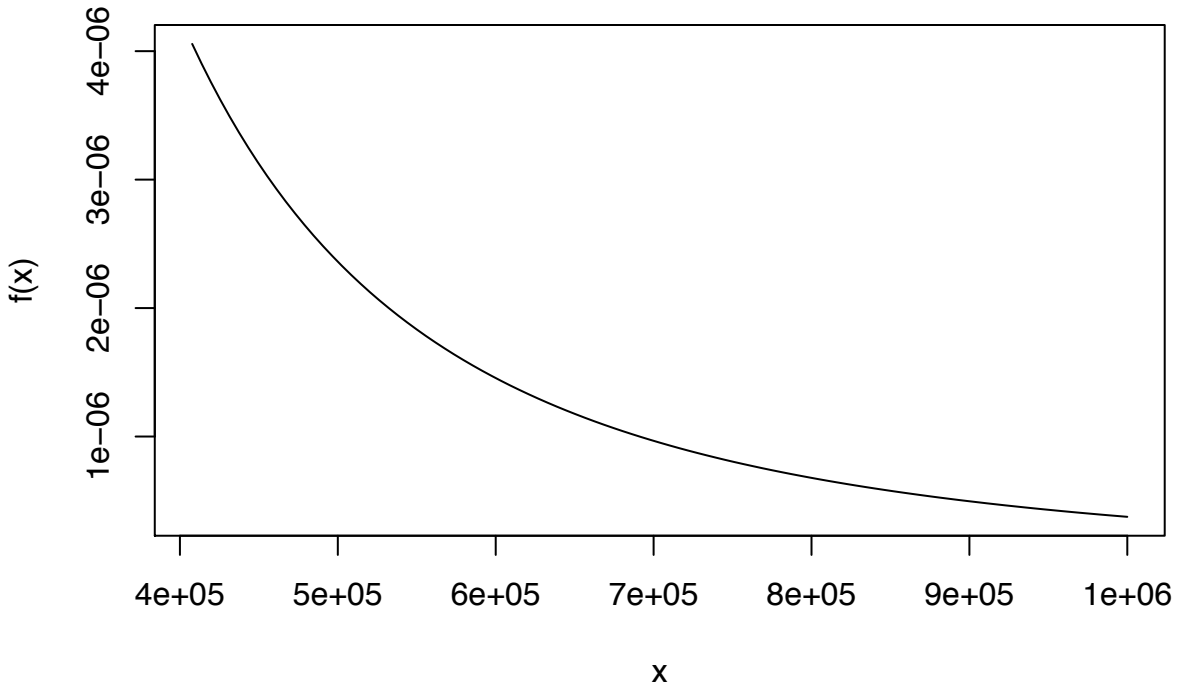
5206hw6_br2498

Bo Rong br2498

November 5, 2016

```
#1.
g1<-function(x,xmin=407760,a=2.654){
  value<-(a-1)/xmin*(x/xmin)^(-a)
  return(value)
}
curve(g1,from=407760,to=1000000,xlab="x",ylab = "f(x)")
```

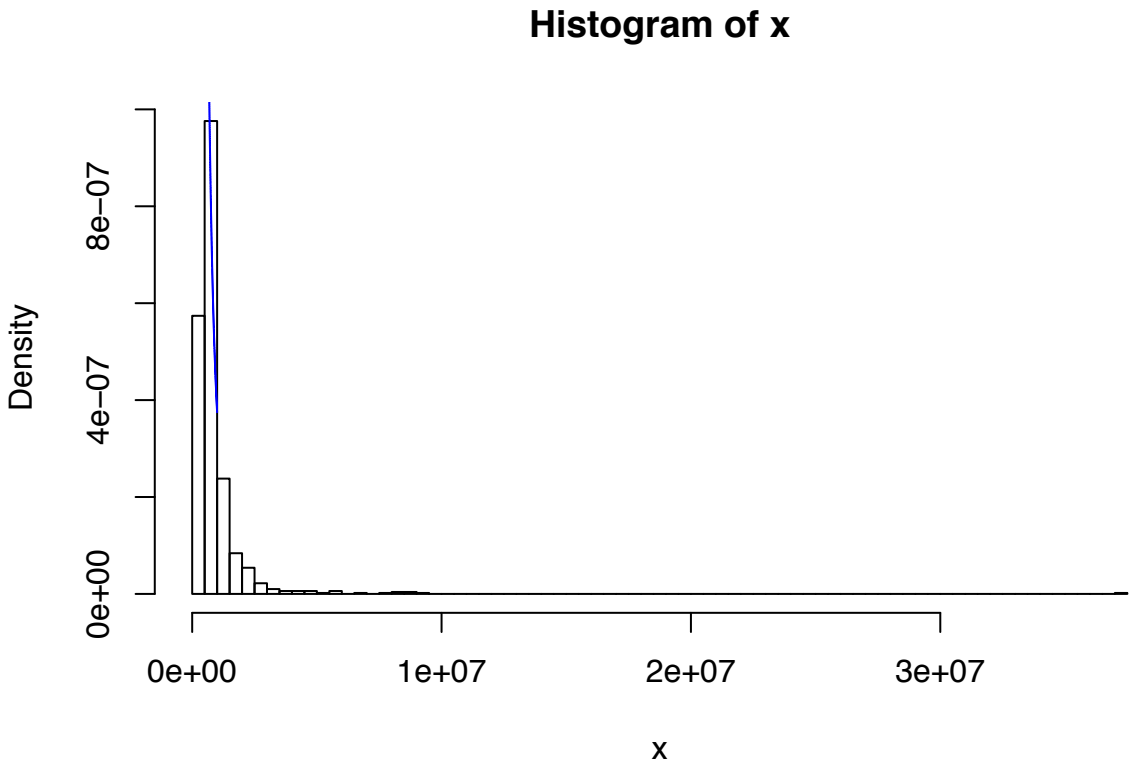



```
#2.
upper.income<-function(x,xmin=407760,a=2.654){
value1<-xmin*(1-x)^(1/(-a+1))
  return(value1)
