Topic :

LMS and LTS Estimation

LMS Estimator

Rousseeuw (1984) proposed Least Median of Squares (LMS) regression which minimizes the median of the squared residuals instead of minimizing the sum of the squared residuals by OLS

$$\min_{\beta} \underset{i}{median}(\varepsilon_i^2)$$

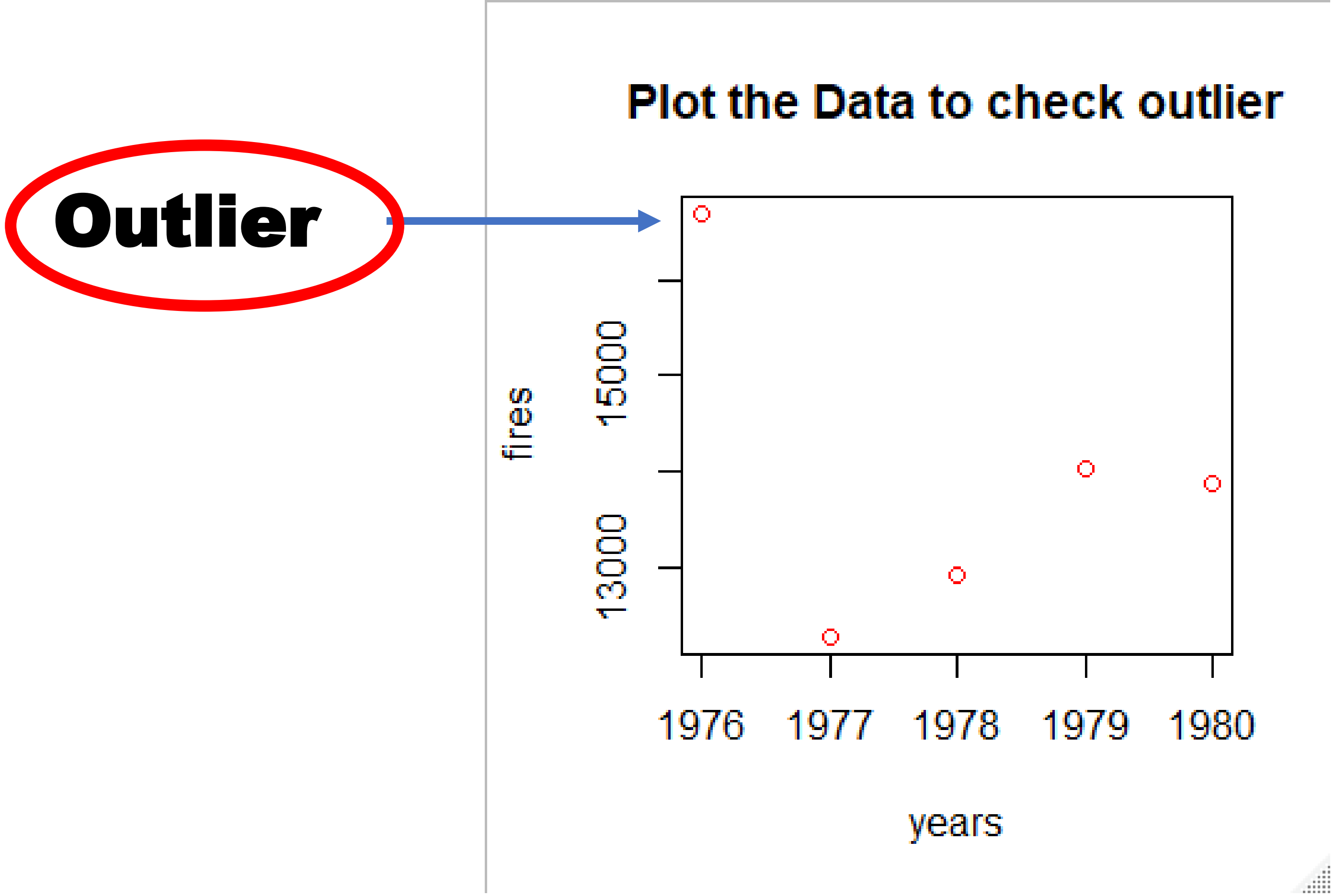
In computational aspects, the LMS estimate would be obtained by evaluating subsets of the data points, where the integer h is defined as

$$h = \left[\frac{n}{2}\right] + \left[\frac{(p+1)}{2}\right]$$

in which n and p denotes the sample size and number of parameters in the model, respectively. The LMS-estimate can be obtained from the subset which provides the smallest median squared residual.

