Nithin S
221IT085

IT302 Endsem Lab Exam

Q1 a) . Suppose a dice is rolled 6 times with probabilities of getting one, two, three, four, five, six are 0.06, 0.19, 0.17, 0.20, 0.22, 0.16, respectively. Write a R/Python code to find the probability that sum of the number is less than 22

Code

```
# Create function to generate all possible combinations
get_dice_prob <- function() {
  p1 <- c(0.06, 0.19, 0.17, 0.20, 0.22, 0.16)
  outcomes <- 1:6
  total_prob <- 0
  for(i in 1:6) {
    for(j in 1:6) {
      for(k in 1:6) {
        for(l in 1:6) {
          for(m in 1:6) {
            for(n in 1:6) {
              sum_dice <- i + j + k + l + m + n
              if(sum_dice < 22) {
                prob <- p1[i] * p1[j] * p1[k] * p1[l] * p1[m] * p1[n]
                total_prob <- total_prob + prob
              }
            }
          }
        }
      }
    }
  }
  return(total_prob)
}

dice_probability <- get_dice_prob()
cat("\n(a) Probability that sum of dice rolls is less than 22:", dice_probability, "\n")
```

Output

```
> dice_probability <- get_dice_prob()
> cat("\n(a) Probability that sum of dice rolls is less than 22:", dice_probability,
"\n")

(a) Probability that sum of dice rolls is less than 22: 0.3565126
>
```

Q1. (b) Suppose that a new born baby is a boy, is 0.5 and a girl is 0.5. Suppose further that the probability that a new born baby is blue eyed is 0.25 and black eyed is, 0.75. If 5 new-born babies are selected at random, write a R/Python code to find probability that there will be exactly 3, blue eyed girls and 2 black eyed boys.

Code

```
p_girl<-0.5
p_boy<-0.5
p_blue_eyed<-0.25
p_black_eyed<-0.75
p_girl_blue <- p_girl* p_blue_eyed
p_girl_blue
p_boy_black <- p_boy * p_black_eyed
p_boy_black
prob_3_blue_girls <- dbinom(3, size = 5, prob = p_girl_blue)
prob_2_black_boys <- dbinom(2, size = 5, prob = p_boy_black)
combined_probability <- prob_3_blue_girls * prob_2_black_boys
cat("The probability of having exactly 3 blue-eyed girls and exactly 2 black-eyed boys among 5 newborns is:", combined_probability, "\n")
```

Output

```
> combined_probability <- prob_3_blue_girls * prob_2_black_boys
> cat("The probability of having exactly 3 blue-eyed girls and exactly 2 black-eyed
oys among 5 newborns is:", combined_probability, "\n")
The probability of having exactly 3 blue-eyed girls and exactly 2 black-eyed boys am
ng 5 newborns is: 0.005133916
>
```

Q2. The repair time (in hours) for an industrial machine has a gamma distribution with mean 1.5 and variance 0.75. Write a R/Python code to determine the probability that [1.5+1.5+1+1] (a) the repair time exceeds 1.2 hours. (b) the repair time is at least 4 hours given that it already exceeds 1.2 hours. (c) Plot the p.d.f. of the given gamma distribution. (d) Plot the c.d.f. of the given gamma distribution

