

Semester	FOUR
Course	Skill
Paper Code	S2ST230411P
Paper Title	Programming in Python and R
No. of Credits	3
Theory / Practical /Composite	Practical
Minimum No. of preparatory hours per week a student has to devote	Module I : 2 periods/week Module II: 2 periods/week
Module	2

Course Outcomes for Module I

1. Remember the syntax for basic input-output operations, arithmetic and logical operators, and standard data structures like lists, tuples, and dictionaries.
2. Understand the conceptual framework of program flow control, including conditions, branching, and various loop structures.
3. Apply specialized libraries such as numpy, pandas, and scipy to perform data manipulation and solve problems related to linear algebra and statistical inference.
4. Analyze Python code to identify and debug programming errors during the development of statistical computation scripts.
5. Evaluate statistical measures and mathematical functions by utilizing built-in functions and the math library.
6. Create visual representations and diagrams of data or functions using the matplotlib library.

Course Outcomes for Module II

1. Remember the specific commands and functions required for professional file handling, including the importing and exporting of data from external sources.
2. Understand the utility of the optim() function and its various arguments for performing mathematical optimization within R.
3. Apply conditional statements, loop structures, and user-defined functions to automate complex statistical procedures and mathematical modeling.
4. Analyze statistical properties such as bias, Mean Squared Error (MSE), and the coverage of confidence intervals through long-run relative frequency simulations.
5. Evaluate the empirical level and power of statistical tests by simulating data from

different finite and infinite probability distributions.
6. Create robust statistical simulations using the <code>set.seed()</code> command and the <code>curve()</code> function to illustrate theoretical sampling distributions and probability estimates.

Syllabus

Unit/Module	Content	No. of lectures	CO mapping	Cognitive levels
Module I				
Unit 1	Python Basics and Data Structures: Introduction to Python programming, basic input output operations, string operations, arithmetic operators, logical operators, lists, tuples, and dictionaries.	6	CO1	K1
Unit 2	Programming in Python: Conditions and branching, loops, user-defined functions.	9	CO2 CO5	K2 K5
Unit 3	Use of libraries: math, numpy, pandas, matplotlib, random, scipy, sympy.	24	CO3 CO4 CO6	K3 K4 K6
Module II				
Unit 1	File Handling: Importing and exporting data from/to other software.	3	CO1	K1
Unit 2	Programming in R: Conditional statements: if, if else. Loop structures. User defined functions. The <code>curve()</code> function.	11	CO2	K2
Unit 3	Statistical Simulations: Drawing random samples from different finite and infinite probability distributions – the <code>set.seed()</code> command. Illustrations through statistical problems: probability estimates by long-run relative frequencies, sampling distribution, bias	25	CO3 CO4 CO5 CO6	K3 K4 K5 K6

	and MSE's of estimates, coverage of confidence intervals, calculating empirical level and power of tests. Optimisation of functions – the optim() function and its various arguments.			
--	---	--	--	--

Reading/Reference list

1. Guttag, J. V. (2021): Introduction to Computation and Programming Using Python, Third Edition, MIT Press.
2. Nelli, F. (2018): Python Data Analytics, 2nd Edition, Apress.
3. Dalgaard, P: Introductory Statistics with R, Springer Publications, 2nd edition, 2008.
4. Maindonald, J. & Braun, J.: Data Analysis and Graphics Using R, Cambridge University Press, Cambridge, 2nd edition, 2007.

Evaluation

Paper Structure	Introduction to Python		Advanced R		Total
	CA	END SEM	CA	END SEM	
	25	NA	25	NA	

CO	CO Description for Module 1	Cognitive levels
CO1	Remember the syntax for basic input-output operations, arithmetic and logical operators, and standard data structures like lists, tuples, and dictionaries.	K1
CO2	Understand the conceptual framework of program flow control, including conditions, branching, and various loop structures.	K2
CO2	Apply specialized libraries such as numpy, pandas, and scipy to perform data manipulation and solve problems related to linear algebra and statistical inference.	K2
CO4	Analyze Python code to identify and debug programming errors during the development of statistical computation scripts.	K4
CO5	Evaluate statistical measures and mathematical functions by utilizing built-in functions and the math library.	K5
CO6	Create visual representations and diagrams of data or functions using the matplotlib library.	K6

CO	CO Description for Module II	Cognitive levels
CO1	Remember the specific commands and functions required for professional file handling, including the importing and exporting of data from external sources.	K1
CO2	Understand the utility of the <code>optim()</code> function and its various arguments for performing mathematical optimization within R.	K2
CO2	Apply conditional statements, loop structures, and user-defined functions to automate complex statistical procedures and mathematical modeling.	K2
CO4	Analyze statistical properties such as bias, Mean Squared Error (MSE), and the coverage of confidence intervals through long-run relative frequency simulations.	K4
CO5	Evaluate the empirical level and power of statistical tests by simulating data from different finite and infinite probability distributions.	K5
CO6	Create robust statistical simulations using the <code>set.seed()</code> command and the <code>curve()</code> function to illustrate theoretical sampling distributions and probability estimates.	K6