Computing Assignment 1

Xinyi Lin

10/6/2019

Problem 1

```r
get_pr = function(n, s, a=0.5, b=1.5, d = delta, N=1000){
  pe = rbeta(N, a+s, b+n-s)
  ps = rbeta(N, 25, 75)
  pr = sum(pe > (ps+delta))/N
  return(pr)
}
```

```r
set.seed(123)
delta = 0.15
alpha = 0.05
# stage 1
n1 = 20
s1 = 5
# stage 2
n2 = 71
s2 = 23
while (get_pr(n1, s1)>alpha | get_pr(n2,s2)>alpha) {
  alpha = alpha + 0.05
}
alpha
```

```
## [1] 0.15
```

Test whether α, δ mimics the two-stage adaptive designs.

```r
set.seed(123)
# stage 1
get_pr(20,6)
```

```
## [1] 0.162
```

```r
get_pr(20,5)
```

```
## [1] 0.063
```

```r
# stage 2
get_pr(71,24)
```

```
## [1] 0.165
```

```r
get_pr(71,23)
```

```
## [1] 0.126
```

Assuming $\delta = 0.15$ and using $S_{20} = 5, S_{71} = 23$ as stopping boundary, I can get the smallest α by updating α until δ, α mimics the two-stage adaptive design's `no-go` criterion. After testing, $\delta = 0.15, \alpha = 0.15$ is the final restult.

Problem 2

Question a

```r
a = 0.5
b = 1.5
delta = 0.15
alpha = 0.15
N=1000
# stage 1
n1 = 20
s1 = 2
set.seed(1)
while (get_pr(n=n1, s=s1)<=alpha) {
  s1 = s1+1
}
s1
```

```
## [1] 6
```

```r
# s20<=5 -> no go
# stage 2
n2 = 40
s2 = 6
while (get_pr(n=n2, s=s2)<=alpha) {
  s2 = s2+1
}
s2
```

```
## [1] 13
```

```r
# s40<=12 -> no go
# stage 3
n3 = 60
s3 = 13
while (get_pr(n=n3, s=s3)<=alpha) {
  s3 = s3+1
}
s3
```

```
## [1] 20
```

```r
# s60<=19 -> no go
# final
n4 = 71
s4 = 20
while (get_pr(n=n4, s=s4)<=alpha) {
  s4 = s4+1
}
s4
```

```
## [1] 24
```

```r
# s71<=23 -> futility
```

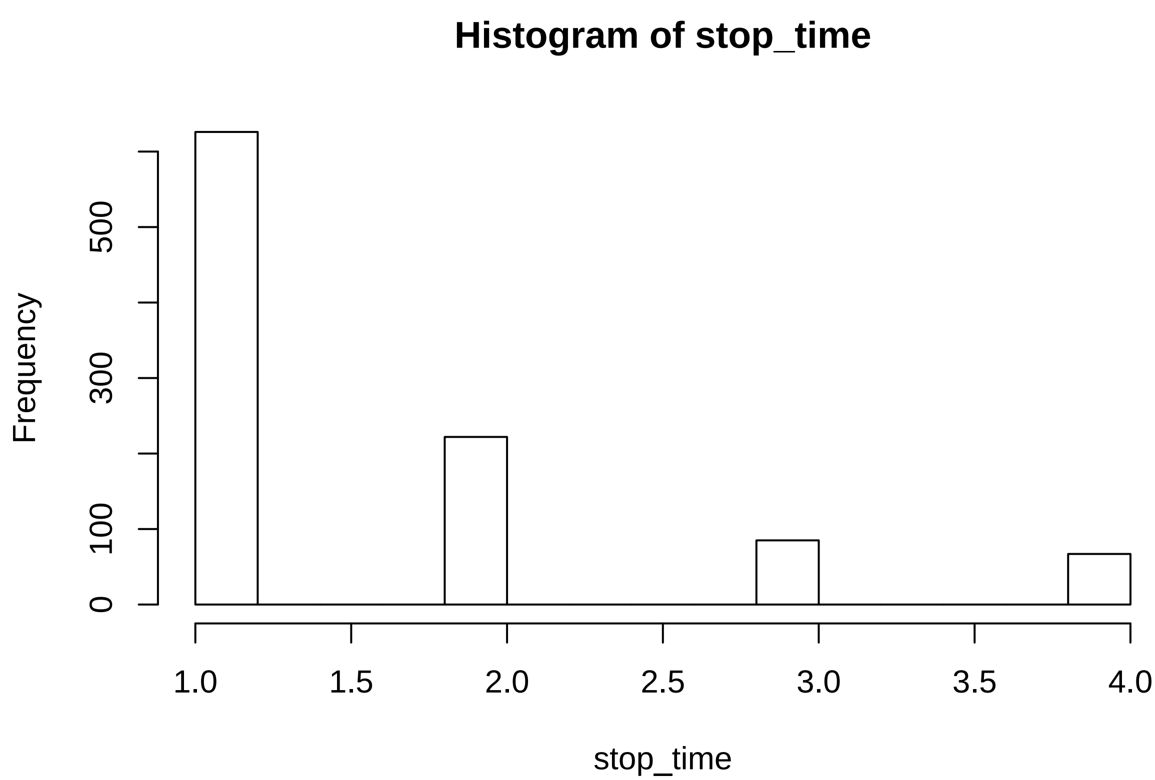
So the stopping boundary at each interim are $s_{20} \leq 5, s_{40} \leq 12, s_{60} \leq 19$.

