# Homework4

Yi Chen

2/16/2020

# Homework 4

## Part A

### quesstion a

It is possible to arrange the four points on a straight line. For example, from left to right, the points can be A, B, C, and D. Besides, $AB = 10$,$BC = 20$, and $CD = 15$. Then all requirements can be satisfied.

## Part B

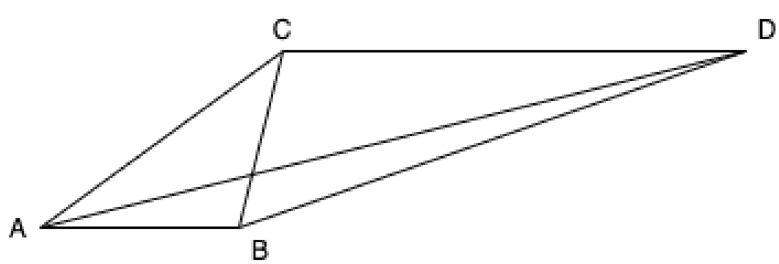
Since AD is the biggest distance, these two points must be the leftest or rightest points. Consequently, B and C muct be in the midde of A and D. There are two possible conditions:

1. ABCD: given $BD < AC$, we have $BC + CD < AB + BC$. This is equal to $CD < AB$, which break the requriement.
2. ACBD: Then $AB > AC$ and $AB > CB$ break the requriements as well.

Consequently, there is no case we can arrage this four points on a line.

## Part C

If we can arrange the points in a two-dimension plat, then it is doable. Example can be seem from plot below.




## Part B

### quesstion 1

```
d1 <- c(0.914,-0.660,-0.649,0.923,-0.589)
d2 <- c(-0.229,-0.078,-0.648,0.172,0.716)
distance <- c()
distance_matrix <- matrix(0,nrow = 5,ncol = 4)
for (i in 1:4){
  for (j in (i+1):5){
    d <- sqrt((d1[i] - d1[j])^2 + (d2[i] - d2[j])^2)
    distance <- c(distance,d)
    distance_matrix[j,i] <- d
  }
}
round(distance_matrix,2)
```

```
##      [,1] [,2] [,3] [,4]
## [1,] 0.00 0.00 0.00 0.00
## [2,] 1.58 0.00 0.00 0.00
## [3,] 1.62 0.57 0.00 0.00
## [4,] 0.40 1.60 1.77 0.00
## [5,] 1.78 0.80 1.37 1.61
```

## quesstion 2

The rank of proximity data is:

$\delta_{14} = 125$, $\delta_{24} = 600$, $\delta_{25} = 670$, $\delta_{35} = 675$, $\delta_{13} = 800$, $\delta_{24} = 850$, $\delta_{12} = 875$, $\delta_{15} = 890$, $\delta_{45} = 890$, and $\delta_{34} = 925$.

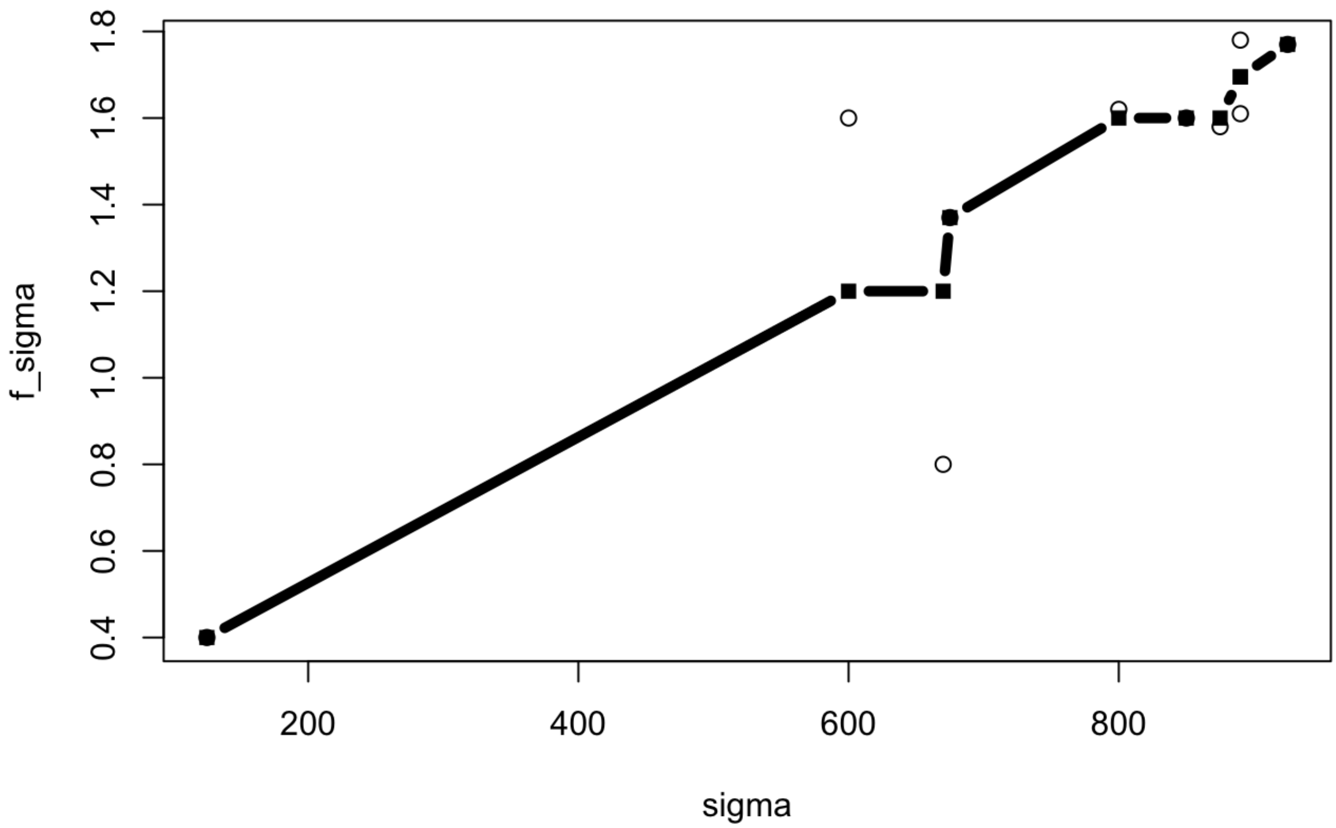
The model distance is:

