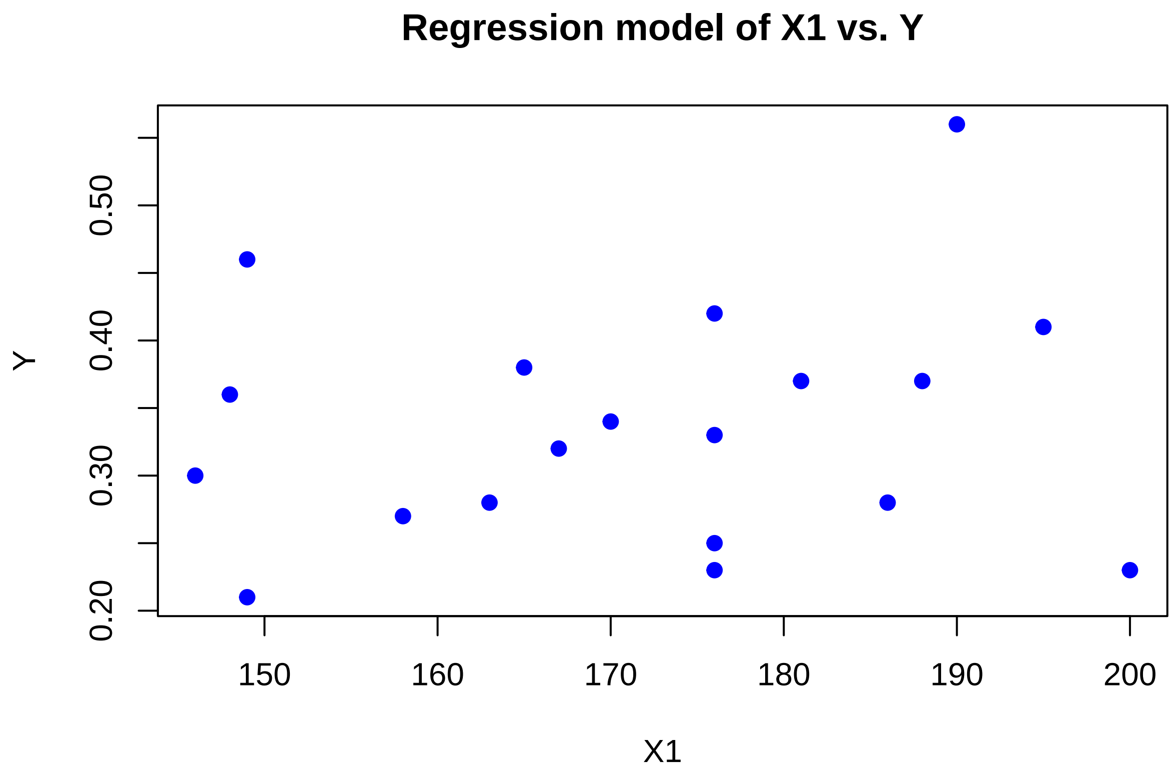
```r
knitr::opts_chunk$set(echo = TRUE)

drug = read.table("drug.txt",header=T)
attach(drug)
```

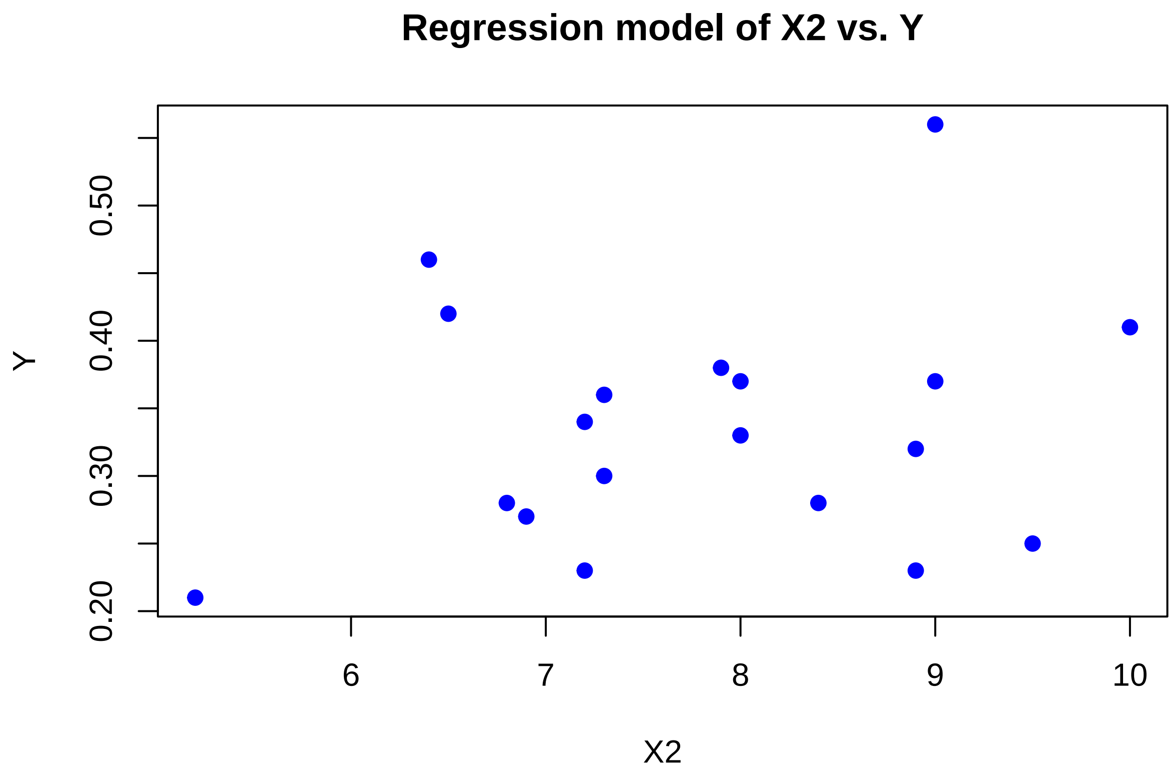
```
## The following object is masked from cp (pos = 3):
##
##     Y

## The following object is masked from demand (pos = 4):
##
##     Y

## The following object is masked from cp (pos = 5):
##
##     Y

## The following object is masked from demand (pos = 6):
##
##     Y

## The following objects are masked from drug (pos = 7):
##
##     x1, x2, x3, Y

## The following objects are masked from drug (pos = 10):
##
##     x1, x2, x3, Y

## The following object is masked from demand (pos = 11):
##
##     Y

## The following object is masked from demand (pos = 16):
##
##     Y

## The following object is masked from cp (pos = 18):
##
##     Y

## The following object is masked from cp (pos = 19):
##
##     Y
```