Question b

```r
N=1000
stop1 = 5
stop2 = 12
stop3 = 19
stop4 = 23
stop_time = rep(NA, N)
futile = rep(NA, N)
pE = 0.25
set.seed(1)
for (i in 1:N){
  s1 = rbinom(1,20,pE)
  if (s1<=stop1) {
    stop_time[i]=1
    futile[i] = T
    } else {
      s2 = rbinom(1,20,pE)
      if (s1+s2<=stop2) {
        stop_time[i]=2
        futile[i] = T
        }else{
          s3 = rbinom(1,20,pE)
          if(s1+s2+s3<=stop3){
            stop_time[i]=3
            futile[i]=T
          }else{
              stop_time[i]=4
              s4 = rbinom(1,11,pE)
              if(s1+s2+s3+s4<=stop4){futile[i]=T}
              else{futile[i]=F}}
    }
  }
}
pro_b = length(futile[futile==T])/N
pro_b
```

```
## [1] 0.963
```

```r
hist(stop_time)
```



The probability of declaring futile is 0.963. The distribution is shown above.

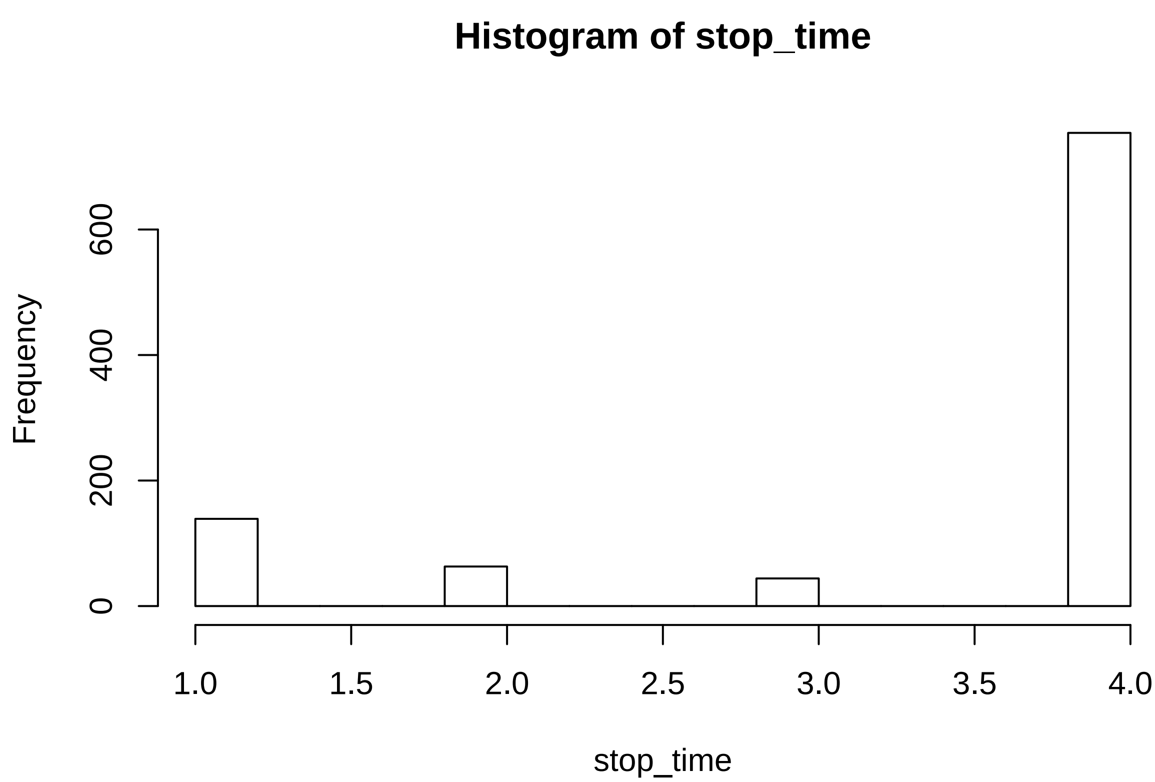
Question c

```r
N=1000
stop1 = 5
stop2 = 12
stop3 = 19
stop4 = 23
stop_time = rep(NA, N)
futile = rep(NA, N)
pE = 0.4
set.seed(1)
for (i in 1:N){
  s1 = rbinom(1,20,pE)
  if (s1<=stop1) {
    stop_time[i]=1
    futile[i] = T
    } else {
      s2 = rbinom(1,20,pE)
```

```r
      if (s1+s2<=stop2) {
        stop_time[i]=2
        futile[i] = T
