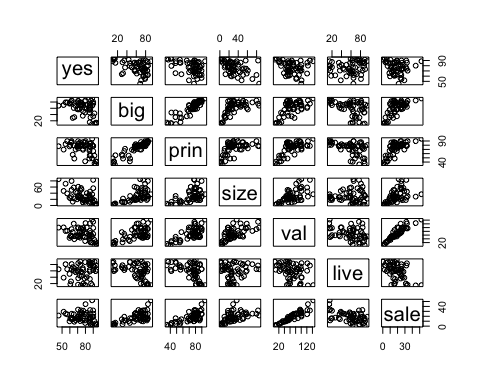
MAST30025 LAB 7 Q2-3

#Question 2: #Part a) Using pairs to plot the data. Is there any evidence of non-linearity or heteroskedasticity?

setwd("~/Desktop/UNIMELB 2021 Material/UNIMELB S1 2021 (Currently)/MAST30025/Tutorials /Tutorials/Rfile/data")  
beef = read.csv("beef.csv")

pairs(beef)

 #There is non-linearity between big vs. yes however there is heteroskedasticity in yes. We can take the logs on yes and live for example. And show that it is a best fit. (This from my understanding) Dunno if we continue to do this for all non linear models from we were given in the pairs model. Because there are other transformations depending on what non linear model you are trying to apply into a best fit model.

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

#Forward selection

str(beef)

## 'data.frame': 56 obs. of 7 variables:  
## $ yes : num 85.9 65 74.9 72.5 76.6 62.3 76.9 80.1 70.7 87.2 ...  
## $ big : num 64.6 61 70.8 57 36 84.2 50.8 85 56.3 84.4 ...  
## $ prin: num 78.4 79.9 79.6 79.2 71.5 87.2 69 89.6 72.5 85.4 ...  
## $ size: num 45.2 58.5 52.5 24 10.5 50.6 17.4 28.8 60.9 21.7 ...  
## $ val : int 122 121 81 74 54 51 88 124 101 60 ...  
## $ live: num 83.9 60.2 43.1 45.4 68.8 81.4 64.1 14.2 84.4 20 ...  
## $ sale: num 32.7 24.2 25.5 25.1 11.6 14.6 16.1 44.6 19.6 16.1 ...

basemodel = lm(beef$yes~1, data = beef)  
add1(basemodel,scope = ~. + beef$big + beef$prin + beef$size+beef$val+beef$live+beef$sale, test = "F")

## Single term additions  
##   
## Model:  
## beef$yes ~ 1  
## Df Sum of Sq RSS AIC F value Pr(>F)   
## <none> 7689.2 277.64   
## beef$big 1 1482.29 6206.9 267.65 12.8960 0.0007112 \*\*\*  
## beef$prin 1 1288.87 6400.3 269.37 10.8744 0.0017287 \*\*   
## beef$size 1 1925.80 5763.4 263.50 18.0439 8.572e-05 \*\*\*  
## beef$val 1 538.85 7150.3 275.57 4.0694 0.0486484 \*   
## beef$live 1 226.19 7463.0 277.97 1.6366 0.2062603   
## beef$sale 1 214.20 7475.0 278.06 1.5474 0.2188955   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

basemodel1 = lm(beef$yes~beef$size, data = beef)  
add1(basemodel1,scope = ~. + beef$big + beef$prin +beef$val+beef$live+beef$sale, test = "F")

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

#Should have end here since Pr(>F) values are all above 0.05!!!!

#Opps :( oh dear!!

basemodel2 = lm(beef$yes~beef$size + beef$live, data = beef)  
add1(basemodel2,scope = ~. + beef$big + beef$prin +beef$val+beef$sale, test = "F")

## Single term additions  
##   
## Model:  
## beef$yes ~ beef$size + beef$live  
## Df Sum of Sq RSS AIC F value Pr(>F)   
## <none> 5415.3 262.01   
## beef$big 1 15.35 5400.0 263.85 0.1479 0.70216   
## beef$prin 1 16.75 5398.6 263.84 0.1613 0.68958   
## beef$val 1 332.68 5082.7 260.46 3.4036 0.07075 .  
## beef$sale 1 622.57 4792.8 257.17 6.7546 0.01214 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

basemodel3 = lm(beef$yes~beef$size + beef$live +beef$sale , data = beef)  
add1(basemodel3,scope = ~. + beef$big + beef$prin +beef$val, test = "F")

## Single term additions  
##   
## Model:  
## beef$yes ~ beef$size + beef$live + beef$sale  
## Df Sum of Sq RSS AIC F value Pr(>F)  
## <none> 4792.8 257.17   
## beef$big 1 92.275 4700.5 258.08 1.0012 0.3217  
## beef$prin 1 198.741 4594.0 256.80 2.2063 0.1436  
## beef$val 1 50.005 4742.8 258.58 0.5377 0.4667

#We use size, live and sale in our final model! Taken 5% significant level!!