Problem 01.

```
##======== Data=======##
## Years        Fires
##  1976        16694
##  1977        12271
##  1978        12904
##  1979        14036
##  1980        13874
```



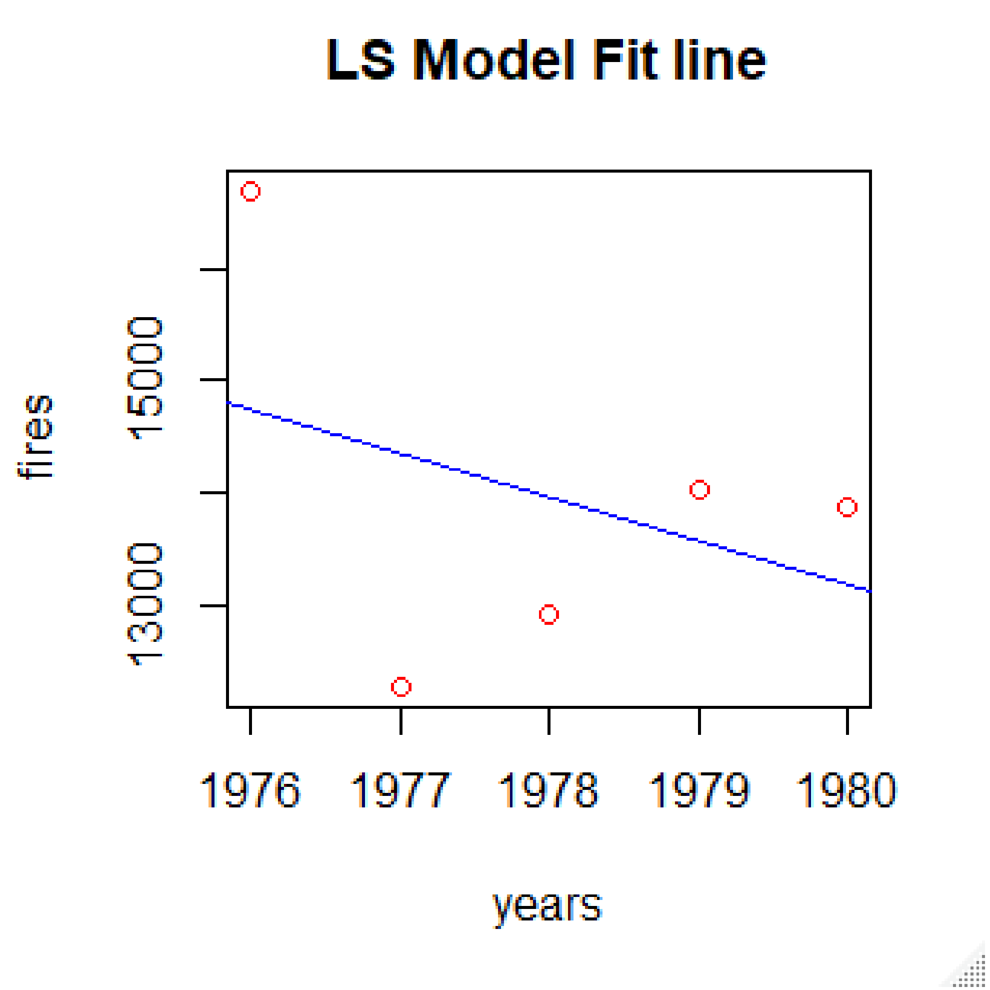
R programming Code :

```
years <- 1976:1980
fires <- c(16694, 12271, 12904, 14036, 13874)
plot(years, fires, main="Plot the Data to check outlier", col="red")
```

R programming Code :

