Homework #1

1. Prepare a one paragraph summary of what we talked about in class so far. I do not want just a bulleted list of topics, I want you to use complete sentences and establish context (Why is what we have learned relevant? How does it connect with other classes?). The more insight you give, the better.

2. Familiarize yourself with Python.

   (a) Install (if necessary) the Anaconda distribution of python at https://www.anaconda.com/products/individual.

   (b) Review the python tutorial at https://cs4540-f18.github.io/notes/python-basics.

   (c) If you feel it would be helpful, especially if you are new to python, consider taking the time to go through one (or both) of these more basic python tutorials:

   - https://developers.google.com/edu/python/
   - https://www.learnpython.org/

3. Suppose you are the manager of a factory that produces two products: toilet paper and paper towels. The profit for paper towels is \$10 per ton of pulp, but the profit for toilet paper is \$100 per ton of pulp. Your factory has a supply of 8 000 tons of pulp per month. In addition, you face the following conditions:

   - You have a government contract that requires you to produce 200 tons of paper towels each month.[1]
   - You must ship all of the products that you produce using a fleet of trucks that has a capacity of 100 000 ton-miles (i.e., you could transport 100 000 tons 1 mile, or 1 ton 100 000 miles, or somewhere in between).
   - Paper towels are delivered to a warehouse 10 miles from your factory, while toilet paper is delivered to a warehouse 35 miles from your factory.

   You would like to decide how much of each product to produce to maximize your profit. Write down a mathematical formulation of this as an optimization problem.

4. Suppose that you are investing for your retirement and you are considering allocating your savings over N different investment opportunities. Let x_n denote the fraction of your savings that you invest in asset n. Suppose that you have historical data from which you have estimated that the expected return of asset n is p_n, where p_n is specified as a multiplier, so $p_n = 1.12$ means that asset n has a historical return of 12%. Furthermore, you have also have

[1] Assume that 1 ton of paper towels or 1 ton of toilet paper each require 1 ton of raw pulp.

an estimate of the covariance matrix $\boldsymbol{R}$ for the assets. One can show that this means that the variance of our overall return is given by

$$\sum_{n=1}^{N}\sum_{m=1}^{N} R_{m,n} x_m x_n.$$

Using this as a measurement of the risk of our portfolio, we would like to find the portfolio allocation that has minimum risk over all allocations that achieve an expected return of at least 8%. Write down a mathematical formulation of this as an optimization problem.[2]

5. Download the file student_debt.csv. This file contains the total student loan debt in the United States (measured quarterly, in *trillions*) beginning in 2004 and ending in early 2014.

   (a) Fit a linear regression model to this data in Python using the method of least squares. (For now, you should not yet resort to tools such as scikit-learn. This is simple enough that you should be able to calculate the slope and intercept directly.) Submit a plot showing both the data and your least squares fit.

   (b) Use your regression model to predict the student debt in the US in 2020. How does this compare with actual student debt?

   (c) Use your model to predict the student debt in the US in 2041 (an estimate of when my youngest son could conceivably graduate from college). Comment on the degree of confidence you have in this prediction.

6. Gordon Moore, co-founder of Intel, predicted in 1965 that the number of transistors on an integrated circuit would double approximately every two years. This conjecture has turned out to be surprisingly accurate over the years – so accurate that it is now known as Moore's *law*. Let's see how well this "law" compares to some real data.

   (a) Go to https://en.wikipedia.org/wiki/Transistor_count. This page contains a large table of microprocessors with both the MOS transistor count and the date of introduction of the processor. Figure out a way to load this information into Python. (You can do this by hand, but I would instead suggest that you do some quick online searches to find examples of people scraping wikipedia tables into Python – there are numerous examples of this if you poke around, and knowing how to do this automatically is a useful skill.)

   (b) Moore's law dictates that there is an *exponential* relationship between time and the transistor count. Propose a transformation of the data so that the relationship should be (approximately) linear.

   (c) Fit a linear regression model to this data in Python using the method of least squares. Submit a plot showing both the data and your least squares fit. Comment on your results and how they compare with Moore's conjecture.

7. In January of 1801, the Italian astronomer Joseph Piazzi discovered a planetoid, which he named Ceres, which he was able to observe for 41 days before the planetoid was lost in the

[2] If you had done this in the 1950's, you would have won a Nobel Prize.

sun, so that astronomers could no longer find the planet. From a limited set of only three observations, Carl Frederich Gauss was able to determine the orbit of the planetoid using the (yet unpublished) method of least squares (together with a lot of ingenuity!) Gauss's prediction differed substantially from that of other mathematicians, and led to Ceres being rediscovered later that year in a position very near to that predicted by Gauss.

Gauss had to solve this problem using measurements of ascension and declination as observed from Earth, which significantly complicates the problem. Here we consider a simplified version of the problem that uses Cartesian coordinates. Specifically, suppose that we have a sequence of noisy observations of the coordinates $(x_1, y_1), \ldots, (x_N, y_N)$ of a planetoid. From Kepler's laws of planetary motion, we know that these coordinates should lie near an ellipsoid, which in standard form can be expressed as

$$\left(\frac{x_n}{\nu_x}\right)^2 + \left(\frac{y_n}{\nu_y}\right)^2 \approx 1,$$

where ν_x and ν_y are tunable parameters that determine the shape of the ellipse.

(a) Re-express the inference problem of determining the parameters of the ellipse as a linear system we would like to solve of the form

$$\boldsymbol{z} = \boldsymbol{A}\boldsymbol{w},$$

where $\boldsymbol{z}$ and $\boldsymbol{A}$ capture what we know about the problem from observations, and $\boldsymbol{w}$ captures the unknown parameters we want to estimate from our data. [Hint: $\boldsymbol{A}$ should be an $N \times 2$ matrix.]

(b) Download the file asteroid_data.csv. This contains some simulated noisy observations. Use the formulation derived above to form a least squares estimate of the ellipse. Submit a plot showing both the data and the ellipse you estimated using least squares. To solve this problem, you may use the fact that the $\boldsymbol{w}$ that minimizes $\|\boldsymbol{z} - \boldsymbol{A}\boldsymbol{w}\|_2^2$ is the solution to the (square) system

$$(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{w} = \boldsymbol{A}^T\boldsymbol{z}.$$

8. Compute the first and second derivatives of the following functions (remember the product and chain rules).

(a) $f(x) = ax^2 + bx + c$, where a, b, c are constants.

(b) $f(x) = -\cos(2\pi x^2) + x^2$.

(c) $f(x) = \sum_{m=1}^{M} \log(1 + e^{-a_m x})$, where $a_1, \ldots, a_M$ are constants.

9. Compute the gradient and Hessian matrix of the following functions. Note that $\boldsymbol{x}$ is a vector in $\mathbb{R}^N$ in all the problems below.

(a) $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c$, where $\boldsymbol{A}$ is an $N \times N$ symmetric matrix (i.e., $\boldsymbol{A} = \boldsymbol{A}^T$),$\boldsymbol{b}$ is an $N \times 1$ vector, and c is a scalar.

(b) $f(\boldsymbol{x}) = -\cos(2\pi\boldsymbol{x}^T\boldsymbol{x}) + \boldsymbol{x}^T\boldsymbol{x}$.

(c) $f(\boldsymbol{x}) = \sum_{m=1}^{M} \log(1 + e^{-\boldsymbol{a}_m^T\boldsymbol{x}})$, where $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M$ are $N \times 1$ vectors.