Yongkang Long 171022

Question 1

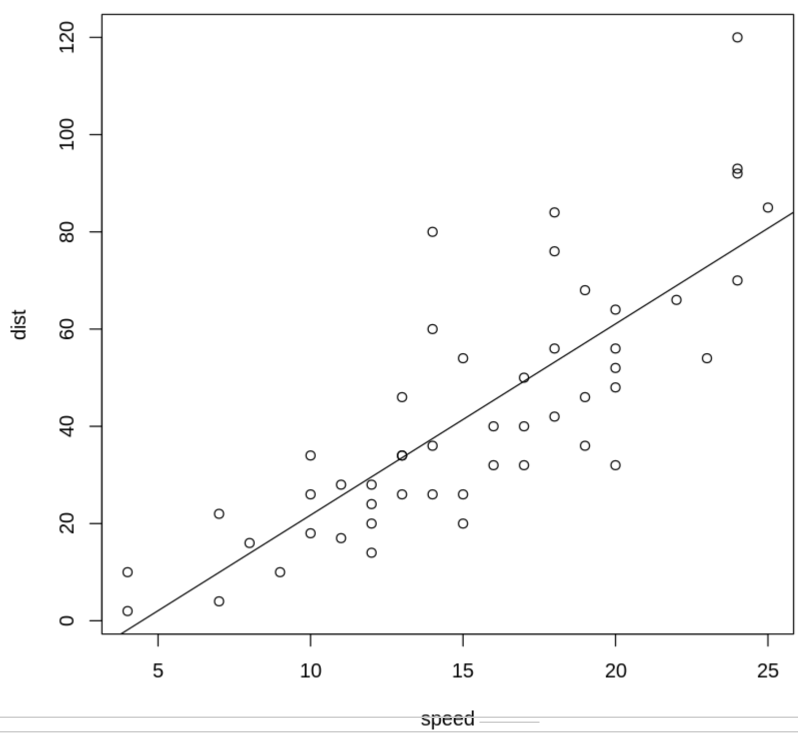
For this question we will use the data setcars.

(i)Plotdistas a function ofspeed. Fit a simple linear regression model ofdistas a function ofspeed.Add the regression line to the previous plot. Obtain the summary for this regression. Obtain anestimator for the error variance. Observe the value for the intercept and comment.

In [1]:
```
library(car)
```

```
Loading required package: carData
```

In [2]:
```
 plot(dist ~ speed, data = cars)
abline(lm(dist ~ speed, data = cars))
```



No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [3]:
```r
 model1 <- lm(dist ~ speed, data = cars)
summary(model1)
```

```
Call:
lm(formula = dist ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-29.069  -9.525  -2.272   9.215  43.201 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -17.5791     6.7584  -2.601   0.0123 *  
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,	Adjusted R-squared:  0.6438 
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

In [4]:
```r
#the estimated variance is
summary(model1)$sigma^2
```

236.531688564477

```
The incercept is -17.5791, which means that the barking distacne is -17.5791 miles for speed 0. However, it is not true in real. Look at the p-value is 0.0123, marginally significant. If we use alpha=0.01, we acept the null hypothesis that incercept is 0.
```

(ii)Based on your comments to the previous section, fit a model without an intersect. Draw a scatterplotand add the two regression lines. Obtain a summary for the new regression and comment on thedifferences with the previous model, including the estimated error variance.

In [5]:
```r
model2 <- lm(dist ~ -1 + speed, data = cars)
summary(model2)
```

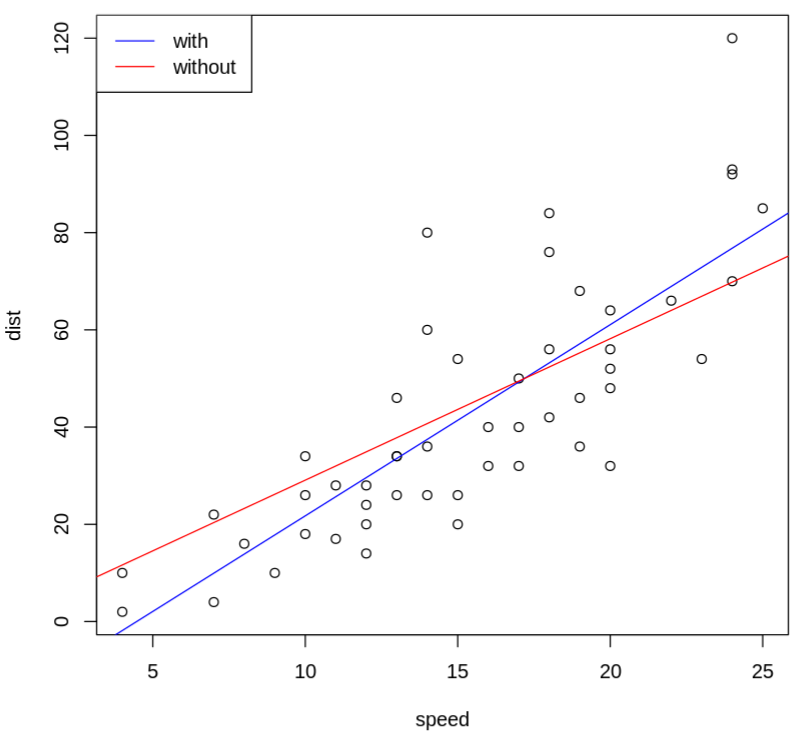
No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

```
Call:
```

In [6]:
```
#the estimated variance is
summary(model2)$sigma^2
```

264.36279259209

In [7]:
```
plot(dist ~ speed, data = cars)
abline(model1,col = 'blue')
abline(model2,col = 'red')
legend('topleft',c('with','without'),col=c('blue','red'),lty=c(1,1))
```




```
The R^2 without intercept is higher that R^2 with intercept and the estimated variance is no difference. It means we prefer the model without intercept. Actually, there is no intercept for dist and speed in real life
```

(iii)We want to compare the predictive power of these two models. Using the same procedure as in exercise1 of the list for week 9, compare the predictive power of both models and comment on your results

In [8]:
```
attach(cars)
```

In [9]:
```
n = length(dist)
```

No documentation for 'cex' in specified packages and libraries: you could try '??cex'

In [10]:
```
predictvalue <- matrix(rep(0,2*n), ncol = 2)
```

In [11]:
```
for (i in 1:n) {
    xx <- speed[-i]
    yy <- dist[-i]
    model1 <- lm(yy ~ xx)
    model2 <- lm(yy ~ -1 + xx)
    pred.values[i,1] = predict(model1,data.frame(xx = speed[i]))
    pred.values[i,2] = predict(model2,data.frame(xx = speed[i]))
}
pred.values <- cbind(pred.values,dist)
colnames(pred.values) <- c('P1', 'P2', 'O')
```

In [12]:
```
head(pred.values)
```

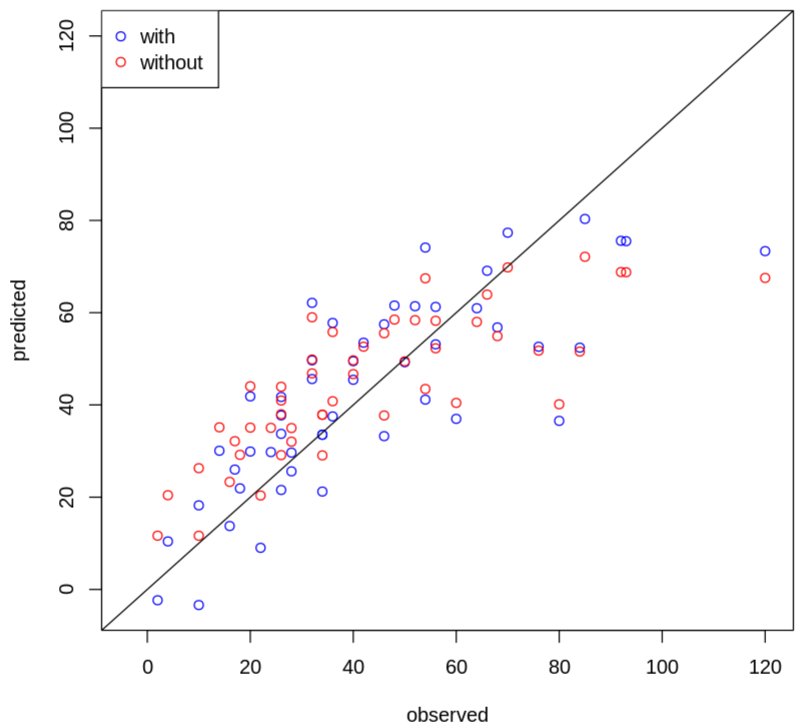
A matrix: 6 × 3 of type dbl

P1	P2	O
-2.348991	11.64820	2
-3.387122	11.63851	10
10.405805	20.42477	4
9.019622	20.35784	22
13.744937	23.30842	16
18.222888	26.28189	10

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [13]:

```
plot(pred.values[,3],pred.values[,1],
     col = 'blue', ylab = 'predicted',
     xlab = 'observed',
     xlim = (range(pred.values)+c(-.5,+.5)),
     ylim = (range(pred.values)+c(-.5,+.5)))
points(pred.values[,3],pred.values[,2], col = 'red')
legend('topleft',c('with','without'),col=c('blue','red'),pch=c(1,1))
abline(0,1)
```

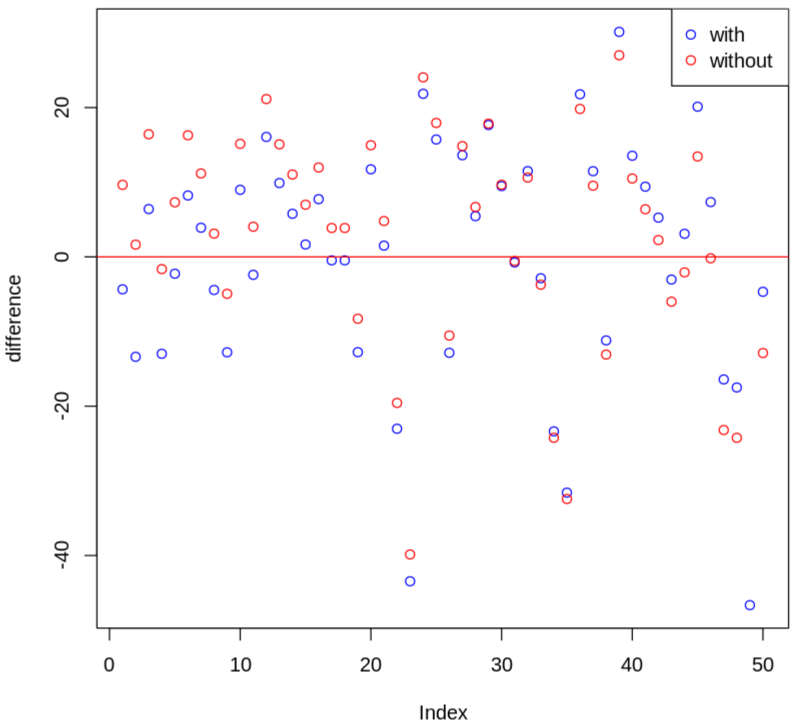



```
Only Slightly different between predicted values with and without intercept
```

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [14]:

```
plot(pred.values[,1]-pred.values[,3], col = 'blue', ylab = 'differenc
abline(h=0,col='red')
points(pred.values[,2]-pred.values[,3], col = 'red')
legend('topright',c('with','without'),col=c('blue','red'),pch=c(1,1)
```




```
Only Slightly different between errors with and without intercept
```

In [15]:

```
#mean absoulte errors (MAE) of model with intercept
mean(abs(pred.values[,1]-pred.values[,3]))
```

12.0591786486375

In [16]:

```
#mean absoulte errors (MAS) of model with outintercept
mean(abs(pred.values[,2]-pred.values[,3]))
```

12.9782367734261

```
No difference betwween MAE with two models. Removing intercept is
powerful
```

Question 2

For this question use the data setdata1

No documentation for 'cex' in specified packages and libraries: you could try '??cex'

(i)Read the data and draw a scatterplot ofvar1as a

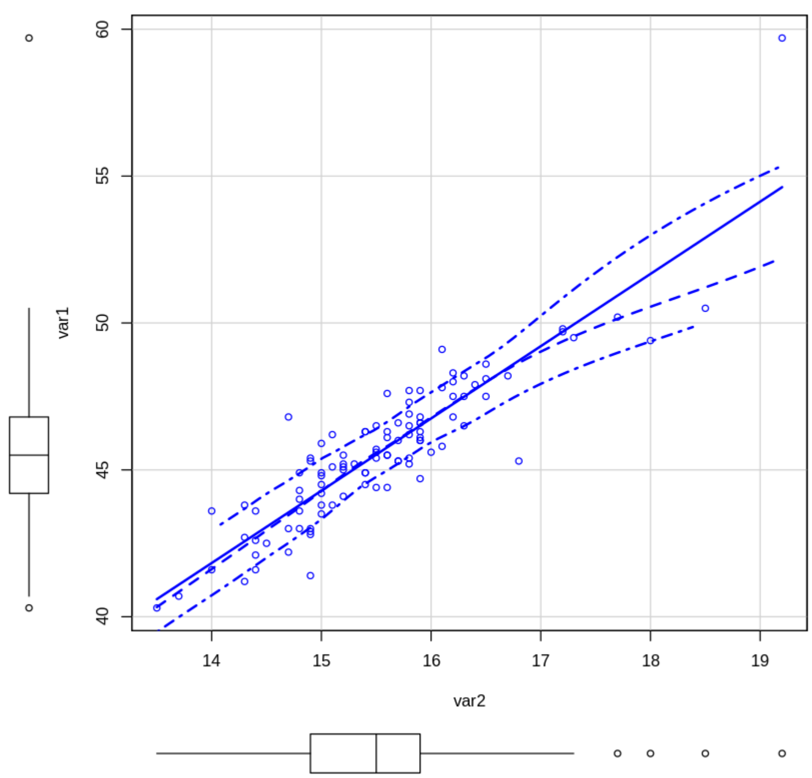
function ofvar2. Fit a simple linear regression andadd the line to the plot. Comment. Obtain a summary of the regression.

In [17]:
```
data1 = read.table('data1')
head(data1)
attach(data1)
```

A data.frame: 6 × 2

	var1	var2
	<dbl>	<dbl>
1	46.8	15.9
2	45.2	15.2
3	46.6	15.9
4	44.9	15.0
5	46.1	15.6
6	45.1	15.2

In [18]:
```
scatterplot(var1 ~ var2, data = data1)
```

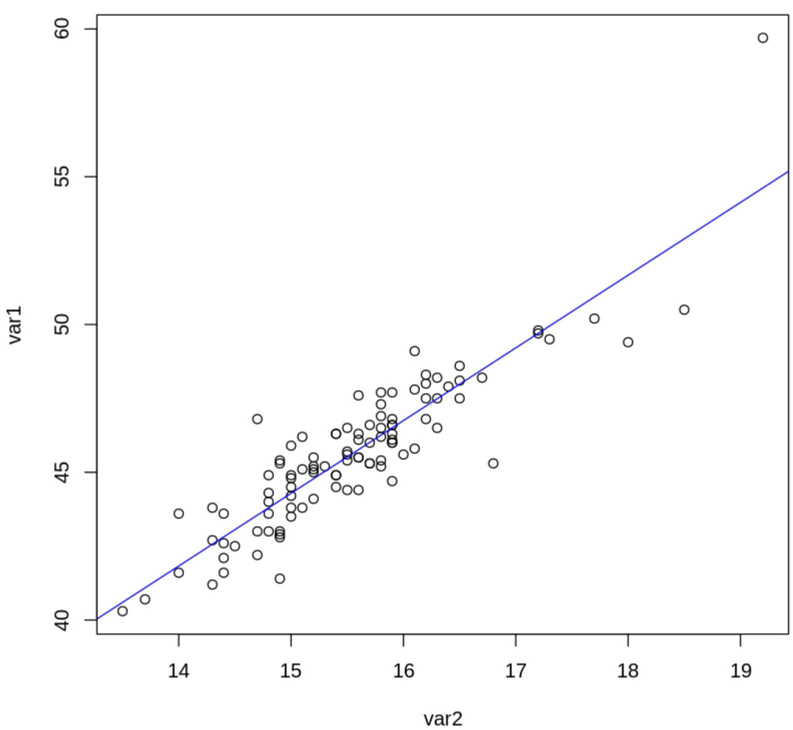