Frist Fit LS model

```
plot(years, fires, main="LS Model Fit line", col = "red")
abline( lm ( fires ~ years ), col= 4)
```




```
Coefficients:
(Intercept)  years
 780430.8    -387.5
```

Divided the data sub group (h)

Formula

$$h = \left[\frac{n}{2}\right] + \left[\frac{(p+1)}{2}\right]$$

R programming Code :

n=5

p=1

h = ((n/2)+(p+1)/2) = 3 # of obs. Each group

Possible group

$^{5}C_{3} = 10$

R programming Code :

```r
## Create afunction to collect median squre of errors

med.er.sq <- function( x, y){
        new <- data.frame(y, x)
        model <- lm(y ~ x)
        errors <- y - predict(model, new)
        med.er.sqr <- median(errors^2)
        return (med.er.sqr)
}
```

Function

R programming Code :

Corresponding Median of errors

```
## medain Erros Squre calculation
e1 <- med.er.sq(years[c(1,2,3)], fires[c(1,2,3)]) ## 1.line
e2 <- med.er.sq(years[c(1,2,4)], fires[c(1,2,4)]) ## 2.line
e3 <- med.er.sq(years[c(1,2,5)], fires[c(1,2,5)]) ## 3.line
e4 <- med.er.sq(years[c(1,3,4)], fires[c(1,3,4)]) ## 4.line
e5 <- med.er.sq(years[c(1,3,5)], fires[c(1,3,5)]) ## 5.line
e6 <- med.er.sq(years[c(1,4,5)], fires[c(1,4,5)]) ## 6.line
e7 <- med.er.sq(years[c(2,3,4)], fires[c(2,3,4)]) ## 7.line
e8 <- med.er.sq(years[c(2,3,5)], fires[c(2,3,5)]) ## 8.line
e9 <- med.er.sq(years[c(2,4,5)], fires[c(2,4,5)]) ## 9.line
e10 <- med.er.sq(years[c(3,4,5)], fires[c(3,4,5)]) ##10.lines
```

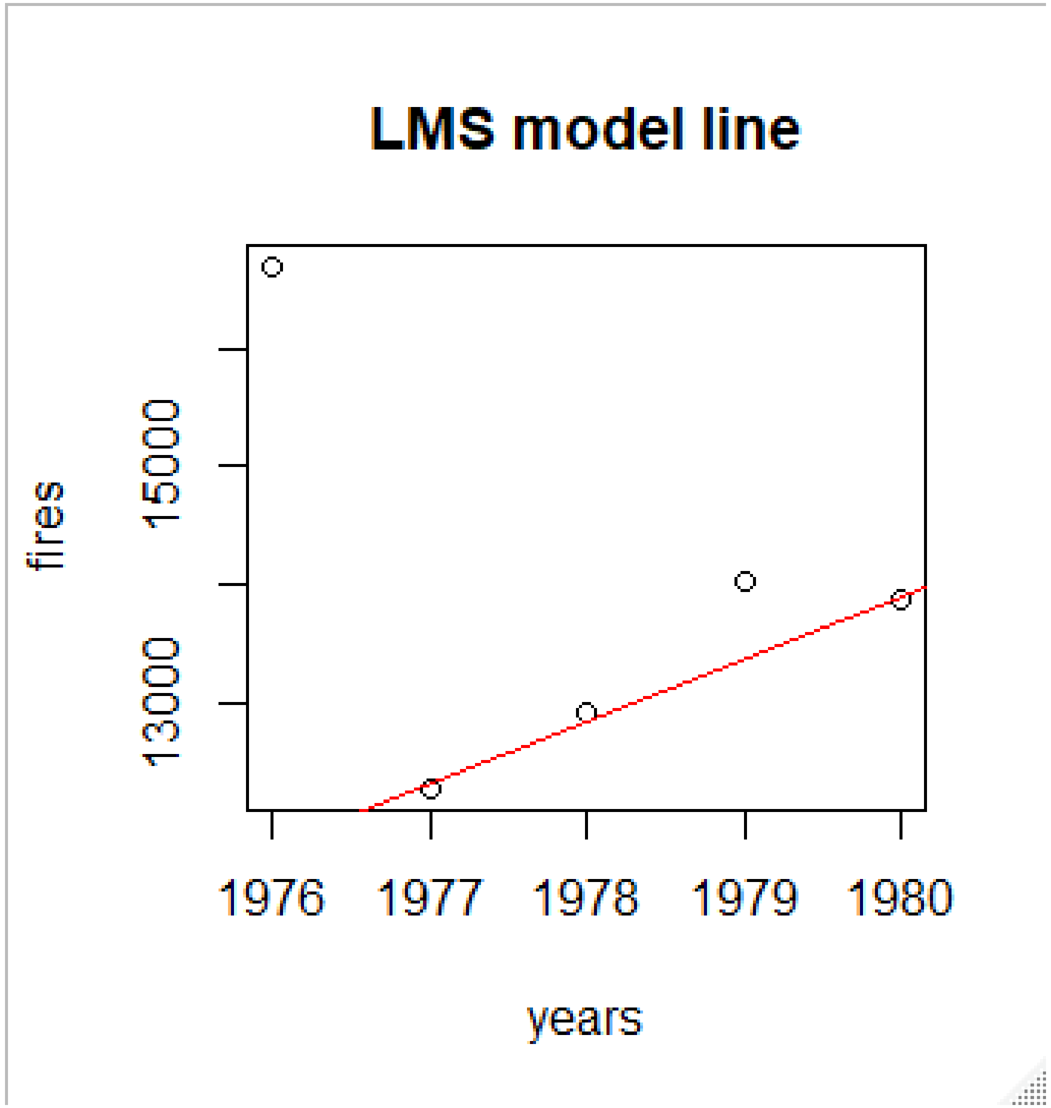
```
710087.1111
574455.7194
327184.0000
186994.4694
629377.7778
6978.6746
6916.6944
447.0204
22264.9031
46512.1111
```

```
## Find least median squre of errors
e <-c(e1, e2, e3, e4, e5, e6, e7, e8, e9, e10)
summary(e)
```

```
Min. 1st Qu. Median Mean 3rd Qu. Max.
447 10800 116753 251122 512638 710087
```

