Introduction to R Programming: Descriptive Statistics

Instructor: Dr. Azadeh Alimadad
Department of Statistics & Data Analytics

Course Overview

This lecture introduces you to R programming with a focus on descriptive statistics. No prior R experience is required. We'll cover basic R syntax, data structures, and statistical measures for summarizing data.

Contents

1 Introduction to R

1.1 What is R?

R is a free, open-source programming language and environment for statistical computing and graphics. It's widely used by statisticians, data analysts, and researchers.

1.2 Getting Started

1.2.1 Installing R and RStudio

- Download R from: `https://cran.r-project.org/`
- Download RStudio (recommended IDE) from: `https://www.rstudio.com/`

1.2.2 RStudio Interface

RStudio has four main panes:

1. Source Editor (top-left): Write and save scripts
2. Console (bottom-left): Execute commands
3. Environment/History (top-right): View objects and command history
4. Files/Plots/Packages/Help (bottom-right): Multiple tabs for various functions

1.3 Basic R Operations

```r
# Basic arithmetic operations
2 + 3           # Addition
5 - 2           # Subtraction
3 * 4           # Multiplication
10 / 2          # Division
2^3             # Exponentiation
sqrt(16)        # Square root
log(10)         # Natural logarithm
exp(2)          # Exponential function
```

1.4 R as a Calculator

```r
# Order of operations follows standard mathematical rules
(3 + 5) * 2     # Result: 16
3 + 5 * 2       # Result: 13

# Mathematical functions
abs(-5)         # Absolute value
round(3.14159, 2) # Round to 2 decimal places
ceiling(3.2)    # Round up
floor(3.8)      # Round down
```

2 Data Structures in R

2.1 Vectors

Vectors are the basic data structure in R.

```
# Creating vectors
x <- c(1, 2, 3, 4, 5)          # Using c() function
y <- 1:10                       # Sequence from 1 to 10
z <- seq(0, 1, by = 0.1)       # Sequence with specific increment
w <- rep(5, times = 3)         # Repeat value

# Vector operations
x * 2                          # Multiply each element by 2
x + y                          # Element-wise addition (if same length)
sum(x)                         # Sum of elements
length(x)                      # Number of elements
```

2.2 Matrices

```
# Creating matrices
mat <- matrix(1:12, nrow = 3, ncol = 4)
mat2 <- matrix(1:12, nrow = 3, ncol = 4, byrow = TRUE)

# Matrix operations
dim(mat)                       # Dimensions
t(mat)                         # Transpose
mat %*% t(mat)                 # Matrix multiplication
```

2.3 Data Frames

Data frames are the most common data structure for statistical analysis.

```
# Creating a data frame
students <- data.frame(
    name = c("Alice", "Bob", "Charlie", "Diana"),
    age = c(20, 21, 19, 22),
    grade = c(85, 92, 78, 88)
)

# Accessing data frame elements
students$name                  # Access column by name
students[["age"]]              # Another way to access column
students[1, ]                  # First row
students[, 2]                  # Second column
students[2, 3]                 # Element at row 2, column 3
```

3 Importing and Exploring Data

3.1 Reading Data

```
# Reading CSV files
data <- read.csv("filename.csv")

# Reading Excel files (requires readxl package)
```

```r
# install.packages("readxl")
library(readxl)
data <- read_excel("filename.xlsx")

# Reading built-in datasets
data(mtcars)                   # Load built-in dataset
head(mtcars)                   # View first few rows
```

3.2 Examining Data

```r
# Basic data examination functions
str(mtcars)                    # Structure of data
summary(mtcars)                # Summary statistics
names(mtcars)                  # Column names
nrow(mtcars)                   # Number of rows
ncol(mtcars)                   # Number of columns
dim(mtcars)                    # Dimensions
class(mtcars)                  # Class/type of object
```

4 Descriptive Statistics: Theory and Formulas

4.1 Measures of Central Tendency

4.1.1 Mean (Arithmetic Average)

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

Where:

- $\bar{x}$ is the sample mean
- x_i are the individual observations
- n is the number of observations

4.1.2 Median

The middle value when data is sorted in ascending order.

- For odd n: Median = middle value
- For even n: Median = average of two middle values

4.1.3 Mode

The value that appears most frequently in a dataset.

4.2 Measures of Dispersion

4.2.1 Range

$$\text{Range} = \text{Maximum} - \text{Minimum} \tag{2}$$

4.2.2 Variance

Sample variance:

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} \tag{3}$$

4.2.3 Standard Deviation

$$s = \sqrt{s^2} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{4}$$

4.2.4 Interquartile Range (IQR)

$$\text{IQR} = Q_3 - Q_1 \tag{5}$$

Where Q_1 is the first quartile (25th percentile) and Q_3 is the third quartile (75th percentile).

