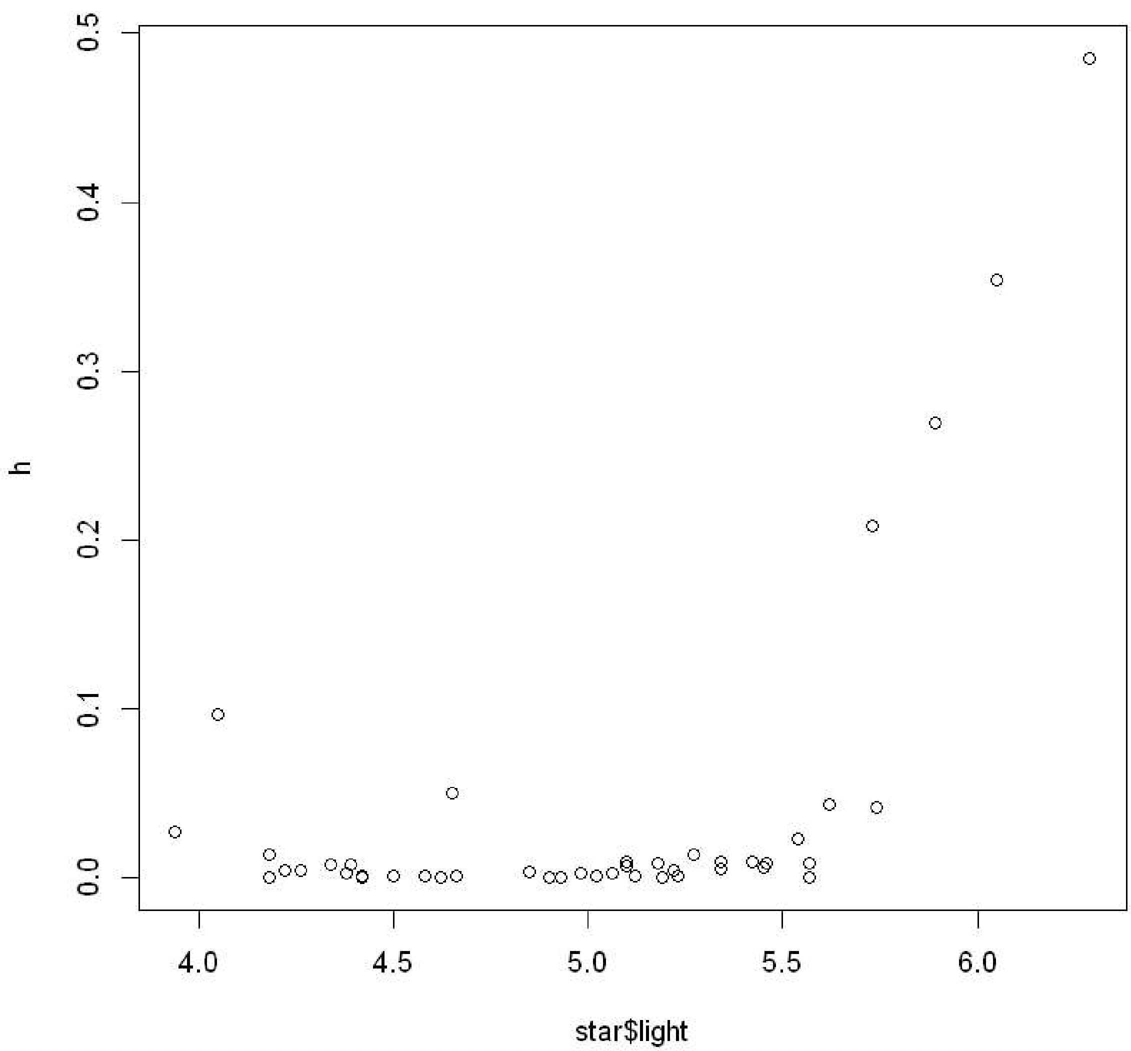
In [ ]:
```
STAT 420 HW 8 Donghan Liu   Donghan2
```

In [23]:
```
library(faraway)
data(star)
```

In [40]:

```
#1a
fit = lm(temp~light,data = star)
h = rep(NA, nrow(star))
n = rep(NA, nrow(star))
sigma2 = sum(fit$residuals^2)/(nrow(star)-2)
X = cbind(1,star$light)
for (i in 1:nrow(star))
{
    stari = star[-i,]
    fiti = lm(temp~light,data = stari)
    H = X %*% solve(t(X)%*%X)%*%t(X)
    h[i] = (H[i,i]/(1-H[i,i])^2)*fit$residuals[i]^2/2/sigma2
}
h
plot(star$light, h)
# By applying for loop, we could define each point at one time in
# terms of cook's Distance,and put them into variable myCooksD
```

0.00107919178252384 0.0413689908401431 0.000474908033849029
0.0413689908401431 1.21614725175368e-05 0.00881561858269054
0.0499671112027003 0.0135564543007699 2.63857782196173e-05
0.000714381192607708 0.208331861967192 0.00615139263081007
0.00954942184479616 0.0969561575855037 0.00407492539609282
0.000886906221601172 0.0276205922423768 0.000204088717779492
0.013416108564253 0.269261494531683 0.00245983936847498
0.0047804144192325 0.000641765487986141 0.00386129803572975
0.000674683473622641 0.000991444942737331 0.000632262302052985
0.000470558276173611 0.00769453593872798 0.353626907748568
1.27743059386415e-05 0.00927319895868729 0.00404152690664204
0.485271903461741 0.00798080364776392 0.0429250719039029
0.00725240690191143 0.00404152690664204 0.00831565435889204
0.00895044173369681 0.000159363516246955 0.0028520601434559
0.00923905592871637 0.00559079211464682 0.022995642442235
0.0025148611025890 4 0.000771528063385228



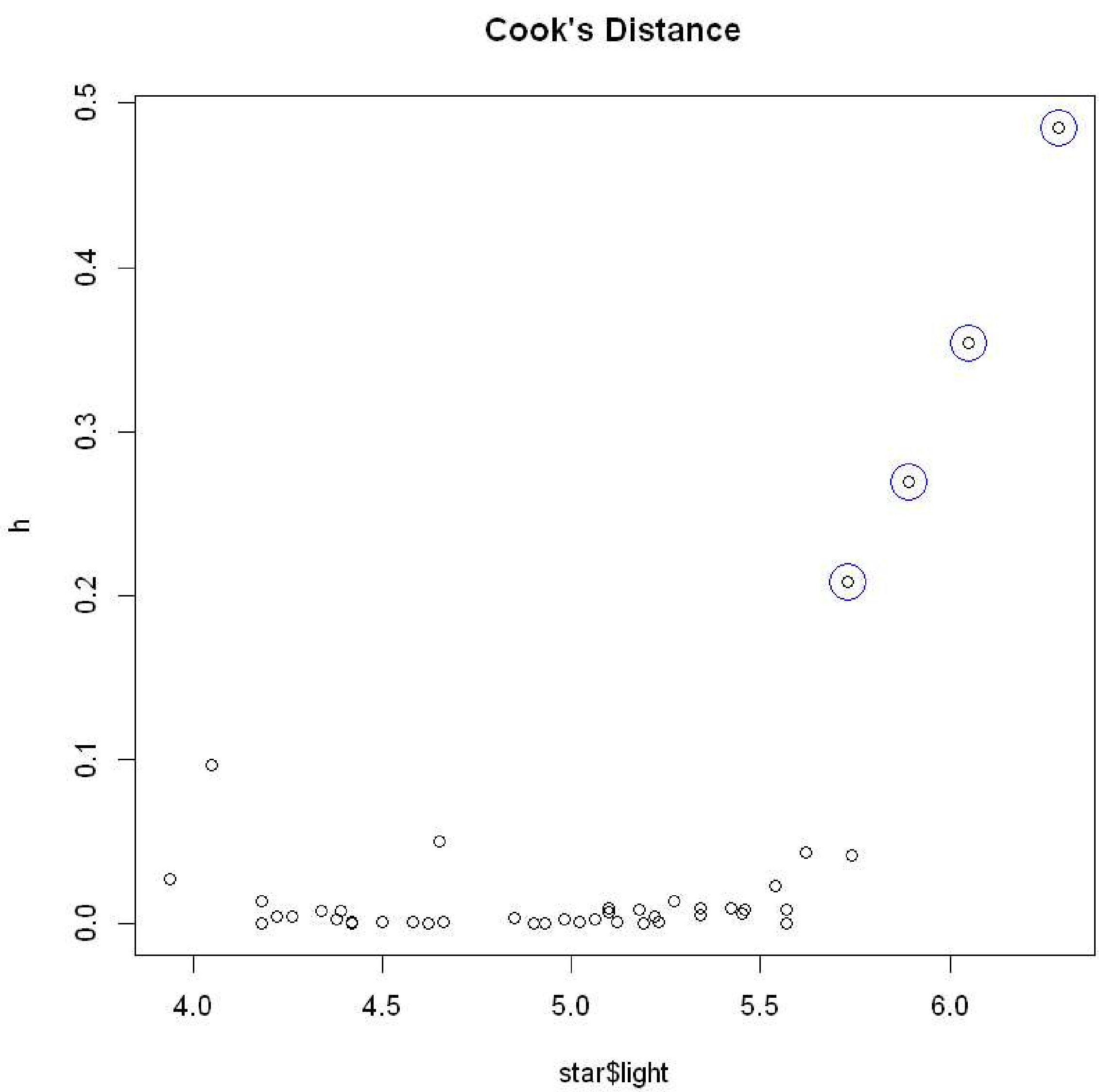
In [29]:

```
#1b
summary(influence.measures(fit))
which.max(cooks.distance(fit))
```

```
Potentially influential observations of
	 lm(formula = temp ~ light, data = star) :

   dfb.1_ dfb.lght dffit   cov.r   cook.d hat    
11  0.49  -0.55    -0.70_*  0.79_*  0.21   0.06  
20  0.61  -0.66    -0.79_*  0.81_*  0.27   0.07  
30  0.74  -0.79    -0.90_*  0.83_*  0.35   0.09  
34  0.91  -0.96    -1.05_*  0.88    0.49   0.13_*
```

