Lab05

John Kaspers

June 8, 2020

Warning

Warning: If you get an error when running the code, make sure that you have the necessary packages installed. You can install them by running the following code in base R (not RStudio or RMarkdown).

> install.packages('alr4')
> install.packages('faraway')
> install.packages('cvTools')

Big Mac 2003

To begin, we'll use the `BigMac2003` data set from the `alr4` package. Further information on the data set can be found here.

1. Create two models that predict `TaxRate`. The first model will use `log(Bread)` and `log(Rice)` as the predictors. The second model will use `log(Bread)`, `log(Rice)`, and `Apt` as predictors. Do not include any interactions of other transformations. Run each of your models through the `summary()` function.

```
data(BigMac2003, package = "alr4")

lm.model1 <- lm(TaxRate ~ log(Bread) + log(Rice), data = BigMac2003)
summary(lm.model1)
```

```
##
## Call:
## lm(formula = TaxRate ~ log(Bread) + log(Rice), data = BigMac2003)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.9987  -5.4434  -0.9278   5.4372  23.6722
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   47.478      5.998   7.916 3.74e-11 ***
## log(Bread)    -6.622      2.133  -3.105  0.00281 **
## log(Rice)     -2.203      2.405  -0.916  0.36295
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.071 on 66 degrees of freedom
## Multiple R-squared:  0.2468, Adjusted R-squared:  0.224
## F-statistic: 10.81 on 2 and 66 DF,  p-value: 8.679e-05
```

```
summary(lm.model1)$adj.r.squared
```

```
## [1] 0.2239514
```

```
lm.model2 <- lm(TaxRate ~ log(Bread) + log(Rice) + Apt, data = BigMac2003)
summary(lm.model2)
```

```
##
## Call:
## lm(formula = TaxRate ~ log(Bread) + log(Rice) + Apt, data = BigMac2003)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.855  -5.410  -0.652   5.711  23.878
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.0120480  8.6360219   5.328 1.33e-06 ***
## log(Bread)  -6.4669762  2.2460609  -2.879  0.00539 **
## log(Rice)   -2.0225567  2.5382125  -0.797  0.42844
## Apt          0.0006923  0.0029141   0.238  0.81296
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.136 on 65 degrees of freedom
## Multiple R-squared:  0.2474, Adjusted R-squared:  0.2127
## F-statistic: 7.124 on 3 and 65 DF,  p-value: 0.0003284
```

```
summary(lm.model2)$adj.r.squared
```

```
## [1] 0.2126958
```

2. Which model is preferred according to adjusted R-squared?

Model 1 is preferred according to adjusted R-squared because model1 has an adjusted R-squared of 0.224 compared to 0.2127 from model 2.

3. Use the `AIC()` function to calculate the AIC for each model.

```
AIC(lm.model1)
```

```
## [1] 505.0436
```

```
AIC(lm.model2)
```

```
## [1] 506.9838
```

4. Which model is preferred according to AIC?

According to AIC, model 1 is preferred because it has a lower AIC (505.0436 compared to 506.9838 in model 2). In other words, the model without "Apt" is prefered because it has a lower AIC.

5. Use the function `logLik()` to calculate the log-likelihood of each model.

```
logLik(lm.model1)
```

```
## 'log Lik.' -248.5218 (df=4)
```

```
logLik(lm.model2)
```

```
## 'log Lik.' -248.4919 (df=5)
```

6. Which model is preferred according to the log-likelihood?

Because model 1 has a slightly lower log-likelihood, model 1 i slightly preferred over model 2.

7. Do your answers to Question 4 and 6 agree with each other? Briefly explain why or why not.

Yes my answers to questions 4 and 6 agree with each other. This said, the answers need not agree.

8. Run a likelihood ratio test between the two models.

```
anova(lm.model1, lm.model2, test = "LRT")
```

```
## Analysis of Variance Table
##
## Model 1: TaxRate ~ log(Bread) + log(Rice)
## Model 2: TaxRate ~ log(Bread) + log(Rice) + Apt
##   Res.Df    RSS Df Sum of Sq Pr(>Chi)
## 1     66 5430.4
## 2     65 5425.6  1    4.7109   0.8122
```

9. Which model is preferred according to the LRT?

The results are not statistically significant due to a very large p-value. This means there is not a statistically significant improvement when adding the "Apt" predictor. Thus the model without "Apt" is prefered.

