

# Data 100/200, Final

Spring 2024

Name: \_\_\_\_\_

Email: \_\_\_\_\_@berkeley.edu

Student ID: \_\_\_\_\_

Exam Classroom and Seat Number: \_\_\_\_\_

Name and SID of the person on your right: \_\_\_\_\_

Name and SID of the person on your left: \_\_\_\_\_

## Instructions:

This exam consists of **100 points in 10 questions** and the Honor Code certification. The exam must be completed in **170 minutes** unless you have accommodations supported by a DSP letter. **You must write your Student ID number at the top of each page.**

Note that you should **select one choice** for questions with **circular bubbles**, and **select all that apply** for questions with **boxes**. There is always at least one correct answer. Please **fully** shade in the box/circle to mark your answer. For all math questions, **please simplify your answer**. Please also **show your work** if there is a large box provided.

For all Python questions, you may assume `Pandas` has been imported as `pd`, `NumPy` has been imported as `np`, the Python `RegEx` library has been imported as `re`. For SQL questions, you may assume that a `duckdb` database has been connected.

## Honor Code [1 Pt]:

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 #tbt [18 Pts]

Angela loves celebrating Throwback Thursday, a weekly tradition of posting old pictures on social media. She records all of her previous social media posts in a `DataFrame` called `pics`. The first 5 rows of `pics` and its column descriptions are given below:

- `post_id`: Unique number for each post, assigned in chronological order (`type = numpy.int64`).
- `date`: The date the picture was posted. Assume only one picture can be posted per day (`type = pandas.Timestamp`).
- `likes`: Number of likes the picture received so far (`type = numpy.int64`).
- `day_of_week`: 1 for Monday, 2 for Tuesday, etc. (`type = numpy.int64`).

	<code>post_id</code>	<code>date</code>	<code>likes</code>	<code>day_of_week</code>
0	1	2024-04-25	120	4
1	2	2024-05-02	75	4
2	3	2024-05-08	103	3
3	4	2024-05-09	84	4
4	5	2024-05-11	95	6

```
pics.head()
```

- (a) [2 Pts] **For this part only:** If 30% of the rows in `pics` have missing values in the `date` column, what is the **BEST** option for dealing with these missing entries?
- ☐ A. Drop all rows with missing values.
  - ☐ B. Impute with the mode of the `date` column.
  - ☒ C. Interpolate values using information from the rest of the `DataFrame`.
  - ☐ D. Leave the `DataFrame` as is.

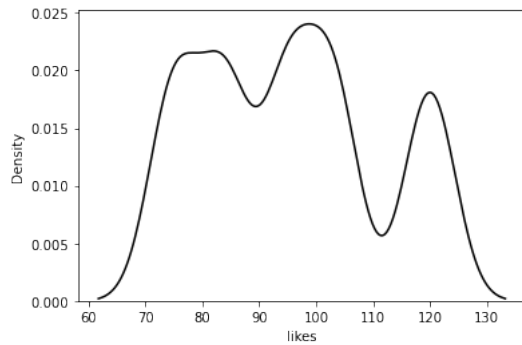
**Solution:** We know that each `post_id` is assigned sequentially, so we can make a good estimate for our missing dates by looking at that column.

- (b) [1 Pt] **In one sentence or less:** What is the granularity of `pics`?

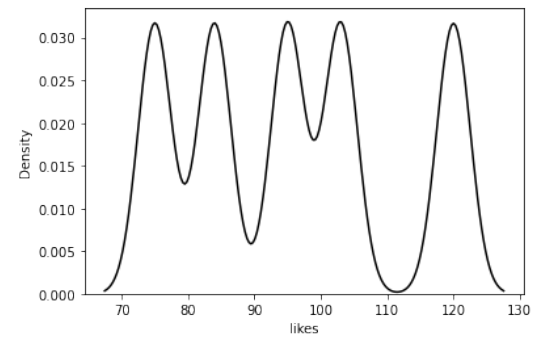
**Solution:** Each row represents one post/picture.

(c) [2 Pts] Angela generates the following KDE curves using a Gaussian kernel. Which KDE curve has the **smallest** bandwidth parameter?

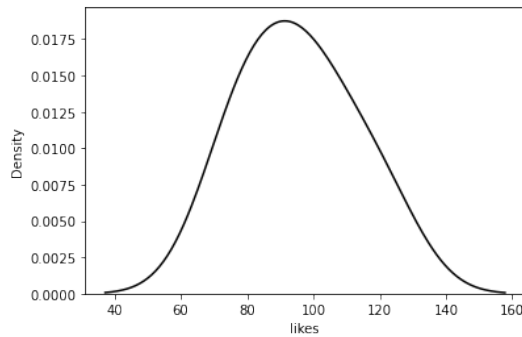
A.



