Homework 3

Section 1 (9:00am to 11:00am)

February 28, 2014

Homework Summary

Problem 1

Part a

For Linear Regression

$\hat{f}(X) = X^T\beta \implies \hat{f}(x_0) = x_0^T\beta = \sum \left(x_0^T \left(x_0^T x_0\right)^{-1} x_0^T\right)_i y_i$

$\therefore l(x_0; X) = \left(x_0^T \left(x_0^T x_0\right)^{-1} x_0^T\right)_i$

For K nearest Neighbor

$\hat{f}(x) = \frac{1}{k}\sum y_i = \sum \frac{1}{k} * y_i$

$\therefore l(x_0; X) = \frac{1}{k}$

Part b

$E_{Y|X}(f(x_0) - \hat{f}(x_0))^2 = E_{Y|X}(f(x_0)^2 - 2 * f(x_0) * \hat{f}(x_0) + \hat{f}(x_0)^2) = f(x_0)^2 - 2 * f(x_0) * E_{Y|X}(\hat{f}(x_0)) + E_{Y|X}(\hat{f}(x_0)^2) = f(x_0)^2 - 2 * f(x_0) * E_{Y|X}(\hat{f}(x_0)) + E_{Y|X}(\hat{f}(x_0)^2) + \left[E_{Y|X}(\hat{f}(x_0))\right]^2 - \left[E_{Y|X}(\hat{f}(x_0))\right]^2 = f(x_0)^2 - 2 * f(x_0) * E_{Y|X}(\hat{f}(x_0)) + E_{Y|X}(\hat{f}(x_0)^2) + \left[E_{Y|X}(\hat{f}(x_0))\right]^2 - \left[E_{Y|X}(\hat{f}(x_0))\right]^2 = \left[f(x_0) - E_{Y|X}(\hat{f}(x_0))\right]^2 + E_{Y|X}(\hat{f}(x_0)^2) - \left[E_{Y|X}(\hat{f}(x_0))\right]^2$

Part c

$E_{Y,X}(f(x_0) - \hat{f}(x_0))^2 = E_{Y,X}(f(x_0)^2 - 2 * f(x_0) * \hat{f}(x_0) + \hat{f}(x_0)^2) = f(x_0)^2 - 2 * f(x_0) * E_{Y,X}(\hat{f}(x_0)) + E_{Y,X}(\hat{f}(x_0)^2) = f(x_0)^2 - 2 * f(x_0) * E_{Y,X}(\hat{f}(x_0)) + \left[E_{Y,X}(\hat{f}(x_0)\right]^2 - \left[E_{Y,X}(\hat{f}(x_0)\right]^2 = f(x_0)^2 - 2 * f(x_0) * E_{Y,X}(\hat{f}(x_0)) + \left[E_{Y,X}(\hat{f}(x_0)\right]^2 + E_{Y,X}(\hat{f}(x_0)^2) - \left[E_{Y,X}(\hat{f}(x_0)\right]^2 = f(x_0)^2 - 2 * f(x_0) * E_{Y,X}(\hat{f}(x_0)) + E_{Y,X}(\hat{f}(x_0)^2) + \left[E_{Y,X}(\hat{f}(x_0))\right]^2 - \left[E_{Y,X}(\hat{f}(x_0))\right]^2 = \left[f(x_0) - E_{Y,X}(\hat{f}(x_0))\right]^2 + E_{Y,X}(\hat{f}(x_0)^2) - \left[E_{Y,X}(\hat{f}(x_0))\right]^2$

Part d

In Part b, the term is better written as $\left[f(x_0) - E_{Y|X}(\hat{f}(x_0))\right]^2 + E_{Y|X}(\hat{f}(x_0)^2) - \left[E_{Y|X}(\hat{f}(x_0))\right]^2 =$
$\left[f(x_0) - E(l(x_0;X)y_i|X=x)^2)\right]^2 + E((l(x_0;X)y_i|X=x)^2) - [E(l(x_0;X)y_i|X=x)]^2$
$= \left[f(x_0) - E_{Y,X}(\hat{f}(x_0))/E_X(\hat{f}(x_0))\right]^2 + E_{Y,X}(\hat{f}(x_0)^2)/E_X(\hat{f}(x_0)^2) - \left[E_{Y,X}(\hat{f}(x_0))/E_X(\hat{f}(x_0))\right]^2$