$d_{14} = 0.40$, $d_{24} = 1.60$, $d_{25} = 0.80$, $d_{35} = 1.37$, $d_{13} = 1.62$, $d_{24} = 1.60$, $d_{12} = 1.58$, $d_{15} = 1.78$, $d_{45} = 1.61$, and $d_{34} = 1.77$.

I use the free ordering approach. After several iteration of transformation, I get:

$f(\sigma) = (0.4, 1.2, 1.2, 1.37, 1.6, 1.6, 1.6, 1.695, 1.695, 1.77)$

```
sigma <- c(125,600,670,675,800,850,875,890,890,925)
f_sigma <- c(0.4, 1.2, 1.2, 1.37, 1.6, 1.6, 1.6, 1.695, 1.695, 1.77)
d <- c(0.4,1.6,0.8,1.37,1.62,1.6,1.58,1.78,1.61,1.77)
plot(sigma,f_sigma,type="b",lwd=5, pch=15)
points(sigma,d)
```




## quesstion 3

I choose to use the recommended stress fomula 2.

```
sqrt((sum((f_sigma-d)^2) / sum((d - mean(d))^2)))
```

```
## [1] 0.4262241
```

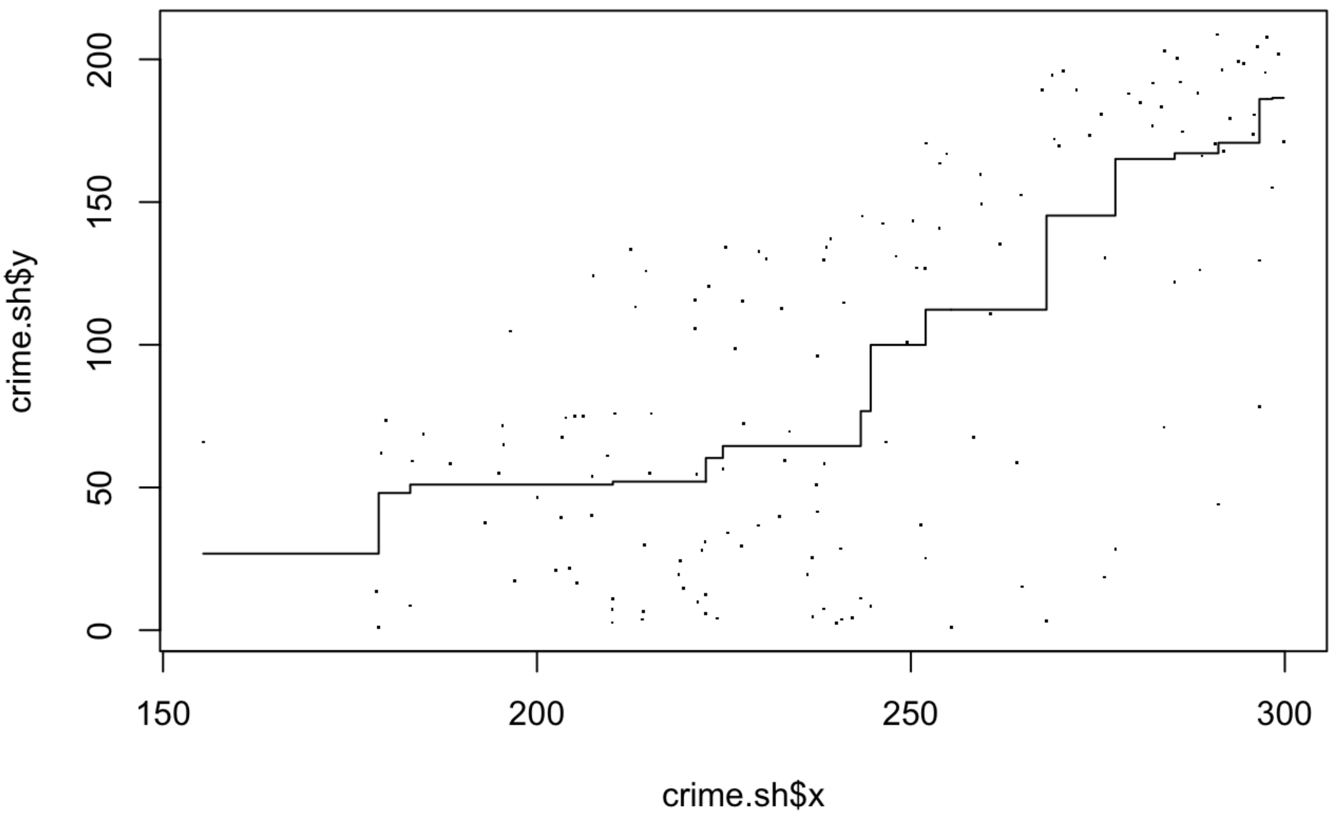
# Part C

## quesstion a

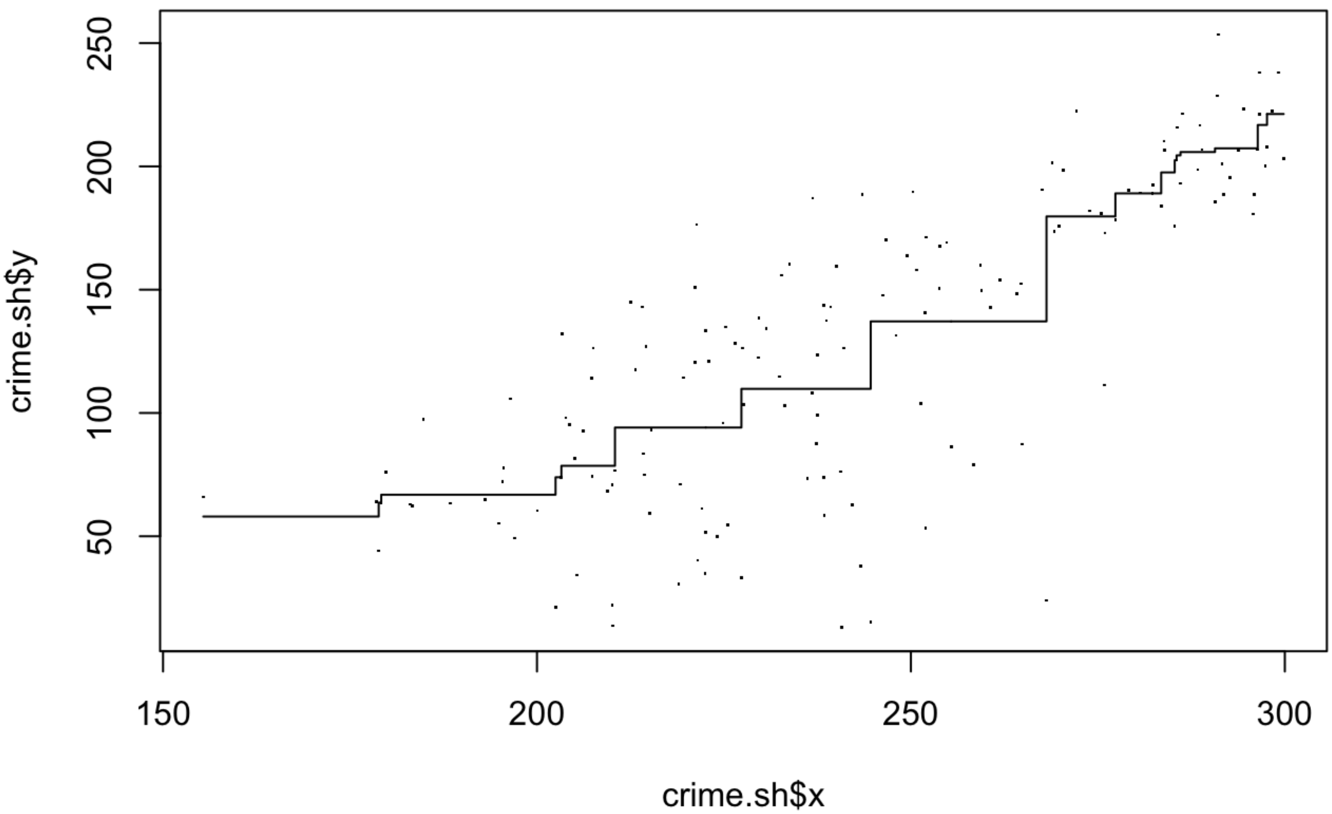
#### Issue (1) and (2)

```