```
The local smooth regression line fit the regression line in low
values but is different in high values
```

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [19]:
```
model1 <- lm(var1 ~ var2, data = data1)
plot(var1 ~ var2, data = data1)
abline(model1,col = 'blue')
summary(model1)
```

```
Call:
lm(formula = var1 ~ var2, data = data1)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.4183 -0.7043 -0.0072  0.6049  5.0765 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   7.3826     1.9110   3.863 0.000199 ***
var2          2.4605     0.1227  20.060  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.152 on 100 degrees of freedom
Multiple R-squared:  0.801,	Adjusted R-squared:  0.799 
F-statistic: 402.4 on 1 and 100 DF,  p-value: < 2.2e-16
```



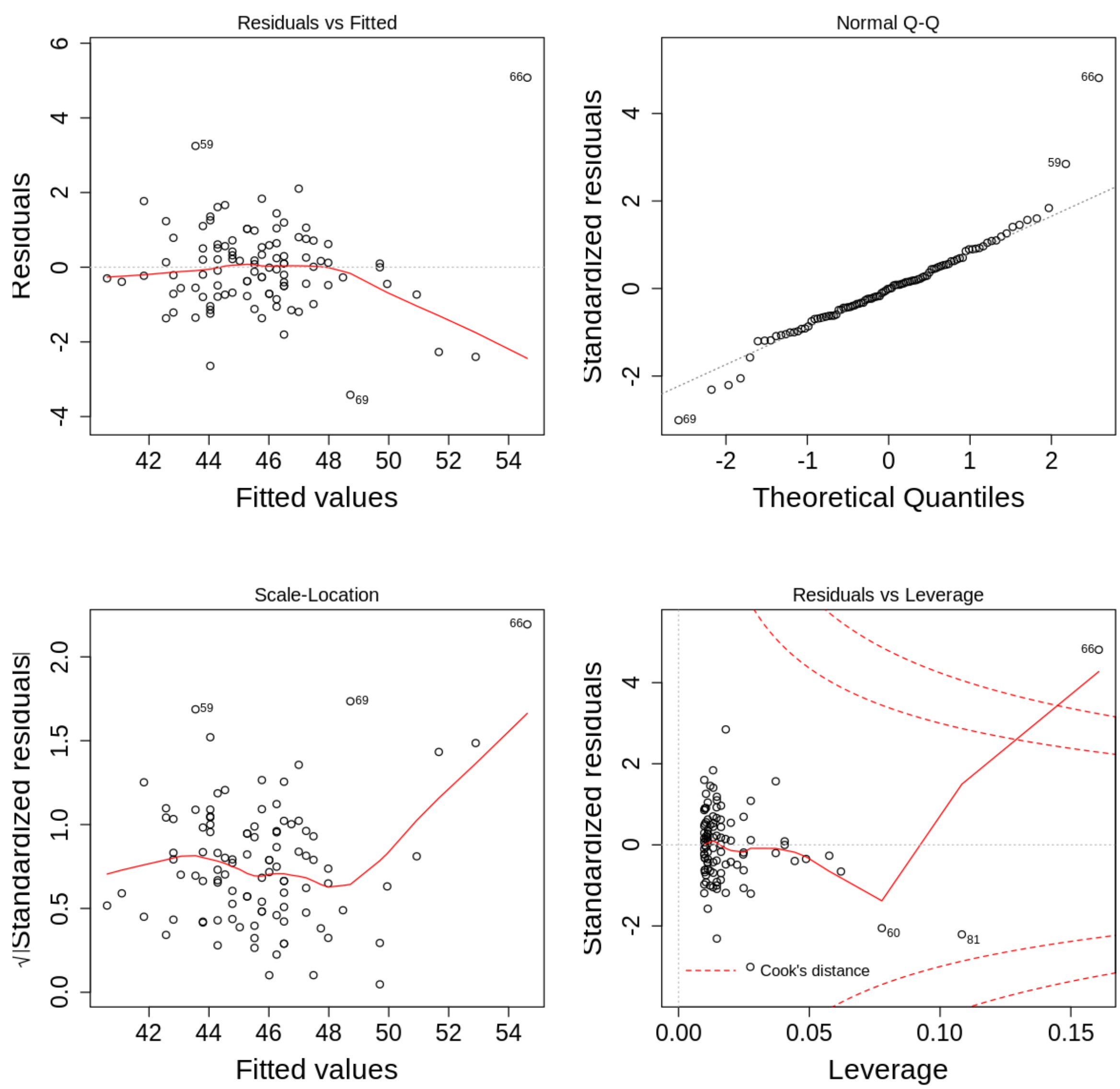

```
The p value is significant and R^2 is 0.8, it seems good. Howwver,
we can obeserve an outlier obviouly which means that there is room
for improvement.
```

No documentation for 'cex' in specified packages and libraries: you could try '??cex'

(ii)Draw the diagnostic plots. Do you identify any point as an outlier? If you do, which point is this? Canyou identify this point in the initial scatterplot?

In [20]:

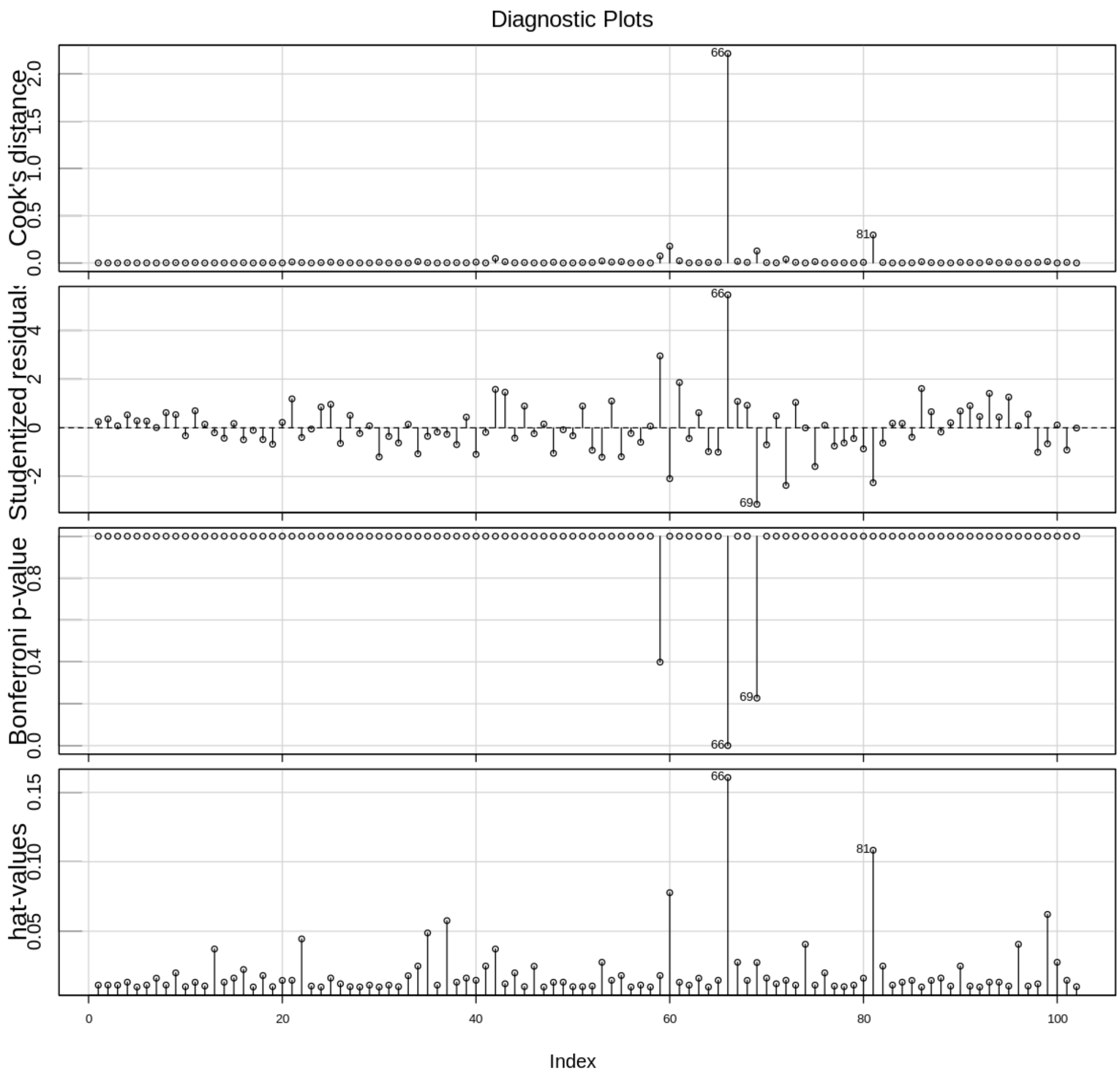
```
options(repr.plot.width=10, repr.plot.height=10)
par(mfrow = c(2,2))
plot(model1,cex.axis=1.5,cex.lab=1.8,ps=10)
par(mfrow = c(1,1))
```




```
The residuals plot and QQ plot shows the variance are uniform and normality is valid. Outlier 66 has very large cook distance.  60 and 81 maybe outlier but with cook's distance < 0.5
```

No documentation for 'cex' in specified packages and libraries: you could try '??cex'

In [21]:
```
influenceIndexPlot(model1,cex.lab=2,cex.axis=1.5)
```



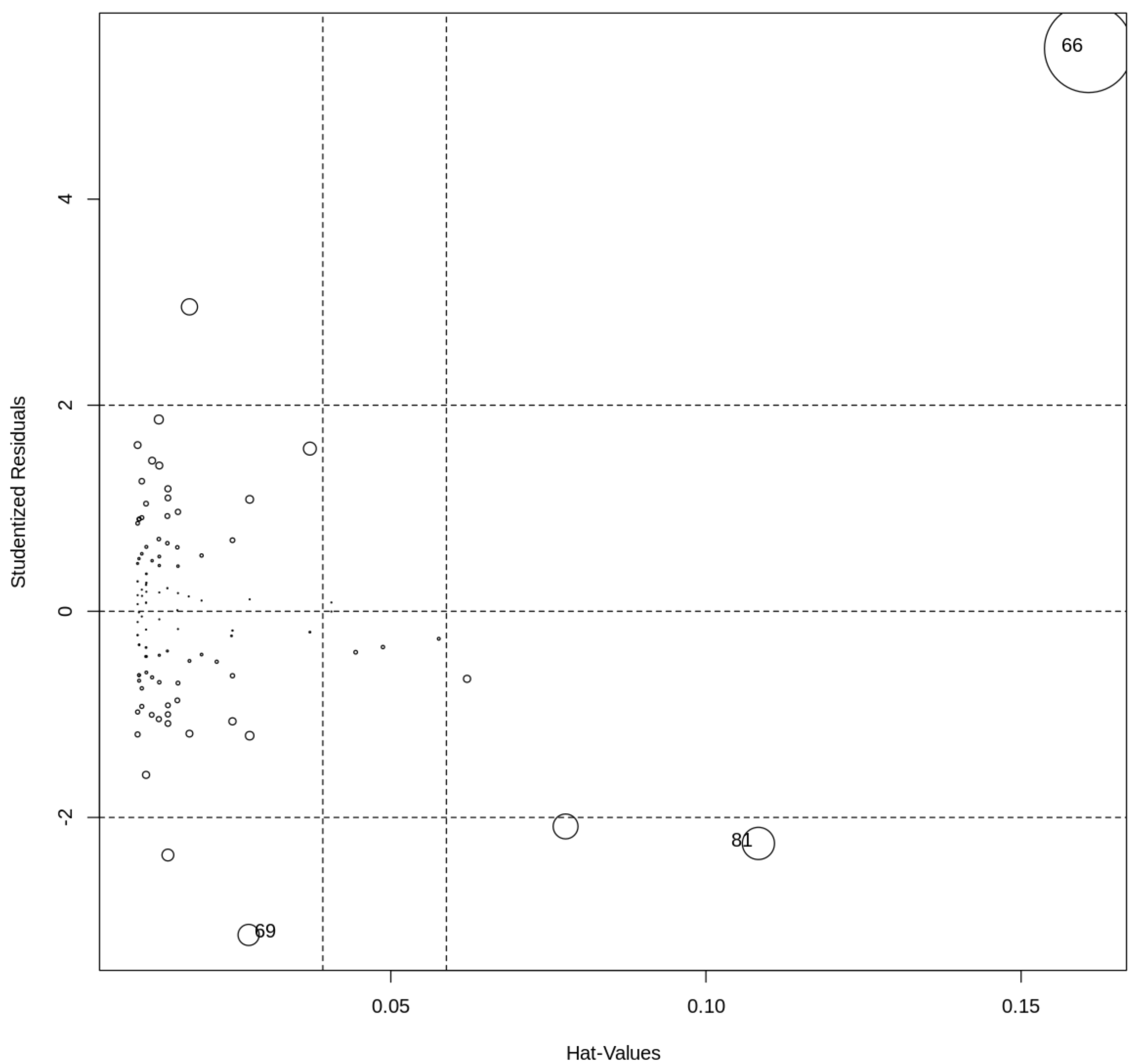
No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [22]:

```
influencePlot(model1)
```

A data.frame: 3 × 3

	StudRes	Hat	CookD
	<dbl>	<dbl>	<dbl>
66	5.461408	0.16068783	2.2163170
69	-3.140454	0.02744525	0.1278291
81	-2.252573	0.10832633	0.2961511



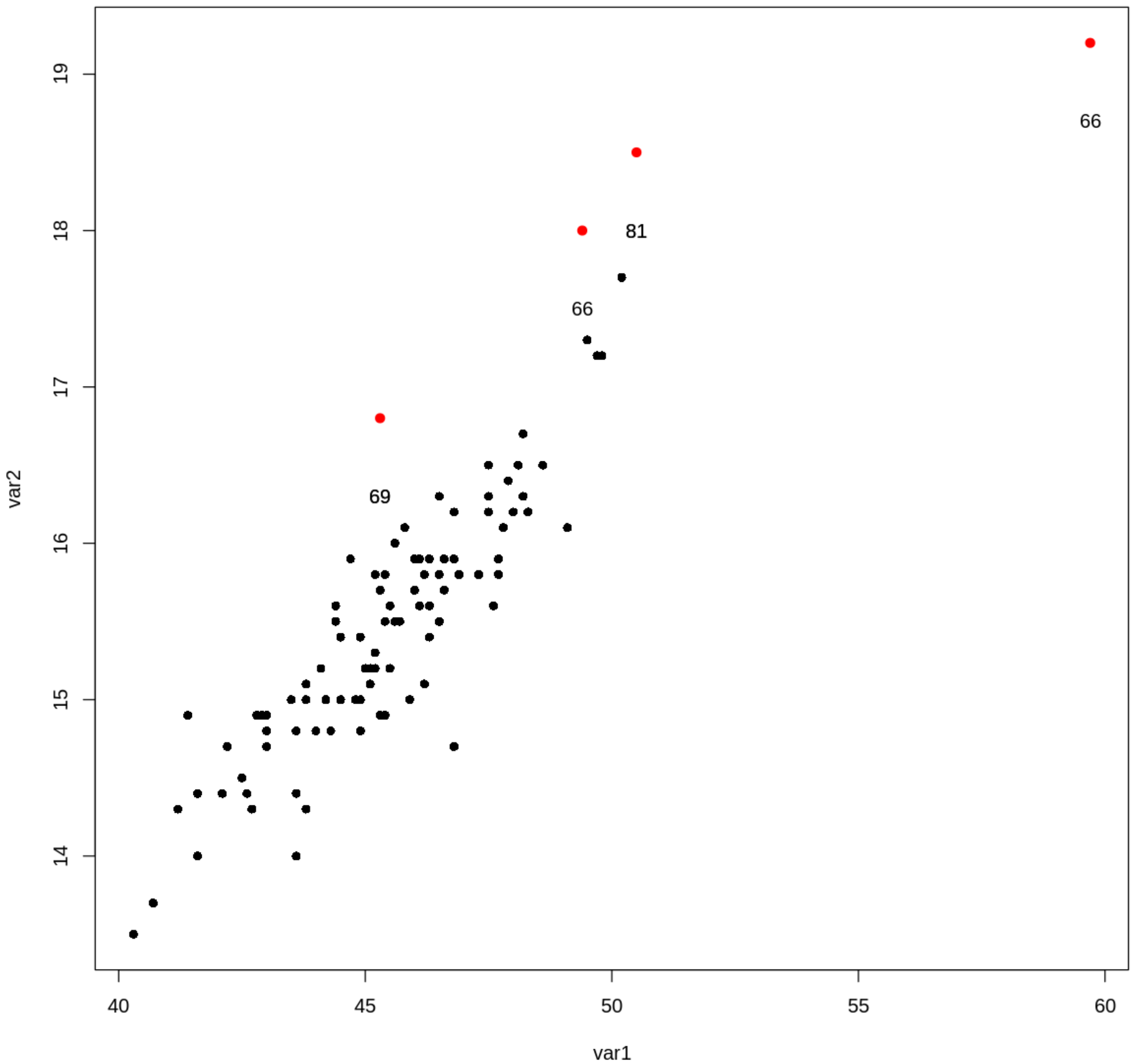

```
We now get three outliers 60,69,66,and 81
```

In [23]:

```
plot(var1 ,var2,pch=16)
#summary(model2)
points(var1[c(60,66,69,81)], var2[c(60,66,69,81)],pch=19, col='red')
text(var1[60],var2[60]-.5, '66');
```

No documentation for 'cex' in specified packages and libraries: you could try '??cex'

```
text(var1[69],var2[69]-.5,'69');
text(var1[81],var2[81]-.5,'81');
text(var1[66],var2[66]-.5,'66');
text(var1[69],var2[69]-.5,'69');
text(var1[81],var2[81]-.5,'81');
```




```
We could not identify 60, 69 and 81 point in the initial scatterplot
```

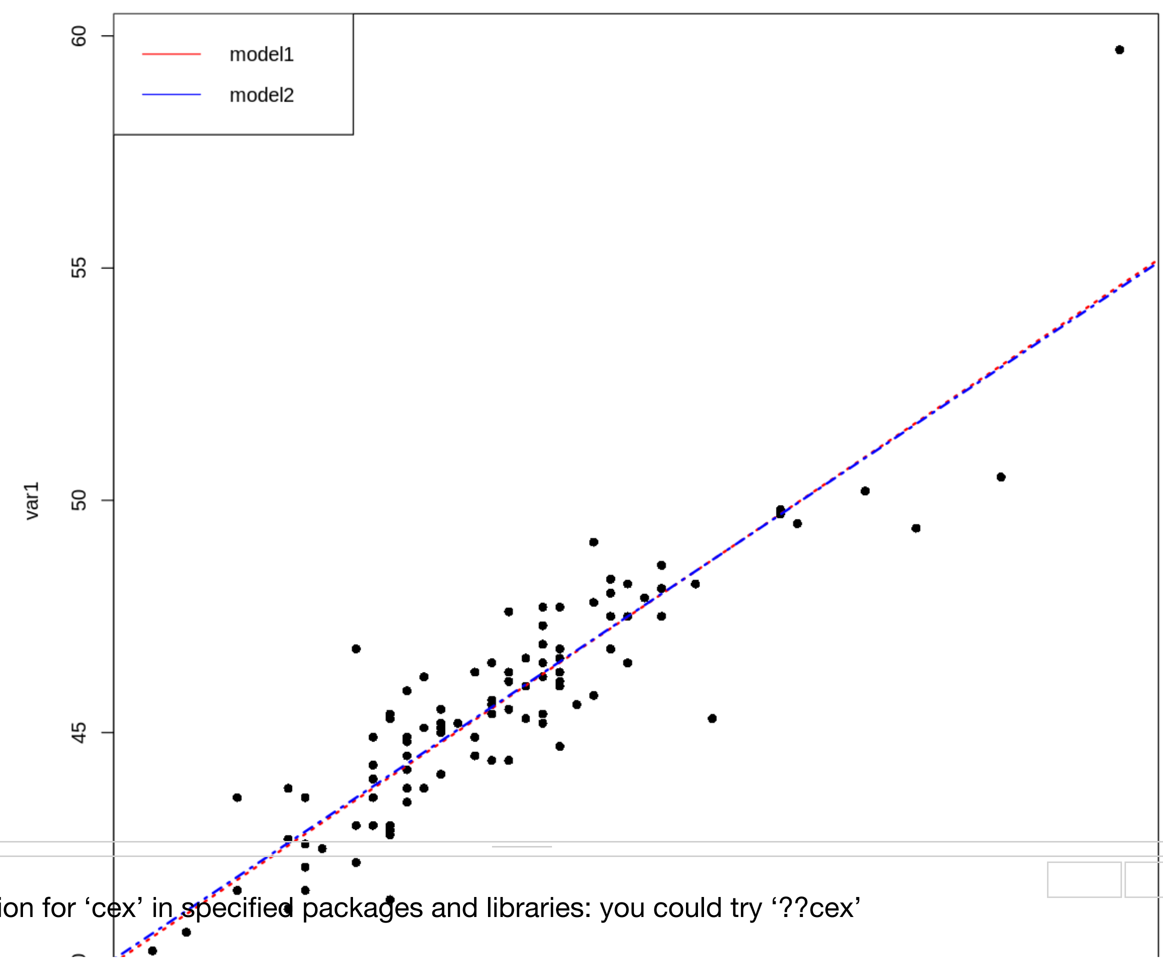
(iii)Fit a new regression model excluding the outlier(s) that you identified in the previous section. Draw ascatterplot with both regression lines. Compare the summary tables. Draw the diagnostic plots andcomment.

In [24]:
```
options(repr.plot.width=10, repr.plot.height=10)
data1_eo = data1[-c(60,66,69,81),]
```

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