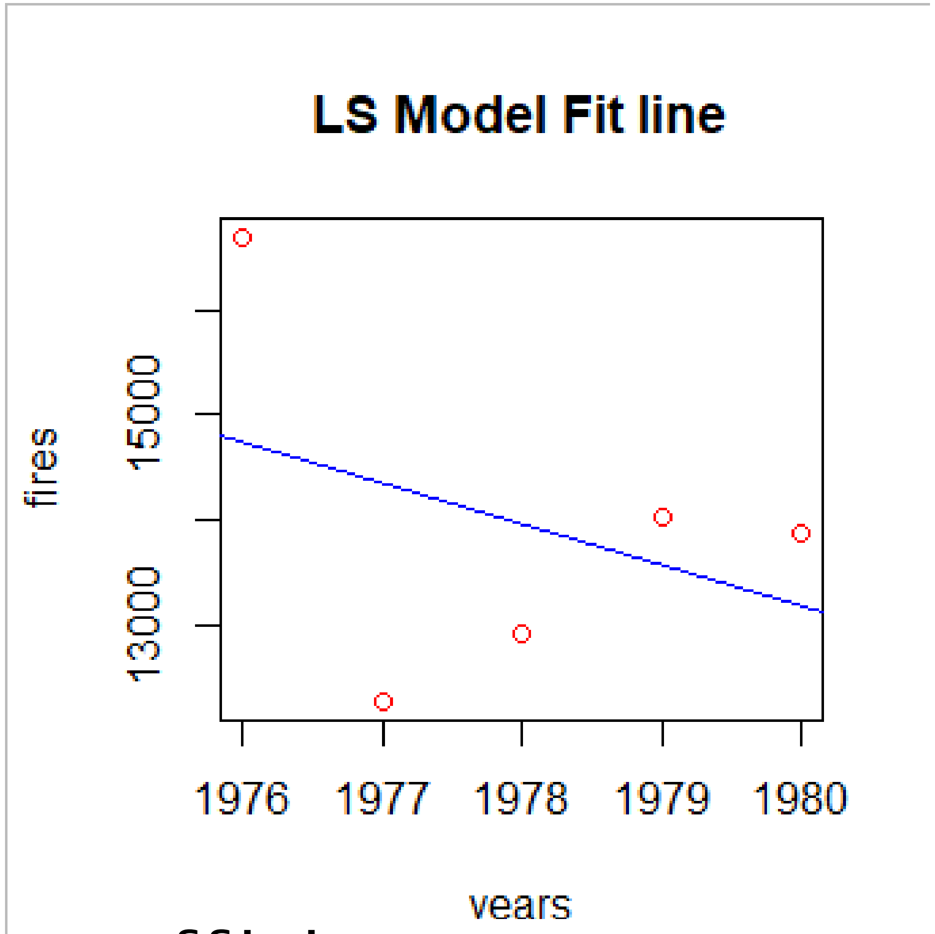
R programming Code :

```
## Fitted LMS model
x <- years[c(2,3,5)]
y <- fires[c(2,3,5)]
lms.model <- lm(y ~ x)
lms.model


## Graph of LMS

plot(years, fires, main= "LMS model line")
abline(lms.model, col= "red")
```

