Institute of Service Science, National Tsing Hua University

Business Analytics Using Computational Statistics (BACS)

Spring 2024

Instructor
Soumya Ray
soumya.ray@iss.nthu.edu.tw
Office: 850 TSMC Building

Teaching Assistants
TBA

Prerequisite knowledge:
You must have completed at least one statistics class before taking this course. This class focuses on the application of computational methods to statistics, and only lightly reviews fundamental statistics topics.

No prior programming knowledge is required but students with basic coding experience will have an advantage. Beginners must be willing to learn basic programming techniques through homework tutorials. We will teach intermediate programming techniques in class that build on the homework tutorials.

Course Description:
This class covers computational statistical methods used to understand and explain business phenomena, and develop tools that support managerial decision-making and assist consumer's decision-making.

Computational focus: We will use a *computational methods of programming and algorithms* to overcome limitations of data quality and quantity. We will learn to reshape data, simulate data and statistics, discover unseen dimensions in data, and create complex models of unobservable phenomena.

Methodological focus: We will use analytics to *understand, explain, and predict business phenomena* to inform decision making. We will describe and visualize data, create statistical models from domain knowledge, test our ideas against data; predict new outcomes; guard against fallacious use of statistics.

Skill focus: We will learn to *write data analytics code in R on par with industry standards*. Students will implement their own algorithms, write code that is highly readable and reusable, produce highly performant code, create bespoke visualizations, and apply different styles of analytic programming.

Software Tools:
Students will use the RStudio development environment to write code in the R programming language. *Our tools are all free and open-source*. Using R requires learning basic concepts in programming and maintaining code, which we will teach in class. If you wish to familiarize yourself with these tools before class begins, please start with the tutorials at `http://swirlstats.com`

Ethics Statement on Generative Artificial Intelligence and other Code Resources:
In accordance with the published Guidelines for Collaboration, Co-learning, and Cultivation of Artificial Intelligence Competencies in University Education, this course adopts the following policy: ***Conditionally open***.

Grounded in the principles of transparency and responsibility, this course encourages students to leverage Generative AI (GAI), like ChatGPT and Github CoPilot, and other coding resources, like StackOverflow, to enhance their learning and improve the quality of their course outputs. This means that students may use GAI or other coding resources but must briefly explain how GAI tools were used in each homework assignment they submit. Code copied from outside resources must be properly cited and credited. Moreover, students must not submit code from GAI or outside sources that they do not fully understand. We expect students to carefully study, verify, and suitably alter their GAI assisted solutions to match homework requirements and reflect their genuine learning. If code is discovered that was not properly understood and adaptated, or if the use of GAI was not disclosed, or if code copied form outside resources is not cited, instructors have the right to reevaluate the assignment or report or withhold scores. Students enrolled in this course agree to the above ethics statement if registering for the class.

Grading:
You will receive grades every week based on:

- *Tutorials and quizzes (2 pts):* occasional interactive tutorials and quizzes on readings.
- *Individual assignment report (4 pts)*: your homework assignments will be *anonymously peer-reviewed* by two other students will chosen at random each week; TAs will assign scores based on peer-review grade suggestions and comments. You may offer a *rebuttal* if you feel that you deserved a better grade, and our teaching team will arbitrate. Peer grading helps us manage a large class and allows students to appreciate alternative solutions.
- *Peer grading and review (2 pts)*: you must also grade two students' assignments each week. You will get a score for giving thoughtful and accurate scores and suggestions.

You can earn extra credit throughout the semester:

- *Outstanding Homework*: your peers may grant you an extra point for noteworthy coding or reports
- *Assistance Credit*: if you are mentioned by peers as having helped on submitted assignments.
- *Participation*: if you participate in class and on online discussions.

Reference Material:
Lecture slides will be provided before every class and made available online at our class website. Students will be occasionally given material from videos, research papers, practitioners' blogs, and so on.

