Homework 5

Multiple Linear Regression Variable Selection Methods

AUTHOR

Zeb Sorenson

Data and Description

For this assignment, we are revisiting the data set used in Homework 4. I think it would be very beneficial for you to review your Homework 4 before starting this one.

Measuring body fat is not simple. One method requires submerging the body underwater in a tank and measuring the increase in water level. A simpler method for estimating body fat would be preferred. In order to develop such a method, researchers recorded age (years), weight (pounds), height (inches), and three body circumference measurements (around the neck, chest, and abdominal (all in centimeters)) for 252 men. Each mans' percentage of body fat was accurately estimated by an underwater weighing technique (the variable brozek is the percentage of body fat). The hope is to be able to use this data to create a model that will accurately predict body fat percentage, by using just the basic variables recorded, without having to use the tank submerging method.

The data can be found in the BodyFat data set on Canvas. Download BodyFat.txt, and put it in the same folder as this R Markdown file.

0. Replace the text "< PUT YOUR NAME HERE >" (above next to "author:") with your full name.

0b. Make sure to set your seed since some of the functions randomly split your data (use `set.seed` in the setup code chunk above)!

1. Read in the data set, and call the data frame "bodyfat_orig". Print a summary of the data and make sure the data makes sense. **Remove the "row" column (which contains row numbers) from the data set.** Make sure the class of "bodyfat_orig" is a *data.frame only*.

```
orginalData <- read.csv("~/Desktop/Stat 330/BodyFat.txt", sep="")

bodyfat_orig <- subset(orginalData, select = -row)
```

```r
bodyfat_orig <- as.data.frame(bodyfat_orig)

head(bodyfat_orig)
```

```
  brozek age weight height neck chest abdom
1   12.6  23 154.25  67.75 36.2  93.1  85.2
2    6.9  22 173.25  72.25 38.5  93.6  83.0
3   24.6  22 154.00  66.25 34.0  95.8  87.9
4   10.9  26 184.75  72.25 37.4 101.8  86.4
5   27.8  24 184.25  71.25 34.4  97.3 100.0
6   20.6  24 210.25  74.75 39.0 104.5  94.4
```

2. Refer back to your Homework 4. In that assignment, you fit this multiple linear regression model: for each of the multiple linear regression assumptions listed below, state if they were met or not met.

1. The X's vs Y are linear: Met!
2. The residuals are normally distributed: Met!
3. The residuals are homoscedastic: Met!
4. There are no influential points: Not Met!
5. No multicollinearity: Not met! We have multicollinearity.

3. There is one clear influential point in the data set. Create a new variable called "bodyfat" that contains the bodyfat_orig data set with the influential point removed. Use the bodyfat data set (not the bodyfat_orig data set) throughout the rest of the assignment.

```r
bodyfat <- bodyfat_orig

bodyfat <-bodyfat[-39, ] #Remove the point...Double checked with HW 4. This should do the tri
```

You should have discovered, from Homework 4, that there is a multicollinearity problem. The goal of this assignment is to continue this analysis by identifying variables to potentially remove from the model to resolve the multicollinearity issues.

4. Briefly explain why multicollinearity is a problem for multiple linear regression.

Some problems that multicollinearity can introduce into our analysis include, making the model more difficult to interpret, cause our estimates of the coefficients to become unreliable, as well as the predictor variables, inflated standard errors to name a few.

5. Briefly explain the similarities and differences between the following variable selection methods: best subset, forward, backward, and sequential replacement. Do not just copy the algorithms from the class notes - use your own words to explain what these methods are doing.

The best subset method involves checking all possible subsets of our fitted model. This is where we choose a metric to which we base the “bestness” of the subset such as (AIC, BIC or PMSE) and then whichever subset of our model who’s combinated of Betas best fits that metric is the model we will proceed with.

For Forward selection, just like above, choose which metric we will base our choice on but now we will begin with an intercept only model (BetaNot) and individually add new betas and checking if the new model reduces the residual sum of squares and our chosen metric improves. We continue this process of adding new Betas/Predictors to the model until adding more no longer improves the model.

Backward selection is the opposite of forward selection. We begin with all p-1 predictor/betas and then based off of our chosen metric, we remove Betas and re analyze the new model, looking for improvements with each removal and continue this process until removing predictors no longer serves our model.

The biggest note to keep in mind with these methods is that they will not all take you the same final model. For example, forward and backward will not necessarily produce the same model.

Sequential Replacement is a mixture between both forward and backward selection. It begins such as a forward, we only have our intercept model, but with each step, we can consider both add a predictor or removing one as well. We are not confined to only adding or subtracting. Again, we continue this process until further modification no longer serves to improve the model.

Above all, no method will ever give you the perfect model. It’s always good to experiment with different selection methods. Also, always recheck your model assumptions when you’ve created your new model.

6. Briefly explain how shrinkage methods work (bias-variance tradeoff).

Shrinkage involves shrinking the coefficients in the model in the direction to zero. There are three main methods of doing this, (at least in this class). These are ridge, LASSO and Elastic Net.

These methods look to decrease variance (By getting closer to zero by having the coefficients penalized), helping to avoid overfitting the model BUT this will introduce a minor bias into in our model. However, this will likely be worth the trade off for the improved model.

7. Briefly explain the similarities/differences between ridge regression and LASSO.

When we are dealing with multicollinearity, Ridge regression tends to produce more precise producted values for our model than OLS. It also keeps all of the predictors in the model.

LASSO is practically a variables selection process as it allows the estimates to be shrunk completely to zero. However, this will run the risk of introducing bias for the estimates for nonzero coefficients.

When we’re dealing with multicollinearity,LASSO will trend to selecting only one variable of multiple correlated predictors, leaving our model with potentially less variables effects.

In summary, when dealing with multicollinerity, ridge regression tends to be a much better choice.