10. Do your answers to Question 4 and 9 agree with each other? Yes or no.

Yes, they do.

Infant Mortality

Now we'll use the `infmort` data set from the `faraway` package. Further information on the data set can be found here.

1. Create two models using this data set. The first model will use `income` to predict `mortality`. The second model will use `log(income)` to predict `log(mortality)`. Use the `AIC()` function to calculate the AIC for each model.

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

lm.model.infant1 <- lm(mortality~income, data = infmort)


lm.model.infant2 <- lm(log(mortality)~log(income), data = infmort)

AIC(lm.model.infant1)
```

```
## [1] 1190.687
```

```
AIC(lm.model.infant2)
```

```
## [1] 214.6845
```

2. Which model is preferred according to AIC?

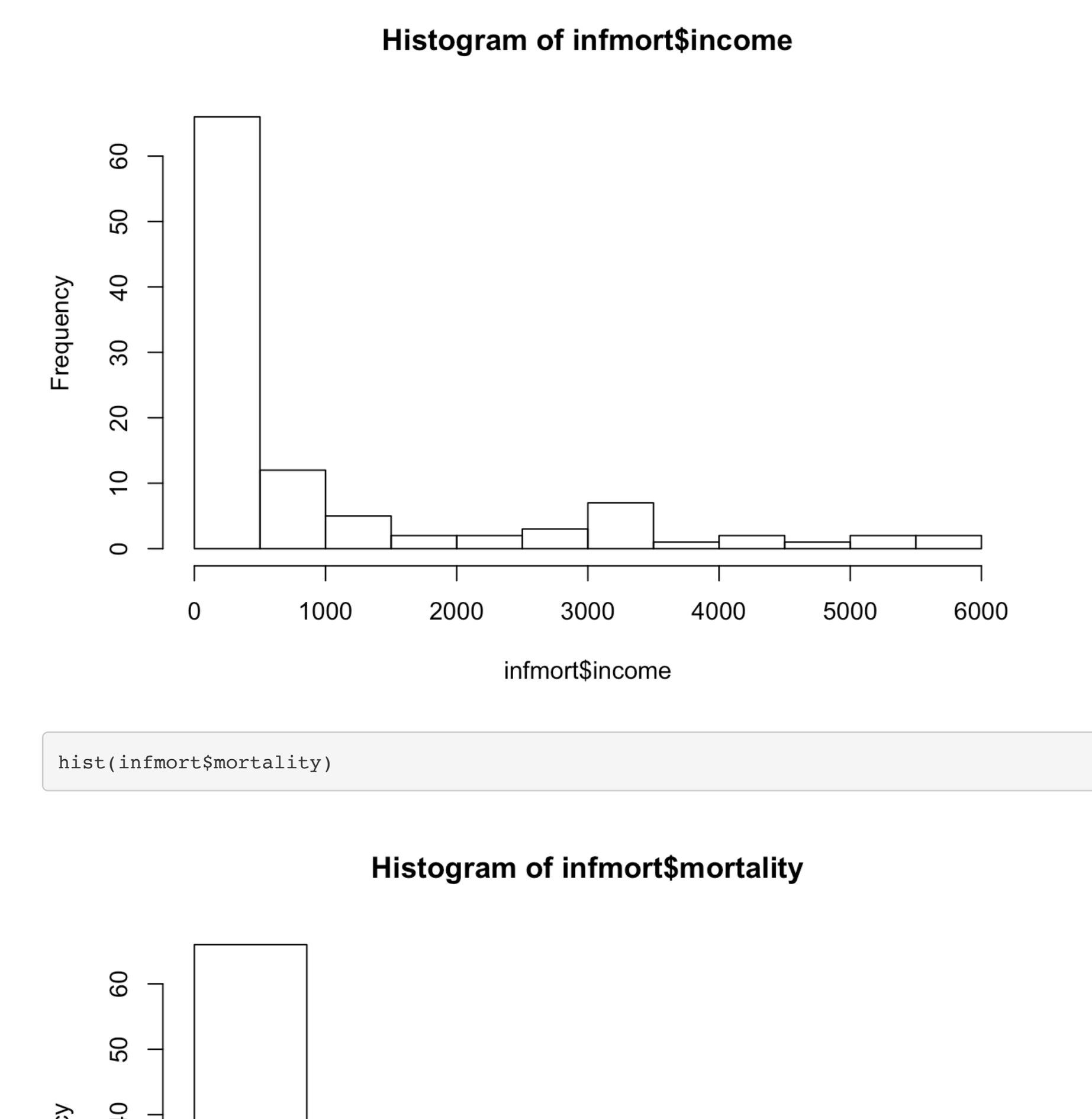
The model without the log transformations has an AIC of 1190.687 whereas the model with log transformations has an AIC of 214.6845. We want a lower AIC so the model with log transformations is prefered.

3. Is it appropriate to use AIC to compare these two models? Briefly explain why or why not.

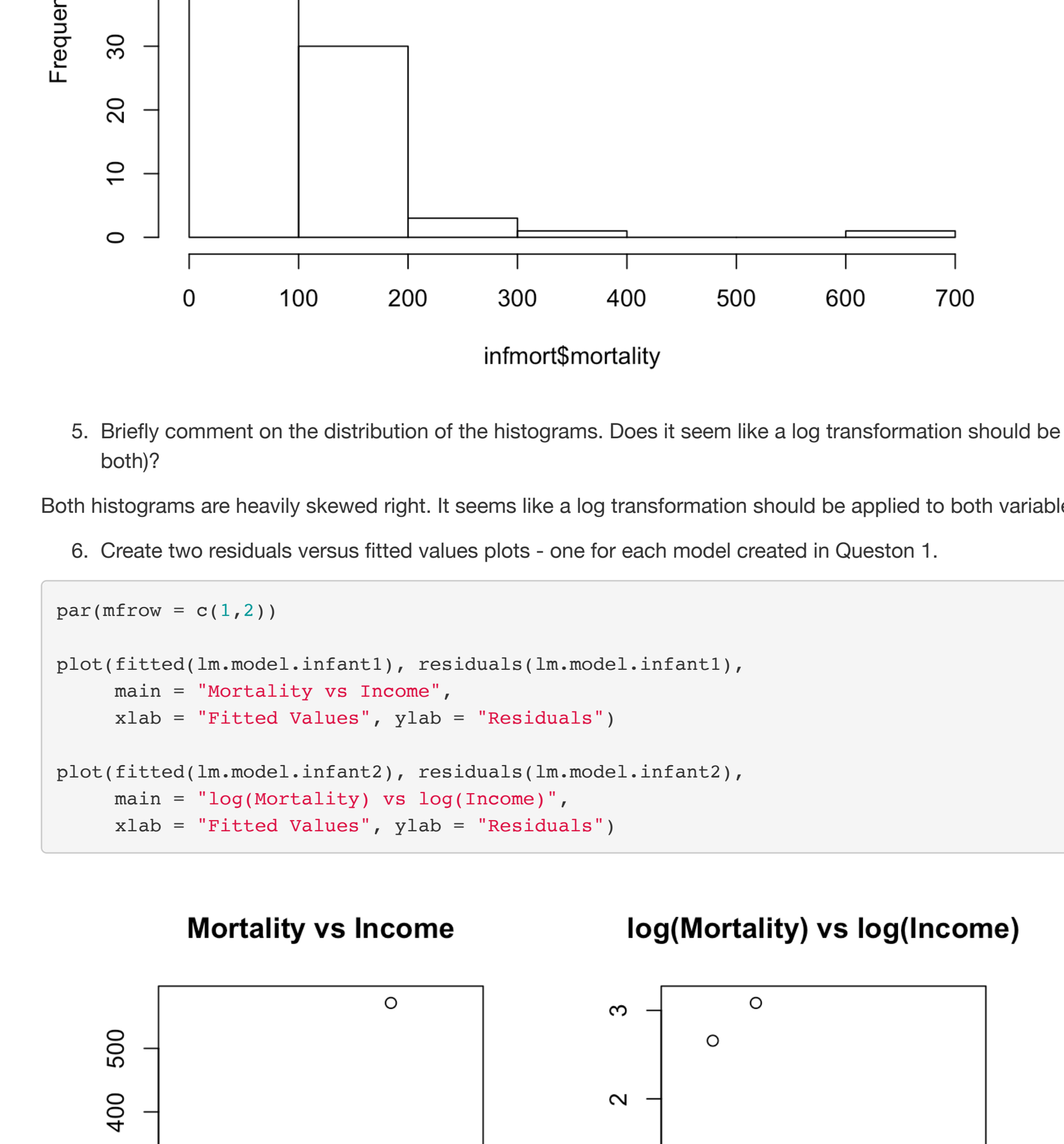
Because we took a log transformation of the outcome, we cannot compare the two models using AIC.

4. Create two histograms - one for `income` and one for `mortality`.

