II Year - I Semester

L	T	P	C
4	0	0	3

STATISTICS WITH R PROGRAMMING

OBJECTIVE:

After taking the course, students will be able to

- Use R for statistical programming, computation, graphics, and modeling,
- Write functions and use R in an efficient way,
- Fit some basic types of statistical models
- Use R in their own research,
- Be able to expand their knowledge of R on their own.

UNIT-I:

Introduction, How to run R, R Sessions and Functions, Basic Math, Variables, Data Types, Vectors, Conclusion, Advanced Data Structures, Data Frames, Lists, Matrices, Arrays, Classes**.**

UNIT-II:

R Programming Structures, Control Statements, Loops, - Looping Over Nonvector Sets,- If-Else, Arithmetic and Boolean Operators and values, Default Values for Argument, Return Values, Deciding Whether to explicitly call return- Returning Complex Objects, Functions are Objective, No Pointers in R, Recursion, A Quicksort Implementation-Extended Extended Example: A Binary Search Tree.

UNIT-III:

Doing Math and Simulation in R, Math Function, Extended Example Calculating Probability- Cumulative Sums and Products-Minima and Maxima- Calculus, Functions Fir Statistical Distribution, Sorting, Linear Algebra Operation on Vectors and Matrices, Extended Example: Vector cross Product- Extended Example: Finding Stationary Distribution of Markov Chains, Set Operation, Input /out put, Accessing the Keyboard and Monitor, Reading and writer Files,

UNIT-IV:

Graphics, Creating Graphs, The Workhorse of R Base Graphics, the plot() Function – Customizing Graphs, Saving Graphs to Files.

UNIT-V:

Probability Distributions, Normal Distribution- Binomial Distribution- Poisson Distributions Other Distribution, Basic Statistics, Correlation and Covariance, T-Tests,-ANOVA.

UNIT-VI:

Linear Models, Simple Linear Regression, -Multiple Regression Generalized Linear Models, Logistic Regression, - Poisson Regression- other Generalized Linear Models-Survival Analysis, Nonlinear Models, Splines- Decision- Random Forests,

OUTCOMES:

At the end of this course, students will be able to:

- List motivation for learning a programming language
- Access online resources for R and import new function packages into the R workspace
- Import, review, manipulate and summarize data-sets in R
- Explore data-sets to create testable hypotheses and identify appropriate statistical tests
- Perform appropriate statistical tests using R Create and edit visualizations with

TEXT BOOKS:

1) The Art of R Programming, Norman Matloff, Cengage Learning
2) R for Everyone, Lander, Pearson

REFERENCE BOOKS:

1) R Cookbook, PaulTeetor, Oreilly.
2) R in Action,Rob Kabacoff, Manning

II Year - I Semester

L	T	P	C
4	0	0	3

PYTHON PROGRAMMING

OBJECTIVES:

- Introduction to Scripting Language
- Exposure to various problems solving approaches of computer science

UNIT – I:

Introduction: History of Python, Need of Python Programming, Applications Basics of Python Programming Using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input-Output, Indentation.

UNIT – II:

Types, Operators and Expressions: Types - Integers, Strings, Booleans; Operators- Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations Control Flow- if, if-elif-else, for, while, break, continue, pass

UNIT – III:

Data Structures Lists - Operations, Slicing, Methods; Tuples, Sets, Dictionaries, Sequences. Comprehensions.

UNIT – IV:

Functions - Defining Functions, Calling Functions, Passing Arguments, Keyword Arguments, Default Arguments, Variable-length arguments, Anonymous Functions, Fruitful Functions(Function Returning Values), Scope of the Variables in a Function - Global and Local Variables.

