HW7

Andrew Shao

2024-11-19

1.1

```
# Load the data
data(prostate,package = "faraway")
# Enter your code for fitting the model below
lmod <- lm(lpsa ~ ., prostate)
summary(lmod)
```

```
##
## Call:
## lm(formula = lpsa ~ ., data = prostate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7331 -0.3713 -0.0170  0.4141  1.6381
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.669337   1.296387   0.516  0.60693
## lcavol       0.587022   0.087920   6.677 2.11e-09 ***
## lweight      0.454467   0.170012   2.673  0.00896 **
## age         -0.019637   0.011173  -1.758  0.08229 .
## lbph         0.107054   0.058449   1.832  0.07040 .
## svi          0.766157   0.244309   3.136  0.00233 **
## lcp         -0.105474   0.091013  -1.159  0.24964
## gleason      0.045142   0.157465   0.287  0.77503
## pgg45        0.004525   0.004421   1.024  0.30886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7084 on 88 degrees of freedom
## Multiple R-squared:  0.6548, Adjusted R-squared:  0.6234
## F-statistic: 20.86 on 8 and 88 DF,  p-value: < 2.2e-16
```

```
# Step 1
lmod1 <- update(lmod, . ~ . - gleason)
# summary(lmod1)

# Step 2
lmod2 <- update(lmod1, . ~ . - lcp)
# summary(lmod2)
```

```r
# Step 3
lmod3 <- update(lmod2, . ~ . - pgg45)
# summary(lmod3)

# Step 4
lmod4 <- update(lmod3, . ~ . - age)
# summary(lmod4)

# Step 5
lmod5 <- update(lmod4, . ~ . - lbph)

# Step 6
summary(lmod5)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + svi, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.72964 -0.45764  0.02812  0.46403  1.57013
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.26809    0.54350  -0.493  0.62298
## lcavol       0.55164    0.07467   7.388  6.3e-11 ***
## lweight      0.50854    0.15017   3.386  0.00104 **
## svi          0.66616    0.20978   3.176  0.00203 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7168 on 93 degrees of freedom
## Multiple R-squared:  0.6264, Adjusted R-squared:  0.6144
## F-statistic: 51.99 on 3 and 93 DF,  p-value: < 2.2e-16
```

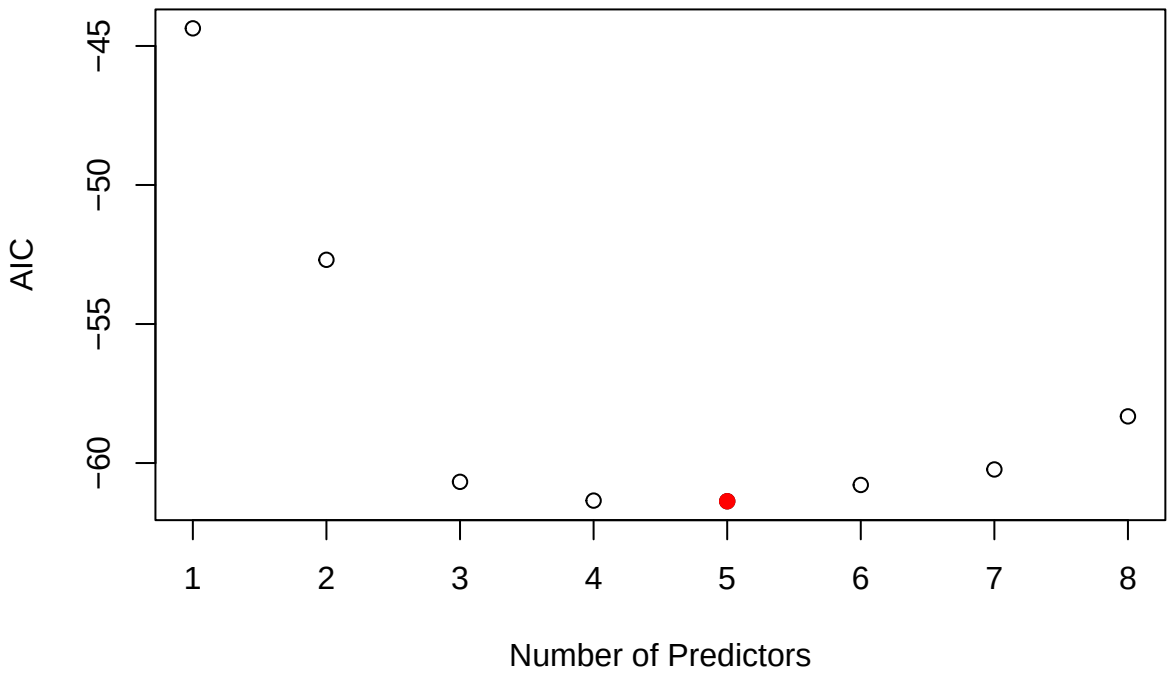
1.2

```r
require(leaps)
```

```
## Loading required package: leaps
```

```
## Warning: package 'leaps' was built under R version 4.3.3
```

```r
prostate.leaps <- regsubsets(lpsa ~ ., data= prostate)
# summary(prostate.leaps)

rs <- summary(prostate.leaps)
# rs$which

n <- nrow(prostate)
AIC <- n*log(rs$rss/n) + (2:9)*2
best_index <- which.min(AIC)
plot(AIC ~ I(1:8), ylab="AIC", xlab="Number of Predictors")
points(best_index, AIC[best_index], col = "red", pch = 19)
```




```
prostate.models <- summary(prostate.leaps)$which;
prostate.models.size <- as.numeric(attr(prostate.models, "dimnames")[[1]]);
prostate.models.rss <- summary(prostate.leaps)$rss;

op <- which(prostate.models.size == best_index)
flag <- op[which.min(prostate.models.rss[op])]

paste('The predictors I selected are:',
↪  paste(names(prostate.models[flag,])[prostate.models[flag,]][-1], collapse = ', '))
```

```
## [1] "The predictors I selected are: lcavol, lweight, age, lbph, svi"
```

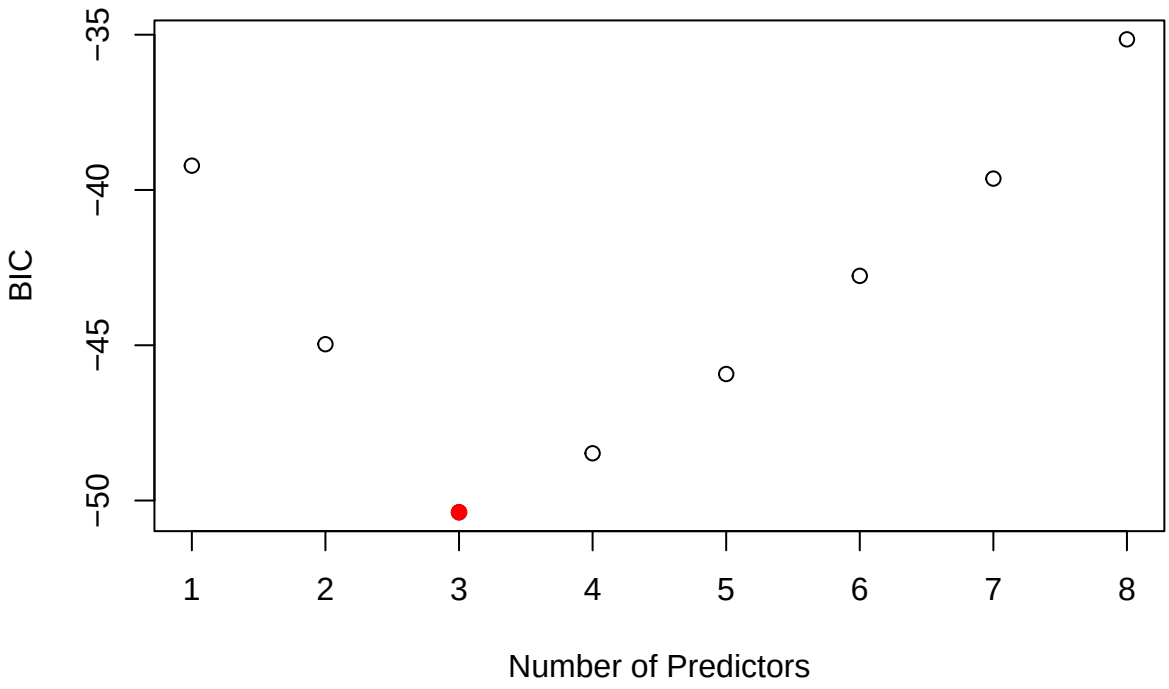
```
best_model <- lm(lpsa ~ lcavol + lweight + age + lbph + svi, data = prostate)
summary(best_model)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.83505 -0.39396  0.00414  0.46336  1.57888
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.95100    0.83175   1.143 0.255882
## lcavol       0.56561    0.07459   7.583 2.77e-11 ***
```

```
## lweight      0.42369    0.16687   2.539 0.012814 *
## age         -0.01489    0.01075  -1.385 0.169528
## lbph         0.11184    0.05805   1.927 0.057160 .
## svi          0.72095    0.20902   3.449 0.000854 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7073 on 91 degrees of freedom
## Multiple R-squared:  0.6441, Adjusted R-squared:  0.6245
## F-statistic: 32.94 on 5 and 91 DF,  p-value: < 2.2e-16
```