library(readr)
library(MASS)

CRIMES_PRX_clean <- read_csv("CRIMES_PRX_clean.csv")
crime.label <- c()
for (i in 2:nrow(CRIMES_PRX_clean)){
  crime.label <- c(crime.label,CRIMES_PRX_clean[i,1])
}
CRIMES_PRX_clean <- as.matrix(CRIMES_PRX_clean)
CRIMES_PRX_clean <- CRIMES_PRX_clean[2:nrow(CRIMES_PRX_clean),2:ncol(CRIMES_PRX
_clean)]
rownames(CRIMES_PRX_clean) <- 1:18
colnames(CRIMES_PRX_clean) <- 1:18



crime <- matrix(NA,nrow = 18,ncol = 18)

for (i in 1:17){
  for (j in (i+1):18){
    crime[i,j] <- as.numeric(CRIMES_PRX_clean[j,i])
    crime[j,i] <- as.numeric(CRIMES_PRX_clean[j,i])
  }
}
diag(crime) <- 0

for (i in 1:4){
  crime.dist <- dist(crime)
  crime.mds <- isoMDS(crime.dist,k = i)
  print(crime.mds$stress)
  crime.sh <- Shepard(crime.dist, crime.mds$points)
  plot(crime.sh, pch = ".")
  lines(crime.sh$x, crime.sh$yf, type = "S")
}
```

