HW4 STAT425

Aldo Sanjoto

Friday, October 27, 2017

```
library("alr4")
```

```
## Loading required package: car
```

```
## Loading required package: effects
```

```
##
## Attaching package: 'effects'
```

```
## The following object is masked from 'package:car':
##
##     Prestige
```

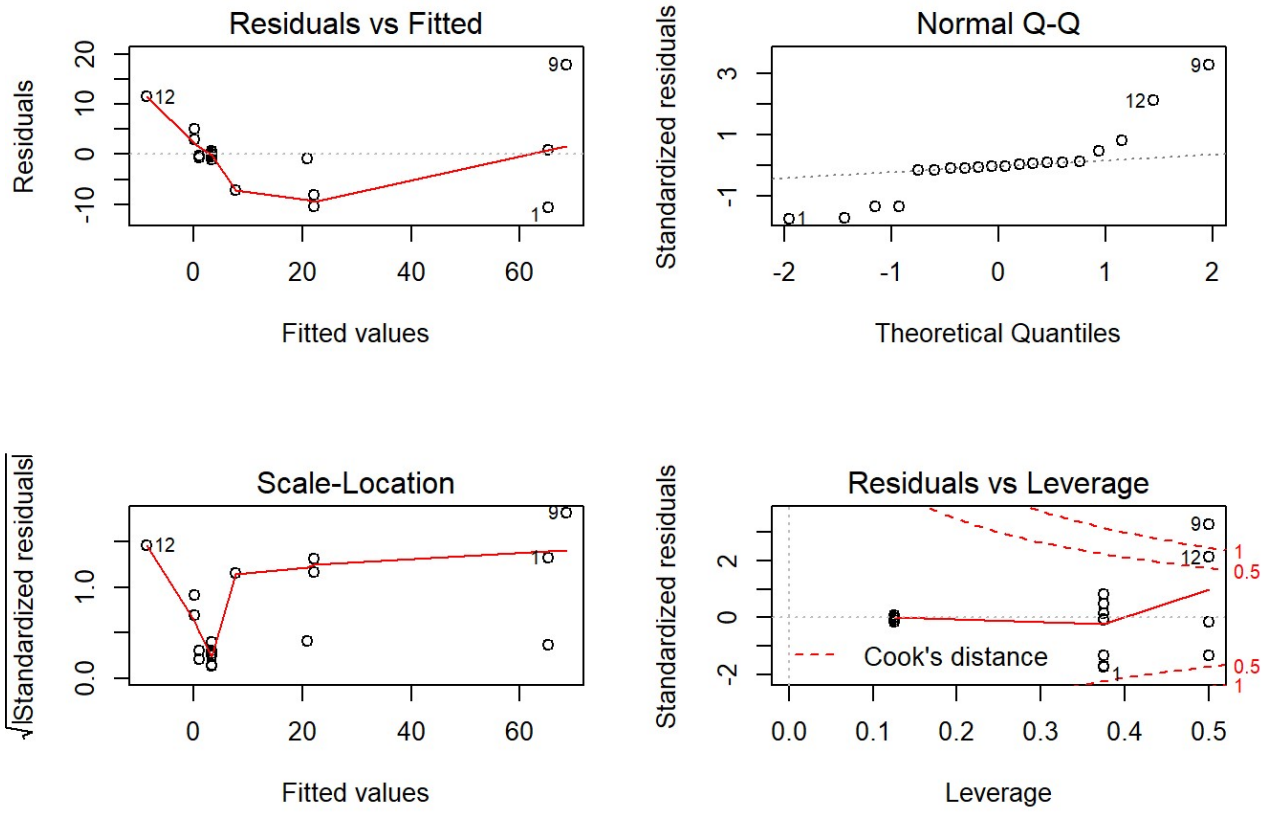
```
data("lathe1")
#head(lathe1)
fit_sop = lm(formula = Life ~ Speed*Feed + I(Feed^2) + I(Speed^2), data = lathe1)
summary(fit_sop)
```

```
##
## Call:
## lm(formula = Life ~ Speed * Feed + I(Feed^2) + I(Speed^2), data = lathe1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6601  -0.9607  -0.1383   0.7062  17.9193
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.338      2.733   1.222 0.241998
## Speed        -21.548      2.231  -9.657 1.44e-07 ***
## Feed         -10.494      2.231  -4.703 0.000339 ***
## I(Feed^2)      1.412      2.617   0.540 0.597837
## I(Speed^2)    17.392      2.617   6.647 1.10e-05 ***
## Speed:Feed    10.975      2.733   4.016 0.001274 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.729 on 14 degrees of freedom
## Multiple R-squared:  0.9267, Adjusted R-squared:  0.9005
## F-statistic:  35.4 on 5 and 14 DF,  p-value: 1.831e-07
```

1a) The interation term appear to be significant at 5% level.