1.3

```
require(leaps)

prostate.leaps <- regsubsets(lpsa ~ ., data= prostate)
# summary(prostate.leaps)

rs <- summary(prostate.leaps)
# rs$which

n <- nrow(prostate)
BIC <- n*log(rs$rss/n) + (2:9)*log(n)
best_index <- which.min(BIC)
plot(BIC ~ I(1:8), ylab="BIC", xlab="Number of Predictors")
points(best_index, BIC[best_index], col = "red", pch = 19)
```




```
prostate.models <- summary(prostate.leaps)$which;
prostate.models.size <- as.numeric(attr(prostate.models, "dimnames")[[1]]);
prostate.models.rss <- summary(prostate.leaps)$rss;

op <- which(prostate.models.size == best_index)
flag <- op[which.min(prostate.models.rss[op])]

paste('The predictors I selected are:',
↪  paste(names(prostate.models[flag,])[prostate.models[flag,]][-1], collapse = ', '))
```

```
## [1] "The predictors I selected are: lcavol, lweight, svi"
```

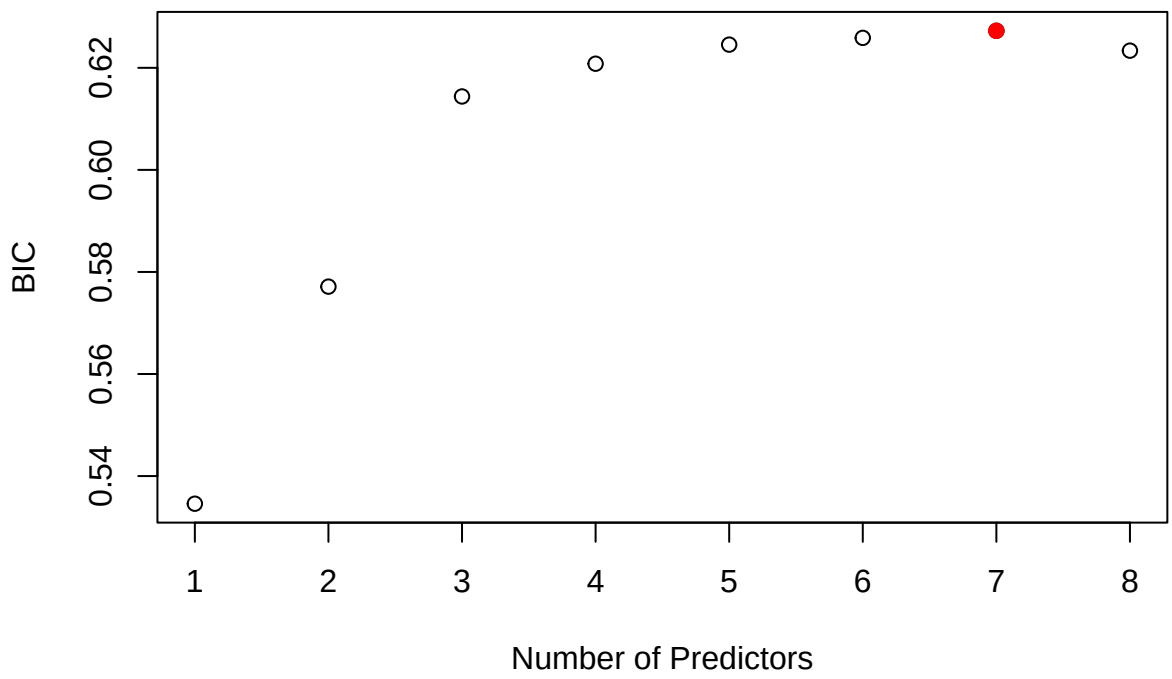
```
best_model <- lm(lpsa ~ lcavol + lweight + svi, data = prostate)
summary(best_model)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + svi, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.72964 -0.45764  0.02812  0.46403  1.57013
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.26809    0.54350  -0.493  0.62298
## lcavol       0.55164    0.07467   7.388  6.3e-11 ***
```

```
## lweight      0.50854    0.15017   3.386  0.00104 **
## svi          0.66616    0.20978   3.176  0.00203 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7168 on 93 degrees of freedom
## Multiple R-squared:  0.6264, Adjusted R-squared:  0.6144
## F-statistic: 51.99 on 3 and 93 DF,  p-value: < 2.2e-16
```

1.4

```
require(leaps)

prostate.leaps <- regsubsets(lpsa ~ ., data= prostate)
# summary(prostate.leaps)

rs <- summary(prostate.leaps)
# rs$which

# n <- nrow(prostate)
best_index <- which.max(rs$adjr2)
plot(rs$adjr2 ~ I(1:8), ylab="BIC", xlab="Number of Predictors")
points(best_index, rs$adjr2[best_index], col = "red", pch = 19)
```




```
prostate.models <- summary(prostate.leaps)$which;
prostate.models.size <- as.numeric(attr(prostate.models, "dimnames")[[1]]);
```

```
prostate.models.rss <- summary(prostate.leaps)$rss;

op <- which(prostate.models.size == best_index)
flag <- op[which.min(prostate.models.rss[op])]

paste('The predictors I selected are:',
↪  paste(names(prostate.models[flag,])[prostate.models[flag,]][-1], collapse = ', '))
```

```
## [1] "The predictors I selected are: lcavol, lweight, age, lbph, svi, lcp, pgg45"
```

```
best_model <- lm(lpsa ~ lcavol + lweight + age + lbph + svi + lcp + pgg45, data =
↪  prostate)
summary(best_model)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi + lcp +
##     pgg45, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73117 -0.38137 -0.01728  0.43364  1.63513
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.953926   0.829439   1.150  0.25319
## lcavol       0.591615   0.086001   6.879 8.07e-10 ***
## lweight      0.448292   0.167771   2.672  0.00897 **
## age         -0.019336   0.011066  -1.747  0.08402 .
## lbph         0.107671   0.058108   1.853  0.06720 .
## svi          0.757734   0.241282   3.140  0.00229 **
## lcp         -0.104482   0.090478  -1.155  0.25127
## pgg45        0.005318   0.003433   1.549  0.12488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7048 on 89 degrees of freedom
## Multiple R-squared:  0.6544, Adjusted R-squared:  0.6273
## F-statistic: 24.08 on 7 and 89 DF,  p-value: < 2.2e-16
```

2.1

```
rm (list = ls())
animalsDf <- read.csv("AnimalsStat.csv")
head(animalsDf)
```

```
##              Name     Body Brain
## 1 Mountain beaver     1.35   8.1
## 2             Cow   465.00 423.0
## 3       Grey wolf    36.33 119.5
## 4            Goat    27.66 115.0
## 5      Guinea pig     1.04   5.5
## 6     Dipliodocus 11700.00  50.0
```

2.2

Body summary statistics

```
summary(animalsDf$Body)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.02     3.10    53.83  4278.44   479.00 87000.00
```

Brain summary statistics

```
summary(animalsDf$Brain)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.40   22.23  137.00  574.52  420.00 5712.00
```

2.3

Which animal has the smallest body mass in the sample?

```
animalsDf[animalsDf$Body == min(animalsDf$Body),]
```

```
##     Name  Body Brain
## 20 Mouse 0.023   0.4
```

Which animal has the largest body mass in the sample?

```
animalsDf[animalsDf$Body == max(animalsDf$Body),]
```

```
##             Name  Body Brain
## 26 Brachiosaurus 87000 154.5
```

Which animal has the smallest brain mass in the sample?

```
animalsDf[animalsDf$Brain == min(animalsDf$Brain),]
```

```
##     Name  Body Brain
## 20 Mouse 0.023   0.4
```

Which animal has the largest brain mass in the sample?

```
animalsDf[animalsDf$Brain == max(animalsDf$Brain),]
```

```
##                Name Body Brain
## 15 African elephant 6654  5712
```

