ASU
ARIZONA STATE UNIVERSITY

GENERAL STUDIES COURSE PROPOSAL COVER FORM

Course information:

*Copy and paste **current** course information from Class Search/Course Catalog.*

College/School: College of Liberal Arts and Sciences | Department/School: **School of Earth and Space Exploration**

Prefix: **SES** | Number: **130** | Title: Coding for Exploration | Units: 3.0

Course description: **A series of lectures and computer labs on data processing and analysis in Earth and Space sciences using Python. Introduction to programming with scratch and python. Numerical methods for data analytics.**

Is this a cross-listed course? No | If yes, please identify course(s):

Is this a shared course? No | If so, list all academic units offering this course:

*Note- For courses that are crosslisted and/or shared, a letter of support from the chair/director of **each** department that offers the course is required for **each** designation requested. By submitting this letter of support, the chair/director agrees to ensure that all faculty teaching the course are aware of the General Studies designation(s) and will teach the course in a manner that meets the criteria for each approved designation.*

Is this a permanent-numbered course with topics? No

If yes, all topics under this permanent-numbered course must be taught in a manner that meets the criteria for the approved designation(s). It is the responsibility of the chair/director to ensure that all faculty teaching the course are aware of the General Studies designation(s) and adhere to the above guidelines.

Chair/Director Initials _____ (Required)

Requested designation: Mathematical Studies–CS

Mandatory Review: (Choose one)

*Note- a **separate** proposal is required for each designation.*

Eligibility: Permanent numbered courses **must** have completed the university's review and approval process. For the rules governing approval of omnibus courses, contact Phyllis.Lucie@asu.edu.

Submission deadlines dates are as follow:

For Fall 2018 Effective Date: October 1, 2017

For Spring 2019 Effective Date: March 10, 2018

Area(s) proposed course will serve:

A single course may be proposed for more than one core or awareness area. A course may satisfy a core area requirement and more than one awareness area requirements concurrently, but may not satisfy requirements in two core areas simultaneously, even if approved for those areas. With departmental consent, an approved General Studies course may be counted toward both the General Studies requirement and the major program of study.

Checklists for general studies designations:

Complete and attach the appropriate checklist

- Literacy and Critical Inquiry core courses (L)
- Mathematics core courses (MA)
- Computer/statistics/quantitative applications core courses (CS)
- Humanities, Arts and Design core courses (HU)
- Social-Behavioral Sciences core courses (SB)
- Natural Sciences core courses (SQ/SG)
- Cultural Diversity in the United States courses (C)
- Global Awareness courses (G)
- Historical Awareness courses (H)

A complete proposal should include:

- ☒ Signed course proposal cover form
- ☒ Criteria checklist for General Studies designation being requested
- ☒ Course catalog description
- ☒ Sample syllabus for the course
- ☒ Copy of table of contents from the textbook and list of required readings/books

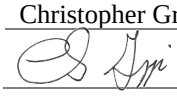
It is respectfully requested that proposals are submitted electronically with all files compiled into one PDF.

Contact information:

Name: Becca Dial | E-mail: bdial@asu.edu | Phone: 480-965-2213

Department Chair/Director approval: *(Required)*

Chair/Director name (Typed): Christopher Groppi | Date: 1/18/18

Chair/Director (Signature): 

Arizona State University Criteria Checklist for

MATHEMATICAL STUDIES [CS]

Rationale and Objectives

The **Mathematical Studies** requirement is intended to ensure that students have skill in basic mathematics, can use mathematical analysis in their chosen fields, and can understand how computers can make mathematical analysis more powerful and efficient. The **Mathematical Studies** requirement is completed by satisfying both the **Mathematics [MA]** requirement and the **Computer/Statistics/Quantitative Applications [CS]** requirement explained below.

The **Mathematics [MA]** requirement, which ensures the acquisition of essential skill in basic mathematics, requires the student to complete a course in College Mathematics, College Algebra, or Pre-calculus; or demonstrate a higher level of skill by completing a mathematics course for which a course in the above three categories is a prerequisite.

