Math 160 (De Loera) **Final Project**
Due date: June 4th 10:30AM

INSTRUCTIONS

This is the final project of the course.
DUE DATE: Saturday, June 04 at 10:30 am
Solutions need to be submitted via SMARTSITE using the assignment boxes.

- **Submit a single folder or a zip file that contains all code files or pdf docs.**
- **Not possible to use DROPBOX.**
- **Make sure the files run the code from simply calling them, e.g., all data files should be included. We will not download files for you or retype or copy paste your code. If it does not run straight out of the box, you will receive zero points.**
- **Other methods of submission without prior approval will receive zero points.**

Write legibly preferably using word processing if your hand-writing is unclear. Be organized and use the notation appropriately. Show your work on every problem. Correct answers with no support work will not receive full credit.

1. **FINAL PROJECT:**

   Here you are supposed to use some of the methods we discuss to make decisions regarding choice of winners in elections and in rating the value of objects or services.

   For the first part of the project, we have collected the ranking of 5 presidential candidates for more than 200 people. Using the ranking data available at

   `https://www.math.ucdavis.edu/~deloera/TEACHING/MATH160/rankingcandidates.dat`

   write MATLAB code to predict the winner of the presidential election based on the following rank-aggregation (aka voting) methods. You will make a total of 8 predictions:

   - Using *Plurality vote method* (voters top choice is counted for each candidate, winner has the most first-place votes.)
   - Using *Average rank method* (in this case each of n candidates has been given a position from 1 to n by each of the voters, the integers representing the positions are averaged to create a rank-aggregated list. If ties occur, then pick a ranking list that appears most often as the "tie-breaking list". If i, j are in a tie, but in the list we choose i is ahead of j then that is the order.)
   - *Borda count method* (For n candidates each voter awards his or her first choice candidate $n-1$ points, second choice $n-2$ points, and so on with 0 points for person last place. these are borda points. Winner is the candidate with the most total Borda points as awarded by the voters.

- *W-borda count method* (For a given vector $W = (w_1, w_2, \ldots, w_n)$ with $w_1 \geq w_2 \geq \ldots \geq w_n$, each voter awards his or her first choice candidate w_1 points, second choice w_2 points, and so on with w_n points for person last place. These are W-Borda points. Winner is the candidate with the most total W-borda points as awarded by the voters. Try five different vectors W making 5 predictions of the election. Can you choose them to make any of the five candidates win?
- Finally, write an SCIP code that uses the *Yamamoto's discrete model* to make a final ranking using the data.
- Report your predictions and compare the situation. Which is in your opinion the fairest method to count votes?

2. **Using SVD's or a network to decide the ranking of difficulty:** An exam with m question is given to n students of MATH 16000. The instructor collects all grades in an $n \times m$ matrix G, with G_{ij} the grade obtained by student i on question j. The professor would like to assign a *difficulty score or rating* to each question based on the available data, rather than use the subjective perception of students.

   - From the theory of SVD's we know G can be decomposed as a sum of rank-many rank-one matrices. Suppose that G is approximated by a rank-one matrix sq^T with $s \in R^n$ and $q \in R^m$ with non-negative components. Can you use this fact to give a difficulty score or rating? What is the possible meaning of the vector s? Note one can use the top singular value decomposition to get this score vector!
   - There is another way to rank the difficulty of test questions using Networks: Each student gives scores for each problem, then we construct a network whose nodes/vertices are the problems of the exam. There is an arc from problem A to B for student k that did better in problem B than in problem A, (i.e., if s_A, s_B are the scores of those problems, $s_B > s_A$). The weight of the arc y_k associated to student k equals (approximately) the difference of score the student received in those two problems $s_B - s_A = y_k$. Explain why the Massey least square method we saw in class for rating movies can be use for rating exam problems by difficulty.
   - The data available at

     `https://www.math.ucdavis.edu/~deloera/TEACHING/MATH160/examscores.dat`

     has the scores of 31 students in a 7 question exam (each problem was graded on a scale of 0 to 6). Use the data and give a rating of the difficulty of each question in the test using
     - SVD method
     - Massey's network method.