2.4

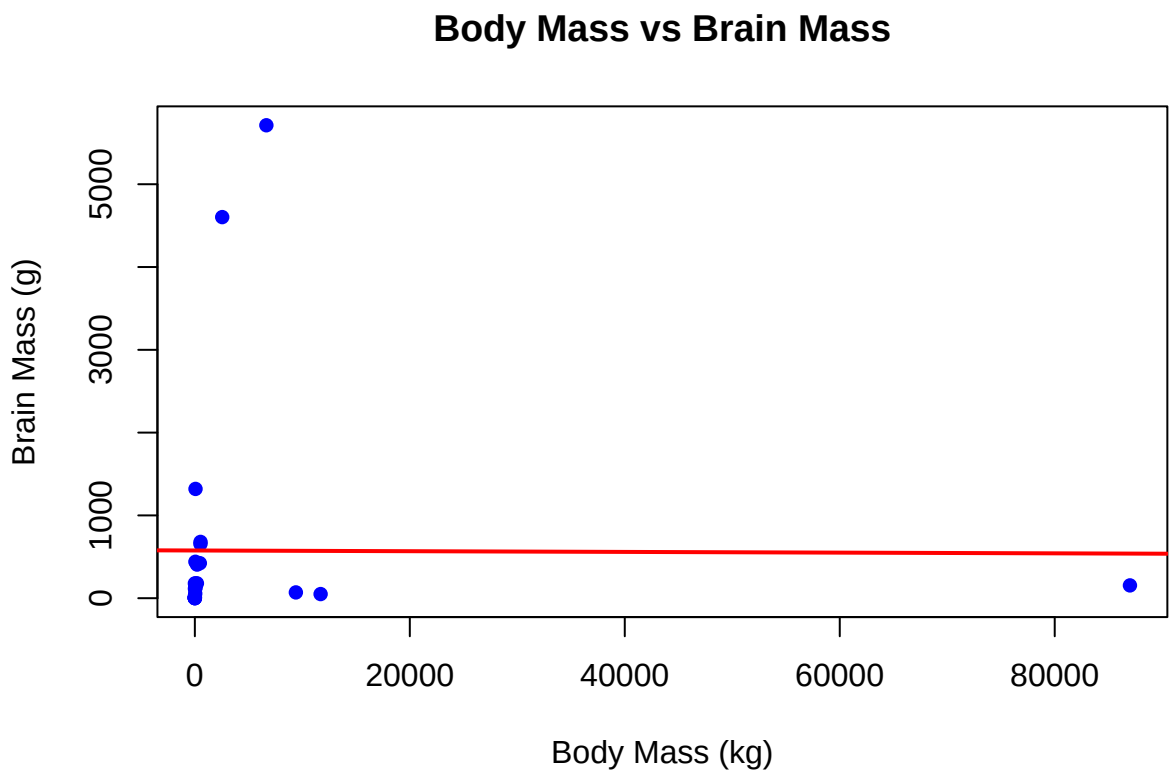
```
animalsDf$brain_to_body_ratio <- animalsDf$Brain / animalsDf$Body
animalsDf[which.max(animalsDf$brain_to_body_ratio), ]
```

```
##             Name Body Brain brain_to_body_ratio
## 17 Rhesus monkey  6.8   179            26.32353
```

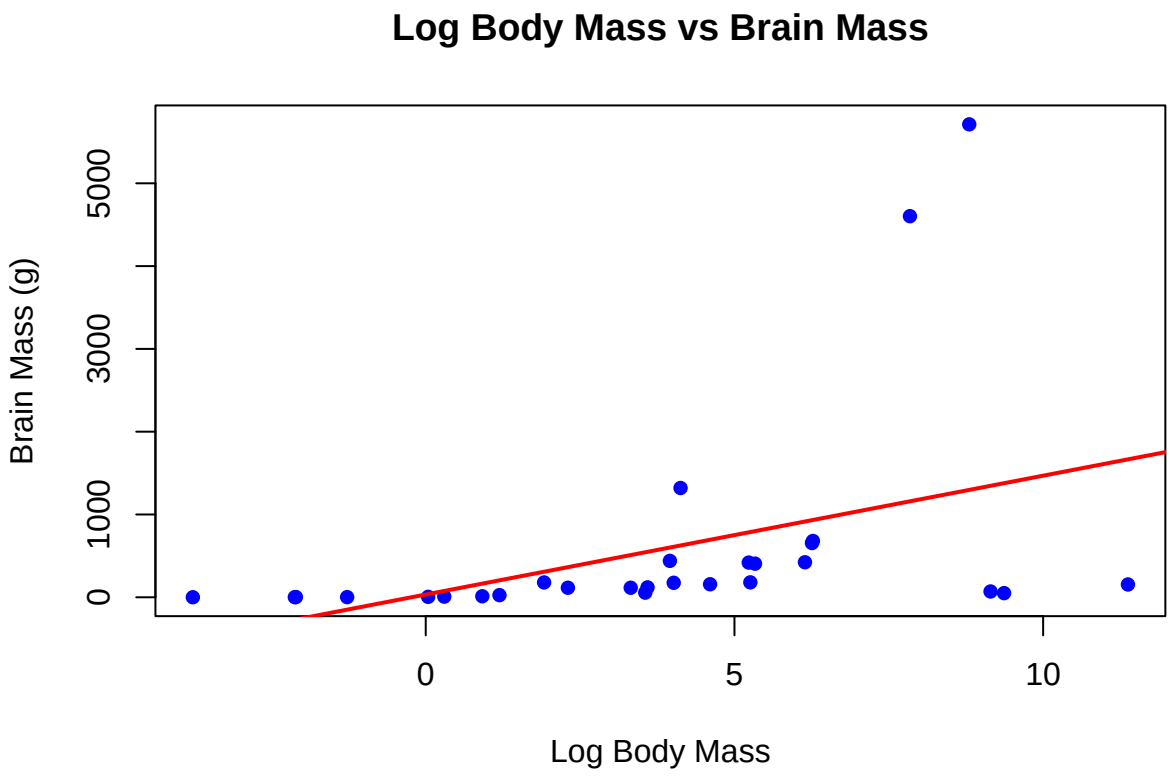
2.5

```
plot(animalsDf$Body, animalsDf$Brain,
     xlab = "Body Mass (kg)", ylab = "Brain Mass (g)",
     main = "Body Mass vs Brain Mass",
```

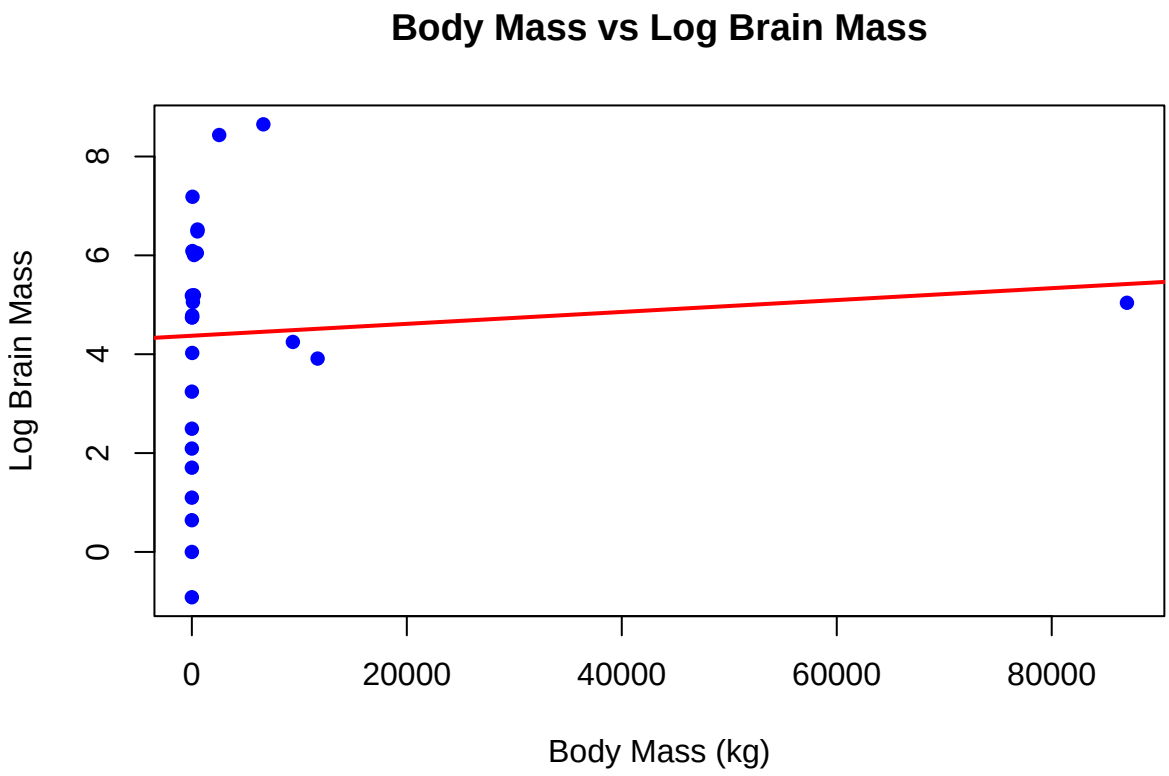
```
     pch = 16, col = "blue")
abline(lm(Brain ~ Body, data = animalsDf), col = "red", lwd = 2)
```



