**Experiment 1: Discrete Uniform**

**Distribution in R**

**1a.Random Selection of IPL Teams**

Code:

teams <- c("Mumbai Indians", "Chennai Super Kings", "Royal Challengers Bangalore",

 "Kolkata Knight Riders", "Delhi Capitals", "Sunrisers Hyderabad","Rajasthan Royals", "Punjab Kings")

set.seed(1)

selected\_team <- sample(teams, 1)

cat(“The selected team is given by”,selected\_team)

**1b: Random Selection of Bollywood Movie Genres**

**Code:**

*genres <- c("Action", "Comedy", "Drama", "Thriller", "Romance")*

*set.seed(2)*

*selected\_genre <- sample(genres, 1)*

*print(selected\_genre)*

**1c: Random Selection of Marvel Comic Book Series**

*# List of Marvel comic series*

*series <- c("The Amazing Spider-Man", "X-Men", "Avengers", "Fantastic Four", "Black Panther")*

*# Randomly select a series*

*set.seed(3)*

*selected\_series <- sample(series, 1)*

*print(selected\_series)*

**2: Bernoulli and Binomial Distributions**

**2a:** Batsman's Success in Scoring a Half-Century

Code:

*runs <- c(55, 45, 60, 30, 75)*

*threshold <- 50*

*success <- ifelse(runs >= threshold, 1, 0)*

*p <- mean(success)*

*print(p)*

**2b:** Number of Matches a Bowler Takes at Least One Wicket

Code:

*wickets <- c(2, 0, 1, 3, 0, 1, 2, 0, 1, 4)*

*success <- ifelse(wickets > 0, 1, 0)*

*n <- length(wickets)*

*p <- mean(success)*

*set.seed(4)*

*simulated\_matches <- rbinom(1, n, p)*

*print(simulated\_matches)*

**2c:** Probability of a Movie Being a Box Office Hit

Code: *# Dataset*

*hits <- c(1, 1, 0, 1, 0, 1, 0, 0, 1, 1,*

*0, 1, 0, 0, 1, 1, 1, 1, 1, 0,*

*0, 1, 1, 0, 0, 1, 1, 1, 1, 1)*

*p\_hat <- mean(hits)*

*p\_hat*

*dbinom(18, size = 30, prob = p\_hat)*

*x <- 0:30*

*prob\_x <- dbinom(x, size=30, prob=p\_hat)*

*library(ggplot2)*

*df <- data.frame(Hits=x, Probability=prob\_x)*

*ggplot(df, aes(x=Hits, y=Probability)) +  geom\_bar(stat="identity", fill="darkgreen") +*

*ggtitle("Binomial Distribution of Box Office Hits") +   xlab("Number of Hits in 30 Movies") +*

*ylab("Probability")*

**EXP3: Poisson and Geometric Distributions**

**Problem 1:** Modeling Character Appearances Using Poisson Distribution

*appearances <- c(1500, 1200, 1100, 1000, 900, 850, 1400, 1300, 800, 700,*

*600, 550, 750, 650, 500, 400, 350, 300, 450, 400,*

*380, 370, 360, 800, 600, 700, 200, 180, 160, 140)*

*lambda <- mean(appearances)*

*library(ggplot2)*

*x <- 0:max(appearances)*

*poisson\_dist <- dpois(x, lambda)*

*qplot(x, poisson\_dist, geom="line", main="Poisson Distribution of Character Appearances",*

*xlab="Number of Appearances", ylab="Probability")*

**Problem 2: Time Until First Appearance Using Geometric Distribution**

*library(ggplot2)*

*first\_appearance <- c(1, 3, 2, 4, 5, 6, 1, 2, 7, 3,*

*4, 5, 2, 6, 7, 8, 9, 10, 3, 4,*

*5, 6, 7, 2, 3, 4, 8, 9, 10, 11)*

*p <- 1 / mean(first\_appearance)*

*x <- 1:max(first\_appearance)*

*geometric\_dist <- dgeom(x - 1, p)*

*qplot(x, geometric\_dist, geom="line", main="Geometric Distribution of Time Until First Appearance",*

*xlab="Issues Until First Appearance", ylab="Probability")*

**Problem 3:** Modeling Movie Release Frequencies Using Poisson Distribution

*movies\_released <- c(95, 102, 110, 98, 105, 115, 120, 125, 118, 122, 100, 97, 110, 130)*

