Lab EXP-5 – Programming for Data Science-28-08-2025

Total Marks: 100

Topic: Solar System Data Representation and Analysis

Understanding Kepler's Third Law (kepler_k)

In this lab exercise, we calculated a value called kepler_k based on Kepler's Third Law of Planetary Motion. This law describes the relationship between a planet's orbital period and its average distance from the Sun.

The Law

Kepler's Third Law states that the square of a planet's orbital period (P) is directly proportional to the cube of its average distance from the Sun (a):

$$P^2 / a^3 = k$$

Where:
- P = Orbital period (in Earth years)
- a = Semi-major axis, i.e., average distance from the Sun (in Astronomical Units, AU)
- k = A constant value, which is approximately equal to 1 for all planets orbiting the Sun.

Why is k ≈ 1?

If we measure orbital period in Earth years and distance from the Sun in AU, then the value of k should be close to 1 for all planets orbiting the Sun. Small deviations from 1 are due to rounding or approximations in the data.

Example: Earth

For Earth:
- Orbital Period (P) = 365 days = 1 year
- Distance (a) = 1 AU

Thus:
$k = P^2 / a^3 = 1^2 / 1^3 = 1$

Example: Jupiter

For Jupiter:
- Orbital Period (P) ≈ 11.86 years
- Distance (a) ≈ 5.2 AU

Thus:
$k = (11.86^2) / (5.2^3) \approx 1.0$

Dr.Trilok Nath Pandey, SCOPE, VIT Chennai, This confirms that the dataset is consistent with Kepler's Third Law.

Conclusion

The kepler_k values calculated for different planets demonstrate that the orbital mechanics of the Solar System align with Kepler's Third Law. This law provides a simple yet powerful way to verify the consistency of planetary data in terms of orbital periods and distances from the Sun.

Instructions for Students

1. Write your solution in R programming language with appropriate comments explaining each step of your code. ***Ensure that the last four digits of your registration number are included in all variable names and user defined functions used in your program***.
   You must use the following R data structures and concepts in your program:
   - Array
   - Matrix
   - List
   - Data Frame
   - User-Defined Functions
2. The solution should be written neatly with correct indentation and comments.
3. Prepare a Lab Report containing:
   - Problem statement
   - Complete R code (with comments)
   - Output screenshots
   - Short analysis of the results (3–4 sentences)
4. The report must be submitted in .docx format with the following naming convention: Name_RegistrationNumber.docx
   Example: RaviKumar_123456.docx
5. Upload the report in the LMS before the deadline.
6. Any form of plagiarism or use of AI tools will be considered academic malpractice.

Problem Statement (100 Marks)

The Solar System consists of 8 planets revolving around the Sun, each with properties such as diameter, distance from the Sun, orbital period, and number of moons.

Your task is to design an R program that organizes and analyzes Solar System data using multiple data structures.

Part A: Using Array (10 Marks)

- Create an array that stores the names of the planets and their corresponding orbital periods in days.
- Display the orbital period of Earth using array indexing.

Part B: Using Matrix (15 Marks)

- Create a numeric matrix that stores the diameter (in km) and distance from Sun (in million km) for all 8 planets.
- Extract and display the values for the outer planets (Jupiter to Neptune).

Part C: Using List (15 Marks)

- Create a list that stores the following information for Earth:
  - Name
  - Diameter (km)
  - Distance from Sun (million km)
  - Number of moons
- Access and display each element of the list separately with proper comments.

Part D: Using Data Frame (30 Marks)

1. Create a data frame with the following columns:
   - Planet
   - Diameter_km
   - Distance_MillionKm
   - Orbital_Period_days
   - No_of_Moons
2. Perform the following operations:
   - Display all terrestrial planets (Mercury, Venus, Earth, Mars).
   - Find and display the planet with the maximum number of moons.
   - Sort planets in ascending order of their distance from the Sun.

Part E: User-Defined Functions (30 Marks)

1. Write a function that converts orbital period in days to Earth years (365 days = 1 year). Apply it to all planets and add the result as a new column in your data frame. 2. Write another function that, given a planet's name, returns a short summary of its properties (diameter, distance, number of moons) using the data frame.
2. Write a function to find Kepler k for all the planets.

Marking Rubric (100 Marks)

Component	Marks	Criteria
Part A: Array	**10**	**Correct array creation and indexing**
Part B: Matrix	**15**	**Proper matrix construction and subsetting**
Part C: List	**15**	**Correct list creation and element access**
Part D: Data Frame	**30**	**Correct data frame creation, subsetting, sorting, and analysis**

Dr.Trilok Nath Pandey, SCOPE, VIT Chennai,

Part E: Functions	**30**	**Correct implementation of functions and application to data**
Report Presentation	**Included**	**Code comments, formatting, and adherence to naming rules**

Total = 100 Marks