```
hist(infmort$income)
```




```
hist(infmort$mortality)
```



5. Briefly comment on the distribution of the histograms. Does it seem like a log transformation should be applied to either of the variables (or both)?

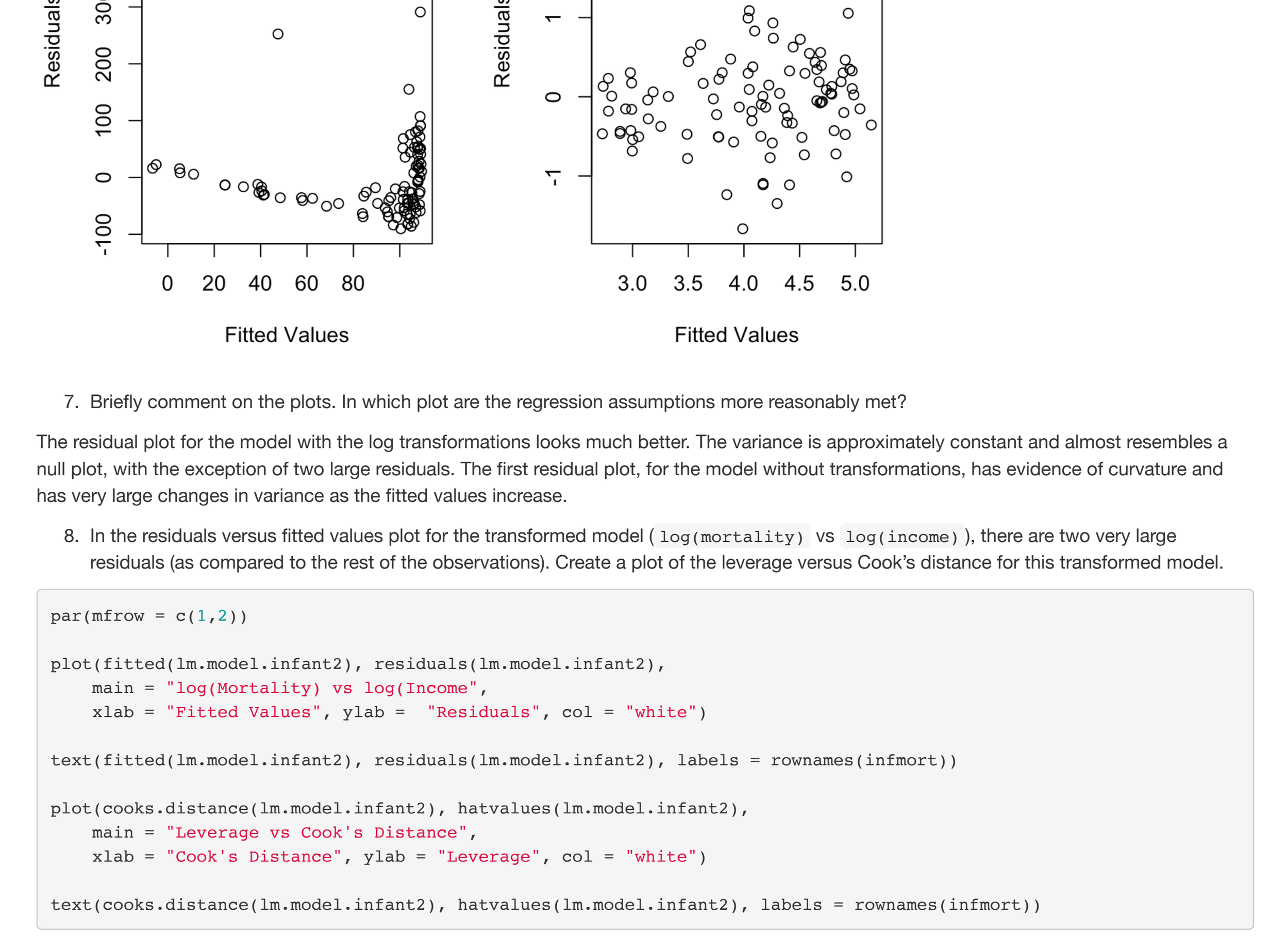
Both histograms are heavily skewed right. It seems like a log transformation should be applied to both variables.

6. Create two residuals versus fitted values plots - one for each model created in Queston 1.