Problem 2

$accuracy = (1 - error)$	Linear Regression	1-nn	3-nn	5-nn	7-nn	15-nn
Training	0.9942405	1	0.9949604	0.9942405	0.9935205	0.9906407
Test	0.9587912	0.9752747	0.9752747	0.9697802	0.967033	0.9615385

With regards, to training, the first and thrid nearest neighbor methods have a higher accuracy, thereby, a lower error rate, than the Linear Regression. This could be due to the number of neighbors being analyzed as a lower number when compared to the Linear Regression method which takes into account the entire set.

The Linear Regression method has a lower accuracy, higher error, compared to all the K-nn

Problem 3

Part 1

I generated an X vector with a length of 30 by assuming $N \sim (0, 1)$ and created a three degree polynomial $y = 3 * x^3 + 2 * x^2 + x$. Following this, I created a Linear Model assuming that Y was the response variable. Using this model, I used the predict function in order find the Standard error for each point. I created the 95% confidence interval, in blue on the plot, for each point by multiplying the SE by 1.96 and adding then substracting from the predicted value. 27 of the 30 points were found to be inside this confidence interval, I'm assuming that since the number wasn't large enough, 2 to 3 points outside the interval was acceptable.

Part II

$C_\beta = \left\{\beta|(\hat{\beta} - \beta)^T X^T X * (\hat{\beta} - \beta) \leq \hat{\sigma}^2 X_{p+1}^{2(1-\alpha)}\right\} \approx X_{3+1}^{2(1-0.025)}\hat{\sigma}^2 = 12.8325 * \hat{\sigma^2}$

The method in Part II would be larger, because of the Chi-square value determined $12.8325 > 1.96$, this is highlighted in green.

Problem 4

assume $X' = X + \sqrt{\lambda}I_p \implies X'^T X' = (X, \sqrt{\lambda}I_p)^T (X, \sqrt{\lambda}I_p) = X^T X + \lambda I_p$,

$\hat{\beta}_{ridge} = \left(X^T X + \lambda I_p\right)^{-1} X^T Y$ and $Y' = (Y, 0_p)$

$\hat{\beta} = \left(X^T X\right)^{-1} X^T Y = \left(X^T X + \lambda I_p\right)^{-1} (X, \sqrt{\lambda})^T Y'$ and $(X, \sqrt{\lambda I})^T Y' = X^T Y + 0_p * \sqrt{\lambda} I_p = X^T Y$

thus, $\hat{\beta} = \left(X^T X\right)^{-1} X^T Y = \left(X^T X + \lambda I_p\right)^{-1} (X)^T Y$

$\therefore$ Since $\hat{\beta} = \hat{\beta}_{ridge}$, ridge regression estimates can be obtained by ordinary least squares regression on an augmented data set.

