

# 2018 CATALOG ADDENDUM

## FACULTY

Roberto Reif, Senior Data Scientist, comes to Metis from Sensoria Inc, where he established and led the Sensoria Artificial Intelligence team, which pioneered a personalized shoe recommendation algorithm using machine learning techniques based on an individual's running form. He also worked at Microsoft, where he served as Program Manager and implemented an algorithm for business driver prioritization and portfolio management optimization. After receiving his Ph.D. in Biomedical Engineering from Boston University, he served as Postdoctoral Fellow at the Biophotonics and Imaging Laboratory at the University of Washington. There he led a team which developed image processing and classification algorithms for analyzing data from several diseases. Roberto is also Founder and Director of Analysis Consulting LLC, a consulting firm that focuses on data analysis, signal/image processing, machine learning and artificial intelligence, business intelligence and visualization.

Alice Zhao, Senior Data Scientist, comes to Metis from Cars.com, where she started as the company's first data scientist, supporting multiple functions from Marketing to Technology. During that time, she also co-founded a data science education startup, Best Fit Analytics Workshop, teaching weekend courses to professionals at 1871 in Chicago. Prior to becoming a data scientist, she worked at Redfin as an analyst and at Accenture as a consultant. She has her M.S. in Analytics and B.S. in Electrical Engineering, both from Northwestern University. She blogs about analytics and pop culture on A Dash of Data. Her blog post, "How Text Messages Change From Dating to Marriage" made it onto the front page of Reddit, gaining over half a million views in the first week. She is passionate about teaching and mentoring, and loves using data to tell fun and compelling stories.

Trent Hauck is currently a Senior Data Scientist at Zymergen, where he builds data products that help biologists and other scientists improve their decision outcomes. Prior to his that role he spent 18 months consulting in the insurance and e-commerce industries, and prior to that worked at Zulily on the relevancy team building recommendation tools for the e-commerce site and conducting operational analysis. He is also the author of two books through Packt Publishing: Instant Data Intensive Apps with pandas How-to, and Scikit-Learn Cookbook. Ask him to get coffee, he can't say no.

Jonathan Balaban is a consultant, data scientist, and entrepreneur with ten years of private, public, and philanthropic experience. As a data scientist, he has worked at McKinsey and Booz Allen Hamilton. He has also taught data science at General Assembly. Jonathan has led teams to design bespoke data science solutions that have driven revolutionary changes in client operations.

Chad Scherrer comes to Metis from a diverse technical background. After earning his mathematics PhD from Indiana University, Chad joined Pacific Northwest National Laboratory to work on statistical and computational challenges ranging from homeland security to high-performance computing and machine learning research. Following several of his publications at top-tier ML conferences, he turned to probabilistic programming, then still in its infancy. He has used these systems for consulting projects for industrial clients, and has led development and publication of several new ones along the way. In his spare time, Chad enjoys a wide range of music, and practices martial arts, where he has black belts in several styles.

Brendan Herger comes to Metis most recently from Capital One where he was a Machine Learning Engineer and where he had the opportunity to build authorization fraud, insider threat, and legal discovery automation platforms. In each of these cases he led a team of data scientists and data engineers to enable and elevate his client's business workflow (and capture some amazing data). He's also held Data Scientist roles at Lawfty, RevUp Software, and Perkins+Will. He received his Master's in Analytics and Bachelor's in Physics from the University of San Francisco.

## **PROFESSIONAL DEVELOPMENT PART-TIME COURSES**

### **ADMISSION INFORMATION**

- 1) Students must be at least 18 years of age or older.
- 2) There are technical skill prerequisites for each course. Every student must bring a laptop to class every day. Metis suggests using an Apple OS X operating system, with at least 4GB RAM, at least 2GHz, and at least 100 GB HD, though some other computers can be accommodated with advance notice. Students may be required to install specific software on their laptops for the practical training.

The courses are 36 hours in length, held twice a week from 6:30-9:30pm for six weeks, unless otherwise specified.

### **Machine Learning: Algorithms and Applications**

Prerequisite: Students should have a firm knowledge of the Python programming environment and a basic understanding of vector and matrix algebra, as well as the notion of a mathematical function. Basic calculus and linear algebra is helpful but not required.

The Machine Learning 36-hour non-occupational professional development course provides an overview of the core principles of machine learning with an intense focus on implementing popular machine learning algorithms to solve real problems using real data. In this course students learn by doing and, with the help of a seasoned machine learning professional, students will implement many of today's most powerful machine learning algorithms.

