Purpose:

- Review essentials of probability
- Review theory of linear regression
- Use Python to solve a linear regression problem

Directions: This homework is to be done individually. Please upload a set of solutions containing your name and `@ucsc.edu` e-mail address to Canvas. Typeset (e.g. TeX) solutions are preferred, but scans or photographs of hand-written solutions are acceptable *provided that they are neat and legible.* The TA may deduct points for poorly organized or illegible solutions.

Question:	1	2	3	4	Total
Points:	23	30	35	12	100
Bonus Points:	0	0	0	6	6
Score:					

Questions: (ordered roughly by increasing difficulty)

1. **Conditional Probability Review:** There are 52 cards in a standard deck (excluding Jokers), with 13 cards per suit. The suits are Hearts, Spades, Diamonds, and Clubs, where Hearts and Diamonds are *red* and Spades and Clubs are *black.* Each suit contains 3 *face* cards (Jack, Queen, King), and 10 additional cards (Ace, Two, ..., Ten). Suppose we remove both *red* Kings and draw a card uniformly at random from the rest of the deck. *Give your answer as a fraction. Correct answers need not show work.*
   (a) (3 points) What is the probability that the card you draw is a *face* card? $P(F)$
   (b) (3 points) What is the probability that the card you draw is *black*? $P(B)$
   (c) (4 points) What is the probability that the card you draw is a *black face* card? $P(BF)$
   (d) (4 points) Given that the card you draw is black, what is the probability that it is a *face* card? $P(F|B)$
   (e) (4 points) Given that the card you draw is a *face* card, what is the probability it is *black*? $P(B|F)$
   (f) (*Ungraded*) *Verify that the five probabilities just calculated are consistent with the definition of conditional probability and Bayes's Rule* (i.e. $P(BF) = P(F|B)P(B) = P(B|F)P(F)$)
   (g) (5 points) Given that the card you draw is *not* a Two, what is the probability it is a Heart?

2. **Linear Regression:**
   For this question, we will consider artificial data. Let us first start with the equation

$$X\theta' = \mathbf{z}'$$

   where we fix X and give an arbitary vector θ from which to determine the value of $\mathbf{y}$'.

$$X = \begin{bmatrix} 1 & 3 & 9 & 2 \\ 1 & 6 & 9 & 1 \\ 1 & 7 & 7 & 7 \\ 1 & 8 & 6 & 4 \\ 1 & 1 & 0 & 8 \end{bmatrix} \quad ; \quad \theta' = \begin{bmatrix} 3 \\ 0 \\ 2 \\ -1 \end{bmatrix} \quad \Longrightarrow \quad \mathbf{z}' = \begin{bmatrix} 19 \\ 20 \\ 10 \\ 11 \\ -5 \end{bmatrix}$$

To $\mathbf{z}'$, we add a small amount of noise $\mathbf{v}$ (which is *not* necessarily orthogonal to the $X\theta$ hyperplane!) and consider the resultant vector

$$\mathbf{y} = \mathbf{z}' + \mathbf{v} = \begin{bmatrix} 19 \\ 19 \\ 10 \\ 11 \\ -3 \end{bmatrix}$$

We now consider the linear regression problem of finding θ in

$$\mathbf{y}' := X\theta$$

such that the mean squared error between the components of and $\mathbf{y}'$ and $\mathbf{y}$ is minimized. For the purpose of a checking the reasonableness of our answer, we have generated our data in this way so θ and θ' will be close(*ish*).

We recommend using Python for this problem.

```
data =
[[3, 9, 2, 19],
 [6, 9, 1, 19],
 [7, 7, 7, 10],
 [8, 6, 4, 11],
 [1, 0, 8, -3]]
```

Our *training* data is a list of examples (or *instances*), where each example has been written on its own line in a row of four values. The first three values of each row are *features* of the data (corresponding to the values of X without the column of ones). The last entry in each line is the *label* (or *target*) of the instance and is the corresponding component of $\mathbf{y}$ (*not* $\mathbf{y}'$).

Our training data consists of 5 instances. We may label the first three features of the data with the variables x_1, x_2, and x_3 (considering $x_0 = 1$ for the omitted column of X). The first instance therefore has features $x_1 = 3, x_2 = 9, x_3 = 2$ and target $y = 19$.

(a) (3 points) Tell us which machine learning method (or library, such as Scikit-learn in Python, which is fine) you will use to solve this linear regression problem (this is your preference).

(b) (11 points) Run a linear regression algorithm on the full training set (other than using the closed-form least square solution). The input features should be a 4-dimension vector $\mathbf{x} = (x_0, x_1, x_2, x_3)$, where $x_0 = 1$ is a constant, and x_1, x_2, x_3 correspond to the first, second, and third column in data. Report the model and "root mean squared error". The root mean squared error is defined as $RMSE = \sqrt{\sum_i (f(\mathbf{x}^{(i)}) - y^{(i)})^2/N}$, where $f(\mathbf{x}^{(i)})$ is the prediction of instance-i, and N is the number of samples. Note in some tools, such as scikit-learn, the constant feature x_0 is added by default thus the input of your feature should simply be (x_1, x_2, x_3).

(c) (8 points) Suppose you had an *unlabeled* instance $\mathbf{x} = [3, 3, 5]$. What prediction for the label would the model from part (b) give?

(d) (8 points) If the examples are re-ordered (so the rows of X and elements of $\mathbf{y}$ are permuted), what happens to the learned θ vector and why?

3. **More Probability Review:** Assume that the probability of obtaining heads when tossing a coin is λ.

(a) (12 points) What is the probability of obtaining the first head at the (k + 1)-th toss?

(b) (13 points) What is the expected number of tosses needed to get the first head?

(c) (10 points) What is the expected number of heads when tossing N times?

4. **A Continuous Variable plus Bayes's Rule:** Suppose it will rain $g(x) = \cot(x\pi/2)$ cm tomorrow, where $x \in (0, 1]$ is some unknown parameter. In this part of the world, $(1-x)$ is precisely the probability of hearing thunder before sunset. This morning you assigned probability density $(n+1)x^n$ to each value of $x \in (0, 1]$ (your prior belief) where $n \geq 1$.

   (a) (12 points) What is your expected value for x at midday?

   (b) (6 points (bonus)) You hear no thunder before sunset. What probability density do you now assign to each value of x (your posterior), following Bayes's Rule? *Give your answer as a function $f(x)$ depending on n. Hint: constant factors may be ignored until the end, when we only need to ensure $\int_0^1 f(x)dx = 1$.*