

ECS 132 - Project

*Teresa Li, sgtli@ucdavis.edu
Wenjing Fu, luffu@ucdavis.edu*

2018-05-18

Contents

Design	2
Question 1	2
Question 2	2
Question 3	4
Question 4	4
Question 5	6
Detection	6
Step 1	6
Step 2	6
Step 3	7
Step 4	8
Step 5	9
Step 6	10
Step 7	11
Step 8	11
Implementation	13
1	13

Design

Question 1

```
Traffic_data_orig <- read.csv("Traffic_data_orig.csv", header=TRUE)
message <- "this is a secret message"
raw <- charToRaw(message)
time = Traffic_data_orig$Time
num = as.integer(rawToBits(raw))

delays = numeric(length(time) - 1)
for (i in (1:(length(time) - 1))) {
  delays[i] = time[i+1] - time[i]
}

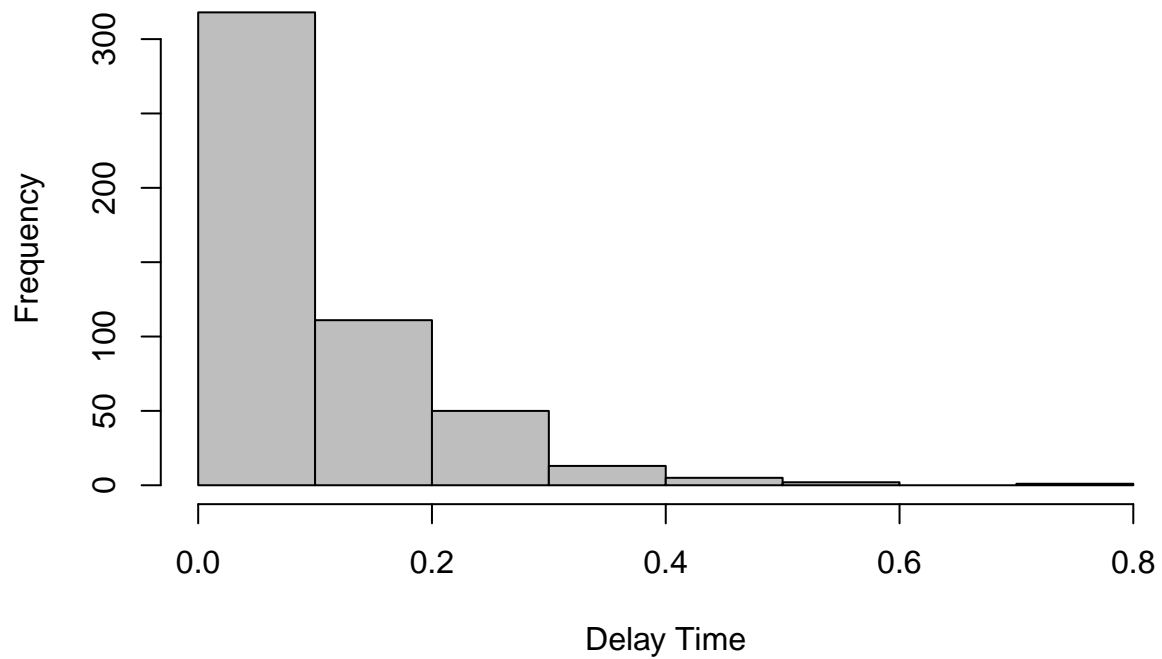
index = 1
bitlen = length(raw)*8
encrpt <- numeric(length(raw)*8)
for (i in (0:(length(raw)-1))) {
  for (j in 1:8) {
    if (num[i*8+j] == 0) {
      encrpt[index] = 0.25
    }
    else {
      encrpt[index] = 0.75
    }
    index = index+1
    j = j-1
  }
}

delays2 = delays
for (i in (1:bitlen)) {
  delays2[i] = encrpt[i]
}
```

Question 2

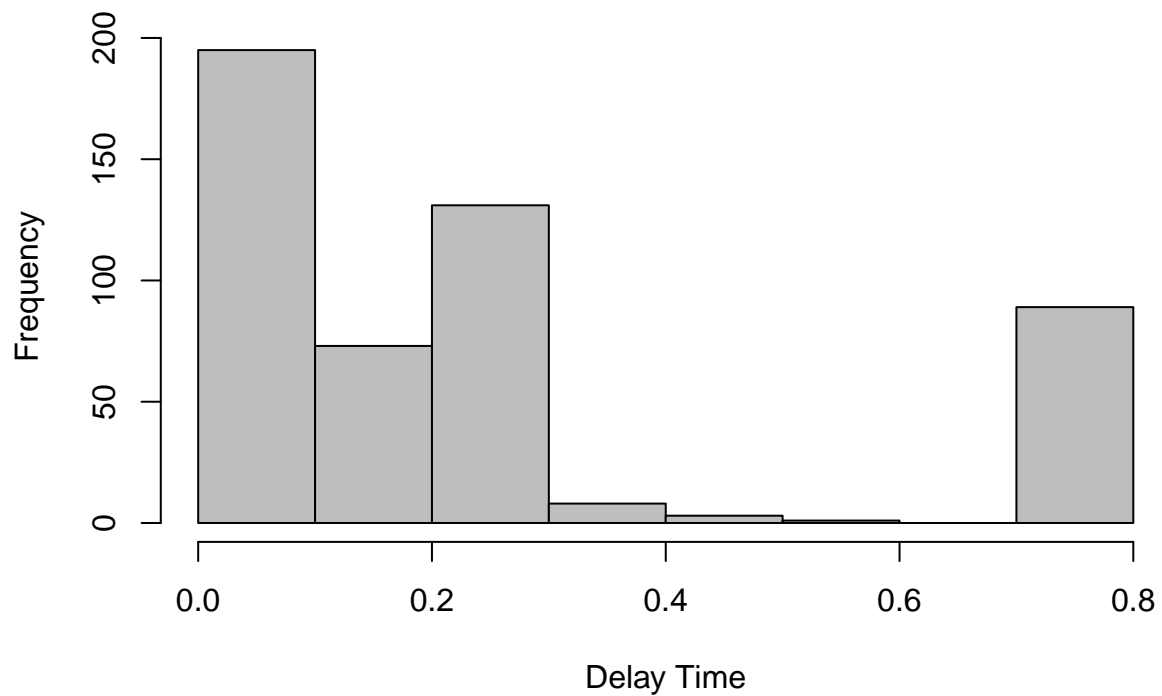
```
hist(delays, col='grey', xlab = 'Delay Time',
     main = 'Histogram of Overt Packet Stream')
```