1b) The diagnostic plots for fit_sop:

```
par(mfrow=c(2,2))
plot(fit_sop, cex = 1)
```



1c)

-As we can see in the Residual vs Fitted plot, the trend is not flat suggesting that it is non-linear and non-constant variance (heteroscedasticity).

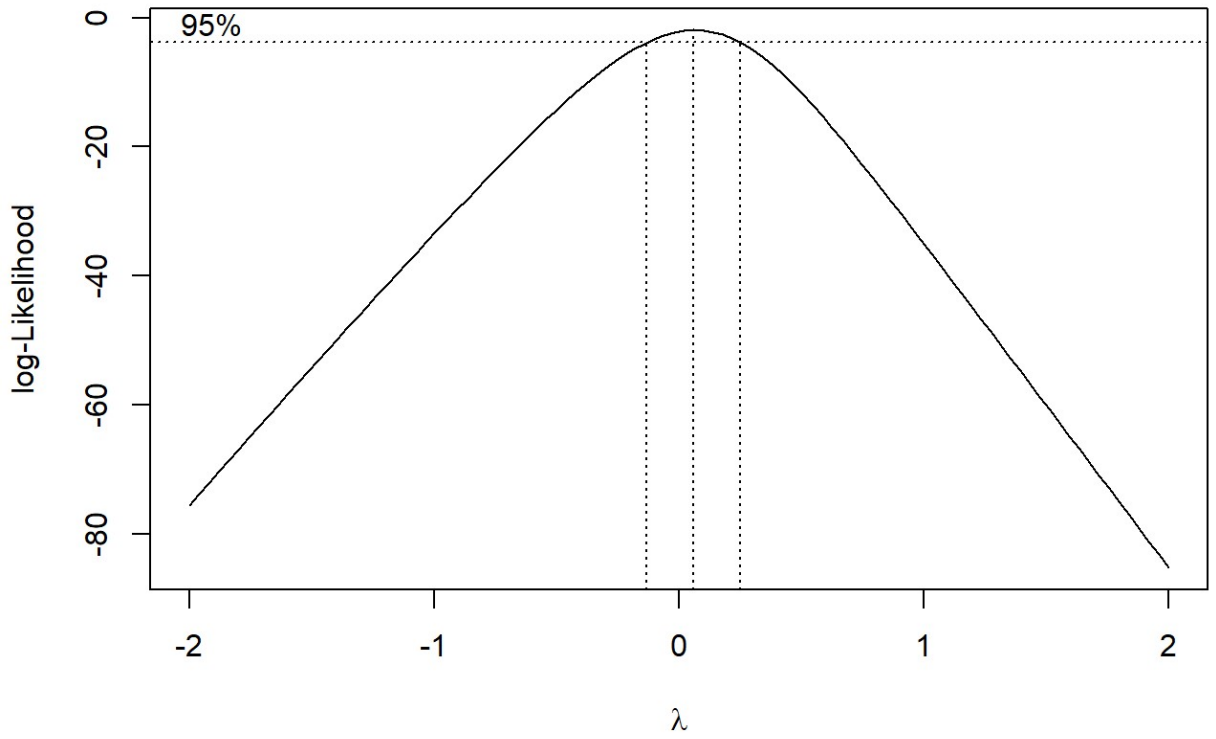
-In the Normal Q-Q plot, the points do not approximate a straight line suggesting there's a problem.

-In Scale-Location plot, the trend is not flat, suggesting there are problems with variance (assumption is false for homoscedasticity)

-In the Residual vs Leverage plot, we can see that the 9th and 12th observation are influential points.