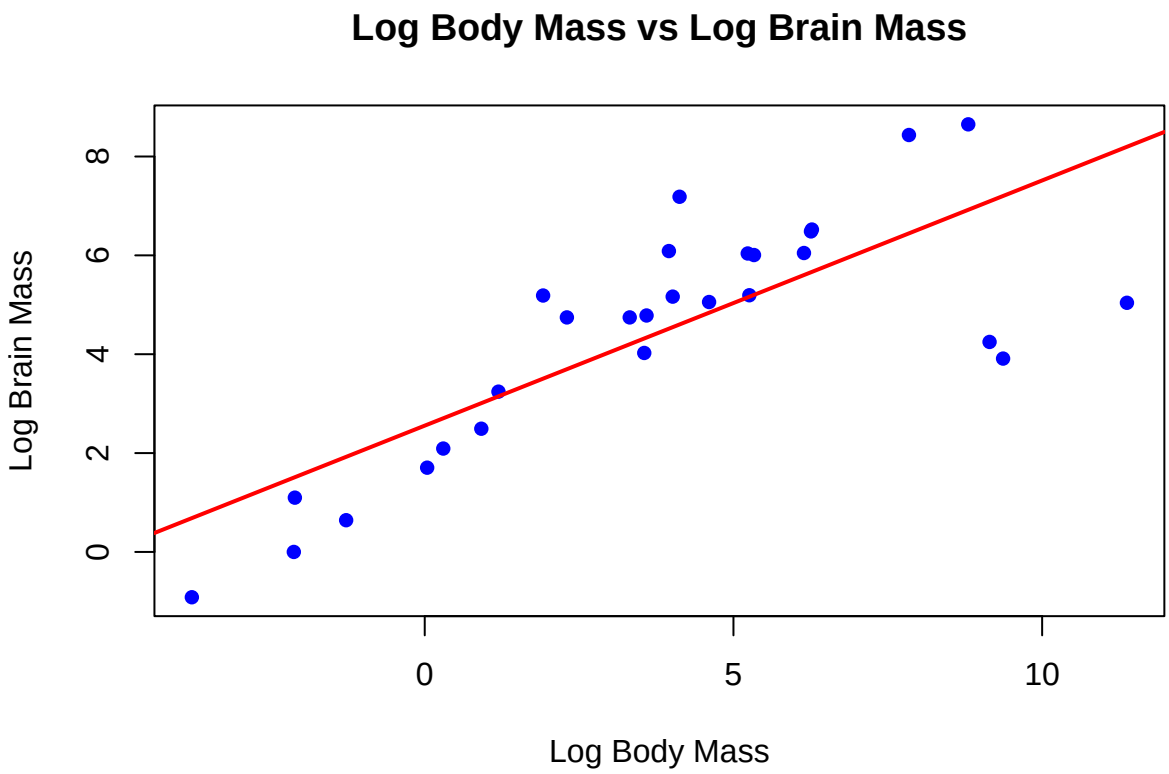

```
plot(log(animalsDf$Body), animalsDf$Brain,
     xlab = "Log Body Mass", ylab = "Brain Mass (g)",
     main = "Log Body Mass vs Brain Mass",
     pch = 16, col = "blue")
abline(lm(Brain ~ log(Body), data = animalsDf), col = "red", lwd = 2)
```




```r
plot(animalsDf$Body, log(animalsDf$Brain),
     xlab = "Body Mass (kg)", ylab = "Log Brain Mass",
     main = "Body Mass vs Log Brain Mass",
     pch = 16, col = "blue")
abline(lm(log(Brain) ~ Body, data = animalsDf), col = "red", lwd = 2)
```




```
plot(log(animalsDf$Body), log(animalsDf$Brain),
     xlab = "Log Body Mass", ylab = "Log Brain Mass",
     main = "Log Body Mass vs Log Brain Mass",
     pch = 16, col = "blue")
abline(lm(log(Brain) ~ log(Body), data = animalsDf), col = "red", lwd = 2)
```



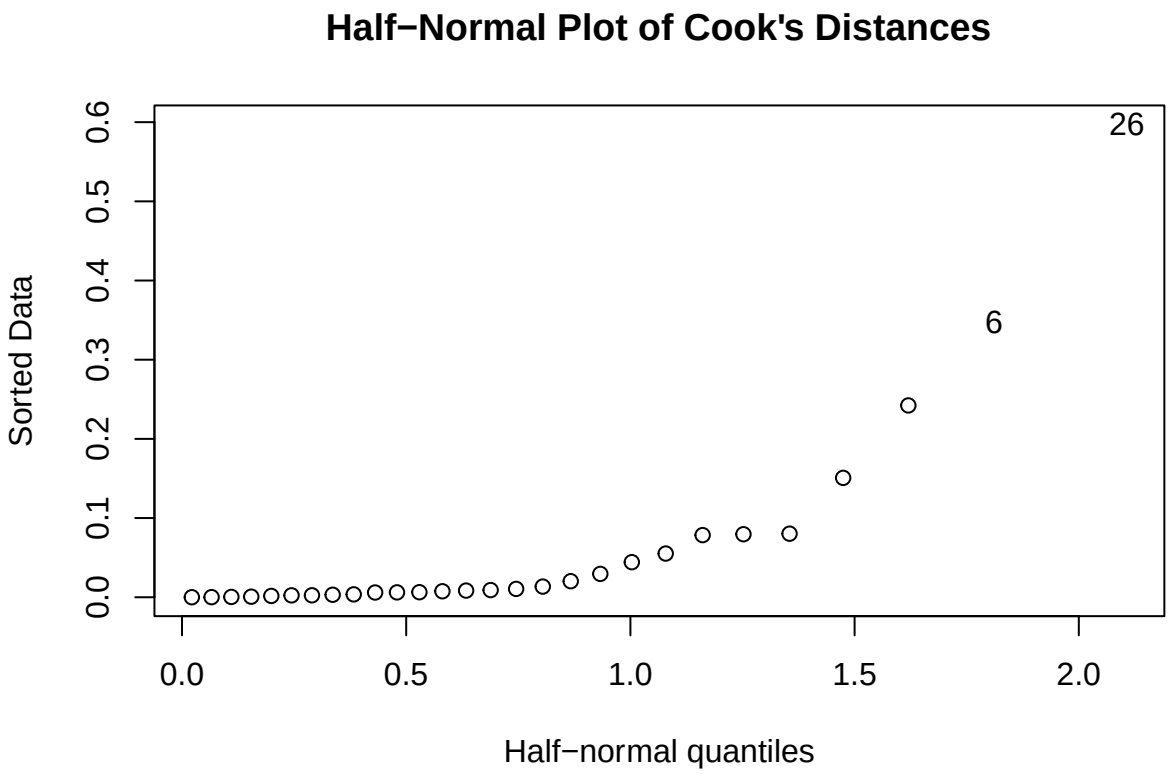
Log body mass and log brain mass model looks the best.

2.6

```
library(faraway)
```

```
## Warning: package 'faraway' was built under R version 4.3.3
```

```
best_model <- lm(log(Brain) ~ log(Body), data = animalsDf)

cooks_distances <- cooks.distance(best_model)

halfnorm(cooks_distances, labs = rownames(animalsDf),
         main = "Half-Normal Plot of Cook's Distances")
```




```
largest_cooks <- sort(cooks_distances, decreasing = T)[1:3]

animalsDf[names(largest_cooks), ]
```

```
##             Name  Body Brain brain_to_body_ratio
## 26 Brachiosaurus 87000 154.5         0.001775862
## 6    Dipliodocus 11700  50.0         0.004273504
## 16   Triceratops  9400  70.0         0.007446809
```

The three largest are Brachiosaurus, Dipliodocus, and Triceratops.

2.7

Original model

```
original_model <- lm(log(Brain) ~ log(Body), data = animalsDf)
summary(original_model)
```

```
##
## Call:
## lm(formula = log(Brain) ~ log(Body), data = animalsDf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2890 -0.6763  0.3316  0.8646  2.5835
##
## Coefficients:
```

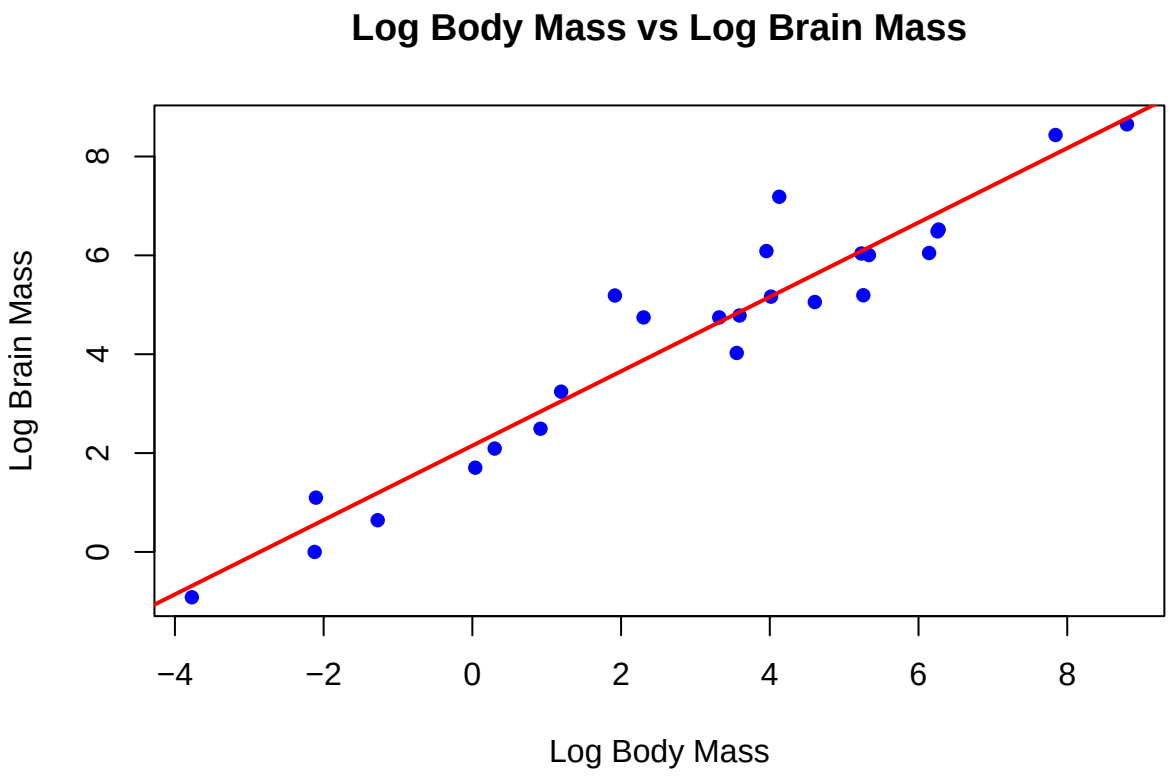
```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.55490    0.41314   6.184 1.53e-06 ***
## log(Body)    0.49599    0.07817   6.345 1.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.532 on 26 degrees of freedom
## Multiple R-squared:  0.6076, Adjusted R-squared:  0.5925
## F-statistic: 40.26 on 1 and 26 DF,  p-value: 1.017e-06
```

