

# R Programming Lab Experiment – 100 Marks

---

Module: Foundations of Data Analytics

Instructor: [Your Name]

Experiment No: [XX]

Title: Crash Site-Based Vector Analysis of 30 Plane Crash Victims

## Scenario:

A plane carrying 30 passengers crashed, and rescue efforts were carried out across multiple locations. Your job is to analyze various survival patterns and passenger characteristics using only one-dimensional vectors and user-defined functions.

You are not allowed to use data frames, lists, matrices, or any visualization techniques.

## Learning Objectives:

- Work with multiple vectors representing real-world data
- Apply filtering and logic operations in R using conditional indexing
- Generate formatted textual summaries using custom functions
- Demonstrate structured thinking with attribute-based segmentation

## Attributes to Initialize (Each as a vector of 30 elements):

1. age – Numeric: Age in years
2. weight – Numeric: Weight in kilograms
3. survival – Binary: 1 = survived, 0 = deceased
4. gender – Character: "M" or "F"
5. class – Character: "E" (Economy), "B" (Business), "F" (First Class)
6. health\_score – Numeric: 0 to 100
7. crash\_site – Character: Indicates where each passenger was found ("Forest", "Sea", "Mountain")

## Write the Following User-Defined Functions

Use `cat()` or `print()` to format all results clearly. Add meaningful comments to explain each step.

### General Survival Analysis

- 1. `calculate_survival_rate()` – Print total survivors and percentage (5 Marks)
- 2. `average_age_survivors()` – Print average age of survivors (5 Marks)
- 3. `average_health_score_deceased()` – Print average health score of deceased (5 Marks)
- 4. `heaviest_survivor()` – Print weight of the heaviest survivor (5 Marks)
- 5. `youngest_first_class_survivor()` – Print age of the youngest first-class survivor (5 Marks)
- 6. `gender_wise_survival_ratio()` – Print M/F survival percentages (10 Marks)

### Health and Class-Based Logic

- 7. `health_category()` – Categorize health into 'Low', 'Medium', 'High'; Return and print (5 Marks)
- 8. `survival_by_health_category()` – Print survivor count per health category (10 Marks)
- 9. `most_common_class_survived()` – Print most common passenger class among survivors (5 Marks)

### Crash Site-Based Analysis (New Section)

- 10. `survivors_by_crash_site()` – Print number of survivors from each site (5 Marks)
- 11. `average_health_by_site()` – Print average health score per crash site (5 Marks)
- 12. `site_with_highest_survival()` – Print which site had the most survivors (5 Marks)

### Final Summary Report

- 13. `print_full_summary()` – Use `cat()` to display:
  - Total passengers
  - Survivors and deceased counts
  - Gender-wise and class-wise survival
  - Average health of survivors
  - Crash site with highest survival (20 Marks)

### Evaluation Rubrics (100 Marks)

| Section                                | Marks |
|----------------------------------------|-------|
| Initialization of 30-element vectors   | 10    |
| General analysis (Functions 1–6)       | 30    |
| Health and class logic (Functions 7–9) | 20    |
| Crash site analysis (Functions 10–12)  | 20    |
| Summary report (Function 13)           | 20    |

## Lab Report Submission Instructions:

- File Name Format:

Experiment No\_Name of Student\_Registration Number.doc or pdf

- Example:

Experiment 02\_AkashReddy\_21BDA1234.docx

- Submit a Lab Report with the following:

- Initialization of all required vectors
- All function definitions with in-line comments
- Function calls demonstrating outputs
- Clear formatting using cat() or print()
- No use of data.frame, list, matrix, ggplot2, or any advanced structures

## Lab Report Format (To Be Included in the Top of Script)

# -----

# Name : [Your Full Name]

# Registration No: [Your Reg. No]

# Experiment No : [XX]

# Title : Crash Site-Based Vector Analysis

# Date : [DD/MM/YYYY]

# -----

# Vector Initialization Section

# [Initialize vectors: age, weight, survival, gender, class, health\_score, crash\_site]

# User-Defined Functions Section

# [Define all 13 functions here with comments]

# Function Calls Section

# [Call each function and display results in formatted output]

# Programming for Data Science – Lab Instructions for Students

All students are required to strictly adhere to the following instructions during lab sessions. Any violation will be treated as **academic malpractice** and appropriate disciplinary action will be taken.

## 1. No Use of AI Tools

- Students **must not use** any Artificial Intelligence (AI) tools (including but not limited to ChatGPT, Bard, Copilot, etc.) for writing or completing their experiments.
- All submissions must reflect **your original work and understanding**.

## 2. Mobile Phones and Smart Gadgets

- The **use of mobile phones or any smart devices (watches, tablets, etc.) is strictly prohibited** during lab hours.
- Possession or usage during the lab will be **taken seriously** and may lead to reporting under academic misconduct.

## 3. No Copying or Plagiarism

- Students must not copy code, answers, or reports from others. **Each student is expected to work independently**.
- Copying from another student's file or sharing files will be considered malpractice for **both parties involved**.

## 4. Maintain Silence in the Lab

- Students are expected to maintain **complete silence** to ensure a focused and disturbance-free environment for all.
- Repeated disruptions will result in a **warning** and may affect lab performance grading.

## 5. Violation = Malpractice

- **Any violation of the above rules** will be strictly dealt with under the category of **malpractice**.
- Consequences may include **marks deduction, lab session cancellation, or escalation to disciplinary authorities**.