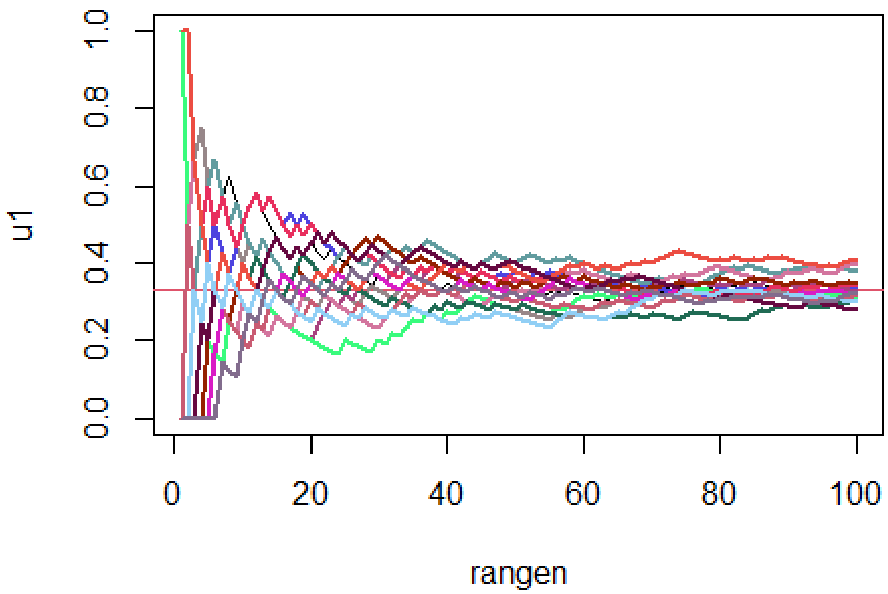
SuchetaWS10

Sucheta

25/11/2021

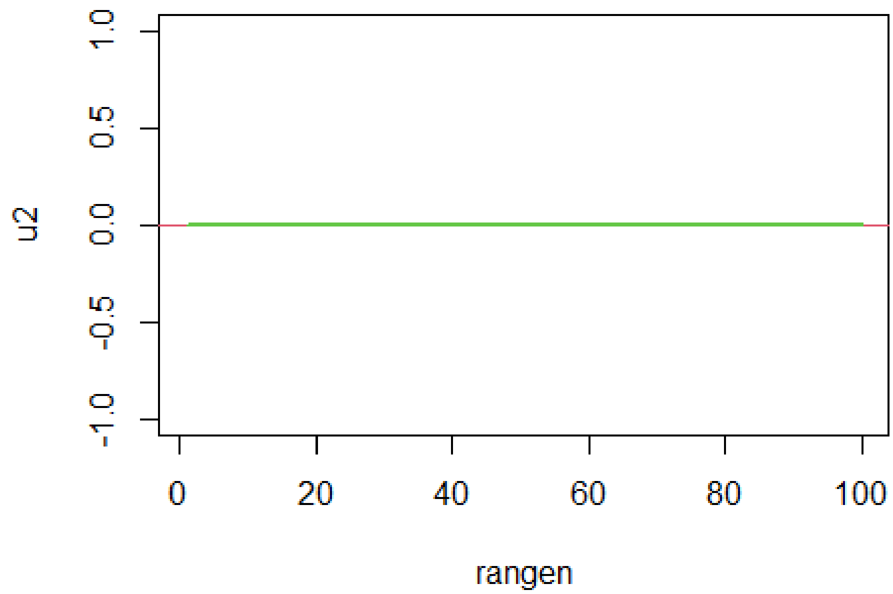
```
#Solution 1.a

library(Rlab)

