Syllabus for DSCI 100 - Introduction to Data Science

Time

Section	Instructor
DSCI 100 - 002	Daniel Chen
DSCI 100 - 003	Melissa Lee

DSCI 100 100 has it's own repository this semester. Please refer to https://github.com/eldadHaber/DSCI100

Course Description

Use of data science tools to summarize, visualize, and analyze data. Sensible workflows and clear interpretations are emphasized.

Long Version: In recent years, virtually all areas of inquiry have seen an uptake in the use of data science tools. Skills in the areas of assembling, analyzing, and interpreting data are more critical than ever. This course is designed as a first experience in honing such skills. Students who have completed this course will be able to implement a data science workflow using either the R or Python programming language, by "scraping" (downloading) data from the internet, "wrangling" (managing) the data intelligently, and creating tables and/or figures that convey a justifiable story based on the data. They will be adept at using tools for finding patterns in data and making predictions about future data. There will be an emphasis on intelligent and reproducible workflow, and clear communications of findings. No previous programming skills necessary; beginners are welcome!

Textbook

This course uses "Data Science: A First Introduction" which is available both in an R-version and a Python-version. This textbook is open source and will always be freely available on the web.

Hardware & Software

Students are required to bring a laptop, chromebook or tablet to both lectures and tutorials. Students who do not own a laptop, chromebook, or tablet may be able to loan a laptop from the UBC library.

All other required software will be provided by the instructors. Students will learn to perform their analysis using the R or Python programming language depending on which section they enrolled in. Worksheets and tutorial problem sets as well as the final project analysis, development, and reports will be done using Jupyter Notebooks accessed via Canvas.

Prerequisite Knowledge

- distance between points on a graph
- percentages, average
- powers, roots, basic operations, logarithm, exponential
- equation of a line / plane

As an example, British Columbia's Math 12 or Pre-Calculus 12 courses would satisfy the prerequisite.

Learning Outcomes

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

- Read data using computation from various sources (local and remote plain text files, spreadsheets and databases)
- Wrangle data from their original format into a fit-for-purpose format.
- Identify the most common types of research/statistical questions and map them to the appropriate type of data analysis.
- Create, and interpret, meaningful tables from wrangled data.
- Create, and interpret, impactful figures from wrangled data.
- Collaborate with others using version control.
- Apply, and interpret the output of simple classifier and regression models.
- Make and evaluate predictions using a simple classifier and a regression model.
- Apply, and interpret the output of, a simple clustering algorithm.
- Distinguish between in-sample prediction, out-of-sample prediction, and cross-validation.
- Calculate a point estimate in the context of statistical inference and explain how that relates to the population quantity being estimated.
- Accomplish all of the above using workflows and communication strategies that are sensible, clear, reproducible, and shareable.

Teaching Team

Note that your TAs are students too; they may have class right before their office hours, and they may run a few minutes late. Please be patient!

Section	Position	Name	Email
All	Course coordinator	Julia Peng	courses[-at-]stat.ubc.ca
002	Instructor	Daniel Chen	daniel.chen[-at-]stat.ubc.ca
003	Instructor	Melissa Lee	melissa.lee[-at-]stat.ubc.ca

Please contact the course coordinator about any administrative questions. Please read the course policy (e.g., late registration, missing exam/assignment due to sickness) below before contacting.

When sending emails, please include your student number and DSCI 100 [Section Number] in the subject line.

Assessments

The course will have exams, worksheets, tutorials, iClicker questions and a project for assessments.

Exams

- One midterm covering ~3 weeks of material
  - Same time & location as week 4 tutorials. Invigilated in-person.
- One cumulative final covering all the material in the course
  - To be scheduled by Classroom Services. Invigilated in-person.

The midterm and final are both closed book exams, where you will only have access to a reference sheet for common functions and operations. We suggest that already now get familiar with the reference sheet relevant to your section (Python version or R version) to use it more efficiently at the exam.

Note: Since DSCI 100 is a large course with multiple sections (hence, multiple versions of exams), the instructors reserve the rights to scale grades in order to maintain equity among sections according the UBC campus wide policies and regulations.