## *Course Objectives*

After completing this course a student is expected to:

- Understand the basic principles of machine learning from both an intuitive and practical level.
- Gain an intuitive understanding of common feature design principles for image, text, and speech data.
- Learn how to use popular machine learning / deep learning software packages in python, as well as be able to implement several popular machine learning algorithms from scratch.
- Gain extensive experience applying machine learning algorithms to real datasets.

## *Course Outline*

### **Week One**

Introduction to the course – an overview of machine learning, this course, and jumping into our first projects

- What kinds of things can you build with machine learning tools?
- How does machine learning work? The 5-minute elevator pitch edition.
- Predictive models – our basic building blocks
- Feature design and learning – what makes things distinct?
- Numerical optimization – the workhorse of machine learning
- Jumping in - getting our hands dirty with python

### **Week Two**

Learning to predict the future – the regression task with applications in forecasting, finance, and basic science

- Linear regression – the first basic building block of machine learning
- Using calculus to build useful algorithms – calculus defined optimality and solving the least squares problem
- Knowledge-driven feature design for regression
- Nonlinear regression and regularization
- Time series extensions

### **Week Three**

Teaching a computer to distinguish between different things - the classification task with applications to object detection, speech recognition, finance, and analytics

- The perceptron/logistic regression/Support Vector Machines
- A brief primer on (stochastic) gradient descent
- Multiclass classification
- Knowledge driven feature design for classification– examples from computer vision (object/face detection and recognition), text mining, and speech recognition

### **Week Four**

Learning and selecting proper features – including a review of deep learning and common python libraries for image and natural language processing applications

- Function approximation and bases of features
- Feed-forward neural network bases, deep learning, and kernels
- Cross-validation for feature learning and selection
- Using deep learning libraries in python

### **Weeks Five & Six**

Making sense of big data - applications in text mining, consumer / product segmentation, recommender systems, image processing, and brain science

- Tools for enormous datasets: K-means clustering and extensions
- Tools for high dimensional data: Principal Component Analysis and random projections
- Getting to the heart of the matter - matrix factorization models and their many applications
- Fixed and learned factorizations including the sparse coding model for redundancy reduction
- A peak under the hood – a closer look at the fundamental optimization algorithms of machine learning

## **SCHEDULE & FEES**

**Schedule:** TBA

**Course Tuition:** \$2,350

## **Data Visualization with D3.js**

Prerequisite: This course is open to beginners, but students should have experience writing HTML, CSS and basic JavaScript. For HTML/CSS, you should know how to work with the DOM and be familiar with CSS selectors. For JavaScript, you should be familiar with variables, data types, arrays, loops and conditional statements, and you should have worked with functions and objects. For Git and GitHub, you should be familiar with forking, cloning, pull requests, and branches. Finally, you should have a general idea of working with and manipulating structured data.

The Data visualization with D3 JS is a 36-hour non-occupational professional development course for anyone who wants to be proficient in the use of D3 and seeks expertise visualizing quantitative information. Learning to make charts form by form – scatter plots, then bar charts, then line charts, and so on – is not the most creative way to learn about data visualization. But because the course is so technical, this structure will help provide a foundation we can build on each week. Each class will roughly be split into two. Half will be discussions and a hands-on, computers off activity about that week's subject, and half will be a deeply technical guided lab making things in D3.

### *Course Objectives:*

Upon completing the Data Visualization with D3.js course, students will have:

- A working conceptual understanding of the field of data visualization, particularly as it relates to the internet and mobile devices.
- Deep knowledge of the forms and techniques of data visualization and effective display of quantitative information; specifically, bar charts, scatterplots, area charts, line charts, choropleth and bubble maps, small multiples, annotation principles; and the strengths and weaknesses of each.
- Proficiency in using D3 to make static and interactive charts and documents, and in using JavaScript to process and manipulate data

### *Course Outline:*

#### **Week 1 | Getting Started and Learning About New Problems**

We'll do boring things like configure our computers, make our first charts, understand why data joins are helpful and get a sense of all the things we need to learn.

- Introduction, configuring machines, intro to data visualization
- Making our first chart...scatterplots
- Charting and intent
- Bar charts
- The fuddly bits: axes, formatting, etc.

#### **Week 2 | Enough to be Dangerous: Mastering Basic Forms**

If you want to be good, you really only need to be good at making a few kinds of forms: bar charts, line charts, scatterplots and maybe a histogram. We'll make demos of all of these and understand when to use which.

