Assingment_1

Anonymous

05/09/2022

Task 1:

probability = mathematical way of describing how probable something is. It is denoted as the range 0 to 1 with 0 being impossible and 1 guaranteed.

probability mass = gives the probability of discrete random variable being exactly one value.

probability density = gives the probability of continuous random variable being within a range.

probability mass function = function that returns the probability mass of a discrete random variable.

probability density function = function when integrated over a range returns the the probability density of that range.

probability distribution function = a distribution that describes the probability of different outcomes.

discrete probability distribution = a probability distribution that only has discrete outcomes.

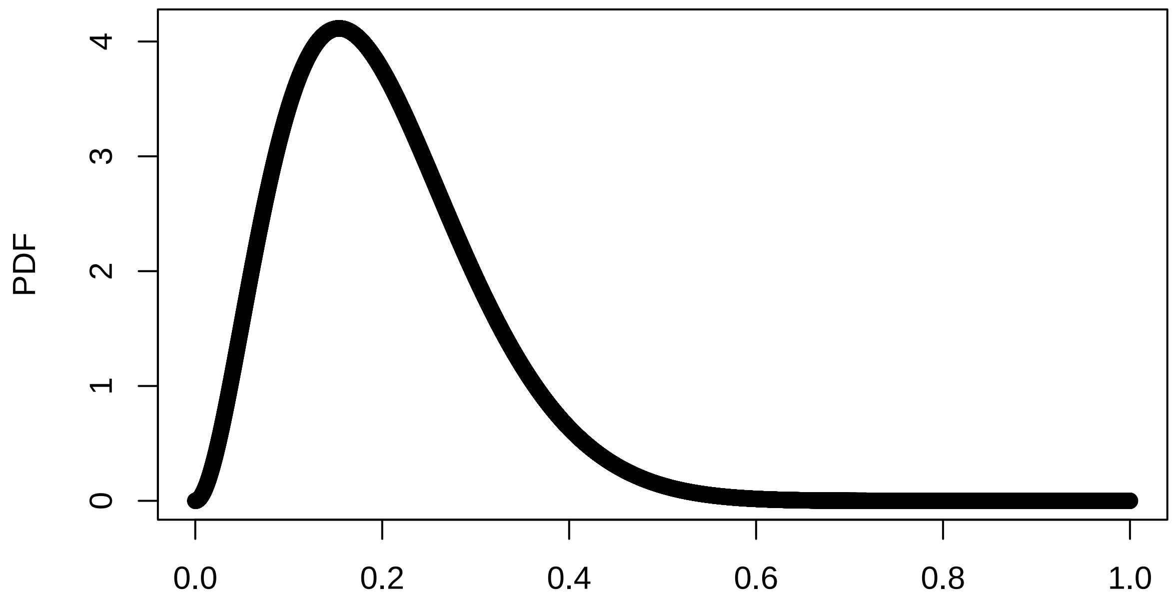
continuous probability distribution = a probability distribution that has a continuous set of outcomes.

cumulative distribution function = it gives a function that gives the probability of random variable being equal or smaller to a given value. It has the range 0 - 1.

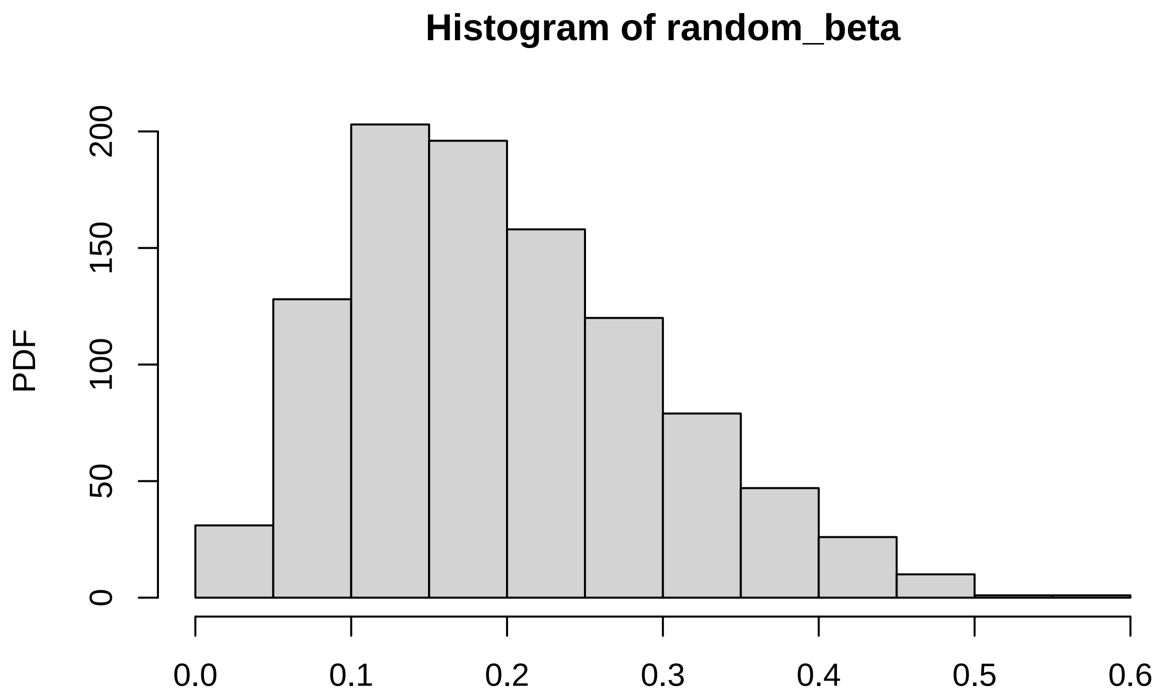
likelihood = non-normalized probability distribution