#ACTUAL ANSWER!! Look above!!! #We use beef size in the final model!!

#Now we do it for the backward elimination!!

fullmodel = lm(yes~., data = beef)  
drop1(fullmodel, 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

model2 = lm(yes~ prin+size+val+live+sale, data = beef)  
drop1(model2, 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

model3 = lm(yes~ prin+size+live+sale, data = beef)  
drop1(model3, 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

model4 = lm(yes~ size+live+sale, data = beef)  
drop1(model4, 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

#Were using size, live and sale in our final model!! #Actual answer –> I did above in the first attempt!

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

model2 = step(basemodel, scope = ~.+big+prin+size+val+live+sale) #Removing step = 1 reveals mutiple steps!!

## Start: AIC=277.64  
## beef$yes ~ 1  
##   
## Df Sum of Sq RSS AIC  
## + size 1 1925.80 5763.4 263.50  
## + big 1 1482.29 6206.9 267.65  
## + prin 1 1288.87 6400.3 269.37  
## + val 1 538.85 7150.3 275.57  
## <none> 7689.2 277.64  
## + live 1 226.19 7463.0 277.97  
## + sale 1 214.20 7475.0 278.06  
##   
## Step: AIC=263.5  
## beef$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  
## beef$yes ~ size + live  
##   
## Df Sum of Sq RSS AIC  
## + sale 1 622.57 4792.8 257.17  
## + val 1 332.68 5082.7 260.46  
## <none> 5415.3 262.01  
## - live 1 348.02 5763.4 263.50  
## + prin 1 16.75 5398.6 263.84  
## + big 1 15.35 5400.0 263.85  
## - size 1 2047.63 7463.0 277.97  
##   
## Step: AIC=257.17  
## beef$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  
## beef$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

## 

lm(formula = yes ~ prin + live + sale + size, data = beef)

##   
## Call:  
## lm(formula = yes ~ prin + live + sale + size, data = beef)  
##   
## Coefficients:  
## (Intercept) prin live sale size   
## 81.8567 -0.2031 0.1728 0.5795 -0.3710

#Part d) Show that the model found in 2c 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.

model3 = step(basemodel, scope = ~.+big+prin+size+val+live+sale+size\*sale)

## Start: AIC=277.64  
## beef$yes ~ 1  
##   
## Df Sum of Sq RSS AIC  
## + size 1 1925.80 5763.4 263.50  
## + big 1 1482.29 6206.9 267.65  
## + prin 1 1288.87 6400.3 269.37  
## + val 1 538.85 7150.3 275.57  
## <none> 7689.2 277.64  
## + live 1 226.19 7463.0 277.97  
## + sale 1 214.20 7475.0 278.06  
##   
## Step: AIC=263.5  
## beef$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  
## beef$yes ~ size + live  
##   
## Df Sum of Sq RSS AIC  
## + sale 1 622.57 4792.8 257.17  
## + val 1 332.68 5082.7 260.46  
## <none> 5415.3 262.01  
## - live 1 348.02 5763.4 263.50  
## + prin 1 16.75 5398.6 263.84  
## + big 1 15.35 5400.0 263.85  
## - size 1 2047.63 7463.0 277.97  
##   
## Step: AIC=257.17  
## beef$yes ~ size + live + sale  
##   
## Df Sum of Sq RSS AIC  
## + size:sale 1 782.10 4010.7 249.20  
## + 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=249.2  
## beef$yes ~ size + live + sale + size:sale  
##   
## Df Sum of Sq RSS AIC  
## <none> 4010.7 249.20  
## + big 1 77.94 3932.7 250.10  
## + val 1 43.83 3966.8 250.58  
## + 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

#In this case our final model includes live and size\*sale

#yes = beta(intercept) + beta1*live + beta2*size\*sale + e

#Actual answer IMPORTANT NOTE: Forgot to change the base model this then becomes my new model!!

newmodel = lm(yes~big+prin+size+val+live+sale+size\*sale, data = beef)  
model3 = step(newmodel, scope = ~.)

