STA 445

6260662

Constant Yaokumah

2023-10-13

Q1 a dunif

```
duniform <- function(x,a,b){
  if(a <= x & x <= b){
    output = 1/(b-a)
  }else{
    output = 0
  }

  return(output)
}
duniform(-20,1,10)
```

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

```
duniform(6,1,10)
```

```
## [1] 0.1111111
```

```
duniform(20,1,10)
```

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

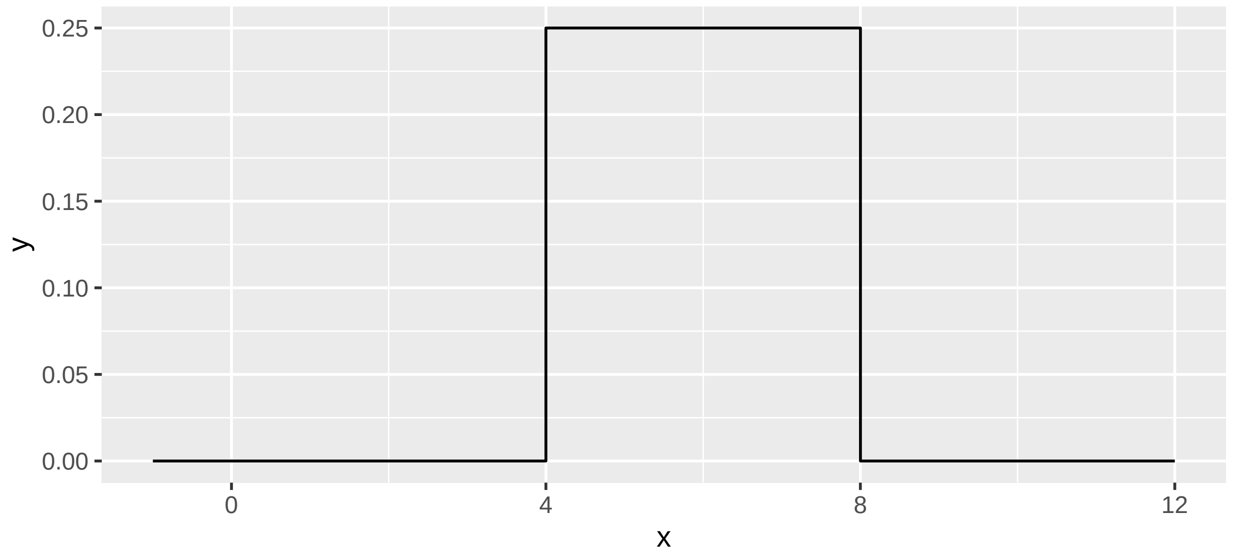
Q1 b using loop(for)

```
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(1:8,2,20)
```

```
## [1] 0.00000000 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
## [8] 0.05555556
```

Q1 bi, verify above code

```
library(dplyr)
library(ggplot2)
data.frame( x=seq(-1, 12, by=.001) ) %>%
    mutate( y = duniform(x, 4, 8) ) %>%
    ggplot( aes(x=x, y=y) ) +
    geom_step()
```



Q1 c using (microbenchmark)

```
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) 84.2277 88.32255 95.07784 92.16765
##       uq      max neval
##  97.5612 172.5128   100
```

