**Introduction**

The goal is to analyze and visualize sports performance data for players across three sports: cricket, football, and basketball. By applying statistical analysis and visualizations, we derive insights into performance trends, identify top scorers, and compare distributions.

**Abstract**

We create a program in R to calculate basic statistics like mean, median, and standard deviation for player scores across sports. This program also generates visualizations such as bar plots, histograms, and box plots to uncover patterns in performance. The focus is on both statistical rigor and ease of interpretation.

**Methodology**

 **Data Preparation**: A dataset of player scores in different sports is created.

 **Statistical Analysis**: Basic statistics (mean, median, standard deviation) are calculated.

 **Visualization**: Visual tools highlight the performance and score distribution.

 **Player Insights**: Identify the highest and lowest scorers in each sport.

**Technology Used**

**The technologies used in this project are as follows:**

**Programming Language**

* R: The core technology for statistical computation and data visualization.

**Libraries and Tools**

1. ggplot2:
   * A popular library in R for creating visually appealing and customizable plots, such as bar plots, histograms, and box plots.
2. tidyr:
   * A modern library used for reshaping and organizing data into formats suitable for analysis or visualization. It allows transformation to a "long" format using pivot\_longer, which is essential for ggplot-based plots.
3. **Base R Functions:**
   * Used for statistical calculations such as mean(), median(), and sd() (standard deviation).
   * Functions like apply() and indexing with which.max() and which.min() are applied for identifying top and lowest scorers.

**Technological Aspects**

* Data Manipulation: Organizing raw data into structured formats (using tidyr) for ease of analysis.
* Statistical Analysis: Deriving insights from player scores using summary statistics.
* Data Visualization: Building descriptive graphs that highlight trends and performance patterns.

**Code:**

# Load libraries

library(ggplot2)

library(tidyr) # Modern library for reshaping data

# Create a sample dataset

sports\_scores <- data.frame(

Player = c("Prachi", "Alo", "Harman", "Ayush", "Dhoni"),

Cricket = c(80, 92, 67, 75, 89),

Football = c(45, 60, 50, 30, 55),

Basketball = c(18, 24, 20, 22, 28)

)

# View the dataset

print(sports\_scores)

View(sports\_scores)

# 1. Compute Mean, Median, and Mode for Each Sport Separately

mean\_cricket <- mean(sports\_scores$Cricket)

median\_cricket <- median(sports\_scores$Cricket)

mode\_cricket <- as.numeric(names(sort(table(sports\_scores$Cricket), decreasing = TRUE))[1])

mean\_football <- mean(sports\_scores$Football)

median\_football <- median(sports\_scores$Football)

mode\_football <- as.numeric(names(sort(table(sports\_scores$Football), decreasing = TRUE))[1])

mean\_basketball <- mean(sports\_scores$Basketball)

median\_basketball <- median(sports\_scores$Basketball)

mode\_basketball <- as.numeric(names(sort(table(sports\_scores$Basketball), decreasing = TRUE))[1])

# Print Results Separately

cat("Cricket - Mean:", mean\_cricket, " Median:", median\_cricket, " Mode:", mode\_cricket, "\n")

cat("Football - Mean:", mean\_football, " Median:", median\_football, " Mode:", mode\_football, "\n")

cat("Basketball - Mean:", mean\_basketball, " Median:", median\_basketball, " Mode:", mode\_basketball, "\n")

# 2. Compute Summary Statistics

summary\_stats <- data.frame(

Sport = c("Cricket", "Football", "Basketball"),

Mean = c(mean\_cricket, mean\_football, mean\_basketball),

Median = c(median\_cricket, median\_football, median\_basketball),

Mode = c(mode\_cricket, mode\_football, mode\_basketball),

StdDev = sapply(sports\_scores[-1], sd) # Standard deviation for each sport

)

print(summary\_stats)

# 3. Visualizations

# Reshape data for plotting

bar\_data <- sports\_scores %>%

pivot\_longer(cols = -Player, names\_to = "Sport", values\_to = "Score")

# Bar plot

ggplot(bar\_data, aes(x = Sport, y = Score, fill = Player)) +

geom\_bar(stat = "identity", position = "dodge") +

labs(title = "Player Scores by Sport", x = "Sport", y = "Score") +

theme\_minimal()

# Histogram

ggplot(bar\_data, aes(x = Score, fill = Sport)) +

geom\_histogram(binwidth = 10, position = "dodge", alpha = 0.7) +

labs(title = "Score Distribution", x = "Scores", y = "Frequency") +

theme\_minimal()

# Box plot

ggplot(bar\_data, aes(x = Sport, y = Score, fill = Sport)) +

geom\_boxplot() +

labs(title = "Score Distribution by Sport", x = "Sport", y = "Score") +

theme\_minimal()