```
par(mfrow = c(1,2))

plot(fitted(lm.model.infant1), residuals(lm.model.infant1),
     main = "Mortality vs Income",
     xlab = "Fitted Values", ylab = "Residuals")

plot(fitted(lm.model.infant2), residuals(lm.model.infant2),
     main = "log(Mortality) vs log(Income)",
     xlab = "Fitted Values", ylab = "Residuals")
```



7. Briefly comment on the plots. In which plot are the regression assumptions more reasonably met?

The residual plot for the model with the log transformations looks much better. The variance is approximately constant and almost resembles a null plot, with the exception of two large residuals. The first residual plot, for the model without transformations, has evidence of curvature and has very large changes in variance as the fitted values increase.

8. In the residuals versus fitted values plot for the transformed model (`log(mortality)` vs `log(income)`), there are two very large residuals (as compared to the rest of the observations). Create a plot of the leverage versus Cook's distance for this transformed model.

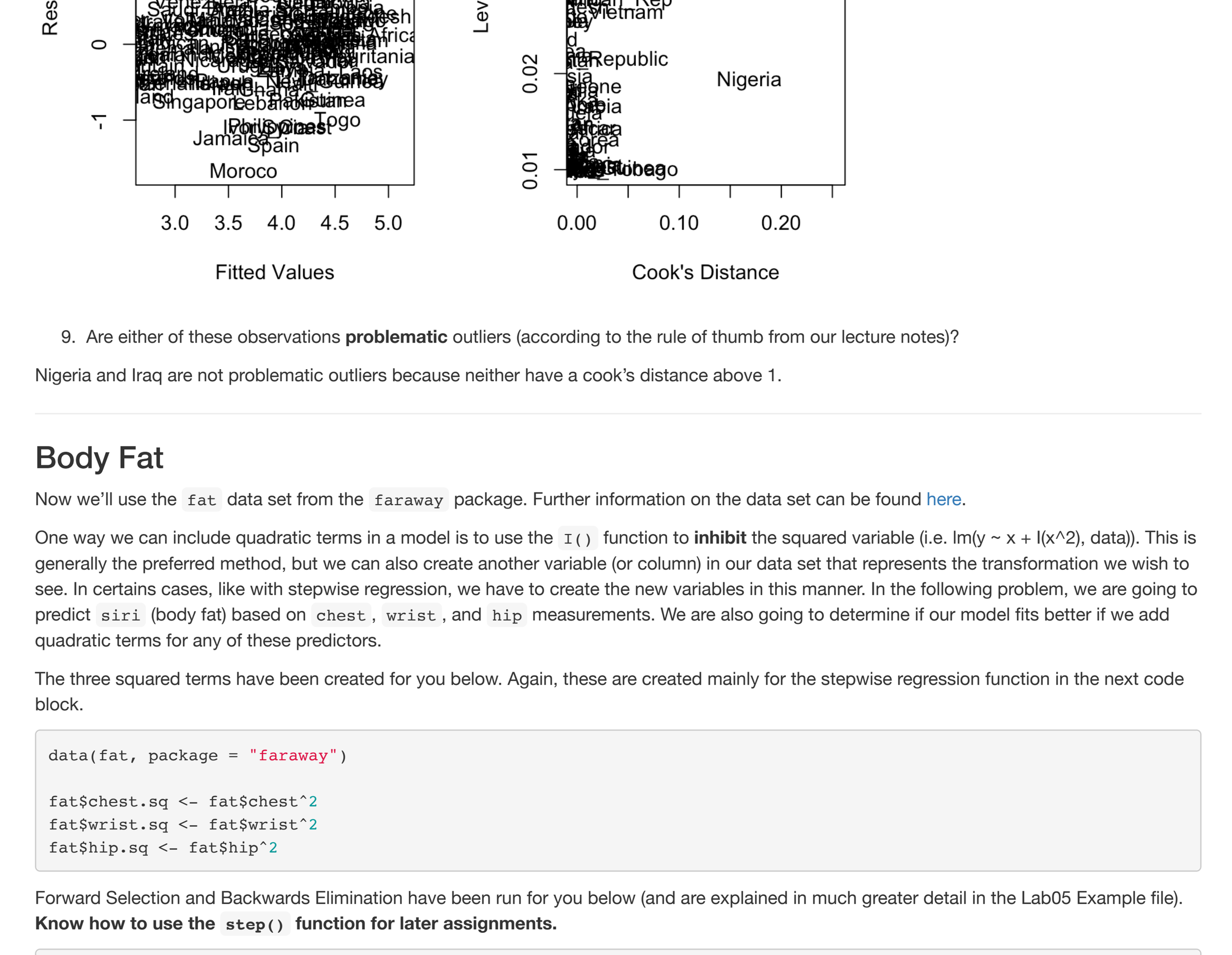
```
par(mfrow = c(1,2))

