MAST90104: Introduction to Statistical Learning

Solutions to Week 6 Lab and Workshop

1. Load the `beef` dataset from the website:

```
beef <- read.csv('../data/beef.csv')
```

In the USA, the Cattlemen's Beef Board and the National Cattlemen's Beef Association promote the consumption of beef with an advertising campaign using the theme "Beef: it's what's for dinner". The campaign is paid for by the "Beef Checkoff", a law that requires all cattle producers to pay $1 per head of cattle sold to support beef/veal promotion and research. In 1988 the Missoulian newspaper surveyed the cattle growers of Montana, and for each of Montana's 56 counties reported the percent of growers voting "yes" for the checkoff.

In this question we explain the size of the yes vote in terms of the characteristics of the farms in each county. Data on farms is taken from the U.S. Bureau of the Census, City and County Data Book, 1986. The variables given in the dataset are:

yes Percentage of farmers voting "yes" for the checkoff

big Percentage of farms with 500 acres or more

prin Percentage of operators whose principle income is farming

size Average size of farm (hundreds of acres)

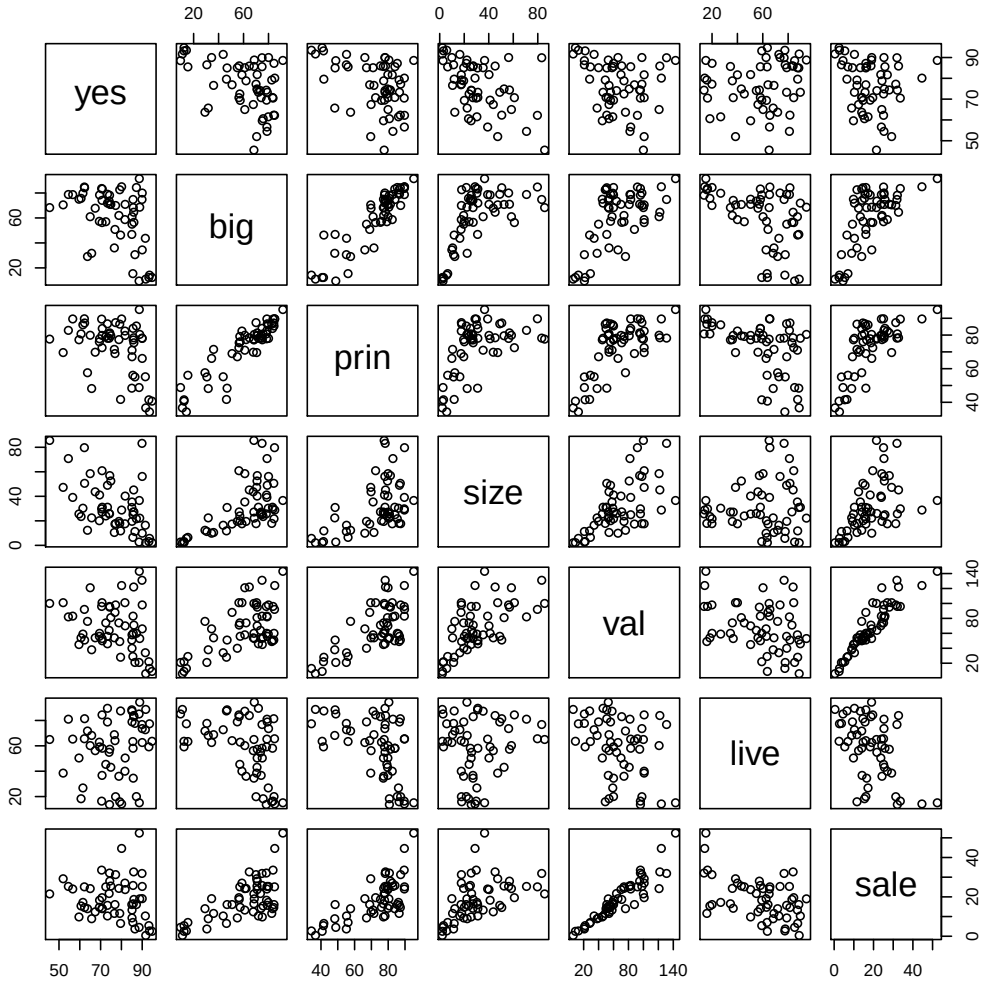
val Average value of products sold ($1000's)

live Percentage of products sold from livestock and poultry

sale Percentage of farms with sales of $100,000 or more

(a) Use `pairs` to plot the data. Is there any evidence of non-linearity or heteroskedasticity?
Solution:

```
pairs(beef)
```



There is some evidence of heteroskedasticity in `yes`, paricularly vs `size` and `val`. We could consider taking logs of `size` and `val` and seeing if that improved the fit, though we won't for the moment.

(b) Using the `add1` and `drop1` commands, use forward and backward selection to find parsimonious models for `yes`.

Solution: We use a 5% significance level. Forward selection:

```
model0 <- lm(yes ~ 1, data = beef)
add1(model0, scope = ~ . + big + prin + size + val + live + sale, test = "F")

## Single term additions
##
## Model:
## yes ~ 1
##        Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>              7689.2 277.64
## big     1   1482.29 6206.9 267.65 12.8960 0.0007112 ***
## prin    1   1288.87 6400.3 269.37 10.8744 0.0017287 **
## size    1   1925.80 5763.4 263.50 18.0439 8.572e-05 ***
## val     1    538.85 7150.3 275.57  4.0694 0.0486484 *
## live    1    226.19 7463.0 277.97  1.6366 0.2062603
## sale    1    214.20 7475.0 278.06  1.5474 0.2188955
## ---
```

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

```
model1 <- lm(yes ~ size, data = beef)
add1(model1, scope = ~ . + big + prin + val + live + sale, test = "F")
```

```
## Single term additions
##
## Model:
## yes ~ size
##        Df Sum of Sq    RSS    AIC F value  Pr(>F)
## <none>              5763.4 263.50
## big     1    218.65 5544.7 263.33  2.0900 0.15415
## prin    1    222.46 5540.9 263.30  2.1279 0.15054
## val     1     35.20 5728.2 265.16  0.3256 0.57064
## live    1    348.02 5415.3 262.01  3.4060 0.07055 .
## sale    1     69.35 5694.0 264.82  0.6455 0.42533
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

None of the variables make a significant difference when added, so we end up with the model

$$yes = \beta_0 + \beta_1\, size + \epsilon.$$

Backward selection:

```
model0 <- lm(yes ~ ., data = beef)
drop1(model0, scope = ~ ., test = "F")
```

```
## Single term deletions
##
## Model:
## yes ~ big + prin + size + val + live + sale
##        Df Sum of Sq    RSS    AIC F value   Pr(>F)
## <none>              4540.7 260.15
## big     1      0.51 4541.2 258.15  0.0055 0.941407
## prin    1     64.15 4604.9 258.93  0.6923 0.409434
## size    1    750.11 5290.8 266.71  8.0946 0.006463 **
## val     1     48.95 4589.7 258.75  0.5282 0.470810
## live    1    461.28 5002.0 263.56  4.9778 0.030283 *
## sale    1    411.63 4952.3 263.01  4.4420 0.040206 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model1 <- lm(yes ~ prin + size + val + live + sale, data = beef)
drop1(model1, scope = ~ ., test = "F")
```

```
## Single term deletions
##
## Model:
## yes ~ prin + size + val + live + sale
##        Df Sum of Sq    RSS    AIC F value   Pr(>F)
## <none>              4541.2 258.15
## prin    1    201.55 4742.8 258.58  2.2191 0.142595
## size    1   1079.23 5620.5 268.09 11.8826 0.001158 **
## val     1     52.81 4594.0 256.80  0.5815 0.449322
## live    1    492.67 5033.9 261.92  5.4244 0.023938 *
## sale    1    436.69 4977.9 261.30  4.8080 0.033008 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model2 <- lm(yes ~ prin + size + live + sale, data = beef)
drop1(model2, scope = ~ ., test = "F")

## Single term deletions
##
## Model:
## yes ~ prin + size + live + sale
##        Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>              4594.0 256.80
## prin    1    198.74 4792.8 257.17  2.2063 0.1436041
## size    1   1566.35 6160.4 271.23 17.3886 0.0001183 ***
## live    1    467.28 5061.3 260.23  5.1875 0.0269741 *
## sale    1    804.56 5398.6 263.84  8.9317 0.0043034 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

model3 <- lm(yes ~ size + live + sale, data = beef)
drop1(model3, scope = ~ ., test = "F")

## Single term deletions
##
## Model:
## yes ~ size + live + sale
##        Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>              4792.8 257.17
## size    1   2600.39 7393.2 279.45 28.2134 2.294e-06 ***
## live    1    901.24 5694.0 264.82  9.7781   0.00289 **
## sale    1    622.57 5415.3 262.01  6.7546   0.01214 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All of the remaining variables make a significant difference when removed, so we end up with the model

$$yes = \beta_0 + \beta_1\, size + \beta_2\, sale + \beta_3\, live + \epsilon.$$

(c) Using the `step` command, starting from a model with just an intercept, use the AIC and stepwise selection to choose a model.