**34**: 34

In [26]:
```
plot(star$light,h,main = "Cook's Distance")
points(star$light[c(11,20,30,34)], h[c(11,20,30,34)], cex = 3, col = "blue")
# summary(influence.measures(fit)) tells us that influential points of index
# 11,20,30,34 are the point have potential influential subjects,
# and points() could help us to locate those points in the cook's
# distance graph.
```




In [27]:
```
star34 = star[-34,]
fit34 = lm(temp~light, data = star34)
X = cbind(1,star$light)
diff = X %*% (fit$coefficients - fit34$coefficients)
sigma2 = sum(fit$residuals^2)/(nrow(star)-2)
t(diff) %*% diff / 2 / sigma2
# As the result shows, the value for cook's distance point 34 is 0.4852719
```

0.4852719

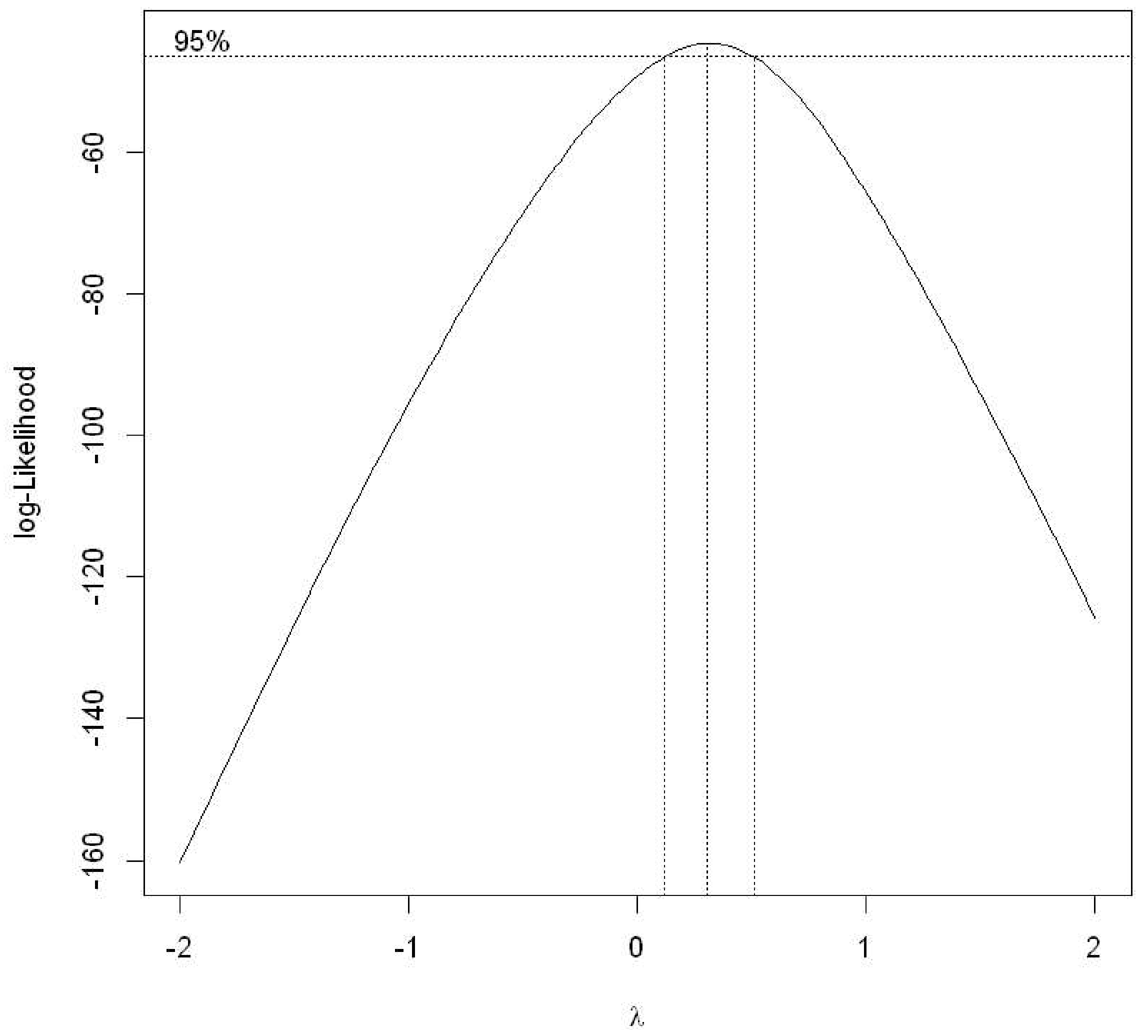
In [30]:
```
#1c
star1 = star[-c(11,20,30,34),]
fit1 = lm(temp~light, data = star1)
summary(fit)
summary(fit1)
# By comparing these two fits, we could obviously notice that intercept
# become smaller, whereas, the paramater estimator for light change
# from negative to positive, which indicates that after removing
# those influential subjects, the relationship between light and
# temp becomes positive correlation. Then, the t value for both terms
# turn out to be large and the p-value correspondingly becomes smaller
# and they are highly significant in the new model, which states that
# they have highly siginficanly correlation with reponse varible. Moreover,
# the value for Multiple R-squared and Adjusted R-squared increased
# approximately 36% as well. Overall, we could define that the new
# model is more appropriate than the old one.
```

```
Call:
lm(formula = temp ~ light, data = star)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.74310 -0.09414  0.06371  0.17744  0.37512 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.84691    0.37422  12.952   <2e-16 ***
light       -0.10712    0.07419  -1.444    0.156    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2875 on 45 degrees of freedom
Multiple R-squared:  0.04427,	Adjusted R-squared:  0.02304 
F-statistic: 2.085 on 1 and 45 DF,  p-value: 0.1557
```

```
Call:
lm(formula = temp ~ light, data = star1)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.49795 -0.05048  0.01922  0.08176  0.16623 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.50511    0.18187  19.272  < 2e-16 ***
light        0.17910    0.03677   4.871  1.7e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.12 on 41 degrees of freedom
Multiple R-squared:  0.3666,	Adjusted R-squared:  0.3511 
F-statistic: 23.73 on 1 and 41 DF,  p-value: 1.697e-05
```

In [31]:
```
#2a
data(gala)
fit2 = lm(Species~Area+Elevation+Nearest+Scruz+Adjacent,data = gala)
```

In [32]:
```
library(MASS)
bc = boxcox(fit2,plotit = T)
```



In [41]:
```
lambda = bc$x[which.max(bc$y)]
lambda
n = nrow(gala)
y = gala$Species
y = y/exp(mean(log(y)))
x1 = gala$Area
x2 = gala$Elevation
x3 = gala$Nearest
x4 = gala$Scruz
x5 = gala$Adjacent
gy = (y^lambda-1)/lambda
LL = -n/2*log(sum(lm(gy~x1+x2+x3+x4+x5)$residuals^2))
LL
bc$y[which.max(bc$y)]
# As we could see the lambda that calculated from maximum likelihood function
# we would make sure that 0.303030303 is the max estimator and we might use
# it to transform the model.
```

