

Data C100, Midterm Exam

Summer 2024

Name: _____

Email: _____@berkeley.edu

Student ID: _____

Name of the student to your left: _____

Name of the student to your right: _____

Instructions:

Do not open the exam until instructed to do so.

This exam consists of **72 points** spread out over **6 questions** on **29 pages** and must be completed in the **110 minute** time period on July 19, 2024, from 9:10 AM to 11:00 AM unless you have pre-approved accommodations otherwise.

Note that some questions have circular bubbles to select a choice. This means that you should only **select one choice**. Other questions have boxes. This means you should **select all that apply**. Please shade in the box/circle to mark your answer.

There is space to write your student ID number (SID) in the upper right-hand corner of each page of the exam. **Make sure to write your SID on each page** to ensure that your exam is graded.

Honor Code [Opt or $-\infty$]:

As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others. I am the person whose name is on the exam, and I completed this exam in accordance with the Honor Code.

Signature: _____

This page has been intentionally left blank.

1 Ain't No Free Chocolate Milk [13 Points]

Aditi and Zekai are interested in conducting market research on the chocolate milk drinking habits of UC Berkeley students. More specifically, they want to find out what the most popular chocolate milk is among above-average spenders. With the limited funds at hand, they decide that they can only survey a small sample of students. They devise several different sampling schemes together.

- (a) [2 Pts] What is the *population of interest* in this study? *Use no more than 15 words.*

Solution: All UC Berkeley students that spend an above-average amount on chocolate milk.

- (b) [4 Pts] Aditi suggests that all TAs in Data 100 select every fifth student in the order that they show up to discussion sections this week and ask them for their favorite chocolate milk brand.

- (i) What is the *sampling frame* of this method? *Use no more than 15 words.*

Solution: All Data 100 students that attend discussion section this week.

- (ii) Which of the following statements are true about Aditi's sampling scheme? **Select all that apply.**

- This sampling scheme is a probability sample whereby everyone in the sampling frame has the same probability of ending up in the sample
- This sampling scheme is not a simple random sample, as not every pair of students in our sampling frame has the same probability of ending up in our sample.**
- It is possible that a sampled student is in the sampling frame and is in the population of interest.**
- It is possible that a sampled student is not in the population of interest and is not in the sampling frame.
- None of the above

Solution:

Option 1 is false. Students who show up 5th, 10th etc. have a 1.0 probability of being included. Others are selected with a probability of 0.

Option 2 is true. By definition, this is not a simple random sample.

Option 3 is true. Students in discussion could be above-average spenders on chocolate milk, our population of interest.

Option 4 is false. A sampled student is trivially part of our sampling frame.

(iii) Which types of biases could occur with this sampling scheme? **Select all that apply.**

- Selection Bias**
- Non-response Bias**
- Response Bias
- None of the above

Solution:

Option 1 is correct. Our sampling frame might not necessarily represent our population of interest. Think about students in discussion who do not spend any money on chocolate milk.

Option 2 is correct. Students who do not show up to discussion are excluded from our samples. We might introduce bias this way, as students who do not show up might exhibit different preferences/behaviors than those who do show up.

Option 3 is incorrect. It is unlikely that sampled students have an incentive to provide false answers to a requested survey about chocolate milk preferences.

(c) [3 Pts] Zekai proposes a different sampling scheme. Instead of selecting every fifth student in order of arrival, he proposes that TAs randomly pick one student out of the first seven students that arrive to their section. The TA then repeats this process for the next seven students.

For instance, for the first seven students that arrive, a TA randomly selects one student. Among the next seven students that arrive, they randomly select one student, and so on.

(i) What is the probability that a student who attends discussion this week ends up in our sample? Assume that 35 students show up.

Solution: $1/7$

(ii) What kind of sample is this? **Select all that apply.**

- Probability Sample**
- Simple Random Sample
- Deterministic Sample
- None of the above

Solution:

Option 1 is correct. We are able to specify the probabilities of each student being selected in our sample. There is also randomness involved.

Option 2 is incorrect. Not each pair of students has the same probability of being selected. For instance, student 1 and 2 cannot occur in the same sample. Student 1 and 8 have a $1/7$ probability of being selected.

Option 3 is incorrect. The sample involves randomness.

Aditi and Zekai happen to have a friend who works at a company that makes software that tracks customer purchases for chocolate milk manufacturers. This friend gives them a dataframe `choco_sum` of which the first rows are displayed below. The dataframe contains a row for each above-average spending Berkeley student (tracked through their `berkeley.edu` email), along with the total amount spent on chocolate milk during 2023. The `email` column contains `str` values. `amount` consists of `float` values.

	email	amount
0	aditi@berkeley.edu	105.0
1	charlie@berkeley.edu	25.0
2	james@berkeley.edu	35.0
3	student1@berkeley.edu	500.0

We have another dataframe `survey` that contains the results from our survey. The table has one row for each student that completed the survey according to the sampling scheme in (c). Here are the first couple of rows:

	email	favorite brand
0	student1@berkeley.edu	Fairlife
1	student2@berkeley.edu	Nestlé
2	student3@berkeley.edu	Horizon
3	student4@berkeley.edu	Hershey's

1. `email` (`str`): The student's Berkeley email
2. `favorite brand` (`str`): The student's favorite chocolate milk brand

- (d) [2 Pts] Finish the blanks below such that the dataframe, `choco_survey`, contains one row for each above-average spending student who filled out the survey. The dataframe should contain three columns: `email`, `amount`, and `favorite brand`.