```r
model2 <- lm(var1 ~ var2, data = data1_eo)
plot(var1 ~ var2,data=data1,pch=16)
abline(model1,col = 'red',lty=3, lwd=2)
abline(model2,col = 'blue',lty=4, lwd=2)
legend('topleft',c('model1','model2'),col=c('red','blue'),lty=c(1,1))
summary(model2)

```

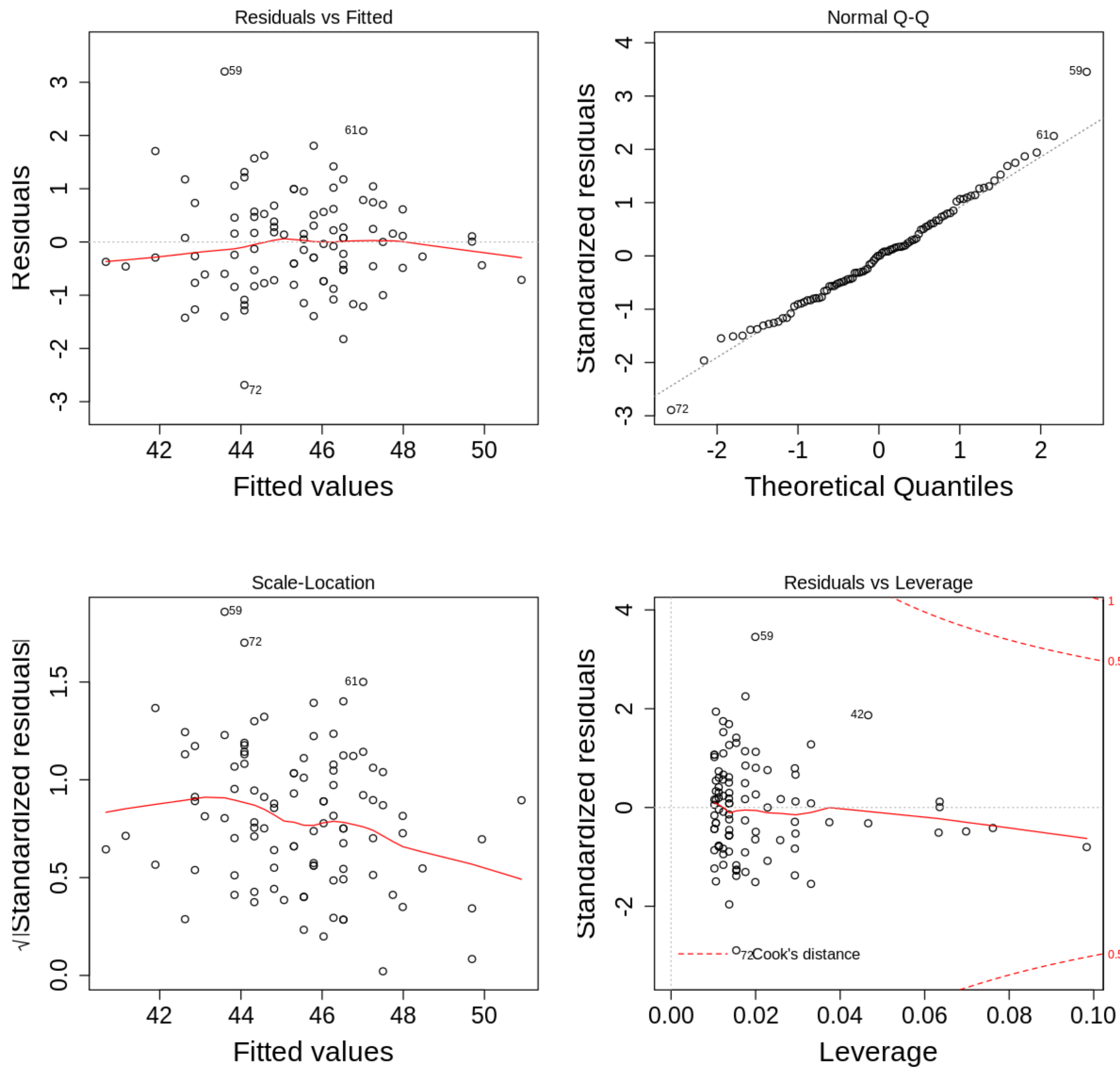
```
Call:
lm(formula = var1 ~ var2, data = data1_eo)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.6866 -0.6084  0.0034  0.5680  3.2010 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   7.7628     1.9086   4.067 9.75e-05 ***
var2          2.4378     0.1234  19.756  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9361 on 96 degrees of freedom
Multiple R-squared:  0.8026,	Adjusted R-squared:  0.8005 
F-statistic: 390.3 on 1 and 96 DF,  p-value: < 2.2e-16
```



No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

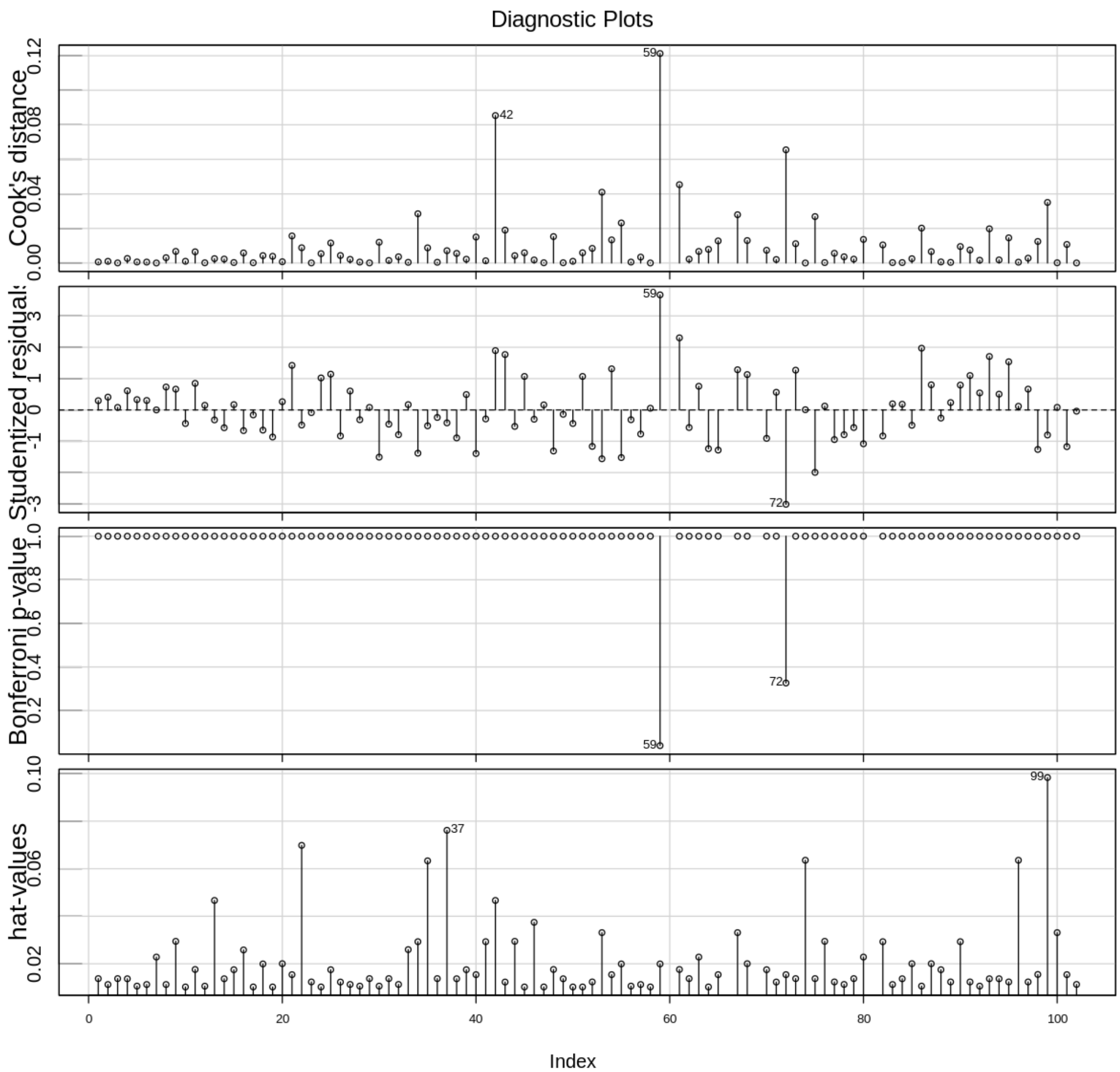
```
The intercept,slope,and R^2 do not change significantly.
```

In [25]:
```
options(repr.plot.width=10, repr.plot.height=10)
par(mfrow = c(2,2))
plot(model2,cex.axis=1.5,cex.lab=1.8,ps=10)
par(mfrow = c(1,1))
```



No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [26]:
```
influenceIndexPlot(model2,cex.lab=2,cex.axis=1.5)
```

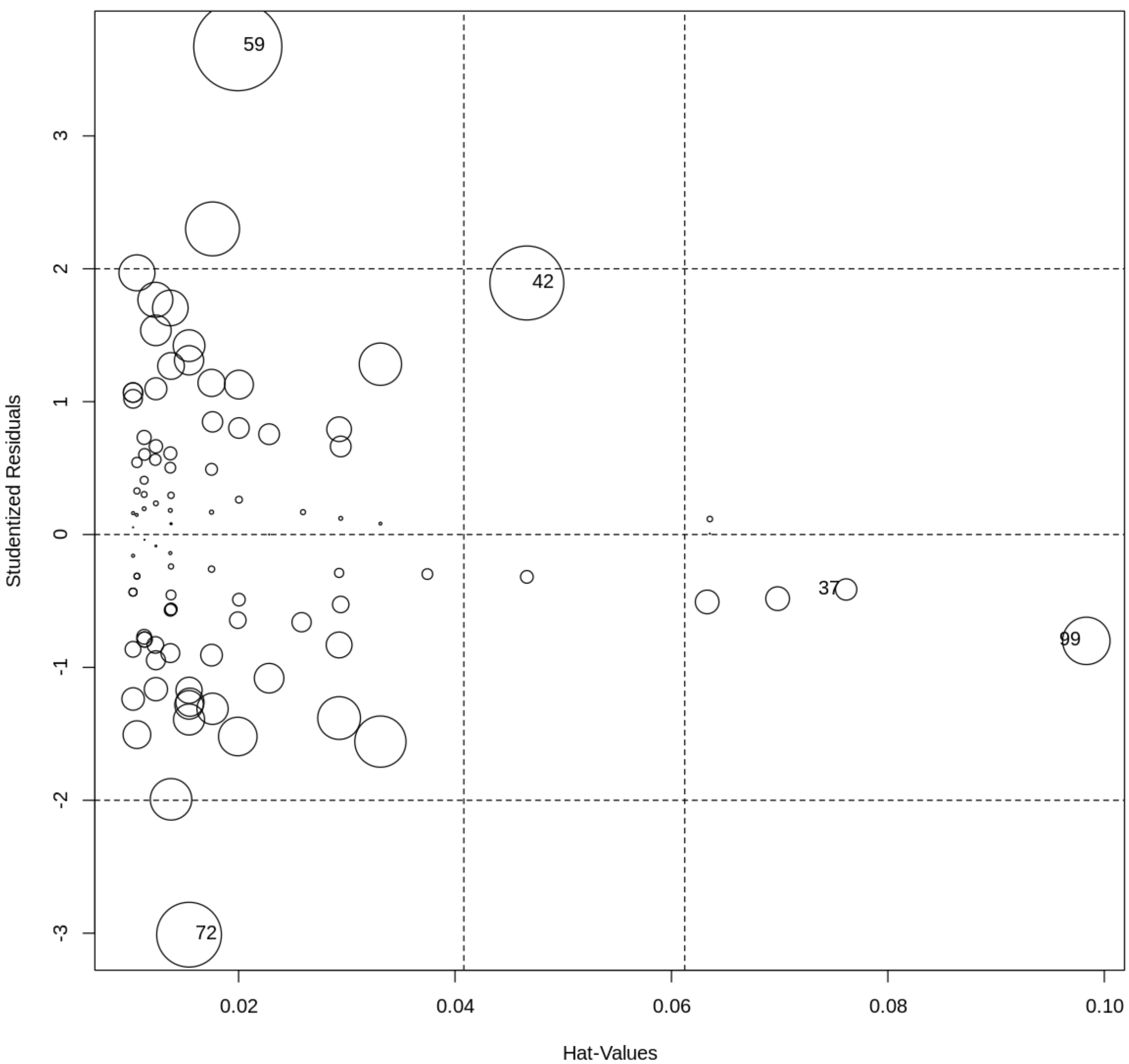


No documentation for 'cex' in specified packages and libraries: you could try '??cex'

In [27]:
```
influencePlot(model2)
```

A data.frame: 5 × 3

	StudRes	Hat	CookD
	<dbl>	<dbl>	<dbl>
37	-0.4134772	0.07614503	0.007106858
42	1.8930925	0.04663819	0.085361742
59	3.6718182	0.01992598	0.121284644
72	-3.0115733	0.01542194	0.065522891
99	-0.8001545	0.09833900	0.035045381




```
Now the diagnostic plot looks better
```

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

Question 3

For this question use the data setdata2.

(i)Readdata2and plotyvalas a function ofxval. Fit a simple linear regression and add the regressionline to the plot. Comment. Obtain a summary for the regression and draw the diagnostic plots.Comment on the results

In [28]:

```
data2 = read.table('data2')
head(data2)
```

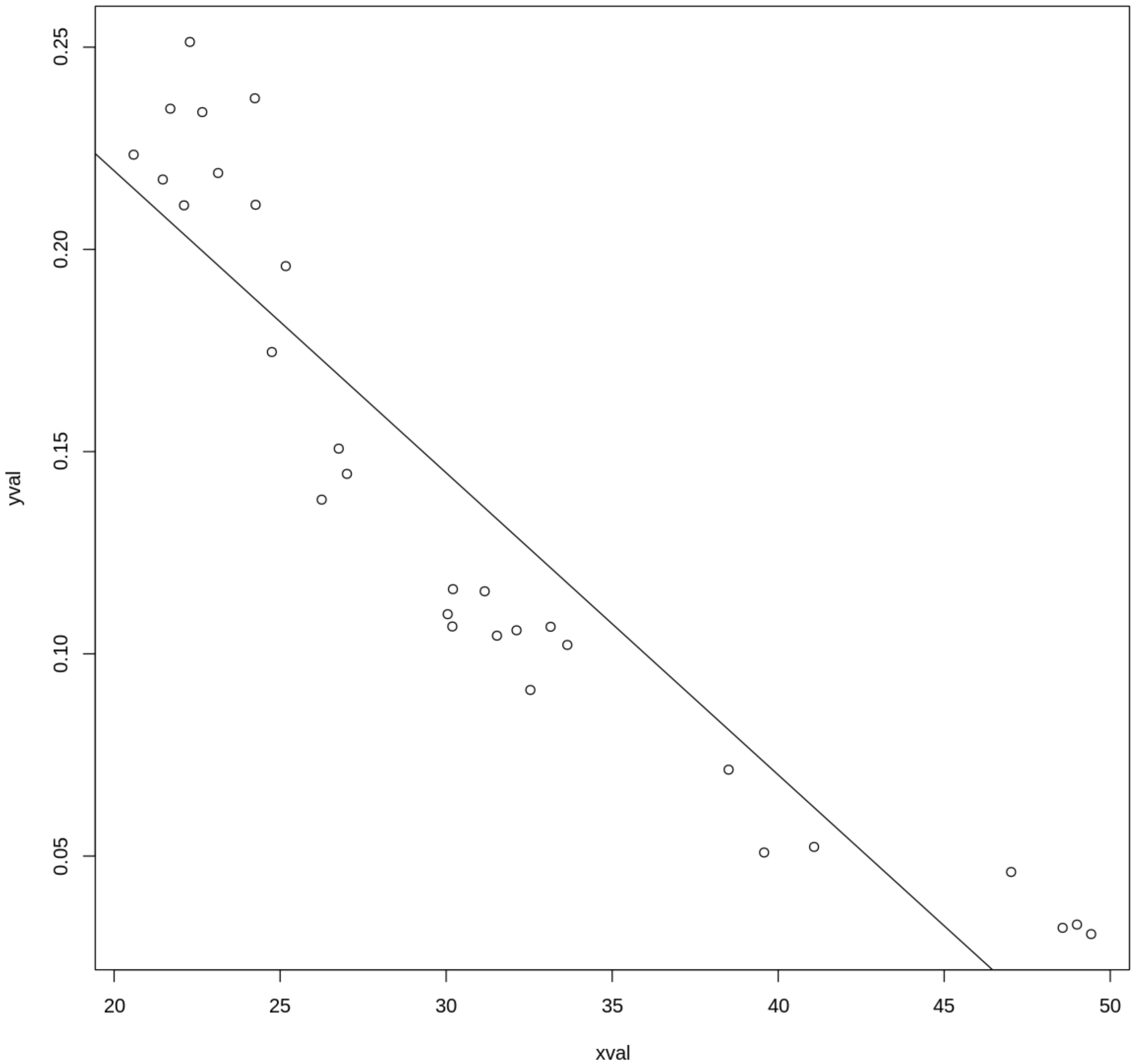
A data.frame: 6 × 2

	xval	yval
	<dbl>	<dbl>
1	31.52991	0.1044869
2	33.14513	0.1066878
3	32.11964	0.1058220
4	31.16068	0.1154637
5	32.53664	0.0910629
6	23.13162	0.2188879

In [29]:

```
options(repr.plot.width=10, repr.plot.height=10)
plot(yval ~ xval, data = data2)
model1 <- lm(yval ~ xval, data = data2)
abline(model1)
summary(model1)
```

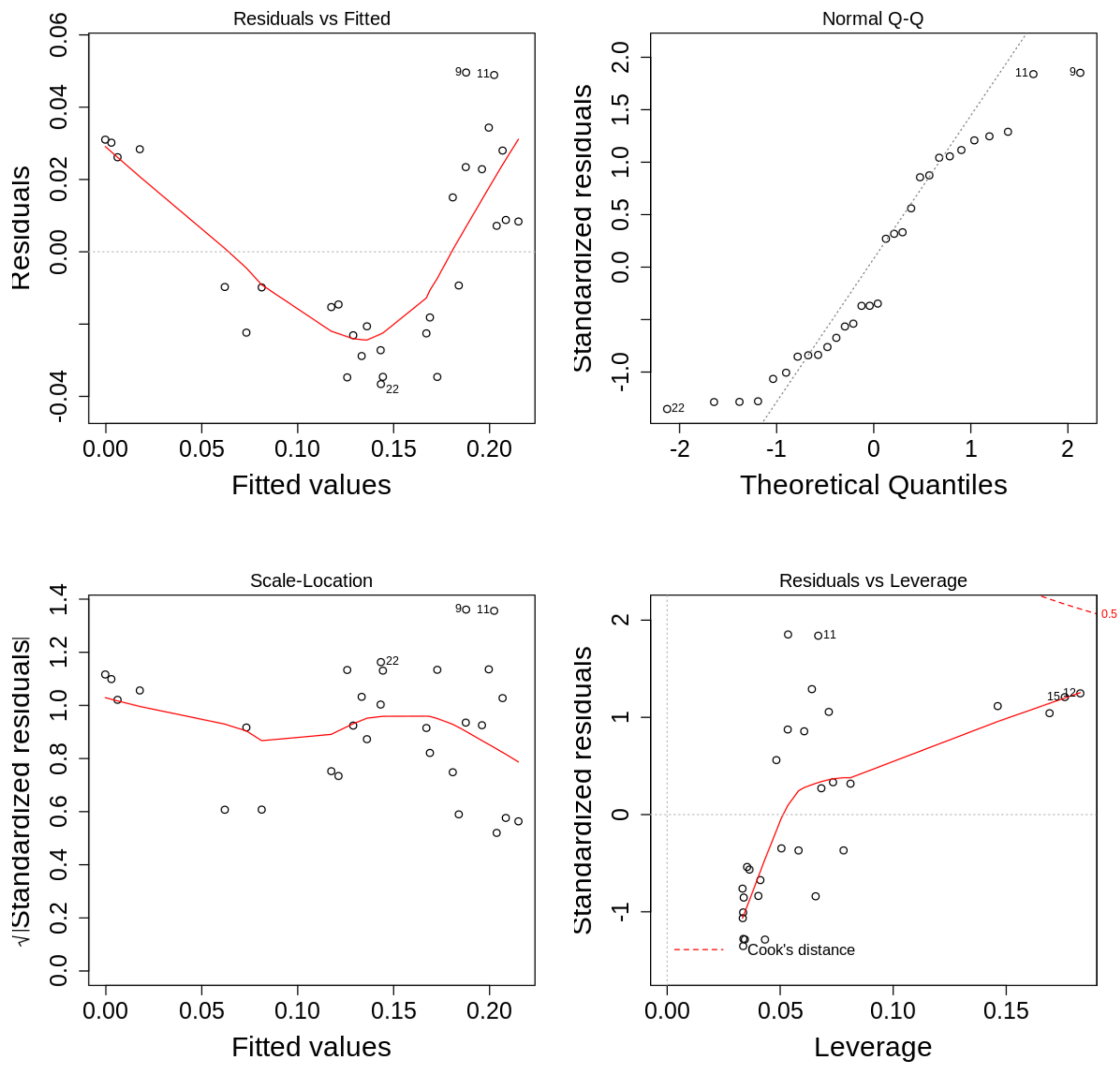
No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’




