Homework #0

Due: January 26, 2023 at 11:59 PM
(with free, no-questions-asked
extension, see below)

Welcome to CS181! The purpose of this assignment is to help assess your readiness for this course. **This assignment will be graded for completion and effort.** If you encounter any difficulty with these problems, fear not! We will have sections in the first week of class reviewing the math, statistics, and coding pre-requisites for this course. TFs will also directly discuss relevant problems from this HW0 in these sections. You are, of course, more than welcome to swing by office hours and post questions on Ed.

1. Please type your solutions after the corresponding problems using this LaTeX template, and start each problem on a new page.
2. Please submit the **writeup PDF to the Gradescope assignment 'HW0'**. Remember to assign pages for each question.
3. Please submit your **LaTeX file and code files (i.e., anything ending in `.py`, `.ipynb`, or `.tex`) to the Gradescope assignment 'HW0 - Supplemental'**.

Free, No-Questions-Asked Extension:

- The official deadline is January 26, in order to not overlap too much with HW1, which covers substantial new course material. With teaching staff discussing these problems in Week 1 sections, a reasonably prepared (in terms of pre-requisites) student should be able to complete this assignment quite comfortably in a week.
- However, given the uncertainty and frenzy of the start of the new semester, we will also give free, no-questions-asked extensions to **February 2, 2023 at 11:59 PM** if you need some extra breathing room and / or if you need to brush up a little more on the pre-requisites. Do note, however, that HW1 will be due a week later on February 9.
- You do not need to formally request this free, no-questions-asked extension (i.e., no Ed post or email is necessary). Please simply submit your assignment before the late deadline of **February 2, 2023 at 11:59 PM** in Gradescope. Given that this is a free extension, it does not count into your late days for the semester.

Problem 1 (Modeling Linear Trends - Linear Algebra Review)

In this class we will be exploring the question of "how do we model the trend in a dataset" under different guises. In this problem, we will explore the algebra of modeling a linear trend in data. We call the process of finding a model that capture the trend in the data, "fitting the model."

Learning Goals: In this problem, you will practice translating machine learning goals ("modeling trends in data") into mathematical formalism using linear algebra. You will explore how the right mathematical formalization can help us express our modeling ideas unambiguously and provide ways for us to analyze different pathways to meeting our machine learning goals.

Let's consider a dataset consisting of two points $\mathcal{D} = \{(x_1, y_1), (x_2, y_2)\}$, where x_n, y_n are scalars for $n = 1, 2$. Recall that the equation of a line in 2-dimensions can be written: $y = w_0 + w_1 x$.

1. Write a system of linear equations determining the coefficients w_0, w_1 of the line passing through the points in our dataset $\mathcal{D}$ and analytically solve for w_0, w_1 by solving this system of linear equations (i.e., using substitution). Please show your work.
2. Write the above system of linear equations in matrix notation, so that you have a matrix equation of the form $\mathbf{y} = \mathbf{X}\mathbf{w}$, where $\mathbf{y}, \mathbf{w} \in \mathbb{R}^2$ and $\mathbf{X} \in \mathbb{R}^{2\times 2}$. For full credit, it suffices to write out what $\mathbf{X}$, $\mathbf{y}$, and $\mathbf{w}$ should look like in terms of x_1, x_2, y_1, y_2, w_0, w_1, and any other necessary constants. Please show your reasoning and supporting intermediate steps.
3. Using properties of matrices, characterize exactly when an unique solution for $\mathbf{w} = (w_0\ w_1)^T$ exists. In other words, what must be true about your dataset in order for there to be a unique solution for $\mathbf{w}$? When the solution for $\mathbf{w}$ exists (and is unique), write out, as a matrix expression, its analytical form (i.e., write $\mathbf{w}$ in terms of $\mathbf{X}$ and $\mathbf{y}$).