*lambda <- mean(movies\_released)*

*x\_vals <- min(movies\_released):max(movies\_released)*

*pois\_probs <- dpois(x\_vals, lambda)*

*library(ggplot2)*

*df <- data.frame(Movies=x\_vals, Probability=pois\_probs)*

*ggplot(df, aes(x=Movies, y=Probability)) +*

*geom\_line(color="blue") +*

*ggtitle("Poisson Distribution of Bollywood Movie Releases") +*

*xlab("Number of Movies Released") +*

*ylab("Probability")*

**Experiment 4: Negative Binomial and Hypergeometric Distributions**

**Problem 1: Modeling Failures Before Success Using Negative Binomial (Cricket IPL Data)**

*matches <- 100*

*half\_centuries <- 20*

*p\_success <- half\_centuries / matches*

*r\_success <- 5  # aiming for 5 half-centuries*

*failures <- 0:20*

*prob\_nb <- dnbinom(failures, size=r\_success, prob=p\_success)*

*library(ggplot2)*

*df <- data.frame(Failures=failures, Probability=prob\_nb)*

*ggplot(df, aes(x=Failures, y=Probability)) +*

*geom\_bar(stat="identity", fill="steelblue") +*

*ggtitle("Negative Binomial: Failures before 5 Half-centuries (Virat)") +*

*xlab("Failures before 5 successes") + ylab("Probability")*

**Problem 2: Modeling Sampling Without Replacement (Marvel Dataset – Hypergeometric)**

*N <- 50     # Total characters*

*K <- 20     # Number of Avengers*

*n <- 10     # Characters selected*

*k <- 4      # Avengers selected*

*phyper\_val <- dhyper(k, K, N - K, n)*

*phyper\_val  # print result*

**Problem 3: Comparing Observed vs. Expected Outcomes in Sports (Hypergeometric Application)**

*total\_players <- 15*

*total\_bowlers <- 8*

*total\_batsmen <- 7*

*playing\_11 <- 11*

*bowlers\_selected <- 6*

*probability <- dhyper(bowlers\_selected, total\_bowlers, total\_batsmen, playing\_11)*

*probability*

**Experiment 5: Application of Discrete Probability Distributions in Data Science**

**Problem 1: Modeling IPL Player Dismissals Using the Poisson Distribution**

*# Dataset*

*dismissals <- c(0,1,1,0,2,1,0,0,1,0,1,2,0,1,0,1,1,2,0,0,1,1,0,0,1,0,1,0,2,1)*

*# Mean rate (lambda)*

*lambda <- mean(dismissals)*

*# Table of observed frequencies*

*table\_obs <- table(dismissals)*

*# Expected Poisson probabilities*

*expected\_probs <- dpois(as.numeric(names(table\_obs)), lambda)*

*expected\_freq <- expected\_probs \* length(dismissals)*

*# Chi-square goodness of fit test*

*chisq.test(x = as.numeric(table\_obs), p = expected\_probs, rescale.p = TRUE)*

**Problem 2: Modeling Marvel Comic Appearances Using the Binomial Distribution**

*# Dataset*

*lead\_appearances <- c(5,7,6,4,5,6,8,4,7,6,5,6,7,3,5,6,5,8,7,5,6,7,4,5,5,6,4,5,5,6)*

*# Parameters*

*n <- 10*

*p\_hat <- mean(lead\_appearances)/n*

*# Table of observed frequencies*

*obs\_table <- table(lead\_appearances)*

*# Expected binomial probabilities*

*expected\_probs <- dbinom(as.numeric(names(obs\_table)), n, p\_hat)*

*expected\_freq <- expected\_probs \* length(lead\_appearances)*

*# Chi-square test*

*chisq.test(x = as.numeric(obs\_table), p = expected\_probs, rescale.p = TRUE)*

**Problem 3: Box Office Hit Probability in Bollywood Movies Using Bernoulli Distribution**

*# Dataset*

*hit\_status <- c(1,0,1,1,0,1,1,0,0,1,1,1,0,1,0,1,1,0,1,1,0,1,1,1,0,0,1,1,1,0)*

*# Estimated probability of hit*

*p\_hat <- mean(hit\_status)*

*# Frequency table*

*table(hit\_status)*

*# Expected frequencies*

*expected\_freq <- c((1 - p\_hat), p\_hat) \* length(hit\_status)*

*# Chi-square goodness of fit test*

*chisq.test(table(hit\_status), p = expected\_freq/length(hit\_status), rescale.p = TRUE)*

**Experiment 6: Uniform and Exponential Distributions**

**Problem 1: Modeling Random Number Generation with Uniform Distribution**

*set.seed(123)*

*uniform\_data <- runif(1000, min = 0, max = 1)*

*# Plotting*

*hist(uniform\_data, breaks = 20, main = "Uniform Distribution", xlab = "Value", col = "lightblue")*