- Line + area charts
- Histograms
- Tables to line + area charts

#### **Week 3 | Data Sketching and Traversing Data Structures**

Since your computer is drawing the charts instead of you, making 100 charts is as easy as making 1. We'll explore the power of exploratory sketching and the data manipulation you'll need to be able to master to do so.

- Making things move
- Sketching in the browser and making applications that scale

## Week 4 | Maps

Mapping with D3 has exploded in the last few years. We might not explore Great Circle Arc Intersects, but we'll learn how to make bubble maps, choropleth maps and create topojson files from scratch.

## Week 5 | Making Dynamic Content

Most things don't need to be interactive, but when they do, you'll be ready. We'll use D3 to make dynamic charts and applications that let us answer questions and solve problems that couldn't have happened in a printed format.

- Data Visualization on Mobile Devices
- D3 and Node

## Week 6 | Editing and Publishing an Idea

Here, we focus on honing ideas and making publication-grade data visualizations. We'll work on small touches, like custom annotations and styles, managing your data visualizations on mobile devices, incorporating feedback and pitching work for publication. We'll also do a "show and tell" of projects we've been working on throughout the 6 weeks.

- Pushing the envelope
- Project show-and-tell

## SCHEDULE & FEES

**Schedule:** TBA

**Course Tuition:** \$2,350

## Introduction to Data Science

Prerequisite: Students should have some experience with Python and have a passing familiarity with basic statistical and linear algebraic concepts (mean, median, mode, standard deviation, correlation, the difference between a vector and a matrix). In Python, it will be helpful to know basic data structures such as lists, tuples, and dictionaries, and what distinguishes them (that is, when they should be used).

Students should skip the prework if they can:

- Write a program in Python that finds the most frequently occurring word in a given sentence.
- Explain the difference between correlation and covariance, and why the difference between the two terms matters.
- Multiply two small matrices together (e.g. 3X2 and 2X4 matrices).

Data science has become the central approach to tackling data-heavy problems in both the business and academic worlds today. The intent of this course is to expose students to the data scientific approach to thinking about and solving problems, and to help students learn to think about data-heavy problems that they'll encounter in the future. Students learn how data science is done in the wild, including data acquisition, cleaning, and aggregation, exploratory data analysis and visualization, feature engineering, and model creation and validation. Students will use the Python scientific stack to work through examples that illustrate all of these concepts, with real-life use cases. Concurrently, students will learn some of the statistical and mathematical foundations that power the data scientific approach to problem solving.

#### *Course Objectives:*

Upon completing the Introduction to Data Science course, students will have:

- An understanding of problems data science can help to solve, and an ability to attack those problems from a statistical perspective
- An understanding of when to use supervised and unsupervised statistical learning methods on labeled and unlabeled data-rich problems
- An ability to create data analytical pipelines and applications in Python
- A familiarity with the Python data science ecosystem and the various tools one can use to continue to develop as a data scientist

#### *Course Outline:*

The class is comprised of a roughly even mix of lectures/instruction and hands-on programming/lab work. The week-by-week breakdown is as follows:

#### **Week 1 | CS/Statistics/Linear Algebra Short Course**

We start with the basics. In the CS portion, we briefly cover basic data structures/types, program control flow, and syntax in Python. In the statistics portion, we go over basic probability and probability distributions, along with general properties of some common distributions. As for linear algebra, we cover matrices, vectors, and some of their properties and how to use them in Python.

## **Week 2 | Exploratory Data Analysis and Visualization**

We spend a considerable amount of time using the Pandas Python package to attack a dataset we've never seen before and to uncover some useful information from it. At this point, students decide on a course project that would benefit from a data scientific approach. The project must involve public (freely-accessible/usable) data and must answer an interesting question, or collection of questions, about that data. Several resources of free data will be provided.

## **Week 3 | Data Modeling: Supervised/Unsupervised Learning and Model Evaluation**

We learn about the two basic kinds of statistical models, which have classically been used for prediction (supervised learning): Linear Regression and Logistic Regression. We also look at one of the ways from which we can glean information from unlabeled data: clustering using K-Means.

## **Week 4 | Data Modeling: Feature Selection, Engineering, and Data Pipelines**

We switch gears from talking about algorithms to talk about features: what they are, how to engineer them, and what can be done (PCA/ICA, regularization) to create and use them given the data at hand. We also cover how to construct complete data pipelines, going from data ingestion and preprocessing to model construction and evaluation.