   Hint: What special property must our $\mathbf{X}$ matrix possess? What must be true about our data points in $\mathcal{D}$ for this special property to hold?
4. Compute $\mathbf{w}$ by hand via your matrix expression in (3) and compare it with your solution in (1). Do your final answers match? What is one advantage for phrasing the problem of fitting the model in terms of matrix notation?
5. In real-life, we often work with datasets that consist of hundreds, if not millions, of points. In such cases, does our analytical expression for $\mathbf{w}$ that we derived in (3) apply immediately to the case when $\mathcal{D}$ consists of more than two points? Why or why not?

1: Modeling Linear Trends - Linear Algebra Review

1.1 Here is a system of linear equations determining the coefficients w_0, w_1 of the line passing through the points in our dataset $\mathcal{D}$:

$$y_1 = w_0 + w_1 x_1 \tag{1}$$

$$y_2 = w_0 + w_1 x_2 \tag{2}$$

Solving this system we get:

$$\begin{aligned}
y_1 - y_2 &= w_1(x_1 - x_2) && \text{(1)-(2)}\\
\Rightarrow w_1 &= \frac{y_1 - y_2}{x_1 - x_2}\\
y_1 &= w_0 + \frac{y_1 - y_2}{x_1 - x_2} x_1 && \text{sub into (1)}\\
\Rightarrow w_0 &= y_1 - \frac{y_1 x_1 - y_2 x_1}{x_1 - x_2} = \frac{y_1 x_1 - y_1 x_2 - y_1 x_1 - y_2 x_1}{x_1 - x_2} = \frac{y_1 x_2 + y_2 x_1}{x_2 - x_1}
\end{aligned}$$

1.2 In matrix notation, we can write the system of linear equations as $\mathbf{y} = \mathbf{X}\mathbf{w}$ as

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

This works because when we carry out the matrix multiplication, we recover the two separate linear equations. For example, the first step gives us exactly that the first element of the resulting vector $\mathbf{y}$ will be $1w_0 + x_1 w_1 = w_0 + w_1 x_1$, as desired.

1.3 A unique solution for $\mathbf{w}$ only exists when the inverse of $\mathbf{X}$ exists. The inverse of a matrix only exists when it has full rank (no linear dependencies), so our 2-point dataset must not have $x_1 = x_2$. We also can't have any 'missing' data points: we must have (x, y) pairs, no singlets.

We write an analytical expression for the coefficients as such:

$$\mathbf{w} = \mathbf{X}^{-1}\mathbf{y}$$

1.4 We solve for $\mathbf{w}$ by hand, which requires computing the inverse of $\mathbf{X}$.
We calculate $\mathbf{X}^{-1}$ as follows:

$$\begin{aligned}
\begin{bmatrix} {[cc|cc]}1 & x_1 & 1 & 0 \\ 1 & x_2 & 0 & 1 \end{bmatrix} &= \begin{bmatrix} {[cc|cc]}1 & x_1 & 1 & 0 \\ 0 & x_2 - x_1 & -1 & 1 \end{bmatrix} && \text{subtract: (2)} \leftarrow \text{(2)-(1)}\\
&= \begin{bmatrix} {[cc|cc]}1 & x_1 & 1 & 0 \\ 0 & 1 & -\frac{1}{x_2 - x_1} & \frac{1}{x_2 - x_1} \end{bmatrix} && \text{divide (2)}\\
&= \begin{bmatrix} {[cc|cc]}1 & 0 & 1 + \frac{x_1}{x_2 - x_1} & -\frac{x_1}{x_2 - x_1} \\ 0 & 1 & -\frac{1}{x_2 - x_1} & \frac{1}{x_2 - x_1} \end{bmatrix} && \text{(1)} \leftarrow \text{(1) - } x_1\text{*(2)}\\
&= \begin{bmatrix} {[cc|cc]}1 & 0 & \frac{x_2}{x_2 - x_1} & -\frac{x_1}{x_2 - x_1} \\ 0 & 1 & -\frac{1}{x_2 - x_1} & \frac{1}{x_2 - x_1} \end{bmatrix}
\end{aligned}$$

This gives us $\mathbf{X}^{-1} = \begin{bmatrix} \frac{x_2}{x_2 - x_1} & -\frac{x_1}{x_2 - x_1} \\ -\frac{1}{x_2 - x_1} & \frac{1}{x_2 - x_1} \end{bmatrix}$