1d) Box-cox log likelihood versus lambda plot:

```
library("MASS")
boxcox(fit_sop)
```




```
bc = boxcox(fit_sop, plotit = FALSE)
bc$x[which.max(bc$y)]
```

```
## [1] 0.1
```

1e) The lambda value selected by the Box-cox procedure is approximately 0.1

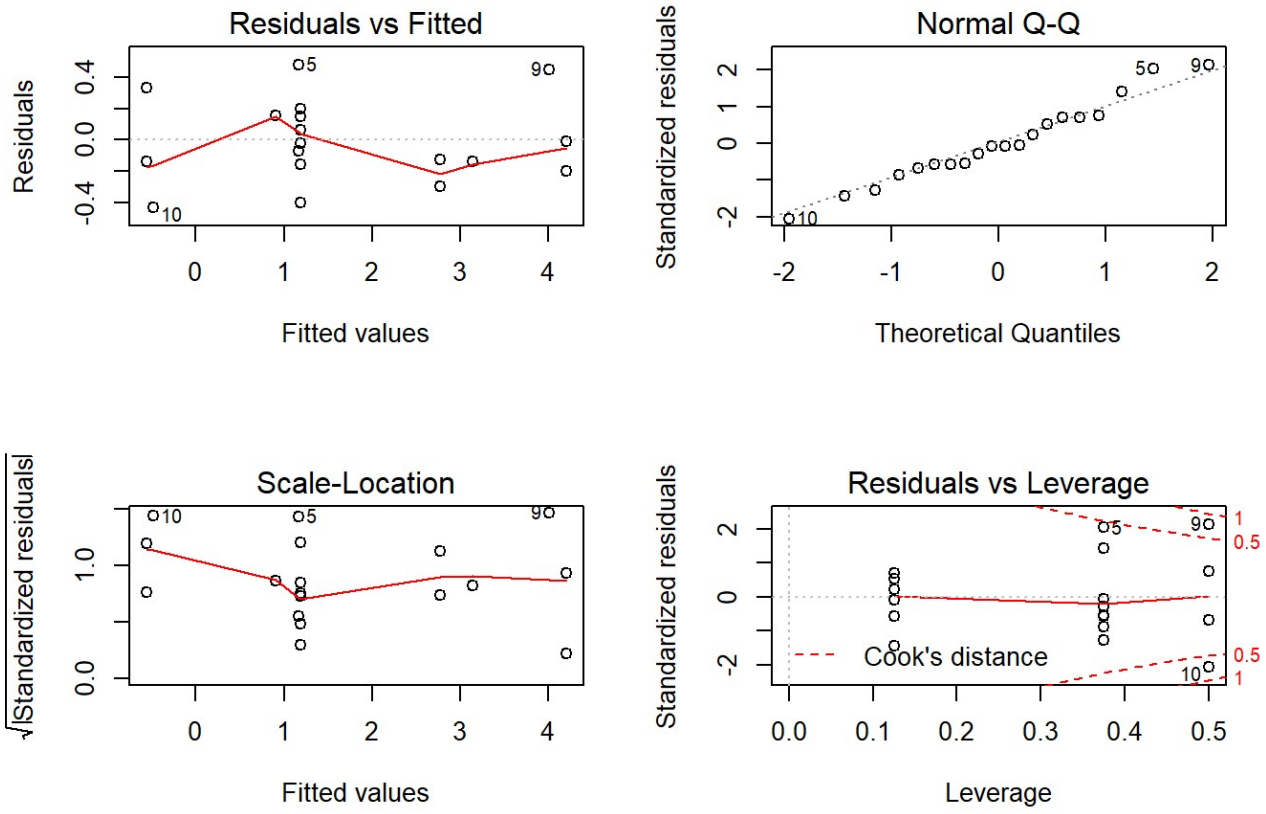
1f) The most “simple” lambda value that is still within the confidence interval limits shown in the box-cox plot is 0. This corresponds to y -> ln(y) which is a log-transformation.

```
fit_log = lm(formula = log(Life) ~ Speed*Feed + I(Feed^2) + I(Speed^2), data = lathe1)
summary(fit_log)
```

```
##
## Call:
## lm(formula = log(Life) ~ Speed * Feed + I(Feed^2) + I(Speed^2),
##     data = lathe1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43349 -0.14576 -0.02494  0.16748  0.47992
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.18809    0.10508  11.307 2.00e-08 ***
## Speed       -1.58902    0.08580 -18.520 3.04e-11 ***
## Feed        -0.79023    0.08580  -9.210 2.56e-07 ***
## I(Feed^2)    0.41851    0.10063   4.159 0.000964 ***
## I(Speed^2)   0.28808    0.10063   2.863 0.012529 *
## Speed:Feed  -0.07286    0.10508  -0.693 0.499426
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2972 on 14 degrees of freedom
## Multiple R-squared:  0.9702, Adjusted R-squared:  0.9596
## F-statistic: 91.24 on 5 and 14 DF,  p-value: 3.551e-10
```