Worksheets and Tutorials

In each class (lecture and tutorial) there will be an assignment:

- Lecture and tutorial worksheet **due dates are posted on Canvas.**
- To open the assignment, click the link (e.g. `worksheet_intro`) from Canvas.
- To submit your assignment, just make sure your work is saved **on our server** (`File -> Save Notebook` to be sure).
- At the deadline, our server will automatically snapshot your work.
- You **must access the lecture and tutorial worksheets through our Canvas course page** (as opposed to the worksheets publicly available via Github). Otherwise your worksheets may not be marked!

iClicker

During each lecture there will be iClicker questions to help check your understanding of the course material. iClicker grade will be based on participation. You **must attend the section you are registered in.** It is your responsibility to make sure that the student ID and name associated with your iClicker account matches the Canvas gradebook. If you need help connecting to iClicker please see iClicker Cloud Student Guide.

Project

The project will provide additional practice with the data science skills we teach in the class. It is an extension to the worksheets and tutorials by providing little to no prompts

and scaffolding code. You will **only** be able to use the dataset(s) we have provided you to use in Canvas. Further details regarding the project will be announced during the term.

Course breakdown

Deliverable	Percent Grade
Worksheets	6
Tutorials	7
iClicker	3
Project	3
Midterm	30
Final	50
Bonus regrade percent	1

DSCI 100 100 has it's own repository this semester. Please refer to it's syllabus and materials at https://github.com/eldadHaber/DSCI100

Schedule

Session	Topic	Description
1	Introduction	Learn to use a programming language and Jupyter notebooks as you walk through a real world data Science application that includes downloading data from the web, wrangling the data into a useable format and creating an effective data visualization.
1	Reading in data locally and from the web	Learn to read in various cases of data sets locally and from the web. Once read in, these data sets will be used to walk through a real world data Science application that includes wrangling the data into a useable format and creating an effective data visualization.
2	Cleaning and wrangling data	This week will be centered around tools for cleaning and wrangling data. Again, this will be in the context of a real world data science application and we will continue to practice working through a whole case study that includes downloading data from the web, wrangling the data into a useable format and creating an effective data visualization.
2	Effective data visualization	Expand your data visualization knowledge and tool set beyond what we have seen and practiced so far. We will move beyond scatter plots and learn other effective ways to visualize data, as well as some general rules of thumb to follow when creating visualations. All visualization tasks this week will be applied to real world data sets. Again, this will be in the context of a real world data science application and we will continue to practice working through a whole case study that includes downloading data from the web,

		wrangling the data into a useable format and creating an effective data visualization.
3	Version control	This chapter will introduce the concept of using version control systems to track changes to a project over its lifespan, to share and edit code in a collaborative team, and to distribute the finished project to its intended audience. This chapter will also introduce how to use the two most common version control tools: Git for local version control, and GitHub for remote version control. We will focus on the most common version control operations used day-to-day in a standard data science project. There are many user interfaces for Git; in this chapter we will cover the Jupyter Git interface.
3	Classification	This chapter and the next together serve as our first foray into answering predictive questions about data. In particular, we will focus on classification, i.e., using one or more variables to predict the value of a categorical variable of interest. This chapter will cover the basics of classification, how to preprocess data to make it suitable for use in a classifier, and how to use our observed data to make predictions. The next chapter will focus on how to evaluate how accurate the predictions from our classifier are, as well as how to improve our classifier (where possible) to maximize its accuracy.
4	Midterm	
4	Classification, continued	This chapter continues the introduction to predictive modeling through classification. While the previous chapter covered training and data preprocessing, this chapter focuses on how to evaluate the performance of a classifier, as well as how to improve the classifier (where possible) to maximize its accuracy.
4	Regression	This chapter continues our foray into answering predictive questions. Here we will focus on predicting numerical variables and will use regression to perform this task. This is unlike the past two chapters, which focused on predicting categorical variables via classification. However, regression does have many similarities to classification: for example, just as in the case of classification, we will split our data into training, validation, and test sets, we will use scikit-learn workflows, we will use a K-nearest neighbors (KNN) approach to make predictions, and we will use cross-validation to choose K. We will focus on prediction in cases where there is a response variable of interest and a single explanatory variable.