0.303030303030303

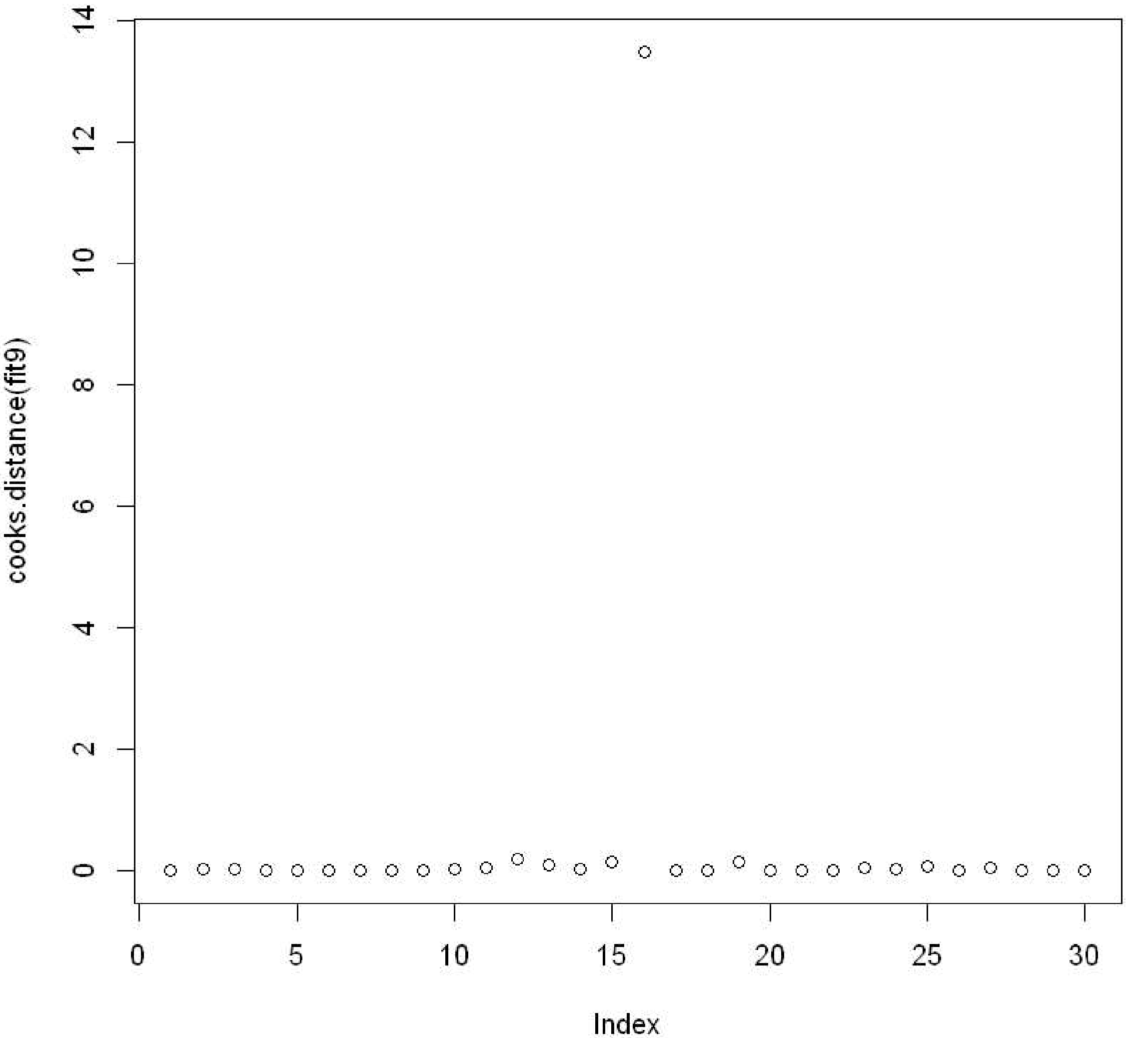
-44.6790794195843

-44.679088891257

In [76]:
```
y = gala$Species
fit9 = lm((y^0.3030303)~x1+x2+x3+x4+x5, data = gala)
summary(fit2)$r.squared
summary(fit9)
plot(cooks.distance(fit9))
# By applying 0.303030303 as the power of y value, the summary()
# function gives us some statistics about how good is this transformation
# First, intercept, elevation, adjacentare significant, but rest of
# predictors are not. Both r-square and adjusted r-square are
# slightly decreased, which indicates that these data represent around
# 70% variation. Also, the cook's distance tells us that only
# one point is very far away from the cut-off line.
```

0.765846944681233

```
Call:
lm(formula = (y^0.3030303) ~ x1 + x2 + x3 + x4 + x5, data = gala)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.26453 -0.42477 -0.07196  0.46992  1.50766 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.0842510  0.2494050   8.357 1.45e-08 ***
x1          -0.0006023  0.0002920  -2.063  0.05010 .  
x2           0.0043787  0.0006987   6.267 1.78e-06 ***
x3           0.0100500  0.0137258   0.732  0.47114    
x4          -0.0037088  0.0028047  -1.322  0.19852    
x5          -0.0008437  0.0002305  -3.661  0.00124 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.794 on 24 degrees of freedom
Multiple R-squared:  0.7465,	Adjusted R-squared:  0.6937 
F-statistic: 14.13 on 5 and 24 DF,  p-value: 1.713e-06
```



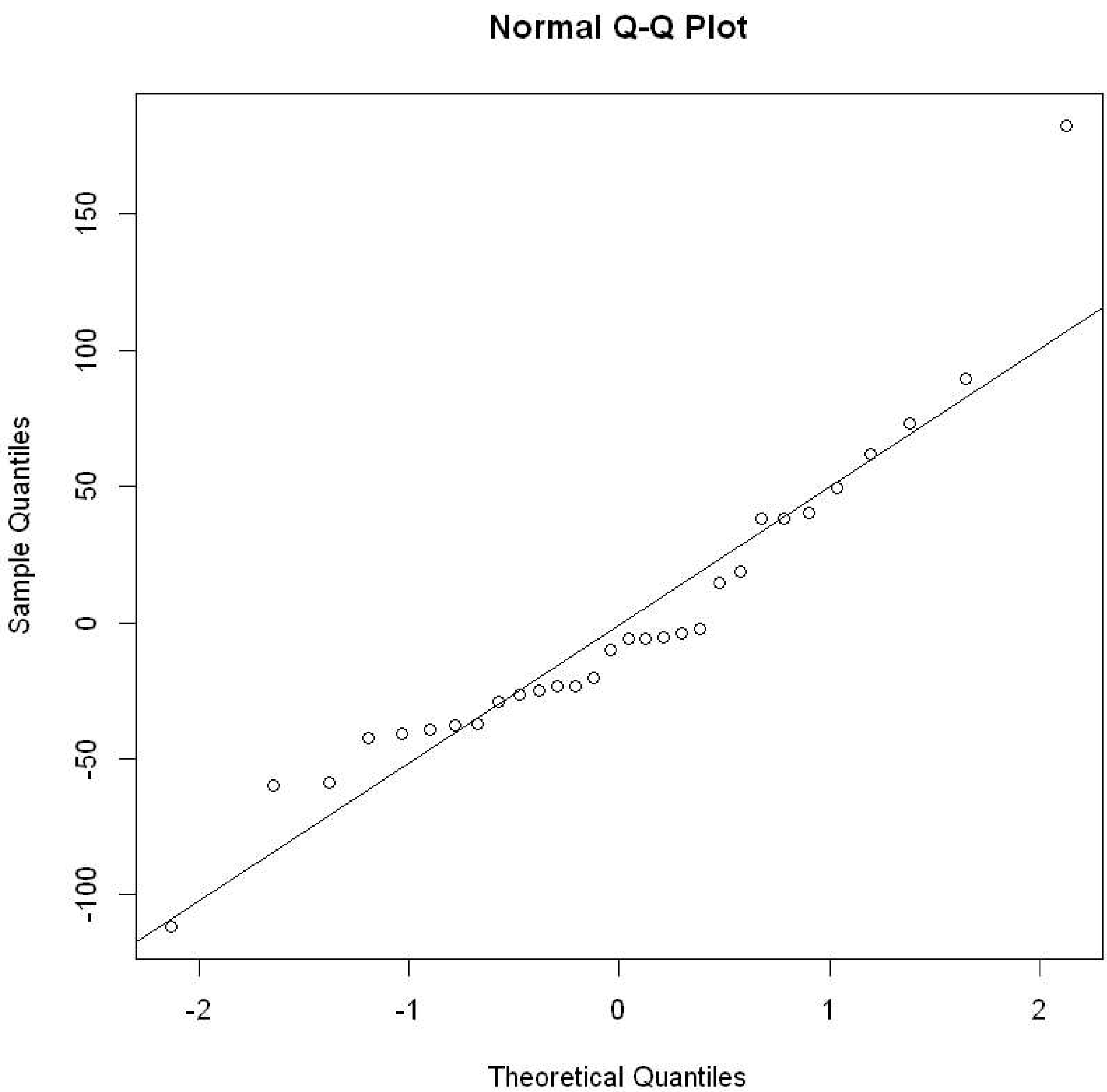
cooks.distance(fit9)
Index
0
2
4
6
8
10
12
14
0
5
10
15
20
25
30

