CS2035a Data Analysis and Visualization 2017

Professor John L. Barron
Dept. of Computer Science
University of Western Ontario
London, Ontario, Canada, N6A 5B7
519-661-2111 x86896
barron@csd.uwo.ca

CS2035 Dealing with Data: Analysis and Visualization

Calendar copy:

It is becoming increasingly common in a number of disciplines to be faced with an overwhelming quantity of data that must be processed, interpreted, and understood in order for it to be of value and truly useful. As a result, skills and background in **data analysis** and **data visualization** are quickly becoming essential to these disciplines. The purpose of this course is to develop and refine these skills and background, using MATLAB as a software platform for understanding and applying the fundamental techniques in statistics, mathematics, and computing necessary for gaining mastery over your data.

Is this a MatLab Course?

YES, this is a MatLab course!!! However, this course assumes no prior programming knowledge, although some knowledge of programming in general would be very helpful. MatLab lets you start programming right away, with just some rudimentary knowledge.

Obtaining a copy of The MatLab software

As a Western student, you are entitled to install a free copy of the MatLab software on your home computer or laptop. See the course webpage for details on how install MatLab on your personal computing device.

Prerequisites

1.0 courses in Applied Mathematics, Calculus, Mathematics, Statistics (including Introductory Statistics), or the former Linear Algebra, or permission of the Department/Instructor. Beware of the following Dean's rule:

> Unless you have either the requisites for this course or written special permission from your Dean to enroll in it, you may be removed from this course and it will be deleted from your record. This decision may not be appealed. You will receive no adjustment to your fees in the event that you are dropped from a course for failing to have the necessary prerequisites.

Course Resources:

1. Recommended text: Mastering MATLAB, Duane Hanselmann and Bruce Littlefield, Pearson (Prentice Hall), 2012. However, any MATLAB textbook would mostly like be sufficient.

2. Another good text: MatLab Programming for Engineers (5^{th} edition), Stephen J. Chapman, Cenage Learning, 2016 [has GUI, Object Oriented, I/O Chapters, updated for MatLab R2014b].

3. The course notes and some MATLAB functions and data will be on the course webpage. The course webpage is at www.csd.uwo.ca/Courses/CS2035b.

4. All MATLAB toolboxes are fully documented on www.mathworks.com/help/documentation-center.html.

5. Mathworks offers a number of Webinars on various topics on www.mathworks.com.

6. Mathworks also offers online documentation, discussion forums, and numerous other resources.

7. Google can find just about anything to do with MATLAB!!!

Class and Lab time schedule:

There are two 1 and $\frac{1}{2}$ hour (75 minutes) classes each week:

- Monday and Friday, 3:30-4:45pm, Health Sciences Building HSB236.

There are two labs at the same time:

- Friday 2:30-3:30pm - Health Sciences Building HSB14 and HSB16.
- The timetable for the University of Western Ontario winter term lectures and labs are posted at:

  https://studentservices.uwo.ca/secure/Timetables/mastertt/ttindex.cfm

Important Dates

The following are important days that may affect students taking CS2035b:

1. First day of classes - Thursday, January 5th
2. First CS2035 class - Friday, January 6th
3. Family day - Monday, February 20th, (holiday)
4. Reading week (also known as slack/skiing week) - Monday to Friday, February 20-24 (this week there are no classes, and yes, it includes family day as the 1st day)
5. Good Friday, April 14th
6. Easter Sunday, April 16th
7. Last day of classes, Friday, April 7th
8. Study days (before exam) - Saturday, April 8th

9. Exam Period - April 9th-30th (21 days) [Typically the final exam date will be released early in the term but students are required to attend the exam. Purchasing a cheap ticket to go home and then finding out the exam is after your travel ternary is not sufficient grounds for as appeal.]

Lab, Assignment and Exam Dates

1. Each lab is due on the Friday of that lab: Lab 1: Jan. 13th, Lab 2: Jan. 20nd, Lab 3: Jan. 27th, Lab 4: Feb. 3th, Lab 5: Feb. 10th, Lab 6: Feb. 17th, Lab 7: Mar. 3th, Lab 8: Mar. 10th, Lab 9: Mar. 17th, Lab 10: Mar. 24th, Lab 11: Mar. 31st, Lab 12: April 7th. It is possible that the April 7th lab may be cancelled as it is on the last day of classes.