## Rlab 2.15.1 attached.

##
## Attaching package: 'Rlab'

## The following objects are masked from 'package:stats':
##
##     dexp, dgamma, dweibull, pexp, pgamma, pweibull, qexp, qgamma,
##     qweibull, rexp, rgamma, rweibull

## The following object is masked from 'package:datasets':
##
##     precip

x_bern = rbern(100,1/3) #generates 100 iid Bernoulli(1/3)

#Solution 1.b

runningmean = function (x,N){  #function to calculate the mean
  y = sample(x,N, replace=TRUE)
  c = cumsum(y)
  n = 1:N
  c/n
}

#Sample with n = 100, p =1/3
n=100
x_bern = rbern(n,1/3) #generates n iid Bernoulli(1/3)
u1 = runningmean(x_bern, n)
rangen=1:n
plot(u1~rangen, type ='l',ylim = c(0,1))
abline(h=1/3, col=2)


#solution 1.c

replicate(15,lines(runningmean(x_bern, 100)~rangen, type="l",lwd=2,
col=rgb(runif(3),runif(3),runif(3))))
```




```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
##
## [[9]]
## NULL
##
```

```
## [[10]]
## NULL
##
## [[11]]
## NULL
##
## [[12]]
## NULL
##
## [[13]]
## NULL
##
## [[14]]
## NULL
##
## [[15]]
## NULL
```

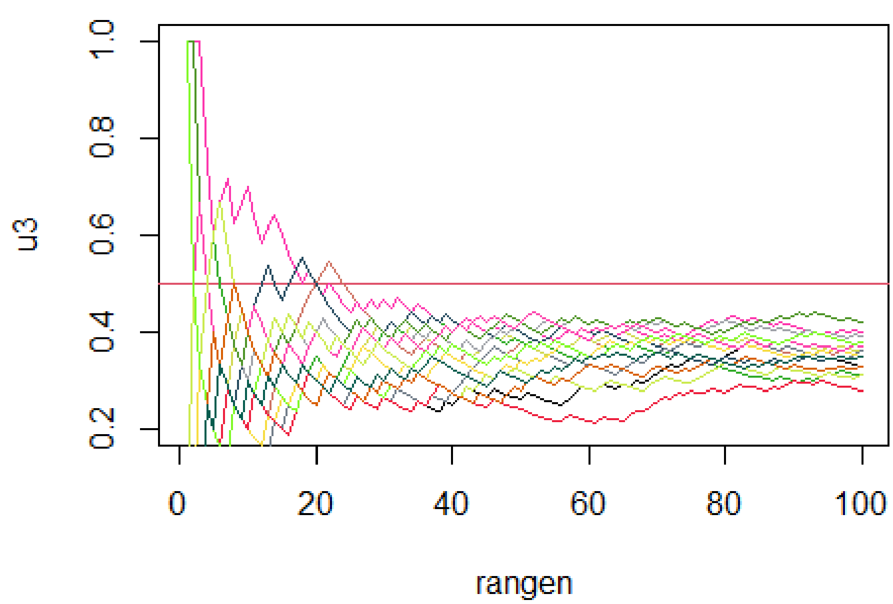
```r
#solution 1.d

#Sample with n = 100, p =0.001
x_bern2 = rbern(n,0.001)  #generates n iid Bernoulli(0.001)
u2 = runningmean(x_bern2, n)
rangen=1:n
plot(u2~rangen, type ='l')
abline(h=0.001, col=2)

replicate(15,lines(runningmean(x_bern2, 100)~rangen, type="l",lwd=2,
col=rgb(runif(3),runif(3),runif(3))))
```




```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
##
## [[9]]
## NULL
##
```

```
## [[10]]
## NULL
##
## [[11]]
## NULL
##
## [[12]]
## NULL
##
## [[13]]
## NULL
##
## [[14]]
## NULL
##
## [[15]]
## NULL
```

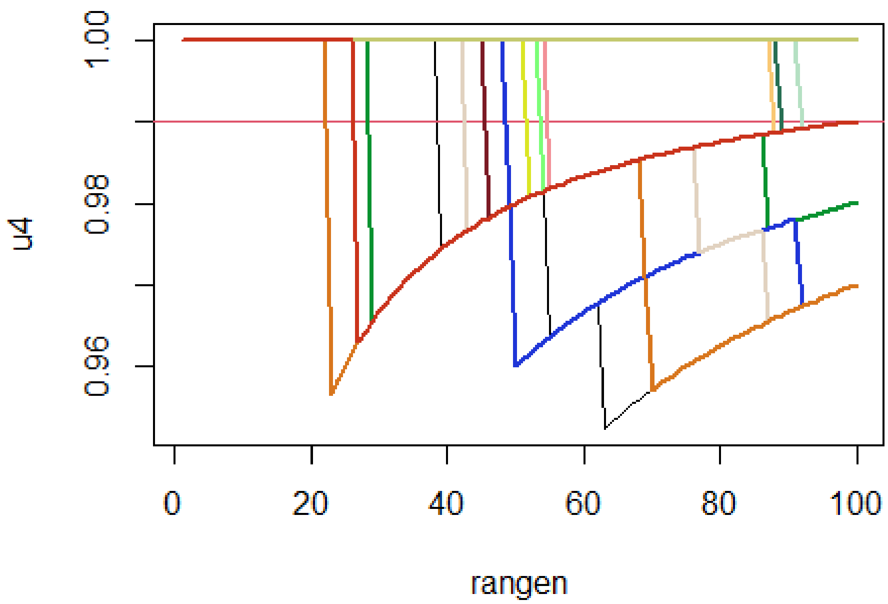
```r
#Sample with n = 100, p = 0.5
x_bern3 = rbern(n,0.5)    #generates n iid Bernoulli(0.5)
u3 = runningmean(x_bern3, n)
rangen=1:n
plot(u3~rangen, type ='l')
abline(h=0.5, col=2)

replicate(15,lines(runningmean(x_bern3, 100)~rangen, type="l",
col=rgb(runif(3),runif(3),runif(3))))
```



