

Syllabus, Spring 2019, for OIDD 311:

Business Computer Languages:

Scripting for Business Analytics

10:30–12:00 T & R, JMHH 380 [sic]

Canvas [sic]: [https :](https://upenn.instructure.com/courses/1344758)

[//upenn.instructure.com/courses/1344758](https://upenn.instructure.com/courses/1344758)

Professor Steven O. Kimbrough, Instructor

Office hours: 565 JMHH, 9:00–10:30 Tuesdays and Thursdays, and by appointment

TAs: Michelle Lok

November 29, 2019, revision 0

Provisional syllabus (key dates set, but class topics may change slightly)

1 Class Description

OPIM 311, “Scripting for Business Analytics,” may be framed as follows. *Business analytics* refers to modeling and analysis undertaken for purposes of management and supporting decision making. This course is about business analytics and exploratory computing in support of business analytics, it is about using MATLAB[®] and Python for these purposes, and the treatment is introductory.

The business analytics concept has received wide uptake and substantial attention in the business press. In fact, the *Zeitgeist* embraces analytics. Business school gurus tout it.¹ Consultants and practitioners proclaim it, repeating “big data” at every turn.² Geeks stun the establishment with impressively accurate predictions.³ Jobs are plentiful even in a down economy and professional organizations start offering certification programs, sponsor conferences on business analytics, and publish specialized news magazines on the subject.⁴ Although hype abounds, there are real achievements at hand.

¹See, for example, “Creating Value with Business Analytics” (Kiron and Shockley, 2011), *Competing on Analytics: The New Science of Winning* (Davenport and Harris, 2007) and *Super Crunchers: Why Thinking-by-Numbers Is the New Way to Be Smart* (Ayres, 2007).

²See, for example, “Big data: The next frontier for competition” (McKinsey & Company, 2012) and *Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics* (Franks, 2012) and *The New York Times* announces “The Age of Big Data” (Lohr, 2012).

³See, for example, *The Signal and the Noise* (Silver, 2012) and various poll aggregators who performed so impressively in the 2012 U.S. presidential election.

⁴See INFORMS, <http://www.informs.org>; <http://informs.informs.org/>

The varieties of techniques and methods are numerous and growing, including simple equational models, constrained optimization models, probabilistic models, visualization, data analysis, various kinds of heuristics (and families of heuristics, called metaheuristics), machine learning (supervised and unsupervised), data mining, text mining, agent-based models, simulation, multiobjective optimization and decision making, and much more. Elementary modeling of this sort can be undertaken in Excel and other spreadsheet programs, but “industrial strength” applications typically use more sophisticated tools, based on scripting languages.

Scripting languages are programming languages that are designed to be learned easily and to be used for special purposes, rather than for large-scale application programming. This course focuses on the special purposes associated with business analytics and teaches MATLAB and Python in this context. MATLAB and Python are widely used in practice (both in management and in engineering), as are the business analytic methods covered in the course.

Prior programming experience is useful, but neither required nor presumed for this course.

2 Texts

- Required text: *Automate the Boring Stuff with Python: Practical Programming for Total Beginners* Sweigart (2015).
- Recommended text: *MATLAB[®] Primer*, Davis (2011).
- Various MATLAB documentation, available at <http://www.mathworks.com/help/matlab/> and as PDFs on Canvas.
- *An Introduction to Doing Analytics with MATLAB[®]: Exploratory Computation and Business Analytics* by Steven Orla Kimbrough (Kimbrough, 2016a). Referred to as the *DAMbook*, it is handed out as a PDF, *DAMbook-311.pdf*.
- *Python Real Fast* by Steven O. Kimbrough (Kimbrough, 2016b). Referred to as the *PRFbook*, it is handed out as a PDF as the file *PRFbook.pdf*.

In addition, various other readings will be assigned, especially various from various MATLAB manuals. These will generally be handed out or made available online.