The **Computer/Statistics/Quantitative Applications [CS]** requirement, which ensures skill in real world problem solving and analysis, requires the student to complete a course that uses some combination of computers, statistics, and/or mathematics.* Computer usage is encouraged but not required in statistics and quantitative applications courses. At a minimum, such courses should include multiple demonstrations of how computers can be used to perform the analyses more efficiently.

*CS does *not* stand for computer science in this context; the "S" stands for statistics. Courses in computer science must meet the criteria stated for CS courses.

Revised April 2014

Proposer: Please complete the following section and attach appropriate documentation.

ASU--[CS] CRITERIA

A COMPUTER/STATISTICS/QUANTITATIVE APPLICATIONS [CS] COURSE MUST SATISFY ONE OF THE FOLLOWING CRITERIA: 1, 2, OR 3

YES	NO		Identify Documentation Submitted
		1. Computer applications*: courses must satisfy both **a** and **b**:	
☒	☐	**a.** Course involves the use of computer programming languages or software programs for quantitative analysis, algorithmic design, modeling, simulation, animation, or statistics.	Course syllabus
		b. Course requires students to analyze and implement procedures that are applicable to at least one of the following problem domains **(check those applicable)**:	
☐	☒	i. Spreadsheet analysis, systems analysis and design, and decision support systems.	
☐	☒	ii. Graphic/artistic design using computers.	
☐	☒	iii. Music design using computer software.	
☐	☒	iv. Modeling, making extensive use of computer simulation.	
☐	☒	v. Statistics studies stressing the use of computer software.	
☒	☐	vi. Algorithmic design and computational thinking.	Course syllabus

***The **computer applications** requirement **cannot** be satisfied by a course, the content of which is restricted primarily to word processing or report preparation skills, the study of the social impact of computers, or methodologies to select software packages for specific applications. Courses that emphasize the use of a computer software package are acceptable only if students are required to understand, at an appropriate level, the theoretical principles embodied in the operation of the software and are required to construct, test, and implement procedures that use the software to accomplish tasks in the applicable problem domains. Courses that involve the learning of a computer programming language are acceptable only if they also include a substantial introduction to applications to one of the listed problem domains.

YES	NO		Identify Documentation Submitted
		2. Statistical applications: courses must satisfy **a**, **b,** and **c**.	
☐	☐	**a.** Course has a minimum mathematical prerequisite of College Mathematics, College Algebra, or Pre-calculus**,** or a course already approved as satisfying the MA requirement.	
		b. The course must be focused principally on developing knowledge in statistical inference and include coverage of all of the following:	
☐	☐	i. Design of a statistical study.	
☐	☐	ii. Summarization and interpretation of data.	
☐	☐	iii. Methods of sampling.	
☐	☐	iv. Standard probability models.	
☐	☐	v. Statistical estimation	
☐	☐	vi. Hypothesis testing.	
☐	☐	vii. Regression or correlation analysis.	
☐	☐	**c.** The course must include multiple demonstrations of how computers can be used to perform statistical analysis more efficiently, if use of computers to carry out the analysis is not required. .	

YES	NO		Identify Documentation Submitted
		3. Quantitative applications: courses must satisfy **a**, **b, and c:**.	
☐	☐	**a.** Course has a minimum mathematical prerequisite of College Mathematics, College Algebra, or Pre-calculus**,** or a course already approved as satisfying the MA requirement.	
		b. The course must be focused principally on the use of mathematical models in quantitative analysis and decision making. Examples of such models are:	
☐	☐	i. Linear programming.	
☐	☐	ii. Goal programming.	
☐	☐	iii. Integer programming.	
☐	☐	iv. Inventory models.	
☐	☐	v. Decision theory.	
☐	☐	vi. Simulation and Monte Carlo methods.	
☐	☐	vii. Other (explanation must be attached).	
☐	☐	**c.** The course must include multiple demonstrations of how computers can be used to perform the above applications more efficiently, if use of computers is not required by students.	

Course Prefix	Number	Title	General Studies Designation
SES	130	Coding for Exploration	

Explain in detail which student activities correspond to the specific designation criteria. Please use the following organizer to explain how the criteria are being met.