`choco_survey` = _____ (A) _____

Fill in the blank (A)

Solution:

```
choco_survey = choco_sum.merge(survey,
                               left_on='email',
                               right_on='email',
                               how='inner')
```

- (e) [0 Pts] Aditi and Zekai decide to resample from their original sample *with replacement* to get an understanding of the uncertainty. What is the probability that a resample of size 20 contains exactly 1 student whose favorite brand is Fairlife **and** exactly 1 student whose favorite brand is Horizon? The proportions of favorite brands in our original sample is shown below:

	Favorite Brand	Proportion
0	Hershey's	0.30
1	Nestlé	0.50
2	Fairlife	0.15
3	Horizon's	0.05

Leave your answer as a mathematical expression.

Solution:

$$\sum_{i=0}^{18} \frac{20!}{i!(18-i)!} (0.3)^i * (0.5)^{18-i} * 0.15 * 0.05$$

OR

$$\frac{20!}{18!} * (0.8)^{18} * 0.15 * 0.05$$

Since the multinomial distribution wasn't provided on the reference sheet, we have decided to convert this question to be worth 2 points of extra credit.

2 Godly Skincare [12 Points]

Dan is scrolling through BikBok one day and discovers the importance of skincare. He starts using products he found online and keeps track of these products in a DataFrame called `dan_products`. The columns are as follows:

- **Product:** The name of the product (type = `str`).
- **Brand:** The brand the product is from (type = `str`).
- **Aura:** The effectiveness of a product according to Dan (type = `numpy.int64`).
- **Price:** The price of a product in USD on Berkazon (type = `numpy.float64`).
- **Stars:** The online rating of the product on Berkazon (type = `numpy.float64`).
- **Repurchased:** Whether or not Dan has repurchased the product (type = `numpy.bool_`).

- (a) [1 Pt] Dan can't remember what Aura he gave the "Lip Glowly Balm" from the brand "Laneige". Fill in the blank below so that `glowy_aura` is the Aura integer value for this product.

`glowy_aura = _____ (A) _____`

Fill in the blank (A)

Solution:

```
dan_products[(dan_products['Product'] == 'Lip Glowly Balm') /  
& (dan_products['Brand'] == 'Laneige')]['Aura'].iloc[0]
```

Other ways to index to get integer value: `.to_numpy()[0]` and `.values[0]`

- (b) [1 Pt] Dan tells you that 1) he has sorted `dan_products` so that `Stars` is in descending order and 2) that there are 719 rows in the DataFrame. Fill in the blank below so that `med_stars` is the median `Stars` value. You are **not** allowed to use `np.median`.

`med_stars = _____ (A) _____`

Fill in the blank (A)

Solution:

- `dan_products.iloc[359]['Stars']`
- `dan_products['Stars'].iloc[359]`

Other ways to get median index position: `719 // 2` and `(719 - 1) / 2`

- (c) [2 Pts] Dan wants to know the average Aura AND maximum Stars for each brand. Fill in the blank below so that `brand_stats` is assigned to a `DataFrame` that can represent this.

`brand_stats = _____ (A) _____`

Fill in the blank (A)

Solution:

```
dan_products.groupby("Brand").agg({"Aura": "mean", \
                                   "Stars": "max"})
```

- (d) [3 Pts] Dan wants to give Xiaorui a great product so that he can begin his K-Drama arc, but he doesn't want to give Xiaorui products from brands he doesn't like. Select the following option(s) (**select all that apply**) that can return the name of the product with the highest Stars that comes from a brand with an average Aura greater than or equal to 3.5. Assume that there are multiple brands in the `DataFrame` that have an average Aura greater than or equal to 3.5.

- `dan_products.groupby("Brand").filter(lambda x: x["Aura"] \ .mean() >= 3.5).sort_values("Stars", ascending = False) \ ["Product"][0]`
- `dan_products.groupby("Brand").filter(lambda x: x["Aura"] \ .mean() >= 3.5).sort_values("Stars", ascending = False) \ ["Product"].iloc[0]`
- `dan_products.filter(lambda x: np.mean(x["Aura"]) >= 3.5)[0]`
- `dan_products.groupby("Brand").filter(lambda x: \ np.mean(x["Aura"]) >= 3.5).sort_values("Stars", \ ascending = False).iloc[0]["Product"]`
- None of the above.

Solution:

Options 1, 2, and 4 all correctly filter out the brands that have a mean Aura of less than 3.5 and sort the resulting `DataFrame` by Stars in descending order. We then just need to select the Product in the first row in the `DataFrame`, aka in integer row position 0.

- Option 1 is incorrect as it selects the product at **index** position 0 rather than integer position 0.
- Option 2 is correct as it selects the product at **integer** position 0.
- Option 4 is correct. It is very similar to Option 2 except it swaps `.iloc[0]` and `["Product"]`. The `.iloc[0]` results in a `Series` containing the values of the row at integer position 0 with the columns as the index. `["Product"]` then selects the product value.

Option 3 is incorrect. `.filter` should come after a `.groupby` call. This code errors.