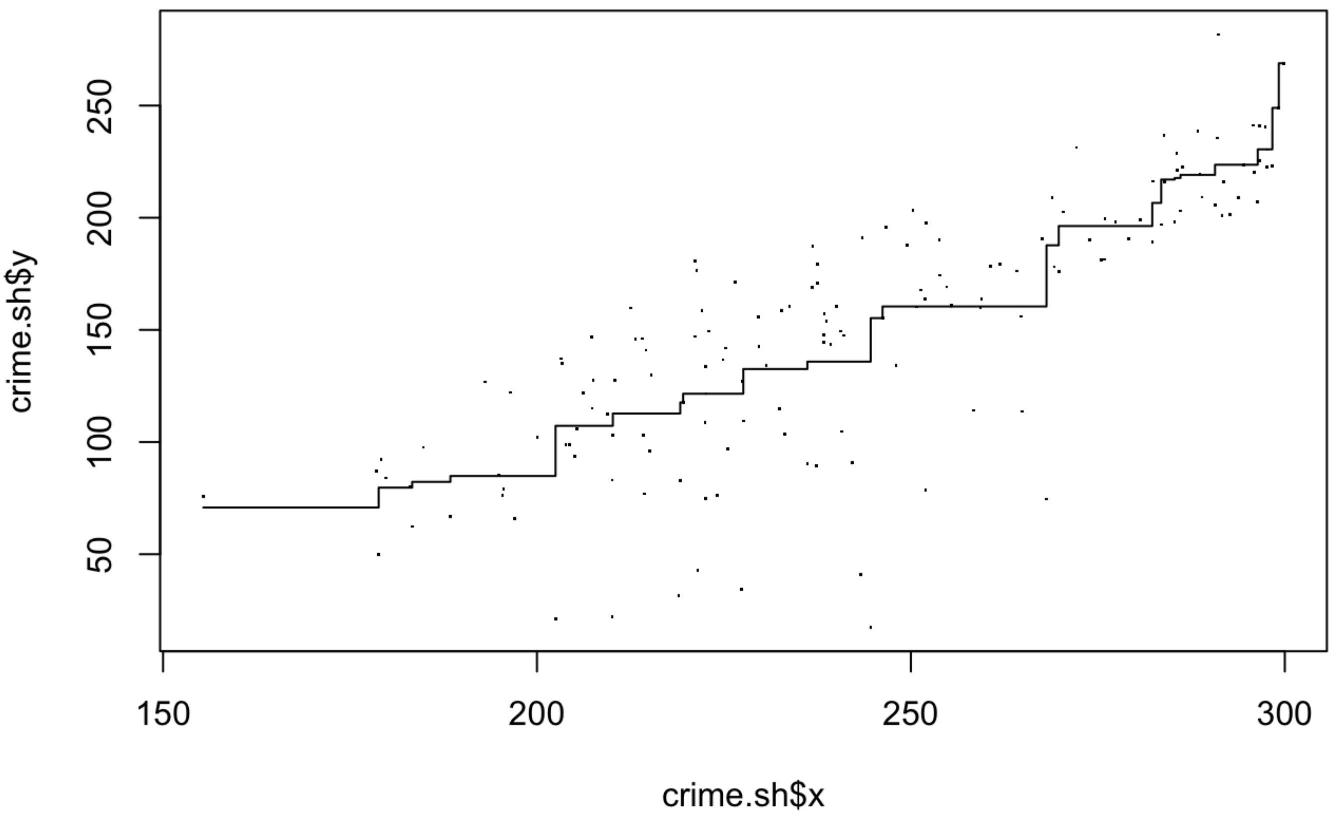
```
## initial  value 39.433243
## final  value 39.421576
## converged
## [1] 39.42158
```



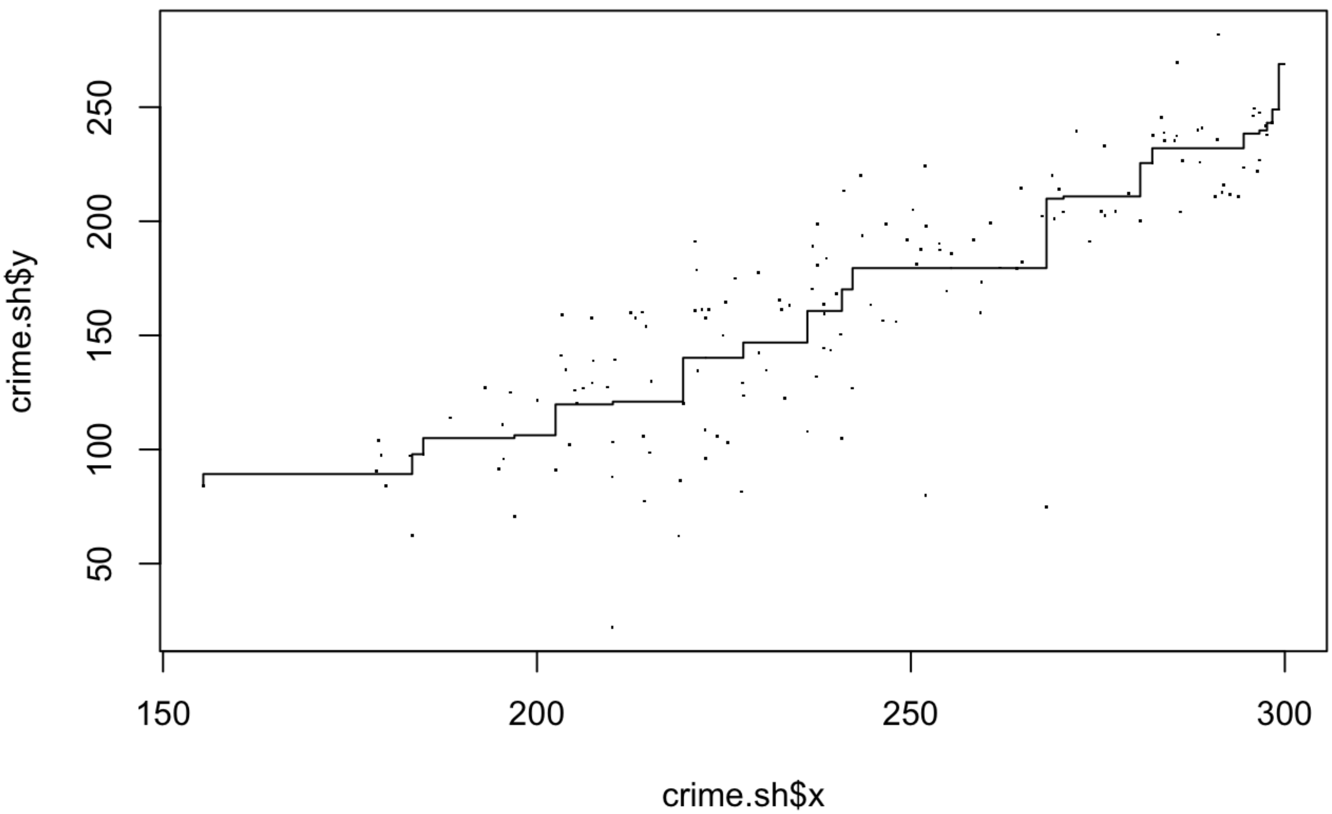

```
## initial  value 24.342310
## final  value 24.331069
## converged
## [1] 24.33107
```




```
## initial  value 19.649508
## final  value 19.641262
## converged
## [1] 19.64126
```




```
## initial  value 15.077546
## final  value 15.072725
## converged
## [1] 15.07272
```