Criteria (from checksheet)	How course meets spirit (contextualize specific examples in next column)	Please provide detailed evidence of how course meets criteria (i.e., where in syllabus)
1a	The course teaches computer programming using languages Scratch and python. The students use computers to design algorithms and analyze Earth and space science data.	As outlined in the course syllabus, problems from The Zelle textbook treat programming in python and algorithmic design. Exercises utilize real earth and Space science data.
1 b vi	Students learn the how to Approach programming tasks And how to design algorithms.	Exercises outlined on the syllabus teach functional and object-oriented program design, designing or selecting algorithms, and applying these to earth And space science data.

SES130: Coding for Exploration

Instructors:

Nathaniel Butler
Office: Goldwater 578
Phone: 480-965-8207
Email: Nathaniel. Butler@asu.edu

Sang-Heon (Dan) Shim
Office: ISTB4 575
Phone: 480-727-2876
Email: SHDShim@asu.edu

Office Hours: To be determined or by appointment

1. Course Description
A series of lectures and computer labs on data processing and analysis in Earth and Space sciences using Python. Introduction to programming with scratch and python. Numerical methods for data analytics.

2. Learning Outcomes
Upon completion, students will be able to:

- Learn to solve problems, in groups, using computers.
- Utilize procedural programming concepts including data types, variables, control structures, arrays, and data I/O.
- Utilize software engineering concepts including testing, incremental development, understanding requirements, and teamwork.
- Design strategies to analyze Earth and Space science data
- Write codes to process Earth and Space science data
- Present key properties of Earth and Space science data
- Interpret data for understanding Earth processes and Astronomy

3. Exercises, Test and Term Project
The course will be evaluated based on seven exercises, one final examination and six individual problem sets. The final examination consists of 5-7 coding questions based on the materials taught in this course. An instructional sheet will be provided for each problem set. All evaluation materials will be available with sufficient time for completion.

4. Grading
The course is for 3 units. Grading is based on A to E with +/- letter grading: 97.5% - 100% = A+, 92.5% - 97.4% = A, 90% - 92.4% = A-, 87.5% - 89.9% = B+, 82.5% - 87.4% = B, 82.4% - 80% = B-, 77.5% - 79.9% = C+, 70% - 77.4% = C, 60% - 69.9% = D, 0% - 59.9% = E.

Item	**Percentage**

Attendance	10.0%
Problem set #1	10%
Problem set #2	10%
Problem set #3	10%
Problem set #4	10%
Problem set #5	10%
Problem set #6	10%
Final examination	30%
Total	100%

EXTRA CREDIT
There will be **no extra credit opportunities** assigned for this course. However, individualized honors contracts and opportunities for independent study credit supervised by the instructor will be made available for projects that go beyond the scope of this course.

5. Readings and weekly itineraries

"Python Programming: An Introduction to Computer Science 3rd Edition", by John Zelle

"Python for Data Analysis, Agile Tools for Real World Data," Wes McKinney

Week 1 Course overview, why python?; Zelle Ch1
Week 2 Computers and Simple Programs; Zelle Ch 2
Week 3 Data Types, numbers; Zelle Ch 3
Week 4 String; Zelle Ch 5
Week 5-6 Files & Functions; Zelle Ch 6
Week 7-8 Decision Structures; Zelle Ch 7
Week 9-10 Loops & Booleans; Zelle Ch 8
Week 11Classes; Zelle Ch 10
Week 12 Data Collections; Zelle Ch 11
Week 13 Numerical Python; McKinney Ch 4
Week 14 Scientific Python; McKinney Ch 12
Week 15 Plotting; McKinney Ch 8

Questions at the end of each chapter serve as the weekly homework assignments, due at the start of class the following week.

A take-home mid-term exam will be given in liue of the week 6 homework. A final exam will be given during the final exam period at the end of the course.

6. Discussion Policy

Students should not discuss problem sets among themselves. No discussions are allowed for the final examination.

7. Other important announcements

Absences
Students should expect to attend all classes. It is the responsibility of the student to inform the instructor(s) of an unexcused absence as soon as possible. Absences for emergency situations may be excused unofficially by the instructors. Instructor-excused absences must be obtained *prior to or on the day of the absence*. Make-ups for such absences will be at the discretion of the instructor(s). *There will be no make-ups for unexcused absences.*

The conditions under which assigned work or tests can be made up include:

- Information on excused absences related to religious observances/practices that are in accordance with ACD 304–04 "Accommodations for Religious Practices."
- Information on excused absences related to university sanctioned events activities that are in accord with ACD 304–02 "Missed Classes Due to University-Sanctioned Activities."