8. When using the `bestglm` function in R for the stepwise methods, the response variable must be the last column in the data set for the `bestglm` function to work. Switch the order of the columns in the data set so that brozek is last.

```r
#Used ChatGPT for this part. Not trying to waste time on something like this.

# Identify the index of the "brozek" column
brozek_column_index <- which(names(bodyfat) == "brozek")

# Move the "brozek" column to the last position in the dataset
bodyfat <- bodyfat[, c(setdiff(1:ncol(bodyfat), brozek_column_index), brozek_column_index)]

head(bodyfat)
```

```
  age weight height neck chest abdom brozek
1  23 154.25  67.75 36.2  93.1  85.2   12.6
2  22 173.25  72.25 38.5  93.6  83.0    6.9
3  22 154.00  66.25 34.0  95.8  87.9   24.6
4  26 184.75  72.25 37.4 101.8  86.4   10.9
5  24 184.25  71.25 34.4  97.3 100.0   27.8
6  24 210.25  74.75 39.0 104.5  94.4   20.6
```

```r
#Hooray! ChatGPT worked
```

9. Apply the best subsets variable selection procedure to this data set using the bestglm function. Try it using AIC and BIC. Output a summary of the “best” model for each metric.

```r
# Modified from completed in class #5 :)

#BIC...

best_subsets_bic <- bestglm(bodyfat,
                            IC = "BIC",
                            method = "exhaustive")

# view variables included in the top 10 models
best_subsets_bic$BestModels
```

```
    age weight height  neck chest abdom Criterion
1 FALSE   TRUE  FALSE FALSE FALSE  TRUE  710.3413
2 FALSE  FALSE   TRUE  TRUE FALSE  TRUE  711.3498
3 FALSE   TRUE  FALSE  TRUE FALSE  TRUE  711.7113
4 FALSE   TRUE   TRUE FALSE FALSE  TRUE  713.7096
5 FALSE  FALSE   TRUE  TRUE  TRUE  TRUE  713.7214
```

```r
# view a summary of the "best" model
summary(best_subsets_bic$BestModel)
```

```
Call:
lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
    drop = FALSE], y = y))
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-10.0656  -2.9685  -0.1073   3.0258   9.9220

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -42.87040    2.45735 -17.446  < 2e-16 ***
weight       -0.12206    0.01966  -6.207 2.26e-09 ***
abdom         0.90426    0.05221  17.321  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.074 on 247 degrees of freedom
Multiple R-squared:  0.7212,    Adjusted R-squared:  0.7189
F-statistic: 319.4 on 2 and 247 DF,  p-value: < 2.2e-16
```

```r
#AIC...

best_subsets_aic <- bestglm(bodyfat,
                            IC = "AIC",
                            method = "exhaustive")

# view variables included in the top 10 models
best_subsets_aic$BestModels
```

```
    age weight height neck chest abdom Criterion
1 FALSE  FALSE   TRUE TRUE  TRUE  TRUE  699.6356
2 FALSE   TRUE   TRUE TRUE FALSE  TRUE  699.7564
3  TRUE  FALSE   TRUE TRUE  TRUE  TRUE  700.4139
4 FALSE   TRUE   TRUE TRUE  TRUE  TRUE  700.5866
5 FALSE  FALSE   TRUE TRUE FALSE  TRUE  700.7854
```

```r
# view a summary of the "best" model
summary(best_subsets_aic$BestModel)
```

```
Call:
lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
    drop = FALSE], y = y))

Residuals:
    Min      1Q  Median      3Q     Max
-8.9719 -3.0782  0.0843  2.9860  9.9388

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.89457    7.29416   0.945  0.34548
height      -0.44669    0.10406  -4.293 2.55e-05 ***
neck        -0.48479    0.17997  -2.694  0.00755 **
chest       -0.14382    0.08160  -1.762  0.07924 .
abdom        0.82586    0.06056  13.638  < 2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.028 on 245 degrees of freedom
Multiple R-squared:  0.7296,    Adjusted R-squared:  0.7252
F-statistic: 165.2 on 4 and 245 DF,  p-value: < 2.2e-16
```

10. Apply the forward selection procedure to this data set using the step() function in R. Try it using AIC and BIC (remember: in order to do BIC with the step() function you need to change the default value of k to be log(n) where n is the number of rows in the dataset!). Output a summary of the “best” models in each case.

```r
# Modified from in class #5 :)

#AIC...

base_mod <- lm(brozek ~ 1, data = bodyfat) # Intercept only model (null model, or base model)
full_mod <- lm(brozek ~ ., data = bodyfat) # All predictors in model (besides response)

forw_AIC <- step(base_mod,trace=0, # starting model for algorithm
     direction = "forward",
     scope=list(lower= base_mod, upper= full_mod))

summary(forw_AIC)
```

```
Call:
lm(formula = brozek ~ abdom + weight + neck + height, data = bodyfat)

Residuals:
    Min      1Q  Median      3Q     Max
-8.9928 -3.0686 -0.0225  2.9802 10.0446

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -12.53229   12.48676  -1.004   0.3165
abdom         0.83583    0.06607  12.651   <2e-16 ***
weight       -0.05594    0.03237  -1.728   0.0852 .
neck         -0.44136    0.19102  -2.310   0.0217 *
height       -0.27105    0.14819  -1.829   0.0686 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.029 on 245 degrees of freedom
Multiple R-squared:  0.7294,    Adjusted R-squared:  0.725
F-statistic: 165.1 on 4 and 245 DF,  p-value: < 2.2e-16
```

```r
forw_AIC$anova
```

```
      Step Df   Deviance Resid. Df Resid. Dev       AIC
1          NA         NA       249  14702.381 1020.5760
2  + abdom -1 9963.26785       248   4739.113  739.5361
3 + weight -1  639.47324       247   4099.640  705.2984
4   + neck -1   67.51570       246   4032.124  703.1469
5 + height -1   54.31443       245   3977.810  701.7564
```

```
#Could use alternative method here but it won't show the steps

#BIC


forw_BIC <- step(base_mod,trace=0, # starting model for algorithm
     direction = "forward",
     k=log(nrow(bodyfat)), #The main difference for BIC is this line of code right here
     scope=list(lower= base_mod, upper= full_mod))

summary(forw_BIC)
```

```
Call:
lm(formula = brozek ~ abdom + weight, data = bodyfat)

Residuals:
     Min       1Q   Median       3Q      Max
-10.0656  -2.9685  -0.1073   3.0258   9.9220

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -42.87040    2.45735 -17.446  < 2e-16 ***
abdom         0.90426    0.05221  17.321  < 2e-16 ***
weight       -0.12206    0.01966  -6.207 2.26e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.074 on 247 degrees of freedom
Multiple R-squared:  0.7212,    Adjusted R-squared:  0.7189
F-statistic: 319.4 on 2 and 247 DF,  p-value: < 2.2e-16
```

```
forw_BIC$anova
```

```
      Step Df  Deviance Resid. Df Resid. Dev       AIC
1          NA        NA       249  14702.381 1024.0974
2  + abdom -1 9963.2679       248   4739.113  746.5790
3 + weight -1  639.4732       247   4099.640  715.8628
```