MATLAB is a commercial product (installed on the Wharton lab machines). The main URL for MATLAB is <http://www.mathworks.com> (and see the Appendix B.2 of the *DAMbook* for related links). There is an open-source product that is in many ways similar to MATLAB: GNU Octave, <http://www.gnu.org/software/octave/>. There are student/educational versions for sale at a hefty discount for MATLAB. Alternatively, you can install Octave on your own computers for free.

Python is an open-source, “free” product. <http://www.python.org/> and <http://docs.python.org/tutorial/>. The following passage, from the Python Web site, is accurate.

Build-Your-Career/Analytics-Certification;
analytics2013//<http://www.analytics-magazine.org/>.

<http://meetings2.informs.org/>

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. <http://docs.python.org/tutorial/>

There are several different builds of Python available. You can download and install versions from the official Python Web site: <http://www.python.org/>. In addition to version 3.1.2 from there, we'll be working with a "Python 3" build from Continuum Analytics <http://continuum.io/>, and their Anaconda Server and IDE, Spyder, http://docs.continuum.io/anaconda/ide_integration.html and <http://docs.continuum.io/anaconda-launcher/index.html>. I encourage you to download this version (and everything that comes with it) for your own computers. It is free for educational purposes.

In the spirit of open source, free software, there is much Python documentation freely available. We'll use some of it. A recommended, but not required, text is: *Programming in Python 3: A Complete Introduction to the Python Language*, (Summerfield, 2009). Also, *Python Essential Reference* by David M. Beazley (Beazley, 2009) is very useful for the day-to-day Python programmer.

Three other useful books: Bird et al. (2009) is an excellent introduction to ...natural language processing with Python, and generally excellent all around. Beyond the scope of this course, but recommended anyway. Kreibich (2010) for a general introduction to SQLite. Friedl (2006) for an authoritative discussion of regular expressions.

3 Grades

Grading will be based on several components, roughly as follows.

40% Class participation and in-class exercises.

20% Python case assignment.

9% MATLAB case assignment.

15% Small group analytics assignment (2–3 people).

16% Quizzes (2, 10% each).

Most of all, I want to see you engaged and involved in the class.

4 Class Schedule

1. Introduction and overview of the course.

Reading (before class): (Sweigart, 2015, chapter 1), "Python Basics."

2. Python basics.

Reading to prepare for this class: *Automate the Boring Stuff with Python* (Sweigart, 2015, chapter 1) and Chapter 1, "Getting Started" in the PRFbook.

3. Lists, 1.

Reading for today: Chapter 2, “Lists,” in the *PRFbook*. Read §2.0 and at least skim §2.5, “Highlights of Lists.” Today: A quick introduction (more later) on lists, tuples, `range`, `random` module, and more use of the `for` loop.

4. Python: Flow Control.

Reading (before class): (Sweigart, 2015, chapter 2), “Flow Control.”

Reading to prepare for this class: *Automate the Boring Stuff with Python* (Sweigart, 2015, Chapter 2), “Flow Control,” and skim as backup Chapter 3, “Flow Control” in the *PRFbook*. Be sure to focus on, read fully, (Sweigart, 2015, Chapter 2); I’ll be assuming you did so and will not repeat many of its points.

Recall: Our text (Sweigart, 2015) works with IDLE and the standard Python distribution. We’re working with Spyder and Anaconda.

5. Python: Functions.

Reading to prepare for this class: *Automate the Boring Stuff with Python* (Sweigart, 2015, Chapter 3), “Functions” and skim as backup Chapter 4, “Functions” in the *PRFbook*.

6. Python: Lists, 2.

Reading to prepare for this class: *Automate the Boring Stuff with Python* (Sweigart, 2015, Chapter 4), “Lists” and skim as backup review Chapter 2, “Lists” in the *PRFbook*.

7. Python: Dictionaries and Structuring Data.

Reading to prepare for this class: *Automate the Boring Stuff with Python* (Sweigart, 2015, Chapter 5), “Dictionaries and Structuring Data.”

Also, skim Chapter 7, “Beyond Lists: Other Data Structures” in the *PRFbook*.