Code

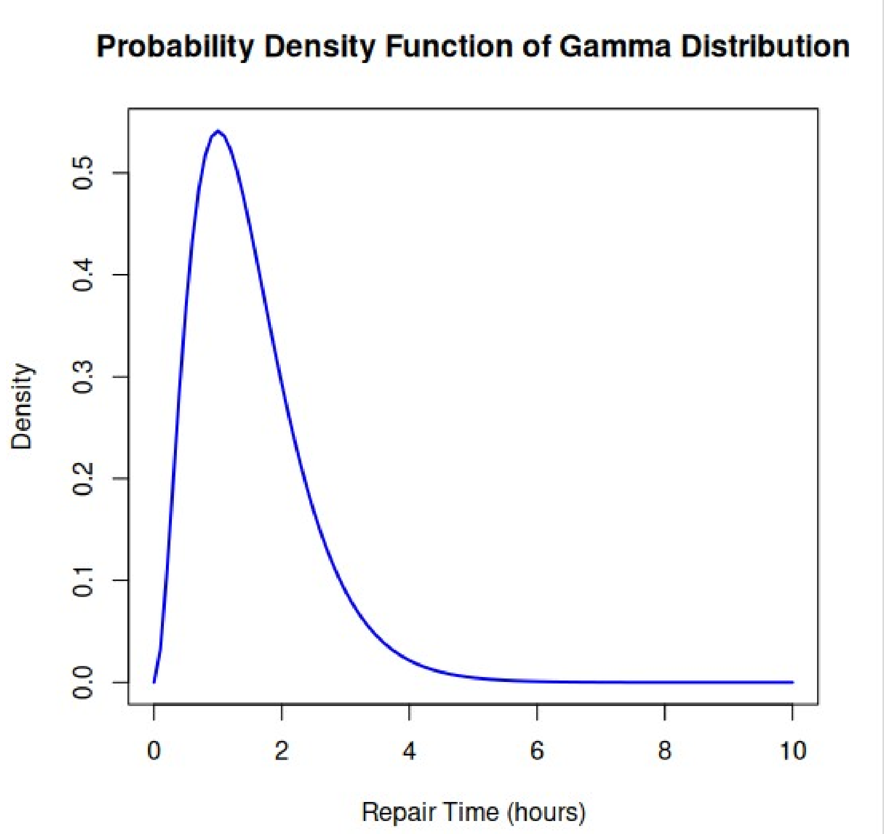
```
mean <- 1.5
variance <- 0.75
beta <- variance / mean
alpha <- mean / beta

# a) P(X > 1.2)
prob_exceeds_1.2 <- 1 - pgamma(1.2, shape = alpha, rate = 1 / beta)
cat("a) Probability that repair time exceeds 1.2 hours:", prob_exceeds_1.2, "\n")

# b) P(X > 4 | X > 1.2)
prob_exceeds_4 <- 1 - pgamma(4, shape = alpha, rate = 1 / beta)
prob_at_least_4_given_exceeds_1.2 <- prob_exceeds_4 / prob_exceeds_1.2
```

```
cat("b) Probability that repair time is at least 4 hours given it exceeds 1.2 hours:",
prob_at_least_4_given_exceeds_1.2, "\n")

# (c) Plot the p.d.f. of the gamma distribution
x_values <- seq(0, 10, by = 0.1)
pdf_values <- dgamma(x_values, shape = alpha, rate = 1 / beta)

plot(x_values, pdf_values, type = "l", col = "blue", lwd = 2,
    main = "Probability Density Function of Gamma Distribution",
    xlab = "Repair Time (hours)", ylab = "Density")

# (d) Plot the c.d.f. of the gamma distribution
cdf_values <- pgamma(x_values, shape = alpha, rate = 1 / beta)

plot(x_values, cdf_values, type = "l", col = "red", lwd = 2,
    main = "Cumulative Distribution Function of Gamma Distribution",
    xlab = "Repair Time (hours)", ylab = "Cumulative Probability")
```