Academic integrity
Academic honesty is expected of all students in all examinations, papers, and laboratory work, academic transactions and records. The possible sanctions include, but are not limited to, appropriate grade penalties, course failure (indicated on the transcript as a grade of E), course failure due to academic dishonesty (indicated on the transcript as a grade of XE), loss of registration privileges, disqualification and dismissal. For more information, see http://provost.asu.edu/academicintegrity

Accommodating students with disabilities
Students who feel they will need disability accommodations in this class but have not registered with the Disability Resource Center (DRC) should contact DRC immediately. The DRC Tempe office is located on the first floor of the Matthews Center Building. DRC staff can also be reached at: (480) 965-1234 (V) or (480) 965-9000 (TTY). For additional information, visit: www.asu.edu/studentaffairs/ed/drc.

Expected classroom behavior

Classroom behavior: Be sure to arrive on time for class. Excessive tardiness will be subject to sanctions. Under no circumstances should you allow your cell phone to ring during class. Any disruptive behavior, which includes ringing cell phones, listening to your mp3/iPod player, text messaging, constant talking, eating food noisily, reading a newspaper will not be tolerated. The use of laptops (unless for note taking), cell phones, MP3, IPOD, etc. are strictly prohibited during class.

Policy against threatening behavior
All incidents and allegations of violent or threatening conduct by an ASU student (whether on-or off campus) must be reported to the ASU Police Department (ASU PD) and the Office of the Dean of Students. If either office determines that the behavior poses or has posed a serious

threat to personal safety or to the welfare of the campus, the student will not be permitted to return to campus or reside in any ASU residence hall until an appropriate threat assessment has been completed and, if necessary, conditions for return are imposed. ASU PD, the Office of the Dean of Students, and other appropriate offices will coordinate the assessment in light of the relevant circumstances.

LATE ASSIGNMENTS
Requests for modifications in assignment due dates must be made in writing and an approved by the instructor **in advance of the due date of the assignment**.

GRADE APPEALS
The College of Liberal Arts and Sciences has formal and informal channels to appeal a grade.

INCOMPLETES
A mark of **"I" (incomplete)** is given by the instructor when you have completed most of the course and are otherwise doing acceptable work but are unable to complete the course because of illness or other conditions beyond your control. Students who are granted a grade of "I" are required to arrange with the instructor for the completion of the course requirements and are recorded using the following form.

STUDENT STANDARDS
Students are required to act in accordance with university and Arizona Board of Regents policies as outlined in the ABOR Code of Conduct: Arizona Board of Regents Policies 5-301 through 5-308.

DROP AND ADD DATES/WITHDRAWALS
Please refer to the academic calendar on the deadlines to drop/withdraw from this course. Consult with your advisor and notify your instructor if you are going to drop/withdraw this course. If you are considering a withdrawal, review the following ASU policies: Withdrawal from Classes and Medical/Compassionate Withdrawal.

EMAIL COMMUNICATIONS
All email communication for this class will be done through your ASU email account. You should be in the habit of checking your ASU email regularly as you will not only receive important information about your classes, but other important university updates and information. You are solely responsible for reading and responding if necessary to any information communicated via email.

CAMPUS RESOURCES
As an ASU student, you have access to many resources on campus. This includes tutoring, academic success coaching, counseling services, financial aid, disability resources, career and internship help, and many opportunities to get involved in student clubs and organizations.

- Tutoring
- Counseling Services
- Financial Aid
- Disability Resource Center

- Major/Career Exploration
- Career Services
- Student Organizations

HARASSMENT PROHIBITIONS
ASU policy prohibits harassment on the basis of race, sex, gender identity, age, religion, national origin, disability, sexual orientation, Vietnam era veteran status, and other protected veteran status. Violations of this policy may result in disciplinary action, including termination of employees or expulsion of students. Contact Student Life (UCB 221) if you feel another student is harassing you based on any of the factors above; contact EO/AA (480-965-5057) if you feel an ASU employee is harassing you based on any of the factors above.