```r
n<-dim(drug)[1]

p<-dim(drug)[2]-1
```

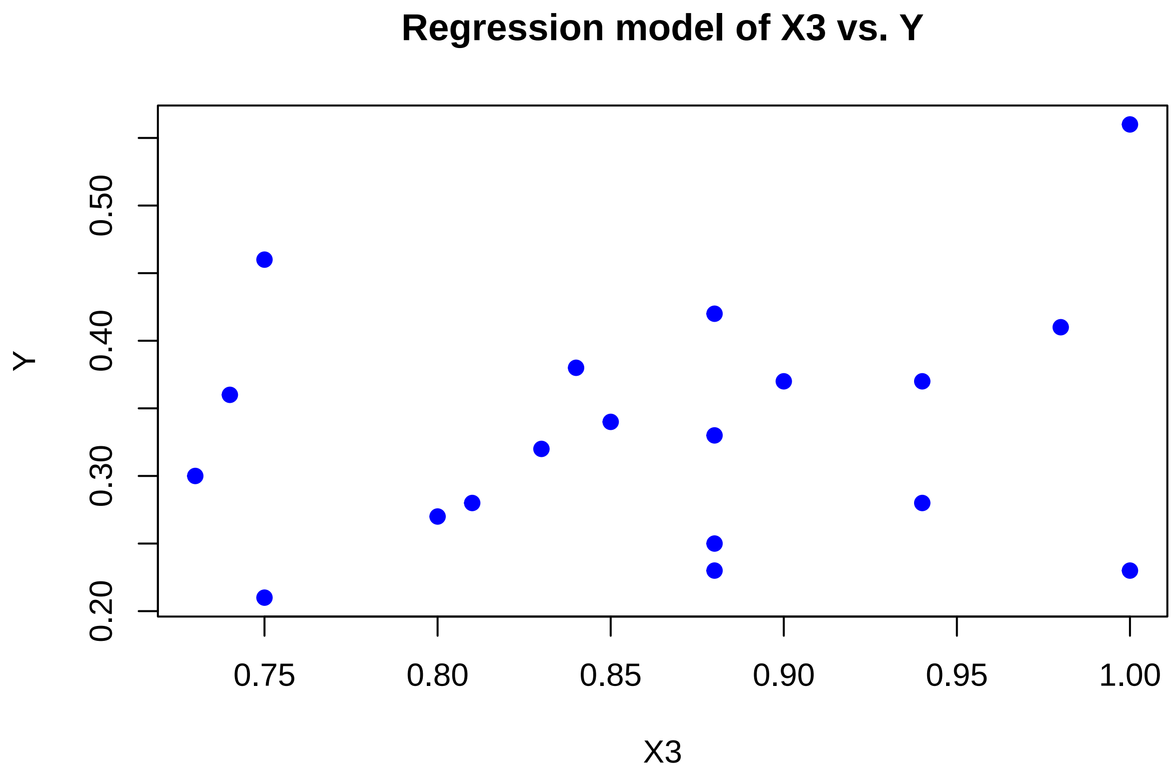
```r
# Y is the proportion of the drug dose in the liver
# x1 is the body weight of each rat in grams
# x2 is the weight of each liver in grams
# x3 is the relative dose of the drug given to each rat as a fraction of the largest dose


# Visualization of correlation of X1 and Y
plot(x1,Y,pch=19,col="blue",xlab="X1",ylab="Y",main="Regression model of X1 vs. Y")
```




```r
# Visualization of correlation of X2 and Y
plot(x2,Y,pch=19,col="blue",xlab="X2",ylab="Y",main="Regression model of X2 vs. Y")
```




```
# Visualization of correlation of X3 and Y
plot(x3,Y,pch=19,col="blue",xlab="X3",ylab="Y",main="Regression model of X3 vs. Y")
```



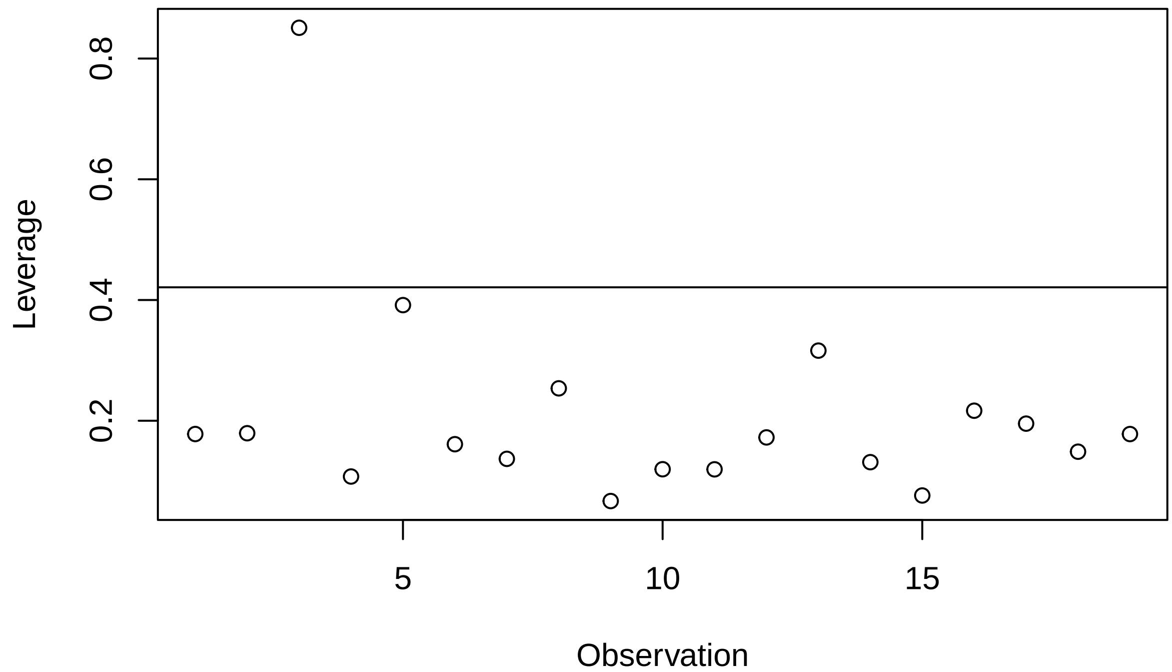

```r
fit<-lm(Y ~ x1 + x2 + x3)
summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ x1 + x2 + x3)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.100557 -0.063233  0.007131  0.045971  0.134691
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.265922   0.194585   1.367   0.1919
## x1          -0.021246   0.007974  -2.664   0.0177 *
## x2           0.014298   0.017217   0.830   0.4193
## x3           4.178111   1.522625   2.744   0.0151 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07729 on 15 degrees of freedom
## Multiple R-squared:  0.3639, Adjusted R-squared:  0.2367
## F-statistic:  2.86 on 3 and 15 DF,  p-value: 0.07197
```

```r
# Influence measurement
influence.measures(fit)
```

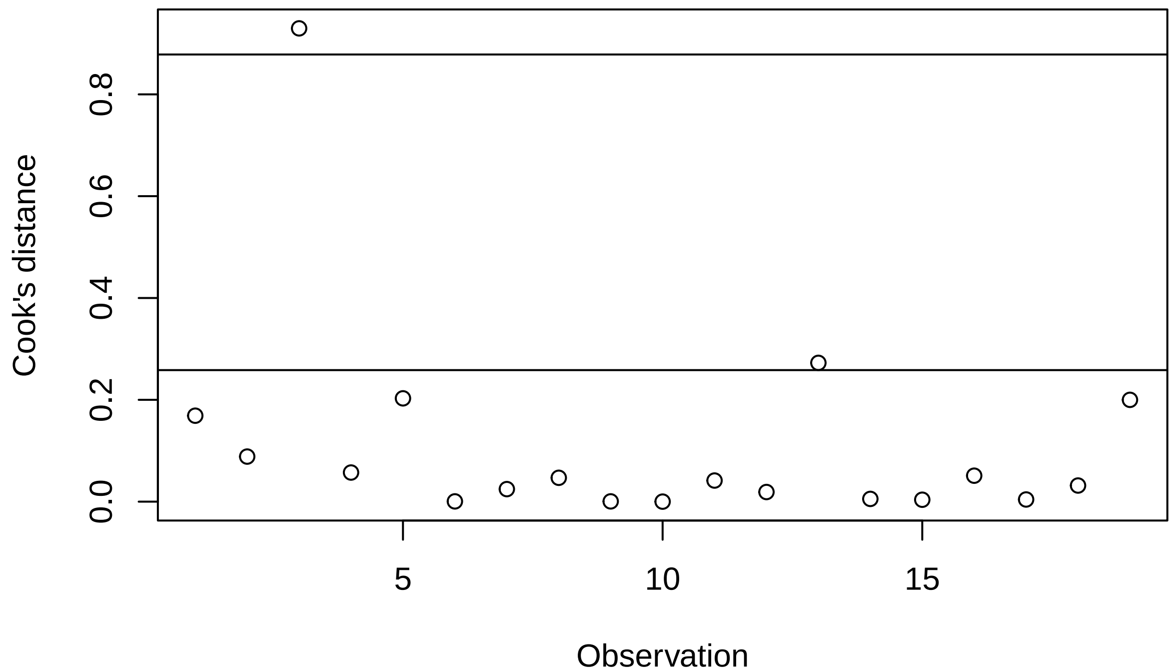
```
## Influence measures of
##   lm(formula = Y ~ x1 + x2 + x3) :
##
##       dfb.1_   dfb.x1    dfb.x2   dfb.x3   dffit cov.r   cook.d    hat
## 1  -0.03835  