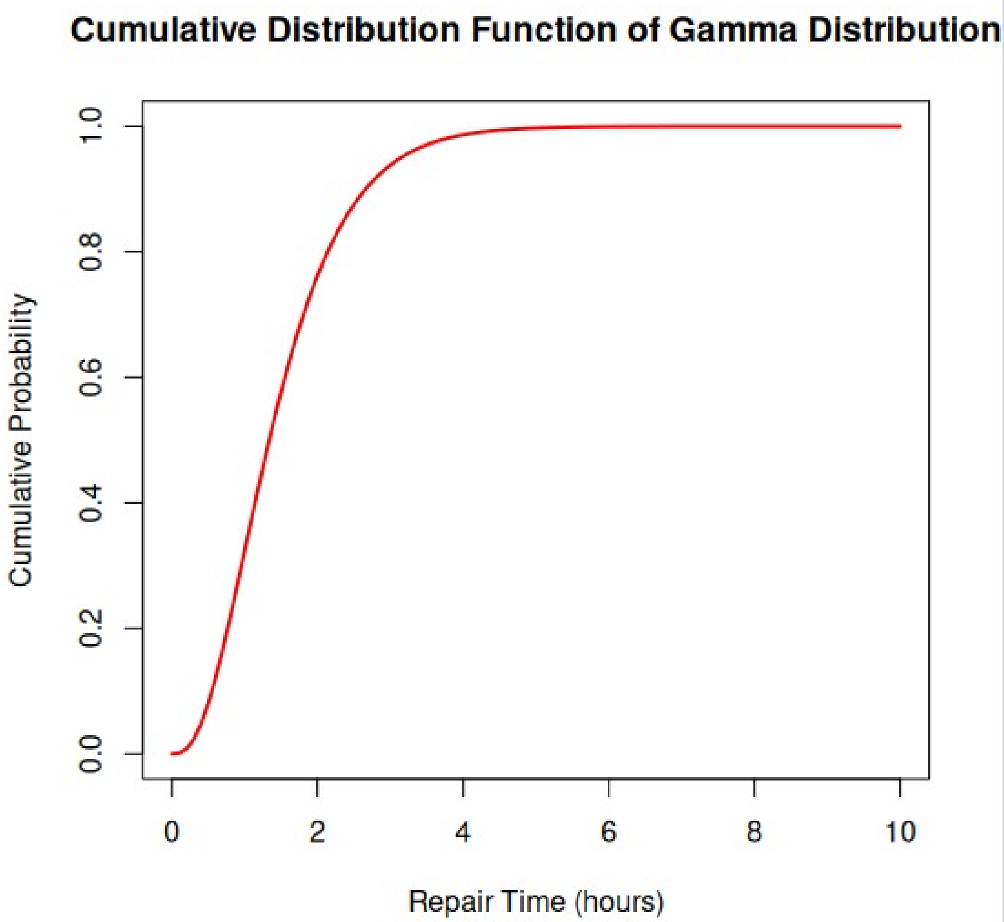
Output

```
> cat("a) Probability that repair time exceeds 1.2 hours:", prob_exceeds_1.2, "\n")
a) Probability that repair time exceeds 1.2 hours: 0.5697087
>
> # b) P(X > 4 | X > 1.2)
> prob_exceeds_4 <- 1 - pgamma(4, shape = alpha, rate = 1 / beta)
> prob_at_least_4_given_exceeds_1.2 <- prob_exceeds_4 / prob_exceeds_1.2
> cat("b) Probability that repair time is at least 4 hours given it exceeds 1.2 hours:", prob_at_least_4_given_exceeds_1.2, "\n")
b) Probability that repair time is at least 4 hours given it exceeds 1.2 hours: 0.0241421
>
```

c)



d)


Cumulative Distribution Function of Gamma Distribution
Cumulative Probability
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
Repair Time (hours)

Q3. The marks (package “bnlearn”) data set contains the examination marks of 88 students on five different subjects. [2.5+2.5] (a) The variable STAT contains marks of the statistics subject of the students. Write a R/Python code to decide whether the average mark of the statistics subject is less than 38 at a 5% level of significance.

H0: $\mu \geq 38$
H1: $\mu < 38$
$\alpha = 0.05$

Code

```
library(bnlearn)
data(marks)

stat_mean <- mean(marks$STAT)
stat_sd <- sd(marks$STAT)
stat_n <- length(marks$STAT)
stat_se <- stat_sd/sqrt(stat_n)
stat_z <- (stat_mean - 38)/stat_se
alpha<-0.05
z_critical <- qnorm(0.05)
stat_pvalue <- pnorm(stat_z)
cat("\nResults for Statistics Marks (Z-test):\n")
cat("Sample Size =", stat_n, "\n")
cat("Sample Mean =", stat_mean, "\n")
cat("Sample SD =", stat_sd, "\n")
cat("Standard Error =", stat_se, "\n")
cat("Z-statistic =", stat_z, "\n")
cat("Critical value =", z_critical, "\n")
cat("P-value =", stat_pvalue, "\n")
cat("Decision: ", ifelse(stat_pvalue < alpha,
                  "Reject H0",
                  "Fail to reject H0"), "\n")
```

