HW5

Manny Fomundam

2023-10-31

```r
# Define the possible outcomes and their probabilities
outcomes <- c(0, 1, 2, 3)
probabilities <- c(1/8, 3/8, 3/8, 1/8)

# Calculate the mean (expected value)
mean_X <- sum(outcomes * probabilities)

# Calculate the variance
var_X <- sum((outcomes - mean_X)^2 * probabilities)

mean_X
```

```
## [1] 1.5
```

```r
var_X
```

```
## [1] 0.75
```

```r
# Set the number of simulations
n_simulations <- 10000

# Simulate tossing three coins 10,000 times
simulations <- replicate(n_simulations, sum(sample(c(0, 1), 3, replace = TRUE)))

# Compute the simulated mean
simulated_mean <- mean(simulations)

# Compute the simulated variance
simulated_var <- var(simulations)

simulated_mean
```

```
## [1] 1.4989
```

```r
simulated_var
```

```
## [1] 0.7496738
```

```r
# Check if the simulated mean is within 2% of the theoretical mean
mean_within_2_percent <- abs((simulated_mean - mean_X) / mean_X) <= 0.02

# Check if the simulated variance is within 2% of the theoretical variance
var_within_2_percent <- abs((simulated_var - var_X) / var_X) <= 0.02

mean_within_2_percent
```

```
## [1] TRUE
```

```r
var_within_2_percent
```

```
## [1] TRUE
```

```r
# Define the values and their probabilities
values <- c(1, 2, 3, 4)
probabilities <- c(0.2, 0.3, 0.1, 0.4)

# Calculate E[X]
expected_X <- sum(values * probabilities)
expected_X
```

```
## [1] 2.7
```

```r
# Calculate 1/E[X]
inverse_expected_X <- 1 / expected_X
inverse_expected_X
```

```
## [1] 0.3703704
```

```r
# Calculate E[1/X]
expected_1_over_X <- sum((1 / values) * probabilities)
expected_1_over_X
```

```
## [1] 0.4833333
```

```r
# Calculate E[X^2]
expected_X_squared <- sum((values^2) * probabilities)
expected_X_squared
```

```
## [1] 8.7
```

```r
# Calculate E[X]^2
expected_X_squared_empirical <- (sum(values * probabilities))^2
expected_X_squared_empirical
```

```
## [1] 7.29
```

```r
# Check if E[X]^2 is equal to E[X^2]
equality1 <- expected_X_squared_empirical == expected_X_squared

# Check if E[1/X] is equal to 1/E[X]
equality2 <- expected_1_over_X == inverse_expected_X

equality1
```

```
## [1] FALSE
```

```r
equality2
```

```
## [1] FALSE
```

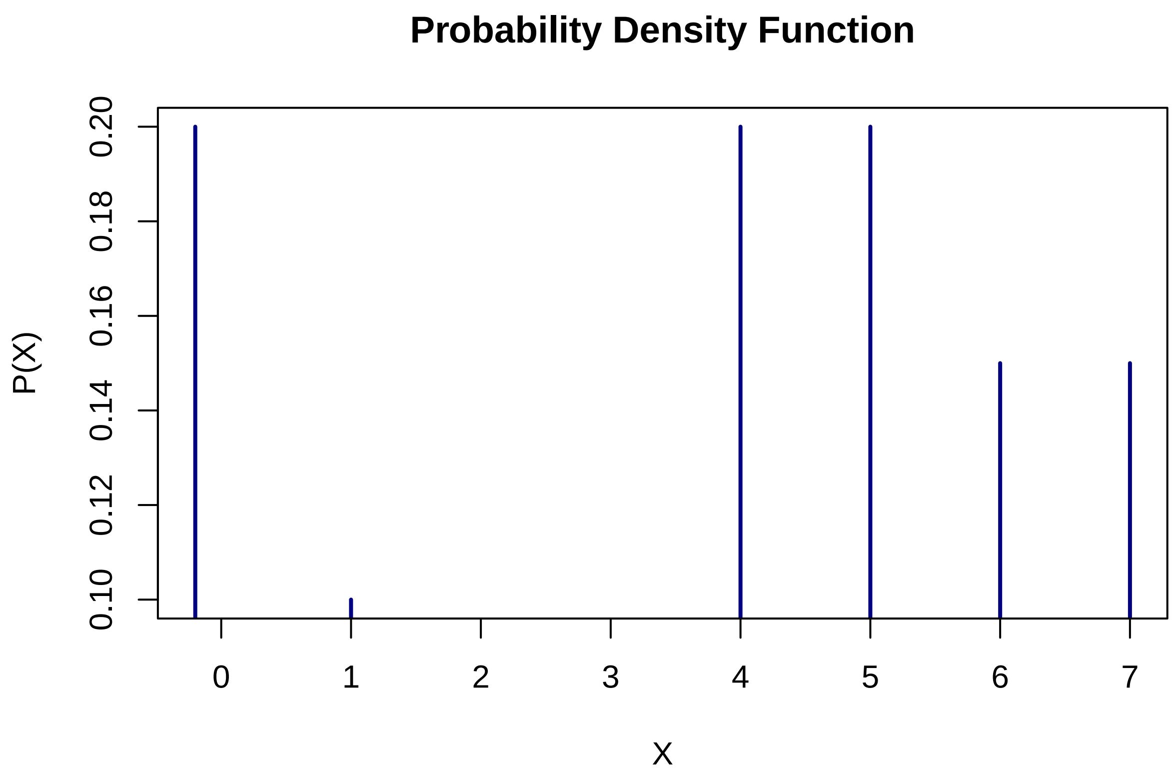
```r
# Define the distribution
x <- c(-0.2, 1, 4, 5, 6, 7)
p <- c(0.2, 0.1, 0.2, 0.2, 0.15, 0.15)

# Check if it's a probability function
if(sum(p) == 1){
```