2. You are required to complete 8 labs. Each lab is worth 1.25 marks (for a maximum of 10 marks). Lab attendance is mandatory and attendance is taken. Photo identification is required. There are (potentially) either 11 or 12 labs but you only have to attend 8 to obtain full marks. There are no bonus marks for completing more than 8 labs but note that there may be a final exam question based on the labs.

3. The purpose of the labs is to introduce or expand on the core material of this course, and to provide programming exercises with concepts. Lab instructions are posted on the course website, and include material that must be read before the lab. Attendance at labs is a required part of the

course. Missing labs is a stupid way to lose easy marks!!!

4. You must attend the lab session for which you are registered. There are no make-up labs and students who are absent from a lab do **NOT** have the option of just submitting the lab online via OWL. Attendance is required.

5. Assignment 1 is worth 9% and is due Sunday, January 23th at 11:55pm (via Owl).

6. Assignment 2 is worth 9% and is due Sunday, February 13th at 11:55pm (via Owl).

7. Assignment 3 is worth 9% and is due Sunday, March 13th at 11:55pm (via Owl).

8. Assignment 4 is worth 8% and is due Sunday, April 2nd at 11:55pm (via Owl).

9. Midterm Exam is Monday, February 27th in class (75 minute exam, open book but NO laptops or cellphones allowed) and worth 20%. There is no makeup midterm exam, rather if you miss the midterm, your final exam

will count for all the exam grade. Note this time was chosen because it is the last Friday before the reading week.

10. Final Exam TBA (3 hours, open book, no laptop or cellphones allowed) and worth 35% (or 55% if you do better on the final exam than on the midterm).

Office Hours

Friday 10am-noon in my office MC355e (note that all assignments are due at 11:55pm on Sundays and this office hour should be a good time to straighten out any last minute problems in timely fashions on your assignments). Office hours start January 20th, 2017.

Email Contact:

We occasionally need to send email messages to the class or to students individually. Email is sent to your Western email address as assigned to you by ITS (Information Technology Services). The is your university email address. It is your responsibility to read this email frequently and regularly (I recommend daily). You may wish to have this email forwarded to an alternative email address. See the ITS website for directions on forwarding email. Verify that any forwarding works! Nevertheless, emails sent out to your uwo email address will assume to have been received even if the forwarding does not work! Important emails about the class, assignments, etc will be sent to these email addresses. You should note that email at ITS and other email providers may have quotas or limits on the amount of space they dedicate to each account. Unchecked mail may accumulate beyond these limits and you may be unable to retrieve important messages from your instructors. It is your responsibility to monitor your email and is not an acceptable excuse for anything to not having received an email!!! You are encouraged to contact the course instructor via email, with concise and appropriate questions you may have regarding course

and lecture materials or clarification of assignments. Note that email sent from accounts different from ITS may not reach its destination (it might be waylaid by a spam filter, for example): hence you are instructed to send your questions with your Western account to be on the safe side.

Course Website:

Point your favourite browser at `www.csd.uwo.ca/courses/CS2035b` for the course webpage. Assignment and lab pdf files will be available there. Also course lectures and MatLab programs referred to in the lectures will be available from there. On the other hand, all assignments and labs will be submitted via OWL and all marks will be posted on OWL.

Lecture Notes:

are be available on the course website, www.csd.uwo.ca/courses/CS2035b/. Pdfs of the lectures and MatLab code relevant to these lectures are password protected and the password will be given out in class. The username is “class”.

Teaching Assistants:

Ayan Chaudhury (achaud29@csd.uwo.ca), Office Hours TBA
Seereen Noorwali (snoorwal@uwo.ca), Office Hours TBA

Computing Facilities:

The labs are in HSB14 or HSB16, general ITS university computing labs. The latest version of MatLab is available there on all the machines. Many students will have their own MatLab software on their laptops: these are acceptable as long as they are version 2009 or better.