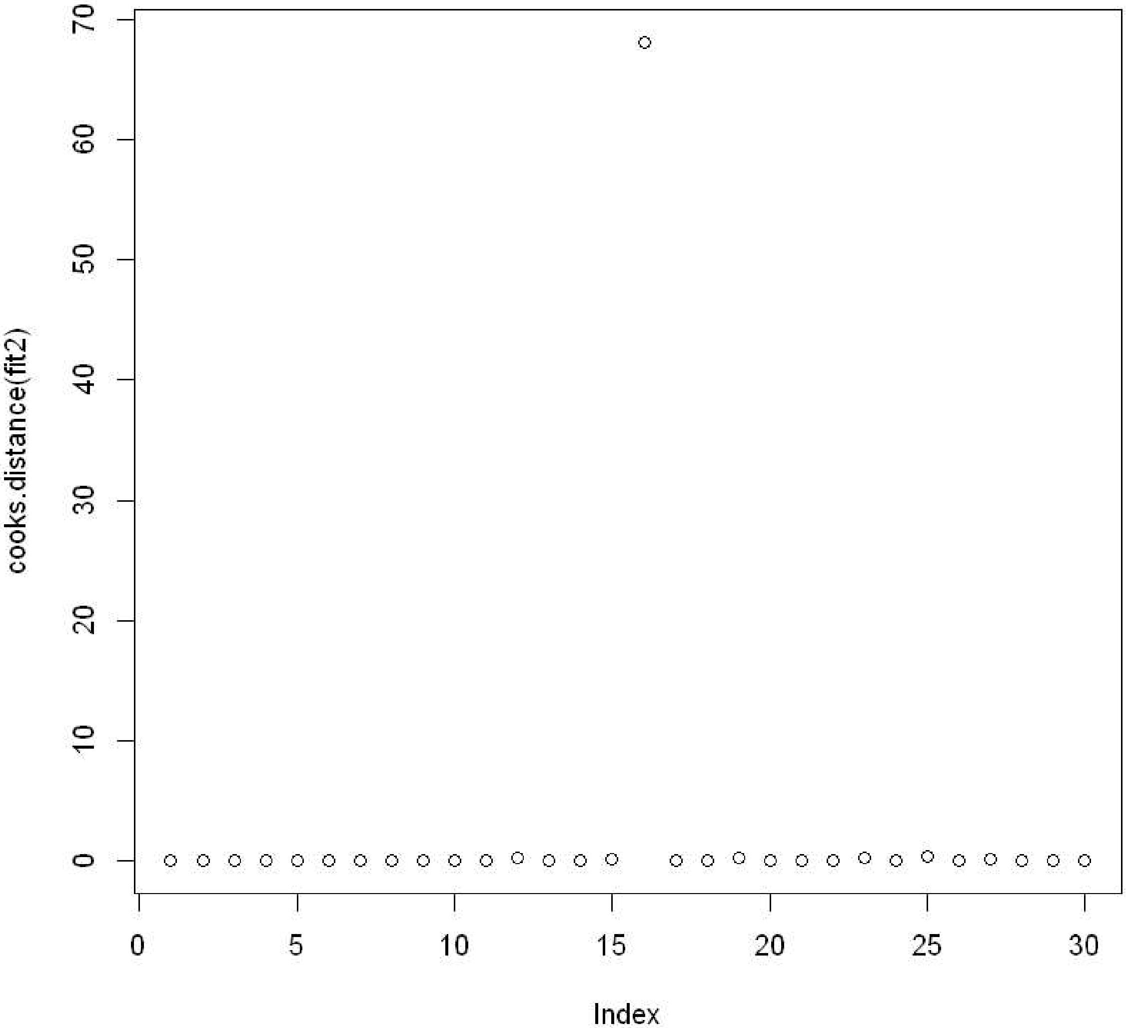
In [62]:
```
#2b
summary(influence.measures(fit2))
# By summary function, Darwin, Fernandina, Genovesa, Isabela, Pinta, SanCristo
bal, SantaCruz, Wolf
# are potential influecial points.
```

```
Potentially influential observations of
	 lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacen
t,      data = gala) :

             dfb.1_   dfb.Area dfb.Elvt dfb.Nrst dfb.Scrz dfb.Adjc dffit   
Darwin       -0.08     0.04    -0.04    -0.06     0.32    -0.02     0.39   
Fernandina    0.16     0.16    -0.12     0.03    -0.06    -0.83    -1.24   
Genovesa      0.17     0.26    -0.35     0.69    -0.34     0.29     0.76   
Isabela      -1.19_* -20.87_*   4.89_*   0.37    -1.02_*  -0.81   -29.59_*
Pinta         0.58     0.85    -1.03_*  -0.21    -0.16     0.66    -1.31   
SanCristobal -0.18    -0.10     0.26     1.02_*  -0.60    -0.12     1.21   
SantaCruz    -0.03    -0.83     1.52_*  -0.54    -0.24    -1.26_*   2.04_* 
Wolf          0.02     0.00     0.00     0.01    -0.06     0.01    -0.08   
             cov.r   cook.d   hat    
Darwin        2.31_*  0.03     0.47   
Fernandina   25.11_*  0.27     0.95_* 
Genovesa      1.86_*  0.10     0.43   
Isabela       0.33   68.08_*   0.97_* 
Pinta         0.50    0.24     0.25   
SanCristobal  1.13    0.23     0.38   
SantaCruz     0.04_*  0.39     0.18   
Wolf          1.93_*  0.00     0.33   
```

In [73]:
```
#2c
# Original qq plot and cook's distance plot
summary(fit2)$r.squared
qqnorm(fit2$residuals)
qqline(fit2$residuals)
plot(cooks.distance(fit2))
```

