# Solar System Data Representation and Analysis - Lab EXP 5

Student Name: Sparsh Karna

Registration Number: 23BDS1172

Date: 28 August 2025

## Problem Statement

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. The objective is to design an R program that organizes and analyzes Solar System data using arrays, matrices, lists, data frames, and user-defined functions. Additionally, Kepler's Third Law (P² / a³ = k) will be used to verify orbital consistency.

## R Code Implementation

# ================================  
# Solar System Data Representation and Analysis  
# Registration Number: 23BDS1172  
# ================================  
  
cat("=== Solar System Lab EXP-5 ===\n\n")  
  
# -------------------------------  
# Part A: Array (10 Marks)  
# -------------------------------  
planet\_names\_1172 <- c("Mercury","Venus","Earth","Mars",  
 "Jupiter","Saturn","Uranus","Neptune")  
orbital\_period\_days\_1172 <- c(88,225,365,687,4333,10759,30687,60190)  
  
# Create array of planets and periods  
planet\_array\_1172 <- array(c(planet\_names\_1172, orbital\_period\_days\_1172),  
 dim=c(8,2),  
 dimnames=list(NULL,c("Planet","Orbital\_Period\_Days")))  
  
# Display orbital period of Earth  
cat("Orbital period of Earth (days):", orbital\_period\_days\_1172[3], "\n\n")  
  
# -------------------------------  
# Part B: Matrix (15 Marks)  
# -------------------------------  
# Diameter (km) and Distance from Sun (million km)  
diameter\_km\_1172 <- c(4879,12104,12742,6779,139820,116460,50724,49244)  
distance\_million\_km\_1172 <- c(57.9,108.2,149.6,227.9,778.5,1434,2871,4495)  
  
planet\_matrix\_1172 <- matrix(c(diameter\_km\_1172, distance\_million\_km\_1172),  
 nrow=8, ncol=2,  
 dimnames=list(planet\_names\_1172,  
 c("Diameter\_km","Distance\_million\_km")))  
  
cat("Outer planets (Jupiter to Neptune):\n")  
print(planet\_matrix\_1172[5:8, ])  
cat("\n")  
  
# -------------------------------  
# Part C: List (15 Marks)  
# -------------------------------  
earth\_list\_1172 <- list(  
 Name="Earth",  
 Diameter\_km=12742,  
 Distance\_million\_km=149.6,  
 Moons=1  
)  
  
cat("Earth List Details:\n")  
cat("Name:", earth\_list\_1172$Name, "\n")  
cat("Diameter (km):", earth\_list\_1172$Diameter\_km, "\n")  
cat("Distance from Sun (million km):", earth\_list\_1172$Distance\_million\_km, "\n")  
cat("Moons:", earth\_list\_1172$Moons, "\n\n")  
  
# -------------------------------  
# Part D: Data Frame (30 Marks)  
# -------------------------------  
moons\_1172 <- c(0,0,1,2,79,83,27,14)  
  
planet\_df\_1172 <- data.frame(  
 Planet=planet\_names\_1172,  
 Diameter\_km=diameter\_km\_1172,  
 Distance\_MillionKm=distance\_million\_km\_1172,  
 Orbital\_Period\_days=orbital\_period\_days\_1172,  
 No\_of\_Moons=moons\_1172,  
 stringsAsFactors=FALSE  
)  
  
cat("Terrestrial planets:\n")  
print(subset(planet\_df\_1172, Planet %in% c("Mercury","Venus","Earth","Mars")))  
cat("\n")  
  
cat("Planet with maximum moons:\n")  
print(planet\_df\_1172[which.max(planet\_df\_1172$No\_of\_Moons), ])  
cat("\n")  
  
cat("Planets sorted by distance:\n")  
print(planet\_df\_1172[order(planet\_df\_1172$Distance\_MillionKm), ])  
cat("\n")  
  
# -------------------------------  
# Part E: User-Defined Functions (30 Marks)  
# -------------------------------  
  
#' @title Convert Orbital Period  
#' @description Converts orbital period from days to Earth years (365 days = 1 year)  
#' @param days Orbital period in days  
#' @return Orbital period in years (rounded to 2 decimals)  
convert\_to\_years\_1172 <- function(days) {  
 return(round(days/365, 2))  
}  
planet\_df\_1172$Orbital\_Period\_years <- convert\_to\_years\_1172(planet\_df\_1172$Orbital\_Period\_days)  
  
#' @title Planet Summary  
#' @description Returns summary information for a given planet  
#' @param name Name of the planet (string)  
#' @return Character string summarizing diameter, distance, and moons  
planet\_summary\_1172 <- function(name) {  
 row <- subset(planet\_df\_1172, Planet==name)  
 if (nrow(row)==0) {  
 return(paste("Planet", name, "not found."))  
 }  
 return(paste("Planet:", row$Planet,  
 "| Diameter:", row$Diameter\_km, "km",  
 "| Distance:", row$Distance\_MillionKm, "million km",  
 "| Moons:", row$No\_of\_Moons))  
}  
  
cat("Summary for Jupiter:\n")  
cat(planet\_summary\_1172("Jupiter"), "\n\n")  
  
#' @title Kepler k Calculator  
#' @description Computes Kepler's constant k = P² / a³ for planets  
#' @param period\_days Orbital period in days  
#' @param distance\_AU Distance from Sun in Astronomical Units (AU)  
#' @return Numeric value of Kepler's k (rounded to 3 decimals)  
kepler\_k\_1172 <- function(period\_days, distance\_AU) {  
 P\_years <- period\_days / 365  
 return(round((P\_years^2) / (distance\_AU^3), 3))  
}  
  
planet\_df\_1172$Distance\_AU <- distance\_million\_km\_1172 / 149.6  
planet\_df\_1172$Kepler\_k <- kepler\_k\_1172(planet\_df\_1172$Orbital\_Period\_days,  
 planet\_df\_1172$Distance\_AU)  
  
cat("Final Data Frame with Orbital Period in Years and Kepler k:\n")  
print(planet\_df\_1172)  
cat("\n=== Analysis Complete ===\n")

## Output Screenshots

[Paste your RStudio/Console output screenshots here]

## Analysis

The Solar System data was successfully represented using arrays, matrices, lists, and data frames. The user-defined functions allowed conversion of orbital periods to years, retrieval of planet summaries, and computation of Kepler's constant (k). The calculated Kepler\_k values are approximately 1 for all planets, which confirms the consistency of the dataset with Kepler’s Third Law. This demonstrates the correctness of planetary data representation and analysis.