Course Topics

Computational Perspective	*Extras*
Computation and Statistics Learning Computation Exploration, Inference, and Prediction Our Tools: R and Rstudio	*Industry: Peer Review and Social Learning* *Tutorial: Swirl to Learn R*
Description and Simulation Kernel Density Plots / Histograms Simulating Distributions Inferential Statistics	*Coding: Conceptualizing Variables* *Coding: Simulating Data from Distributions* *Computing: Binary Representation of Numbers* *Tutorial: Swirl to Learn R*
Computational Intervals Functions and Iterations Describing Distributions Confidence Intervals Resampling	*Coding: Writing Your Own Functions* *Coding: Functional and Vectorized Iteration* *Coding: Performance Benchmarking* *Simulation: Sampling Statistics* *Tutorial: Swirl to Learn R*
Computational Tests	
Bootstrapping Review of Descriptives Classical Hypothesis Testing Bootstrapping the Alternative	*Reading: Random Walks* *Reading: Android Malware Detector* *Statistics: Rescaling Data*
Nonparametric Testing Bootstrapped Hypothesis Testing Empirical Distributions and Power	*Simulation: Null and Alternative* *Statistics: Type I, Type II Errors* *Data: Website Performance*
Permutation Tests Reshaping Data Permutation of Data Samples Wilcoxon Test: Permutation vs. Sum of Ranks	*Tutorial: Swirl to Learn R* *Data: Verizon Customer Service*
Multigroup Tests Normality and Quantiles – the QQ Plot ANOVA: Parametric Test for Multiple Groups Kruskal Wallis: Nonparametric Test of Independent Groups	*Coding: How to Choose R Packages* *Statistics: Familywise Errors* *Data: Media Experiment*

(continued on next page)

Inferring Relationships in Data

Data Similarity
Data as Vectors
Similarity: Cosine, Correlation
Item-Item Collaborative Filtering

Reading: Collaborative filtering at Amazon
Statistics: p-hacking and Frequentist Mistakes
Statistics: Dot Products
Reading: Collaborative Filtering at Amazon

Dual Perspectives of Linear Regression
Review of Linear Regression
Geometric Representation of Regression
Linear Algebraic Representation of Regression

Simulation: Interactive Regression
Data: Cars Dataset
Videos: Essence of Linear Algebra
Reading: Amazon Retrospective on Recommender Systems

Applied Regression
The Hat Matrix
Diagnosing and Managing Non-Linearity
Diagnosing and Managing Multi-Collinearity

Statistics: Stepwise-VIF
Videos: Essence of Linear Algebra

Moderation and Mediation
The Contingency Perspective as Moderation
Partial Orthogonalization
Bootstrapped Test of Indirect Effects

Videos: Essence of Linear Algebra

Data Dimensions and Latent Variables

Composites and Components
Multi-item Constructs
Principal Components
Transforming Dimensions
Reducing Dimensions

Data: Online Security Survey
Data: Decathlon Athletics
Simulation: Interactive PCA

Principal Components Analysis
Composite Variables
Composites vs. Factors
Component Rotation as Perspective

Coding: Anonymous Functions
Coding: Pipes & Forward Moving Code
Statistics: Parallel Analysis

Structural Equation Modeling
Structural Models
Composite Structural Models
Common Factor Structural Models

Coding: SEMinR package by class alumni
Coding: Domain-Specific Languages
Coding: Functional Currying
Coding: Open-Source Communities

Predictions

Predictions
Out-of-sample Predictions
Split-sample Testing
k-Fold Cross Validation

Statistics: Polynomial Regression
Statistics: leave-one-out cross validation
Machine Learning: Decision Trees

Ensemble Predictions
Stable vs. Unstable Algorithms
Bagging Algorithms
Boosting Algorithms

Data: Insurance Dataset
Coding: Updating Estimated Models
Coding: expand.grid vs. nested for-loops

Validation and Conclusions
Hyperparameter Tuning
Validation Sets
What's Next?

Coding: High Performing Data.Table package
Coding: RStudio Server
Coding: Shiny Web Applications
Coding: Matrix package for sparse matrices