And so

$$\begin{aligned}
\mathbf{w} &= \mathbf{X}^{-1}\mathbf{y} \\
&= \begin{bmatrix} \frac{x_2}{x_2-x_1} & -\frac{x_1}{x_2-x_1} \\ -\frac{1}{x_2-x_1} & \frac{1}{x_2-x_1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\
&= \begin{bmatrix} \frac{y_1 x_2}{x_2-x_1} - \frac{y_2 x_1}{x_2-x_1} \\ -\frac{y_1}{x_2-x_1} + \frac{y_2}{x_2-x_1} \end{bmatrix} \\
&= \begin{bmatrix} \frac{y_1 x_2 - y_2 x_1}{x_2-x_1} \\ \frac{y_2-y_1}{x_2-x_1} \end{bmatrix}
\end{aligned}$$

As expected, using matrix notation gives the same result for $\mathbf{w}$ as solving the system one coefficient at a time. One advantage of using matrix notation is that the operations are vectorized: we can compute all of the coefficients simultaneously. This is much more efficient for large datasets.

1.5 Yes, the analytical expression that we derived in (1.3) immediately applies to the case when our dataset consists of more than two points: our matrices and vectors will change size, but the algebraic operations will remain the same (assuming that the conditions for a unique solution to exist are met). This is assuming that the number of coefficients has also increased to match the number of data points you have: the w vector needs to have the same length as the y vector for a unique solution to exist.

Problem 2 (Optimizing Objectives - Calculus Review)

In this class, we will write real-life goals we want our model to achieve into a mathematical expression and then find the optimal settings of the model that achieves these goals. The formal framework we will employ is that of mathematical optimization. Although the mathematics of optimization can be quite complex and deep, we have all encountered basic optimization problems in our first calculus class!

Learning Goals: In this problem, we will explore how to formalize real-life goals as mathematical optimization problems. We will also investigate under what conditions these optimization problems have solutions.

In her most recent work-from-home shopping spree, Nari decided to buy several house plants. *Her goal is to make them to grow as tall as possible.* After perusing the internet, Nari learns that the height y in mm of her Weeping Fig plant can be directly modeled as a function of the oz of water x she gives it each week:

$$y = -3x^2 + 72x + 70.$$

1. Based on the above formula, is Nari's goal achievable: does the plant have a maximum height? Why or why not? Does her goal have a unique solution - i.e. is there one special watering schedule that would acheive the maximum height (if it exists)?

   Hint: plot this function. In your solution, words like "convex" and "concave" may be helpful.

2. Using calculus, find how many oz per week should Nari water her plant in order to maximize its height. With this much water, how tall will her plant grow?

   Hint: solve analytically for the critical points of the height function (i.e., where the derivative of the function is zero). For each critical point, use the second-derivative test to identify if each point is a max or min point, and use arguments about the global structure (e.g., concavity or convexity) of the function to argue whether this is a local or global optimum.

Now suppose that Nari want to optimize both the amount of water x_1 (in oz) *and* the amount of direct sunlight x_2 (in hours) to provide for her plants. After extensive research, she decided that the height y (in mm) of her plants can be modeled as a two variable function:

$$y = f(x_1, x_2) = \exp\left(-(x_1 - 2)^2 - (x_2 - 1)^2\right)$$

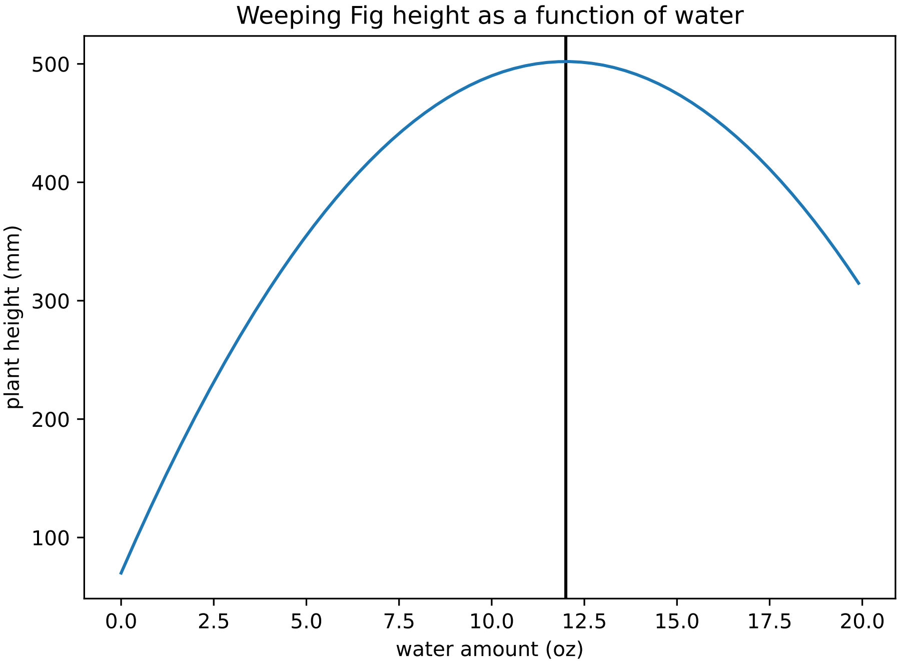
3. Using `matplotlib`, visualize in 3D the height function as a function of x_1 and x_2 using the `plot_surface` utility for $(x_1, x_2) \in (0, 6) \times (0, 6)$. Use this visualization to argue why there exists a unique solution to Nari's optimization problem on the specified intervals for x_1 and x_2.

   Remark: in this class, we will learn about under what conditions do *multivariate* optimization problems have unique global optima (and no, the second derivative test doesn't exactly generalize directly). Looking at the visualization you produced and the expression for $f(x_1, x_2)$, do you have any ideas for why this problem is guaranteed to have a global maxima? You do not need to write anything responding to this – this is simply food for thought and a preview for the semester.

2: Optimizing Objectives - Calculus Review

2.1 The function governing the height of the Weeping Fig (namely $y = -3x^2 + 72x + 70$) is concave: we see this because the highest degree of this polynomial is 2, and the coefficient in front is negative. With a highest degree of 2, this polynomial will have a single unique solution. This means that the height function will have a single global maximum.

We also see this global maximum in the plot of the growth function below (I added a vertical line at 12 oz):



2.2 We want to determine whether the Weeping Fig plant will have a maximum height using calculus. We will check this by computing the first and second derivatives.

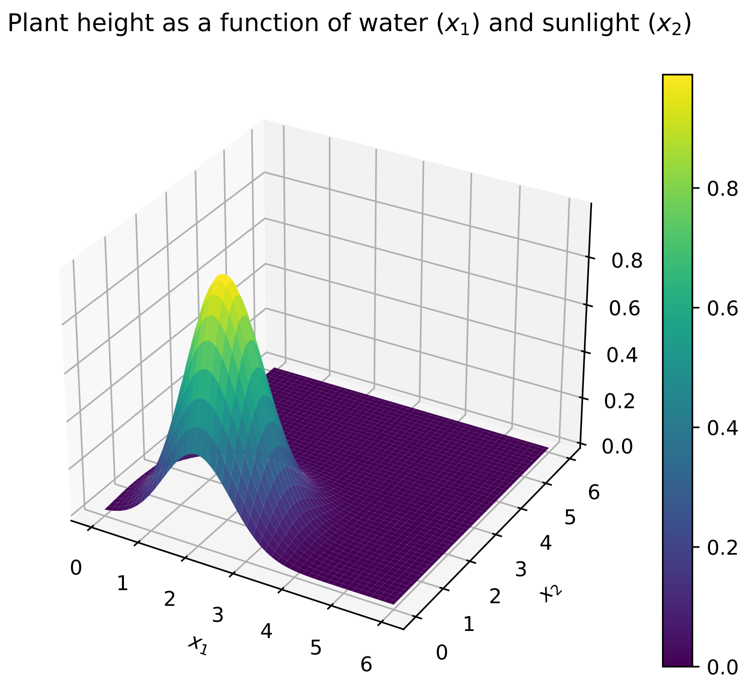
First derivative tells us where the critical points are: $y' = -6x + 72 = 0 \Rightarrow x = 12$ oz.