```
1 The p value is significant and R^2 is 0.85, it seems good.
```

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [30]:
```
options(repr.plot.width=10, repr.plot.height=10)
par(mfrow = c(2,2))
plot(model1,cex.axis=1.5,cex.lab=1.8,ps=10)
par(mfrow = c(1,1))
```




```
From residuals plot, we know that there is a quadratic pattern but not for the standardized residuals plot and normality do not fit the model
```

In [31]:
```
ncvTest(model1)
#### The variances are uniform
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.03720501, Df = 1, p = 0.84705
```

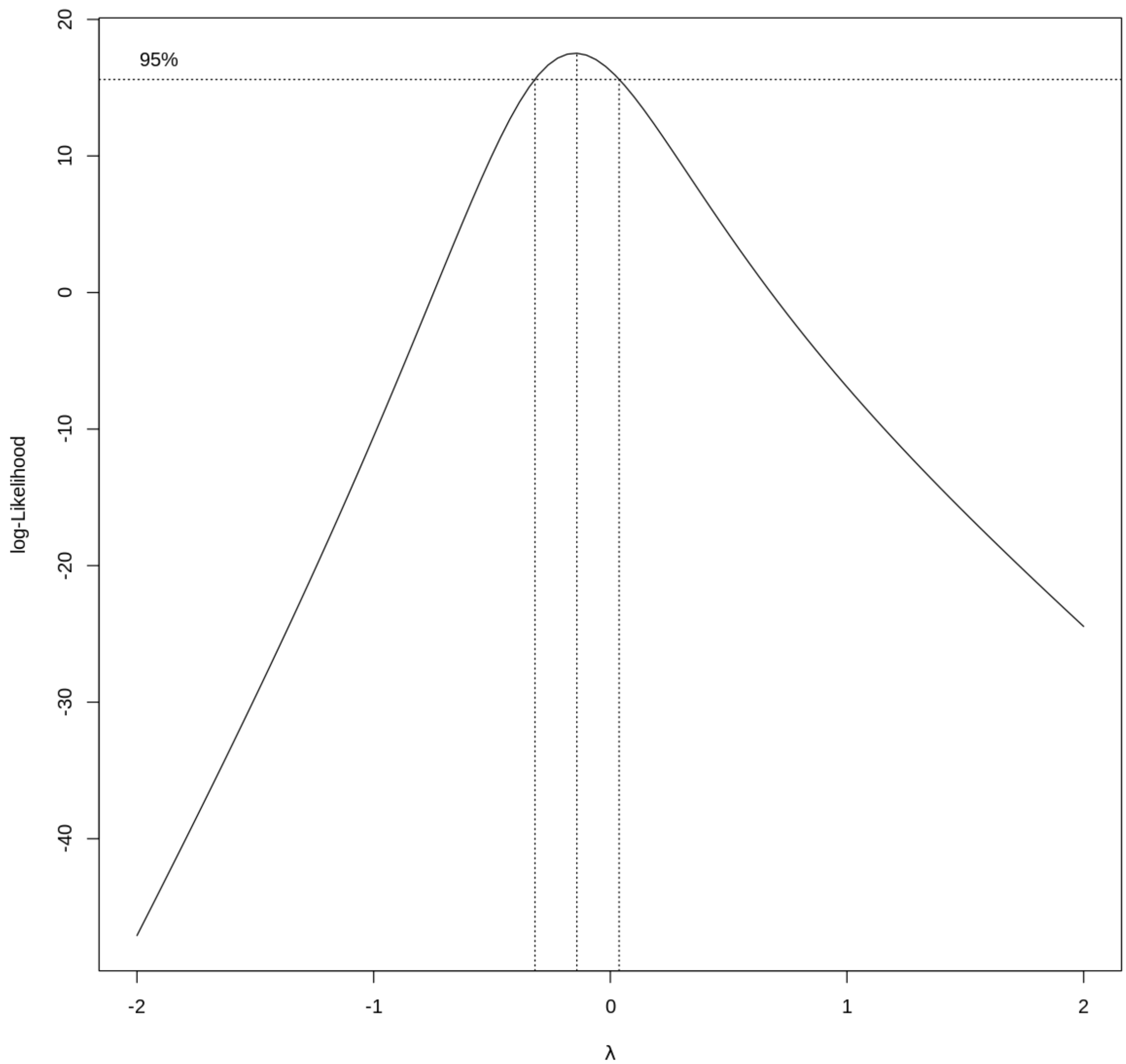
In [32]:
```
shapiro.test(model1$residuals)
#####normality assumption is not satisfied
```

No documentation for 'cex' in specified packages and libraries: you could try '??cex'

```
        Shapiro-Wilk normality test

data:  model1$residuals
W = 0.91828, p-value = 0.0242
```

(ii) Use the functionboxcoxon the packageMASSwith the argument set to the model you fitted in (i).

In [33]:
```
library(MASS)
boxcox(model1)
```




```
Becuase 0 is i the interval, we can chose lamda = 0 and use
log(transform)
```

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

(iii)If the confidence interval in the graph includes zero,

use a logarithmic transformation foryvaland fit anew model. Obtain a summary of the new regression and compare with the previous one. Draw thediagnostic plots and compare with the previous results.

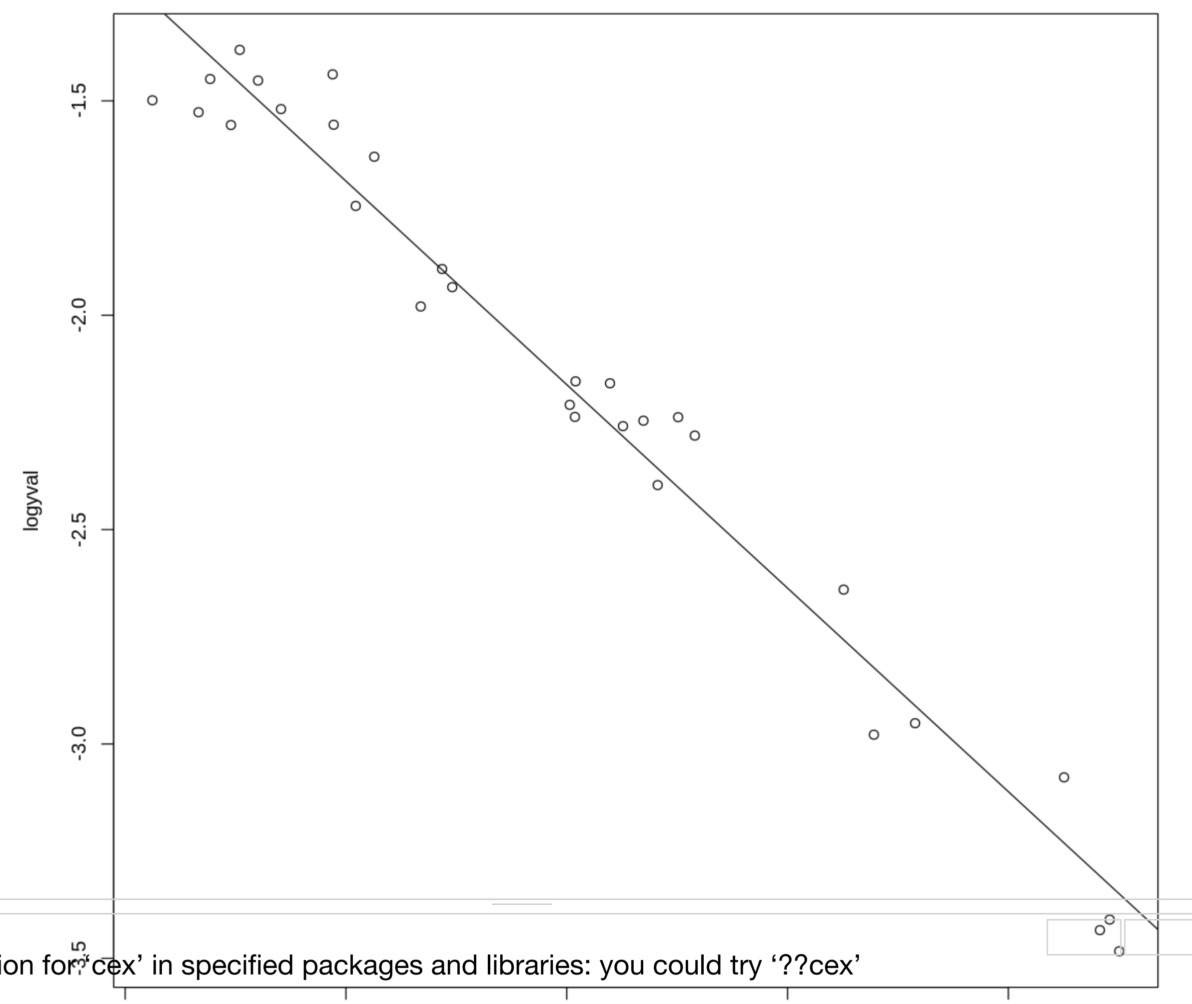
In [34]:

```
data2$logyval = log(data2$yval)
data2$logxval = log(data2$xval)
```

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [35]:
```
plot(logyval ~ logxval, data = data2)
model2 <- lm(logyval ~ logxval, data = data2)
abline(model2)
summary(model2)
```

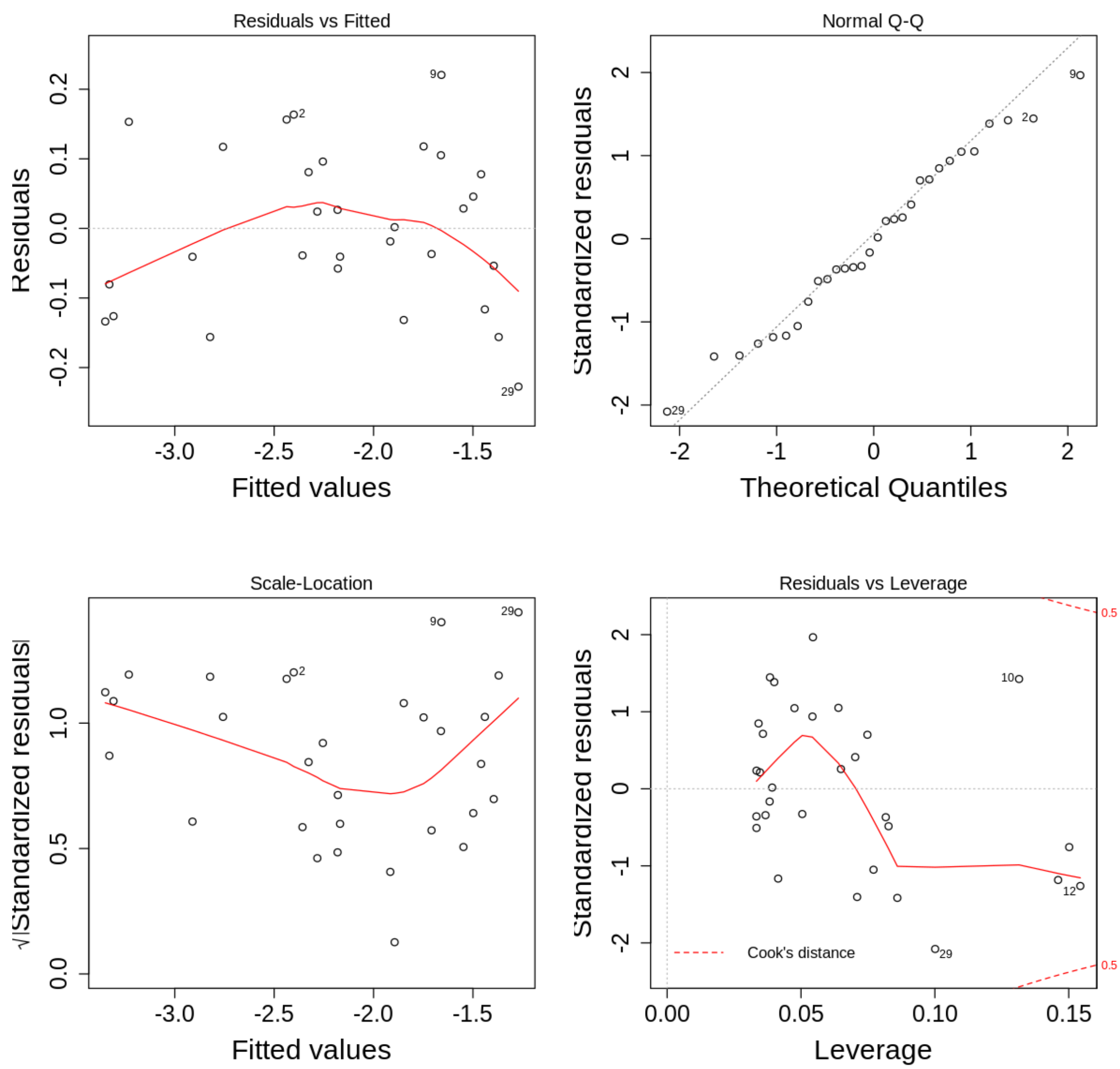
```
Call:
lm(formula = logyval ~ logxval, data = data2)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.227479 -0.074795 -0.008457  0.092254  0.220557 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.90701    0.27207   21.71   <2e-16 ***
logxval     -2.37322    0.07983  -29.73   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1153 on 28 degrees of freedom
Multiple R-squared:  0.9693,    Adjusted R-squared:  0.9682 
F-statistic: 883.8 on 1 and 28 DF,  p-value: < 2.2e-16
```



No documentation for 'cex' in specified packages and libraries: you could try '??cex'

```
It is much better than the previous one!  R^2 reach to 0.97!
```

In [36]:
```
par(mfrow = c(2,2))
plot(model2,cex.axis=1.5,cex.lab=1.8,ps=10)
par(mfrow = c(1,1))
```




```
The plot shows that varivance is uniform and normality is stasfied
```

In [37]:
```
ncvTest(model2)
#### The variances are uniform
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.03014715, Df = 1, p = 0.86216
```

In [38]:
```
shapiro.test(model2$residuals)
#### normality assumption is now satisfied
```

No documentation for 'cex' in specified packages and libraries: you could try '??cex'