```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
##
## [[9]]
## NULL
##
## [[10]]
## NULL
##
## [[11]]
## NULL
##
## [[12]]
## NULL
##
## [[13]]
## NULL
##
## [[14]]
## NULL
##
## [[15]]
## NULL
```

```
#Sample with n = 100, p =0.99
x_bern4 = rbern(n,0.99)   #generates n iid Bernoulli(0.99)
u4 = runningmean(x_bern4, n)
rangen=1:n
plot(u4~rangen, type ='l')
```

```r
abline(h=0.99, col=2)

replicate(15,lines(runningmean(x_bern4, 100)~rangen, type="l",lwd=2,
col=rgb(runif(3),runif(3),runif(3))))
```




```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
```

```
## NULL
##
## [[9]]
## NULL
##
## [[10]]
## NULL
##
## [[11]]
## NULL
##
## [[12]]
## NULL
##
## [[13]]
## NULL
##
## [[14]]
## NULL
##
## [[15]]
## NULL
```

```
#Solution 2.a
sum_fun = function(a,b,f,n){
  x_unif = runif(n,min=a,max=b) #uniformly distributed random variables
  f_x = sapply(x_unif,f) #apply the function to each random variable
  (b-a)*sum(f_x)/n  #calculate the left hand side of the given equation
}

#Solution 2.b

a=0 #lower bound
b=7 #upper bound
n=400
f = function(x){  #given function
  (16+sin(x))/(x^2+4)
}

sum_f = sum_fun(a,b,f,n)
print(paste("The approximate value of the integration is :",sum_f))
```

```
## [1] "The approximate value of the integration is : 10.8776963698781"
```

```
#Solution 2.c

sum_f1=rep(NA,100)
for (i in 1:100){  #repeat 100 times
  x_unif = runif(n,min=a,max=b)
  f_x = sapply(x_unif,f)
```

```
  sum_f1[i] = (b-a)*sum(f_x)/n
}
mean_sum = mean(sum_f1)
print(paste("The approximate value comes as: ",mean_sum))

## [1] "The approximate value comes as:  10.6081065357171"

#Solution 2.d

int_f= integrate(f,a,b)
error= mean_sum - int_f$value
print(paste("The value of the integral is: ",int_f$value))

## [1] "The value of the integral is:  10.5820421004761"

print(paste("The integral and the sum differ by: ",error))

## [1] "The integral and the sum differ by:  0.0260644352410484"

#We can see that the two answers are very close to each other.