0.765846944681233

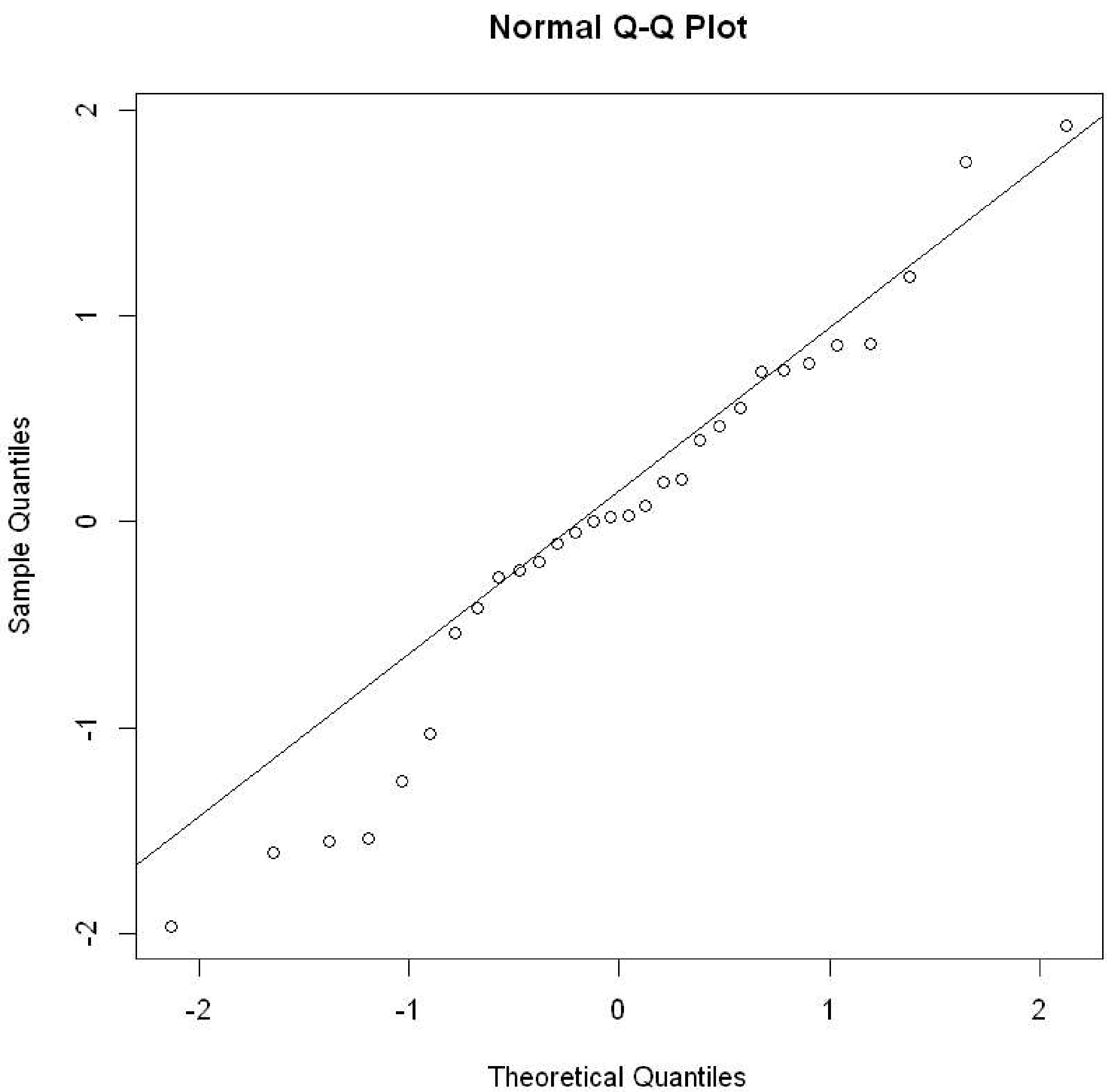


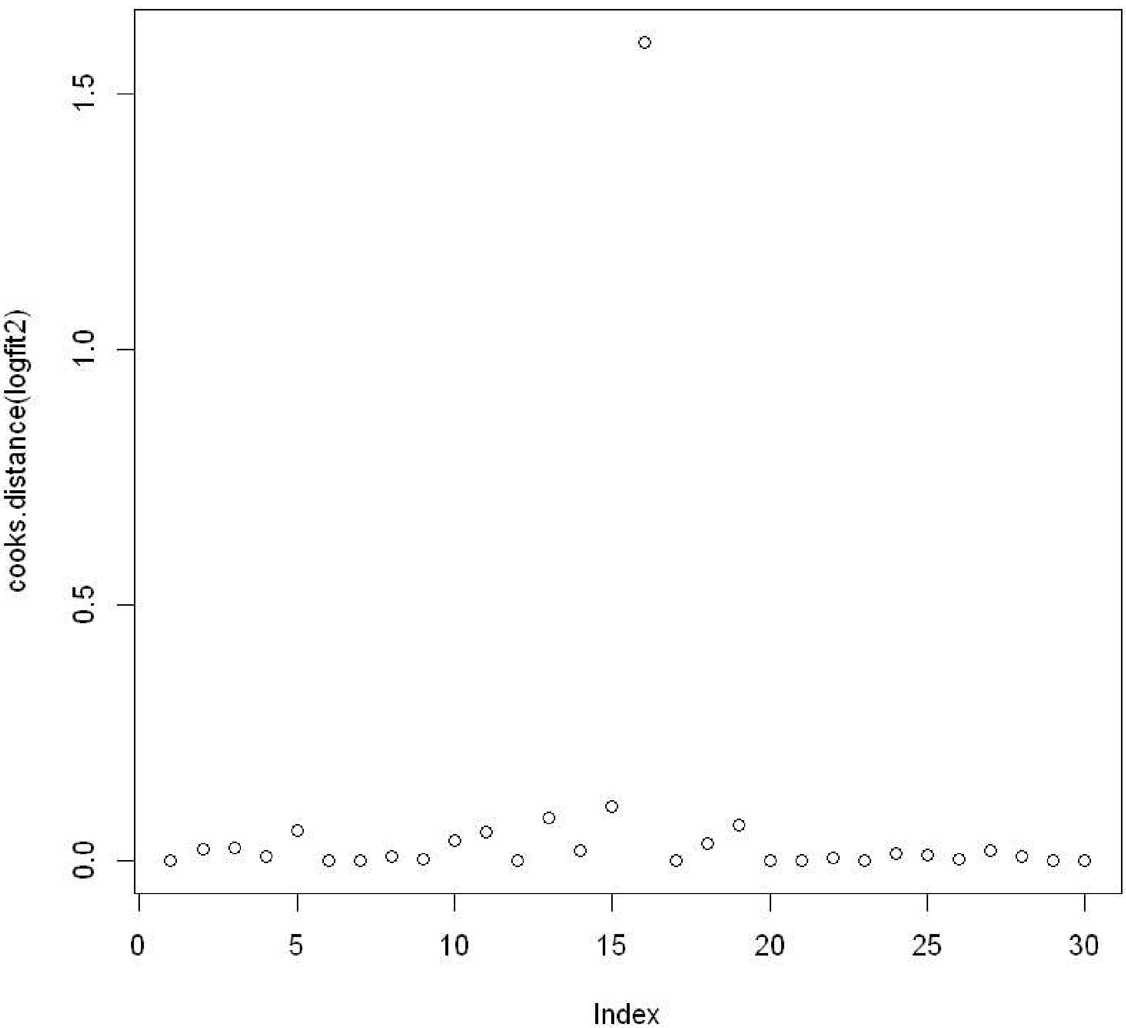

cooks.distance(fit2)
Index
0
10
20
30
40
50
60
70
0
5
10
15
20
25
30

In [72]:
```
#2c log transformation
logfit2 = lm(log(gala$Species)~gala$Area+gala$Elevation+gala$Nearest+gala$Scru
z+gala$Adjacent,data = gala)
qqnorm(logfit2$residuals)
qqline(logfit2$residuals)
res = logfit2$residuals
shapiro.test(res)
summary(logfit2)$r.squared
plot(cooks.distance(logfit2))
# The qq plot looks better than the original one, and only one point is far aw
ay from other points, but
# the r-squared decreased
```

```
        Shapiro-Wilk normality test

data:  res
W = 0.96628, p-value = 0.4431
```