```
        Shapiro-Wilk normality test

data:  model2$residuals
W = 0.97749, p-value = 0.7557
```

(iv)Write down the final model in terms of the original variables. Draw a scatterplot ofyvalagainstxvaland add the regression line for the first model and the curve you obtained with the second regression.

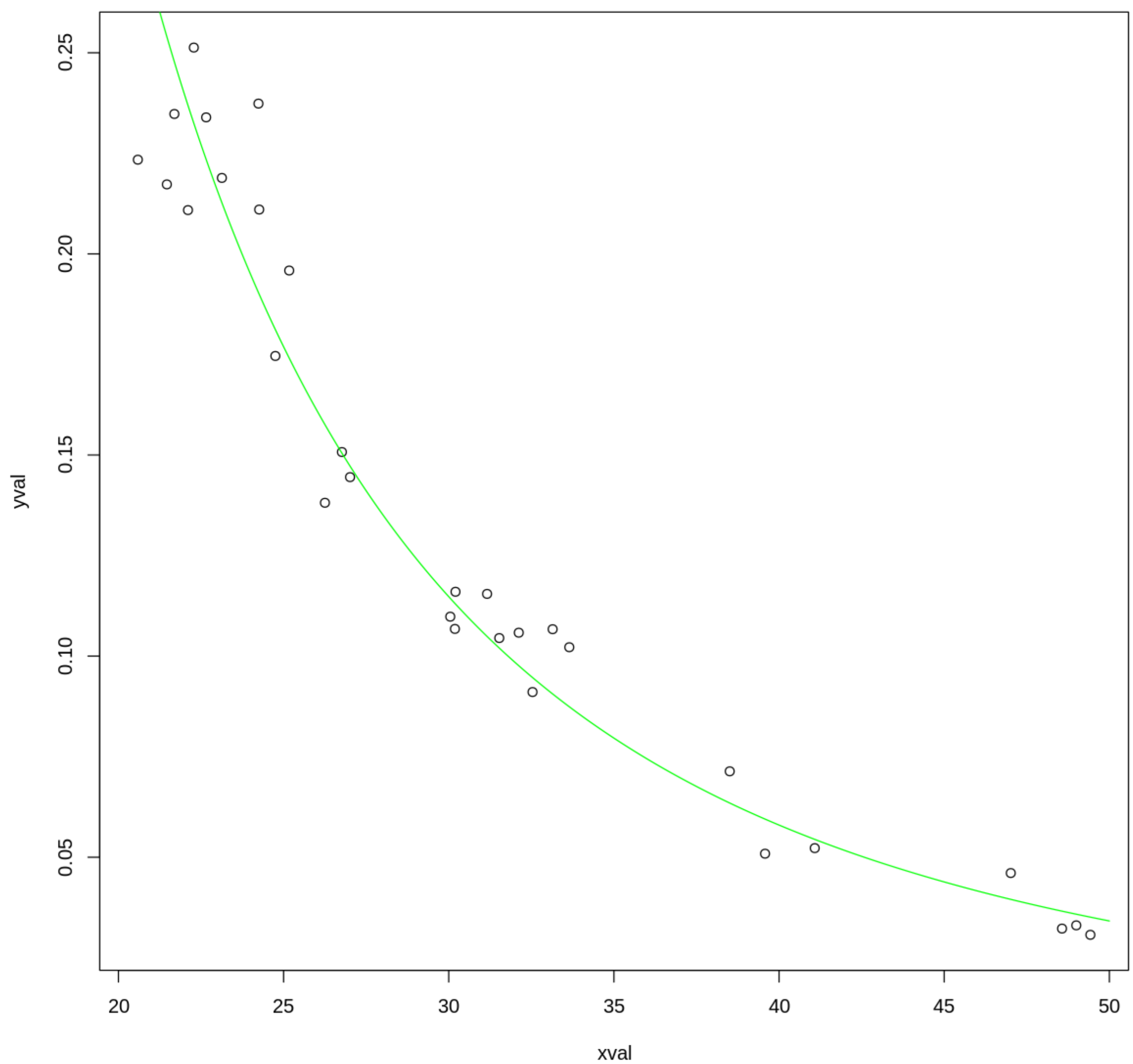
```
log(yval) = 5.90701-2.37322*log(xval)  => yval = exp(5.90701)*xval^(-2.37322)
```

In [39]:

```
xCurve <- seq(20, 50, 0.001)
yCurve <- exp(5.90701)*xCurve^(-2.37322)
```

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [40]:
```
plot(yval ~ xval, data = data2)
lines(xCurve, yCurve, col = 'green', lty = 1) ## Plot the curve
```



Question 4

For this question use the data setdata3.

(i)Readdata3and plotvaryas a function ofvarx. Fit a simple linear regression and add the regressionline to the plot. Comment. Obtain a summary for the regression and draw the diagnostic plots.Comment on the results.

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [41]:
```
data3 = read.table('data3')
head(data3)
```

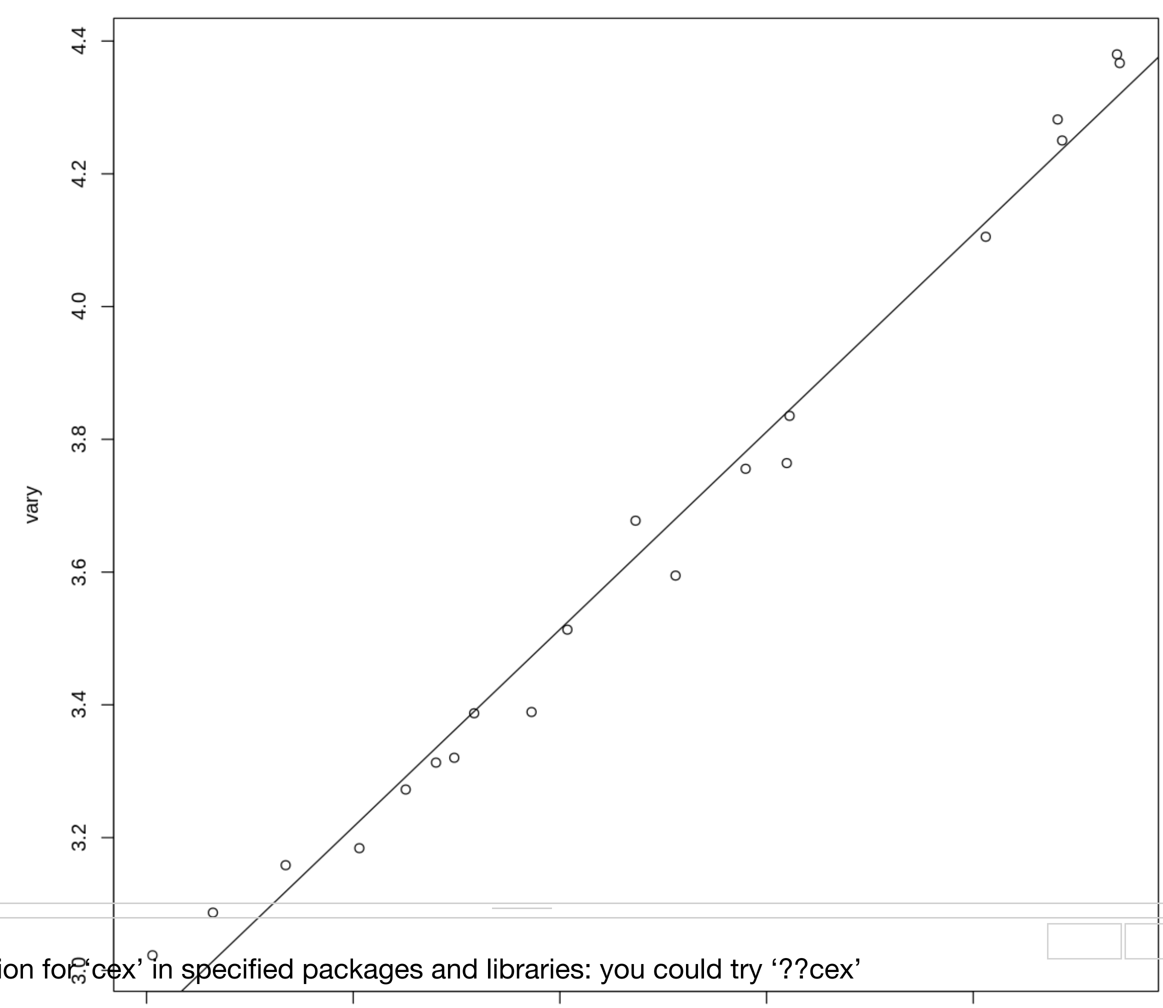
A data.frame: 6 × 2

	varx	vary
	<dbl>	<dbl>
1	8.121116	4.105132
2	2.800989	3.313017
3	2.977722	3.320299
4	5.797381	3.755601
5	4.732074	3.677448
6	9.417352	4.366827

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [42]:
```r
plot(vary ~ varx, data = data3)
model1 <- lm(vary ~ varx, data = data3)
abline(model1)
summary(model1)
```

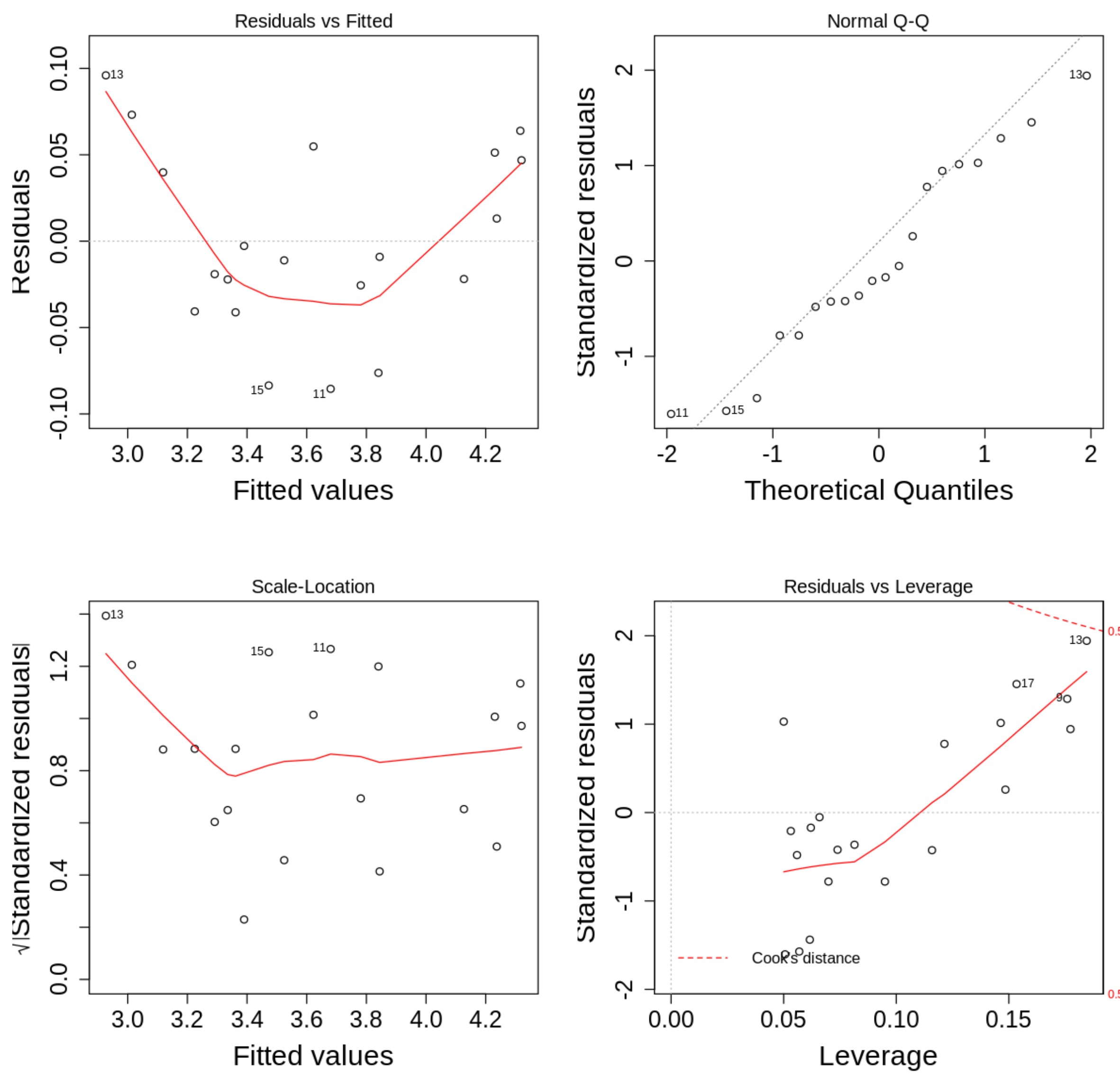
```
Call:
lm(formula = vary ~ varx, data = data3)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.08550 -0.02935 -0.01009  0.04794  0.09598 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.918281   0.023713  123.06   <2e-16 ***
varx        0.148842   0.004231   35.18   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05471 on 18 degrees of freedom
Multiple R-squared:  0.9857,	Adjusted R-squared:  0.9849 
F-statistic:  1238 on 1 and 18 DF,  p-value: < 2.2e-16
```



No documentation for 'cex' in specified packages and libraries: you could try '??cex'

```
The p value is significant and R^2 is 0.0849, it seems very good.
```

In [43]:
```
par(mfrow = c(2,2))
plot(model1,cex.axis=1.5,cex.lab=1.8,ps=10)
par(mfrow = c(1,1))
```