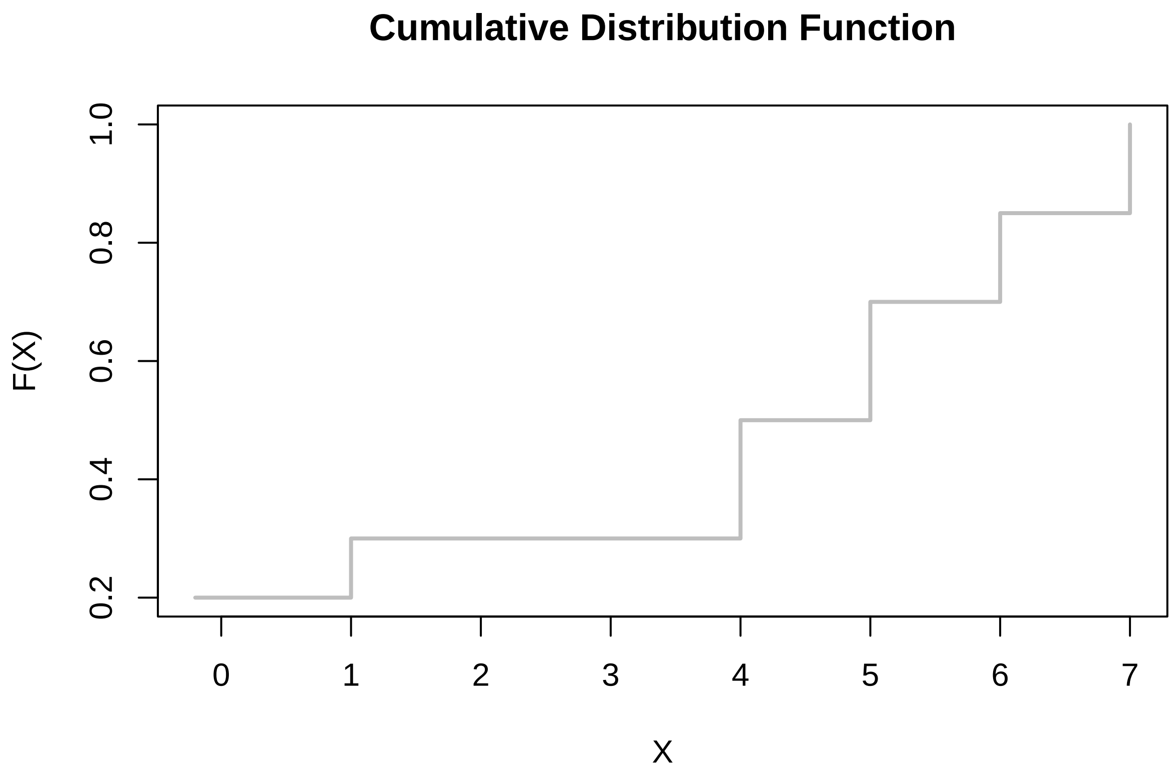
```r
  print("The distribution is a probability function.")
} else {
  print("The distribution is not a probability function.")
}
```

```
## [1] "The distribution is a probability function."
```

```r
# Plot the PDF
plot(x, p, type = "h", lwd = 2, col = "navy", xlab = "X", ylab = "P(X)", main = "Probability Density Fu
```




```r
# Obtain and plot the CDF
cdf <- cumsum(p)
plot(x, cdf, type = "s", lwd = 2, col = "gray", xlab = "X", ylab = "F(X)", main = "Cumulative Distribut
```




```r
#Function A
# Define the integral equation and solve for k
k_a <- 1 / (integrate(function(x) x^(4)/5, lower = 0, upper = 1)$value)
k_a
```

```
## [1] 25
```

```r
#Function B
# Define the integral equation and solve for k
k_b <- 1 / (integrate(function(x) x^2, lower = 0, upper = 2)$value)
k_b
```

```
## [1] 0.375
```

```r
#Function C
# Define the integral equation and solve for k
k_c <- 1 / (integrate(function(x) sqrt(x)/2, lower = 0, upper = 1)$value)
k_c
```

```
## [1] 2.999999
```

```r
# (Example problem to help)
# Define the interval
a <- 2
b <- 6

# Generate a uniform random variable
X <- runif(10000, min=a, max=b)
```

```r
# Calculate the expected value (mean)
E_X <- mean(X)
print(paste("E[X] = ", E_X))
```

```
## [1] "E[X] =  4.00568534658374"
```

```r
# Calculate the variance
Var_X <- var(X)
print(paste("Var(X) = ", Var_X))
```

```
## [1] "Var(X) =  1.34230108036966"
```