4.3 Measures of Shape

4.3.1 Skewness

$$\text{Skewness} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^3}{(n-1)s^3} \tag{6}$$

- Positive: Right-skewed (tail on right)
- Negative: Left-skewed (tail on left)
- Zero: Symmetrical

4.3.2 Kurtosis

$$\text{Kurtosis} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^4}{(n-1)s^4} - 3 \tag{7}$$

- Positive: Heavy tails (leptokurtic)
- Negative: Light tails (platykurtic)
- Zero: Normal distribution (mesokurtic)

5 Descriptive Statistics in R

5.1 Basic Statistical Functions

```r
# Create example data
data <- c(23, 45, 67, 34, 89, 56, 78, 12, 45, 67)

# Measures of central tendency
mean(data)                      # Mean
median(data)                    # Median

# Mode (no built-in function, but we can create one)
get_mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
get_mode(data)
```

```

# Measures of dispersion
min(data)                      # Minimum
max(data)                      # Maximum
range(data)                    # Range
var(data)                      # Variance
sd(data)                       # Standard deviation
IQR(data)                      # Interquartile range
quantile(data)                 # All quartiles
quantile(data, probs = c(0.25, 0.5, 0.75))  # Specific quartiles

# Summary statistics
summary(data)                  # Five-number summary + mean
```

5.2 Descriptive Statistics for Data Frames

```
# Using mtcars dataset
data(mtcars)

# Summary for entire data frame
summary(mtcars)

# Summary for specific column
summary(mtcars$mpg)

# Calculate multiple statistics
mean(mtcars$mpg)
sd(mtcars$mpg)
median(mtcars$mpg)
```

5.3 Advanced Descriptive Statistics

```
# Using psych package for comprehensive statistics
# install.packages("psych")
library(psych)

describe(mtcars)               # Comprehensive descriptive statistics
describeBy(mtcars$mpg, mtcars$cyl)  # Statistics by group

# Skewness and Kurtosis
# install.packages("moments")
library(moments)
skewness(mtcars$mpg)           # Skewness
kurtosis(mtcars$mpg)           # Kurtosis

# Coefficient of Variation
cv <- function(x) sd(x)/mean(x) * 100
cv(mtcars$mpg)
```

6 Data Visualization for Descriptive Statistics

6.1 Histogram

```r
# Basic histogram
hist(mtcars$mpg,
     main = "Distribution of MPG",
     xlab = "Miles per Gallon",
     ylab = "Frequency",
     col = "lightblue",
     border = "black")

# Histogram with density curve
hist(mtcars$mpg,
     prob = TRUE,             # Plot as probability density
     main = "Distribution of MPG with Density Curve",
     xlab = "Miles per Gallon",
     col = "lightgreen")
lines(density(mtcars$mpg), col = "red", lwd = 2)
```

6.2 Box Plot

```r
# Basic box plot
boxplot(mtcars$mpg,
        main = "Box Plot of MPG",
        ylab = "Miles per Gallon",
        col = "lightyellow")

# Box plot by group
boxplot(mpg ~ cyl, data = mtcars,
        main = "MPG by Number of Cylinders",
        xlab = "Number of Cylinders",
        ylab = "Miles per Gallon",
        col = c("red", "green", "blue"))
```

6.3 Scatter Plot

```r
# Scatter plot
plot(mtcars$wt, mtcars$mpg,
     main = "Weight vs MPG",
     xlab = "Weight (1000 lbs)",
     ylab = "Miles per Gallon",
     pch = 19,                # Point shape
     col = "blue")

# Add regression line
abline(lm(mpg ~ wt, data = mtcars), col = "red", lwd = 2)
```

6.4 Bar Plot

```r
# Create frequency table
cyl_counts <- table(mtcars$cyl)

# Bar plot
barplot(cyl_counts,
        main = "Number of Cars by Cylinders",
        xlab = "Number of Cylinders",
        ylab = "Frequency",
        col = "steelblue",
        border = "black")
```

7 Handling Missing Data

```
# Create data with missing values
data_with_na <- c(23, 45, NA, 67, NA, 89, 56)

# Check for missing values
is.na(data_with_na)           # Returns TRUE/FALSE for each element
any(is.na(data_with_na))      # Check if any missing values exist
sum(is.na(data_with_na))      # Count missing values

# Remove missing values
data_clean <- na.omit(data_with_na)

# Calculate statistics with missing values
mean(data_with_na)                    # Returns NA
mean(data_with_na, na.rm = TRUE)      # Removes NA before calculation
```

8 Practical Examples

8.1 Example 1: Student Grades Analysis

```
# Create student data
students <- data.frame(
    student_id = 1:30,
    math = round(rnorm(30, mean = 75, sd = 10), 1),
    science = round(rnorm(30, mean = 80, sd = 8), 1),
    english = round(rnorm(30, mean = 70, sd = 12), 1)
)

# Calculate overall statistics
students$average <- rowMeans(students[, 2:4])

# Summary statistics for each subject
summary(students$math)
summary(students$science)
summary(students$english)

# Compare subjects
cat("Math - Mean:", mean(students$math),
    "SD:", sd(students$math), "\n")
cat("Science - Mean:", mean(students$science),
    "SD:", sd(students$science), "\n")
cat("English - Mean:", mean(students$english),
    "SD:", sd(students$english), "\n")

# Visualization
par(mfrow = c(2, 2))  # Create 2x2 plot grid
hist(students$math, main = "Math Scores", col = "lightblue")
hist(students$science, main = "Science Scores", col = "lightgreen")
hist(students$english, main = "English Scores", col = "lightpink")
```

```r
boxplot(students[, 2:4], main = "Subject Comparison",
        col = c("lightblue", "lightgreen", "lightpink"))
```

