Body Fat Regression Analysis

Jeremiah Anderson (5744842)

February 04, 2023

Objective

The objective of this assignment is to run regression diagnostics on a linear model where body fat percentage is the response variable.

Introduction

First, lets read in out dataset and take a look at it.

```
fat <- read.csv(file = "fat.csv")
head(fat)
```

```
##   brozek age weight height neck chest abdom   hip
## 1   12.6  23  154.2  67.75 36.2  93.1  85.2  94.5
## 2    6.9  22  173.2  72.25 38.5  93.6  83.0  98.7
## 3   24.6  22  154.0  66.25 34.0  95.8  87.9  99.2
## 4   10.9  26  184.8  72.25 37.4 101.8  86.4 101.2
## 5   27.8  24  184.2  71.25 34.4  97.3 100.0 101.9
## 6   20.6  24  210.2  74.75 39.0 104.5  94.4 107.8
```

Here, wee see the 8 numeric variables. Brozek, which is percent body fat, is going to be our response variable, while the other 7 will be our predictors. For information on the predictor variables, see the codebook.

Creating the Model

Now we can fit our model.

```
# Set up linear model
fatMod <- lm(formula = brozek ~ age + weight + height + neck + chest + abdom + hip, data = fat)

# Take a look at the first 5 observations and the first 5 residuals
head(fitted(fatMod))
```

```
##     1     2     3     4     5     6
## 15.87 10.43 18.82 12.80 26.65 16.45
```

```
head(residuals(fatMod))
```

```
##      1      2      3      4      5      6
## -3.266 -3.525  5.779 -1.900  1.149  4.146
```

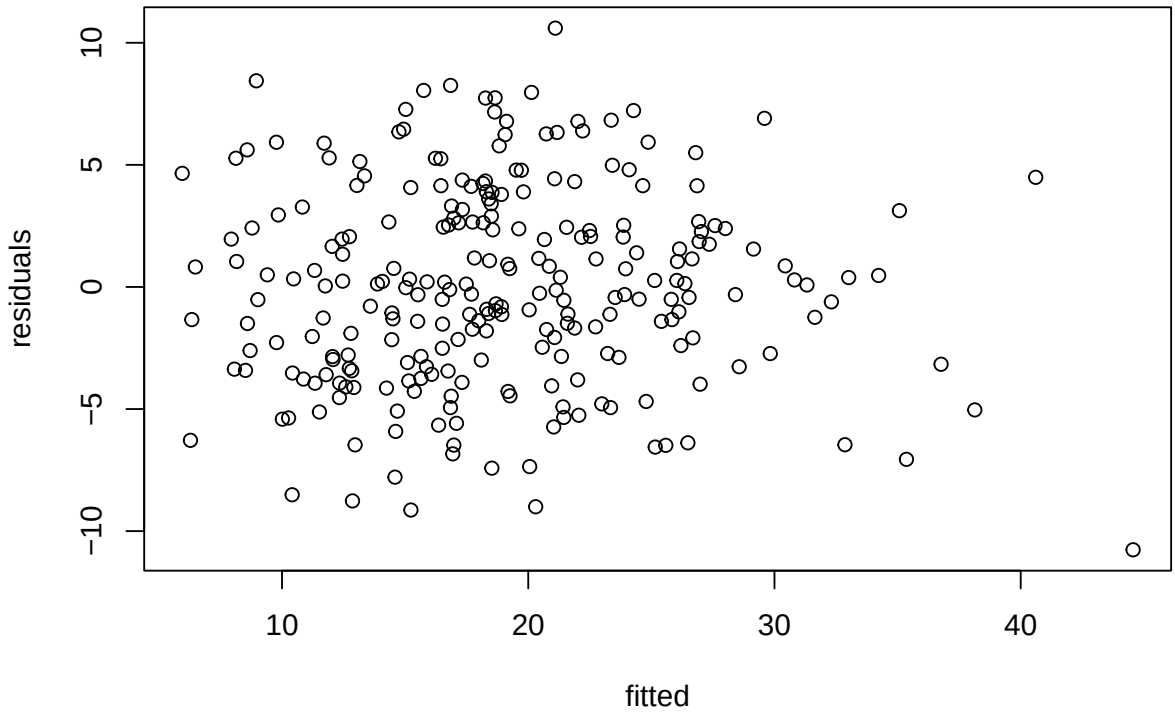
Here we see both the predicted values as well as the residuals. The residuals tell us the difference between the actual values and the predicted values. Now that we have the residuals, we can run our first diagnostic.