ESTABLISHING A SAFE LEARNING ENVIRONMENT
Learning takes place best when a safe environment is established in the classroom. Students enrolled in this course have a responsibility to support an environment that nurtures individual and group differences and encourages engaged, honest discussions. The success of the course rests on your ability to create a safe environment where everyone feels comfortable to share and explore ideas. We must also be willing to take risks and ask critical questions. Doing so will effectively contribute to our own and others intellectual and personal growth and development. We welcome disagreements in the spirit of critical academic exchange, but please remember to be respectful of others' view points, whether you agree with them or not.

SYLLABUS DISCLAIMER
The course syllabus is an educational contract between the instructor and students. Every effort will be made to avoid changing the course schedule but the possibility exists that unforeseen events will make syllabus changes necessary. The instructor reserves the right to make changes to the syllabus as deemed necessary. You will be notified in a timely manner of any syllabus changes via email or through Blackboard.

STUDENT CONDUCT STATEMENT
Students will be required to adhere to the behavior standards listed below:
- Arizona Board of Regents Policy Manual Chapter V – Campus and Student Affairs: Code of Conduct.
- ACD 125: Computer, Internet, and Electronic Communications.
- ASU's Student Academic Integrity Policy.

Students are entitled to receive instruction free from interference by other members of the class. If a student is disruptive, an instructor may ask the student to stop the disruptive behavior and warn the student that such disruptive behavior can result in withdrawal from the course. An instructor may withdraw a student from a course when the student's behavior disrupts the educational process under USI 201-10.
Course discussion messages should remain focused on the assigned discussion topics. Students must maintain a cordial atmosphere and use tact in expressing differences of opinion. Inappropriate discussion board messages may be deleted if an instructor feels it is necessary. Students will be notified privately that their posting was inappropriate. Student access to the course Send Email feature may be limited or removed if an instructor feels that particular students is sending inappropriate electronic messages to other students in the course.

RELIGIOUS ACCOMMODATIONS

Students who need to be absent from class due to the observance of a religious holiday or participate in required religious functions must notify the faculty member in writing as far in advance of the holiday or obligation as possible. Students will need to identify the specific holiday or obligatory function to the faculty member. Students will not be penalized for missing class due to religious obligations or holiday observance and a responsible for contacting the instructor to make arrangements for making up tests/assignments within a reasonable time.

SES 130 Coding for Exploration

Course description:

A series of lectures and computer labs on data processing and analysis in Earth and Space sciences using Python. Introduction to programming with scratch and python. Numerical methods for data analytics.

Textbooks:

"Python Programming: An Introduction to Computer Science 3rd Edition", by John Zelle

"Python for Data Analysis, Agile Tools for Real World Data," Wes McKinney

Weekly Readings:

Week 1 Course overview, why python?; Zelle Ch1
Week 2 Computers and Simple Programs; Zelle Ch 2
Week 3 Data Types, numbers; Zelle Ch 3
Week 4 String; Zelle Ch 5
Week 5-6 Files & Functions; Zelle Ch 6
Week 7-8 Decision Structures; Zelle Ch 7
Week 9-10 Loops & Booleans; Zelle Ch 8
Week 11Classes; Zelle Ch 10
Week 12 Data Collections; Zelle Ch 11
Week 13 Numerical Python; McKinney Ch 4
Week 14 Scientific Python; McKinney Ch 12
Week 15 Plotting; McKinney Ch 8

Table of Contents for Python Programming: An Introduction to Computer Science, 3rd Edition, John Zelle

Chapter 1 Computers and Programs
1.1 The Universal Machine
1.2 Program Power
1.3 What Is Computer Science?
1.4 Hardware Basics
1.5 Programming Languages
1.6 The Magic of Python
1.7 Inside a Python Program
1.8 Chaos and Computers
1.9 Chapter Summary
1.10 Exercises

Chapter 2 Writing Sample Programs
2.1 The Software Development Process
2.2 Example Program: Temperature Converter
2.3 Elements of Programs
2.4 Output Statements
2.5 Assignment Statements
2.6 Definite Loops
2.7 Example Program: Future Value
2.8 Chapter Summary
2.9 Exercises

Chapter 3 Computing with Numbers
3.1 Numeric Data Types
3.2 Type Conversions and Rounding
3.3 Using the Math Library
3.4 Accumulating Results: Factorials

