A4Q2

The dataset contains various mainstream video games with basic information, critic scores and user scores. In this particular question, we will use critic score: ratings received from critics, which ranges from 0-100. The population is critics scores that are greater than 40 (excluding null). This question introduces a statistical terminology: **The Range Rule**, which is used to get a rough estimate of the standard deviation. The formula is simple:

$$\hat{\sigma} = \frac{max(y_1, y_2, ..., y_N) - min(y_1, y_2, ..., y_N)}{4}$$

We will calculate the standard deviation and the range rule estimate of the population and generate $m = 5000$ samples for n=40 and n=70 with sampling with replacement. We then calculate sample bias standard deviation and RMSE. We plot histograms of samples with Gaussian distribution overlaid.

```
game <- read.csv('video_games.csv', header= T)
ratings <- na.omit(game$Critic_Score)
ratings <- ratings[which(ratings>40)]
```

Calculate standard deviation and range rule estimate:

```
y = ratings
N = length(ratings)

sdn <- function(x) {  sd(x)*sqrt( (length(x-1))/length(x) ) }
range_rule <- function(x) {(max(x)-min(x))/4}
c(sdn(y), (max(y)-min(y))/4 )
```

```
## [1] 12.24537 14.25000
```

```
popsd = 12.24537
```

Sampling with replacement (m=5000):

```
sim.hat <- function(pop=NULL, n=NULL, m=5000) {
  N = length(pop);
  s.hat = sapply( 1:m, function(rep) {
                       sam.y = pop[sample(N, n, replace=T)]
                       c(sdn(sam.y), range_rule(sam.y) )
                       } )
  row.names(s.hat) = c("stdev", "range_rule")
  return(s.hat)
}
```

Gaussian function:

```r
overlay <- function(pop=NULL, popsd=NULL) {
    # pop is the population of values
    # pop.val is the value we want to estimate.

    abline(v=popsd, lty=2, col=4, lwd=1.5)
    aseq = seq(min(pop), max(pop), length.out=100)
    lines(aseq, dnorm(aseq, mean=mean(pop), sd=sd(pop)), col=2)
}
```

generate a table of sample bias, standard deviation and RMSE:

```r
sampling <- function(estimate=NULL, popsd=NULL){
   # estimate is a [(p >1) x m  ]  matrix

   bias = apply(estimate, 1, mean) - popsd
   var  = apply(estimate, 1, var)
   MSE  = bias^2 + var

   tab = rbind( Sampling.bias = bias, Sampling.stdev= sqrt(var), sqrt.MSE=sqrt(MSE) )
   tab
}
```

when n=40:

```r
set.seed(20775633)
s40  <- sim.hat(ratings, 40)
MSE40 <- sampling(s40, popsd)

round(MSE40,2)
```

```
##                stdev range_rule
## Sampling.bias  -0.08      -0.32
## Sampling.stdev  1.16       1.02
## sqrt.MSE        1.16       1.07
```

