# DSC 40A - Homework 1

Due: Friday, January 14, 2022 at 11:59pm

Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Homeworks are due to Gradescope by 11:59pm on the due date. You can use a slip day to extend the deadline by 24 hours.

Homework will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should **always explain and justify** your conclusions, using sound reasoning. Your goal should be to convince the reader of your assertions. If a question does not require explanation, it will be explicitly stated.

Homeworks should be written up and turned in by each student individually. You may talk to other students in the class about the problems and discuss solution strategies, but you should not share any written communication and you should not check answers with classmates. You can tell someone how to do a homework problem, but you cannot show them how to do it.

For each problem you submit, you should **cite your sources** by including a list of names of other students with whom you discussed the problem. Instructors do not need to be cited.

This homework will be graded out of 50 points. The point value of each problem or sub-problem is indicated by the number of avocados shown.

**Note:** For Problem 5, parts (a), (c), (d), and (e), you'll need to code your answers in Python. We've provided some starter code, linked here. You'll need to turn in your completed Python file to Gradescope separately from the rest of this homework, in a file called `hw1code.py` . We'll grade your code using an autograder, so explanations are not necessary for Problem 5, parts (a), (c), (d), and (e),

### Problem 1. Linear Functions

Consider the linear function $f(x) = 4x - 7$.

**a)** 🥑 If $a \leq b$, show that $f(a) \leq f(b)$.

**b)** 🥑🥑 Both of the statements below are true, but only one is a consequence of the property you proved in part (a). Which is it? Show that this statement is true, using the result of part (a).

1. $\text{Mean}(f(x_1), \ldots, f(x_n)) = f(\text{Mean}(x_1, \ldots, x_n))$
2. $\text{Median}(f(x_1), \ldots, f(x_n)) = f(\text{Median}(x_1, \ldots, x_n))$

**c)** 🥑🥑 Now, prove the other statement.

**d)** 🥑🥑🥑🥑🥑🥑 Suppose we consider a different linear function $g(x) = -3x + 2$. Prove or find a counterexample to disprove each of the following:

1. If $a \leq b$, then $g(a) \leq g(b)$.
2. $\text{Mean}(g(x_1), \ldots, g(x_n)) = g(\text{Mean}(x_1, \ldots, x_n))$
3. $\text{Median}(g(x_1), \ldots, g(x_n)) = g(\text{Median}(x_1, \ldots, x_n))$
4. $\text{Mode}(g(x_1), \ldots, g(x_n)) = g(\text{Mode}(x_1, \ldots, x_n))$

**Problem 2. Variations of the Mean Squared Error**

Suppose you have a data set $y_1, y_2, \ldots, y_n$ with at least three values, $n \geq 3$, and the values are arranged such that $y_1 \leq y_2 \leq \cdots \leq y_n$.

We know from class that the mean of the data minimizes mean squared error,

$$R_{sq}(h) = \sum_{i=1}^{n} (h - y_i)^2.$$

In this problem, we'll consider some variations of this risk function.

**a)** 🥑🥑 Define a new function that considers only the two extreme points:

$$E(h) = (h - y_1)^2 + (h - y_n)^2.$$

What value of $h$ minimizes $E(h)$? We'll call the value of $h$ that minimizes $E(h)$ the **extreme mean**, since it's based on the extreme data values.

**b)** 🥑🥑🥑🥑 Define a new function that weights larger data points less heavily:

$$S(h) = \left( \sum_{i=1}^{n-2} (h - y_i)^2 \right) + 0.5 \cdot (h - y_{n-1})^2 + 0.1 \cdot (h - y_n)^2.$$

What value of $h$ minimizes $S(h)$? We'll call the value of $h$ that minimizes $S(h)$ the **sloped mean**, since the coefficients of the data values decrease for larger data.

**c)** 🥑🥑 Which do you think is a better hypothesis, the mean or the sloped mean? Is your answer always the same, or does it depend on some property of the data set? Give an example of when you might prefer to use the sloped mean, and when you might prefer the (regular) mean.

**d)** 🥑🥑🥑 Find a function $P(h)$, a variant of the mean squared error $R_{sq}(h)$, such that $P(h)$ is minimized at

$$h = \frac{0.7 \cdot y_1 + 0.8 \cdot y_2 + \sum_{i=3}^{n} y_i}{n - 0.5}.$$

**Hint:** Look closely at the work you did in part (b).

**Problem 3. Constructing Examples**

Given a data set $y_1 \leq y_2 \leq \cdots \leq y_n$, define the mean absolute error and mean squared error functions as usual,

$$R_{abs}(h) = \sum_{i=1}^{n} |h - y_i|,$$

$$R_{sq}(h) = \sum_{i=1}^{n} (h - y_i)^2.$$

**a)** 🥑🥑 Give an example of a data set of size $n = 4$ for which $R_{abs}(h) > R_{sq}(h)$ whenever $y_1 \leq h \leq y_4$.

**b)** 🥑🥑 Give an example of a data set of size $n = 4$ for which $R_{abs}(h) < R_{sq}(h)$ whenever $y_1 \leq h \leq y_4$.

**c)** 🥑🥑 Give an example of a data set of size $n = 4$ for which $R_{abs}(h) = R_{sq}(h)$ whenever $y_1 \leq h \leq y_4$.

**d)** 🥑🥑 Give an example of a data set of size $n = 4$ for which $R_{abs}(h) > R_{sq}(h)$ for at least one value of $h$ with $y_1 \leq h \leq y_4$, and $R_{abs}(h) < R_{sq}(h)$ for some other value of $h$ with $y_1 \leq h \leq y_4$.