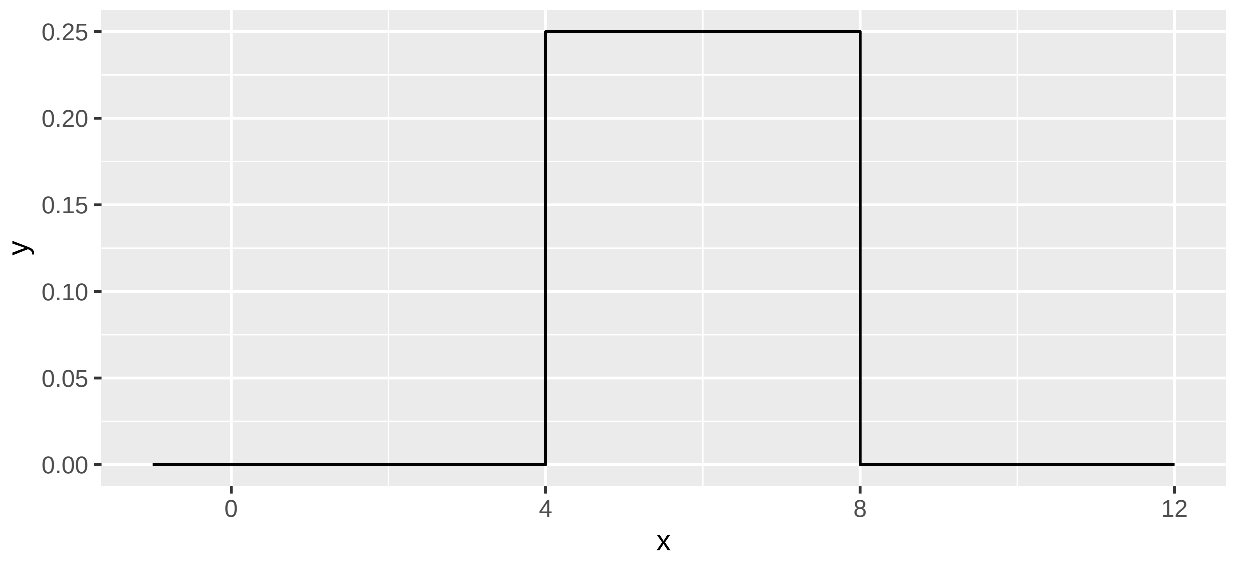
Q1 d using ifelse statement

```
duniform1 <- function(x, a, b){
   output <- ifelse(x>=a & x <= b, 1/(b-a),0)
   return(output)
}
duniform1(1:21,2,20)
```

```
##  [1] 0.00000000 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
##  [7] 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
## [13] 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
## [19] 0.05555556 0.05555556 0.00000000
```

Q 1d verify plot

```
library(dplyr)
library(ggplot2)
data.frame( x=seq(-1, 12, by=.001) ) %>%
    mutate( y = duniform1(x, 4, 8) ) %>%
    ggplot( aes(x=x, y=y) ) +
    geom_step()
```



## Q 1 d verify microbenchmark

```
library(microbenchmark)
microbenchmark::microbenchmark( duniform1( seq(-4,12,by=.0001), 4, 8), times=100)
```

```
## Unit: milliseconds
##                                       expr    min      lq     mean  median
##  duniform1(seq(-4, 12, by = 1e-04), 4, 8) 4.4142 5.03255 7.500036 6.27805
##      uq     max neval
##  8.5836 68.5286   100
```

Codes in Question 1d are much easier to write and run faster compared to Question 1 b

Q2 setting default values

```
duniform2 <- function(x,min = 0, max = 1){
  output <- ifelse(x>=min & x <= max, 1/(max-min),0)
  return(output)
}
duniform2(1:7,2,20)
```

