BT307 Biological Data Analysis   Lab Assignment No. 4   15.02.2024

Note: Submit the assignment online through Moodle either in .doc or .pdf format. Your final report file should be named as "**YourName_BT307_Lab4_15022024**". Make sure that your name and roll numbers are written at the first page of your final report. Note that you can upload only one file; thus, put together all the answers in a single file.

Goal of this exercise is to learn about the main probability distributions in R.

(A) ########## Binomial Probability Distribution ################################

(1) # dbinom is the R function that calculates the PMF of the binomial distribution.
 # dbinom(k, n, p). For example, P(X = 27)=?

```
dbinom(27, size=100, prob=0.25)
probabilities <- dbinom(x = c(0:10), size = 10, prob = 1 / 6)
data.frame(probabilities)
plot(0:10, probabilities, type = "l")
```

(2) # pbinom is the R function that calculates the CDF of the binomial distribution.
 # pbinom(k, n, p). For example, P(X <= 27) = ?

```
pbinom(27, size=100, prob=0.25)
plot(0:10, pbinom(0:10, size = 10, prob = 1 / 6), type = "l")
```

(3) # qbinom is the R function that calculates the "inverse CDF" of the binomial distribution. qbinom(P, n, p).

```
qbinom(0.8419226, size = 13, prob = 1 / 6)
x <- seq(0, 1, by = 0.1)
y <- qbinom(x, size = 13, prob = 1 / 6)
plot(x, y, type = 'l')
```

(4) # rbinom() function generates n random variables of a particular probability.
rbinom(n, N, p).

```
rbinom(8, size = 13, prob = 1 / 6)
hist(rbinom(8, size = 13, prob = 1 / 6))
```

(B) ############# Multinomial Probability Distribution ########################

(5) # Generate 1000 samples from a multinomial distribution with 3 trials and
 # unequal probabilities for each outcome

```
set.seed(123)
samples <- rmultinom(n = 1000, size = 3, prob = c(0.3, 0.4, 0.3))
tab <- table(samples)
pie(prop.table(tab),
   labels = paste0(names(tab), ": ", round(prop.table(tab) * 100, 1), "%"),
   main = "Multinomial Distribution")
```

(C)############ Poisson Probability Distribution############################

(6) # dpois() function calculates the PMF for a given Poisson distribution
 # Plot PMF for Poisson distribution with mean of 2.5 for x = 0 to 10

```
x <- seq(0, 10, by = 1)
pmf <- dpois(x, 2.5)
plot(x, pmf, type = "h", lwd = 3, main = "Poisson PMF", xlab = "Number of events",
  ylab = "Probability")
```

(7) # ppois() function calculates the CDF for a Poisson distribution.
 # Plot CDF for Poisson distribution with mean of 2.5 for q = 0 to 10

```
q <- seq(0, 10, by = 1)
cdf <- ppois(q, 2.5)
plot(q, cdf, type = "s", lwd = 3, main = "Poisson CDF", xlab = "Number of events",
  ylab = "Probability")
```

(8) # qpois() function calculates the quantiles of a Poisson distribution.
 # Plot ICDF for Poisson distribution with mean of 2.5 for p = 0.1 to 0.9

```
p <- seq(0.1, 0.9, by = 0.1)
icdf <- qpois(p, 2.5)
plot(p, icdf, type = "o", pch = 19, lwd = 3, main = "Poisson ICDF",
  xlab = "Probability", ylab = "Number of events")
```

(9) # rpois() function generates random numbers from a Poisson distribution.

```
x <- 0:10
pmf <- dpois(x, lambda = 3)
plot(x, pmf, type = "h", lwd = 2, col = "blue", xlab = "x", ylab = "P(X = x)",
  main = "Poisson PMF for lambda = 3")
```

(D)############ Normal Probability Distribution############################

(10) # dnorm() function computes the PDF of the normal distribution

```
x <- seq(-4, 4, length.out = 100)
y <- dnorm(x, mean = 0, sd = 1)
plot(x, y, type = "l")
```