New model

```r
animalsDf_removed <- animalsDf[-1 * as.numeric(names(largest_cooks)), ]
new_model <- lm(log(Brain) ~ log(Body), data = animalsDf_removed)
summary(new_model)
```

```
##
## Call:
## lm(formula = log(Brain) ~ log(Body), data = animalsDf_removed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9125 -0.4752 -0.1557  0.1940  1.9303
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.15041    0.20060   10.72 2.03e-10 ***
## log(Body)    0.75226    0.04572   16.45 3.24e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7258 on 23 degrees of freedom
## Multiple R-squared:  0.9217, Adjusted R-squared:  0.9183
## F-statistic: 270.7 on 1 and 23 DF,  p-value: 3.243e-14
```

```r
plot(log(animalsDf_removed$Body), log(animalsDf_removed$Brain),
     xlab = "Log Body Mass", ylab = "Log Brain Mass",
     main = "Log Body Mass vs Log Brain Mass",
     pch = 16, col = "blue")
abline(new_model, col = "red", lwd = 2)
```



Original R-squared: 0.6076
New R-squared: 0.7258

3.1

```
rm (list = ls())
require(MASS)
```

```
## Loading required package: MASS
```

```
data(ozone, package = "faraway")

model_31 <- lm(O3 ~ temp + humidity + ibh + temp * humidity, data = ozone)
summary(model_31)
```

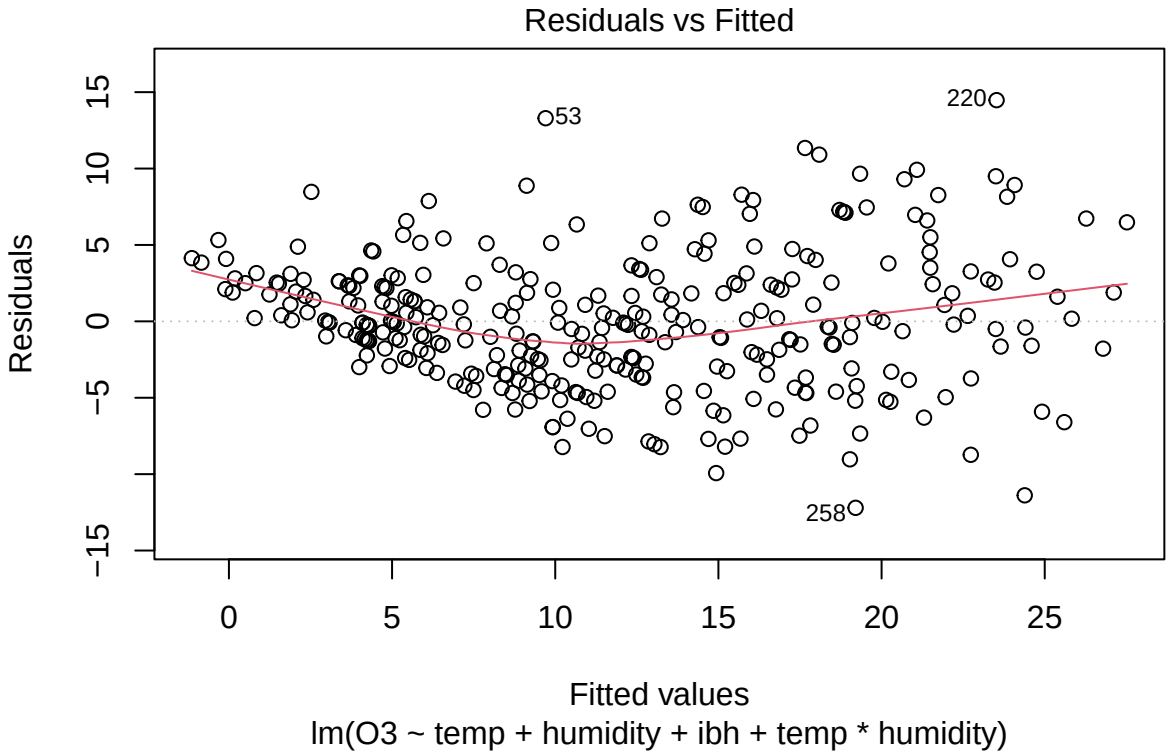
```
##
## Call:
## lm(formula = O3 ~ temp + humidity + ibh + temp * humidity, data = ozone)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.204  -2.890  -0.176   2.508  14.476
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   10.9318952  4.0129533   2.724   0.0068 **
## temp          -0.0479114  0.0683146  -0.701   0.4836
```

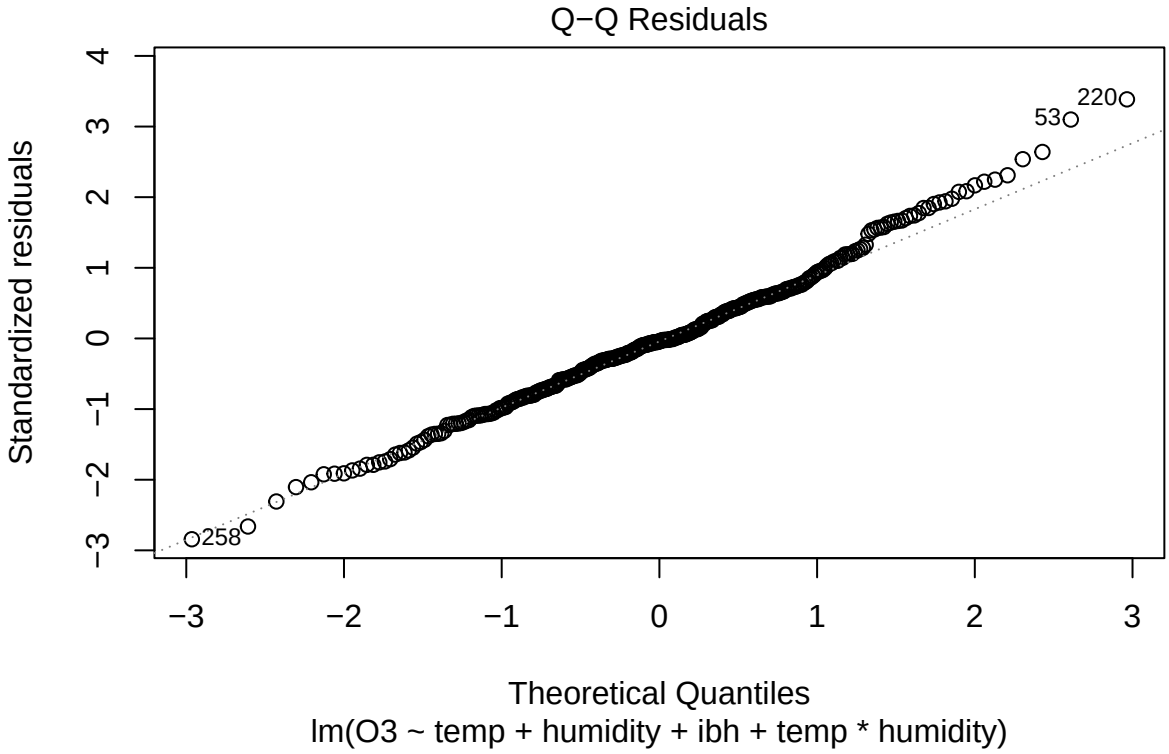
```
## humidity      -0.2741679  0.0621176  -4.414 1.38e-05 ***
## ibh           -0.0010115  0.0001563  -6.472 3.56e-10 ***
## temp:humidity  0.0060593  0.0010478   5.783 1.72e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.315 on 325 degrees of freedom
## Multiple R-squared:  0.7135, Adjusted R-squared:  0.7099
## F-statistic: 202.3 on 4 and 325 DF,  p-value: < 2.2e-16
```