## **Week 5 | Data Modeling: Advanced Supervised/Unsupervised Learning**

We delve into more advanced supervised learning approaches, during which we get a feel for linear support vector machines, decision trees, and random forest models for regression and classification. We also explore an additional unsupervised learning approach: DBSCAN.

## **Week 6 | Data Modeling: Advanced Model Evaluation and Data Pipelines; Presentations**

We explore more sophisticated model evaluation approaches (cross-validation, bootstrapping) with the goal of understanding how we can make our models as generalizable as possible. Students complete their data science projects and share their learnings and discoveries.

## **SCHEDULE & FEES**

**Schedule:** TBA

**Course Tuition:** \$2,350

# Big Data Processing with Apache Spark

## Prerequisites

### **Programming:**

This course is open to both beginner and experienced programmers, and we make every attempt to make it self-sufficient. It is encouraged for students to have familiarity with programming in languages such as Java, C++, Python -- experience in Java or Scala is a plus. Students are expected to know the basic concepts of compiling/running developed programs and object-oriented programming. They are also expected to be familiar with basic Unix/Linux command line utilities and be able to install open-source software on their laptops.

### **Source code management:**

We will be using GitHub throughout the course for sharing and maintaining code. So, please familiarize yourselves with GitHub activities such as code committing, forking and cloning repositories, creating pull requests and branches.

### **Data processing:**

Beginner-level experience with SQL and related database query processing is recommended.

Organizations from diverse disciplines are inundated with data. The ability to process these large data sets effectively and efficiently is critical for making data-driven business decisions and for building data-intensive services (e.g., recommendations, predictions, diagnosis, etc.). In this course, we will learn to use Apache Spark framework for the purposes of such big data management and analysis. It focuses on fundamental concepts, overall architecture, different components, APIs and language interfaces of Spark. The emphasis will be on learning through practical examples and use cases in the real-world applications. While Spark allows interfacing via multiple different languages, we will primarily concentrate on using the Scala API. In preparation for deep-dive into the world of Spark, we will first learn the basics of Scala language and the relevant aspects of functional programming upon which Spark is designed and developed.

## *Course Objectives:*

Upon completion of this course, students will:

- Have an in-depth understanding of fundamental concepts, design principles, and system architecture of Apache Spark.
- Have hands-on experience in tackling problems in processing and analyzing big data sets.
- Be well-prepared for Spark certification exams from leading companies such as Databricks and Cloudera.

## *Course Outline:*

## Week 1

We will get introduced to big data processing frameworks -- history and use cases of Hadoop and what brought us to Spark. We will also learn the fundamentals of Scala programming.

## Week 2

We will learn to do object-oriented programming and functional programming in Scala. We will cover some important aspects of functional programming (such as higher-order functions, closures, collections, currying, etc.) that will be extremely useful in writing good code with Spark.

## Week 3

We will start by getting introduced to the basics of Spark -- including, the data model of RDDs; supported operations on RDDs; and working with data sets with key-value pairs. We will then slowly deep-dive into the concepts of dependencies, lineage, and stages to understand the underlying architecture providing scalability and fault tolerance.

## Week 4

We will continue the exploration of Spark with more advanced concepts of partitioning and broadcasts, and cover topics relevant to debugging and performance tuning.

## Week 5

We will move up the level of abstraction by discussing Spark libraries. In this week, we will focus on SQL-style processing with Spark DataFrames and DataSets. We will also learn to create machine learning and predictive models using MLlib by considering the use case of building a recommendation engine.

## Week 6

In this final week, we will learn to process streaming data (e.g., tweets from Twitter) and graph-structured data using Spark Streaming and GraphX/Bagel, respectively. We will conclude by discussing the future directions and potential next steps.

## SCHEDULE & FEES

**Schedule:** TBA

**Course Tuition:** \$2,350

## Deep Learning with Tensor Flow

Pre-requisite: Deep learning is not an entry-level subject. In order to get the most out of the course, students should be familiar with the following:

- Basic statistics
- Basic linear algebra (matrix multiplication, transposing matrices)
- Basic calculus (derivatives, summations)
- Programming: Python preferred, but those comfortable with another language should be able to learn the material

There will be a pre-course workshop to refresh students on the requisite linear algebra and calculus techniques. Students with familiarity with NumPy will find it easier to pick up the material, but necessary components will be taught along with the rest of the course. Course materials will be hosted on GitHub, so knowledge of the bash console, Git, and the GitHub platform are beneficial, but not required.