Problem 5

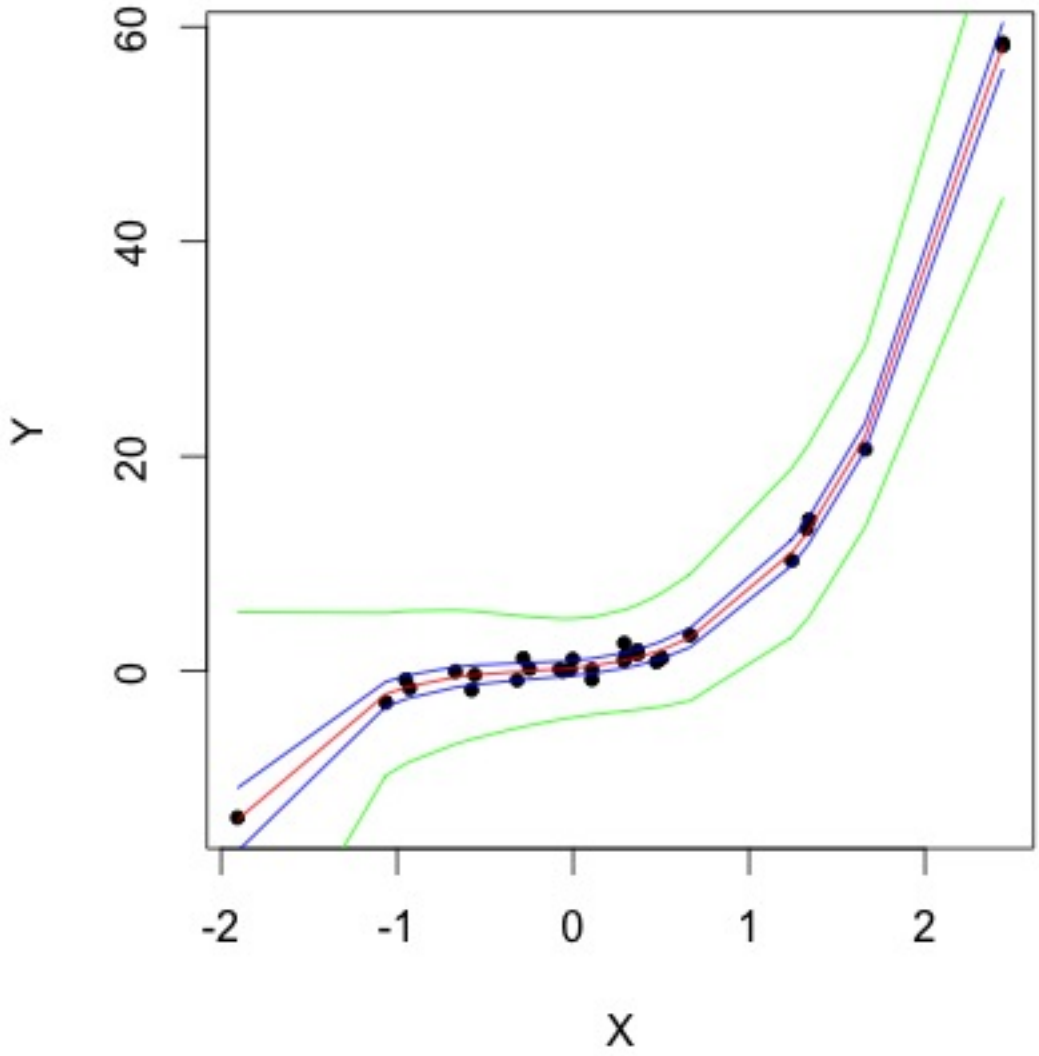
When first analyzing the the Brains Weight Data, I constructed a Linear Model with all the data to look for any outlying points. The Brain Weight was used as the response variable, while body weight was used as the exploratory variable. Based on the cook's distance, I identified three species that could affect the building of the model, African Elephant, Human, and Asian Elephant. Human had a cook's distance that was close to 0.5 and was not an outlier yet was still considered affecting data, while both Elephant points were greater than one, concluding that they were outliers. This conclusion was also confirmed by looking at a Normal Q-Q plot. Had these three points not been eliminated, the regression line would have been $Brain.Weight = 0.9665 * Body.Weight + 91.0086$.

After eliminating the three outliers, three additional points were found to be influencing the new Linear Model; however only one, cow, was considered an outlier because it's cook's distance was greater than or equal to 1. The Regression line for this model was $Brain.Weight = 1.228 * Body.Weight + 36.573$.