Solution:

```
basemodel <- lm(yes ~ 1, data = beef)
model <- step(basemodel, scope = ~ . + big + prin + size + val + live + sale)

## Start:  AIC=277.64
## yes ~ 1
##
##        Df Sum of Sq    RSS    AIC
## + size  1   1925.80 5763.4 263.50
## + big   1   1482.29 6206.9 267.65
## + prin  1   1288.87 6400.3 269.37
## + val   1    538.85 7150.3 275.57
## <none>              7689.2 277.64
## + live  1    226.19 7463.0 277.97
## + sale  1    214.20 7475.0 278.06
##
## Step:  AIC=263.5
## yes ~ size
##
##        Df Sum of Sq    RSS    AIC
## + live  1    348.02 5415.3 262.01
```

```
## + prin  1    222.46 5540.9 263.30
## + big   1    218.65 5544.7 263.33
## <none>              5763.4 263.50
## + sale  1     69.35 5694.0 264.82
## + val   1     35.20 5728.2 265.16
## - size  1   1925.80 7689.2 277.64
##
## Step:  AIC=262.01
## yes ~ size + live
##
##        Df Sum of Sq    RSS    AIC
## + sale  1    622.57 4792.8 257.17
## + val   1    332.68 5082.7 260.46
## <none>              5415.3 262.01
## - live  1    348.02 5763.4 263.50
## + prin  1     16.75 5398.6 263.84
## + big   1     15.35 5400.0 263.85
## - size  1   2047.63 7463.0 277.97
##
## Step:  AIC=257.17
## yes ~ size + live + sale
##
##        Df Sum of Sq    RSS    AIC
## + prin  1    198.74 4594.0 256.80
## <none>              4792.8 257.17
## + big   1     92.28 4700.5 258.08
## + val   1     50.00 4742.8 258.58
## - sale  1    622.57 5415.3 262.01
## - live  1    901.24 5694.0 264.82
## - size  1   2600.39 7393.2 279.45
##
## Step:  AIC=256.8
## yes ~ size + live + sale + prin
##
##        Df Sum of Sq    RSS    AIC
## <none>              4594.0 256.80
## - prin  1    198.74 4792.8 257.17
## + val   1     52.81 4541.2 258.15
## + big   1      4.37 4589.7 258.75
## - live  1    467.28 5061.3 260.23
## - sale  1    804.56 5398.6 263.84
## - size  1   1566.35 6160.4 271.23
```

In this case our model is even larger:

$$yes = \beta_0 + \beta_1\, size + \beta_2\, sale + \beta_3\, live + \beta_4\, prin + \epsilon.$$

(d) Show that the model found in 1c can be improved by adding the interaction term `size*sale`. (Important here is how you judge "improved".)

Use stepwise selection again to see if adding `size*sale` can let you remove any other variables from the model.

Solution:

```
model1 <- lm(yes ~ size + live + sale + prin + size*sale, data = beef)
model2 <- step(model1, scope = ~ .)

## Start:  AIC=251.08
## yes ~ size + live + sale + prin + size * sale
```

```
##
##             Df Sum of Sq    RSS    AIC
## - prin       1      8.54 4010.7 249.20
## <none>                   4002.1 251.08
## - live       1    535.79 4537.9 256.11
## - size:sale  1    591.90 4594.0 256.80
##
## Step:  AIC=249.2
## yes ~ size + live + sale + size:sale
##
##             Df Sum of Sq    RSS    AIC
## <none>                   4010.7 249.20
## + prin       1      8.54 4002.1 251.08
## - live       1    563.60 4574.3 254.56
## - size:sale  1    782.10 4792.8 257.17
```

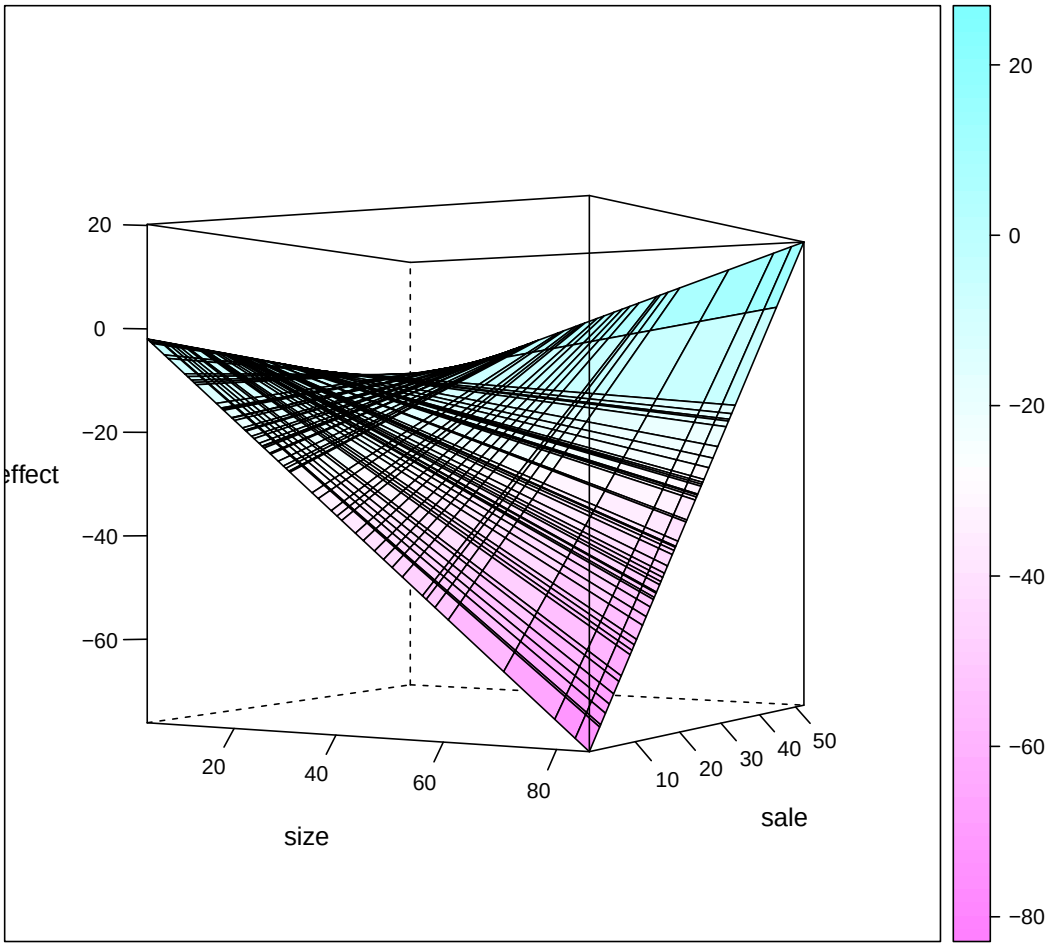
The model `yes ~ size + live + sale + prin` has AIC 256.80 and the model `yes ~ size + live + sale + prin + size*sale` has AIC 251.08, indicating a better fit. Removing `prin` improves the AIC further. Note that R does not consider removing `size` or `sale` while `size:sale` is still in the model. Current model is

$$yes = \beta_0 + \beta_1\, size + \beta_2\, sale + \beta_3\, live + \beta_{12}\, size \times sale + \epsilon.$$

(e) Suppose that β_1, β_2 and β_{12} are the coefficients of $x_1 =$ `size`, $x_2 =$ `sale` and `size*sale`, in the model from 1d. Plot $\beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 * x_2$ as a function of (x_1, x_2), to see the combined effect of these variables on the yes vote. You may need the `wireframe` function from the `lattice` library, and also `expand.grid`.

Solution:

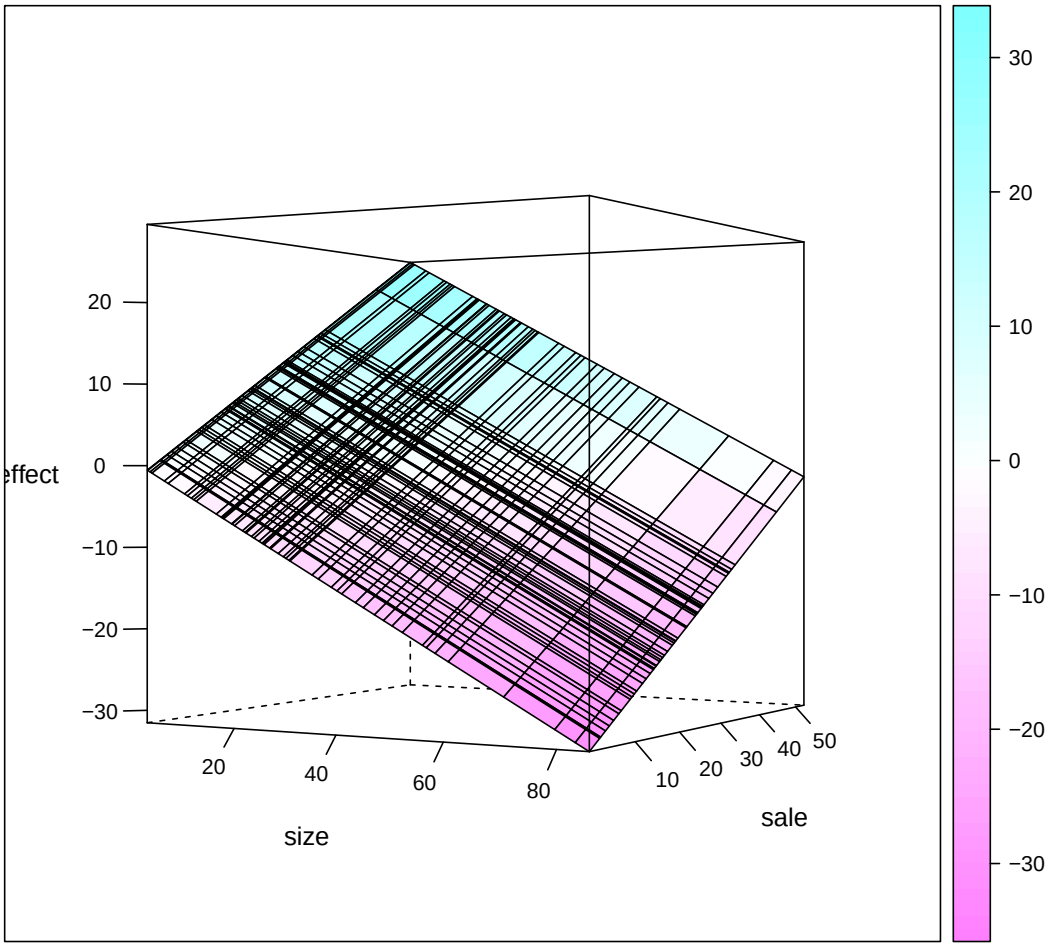
```
library(lattice)
df <- expand.grid(size=beef$size, sale=beef$sale)
f <- function(x, y) sum( model2$coefficients[c(2, 4, 5)] * c(x, y, x*y) )
df$effect <- mapply(f, df$size, df$sale)
wireframe(effect ~ size + sale, data = df, drape = T,
scales = list(arrows=F), screen = list(z = 30, x = -90, y = -60))
```



We see that `sale` has a big effect on `yes` when `size` is large, but when `size` is small `sale` isn't so important.

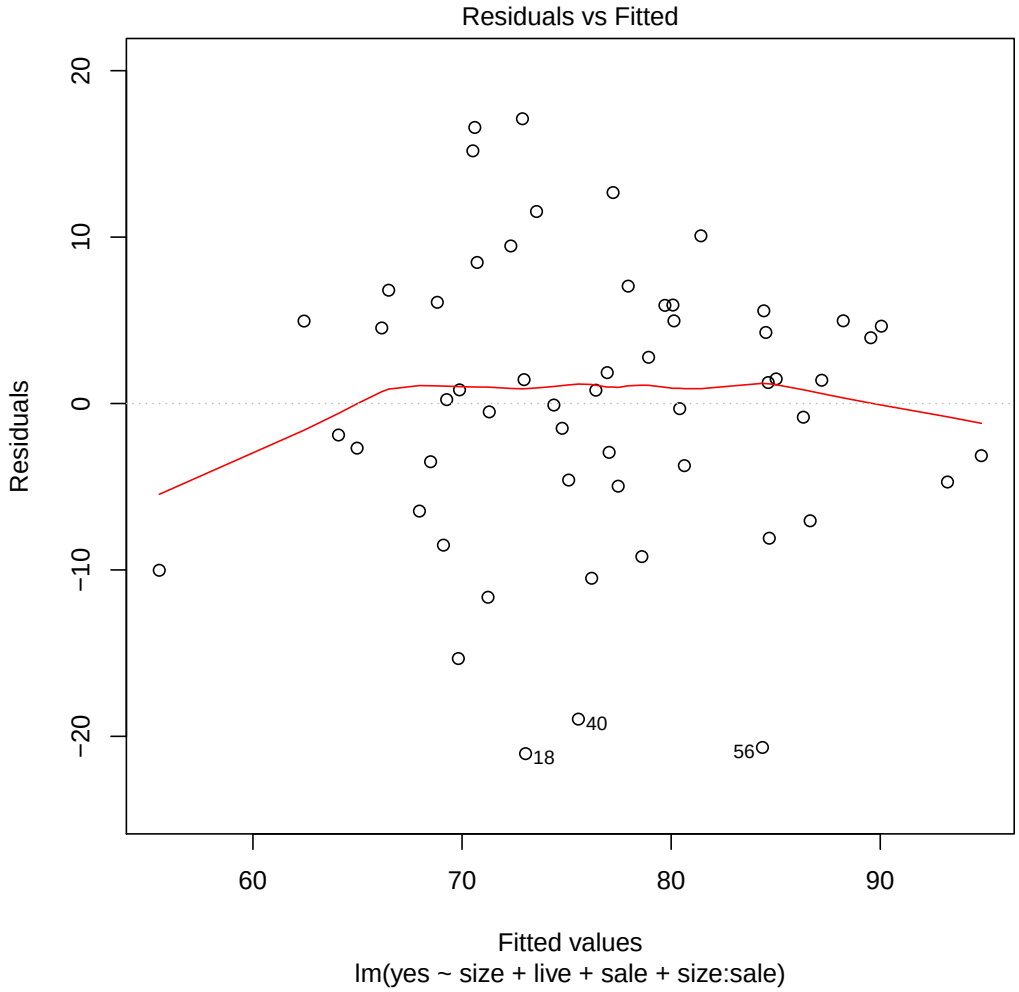
(f) Repeat the above question using the model with no `size*sale` interaction term from 1c.
Solution:

```
f <- function(x, y) sum( model$coefficients[c(2, 4)] * c(x, y) )
df$effect <- mapply(f, df$size, df$sale)
wireframe(effect ~ size + sale, data = df, drape = T,
scales = list(arrows=F), screen = list(z = 30, x = -90, y = -60))
```

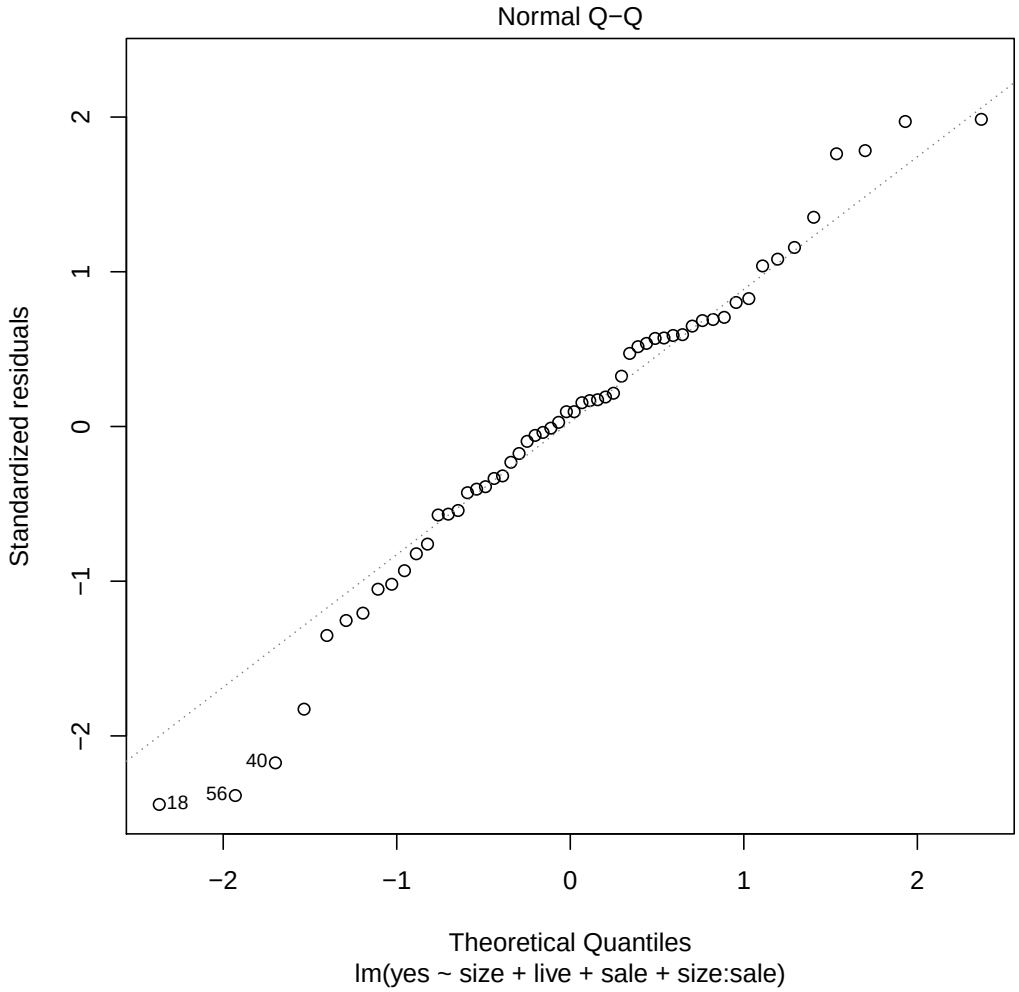


(g) Use the diagnostic plots provided by R to assess the model from 1d.
Refer back to 1a; do you need to transform the data and start again?
Solution:

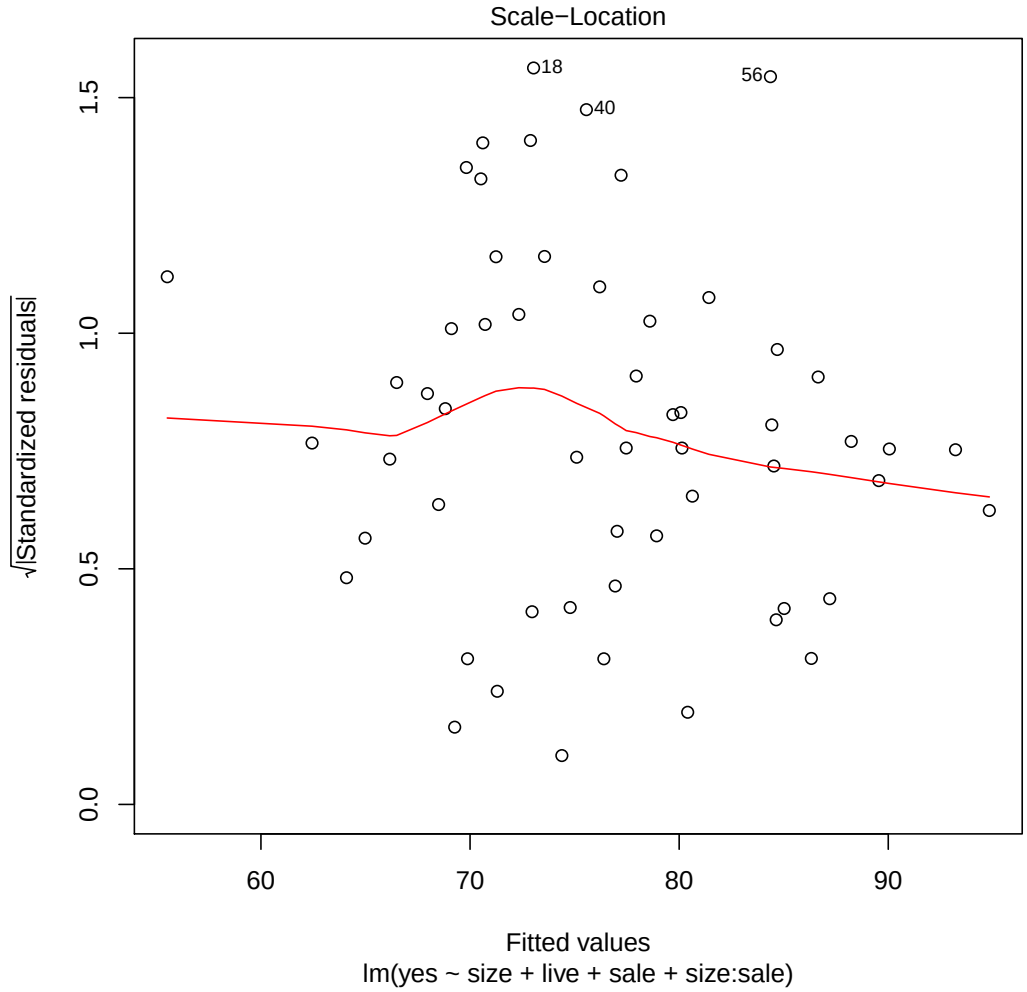
```
plot(model2, which = 1)
```



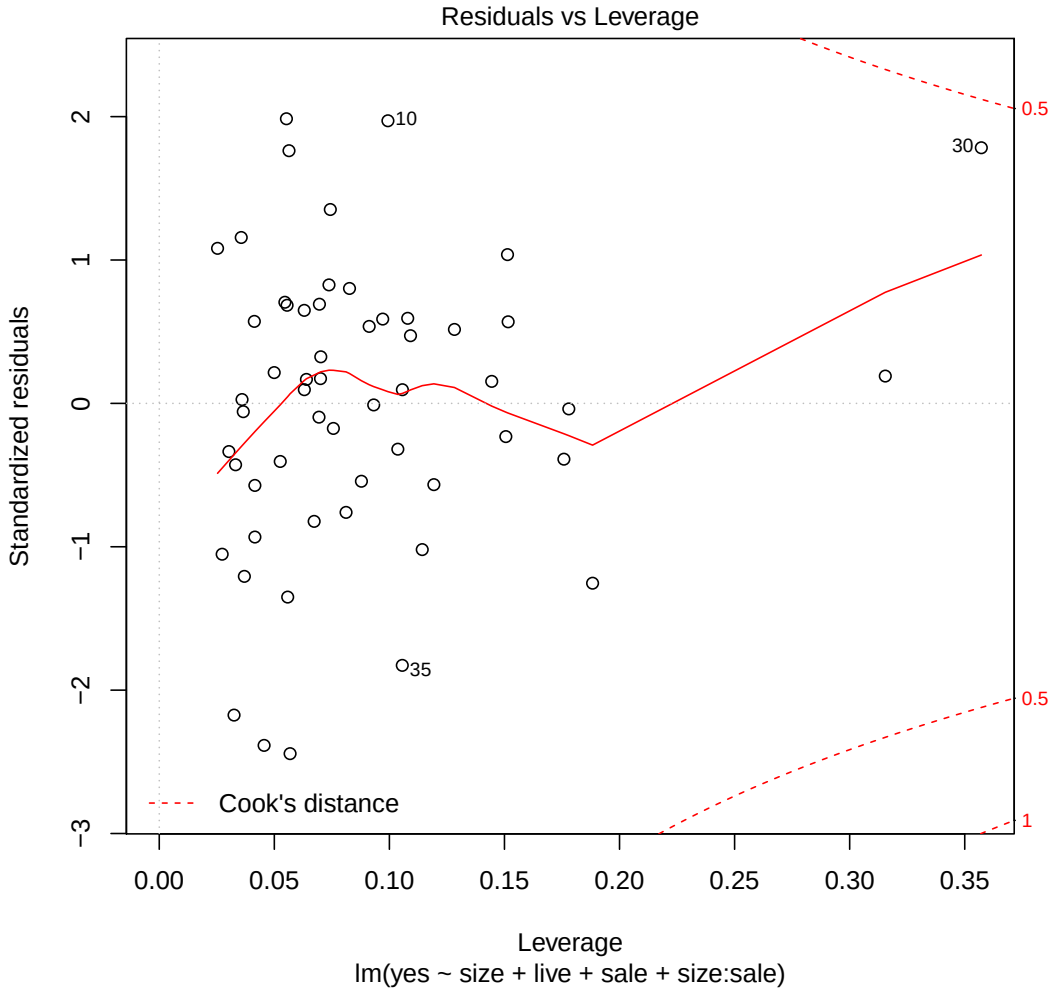

```
plot(model2, which = 2)
```




```
plot(model2, which = 3)
```




```
plot(model2, which = 5)
```



There is perhaps some evidence of heteroskedasticity in the plot of the square root of the absolute standardised residuals against fitted values, but not enough to be a problem, so no need to consider transforming the data.

(h) Which are the most important variables when it comes to predicting the yes vote? In deciding this, take into account the average size of the variables as well as the size of the fitted coefficients.

Solution: Significance is not the same as importance. The average contribution of each variable to the overall mean can be calculated as follows