0.624922333460508

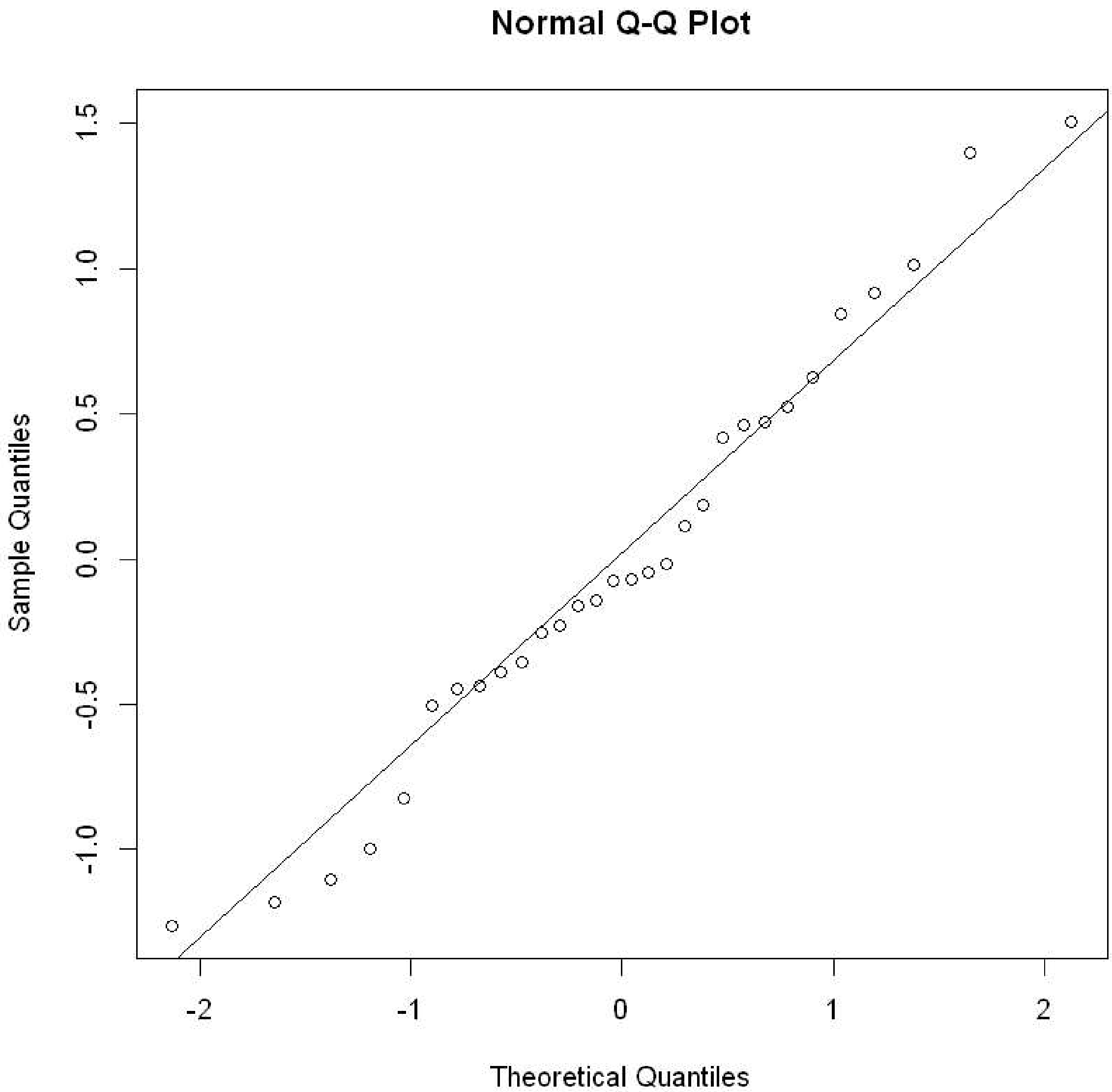


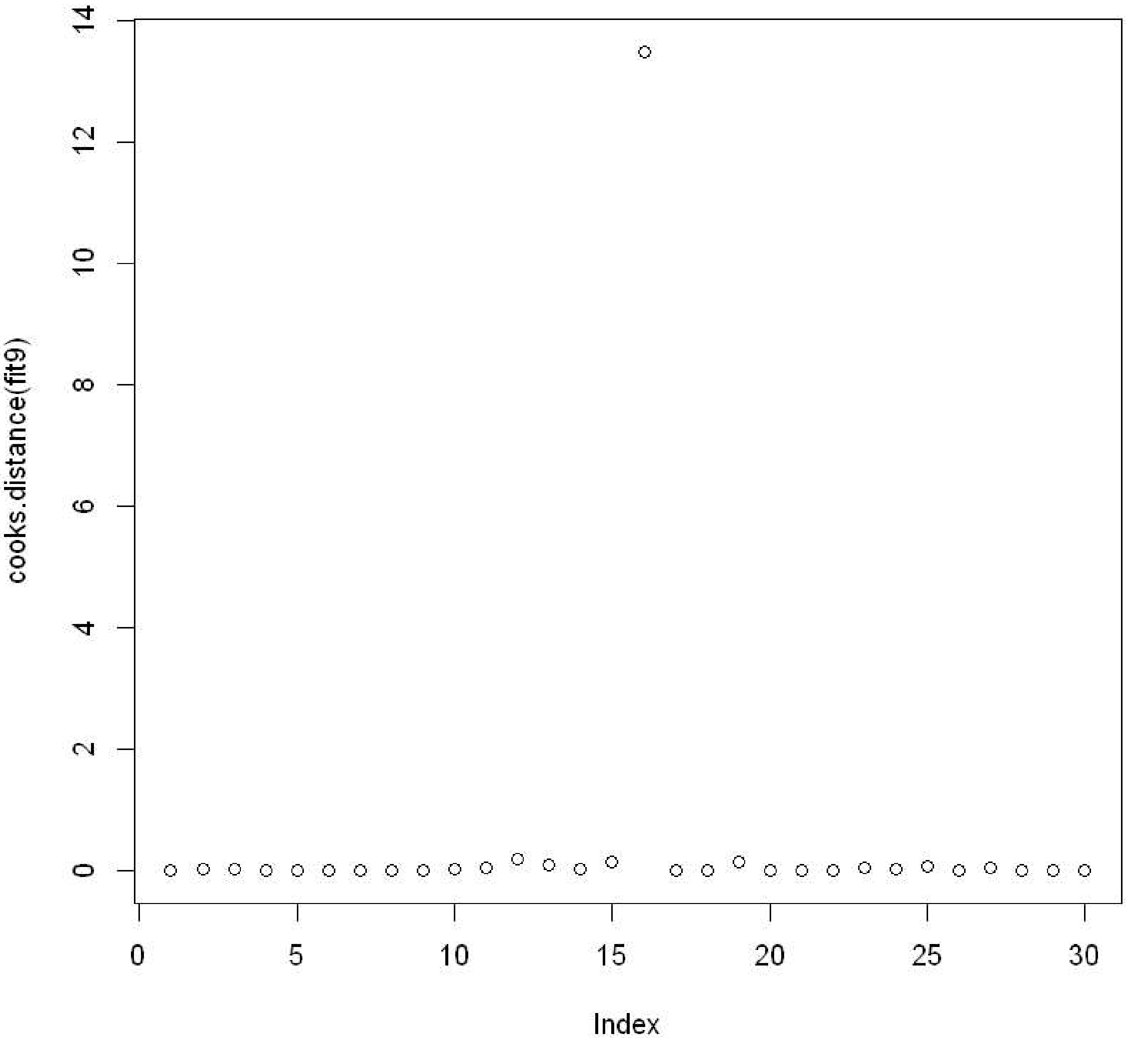

cooks.distance(logfit2)
0.0
0.5
1.0
1.5
0
5
10
15
20
25
30
Index

In [71]:

```
# Box-cox transformation
summary(fit9)$r.squared
qqnorm(fit9$residuals)
qqline(fit9$residuals)
plot(cooks.distance(fit9))
# Simliar situation with the above one, the points in normal qq plot are
# mostly line up for the qqline, and only one point is outlier. Whereas,
# both r-squared and adjusted r-squared has slightly decreasing.
```