11. Apply the backward selection procedure to this data set using the step() function in R. Try it using AIC and BIC (remember: in order to do BIC with the step() function you need to change the default value of k to be log(n) where n is the number of rows in the dataset!). Output a summary of the “best” models in each case.

```r
base_mod <- lm(brozek ~ 1, data = bodyfat) # Intercept only model (null model, or base model)
full_mod <- lm(brozek ~ ., data = bodyfat) # All predictors in model (besides response)


#AIC...Backward


back_AIC <- step(full_mod,trace=0, # starting model for algorithm
     direction = "backward",
     scope=list(lower= base_mod, upper= full_mod))

summary(back_AIC)
```

```
Call:
lm(formula = brozek ~ height + neck + chest + abdom, data = bodyfat)

Residuals:
    Min      1Q  Median      3Q     Max
-8.9719 -3.0782  0.0843  2.9860  9.9388

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.89457    7.29416   0.945  0.34548
height      -0.44669    0.10406  -4.293 2.55e-05 ***
neck        -0.48479    0.17997  -2.694  0.00755 **
chest       -0.14382    0.08160  -1.762  0.07924 .
abdom        0.82586    0.06056  13.638  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.028 on 245 degrees of freedom
Multiple R-squared:  0.7296,    Adjusted R-squared:  0.7252
F-statistic: 165.2 on 4 and 245 DF,  p-value: < 2.2e-16
```

```r
back_AIC$anova
```

```
      Step Df  Deviance Resid. Df Resid. Dev      AIC
1          NA        NA       243   3951.591 704.1032
2 - weight  1  4.915087       244   3956.506 702.4139
3    - age  1 19.381489       245   3975.888 701.6356
```

```r
#BIC...Backward Step


back_BIC <- step(full_mod,trace=0, # starting model for algorithm
     direction = "backward",
     k = log(nrow(bodyfat)),
     scope=list(lower= base_mod, upper= full_mod))
```

```
summary(back_BIC)
```

```
Call:
lm(formula = brozek ~ height + neck + abdom, data = bodyfat)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.7685 -3.0213 -0.0948  3.0441 10.1815 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.07086    7.25124   0.699 0.485019    
height      -0.45346    0.10443  -4.342 2.06e-05 ***
neck        -0.59792    0.16885  -3.541 0.000477 ***
abdom        0.74045    0.03646  20.307  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.046 on 246 degrees of freedom
Multiple R-squared:  0.7261,    Adjusted R-squared:  0.7228 
F-statistic: 217.4 on 3 and 246 DF,  p-value: < 2.2e-16
```

```
back_BIC$anova
```

```
      Step Df  Deviance Resid. Df Resid. Dev      AIC
1          NA        NA       243   3951.591 728.7534
2 - weight  1  4.915087       244   3956.506 723.5427
3    - age  1 19.381489       245   3975.888 719.2429
4  - chest  1 50.409597       246   4026.297 716.8712
```

12. Apply the sequential replacement selection procedure to this data set using the step() function. You may choose which metric you would like to use (eirther AIC or BIC). Try initializing it from the full model and the intercept only model. Output a summary of the single “best” model for the metric you chose.

```
# your code here

#Using BIC

base_mod <- lm(brozek ~ 1, data = bodyfat) # Intercept only model (null model, or base model)
full_mod <- lm(brozek ~ ., data = bodyfat) # All predictors in model (besides response)

step_BIC <- step(base_mod, trace=0, # starting model for algorithm
     direction = "both",
     k = log(nrow(bodyfat)),
     scope=list(lower= base_mod, upper= full_mod))
```

```
#Add the full model here..Then look at which looks better and then choose that one and give t

#Base is slightly better based off of the given values

step_BIC$anova #Add this to the other questions as well.
```

```
      Step Df  Deviance Resid. Df Resid. Dev       AIC
1          NA        NA       249  14702.381 1024.0974
2  + abdom -1 9963.2679       248   4739.113  746.5790
3 + weight -1  639.4732       247   4099.640  715.8628
```

```
summary(step_BIC)
```

```
Call:
lm(formula = brozek ~ abdom + weight, data = bodyfat)

Residuals:
     Min       1Q   Median       3Q      Max
-10.0656  -2.9685  -0.1073   3.0258   9.9220

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -42.87040    2.45735 -17.446  < 2e-16 ***
abdom         0.90426    0.05221  17.321  < 2e-16 ***
weight       -0.12206    0.01966  -6.207 2.26e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.074 on 247 degrees of freedom
Multiple R-squared:  0.7212,    Adjusted R-squared:  0.7189
F-statistic: 319.4 on 2 and 247 DF,  p-value: < 2.2e-16
```

```
step_BIC_full<- step(full_mod, trace=0, # starting model for algorithm
     direction = "both",
     k = log(nrow(bodyfat)),
     scope=list(lower= base_mod, upper= full_mod))

#Add the full model here..Then look at which looks better and then choose that one and give t


#step_BIC_full$anova #Add this to the other questions as well.
#summary(step_BIC_full)


#Commented out the summary above because we'll go with the base model
```

13. Apply LASSO to this data set using the MSE metric. Output the coefficient values corresponding to the 1 standard error rule (do not output any plots).

```r
#Lasso will probably be better


bodyfat_x <- as.matrix(bodyfat[, 1:6]) # predictors
bodyfat_y <- bodyfat[, 7] # response

# use cross validation to pick the "best" (based on MSE) lambda
set.seed(50)
env_ridge_cv <- cv.glmnet(x = bodyfat_x, # automatically includes a column of ones for the in
                          y = bodyfat_y,
                          type.measure = "mse",
                          alpha = 0)  # 0 is code for "ridge regression"


# use cross validation to pick the "best" (based on MSE) lambda


bodyfat_lasso_cv <- cv.glmnet(x = bodyfat_x, # automatically includes a column of ones for th
                          y = bodyfat_y,
                          type.measure = "mse",
                          alpha = 1)  # 1 makes this the LASSO


# lambda.min: value of lambda that gives minimum mean cross-validated error
bodyfat_lasso_cv$lambda.min
```

```
[1] 0.01492685
```

```r
# lambda.1se: value of lambda within 1 standard error of the minimum
# cross-validated error
bodyfat_lasso_cv$lambda.1se
```

```
[1] 0.466571
```

```r
coef(bodyfat_lasso_cv, s = "lambda.min")
```

```
7 x 1 sparse Matrix of class "dgCMatrix"
                     s1
(Intercept) -1.37687024
age          0.01912264
weight      -0.01576586
height      -0.36558206
neck        -0.45200095
chest       -0.10177095
abdom        0.81549495
```

```r
coef(bodyfat_lasso_cv, s = "lambda.1se")
```

```
7 x 1 sparse Matrix of class "dgCMatrix"
                     s1
(Intercept) -13.17110233
age           0.00872195
weight        .
height       -0.32275286
neck          .
chest         .
abdom         0.58851431
```

