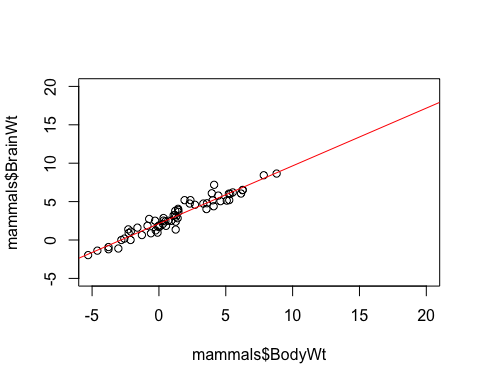
MAST30025 LAB 6 (WEEK 6)

#Questions 1–7 use the ‘sleep’ dataset, which you can download from the course website. This dataset contains (among other things) data on the body weight (kg) and brain weight (g) of 62 mammals. Use the following commands to read the data:

setwd("~/Desktop/UNIMELB S1 2021 (Currently)/MAST30025/Tutorials /Tutorials/Rfile/data")  
mammals <- read.csv("sleep.csv")   
mammals$BodyWt = log(mammals$BodyWt)  
mammals$BrainWt = log(mammals$BrainWt)

#Question 1. Fit a linear model explaining brain weight from body weight, using the lm command. #Display the summary of the fitted model, and then create a scatter plot of the data and superimpose #the fitted regression line on it. Does it look like a reasonable fit? Use diagnostic plots to assess if the model assumptions are satisfied.

plot(mammals$BodyWt,mammals$BrainWt,xlim = c(-5,20), ylim = c(-5,20))  
   
abline(lm(mammals$BrainWt~mammals$BodyWt),col = 'red')



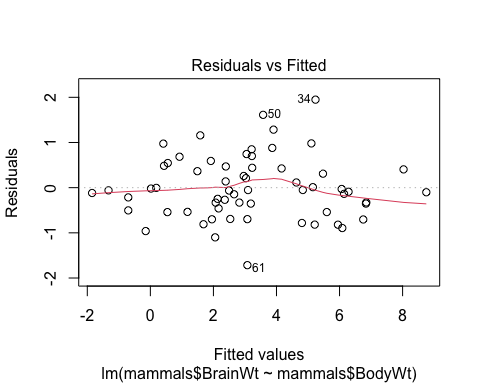
#Looks best fit

m = lm(mammals$BrainWt~mammals$BodyWt, data = mammals)  
m

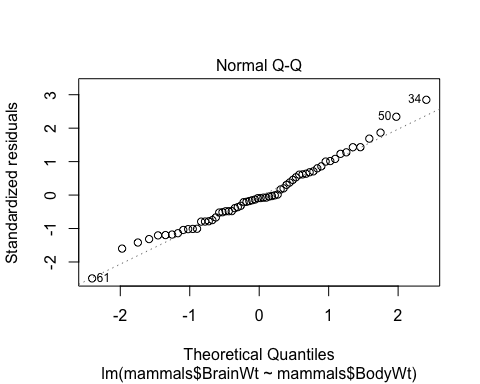
##   
## Call:  
## lm(formula = mammals$BrainWt ~ mammals$BodyWt, data = mammals)  
##   
## Coefficients:  
## (Intercept) mammals$BodyWt   
## 2.1348 0.7517

#Diagonistic plots

plot(m, which = 1)

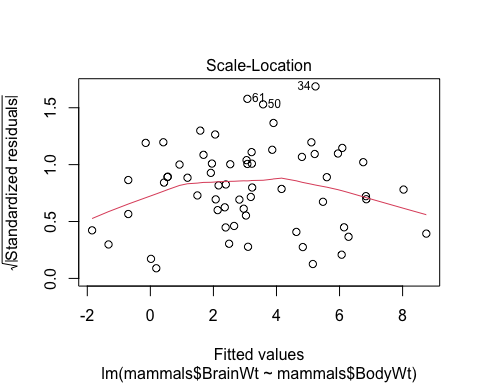
 #Some of the residuals appear more under the line than above!

plot(m, which = 2)



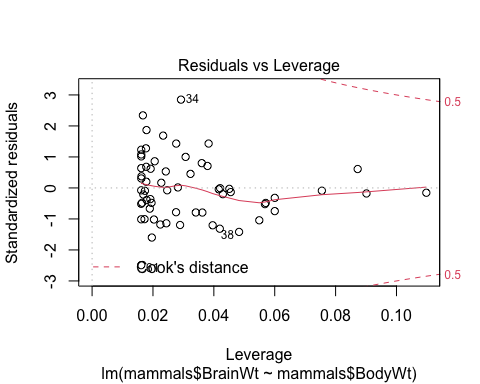
#Looks normally distributed and linear

plot(m, which = 3)



#There are many residuals over and under the line. Unable to determine how many they are in both sides!

plot(m, which = 5)

 #Seems decent alright!