        }else{
          s3 = rbinom(1,20,pE)
          if(s1+s2+s3<=stop3){
            stop_time[i]=3
            futile[i]=T
          }else{
              stop_time[i]=4
              s4 = rbinom(1,11,pE)
              if(s1+s2+s3+s4<=stop4){futile[i]=T}
              else{futile[i]=F}}
    }
  }
}
pro_c = length(futile[futile==T])/N
pro_c
```

```
## [1] 0.264
```

```r
hist(stop_time)
```



The probability of declaring futile is 0.264. The distribution is shown above.

Problem 3

Question a

```r
new_pr = function(n, s, a=0.5, b=1.5){
  pe = rbeta(1, a+s, b+n-s)
  pr = pbinom(25, 71, pe, lower.tail = F)
  return(pr)
}
```

```r
# stage 1
n1 = 20
s1 = 2
alpha = 0.9
set.seed(1)
n1 = 20
n2 = 3
while (new_pr(n=n1, s=s1)<alpha) {
  s1 = s1+1
}
s1
```

```
## [1] 8
```

```r
# s20<=7 -> no go
# stage 2
n2 = 40
s2 = 6
while (new_pr(n=n2, s=s2)<alpha) {
  s2 = s2+1
}
s2
```

```
## [1] 15
```

```r
# s40<=14 -> no go
# stage 3
n3 = 60
s3 = 13
while (new_pr(n=n3, s=s3)<alpha) {
  s3 = s3+1
}
s3
```

```
## [1] 24
```

```r
# s60<=23 -> no go
# final
n4 = 71
s4 = 20
while (new_pr(n=n4, s=s4)<alpha) {
```

```r
  s4 = s4+1
}
s4
```

```
## [1] 26
```

```r
# s71<=25 -> futility
```