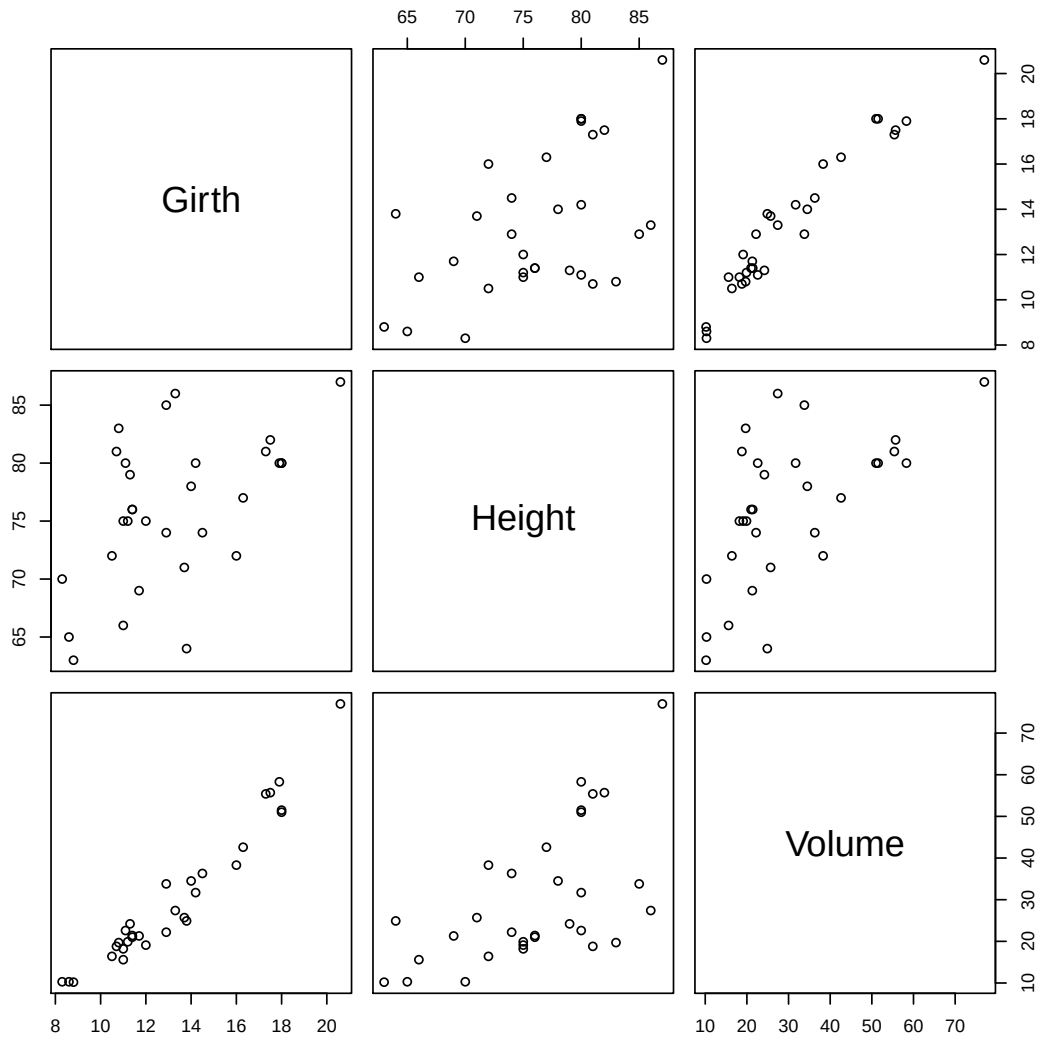
```
mean(beef$size)*model2$coefficients[2]

##      size
## -27.85712

mean(beef$live)*model2$coefficients[3]

##     live
## 10.38959

mean(beef$sale)*model2$coefficients[4]

##      sale
## -3.225588

mean(beef$size*beef$sale)*model2$coefficients[5]
```

```
## size:sale
##  16.12498
```

So `size` has the most influence, followed by `sale` because of the interaction term.

2. Load and examine the dataset `trees` using

```
data(trees)
?trees
pairs(trees)
```



We will model the volume of a black cherry tree as a function of its girth and height.

(a) By calculating $R(\gamma_1|\gamma_2)$ and SS_{Res} from the data $\mathbf{y}$ and design matrix X, use an F test to determine if including the variable `Height` significantly improves the model fitted using only `Girth` (and an intercept).

Repeat the test using the `lm` and `anova` commands, to see if you get the same numbers.

Solution: By "hand":

```
y <- trees$Volume
n <- length(y)
```

```
X <- cbind(1, trees$Girth, trees$Height)
b <- solve(t(X) %*% X, t(X) %*% y)
(SS_res <- sum((y - X %*% b)^2))

## [1] 421.9214

SS_reg <- sum((X %*% b)^2)
X2 <- X[,-3]
b2 <- solve(t(X2) %*% X2, t(X2) %*% y)
SS_reg2 <- sum((X2 %*% b2)^2)
(R_g1g2 <- SS_reg - SS_reg2)

## [1] 102.3812

(Fstat <- (R_g1g2/1)/(SS_res/(n - 3)))

## [1] 6.79433

pf(Fstat, 1, n - 3, lower.tail = F)

## [1] 0.01449097
```

Using `lm` and `anova`:

```
model1 <- lm(Volume ~ Girth, data = trees)
model2 <- lm(Volume ~ Girth + Height, data = trees)
anova(model1, model2)

## Analysis of Variance Table
##
## Model 1: Volume ~ Girth
## Model 2: Volume ~ Girth + Height
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     29 524.30
## 2     28 421.92  1    102.38 6.7943 0.01449 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(b) Add variables `Girth` squared and `Girth` squared times `Height` to the model, then use stepwise selection to simplify the model. (You can use `step` for this step.)
Comment on the form of your final model.
Solution:

```
trees$GirthSq <- trees$Girth^2
model <- lm(Volume ~ Girth + Height + GirthSq + GirthSq*Height, data = trees)
model <- step(model, scope = ~ .)