- (e) [2 Pts] Dan wants to find out if a product being repurchased has a greater amount of Aura. Fill in the blank below to create a DataFrame called `repurch_aura` that finds the average Aura for each brand, separating it into two columns on whether or not a group of products has been repurchased. If there's a category with no data, keep the value `NaN`.

```
repurch_aura = _____ (A) _____
```

Fill in the blank (A)

Solution:

```
dan_products.pivot_table(  
    values = "Aura",  
    index = "Brand",  
    columns = "Repurchased",  
    aggfunc = "mean",  
    fill_value = None # Optional  
)
```

- (f) [3 Pts] Dan is browsing Berkazon and comes across a review from a skincare influencer that he really likes.

```
review = "This was the WORST (!!!) thing EVER!! i only liked the  
CUT3 pets on the packaging.ZZZ..."
```

```
mysterypattern = r"([A-Z!]+)[^\.]"
```

```
re.findall(mysterypattern , review)
```

Select all the strings that are returned in the call to `re.findall` above (**select all that apply**). Note that spaces are represented as an underscore.

- | | |
|--|---|
| <input type="checkbox"/> "CUT3" | <input type="checkbox"/> "ZZZ" |
| <input type="checkbox"/> "EVER!!" | <input type="checkbox"/> "T" <input type="checkbox"/> None of these strings. |
| <input type="checkbox"/> "WORST_" | <input type="checkbox"/> "!!!!" |

Solution:

This pattern looks for one or more (+) capital letters or ! ([A-Z]!) and then any character except backslash (\) or period (.). Because there is a capturing group in the pattern (parentheses), `re.findall` will return only the characters contained in the capturing group: ([A-Z!]+). Remember that spaces are represented as an underscore.

Option A1 is incorrect as it includes `3` which would not be included in the capture group (not a capital letter or !).

Option A2 is correct as the string "EVER!!_" is matched and "EVER!!" is returned.

Option A3 is incorrect as it includes `_` which would not be included in the capture group (not a capital letter or !).

Option B1 is incorrect as the character after `ZZZ` is a period.

Option B2 is correct as the string "Th" is matched and "T" is returned.

Option B3 is correct as the string "!!!)" is matched and "!!!" is returned.

3 Spill the Bubble Tea [8 Points]

Angela loves TPTEa and has taken to exploring the different drink options available. She compiled this data in a DataFrame called `boba`. Below you can find the descriptions of its columns and the rows. We have also provided the first couple of rows.

- `drink`: name of the boba drink (type=`str`)
- `price`: price of the boba drink (type=`numpy.float64`)
- `bestseller`: whether the drink is a bestseller drink or not (type=`numpy.bool_`)
- `num_times`: the number of times Angela has gotten the drink (type=`numpy.int64`)
- `rating`: Angela's rating for the drink between [1, 10] (type=`numpy.int64`)
 - 1: "I would never get this drink again"
 - 10: "This is my go-to order"

	<code>drink</code>	<code>price</code>	<code>bestseller</code>	<code>num_times</code>	<code>rating</code>
0	Tie Guan Yin Milk Tea	5.95	False	10	10
1	Boba Milk Tea	5.95	True	4	8
2	Strawberry Milk Tea	6.75	False	6	9
3	Passion Fruit Green Tea	6.25	False	1	4
4	Peach Fantasy Ice Tea	6.50	False	0	5
5	Mango Cheese Cream Foam	6.95	True	5	10
6	Thai Milk Tea	5.75	False	1	2
7	Winter Melon Lemon	5.99	False	0	5

(a) [2 Pts] Angela wants to perform exploratory data analysis on the dataset. To familiarize herself with the data, she first identifies the variable types of the columns in `boba`. Determine the variable type that best describes each of the following columns in `boba`.

(i) `drink`

- Qualitative Nominal**
- Qualitative Ordinal
- Quantitative Continuous
- Quantitative Discrete

(iii) `bestseller`

- Qualitative Nominal**
- Qualitative Ordinal
- Quantitative Continuous
- Quantitative Discrete

(ii) `price`

- Qualitative Nominal
- Qualitative Ordinal
- Quantitative Continuous**
- Quantitative Discrete

(iv) `num_times`

- Qualitative Nominal
- Qualitative Ordinal
- Quantitative Continuous
- Quantitative Discrete**

(b) [2 Pts] Which of the following are possible visualizations for `price` only? **Select all that apply.**

- Bar chart
- Histogram**
- Box plot**
- Line Plot
- Contour Plot
- None of the Above

Solution: `price` is a continuous variable. The question asked for valid visualizations for **only** visualizing `price` (so not `price` combined with the other variables) Bar chart is incorrect. This visualization type is normally used with categorical variables.

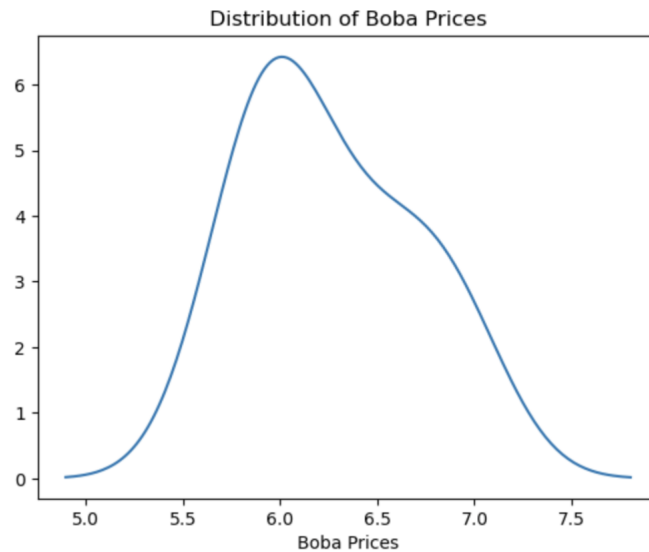
Histogram is correct. This visualization type is used for plotting the density of a continuous variable.