Second derivative tells us about the critical points: $y'' = -6$. Since the second derivative is always negative across the domain (regardless of the value of x), we see that the critical point at 12 oz of water represents a global maximum. This means that **Nari should give her Weeping Fig 12 oz of water each week** for maximal growth to a height of $-3 * 12^2 + 72 * 12 + 70 =$ **502 mm**.

Now we're optimizing both water (x_1) and sunlight hours (x_2) and have this equation:

$$y = f(x_1, x_2) = exp(-(x_1 - 2)^2 - (x_2 - 1)^2)$$

2.3 From the 3D plot of plant height as a function of sunlight and water over the domain specified, it appears that there is a single unique global maximum (approximately near 1 oz and 2 hrs of sunlight).


Plant height as a function of water (x_1) and sunlight (x_2)
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4
5
6
x_1
0
1
2
3
4
5
6
x_2
0.0
0.2
0.4
0.6
0.8

Problem 3 (Reasoning about Randomness - Probability and Statistics Review)

In this class, one of our main focuses is to model the unexpected variations in real-life phenomena using the formalism of random variables. In this problem, we will use random variables to model how much time it takes an USPS package processing system to process packages that arrive in a day.

Learning Goals: In this problem, you will analyze random variables and their distributions both analytically and computationally. You will also practice drawing connections between said analytical and computational conclusions.

Consider the following model for packages arriving at the US Postal Service (USPS):

- Packages arrive randomly in any given hour according to a Poisson distribution. That is, the number of packages in a given hour N is distributed $Pois(\lambda)$, with $\lambda = 3$.
- Each package has a random size S (measured in in^3) and weight W (measured in pounds), with joint distribution

$$(S, W)^T \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right), \text{ with } \boldsymbol{\mu} = \begin{bmatrix} 120 \\ 4 \end{bmatrix} \text{ and } \boldsymbol{\Sigma} = \begin{bmatrix} 1.5 & 1 \\ 1 & 1.5 \end{bmatrix}.$$

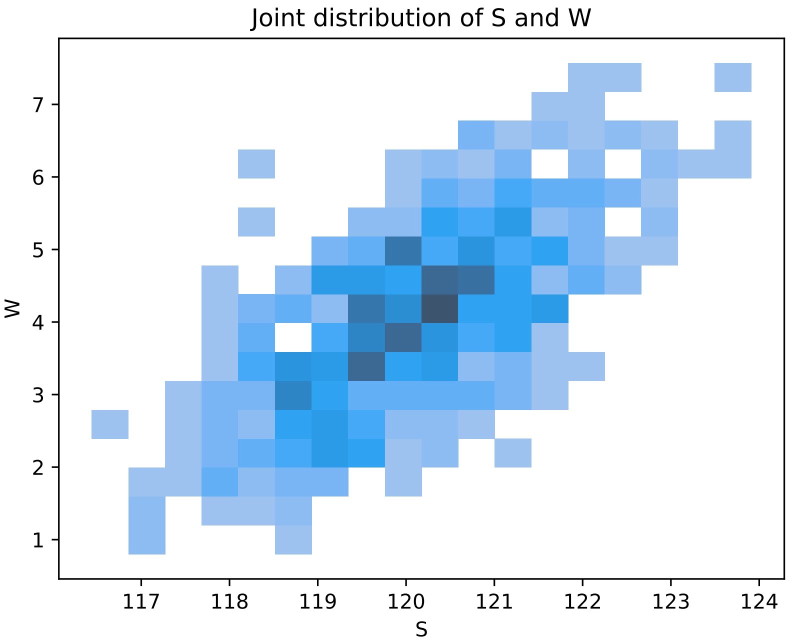
- Processing time T (in seconds) for each package is given by $T = 60 + 0.6W + 0.2S + \epsilon$, where ϵ is a random noise variable with Gaussian distribution $\epsilon \sim \mathcal{N}(0, 5)$.

