Assignment_STA445

BANDELLE

2023-10-13

Q1

a)

```
funtn_1 <- function(x,a,b){
  if( x>= a & x<=b){ v= 1/(b-a)}
  else{ v=0}
  return(v)
}

funtn_1(0.9,1,5)
```

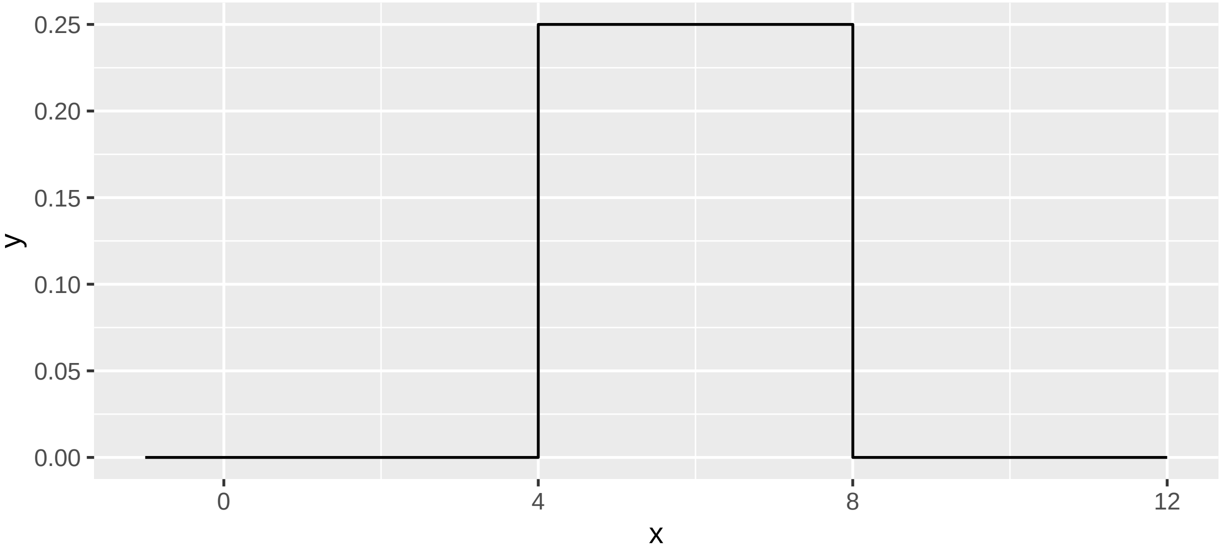
```
## [1] 0
```

b)

```
        duniform <- function(x, a, b){
          output <- NULL
          for( i in 1:length(x) ){
            if( x[i] >= a & x[i] <= b ){
              output[i] <- 1/(b-a)
            }else{
              output[i] <- 0 }
            }
          return(output)
        }
duniform(c(4,8,9,11),5,10)
```

```
## [1] 0.0 0.2 0.2 0.0
```

```
        data.frame( x=seq(-1, 12, by=.001) ) %>%
          mutate( y = duniform(x, 4, 8) ) %>%
          ggplot( aes(x=x, y=y) ) +
          geom_step()
```



c)

```
library(microbenchmark)
microbenchmark::microbenchmark( duniform(seq(-4,12,by=.0001), 4, 8), times=100)
```

```
## Unit: milliseconds
##                                        expr     min      lq     mean   median
##  duniform(seq(-4, 12, by = 1e-04), 4, 8) 59.0421 61.7358 68.29806 63.02985
##       uq      max neval
##  68.9357 144.7444   100
```

The median time is 56.99265 seconds

d)