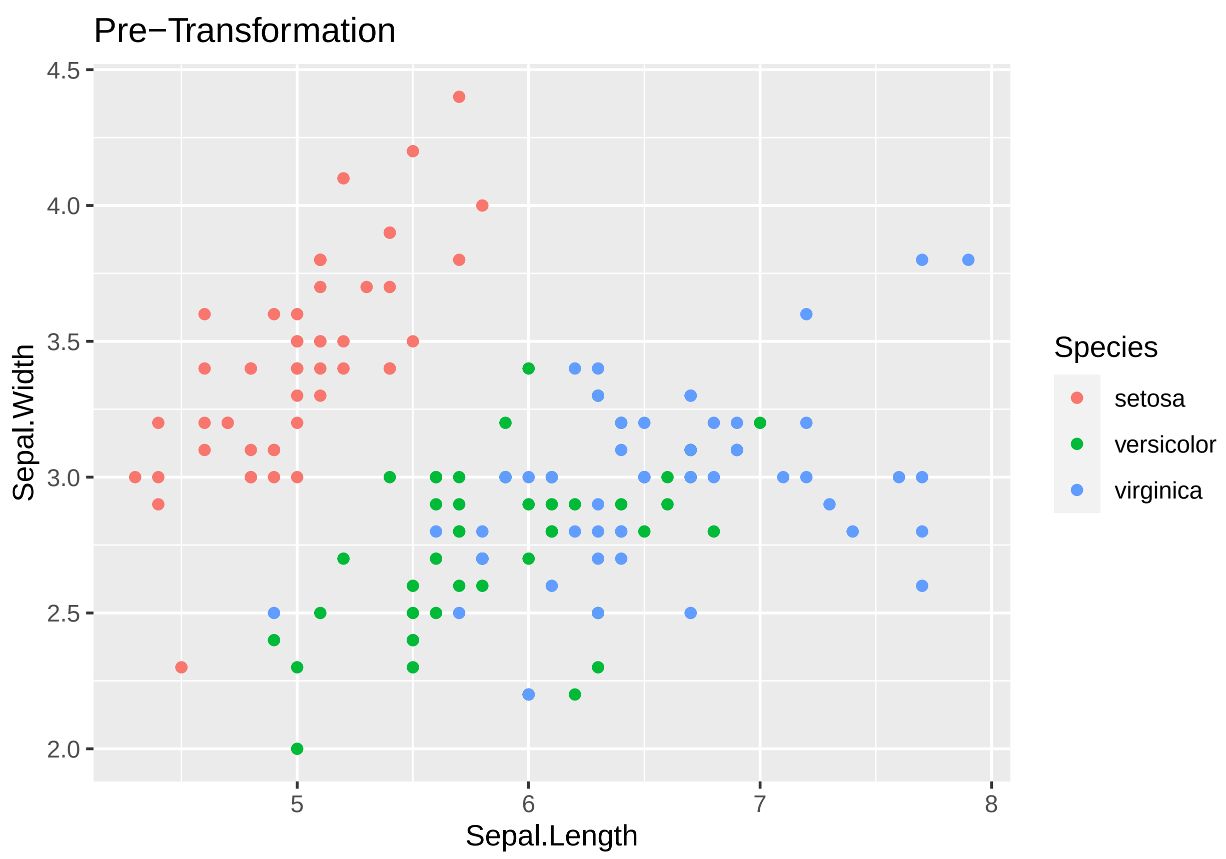
```
## [1] 0.00000000 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556 0.05555556
```

```
duniform2(3)
```