For this problem, you may find the `multivariate_normal` module from `scipy.stats` especially helpful. You may also find the `seaborn.histplot` function quite helpful.

1. Perform the following tasks:
   (a) Visualize the Bivariate Gaussian distribution for the size S and weight W of the packages by sampling 500 times from the joint distribution of S and W and generating a bivariate histogram of your S and W samples.
   (b) Empirically estimate the most likely combination of size and weight of a package by finding the bin of your bivariate histogram (i.e., specify both a value of S and a value of W) with the highest frequency. A visual inspection is sufficient – you do not need to be incredibly precise. How close are these empirical values to the theoretical expected size and expected weight of a package, according to the given Bivariate Gaussian distribution?
2. For 1001 evenly-spaced values of W between 0 and 10, plot W versus the joint Bivariate Gaussian PDF $p(W, S)$ with S fixed at $S = 118$. Repeat this procedure for S fixed at $S = 122$. Comparing these two PDF plots, what can you say about the correlation of random variables S and W?
3. Give one reason for why the Gaussian distribution is an appropriate model for the size and weight of packages. Give one reason for why it may not be appropriate.
4. Because T is a linear combination of random variables, it itself is a random variable. Using properties of expectations and variance, please compute $\mathbb{E}(T)$ and $\text{Var}(T)$ analytically.
5. Let us treat the *total* amount of time it takes to process *all* packages received at the USPS office within *an entire day* (assuming a single day is 24 hours long) as a random variable T^*.
   (a) Write a function to simulate draws from the distribution of T^*.
   (b) Using your function, empirically estimate the mean and standard deviation of T^* by generating 1000 samples from the distribution of T^*.

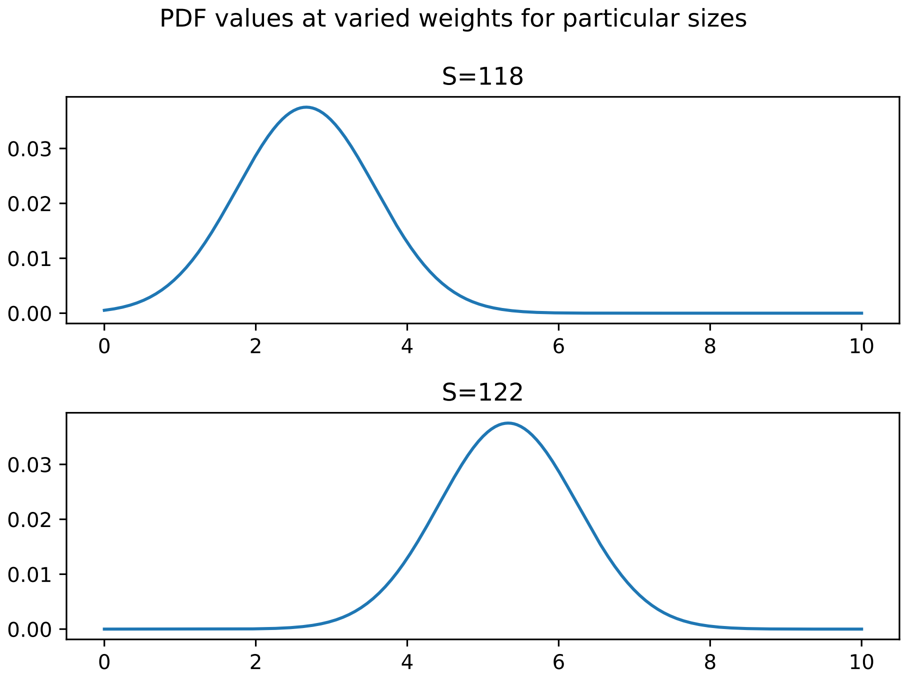
3: Reasoning about Randomness - Probability and Statistics Review

3.1 (a) In the plot below, we visualize the bivariate gaussian distribution for the size S and weight W of packages, generated by sampling randomly 500 times from the MVN.