```
From residuals plot, we know that there is a quadratic pattern but not for the standardized residuals plot
and normality seems not satisfied.
We need further check.
Point 13 is very close to cook's distance 0.5 and it seems that larger residuals have largerr leverage
```

In [44]:
```
ncvTest(model1)
#### The variances are uniform
```

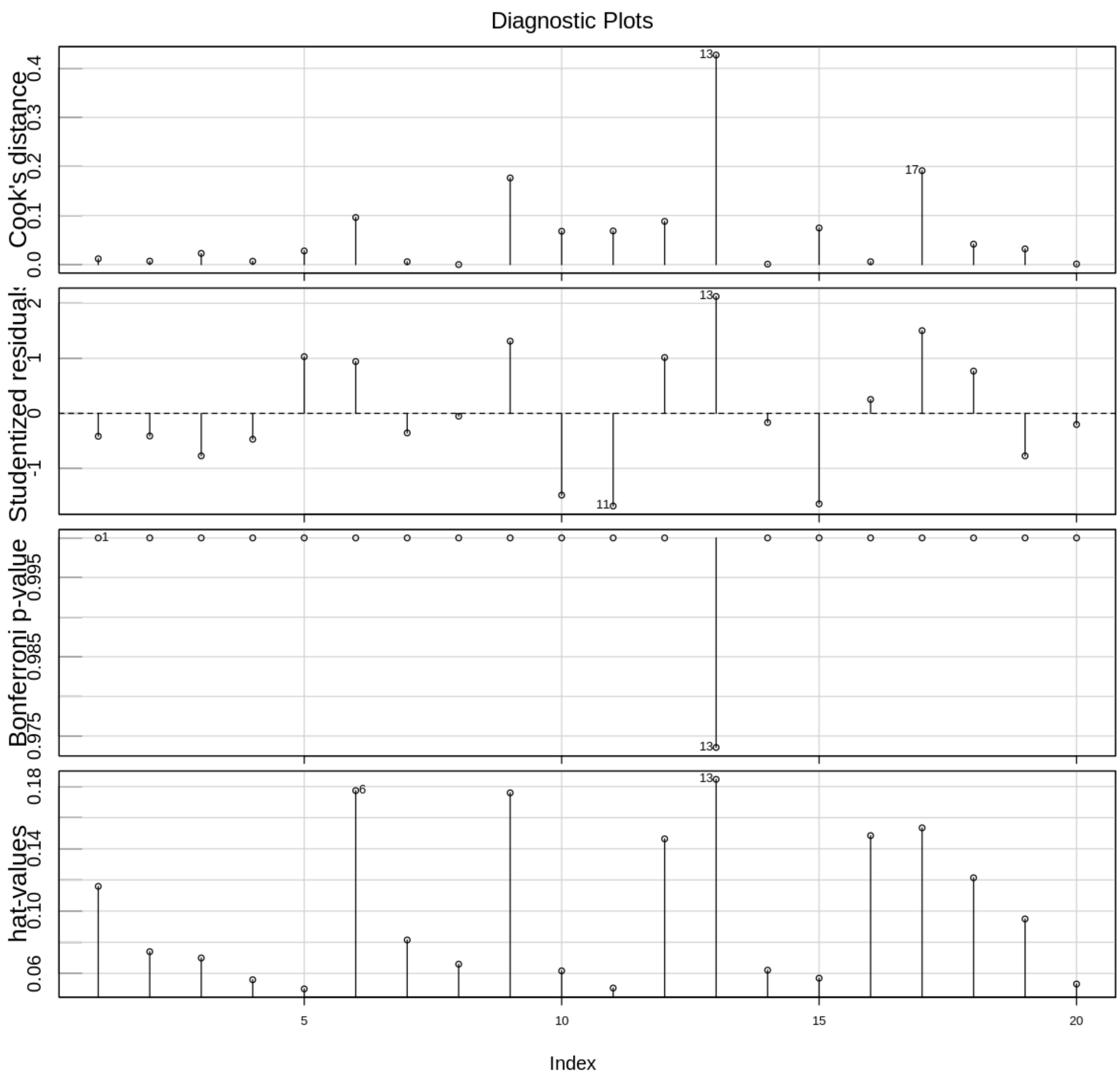
No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

```
Non-constant Variance Score Test
```

In [45]:
```
shapiro.test(model1$residuals)
######normality assumption is satisfied.
```

```
	Shapiro-Wilk normality test

data:  model1$residuals
W = 0.95481, p-value = 0.446
```

In [46]:
```
influenceIndexPlot(model1,cex.lab=2,cex.axis=1.5)
```



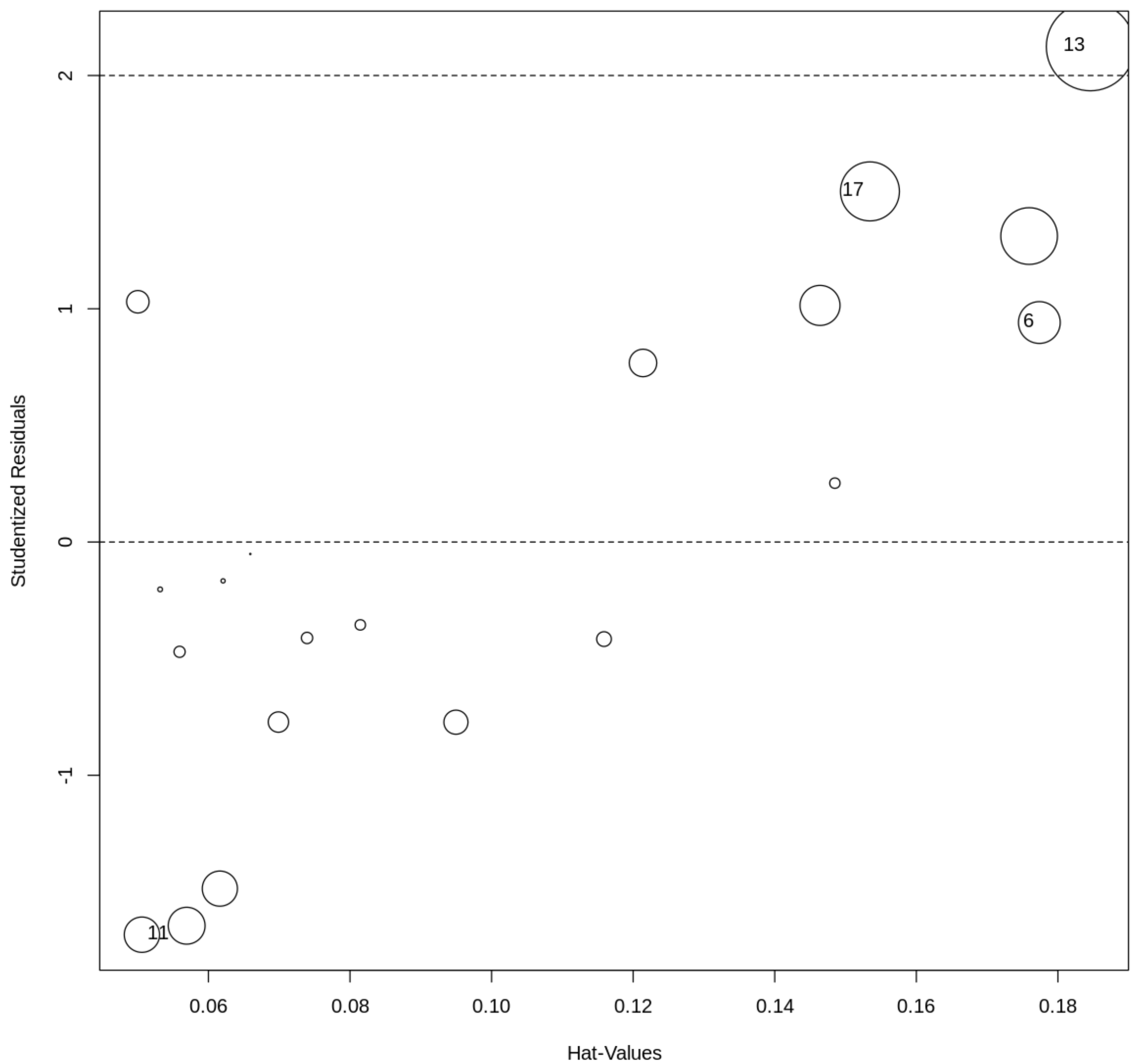
In [47]:
```
influencePlot(model1)
```

A data.frame: 4 × 3

StudRes	Hat	CookD
<dbl>	<dbl>	<dbl>

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

	StudRes	Hat	CookD
	<dbl>	<dbl>	<dbl>
6	0.941004	0.1773770	0.09607734
11	-1.683430	0.0506035	0.06854195



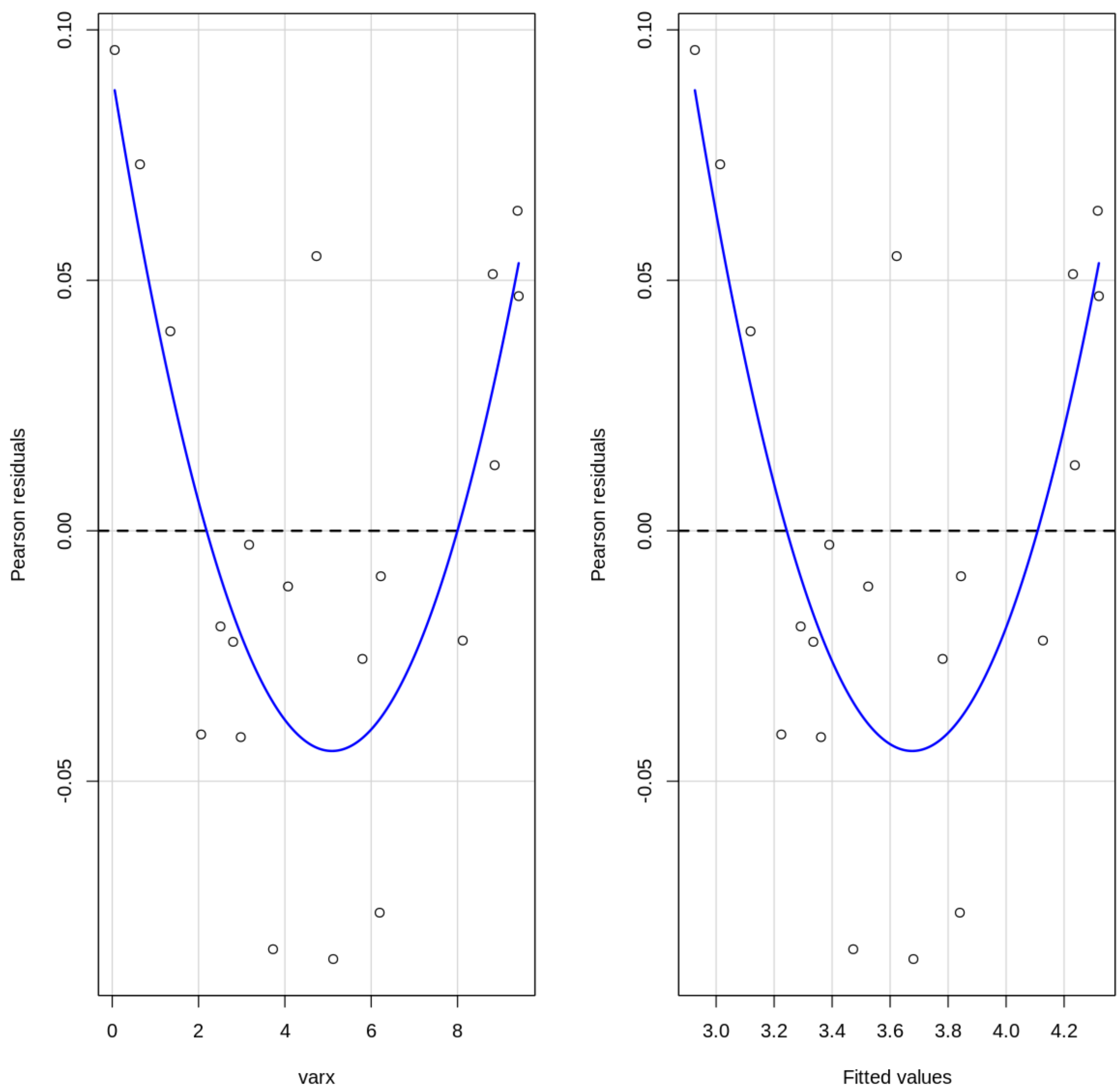

```
1  The influence plots we see that 13 have large standard residals
```

(ii)Use the functionresidualPlotsin packagecarand interpret the test produced by the function. Whatis your conclusion?

In [48]:
```
1  residualPlots(model1)
```

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

```
             Test stat Pr(>|Test stat|)
varx            4.9408        0.0001241 ***
Tukey test      4.9408        7.781e-07 ***
```




```
Very highly to be quandratic term, it means that the model is not
sufficient
```

(iii)Add a quadratic term to the regression model and obtain a summary, draw the diagnostic plots andcomment. Draw a scatterplot of the data and add the lines/curves for both models. Write down youfinal model.

In [49]:
```r
model2 <- lm(vary ~ poly(varx,2,raw=TRUE),
                   data=data3)
summary(model2)
```

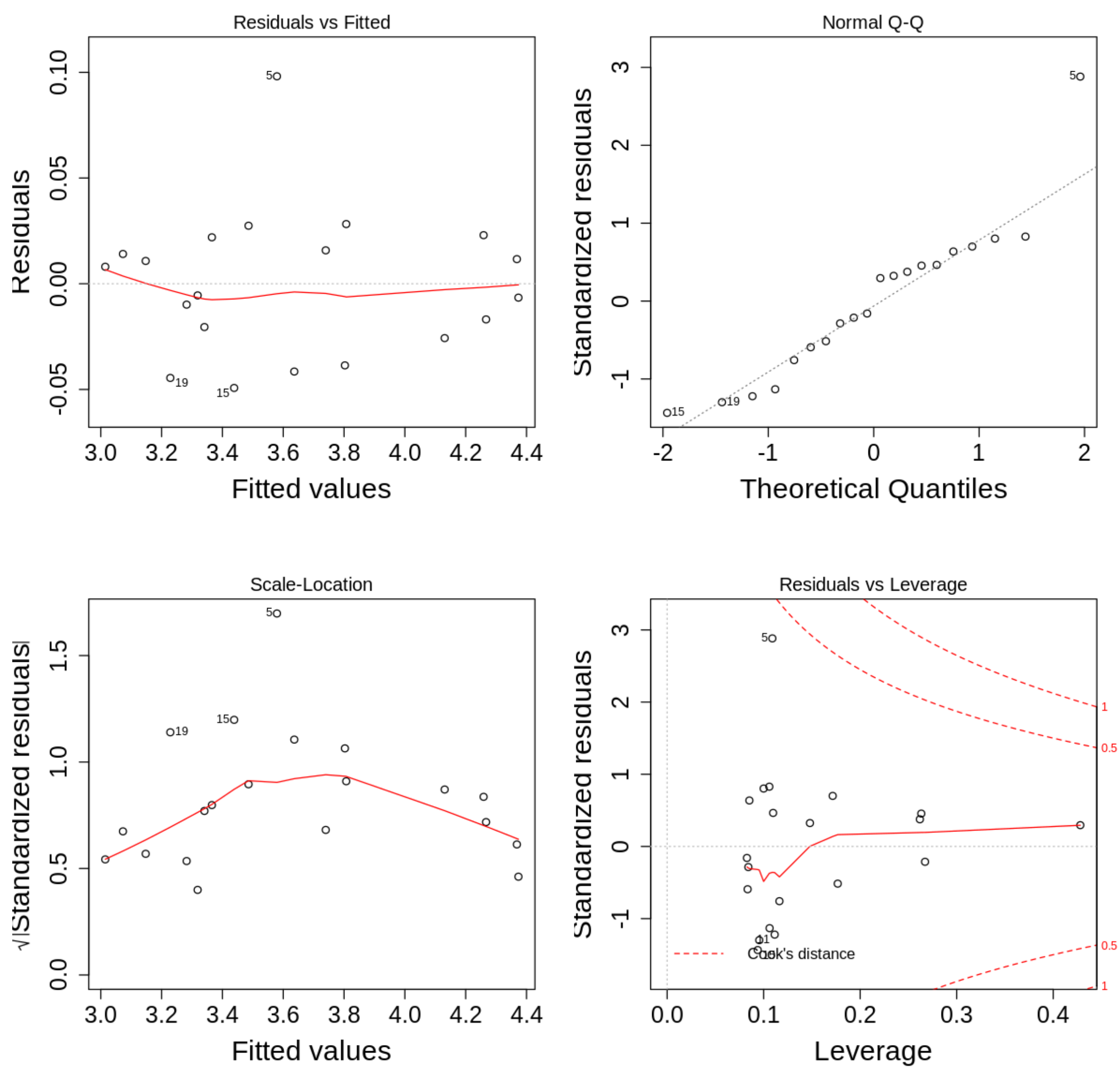
No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