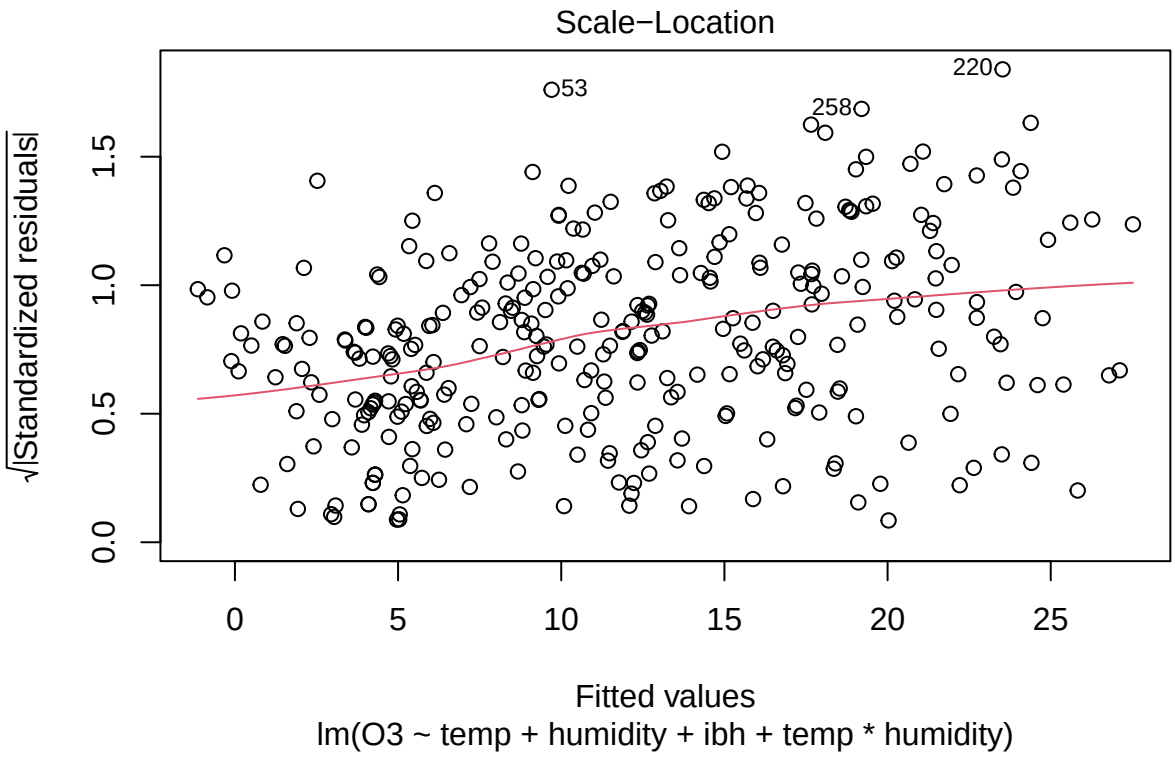
The interaction coefficient is significant. `temp` variable isn't significant. We shouldn't remove `temp` because removing it would affect the interpretation of the model and interaction term.

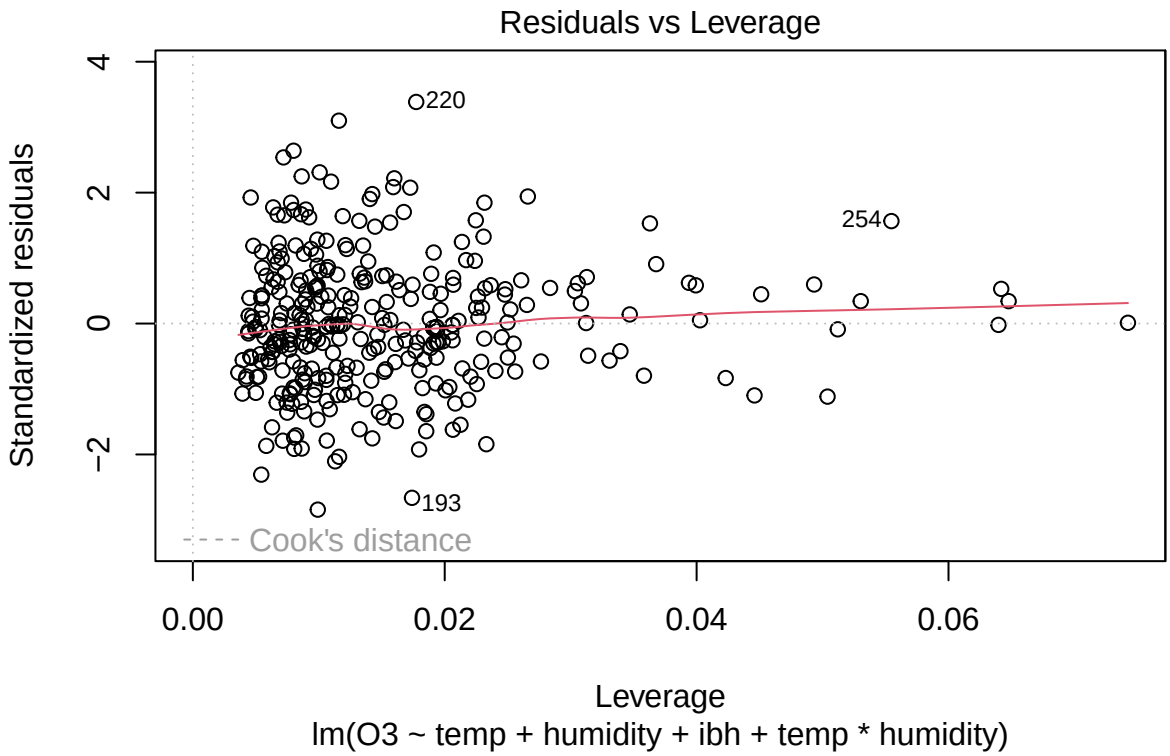
3.2

```
plot(model_31)
```




Q−Q Residuals
Standardized residuals
−3
−2
−1
0
1
2
3
4
220
53
258
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(O3 ~ temp + humidity + ibh + temp * humidity)

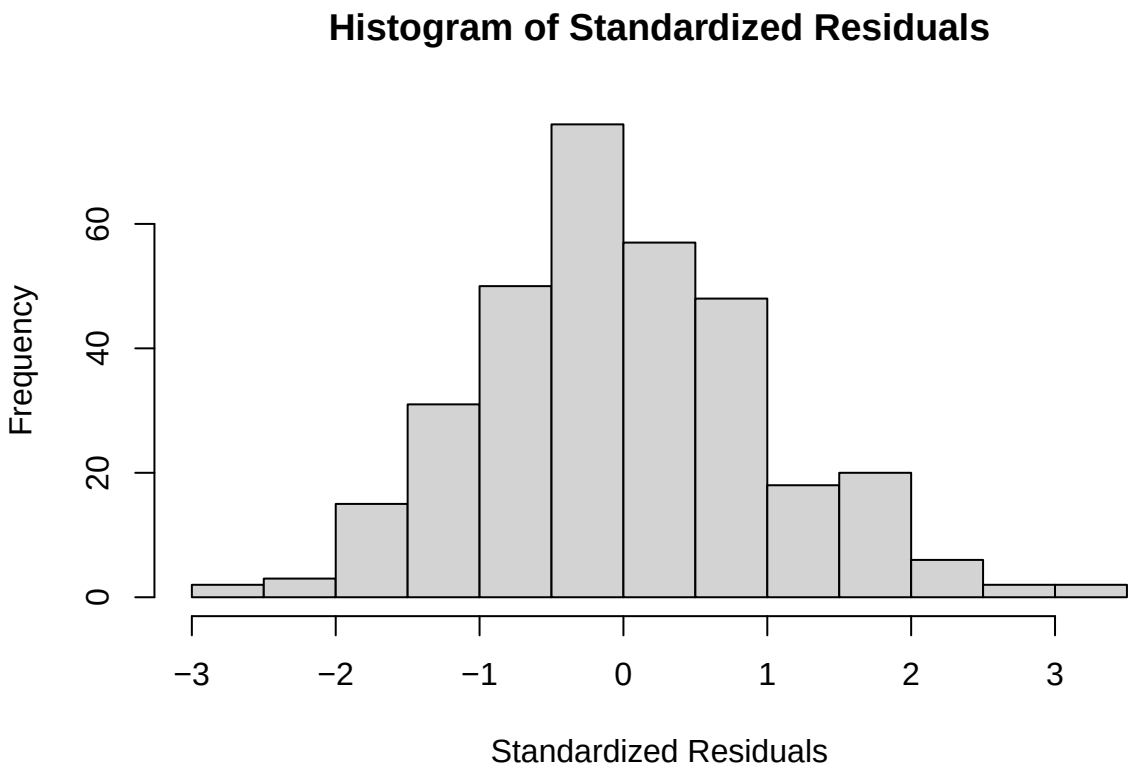

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
53
258
220
0
5
10
15
20
25
Fitted values
lm(O3 ~ temp + humidity + ibh + temp * humidity)



Spread of the residuals increases as fitted values increases, which indicates heteroscedascity, which is an issue as it suggests the constant variance assumption is violated. The curve in the residuals also suggests a non-linear relationship.

