CURRICULUM CHANGE PROPOSAL

APPROVAL PAGE

Proposal Title: Creation of STAT 4329: Statistical Programming

College: Science **Department: Mathematical Sciences**

DEPARTMENT CHAIR

I have read the enclosed proposal and approve this proposal on behalf of the department.

Signature _____ Date 12/13/19

COLLEGE CURRICULUM COMMITTEE CHAIR

I have read the enclosed documents and approve the proposal on behalf of the college curriculum committee.

Signature _____ Date 1/30/20

COLLEGE DEAN

I have read the enclosed documents and approve the proposal on behalf of the college. I certify that the necessary funds will be allocated by the college in support of this proposal.

Signature _____ Date 1/30/20

UNDERGRADUATE CURRICULUM CHANGE MEMO

Date: 10/15/2019

From: Amy Wagler, Dept of Mathematical Sciences

Through: Christina Mariani, Dept of Mathematical Sciences

Through: Robert A. Kirken, College of Science

To: Carla Ellis, Chair of UGCC

Proposal Title: Creation of STAT 4329: Statistical Programming

---

We propose creation of a new undergraduate course covering the foundations of machine learning. It will focus on the two preeminent programming language for statistical analysis, R and Python. However, this course already runs as a beginning graduate level course, STAT 5329. Due to large demand for statistical programming coursework at the senior undergraduate level, we would like to cross-list our Master's level course STAT 5329: Statistical Programming to be a senior undergraduate course STAT 4329: Statistical Programming. This course focuses on teaching fundamental concepts in R and Python programming to prepare students for advanced statistical analysis in these languages.

COURSE ADD

All fields below are required

College : Science **Department :** Mathematical Sciences

Rationale for adding the course:
We propose creation of a new undergraduate course on statistical programming. It will focus on the two preeminent programming language for statistical analysis, R and Python. However, this course already runs as a beginning graduate level course, STAT 5329. Due to large demand for statistical programming coursework at the senior undergraduate level, we would like to cross-list our Master's level course STAT 5329: Statistical Programming to be a senior undergraduate course STAT 4329: Statistical Programming. This course focuses on teaching fundamental concepts in R and Python

All fields below are required

Subject Prefix and # STAT 4329

Title (29 characters or fewer): Statistical Programming

Dept. Administrative Code : 1870

CIP Code 27.0501.00.01

Departmental Approval Required ☐Yes ☐No

Course Level ☒UG ☐GR ☐DR ☐SP

Course will be taught: ☒ Face-to-Face ☐ Online ☒ Hybrid

How many times may the course be taken for credit? (Please indicate 1-9 times): 1

Should the course be exempt from the "Three Repeat Rule?" ☐Yes ☒No

Grading Mode: ☒Standard ☐Pass/Fail ☐Audit

Description (600 characters maximum):
Introduces students to the principles and concepts of programming in Python and R. Students will be able to manipulate data, create summary reports and lists, edit and interactively debug code, manage complex data sets, transform and generate data, create effective graphics for data visualization, create user-defined functions and Python calculations, handle various data formats.

Contact Hours (per week): 3 Lecture Hours Lab Hours Other

Types of Instruction (Schedule Type): Select all that apply

☒A	Lecture	☐ H	Thesis
☐ B	Laboratory	☐ I	Dissertation
☐ C	Practicum	☐ K	Lecture/Lab Combined
☐ D	Seminar	☐ O	Discussion or Review (Study Skills)
☐ E	Independent Study	☐ P	Specialized Instruction

☐ F Private Lesson ☐ Q Student Teaching

Fields below if applicable

If course is taught during a part of term in addition to a full 16-week term please indicate the length of the course (ex., 8 weeks):

TCCN (Use for lower division courses) :

Prerequisite(s):		
Course Number/ Placement Test	**Minimum Grade Required/ Test Scores**	**Concurrent Enrollment Permitted? (Y/N)**
STAT 4385 or equivalent	C	Y

Corequisite Course(s):

Equivalent Course(s):

Restrictions:

Classification	
Major	

THE UNIVERSITY OF TEXAS AT EL PASO
COLLEGE OF SCIENCE
DEPARTMENT OF MATHEMATICAL SCIENCES

Course #:	STAT 4329	
Course Title:	Statistical Programming	
Credit Hrs:	3	
Term:	Fall 2019	
Course Meetings & Location:	Online and Bell Hall Computer Lab	
Prerequisite Courses:	Department Approval Required	
Course Fee: (if applicable)	NA	
Instructor:		
Office Location:		
Contact Info:		Phone #
		E-mail address
	744-6502	Fax #
		Emergency Contact
Office Hrs:	TBA	
Textbook(s), Materials:	Required:	none
	Suggested:	The Art of R Programming, Matloff
		Python for Data Analysis McKinney

Course Objectives (Learning Outcomes): Introduces students to the principles and concepts of programming in Python and R. Students will be able to manipulate data, create summary reports and lists, edit and interactively debug code, manage complex data sets, transform and generate data, create effective graphics for data visualization, create user-defined functions and Python calculations, handle various data formats.