#Question 2. Using the fitted model or otherwise, obtain:

#Case 1 and 2 without lm and with lm respectively!!

#(a) The least squares estimator of the parameters, b;

y = mammals$BrainWt  
X = matrix(c(rep(1,62),mammals$BodyWt),62,2)

#Solving for b

b = solve(t(X)%\*%X,t(X)%\*%y)  
b

## [,1]  
## [1,] 2.1347887  
## [2,] 0.7516859

#(b) The vector of residuals, e;

e = y - X%\*%b  
e

## [,1]  
## [1,] -0.101535517  
## [2,] -0.247719028  
## [3,] 0.744129330  
## [4,] -0.331645719  
## [5,] 0.404449017  
## [6,] 1.284319858  
## [7,] -0.503205892  
## [8,] -0.819826743  
## [9,] 0.210348946  
## [10,] 0.979582387  
## [11,] 0.364701495  
## [12,] -0.704299635  
## [13,] -0.809934274  
## [14,] -0.029470399  
## [15,] -0.005399721  
## [16,] 0.258275741  
## [17,] -0.700063920  
## [18,] 0.684441080  
## [19,] 0.469140656  
## [20,] -0.818000748  
## [21,] -0.326509689  
## [22,] 0.114555670  
## [23,] -0.541016395  
## [24,] -0.136965236  
## [25,] 0.309558036  
## [26,] -0.052012159  
## [27,] 0.974846992  
## [28,] -0.459522243  
## [29,] -0.352512768  
## [30,] -0.540184534  
## [31,] -0.781942122  
## [32,] -0.118230883  
## [33,] -0.059441375  
## [34,] 1.948289465  
## [35,] 0.545171049  
## [36,] -0.268509017  
## [37,] -0.215523820  
## [38,] -0.960916163  
## [39,] -0.693104837  
## [40,] -0.696926847  
## [41,] -0.090787805  
## [42,] 1.157765654  
## [43,] 0.879322620  
## [44,] -0.063808273  
## [45,] -0.893817157  
## [46,] -0.330346081  
## [47,] 0.439568332  
## [48,] -0.536064395  
## [49,] 0.700231550  
## [50,] 1.611673361  
## [51,] 0.591057154  
## [52,] -0.053126196  
## [53,] 0.425182038  
## [54,] 0.010938666  
## [55,] 0.138018519  
## [56,] -0.019987408  
## [57,] -1.100079214  
## [58,] -0.146218402  
## [59,] 0.482841228  
## [60,] 0.848713341  
## [61,] -1.715496428  
## [62,] -0.352971141

#(c) The residual sum of squares, SSRes;

SSRes = sum(e^2)  
SSRes

## [1] 28.92271

#(d) The regression sum of squares, SSReg;

SSTotal = sum(y^2)  
SSReg = SSTotal - SSRes  
SSReg

## [1] 947.5602

#(e) The regression sum of squares, SSReg;

s2 = SSRes/(62-2) # p = k + 1, hence p = 2   
s2

## [1] 0.4820452

#(f) The standardised residuals; #Using lm

str(rstandard(m))

## Named num [1:62] -0.155 -0.36 1.081 -0.482 0.61 ...  
## - attr(\*, "names")= chr [1:62] "1" "2" "3" "4" ...

#(g) The leverages of the points; # Using lm

str(lm.influence(m)$hat)

## Named num [1:62] 0.1098 0.0191 0.0162 0.0195 0.0873 ...  
## - attr(\*, "names")= chr [1:62] "1" "2" "3" "4" ...

# (h) The leverages of the points;

# Using lm

str(cooks.distance(m))

## Named num [1:62] 0.00148 0.00127 0.00958 0.00232 0.01777 ...  
## - attr(\*, "names")= chr [1:62] "1" "2" "3" "4" ...

#(i) 95% confidence intervals for each of the parameters. #Using lm

(More to be added)