## Start: AIC=253.76  
## yes ~ big + prin + size + val + live + sale + size \* sale  
##   
## Df Sum of Sq RSS AIC  
## - prin 1 8.05 3916.9 251.87  
## - val 1 9.78 3918.6 251.90  
## - big 1 50.54 3959.4 252.47  
## <none> 3908.8 253.76  
## - live 1 608.72 4517.6 259.86  
## - size:sale 1 631.88 4540.7 260.15  
##   
## Step: AIC=251.87  
## yes ~ big + size + val + live + sale + size:sale  
##   
## Df Sum of Sq RSS AIC  
## - val 1 15.85 3932.7 250.10  
## - big 1 49.95 3966.8 250.58  
## <none> 3916.9 251.87  
## + prin 1 8.05 3908.8 253.76  
## - live 1 625.57 4542.5 258.17  
## - size:sale 1 687.97 4604.9 258.93  
##   
## Step: AIC=250.1  
## yes ~ big + size + live + sale + size:sale  
##   
## Df Sum of Sq RSS AIC  
## - big 1 77.94 4010.7 249.20  
## <none> 3932.7 250.10  
## + val 1 15.85 3916.9 251.87  
## + prin 1 14.12 3918.6 251.90  
## - live 1 635.83 4568.6 256.49  
## - size:sale 1 767.76 4700.5 258.08  
##   
## Step: AIC=249.2  
## yes ~ size + live + sale + size:sale  
##   
## Df Sum of Sq RSS AIC  
## <none> 4010.7 249.20  
## + big 1 77.94 3932.7 250.10  
## + val 1 43.83 3966.8 250.58  
## + 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

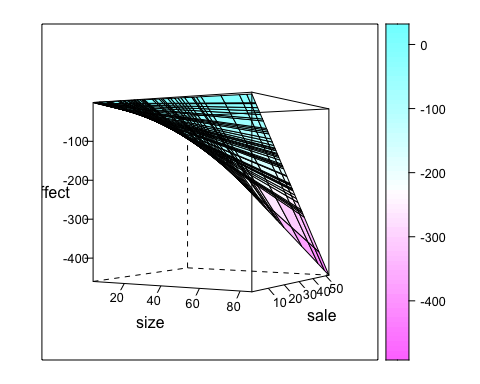
#In comparsion the AIC from Q2c was 256.80 here the AIC in Q2d with size\*sale is 249.20. #removing prin would improve the AIC in the final model! for this part of the question!—>ASK

#yes = beta0 + beta1*size + beta2*sale + beta3*live + beta12*size\*sale + e

#Question 2e) Suppose that beta1,beta2 and beta3 are the coefficients of x1 = size, x2 = sale and x12 = size*sale (interaction term). In the model from 2d.Plot beta1*x1 + beta2*x2 + beta12*x1\*x2 as a function of (x1,x2), 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.

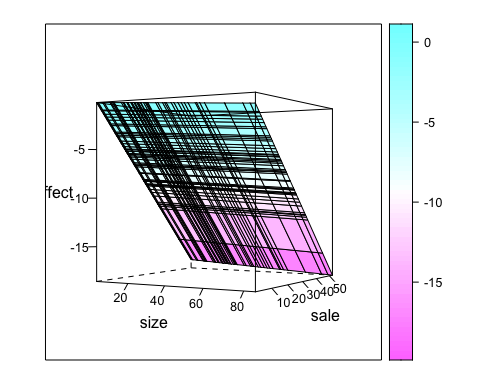
#ASK: Why we didn’t use the lattice library and expand.grid example in the lectures!!

library(lattice)  
model5 = lm(yes~big+prin+size+val+live+sale, data = beef)  
df = expand.grid(size = beef$size, sale = beef$sale)  
f = function(x,y)  
sum(model5$coefficients[c(2,4,5)]\*c(x,y,x\*y))  
df$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))



#Question 2f) Repeat the above question using the model with no size\*sale interaction term from 2c.

