1. Getting Started

Emmanuel Masavo DJEGOU

2025-04-06

Contents

1 Executing R Scripts **1**
1.1 Interactive Mode 1
1.2 Batch Mode 1

2 Exploring Your First R Session **1**

3 Creating and Using Functions in R **4**

4 Learning More About R Functions **4**

1 Executing R Scripts

R can be run in two modes: **interactive** and **batch**. Here's a brief overview of both:

1.1 Interactive Mode

In interactive mode, you can run commands directly, and R will display results immediately in the console. This mode is useful for experimenting and quickly testing your ideas.

```
# Example of Interactive Mode: Creating a numeric vector and calculating its mean
x <- 1:5
mean(x)  # Output shown directly in the console
```

```
## [1] 3
```

1.2 Batch Mode

Batch mode runs scripts non-interactively, meaning you can execute a script without needing to type any commands in the console. This is especially useful for automation, such as running scripts on a schedule (e.g., once per day).

To run a script in batch mode, use the following command outside of R:

- Example: Running a script in batch mode (via terminal)
- Rscript my_script.R Executes the R script without user interaction

2 Exploring Your First R Session

In this section, we will explore how to create and manipulate vectors, compute summary statistics, and visualize data using R.

```r
# Creating a numeric vector
x <- c(1, 2, 4)

# Creating a new vector by repeating x and adding 8
q <- c(x, x, 8)

# Accessing the third element of x
print(x[3])
```

```
## [1] 4
```

```r
# Subsetting: extracting the first two elements of x
subset_x <- x[1:2]
print(subset_x)
```

```
## [1] 1 2
```

```r
# Calculating the mean of x
mean_x <- mean(x)
print(mean_x)
```

```
## [1] 2.333333
```

```r
# Calculating the standard deviation of x
std_x <- sd(x)
print(std_x)
```

```
## [1] 1.527525
```

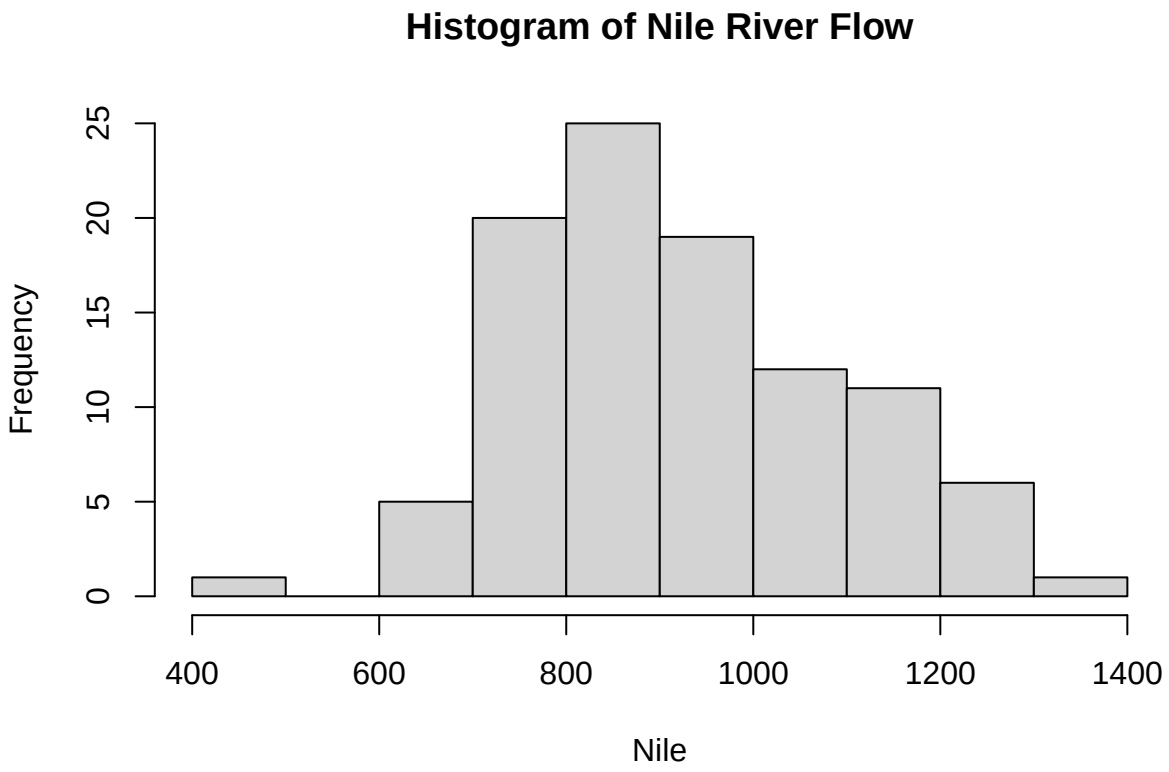
```r
# Listing available datasets
data()

# Working with the "Nile" dataset
print(mean(Nile))      # Mean of the dataset
```

```
## [1] 919.35
```