0.746500088829006

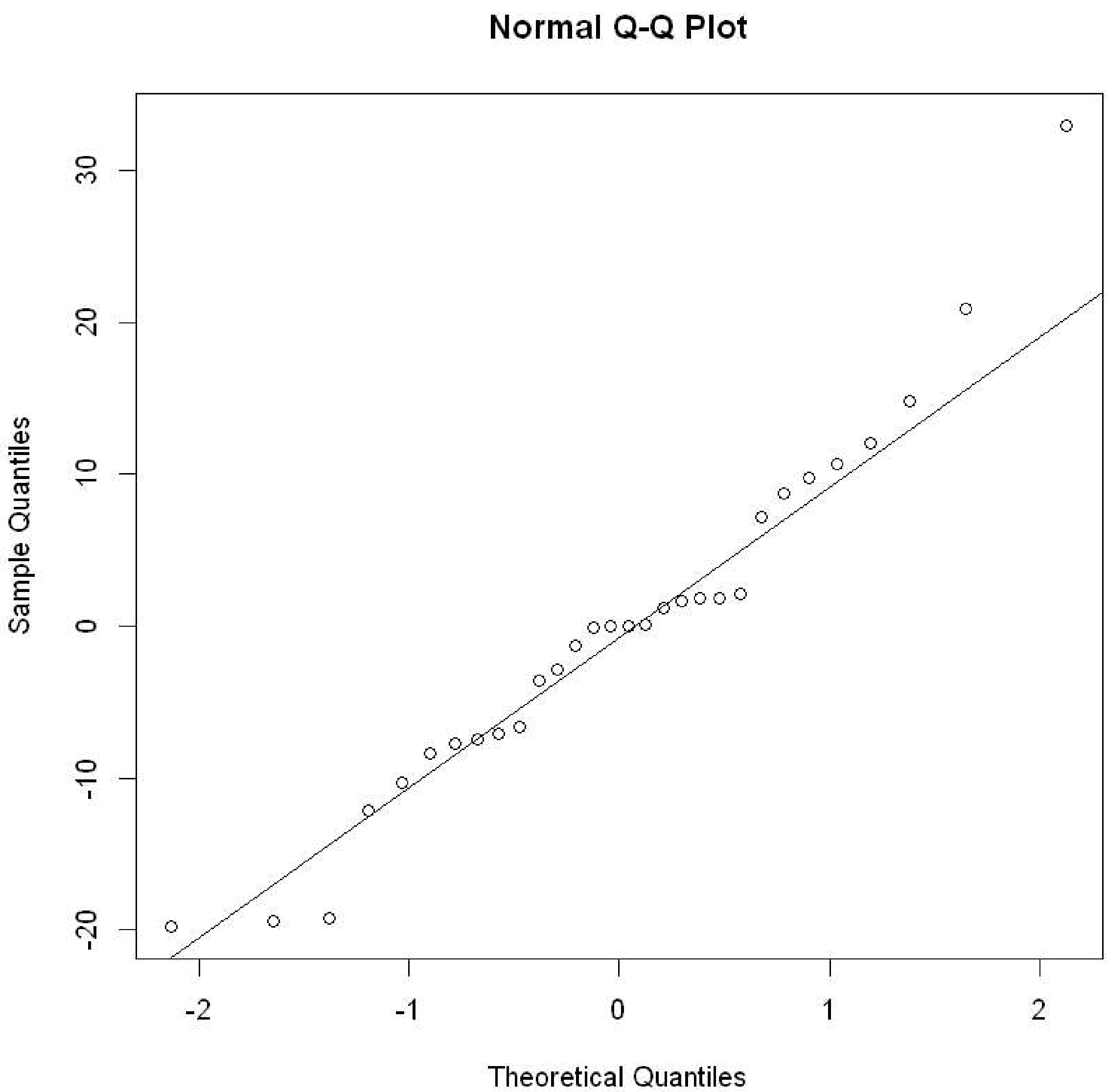


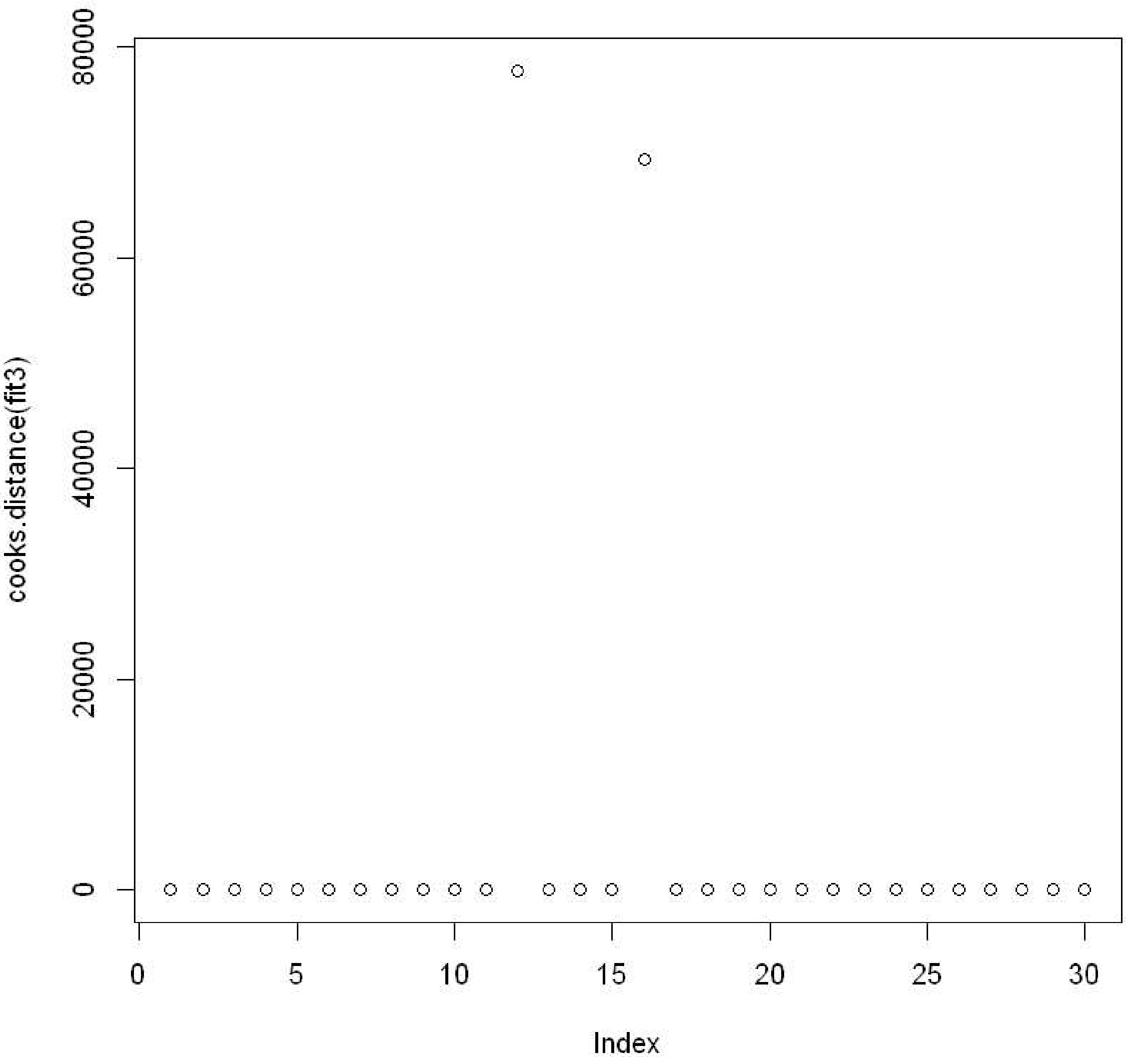

cooks.distance(fit9)
Index
0
2
4
6
8
10
12
14
0
5
10
15
20
25
30

In [68]:
```
#poly function
fit3 = lm(Species~poly(Area,Elevation,Nearest,Scruz,Adjacent,degree = 2),data
= gala)
qqnorm(fit3$residuals)
qqline(fit3$residuals)
res = fit3$residuals
shapiro.test(res)
summary(fit3)$r.squared
plot(cooks.distance(fit3))
# In poly function's qq plot, we could see that there are couple points in the
 middle
# are not line up in the qqline, and there are two points are extremely higher
 than the
# cut-off line, even though the r-squared is 0.989637, we consider it as bad t
ransformation.
```

```
	Shapiro-Wilk normality test

data:  res
W = 0.9574, p-value = 0.2654
```

0.989636980833444

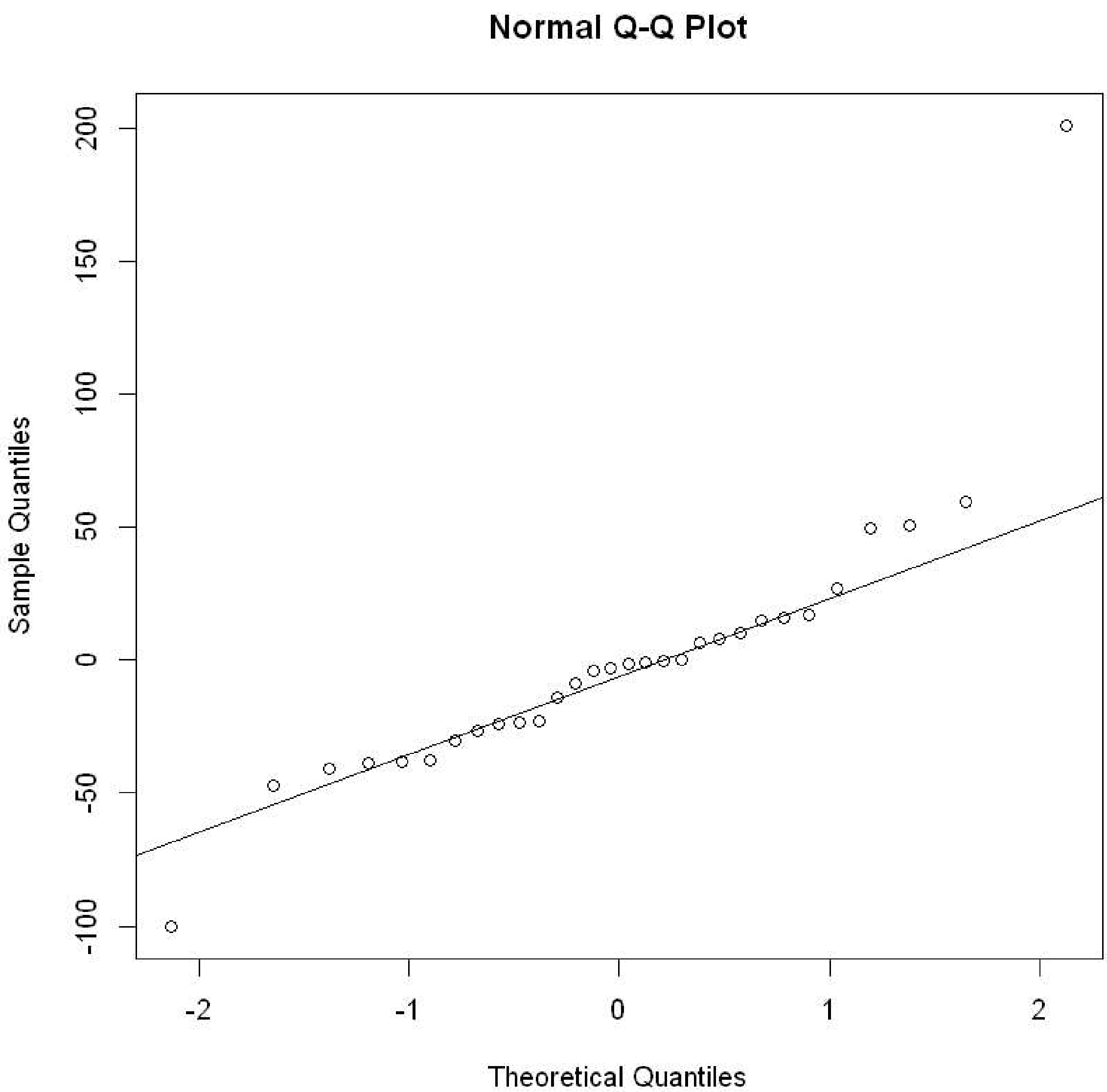


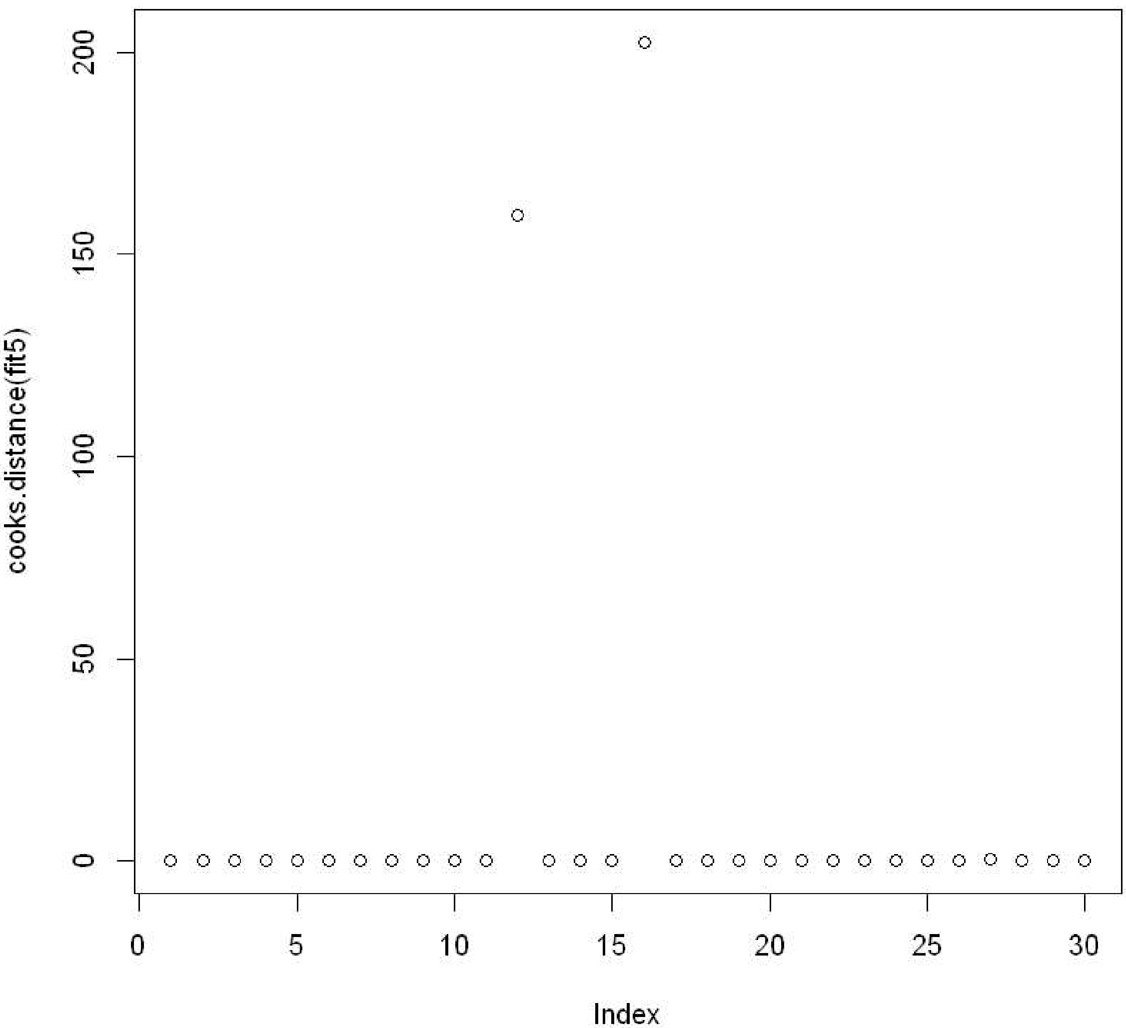

80000
60000
40000
20000
0
cooks.distance(fit3)
0
5
10
15
20
25
30
Index

In [74]:
```
#polym function
fit5 = lm(Species~polym(Area+Elevation+Nearest+Scruz+Adjacent,degree = 5),data
 = gala)
qqnorm(fit5$residuals)
qqline(fit5$residuals)
res = fit5$residuals
shapiro.test(res)
summary(fit5)$r.squared
plot(cooks.distance(fit5))
# In ploym function, the normal qq plot seems very ordered and line up
# on the qqline, and only two points are higher than the cut-off line, also th
e r-squared
# value increased as well, so this transformation might be a better fitting.
```

```
	Shapiro-Wilk normality test

data:  res
W = 0.80843, p-value = 9.374e-05
```