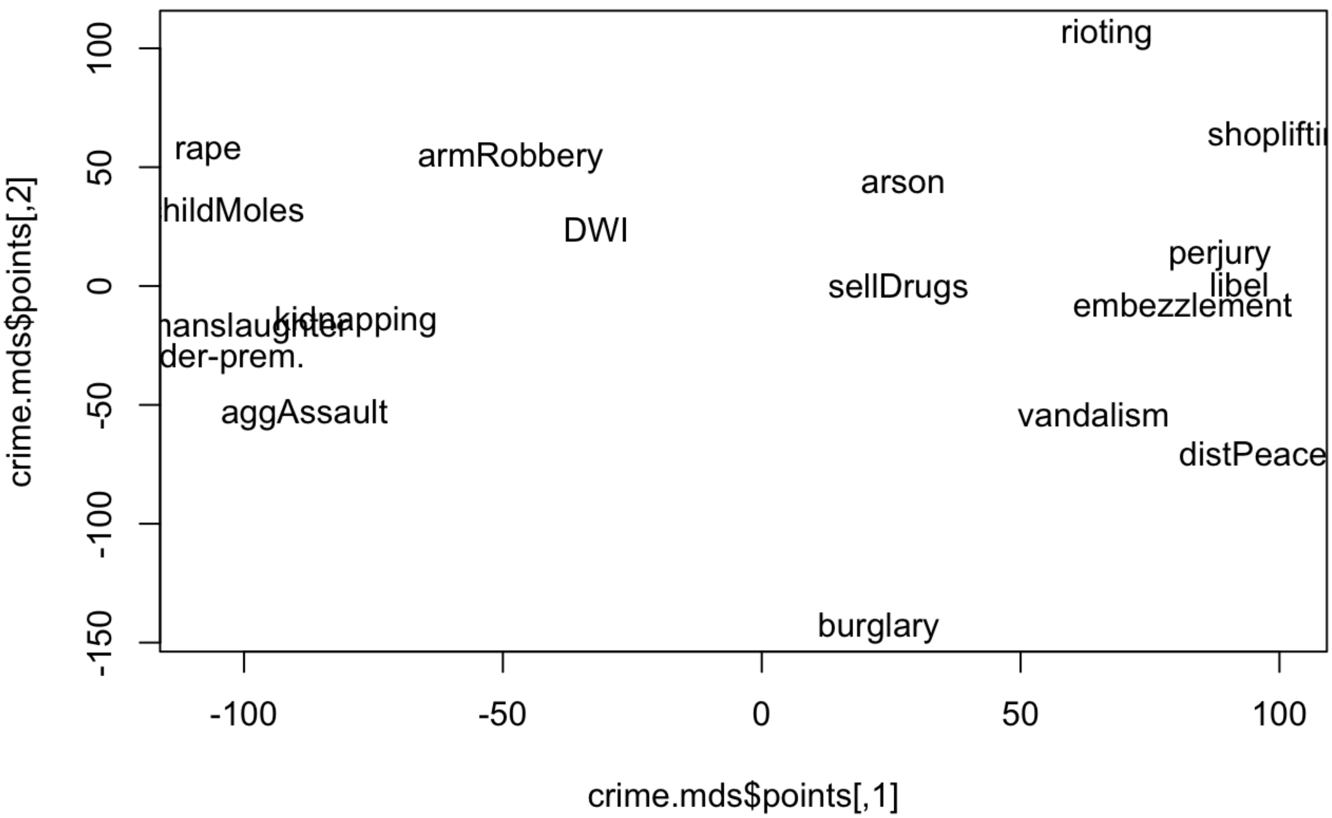

With more dimension, the stress will always decrease. But the difficulity of interpretaion will also increase. The decrease of stree is big between dimension of 1 and 2, but the decreases are not obvious after that. I pick 2 dimension.

**Issue (3)**

```
crime.mds <- isoMDS(crime.dist,k = 2)
```

```
## initial  value 24.342310
## final  value 24.331069
## converged
```

```
plot(crime.mds$points, type = "n")
text(crime.mds$points, labels = as.character(crime.label))
```




The first dimension is about Criminal motivation.

The second dimension is about level of vialence.