}
upper.income(0.5)
```

```
## [1] 620020.2
```

```
#3.
n<- 1000
u<-runif(n,0,1)
xmin=407760
a=2.654
F.inverse <- function(u,xmin,a) {return(xmin*(1-u)^(1/(-a+1)))}
x<- F.inverse(u,xmin,a)
hist(x,breaks=100, prob = TRUE) # histogram
```

```r
y<-seq(407760,1000000)
lines(y, (a-1)/xmin*(y/xmin)^(-a),col="blue") # density curve f(x)
```

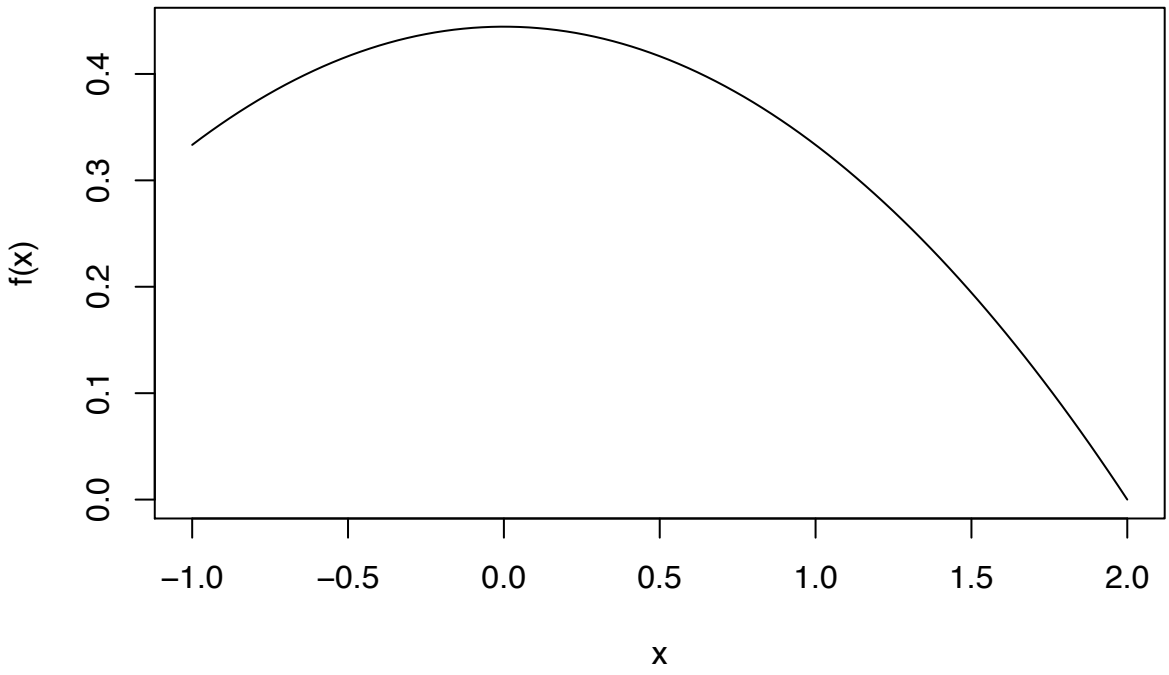



```r
#4.