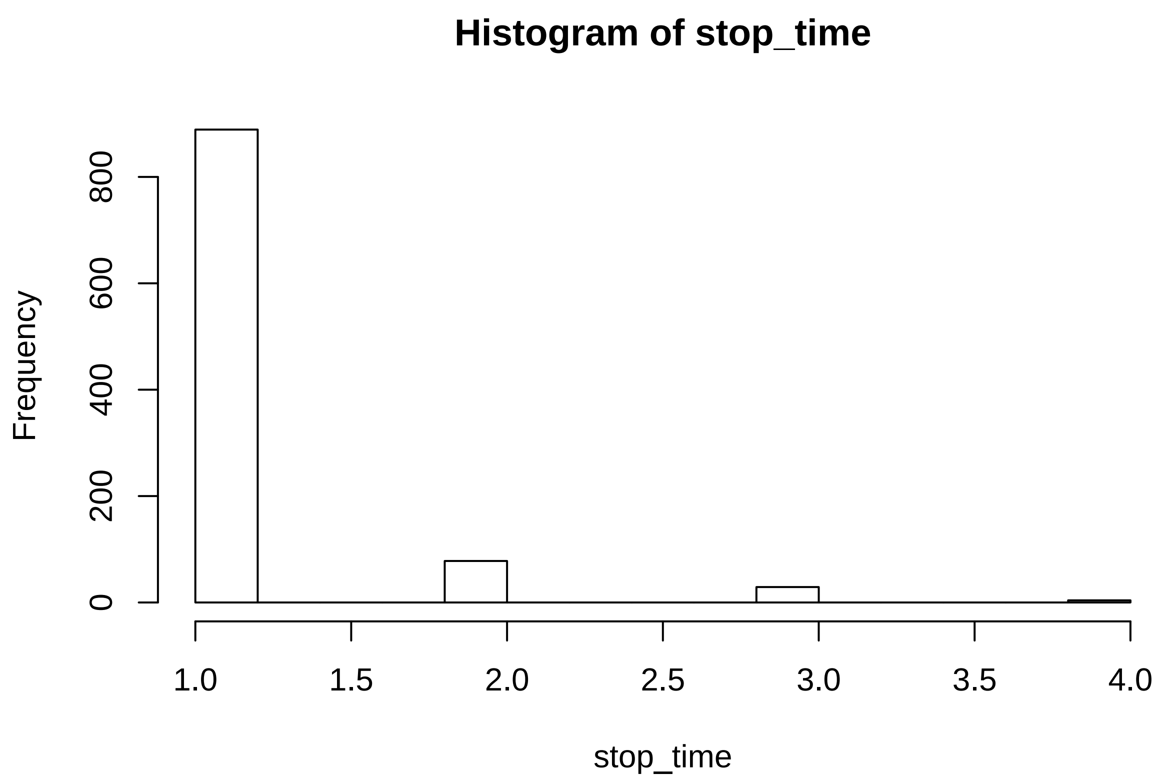
So the stopping boundary at each interim are $s_{20} \leq 7, s_{40} \leq 14, s_{60} \leq 23$.

Question b

```r
N=1000
stop1 = 7
stop2 = 14
stop3 = 23
stop4 = 25
stop_time = rep(NA, N)
futile = rep(NA, N)
pE = 0.25
set.seed(1)
for (i in 1:N){
  s1 = rbinom(1,20,pE)
  if (s1<=stop1) {
    stop_time[i]=1
    futile[i] = T
    } else {
      s2 = rbinom(1,20,pE)
      if (s1+s2<=stop2) {
        stop_time[i]=2
        futile[i] = T
        }else{
          s3 = rbinom(1,20,pE)
          if(s1+s2+s3<=stop3){
            stop_time[i]=3
            futile[i]=T
          }else{
              stop_time[i]=4
              s4 = rbinom(1,11,pE)
              if(s1+s2+s3+s4<=stop4){futile[i]=T}
              else{futile[i]=F}}
    }
  }
}
pro_b = length(futile[futile==T])/N
pro_b
```

```
## [1] 0.996
```

```r
hist(stop_time)
```



The probability of declaring futile is 0.996. The distribution is shown above.

