College of San Mateo
Official Course Outline

1. **COURSE ID:** CIS 140 **TITLE:** Big Data Analytics
   Units: 4.0 units **Hours/Semester:** 48.0-54.0 Lecture hours; 48.0-54.0 Lab hours; and 96.0-108.0 Homework hours
   Method of Grading: Grade Option (Letter Grade or Pass/No Pass)
   Recommended Preparation:
   Completion of CIS 254.

2. **COURSE DESIGNATION:**
   Degree Credit
   Transfer credit: CSU; UC

3. **COURSE DESCRIPTIONS:**
   Catalog Description:
   Introduction to the field of Big Data, its concepts and technologies, as well as current programming environments such as R and Python. Students will explore the roles of a data scientist in terms of network architecture, data analytics and predictive analysis. Fundamental questions of data science and scenarios appropriate for each will be discussed. Differentiation among raw data, clean data, and tidy data; and tools to convert data to/from these formats will be covered. Effective management of large data in single and distributed computing environments, including managing data redundancy and failures, will be covered. Introduction to Data Mining and Machine Learning techniques: classification, correlation, cluster analysis, frequent patterns and data visualization will be introduced. Intended for students with previous programming experience.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**
   Upon successful completion of this course, a student will meet the following outcomes:
   1. Read Structured Data from various sources
   2. Write user-defined functions
   3. Reshape data to support different analyses
   4. Use program functions for descriptive and inferential statistics
   5. Manage Big Data in single and distributed computing environments.
   6. Display data using graphics and data visualization.

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**
   Upon successful completion of this course, a student will be able to:
   1. Read Structured Data from various sources
   2. Write user-defined functions
   3. Reshape data to support different analyses
   4. Use program functions for descriptive and inferential statistics
   5. Manage Big Data in single and distributed computing environments
   6. Display data using graphics and data visualization

6. **COURSE CONTENT:**
   Lecture Content:
   1. Introduction to R
   - What is R?
   - Installing R and RStudio
   - R language resources
   - Installing and using packages
   - Workspace

   2. Programming with R
   - Data Objects: Vectors, Matrices, Data Frames, and Lists
   - Variables
   - Local data import/export
   - Functions
   - Control statements
   - Data sorting
   - Merging data
   - Remodeling data
   - String manipulation
   - Regular expressions
   - Dates and time stamps

- Web data capture
- API data sources
- Connecting to an external database

3. Introduction to Data Science

- Data
- Scenarios
- Data architecture patterns
- Data analytics
- Data conversion tools
- Data in single and distributed environments
- Predictive analysis
- Correlation clustering
- Management of data redundancy and failure
- Testing
- Raw data
- Clean data
- Tidy data
- Network architecture

4. Principal Statistical Methods

- Descriptive Statistics
- Hypothesis testing
- Linear Regression
- Logistic Regression
- Non-parametric statistics

5. Data Graphics and Data Visualization

- Core concepts of data graphics and data visualization
- R graphics engines
    - Base
    - Grid
    - Lattice
    - ggplot2

Lab Content:

Lab assignments will include:
Data types and data structures
Flow control and looping
Writing and calling functions
Top-down design, testing
Debugging and functions as objects
Simple optimization and refactoring
Split/apply/combine pattern
Simulation
Optimization
Working with character data and regular expressions
Regular expressions and web scraping
Reshaping data and database access
Using data analytics and predictive analysis
Sample Big Data sets may include tax data, automotive data, social media data, stock market data, employment data, sports data, etc.
Final projects

7. **REPRESENTATIVE METHODS OF INSTRUCTION:**

Typical methods of instruction may include:

A. Lecture
B. Lab
C. Activity
D. Directed Study
E. Discussion
F. Experiments
G. Observation and Demonstration
H. Other (Specify): teacher will model problem solving techniques; teacher will create and manage an Internet conference for discussion of course topics; and students will work alone and in small groups to solve programming assignments.

8. **REPRESENTATIVE ASSIGNMENTS**

Representative assignments in this course may include, but are not limited to the following:

Writing Assignments:

The computer programming assignments provide hands on practice of the concepts covered in the readings. Weekly programming assignments will cover: Data types and data structures Flow control and looping Writing and calling functions Top-down design, testing Debugging and functions as objects Simple optimization and refactoring Split/apply/combine pattern Simulation Optimization Working with character data and regular

expressions Regular expressions and web scraping Reshaping data and database access Using data analytics and predictive analysis Sample Big Data sets may include tax data, automotive data, social media data, stock market data, employment data, sports data, etc. Final projects

Reading Assignments:

Reading assignments accompanied by self-test questions and running code examples. Studying posted lecture notes and relevant handouts. The reading assignment frames the concepts covered and provides the basic knowledge necessary to do the self-test questions and understand the sample code. The lecture notes and handouts provide a more in-depth look at topics and distills the information down to what the faculty thinks is most important.

9. **REPRESENTATIVE METHODS OF EVALUATION**

Representative methods of evaluation may include:

A. Exams/Tests
B. Group Projects
C. Homework
D. Lab Activities
E. Projects
F. Quizzes
G. Simulation
H. Written examination
I. Bi-weekly quizzes (short answer--from textbook material) to provide feedback to students and teacher. Assessment of student contributions during class discussion and project time. Individual programming assignments. Midterm and Final exams (short answer from textbook material, general problem solving (similar to in-class work), short program segments (similar to programming assignments). Assessment of group participation on course projects, including peer-assessment of participation and contribution to the group effort.

10. **REPRESENTATIVE TEXT(S):**

Possible textbooks include:

A. Lander. *R for Everyone: Advanced Analytics and Graphics*, 2nd ed. Addison-Wesley, 2017
B. Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. ed. O'Reilly Media, 2019
C. Ramasubramanian & Singh. *Machine Learning Using R: With Time Series and Industry-Based Use Cases in R*, 2nd ed. Packt Publishing, 2019
D. James, Witten et al. *An Introduction to Statistical Learning: with Applications in R*, 1st ed. Springer, 2017
E. Grolemund & Wickham. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*, 1st ed. O'Reilly, 2017
F. Leskovec, Jure, Anand Rajaraman, Jeffrey David Ullman. *Mining of Massive Datasets*, 3rd ed. ed. Cambridge University Press, 2020

Origination Date: November 2020
Curriculum Committee Approval Date: January 2021
Effective Term: Fall 2021
Course Originator: Mounjed Moussalem