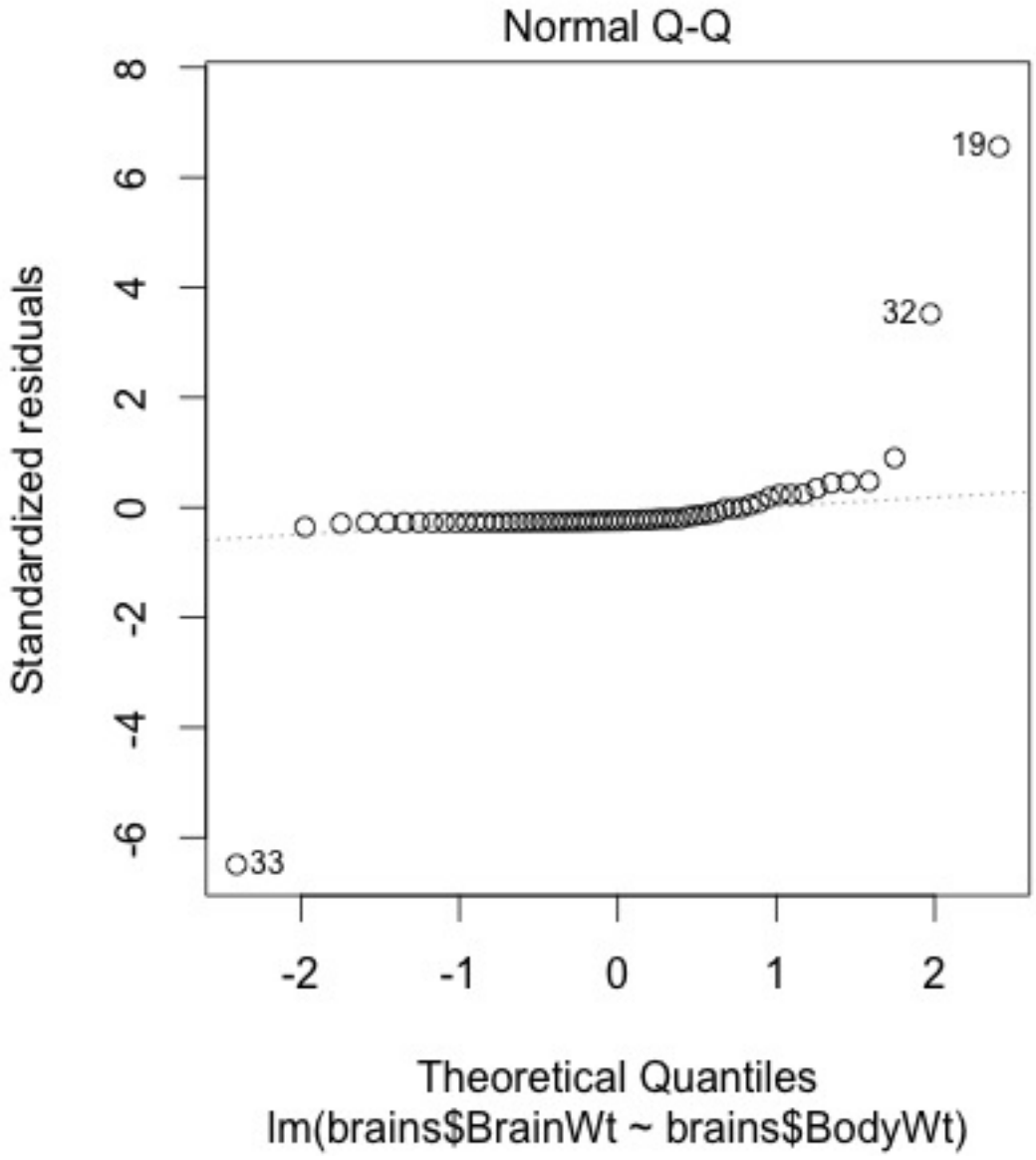
Appendix

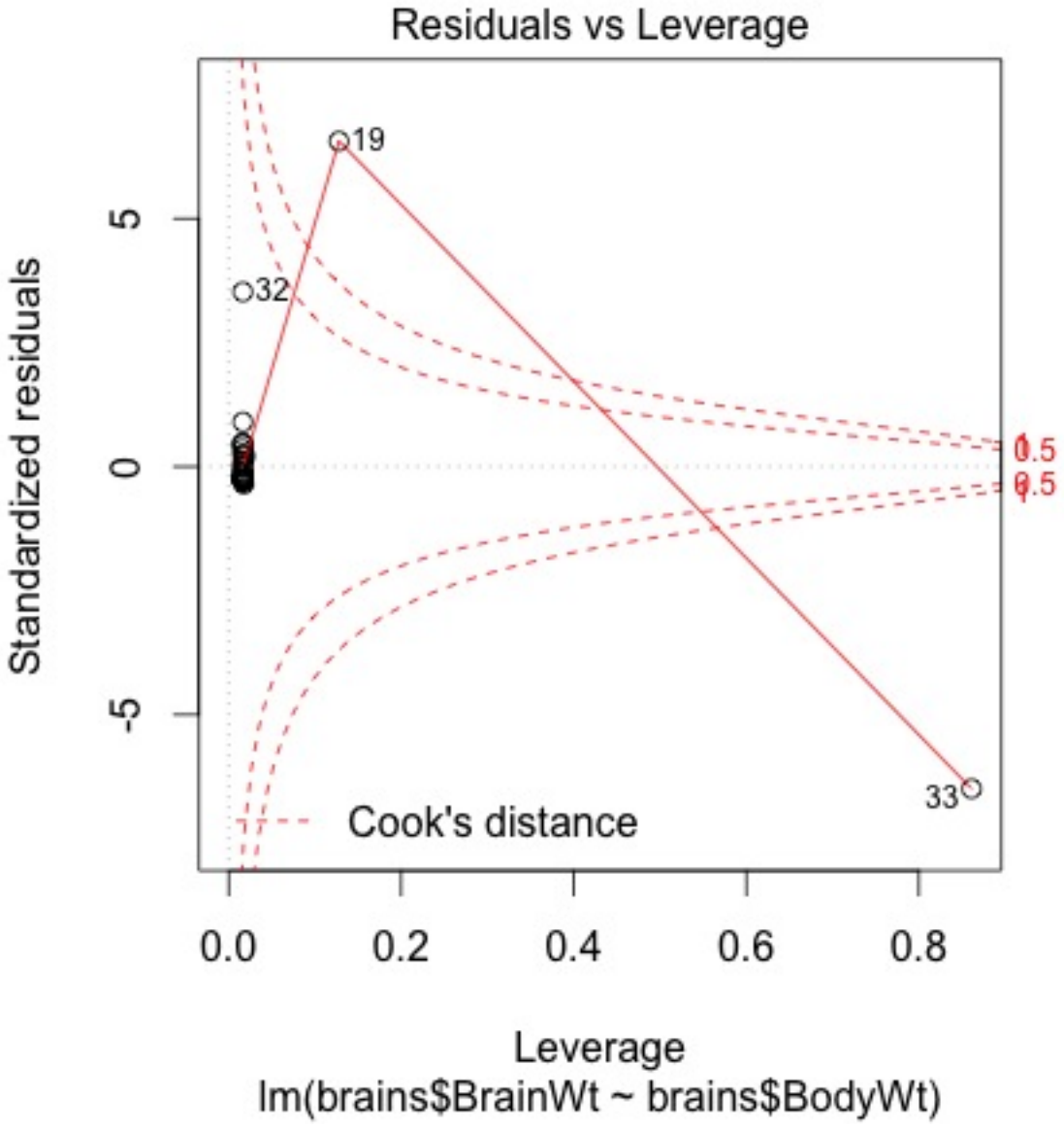
Graphs

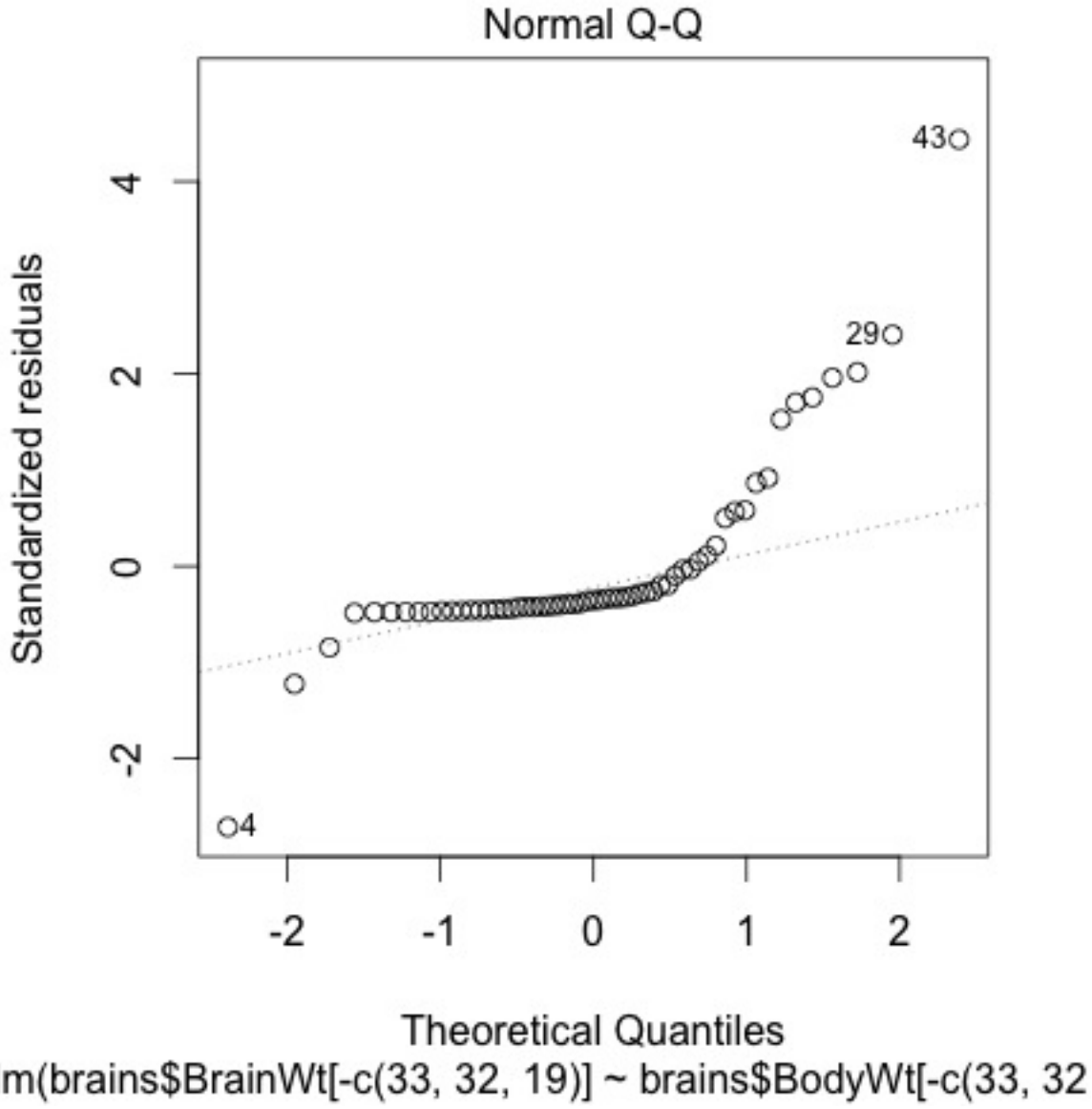