14. Apply Elastic Net to this data set using the MSE metric. Output the coefficient values corresponding to the 1 standard error rule (do not output any plots).

```r
set.seed(50)

bodyfat_elastic_cv <- cv.glmnet(x = bodyfat_x, # automatically includes a column of ones for
                          y = bodyfat_y,
                          type.measure = "mse",
                          alpha = .5)  # .5 - this fits the elastic net with half ridge penal


# lambda.min: value of lambda that gives minimum mean cross-validated error
bodyfat_elastic_cv$lambda.min
```

```
[1] 0.03276441
```

```r
# lambda.1se: value of lambda within 1 standard error of the minimum
# cross-validated error
bodyfat_elastic_cv$lambda.1se
```

```
[1] 0.5860404
```

```r
coef(bodyfat_elastic_cv, s = "lambda.min")
```

```
7 x 1 sparse Matrix of class "dgCMatrix"
                     s1
(Intercept) -0.30691450
age          0.02122208
weight      -0.01068811
height      -0.37934042
neck        -0.45669044
chest       -0.09411263
abdom        0.79713819
```

```r
coef(bodyfat_elastic_cv, s = "lambda.1se")
```

```
7 x 1 sparse Matrix of class "dgCMatrix"
                     s1
(Intercept) -10.58193343
```

```
age           0.02146324
weight        .
height       -0.35221851
neck         -0.01698972
chest         .
abdom         0.58369704
```

15. Fill in the table below with "X"s (like the one at the end of the Module 5 course notes: a row for each variable, a column for each variable selection method, an "X" in a cell means the variable was included for that variable selection method). For the best subset, forward, backward, and sequential replacement columns, use either AIC or BIC. In other words, only report the models for either AIC or BIC, not both, but make sure you're consistent in the table.

BIC

Variable	Best Subset	Forward	Backward	Sequential Replacement	LASSO	Elastic Net
age					x	x
weight	x	x		x		
height			x		x	x
neck			x			x
chest						
abdom	x	x	x	x	x	x

16. Now that you have seen the various results from the different methods, pick a subset of variables that you will include in the model. Which variables do you choose to include in the model? Why?

I will most definitely include abdomen, as it appears in every method of selection. I will also include height and weight as they are the second most frequent variables to occur in our various tests. These variables also make sense intuitively that they would be good predictors for body fat percentage, we now have data to back up this decision.

17. Create the multiple linear regression model with the variables you listed in the previous question (alternatively, you can call the best model using $BestModel). Print a summary of the results. Save the residuals from this model to the bodyfat dataframe.

```
# your code here

bodyFat_lm <- lm(brozek ~ weight+height+abdom, data = bodyfat) #Add each one individually her
summary(bodyFat_lm)
```

```
Call:
lm(formula = brozek ~ weight + height + abdom, data = bodyfat)

Residuals:
     Min       1Q   Median       3Q      Max
-10.2697  -3.0163   0.0479   3.0928   9.3556

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -27.62429   10.73546  -2.573  0.01066 *
weight       -0.09141    0.02875  -3.180  0.00166 **
height       -0.21514    0.14749  -1.459  0.14592
abdom         0.84382    0.06656  12.678  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.065 on 246 degrees of freedom
Multiple R-squared:  0.7235,    Adjusted R-squared:  0.7202
F-statistic: 214.6 on 3 and 246 DF,  p-value: < 2.2e-16
```

```r
bodyfat <- bodyfat[, !names(bodyfat) %in% c("age", "neck", "chest")] #Get rid of the other va

bodyfat$residuals <- bodyFat_lm$residuals
bodyfat$fits <- bodyFat_lm$fitted.values

head(bodyfat)
```

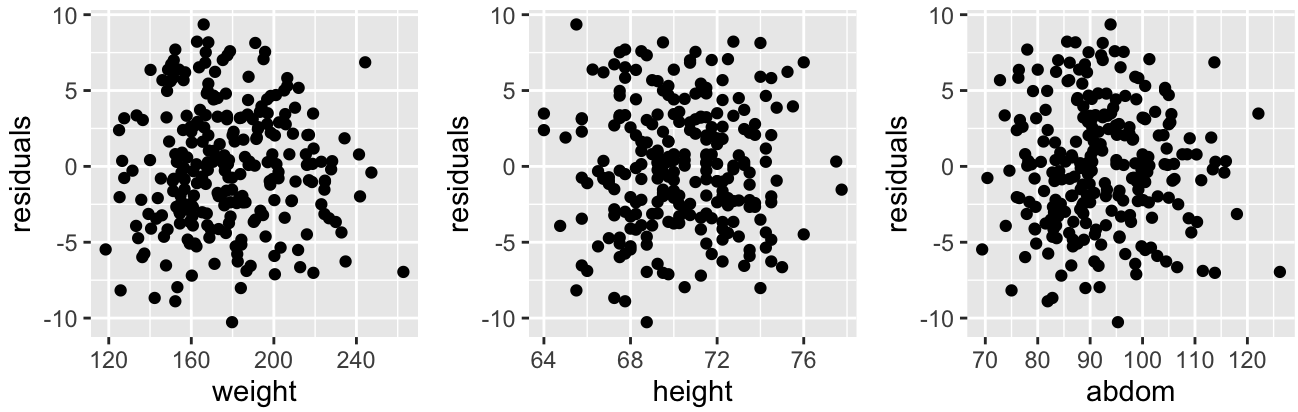
```
  weight height abdom brozek residuals     fits
1 154.25  67.75  85.2   12.6 -2.993173 15.59317
2 173.25  72.25  83.0    6.9 -4.131854 11.03185
3 154.00  66.25  87.9   24.6  6.382951 18.21705
4 184.75  72.25  86.4   10.9 -1.949630 12.84963
5 184.25  71.25 100.0   27.8  3.213597 24.58640
6 210.25  74.75  94.4   20.6  3.868618 16.73138
```

```r
#looks good!
```