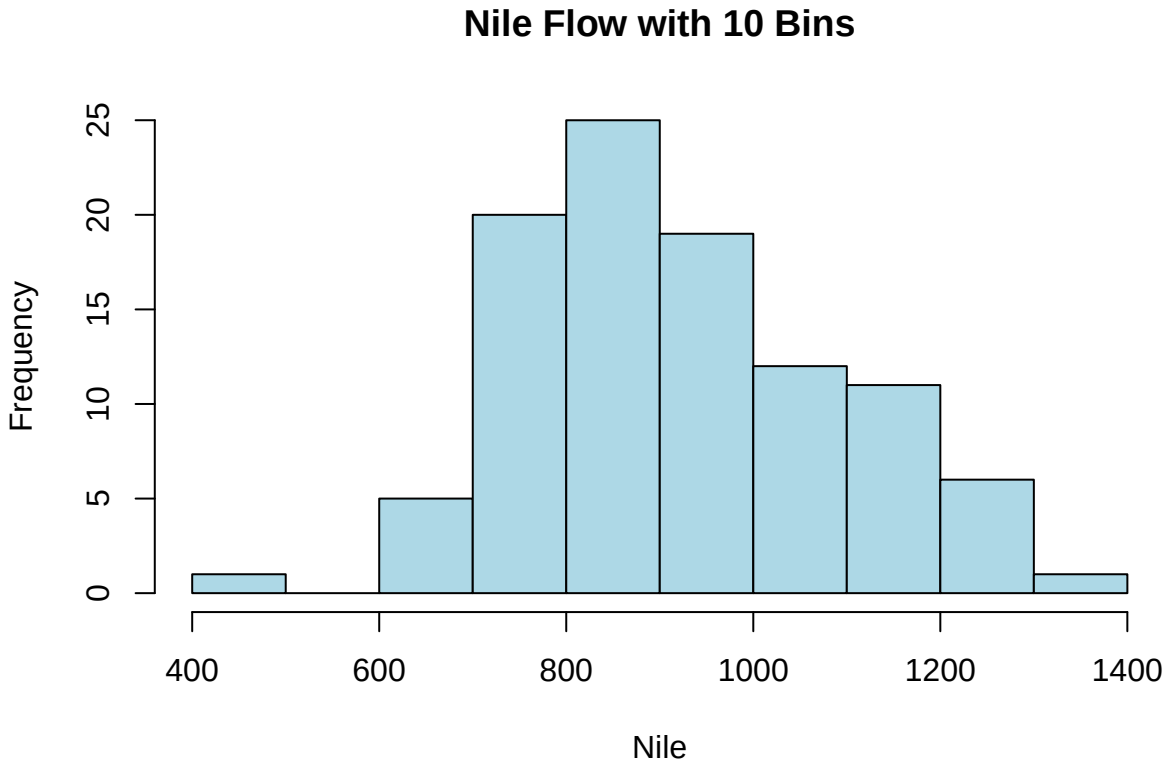
```r
print(sd(Nile))        # Standard deviation
```

```
## [1] 169.2275
```

```r
# Plotting histogram
hist(Nile, main = "Histogram of Nile River Flow")
```




```r
hist(Nile, breaks = 10, col = "lightblue", main = "Nile Flow with 10 Bins")
```



3 Creating and Using Functions in R

```
# Function to count the number of odd numbers in a vector
oddcount <- function(x) {
     k <- 0
     for (n in x) {
          if (n %% 2 == 1) k <- k + 1
     }
     return(k)
}

# Testing the oddcount function
y <- c(1, 2, 3, 7, 9)
print(oddcount(y))
```

```
## [1] 4
```

```
# Modulo operator example
print(38 %% 7)
```

```
## [1] 3
```

```
# Global variable example
f <- function(x) return(z + y)
z <- 3
print(f(z))
```

```
## [1]  4  5  6 10 12
```

```
# Function with default arguments
g <- function(x, y = 2, z = TRUE) {
     print(paste("x:", x, "| y:", y, "| z:", z))
}

g(12, z = FALSE)
```

```
## [1] "x: 12 | y: 2 | z: FALSE"
```

```
# To quit R (uncomment if needed)
# q()
```

4 Learning More About R Functions