0.31492 -0.704363 -0.24375  0.8920 0.631 1.69e-01 0.1780
## 2   0.14256 -0.09774 -0.481778  0.12561 -0.6088 1.016 8.85e-02 0.1793
## 3  -0.23100 -1.66770  0.304572  1.74720  1.9048 7.401 9.30e-01 0.8509
## 4   0.12503 -0.12686 -0.303651  0.14009 -0.4944 0.860 5.72e-02 0.1076
## 5   0.52161 -0.39627  0.550016  0.27474 -0.9095 1.524 2.03e-01 0.3915
## 6   0.00229  0.01360  0.029003 -0.01715  0.0427 1.567 4.87e-04 0.1612
## 7  -0.18376  0.15044 -0.083355 -0.11839  0.3096 1.289 2.46e-02 0.1369
## 8  -0.29725  0.05936  0.246500 -0.04042  0.4262 1.520 4.69e-02 0.2537
## 9  -0.00968  0.01791  0.000168 -0.01673  0.0427 1.402 4.88e-04 0.0670
## 10 -0.00566  0.00993 -0.003365 -0.00929 -0.0140 1.496 5.23e-05 0.1197
## 11 -0.29053  0.19381  0.100742 -0.17288 -0.4104 1.066 4.14e-02 0.1195
## 12  0.21742 -0.02526  0.051721 -0.00920  0.2689 1.444 1.89e-02 0.1724
## 13 -0.77232  0.14391  0.766461 -0.12005 -1.0989 0.972 2.73e-01 0.3162
## 14 -0.03482 -0.04618 -0.076534  0.05990 -0.1423 1.461 5.37e-03 0.1314
## 15  0.01868  0.04063 -0.054603 -0.03817  0.1188 1.359 3.73e-03 0.0762
## 16  0.12309 -0.00557  0.328720 -0.04910 -0.4475 1.375 5.10e-02 0.2166
## 17 -0.10359  0.01454 -0.027611  0.00215 -0.1262 1.607 4.25e-03 0.1952
## 18 -0.15423  0.19035  0.162330 -0.18929  0.3522 1.270 3.16e-02 0.1487
## 19  0.85580 -0.25010 -0.294617  0.17123  0.9952 0.517 2.00e-01 0.1780
##    inf
## 1
## 2
## 3    *
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
```

```r
library(car)

# Leverage for each observation
plot(1:n,hatvalues(fit),xlab="Observation",ylab="Leverage")
abline(h=(2*(p + 1)/n))
identify(hatvalues(fit))
```



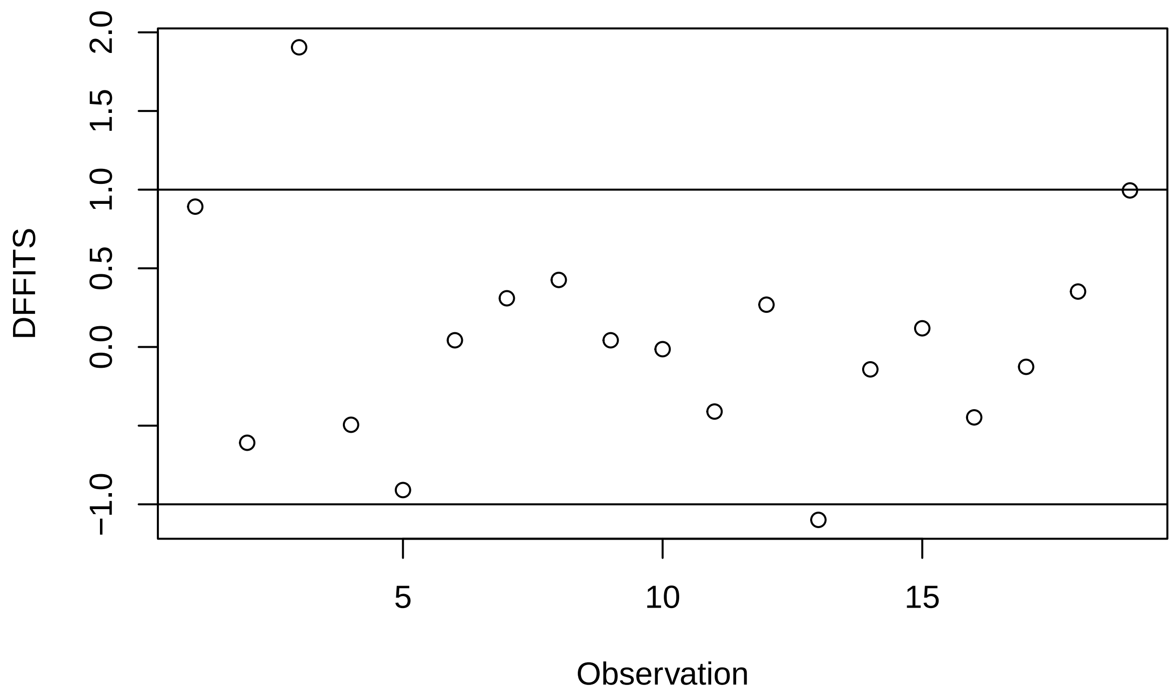

```
## integer(0)
```

```
# Leverage > 2(p + 1)/n is "large"