3.5 Limitations of Computer Arithmetic
3.6 Chapter Summary
3.7 Exercises

Chapter 4 Objects and Graphics
4.1 Overview
4.2 The Object of Objects
4.3 Simple Graphics Programming
4.4 Using Graphical Objects
4.5 Graphing Future Value
4.6 Choosing Coordinates
4.7 Interactive Graphics
4.8 Graphics Module Reference
4.9 Chapter Summary
4.10 Exercises

Chapter 5 Sequences: Strings, Lists, and Files
5.1 The String Data Type
5.2 Simple String Processing
5.3 Lists as Sequences
5.4 String Representation and Message Encoding
5.5 String Methods
5.6 List Have Methods, Too
5.7 From Encoding to Encryption
5.8 Input/Output as String Manipulation
5.9 File Processing
5.10 Chapter Summary
5.11 Exercises

Chapter 6 Defining Functions
6.1 The Function of Functions
6.2 Functions, Informally
6.3 Future Value with a Function
6.4 Functions and Parameters: The Exciting Details
6.5 Functions that Return Values
6.6 Functions that Modify Parameters
6.7 Functions and Program Structure
6.8 Chapter Summary
6.9 Exercises

Chapter 7 Decision Structures
7.1 Simple Decisions
7.2 Two-Way Decisions
7.3 Multi-Way Decisions
7.4 Exception Handling
7.5 Study in Design: Max of Three
7.6 Chapter Summary
7.7 Exercises

Chapter 8 Loop Structures and Booleans
8.1 For Loops: A Quick Review
8.2 Indefinite Loops
8.3 Common Loop Patterns
8.4 Computing with Booleans
8.5 Other Common Structures
8.6 Example: A Simple Event Loop
8.7 Chapter Summary
8.8 Exercises

Chapter 9 Simulation and Design
9.1 Simulating Racquetball
9.2 Pseudo-random Numbers
9.3 Top-Down Design
9.4 Bottom-Up Implementation
9.5 Other Design Techniques
9.6 Chapter Summary
9.7 Exercises

Chapter 10 Defining Classes
10.1 Quick Review of Objects
10.2 Example Program: Cannonball
10.3 Defining New Classes
10.4 Data Processing with Class
10.5 Objects and Encapsulation
10.6 Widgets
10.7 Animated Cannonball
10.8 Chapter Summary
10.9 Exercises

Chapter 11 Data Collections
11.1 Example Problem: Simple Statistics
11.2 Applying Lists
11.3 Lists of Records
11.4 Designing with Lists and Classes
11.5 Case Study: Python Calculator
11.6 Case Study: Better Cannonball Animation
11.7 Non-sequential Collections
11.8 Chapter Summary
11.9 Exercises

Chapter 12 Object-Oriented
12.1 The Process of OOD
12.2 Case Study: Racquetball
12.3 Case Study: Dice Poker
12.4 OO Concepts
12.5 Chapter Summary
12.6 Exercises

Chapter 13 Algorithm Design and Recursion
13.1 Searching
13.2 Recursive Problem Solving
13.3 Sorting Algorithms
13.4 Hard Problems
13.5 Chapter Summary
13.6 Exercises

Table of Contents for Python for Data Analysis, Wes McKinney

Preface xi

1. Preliminaries 1
What Is This Book About? 1
Why Python for Data Analysis? 2
Python as Glue 2
Solving the "Two-Language" Problem 2
Why Not Python? 3
Essential Python Libraries 3
NumPy 4
pandas 4
matplotlib 5
IPython 5
SciPy 6
Installation and Setup 6
Windows 7
Apple OS X 7
GNU/Linux 8
Python 2 and Python 3 8
Integrated Development Environments (IDEs) 9
Community and Conferences 9
Navigating This Book 9
Code Examples 10
Data for Examples 10
Import Conventions 10
Jargon 11
Acknowledgements 11

2. Introductory Examples 13
1.usa.gov data from bit.ly 14
Counting Time Zones in Pure Python 15
Counting Time Zones with pandas 17
MovieLens 1M Data Set 22
Measuring rating disagreement 27
US Baby Names 1880-2010 28
Analyzing Naming Trends 33
Conclusions and The Path Ahead 40