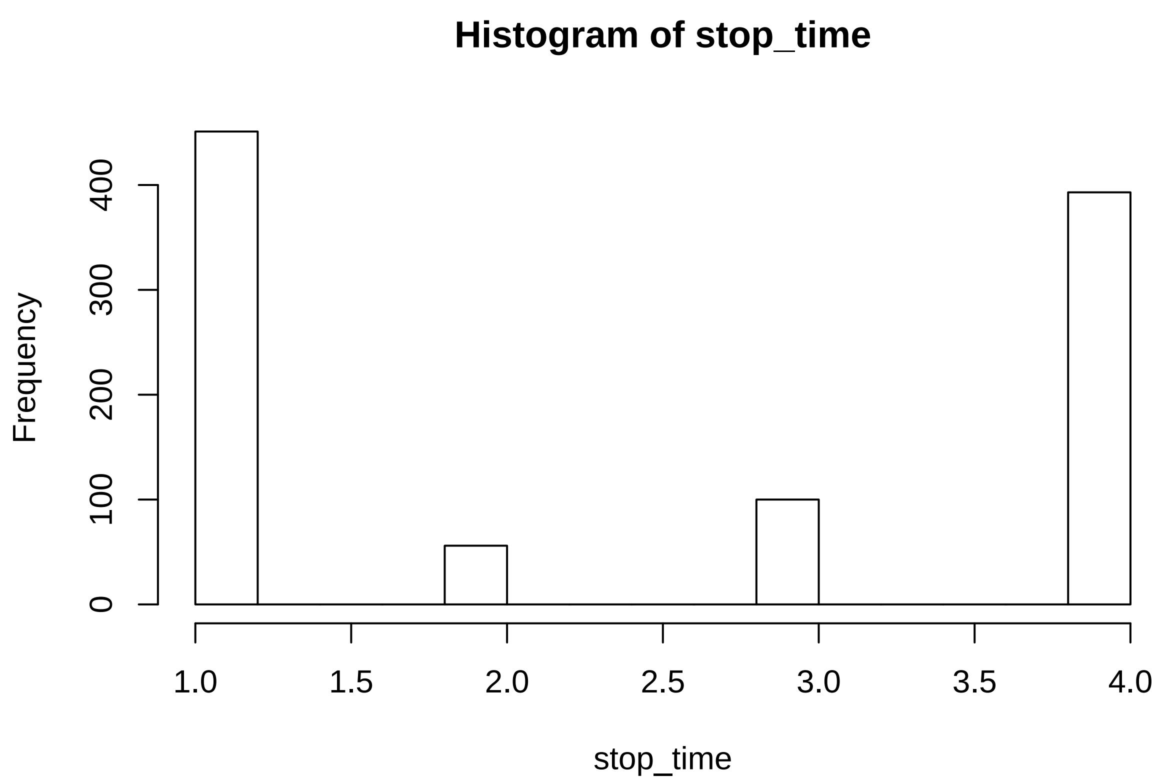
Question c

```r
N=1000
stop1 = 7
stop2 = 14
stop3 = 23
stop4 = 25
stop_time = rep(NA, N)
futile = rep(NA, N)
pE = 0.4
set.seed(1)
for (i in 1:N){
  s1 = rbinom(1,20,pE)
  if (s1<=stop1) {
    stop_time[i]=1
    futile[i] = T
    } else {
      s2 = rbinom(1,20,pE)
      if (s1+s2<=stop2) {
        stop_time[i]=2
```

```
        futile[i] = T
        }else{
          s3 = rbinom(1,20,pE)
          if(s1+s2+s3<=stop3){
            stop_time[i]=3
            futile[i]=T
          }else{
              stop_time[i]=4
              s4 = rbinom(1,11,pE)
              if(s1+s2+s3+s4<=stop4){futile[i]=T}
              else{futile[i]=F}}
    }
  }
}
pro_c = length(futile[futile==T])/N
pro_c
```

```
## [1] 0.608
```

```
hist(stop_time)
```



The probability of declaring futile is 0.608. The distribution is shown above.