Assignment 1

Mauricio de Oliveira Reis | D21125621

### Two dice and are rolled and let be the sum

#### 1. Find [**10 marks**]

# Exercise 1 and 2 data setup:  
die1 <- c(1:6)  
die2 <- c(1:6)  
rolls <- expand.grid(die1, die2)   
R <- data.frame(rolls, rowSums(rolls))  
colnames(R) <- c("D\_1", "D\_2", "S")  
totalDifferentRolls <- dim(R)[1]  
#Exercise 1:  
answer1 <- sum(R$D\_1 < 4 | R$S > 9) / totalDifferentRolls  
answer1

## [1] 0.6666667

#### 2. Find [**10 marks**]

answer2 <- sum(R$D\_1 < 4 & R$S == 8) / sum(R$S == 8)  
answer2

## [1] 0.4

### Suppose

#### 3. Find [**10 marks**]

#data setup exercise 3 and 4  
n <- 10  
p <- 0.3  
#exercise 3  
xSmallerEqualSix <- c(0:6)  
xBetweenTwoAndEight <- c(3:7)  
intersection <- intersect(xSmallerEqualSix, xBetweenTwoAndEight)  
answer3 <- sum(dbinom(intersection,n,p))  
answer3

## [1] 0.6066251

#### 4. Find [**10 marks**]

#Pr(X<4 And X<= 6) / Pr(X<=6) -> Pr(X<4) / Pr(X<=6)   
probSmallerThanFour <- pbinom(3, n, p)  
probSmallerEqualSix <- pbinom(6, n, p)  
answer4 <- probSmallerThanFour / probSmallerEqualSix  
answer4

## [1] 0.6565651

### Suppose

#### 5. Find [**10 marks**]

#data setup exercise 5 and 6  
lambda <- 5  
  
# exercise 5  
xBetweenThreeAndSevenInclusive <- c(4:7)  
answer5 <- sum(dpois(xBetweenThreeAndSevenInclusive, lambda))  
answer5

## [1] 0.6016024

#### 6. Find [**10 marks**]

# insert code  
probSmallerThanThree <- ppois(2,lambda)  
probBiggerOrEqualSeven <- ppois(6, lambda, lower.tail = FALSE)  
probOfIntersection <- 0  
answer6 <- probSmallerThanThree + probBiggerOrEqualSeven - probOfIntersection  
answer6

## [1] 0.3624686

### Create a function [**40 marks**]

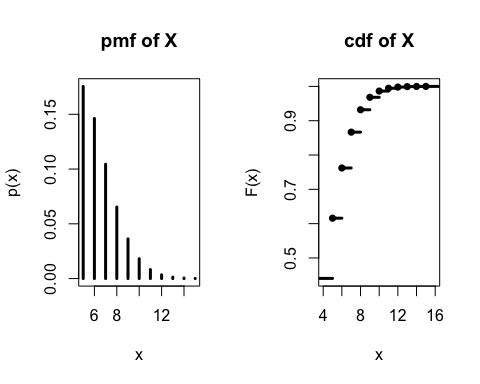
Suppose Create a function which plots the pmf and cdf for the interval and calculate the and for a given value of

* Input: and
* Output: and
* Extra feature: stop the function if is negative and print a warning message.  
  [**40 marks**]

poisson <- function(x1, lambda, plot = FALSE) {  
 if (lambda <= 0) stop('lambda must be a positive number!')  
 p\_x <- dpois(x1, lambda)  
 F\_x <- ppois(x1, lambda)  
 if (plot == TRUE) {  
 par(mfrow = c(1, 2))  
 INTERVAL\_SIZE <- 5  
 L <- x1 - INTERVAL\_SIZE  
 U <- x1 + INTERVAL\_SIZE  
 x <- c(L : U)  
 pmf <- dpois(x, lambda)  
 plot(x, pmf, lwd = 3, type = 'h', main = 'pmf of X', ylab = 'p(x)')  
   
 cdf <- stepfun(x, c(ppois(L - 1, lambda), ppois(x, lambda)))  
 plot.stepfun(cdf,   
 verticals = FALSE,   
 do.points = TRUE,   
 pch = 16, lwd = 3,  
 main = 'cdf of X',  
 ylab = 'F(x)')  
 }   
 list(p\_x = p\_x,  
 F\_x = F\_x)  
}  
#poisson(10, 0) #This should throw an error  
poisson(10, 5) #Only returns p\_x and F\_x

## $p\_x  
## [1] 0.01813279  
##   
## $F\_x  
## [1] 0.9863047

poisson(10, 5, TRUE) #Prints plot and returns p\_x and F\_x



## $p\_x  
## [1] 0.01813279  
##   
## $F\_x  
## [1] 0.9863047