3.1 (b) With just visual inspection (not being overly precise), the bin in the histogram that appears to have the highest frequency is approximately $(S, W) = (119.5, 3.5)$. This corresponds pretty well with the theoretical expected size and weight of $(120, 4)$.

3.2 Here we plot the package weight vs the PDF values at two fixed package sizes. By comparing the plots where $S = 118$ and $S = 122$, we see that the random variables S and W appear positively correlated: the larger the size, the larger the weight. This is also evident from the plot in 3.1(a).



3.3 Using a Gaussian distribution to model the size and weight of packages is appropriate because it is a continuous distribution and packages are likely to show some central tendencies (the mean package weight and the mean size are meaningful measures to parameterize the distribution) and because these variables should be continuous. On the other hand, it may not be wholly appropriate because it seems plausible that multiple particular combinations of sizes and weights might be more likely: perhaps 12x24, 6x12, and 6x18 are very common package sizes. A Gaussian model of the size is only parameterized by a single mean and variance value. The same could be true of weight (e.g. each shipped product has the same weight, like a particular shoe model). Also, it's probably unlikely that the size and weight distributions will be symmetric: it should be more likely to have small/close to 0 than to have very large values. We also have physical constraints that are not obeyed by Gaussian distributions: we can't have negative weight or sizes.

3.4 The processing time T is a linear combination of the package size, weight, and some noise.
The expected value:

$$\begin{aligned}
\mathbb{E}[T] &= \mathbb{E}[60 + 0.6W + 0.2S + \epsilon] \\
&= 60 + 0.6\mathbb{E}[W] + 0.2\mathbb{E}[S] + \mathbb{E}[\epsilon] && \text{by linearity of expectation} \\
&= 60 + 0.6 * 4 + 0.2 * 120 + 0 && \text{by props of MVN and N} \\
&= 86.4 \text{ seconds}
\end{aligned}$$

The variance:

$$\begin{aligned}
\mathrm{Var}(T) &= \mathrm{Var}(60 + 0.6W + 0.2S + \epsilon) = \mathrm{Var}(0.6W + 0.2S + \epsilon) \\
&= \mathrm{Var}(0.6W) + \mathrm{Var}(0.2S) + \mathrm{Var}(\epsilon) + 2\mathrm{Cov}(0.6W, 0.2S) + 2\mathrm{Cov}(0.6W, \epsilon) + 2\mathrm{Cov}(0.2S, \epsilon) \quad \text{by props variance} \\
&= 0.36\mathrm{Var}(W) + 0.04\mathrm{Var}(S) + \mathrm{Var}(\epsilon) + \\
&\quad 2 * 0.6 * 0.2\mathrm{Cov}(W, S) + 2 * 0.6\mathrm{Cov}(W, \epsilon) + 2 * 0.2\mathrm{Cov}(S, \epsilon) \quad \text{by props (co)variance} \\
&= 0.36 * 1.5 + 0.04 * 1.5 + 5 + (2 * 0.6 * 0.2) * 1 + 0 \quad \text{b/c } \epsilon \text{ is ind.} \\
&= 5.84 \text{ squared seconds}
\end{aligned}$$

3.5 Here we work with the total amount of time it takes to process all packages received within a 24 hr period, calling this random variable T^*.

(a) Here is a function to simulate draws from the distribution of T^*.

```python
def draw_from_tstar():
    pkgs_per_each_hr = poisson.rvs(3, size=24)
    tot_num_pkgs = np.sum(pkgs_per_each_hr)

    ss_ws = mvn.rvs(mu, sigma, size=tot_num_pkgs)
    eps = mvn.rvs(0,5,size=tot_num_pkgs)

    # T = 60+0.6W+0.2S+\epsilon
    ts = 60 + 0.2*ss_ws[:,0] + 0.6*ss_ws[:,1] + eps
    # t* = sum of processing times ts
    tstar = np.sum(ts)
    return tstar
```

(b) Using that function to generate 1000 samples of T^*, we empirically estimate the mean and standard deviation as 6258 seconds and 698 seconds respectively.