median(x)
```

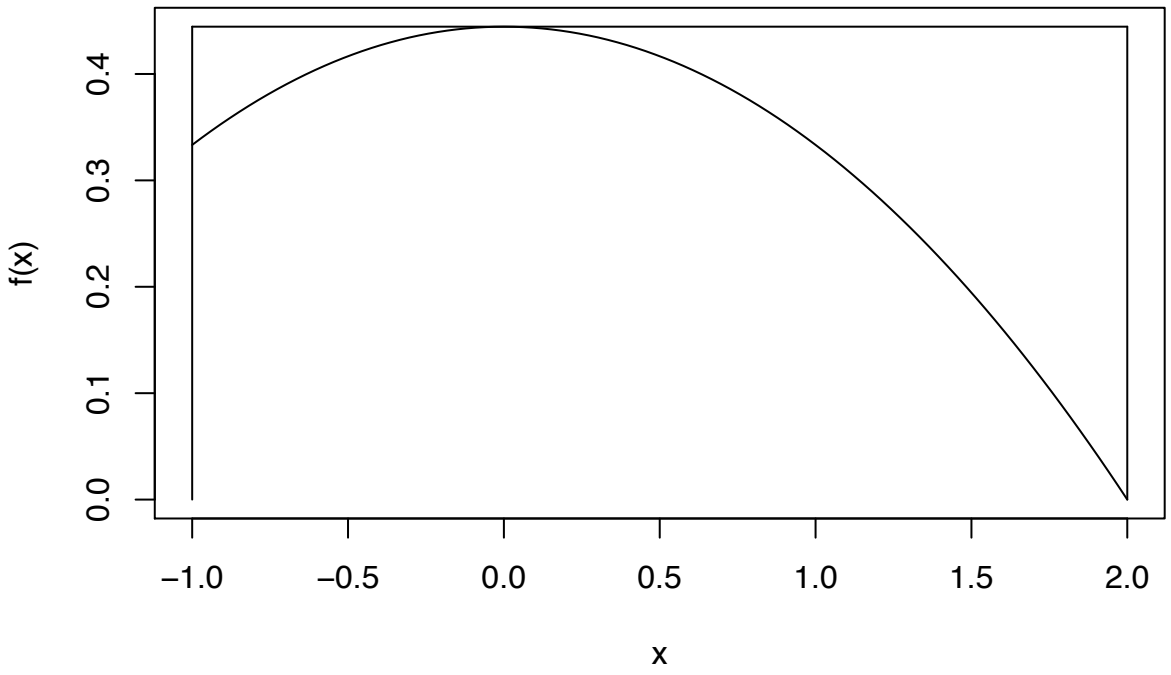
```
## [1] 622147.9
```

```r
# the median income in the simulated set is quite close to 620020.2
```

```r
#5.