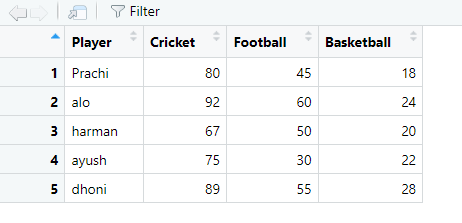
# 4. Identify highest and lowest scorers

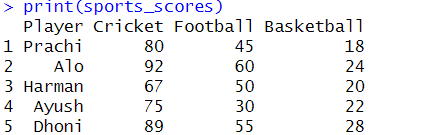
highest\_scorers <- apply(sports\_scores[-1], 2, function(x) sports\_scores$Player[which.max(x)])

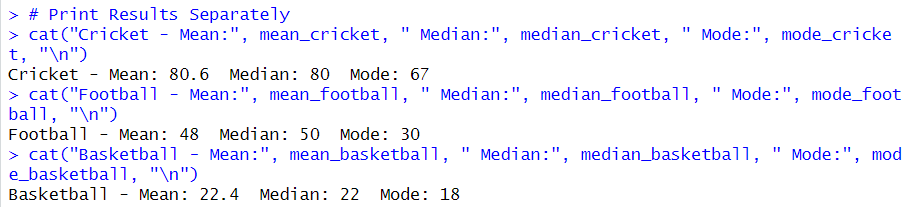
lowest\_scorers <- apply(sports\_scores[-1], 2, function(x) sports\_scores$Player[which.min(x)])

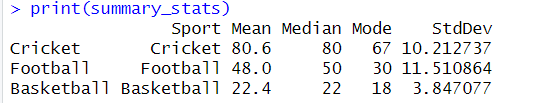
print(data.frame(Sport = colnames(sports\_scores[-1]), Highest = highest\_scorers, Lowest = lowest\_scorers))