3.3

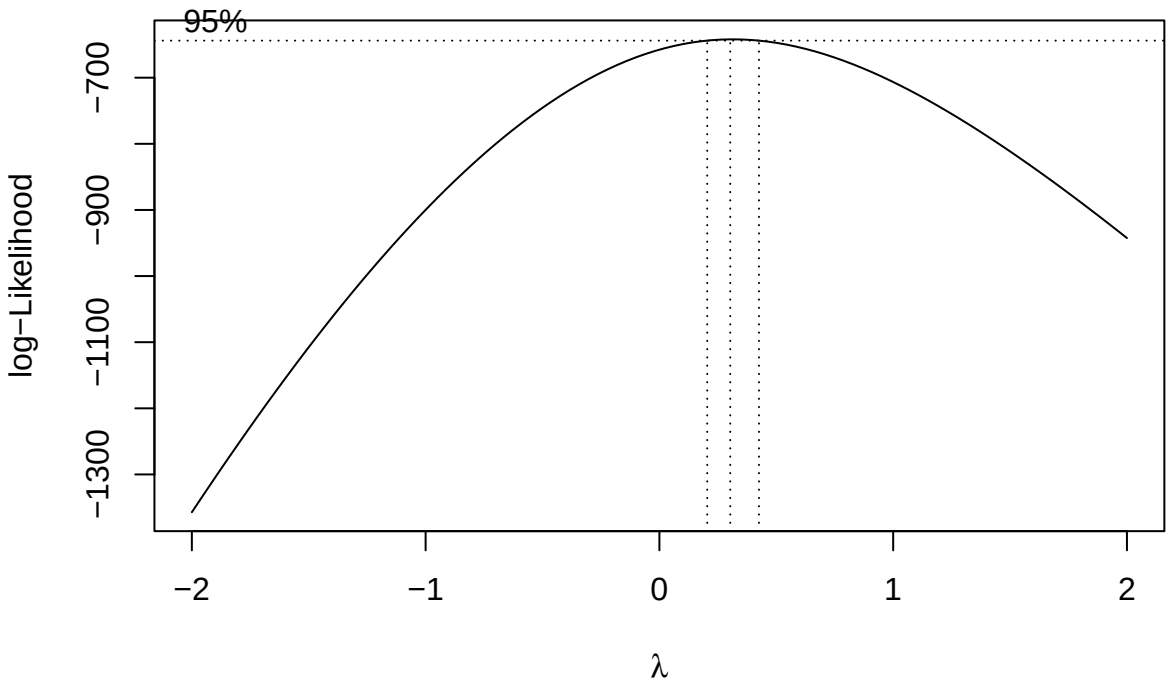
```
hist(rstandard(model_31),
     main = "Histogram of Standardized Residuals",
     xlab = "Standardized Residuals")
```



See above in 3.2 for the Q-Q plot. The residuals in the Q-Q plot lie close to the diagonal and the histogram looks roughly normal, which suggests that the normality assumption holds.

3.4

```
boxcox_result <- boxcox(model_31, lambda = seq(-2, 2, by = 0.1))
```




```
boxcox_result$x[which.max(boxcox_result$y)]
```

```
## [1] 0.3030303
```

The exponent I found was 0.303.

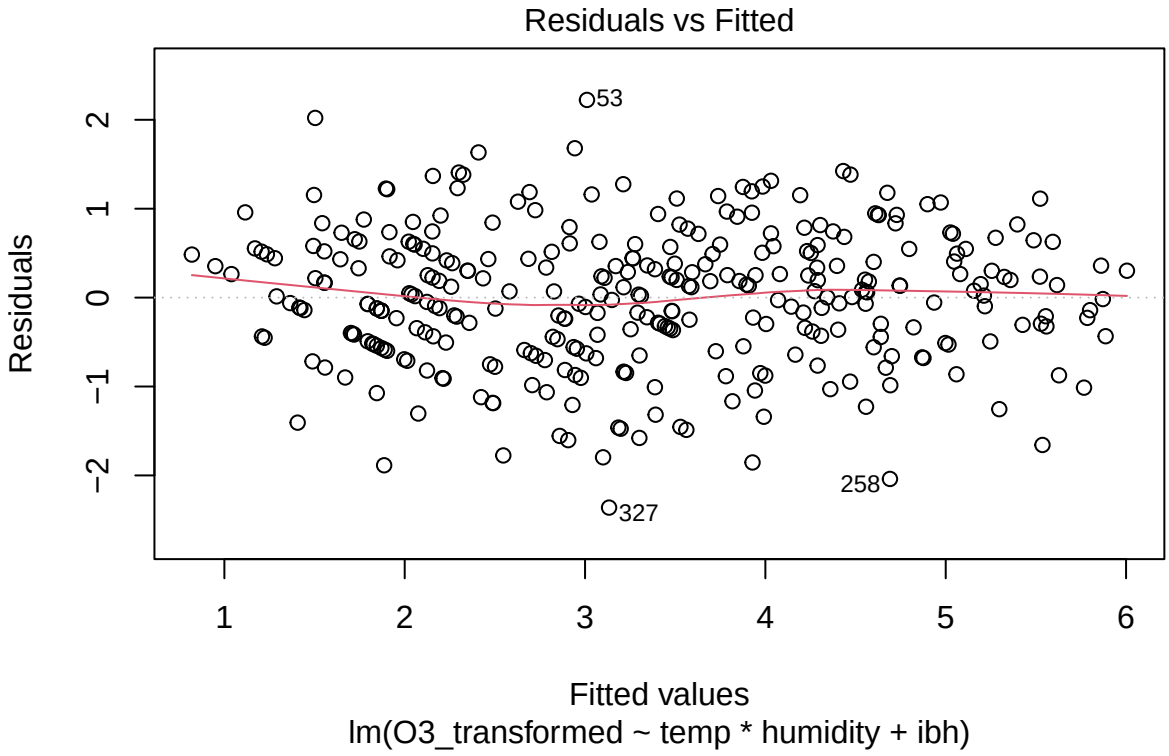
3.5

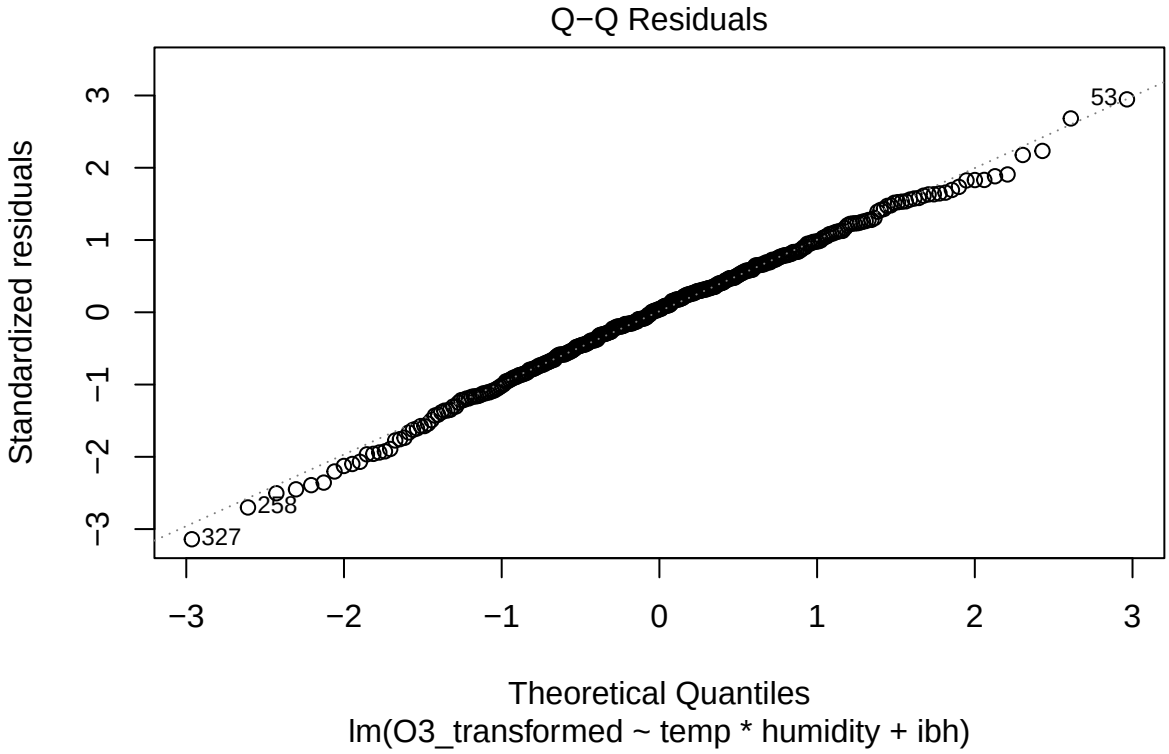
```
lambda <- boxcox_result$x[which.max(boxcox_result$y)]
ozone$O3_transformed <- (ozone$O3^lambda - 1) / lambda

new_model <- lm(O3_transformed ~ temp * humidity + ibh, data = ozone)
summary(new_model)
```