g3<-function(x){
  value3<-1/9*(4-x^2)
  return(value3)
}
curve(g3,from=-1,to=2,xlab="x",ylab = "f(x)")
```

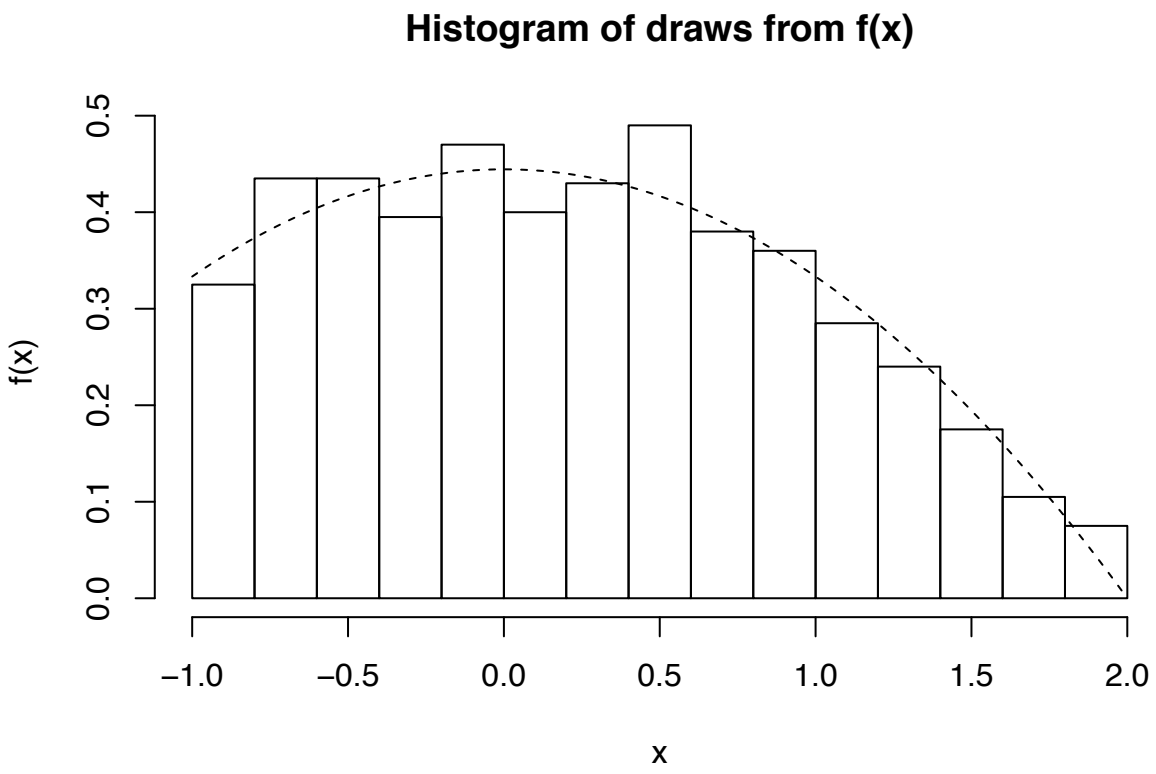



```r
#6.
f <- function(x) {
return(ifelse((x < -1 | x > 2), 0, 1/9*(4-x^2)))}
f.max<-4/9
e <- function(x) {
return(ifelse((x < -1 | x > 2), Inf, f.max))}
x <- seq(-1, 2, length = 100)
plot(x, f(x), type="l", ylab="f(x)")
lines(c(-1, -1), c(0, e(-1)), lty = 1)
lines(c(2, 2), c(0, e(2)), lty = 1)
lines(x, e(x), lty = 1)
```




```
#7.