# Observation #3 is significant.


# Cook's distance for each observation
plot(1:n,cooks.distance(fit),xlab="Observation",ylab="Cook's distance")
abline(h=qf(0.5,p+1,n-(p+1)))
abline(h=qf(0.1,p+1,n-(p+1)))
identify(cooks.distance(fit))
```



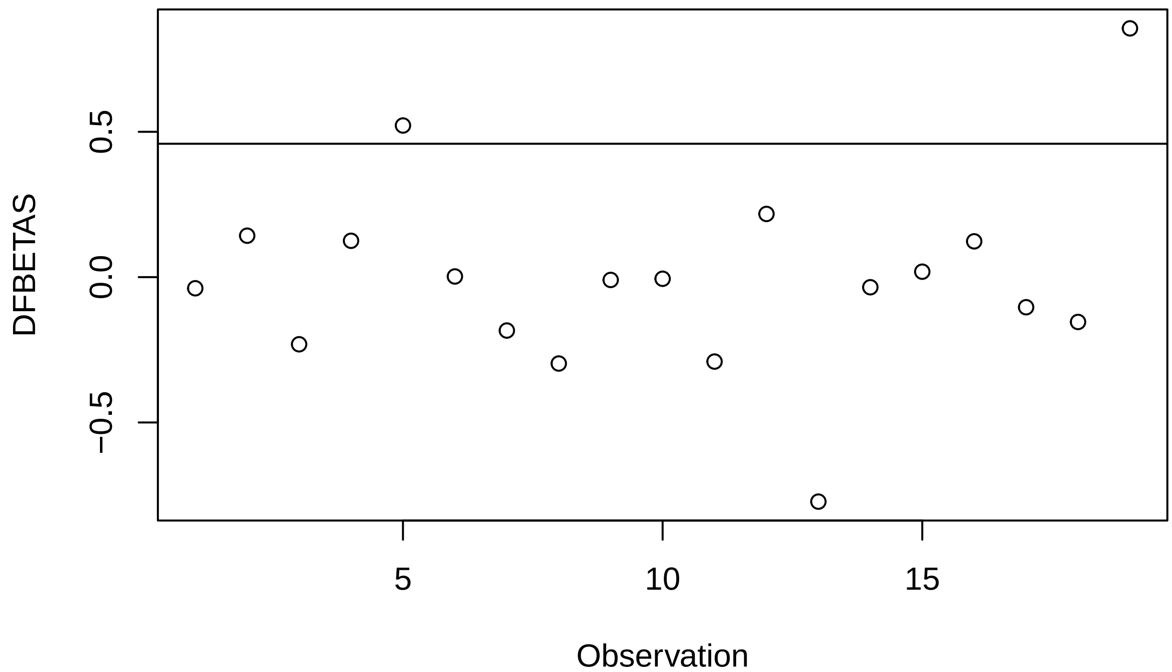

```
## integer(0)
```

```
# Cook's distance > qf(0.5,p+1,n-(p+1)) means large impact
# Cook's distance > qf(0.1,p+1,n-(p+1)) means small impact

# Observation #3 has large impact.
# Observation #13 has small impact.


# DFFITS
plot(1:n,dffits(fit),xlab="Observation",ylab="DFFITS")
abline(h=1)
abline(h=-1)
identify(dffits(fit))
```




```
## integer(0)
```

```
# absolute_value(DFFITS) > 1 are observations to look out for in small to medium datasets

# Observations 3, 13, 19 are observations to look out for


# DFBETAS
plot(1:n,dfbetas(fit)[,1],xlab="Observation",ylab="DFBETAS")
abline(h=2)
abline(h=(2/sqrt(n)))
identify(dfbetas(fit)[,1])
```




```
## integer(0)
```

```
# If absolute_value(DFBETAS) > 2, then major impact
# If absolute_value(DFBETAS) > 2/sqrt(n), then there is impact; the value of
↪  absolute_value(DFBETAS) is the "amount" of impact
```

```
# Perform the outlier test
outlierTest(fit)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##    rstudent unadjusted p-value Bonferroni p
## 19 2.138833           0.050557      0.96058
```

```
# Evaluate Collinearity
vif(fit) # variance inflation factors
```

```
##        x1        x2        x3
## 52.101917  1.335679 51.427154
```

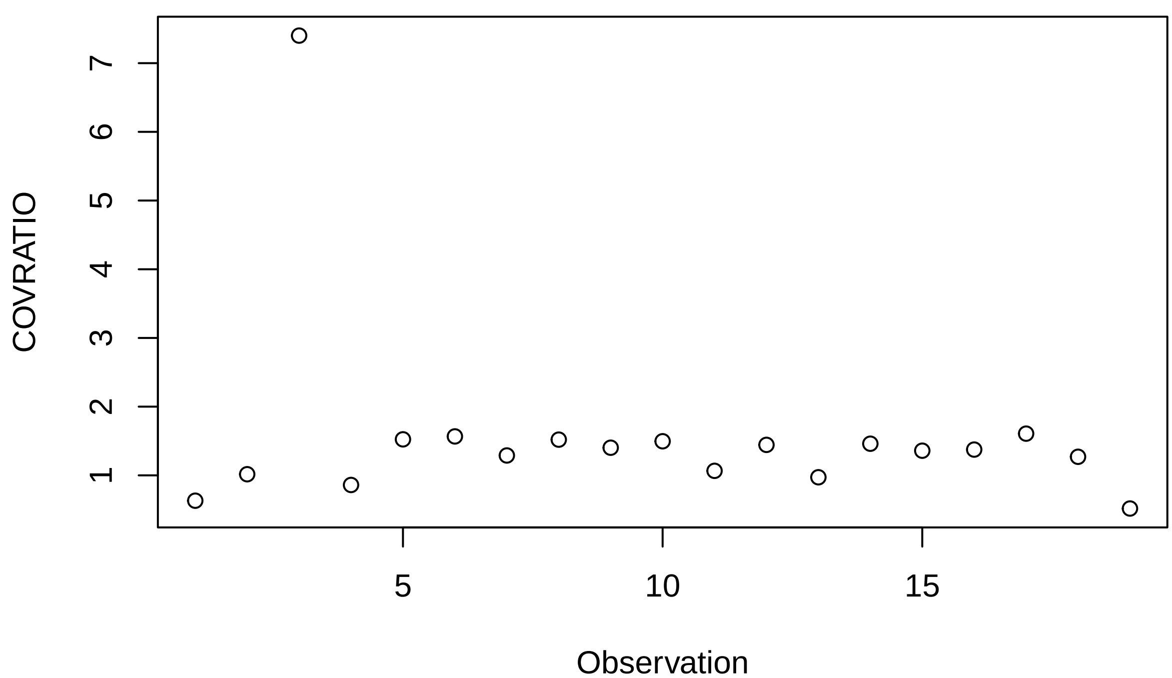
```
sqrt(vif(fit)) > 2
```

```
##    x1    x2    x3
##  TRUE FALSE  TRUE
```

```
# Find the covratios
covratio(fit)
```

```