8.2 Example 2: Sales Data Analysis

```r
# Create sales data
set.seed(123)  # For reproducibility
sales_data <- data.frame(
    day = 1:100,
    sales = round(rnorm(100, mean = 5000, sd = 1500), 0),
    customers = round(rnorm(100, mean = 200, sd = 50), 0)
)

# Add some outliers
sales_data$sales[c(10, 50, 90)] <- c(10000, 12000, 8000)

# Calculate daily statistics
sales_data$revenue_per_customer <- sales_data$sales / sales_data$
    customers

# Weekly summary (assuming 7-day weeks)
sales_data$week <- ceiling(sales_data$day / 7)

# Aggregate by week
weekly_summary <- aggregate(sales ~ week, data = sales_data,
                           FUN = function(x) c(mean = mean(x),
                                               sd = sd(x),
                                               median = median(x)))
# Display results
print("Weekly Sales Summary:")
print(weekly_summary)

# Identify outliers
Q1 <- quantile(sales_data$sales, 0.25)
Q3 <- quantile(sales_data$sales, 0.75)
IQR_value <- IQR(sales_data$sales)
lower_bound <- Q1 - 1.5 * IQR_value
upper_bound <- Q3 + 1.5 * IQR_value

outliers <- sales_data$sales[sales_data$sales < lower_bound |
                            sales_data$sales > upper_bound]
cat("\nOutliers detected:", length(outliers), "\n")
cat("Outlier values:", outliers, "\n")
```

9 Common Errors and Troubleshooting

9.1 Common R Errors

- `Error: object not found`: Check spelling and if object exists
- `Error: unexpected symbol`: Check syntax, missing commas or parentheses
- `Warning: NAs introduced`: Missing values in calculations
- `Error: non-numeric argument`: Trying to do math on non-numeric data

9.2 Debugging Tips

```r
# Use these functions to debug:
str(your_data)                 # Check structure
class(your_variable)           # Check data type
head(your_data)                # View first few rows
summary(your_data)             # Get summary statistics

# Check for common issues:
is.numeric(your_variable)      # Is it numeric?
any(is.na(your_data))          # Any missing values?
length(unique(your_variable))  # How many unique values?
```

10 Best Practices

1. **Comment your code**: Use # for comments
2. **Use meaningful variable names**: e.g., `student_grades` not `x`
3. **Save your scripts**: Use .R extension
4. **Check your data**: Always examine data before analysis
5. **Handle missing values**: Decide how to treat NAs
6. **Visualize first**: Plot data before analyzing
7. **Reproducibility**: Use `set.seed()` for random numbers

11 Exercises

11.1 Basic Exercises

1. Create a vector of 20 random numbers between 1 and 100 and calculate:
   - Mean, median, and mode
   - Variance and standard deviation
   - Range and IQR
2. Load the `iris` dataset using `data(iris)` and:
   - Calculate summary statistics for Sepal.Length
   - Compare statistics between different species
   - Create histograms for each numeric variable
3. Create a data frame with 50 observations of:
   - Age (random between 18-65)
   - Income (random between 30000-120000)
   - Education (random categories: High School, Bachelor, Master, PhD)

   Calculate appropriate statistics for each variable.

11.2 Advanced Exercises

1. Analyze the relationship between mpg and weight in the mtcars dataset:
   - Calculate correlation coefficient
   - Create a scatter plot with regression line
   - Calculate descriptive statistics for mpg by transmission type (am)
2. Simulate exam scores for 200 students across 5 subjects and:
   - Identify outliers in each subject
   - Calculate overall performance statistics
   - Create a comprehensive report of findings

12 Resources for Further Learning

- **Books**:
  - "R for Data Science" by Hadley Wickham
  - "The R Book" by Michael J. Crawley
  - "An Introduction to R" by Venables and Smith
- **Online Resources**:
  - R Documentation: `https://www.r-project.org/help.html`
  - R-bloggers: `https://www.r-bloggers.com/`
  - Stack Overflow: `https://stackoverflow.com/questions/tagged/r`
- **Courses**:
  - DataCamp: Introduction to R
  - Coursera: R Programming by Johns Hopkins University
  - edX: Introduction to R for Data Science

Conclusion

This lecture covered the fundamentals of using R for descriptive statistics. Remember that descriptive statistics are the foundation of any data analysis, providing insights into your data's characteristics. Practice regularly and explore R's extensive capabilities for data analysis and visualization.

Remember: The best way to learn R is by using it!