8. Python: Manipulating Strings.

Reading to prepare for this class: *Automate the Boring Stuff with Python* (Sweigart, 2015, Chapter 6), “Manipulating Strings.”

Also recommended (skim) (Sweigart, 2015, Chapter 13).

Also, skim Chapter 5, “Strings and Their Methods” in the *PRFbook*.

9. Python: Pattern Matching with Regular Expressions, 1.

Reading (before class): (Sweigart, 2015, chapter 7), “Pattern Matching with Regular Expressions.” Also, Chapter 15 of the *PRFbook*, “Regular Expressions.”

10. Python: Pattern Matching with Regular Expressions, 2.

Reading (before class): (Sweigart, 2015, chapter 7), “Pattern Matching with Regular Expressions.” Also, Chapter 15 of the *PRFbook*, “Regular Expressions.”

11. Python: Pattern Matching with Regular Expressions, 3.

Reading (before class): (Sweigart, 2015, chapter 7), “Pattern Matching with Regular Expressions.” Also, Chapter 15 of the *PRFbook*, “Regular Expressions.”

12. Python: Quiz. **Quiz #1:
February
26, 2019**
13. Python: Web Scraping, Regular Expressions, 4.
 Reading (before class): (Sweigart, 2015, chapter 11), “Web Scraping.”
 Reading (before class): (Sweigart, 2015, chapter 7), “Pattern Matching with Regular Expressions.” Also, Chapter 15 of the *PRFbook*, “Regular Expressions.”
 * * * **Spring Break Saturday-Sunday March 2–10, 2019** * * *
14. Python: Numpy, 1. **Python case
handed out.**
 Reading: Chapter 11, “NumPy” in the *PRFbook*.
15. Python: Numpy, 2.
 Python case released. Q&A in class 18. Due date is April 15, 2019 at 11:59 p.m. Note that the case uses pandas extensively. We’ll get to pandas in class 17.
 Reading: Chapter 11, “NumPy” in the *PRFbook*.
16. Python: Files: I/O and Navigation. Cases and Projects.
 Discussion of term projects. Read: Chapter 8, “Files: I/O and Navigation” in the *PRFbook*; (Sweigart, 2015, chapters 8–9).
17. Python: Introduction to pandas.
 Read (before class): *PRFbook* Chapter 12, “Pandas.” Also, read and even study: “10 Minutes to pandas” at <http://pandas.pydata.org/pandas-docs/stable/10min.html>.
 Recommended for background and depth: <http://pandas.pydata.org/pandas-docs/stable/basics.html#basics-binop> and <http://pandas.pydata.org/pandas-docs/stable/tutorials.html>. The general resource is <http://pandas.pydata.org/pandas-docs/stable/index.html>.
18. Getting started with MATLAB.
 Python case Q&A.
 Read: “Preface” of the *DAMbook* (Kimbrough, 2016a, pages xi–xiii); Chapter 1 of the *DAMbook*, “Quick Start in MATLAB”.
 The MATLAB documentation is now online at <http://www.mathworks.com/help/matlab/index.html>
 Start reading relevant parts of *Getting Started with MATLAB* at <http://www.mathworks.com/help/matlab/getting-started-with-matlab.html>.
19. MATLAB, 2. Further Introduction.
 Read in preparation for class: *DAMbook* Chapters 1 and 3. Also, from last time: You should prepare by going through the following tutorials at the “Getting Started with MATLAB” site:
- Desktop Basics

- Matrices and Arrays
- Array Indexing
- Workspace Variables
- Calling Functions

Get “Getting Started with MATLAB” at

<http://www.mathworks.com/help/matlab/getting-started-with-matlab.html>

20. User-defined functions and flow control.

Read: Chapter 3 of the *DAMbook* and Chapter 4 of the *DAMbook*.

Recommended reading: Chapters “Simple Knapsack Models” and “Evolutionary Algorithms” in the *DAMbook*.

Recommended: In the MATLAB documentation, read relevant parts of relevant parts of “Getting Started with MATLAB” at <http://www.mathworks.com/help/matlab/getting-started-with-matlab.html> and “Programming Scripts and Functions” at <http://www.mathworks.com/help/matlab/programming-and-data-types.html>.