Boxplot is correct. This visualization type is used for visualizing numerical distributions.

Line Plot is incorrect. `price` does not contain the notion of temporality/order/sequence (e.g., time) nor does it have any guarantees about unique values.

Contour plots is incorrect. This visualization type is used for comparing the distributions of two continuous variables.

- (c) [2 Pts] Angela wants to create a KDE curve of the distribution of boba prices in the dataset `boba`. She generated the following KDE curve.



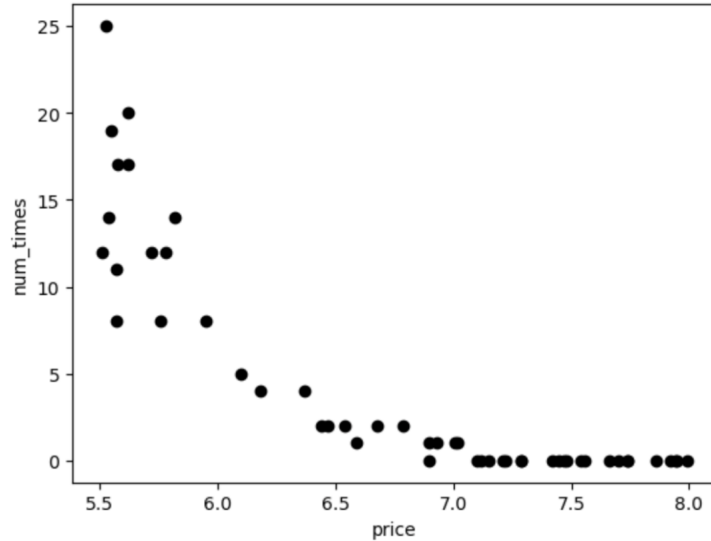
Given the plot above, is this curve a valid KDE curve? **Select one.**

- Yes
- No**
- Not enough information

Explain your answer to the previous part. No credit will be awarded to the previous part if no explanation is provided. *Do not use more than 15 words.*

Solution: No, because the area under the curve is not 1

- (d) [2 Pts] Angela collects more data on the other drinks offered at TPTEa from the students in her section. She decides to plot the relationship between `price` and `num_times` and gets the following plot:



Which transformations should Angela apply to make it so that the relationship between `price` and `num_times` is linear? **Select all that apply.**

- $\sqrt[3]{x}$
- x^3
- \sqrt{y}
- $\log(y)$
- None of the above

Solution:

Using the Tukey-Mosteller bulge diagram:

Option 1, 2 and 4 are correct. As all these transformations would aid in creating a more linear relationship.

Option 2 is incorrect. This would enlarge the differences on the x-axis, resulting in a nonlinear relationship.

4 Lotsa Loss [13 Points]

Suppose we have a dataset with n datapoints and one input, $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}$ is the input and $y_i \in \mathbb{R}$ is the target output. We want to use regression on the inputs x_i to predict y_i . Boyu suggests we use the constant model:

$$\hat{y}_i = \theta \text{ where } \theta \in \mathbb{R}.$$

Additionally, he has a new loss function for us, Weighted Mean Squared Error (WMSE). This loss function allows us to up- and down-weight datapoints. For example, perhaps we want to assign higher weights to more recent datapoints. The formula for WMSE is as follows:

$$\text{WMSE} = \frac{1}{n} \sum_{i=1}^n w_i (y_i - \hat{y}_i)^2$$

- (a) [2 Pts] Say we have the following table which includes our datapoints and the weights we have assigned them:

i	x_i	y_i	w_i
1	100	10	0.5
2	50	20	1.0
3	30	30	1.5
4	20	10	1.0

Calculate the weighted WMSE for $\theta = 20$. **Draw a box around your final answer.**

Solution: Given $\theta = 20$ and:

i	x_i	y_i	w_i
1	100	10	0.5
2	50	20	1.0
3	30	30	1.5
4	20	10	1.0

and $\theta = 20$:

$$\text{WMSE} = \frac{1}{4} \sum_{i=1}^4 w_i (y_i - \theta)^2$$

$$\begin{aligned} \text{WMSE} &= \frac{1}{4} [0.5(10 - 20)^2 + 1.0(20 - 20)^2 + 1.5(30 - 20)^2 + 1.0(10 - 20)^2] \\ &= \frac{1}{4} [0.5(-10)^2 + 1.0(0)^2 + 1.5(10)^2 + 1.0(-10)^2] \\ &= \frac{1}{4} [0.5 \cdot 100 + 0 + 1.5 \cdot 100 + 1.0 \cdot 100] \\ &= \frac{1}{4} [50 + 0 + 150 + 100] = \frac{1}{4} \cdot 300 = \boxed{75} \end{aligned}$$

- (b) [3 Pts] Take the derivative with respect to θ of the WMSE of the model. **Draw a box around your final answer and be sure to simplify it in terms of x_i, y_i, θ, w_i , and/or n .**

Solution:

The Weighted Mean Squared Error (WMSE) is given by:

$$\text{WMSE} = \frac{1}{n} \sum_{i=1}^n w_i (y_i - \hat{y}_i)^2$$

And our model is the constant model:

$$\hat{y}_i = \theta$$

Thus, the loss function we want to minimize is:

$$L_{\text{WMSE}}(\theta) = \frac{1}{n} \sum_{i=1}^n w_i (y_i - \theta)^2$$

$$\begin{aligned} \frac{\partial L_{\text{WMSE}}}{\partial \theta} &= \frac{\partial}{\partial \theta} \left(\frac{1}{n} \sum_{i=1}^n w_i (y_i - \theta)^2 \right) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial \theta} [w_i (y_i - \theta)^2] && \text{Deriv. of a sum = the sum of the derivatives} \\ &= \frac{1}{n} \sum_{i=1}^n w_i \frac{\partial}{\partial \theta} [(y_i - \theta)^2] && w_i \text{ is a constant with respect to } \theta \\ &= \frac{1}{n} \sum_{i=1}^n w_i \cdot 2(y_i - \theta) \frac{\partial}{\partial \theta} (y_i - \theta) && \text{Chain rule} \\ &= \frac{1}{n} \sum_{i=1}^n w_i \cdot 2(y_i - \theta)(-1) \\ &= \boxed{-\frac{2}{n} \sum_{i=1}^n w_i (y_i - \theta)} \end{aligned}$$

- (c) [2 Pts] Now, calculate the value of $\hat{\theta}$, which minimizes the Weighted Mean Squared Error (WMSE) of our model. You may assume, without proof, that the critical point of this loss function is guaranteed to be a minimum. **Draw a box around your final answer and be sure to simplify it in terms of x_i, y_i, θ, w_i , and/or n .**

Solution: To find the value of θ that minimizes the WMSE of our model ($\hat{\theta}$), we set the derivative from (b) equal to zero and solve for θ :

$$-\frac{2}{n} \sum_{i=1}^n w_i (y_i - \theta) = 0 \quad \text{Set the derivative we calculated in (b) equal to 0.}$$

$$\sum_{i=1}^n w_i (y_i - \theta) = 0 \quad \text{Divide both sides by } -\frac{2}{n}.$$

$$\sum_{i=1}^n (w_i y_i - w_i \theta) = 0 \quad \text{Multiply the } w_i \text{ out.}$$

$$\sum_{i=1}^n w_i y_i - \sum_{i=1}^n \theta w_i = 0 \quad \text{Separate into two summations.}$$

$$\sum_{i=1}^n w_i y_i - \theta \sum_{i=1}^n w_i = 0 \quad \text{Pull } \theta \text{ out of second summation as it a constant.}$$

$$\theta \sum_{i=1}^n w_i = \sum_{i=1}^n w_i y_i \quad \text{Move } \theta \sum_{i=1}^n w_i \text{ to other side (add } \theta \sum_{i=1}^n w_i \text{ to both sides).}$$

$$\theta = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i} \quad \text{Divide both sides by } \sum_{i=1}^n w_i.$$

$$\hat{\theta} = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i}$$

(d) [2 Pts] Which of the following statements are true about the model and the loss function defined above? **Select all that apply.**

- The model assumes that all observations have the same variance.
- The WMSE gives more importance to observations with higher weights.**
- The sum of the residuals is always zero.
- The weighted sum of the residuals is always zero.**
- For a fixed set of weights, there is always a single unique optimal value for $\hat{\theta}$ for a given dataset.**
- None of the above.

Solution:

Option 1 is false. The method does not assume that the observations have the same variance. In fact, because we are able to assign different weights, the model explicitly allows for the observations to have different variances.

Option 2 is true. As we can see from (c), our model always predicts the constant $\hat{y} = \hat{\theta} = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i}$. The value that our model predicts is a function of the weights and y -values. Observations with a higher weight will contribute more to the constant prediction value (more importance).

Option 3 is false. Here is a quick counterexample: Say we have datapoints $y_1 = 1$ and $y_2 = 5$ and corresponding weights $w_1 = 1$ and $w_2 = 3$. Then $\hat{\theta} = \frac{(1 \cdot 1) + (5 \cdot 3)}{1 + 3} = \frac{16}{4} = 4$. Then our residuals are $1 - 4 = -3$ and $5 - 4 = 1$. The sum of residuals = $-3 + 1 = -2 \neq 0$.

Option 4 is true. Using $\hat{\theta}$ from (c):

$$\begin{aligned}
 \sum_{j=1}^n w_j (y_j - \hat{y}_j) &= \sum_{j=1}^n w_j (y_j - \hat{\theta}) = \sum_{j=1}^n w_j \left(y_j - \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i} \right) \\
 &= \sum_{j=1}^n \left(w_j y_j - w_j \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i} \right) = \sum_{j=1}^n (w_j y_j) - \sum_{j=1}^n \left(w_j \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i} \right) \\
 &= \sum_{j=1}^n (w_j y_j) - \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i} \sum_{j=1}^n w_j = \sum_{j=1}^n (w_j y_j) - \sum_{i=1}^n (w_i y_i) \\
 &= 0
 \end{aligned}$$

Option 5 was meant to be correct. However, if the weights sum to 0, $\hat{\theta}$ is undefined. Thus, we awarded credit to both answers. The corrected version of this option is: For a fixed set of weights where $\sum_{i=1}^n w_i \neq 0$, there is always a single unique optimal value for $\hat{\theta}$ for a given dataset. Technically (b) and (c) also assume that $\sum_{i=1}^n w_i \neq 0$. In practice, people usually set all weights to be positive.

- (e) [2 Pts] Let's now think about Mean Squared Error (MSE). Recall that MSE is defined as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Which of the following functions are appropriate loss functions that penalize outliers more harshly than MSE? **Select all that apply.**

- $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)$
 $\frac{1}{n} \sum_{i=1}^n [|y_i - \hat{y}_i|^3 + (y_i - \hat{y}_i)^2]$
- $\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$
 $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^4$
- $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^3$
 $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^5$
- $\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|^3$
 $\frac{1}{n} \sum_{i=1}^n |(y_i - \hat{y}_i)^5|$
- None of the above.