Other Labs:

- There are other labs available to you that are open on the weekend.
- These include NCB105 and SS1032 as well as the Genlab located in Taylor Library.

- Hours for the these labs can be found at:

  https://www.uwo.ca/its/genlabs/hours.html

  The locations of all Western labs can be found at:

  https://www.uwo.ca/its/genlabs/genlabs-western.pdf

- All computers in the university computing labs will have MatLab available on them (probably MatLab R2015b or better).

Course Outline:

This course is broken down into down into 3 modules.

Module 1: Introduction to MATLAB and the MATLAB toolboxes

1. The components of MATLAB (command window, editor, figures, toolboxes)
2. Simple MATLAB programming
3. Data types (single, double, integer, character arrays, records, cells)
4. Variables and arrays
5. Control flow (loops, while, if-then-else, switch (case) statements)
6. Simple I/0 (reading/writing binary, ASCII and mat files)

7. Some built-in mathematical MATLAB functions
8. Scripts and functions (*.m files)
9. Arrays and simple array operations
10. Multidimensional arrays
11. Simple 2D/3D plots and the print statement
12. Matrix algebra
13. Serialization versus Vectorization, JIT compilation
14. Serialized versus Vectorized I/O
15. Graphical User Interfaces (GUIs) using GUIDE
16. MATLAB Programming Interfaces (such as C, Fortran and Java)
17. Object Oriented MatLab

Module 2: Basic Data Visualization

1. Setting the camera and the lighting model
2. Mesh and surface plots
3. Colormaps and texture
4. Representation arbitrary shaped 3D objects using patches
5. Using transparency to display data
6. Volume Visualization: scalar values, slice planes, isosurfaces, vector data
7. Stream lines/ribbons and tubes
8. Images, movies and sound

Module 3: Basic Data Analysis

1. Some basic operations: mean, standard deviation, weighted average, median, covariance matrices
2. Random number generation
3. Histograms
4. Data correlation (Pearson's coefficient)
5. Hypothesis testing (z-test and t-test)
6. Chi-square goodness-of-fit and other variance tests
7. Regression analysis (including linear, nonlinear and robust regression)
8. Scatter/Box/Distribution plotting
9. Probability Density/Cumulative distributions
10. Normal, Exponential, Poisson, Rayleigh, Rican distributions

11. Performance curves (ROC)

Note: This list of topics may be too ambitious to teach in a 0.5 credit 1 term course. In this case, an appropriate subset of this material will be taught.

Other Grading Considerations

- If for any reason the assignment schedule cannot be adhered to, the assignment marks will be pro-rated. The assignments are worth 35% of the overall mark for the course. If an assignment has to be cancelled for any reason, the remaining assignment weights will be prorated to add up to 35%.
- If you obtain a higher grade on the final than on the midterm the final grade make will count for the complete exam grade.
- If you miss the midterm exam for any reason, the final exam make will comprise the entire exam mark. There will be no midterm makeup exam.
- You need to pass the assignments (average 50%) to pass the course.
- You need to obtain 40% on the exams to pass the course.
- You need to obtain 50% on the exams to receive a grade of 65% or more in the course.

- Neither cellphones or laptop computed can be brought to exams. We cannot be responsible for the storage of these devices at the front of the class. Procession of either of these devices will be considered to constitute cheating!!!

Appeal of Assignment Marks

1. Appeals of assignment marks should be addressed to your T.A. first. If you and the T.A. cannot agree, then the T.A. and the student will discuss the situation with the instructor. That decision will be final.

2. Appeals must occur within 1 week from the first day that the marked assignments or midterm exam were made available to students. After that 1 week period has gone by, no further appeals will be considered and the marks are considered final. Note that this rule applies even if assignments are not picked up when passed back. The week (8 day) count down starts from the date the assignment is passed back.