## Start:  AIC=64.36
## Volume ~ Girth + Height + GirthSq + GirthSq * Height
##
##                  Df Sum of Sq    RSS    AIC
## - Girth           1    0.2288 179.27 62.402
## - Height:GirthSq  1    6.9694 186.01 63.547
## <none>                        179.04 64.363
##
## Step:  AIC=62.4
## Volume ~ Height + GirthSq + Height:GirthSq
##
##                  Df Sum of Sq    RSS    AIC
## <none>                        179.27 62.402
## + Girth           1     0.229 179.04 64.363
```

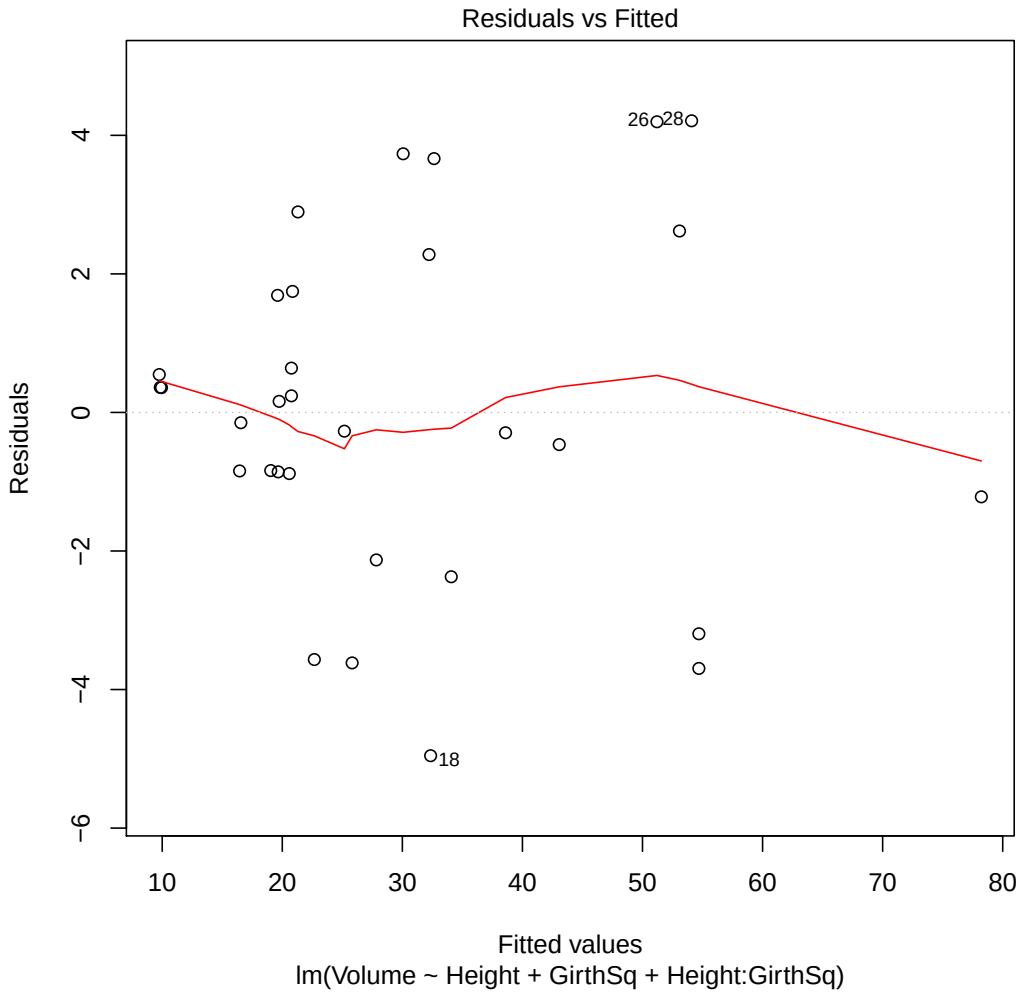
```
## - Height:GirthSq  1    40.164 219.44 66.669
```

Note that R will not attempt to drop `GirthSq` and `Height` while `GirthSq*Height` is still in the model.

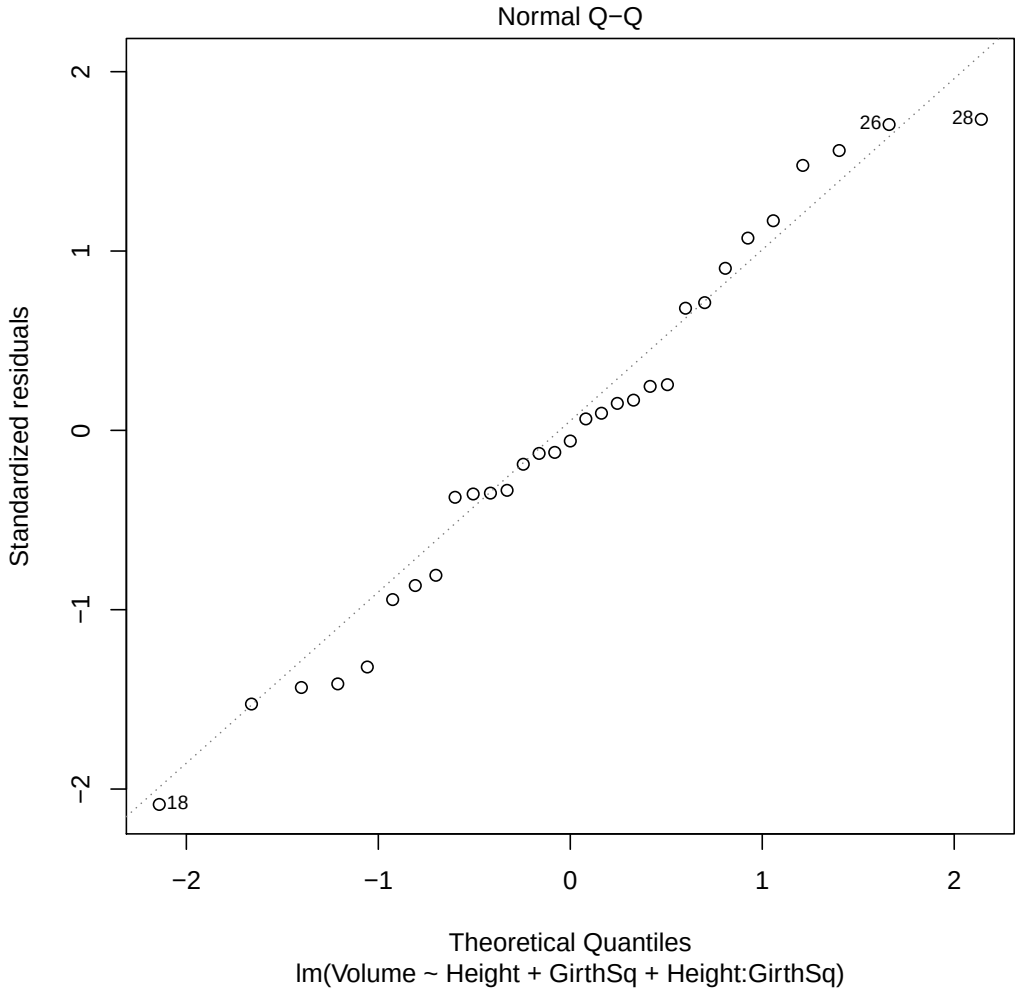
(c) Use diagnostic plots to check the fit of your final model.

Solution:

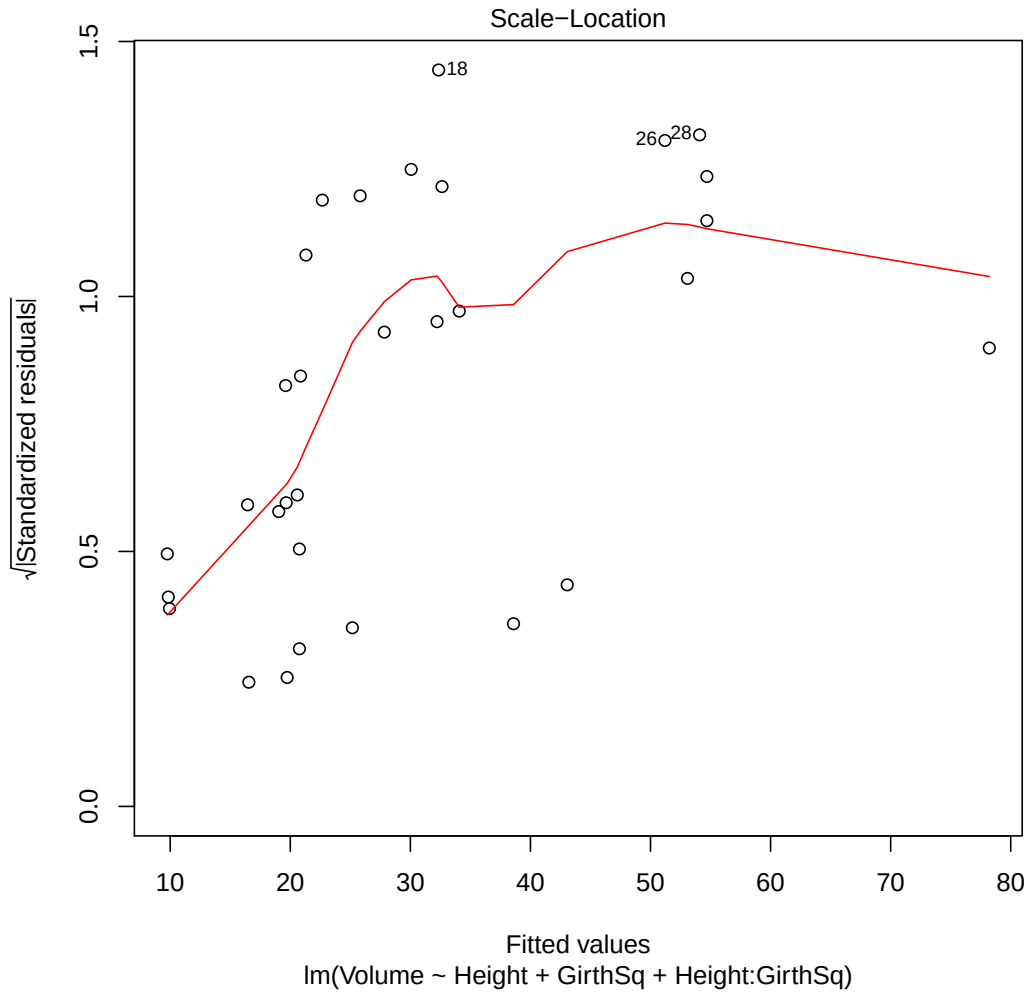
```
plot(model, which = 1)
```



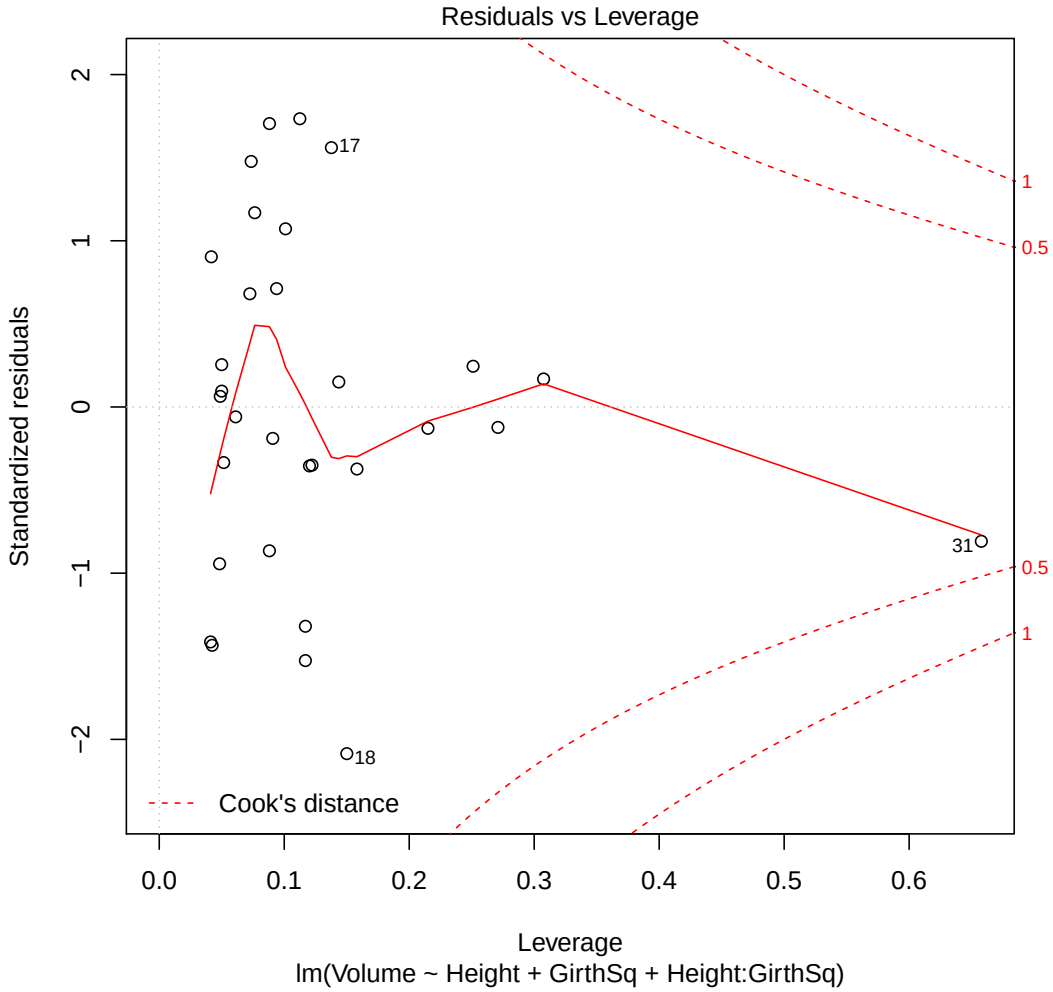