Course Activities/Assignments: Each week will require in-class work completed by the end of the week. Additionally, bi-weekly assignments are given. A midterm and final exam will also be administered during the semester.

A grading rubric will be used for the bi-weekly assignments. There will be a mid-term and final in-class exam. Daily in-class assignments are graded for completeness only. There is a minimum score (40%) that you are expected to obtain on both the midterm and final exams. If the mean of both exams does not exceed 40%, then the homework grade will be dropped from your course grade.

Course Schedule: R Schedule: This is a tentative schedule for this semester.

Week 1: R Intro, help and packages
Week 2: vectorized calculations, matrices and arrays
Week 3: lists and data frames, programming structures
Week 4: simulations and efficient programming
Week 5: permuting and bootstrapping
Week 6: creating R functions (packages)
Week 7: final topics and MIDTERM EXAM (in class)

Python Schedule: This is a tentative schedule for this semester.

Week 8: Python Introduction: basics, variables, and operators
Week 9: Loops and conditionals
Week 10: Functions
Week 11: Lists, Strings and Tools for Statistical Modeling
Week 12: Stings and Control flow
Week 13: File I/O and Modules
Week 14: Analysis of algorithms and FINAL EXAM (in class)

Grading Policy: 20% Weekly Lab assignments
20% Midterm Exam (there is an in-class and take-home portion)
20% Final Exam (there is an in-class and take-home portion)
40% Homework Assignments (about 5)

Make-up Policy: If class is missed for a valid and documented reason, the daily in-class assignments may be made-up for full credit. All other assignments must be turned in on time.

Attendance Policy: You are expected to attend class so that you may turn in the in-class assignments and bi-weekly homework assignments.

Academic Integrity Policy: Please see http://academics.utep.edu/Default.aspx?tabid=23785

Civility Statement: This is a class where participation is required. You will be seated in front of a computer all class period and you are expected to follow the lecture/discussion and at various times complete in-class assignments. You are not to browse the internet during class time or work on any other material. If you regularly do not complete in-class assignments in a satisfactory manner, participate in class, or if you work on other material in class you will have points deducted from your in-class assignments portion of your grade.

Disability Statement: If a student has or suspects she/he has a disability and needs an accommodation, he/she should contact the Disabled Student Services Office (DSSO) at 747-5148 or at <dss@utep.edu> or go to Room 106 Union East Building. The student is responsible for presenting to the instructor any DSS accommodation letters and instructions.

Military Statement: If you are a military student with the potential of being called to military service and /or training during the course of the semester, you are encouraged to contact me as soon as possible.

UTEP College of Science Policies: The UTEP Spring 2019 drop deadline is November 1, 2019. The College of Science will remain aligned with the University and not approve any drop requests after that date.

All grades of Incomplete must be accompanied by an Incomplete Contract that has been signed by the instructor of record, student, departmental chair, and the dean. Although UTEP will allow a maximum of one year to complete this contract, the College of Science requests it be limited to month based upon completion data. A grade of Incomplete is only used in extraordinary circumstances confined to a limited event such as a missed exam, project, or lab. If the student has missed a significant amount of work (e.g. multiple assignments or tasks), a grade of Incomplete is not appropriate or warranted.

Criterion	Approx. % of Grade	Excellent (100%)	Adequate (80%)	Poor (60%)	Not Met (0%)
Program Specifications / Correctness	50%*	No errors, program always works correctly and meets the specification(s).	Minor details of the program specification are violated, program functions incorrectly for some inputs.	Significant details of the specification are violated, program often exhibits incorrect behavior.	Program only functions correctly in very limited cases or not at all.
Readability	20%	Code is clean, understandable, and well-organized.	Minor issues with consistent indentation, use of whitespace, variable naming, or general organization.	At least one major issue with indentation, whitespace, variable names, or organization.	Major problems with at three or four of the readability subcategories.
Documentation	5%	Code is well-commented.	One or two places that could benefit from comments are missing them **or** the code is *overly* commented	File header missing, complicated lines or sections of code uncommented or lacking meaningful comments.	No file header or comments present.
Code Efficiency	20%	Code uses the best approach in every case.	Code uses poorly-chosen approaches (though correct in result) in at least one place.	Code uses poorly-chosen approaches (though correct in result) in at least two places.	Many things in the code could have been accomplished in an easier, faster, or otherwise better fashion.
Assignment Specifications	5%	No errors	Minor details of the assignment specification are violated, such as files named incorrectly or extra instructions slightly misunderstood.	Minor details of the assignment specification are violated, such as files named incorrectly or extra instructions significantly misunderstood.	Significant details of the specification are violated, such as extra instructions ignored or entirely misunderstood.