#Without lm #For b0

c00 = solve(t(X)%\*%X)[1,1]  
s = sqrt(s2)  
alpha = 0.05  
ta <- qt(1-alpha/2, df=62-1) #One because one of the estimators is unbiased   
c(b[1] - ta\*s\*sqrt(c00), b[1] + ta\*s\*sqrt(c00))

## [1] 1.942738 2.326839

#For b1

c11 = solve(t(X)%\*%X)[2,2]  
s = sqrt(s2)  
alpha = 0.05  
ta = qt(1-alpha/2, df=62-1) # One because one of the estimators is unbiased   
c(b[2] - ta\*s\*sqrt(c11), b[2] + ta\*s\*sqrt(c11))

## [1] 0.6947695 0.8086023

#Question 3. Find a 95% confidence interval for a mammal weighing 50 kg

xst = c(1,log(50))  
xst%\*%b

## [,1]  
## [1,] 5.075401

#Lower Bound

xst%\*%b - ta\*s\*sqrt(t(xst)%\*%solve(t(X)%\*%X)%\*%xst)

## [,1]  
## [1,] 4.846144

#Upper Bound

xst%\*%b + ta\*s\*sqrt(t(xst)%\*%solve(t(X)%\*%X)%\*%xst)

## [,1]  
## [1,] 5.304659

#Question 4. Find a 95% prediction interval for a mammal weighing 50 kg

#Lower Bound

xst%\*%b - ta\*s\*sqrt(1 + t(xst)%\*%solve(t(X)%\*%X)%\*%xst)

## [,1]  
## [1,] 3.668272

#Upper Bound

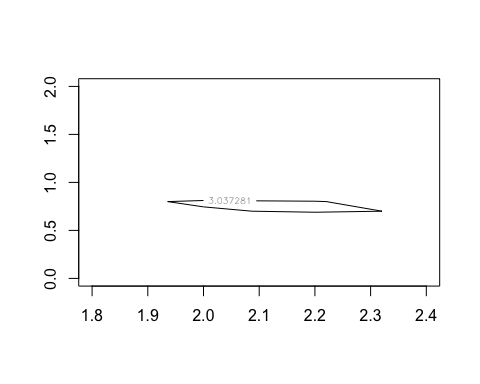
xst%\*%b + ta\*s\*sqrt(1 + t(xst)%\*%solve(t(X)%\*%X)%\*%xst)

## [,1]  
## [1,] 6.482531

#Question 5. Find and draw a 95% joint confidence region for the parameters.

#Ask the demonstator

b1 = seq(1.8,2.5,.2)  
b2 = seq(0,2,.1)  
f = function(beta1,beta2){  
 b = matrix(c(2.134,0.7517),2,1)  
 XTX = matrix(c(62,82.93,82.93,705.91),2,2)  
 f.out = rep(0,length(beta1))  
 for (i in 1:length(beta1)){  
 beta = matrix(c(beta1[i], beta2[i]),2,1)  
 f.out[i] = t(b - beta) %\*% XTX %\*% (b - beta)  
 }  
 return(f.out)  
}  
  
z = outer(b1,b2,f)  
  
contour(b1, b2, z, levels=2\*0.4820452\*qf(0.95, 2, 60))



#Question 6. Test the following hypotheses, using the anova function.

#(a) H0 :β=0

n = 62  
p = 2  
SSTotal = sum(y^2)  
SSRes = sum((y - X%\*%b)^2)  
SSReg = SSTotal - SSRes  
Fstat = (SSReg/p)/(SSRes/(n-p))  
pf(Fstat,p, n-p, lower.tail = FALSE)

## [1] 1.403557e-46

m1 = lm(mammals$BrainWt~0, data = mammals)  
m = lm(mammals$BrainWt~mammals$BodyWt, data = mammals)  
anova(m1,m)

## Analysis of Variance Table  
##   
## Model 1: mammals$BrainWt ~ 0  
## Model 2: mammals$BrainWt ~ mammals$BodyWt  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 62 976.48   
## 2 60 28.92 2 947.56 982.85 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#Reject the null

#(b) H0: β1 =0

null = lm(mammals$BrainWt~1, data = mammals) #1 is just temporarily because we   
anova(null,m)

## Analysis of Variance Table  
##   
## Model 1: mammals$BrainWt ~ 1  
## Model 2: mammals$BrainWt ~ mammals$BodyWt  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 61 365.11   
## 2 60 28.92 1 336.19 697.42 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#(c) H0 :β0 =0