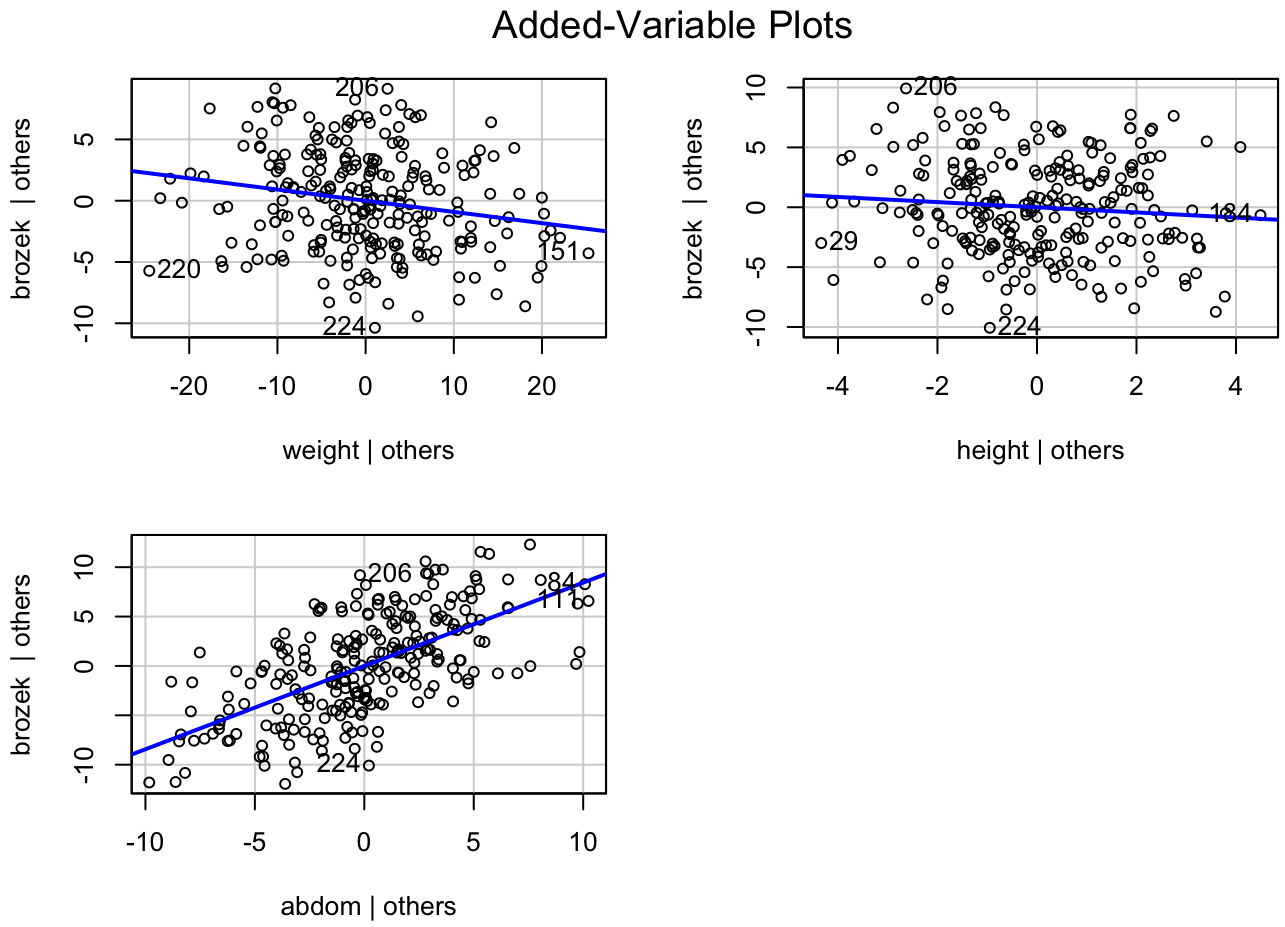
Now that you have chosen a model, the next several questions ask you to check some of the model assumptions. For each assumption, (1) perform appropriate diagnostics to determine if the assumption is violated, and (2) explain whether or not you think the assumption is violated and why you think that. Note: you can copy (then modify) a lot of your code from Homework 4 to answer these questions.

18. (L) The Xs vs Y are linear (use the residual by predictor plots and the partial regression plots)

```r
# residual vs. predictor plots


resid_vs_weight <- ggplot(data = bodyfat) +
  geom_point(mapping = aes(x = weight, y = residuals)) +
  theme(aspect.ratio = 1)

resid_vs_height <- ggplot(data = bodyfat) +
  geom_point(mapping = aes(x = height, y = residuals)) +
  theme(aspect.ratio = 1)


resid_vs_abdom <- ggplot(data = bodyfat) +
  geom_point(mapping = aes(x = abdom, y = residuals)) +
  theme(aspect.ratio = 1)


(resid_vs_weight | resid_vs_height) |
  (resid_vs_abdom)
```




```r
# partial regression plots

avPlots(bodyFat_lm)
```



Linearity is looking great!

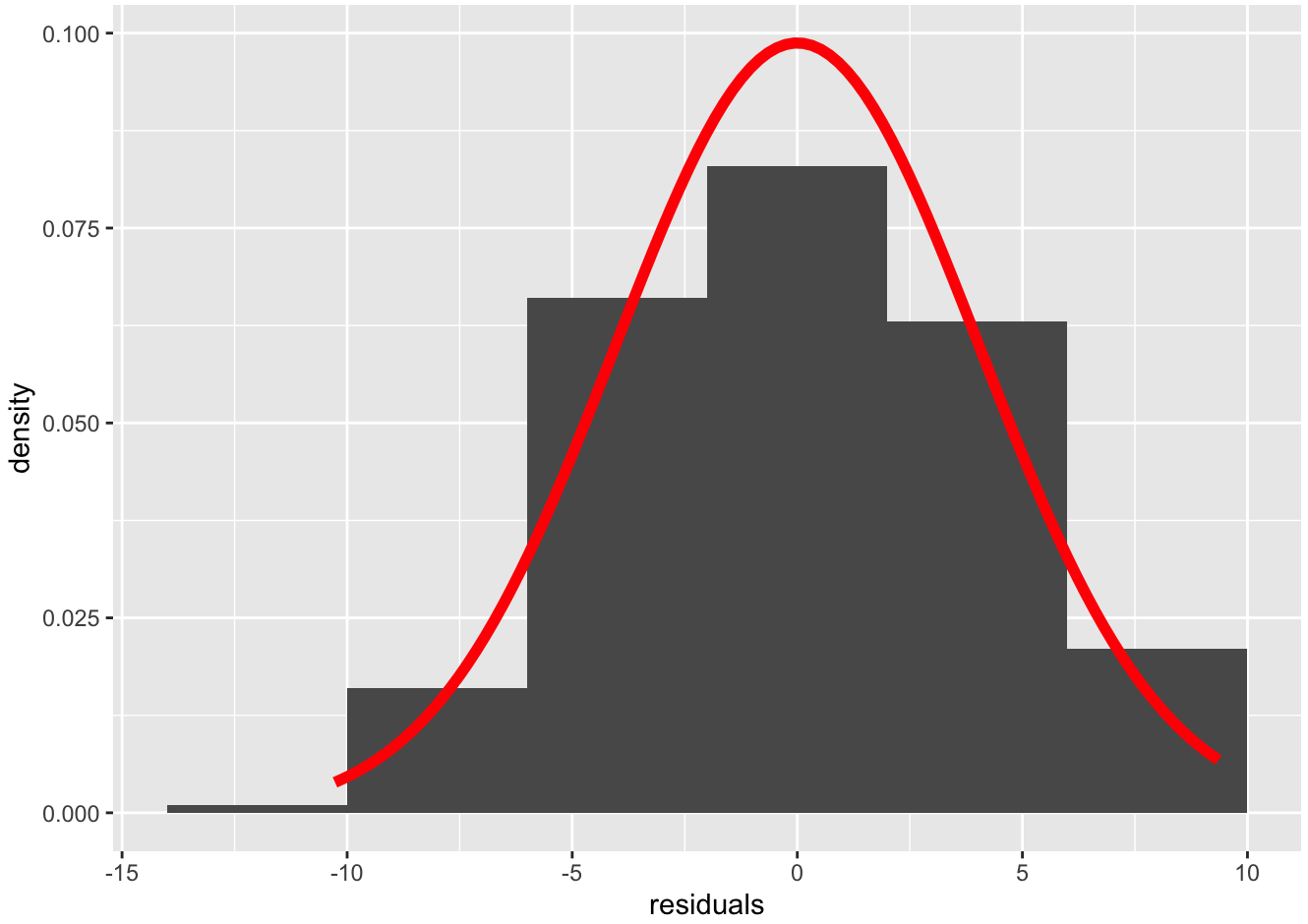
With our residual vs predictor plots, we do not see any striking trend for any of the variables. There are no curves at any of our x values.