### Problem 4. Max's Idea

In our lecture, we argued that one way to make a good prediction $h$ is to minimize the mean absolute error:

$$R(h) = \frac{1}{n} \sum_{i=1}^{n} |h - x_i|.$$

We saw that the median of $y_1, \ldots, y_n$ is the prediction with the smallest mean error. Your friend Max thinks that instead of minimizing the mean error, it is better to maximize the following quantity:

$$M(h) = \prod_{i=1}^{n} e^{-|h-x_i|}.$$

The above formula is written using product notation, which is similar to summation notation, except terms are multiplied and not added. For example,

$$\prod_{i=1}^{n} a_i = a_1 \cdot a_2 \cdot \ldots \cdot a_n.$$

Max's reasoning is that for some models, $e^{-|h-x_i|}$ is used to compute how likely prediction $h$ will appear given the observation $x_i$ – hence it is called "likelihood." Then, we should attempt to maximize the chance of getting the prediction $h$, given the set of observations. In this problem, we'll see if Max has a good idea.

**a)** 🥑🥑 For an arbitrary fixed value of $x_i$, sketch a graph of the basic shape of the likelihood function $L(h) = e^{-|h-x_i|}$. Explain, based on the graph, why larger values of $L(h)$ correspond to better predictions $h$.

**b)** 🥑🥑🥑🥑 Informally, a minimizer of a function $f$ is an input $x_{\min}$ where $f$ achieves its minimum value. More formally, $x_{\min}$ is a minimizer of $f$ if $f(x_{\min}) \leq f(x)$ for all values of $x$. In the same way, $x_{\max}$ is a maximizer of $f$ if $f(x_{\max}) \geq f(x)$ for all values of $x$.

Suppose that $f$ is some unknown function which takes in a real number and outputs a real number. Suppose that $c$ is an unknown positive constant, and define the function $g(x) = e^{-c \cdot f(x)}$. Prove that if $x_{\min}$ is a minimizer of $f$, then it is also a maximizer of $g$.

**Hint:** Use the definitions of maximizer and minimizer given here, plus the basic properties of inequalities from Groupwork 1.

**c)** 🥑🥑 At what value $h^*$ is $M(h)$ maximized? Did Max have a reasonable idea?

### Problem 5. Side Hustle

**Note:** For parts (a), (c), (d), and (e) of this problem, you'll need to code your answers in Python. We've provided some starter code, linked here. You'll need to turn in your completed Python file to Gradescope separately from the rest of this homework, in a file called `hw1code.py` . We'll grade your code using an autograder, so explanations are not necessary for parts (a), (c), (d), and (e) of this question.

Suppose your backyard avocado tree produces more avocados than you can eat, so you decide to sell some to help offset the rising costs of UCSD tuition. Instead of selling each avocado for a fixed price, like most grocery stores do, you prefer to sell them per pound.

**a)** 🥑🥑🥑 Suppose you want to keep track of the mean avocado weight of all your sales, but your scale does not record or remember the weights of previous measurements. Instead of writing down the weight of each sale and re-calculating the mean with each transaction, you want to come up with a way to only keep track of the mean weight.

Complete the function `avo_mean` which should calculate the mean weight of all avocado sales based on three parameters:

- `prev_mean`, the mean weight of all avocado sales so far,
- `n`, the number of avocado sales so far, and
- `weight`, the weight of the very next avocado sale.

The function should return the mean weight of all `n+1` avocado sales.

**b)** 🥑 Suppose that so far, you've sold avocados to 14 customers, and the mean avocado weight for these 14 sales is 1.24 pounds. How much would your next sale have to weigh in order to bring the mean weight up to 1.5 pounds?

**c)** 🥑🥑 We've shown that it's possible to update the mean after each sale without keeping track of all the individual weights of each sale. Show that we cannot do the same for the median, by giving an example of two different lists of avocado weights such that:

- both lists are of the same length,
- both lists have the same last number,
- both lists have the same median value when we exclude the last number, and
- both lists have different median values when we include the last number.

This shows that the new median weight cannot be determined by the value of the previous median weight, the number of transactions, and the new weight alone. Save your lists as `list1c` and `list2c`.

**d)** 🥑🥑 Suppose now that instead of determining the mean weight of all purchases, you want to find the **extreme mean** weight of all sales, where the extreme mean is how we defined it in Problem 2(a). Again, we don't want to have to keep track of every sale's weight in order to do this.

Unfortunately, knowledge of the previous extreme mean weight, number of transactions, and the weight of the next sale are insufficient to determine the extreme mean. Show that this is the case by providing an example of two lists of avocado weights such that:

- both lists are of the same length,
- both lists have the same last number,
- both lists have the same extreme mean when we exclude the last number, and
- both lists have different extreme means when we include the last number.

Save your lists as `list1d` and `list2d`.

**e)** 🥑🥑🥑🥑 It turns out that if we keep track of just one additional piece of information, the maximum weight of all sales, then we can calculate the extreme mean without recording all individual sale weights!

Complete the function `avo_extreme_mean` which should calculate the extreme mean weight of all avocado sales based on four parameters:

- `prev_extreme_mean`, the extreme mean weight of all avocado sales so far,
- `prev_max`, the largest weight of all avocado sales so far,

- `n`, the number of avocado sales so far, and
- `weight`, the weight of the very next avocado sale.

The function should return a **list** of two elements, the first being the extreme mean weight of all `n+1` sales, the second being the maximum weight of all `n+1` sales.