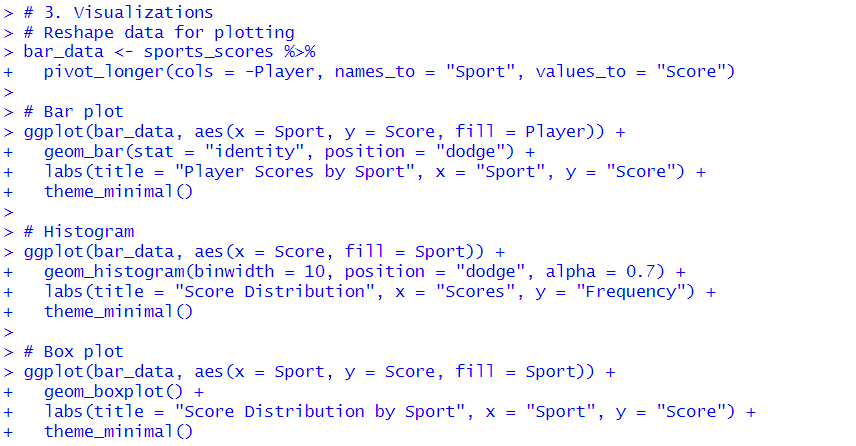
OUTPUT:-

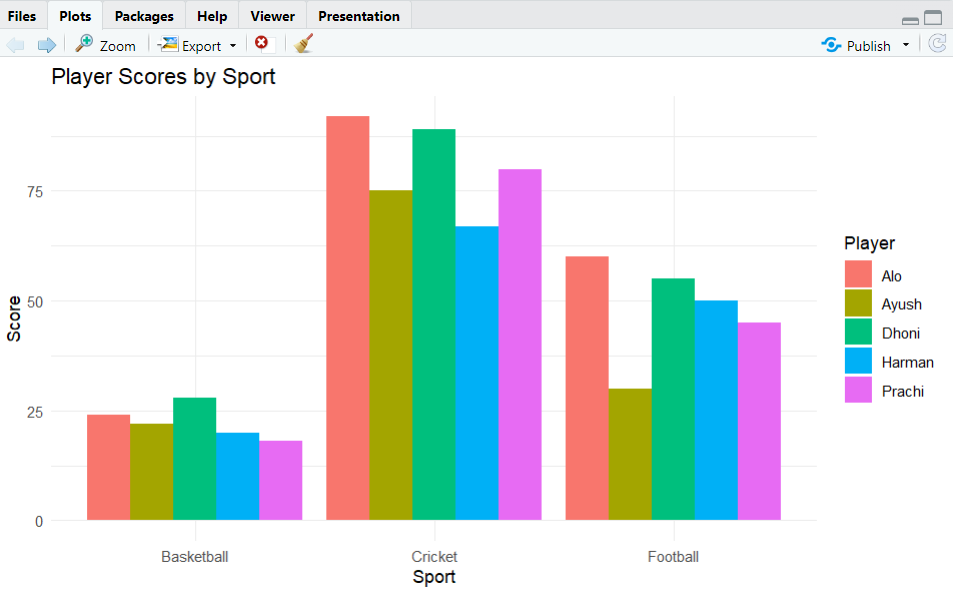


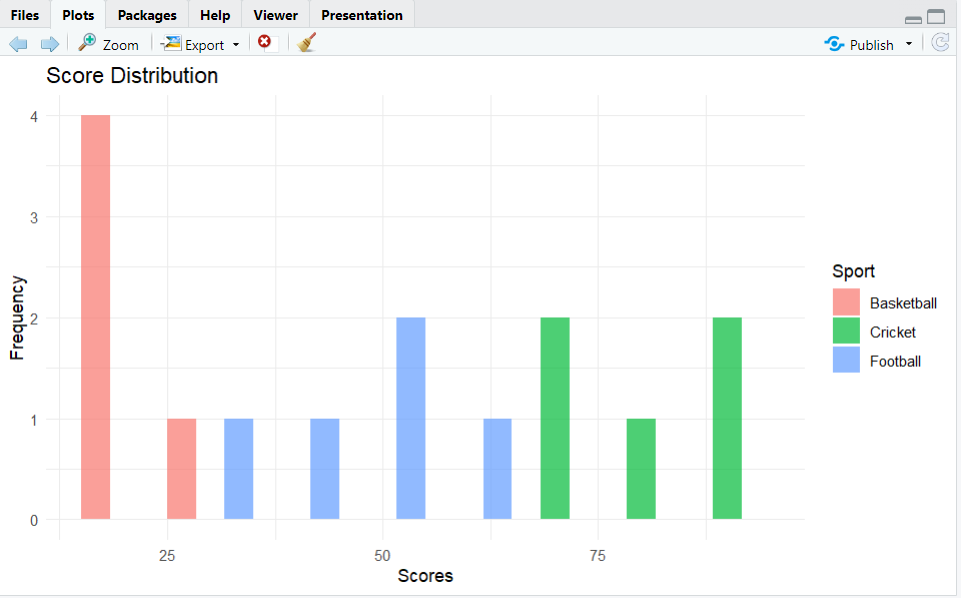




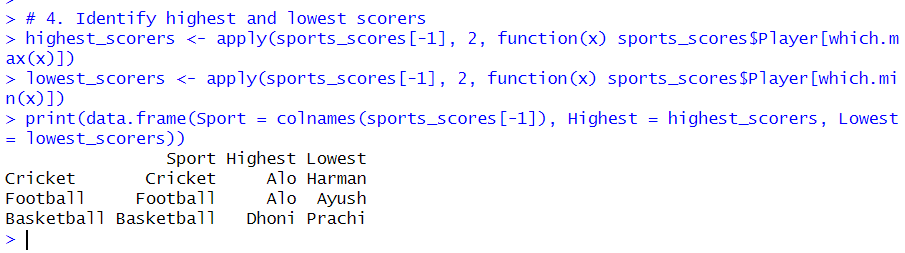












**Explanation of the Code:**

**Step 1: Load Libraries**

R

library(ggplot2)

library(tidyr) # Modern library for reshaping data

* **Purpose**:
  + ggplot2 is used for creating plots such as bar plots, histograms, and box plots.
  + tidyr is used for transforming the dataset into a "long format" that is suitable for plotting.

**Step 2: Create a Sample Dataset**

R

sports\_scores <- data.frame(

Player = c("Prachi", "Alo", "Harman", "Ayush", "Dhoni"),

Cricket = c(80, 92, 67, 75, 89),

Football = c(45, 60, 50, 30, 55),

Basketball = c(18, 24, 20, 22, 28)

)

* **Purpose**: Create a dataset containing player names (Player) and their scores across three sports (Cricket, Football, and Basketball).

**Step 3: View the Dataset**

R

print(sports\_scores)

View(sports\_scores)

* **Purpose**:
  + print() displays the dataset in the console.
  + View() opens the dataset in a spreadsheet-like interface for easier examination (in RStudio).

**Step 4: Compute Mean, Median, and Mode Separately**

**Cricket:**

R

mean\_cricket <- mean(sports\_scores$Cricket)

median\_cricket <- median(sports\_scores$Cricket)

mode\_cricket <- as.numeric(names(sort(table(sports\_scores$Cricket), decreasing = TRUE))[1])

* **Purpose**:
  + mean(): Computes the average cricket score.
  + median(): Finds the midpoint score (central tendency).
  + mode(): Identifies the most frequently occurring score using table() (to count occurrences) and sort() (to rank them).