**Problem 2: Modeling Waiting Times Using Exponential Distribution**

*set.seed(123)*

*lambda <- 1/5  # Average time between arrivals is 5 minutes*

*waiting\_times <- rexp(1000, rate = lambda)*

*# Plotting*

*hist(waiting\_times, breaks = 30, main = "Exponential Distribution of Waiting Times", xlab = "Time (minutes)", col = "lightgreen")*

**Problem 3: Comparing Uniform and Exponential Distributions**

*par(mfrow = c(1, 2))*

*hist(uniform\_data, breaks = 20, main = "Uniform Distribution", xlab = "Value", col = "lightblue")*

*hist(waiting\_times, breaks = 30, main = "Exponential Distribution", xlab = "Time (minutes)", col = "lightgreen")*

**Experiment 7: Normal and Cauchy Distributions**

**Problem 1: Simulating Normally Distributed Data**

*set.seed(123)*

*normal\_data <- rnorm(1000, mean = 0, sd = 1)*

*# Plotting*

*hist(normal\_data, breaks = 30, main = "Normal Distribution", xlab = "Value", col = "lightblue"*)

**Problem 2: Simulating Cauchy Distributed Data**

*set.seed(123)*

*cauchy\_data <- rcauchy(1000, location = 0, scale = 1)*

*# Plotting*

*hist(cauchy\_data, breaks = 30, main = "Cauchy Distribution", xlab = "Value", col = "lightgreen", xlim = c(-25, 25))*

**Problem 3: Comparing Normal and Cauchy Distributions**

*par(mfrow = c(1, 2))*

*hist(normal\_data, breaks = 30, main = "Normal Distribution", xlab = "Value", col = "lightblue")*

*hist(cauchy\_data, breaks = 30, main = "Cauchy Distribution", xlab = "Value", col = "lightgreen", xlim = c(-25, 25))*

**Experiment 8: Gamma, Beta, and Weibull Distributions**

**Problem 1: Modeling Reliability with Weibull Distribution**

*set.seed(123)*

*weibull\_data <- rweibull(1000, shape = 2, scale = 1)*

*# Plotting*

*hist(weibull\_data, breaks = 30, main = "Weibull Distribution", xlab = "Time", col = "lightblue")*

**Problem 2: Exploring Shape Variations in Gamma Distribution**

*set.seed(123)*

*gamma\_data <- rgamma(1000, shape = 2, rate = 1)*

*# Plotting*

*hist(gamma\_data, breaks = 30, main = "Gamma Distribution", xlab = "Value", col = "lightgreen")*

**Problem 3: Exploring Shape Variations in Beta Distribution**

*set.seed(123)*

*beta\_data <- rbeta(1000, shape1 = 2, shape2 = 5)*

*# Plotting*

*hist(beta\_data, breaks = 30, main = "Beta Distribution", xlab = "Value", col = "lightcoral")*

**Experiment 9: Application of Continuous Probability Distributions in Data Science**

**Problem 1: Predictive Modeling Using Normal Distribution**

set.seed(123)

data <- rnorm(1000, mean = 50, sd = 10)

# Probability of a value less than 60

pnorm(60, mean = 50, sd = 10)

**Problem 2: Estimating Real-World Probabilities Using Fitted Distributions**

set.seed(123)

data <- rnorm(1000, mean = 50, sd = 10)

# Fit a normal distribution

fit <- fitdistrplus::fitdist(data, "norm")

# Summary of the fit

summary(fit)

**Problem 3: Applying Continuous Distributions in Predictive Modeling**

set.seed(123)

x <- rnorm(1000, mean = 50, sd = 10)

y <- 2 \* x + rnorm(1000, mean = 0, sd = 5)

# Linear model

model <- lm(y ~ x)

# Summary of the model

summary(model)

**Experiment 10: Tchebycheff’s Inequality and Convergence in Probability**

**Problem 1: Tchebycheff's Inequality on IPL Runs**

ipl\_scores <- c(45, 50, 55, 60, 48, 70, 35, 40, 61, 52,

                39, 72, 47, 49, 51, 43, 44, 65, 55, 60,

                62, 37, 38, 56, 58, 59, 46, 42, 41, 53)