Solution:

$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)$, $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^3$, and $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^5$ are inappropriate loss functions as they take the residuals to an odd power. We discussed this in Lecture 10.

$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$ is Mean Absolute Error which does not penalize outliers.

$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|^3$, $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^4$, $\frac{1}{n} \sum_{i=1}^n |(y_i - \hat{y}_i)^5|$ are all appropriate loss functions (the absolute values take care of the odd powers) that penalize outlier more harshly than MSE as their powers are greater than 2.

The second term of $\frac{1}{n} \sum_{i=1}^n [|y_i - \hat{y}_i|^3 + (y_i - \hat{y}_i)^2]$ is MSE. The absolute value takes care of the odd power in the first term and this term has a larger power than 2. Thus, this loss function penalizes outliers more harshly than MSE.

(f) [2 Pts] Rayna brings us some other models to consider. Which of the following are linear models? **Select all that apply.**

$\hat{y}_i = \theta_0 + \theta_1 x_i$

$\hat{y}_i = \theta_0 e^{x_i}$

$\hat{y}_i = \theta_0 + \theta_1 x_i + \sin(\theta_2) x_i^2$

$\hat{y}_i = \theta_0 + x_i^{\theta_1}$

$\hat{y}_i = \theta_0 \theta_1 x_i$

$\hat{y}_i = \theta_0 + \ln(x_i) \theta_1$

None of the above.

Solution:

$\hat{y}_i = \theta_0 + \theta_1 x_i$, $\hat{y}_i = \theta_0 e^{x_i}$, and $\hat{y}_i = \theta_0 + \ln(x_i) \theta_1$ are all linear models as they are linear functions of the **parameters** $(\theta_0, \theta_1, \theta_2)$.

$\hat{y}_i = \theta_0 + \theta_1 x_i + \sin(\theta_2) x_i^2$ is not a linear model as it has parameter θ_2 in a sin function.

$\hat{y}_i = \theta_0 \theta_1 x_i$ is not a linear function as it contains a product of two parameters $\theta_0 \theta_1$.

$\hat{y}_i = \theta_0 + x_i^{\theta_1}$ is not a linear function as it has parameter θ_1 in a power.

5 A Sleep Descent [18 Points]

James and Rohan want to predict how well students do on an exam, y_i , using how much they sleep, x_i . They decide to use the following model:

$$\hat{y}_i = \theta_0^3 x_i^2 + e^{\theta_1} \theta_2^2 x_i,$$

where the three parameters, $\theta_0, \theta_1, \theta_2 \in \mathbb{R}$, are stored in the parameter vector $\vec{\theta} = [\theta_0, \theta_1, \theta_2]^T$.

- (a) [4 Pts] James and Rohan want to use Mean Squared Error (MSE) as it is their favorite loss function. They decide to use batch gradient descent to select their optimal $\vec{\theta}$. What is the gradient vector, $\nabla_{\vec{\theta}} L$, for their choice of loss function (MSE)?

Show all work in the space below and write your final answer on the provided lines. Use \hat{y}_i in your answers where possible and be sure to simplify (e.g. $2(3x)$ should be written as $6x$).

$$\nabla_{\vec{\theta}} L = \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

Solution:

Given our model: $\hat{y}_i = \theta_0^3 x_i^2 + e^{\theta_1} \theta_2^2 x_i$ and loss (MSE): $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$, our loss function is:

$$L(\vec{\theta}) = \frac{1}{n} \sum_{i=1}^n (y_i - (\theta_0^3 x_i^2 + e^{\theta_1} \theta_2^2 x_i))^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \theta_0^3 x_i^2 - e^{\theta_1} \theta_2^2 x_i)^2$$

Using the chain rule:

$$A = \frac{\partial}{\partial \theta_0} L(\vec{\theta}) = -\frac{6}{n} \sum_{i=1}^n (y_i - \hat{y}_i) \theta_0^2 x_i^2$$

$$B = \frac{\partial}{\partial \theta_1} L(\vec{\theta}) = -\frac{2}{n} \sum_{i=1}^n (y_i - \hat{y}_i) e^{\theta_1} \theta_2^2 x_i$$

$$C = \frac{\partial}{\partial \theta_2} L(\vec{\theta}) = -\frac{4}{n} \sum_{i=1}^n (y_i - \hat{y}_i) e^{\theta_1} \theta_2 x_i$$

$$A = \underline{\quad -\frac{6}{n} \sum_{i=1}^n (y_i - \hat{y}_i) \theta_0^2 x_i^2 \quad}$$

$$B = \underline{\quad -\frac{2}{n} \sum_{i=1}^n (y_i - \hat{y}_i) e^{\theta_1} \theta_2^2 x_i \quad}$$

$$C = \underline{\quad -\frac{4}{n} \sum_{i=1}^n (y_i - \hat{y}_i) e^{\theta_1} \theta_2 x_i \quad}$$

- (b) [2 Pts] James and Rohan realize that their dataset actually contains billions of rows. Which of the following are true for computing optimal θ ? **Select all that apply.**
- Batch gradient descent is guaranteed to converge (not necessarily to the optimal parameters)
 - Each iteration of batch gradient descent will be very slow**
 - Stochastic gradient descent will always find the optimal parameters
 - Stochastic or mini-batch gradient descent may be a suitable choice**
 - None of the above

Solution:

Option 1 is false. There is never a guarantee of convergence for batch gradient descent without further information, e.g. proper learning rate, sufficient time steps, convexity.