For our partial regression plots, we have mostly flat lines. We do some some curves in abdom, but nothing too concerning. For the most part, straight lines.

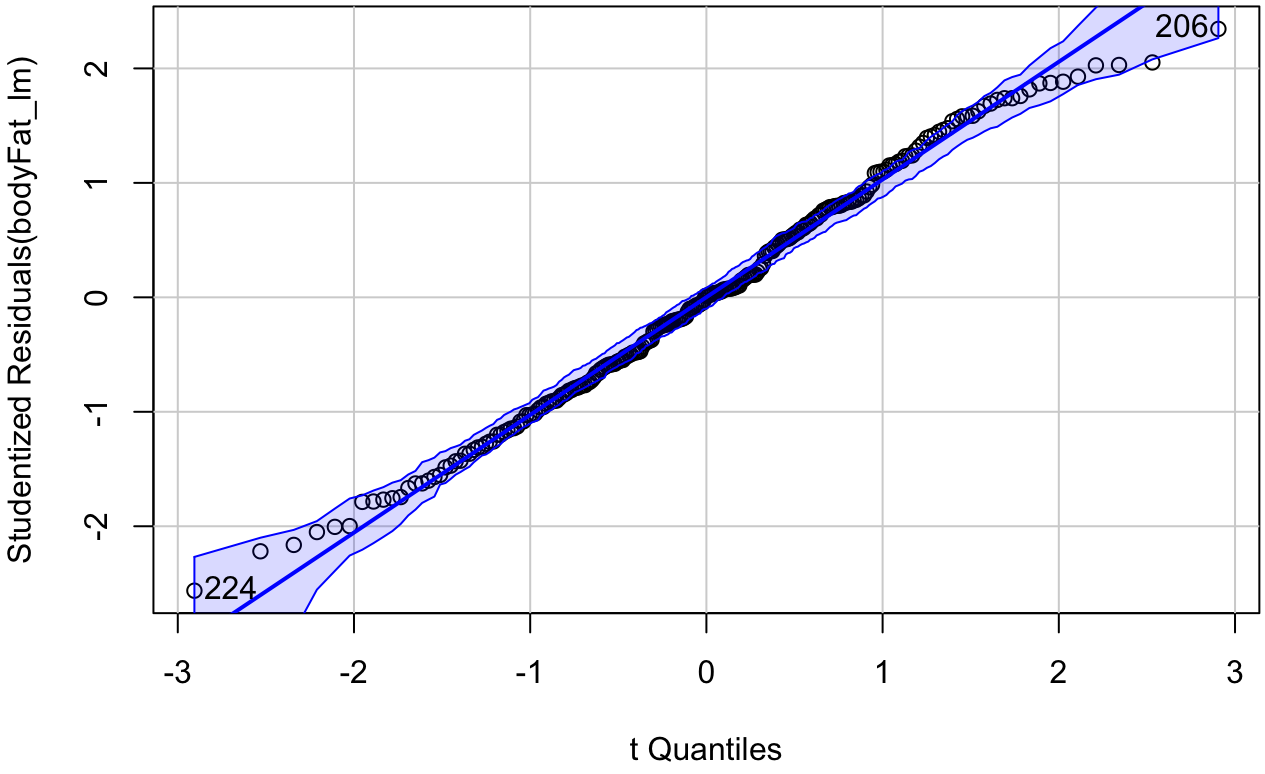
I'm confident in stating that linearity is met based off these tests.

19. (N) The residuals are normally distributed (use a histogram, qq plot, and shapiro wilk test)

```
# Diagnostic 1 Histogram


ggplot(data = bodyfat) +
  geom_histogram(aes(x = residuals, y = after_stat(density)),
                 binwidth = 4) +
  stat_function(fun = dnorm, color = "red", linewidth = 2,
                args = list(mean = mean(bodyfat$residuals),
                            sd = sd(bodyfat$residuals)))
```




```
# Diagnostic 2 qq plot


qqPlot(bodyFat_lm)
```




```
206 224
205 223
```

```
# Diagnostic 3 shapiro Wilk


shapiro.test(bodyfat$residuals)
```