Histogram of Overt Packet Stream



```
hist(delays2, col='grey', xlab = 'Delay Time',  
     main = 'Histogram of Convert Packet Stream')
```

Histogram of Convert Packet Stream



will be suspicious because it is obvious that the distribution changed.

Yes, Eve

Question 3

```
Traffic_data_orig <- read.csv("Traffic_data_orig.csv", header=TRUE)
message <- "this is a secret message"
raw <- charToRaw(message)
time = Traffic_data_orig$Time
num = as.integer(rawToBits(raw))
delays = numeric(length(time) - 1)
for (i in (1:(length(time) - 1))) {
  delays[i] = time[i+1] - time[i]
}
m = median(delays)
max = max(delays)
min = min(delays)

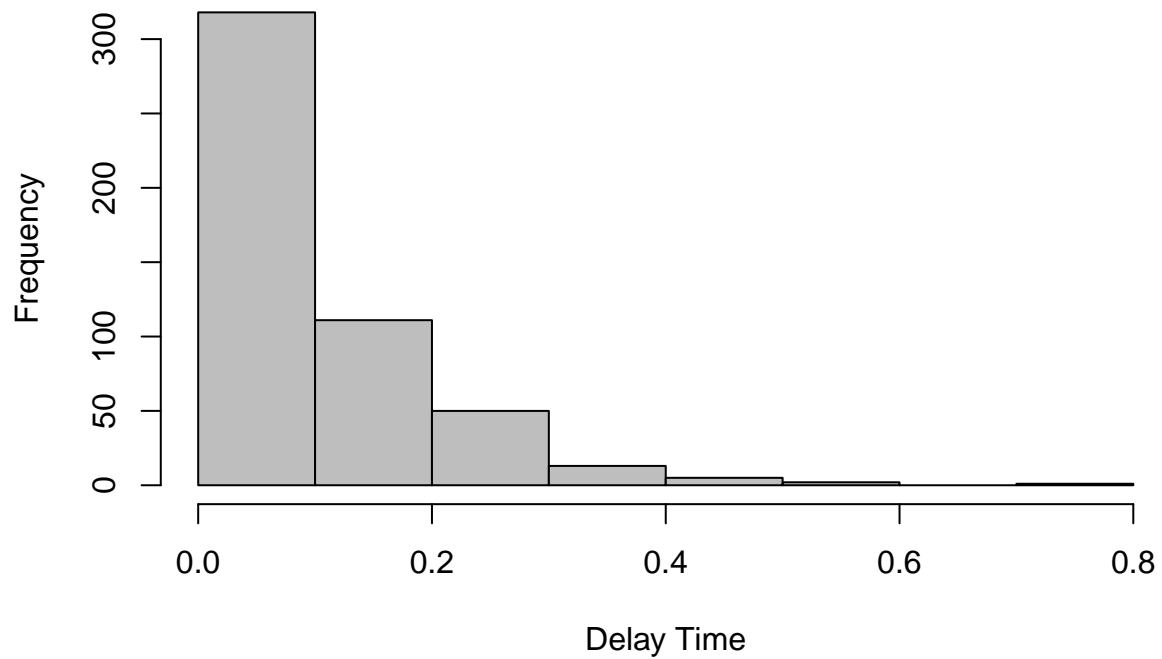
index = 1
bitlen = length(raw)*8
encrpt <- numeric(length(raw)*8)
for (i in (0:(length(raw)-1))) {
  for (j in 1:8) {
    if (num[i * 8 + j] == 0) {
      encrpt[index] = runif(1, min, m)
    }
    else {
      encrpt[index] = runif(1, m, max)
    }
    index = index + 1
    j = j - 1
  }
}

delays3 = delays
for (i in (1:bitlen)) {
  delays3[i] = encrpt[i]
}
```