X2 = X[,-1]  
b2 = solve(t(X2)%\*%X2, t(X2)%\*%y)  
SSRes2 = sum((y - X2%\*%b2)^2)  
Rg2 = SSTotal - SSRes2  
Rg2

## [1] 709.4036

Rg2 = t(y)%\*%X2%\*%b2  
Rg2

## [,1]  
## [1,] 709.4036

Rg1g2 = SSReg - Rg2  
Rg1g2

## [,1]  
## [1,] 238.1566

r = 1 #Rank 1 (C\*beta = 0, C = 1)  
Fstat = (Rg1g2/r)/(SSRes/(n-p))  
Fstat

## [,1]  
## [1,] 494.0545

pf(Fstat, r, n-p, lower.tail = FALSE)

## [,1]  
## [1,] 1.183207e-30

null = lm(mammals$BrainWt~0+mammals$BodyWt, data = mammals)  
anova(null,m)

## Analysis of Variance Table  
##   
## Model 1: mammals$BrainWt ~ 0 + mammals$BodyWt  
## Model 2: mammals$BrainWt ~ mammals$BodyWt  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 61 267.079   
## 2 60 28.923 1 238.16 494.05 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#(d) H0 :β=(2,1)

bst = as.vector(c(2,1))  
Fstat = (t(b-bst)%\*%t(X)%\*%X%\*%(b-bst))/p/(SSRes/(n-p))   
  
Fstat

## [,1]  
## [1,] 40.55789

pf(Fstat,p,n-p,lower.tail = FALSE)

## [,1]  
## [1,] 7.19895e-12

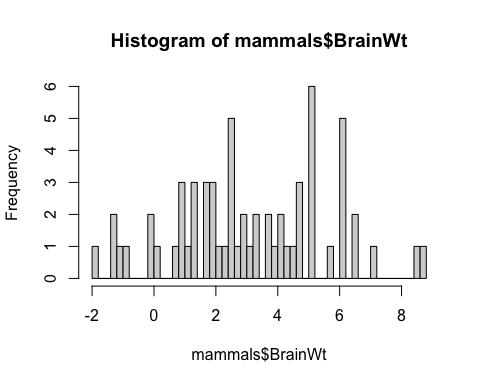
h0 = X%\*%bst  
basemodel = lm(mammals$BrainWt~0, data=mammals, offset = h0)  
anova(basemodel,m)

## Analysis of Variance Table  
##   
## Model 1: mammals$BrainWt ~ 0  
## Model 2: mammals$BrainWt ~ mammals$BodyWt  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 62 68.024   
## 2 60 28.923 2 39.101 40.558 7.199e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

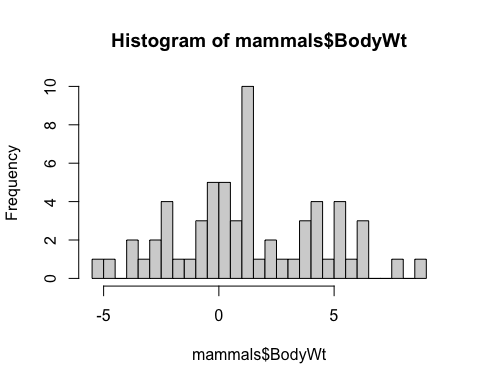
#Reject all hypothesis!!

#Question 7. By visualising the raw data, justify the use of a double logarithmic transformation. Write down the final model for the (untransformed) brain weight vs. body weight.

hist(mammals$BrainWt,breaks=50)



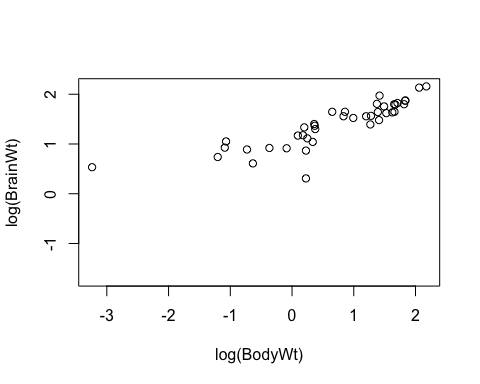
hist(mammals$BodyWt,breaks=50)



plot(log(BrainWt)~log(BodyWt),data=mammals)

## Warning in log(BrainWt): NaNs produced

## Warning in log(BodyWt): NaNs produced



#The final model see my annotation in MAST30025 Lab 6!! first find beta0 and beta1!