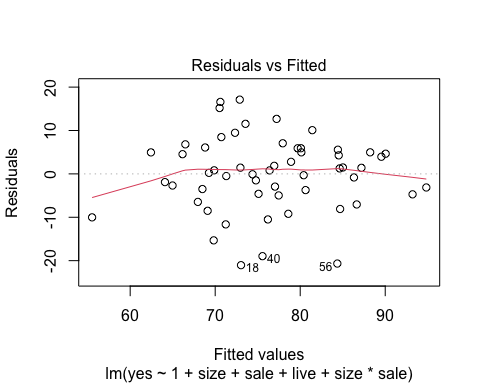
df = expand.grid(size = beef$size, sale = beef$sale)  
f = function(x,y) sum(model5$coefficients[c(2,4)]\*c(x,y))  
df$effect = mapply(f, df$size, df$sale)  
wireframe(effect ~ size + sale, data = df, drape = T, scales = list(arrows = F), screen = list(z=30, x = -90, y = -60))



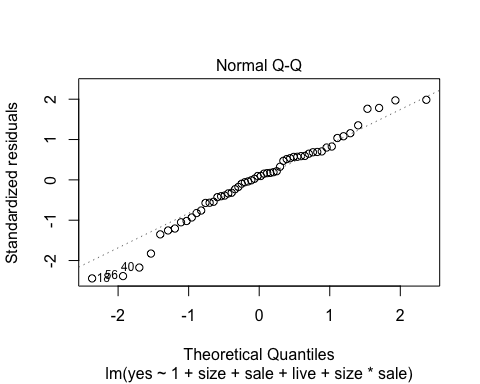
#Question 2g) Use the diagnostic plots provided by R to assess the model from 2d. Refer back to 2a; do you need to transform the data and start again?

#yes = beta0 + beta1*size + beta2*sale + beta3*live + beta12*size\*sale + e

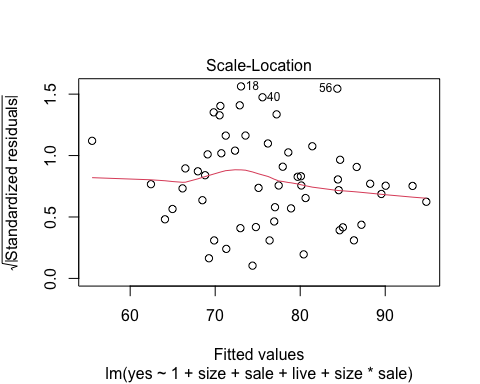
model2g = lm(yes~ 1 + size + sale + live + size\*sale, data = beef)  
plot(model2g, which = 1)



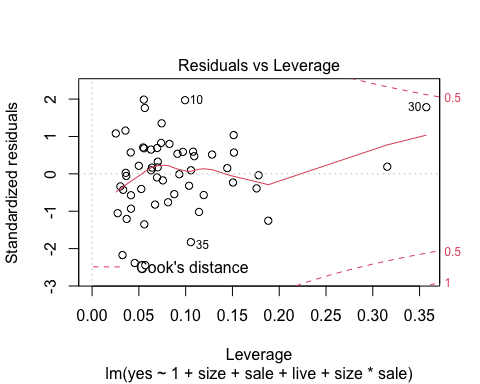
plot(model2g, which = 2)



plot(model2g, which = 3)



plot(model2g, which = 5)



#Depends on our response variable since we only choosen yes, we could have chosen a different responsible variable. But we only interested in transforming non linearity to a linear model (into a best fit)!! This means we have to keep changing our response and design variables for big, prin,size,val,live and sale!!!

#There is some evidence of heteroskedasticity in the plot of the square root of the absolute standardised residuals against fitted values.

#Question 2h) 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$size)\*model3$coefficients[2]

## size   
## -27.85712

mean(beef$live)\*model3$coefficients[3]

## live   
## 10.38959

mean(beef$sale)\*model3$coefficients[4]

## sale   
## -3.225588

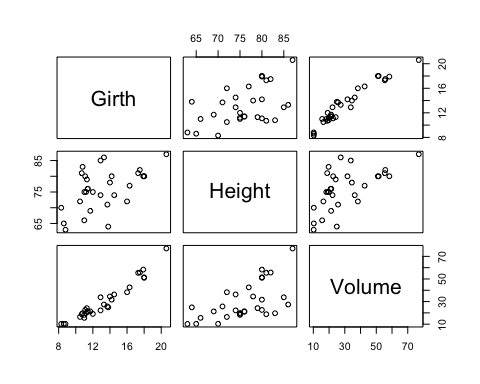
mean(beef$size\*beef$sale)\*model3$coefficients[5]