```
plot(model, which = 2)
```




```
plot(model, which = 3)
```




```
plot(model, which = 5)
```

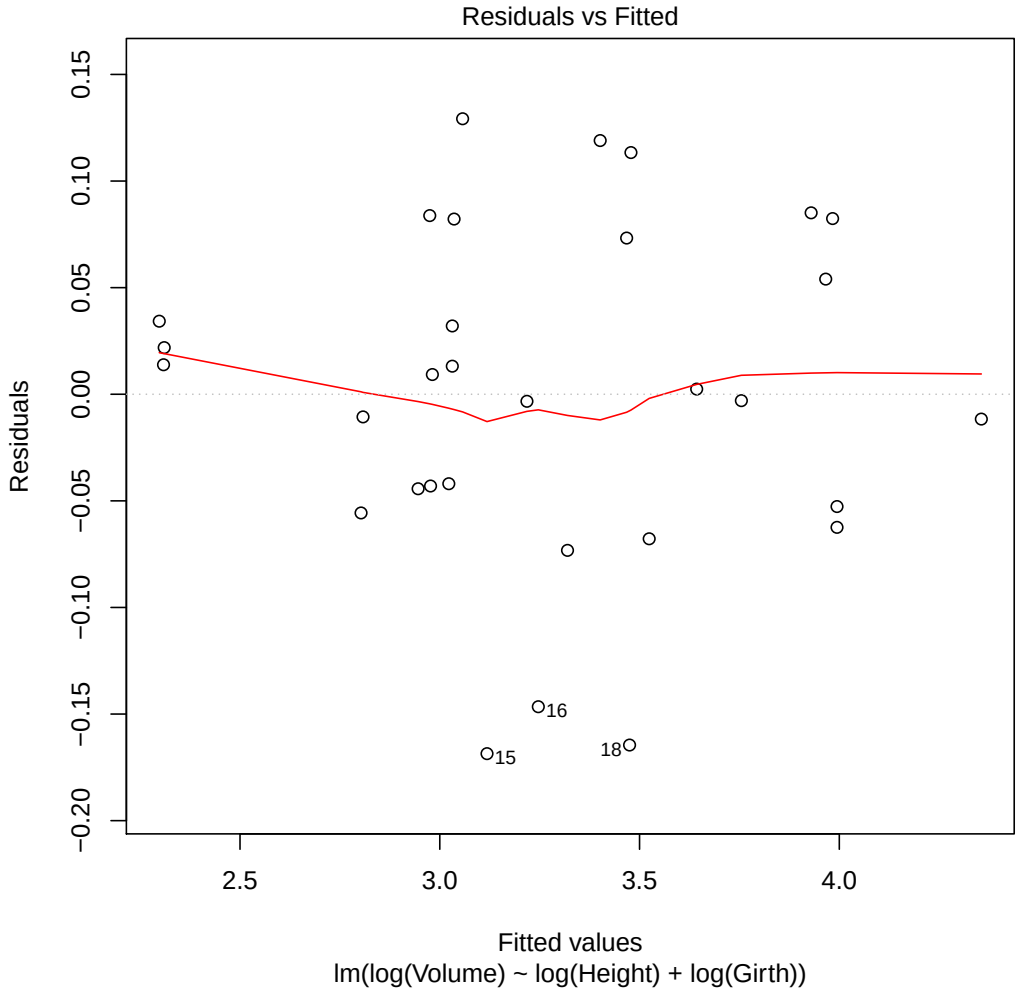


(d) What transformation might be indicated from the plot of residuals versus fitted values? Transform all variables with this transformation. What might the appropriate model be? Fit it and comment on the resulting residuals.

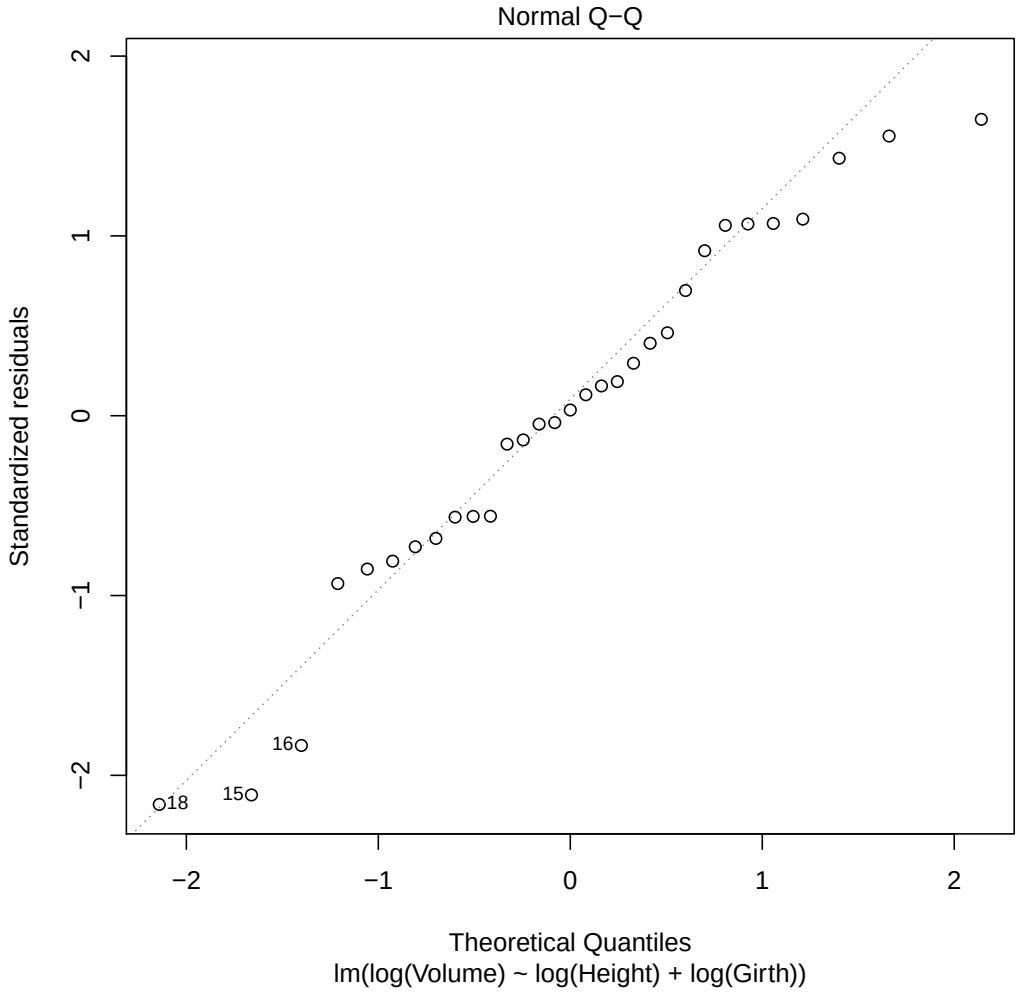
Solution: From the third plot we see that the residuals get larger as the fitted values increase. Perhaps, rather than including the girth squared term, we should take logs. The only way to be sure is to try and see if the residuals look better. If you do this you will see that the diagnostic plots are much the same for the transformed model as for the previous one, making it hard to choose between them. (Note that because we have transformed the response, we can't meaningfully compare the AIC scores for the two models.)