1g) The interaction term appear not to be significant at 5% level.

```
par(mfrow=c(2,2))
plot(fit_log, cex = 1)
```



1h) From the above log-transformation diagnostic plot, we can observe that Residuals vs Fitted and Scale-location plots trend are roughly flat suggesting linearity and homoscedasticity (constant variance). The normal Q-Q points are forming a straight line suggesting no problems. Therefore, it has improved. Additionally, as we can see in the Residual vs Leverage plot, there are still influential points; 9th and 10th.

2a) We make each of the variable to log base 10 which is equal to ln. By doing this, multiplication becomes addition. And power comes down in front of a variable…

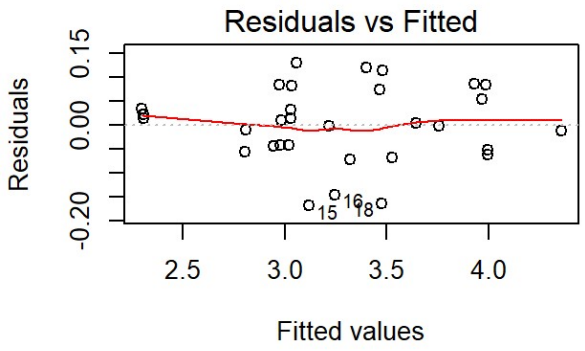
The formula then becomes *ln(Volume) = ln(lowercase(gamma)) + beta1ln(Girth) + beta2ln(height) + ln(e)*

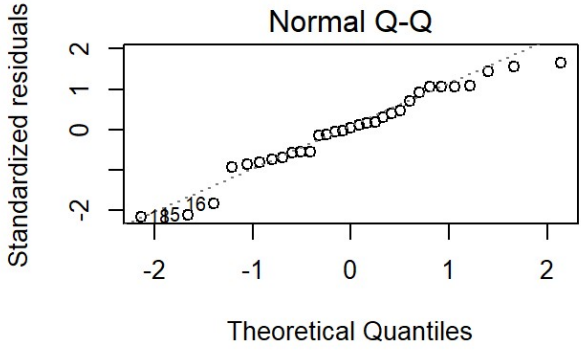
2b) Summary of linearized model

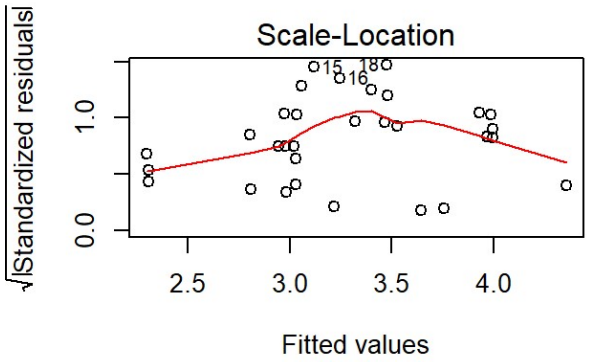
```
data("trees")
#head(trees)
fit_linearized_tree = lm(formula = log(Volume) ~ log(Girth) + log(Height), data = tree
s)
summary(fit_linearized_tree)
```

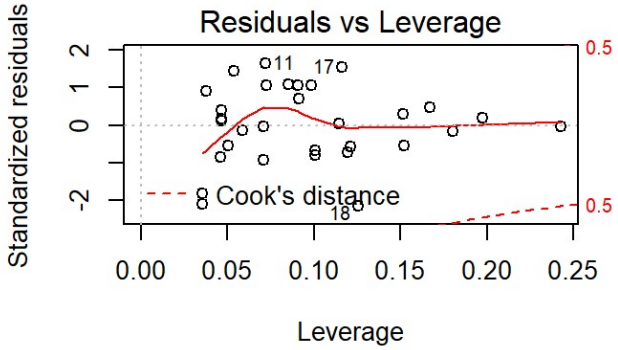
```
##
## Call:
## lm(formula = log(Volume) ~ log(Girth) + log(Height), data = trees)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.168561 -0.048488  0.002431  0.063637  0.129223
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.63162    0.79979  -8.292 5.06e-09 ***
## log(Girth)   1.98265    0.07501  26.432  < 2e-16 ***
## log(Height)  1.11712    0.20444   5.464 7.81e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08139 on 28 degrees of freedom
## Multiple R-squared:  0.9777, Adjusted R-squared:  0.9761
## F-statistic: 613.2 on 2 and 28 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(fit_linearized_tree, cex = 1)
```