Question 4

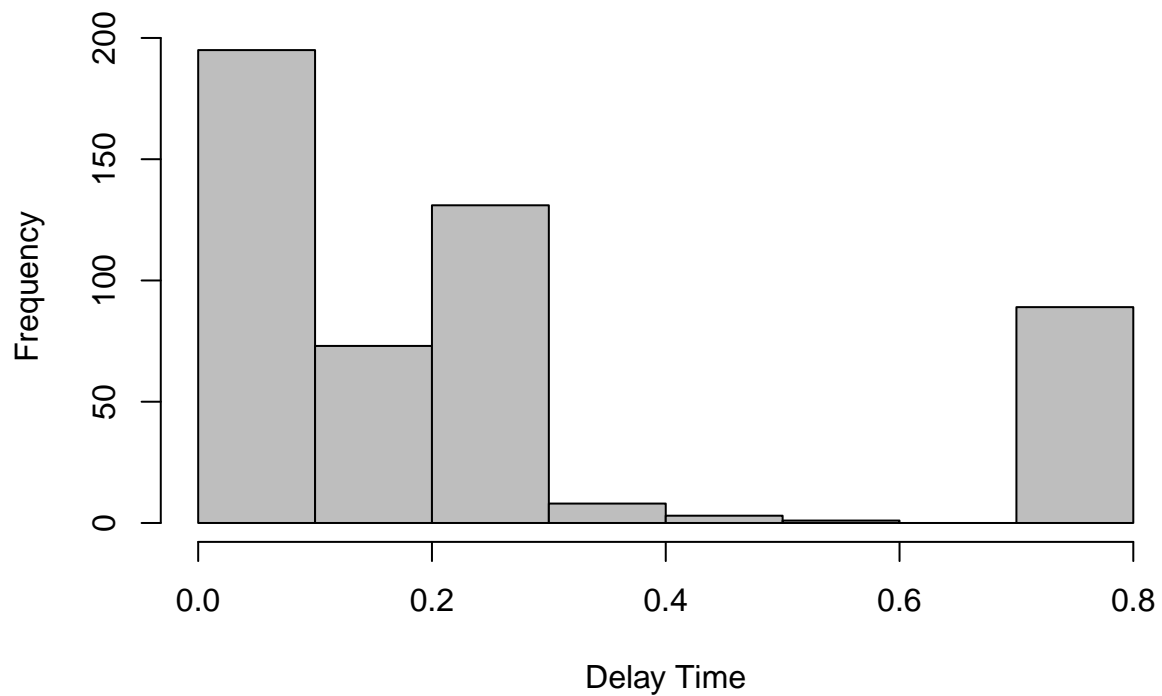
```
hist(delays, col='grey', xlab = 'Delay Time',
     main = 'Histogram of Overt Packet Stream')
```

Histogram of Overt Packet Stream



```
hist(delays2, col='grey', xlab = 'Delay Time',  
     main = 'Histogram of Convert Packet Stream')
```

Histogram of Convert Packet Stream



Eva will not be suspicious.

I think

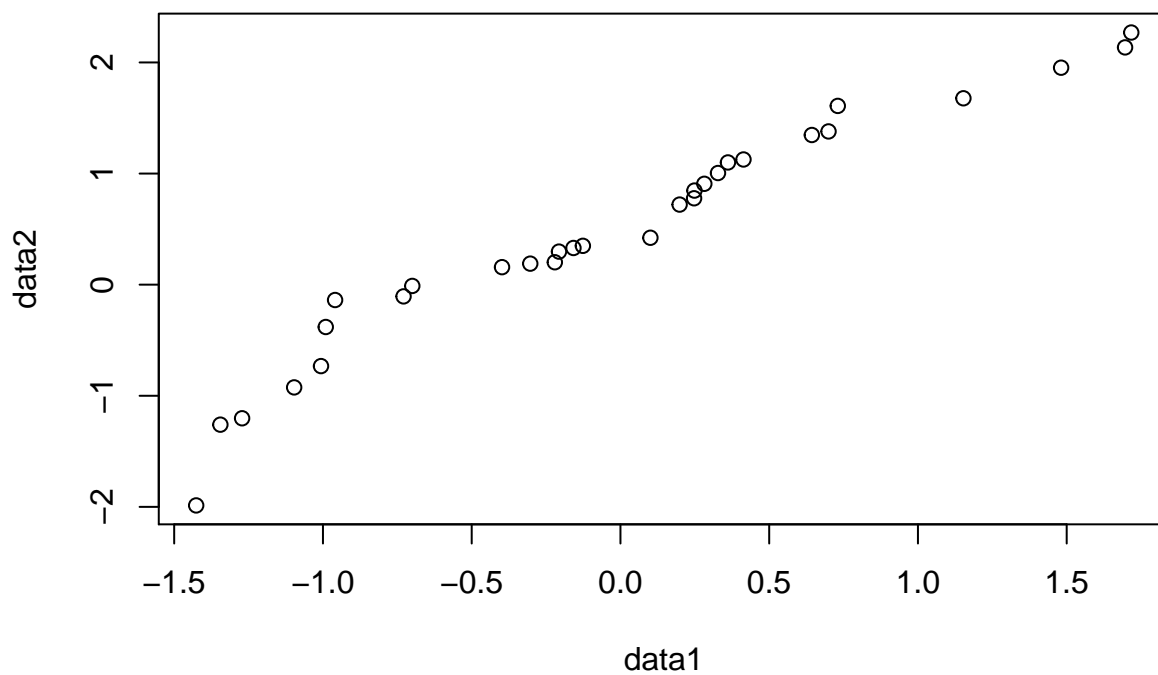
Question 5

1. Instead of generating random number from m to max, and min to m, we can choose one of the existing one from m to max, and min to m.
- 2.
- 3.

