

prova 1

November 23, 2020

```
[173]: # Questão 1
#a)
BIGINT <- 100000
sequencia <- seq(from=0, to=6, by=6/BIGINT)
normal <- runif(n=BIGINT, 0, 0.5)
under <- 0

integral <- function(x) {
  return (x/(1 + (2*x^2) + x^4))
}

for (i in 1:BIGINT) {
  results <- integral(sequencia[[i]])
  if (normal[[i]] < results) {
    under <- under + 1
  }
}

resultado <- (under / length(sequencia)) * (6 - 0) * (0.5 - 0)
print(resultado)
```

```
[1] 0.4877051
```

```
[174]: #b)

# sobre os inptus na função estimaIntegral:
#a e n são os valores da intgral
#c e d são valores do retângulo que fecham em torno da integral

estimaIntegral <- function( a, n, c, d) {
  sequencia <- seq(from=a, to=n, by=n/BIGINT)
  normal <- runif(n=BIGINT, c, d)
  under <- 0

  integral <- function(x) {
    return (x/(1 + (2*x^2) + x^4))
  }
```

```

for (i in 1:BIGINT) {
  results <- integral(sequencia[[i]])
  if (normal[[i]] < results) {
    under <- under + 1
  }
}

return((under / length(sequencia)) * (n - a) * (d - c))
}

#usei a funçãoq eu criei para a questão 2 para responder a questão 1, sendo o
↪ intervalo da integral de 0 à 6:
print(estimaIntegral(0, 6, 0, 0.5))
print(estimaIntegral(0, 10, 0, 0.5))

```

```

[1] 0.4900451
[1] 0.4900951

```

[175]: #Questão 2.

```

countA <- 0
countB <- 0
countC <- 0

allDices <- vector()
for (i in 1:BIGINT) {

  d1 <- sample(1:6, size = 1, replace = TRUE)
  d2 <- sample(1:6, size = 1, replace = TRUE)
  d3 <- sample(1:6, size = 1, replace = TRUE)
  d4 <- sample(1:6, size = 1, replace = TRUE)
  d5 <- sample(1:6, size = 1, replace = TRUE)

  #contando a probabilidade de algum deles ser 2
  if (d1 == 2 || d2 == 2 || d3 == 2 || d4 == 2 || d5 == 2){
    countA <- countA + 1
  }

  # contando a probabilidade da soma dos resultados ser 14
  if(d1+d2+d3+d4+d5==14){
    countB <- countB + 1
  }
}

```

```

}

dados <- c(d1,d2,d3,d4,d5)
# checando se todos os resultados são diferentes, e se forem, incrementa o
↪countadorC
#Se encontrar algum TRUE no array gerado por match, significa que tem uma
↪duplicata.
duplicado <- match(TRUE,duplicated(dados))
#Se não houver um TRUE (ou seja, nenhum duplicado), vai retornar <NA>.
↪Nesse caso, tem que incrementar
#como amostra
if(is.na(duplicado)){
countC <- countC + 1
}
}

#resultado item a)
x <- countA/BIGINT
print(x)

#resultado item b)
x <- countB/BIGINT
print(x)

#resultado do item c)
x <- countC/BIGINT
print(x)

```

```

[1] 0.59852
[1] 0.07207
[1] 0.09298

```