Flash

Here’s a **combined summary of all flashpoints** from **Units 1 to 5** of R Programming. This is designed as a one-stop revision tool for last-minute prep, memory refresh, or note-making in Joplin.

✅ R Programming Summary Flashpoints (Units 1–5)

🔵 Unit 1: Introduction to R Programming

- **R is a language** for statistical computing, data analysis, and visualization.
- **RStudio is an IDE** that makes writing and managing R code easier.
- **Packages** extend R’s features; CRAN has 15,000+.
- R supports **various data structures**: vectors, matrices, data frames, lists.
- **R integrates** with C/C++, Python, Java.
- **RStudio’s 4 main panels**: Source, Console, Environment, Files/Plots/Help.
- Quit RStudio via GUI or `q()`; save workspace optionally.
- **Always save scripts**, not just the workspace image.

🟣 Unit 2: Data Structures and Manipulation

- Use `<-` (recommended), `=`, or `->` to assign variables.
- **Conditional statements**: `if`, `else if`, `else` to control logic flow.
- Use **functions** to write modular code with `function()`.
- **Vectors** are the most basic structure: homogeneous and 1D.
- R is **dynamically typed** — it infers data type automatically.
- Use logical operators: `==`, `!=`, `>`, `<`, `&`, `|`.

🟡 Unit 3: Packages and Functions

- **Packages** = mini toolkits to add functionality.
- Install using `install.packages()`; load using `library()`.
- Define functions using:

  ```r
  name <- function(args) { return(value) }
  ```

- Use `read.csv()`, `read.table()`, `read_excel()` to import data.
- Built-in vs user-defined vs anonymous functions.
- Functions make code **cleaner, reusable, and modular**.

🟢 Unit 4: Matrices, Arrays, and Lists

- **Matrix**: 2D, homogeneous; created using `matrix()`.
- Operations: `+`, `-`, `*`, `%*%` (matrix multiplication), `t()` (transpose).
- Use `rbind()`, `cbind()` to add rows/cols; use negative indexing to remove.
- **Array**: multi-dimensional matrix; created with `array(dim = c(x,y,z))`.
- **List**: flexible container for **mixed-type data**; use `list()`.
- Access list items using `$name`, `[[index]]`.

🔴 Unit 5: Data Frames

- **Data frame** = table where **columns can have different types**.
- Create using `data.frame()` with vectors of same length.
- Access with `$`, `df[row,col]`, or `df[ , col]`.
- **Factors** store categorical data efficiently, used in modeling.
- Use `merge()` to join data frames by common column.
- Use `apply(X, MARGIN, FUN)` for row/column-wise ops.
- `sapply()` applies a function to each element of a vector.
- Use custom functions + logic to transform data (e.g., grading).
- **Scope issue**: inside function vs global — use `<<-` to assign globally.

Would you like me to give this in **Markdown format** for Joplin, a downloadable **PDF**, or flashcard style for fast revision?