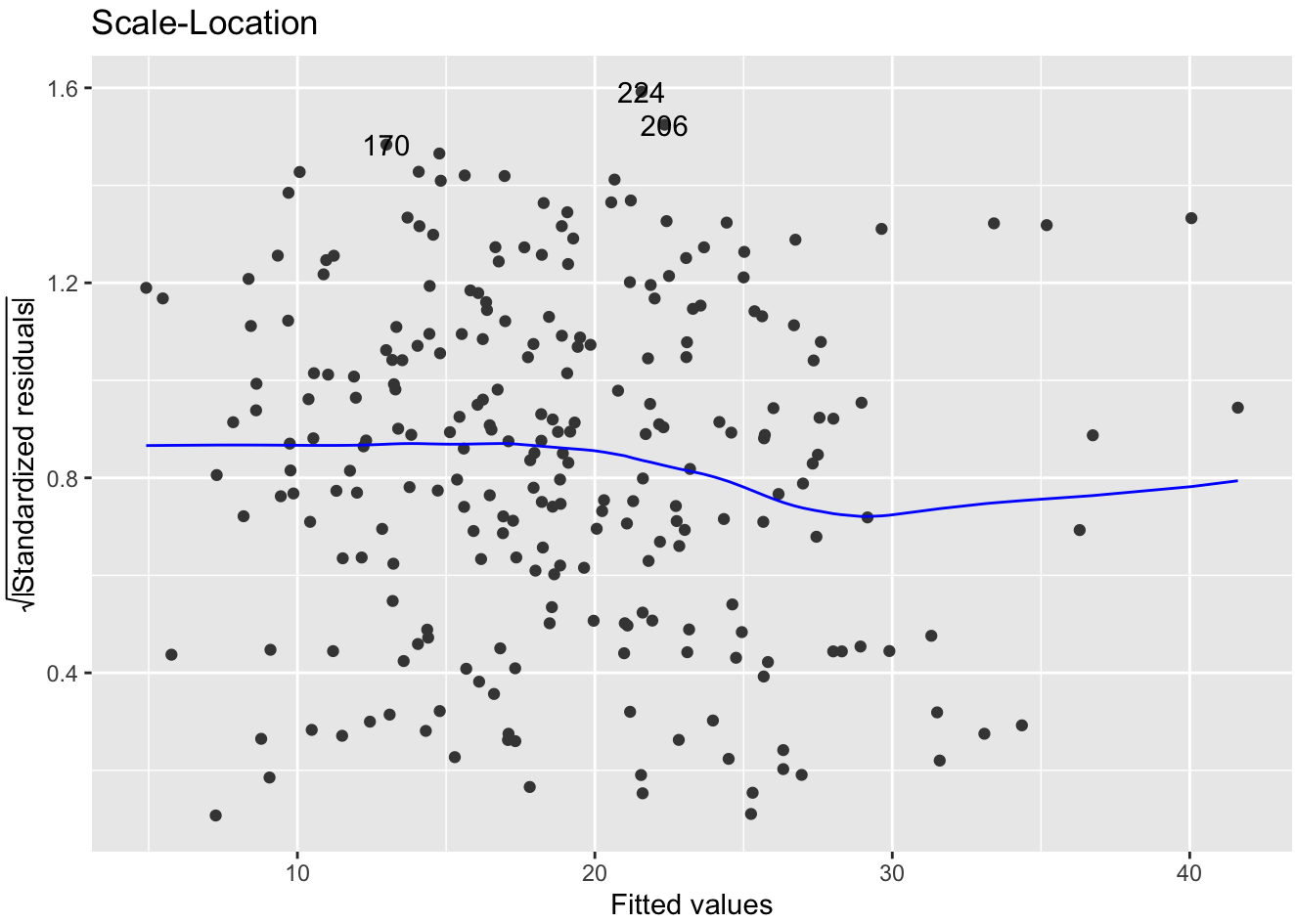
```

    Shapiro-Wilk normality test

data:  bodyfat$residuals
W = 0.99129, p-value = 0.1438
```

Normally distributed residuals are also looking great!

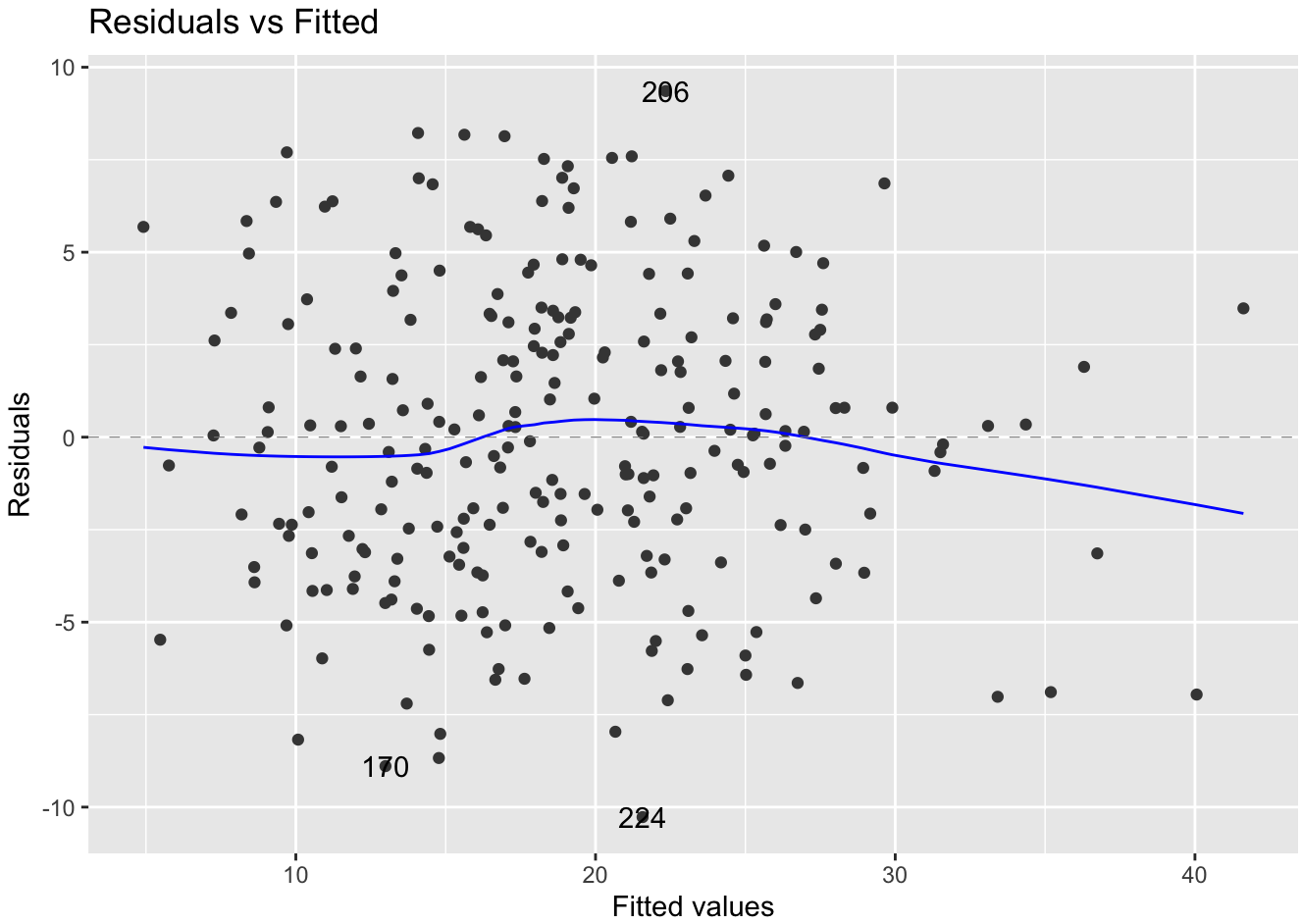
With the the histogram and qq plot as we can see a nice normal curve and our values mostly stay within the boundaries in the qq plot (we see some values leave, but nothing too concerning). Finally, our Shaprio Wilk test gives us a large P value, which could be larger, but this serves confirm normal distrubtion in conjuction with the other two tests. Further confirming that the residuals are indeed normally distributed and that this assumption is met.