Late Assignment Policy:

Assignment due dates are always at 11:55pm (via Owl). It is not necessary to skip a class to put the final touches on an assignment. Hardcopies of your assignments are not necessary and the Owl date of submission will be the "official" date of submission. Assignments mailed to the instructor or TA will not be accepted. Assignments passed in 1 day late will have 5% deducted while assignments passed in 2 days late will have 10% deducted. No assignments will be accepted after 2 days. Saturday and Sunday count as 1 day in determining the lateness of an assignment but since all assignments are due on a Sunday this rule will not apply unless an assignment date is changed. Extensions can only be granted by the course instructor. If you have serious medical or compassionate grounds for an extension, you should take supporting documentation to the Academic Counseling office of your faculty, who will contact the instructor. Workload, exams, minor illnesses, and home computer problems are not valid reasons for being unable to complete an assignment within the allotted time (unless your academic councillor thinks otherwise).

Academic Accommodation for Medical Illness:

If you are unable to meet a course requirement due to illness or other serious circumstances, you must provide valid medical or other supporting documentation to your Academic Counseling office as soon as possible and contact your instructor immediately. It is the student's responsibility to make alternative arrangements with their instructor once the accommodation has been approved and the instructor has been informed. For further information please see: www.uwo.ca/univsec/handbook/appeals/accommodation_medical.pdf A student requiring academic accommodation due to illness should use the Student Medical Certificate when visiting an off-campus medical facility or an Accommodation Certificate from Student Health Services. This form can be found at:

http://www.uwo.ca/univsec/handbook/appeals/medicalform.pdf

Ethical Conduct:

Scholastic offenses are taken seriously and students are directed to read the appropriate policy, specifically, the definition of what constitutes a Scholastic Offense, at the following Web site:

http://www.uwo.ca/univsec/handbook/appeals/scholoff.pdf

Plagiarism: Students must write their essays and assignments in their own words. Whenever students take an idea, or a passage from another author, they must acknowledge their debt both by using quotation marks where appropriate and by proper referencing such as footnotes or citations. Plagiarism is a major academic offense. All assignments are individual assignments. You may discuss approaches to problems among yourselves; however, the actual details of the work (assignment coding, answers to concept questions, etc.) must be an individual effort. Assignments that are judged to be the result of academic dishonesty will, for the student's first offense, be given a mark of zero with an additional penalty equal to the weight of the assignment also being applied. You are responsible for reading and respecting the Department of Computer

Science policy on Scholastic Offenses and Rules of Ethical Conduct. The University of Western Ontario may use software for plagiarism checking. Students may be required to submit their written work and programs in electronic form for plagiarism checking.

Statement on Academic Offenses

Scholastic offenses are taken seriously and students are directed to read the appropriate policy, specifically, the definition of what constitutes a Scholastic Offense, at the following web site:

www.uwo.ca/univsec/handbook/appeals/scholastic_discipline_undergrad.pdf.

Additionally,

1. All required papers may be subject to submission for textual similarity review to the commercial plagiarism detection software under license to the University for the detection of plagiarism. All papers submitted for such checking will be included as source documents in the reference database for the purpose of detecting plagiarism of papers subsequently submitted to the system. Use of the service is subject to the licensing agreement, currently between The University of Western Ontario and Turnitin.com (http://www.turnitin.com).

2. Computer-marked multiple-choice tests and/or exams may be subject to submission for similarity review by software that will check for unusual coincidences in answer patterns that may indicate cheating.

Tutoring:

The role of tutoring is to help students understand course material. Tutors should not write part or all of an assignment. Having employed the same tutor as another student is not a legitimate defense against an accusation of collusion, should two or more students hand in assignments considered similar beyond the possibility of coincidence.

Mental Health:

Students who are in emotional/mental distress should refer to Mental Health website:

http://www.uwo.ca/uwocom/mentalhealth/

for a complete list of options about how to obtain help.

Accessibility:

Please contact the course instructor if you require material in an alternate format or if you require any other arrangements to make this course more accessible to you. You may also wish to contact Services for Students with Disabilities (SSD) at 661-2111 x 82147 for any specific question regarding an accommodation.

What is MatLab?