3. IPython: An Interactive Computing and Development Environment 41
IPython Basics 42
Tab Completion 43
Introspection 44
The %run Command 45
Executing Code from the Clipboard 46
Keyboard Shortcuts 48
Exceptions and Tracebacks 49
Magic Commands 50
Qt-based Rich GUI Console 51
Matplotlib Integration and Pylab Mode 52
Using the Command History 54
Searching and Reusing the Command History 54
Input and Output Variables 54
Logging the Input and Output 55
Interacting with the Operating System 56
Shell Commands and Aliases 56
Directory Bookmark System 58
Software Development Tools 58
Interactive Debugger 58
Timing Code: %time and %timeit 63
Basic Profiling: %prun and %run -p 64
Profiling a Function Line-by-Line 66
IPython HTML Notebook 68
Tips for Productive Code Development Using IPython 68
Reloading Module Dependencies 70
Code Design Tips 70
Advanced IPython Features 72
Making Your Own Classes IPython-friendly 72
Profiles and Configuration 73
Credits 74

4. NumPy Basics: Arrays and Vectorized Computation 75
The NumPy ndarray: A Multidimensional Array Object 76
Creating ndarrays 77
Data Types for ndarrays 79

Operations between Arrays and Scalars 81
Basic Indexing and Slicing 82
Boolean Indexing 85
Fancy Indexing 88
Transposing Arrays and Swapping Axes 89
Universal Functions: Fast Element-wise Array Functions 91
Data Processing Using Arrays 93
Expressing Conditional Logic as Array Operations 94
Mathematical and Statistical Methods 96
Methods for Boolean Arrays 97
Sorting 97
Unique and Other Set Logic 98
File Input and Output with Arrays 99
Storing Arrays on Disk in Binary Format 99
Saving and Loading Text Files 100
Linear Algebra 101
Random Number Generation 102
Example: Random Walks 104
Simulating Many Random Walks at Once 105

5. Getting Started with pandas 107
Introduction to pandas Data Structures 108
Series 108
DataFrame 111
Index Objects 116
Essential Functionality 118
Reindexing 118
Dropping entries from an axis 121
Indexing, selection, and filtering 122
Arithmetic and data alignment 125
Function application and mapping 128
Sorting and ranking 130
Axis indexes with duplicate values 132
Summarizing and Computing Descriptive Statistics 133
Correlation and Covariance 136
Unique Values, Value Counts, and Membership 137
Handling Missing Data 139
Filtering Out Missing Data 140
Filling in Missing Data 142
Hierarchical Indexing 143
Reordering and Sorting Levels 146
Summary Statistics by Level 147
Using a DataFrame's Columns 147
Other pandas Topics 148
Integer Indexing 148
Panel Data 149

6. Data Loading, Storage, and File Formats 153
Reading and Writing Data in Text Format 153
Reading Text Files in Pieces 158
Writing Data Out to Text Format 160
Manually Working with Delimited Formats 161
JSON Data 163
XML and HTML: Web Scraping 164
Binary Data Formats 169
Using HDF5 Format 169
Reading Microsoft Excel Files 170
Interacting with HTML and Web APIs 171
Interacting with Databases 172
Storing and Loading Data in MongoDB 173

7. Data Wrangling: Clean, Transform, Merge, Reshape 175
Combining and Merging Data Sets 175
Database-style DataFrame Merges 176
Merging on Index 180
Concatenating Along an Axis 183
Combining Data with Overlap 187
Reshaping and Pivoting 188
Reshaping with Hierarchical Indexing 188
Pivoting "long" to "wide" Format 190
Data Transformation 192
Removing Duplicates 192
Transforming Data Using a Function or Mapping 193
Replacing Values 195
Renaming Axis Indexes 196
Discretization and Binning 197
Detecting and Filtering Outliers 200
Permutation and Random Sampling 201
Computing Indicator/Dummy Variables 202
String Manipulation 204
String Object Methods 204
Regular expressions 206
Vectorized string functions in pandas 209
Example: USDA Food Database 211