Regression Diagnostics

Constant Variation Assumption

The constant variance assumption assumes that the variance in the residuals is the same for every observation. If the assumption is not met, there will be inaccuracy in both our confidence intervals and our p-values. To test this assumption, we will plot a the residuals against the fitted values.

```
plot(fitted(fatMod), residuals(fatMod), xlab = "fitted", ylab = "residuals")
```



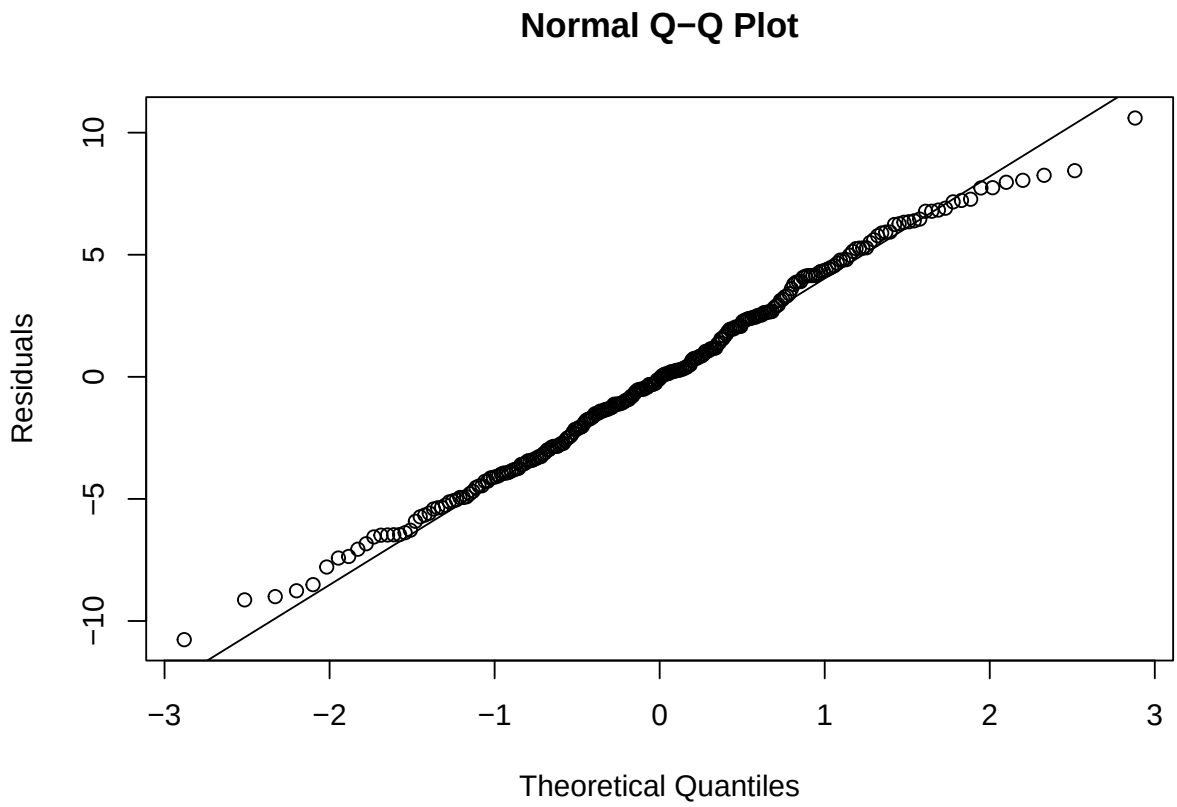
Since the spread of the residuals is roughly even throughout the fitted values, we can say that the constant variation assumption has been met.

The Normal Assumption

Next, we can use the residuals to check for normality. We do this by creating a Q-Q plot, or Quantile-Quantile plot, and plotting the residuals to see if they follow the Q-Q line.

```
# Plot the residuals from our model
qqnorm(residuals(fatMod), ylab = "Residuals")

# Plot a normal line
qqline(residuals(fatMod))
```



It is evident from this plot that the residuals, also known as random errors, follow a normal distribution. So, it seems that the normal assumption has been met. However, we can go one step further and test for normality with the Shapiro-Wilks test in order to double check.

```
shapiro.test(residuals(fatMod))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(fatMod)
## W = 0.99, p-value = 0.5
```