2c) The plot of Residuals vs Fitted and Scale-location show a roughly flat trend indicating linearity and homoscedasticity. Normal Q-Q plot also approximates into a straight line indicating no problem. As we can see, there are no influential points.

```
confint(fit_linearized_tree)
```

```
##                 2.5 %    97.5 %
## (Intercept) -8.269912 -4.993322
## log(Girth)   1.828998  2.136302
## log(Height)  0.698353  1.535894
```

2d) As we can observe, the 95% CI for Girth and Height contains its slope theoretical values which are beta1=2, beta2=1

```
new_tree = data.frame(Girth = 10.9, Height = 75)
CI = predict(fit_linearized_tree, newdata = new_tree, interval = "prediction")
print(CI)
```

```
##       fit     lwr    upr
## 1 2.92763 2.75656 3.0987
```

2e) Prediction value: 2.92763, with interval (2.75656, 3.0987)

```
ori_CI = exp(CI)
print(ori_CI)
```

```
##       fit      lwr     upr
## 1 18.6833 15.74559 22.1691
```

2f) Prediction value: 18.6833, with inteval (15.74559, 22.1691)

3a) Using Forward selection with Fin = 3, we have SSF and Sex as the final variables.

```
data("ais")
#head(ais)
possible_pred = ~ Sex + Ht + Wt + LBM + BMI + SSF

fit_forward = lm(formula = Bfat ~ 1, data = ais)
add1(fit_forward, possible_pred, test = "F")
```

```
## Single term additions
##
## Model:
## Bfat ~ 1
##        Df Sum of Sq    RSS    AIC   F value    Pr(>F)
## <none>              7701.1 737.45
## Sex     1    3733.1 3968.0 605.51  188.1568 < 2.2e-16 ***
## Ht      1     272.3 7428.9 732.18    7.3295  0.007370 **
## Wt      1       0.0 7701.1 739.45    0.0000  0.998176
## LBM     1    1008.3 6692.8 711.10   30.1326 1.214e-07 ***
## BMI     1     270.9 7430.2 732.22    7.2921  0.007519 **
## SSF     1    7142.0  559.1 209.64 2554.8760 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit_forward = update(fit_forward, . ~ . + SSF)
add1(fit_forward, possible_pred, test = "F")
```

```
## Single term additions
##
## Model:
## Bfat ~ SSF
##        Df Sum of Sq    RSS     AIC F value    Pr(>F)
## <none>              559.09 209.644
## Sex     1    315.09 244.00  44.155 256.984 < 2.2e-16 ***
## Ht      1    110.36 448.73 167.228  48.940 3.929e-11 ***
## Wt      1    174.40 384.69 136.123  90.216 < 2.2e-16 ***
## LBM     1    210.66 348.43 116.122 120.318 < 2.2e-16 ***
## BMI     1    127.14 431.95 159.529  58.572 8.278e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit_forward = update(fit_forward, . ~ . + Sex)
add1(fit_forward, possible_pred, test = "F")
```

```
## Single term additions
##
## Model:
## Bfat ~ SSF + Sex
##        Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>              244.00 44.155
## Ht      1   0.61479 243.38 45.646  0.5002 0.4803
## Wt      1   0.26549 243.73 45.935  0.2157 0.6429
## LBM     1   0.79043 243.21 45.500  0.6435 0.4234
## BMI     1   0.04465 243.95 46.118  0.0362 0.8492
```