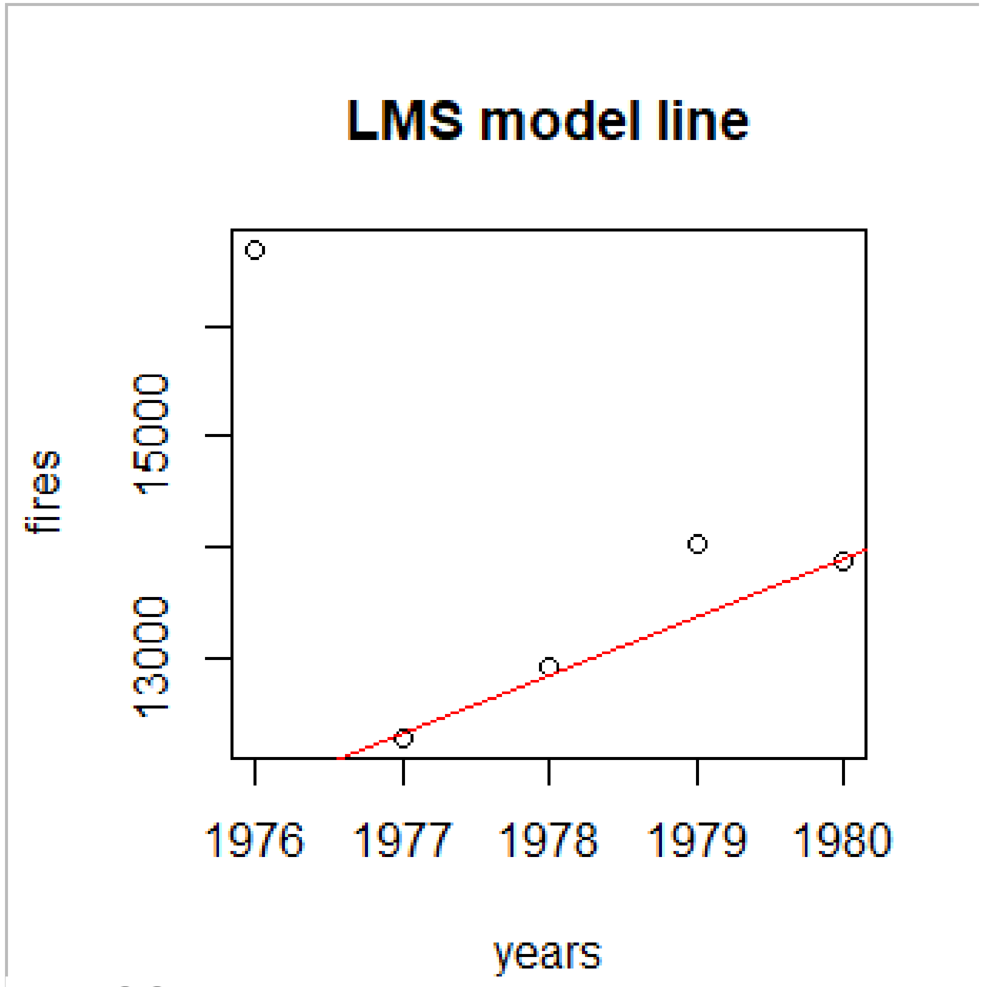



```
Coefficients:
(Intercept)          x
-1030130.6       527.3
```