Problem 3



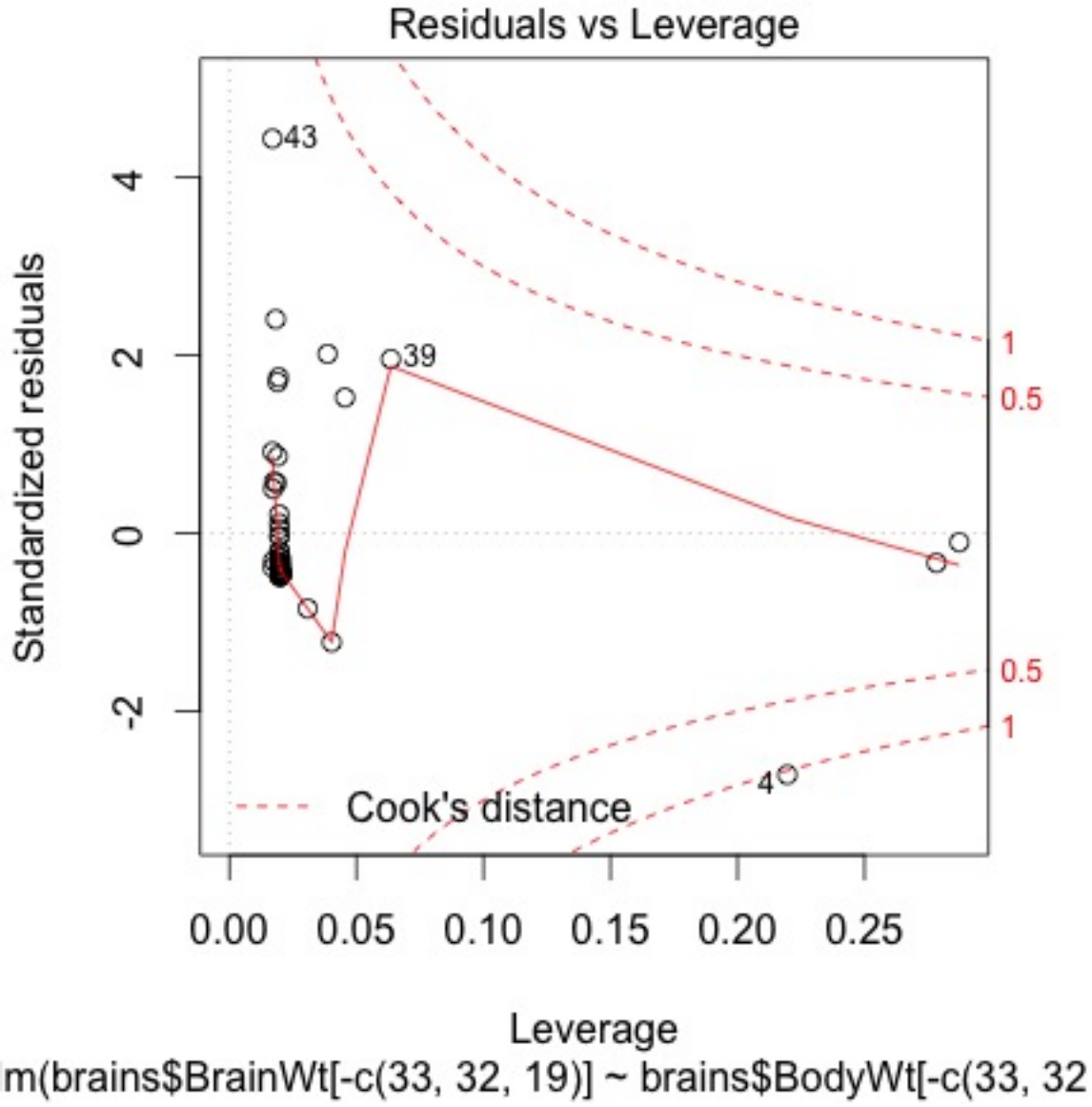
Problem 5


Normal Q-Q
Standardized residuals
8
6
4
2
0
-2
-4
-6
19
32
33
-2
-1
0
1
2
Theoretical Quantiles
lm(brains$BrainWt ~ brains$BodyWt)


Residuals vs Leverage
Standardized residuals
5
0
-5
19
32
33
0.5
0.5
Cook's distance
0.0
0.2
0.4
0.6
0.8
Leverage
lm(brains$BrainWt ~ brains$BodyWt)