### quesstion b

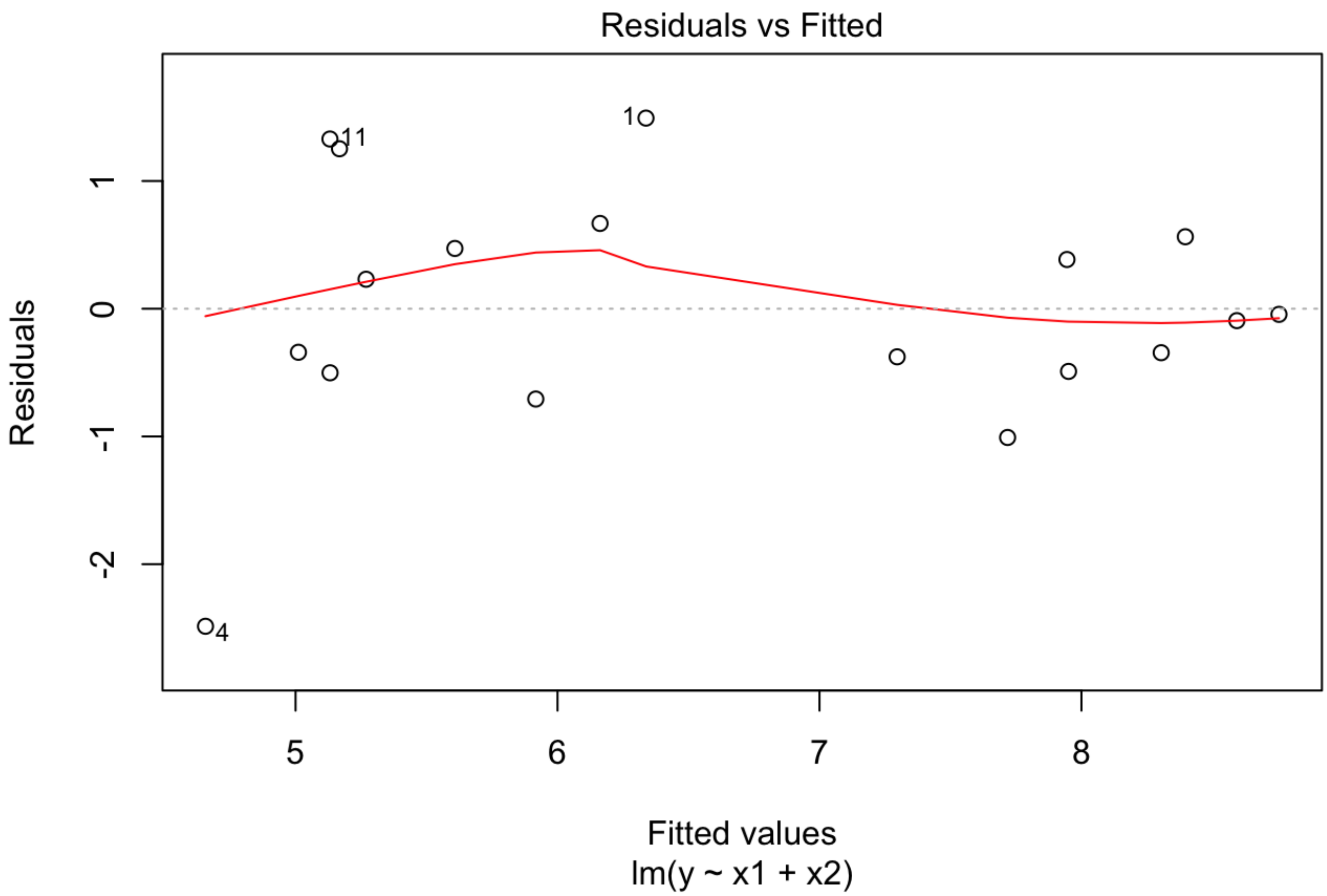
```
y <- c(7.83,6.08,8.50,2.17,6.92,6.42,8.33,4.67,7.96,8.96,6.46,7.46,8.71,5.21,6.
71,6.83,4.63,5.50)
x1 <- as.numeric(crime.mds$points[,1])
x2 <- as.numeric(crime.mds$points[,2])
data <- cbind(x1,x2,y)
data <- cbind(data,as.character(crime.label))
colnames(data) <- c("x1","x2","Y","stim")
data <- as.data.frame(data)
data
```