##         1         2         3         4         5         6         7
## 0.6310012 1.0164073 7.4008047 0.8599033 1.5241607 1.5667434 1.2892824
##         8         9        10        11        12        13        14
## 1.5200517 1.4022535 1.4963647 1.0656374 1.4437298 0.9722483 1.4605453
##        15        16        17        18        19
## 1.3588186 1.3749188 1.6071057 1.2700823 0.5173619
```

```
plot(1:n,covratio(fit),xlab="Observation",ylab="COVRATIO")
abline(h=(1+3*(p+1)))
abline(h=(1-3*(p+1)))
identify(covratio(fit))
```




```
## integer(0)
```

```
# No observations are considered influential under the COVRATIO test
```

```r
# If we want to remove the most influential point, we will remove observation 3.

# Removing influential point observation 3,

# Create a matrix with row 3 removed
drug_outlier_removed <- drug[-c(3),]


# You need to use unlist() to convert the lists to data frames before fitting the data
Y_new<-unlist(drug_outlier_removed[4])
x1_new<-unlist(drug_outlier_removed[1])
x2_new<-unlist(drug_outlier_removed[2])
x3_new<-unlist(drug_outlier_removed[3])

fit_outlier_removed<-lm(Y_new ~ x1_new + x2_new + x3_new)
summary(fit_outlier_removed)
```

```
##
## Call:
## lm(formula = Y_new ~ x1_new + x2_new + x3_new)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.102154 -0.056486  0.002838  0.046519  0.137059
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.311427   0.205094   1.518    0.151
## x1_new      -0.007783   0.018717  -0.416    0.684
## x2_new       0.008989   0.018659   0.482    0.637
## x3_new       1.484877   3.713064   0.400    0.695
##
## Residual standard error: 0.07825 on 14 degrees of freedom
## Multiple R-squared:  0.02106,    Adjusted R-squared:  -0.1887
## F-statistic: 0.1004 on 3 and 14 DF,  p-value: 0.9585
```