University of Illinois – Deloitte Foundation Center for Business Analytics

Instructor's Guide for the First Course in Foundations of Data Analytics

Course Description

This course will build a practical foundation for data analytics by teaching students basic tools and techniques that can scale to large computational systems and massive data sets. Specifically, students will first learn how to use the Python programming language, with a focus on specific aspects of the language and associated Python modules that are relevant for data analytics. The Python programming language will be introduced, for which students will use Jupyter Notebooks. This introduction will include the NumPy, SciPy, Matplotlib, Pandas, and Seaborn Python modules. These capabilities will be demonstrated through simple data analytic tasks such as obtaining data, cleaning data, visualizing data, and basic data analysis. In addition, students will learn how to work about the Unix file system, which is used by most big data tools or technologies.

Course Organization

This course is organized into eight modules. Each module consists of multiple lessons, which each contain an associated video explaining the lesson content, one or more external readings, and included course Jupyter notebooks. Each module also includes a quiz (or assessment) that tests basic mastery of the lesson contents and a programming assignment that tests synthesis of the lesson contents.

Typically, a module is covered in one week, although the pace can be adjusted based on student needs. In addition, content can be selectively used from lessons and modules to provide supplemental learning content for an existing course. The module quiz is traditionally given through the course LMS in a timed fashion (e.g., thirty minutes). The assignments are completed on the course JupyterHub server, collected at the assignment deadline, and processed by an autograder.

Jupyter notebooks: *A document that contains both computer code (e.g., Python), the output of the included computer code (e.g., from print statements or visualizations and plots), and rich text including paragraphs, figures, equations, and hyperlinks. Notebooks are human readable and also able to be processed automatically by the Jupyter kernel. A notebook can also be easily converted into other document formats including HTML and PDF.*
For more information, see jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html.

LMS: *Learning Management System, a software system to administer, document, track, and deliver educational content. For example, Blackboard or Moodle.*
For more information, see en.wikipedia.org/wiki/Learning_management_system.

ILLINOIS
Gies College of Business

Course Contents

Module 1: Foundations

This module introduces the course content and the course Jupyter server, where you will run your analytics scripts. First, you will read about specific examples of how analytics is being employed by accounting firms. Next, you will learn about the capabilities of the course Jupyter server and how to create, edit, and run notebooks on the course server. After this, you will learn how to write Markdown formatted documents, which is an easy way to quickly write formatted text, including descriptive text inside a course notebook. Finally, you will begin learning about Python, the programming language used in this course for data analytics.

Learning Objectives

By the end of this module, you should be able to:

- Appreciate the importance of data analytics in accountancy
- Work effectively with a Jupyter notebook on the course server
- Write basic Markdown encoded documents
- Write simple Python scripts

Module 2: Introduction to Python

This module focuses on the basic features in the Python programming language that underlie most data analytics scripts. First, you will read about why accounting students should learn to write computer programs. Second, you will learn about basic data structures commonly used in Python programs. Third, you will learn how to write functions, which can be repeatedly called, in Python, and how to use them effectively in your own programs. Finally, you will learn how to control the execution process of your Python program by using conditional statements and looping constructs. At the conclusion of this module, you will be able to write Python scripts to perform basic data analytic tasks.

Learning Objectives

By the end of this module, you should be able to:

- Articulate how accounting students can benefit from learning to write computer programs
- Write simple Python scripts
- Effectively use the basic Python data structures: list, string, dictionary, and tuple
- Write and use simple Python functions
- Use conditional statements and loops in a Python script

Markdown: *A markup language used to encode rich text by using a plain text formatting. This format can easily be inverted into other output formats including HTML and PDF. For more information, see en.wikipedia.org/wiki/Markdown.*

Module 3: Introduction to Data Analysis

This module introduces fundamental concepts in data analysis. First, you will read a report from the Association of Accountants and Financial Professionals in Business that explores Big Data in Accountancy, and read a blog article that discusses how a programming language like Python offers specific advantages over performing data analytics in Excel. Next, you will learn about the Unix file system, which is the operating system used for most big data processing (as well as Linux and Mac OSX desktops and many mobile phones). Third, you will learn how to read and write data to a file from within a Python program. Finally, you will learn about the Pandas Python module that can simplify many challenging data analysis tasks, and includes the DataFrame, which programmatically mimics many of the features of a traditional spreadsheet.