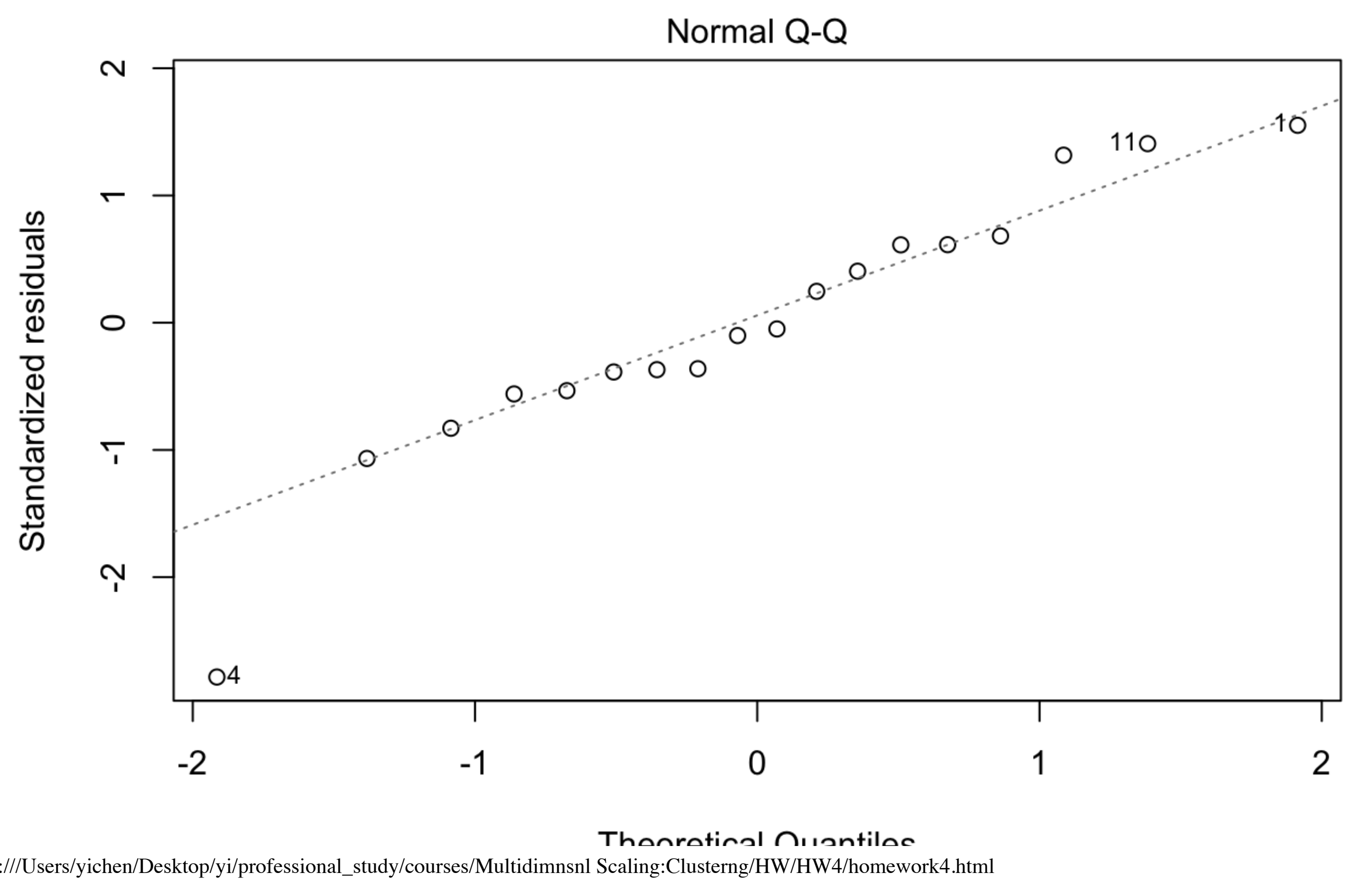
```
##                   x1                  x2    Y        stim
## 1   27.3428766803534   43.2692387521394 7.83        arson
## 2   22.6149481596888  -143.814153923229 6.08     burglary
## 3  -103.631981578667   32.2851216373103  8.5   childMoles
## 4   94.9567113564059  -70.1122164313839 2.17    distPeace
## 5  -31.9596464862374   23.9183851272305 6.92          DWI
## 6    81.306826413004  -7.61462113389714 6.42 embezzlement
## 7   -78.366685535533  -14.7375411226057 8.33   kidnapping
## 8   92.2692057082945  0.680000123324426 4.67        libel
## 9  -99.4722572115975  -17.3736335597607 7.96 manslaughter
## 10 -107.870356690644  -30.1688593978231 8.96 murder-prem.
## 11  88.4975020911547   13.1046712188385 6.46      perjury
## 12 -88.3375136713933  -53.7207790383159 7.46   aggAssault
## 13 -106.922539112357   55.9823434724491 8.71         rape
## 14  66.7151031851849   105.839705575384 5.21      rioting
## 15 -48.5151030216206   53.9565325917624 6.71   armRobbery
## 16  26.4139391572079 -0.868134195669461 6.83    sellDrugs
## 17  100.816193013619   63.1004974559511 4.63  shoplifting
## 18  64.1427775431369  -53.7265571517047  5.5    vandalism
```

```
regression <- lm(formula = y~x1+x2)
regression
```

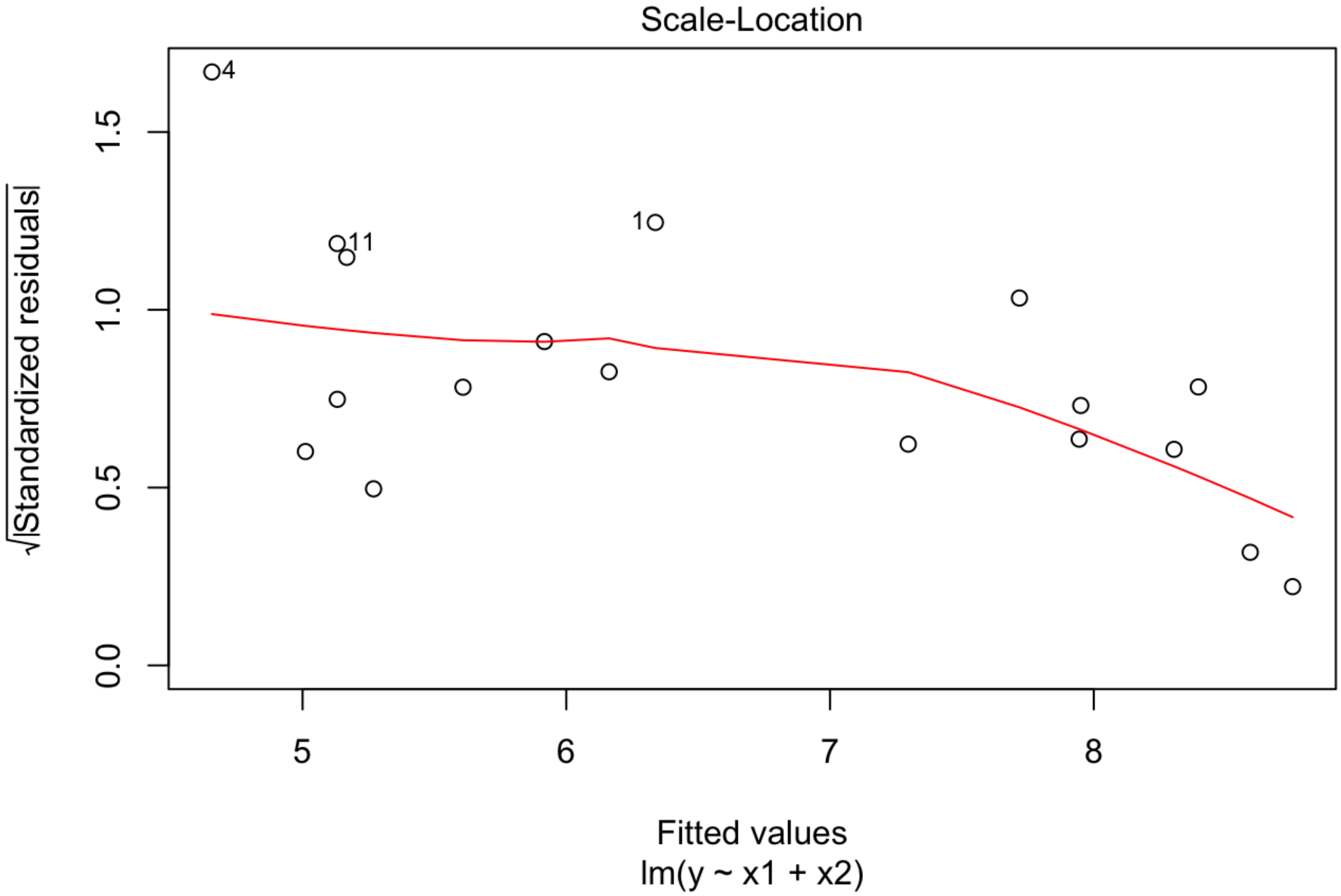
```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Coefficients:
## (Intercept)           x1           x2
##    6.630556    -0.017586     0.004343
```