#solution 3.a

x_pois = rpois(500,10) #generating samples from distribution function
print(x_pois)

##   [1] 12 14  8 10 10  7 15 11 16 11 11  9  8 13  4 13 11 11 10  7 13 11 12
10  9
##  [26] 12 13 11 12 12 10  9 12  3 11  5  6  3  5 15  7  8  8 10 11 10  7 11
7  5
##  [51]  9  8  9  8 11 13 11  5 10  9  9 14 13  6 10  7 11 10  8  6  8 11 14
11 10
##  [76]  7 13  6 13 11 13  8  9 11 13  4  8 13 13 10 11 12 12  9 11  4  8  7
14 10
## [101] 14  7  8  5 13 11 14  7  9 15 11  7 10  9 10  9 10 12  9 10 13 13 12
12  6
## [126] 10  9  6  7 10  7  8 14 15 14 16 10  7  8 10  8 10 13  6 10 23 18 14
12 17
## [151]  4 11 16 16  8 14 10  9  6 11 10  8  8 10  8  8  8 11 10 10 13  8 10
11 10
## [176] 13  8  4 12 12 12 10 14  8 13  5  6 12 10 12  8 17 13  5  8 17 15 15
10  7
## [201]  8 12  8  8 14 10  8 10 13  8 12  6 14  6 12  8  9  5 12  8 11 14  9
12  5
## [226] 15  7  8 11 10  9 11  9 15  7  7  5 12 15  6 11  5  8 16 12 12 12  9
11 12
## [251] 14  6 13  9  9 10  4  6 11  9  8 12 12  6 12 13  7 14  8 10  7  5  9
16 12
## [276]  7  9  6  6  6  9  8  7 12 12 13 13  9 10 12  7  9  7  6 12 12 11 11
11 13
```

```
## [301] 12  7  9  8  9  8 11 12 10 15 11 11 10  7 10 14  8  4  5 10 12 11  6
8  5
## [326] 10 10 10 17  6 11  9 11  7 15  8  5 15  7  9  6  7  7 13  7 10  5  9
6 10
## [351]  8 10 10  8 13 10 15 11 10  8 21 11 12 19 10  5  4  9  5  9 13 10  4
16  7
## [376] 10  9  7  9 13 13 10 11  8 14 15 18 11  8 10  8 10 10  8 16 14 12 11
9  6
## [401] 13  9  5 15  5 10  9  3 10  9  9  9 12 10  8 11  8  7  6 14 14 17  9
11 17
## [426] 14  9 12  9  8  7  8 10 10  7 10  7  9 20  5  9 14  7  4  8  3 16  6
8  7
## [451]  7 11 11  9  9  7  8 13 13 11 10  6  5 11  5  9 13  8 10 11 11 14  9
9 13
## [476] 13 15  9 12  9  6 13  9  8 10  5  8 13 12 12  9 10 11 14 10  7 14 18
8 10

x_sim_pois = rep(0,500)
for (i in 1:500){   #generating samples using runif
  sum = 0
  while(sum<1){
    u_pois = runif(1,min=0,max=1)
    sum = sum +(-(1/10)*log(u_pois))
    x_sim_pois[i]=x_sim_pois[i]+1
  }
}
print(x_sim_pois[11:15])

## [1]  5 14 13  7 13

#solution 3.b

u = runif(500,min=0,max=1)   #generating samples using runif
x_sim = (2*u)^0.5
print(x_sim[1:5])

## [1] 1.2671578 0.2050903 1.1747872 0.6938349 1.3218337

#solution 3.c

norm_dist = rnorm(500,mean =3,sd =2) #generating samples from distribution
function

n = 500  #Generating samples using runif
u1 = runif(n, min = 0, max = 1)
u2 = runif(n, min = 0, max = 1)
x1 = (3+sqrt(-8*log(u1)))*cos(8*pi*u2)
x2 = (3+sqrt(-8*log(u1)))*sin(8*pi*u2)
print(x1[1:5])

## [1] -0.3937524 -1.9529016 -6.7004239  1.2213196 -2.2604175
```

```r
print(x2[1:5])

## [1]  4.104469 -5.001463 -4.410409 -4.343268 -4.954600
```