Detection

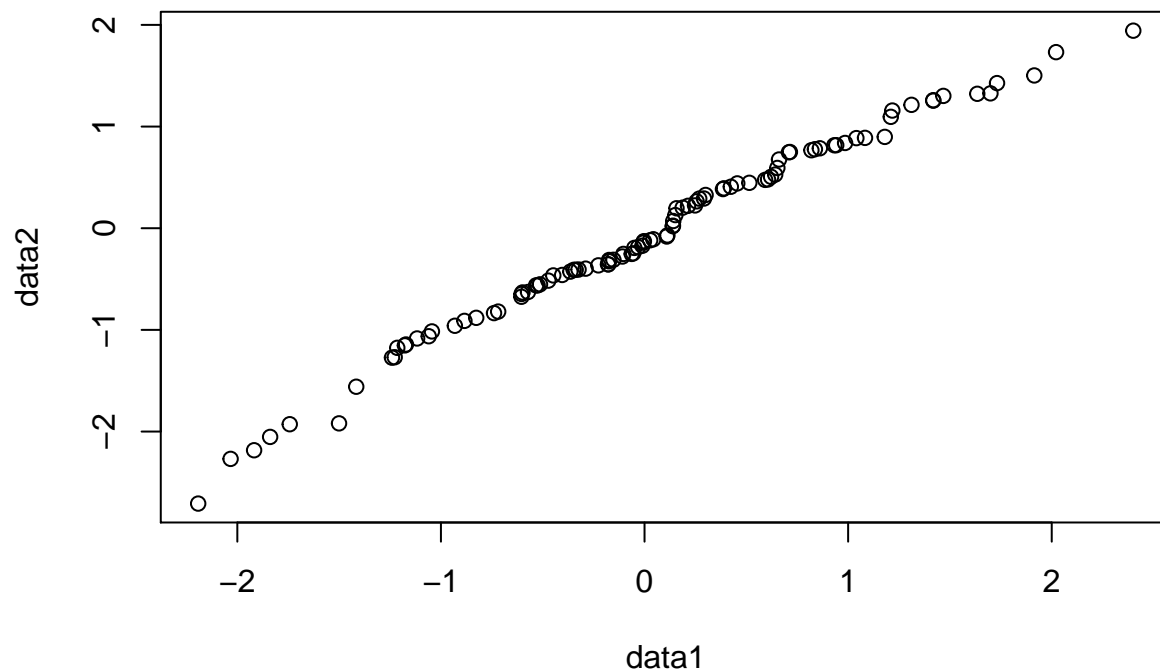
Step 1

```
data1 <- rnorm(30)
data2 <- rnorm(30)
qqplot(data1, data2)
```

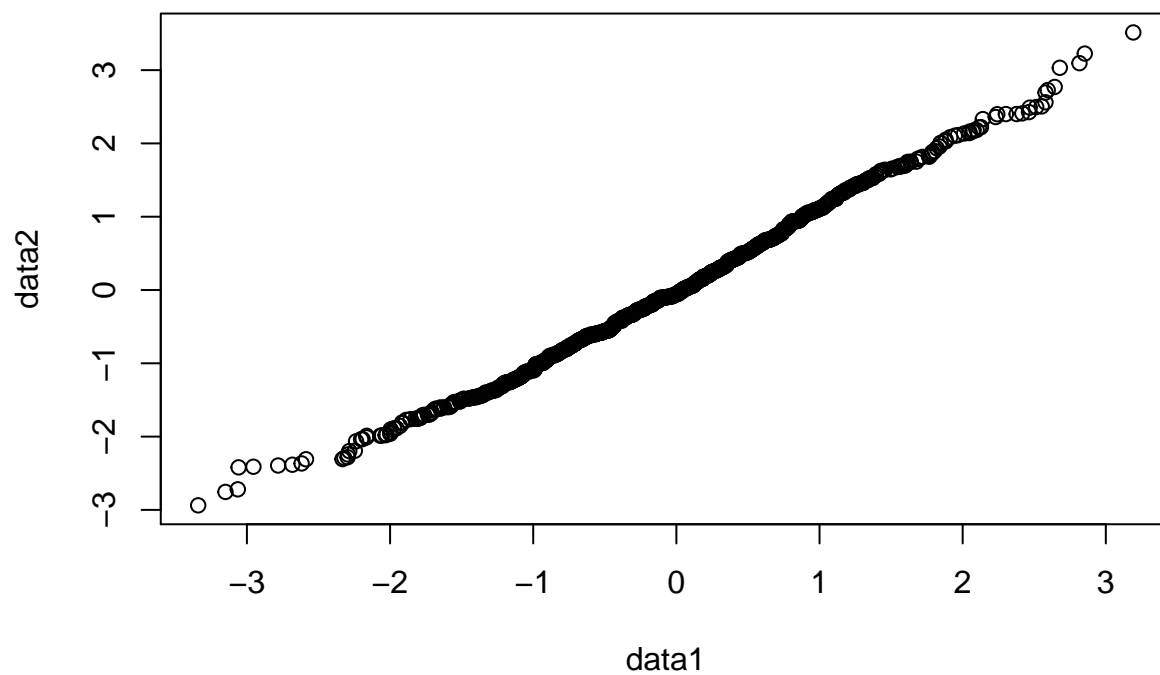


Step 2

```
data1 <- rnorm(100)
data2 <- rnorm(100)
qqplot(data1, data2)
```



```
data1 <- rnorm(1000)
data2 <- rnorm(1000)
qqplot(data1, data2)
```