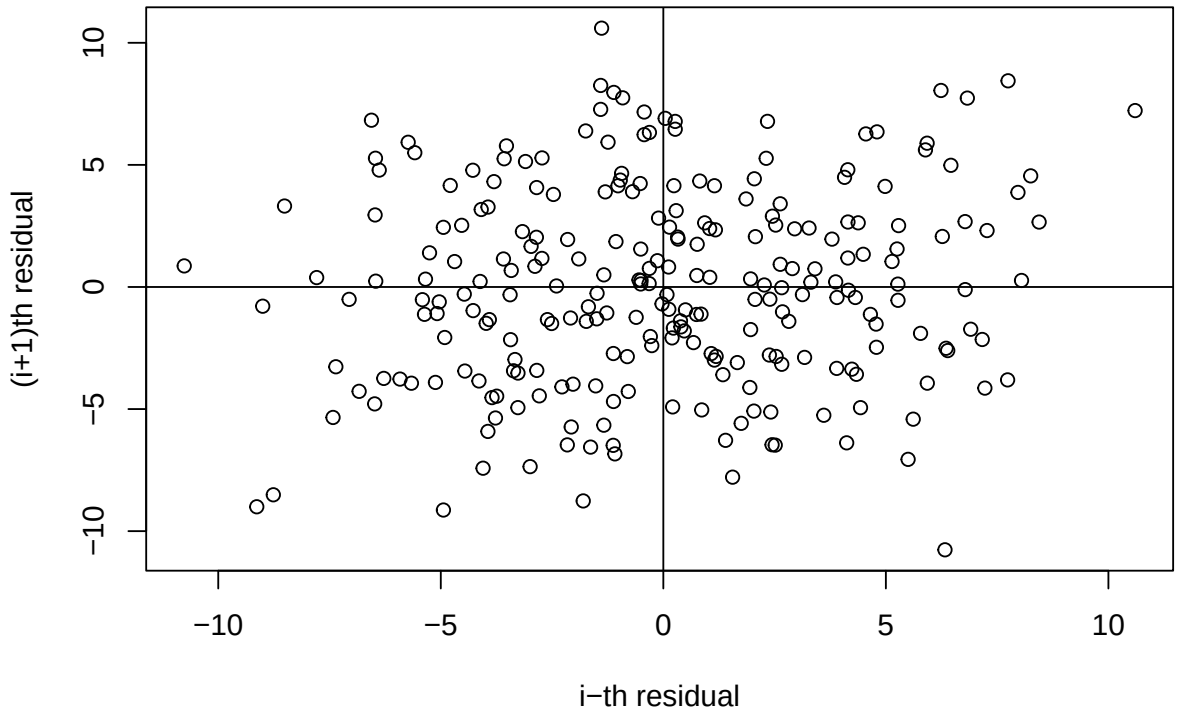
This test gives us two things, a p-value and the W statistic. The W statistic is a measure of how well the standardized residuals would fit the corresponding standard normal quantiles. At W = 0.99, we can

assume that the normality assumption is met. Additionally, for this test, the null hypothesis is that the random errors follow a normal distribution. Since out p -value is >= 0.5, we would fail to reject such a null hypothesis.

Serial Correlation

Serial correlation is when the i-th residual and the i-th + 1 residual are more similar than a randomly selected pair on average. This becomes an issue because it effects the standard error of our estimators, causing use to believe they are more accurate than they are. For this reason, linear regression assumes there is no serial correlation among observations. Lets take a look at a plot of successive pairs of residuals to see whether this assumption is met.

```
n <- nrow(fat)
plot(tail(residuals(fatMod), n-1) ~ head(residuals(fatMod), n-1),
     xlab = "i-th residual", ylab = "(i+1)th residual")
abline(h=0, v=0)
```



We can see that there is no trend in the plot, and the residual pairs seem to be spread randomly. This is evidence that there is no serial correlation among pairs.

High Leverage Points

Next we will test for high leverage points. High leverage points are data points with an extremely high or extremely low predictor value. When we have high leverage points, a single predictor can have too much

or too little influence on the response variable. For this diagnostic, we will say any leverage point above 3(p+1)/n is a high leverage point, where p is the number of predictors.