5	Regression, continued	Up to this point, we have solved all of our predictive problems—both classification and regression—using K-nearest neighbors (KNN)-based approaches. In the context of regression, there is another commonly used method known as linear regression. This chapter provides an introduction to the basic concept of linear regression, shows how to use scikit-learn to perform linear regression in Python, and characterizes its strengths and weaknesses compared to KNN regression. The focus is, as usual, on the case where there is a single predictor and single response variable of interest; but the chapter concludes with an example using multivariable linear regression when there is more than one predictor.
5	Clustering	As part of exploratory data analysis, it is often helpful to see if there are meaningful subgroups (or clusters) in the data. This grouping can be used for many purposes, such as generating new questions or improving predictive analyses. This chapter provides an introduction to clustering using the K-means algorithm, including techniques to choose the number of clusters.
6	Introduction to statistical inference	A typical data analysis task in practice is to draw conclusions about some unknown aspect of a population of interest based on observed data sampled from that population; we typically do not get data on the entire population. Data analysis questions regarding how summaries, patterns, trends, or relationships in a data set extend to the wider population are called inferential questions. This chapter will start with the fundamental ideas of sampling from populations and then introduce two common techniques in statistical inference: point estimation and interval estimation.
6	Introduction to statistical inference, continued	Unfortunately, we cannot construct the exact sampling distribution without full access to the population. However, if we could somehow approximate what the sampling distribution would look like for a sample, we could use that approximation to then report how uncertain our sample point estimate is (as we did above with the exact sampling distribution). There are several methods to accomplish this; in this course, we will use the bootstrap. We will discuss interval estimation and construct confidence intervals using just a single sample from a population. A confidence interval is a range of plausible values for our population parameter.
6	Project report due	
7	Final	Cumulative. Covers all the material. To be Scheduled by

Classroom Services

Policies

Code of Conduct

All participants in our course and communications are expected to show respect and courtesy to others. To creating a friendly and respectful place for learning, teaching and contributing, you are expected to read and follow the DSCI 100 Code of Conduct.

Late Registration

Students who register for the class late have 1 week from their registration date on Canvas to complete all prior assignments.

Late Assignments / Absences

For examinations, students **must be present** at the invigilation venue (in class, examination centre, etc) to take exams; otherwise they will be considered to have missed the exam and will be assigned a grade of zero.

Students who will miss an exam **must provide a self-declaration of academic concession prior to the exam** (see Canvas homepage for the academic concession form) and make arrangements with the Instructor. Failing to present a declaration within a reasonable timeframe before the exam will result in a grade of zero.

There will be **no extensions for the lecture and tutorial worksheets**; late assignments will receive a grade of zero. Instead, we will drop the lowest 2 grades on tutorials and worksheets for the semester (1 lowest tutorial + 1 lowest worksheet). This policy is meant to cover illness/unexpectancies. The worksheet and tutorial that you were not able to complete before the deadline will be covered by this policy. However, if you have extenuating circumstances and require further accommodations for subsequent requests, please contact the course coordinator with supporting documents, and we will deal with them case by case.

Students who miss a lecture for their registered section will receive a iClicker grade of 0 for that lecture. There will be no make ups or accomodations for missing i-clickers. Instead, we will drop the 1 lowest grade on iClicker lectures to accomodate late registration and/or unforseeable events.

For all other assignments and the course project, a **late submission will receive a 50% penalty**.

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. No assignment, when completed, should take longer than 5 minutes to run on the server. The

autograder will automatically stop (time out) for each student assignment after a maximum of 5 minutes; **any ungraded questions at that point will receive a score of 0.**

Students are responsible for making 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.** A common mistake is to manually download data when working on the assignment, making the autograder unable to find the data and often resulting in an assignment grade of 0.

In short: whatever grade the autograder returns after 5 minutes (assuming the teaching team did not make an error) is the grade that will be assigned.

Re-grading

To account for minor grading errors throughout the course, every student will get a bonus of one percentage point at the end of the semester. We only accept tutorial regrade requests for major errors in grading. If you think the grading team made an error of more than 25% on a single assignment, you may fill out a tutorial regrade form at the end of the semester (regrade form will be released near the end).

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 exams in the past.

Missed Final Exam

Students who miss the final exam must report to their faculty advising office within 72 hours of the missed exam, and must supply supporting documentation. Only your faculty advising office can grant deferred standing in a course. You must also notify your instructor prior to (if possible) or immediately after the exam. Your instructor will let you know when you are expected to write your deferred exam. Deferred exams will ONLY be provided to students who have applied for and received deferred standing from their faculty.

Academic Concession Policy

Please see UBC's concession policy for detailed information on dealing with missed coursework and exams under circumstances of an acute and unanticipated nature.

See our Canvas homepage for the academic concession form.

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 and/or text by automatic code-generation software or large language model

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 an exam. Answers must always be in your own words. At a minimum, copying will result in a grade of 0 for the related assignment.**

Repeated plagiarism of any form could result in larger penalties, including failure of the course.

Resources

For additional information, please check out these useful student resources, the survival tips from your TAs, and the Frequently Asked Questions. If you want to use any of this material elsewhere, please read the license.

Attribution

Parts of this syllabus (particularly the policies) have been copied and derived from the UBC MDS Policies.