3b) Using Backward selection with Fout = 3, we have Sex, Ht, Wt, LBM, SSF as the final variables.

```
fit_backward = lm(Bfat ~ Sex + Ht + Wt + LBM + BMI + SSF, data = ais)
drop1(fit_backward, test="F")
```

```
## Single term deletions
##
## Model:
## Bfat ~ Sex + Ht + Wt + LBM + BMI + SSF
##        Df Sum of Sq    RSS      AIC  F value    Pr(>F)
## <none>              105.03 -118.120
## Sex     1    22.694 127.72  -80.603  42.1354 6.888e-10 ***
## Ht      1     1.719 106.74 -116.842   3.1909    0.0756 .
## Wt      1    59.652 164.68  -29.264 110.7556 < 2.2e-16 ***
## LBM     1   136.496 241.52   48.096 253.4299 < 2.2e-16 ***
## BMI     1     0.695 105.72 -118.788   1.2907    0.2573
## SSF     1    24.310 129.34  -78.064  45.1351 1.969e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit_backward = update(fit_backward, .~. - BMI)
drop1(fit_backward, test="F")
```

```
## Single term deletions
##
## Model:
## Bfat ~ Sex + Ht + Wt + LBM + SSF
##        Df Sum of Sq    RSS      AIC  F value    Pr(>F)
## <none>              105.72 -118.788
## Sex     1    22.163 127.88  -82.343  41.0886 1.061e-09 ***
## Ht      1     4.136 109.86 -113.035   7.6687  0.006158 **
## Wt      1   134.239 239.96   44.786 248.8715 < 2.2e-16 ***
## LBM     1   137.661 243.38   47.646 255.2150 < 2.2e-16 ***
## SSF     1    24.709 130.43  -78.362  45.8081 1.474e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3c) Using Mallow Cp selection, we have Sex, Ht, Wt, LBM, SSF as the final variables.

```
#install.packages("leaps")
library(leaps)
x = model.matrix(Bfat ~ Sex + Ht + Wt + LBM + BMI + SSF - 1, data = ais)
y = ais$Bfat
cp_mod = leaps(x, y, nbest=1)
cp_mod
```

```
## $which
##       1     2     3     4     5     6
## 1 FALSE FALSE FALSE FALSE FALSE  TRUE
## 2 FALSE FALSE  TRUE  TRUE FALSE FALSE
## 3 FALSE FALSE  TRUE  TRUE FALSE  TRUE
## 4  TRUE FALSE  TRUE  TRUE FALSE  TRUE
## 5  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
## 6  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##
## $label
## [1] "(Intercept)" "1"           "2"           "3"           "4"
## [6] "5"           "6"
##
## $size
## [1] 2 3 4 5 6 7
##
## $Cp
## [1] 840.052675  81.674247  52.218643  11.970682   6.290651   7.000000
```

3d) Using Stepwise selection we have Sex and SSF as the final variables.

```
step(object= lm(Bfat ~ 1, data=ais), scope= ~ Sex + Ht + Wt + LBM + BMI + SSF, directi
on= "both")
```

```
## Start:  AIC=737.45
## Bfat ~ 1
##
##        Df Sum of Sq    RSS    AIC
## + SSF   1    7142.0  559.1 209.64
## + Sex   1    3733.1 3968.0 605.51
## + LBM   1    1008.3 6692.8 711.10
## + Ht    1     272.3 7428.9 732.18
## + BMI   1     270.9 7430.2 732.22
## <none>              7701.1 737.45
## + Wt    1       0.0 7701.1 739.45
##
## Step:  AIC=209.64
## Bfat ~ SSF
##
##        Df Sum of Sq    RSS    AIC
## + Sex   1     315.1  244.0  44.16
## + LBM   1     210.7  348.4 116.12
## + Wt    1     174.4  384.7 136.12
## + BMI   1     127.1  432.0 159.53
## + Ht    1     110.4  448.7 167.23
## <none>               559.1 209.64
## - SSF   1    7142.0 7701.1 737.45
##
## Step:  AIC=44.16
## Bfat ~ SSF + Sex
##
##        Df Sum of Sq    RSS    AIC
## <none>               244.0  44.16
## + LBM   1       0.8  243.2  45.50
## + Ht    1       0.6  243.4  45.65
## + Wt    1       0.3  243.7  45.94
## + BMI   1       0.0  244.0  46.12
## - Sex   1     315.1  559.1 209.64
## - SSF   1    3724.0 3968.0 605.51
```

```
##
## Call:
## lm(formula = Bfat ~ SSF + Sex, data = ais)
##
## Coefficients:
## (Intercept)          SSF          Sex
##      1.1307       0.1579       2.9844
```