plot(fitted(lm.model.infant2), residuals(lm.model.infant2),
    main = "log(Mortality) vs log(Income",
    xlab = "Fitted Values", ylab =  "Residuals", col = "white")

text(fitted(lm.model.infant2), residuals(lm.model.infant2), labels = rownames(infmort))

plot(cooks.distance(lm.model.infant2), hatvalues(lm.model.infant2),
    main = "Leverage vs Cook's Distance",
    xlab = "Cook's Distance", ylab = "Leverage", col = "white")

text(cooks.distance(lm.model.infant2), hatvalues(lm.model.infant2), labels = rownames(infmort))
```



9. Are either of these observations **problematic** outliers (according to the rule of thumb from our lecture notes)?

Nigeria and Iraq are not problematic outliers because neither have a cook's distance above 1.

Body Fat

Now we'll use the `fat` data set from the `faraway` package. Further information on the data set can be found here.

One way we can include quadratic terms in a model is to use the `I()` function to **inhibit** the squared variable (i.e. lm(y ~ x + I(x^2), data)). This is generally the preferred method, but we can also create another variable (or column) in our data set that represents the transformation we wish to see. In certains cases, like with stepwise regression, we have to create the new variables in this manner. In the following problem, we are going to predict `siri` (body fat) based on `chest`, `wrist`, and `hip` measurements. We are also going to determine if our model fits better if we add quadratic terms for any of these predictors.

The three squared terms have been created for you below. Again, these are created mainly for the stepwise regression function in the next code block.

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

fat$chest.sq <- fat$chest^2
fat$wrist.sq <- fat$wrist^2
fat$hip.sq <- fat$hip^2
```