- MatLab (**Mat**rix **Lab**oratory) is a software package sold by Mathworks (started in 1984, located in Natick, Massachusetts) that provides the user with software for high performance numerical computation and visualization capabilities. It has more than 80 toolboxes that can be used for various computation, graphics and animation tasks.
- Available toolboxes include those for linear algebra, data analysis, signal processing, image processing, image acquisition, optimization, parallelization, computer vision, distributed computing, bioinformatics, data acquisition, database, report generation, financial analysis, fuzzy logic, neural networks, robust systems, control systems, simulink, spreadsheet links (Excel), statistics, symbolic math and wavelets, to list a few.
- MatLab provides an interface for the programming languages C, Java and Fortran so that one can integrate functions/objects written in these languages into a MatLab program.
- The main building block of MatLab is the matrix (vectors, scalars are

special cases of matrices). MatLab is optimized for matrix operations, i.e. has *vectorized* code. MatLab can execute instructions as fast as C or Fortran if properly coded via vectorization or using Just In Time [JIT] compilation of loops.

- Got a machine with multi-cores? You may be able to use the parallel computing toolbox to significantly speed up existing MatLab code by running code in parallel on the cores or running code on your machine's GPU (Graphics Processing Unit).

- The latest versions of MatLab runs on Windows (for both 32 and 64 bit machines) and on Unix, Linux and Mac OSX (for 64 bit machines only).

Representation of Numbers on Computers

- Normally, we view numbers in base 10 (decimal). For example 148 or 148_{10}, where the subscript represents the base of the decimal number.
- Computers use base 2 (binary) to represent numbers.
- For example, 10010100_2 is equal to 148_{10}. 10010100_2 has 8 bits (which comprise 1 byte) and they are numbered bits 7, 6, 5, 4, 3, 2, 1, 0 from left to right. So 10010100_2 can be written as $1 \times 2^7 + 0 \times 2^5 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$. Note that anything raised to the power of 0 is 1, i.e. $2^0 = 10^0 = 1$. Thus 10010100_2 is equal to $128 + 16 + 4 = 148$. This is how we convert a binary number to its decimal equivalent.
- Note that 148 can also be expressed as a summation of powers of 10, i.e. $128 = 1 \times 10^2 + 2 \times 10^1 + 8 \times 10^0$.
- How about the other way: decimal to binary? This is a bit more complicated. Assuming our decimal number can be expressed as an 8 bit

binary number (a byte), check how many times $2^7 = 128$ divides that number. If it is 0 or 1 we have our first binary bit. If it is 2 or more, the number can't fit into a byte (and we need to consider more bytes). If the division produces a 1 bit, then subtract 128 from the number to get a new remainder. Divide that remainder by $2^6 = 64$, obtaining a 0 or a 1 for our next binary bit. Again subtract 64 from our current number to get the new remainder. Check if $2^5 = 32$ divides the new remainder, obtaining a 0 or a 1 for the 4rd binary bit. Continue this process until $2^0 = 1$ has been processed.

- Consider 148_{10}. We divide by $2^7 = 128$ to obtain a 1 with a remainder of 20. We divide 20 by $2^6 = 64$ to obtain a 0 with the remainder staying at 20. We divide that remainder by $2^5 = 32$ to obtain a 0 with the remainder still staying at 20. We divide the remainder 20 by $2^4 = 16$ to obtain a 1 with a new remainder of 4. We divide the remainder 4 by $2^3 = 8$ to obtain 0 with the remainder staying 4. We divide the remainder 4 by $2^2 = 4$ to obtain 1 with a new remainder of 0. Now, of course, the remaining bits for $2^1 = 2$ and $2^0 = 1$ are both 0. Thus, we obtain

$148_{10} = 10010100_2$.

- MatLab supports a number of different types of signed and unsigned integers, including 8 bit integers (unsigned characters), 16 bit integers (shorts), 32 bit integers and 64 bit integers (longs).

- On the one hand, an unsigned 8 bit integer has values from 0_{10} to 255_{10} (in binary, from 00000000_2 to 11111111_2).