Comparison between LS and LMS




```
Coefficients:
(Intercept) years
 780430.8   -387.5
```




```
Coefficients:
(Intercept)         x
-1030130.6       527.3
```

LTS- Estimator

Frist the fit the LS model from the given data then,

Let $\varepsilon^2_{(1):n} \le \varepsilon^2_{(2):n} \le \ldots \le \varepsilon^2_{(n):n}$ be the ordered squared residuals.

Thus, the objective function of the LTS estimator is given by

$$\min_{\beta} \sum_{i=1}^{h} \varepsilon^2_{(i):n}$$

Where $h = \left[(1-\alpha)n\right] + \left[(p+1)/2\right]$

α is the percentage level of trimming.

LTS is the special case of LMS when $\alpha = 50\%$

U.S Air Force Data

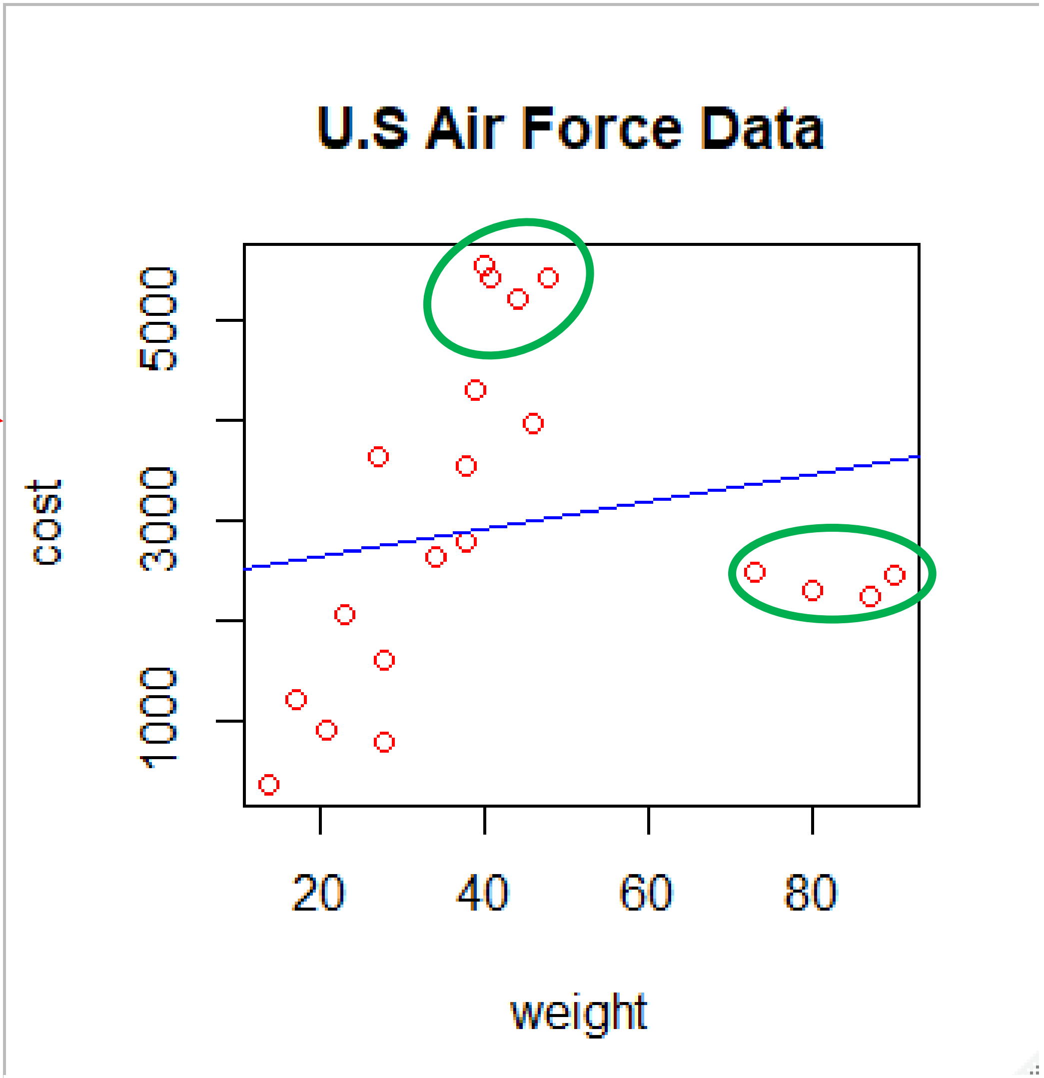
Satellite cost vs weight

R programming Code :

```
## U.S Air Force Data
## Cost and Weight

cost <- c(2449, 2248, 3545, 789, 1619, 2079, 918, 1231, 3641, 4314, 2628, 3989,
2308, 376, 5428, 2786, 2497, 5551, 5208, 5438)

weight <-c(90, 87, 38, 28, 28, 23, 21, 17, 27, 39, 34, 46, 80, 14, 48, 38, 73, 40, 44,
41)
```

R programming Code :

```
## Graph
plot(weight, cost, main = "U.S Air Force Data ")
abline(lm(cost~weight), col = "red") ## SL fit line

## create a data frame
myData <- data.frame(cost,weight)

