# Question 7 - Part 1

Jin Barai

Let us first write down a general purpose R-function to perform Power Transformations

```
# General Purpose R function to perform power-transformation
powerfun <- function(x, alpha) {
if(sum(x <= 0) > 0) stop("x must be positive")
if (alpha == 0)
  log(x)
else if (alpha > 0) {
  x^alpha
} else -x^alpha
}
```

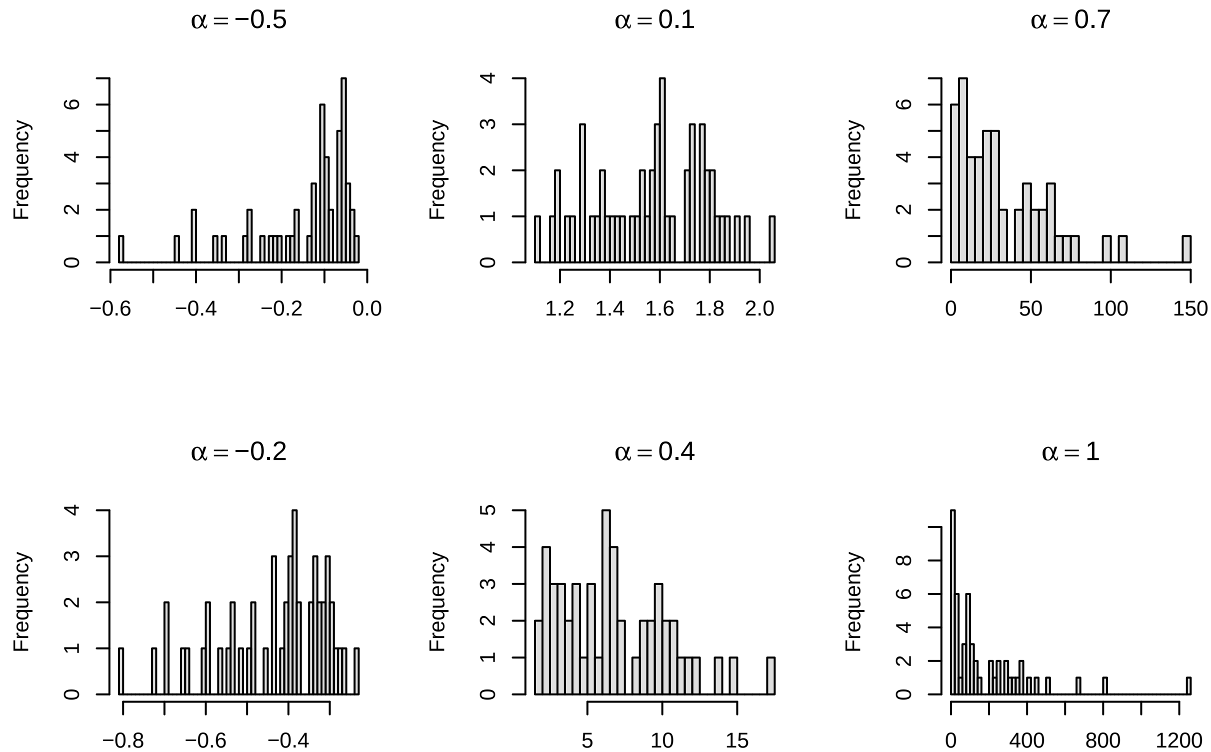
### Bump Rule #1

Let us use the data from Question 6 - "murders.csv".

```
data <- read.csv("murders.csv", header = TRUE)
y <- data$total
```

