Test

Maria Nathalia Vinter Kristensen;
mkris13@student.sdu.dk
(ST516 exercises)

Spring 2016

April 25, 2016

1 Chapter 3

1.1 Exercise 1 + 2

Calculate pseudorandom numbers $x_1, \ldots, x_10$ using multiplivative congruential method and mixed congruential method with seed $x_0 = 5$, $a = 3, m = 150$ and with seed $x_0 = 3$, $a = 5, c = 7, m = 200$ repectively

```
## Define variables
x0=5
a=3
m = 150

x = data.frame("x0"=x0)
## Compute x1,...,x10 using multiplicative congruential method
for (i in 1:10){
  x[1,i+1] = a*x[1,i] %% m
}
x

##   x0 V2 V3  V4  V5  V6 V7  V8  V9 V10 V11
## 1  5 15 45 135 405 315 45 135 405 315  45

## Define variables
x0=3
a=5
c = 7
m = 200

x = rbind(x,x0)

## Compute x1,...,x10 using mixed congruential method
for (i in 1:10){
  x[2,i+1] = (a*x[2,i]+c) %% m
}
## Style and show result
colnames(x) = paste("x", seq(0,10,1),sep="")
```

```
rownames(x) = c("Multiplicative GM","Mixed GM")
x[1:2,2:11]

##                     x1  x2  x3  x4  x5 x6  x7  x8  x9 x10
## Multiplicative GM 15  45 135 405 315 45 135 405 315  45
## Mixed GM          22 117 192 167  42 17  92  67 142 117
```

1.2 Exercises 3-9

For all exercises 3-9 we can use the Monte Carlo approach to evaluate the integral given. The general function for doing so is given below. Here `k` is the number estimates used in the evalutation and `expr` is the integral to be evaluated. **NOTE** that `expr` need to be an integral from 0 to 1, a function for rewriting the expression is also rewritten. (only for cases used in the exercises)

```
##################################################################
#
#  variables:
#   - k     number of iteration to go through
#   - a     lower bound for integration
#   - b     upper bound for integration
#   - expr  expression to be integrated
#   - rewrite TRUE, when we integrate something
#     else than 0 to 1
#
#  output:
#   approximated integral
#
##################################################################

monte_carlo_approach <-
  function(k = 10000, a=0, b=1, expr = function(x){
    return(x)},rewrite=FALSE, N=1) {
  h = c(0)
  for (i in 1:k) {
    if (rewrite){
      h[i] = rewriten(expr,a,b, N)
    } else {
      h[i] = expr(runif((N)))
    }
  }
  return(sum(h)/k)
}

rewriten <- function(expr,a,b,N) {
```

```r
  a = ifelse(a<b,a,b)
  b = ifelse(a<b,b,a)
  u = runif(N)
  if (a == 0 && b == 1) {
    return(expr(u))
  }
  if (is.finite(a) && is.finite(b)) {
    return(expr(a+(b-a)*u)*(b-a))
  }
  if (is.infinite(a) || is.infinite(b)) {
    if(abs(a) == abs(b)) {
      return(2*(expr(1/u-1)/u^2))
    } else {
      return(expr(1/u-1)/(u^2))
    }
  }
  return(0)
}
```

1.3 Exercise 3

We want to evaluate

$$\int_0^1 e^{e^x} dx = 6.31656$$

We can use the Monte Carlo approach directly, where `expr`$= e^{e^x}$

```r
monte_carlo_approach(expr=function(x){
  return(exp(exp(x)))})

## [1] 6.265319
```

1.4 Exercise 4

We want to evaluate

$$\int_0^1 (1-x^2)^{3/2} dx = 0.58905$$

We can use the Monte Carlo approach directly, where `expr`$= (1-x^2)^{3/2}$

```r
monte_carlo_approach(expr=function(x){
  return((1-x^2)^(3/2))})

## [1] 0.5863435
```

1.5 Exercise 5

We want to evaluate

$$\int_{-2}^{2} e^{x+x^2} dx = 93.1628$$

We cannot use the Monte Carlo approach directly, but using the function `rewriten` we can substitute to get an integral from 0 to 1

```
monte_carlo_approach(expr = function(x){
  return(exp(x+x^2))},a=-2,b=2,rewrite=TRUE)

## [1] 90.96126
```

1.6 Exercise 6

We want to evaluate

$$\int_{0}^{\infty} x \cdot (1+x^2)^{-2} dx = 0.5000$$

Again we cannot do it directly:

```
monte_carlo_approach(k=10000,expr = function(x){
  return(x*((1+x^2)^(-2)))},a=0,b=Inf,rewrite=TRUE)

## [1] 0.500781
```

1.7 Exercise 7

We want to evaluate

$$\int_{-\infty}^{\infty} x \cdot e^{-x^2} dx = 1.77245$$

Again we cannot do it directly:

```
monte_carlo_approach(expr = function(x){
  return(exp(-x^2))},a=-Inf,b=Inf,rewrite=TRUE)

## [1] 1.793106
```

1.8 Exercise 8

We want to evaluate

$$\int_0^1 \int_0^1 x \cdot e^{(x+y)^2} dx = 4.89916$$

This we can do directly, using the optinal parameter $N = 2$ in `monte_carlo_approach`:

```
monte_carlo_approach(expr = function(x){
  return(exp((x[1]+x[2])^2))},N=2)

## [1] 4.834441
```