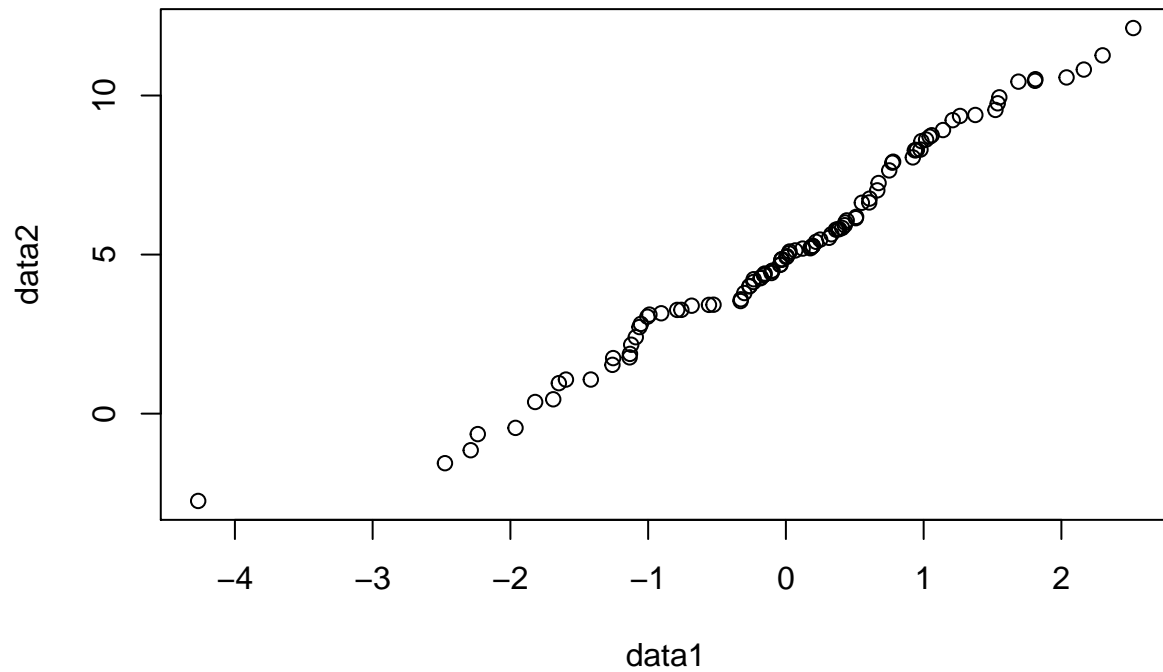


Two plots are directly proportional to each other.

