Programming in Python For Data Science

Syllabus

Programming in Python for Data Science

Time and Place

Online Sept 22 – Oct 26, 2020

Description

This course is part of the Key Capabilities for Data Science program and will teach you how to conduct data analysis in Python. During the course, you will work with powerful Python packages made for data-science, including Pandas for processing tabular data, Altair for data visualization and NumPy for working with numerical data types. You will also learn about iteration, flow control, and the data types relevant to data exploration and analysis. You will leave this course capable of processing raw data into a format suitable for analysis, writing your own analysis functions, and deriving data-driven insights via the creation of interactive visualizations and summary tables.

Prerequisites

- None

Content

We are using an interactive platform that contains a series of slide decks accompanied with multiple choice questions and interactive coding exercises to apply what you learned during the module: https://prog-learn.mds.ubc.ca/. We recommend using Chrome, Firefox or Safari to complete the content as other browsers are incompatible with the content platform.

Course Software Platforms

Students will learn to write code and perform their analysis using the Python Programming Language. Assignments as well as the final project analysis will be done using Jupyter Notebooks. Students will access the worksheets and tutorials in Jupyter Notebooks through Canvas. Students will require a laptop, chromebook or tablet to complete this course.

Learning Outcomes

By the end of the course, students will be able to:

1. Define tidy data and explain why it is an optimal format for data analysis.
2. Transform data into the tidy data format using Altair.
3. Demonstrate fundamental programming concepts such as loops and conditionals.
4. Understand the key data structures in Python.
5. Read data into Python data from vanilla (e.g., .csv) and non-standard plain text files, as well as common spreadsheet file types (e.g., .xls).

6. Construct simple plots using pandas
7. Manipulate a single data table by:

   7.1 Filtering rows based on a criterion or combination of criteria
   7.2 Selecting variables
   7.3 Creating new variables and modifying pre-existing ones
   7.4 Rearranging the observations or variables by sorting.
8. Manage and manipulate data with dates and times, missing values and categorical variables as well as renaming dataframe columns.
9. Produce human-readable code that incorporates best practices of programming and coding style.

Course Facilitator

Position	Name	Email	Office Hours	Office Location
Course Facilitator	Socorro Dominguez	sedv8808@gmail.com	TBD	Zoom

Course Structure

There are 8 modules in total for this course. Each module will be accompanied by an autograded assignment to be submitted. The students will be responsible to complete the material of 2 modules per week. In weeks 3 and 5, students will need to take an online 45 minute open book quiz testing, material from Modules 1-4 and 5-8 respectively. This quiz can be taken at any point during the week. At the end of the 8th module, students will have an individual project to complete where they are required to use the skills they learned in the course and produce an analysis regarding an assigned dataset.

Assignments

Each week students will submit 2 assignments for a grade. **Assignment due dates are posted on Canvas.** To open the assignment, click the link (e.g. `assignment_01`) from Canvas. To submit your assignment, just make sure your work is saved (File -> Save and Checkpoint to be sure) **on our server** (i.e., using the link from Canvas) before the deadline. Our server will automatically snapshot at the due date/time. We have supplied informative videos on how to do this in assignment 1 and referenced the links in subsequent assignments.

Course breakdown

Deliverable	Percent Grade
Assignments	48
2 Quizzes	30(15% each)
Final project	22

Schedule

Module	Title	Description
Module 1	Python & Pandas - An Unexpected Friendship	An Introduction to dataframes, the Python package Pandas, simple manipulations and Visualizations.

Module	Title	Description
Module 2	Not So Scary Wrangling (Table Manipulation and Chaining)	Importing different types of files, performing more advanced table manipulations (modifying and creating new columns) as well as method chaining conventions (style, including multi-line).
Module 3	Tidy Data and Joining Dataframes	Tidy data and how to transform your dataset into a tidy format.This module will also focus on how to concatenate and join multiple dataframes.
Module 4	Python Without the "Eek" (Basic Python)	Basic Python data types and structures. You will explore what data types and structures are used to create a Pandas dataframe and how understanding column dtypes is important to data analysis.
Module 5	Making Choices and Repeating Iterations	Conditionals Statements and the fundamentals on creating code that efficiently repeats the same operations by adhering the DRY principle.
Module 6	Functions Fundamentals and Best Practices	Developing good habits when writing functions like including docstrings, defensive programming, test-driven development and how to compose useful functions.
Module 7	Importing Files and the Coding Style Guide	Importing files and libraries from other directories and stylize your code for optimal readability.
Module 8	A Slice of NumPy and Advanced Data Wrangling	NumPy arrays and more advanced wrangling techniques such as handling columns with dates and strings and identifying null values.