n.samps <- 1000
n <- 0
samps <- numeric(n.samps)
while (n < n.samps) {
y<-runif(1,-1,2)
u<-runif(1)
if(u<f(y)/e(y)){
  n<-n+1
  samps[n]<-y}
}
```

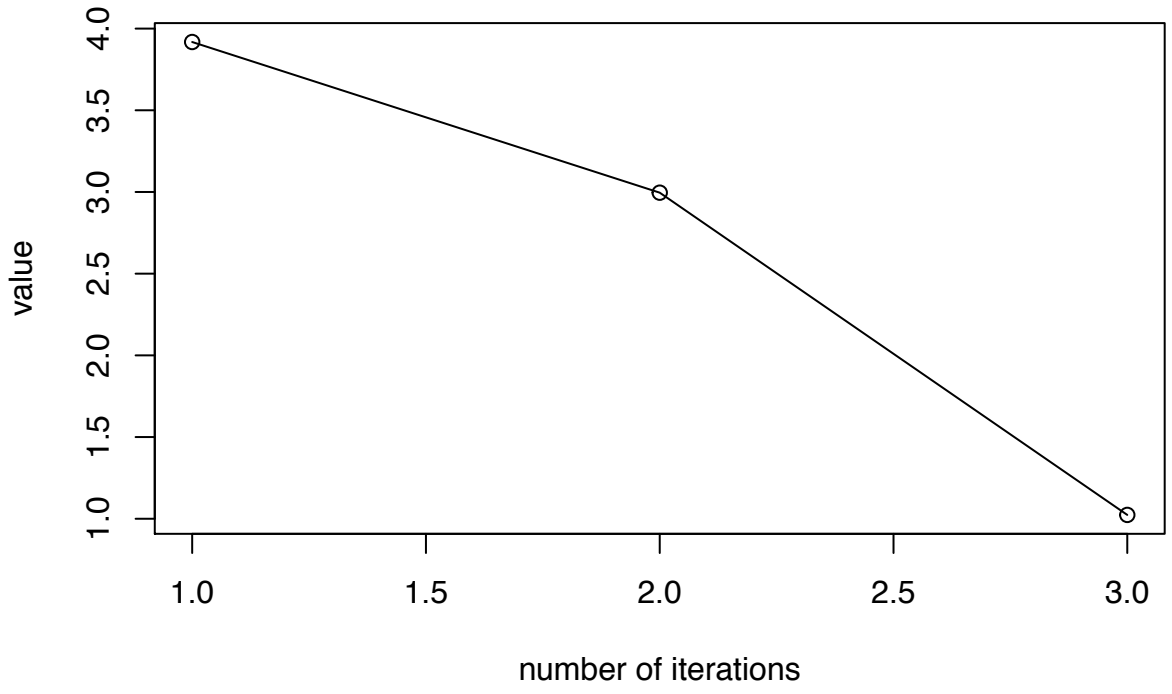
```
#8.
x <- seq(-1, 2, length = 100)
hist(samps, prob = T, ylab = "f(x)", xlab = "x", main = "Histogram of draws from f(x)")
lines(x, f(x), lty = 2)
```



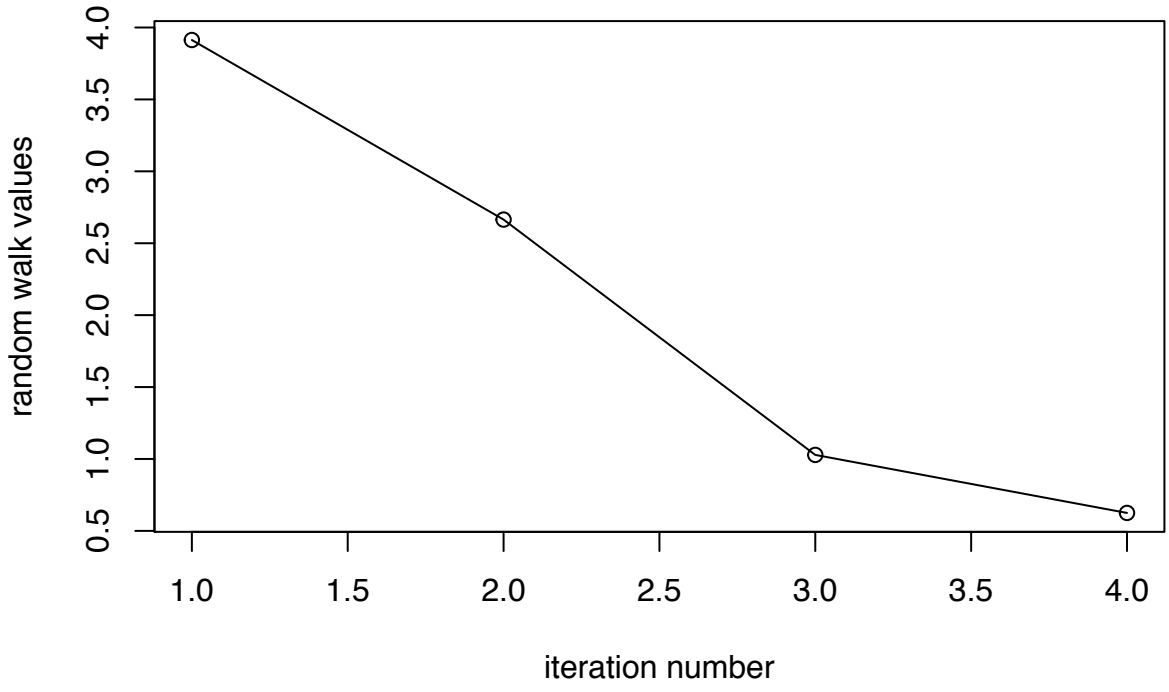

```
#9.