Step 3

```
data1 <- rnorm(100)
data2 <- rnorm(100, mean = 5, sd = 3)
```

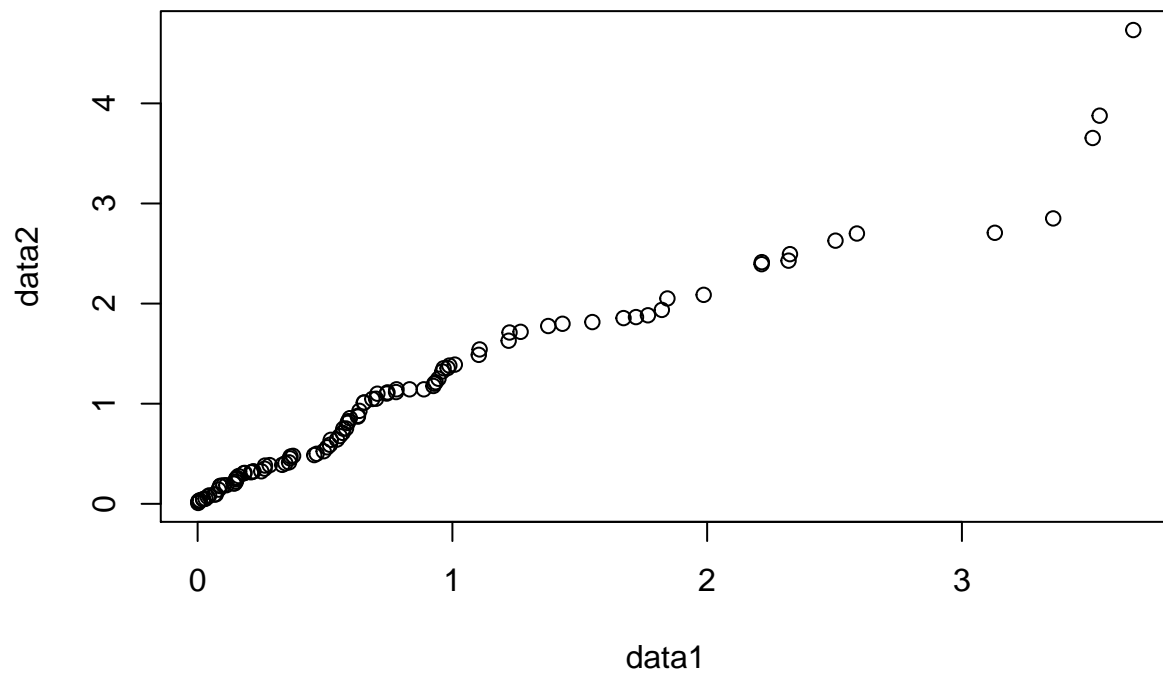
```
qqplot(data1, data2)
```



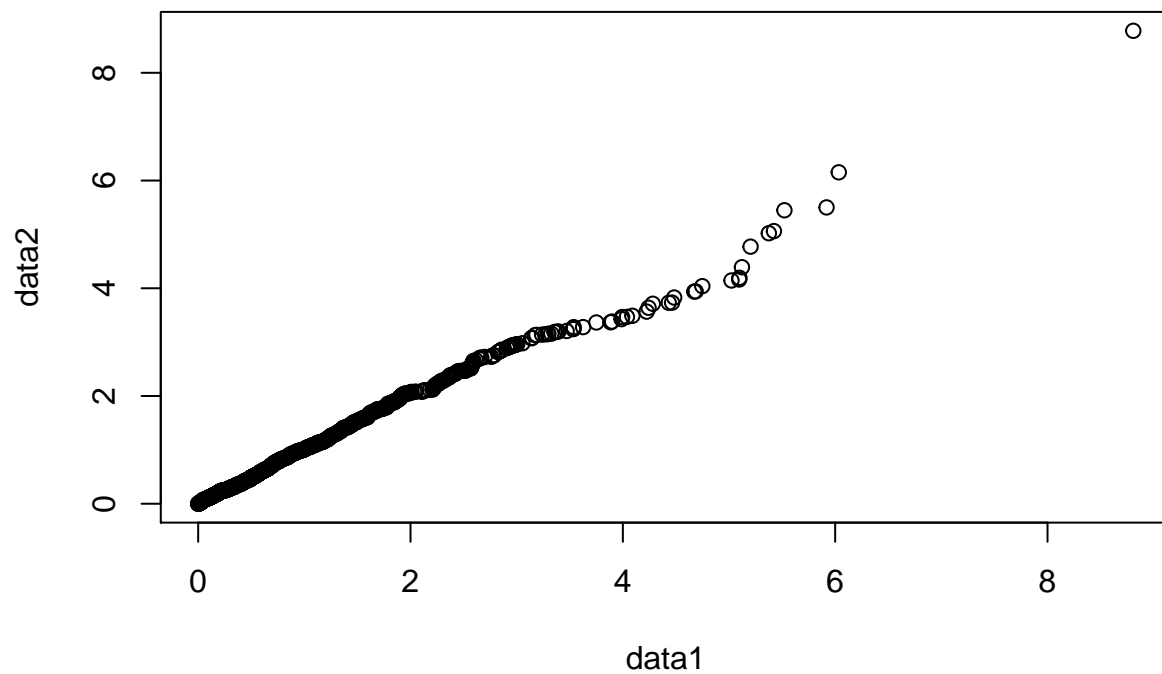
data2 is directly proportional to data1, but the slope is different this time.

Step 4

```
data1 <- rexp(100)  
data2 <- rexp(100)  
qqplot(data1, data2)
```



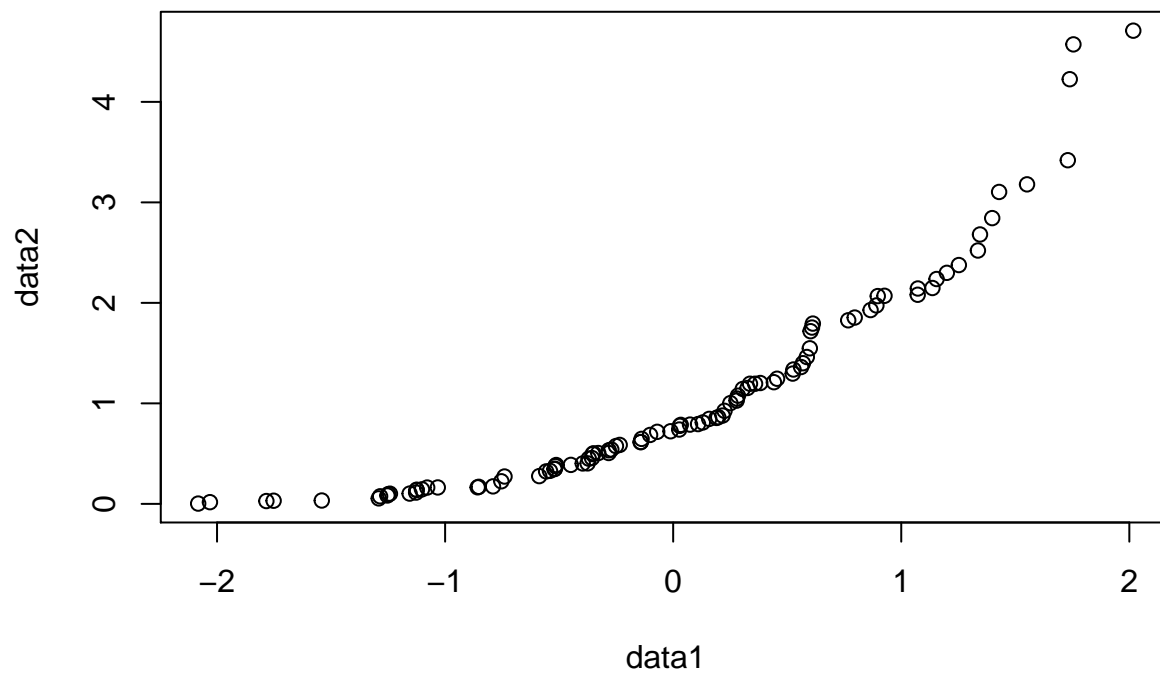

```
data1 <- rexp(1000)
data2 <- rexp(1000)
qqplot(data1, data2)
```