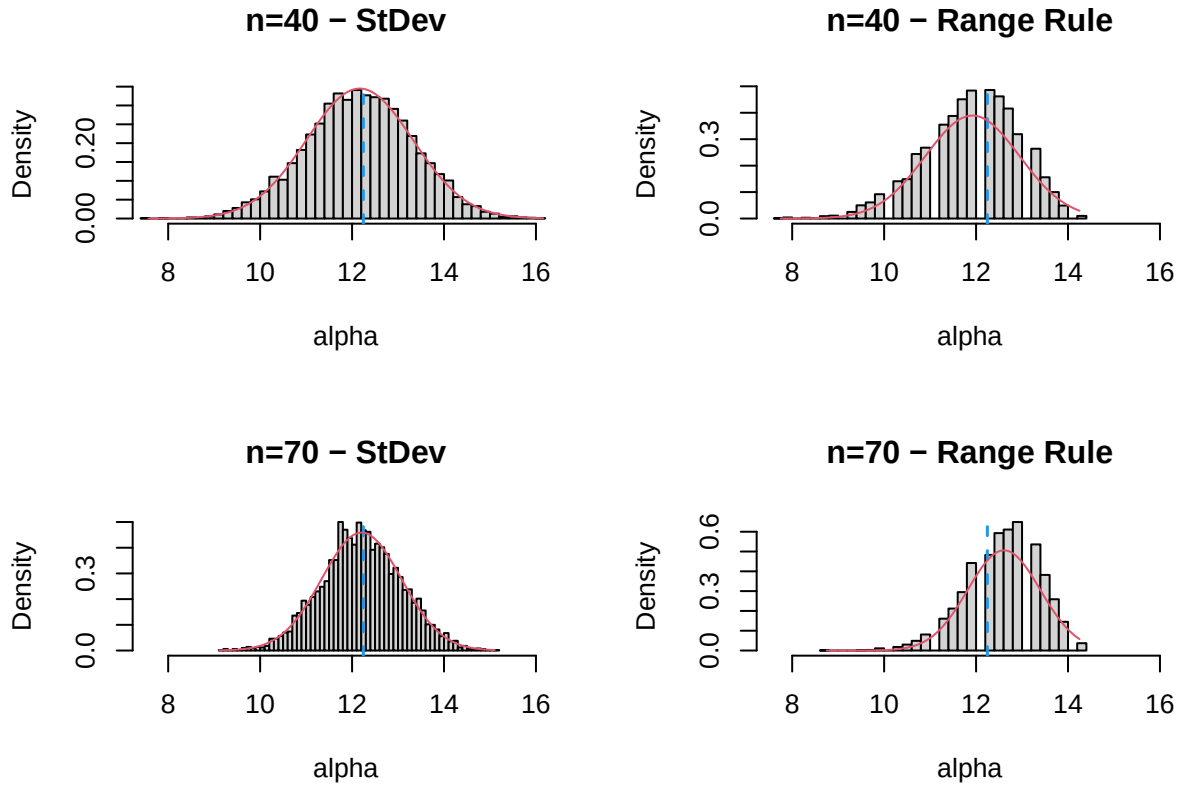
When n=70:

```r
set.seed(20775633)
s70  <- sim.hat(ratings, 70)
MSE70 <- sampling(s70, popsd)

round(MSE70,2)
```

```
##                stdev range_rule
## Sampling.bias  -0.04       0.36
## Sampling.stdev  0.87       0.79
## sqrt.MSE        0.87       0.87
```

Histograms of both sample sizes:

```
par(mfrow=c(2,2),oma=c(0,0,0,0))
hist( s40[1,], breaks="FD", prob=TRUE,
      xlab="alpha", main="n=40 - StDev", xlim=range(s40))
overlay(s40[1,], sdn(y))
hist( s40[2,], breaks="FD", prob=TRUE,
      xlab="alpha", main="n=40 - Range Rule", xlim=range(s40))
overlay(s40[2,], sdn(y))

hist( s70[1,], breaks="FD", prob=TRUE,
      xlab="alpha", main="n=70 - StDev", xlim=range(s40))
overlay(s70[1,], sdn(y))
hist( s70[2,], breaks="FD", prob=TRUE,
      xlab="alpha", main="n=70 - Range Rule", xlim=range(s40))
overlay(s70[2,], sdn(y))
```



Comment: For Stdev, the standard derivations of samples group around the true standard derivation, and they match nearly as perfect as the gaussian distribution. For range rule, there are some gaps in the histogram as expected as we are calculating the difference in range. Both range rule histograms has a long left tail and the peak is over the guassian distribution. With sample size increases, the distribution becomes more skewed. We can also see in the table. With n=40, the sampling bias is close, but when the sample size increases, range rule sampling bias and RMSE increase drastically. Therefore, range rule can't be an accurate estimator of standard deviation. This proves that it can only get a rough value.