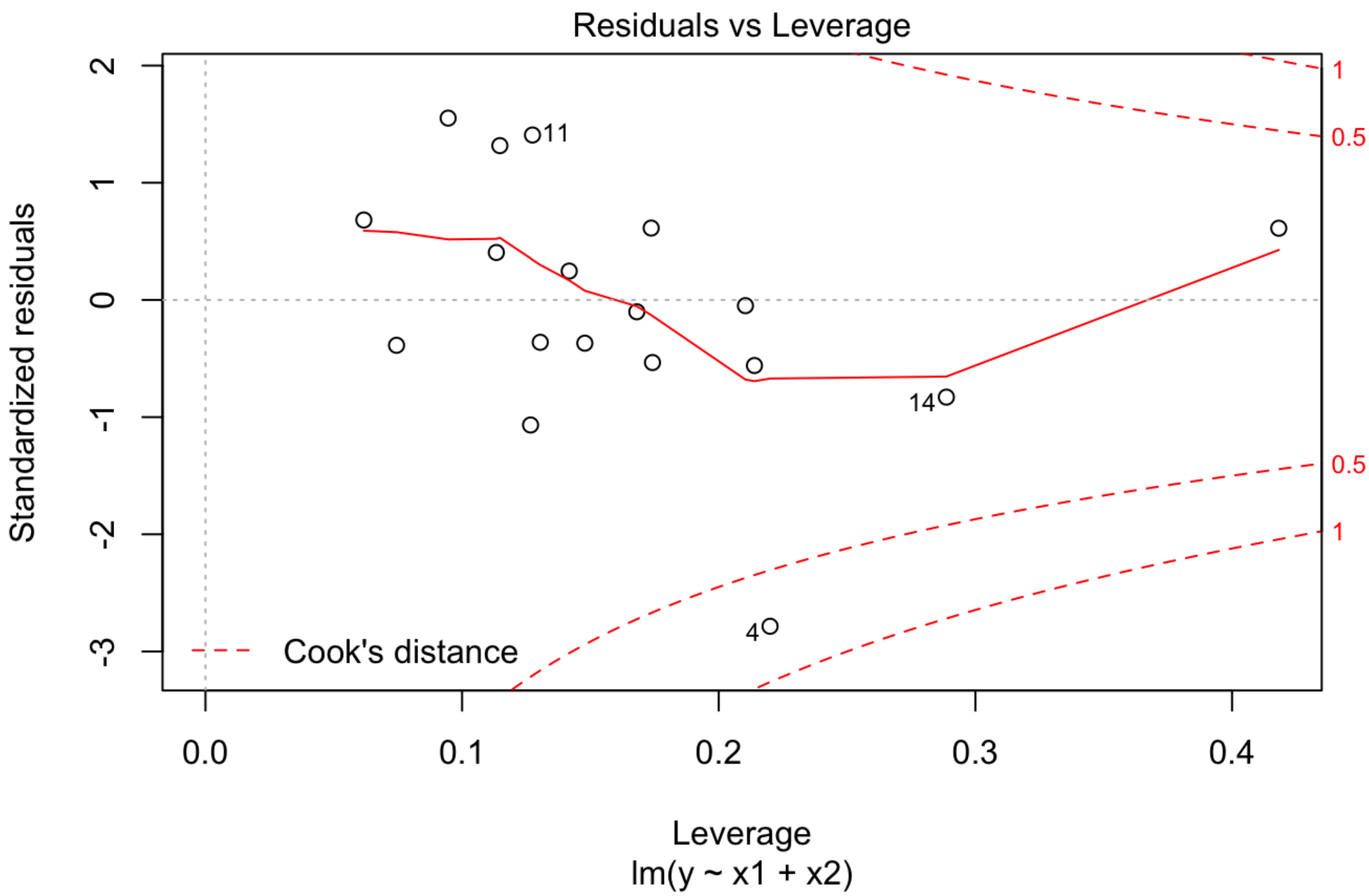
```
plot(regression)
```


Residuals vs Fitted
Residuals
Fitted values
lm(y ~ x1 + x2)
11
1
4
1
0
-1
-2
5
6
7
8


Normal Q-Q
Standardized residuals
11
1
4
2
1
0
-1
-2
-2
-1
0
1
2

Theoretical Quantiles
lm(y ~ x1 + x2)




Residuals vs Leverage
Standardized residuals
Leverage
lm(y ~ x1 + x2)
Cook's distance
11
14
4
1
0.5
0.5
1
2
1
0
-1
-2
-3
0.0
0.1
0.2
0.3
0.4