Option 2 is true. Since the dataset contains **billions** of rows and batch gradient descent uses the whole dataset each iteration, each iteration of batch gradient descent will take a long time.

Option 3 is false. There is never a guarantee of convergence to the optimal parameters for stochastic gradient descent without further information, e.g. convexity, proper learning rate, sufficient time steps.

Option 4 is true. Remember that stochastic and mini-batch gradient descent use a subset of the whole dataset each iteration. (More specifically, stochastic gradient descent uses one datapoint at a time while mini-batch gradient descent uses some subset of size b . Note: Stochastic gradient descent is a special case of mini-batch where $b = 1$.) Thus, due to the size of the dataset, stochastic or mini-batch gradient descent may be a suitable choice here.

- (c) [4 Pts] Rohan's friend suggests they use the following loss function and stochastic gradient descent instead:

$$L^*(\vec{\theta}) = y_i - (\ln \theta_0 + \theta_0 \theta_1 x_i)$$

This loss function has two parameters, $\theta_0, \theta_1 \in \mathbb{R}$ which are stored in the parameter vector $\vec{\theta} = [\theta_0, \theta_1]^T$.

Pushing aside their concerns about this new loss function, James and Rohan decide to proceed and see what the optimal θ_0 and θ_1 estimates are after one iteration using the following row of data:

i	x_i	y_i
1	3	60

They pick the following starting values for θ_0 and θ_1 : $\theta_0^{(0)} = 1$ and $\theta_1^{(0)} = 2$ and set learning rate $\alpha = 0.1$.

Show all work in the space below and write your final answers on the provided lines.

Solution:

$$\frac{\partial}{\partial \theta_0} L^*(\vec{\theta}) = -\frac{1}{\theta_0} - \theta_1 x_i$$

$$\frac{\partial}{\partial \theta_1} L^*(\vec{\theta}) = -\theta_0 x_i$$

Recall the gradient descent update rule: $\vec{\theta}^{(t+1)} = \vec{\theta}^{(t)} - \alpha \nabla_{\vec{\theta}} L(\vec{\theta}^{(t)})$

$$\begin{aligned} \theta_0^{(1)} &= \theta_0^{(0)} - \alpha \left(-\frac{1}{\theta_0^{(0)}} - \theta_1^{(0)} x_i \right) \\ &= 1 - 0.1 \left(-\frac{1}{1} - 2 \cdot 3 \right) = 1 - 0.1(-7) = 1 + 0.7 \\ &= 1.7 \end{aligned}$$

$$\begin{aligned} \theta_1^{(1)} &= \theta_1^{(0)} - \alpha (-\theta_0^{(0)} x_i) \\ &= 2 - 0.1(-1 \cdot 3) = 2 - 0.1(-3) = 2 + 0.3 \\ &= 2.3 \end{aligned}$$

$$\theta_0^{(1)} = \underline{\quad 1.7 \quad}$$

$$\theta_1^{(1)} = \underline{\quad 2.3 \quad}$$

(d) [2 Pts] In general, which of the following statements are true? **Select all that apply.**

- Initial parameter values do not affect the gradient descent convergence.
- If a gradient descent process converged at a local minimum, then it must also have converged at the global minimum.
- The learning rate affects the speed of the gradient descent process and whether or not the model converges.**
- Negative values are an appropriate choice for learning rate.
- Each update of θ may jump back and forth between two sides of the optimal $\hat{\theta}$.**
- None of the above.

Solution:

Option 1 is false. Initial parameter values can affect gradient descent convergence.

Option 2 is false. There is no guarantee that this local minimum is the global minimum unless we know this loss function is convex.

Option 3 is true. The learning rate affects both the speed of the gradient descent process and whether or not the model converges. For example, too large of a learning rate may cause the algorithm to bounce back and forth forever, never converging.

Option 4 is true. Certain choices of learning rate may cause each update of θ to jump back and forth between the two sides of the optimal $\hat{\theta}$. Additionally, this is also possible if one is using stochastic or mini-batch gradient descent.

(e) [2 Pts] In the case that the validation and test performance are significantly worse than the training performance, which of the following statements could explain this behavior? **Select all that apply.**

- The model is underfitting. It might be possible that the model is not expressive enough to model the underlying trend in the data.
- The model is underfitting. It is possible that there is no underlying trend in our data.
- The model is overfitting. It is possible that our model is too expressive and started fitting too closely to the noise in our training data.**
- The training data is not representative of the validation and test data, leading to poor performance.**
- None of the above.

Solution:

Option 1 is incorrect. The model is expressive enough, as exhibited by the high training performance.

Option 2 is incorrect. There appears to be some kind of pattern, otherwise the training performance would not be high.

Option 3 is correct. Since our validation/testing performance is lower, it might be possible that our model fitted to the variance present in our training data.

Option 4 is correct. If there is a mismatch between our training and validation/testing sets, it can lead to this current result. For instance: training a model on bananas and evaluating it on strawberries.

(f) [2 Pts] Which of the following statements about regularization are true? Assume we are using OLS. **Select all that apply.**

- L1 regularization is more likely to lead to a solution in which the coefficients are set to 0 than L2 regularization.**
- L1 regularization is more sensitive to extreme outliers than L2 regularization.
- L2 regularization has a closed form solution, while L1 regularization does not.**
- Introducing regularization always leads to higher performance on the training set.
- None of the above.