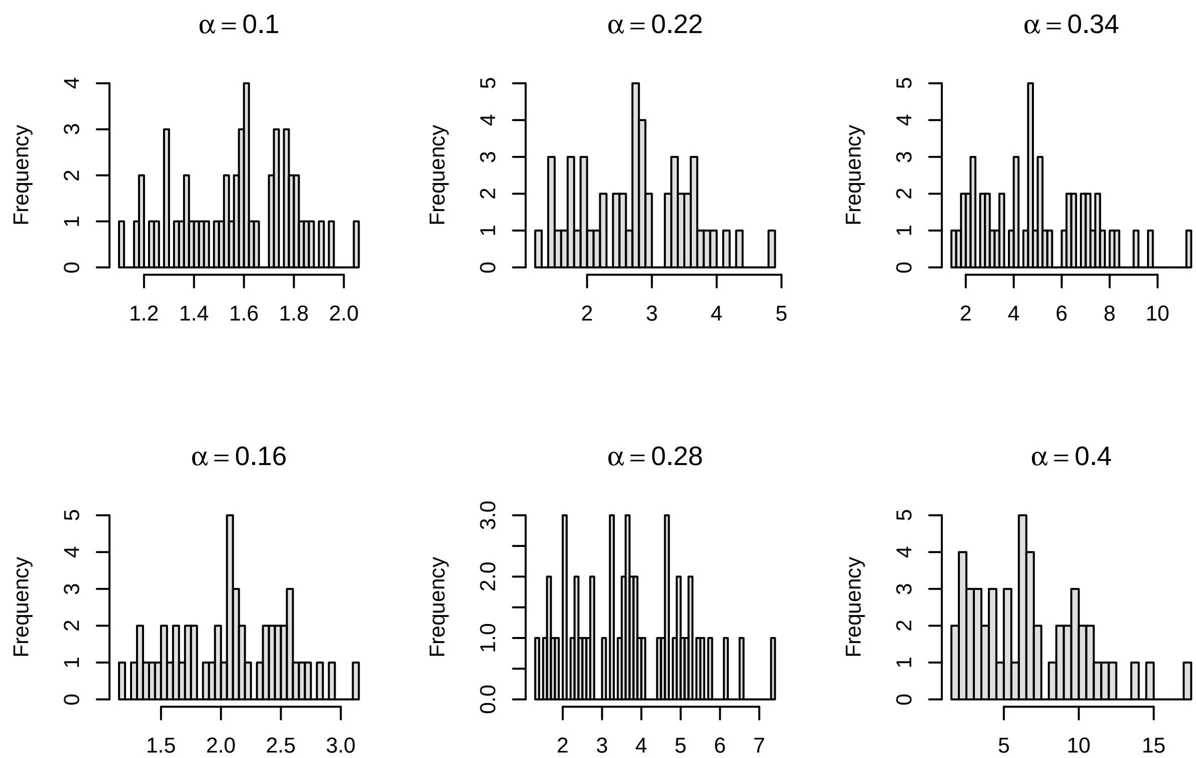
We will vary the power on number of murders.

```
par(mfcol = c(2, 3))
a = seq(-1/2, 1, length.out=6)
for (i in 1:6) {
hist( powerfun( y + 1, a[i]), col=adjustcolor("grey", alpha = 0.5),
main= bquote(alpha == .(a[i])), xlab="", breaks=50 )
}
```




Clearly, alpha seems to lie between 0.1 and 0.4 using bump rule for symmetric histograms.

```
par(mfcol = c(2, 3))
c = seq(0.1, 0.4, length.out=6)
for (i in 1:6) {
hist( powerfun( y + 1, c[i]), col=adjustcolor("grey", alpha = 0.5),
main= bquote(alpha == .(c[i])), xlab="", breaks=50 )
}
```