m(brains$BrainWt[-c(33, 32, 19)] ~ brains$BodyWt[-c(33, 32



Code

```
setwd("Dropbox/School/Statistics/Stat 154 Spring 2014/HW3")

###############Problem 2###############
ZipTrain = read.table(file="zip.train", header=F)
ZipTest = read.table('zip.test', header=F)
ZipTrain = ZipTrain[ZipTrain$V1== 2 | ZipTrain$V1== 3, ]
ZipTest = ZipTest[ZipTest$V1== 2 | ZipTest$V1== 3, ]
ZipTrain = na.omit(ZipTrain)
Problem2 = glm(V1~., data=ZipTrain) summary(Problem2)$coefficients[,1]#errors
```

```
error.for.test = function(ZipTest){Testing = t(ZipTest[, -1])
print(dim(Testing))
Test.yhat = c()
for(i in 1:ncol(Testing)){
Test.yhat[i] = Problem2$coefficients[1]+sum(Problem2$coefficients[-1]*(Testing[,i]))}
Test.yhat = round(Test.yhat)
Test.error = mean(ZipTest$V1-Test.yhat)^2 return(Test.error)}
error.for.test(ZipTrain)
error.for.test(ZipTest)
library("FNN") k.error.train.test = function(k){
Train.error.k = sum(ZipTrain$V1-knn(ZipTrain[,-1], ZipTrain[, -1], ZipTrain$V1, k =k))/leng
Test.error.k = sum(ZipTest$V1-knn(ZipTrain[,-1], ZipTest[, -1], ZipTrain$V1, k =k))/length(
return(data.frame(Train.error.k, Test.error.k)) }
k.error.train.test(1)
k.error.train.test(3)
k.error.train.test(5)
k.error.train.test(7)
k.error.train.test(15)
###############Problem 3###############
X = rnorm(30, 0,1)
Y = 0 + 1*X + 2*X^2 + 3*X^3 + rnorm(30, 0, 1) X.lm = lm(Y~1+X+I(X^2)+I(X^3))
X.lm
#  Coefficients:
# (Intercept)            X       I(X^2)       I(X^3)
#      51.421       -1.154        2.030        3.000
X.Sum.lm = summary(X.lm) X.Sum.lm
# Call:
#   lm(formula = Y ~ 1 + X + I(X^2) + I(X^3))
#
# Residuals:
#    Min      1Q  Median      3Q     Max
# -1.6499 -0.4746 -0.1368  0.5540  1.3134
#
# Coefficients:
#    Estimate Std. Error t value Pr(>|t|)
# (Intercept)  0.06983    0.15975   0.437  0.66561
# X            0.87570    0.28053   3.122  0.00437 **
#    I(X^2)       2.15108    0.11686  18.407  < 2e-16 ***
#    I(X^3)       2.98545    0.08381  35.620  < 2e-16 ***
#    ---
#    Signif. codes:  0 '*** 0.001 '** 0.01 '* 0.05 '. 0.1 ' ' 1
#
# Residual standard error: 0.7275 on 26 degrees of freedom
# Multiple R-squared:  0.9953,  Adjusted R-squared:  0.9947
# F-statistic:  1820 on 3 and 26 DF,  p-value: < 2.2e-16
errors = X.Sum.lm$coefficients[,1]
plot(X, Y, pch = 20)
y = X^3*X.lm$coefficients[4] + X^2*X.lm$coefficients[3] +
(X)*X.lm$coefficients[2] + 1*X.lm$coefficients[1]
X.x = X[order(X)]
y = y[order(X)] points(X.x, y, type="l", col = "red")
CI.X.lm = predict(X.lm, data.frame(X.x,y), interval="confidence")
upper.CI=1.96*(CI.X.lm[,3]-CI.X.lm[,1])+CI.X.lm[,1]
lower.CI=1.96*(CI.X.lm[,2]-CI.X.lm[,1])+CI.X.lm[,1]
```

```
points(X.x, upper.CI[order(X)],type="l", col = "blue")
points(X.x, lower.CI[order(X)],type="l", col = "blue")
upper.CI = qchisq(1-0.025, 5)*(CI.X.lm[,3]-CI.X.lm[,1]) + CI.X.lm[,1]
lower.CI = qchisq(1-0.025, 5)*(CI.X.lm[,2]-CI.X.lm[,1]) + CI.X.lm[,1]
points(X.x, upper.CI[order(X)],type="l", col = "green")
points(X.x, lower.CI[order(X)],type="l", col = "green")
##############Problem 5##############
brains = read.csv("brains.csv")
plot(brains$BrainWt, brains$BodyWt)
brains.lm = lm(brains$BrainWt~brains$BodyWt) plot(brains.lm)
plot(brains$BrainWt[-c(33,32,19)],
brains$BodyWt[-c(33,32,19)])
brains.lm = lm(brains$BrainWt[-c(33,32,19)]~brains$BodyWt[-c(33,32,19)])
plot(brains.lm)
```