21. Arrays and operations on them, 1.

Read: Chapter 5 of the *DAMbook*, “Arrays and Operations on Them, 1”; (Davis, 2011, chapters 3–5). Recommended: “Matrices and Arrays” and “Array Indexing” in “Getting Started with MATLAB” at

<http://www.mathworks.com/help/matlab/getting-started-with-matlab.html>.

22. Arrays and operations on them, 2.

Read: Chapter 6 of the *DAMbook*, “Arrays and Operations on Them, 2;” (Davis, 2011, chapters 3–5). Recommended for those interested (not required for the course).

23. Beyond Numeric Arrays and I/O.

Read: Chapters 7 and 8 of the *DAMbook*, “Beyond Numeric Arrays” and “I/O: Interacting with Files.” Recommended: (Davis, 2011, chapter 8). Recommended for reference purposes: “Data Import and Analysis” at <http://www.mathworks.com/help/matlab/data-import-and-analysis.html>.

April 15, 2019 Python case due.

24. April 16, 2019: Quiz #2.

April 16, 2019 Quiz #2

25. MATLAB I/O and toolboxes.

Readings: TBA.

MATLAB assignment handed out and discussed.

26. Graphics

Readings: TBA.

27. Python: SQL, SQLite, 1.

Readings: from Chapters 20–3 of the *PRFbook*, TBA.

28. Tuesday, April 30, 2019: Last day of class. Python: SQL, SQLite, 1.

Readings: from Chapters 20–3 of the *PRFbook*, TBA.

Tuesday, April 30, 2019: MATLAB case due, 5:00 p.m.

**MATLAB
case due.**

Small group assignment hand-ins due: Tuesday, 7 May, 2019, 11:59 p.m..

**Small
group
hand-ins.**

References

Ayres, I. (2007). *Super Crunchers: Why Thinking-by-Numbers Is the New Way to Be Smart*. Bantam Books, New York, NY.

Beazley, D. M. (2009). *Python: Essential Reference*. Addison-Wesley, Upper Saddle River, NJ, fourth edition.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly, Sebastopol, CA.

Davenport, T. H. and Harris, J. G. (2007). *Competing on Analytics: The New Science of Winning*. Harvard Business School Press, Boston, MA.

Davis, T. A. (2011). *MATLAB[®] Primer*. CRC Press, Boca Raton, eighth edition.

Franks, B. (2012). *Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics*. John Wiley & Sons, Hoboken, NJ.

Friedl, J. E. (2006). *Mastering Regular Expressions*. O'Reilly & Associates, Inc., Sebastopol, California, third edition.

Kimbrough, S. O. (2016a). *An Introduction to Business Analytics with MATLAB[®]: Exploratory Computation and Business Analytics*. Draft manuscript. File: DAMbook-master.pdf. In process and changeable.

Kimbrough, S. O. (2016b). *Python Real Fast*. Draft manuscript.

Kiron, D. and Shockley, R. (2011). Creating business value with analytics. *MIT Sloan Management Review*, 53(1):56–64.

Kreibich, J. A. (2010). *Using SQLite: Small. Fast. Reliable. Choose Any Three*. O'Reilly, Sebastopol, CA.

Lohr, S. (2012). The age of big data. *The New York Times*.

McKinsey & Company (2012). Big data: The next frontier for competition. http://www.mckinsey.com/Features/Big_Data Accessed: 2012-12-25.

Silver, N. (2012). *The Signal and the Noise: Why So Many Predictions Fail—but Some Don't*. The Penguin Press, New York, NY.

Summerfield, M. (2009). *Programming in Python 3: A Complete Introduction to the Python Language*. Addison-Wesley, Upper Saddle River, NJ.

Sweigart, A. (2015). *Automate the Boring Stuff with Python: Practical Programming for Total Beginners*. No Starch Press. ISBN-10: 1593275994, ISBN-13: 978-1593275990.