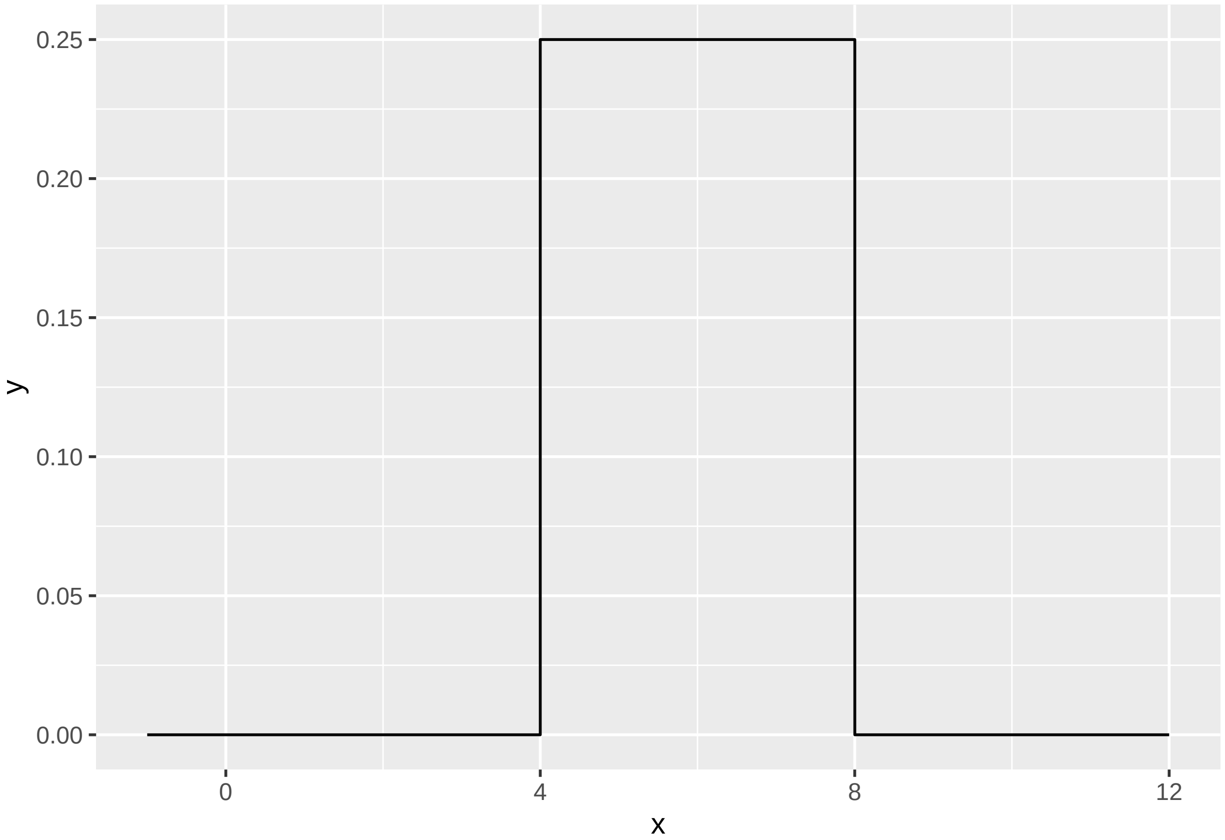
```
        duniform <- function(x, a, b){

          ifelse(x>=a & x<=b, 1/(b-a) , 0)

        }
duniform(c(4,8,9,11),5,10)
```

```
## [1] 0.0 0.2 0.2 0.0
```

```
data.frame( x=seq(-1, 12, by=.001) ) %>%
          mutate( y = duniform(x, 4, 8) ) %>%
          ggplot( aes(x=x, y=y) ) +
          geom_step()
```



```
library(microbenchmark)
microbenchmark::microbenchmark( duniform(seq(-4,12,by=.0001), 4, 8), times=100)
```

```
## Unit: milliseconds
##                                        expr      min       lq     mean   median
##  duniform(seq(-4, 12, by = 1e-04), 4, 8) 4.064301 4.471351 7.557013 5.986501
##        uq      max neval
##  7.719101 101.4646   100
```