Task 2:

```
mu <- 0.2
delta_2 <- 0.01
alpha <- mu*(mu*(1-mu)/(delta_2)-1)
beta <- alpha*(1-mu)/mu
x<- seq(0,1,length.out = 10000)
plot(x,dbeta(x,alpha,beta),ylab = "PDF", xlab="")
```




```
random_beta <- rbeta(1000,alpha,beta)
hist(random_beta,ylab = "PDF", xlab="")
```




```
calc_var = var(random_beta)
calc_mean = mean(random_beta)
print(calc_var)
```

```
## [1] 0.009768922
```

```
print(calc_mean)
```

```
## [1] 0.1975865
```

```
quantile(random_beta, probs = c(0.025,0.975))
```

```
##       2.5%      97.5%
## 0.04630759 0.41555060
```

The above code gives the mean 0.19 and variance 0.010. The 2.5th quantile is 0.049 and the 97.5th quantile is 0.41

Task 3:

p(positiveTest|hasCancer) = 0.98 p(negativeTest|noCancer) = 0.96 p(hasCancer) = 1/1000

Showing chance of false negative:

p(hasCancer|negativeTest) = p(negativeTest|hasCancer)*p(hasCancer)/p(negativeTest) p(negativeTest|hasCancer) = 1 - p(postiveTest|hasCancer) = 0.02

p(negativeTest) = p(hasCancer) * p(negativeResult|hasCancer) + p(noCancer) * p(negativeResult|noCancer) = 1/1000 * 0.02 + 999/1000 * 0.96

=>

```
p_hasCancer_negativeTest <- 0.02*(1/1000)/(1/1000*0.02+999/1000*0.96)
print(p_hasCancer_negativeTest)
```

```
## [1] 2.085375e-05
```

Showing chance of false positive:

p(noCancer|positiveTest) =p(positiveTest|noCancer)*p(noCancer)/p(positiveTest)

p(positiveTest|noCancer) = 1-p(negativeTest|noCancer) = 0.04

p(positiveTest) = p(hasCancer) * p(positiveTest|hasCancer) + p(noCancer) * p(positiveTest|noCancer) =1/1000 * 0.98 + 999/1000 * 0.04

```
p_noCancer_positiveTest <- 0.04*(999/1000)/(0.98*1/1000+0.04*999/1000)
print(p_noCancer_positiveTest)
```

```
## [1] 0.9760625
```

As the rate of false positives is over 97% the test is useless

Task 4:

```
p_a <- 0.4
p_b <- 0.1
p_c <- 0.5

p_red <- function(boxes){
  A <- boxes[1,]
  B <- boxes[2,]
  C <- boxes[3,]
  p_red <- p_a*A[1]/(A[1]+A[2]) + p_b*B[1]/(B[1]+B[2])+p_c*C[1]/(C[1]+C[2])
  p_red
}

boxes <- matrix(c(2,4,1,5,1,3),ncol =2, dimnames = list(c("A","B","C"),c("red","white")))
boxes
```

```
##   red white
## A   2     5
## B   4     1
## C   1     3
```

```
p_red(boxes)
```

```
##       red
## 0.3192857
```

```r
p_box <- function(boxes){
  A <- boxes[1,]
  B <- boxes[2,]
  C <- boxes[3,]
  p_red <- p_red(boxes)
  p_red_a <- A[1]/(A[1]+A[2])
  p_red_b <- B[1]/(B[1]+B[2])
  p_red_c <- C[1]/(C[1]+C[2])
  c(p_red_a*p_a/p_red,p_red_b*p_b/p_red,p_red_c*p_c/p_red)
}

p_box(boxes)
```

```
##       red       red       red
## 0.3579418 0.2505593 0.3914989
```

Red ball is most probably from box c

Task 5: P(identicak | boy)

= p(boy | identical) * p(identical) / p(boy)

p(boy | identical) = 1 p(identical) = identical_prob/(identical_prob + fraternal_prob) p(fraternal) = fraternal_prob/(identical_prob + fraternal_prob) p(boy) = p(identical) + 0.5 * p(fraternal)

thus the function below calculates the odds of Elvis having an identical twin brother.

```r
p_identical_twin <- function(fraternal_prob,identical_prob){
  p_boy <- 0.5 * fraternal_prob/(identical_prob+fraternal_prob) + identical_prob/(identical_prob+fratern
  identical_prob/((identical_prob + fraternal_prob)*p_boy)
}
p_identical_twin(fraternal_prob = 1/150,identical_prob = 1/400)
```

```
## [1] 0.4285714
```

The odds are 43%