8. Plotting and Visualization 217
A Brief matplotlib API Primer 217
Figures and Subplots 218
Colors, Markers, and Line Styles 222
Ticks, Labels, and Legends 223
Annotations and Drawing on a Subplot 226
Saving Plots to File 229
matplotlib Configuration 229
Plotting Functions in pandas 230
Line Plots 230
Bar Plots 233
Histograms and Density Plots 236
Scatter Plots 237
Plotting Maps: Visualizing Haiti Earthquake Crisis Data 239
Python Visualization Tool Ecosystem 245
Chaco 246
mayavi 247
Other Packages 247
The Future of Visualization Tools? 247

9. Data Aggregation and Group Operations 249
GroupBy Mechanics 250
Iterating Over Groups 253
Selecting a Column or Subset of Columns 254
Grouping with Dicts and Series 255
Grouping with Functions 256
Grouping by Index Levels 257
Data Aggregation 258
Column-wise and Multiple Function Application 260
Returning Aggregated Data in "unindexed" Form 262
Group-wise Operations and Transformations 263
Apply: General split-apply-combine 264
Quantile and Bucket Analysis 267
Example: Filling Missing Values with Group-specific Values 268
Example: Random Sampling and Permutation 269
Example: Group Weighted Average and Correlation 271
Example: Group-wise Linear Regression 272
Pivot Tables and Cross-Tabulation 273
Cross-Tabulations: Crosstab 275
Example: 2012 Federal Election Commission Database 276
Donation Statistics by Occupation and Employer 278
Bucketing Donation Amounts 281
Donation Statistics by State 283

10. Time Series 285
Date and Time Data Types and Tools 286
Converting between string and datetime 287
Time Series Basics 289
Indexing, Selection, Subsetting 290
Time Series with Duplicate Indices 292
Date Ranges, Frequencies, and Shifting 293
Generating Date Ranges 294
Frequencies and Date Offsets 295
Shifting (Leading and Lagging) Data 297
Time Zone Handling 299
Localization and Conversion 300
Operations with Time Zone–aware Timestamp Objects 301
Operations between Different Time Zones 302
Periods and Period Arithmetic 303
Period Frequency Conversion 304
Quarterly Period Frequencies 305
Converting Timestamps to Periods (and Back) 307
Creating a PeriodIndex from Arrays 308
Resampling and Frequency Conversion 309
Downsampling 310
Upsampling and Interpolation 313
Resampling with Periods 314
Time Series Plotting 315
Moving Window Functions 317
Exponentially-weighted functions 320
Binary Moving Window Functions 321
User-Defined Moving Window Functions 322
Performance and Memory Usage Notes 323

11. Financial and Economic Data Applications 325
Data Munging Topics 325
Time Series and Cross-Section Alignment 326
Operations with Time Series of Different Frequencies 328
Time of Day and "as of" Data Selection 330
Splicing Together Data Sources 332
Return Indexes and Cumulative Returns 334
Group Transforms and Analysis 336
Group Factor Exposures 338
Decile and Quartile Analysis 339
More Example Applications 341
Signal Frontier Analysis 342
Future Contract Rolling 344

Rolling Correlation and Linear Regression 347

12. Advanced NumPy 349
ndarray Object Internals 349
NumPy dtype Hierarchy 350
Advanced Array Manipulation 351
Reshaping Arrays 351
C versus Fortran Order 352
Concatenating and Splitting Arrays 353
Repeating Elements: Tile and Repeat 356
Fancy Indexing Equivalents: Take and Put 357
Broadcasting 358
Broadcasting Over Other Axes 360
Setting Array Values by Broadcasting 363
Advanced ufunc Usage 363
ufunc Instance Methods 364
Custom ufuncs 366
Structured and Record Arrays 366
Nested dtypes and Multidimensional Fields 367
Why Use Structured Arrays? 368
Structured Array Manipulations: numpy.lib.recfunctions 368
More About Sorting 369
Indirect Sorts: argsort and lexsort 370
Alternate Sort Algorithms 371
numpy.searchsorted: Finding elements in a Sorted Array 372
NumPy Matrix Class 373
Advanced Array Input and Output 375
Memory-mapped Files 375
HDF5 and Other Array Storage Options 376
Performance Tips 376
The Importance of Contiguous Memory 377
Other Speed Options: Cython, f2py, C 378

Appendix: Python Language Essentials 381

Index 429