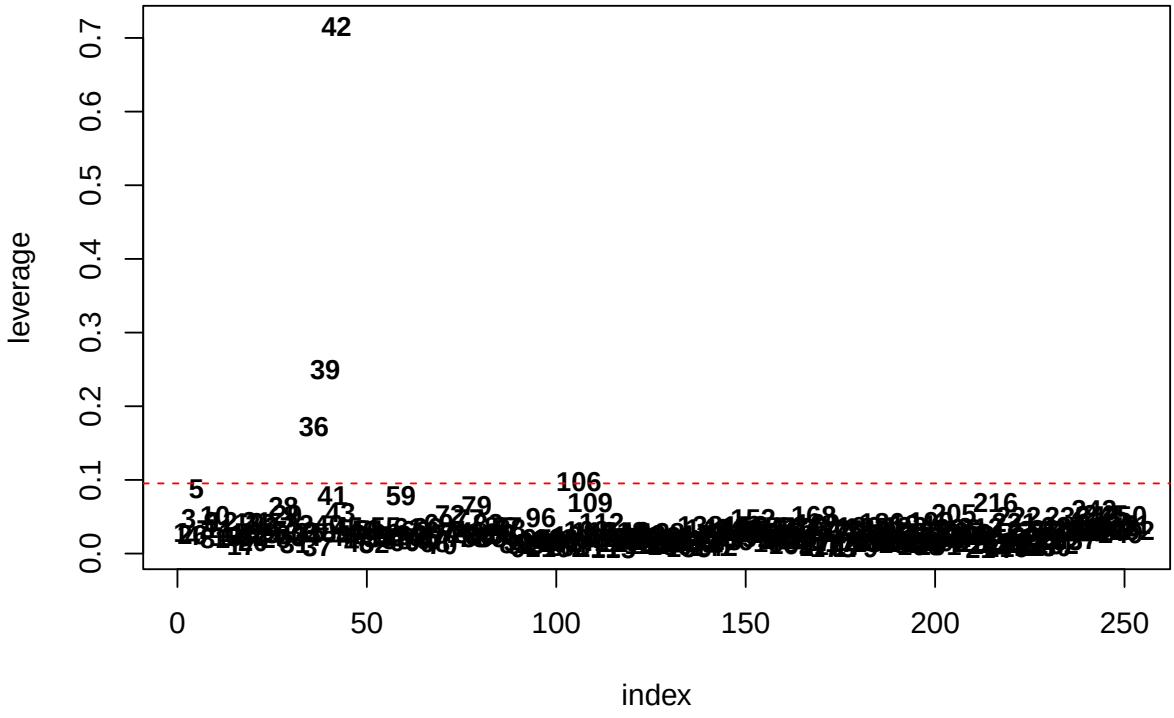
```
# Calculate leverage points
x <- model.matrix(fatMod)
H <- x %*% solve(crossprod(x), t(x))
lev <- diag(H)
sum(lev)
```

```
## [1] 8
```

```
# Store the number of predictors in a variable
p <- 7

# Create a dataframe with the leverage points
dat <- data.frame(index = seq(length(lev)), leverage = lev)

# Plot the leverage points as well as the cutoff line\
plot(leverage ~ index, col = "white", data = dat, pch = NULL)
text(leverage ~ index, labels = index, data = dat, cex = 0.9, font =2)
abline(h = 3*(p+1)/n, col = "red", lty = 2)
```



We can see from the graph that points 36, 39, 106, and 42 are high leverage points. Since the objective of this program is analysis, we won't change anything in the model. Still, it is important to know that these points have extreme predictor values and influence the accuracy of the model.

