Introduction to R and Basic Data Analysis

Introduction to R and Basic Data Analysis

This laboratory introduces you to the R programming environment and basic data analysis techniques. R was specifically designed for statistics and data analysis, so tasks such as loading data, computing summaries, and creating visualisations are native parts of the language and its standard ecosystem.

As a first quick-and-dirty example, run the following code, which reads measurement data directly from the course website and inspects it:

```
tab = read.table("https://ccfd.github.io/courses/data/info1/przebieg.txt

str(tab)
summary(tab)
plot(tab)
```

1. Getting Started with R

1.1 Installing and Starting R

R can be downloaded from CRAN. For this course, we recommend using RStudio, which provides a user-friendly interface for R.

Start RStudio and familiarize yourself with the interface: - **Console**: Where you type and execute R commands - **Script Editor**: Where you write and save your R scripts - **Environment**: Shows objects in your workspace - **Plots/Help**: Displays plots and help documentation

1.2 Basic R Operations

Execute the following commands in the R console:

```
# Basic arithmetic operations
2 + 3
5 * 4
10 / 2
2^3

# Variable assignment
x <- 5
y <- 10
x + y

# Creating vectors
numbers <- c(1, 2, 3, 4, 5)
numbers
```

Task: Create a vector containing the first 10 even numbers and calculate their sum.

2. Working with Data

2.1 Creating Data Frames

A data frame is R's primary structure for storing tabular data. Create a simple data frame:

```
# Create a data frame
students <- data.frame(
  name = c("Alice", "Bob", "Charlie", "Diana"),
  age = c(20, 21, 19, 22),
  score = c(85, 92, 78, 95)
)
students

# View structure of the data frame
str(students)
```

```
# Summary statistics
summary(students)
```

Task: Create a data frame with information about 5 products, including columns for product name, price, and quantity in stock.

2.2 Reading Data from Files

R can read data from various file formats. For CSV files:

```
# Read a CSV file (if it exists)
# data <- read.csv("filename.csv")

# For this exercise, create sample data
set.seed(123)
sample_data <- data.frame(
  id = 1:20,
  value = rnorm(20, mean = 50, sd = 10)
)
```

Task: Create a data frame with 30 observations of a variable following a normal distribution with mean 100 and standard deviation 15.

3. Basic Statistical Analysis

3.1 Descriptive Statistics

Calculate basic descriptive statistics:

```
# Generate sample data
data <- rnorm(100, mean = 50, sd = 10)

# Mean
mean(data)

# Median
median(data)
```

```
# Standard deviation
sd(data)

# Variance
var(data)

# Minimum and maximum
min(data)
max(data)

# Range
range(data)

# Quantiles
quantile(data, c(0.25, 0.5, 0.75))
```

Task: Generate 200 random numbers from a uniform distribution between 0 and 100, and calculate all descriptive statistics mentioned above.

3.2 Data Visualization

Create basic plots:

```
# Histogram
hist(data, main = "Histogram of Data", xlab = "Values")

# Boxplot
boxplot(data, main = "Boxplot of Data")

# Scatter plot (if you have two variables)
x <- 1:50
y <- x + rnorm(50, 0, 5)
plot(x, y, main = "Scatter Plot", xlab = "X", ylab = "Y")
```

Task: Create a histogram and boxplot for the uniform distribution data from the previous task.

4. Working with Real Data

4.1 Exploring Built-in Datasets

R comes with many built-in datasets. Explore one:

```
# Load a built-in dataset
data(mtcars)

# View first few rows
head(mtcars)

# View structure
str(mtcars)

# Summary statistics
summary(mtcars)

# Access specific columns
mtcars$mpg
mtcars[, "mpg"]
```

Task: Using the `mtcars` dataset: 1. Calculate the mean and standard deviation of miles per gallon (mpg) 2. Find the car with the highest horsepower (hp) 3. Create a scatter plot of mpg vs. hp

4.2 Data Subsetting

Select specific rows and columns:

```
# Select rows where mpg > 20
high_mpg <- mtcars[mtcars$mpg > 20, ]
high_mpg

# Select specific columns
mpg_hp <- mtcars[, c("mpg", "hp")]
mpg_hp

# Combine conditions
fast_efficient <- mtcars[mtcars$mpg > 20 & mtcars$hp > 100, ]
fast_efficient
```

Task: From the `mtcars` dataset, create a subset containing only cars with: - More than 6 cylinders - Weight (wt) less than 3.5 Display the mean horsepower for this subset.

5. Basic Data Manipulation

5.1 Adding and Modifying Columns

```
# Add a new column
mtcars$km_per_liter <- mtcars$mpg * 0.425144
head(mtcars)

# Modify existing column (convert weight to kg)
mtcars$wt_kg <- mtcars$wt * 453.592
head(mtcars)
```

Task: Add a column to `mtcars` that categorizes cars as "Light" (wt $<$ 3), "Medium" ($3 \leq$ wt < 4), or "Heavy" (wt ≥ 4).

5.2 Basic Aggregation

```
# Calculate mean by group
tapply(mtcars$mpg, mtcars$cyl, mean)

# Count observations by group
table(mtcars$cyl)
```

Task: Calculate the mean horsepower for each number of cylinders in the `mtcars` dataset.

6. Exercises

Complete the following exercises:

1. Create a data frame with 100 observations containing:
   - An ID column (1 to 100)
   - A column with random values from a normal distribution (mean=50, sd=10)
   - A column with random values from a uniform distribution (0 to 100)
2. For the normal distribution column:
   - Calculate mean, median, standard deviation, and variance
   - Create a histogram and boxplot
   - Identify any values that are more than 2 standard deviations from the mean
3. Using the `mtcars` dataset:
   - Create a summary table showing the mean mpg, hp, and wt for each number of cylinders
   - Create scatter plots of mpg vs. hp, colored by number of cylinders
   - Calculate the correlation coefficient between mpg and hp
4. Create a new dataset by combining two vectors:
   - A vector of 50 student names (you can use paste0("Student", 1:50))
   - A vector of 50 exam scores (random values between 0 and 100)
   - Add a column that categorizes scores as "A" ($\geq$90), "B" (80-89), "C" (70-79), "D" (60-69), or "F" ($<$60)
   - Calculate the percentage of students in each grade category

7. Additional Resources

- R documentation: Use `?function_name` or `help(function_name)` for help on any function
- RStudio cheatsheets: Available in Help menu
- CRAN Task Views: https://cran.r-project.org/web/views/