Submission Schedule

Assignments are due on Sunday at 6pm. Office hours will be held twice a week to support and answer questions regarding the concepts learned in lecture. Quizzes will be opened for up to 7 days, please check Canvas for quiz submission dates.

Week	Modules to Complete	Submissions Due
Week 1	Module 1: Python & Pandas - An Unexpected Friendship Module 2: Not So Scary Wrangling (Table Manipulation and Chaining)	assignment_01, assignment_02
Week 2	Module 3: Tidy Data and Joining DataframesModule 4: Python Without the "Eek" (Basic Python)	assignment_03, assignment_04
Week 3	Module 5: Making Choices and Repeating IterationsModule 6: Functions Fundamentals and Best Practices	assignment_05, assignment_06, quiz_01

Week	Modules to Complete	Submissions Due
Week 4	Module 7: Importing Files and the Coding Style GuideModule 8: A Slice of NumPy and Advanced Data Wrangling	assignment_07, assignment_08
Week 5	NA	quiz_02, final_project

Policies

Late Assignments

Students should submit all assignments by the due date. If for any reason a student is unable to do so, they can submit the assignments before the course end date. This is the final allowed submission date and is non-negotiable.

Late Quizzes

We allow 1 week to complete each quiz. if you've submitted a quiz late or did not submit a quiz at all, please contact the course facilitator and we will try to make accommodations.

Autograder Policy

Many of the questions in assignments are graded automatically by software. The grading computer has exactly the same hardware setup as the server that students work on. Students should make sure their assignments are *reproducible*, and run from beginning to end on the autograding computer. In particular, **please ensure that any data that needs to be downloaded is done so by the assignment notebook with the correct filename to the correct folder**.

Re-grading

If you have concerns about the way your work was graded, please contact the course facilitator.

Device/Browser

Students are responsible for using a device and browser compatible with all functionality of Canvas. Chrome or Firefox browsers are recommended; Safari has had issues with Canvas quizzes in the past.

Academic Integrity

The academic enterprise is founded on honesty, civility, and integrity. As members of this enterprise, all students are expected to know, understand, and follow the codes of conduct regarding academic integrity. At the most basic level, this means submitting only original work done by you and acknowledging all sources of information or ideas and attributing them to others as required. This also means you should not cheat, copy, or mislead others about what is your work. Violations of academic integrity (i.e., misconduct) lead to the breakdown of the academic enterprise, and therefore serious consequences arise and harsh sanctions are imposed. For example, incidences of plagiarism or cheating may result in a mark of zero on the assignment or exam and more serious consequences may apply if the matter is referred to the President's Advisory Committee on Student Discipline. Careful records are kept in order to monitor and prevent recurrences.

A more detailed description of academic integrity, including the University's policies and procedures, may be found in the Academic Calendar at http://calendar.ubc.ca/vancouver/index.cfm?tree=3,54,111,0.

Plagiarism

Students must correctly cite any code or text that has been authored by someone else or by the student themselves for other assignments. Cases of plagiarism may include, but are not limited to:

- the reproduction (copying and pasting) of code or text with none or minimal reformatting (e.g., changing the name of the variables)
- the translation of an algorithm or a script from a language to another
- the generation of code by automatic code-generation software

An "adequate acknowledgement" requires a detailed identification of the (parts of the) code or text reused and a full citation of the original source code that has been reused.

The above attribution policy applies only to assignments. **No code or text may be copied (with or without attribution) from any source during a quiz or exam. Answers must always be in your own words. At a minimum, copying will result in a grade of 0 for the related question.**

Repeated plagiarism of any form could result in larger penalties

Attribution

Parts of this syllabus (particularly the policies) have been copied and derived from the UBC MDS Policies as well as the Syllabus from UBC's DSCI 100 course.