Forward Selection and Backwards Elimination have been run for you below (and are explained in much greater detail in the Lab05 Example file). **Know how to use the `step()` function for later assignments.**

```
null.equation <- siri ~ 1
lm.null <- lm(null.equation, data = fat)

saturated.equation <- siri ~ chest + chest.sq + wrist + wrist.sq + hip + hip.sq
lm.saturated <- lm(saturated.equation, data = fat)

forward.results <- step(lm.null, scope = saturated.equation, direction = "forward", trace = FALSE)
forward.results
```

```
##
## Call:
## lm(formula = siri ~ chest + wrist.sq + hip + hip.sq, data = fat)
##
## Coefficients:
## (Intercept)        chest     wrist.sq          hip       hip.sq
##  -1.449e+02    6.607e-01   -6.136e-02    2.009e+00   -8.252e-03
```

```
backward.results <- step(lm.saturated, scope = null.equation, direction = "backward", trace = FALSE)
backward.results
```

```
##
## Call:
## lm(formula = siri ~ chest + wrist + hip + hip.sq, data = fat)
##
## Coefficients:
## (Intercept)        chest        wrist          hip       hip.sq
##  -1.278e+02    6.609e-01   -2.276e+00    2.084e+00   -8.608e-03
```

The final models differ for the two methods ('forward.results' and 'backward.results') so we are going to use Cross Validation to see which model fits the data better.

1. Create two models using this data set. Create the first model using the predictors chosen from 'forward.results' to predict `siri`. Create the second model using the predictors chosen from 'backward.results' to predict `siri`. Do not include any interaction terms. Run each of your models through the `summary()` function.

```
lm.forward <- lm(siri ~ chest + I(wrist^2) + hip + I(hip^2), data = fat)
summary(lm.forward)$adj.r.squared
```

```
## [1] 0.54088
```

```
lm.backward <- lm(siri ~ chest + wrist + hip + I(hip^2), data = fat)
summary(lm.backward)$adj.r.squared
```

```
## [1] 0.5416903
```

2. Which model is preferred according to adjusted R-squared?

The forward selection model has an adjusted R-squared value of 0.5409, while the backward elimination model has an adjusted value of 0.5416903. We want a higher adjusted R-squared value so the backward elimination model is preferred - by a small margin.

3. Using the `cvFit()` function, run 10 replications of 5-fold CV for both of your models.

```
library(cvTools)
```

```
## Loading required package: lattice
```

```
## Loading required package: robustbase
```

```
cv5.forward <- cvFit(lm.forward, data = fat, y = fat$siri, cost = rmspe, K = 5, R = 10)
cv5.backward <- cvFit(lm.backward, data = fat, y = fat$siri, cost = rmspe, K = 5, R = 10)

cv.df <- data.frame(cv5.forward$reps, cv5.backward$reps)
colnames(cv.df) <- c("Forward", "Backward")
cv.df
```

```
##     Forward Backward
## 1  5.720039 5.729386
## 2  5.716864 5.723579
## 3  5.713399 5.702318
## 4  5.696736 5.752366
## 5  5.717195 5.643313
## 6  5.717307 5.736301
## 7  5.678698 5.812368
## 8  5.752713 5.724199
## 9  5.665681 5.818780
## 10 5.756821 5.656584
```

4. Although the variations from replication to replication are small, notice the randomness of k-fold CV. Is one model always preferred over the other using this method?

No, both root mean squared prediction errors are extremely close and in some cases one model is lower than the other.

5. Using the `cvFit()` function, run LOOCV for both of your models.

```
cvFit(lm.forward, data = fat, y = fat$siri, cost = rmspe, K = nrow(fat))
```

```
## Leave-one-out CV results:
##       CV
## 5.726378
```

```
cvFit(lm.backward, data = fat, y = fat$siri, cost = rmspe, K = nrow(fat))
```

```
## Leave-one-out CV results:
##       CV
## 5.726329
```

6. LOOCV is not random (and would return the same result if replicated). Which model is preferred according to LOOCV?

The forward selection model has an RMSPE of 5.7264, while the backward elimination model has an RMSPE of 5.7263. We want a lower RMSPE so technically we would choose the backward elimination model, but the difference is so minimal we will decide based on practicality.

7. The results of cross validation are very, very similar for this analysis. If **you** had to choose a model based on how practical the terms of each model are, which model would you prefer? Briefly explain why.

The model with backward because on the basis of the number of predictors we have, since model 1, or the forward model, does not include wrist's linear term.