Description: This course teaches the core concepts of deep learning using TensorFlow, Google's open-source computation graph library.

Deep learning has become standard in the tech industry, achieving state-of-the-art results in computer vision, natural language processing, and artificial intelligence. TensorFlow provides the flexibility needed to implement and research cutting edge architectures while allowing users to focus on the structure of their models as opposed to mathematical minutiae. Students will learn modern techniques with hands-on model building, data collection/transformation, and deployment.

### *Course Objectives*

After completing this course a student is expected to:

- Understand the techniques, concepts, terminology, and mathematics of deep machine learning for computer vision and natural language processing.
- Be proficient in TensorFlow to implement, train, visualize, export, and deploy deep models from scratch, as well as utilize pre-trained sources.
- Have skills for cleaning, normalizing, and generating data to make the most out of available datasets.
- Have foundational knowledge of recent deep learning literature and experience implementing concepts introduced in those papers.

### *Course Outline*

## Unit 0 – Linear Algebra and Calculus Workshop (Optional)

Before class officially begins, we'll have a brief, optional, on-site workshop for those in the class who need to brush up on their linear algebra and calculus.

We won't need all of the theoretical knowledge from those subjects, but certain practical techniques are necessary:

- Single variable derivatives
- Summations (and derivatives of summations)
- Matrix multiplication
- Matrix manipulation (transpose, inverse)

## Unit 1 - TensorFlow and Machine Learning Fundamentals

This unit focuses on getting students acquainted with the TensorFlow framework as well as developing an intuition for techniques that are used throughout machine learning.

- Installing TensorFlow
- TensorFlow core API
- Introduction to TensorBoard
- Linear, logistic regression
- Gradient descent

## Unit 2 – Neural Network Basics

Once we have familiarity with basic machine learning concepts, we will introduce the feedforward neural network, which lays the foundation for the rest of the models in the class.

- Neural network structure
- Neurons, activation functions (sigmoid, tanh, ReLU)
- Forward propagation and backpropagation
- Dropout, softmax, cross-entropy

## Unit 3 – Convolutional Neural Networks

Using pixels as independent inputs has several weaknesses, is there a better way? Yes! Convolutional layers are inspired from traditional image processing techniques, and are currently used in all manner of state of the art convolutional neural networks, or CNNs.

- Convolutions, pooling
- AlexNet, ResNet, Inception, and beyond
- Transfer learning (for fun and profit!)
- Object detection

## Unit 4 – Recurrent Neural Networks

Sometimes your inputs come in various lengths, such as text (each sentence could be of different length), or be naturally sequential (such as time series). Recurrent neural networks, or RNNs, are designed to handle data that comes in different lengths, and have become incredibly popular for various natural language processing tasks.

- RNN basics
- Long short term memory (LSTM)
- Sentence classification
- Text generation

## Unit 5 – Deploying Tensorflow Models

TensorFlow includes functionality to help developers easily put their models up in a deployment setting, which is one of the more attractive features of the software. With this, students should be able to take their trained models and get them ready to be used out in the wild.

- Rudimentary TensorFlow deployment in pure Python
- Setting up a basic TensorFlow Serving server (with some C++)
- Accessing your model server remotely

## SCHEDULE & FEES

**Schedule:** TBA

**Course Tuition:** \$2,350

## Statistical Foundations for Data Science and Machine Learning

Pre-requisites: This course is open to beginners, but students should have some experience with coding (Python or R preferable but not required) and have a basic understand of calculus, linear algebra and probability. A brief review will be provided but prior experience would be very helpful.

Students may opt to skip the pre-work if they:

1. Have taken an introductory course to statistics or probability in college
2. Are familiar with Linear Algebra (either coursework or work experience)
3. Are able to do a hypothesis test to determine:

- If a coin is fair given 100 flips
- Calculate a confidence interval for the mean height given 100 observations
- Explain how to test if events are independent
- Use Bayes Rules to see what the probability of an event is given another event
- Fit a linear model in R.

Otherwise, students should familiarize themselves with Chapters 1-6 of CK-12 Foundation’s Basic Probability and Statistics – A Short Course. Each chapter should take between 1-2 hours.

**Description:** This course will serve as introduction to basic statistical principles that are often used by data scientists and applied statisticians. Many of the concepts will be reinforced by using the statistical programming language R, one of the two most popular languages for Data Science.