Learning Objectives

By the end of this module, you should be able to:

- Work with the Unix file system and understand the concept of directories and files
- Read and write data to files from within a Python script
- Use a Pandas Series and DataFrame to manipulate data from within a Python script

Module 4: Statistical Data Analysis

This module introduces fundamental concepts in data analysis. First, you will read about how to perform many basic tasks in Excel by using the Pandas module in Python. Second, you will learn about the NumPy module, which provides support for fast numerical operations within Python. This module will focus on using NumPy with one-dimensional data (i.e., vectors or 1-D arrays), but a later module will explore using NumPy for higher-dimensional data. Third, you will learn about descriptive statistics, which can be used to characterize a data set by using a few specific measurements. Finally, you will learn about advanced functionality within the Pandas module including masking, grouping, stacking, and pivot tables.

Learning Objectives

By the end of this module, you should be able to:

- Understand how to move from analyzing data in Excel to Pandas
- Work with one dimensional numerical data by using the NumPy module
- Compute and interpret descriptive statistics, and apply advanced features to more effectively analyze data using a Pandas DataFrame

Module 5: Introduction to Visualization

This module introduces visualization as an important tool for exploring and understanding data. First, the basic components of visualizations are introduced with an emphasis on how they can be used to convey information. Also, you will learn how to identify and avoid ways that a visualization can mislead or confuse a viewer. Next, you will learn more about conveying information to a user visually, including the use of form, color, and location. Third, you will learn how to actually create a simple visualization (basic line plot) in Python, which will introduce creating and displaying a visualization within a notebook, how to annotate a plot, and how to improve the visual aesthetics of a plot by using the Seaborn module. Finally, you will learn how to explore a one-dimensional data set by using rug plots, box plots, and histograms.

Learning Objectives

By the end of this module, you should be able to:

- Understand the basic concepts in conveying information visually
- Identify and avoid visualization techniques that can mislead viewers
- Create basic visualizations by using Python and the Matplotlib module
- Visually explore one-dimensional data by using Python and the Seaborn module

Module 6: Introduction to Probability

In this module, you will learn the basics of probability and how it relates to statistical data analysis. First, you will learn about the basic concepts of probability, including random variables, the calculation of simple probabilities, and several theoretical distributions that commonly occur in discussions of probability. Next, you will learn about conditional probability and Bayes theorem. Third, you will learn to calculate probabilities and apply Bayes theorem directly by using Python. Finally, you will learn to work with both empirical and theoretical distributions in Python and how to model an empirical data set by using a theoretical distribution.

Learning Objectives

By the end of this module, you should be able to:

- Calculate and interpret basic probability concepts
- Calculate and interpret conditional probability
- Understand and apply Bayes theorem
- Understand the most important empirical distribution measures
- Understand and know how to apply the most important theoretical distributions

Module 7: Exploring Two-Dimensional Data

This module extends what you have learned in previous modules to the visual and analytic exploration of two-dimensional data. First, you will learn how to make two-dimensional scatter plots in Python and how they can be used to graphically identify a correlation and outlier points. Second, you will learn how to work with two-dimensional data by using the NumPy module, including a discussion on analytically quantifying correlations in data. Third, you will read about statistical issues that can impact understanding multi-dimensional data, which will allow you to avoid them in the future. Finally, you will learn about ordinary linear regression and how this technique can be used to model the relationship between two variables.

Learning Objectives

By the end of this module, you should be able to:

- Display and interpret two-dimensional data by using Python
- Work with multi-dimensional arrays in NumPy
- Quantify correlations between data by using Python
- Understand statistical concerns when working with multi-dimensional data
- Compute and interpret linear models

Module 8: Introduction to Density Estimation

Often, as part of exploratory data analysis, a histogram is used to understand how data are distributed, and in fact this technique can be used to compute a probability mass function (or PMF) from a data set as was shown in an earlier module. However, the binning approach has issues, including a dependance on the number and width of the bins used to compute the histogram. One approach to overcome these issues is to fit a function to the binned data, which is known as parametric estimation. Alternatively, we can construct an approximation to the data by employing a non-parametric density estimation. The most commonly used non-parametric technique is kernel density estimation (or KDE). In this module, you will learn about density estimation and specifically how to employ KDE. One often overlooked aspect of density estimation is the model representation that is generated for the data, which can be used to create new data. This concept is demonstrated by applying density estimation to images of handwritten digits, and sampling from the resulting model.

Learning Objectives

By the end of this module, you should be able to:

- Understand both parametric and non-parametric density estimation
- Understand the basic concepts behind kernel density estimation
- Use density estimation to approximate or smooth discrete data
- Construct a kernel density estimate by using the Python scikit-learn module
- Sample from a density estimate model to generate new data.