```
model2 <- lm(log(Volume) ~ log(Height) + log(Girth), data = trees)
```

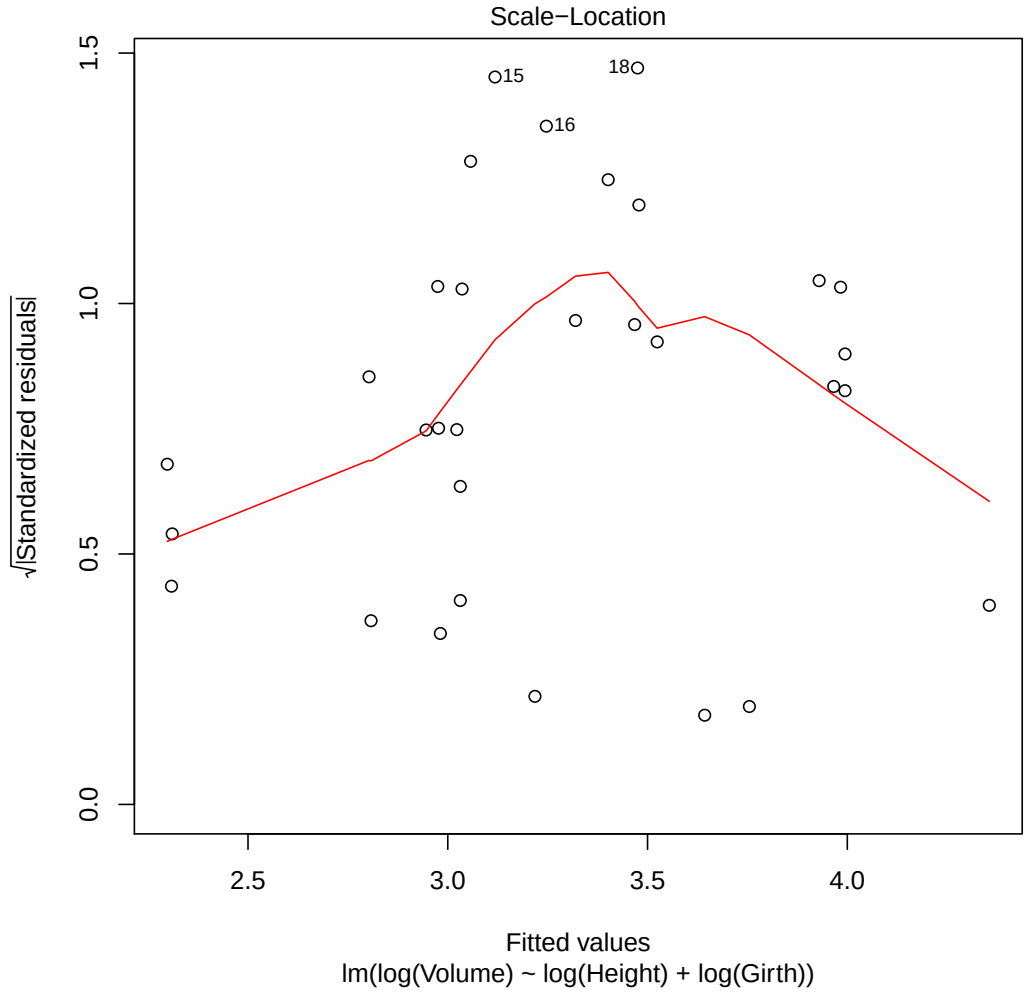
```
plot(model2, which = 1)
```



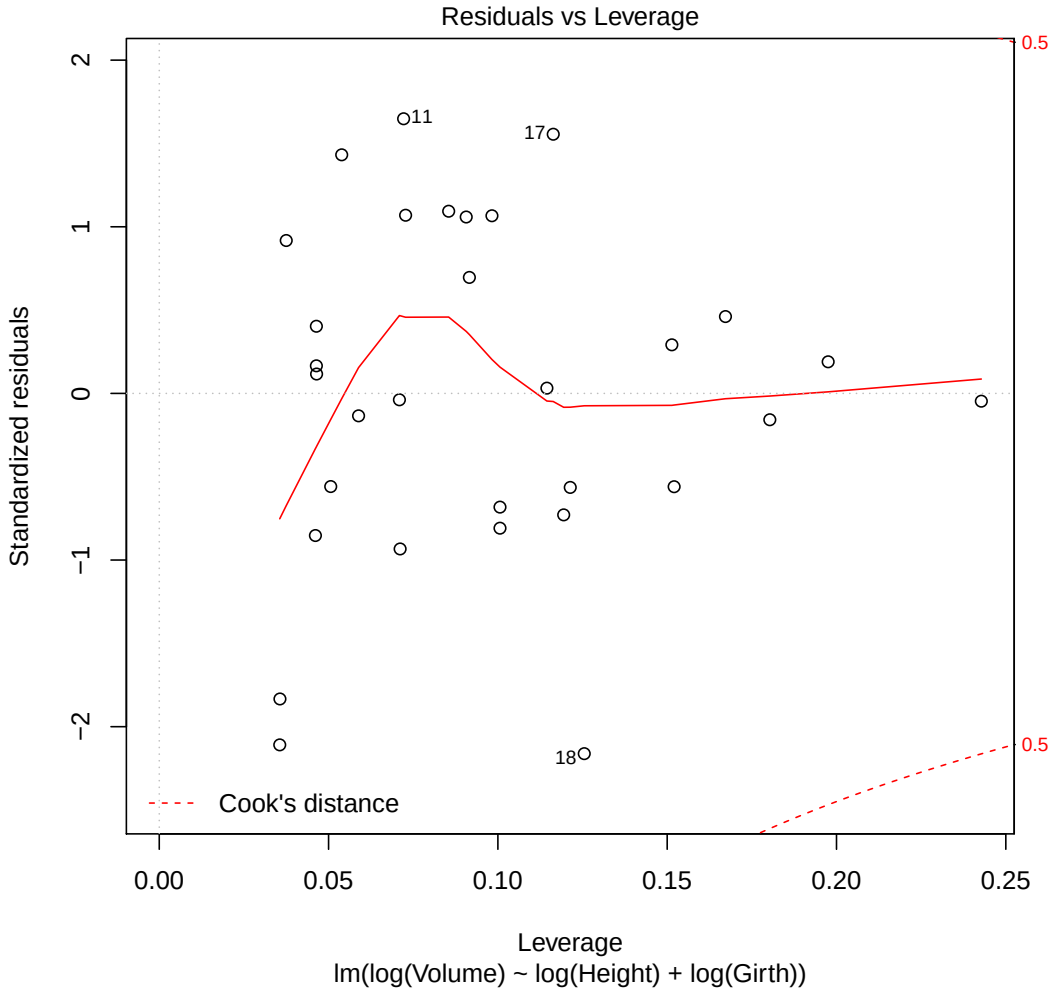

```
plot(model2, which = 2)
```




```
plot(model2, which = 3)
```




```
plot(model2, which = 5)
```



3. If the model has an intercept term, what is the mean of the fitted values? [*Hint: use the normal equation for the intercept term*]

   Solution:

   The normal equation for the intercept term is the first one from $X^T X\mathbf{b} = X^T\mathbf{y}$:

   $$nb_0 + \mathbf{1}^T\mathbf{x}_1 b_1 + \cdots + \mathbf{1}^T\mathbf{x}_k b_k = \mathbf{1}^T\mathbf{y},$$

   where $\mathbf{1}$ is the $n \times 1$ column vector of 1's and $\mathbf{x}_1, \cdots, \mathbf{x}_k$ are the column vectors of X. Dividing through by n gives

   $$\bar{\hat{y}} = b_0 + b_1\bar{\mathbf{x}_1} + \cdots + b_k\bar{\mathbf{x}_k} = \bar{\mathbf{y}},$$

   so the mean of the fitted values is the mean of the responses.

4. Show that R^2 is the square of the correlation coefficient between the data and the fitted values. [*Hint: write the response as fit + residual. Express the correlation coefficient as that of a random vector which chooses randomly from the data (both y and corresponding row of X) and records the response value and fitted. Use rules for covariance and the correlation between fitted values and residuals.*]

   Solution:

   Let $(Y, \hat{Y})$ be the random vector which records response and fitted value when one of the data rows is picked at random. As discussed in MAST90105, expectation for functions of $(Y, \hat{Y})$ are

just simple arithmetic averages. Writing $E = Y - \hat{Y}$ for the residual,.the square of the correlation between the fitted and data values is:

$$\begin{aligned}\rho^2(Y,\hat{Y}) &= \frac{cov^2(Y,\hat{Y})}{var(Y)var(\hat{Y})} \\ &= \frac{cov^2(\hat{Y}+E,\hat{Y})}{var(Y)var(\hat{Y})}.\end{aligned}$$

Since the residuals and fitted values are orthogonal to each other,

$$\begin{aligned}cov(\hat{Y}+E,\hat{Y}) &= cov(\hat{Y},\hat{Y}) + cov(E,\hat{Y}) \\ &= var(\hat{Y}),\end{aligned}$$

so

$$\begin{aligned}\rho^2(Y,\hat{Y}) &= \frac{var^2(\hat{Y})}{var(Y)var(\hat{Y})} \\ &= \frac{var(\hat{Y})}{var(Y)}.\end{aligned}$$

On the other hand, R^2 is also equal to this ratio because

$$\begin{aligned}R^2 &= 1 - \frac{SS_{Res}}{SS_{Total} - (\mathbf{1}^T\mathbf{y})^2/n} \\ &= \frac{SS_{Total} - (\mathbf{1}^T\mathbf{y})^2/n - SS_{Res}}{SS_{Total} - (\mathbf{1}^T\mathbf{y})^2/n} \\ &= \frac{SS_{Reg} - (\mathbf{1}^T\mathbf{y})^2/n}{SS_{Total} - (\mathbf{1}^T\mathbf{y})^2/n}\end{aligned}$$

and this is the ratio of variances using the previous question and dividing both numerator and denominator by n.

5. Show that the adjusted R^2 satisfies:

$$\text{adjusted } R^2 = 1 - \frac{\text{estimate of } \sigma^2 \text{using the model}}{\text{estimate of } \sigma^2 \text{assuming equal means}}$$

Solution:

$$\begin{aligned}\text{adjusted } R^2 &= 1 - \frac{n-1}{n-p}(1-R^2) \\ &= 1 - \frac{n-1}{n-p} \times \frac{SS_{Res}}{SS_{Total} - (\mathbf{1}^T\mathbf{y})^2/n} \\ &= 1 - \frac{SS_{Res}/(n-p)}{(SS_{Total} - (\mathbf{1}^T\mathbf{y})^2/n)/(n-1)},\end{aligned}$$

The numerator is the estimate of σ^2 using the model, and the denominator is the estimate of σ^2 assuming equal means.

As discussed in class, the rest of the items will be deferred to next week, so that you can concentrate on preparation of the material for the first exam.