The median time is 4.168 seconds

Q2

```
duniform <- function(x, a=1, b=5){

          ifelse(x>=a & x<=b, 1/(b-a) , 0)

        }
duniform(c(4,8,9,11))
```

```
## [1] 0.25 0.00 0.00 0.00
```

Q3

```
    standardize <- function(x){
      x_bar<-mean(x)
      s<-sd(x)
      output_z<-(x-x_bar)/s
  return(output_z)

    }

    data( 'iris' )
    # Graph the pre-transformed data.
    ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, color=Species)) +
      geom_point() +
      labs(title='Pre-Transformation')

    # Standardize all of the numeric columns
    # across() selects columns and applies a function to them
    # there column select requires a dplyr column select command such
    # as starts_with(), contains(), or where().  The where() command
    # allows us to use some logical function on the column to decide
    # if the function should be applied or not.
    iris.z <- iris %>% mutate( across(where(is.numeric), standardize) )

    # Graph the post-transformed data.
    ggplot(iris.z, aes(x=Sepal.Length, y=Sepal.Width, color=Species)) +
      geom_point() +
      labs(title='Post-Transformation')
```

```
library(microbenchmark)
microbenchmark::microbenchmark( duniform(seq(-4,12,by=.0001), 4, 8), times=100)
```

```
## Unit: milliseconds
##                                     expr      min       lq     mean   median
##  duniform(seq(-4, 12, by = 1e-04), 4, 8) 4.212602 4.287751 6.691566 4.910251
##      uq     max neval
##  7.2876 101.504   100
```

Q4

```
 bb <- function(x){
   for(i in 1:length(x))
  if(i%%3==0 & i%%5==0){x[i] <-"Fizz Buzz"}
   else if(i%%3==0){x[i] <-"Fizz"}
   else if(i%%5==0){x[i]<-"Buzz"}
   else{}

    return(x)
 }
 bb(x=seq(1:100))
```

```
##   [1] "1"         "2"         "Fizz"      "4"         "Buzz"      "Fizz"
##   [7] "7"         "8"         "Fizz"      "Buzz"      "11"        "Fizz"
##  [13] "13"        "14"        "Fizz Buzz" "16"        "17"        "Fizz"
##  [19] "19"        "Buzz"      "Fizz"      "22"        "23"        "Fizz"
##  [25] "Buzz"      "26"        "Fizz"      "28"        "29"        "Fizz Buzz"
##  [31] "31"        "32"        "Fizz"      "34"        "Buzz"      "Fizz"
##  [37] "37"        "38"        "Fizz"      "Buzz"      "41"        "Fizz"
##  [43] "43"        "44"        "Fizz Buzz" "46"        "47"        "Fizz"
##  [49] "49"        "Buzz"      "Fizz"      "52"        "53"        "Fizz"
##  [55] "Buzz"      "56"        "Fizz"      "58"        "59"        "Fizz Buzz"
##  [61] "61"        "62"        "Fizz"      "64"        "Buzz"      "Fizz"
##  [67] "67"        "68"        "Fizz"      "Buzz"      "71"        "Fizz"
##  [73] "73"        "74"        "Fizz Buzz" "76"        "77"        "Fizz"
##  [79] "79"        "Buzz"      "Fizz"      "82"        "83"        "Fizz"
##  [85] "Buzz"      "86"        "Fizz"      "88"        "89"        "Fizz Buzz"
##  [91] "91"        "92"        "Fizz"      "94"        "Buzz"      "Fizz"
##  [97] "97"        "98"        "Fizz"      "Buzz"
```

Q5

~~~~
```r
myFill <- function(x){

 for(i in which(is.na(x)))
if( i==2 | i==3){ x[i] <- "A"}
  else if( i==6 | i==7 | i==8 ){ x[i] <- "C"}
  else{}
  return(x)
  # Stuff in here!
}
test.vector <- c('A',NA,NA, 'B','C', NA,NA,NA)
myFill(test.vector)
```
~~~~