*mu <- mean(ipl\_scores)*

*sigma <- sd(ipl\_scores)*

*k <- 2*

*upper\_bound <- 1 / k^2*

*# Actual proportion*

*deviation <- abs(ipl\_scores - mu) >= k \* sigma*

*actual <- mean(deviation)*

*list(mean = mu, sd = sigma, bound = upper\_bound, actual\_prob = actual)*

**Problem 2: Marvel Movie Revenue Deviation**

**revenue <- c(850, 900, 870, 860, 890, 910, 905, 925, 950, 940,**

**935, 960, 970, 980, 1000, 930, 920, 940, 950, 945,**

**960, 980, 1010, 1020, 1030, 1040, 1060, 1080, 1090, 1100)**

*mu <- mean(revenue)*

*sigma <- sd(revenue)*

*k <- 3*

*bound <- 1 / k^2*

*actual\_prob <- mean(abs(revenue - mu) >= k \* sigma)*

*list(mean = mu, sd = sigma, bound = bound, actual\_prob = actual\_prob)*

**Problem 3: Bollywood Actor's Earnings Variation**

**earnings <- c(50, 55, 53, 60, 48, 62, 59, 63, 57, 61,**

**66, 52, 49, 64, 58, 65, 67, 51, 47, 68,**

**70, 45, 46, 69, 71, 43, 42, 44, 40, 41)**

*mu <- mean(earnings)*

*sigma <- sd(earnings)*

*k <- 1.5*

*bound <- 1 / (k^2)*

*actual\_prob <- mean(abs(earnings - mu) >= k \* sigma)*

*list(mean = mu, sd = sigma, bound = bound, actual\_prob = actual\_prob)*

**Experiment 11: Weak and Strong Law of Large Numbers (LLN)**

**Problem 1: IPL Strike Rate and Sample Mean Convergence**

strike\_rates <- c(140, 142, 138, 145, 143, 139, 144, 146, 147, 148,

                  150, 141, 137, 136, 149, 135, 134, 151, 152, 153,

                  154, 133, 132, 155, 156, 131, 130, 129, 128, 127)

*cumulative\_avg <- cumsum(strike\_rates) / seq\_along(strike\_rates)*

*plot(cumulative\_avg, type = "l", col = "blue", main = "LLN Demonstration", ylab = "Cumulative Mean", xlab = "Match Number")*

*abline(h = mean(strike\_rates), col = "red", lty = 2)*

**Problem 2: Marvel Movie Lengths**

lengths <- c(120, 118, 119, 121, 122, 123, 124, 125, 126, 127,

             130, 128, 129, 131, 132, 133, 134, 135, 136, 137,

             140, 138, 139, 141, 142, 143, 144, 145, 146, 147)

*c\_avg <- cumsum(lengths) / seq\_along(lengths)*

*plot(c\_avg, type = "l", col = "darkgreen", main = "LLN in Movie Durations", ylab = "Running Mean", xlab = "Movie Count")*

*abline(h = mean(lengths), col = "purple", lty = 2)*

**Problem 3: Bollywood Actor's Yearly Earnings (Strong LLN via multiple samples)**

*set.seed(123)*

*samples <- replicate(1000, mean(sample(earnings, 10, replace = TRUE)))*

*hist(samples, main = "Distribution of Sample Means", xlab = "Sample Mean", col = "skyblue")*

*abline(v = mean(earnings), col = "red", lty = 2)*

**Experiment 12: Central Limit Theorem (CLT) and Its Applications**

**Problem 1: Marvel Movie Revenues (Sampling)**

revenue <- c(

  850, 900, 870, 860, 890, 910, 905, 925, 950, 940,

  935, 960, 970, 980, 1000, 930, 920, 940, 950, 945,

  960, 980, 1010, 1020, 1030, 1040, 1060, 1080, 1090, 1100

)

*set.seed(100)*

*sample\_means <- replicate(1000, mean(sample(revenue, 10, replace = TRUE)))*

*hist(sample\_means, breaks = 20, col = "orange", main = "CLT - Marvel Revenues", xlab = "Sample Means")*

**Problem 2: IPL Scores CLT**

**Dataset:**

ipl\_scores <- c(

  45, 50, 55, 60, 48, 70, 35, 40, 61, 52,

  39, 72, 47, 49, 51, 43, 44, 65, 55, 60,

  62, 37, 38, 56, 58, 59, 46, 42, 41, 53

)

**R Code:**

*set.seed(101)*

*sample\_means <- replicate(1000, mean(sample(ipl\_scores, 15, replace = TRUE)))*

*hist(sample\_means, breaks = 20, col = "lightblue", main = "CLT - IPL Scores", xlab = "Sample Means")*

**Problem 3: Bollywood Actor Earnings CLT**

**Dataset**:

earnings <- c(

  50, 55, 53, 60, 48, 62, 59, 63, 57, 61,

  66, 52, 49, 64, 58, 65, 67, 51, 47, 68,

  70, 45, 46, 69, 71, 43, 42, 44, 40, 41

)

**R Code**:

*set.seed(102)*

*sample\_means <- replicate(1000, mean(sample(earnings, 20, replace = TRUE)))*

*hist(sample\_means, breaks = 20, col = "lightgreen", main = "CLT - Bollywood Earnings", xlab = "Sample Means")*