x=5
c=1
x.vals = NULL
while(x>0) {
  r<-runif(1,-2,1)
  x= x + r
  if(x>0)
  x.vals[c] <- x
  c  <- c + 1
}
x.vals
```

```
## [1] 3.917820 2.995729 1.023501
```

```
#10.
plot(x.vals,type="o",ylab = "value",xlab = "number of iterations")
```

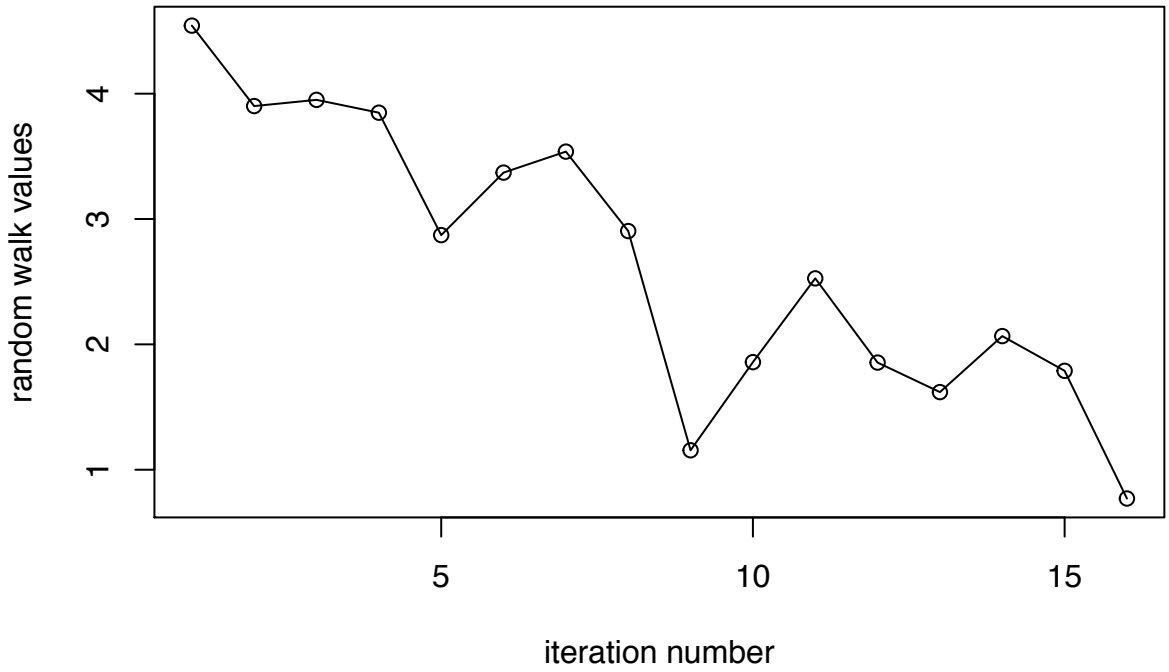



```
#11.
