Exam1

Sebastian Kirkpatrick

2024-10-15

Question 1

```r
f <- function(x){
  out <- ((x^2)/(100)) + sin(x)
  return(out)
}

secant <- function(f, x0 = c(0,1), max.iter = 100, tol = 0.0000000001){

  x <- x0
  n <- 2
  eps <- 1000
  while (n < max.iter & eps > tol) {

    x[n + 1] <- x[n] - f(x[n]) * ((x[n] - x[n - 1]) / (f(x[n]) - f(x[n - 1])))

    eps <- abs(f(x[n+1]))
    n <- n + 1

  }

return(x)

}

secant(f,c(-100,-75))
```

```
##  [1] -100.000000  -75.000000  -42.723040  -26.163974  -18.662250   -6.791313
##  [7]   -6.872684   -6.755959   -6.757437   -6.757380   -6.757380
```

```r
secant(f,c(-100,-50))
```

```
##  [1] -100.000000  -50.000000  -33.213028  -22.110768  -11.096830   -2.282756
##  [7]   -4.402565   -3.090027   -3.037705   -3.048502   -3.048523   -3.048523
```

```r
secant(f,c(-50,-8))
```

```
## [1] -50.000000  -8.000000  -8.572903  -8.603241  -8.593779  -8.593867  -8.593867
```

```
secant(f,c(-5,5))
```

```
## [1] -5.000000  5.000000  1.303544  3.449786  3.110444  3.249446  3.247247
## [8]  3.247234  3.247234
```

```
secant(f, c(-1,1))
```

```
## [1] -1.000000e+00  1.000000e+00 -1.188395e-02  2.042524e-03 -2.825797e-07
## [6] -5.575192e-12
```

```
secant(f, c(5,50))
```

```
## [1]  5.000000 50.000000  6.253671  5.604775  5.906077  5.925975  5.924539
## [8]  5.924544  5.924544
```

Zeroes: -6.757, -3.049, -8.594, 3.247, 0, 5.925

Question 2

```
set.seed(740)
n <- 10000

x <- rexp(n,1)
x <- x[x>=4]
c <- exp(-4)

c * mean(log(x))
```

```
## [1] 0.02917288
```

Approx. Value = 0.0292

Question 3

a)

```
set.seed(740)
n <- 1000000
small_stick <- rep(0,n)

for (i in 1:n){
  cut <- runif(1)
  if (cut > 0.5){
    small_stick[i] <- 1 - cut
  } else {
    small_stick[i] <- cut
  }
}

mean(small_stick)
```

```
## [1] 0.2499357
```

Average length: 0.250 feet

b)

```
set.seed(740)
n <- 1000000
four_cuts <- rep(0,4)
ratio <- rep(0,n)


for (i in 1:n){
  c1 <- runif(1)

  c2 <- runif(1)
  c3 <- runif(1)

  four_cuts[1] <- c1 * c2
  four_cuts[2] <- c1 * (1-c2)

  four_cuts[3] <- (1-c1) * c3
  four_cuts[4] <- (1-c1) * (1-c3)

  ratio[i] <- max(four_cuts) / min(four_cuts)
}

mean(ratio)
```

```
## [1] 399.4608
```

Average ratio for my sim is 399.46

Question 4

```
q4 <- function(x) {
  return(-cos(x[1]) * cos(x[2]) * exp(-(x[1]^2 + x[2]^2)))
}

test <- c(0,0)
q4(test)
```

```
## [1] -1
```

```
set.seed(740)
niter <- 100000
x <- matrix(NA,ncol=2,nrow=niter)
x[1,] <- c(1,1)


temp <- 500
sigma <- 1


for (i in 1:(niter-1)) {
  xnew <- x[i,] + rnorm(2,0,sigma)
  if(q4(xnew) < q4(x[i,])){
    x[i+1,] <- xnew
  } else {
    p <- exp(-(q4(xnew) - q4(x[i,])) / temp)
    if (runif(1) < p) {
      x[i+1,] <- xnew
    } else {
      x[i+1,] <- x[i,]
    }
  }
  temp <- 0.99*temp
}

x[100000,]
```

```
## [1] 0.0023064152 0.0005838405
```

```
q4(c(0.002,0.0005))
```

```
## [1] -0.9999936
```

Minimum: 0,0

Question 5

```
set.seed(740)
n <- 100000
deck <- rep(1:13,4)
count <- rep(0,n)

for (i in 1:n){

  shuffle <- sample(deck)

  for (j in 2:52){

    if (shuffle[j] == shuffle[j-1]){
      count[i] <- j
      break
    }
  }
  if (count[i] == 0){
    count[i] <- 52
  }
}

mean(count)
```

```
## [1] 17.04733
```

Average number of cards is 17

Question 6

```
set.seed(740)
n <- 10000
emp <- 400

days_worked <- rep(0,emp)

for (j in 320:emp){

  days_open <- rep(0,n)

  for (i in 1:n){

    births <- sample(c(1:365),j,replace=T)

    days_open[i] <- 365 - n_distinct(births)
  }

  days_worked[j] <- mean(days_open) * j
}

which.max(days_worked)
```

```
## [1] 369
```

369 workers (though I am guessing with more sims it is about 365)