Solution:

Option 1 is true. L1 loss (also LASSO) promotes sparsity more. (See Lecture 15 figures)

Option 2 is false. L2 uses the euclidean norm, which has a square term. Vectors with larger values will have a larger norm than those with smaller values, leading to them being penalized more.

Option 3 is true. (See Lecture 15)

Option 4 is false. Regularization might reduce training performance. Think: a model that is perfectly fit, which we then regularize. The resulting model will be penalized for fitting too closely to the variance (present in the training set), leading to a worse training performance (but better test performance).

- (g) [2 Pts] You decide to use two hyperparameters, α (the learning rate), and λ (the regularization penalty). You are considering 3 choices of α and 3 choices of λ . If you perform five-fold cross-validation to choose the best pair of hyperparameters, how many validation errors would you have to calculate? **Draw a box around your final answer.**

Solution: With two hyperparameters to tune and three possible choices of each, we have $3 \cdot 3 = 9$ pairs of possible hyperparameters. As we are performing five-fold cross-validation, for each pair of hyperparameters we need to calculate 5 validation errors. Thus, in total we need to calculate validation error $9 \cdot 5 = \boxed{45}$ times.

6 Least Squares, Most Fun [8 Points]

Given the linear model $\hat{Y} = \mathbb{X}\theta$ and our design matrix $\mathbb{X} \in \mathbb{R}^{n \times (p+1)}$, we have used geometric properties to show that the most optimal solution to the least squares solution is, assuming $\mathbb{X}^T \mathbb{X}$ is invertible:

$$\hat{\theta}_{OLS} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbb{Y}$$

In some cases, our matrix \mathbb{X} might be *wide* whereby $n \ll (p+1)$. In other words, we have many more features than actual observations. We can then take the right pseudo-inverse (also called Moore-Penrose Inverse) if \mathbb{X} is full row rank, allowing us to calculate a least squares solution:

$$\hat{\theta}_{MP} = \mathbb{X}^T (\mathbb{X} \mathbb{X}^T)^{-1} \mathbb{Y}$$

For this question, you can assume that \mathbb{X} contains a column of ones for the intercept, i.e. $\mathbb{X}_{:,0} = \mathbb{1}_n$.

- (a) [2 Pts] During lecture, we covered the computational complexity of solving for $\hat{\theta}_{OLS}$ using the normal equation. What is the computational complexity of calculating $\hat{\theta}_{MP}$ if we use the equation above? Use big O notation and give the tightest bound possible.

Solution:

$$O(pn^2)$$

In Lecture 14, we have seen that $\hat{\theta}_{OLS}$ has a runtime of $O(p^3) + O(np^2)$. In this problem, calculating $\hat{\theta}_{MP}$ involves slight differences in computation time mainly due to the left \mathbb{X}^T and $(\mathbb{X} \mathbb{X}^T)^{-1}$ terms, yielding a runtime of $O(n^3) + O(pn^2)$ instead. Note that by definition, $n \ll p$. Therefore, we drop the $O(n^3)$ term in our final answer.

- (b) [3 Pts] Show that when \mathbb{X} is full rank and square, $\hat{\theta}_{MP} = \hat{\theta}_{OLS}$. In other words, show that $\mathbb{X}^T (\mathbb{X} \mathbb{X}^T)^{-1} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T$.

Solution:

$$\begin{aligned} \mathbb{X}^T (\mathbb{X} \mathbb{X}^T)^{-1} &= (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \\ \mathbb{X}^T (\mathbb{X} \mathbb{X}^T)^{-1} \mathbb{X} &= (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbb{X} = \mathbb{I} \\ \mathbb{X} \mathbb{X}^T (\mathbb{X} \mathbb{X}^T)^{-1} \mathbb{X} &= \mathbb{X} \mathbb{I} \\ \mathbb{X} &= \mathbb{X} \end{aligned}$$

□

- (c) [3 Pts] From class, we also know that $\sum_{i=1}^n e_i = 0$ when $\hat{\theta}_{OLS} = (\mathbb{X}^\top \mathbb{X})^{-1} \mathbb{X}^\top \mathbb{Y}$ and $\mathbb{X}_{:,0} = \mathbb{1}_n$. Show that this property holds when $n \ll (p+1)$ for $\hat{\theta}_{MP} = \mathbb{X}^\top (\mathbb{X} \mathbb{X}^\top)^{-1} \mathbb{Y}$ as well.

Solution:

$$\vec{e} = \mathbb{Y} - \hat{\mathbb{Y}} = 0$$

$$\vec{e} = \mathbb{Y} - \mathbb{X} \hat{\theta}_{MP} = 0$$

$$\vec{e} = \mathbb{Y} - \mathbb{X} \mathbb{X}^\top (\mathbb{X} \mathbb{X}^\top)^{-1} \mathbb{Y}$$

$$\vec{e} = \mathbb{Y} - \mathbb{Y} = 0$$

$$\sum_{i=1}^n e_i = 0$$

□

End of Graded Questions

7 Congratulations [0 Pts]

Congratulations! You have completed the Midterm Exam.

- **Make sure that you have written your student ID number on *each page* of the exam.** You may lose points on pages where you have not done so.
- Also ensure that you have **signed the Honor Code** on the cover page of the exam for 1 point.

[Optional, 0 pts] Draw a picture (or graph) describing your experience in Data 100.