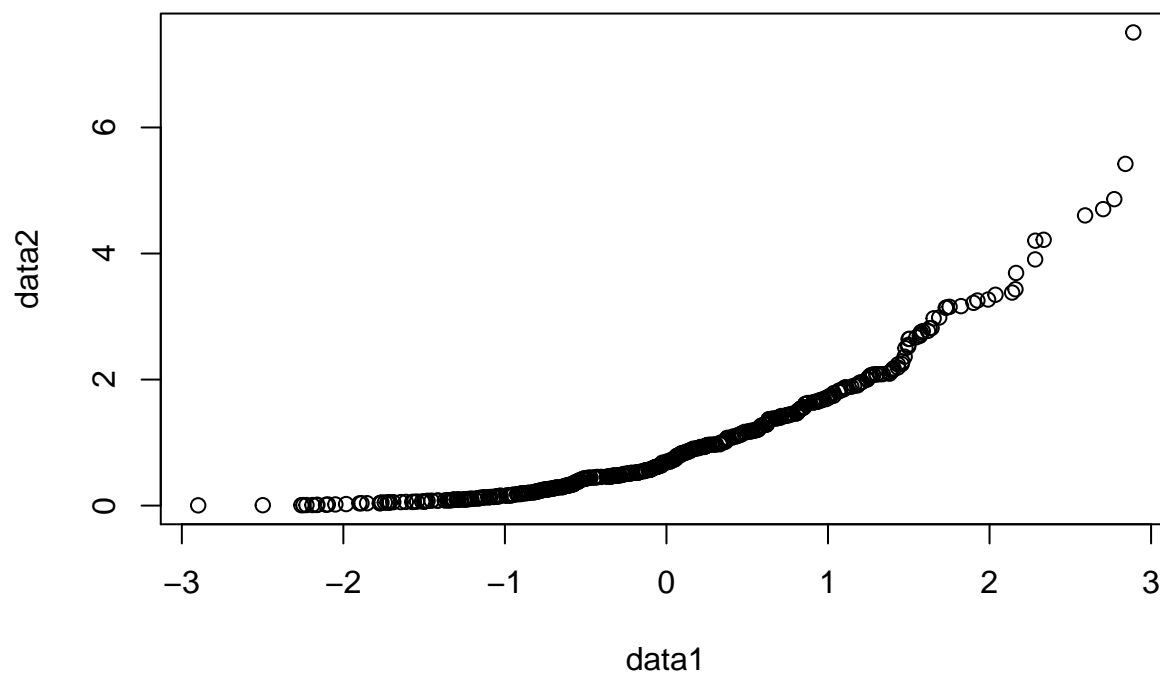
With a bigger size, two data sets are more consistent.

Step 5

```
data1 <- rnorm(100)
data2 <- rexp(100)
qqplot(data1, data2)
```



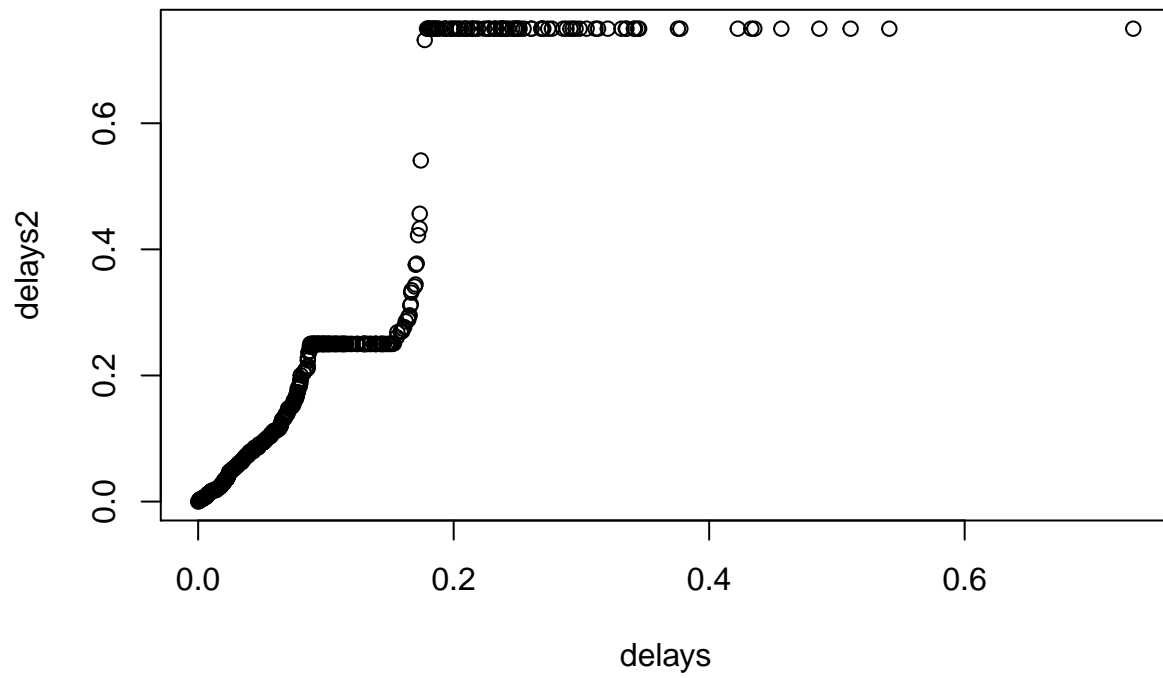
```
data1 <- rnorm(500)
data2 <- rexp(500)
qqplot(data1, data2)
```



Their relation is exponential.