0.809345275919651

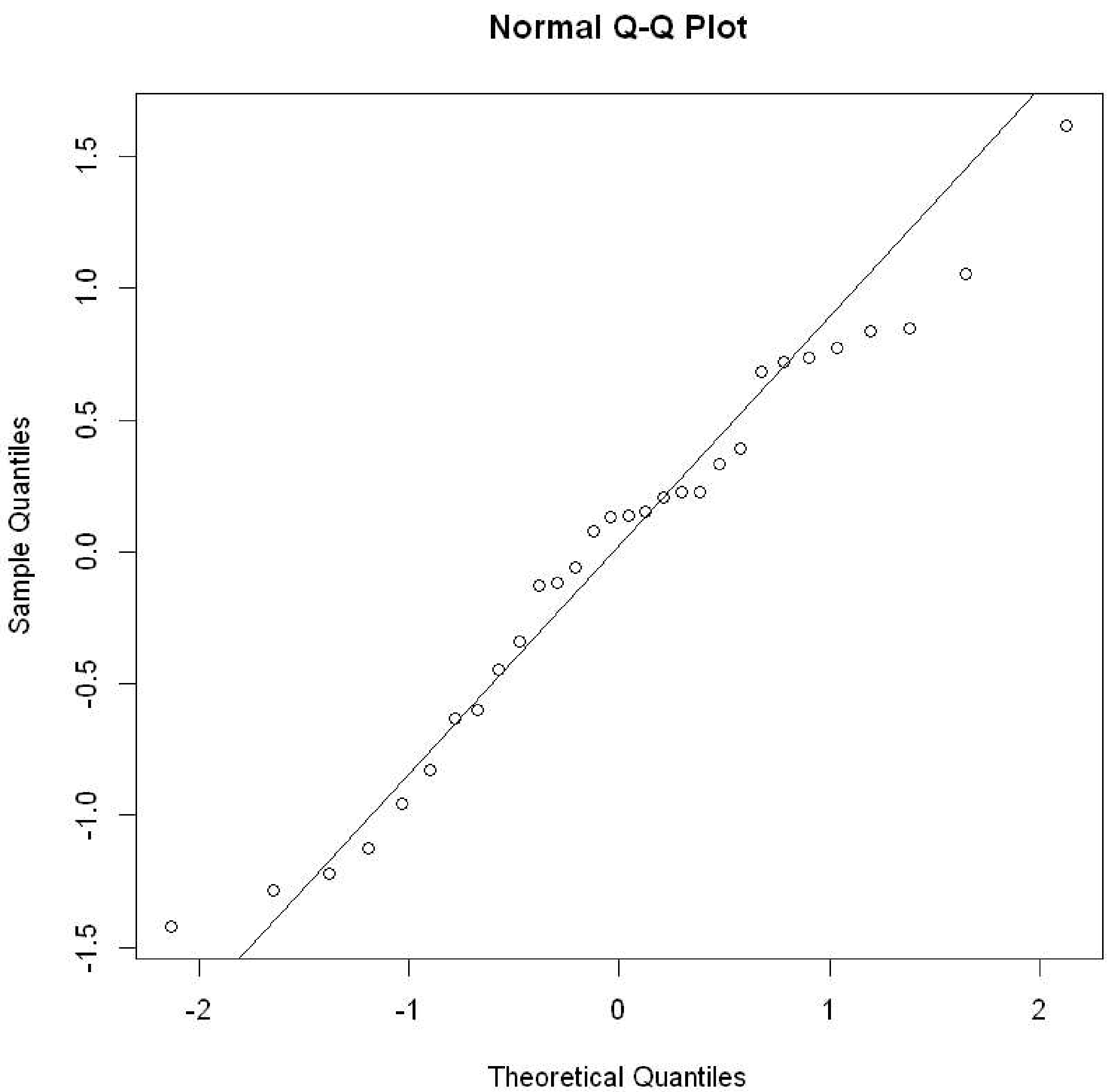


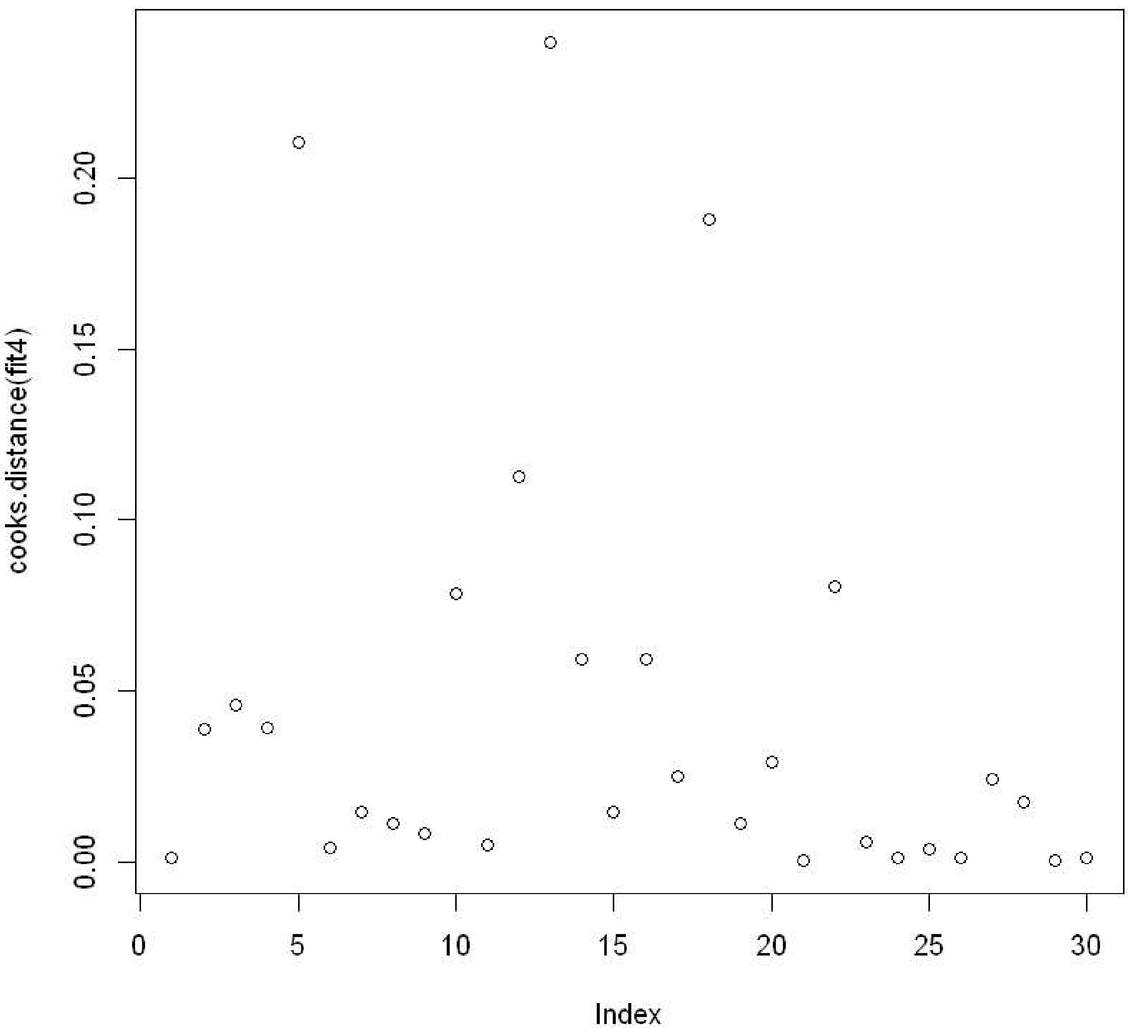

cooks.distance(fit5)
0
50
100
150
200
0
5
10
15
20
25
30
Index

In [75]:
```
# tranforming the X variables
fit4 =
lm(log(Species+1)~log(Area+1)+log(Elevation+1)+log(Nearest+1)+log(Scruz+1)+log
djacent+1),data = gala)
qqnorm(fit4$residuals)
qqline(fit4$residuals)
res = fit4$residuals
shapiro.test(res)
summary(fit4)$r.squared
plot(cooks.distance(fit4))
# First, we would like to add one in each terms in order to let this data be p
ositive.
# then, as the analysis of normal qqplot, the graph still has some gaps and th
e points
# are not actually lined up, and its p-value of normality test is 0.5543, whic
h says that
# it might not be a good model fitting.
# Thus, overall, we would like to conclude that ploynomial transformation or o
riginal model
# fitting is likely to have good fit.
```

```
	Shapiro-Wilk normality test

data:  res
W = 0.97055, p-value = 0.5543
```

0.726412821530776




cooks.distance(fit4)
0.00
0.05
0.10
0.15
0.20
0
5
10
15
20
25
30
Index