As you can see from above, the histogram seems most symmetric for

$$\alpha = 0.16$$

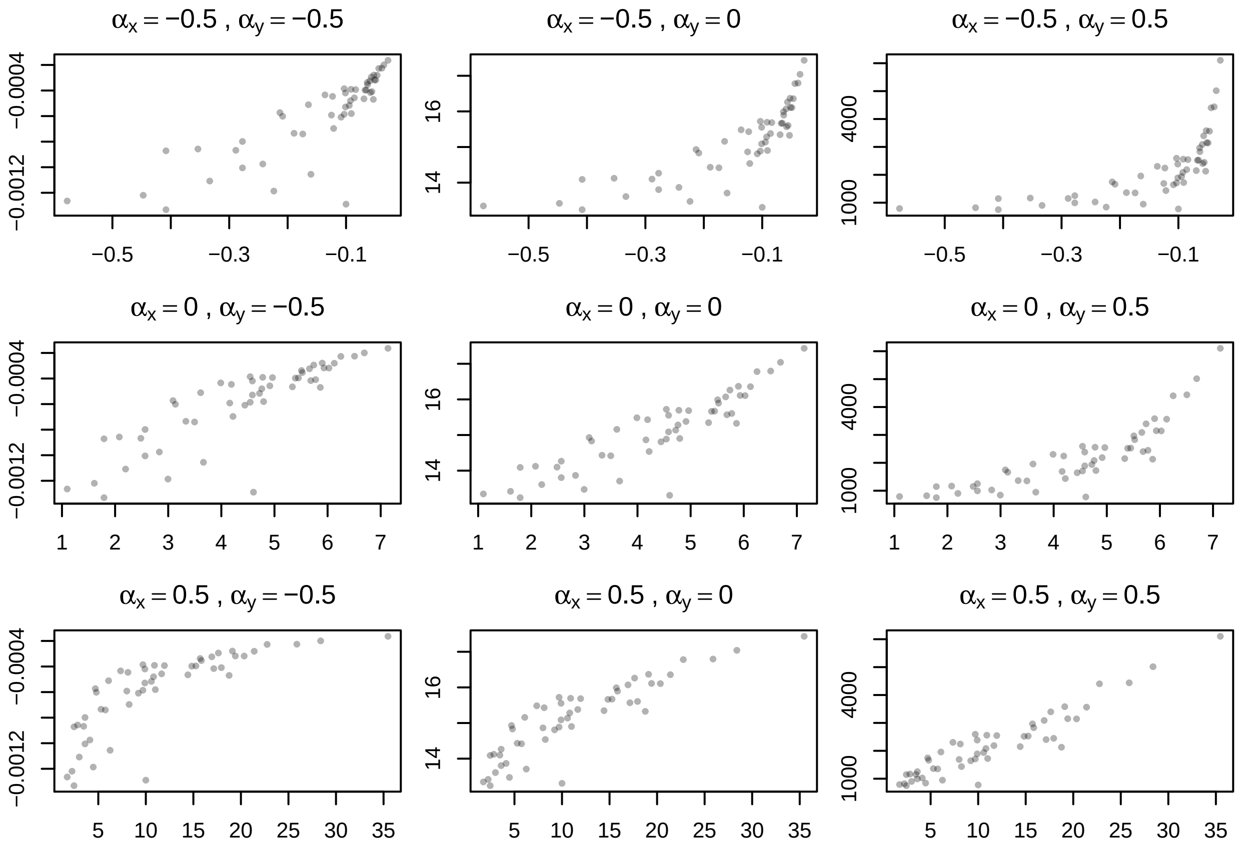
Hence using Bump Rule #1, we can pick our $\alpha$ to be ***0.16***

### Bump Rule #2

We will apply a different power transformation to each variate (x = number of murders in the state, y = population of the state)