Outliers

Outliers are observations that are far from the other observations. Outliers can interfere with the results of a hypothesis test by skewing the data. It can to rejection of a true null hypothesis or acceptance of a false null hypothesis. To test for outliers, we will compute the standardized residuals of out model. Any observation above 3, or below -3 will be considered and outlier.

```
# Use the residual standard error to standardize the residuals
rse <- summary(fatMod)$sigma
r <- residuals(fatMod)/(rse *sqrt(1-lev))
r
```

```
##         1         2         3         4         5         6         7         8
## -0.806902 -0.869815  1.441813 -0.468412  0.292817  1.023026  0.293699 -0.699794
##         9        10        11        12        13        14        15        16
## -0.850602  0.169322 -0.564687 -1.018718  0.781838 -0.709994  0.209246 -0.273962
##        17        18        19        20        21        22        23        24
##  1.950935  0.956339  0.052002 -1.205830 -0.515342 -1.425274  1.463482  1.450999
##        25        26        27        28        29        30        31        32
##  1.386882 -1.331590 -0.129097  1.065889 -0.842491 -0.846901 -0.529040 -1.606785
##        33        34        35        36        37        38        39        40
##  1.303393 -0.134566  0.071177  0.836547 -0.076489  1.563717 -3.025477  0.214299
##        41        42        43        44        45        46        47        48
## -1.278144 -0.278659 -0.311235  1.461326 -0.977408  0.808014  0.594925 -1.256263
##        49        50        51        52        53        54        55        56
## -0.967091 -0.331021 -1.401149 -0.966495 -1.456610 -0.928888 -1.331397 -0.275938
##        57        58        59        60        61        62        63        64
## -1.155827  0.255857  0.607208 -0.122860  0.066051  1.671935  0.662519 -0.250111
##        65        66        67        68        69        70        71        72
##  1.021171  1.190458  1.562130 -0.615973 -0.371944 -0.319676  0.961560 -0.834152
##        73        74        75        76        77        78        79        80
## -0.733355  0.411140 -0.762594  1.268364  0.260255  0.097558 -0.411686 -1.616921
##        81        82        83        84        85        86        87        88
##  1.687724  1.923790 -0.937128  1.064125 -0.106614  1.767941 -0.533385  0.481660
##        89        90        91        92        93        94        95        96
## -1.009585  0.055376 -0.412552 -0.198080 -0.699094  0.499216 -1.250739 -0.271948
##        97        98        99       100       101       102       103       104
## -1.674885 -1.051029 -0.237685  1.076940  0.644656  0.227130  0.642625  0.836125
##       105       106       107       108       109       110       111       112
##  0.183518 -0.288831 -1.595760 -1.184890  1.049648 -0.032957  0.263016 -0.679123
##       113       114       115       116       117       118       119       120
##  0.288089  0.576417  1.657569 -0.026324  0.692040 -0.348148  2.040147  1.120098
##       121       122       123       124       125       126       127       128
##  1.542361  0.509549  0.508393 -0.124496  0.030465 -0.224384  1.897901  2.086410
##       129       130       131       132       133       134       135       136
##  0.652446 -0.007145 -0.170552  0.957114 -0.105727  1.536441  1.967883  0.065524
##       137       138       139       140       141       142       143       144
##  1.585869  1.237505  1.016417 -1.582004  1.171893 -0.604215  0.930911  0.483677
##       145       146       147       148       149       150       151       152
##  0.081844  0.481934 -0.431668  1.581422 -0.643753 -0.331764  0.122334 -0.233934
##       153       154       155       156       157       158       159       160
##  1.149494 -0.278713 -0.674907  1.305098  0.623818 -1.589170  0.726472  0.585673
##       161       162       163       164       165       166       167       168
## -0.686263 -1.092471 -0.852833 -0.079069  0.034224  0.607484  0.713526  0.188306
```

```
##       169       170       171       172       173       174       175       176
##  0.116851 -0.440253 -2.151757 -2.105655  0.810852  0.047999 -0.516235 -0.314102
##       177       178       179       180       181       182       183       184
## -0.261168  0.459698  0.881599 -1.303268  0.343331 -1.557534 -0.919265 -1.097515
##       185       186       187       188       189       190       191       192
## -0.071785 -0.505093 -0.984235 -0.367594 -0.063749 -0.589821  0.010267  1.713492
##       193       194       195       196       197       198       199       200
## -0.424622 -0.349921  1.786849  0.565722  1.292689  0.029085  0.203832  1.072091
##       201       202       203       204       205       206       207       208
## -0.875857  1.292347  0.382026 -1.917354  0.096403 -0.343490  2.620990  1.781331
##       209       210       211       212       213       214       215       216
## -1.022387 -0.948907 -1.111854  0.626206  0.622006 -0.695740  0.997392  1.133314
##       217       218       219       220       221       222       223       224
##  0.330597 -0.880765  0.280583 -0.735173 -1.830618 -0.816361 -1.212340 -2.239930
##       225       226       227       228       229       230       231       232
## -2.226569 -0.194061 -1.047664  1.182810 -0.374054 -0.991366 -1.837678 -1.309666
##       233       234       235       236       237       238       239       240
##  0.078800  0.507343  1.107327 -1.221900  0.600740 -1.601969  0.058813  1.025701
##       241       242       243       244       245       246       247       248
##  0.658143 -0.794942  0.567286  0.020229 -0.078558  0.188157  0.433472 -1.385198
##       249       250       251       252
##  1.357858 -1.764931 -0.126146  0.384023
```

From our data we can see that there is one outlier, observation 39. Since this observation is so far from the others, the results of a hypothesis test could be skewed towards it.