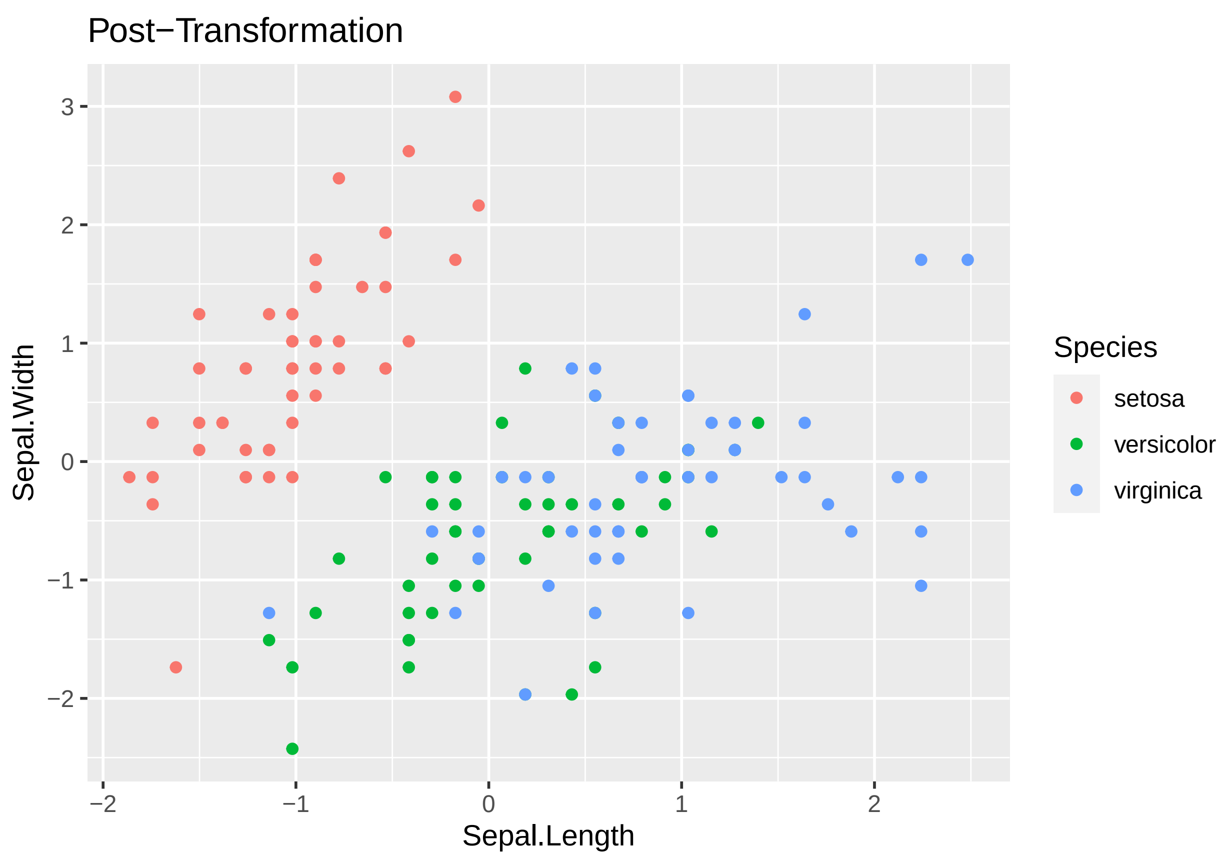
```
## [1] 0
```

Q3

```
standardize <- function(x){
  (x-mean(x))/sd(x)
    }

data( 'iris' )

ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width, color=Species)) +
  geom_point() +
  labs(title='Pre-Transformation')
```



```r
iris.z <- iris %>% mutate( across(where(is.numeric), standardize) )

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



Q4 Using paste and %%

```
Fizz_Buzz_Game <- function(n){
 output <- c()
  for( i in 1:length(n)){
    output[i] <- ""
    if(i %% 3 == 0){output[i] <- paste(output[i], "Fizz")}
    if(i %% 5 == 0){output[i] <-  paste(output[i], "Buzz")}
    if(output[i] == ""){output[i] <- i}
  }
 return(output)
}

Fizz_Buzz_Game(1:50)
```

```
##  [1] "1"          "2"          " Fizz"      "4"          " Buzz"
##  [6] " Fizz"      "7"          "8"          " Fizz"      " Buzz"
## [11] "11"         " Fizz"      "13"         "14"         " Fizz Buzz"
## [16] "16"         "17"         " Fizz"      "19"         " Buzz"
## [21] " Fizz"      "22"         "23"         " Fizz"      " Buzz"
## [26] "26"         " Fizz"      "28"         "29"         " Fizz Buzz"
## [31] "31"         "32"         " Fizz"      "34"         " Buzz"
## [36] " Fizz"      "37"         "38"         " Fizz"      " Buzz"
## [41] "41"          " Fizz"       "43"          "44"          " Fizz Buzz"
## [46] "46"          "47"          " Fizz"       "49"          " Buzz"
```

Q5 Filling NA

```r
myFill <- function(x){
for(i in 1:length(x)){
  if(is.na(x[i])){
    x[i] =x[i-1]
    }
  }
return(x)
}

test.vector <- c('A',NA,NA, 'B','C', NA,NA,NA)
myFill(test.vector)
```

```
## [1] "A" "A" "A" "B" "C" "C" "C" "C"
```