## LS model Fit And Take Residuals
LS <- lm(cost ~ weight, data= myData)
errors <- LS$residuals
myData$residual <- errors
sort(errors)
```




```
Coefficients:
(Intercept)        weight
2373.70             13.51
```

R programming Code :

sort(errors)

```
-2186.8992 -1963.0940 -1739.4966 -1372.4410 -1301.4151 -1146.8177 -1140.9569 -1133.0940
-863.2203 -605.5245 -205.1775 -101.2332 657.7668 902.4199 993.6555 1413.2529
 2239.6833 2405.6276 2510.2250 2636.7390
```

```
## Graph of Residual
plot(weight, errors, main = "Residual plot")

## trimmed the data
n = 20
p = 1
alpha = .20      ## 20% data trimmed
h = ((1-alpha)*n)+((p+1)/2)
h= 17 ## 3 large residual omit from data respectively
```

R programming Code :

```
## trim data collection
t1.data <- myData[myData$residual < 2405, ]
t1.data
```

```
## model fit with trim data
LTS_model_1 <- lm(cost ~ weight,data = t1.data)
```

	cost	weight	residual
1	2449	90	-1140.9569
2	2248	87	-1301.4151
3	3545	38	657.7668
4	789	28	-1963.0940
5	1619	28	-1133.0940
6	2079	23	-605.5245
7	918	21	-1739.4966
8	1231	17	-1372.4410
9	3641	27	902.4199
10	4314	39	1413.2529
11	2628	34	-205.1775
12	3989	46	993.6555
13	2308	80	-1146.8177
14	376	14	-2186.8992
16	2786	38	-101.2332
17	2497	73	-863.2203
19	5208	44	2239.6833