Output

```
> library(bnlearn)
> data(marks)
>
> stat_mean <- mean(marks$STAT)
> stat_sd <- sd(marks$STAT)
> stat_n <- length(marks$STAT)
> stat_se <- stat_sd/sqrt(stat_n)
> stat_z <- (stat_mean - 38)/stat_se
> alpha<-0.05
> z_critical <- qnorm(0.05)
> stat_pvalue <- pnorm(stat_z)
> cat("\nResults for Statistics Marks (Z-test):\n")

Results for Statistics Marks (Z-test):
> cat("Sample Size =", stat_n, "\n")
Sample Size = 88
> cat("Sample Mean =", stat_mean, "\n")
Sample Mean = 42.30682
> cat("Sample SD =", stat_sd, "\n")
Sample SD = 17.25559
> cat("Standard Error =", stat_se, "\n")
Standard Error = 1.839452
> cat("Z-statistic =", stat_z, "\n")
Z-statistic = 2.341359
> cat("Critical value =", z_critical, "\n")
Critical value = -1.644854
> cat("P-value =", stat_pvalue, "\n")
P-value = 0.9903932
> cat("Decision: ", ifelse(stat_pvalue < alpha,
+                          "Reject H0",
+                          "Fail to reject H0"), "\n")
Decision:  Fail to reject H0
> |
```

The average marks is greater than or equal to 38

(b) The variable MECH contains marks of the mechanics subject of the students. Consider only the first 25 students' mechanics marks. Write a R/Python code to decide whether the average mark of those 25 students is greater than 56 at a 10% level of significance

$H0: \mu \leq 56$
$H1: \mu > 56$
$\alpha = 0.10$

Code

```
mech_25 <- marks$MECH[1:25]
mech_25
mech_mean <- mean(mech_25)
mech_sd <- sd(mech_25)
mech_n <- length(mech_25)
mech_se <- mech_sd/sqrt(mech_n)
mech_t <- (mech_mean - 56)/mech_se
alpha<-0.10
mech_critical <- qt(0.90, df = mech_n - 1)
```

```
mech_pvalue <- 1 - pt(mech_t, df = mech_n - 1)
cat("\nResults for Mechanics Marks (t-test):\n")
cat("Sample Size =", mech_n, "\n")
cat("Sample Mean =", mech_mean, "\n")
cat("Sample SD =", mech_sd, "\n")
cat("Standard Error =", mech_se, "\n")
cat("t-statistic =", mech_t, "\n")
cat("Critical value =",mech_critical, "\n")
cat("P-value =", mech_pvalue, "\n")
cat("Decision: ", ifelse(mech_pvalue < alpha,
                         "Reject H0",
                         "Fail to reject H0"), "\n")
```

Output

```
> mech_25 <- marks$MECH[1:25]
> mech_25
 [1] 77 63 75 55 63 53 51 59 62 64 52 55 50 65 31 60 44 42 62 31 44 49 12 49 54
> mech_mean <- mean(mech_25)
> mech_sd <- sd(mech_25)
> mech_n <- length(mech_25)
> mech_se <- mech_sd/sqrt(mech_n)
> mech_t <- (mech_mean - 56)/mech_se
> alpha<-0.10
> mech_critical <- qt(0.90, df = mech_n - 1)
> mech_pvalue <- 1 - pt(mech_t, df = mech_n - 1)
> cat("\nResults for Mechanics Marks (t-test):\n")

Results for Mechanics Marks (t-test):
> cat("Sample Size =", mech_n, "\n")
Sample Size = 25
> cat("Sample Mean =", mech_mean, "\n")
Sample Mean = 52.88
> cat("Sample SD =", mech_sd, "\n")
Sample SD = 14.08107
> cat("Standard Error =", mech_se, "\n")
Standard Error = 2.816215
> cat("t-statistic =", mech_t, "\n")
t-statistic = -1.10787
> cat("Critical value =",mech_critical, "\n")
Critical value = 1.317836
> cat("P-value =", mech_pvalue, "\n")
P-value = 0.8605511
> cat("Decision: ", ifelse(mech_pvalue < alpha,
+                          "Reject H0",
+                          "Fail to reject H0"), "\n")
Decision:  Fail to reject H0
> 
```

The average marks is less than equal to 56