Influential Points

Influential points are points the have a high impact on the slope of the regression line. Removing an influential point will always change the model significantly. It is important to test for influential points and to see if they may be cause by an error. It is also important to decide whether or not to remove the influential points. We will use an estimate called Cook's Distance, in order to test for influential points. Any observation with a Cook's Distance above 0.02 will be called high influence points.

```
# Calculate cook's distance for all of the leverage points
d <- r^2*lev / (1-lev) / (p+1)

# FIlter out observations with a Cook's distance greater thanm 0.02
d[d>0.02]
```

```
##      39      42     207     250
## 0.37987 0.02438 0.02657 0.02099
```

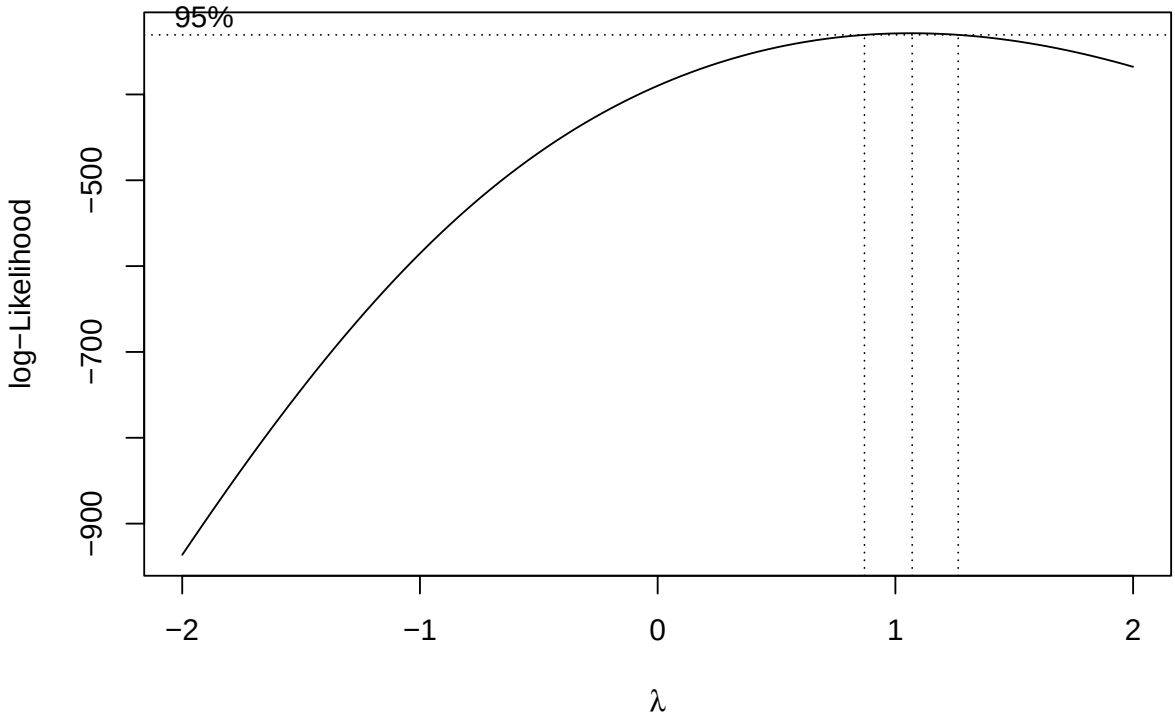
We see 4 observations with a Cook's distance greater than 0.02. We will say that observations 39, 42, 207, and 250 are influential points.

Box-Cox Transformation

Lastly, just for fun, we will test to see if a box-cox transformation is needed. Box-cox transformations are used to transform data that is non-normally distributed into a normal shape. Both the Q-Q plot and the Shapiro-Wilks test implied normality so this test isn't necessary, but it's good pracitce. We will plot the model and test whether the 95% confidence interval contains 1.

```r
# Remove observations where the response variable is 0
fat_fixed <-fat[fat$brozek !=0,]

# Create a new model that is stricly positive and plot
fatmod_fixed <- lm(formula = brozek ~ age + weight + height + neck + chest + abdom + hip, data = fat_fi
boxcox(fatmod_fixed)
```



As expected, Boxcox is not needed as the confidence interval contains 1. The boxcox transformation is designed for strictly positive responses, so that's why we had to remove observations where brozek = 0.

Summary

In the end we found that the data is normally distributed, has constant variance, and has no serial correlation among observations. There were 4 high leverage points, 1 outlier, and 4 influential points. Observations 39 and 42 were both high-leverage points and influential points. Observation 39 was also an outlier.