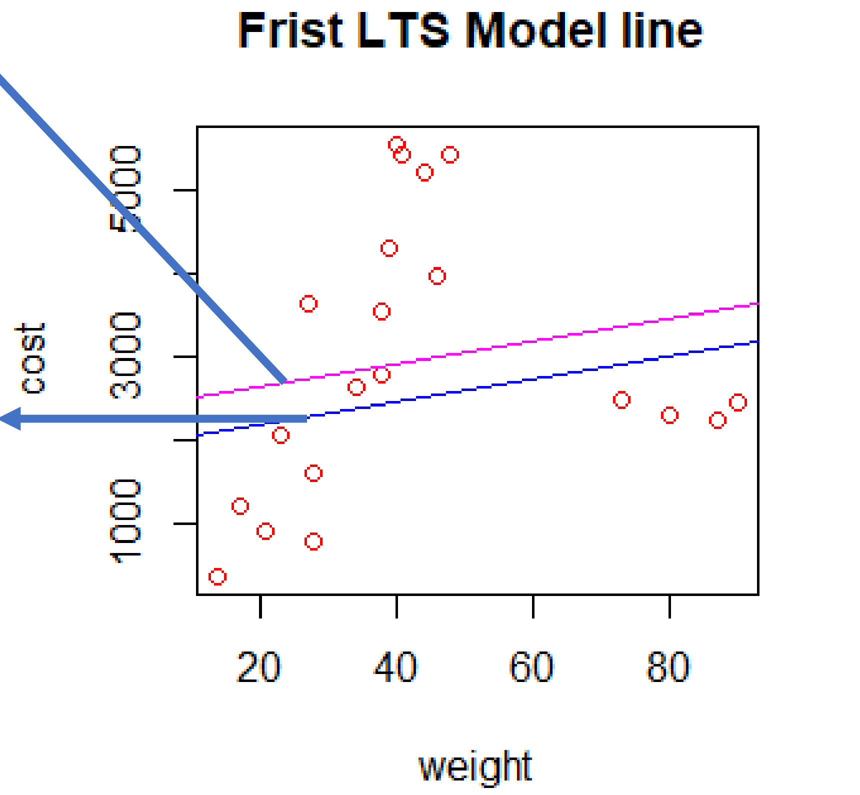
2405.6276 2510.2250 2636.7390

R programming Code :

```r
## See the model cofficient
LTS_model_1
##Graph
plot(weight, cost, main = "Frist LTS MOdel line ",
        col = "red")
abline(LTS_model_1, col = 4)
abline(lm(cost~weight), col = 6)
```

LS

LTS
1




```
Coefficients:
(Intercept)      weight
 2373.70         13.51
```

```
Coefficients:
(Intercept)      weight
1933.30          13.42
```

R programming Code :

```
## again check residual omit h obs.
t1.data$residual <- LTS_model_1$residuals
n=17
alpha = .2
p=1
h = ((1-alpha)*n)+((p+1)/2) ## h =14.6 close to 15

# sort Residuals
sort(t1.data$residual) ## sort residuals and omit 2
large residauls.
```

```
 [1] -1745.2287 -1520.1583 -1297.1935 -930.4993 -853.1473 -699.1825 -692.4179 -690.1583
 [9] -416.2177 -163.0406 238.3005 342.6063 1101.6063 1345.2653 1438.2180 1857.1827
[17] 2684.0650
```

```
## trim two data
t2.data <- t1.data[t1.data$residual < 1857, ]
```

R programming Code :

```
LTS_model_2 <- lm(cost ~ weight, data = t2.data )
## LTS 2 model

## trimmed three data
t3.data <- t2.data[t2.data$residual < 1655, ]
LTS_model_3 <- lm(cost ~ weight, data = t3.data )
## LTS 3 model
t3.data$residual <- LTS_model_3$residuals
sort(t3.data$residual)

## trimmed  four data
t4.data <- t3.data[t3.data$residual < 1406, ]
LTS_model_4 <- lm(cost ~ weight, data = t4.data )
## LTS 4 model
t4.data$residual <- LTS_model_4$residuals
sort(t4.data$residual)
```

```
Coefficients:
(Intercept)     weight
1610.81         13.88
```

```
Coefficients:
(Intercept)     weight
1235.46         16.48
```

```
Coefficients:
(Intercept)  weight
1043.88       17.64
```

R programming Code :

```
## trim five data
t5.data <- t4.data[t4.data$residual < 1071.84893, ]
LTS_model_5 <- lm(cost ~ weight, data = t5.data )
## LTS 5 model
```

```
Coefficients:
(Intercept)  weight
1043.88       17.64
```

LTS 4

Model

```
Coefficients:
(Intercept)  weight
1043.88       17.64
```

R programming Code :

```
## all model graph
plot(weight, cost, main = "All Model graph ")
abline(lm(cost~weight))
abline(LTS_model_1, col= 2)
abline(LTS_model_2, col= 3)
abline(LTS_model_3, col= 4)
abline(LTS_model_4, col= 5)
abline(LTS_model_5, col= 6)
```

Final LTS estimated is

```
Coefficients:
(Intercept)   weight
1043.88        17.64
```