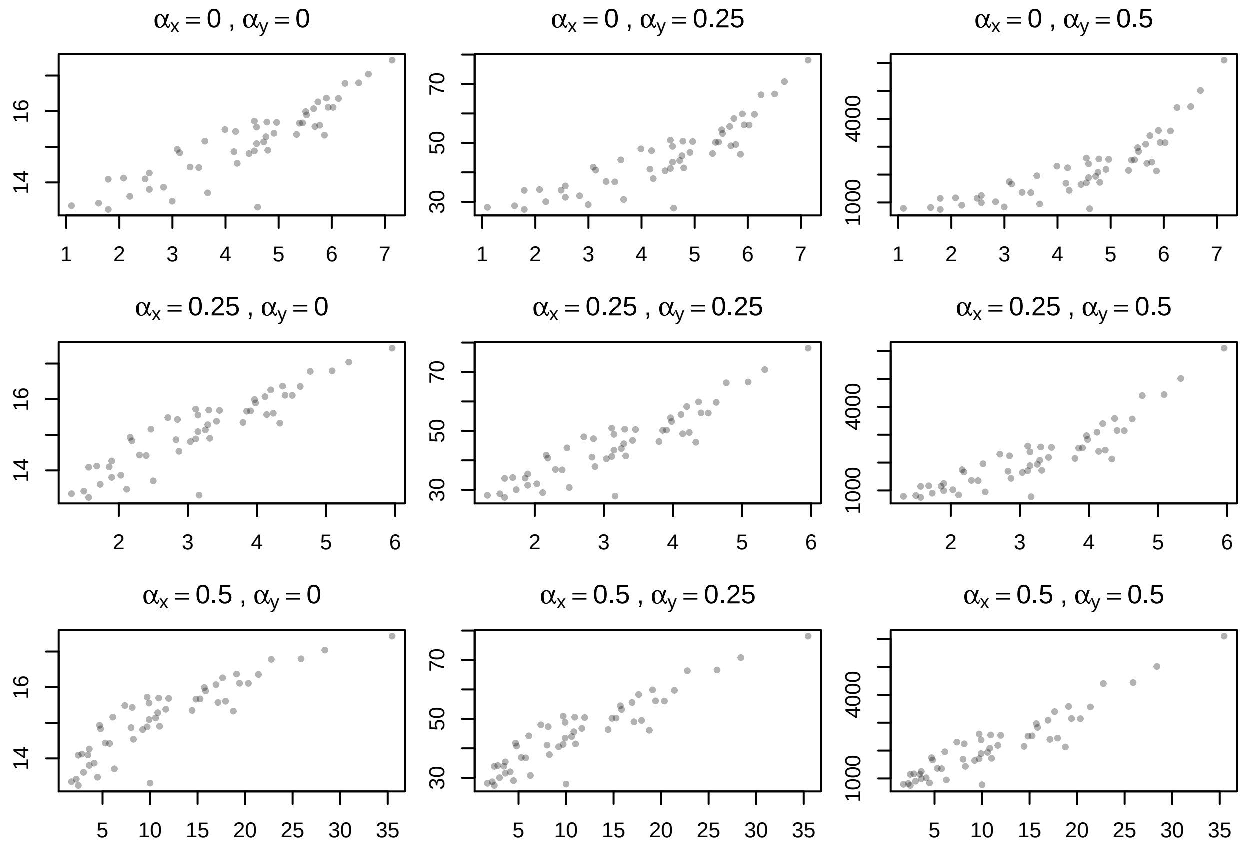
```
par(mfrow=c(3,3), mar=2.5*c(1,1,1,0.1))
a = rep(c(-1/2,0,1/2),each=3)
b = rep(c(-1/2,0,1/2),times=3)
subdata = data[,c(’total’, ’population’)]
subdata = na.omit(subdata)

for (i in 1:9) {
plot( powerfun(subdata$total+1, a[i]), powerfun(subdata$population+1, b[i]), pch = 19, cex=0.5,
col=adjustcolor("black", alpha = 0.3), xlab = "", ylab = "",
main = bquote(alpha[x] == .(a[i]) ~ "," ~ alpha[y] == .(b[i])))
}
```




Clearly it seems to lie between $\alpha$ values of 0,0 and 0.5,0.5. Let's do one more refined search

```
par(mfrow=c(3,3), mar=2.5*c(1,1,1,0.1))
a = rep(c(0,1/4,1/2),each=3)
b = rep(c(0,1/4,1/2),times=3)
for (i in 1:9) {
plot( powerfun(subdata$total+1, a[i]), powerfun(subdata$population+1, b[i]), pch = 19, cex=0.5,
col=adjustcolor("black", alpha = 0.3), xlab = "", ylab = "",
main = bquote(alpha[x] == .(a[i]) ~ "," ~ alpha[y] == .(b[i])))
}
```




From above, using Bump Rule #2, we can pick our $\alpha$ to 0.25,0.25 for x,y respectively.