模擬書櫃問題: 一個書櫃上有 10 本各不同的書，每本書與使用者的距離為 1~10 個單位，使用者每次會拿取一本書，爾後馬上放回，假設這 10 本書拿取的頻率未知，如何在一邊拿取，一邊更新這 10 本書的相對位置下，達到平均每次拿書的距離為最短。

提供三個 Policy：

A. 若這次拿到的書為 i 書，則把 i 書往前移一個，每次都如此操作
B. 若這次拿到的書為 i 書，則把 i 書移到最前面一個，每次都如此操作
C. 可以給每本書一個計數器，每拿一次書則該計數器加一；計數器滿五次則書往前移一格並且計數器歸零。因此我突發奇想，既然計數器一定會占用到 memery 了，那就不要歸零，依然維持「每拿一次書則該計數器加一」的原則(根據大數法則，每本書對應的計數器/總共拿書次數會機率收斂至拿到那本書的機率)。我們就把書由近到遠依序以計數器頻率多寡來排列。
   並利用 Batch means 方法將平均每次拿書距離估計出來。

假設十本書編號 1~10，並設定第 i 本書拿到的機率 p(i)如下：

P(1)=0.06
P(2) = 0.01
P(3) = 0.12
P(4) = 0.14
P(5) = 0.08
P(6) = 0.18
P(7) = 0.04
P(8) = 0.10
P(9) = 0.22
P(10) = 0.05

以下使用 Two-Stage Procedure 方法來做比較，最後每個 Policy 產出的統計量，其中 h=2.786 為查表 n_0=40、p^*=.95、k=3。

Policy A：5.22
Policy B：3.86
Policy C：3.65

此方法也顯示，Policy C 優於其他兩個 Policy。

下附程式碼:

Function and some options

```
options(digits = 16)
g=as.integer(Sys.time()) #initial garbled for generating event
Ge<-function(g){
  g<<-(g*16807)%%(2^55-1)
  return(g/(2^55-1))
}
which.book<-function(e){ #which book I take next time
if(e<.06){
    return(x=1)
  }else if(e<.07){
```

```
    return(x=2)
    }else if(e<.19){
      return(x=3)
      }else if(e<.33){
        return(x=4)
        }else if(e<.41){
          return(x=5)
          }else if(e<.59){
            return(x=6)
            }else if(e<.63){
              return(x=7)
              }else if(e<.73){
                return(x=8)
                }else if(e<.95){
                  return(x=9)
                  }else{
                    return(x=10)
                  }
}
```

```
######Policy A######
S=c(1:10) #state space
hand.length.m.A=0 #stat. counter mean
hand.length.v.A=0 #stat. counter second moment mean
m.A=0
for(i in 1:10000){
  x=which.book(Ge(g)) #eventlist
  S[c(1,which(S==x))]=S[c(which(S==x),1)]
}
for(i in 1:200){
  x=which.book(Ge(g)) #eventlist
  m.A=m.A+which(S==x)
  if(i%%5==0){
    hand.length.m.A=hand.length.m.A+(m.A/5)
    hand.length.v.A=hand.length.v.A+(m.A/5)^2
    m.A=0
  }
  S[c(1,which(S==x))]=S[c(which(S==x),1)]
}
b40=hand.length.m.A*5/i
v40=(hand.length.v.A-(i/5)*(hand.length.m.A*5/i)^2)/(i/5-1)
N.A=max(41,ceiling(2.786^2*v40/.01))
```

```
hand.length.m.A.second=0
for(i in 1:(N.A*5-200)){
  x=which.book(Ge(g)) #eventlist
  m.A=m.A+which(S==x)
  if(i%%5==0){
    hand.length.m.A.second=hand.length.m.A.second+(m.A/5)
    m.A=0
  }
  S[c(1,which(S==x))]=S[c(which(S==x),1)]
}
W=40*(1+sqrt(1-N.A*(1-(N.A-40)*.01/(2.786^2*v40))/40))/N.A
W*hand.length.m.A/200+(1-W)*hand.length.m.A.second/(N.A-40)
######Policy A######
######Policy B######
S=c(1:10) #state space
hand.length.m.B=0 #stat. counter mean
hand.length.v.B=0 #stat. counter second moment mean
m.B=0 #local
for(i in 1:10000){
  x=which.book(Ge(g)) #eventlist
  if(which(S==x)!=1){
  S[c((which(S==x)-1),which(S==x))]=S[c(which(S==x),(which(S==x)-1))]
  }
}
for(i in 1:200){
  x=which.book(Ge(g)) #eventlist
  m.B=m.B+which(S==x)
  if(i%%5==0){
    hand.length.m.B=hand.length.m.B+(m.B/5)
    hand.length.v.B=hand.length.v.B+(m.B/5)^2
    m.B=0
  }
  if(which(S==x)!=1){
    S[c((which(S==x)-1),which(S==x))]=S[c(which(S==x),(which(S==x)-1))]
  }
}
b40=hand.length.m.B*5/i
v40=(hand.length.v.B-(i/5)*(hand.length.m.B*5/i)^2)/(i/5-1)
N.B=max(41,ceiling(2.786^2*v40/.01))
hand.length.m.B.second=0
for(i in 1:(N.B*5-200)){
  x=which.book(Ge(g)) #eventlist
```

```
  m.B=m.B+which(S==x)
  if(i%%5==0){
    hand.length.m.B.second=hand.length.m.B.second+(m.B/5)
    m.B=0
  }
  if(which(S==x)!=1){
    S[c((which(S==x)-1),which(S==x))]=S[c(which(S==x),(which(S==x)-1))]
  }
}
W=40*(1+sqrt(1-N.B*(1-(N.B-40)*.01/(2.786^2*v40))/40))/N.B
W*hand.length.m.B/200+(1-W)*hand.length.m.B.second/(N.B-40)

######Policy B######
######Policy C######
S=c(1:10) #state space
counter=numeric(10)
hand.length.m.C=0 #stat. counter mean
hand.length.v.C=0 #stat. counter second moment mean
m.C=0 #local
for(i in 1:10000){
  x=which.book(Ge(g)) #eventlist
  counter[x]=counter[x]+1
  S=order(counter)[10:1]
}
for(i in 1:200){
  x=which.book(Ge(g)) #eventlist
  m.C=m.C+which(S==x)
  if(i%%5==0){
    hand.length.m.C=hand.length.m.C+(m.C/5)
    hand.length.v.C=hand.length.v.C+(m.C/5)^2
    m.C=0
  }
  counter[x]=counter[x]+1
  S=order(counter)[10:1]

}
b40=hand.length.m.C*5/i
v40=(hand.length.v.C-(i/5)*(hand.length.m.C*5/i)^2)/(i/5-1)
N.C=max(41,ceiling(2.786^2*v40/.01))
hand.length.m.C.second=0
for(i in 1:(N.C*5-200)){
  x=which.book(Ge(g)) #eventlist
```

```
  m.C=m.C+which(S==x)
  if(i%%5==0){
    hand.length.m.C.second=hand.length.m.C.second+(m.C/5)
    m.C=0
  }
  counter[x]=counter[x]+1
  S=order(counter)[10:1]
}
W=40*(1+sqrt(1-N.C*(1-(N.C-40)*.01/(2.786^2*v40))/40))/N.C
W*hand.length.m.C/200+(1-W)*hand.length.m.C.second/(N.C-40)
######Policy C######
```