## size:sale   
## 16.12498

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

#Question 3 Load & Examine the dataset trees using

data("trees")  
?trees  
pairs(trees)



#We will model the volume of the black tree as a function of its girth and height. #Part a. By calculating R(r1|r2) and SSRes from the data 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).

treesdata = trees  
y = trees$Volume  
X = cbind(1,trees$Girth,trees$Height)  
b = solve(t(X)%\*%X,t(X)%\*%y)  
n = length(y)  
p = dim(X)[2]  
SSTotal = sum(y^2)  
SSRes = sum((y-X%\*%b)^2)  
SSRes

## [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  
R\_g1g2

## [1] 102.3812

Fstat = (R\_g1g2/1)/(SSRes/(n-3))  
Fstat

## [1] 6.79433

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

## [1] 0.01449097

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.

model3 = lm(Volume~Girth+Height+Girth^2+Girth^2\*Height, data = trees)  
  
#Now using stepwise selection!  
model3 = step(model3,scope = ~.)

## Start: AIC=65.49  
## Volume ~ Girth + Height + Girth^2 + Girth^2 \* Height  
##   
## Df Sum of Sq RSS AIC  
## <none> 198.08 65.495  
## - Girth:Height 1 223.84 421.92 86.936

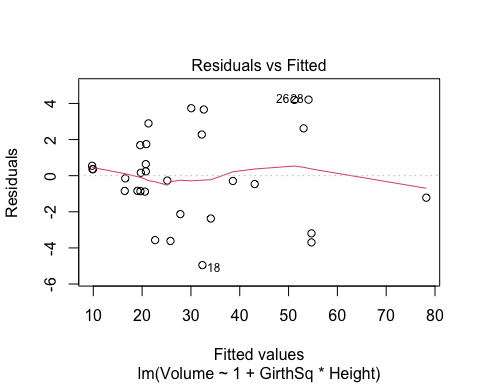
trees$GirthSq = trees$Girth^2 #Ensure to add to the trees dataframe!  
model3 = lm(Volume~Girth+Height+GirthSq+GirthSq\*Height, data = trees)  
  
#Now using stepwise selection!  
model3 = step(model3,scope = ~.)

## 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

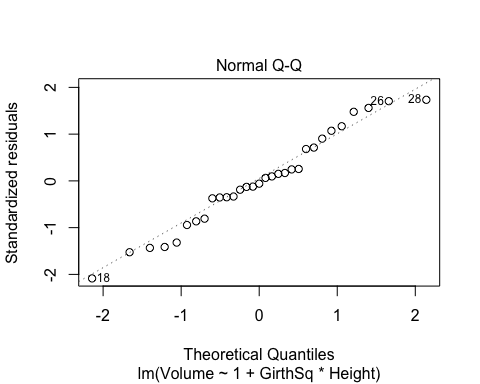
#Height\*GirthSq are in the final model!

#Part c: Use diagnostic plots to check the fit of your final model.

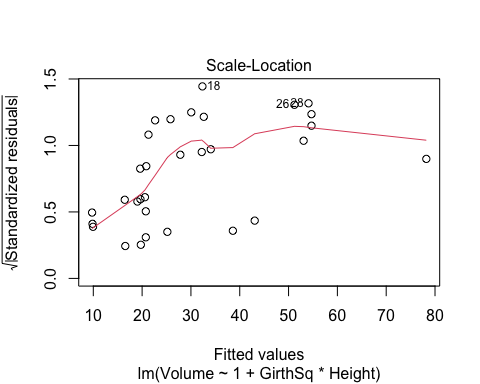
finalmodel = lm(Volume~1+ GirthSq\*Height,data= trees)  
plot(finalmodel, which= 1)



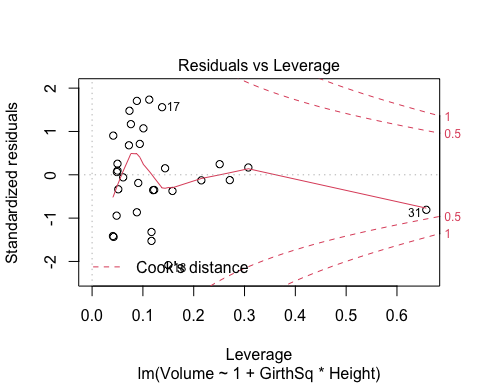
plot(finalmodel, which= 2)



plot(finalmodel, which= 3)



plot(finalmodel, which= 5)

 #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.)