- On the other hand, a signed 8 bit integer has values from -127_{10} to $+128_{10}$. 2's complement is used to represent integers (you can search on the web for details but basically you "flip" the bits and add 1).

- We have been talking only about integers, but what about real numbers (or floating point numbers)? 4.25_{10} is 100.01_2 as it is equal to $1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$.

- Of course, it may not be possible to exactly represent a decimal number as a binary number. What is 3.3_{10} in binary? Well $0 \times 2^2 + 1^1 + 1 \times 2^0$ gives us 011_2 for the 3. What about the .3? There is $0\ 2^{-1} = 0.5$. There

is 1 $2^{-2} = 0.25$ leaving a remainder of 0.05. There are 0 $2^{-3} = 0.125$ and 0 $2^{-4} = 0.0625$ in 0.05. For $2^{-5} = 0.03125$ we get a 1 from division and a remainder of 0.01785. For $2^{-6} = 0.015625$, we get a 1 bit with a remainder of 0.002225. So far, we have .010011. We can continue this calculation infinitely as 0.3 cannot be precisely represented in binary. We still have a very good approximation to 3.3 but its not 100% precise.

- This is called **roundoff** error and can cause seriously problems with large computations. For example, solving a 1000×1000 linear system of equations using Gaussian elimination is generally not possible in practice although it is possible in theory. Roundoff error also occurs when arithmetic is done on two numbers: even if the numbers can be precisely represented, the arithmetic operation on the them may not be able to be precisely represented. Once some roundoff error is present it will propagate through all your calculations potentially growing too large! A branch of Computer Science called Numerical Methods deals with how to reduce roundoff error in algorithms, among many other things. In general, roundoff error probably won't be a problem in this course.

- In MatLab, all numbers by default are **double**. These are 64 bit numbers, where a mantissa occupies 52 bits, 1 bit contains the sign and the remaining 11 bits are for the exponent (a signed integer).
- MatLab also supports **singles**. These are 32 bit numbers where a mantissa contains 23 bits, 1 bit contains the sign and the remaining 8 bits are the exponent (a signed integer). **singles** can also be called **words** (4 bytes).
- Consider the number 52372.453. The **scientific notation** for this number is 5.2372453×10^4. The mantissa (5.2372453) is greater than or equal to 1 and less than or equal to 9. The exponent is the number of rightward shifts needed to convert the original number, 52372.453 to 5.2372453. That is the same as multiplying 5.2372453 by 1×10^4 (usually written as 10^4). Thus 4 is the exponent that the base (10) has to be raised to in the number's representation.
- 5.2372453×10^4 is positive (so its sign bit is 1) and the mantissa is the binary representation of 5.2372453 with the left most bit being 1 while the exponent is the 2's complement of integer 4.

- We can see that both single and double precision floating point numbers has some limitations on how accurately they can represent real numbers! Basically, computers represent number discretely rather than continuously.

- There is also potential problems with doing arithmetic with large and small numbers (even if the numbers can be precisely represented).

- How do you add 2 numbers represented in scientific notation? We have to make the exponents equal by shifting the smaller number to the right (each rightward shift make the exponent 1 bigger) to make the exponents equal. For example, 5.2×10^2 + 5.2×10^4 is computed by shifting 5.2×10^2 to the right twice and then adding the mantissas. So 5.2×10^2 becomes 0.052×10^4 before it is added to 5.2×10^2 and the sum is computed as 0.572×10^4.

- But problems can arise if the smaller mantissa is shifted to the right too much, as it can be significantly truncated or even become 0.

- Consider adding 6×10^{11} to 6×10^1 using a 10 digit representation for

decimal numbers. Then 6×10^1 becomes $0.00000000006 \times 10^{11}$ which is actually $0.0000000000 \times 10^{11}$ using only 10 decimal digits. So 6×10^1 becomes 0. Yikes! Thus $6 \times 10^{11} + 6 \times 10^1 = 6 \times 10^{11}$ in 10 digit arithmetic even though 6×10^1 is not zero. 6×10^1 is a **relative zero** with respect to 6×10^{11} in 10 decimal digit arithmetic.