20. (E) The residuals have equal/constant variance across all values of X (use the residuals vs. fitted values plot and scale - location plot)

```
#Scale location
autoplot(bodyFat_lm, which = 3, nrow = 1, ncol = 1)
```




```
# residuals vs fitted values

autoplot(bodyFat_lm, which = 1, nrow = 1, ncol = 1)
```

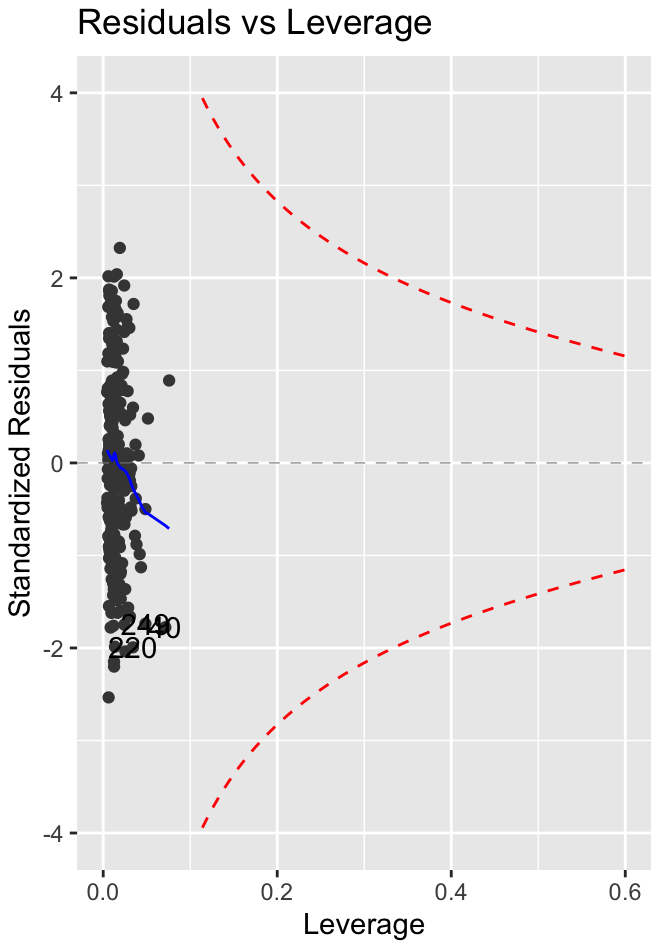


Here we have a consistent spread in both of plots without any blaring patterns. Our line mostly stays straight, not perfect but already a huge improvement after the modifications with the model compared to our previous analysis #4

Constant variance is satisfied.

21. (A) The model describes all observations (i.e., there are no influential points) (use the cooks distance > 0.5 plot).

```
# Cook's Distance

cd_cont_pos <- function(leverage, level, model) {sqrt(level*length(coef(model))*(1-leverage)/
cd_cont_neg <- function(leverage, level, model) {-cd_cont_pos(leverage, level, model)}

cd_threshold <- 0.5
autoplot(bodyFat_lm, which = 5) +
  stat_function(fun = cd_cont_pos,
                args = list(level = cd_threshold, model = bodyFat_lm),
                xlim = c(0, 0.6), lty = 2, colour = "red") +
  stat_function(fun = cd_cont_neg,
                args = list(level = cd_threshold, model = bodyFat_lm),
                xlim = c(0, 0.6), lty = 2, colour = "red") +
  scale_y_continuous(limits = c(-4, 4))
```



Beautiful! No more influential point! Meaning that the model describes all observations. All points stay within our specified 0.5 boundary.

22. No multicollinearity (use the scatterplot matrix, correlation matrix, and variance inflation factors).

```
fat_vifs <- vif(bodyFat_lm)
fat_vifs
```

```
  weight   height    abdom 
9.109057 2.231029 6.927957 
```

```
max(fat_vifs)
```

```
[1] 9.109057
```

```
mean(fat_vifs)
```

```
[1] 6.089347
```

Our VIFS could be better. Individually they meet our needed threshold of staying below 10. Although weight is a bit high. However, the mean is still above what we would like. (We want it to be at 5 or below). But it is a large improvement from our previous analysis which had the mean at roughly 7.63

I will say that there is slight multicolinearity but it shouldn't impede us from using this model. Just something to be aware of when we use this model to make any predictions or decisions.

23. Given the results from your model assumption checking, what would you do next to continue this analysis?

Pass on my findings to someone much smarter than me that is being paid to do these things. This seems to be a good starting point but preferably more analysis should be done by real data analysis people.

24. Briefly summarize what you learned, personally, from this analysis about the statistics, model fitting process, etc.

This was a great follow up to homework number 4. Because, with the knowledge I had previously, I wouldn't see much of an issue with using the model we already had. Sure there were some issues but we can always argue that this is just the cost of working with data in the real world. I learned a lot in this homework that this isn't the case. It was interesting to see the adjustments we can make to our model in order to make more sound predictions. I think this was quite valuable in showing us that we should never settle with the first model that we fit but should always be looking for possible improvements.

25. Briefly summarize what you learned from this analysis *to a non-statistician*. Write a few sentences about (1) the purpose of this data set and analysis and (2) what you learned about this data set from your analysis. Write your response as if you were addressing a business manager (avoid using statistics jargon) and just provide the main take-aways.

We previously looked at what kind of things we could use to predict a person's body fat percentage. Such as a person's neck, age, height etc. Instead of going through a traditional process of measuring body fat percentage.

We originally started with a person's age, weight, height, neck, chest and abdomen. But we should that we were running into issues with having this things to try and make this prediction. So we ran a number of tests to see if we could get rid of any of these things to try and help us make a more accurate prediction. This was an extensive process and found that the best things to help us predict a person's body fat percentage was their weight, height and size of their abdomen. We now feel confident that we are come up with a mathematical method that can take these numbers of a person and give a robust estimate of their body fat percentage.

Cool stuff!