**Football:**

R

mean\_football <- mean(sports\_scores$Football)

median\_football <- median(sports\_scores$Football)

mode\_football <- as.numeric(names(sort(table(sports\_scores$Football), decreasing = TRUE))[1])

* **Purpose**: Same operations as above but applied to football scores.

**Basketball:**

R

mean\_basketball <- mean(sports\_scores$Basketball)

median\_basketball <- median(sports\_scores$Basketball)

mode\_basketball <- as.numeric(names(sort(table(sports\_scores$Basketball), decreasing = TRUE))[1])

* **Purpose**: Same operations but applied to basketball scores.

**Step 5: Print Results Separately**

R

cat("Cricket - Mean:", mean\_cricket, " Median:", median\_cricket, " Mode:", mode\_cricket, "\n")

cat("Football - Mean:", mean\_football, " Median:", median\_football, " Mode:", mode\_football, "\n")

cat("Basketball - Mean:", mean\_basketball, " Median:", median\_basketball, " Mode:", mode\_basketball, "\n")

* **Purpose**: Display the calculated mean, median, and mode for each sport in the console using cat() for formatted output.

**Step 6: Compute Summary Statistics**

R

summary\_stats <- data.frame(

Sport = c("Cricket", "Football", "Basketball"),

Mean = c(mean\_cricket, mean\_football, mean\_basketball),

Median = c(median\_cricket, median\_football, median\_basketball),

Mode = c(mode\_cricket, mode\_football, mode\_basketball),

StdDev = sapply(sports\_scores[-1], sd) # Standard deviation for each sport

)

print(summary\_stats)

* **Purpose**:
  + Create a summary table combining all statistical measures (mean, median, mode, and standard deviation) for each sport.
  + sapply() applies sd() (standard deviation) column-wise to calculate variability of scores.

**Step 7: Reshape Data for Plotting**

R

bar\_data <- sports\_scores %>%

pivot\_longer(cols = -Player, names\_to = "Sport", values\_to = "Score")

* **Purpose**: Use pivot\_longer() to transform the dataset from wide format (columns for sports) to long format:
  + Columns Sport and Score are created, making it easier for ggplot to plot data.

**Step 8: Create Visualizations**

**Bar Plot:**

R

ggplot(bar\_data, aes(x = Sport, y = Score, fill = Player)) +

geom\_bar(stat = "identity", position = "dodge") +

labs(title = "Player Scores by Sport", x = "Sport", y = "Score") +

theme\_minimal()

* **Purpose**: Visualize player scores for each sport using grouped bars. geom\_bar() creates the bar plot, and position = "dodge" separates bars by player.

**Histogram:**

R

ggplot(bar\_data, aes(x = Score, fill = Sport)) +

geom\_histogram(binwidth = 10, position = "dodge", alpha = 0.7) +

labs(title = "Score Distribution", x = "Scores", y = "Frequency") +

theme\_minimal()

* **Purpose**: Plot score distribution using geom\_histogram():
  + binwidth defines the size of score intervals.
  + alpha adds transparency for overlapping histograms.

**Box Plot:**

R

ggplot(bar\_data, aes(x = Sport, y = Score, fill = Sport)) +

geom\_boxplot() +

labs(title = "Score Distribution by Sport", x = "Sport", y = "Score") +

theme\_minimal()

* **Purpose**: Create box plots to show score spread, median, and outliers for each sport using geom\_boxplot().

**Step 9: Identify Highest and Lowest Scorers**

R

highest\_scorers <- apply(sports\_scores[-1], 2, function(x) sports\_scores$Player[which.max(x)])

lowest\_scorers <- apply(sports\_scores[-1], 2, function(x) sports\_scores$Player[which.min(x)])

print(data.frame(Sport = colnames(sports\_scores[-1]), Highest = highest\_scorers, Lowest = lowest\_scorers))

* **Purpose**:
  + apply() operates column-wise (across sports), finding the player with the highest (which.max()) and lowest (which.min()) score for each sport.
  + Creates a summary table showing the top and bottom scorers.
* **Conclusion**

The R program effectively analyzes and visualizes sports scores using statistical measures like mean, median, mode, and standard deviation, combined with visual tools such as bar plots, histograms, and box plots. It identifies top and lowest scorers, offering a clear understanding of player performance trends. Leveraging modern libraries like ggplot2 and tidyr, this approach is efficient, adaptable, and suitable for diverse sports analytics applications. 🚀

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*