Modules: Creating modules, import statement, from. Import statement, name spacing,

Python packages, Introduction to PIP, Installing Packages via PIP, Using Python Packages

UNIT – V:

Object Oriented Programming OOP in Python: Classes, 'self variable', Methods, Constructor Method, Inheritance, Overriding Methods, Datahiding,

Error and Exceptions: Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User Defined Exceptions

UNIT – VI:

Brief Tour of the Standard Library - Operating System Interface **-** String Pattern Matching, Mathematics, Internet Access, Dates and Times, Data Compression, Multithreading, GUI Programming, Turtle Graphics

Testing: Why testing is required ?, Basic concepts of testing, Unit testing in Python, Writing Test cases, Running Tests.

OUTCOMES:

- Making Software easily right out of the box.
- Experience with an interpreted Language.
- To build software for real needs.
- Prior Introduction to testing software

TEXT BOOKS

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson
2. Learning Python, Mark Lutz, Orielly

Reference Books:

1. Think Python, Allen Downey, Green Tea Press
2. Core Python Programming, W.Chun, Pearson.
3. Introduction to Python, Kenneth A. Lambert, Cengage

II Year - I Semester

L	T	P	C
0	0	3	2

PYTHON PROGRAMMING LAB

Exercise 1 - Basics

a) Running instructions in Interactive interpreter and a Python Script
b) Write a program to purposefully raise Indentation Error and Correct it

Exercise 2 - Operations

a) Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)
b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

Exercise - 3 Control Flow

a) Write a Program for checking whether the given number is a even number or not.

b) Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, . . . , 1/10

c) Write a program using a for loop that loops over a sequence. What is sequence ?

d) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

Exercise 4 - Control Flow - Continued

a) Find the sum of all the primes below two million.
Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

b) By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

Exercise - 5 - DS

a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure
b) Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

Exercise - 6 DS - Continued

a) Write a program combine_lists that combines these lists into a dictionary.

b) Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?

Exercise - 7 Files

a) Write a program to print each line of a file in reverse order.
b) Write a program to compute the number of characters, words and lines in a file.

Exercise - 8 Functions

a) Write a function ball_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.

   Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius

   If (distance between two balls centers) <= (sum of their radii) then (they are colliding)

b) Find mean, median, mode for the given set of numbers in a list.

Exercise - 9 Functions - Continued

a) Write a function nearly_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.
b) Write a function dups to find all duplicates in the list.
c) Write a function unique to find all the unique elements of a list.

Exercise - 10 - Functions - Problem Solving

a) Write a function cumulative_product to compute cumulative product of a list of numbers.
b) Write a function reverse to reverse a list. Without using the reverse function.
c) Write function to compute gcd, lcm of two numbers. Each function shouldn't exceed one line.

Exercise 11 - Multi-D Lists

a) Write a program that defines a matrix and prints
b) Write a program to perform addition of two square matrices
c) Write a program to perform multiplication of two square matrices

Exercise - 12 - Modules

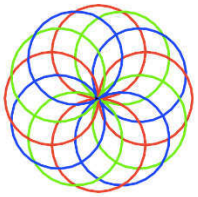
a) Install packages requests, flask and explore them. using (pip)
b) Write a script that imports requests and fetch content from the page. Eg. (Wiki)
c) Write a simple script that serves a simple HTTPResponse and a simple HTML Page

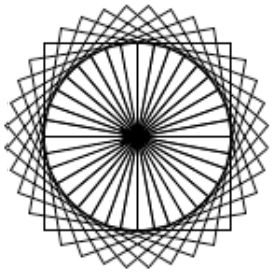
Exercise - 13 OOP

a) Class variables and instance variable and illustration of the self variable
   i) Robot
   ii) ATM Machine

Exercise - 14 GUI, Graphics

1. Write a GUI for an Expression Calculator using tk
2. Write a program to implement the following figures using turtle





Exercise - 15 - Testing

a) Write a test-case to check the function even_numbers which return True on passing a list of all even numbers
b) Write a test-case to check the function reverse_string which returns the reversed string

Exercise - 16 - Advanced

a) Build any one classical data structure.
b) Write a program to solve knapsack problem.