The intent of this course is to expose students to common statistical issues and teach them how to avoid statistical fallacies. We begin with a high-level overview of probability and common statistical estimates and then proceed to move advanced topics like multiple hypothesis testing, independence, sample size and power calculations as well as bootstrapping.

By the end of the course, students will have a fundamental understanding of many of the statistical principles that underlie machine learning and data science.

### *Course Objectives*

After completing this course a student is expected to:

- Understand basic statistical hypothesis testing and confidence intervals.
- Have the ability to model data using well known statistical distributions as well as handle data that is both continuous and categorical.
- Have the ability to perform linear regression and adjust for multiple hypothesis.
- Understand how to calculate the number of samples needed to achieve required sensitivity and specificity.
- Understand bootstrapping and Monte Carlo simulation.

### *Course Outline*

#### **Class 1 - Basic Probability, Expected Value, Variance, Point Estimates, Introduction to R**

We will start the course with a review of basic probability and how to compute basic properties of a random variable such as the expected value and variance. We will also clearly define what is a point estimate and how that varies from a statistical estimate. How to compute these properties will be examined via R.

## **Class 2 - Further Probability, Central Limit Theorem, Law of Large Numbers, Hypothesis Testing**

We will use probability to calculate probabilities about binomial and normal distribution. We will explore the central limit theorem and the law of large numbers to understand how to calculate probabilities of events for averages. This will lead us into basic hypothesis testing and an exploration of how to interpret testing results.

## **Class 3 - P-Values, Multiple Comparisons, Bonferroni Adjustment**

We will explore the formal definition of a confidence interval as well as its interpretation. We will also discuss the issue of multiple comparisons and provide an example of a false positive. We will then explain the use of a Bonferroni Adjustment as well as the False Discovery Rate.

## **Class 4 - Introduction to Regression, Prediction, Hypothesis Testing for Regression**

Given a set of continuous outcomes and predictive variables, we will create a linear regression model using R. We will then explain how to use that model to generate predictions for new observations as well as test if any of the coefficients have statistically significant parameters.

## **Class 5 - Model Selection for Regression, Backwards/Forwards, $R^2$ and other selection criteria**

We'll look at how to select models when using a variety selection criteria such as  $R^2$  and adjusted  $R^2$ . We'll also look at backwards, forwards and best subset regression. Finally, we'll briefly cover logistic regression and how/why it's used.

## **Class 6 - Categorical Data, 2x2 tables, Simpson's Paradox**

We will introduce the odds ratio for a 2x2 table as well as a statistical test for independence. We will also introduce 2x2xk table with an example of Simpson's paradox.

## **Class 7 - Independence, MxN tables and trend, Fisher's Permutation Test**

We provide further examples of independence along with the introduction of larger tables. Trends and advanced categorical analysis will be covered. We will then go into Fisher's exact permutation test to explore what hypothesis testing can be done on small sample sets.

## **Class 8 - Correlation & Causation**

We will provide several examples of how to calculate correlation for both continuous and categorical variables. We will also provide how to calculate confidence intervals to determine if the correlation is significant. Finally we will explore the correlation implies causation fallacy and provide some counter examples.

## **Class 9 - A/B testing, Hypothesis Testing proportions, More General Hypothesis**

Here we provide several examples of hypothesis testing as it relates to Data Science and web design. We'll also cover hypothesis testing & confidence intervals for proportions and variance.

### **Class 10 - Sample Size & Power Calculation / Method of Moments Estimation**

We will work through several examples on how to calculate the required sample size given a specific level of false positives and a pre-specified power level. We will go into more detail why it's only possible to reject or fail to reject a null hypothesis (and not to accept a null hypothesis). Next, we will switch gears and cover Method of Moments, compare it to MLE and take a look at a few examples.

### **Class 11 - Bootstrapping, the Information Matrix & Variance Bound**

We discuss some options one can use if they are dealing with small amounts of data, specifically the bootstrap method. We'll then switch gears and touch upon the information matrix and how to calculate a theoretical lower bound on the variance of any statistic of interest.

### **Class 12 - Expectation-Maximization Algorithm, Bias/Variance Trade Off**

We'll explore the details of the expectation maximization algorithm and how it's used in the presence of latent variables for estimation. We'll work through an analytical example as well as how to use R to do it. We will also cover the Bias/Variance tradeoff when modeling and the pitfalls of overfitting.

## **SCHEDULE & FEES**

**Schedule:** TBA

**Course Tuition:** \$2,350