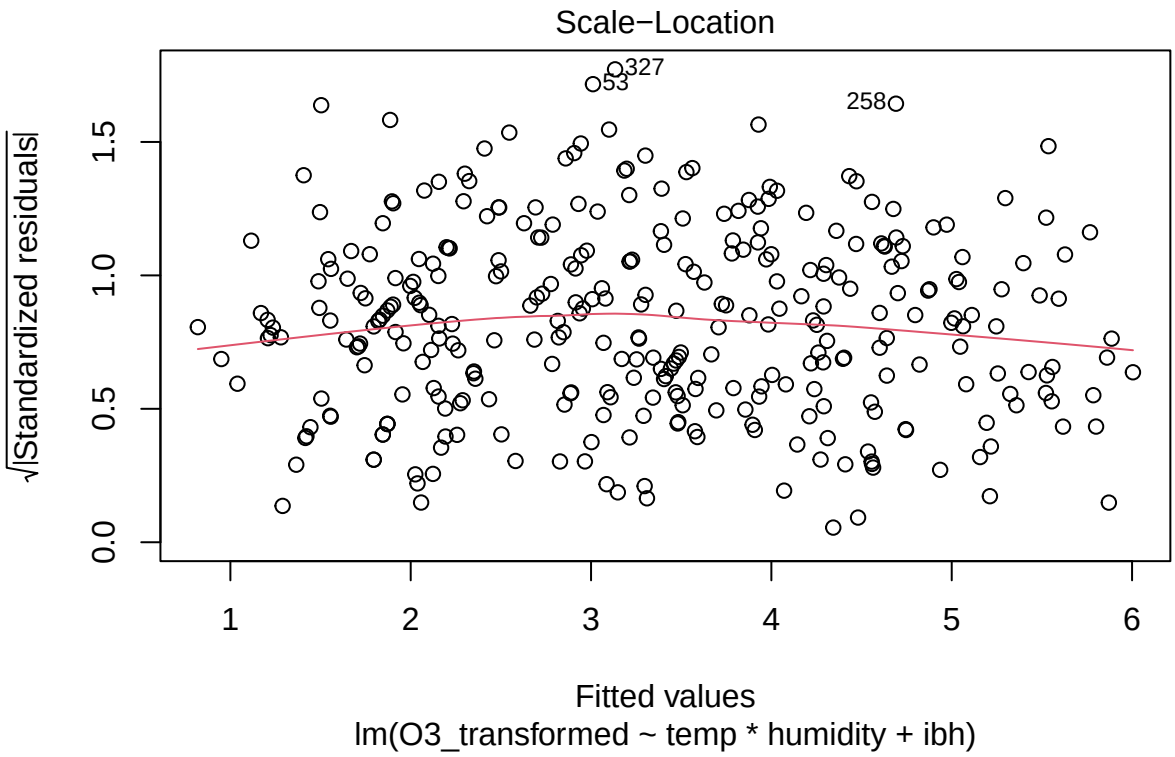
```
##
## Call:
## lm(formula = O3_transformed ~ temp * humidity + ibh, data = ozone)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.36193 -0.49271  0.03453  0.51233  2.22343
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.198e+00  7.059e-01   3.113  0.00201 **
## temp           1.041e-02  1.202e-02   0.866  0.38693
```

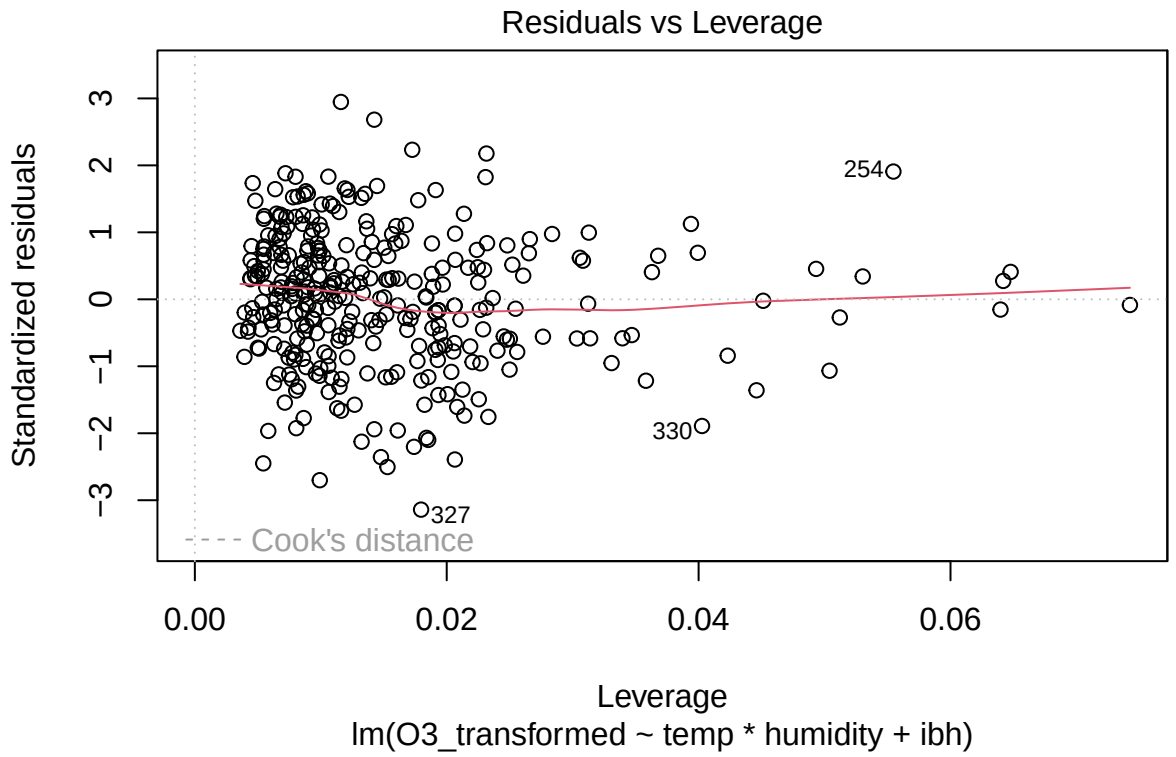
```
## humidity      -3.179e-02  1.093e-02  -2.910  0.00387 **
## ibh           -2.175e-04  2.749e-05  -7.912 4.02e-14 ***
## temp:humidity  7.820e-04  1.843e-04   4.243 2.88e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7589 on 325 degrees of freedom
## Multiple R-squared:  0.7317, Adjusted R-squared:  0.7284
## F-statistic: 221.6 on 4 and 325 DF,  p-value: < 2.2e-16
```

```
plot(new_model)
```

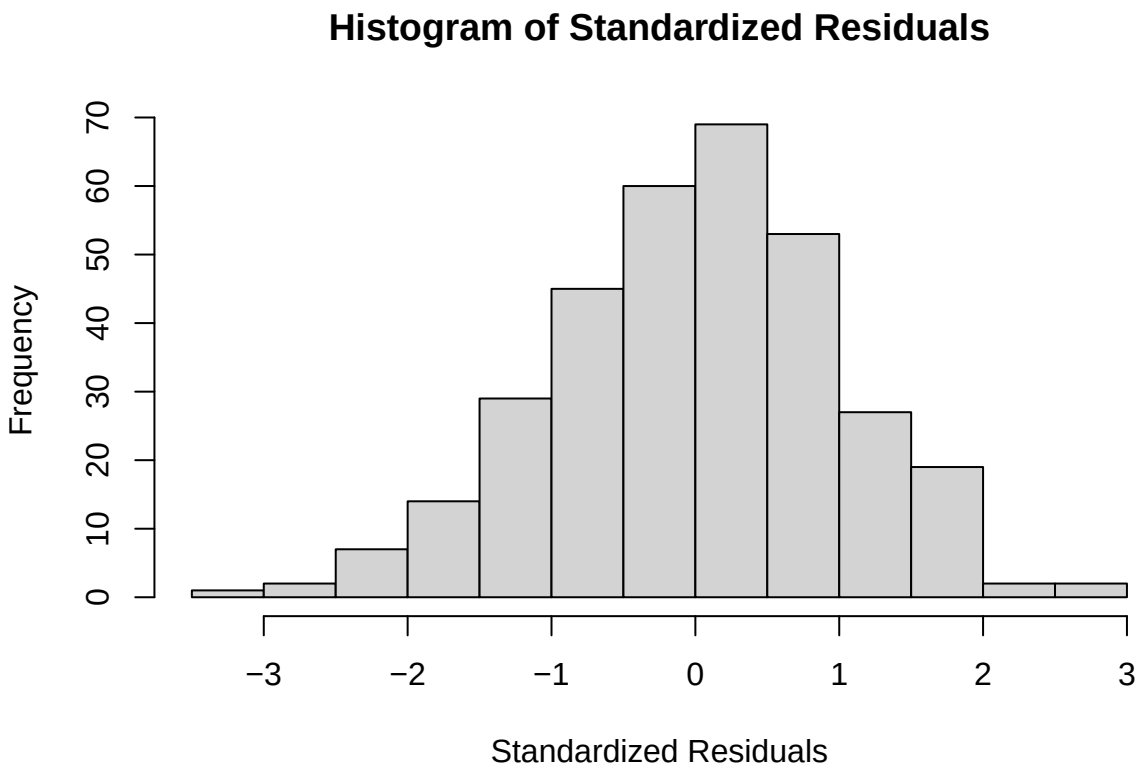



Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(O3_transformed ~ temp * humidity + ibh)
53
258
327
−3
−2
−1
0
1
2
3


Scale−Location
327
53
258
√|Standardized residuals|
0.0
0.5
1.0
1.5
1
2
3
4
5
6
Fitted values
lm(O3_transformed ~ temp * humidity + ibh)




```r
hist(rstandard(new_model),
     main = "Histogram of Standardized Residuals",
     xlab = "Standardized Residuals")
```



The spread is now roughly evenly spread across fitted values with less curvature, suggesting heteroscedascity has been reduced. The Q-Q plot residuals are closer to the diagonal line, especially in the tails, and the histogram appears more normal suggesting better alignment with the normality assumption. Outliers may be present, however.