```
Call:
lm(formula = vary ~ poly(varx, 2, raw = TRUE), data = data3)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.04929 -0.02179  0.00126  0.01733  0.09812 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)                3.009235   0.024152 124.597  < 2e-16 ***
poly(varx, 2, raw = TRUE)1 0.095847   0.011083   8.648 1.24e-07 ***
poly(varx, 2, raw = TRUE)2 0.005204   0.001053   4.941 0.000124 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03607 on 17 degrees of freedom
Multiple R-squared:  0.9941,	Adjusted R-squared:  0.9934 
```

The R^2 square is better thant the previous model reach to 0.994, the intercept do not change very much. The slope of varx reduce to less than 0.1 becuase of adding varx^2

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [50]:
```
par(mfrow = c(2,2))
plot(model2,cex.axis=1.5,cex.lab=1.8,ps=10)
par(mfrow = c(1,1))
```



In [51]:
```
ncvTest(model2)
shapiro.test(model2$residuals)
```

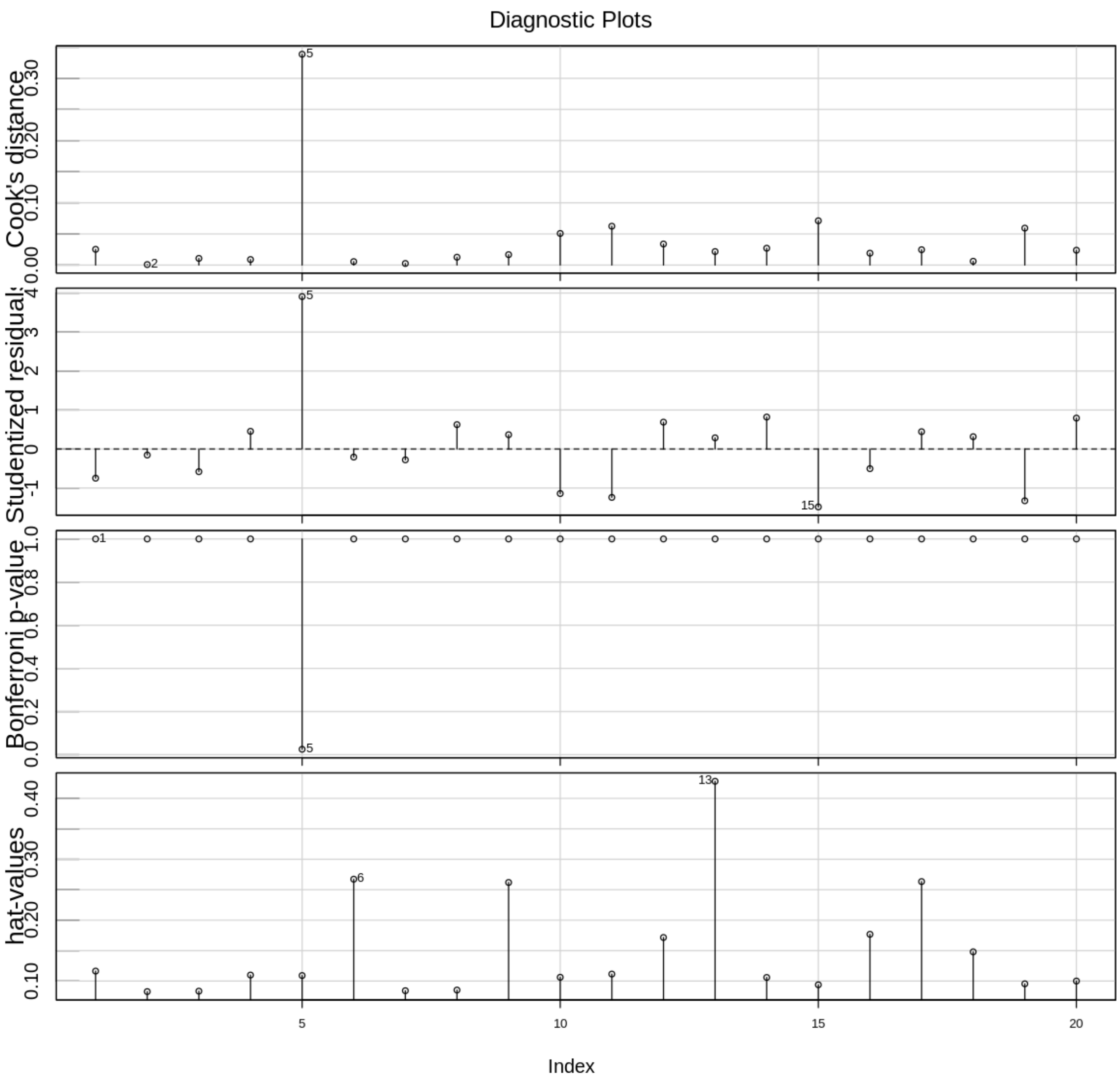
```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.1112209, Df = 1, p = 0.73876


        Shapiro-Wilk normality test

data:  model2$residuals
W = 0.91348, p-value = 0.07427
```

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

```
1 Both homoscedasticity and normality are satisfied. But ther is one outliers 5
```

In [52]:

```
influenceIndexPlot(model2,cex.lab=2,cex.axis=1.5)
```

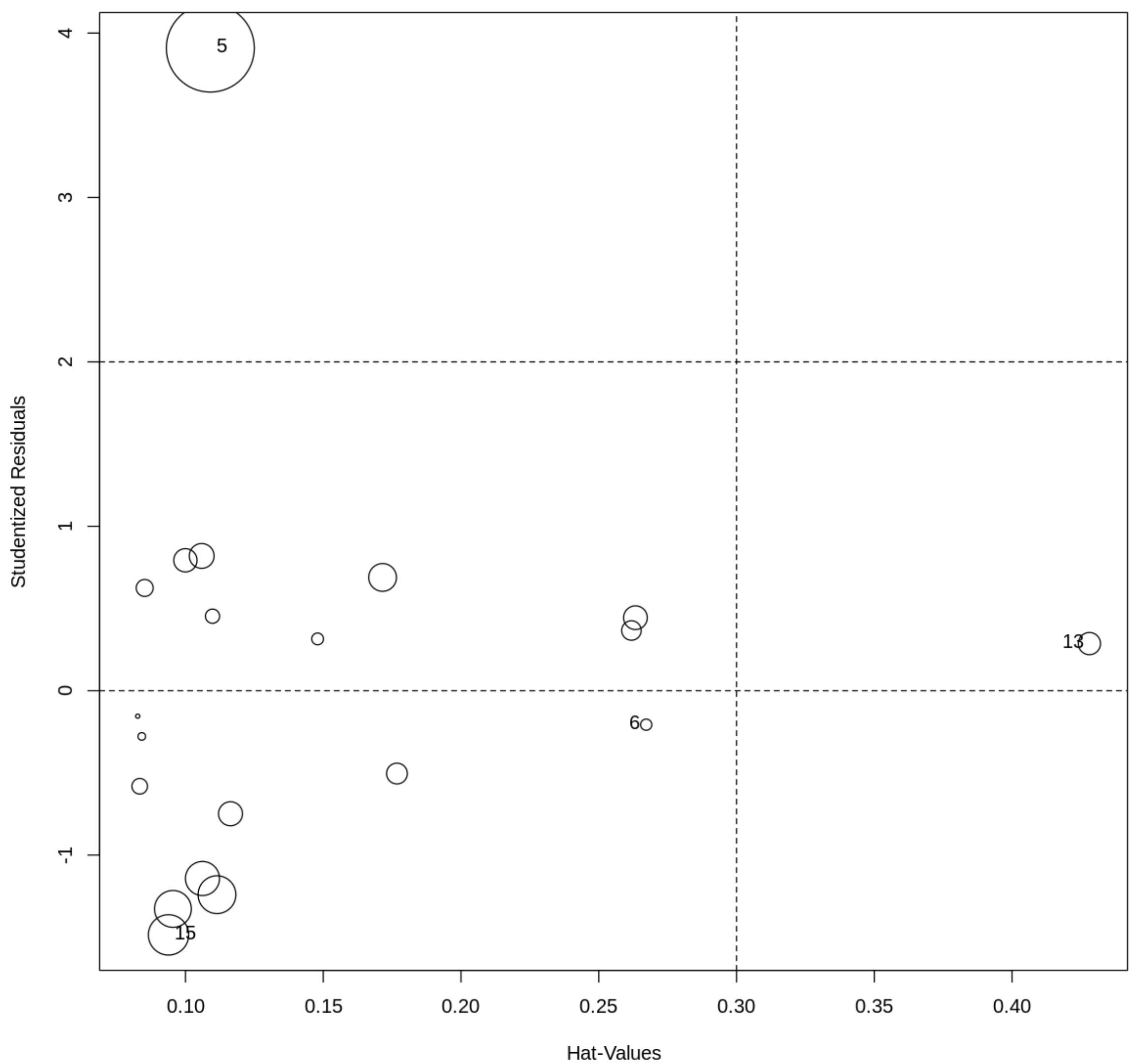


No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

In [53]:
```
1 influencePlot(model2)
```

A data.frame: 4 × 3

	StudRes	Hat	CookD
	<dbl>	<dbl>	<dbl>
5	3.9090487	0.10901910	0.33871007
6	-0.2068640	0.26722267	0.00551212
13	0.2864369	0.42808499	0.02163928
15	-1.4854910	0.09384668	0.07113021



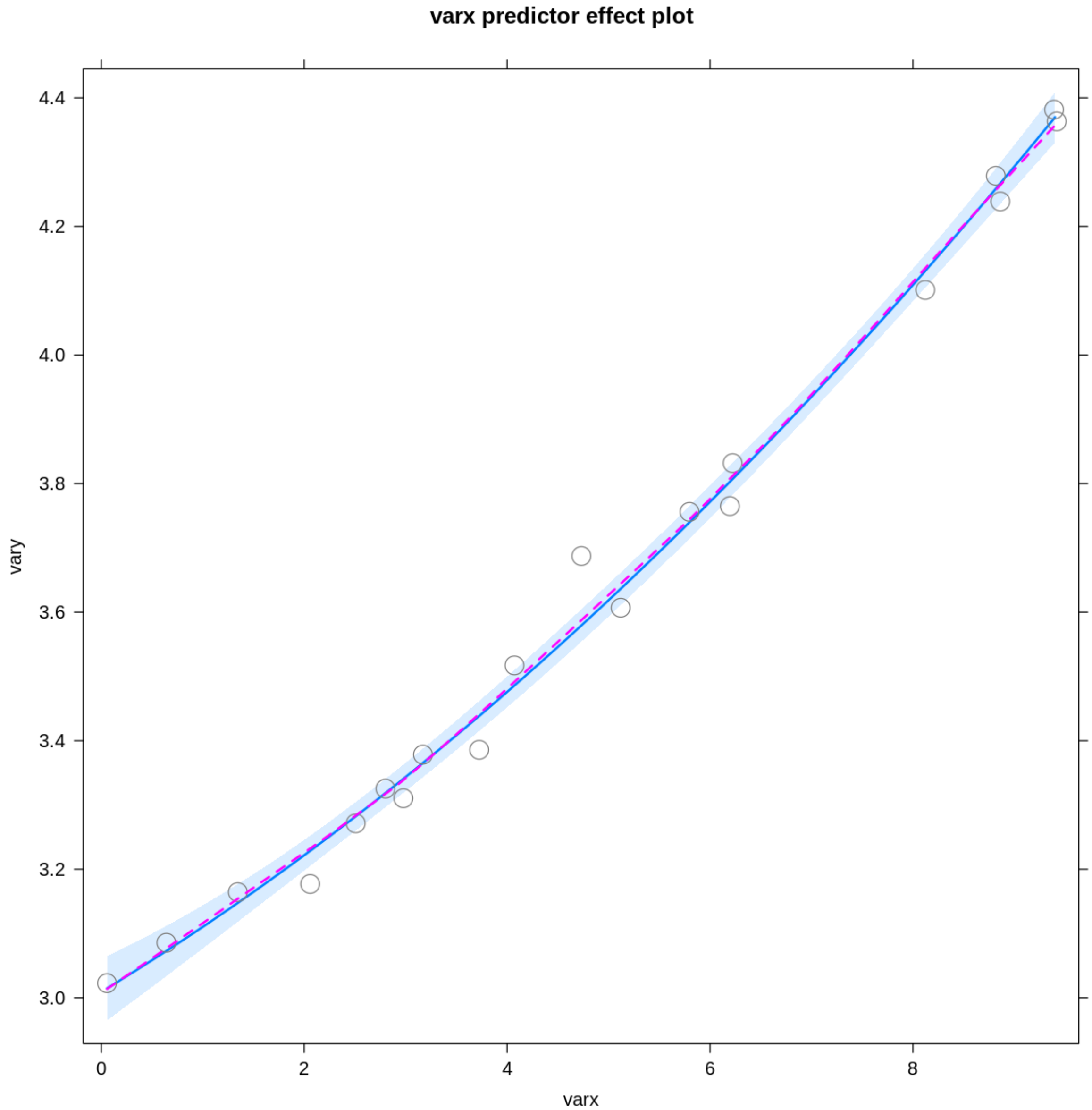

```
1 The influence plot show outlier 5
```

In [54]:
```
1 library(alr4)
2 plot(predict(model2, ... = TRUE),
```

No documentation for 'cex' in specified packages and libraries: you could try '??cex'

```
3         partial.residuals = list(cex=2, col=gray(0.5),
4                                  lty = 2))
```

```
Loading required package: effects

Registered S3 methods overwritten by 'lme4':
  method                          from
  cooks.distance.influence.merMod car
  influence.merMod                car
  dfbeta.influence.merMod         car
  dfbetas.influence.merMod        car

lattice theme set by effectsTheme()
See ?effectsTheme for details.
```



No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’

No documentation for ‘cex’ in specified packages and libraries: you could try ‘??cex’