_final %>%
  last_fit(split = split) %>%
  collect_metrics()
final_fit
```

```
## # A tibble: 2 x 4
##   .metric .estimator .estimate .config
##   <chr>   <chr>          <dbl> <chr>
## 1 rmse    standard       2.06  Preprocessor1_Model1
## 2 rsq     standard       0.734 Preprocessor1_Model1
```

Comment: Now that I have my data and method of fitting together, I have to tune my parameter penalty.

To do this I first created a grid of 50 possible penalty values of which I would test in order to find the best one. I then combined these possible penalty values with my workflow. I also put in my 10 folds which split the train data into even smaller categories. After this I selected the best penalty according to rmse. Basically, the penalty that minimized rmse was chosen and this penalty is 0.095. At this section I could have selected my penalty based upon rmse or rsq. However, in the earlier analysis we saw that rmse was a better representation of our different models as rsq was unchanged.

Now that the best penalty has been determined, I put it into the original workflow in which penalty was initially set as tune. At this point I have effectively tuned my parameters and created my final workflow for the model. Now that I have this model, I can finally train my data and as a result get the rmse and rsq. For this model we see that rmse is 2.216 and rsqu is 81.7%.

Predicting

```
#gives predictions of final model with best penalty using test data
opossum_work_final %>%
    last_fit(split) %>%
    collect_predictions()
```

```
## # A tibble: 21 x 5
##    id               .pred  .row total_length .config
##    <chr>            <dbl> <int>        <dbl> <chr>
##  1 train/test split  88.7     1         89   Preprocessor1_Model1
##  2 train/test split  89.1     7         89.5 Preprocessor1_Model1
##  3 train/test split  93.3    11         89.5 Preprocessor1_Model1
##  4 train/test split  91.8    20         89   Preprocessor1_Model1
##  5 train/test split  95.0    21         96.5 Preprocessor1_Model1
##  6 train/test split  93.9    22         91   Preprocessor1_Model1
##  7 train/test split  87.0    29         88   Preprocessor1_Model1
##  8 train/test split  84.4    30         84   Preprocessor1_Model1
##  9 train/test split  86.2    36         88   Preprocessor1_Model1
## 10 train/test split  89.0    46         85   Preprocessor1_Model1
## # ... with 11 more rows
```

```
#provides the estimates of the predictors when the optimal penalty is applied
estimate_fit <- opossum_work_final %>%
  last_fit(split = split)
estimate_fit$.workflow[[1]] %>%
  extract_fit_parsnip() %>%
  tidy()
```

```
## # A tibble: 12 x 3
##    term             estimate penalty
##    <chr>               <dbl>   <dbl>
##  1 (Intercept)        87.1    0.0954
##  2 site               -1.26   0.0954
##  3 sex                -0.272  0.0954
##  4 age                 0      0.0954
##  5 head_length         1.50   0.0954
##  6 skull_width         0      0.0954
##  7 tail_length         2.55   0.0954
##  8 foot_length         0.488  0.0954
##  9 ear_conch_length    0      0.0954
## 10 eye                 0      0.0954
## 11 chest_girth         0.251  0.0954
## 12 belly_girth         0      0.0954
```

Comments: Next I wanted to look at my predicted values against the actual value just to visualize how well the model did. While the predictions are not exact we can see that they are doing a pretty good job as the difference between the actual and predicted (residual) appears rather small.

I then just wanted to look at the estimates of of the predictors for the selected penalty. This tells me which variables are not being used and which have high predictive power. We see that several of the estimates go to zero: age, ear_conch_length, eye, and belly girth. We also see several with rather high estimates: tail_length, head_length, and site. The rest of the variables lie in a rather middle ground. This leads me to believe that in this model, total_length is mainly being predicted by tail_length, head_length, and site with some input from those not at zero.

Comparing Methods

In the multiple linear regression model we saw that total_length was being predicted by head_length, tail_length, and site. This model had an rmse of 1.98 and an adj rsq of 73%. In the penalized regression model we saw that total_length was mainly being predicted by head_length, tail_length, and site with some input from sex, skull_width, foot_length, and chest_girth. This model had an rmse of 2.21 and an rsq of 81.75%.

Here we see that the multiple linear regression has a lower rmse than that in the penalized regression. On the other hand the penalized regression has an higher rsq than the multiple linear regression. This is interesting considering we selected the penalty based upon rmse.

Overall I feel as if the rmse is slightly lower for the mlr due to the fact that it only has the three strongest predictors. In the penalized regression, we see the same three main predictors but there are also other variables still being used, even if it is at a smaller degree. I really cannot explain why the rsq is higher for penalized aside from potentially being that it is regular rsq and not adjusted so the addition of predictors is not being taken into account.

Regardless, I feel as if mlr created the best model given the data as the rmse is the lowest and it has the least amount of predictors, making it the simplest model.

Revisiting Question

My question was whether an opossum's environment and physical attributes determine the overall length of their body. I feel that my question was nicely answered. When using these types of variables as predictors, we see two models with high Rsq and low rmse. When analyzing the predictions, we also see that predicted value is very close to actual value. While not every physical attribute determines the total_length, there are two that do rather accurately. We also see that the site or location at which they live determines overall length. For all of these reasons I feel as if the answer to the question is that environment and physical features do play a role in determining the total length of the body.

Shortcomings

I feel that the greatest shortcoming was with the categorical variables. In order to make them work in the model, I had to make them numeric (ex: 0 for male and 1 for female). This makes them seem as if one is better than the other and very likely messes up the regression. Considering site is one of my selected predictors I feel as if this is a problem that would be found in my analysis. If I were to do it again I would be sure to deal with these categorical variables in a better way so that they were in fact represented as categories and not rated. Another issue was the size of my dataset. It was on the smaller side and as a result, my testing data was only 11 observations. I feel that this does not give the best idea of how the model is actually fitting purely based upon size. Having more observations would likely result in more accurate models.

Take Aways

Overall I did learn a lot from this project. I was so confused that I had to individual analyze each step of the LASSO regression and tuning the penalty. However as a result of all of this struggle I feel I do have a rather good understanding of the process and could do it again for other data. This also helped me understand how useful putting comments explain steps of code can be when you begin to work with more complex problems. By having these little comments I was able to begin to form a more cohesive idea of the overall process.

I also learned a lot about how testing and training data works. Now that I understand it makes a lot of sense that you do not want to check a models ability to predict on the same data by which you formed it (kind of like a conflict of interest).

Data science as a whole is a lot more statistically based than I ever imagine. This is perhaps a late realization as the prerequisites of the course are in fact statistics. However I never imagined quite how strong it would be. I really enjoyed the data manipulation part of data science 1 and adding in this level of statistics is an adjustment.

My biggest question left after all of this is how this would be used in the workplace. Opossums are cool and all but I do not see myself working somewhere that I need to predict the total_length of one. I can see the type of data by which this is useful, but in what type of business would you need something like this as a practicing data scientist?

**When I knit to a pdf the penalty is different than when I run it here. This causes an extra value to go to 0. I left the comments representative to what I am seeing when I run it in R and not in the pdf.