random.walk<-function(x.start,plot.walk){
x<-x.start
c=1
x.vals<-c(0)
while(x>0) {
r<-runif(1,-2,1)
x= x + r
 if(x>0)
x.vals[c] <- x
c  <- c + 1
}
num.steps<-c-1
result<-list(x.vals,num.steps)
names(result)<-c("x.vals","num.steps")
if(plot.walk==TRUE)
{plot(x.vals,type="o",ylab = "random walk values", xlab = "iteration number")
  return(result)}
else
  {return(result)}
 }
  random.walk(5,TRUE)
```




```
## $x.vals
## [1] 3.913016 2.664430 1.028225 0.624788
##
## $num.steps
## [1] 5
```

```
random.walk(5,TRUE)
```




```
## $x.vals
```

```
##  [1] 4.5427312 3.9009547 3.9506113 3.8483000 2.8713270 3.3702532 3.5376708
##  [8] 2.9039863 1.1550996 1.8581398 2.5264483 1.8548072 1.6191866 2.0659745
## [15] 1.7890309 0.7703217
##
## $num.steps
## [1] 17
```

```
  random.walk(10,FALSE)
```

```
## $x.vals
##  [1] 8.1625800 6.4842772 7.1190493 7.7265503 8.2442820 6.7301309 6.2818642
##  [8] 6.8269980 6.6147268 4.8961409 5.7180768 6.2113251 6.4502199 5.6561061
## [15] 4.6983943 2.8367804 3.7274640 3.7048551 1.7823815 2.5829804 1.1730224
## [22] 0.1644123
##
## $num.steps
## [1] 23
```

```
  random.walk(10,FALSE)
```

```
## $x.vals
##  [1] 8.1531196 7.0238621 7.3474862 5.5394165 3.7877488 1.8498421 2.2561228
##  [8] 3.0483605 3.2015936 4.0275791 5.0181000 3.5280475 3.5105513 3.5937990
## [15] 3.6771281 3.7755954 2.3120413 2.4805379 3.3323613 3.9609172 2.1706362
## [22] 2.4943122 2.5971947 1.4091841 1.4018698 0.1965450 0.8073977
##
## $num.steps
## [1] 28
```

```
#12.
n<-10000
samplesize <-rep(NA, n)
for(i in 1:n){
  samplesize[i] <-random.walk(5,FALSE)$num.steps
}
vector_sample<-samplesize
mean(vector_sample)
```

```
## [1] 11.2576
```

```
#13.
random.walk<-function(x.start,plot.walk,seed){
 set.seed(seed)
x<-x.start
c=1
x.vals<-c(0)
while(x>0) {
r<-runif(1,-2,1)
x= x + r
 if(x>0)
x.vals[c] <- x
c  <- c + 1
```

```
}
num.steps<-c-1
result<-list(x.vals,num.steps)
names(result)<-c("x.vals","num.steps")
if(plot.walk==TRUE)
{plot(x.vals,type="o",ylab = "random walk values", xlab = "iteration number")
  return(result)}
else
  {return(result)}
}
random.walk(5,FALSE,NULL)
```

```
## $x.vals
## [1] 4.688501 2.914825 1.161845 1.465457 1.989958 1.283236
##
## $num.steps
## [1] 7
```

```
random.walk(5,FALSE,NULL)
```

```
## $x.vals
## [1] 3.674235 3.676329 2.750122 1.043769 1.246978
##
## $num.steps
## [1] 6
```

```
random.walk(5,FALSE,33)
```

```
## $x.vals
## [1] 4.3378214 3.5217724 2.9729590 3.7295869 4.2612312 3.8132800 3.1246550
## [8] 2.1542497 0.2008006
##
## $num.steps
## [1] 10
```

```
random.walk(5,FALSE,33)
```

```
## $x.vals
## [1] 4.3378214 3.5217724 2.9729590 3.7295869 4.2612312 3.8132800 3.1246550
## [8] 2.1542497 0.2008006
##
## $num.steps
## [1] 10
```