(11) # pnorm() function computes the CDF of the normal distribution

```
x <- seq(-3, 3, length.out = 100)
plot(x, pnorm(x), type = "l", lty = 1,
    xlab = "x", ylab = "Cumulative Probability",
    main = "CDF of Standard Normal Distribution")
legend("topleft", legend = c("Mean = 0", "SD = 1"),
      lty = 1, col = 1, bg = "white")
```

(12) # The qnorm() function calculates the quantiles of the normal distribution.

```
set.seed(123)
x <- rnorm(100)
plot(qnorm(seq(0.01, 0.99, length.out = 100)), sort(x),
    xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
    main = "Normal Probability Plot")
```

(13) # rnorm() function generates random numbers from a normal distribution with a specified mean and standard deviation.

```
data <- rnorm(1000, mean = 0, sd = 1)
hist(data, main = "Normal Distribution", xlab = "Data", ylab = "Frequency")
```

(E) ############ Exponential Probability Distribution#########################

(14) # Generate 1000 random values from an Exp. distribution with rate = 0.5

```
set.seed(123)
values <- rexp(n = 1000, rate = 0.5)
hist(values, freq = FALSE, main = "Exponential Distribution",
  xlab = "Values", ylab = "Density")
curve(dexp(x, rate = 0.5), add = TRUE, col = "red", lwd = 2)
```

(F) ############ Continuous Uniform Probability Distribution################

(15) # Generate the values

```
random_numbers <- runif(10, min = 0, max = 1)
print(random_numbers)

pdf <- dunif(seq(-1, 2, by = 0.1), min = 0, max = 1)
print(pdf)

cdf <- punif(seq(-1, 2, by = 0.1), min = 0, max = 1)
print(cdf)

quantiles <- qunif(seq(0, 1, by = 0.1), min = 0, max = 1)
print(quantiles)
```

(16) # Plot the values

```
library(ggplot2)
pdf <- dunif(seq(-1, 2, by = 0.01), min = 0, max = 1)
cdf <- punif(seq(-1, 2, by = 0.01), min = 0, max = 1)
data <- data.frame(x = seq(-1, 2, by = 0.01), pdf = pdf, cdf = cdf)

pdf_plot <- ggplot(data, aes(x = x, y = pdf)) +
  geom_line(color = "blue") +
  labs(title = "Probability Density Function (PDF)", x = "x", y = "Density") +
  theme_minimal()

cdf_plot <- ggplot(data, aes(x = x, y = cdf)) +
  geom_line(color = "red") +
  labs(title = "Cumulative Distribution Function (CDF)",
       x = "x", y = "Probability") +
  theme_minimal()
pdf_plot
cdf_plot
```

(G)################### Central Limit Theorem ##############################

```
set.seed(42)
population <- runif(5000, min = 0, max = 1)
par(mfrow = c(1, 2))
hist(population, breaks = 30, prob = TRUE, main = "Population Distribution",
     xlab = "Value", col = "lightblue")

sample_size <- 30
num_samples <- 300

sample_means <- c()

for (i in 1:num_samples) {
  sample <- sample(population, size = sample_size, replace = TRUE)
  sample_means[i] <- mean(sample)}

x_bar <- mean(sample_means)
std <- sd(sample_means)
print('Sample Mean and Variance')
print(x_bar)
print(std**2)

mu <- mean(population)
sigma <- sd(population)
print('Population Mean and Variance')
print(mu)
print((sigma**2)/sample_size)

hist(sample_means, breaks = 30, prob = TRUE, main = "Distribution of Sample Means",
     xlab = "Sample Mean", col = "lightgreen")

curve(dnorm(x, mean = x_bar, sd = std), col = "black", lwd = 2, add = TRUE)

legend("topright", legend = c("Distribution Curve"),
       col = c("black"), lwd = 2)

par(mfrow = c(1, 1))
```