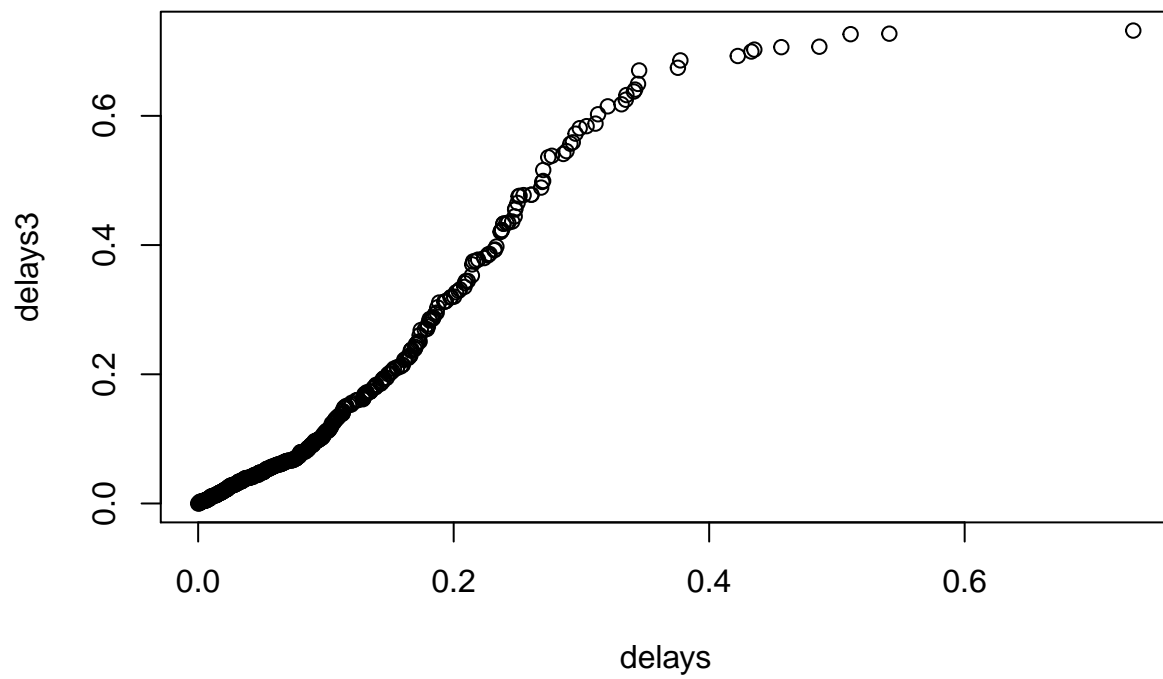
Step 6

```
qqplot(delays, delays2)
```



Step 7

```
qqplot(delays, delays3)
```



Step 8

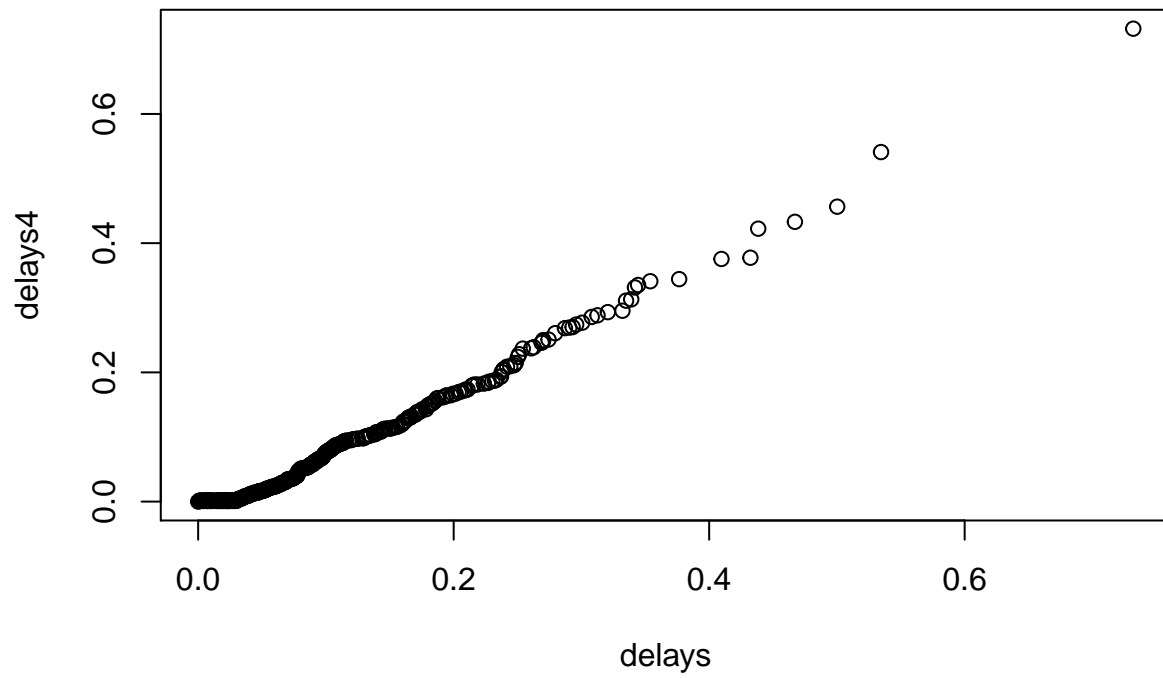
```

Traffic_data_orig <- read.csv("Traffic_data_orig.csv", header=TRUE)
message <- "this is a secret message"
raw <- charToRaw(message)
time = Traffic_data_orig$Time
num = as.integer(rawToBits(raw))
delays = numeric(length(time) - 1)
for (i in (1:(length(time) - 1))) {
  delays[i] = time[i+1] - time[i]
}
m = median(delays)
max = max(delays)
min = min(delays)

index = 1
bitlen = length(raw)*8
encrpt <- numeric(length(raw)*8)
for (i in (0:(length(raw)-1))) {
  for (j in 1:8) {
    if (num[i * 8 + j] == 0) {
      encrpt[index] = sample(delays[which(delays >= min && delays <= m))][1]
    }
    else {
      encrpt[index] = sample(delays[which(delays >= m && delays <= max))][1]
    }
    index = index + 1
    j = j - 1
  }
}

delays4 = delays
for (i in (1:bitlen)) {
  delays4[i] = encrpt[i]
}
qqplot(delays, delays4)

```



Implementation

1