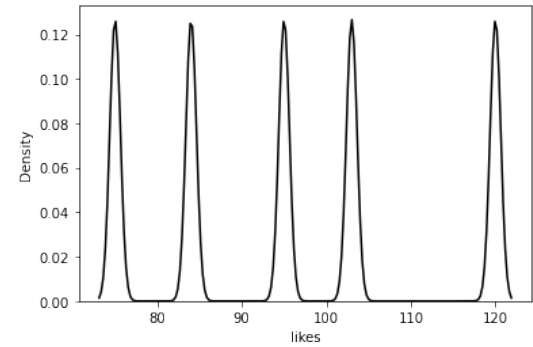
B.



C.



D.



☐ A

☐ C

☐ B

☒ D

**Solution:** D. The smoother the curve, the higher the bandwidth parameter.

Angela has a second DataFrame called `comments` which contains all the comments left on her pictures and the users who wrote them. The first 5 rows can be seen below:

	post_id	comment_user	comment_text
0	1	swei	lol
1	1	yashdave	lol lol :smile_cat:
2	4	lillian	looks fun!
3	4	swei	nice pic! :smile:
4	5	lillian	spotted :open_mouth: :camera:

```
comments.head()
```

- (d) [3 Pts] In each `comment_text`, emojis are represented by combinations of alphabet characters and underscores ("`_`") in between two colons (i.e., "`:laughing_face:`").

Angela creates a RegEx pattern called `emoji_pattern`, which she uses in the following code snippet to generate the given output:

```
example_comment = comments.iloc[4]["comment_text"]
```

```
emoji_pattern = _____A_____
```

```
_____B_____ (emoji_pattern, example_comment)
```

```
['open_mouth', 'camera']
```

- (i) Fill in the blank A by choosing the RegEx patterns which could be `emoji_pattern`.  
**Select all that apply.**

- ☐ A. `r": ([a-zA-Z_]+) :`
- ☐ B. `r": (.*) :`
- ☐ C. `r": (\w)+ :`
- ☐ D. `r": (\w+) :`

**Solution:**

Option A is correct. This allows for any alphabetical character or underscore to appear at least once.

Option B is incorrect. The `.` symbol represents any character, which would lead

to all the emojis being captured in one string rather than separately.

Option C is incorrect. The + is outside the capturing group, capturing only one character from each emoji.

Option D is correct. This allows for any combination of at least 1 letter and/or underscores.

(ii) Fill in the blank B:

- ☐ A. `re.match`
- ☐ B. `re.search`
- ☒ C. `re.findall`
- ☐ D. `str.extract`

**Solution:** We want the output to be a list of all individual matches, so C is the correct option.

Angela uses `pd.merge` to join `pics` and `comments` on the `post_id` column into a new DataFrame called `merged`. The first few rows are shown below:

	post_id	date	likes	day_of_week	comment_user	comment_text
0	1	2024-04-25	120	4	swei	lol
1	1	2024-04-25	120	4	yashdave	lol lol :smile_cat:
2	2	2024-05-02	75	4	NaN	NaN
3	3	2024-05-08	103	3	NaN	NaN
4	4	2024-05-09	84	4	lillian	looks fun!
5	4	2024-05-09	84	4	swei	nice pic! :smile:
6	5	2024-05-11	95	6	lillian	spotted :open_mouth: :camera:

`merged.head(7)`

(e) [2 Pts] If `pics` was the left table and `comments` was the right table, what kind of join did Angela use? Assume that there are no missing values in `comments`.

☐ A. Inner Join

☐ C. Right Join

☒ B. Left Join

☐ D. Not Enough Information

**Solution:** Rows from the left DataFrame (`pics`) which do not have a match in the right DataFrame (`comments`) still appear in `merged`, with the columns from `comments` filled with null values.

(f) [3 Pts] Using `merged`, write a Python statement that creates a DataFrame displaying the **number of times** each `comment_user` commented on each `day_of_week`, including rows with NaN values. Each **column** should represent one `day_of_week` and each **row** should represent one `comment_user`. If a `comment_user` has never commented on a certain `day_of_week` **there should be a value of 0**.

**Solution:**

```
merged.pivot_table(index = "comment_user",
                    columns = "day_of_week",
                    aggfunc = "size",
                    fill_value = 0)
```

Nothing is needed for the `values` argument, but any column in `merged` would also work there.



The first few rows of `merged` are shown again here for your convenience:

	post_id	date	likes	day_of_week	comment_user	comment_text
0	1	2024-04-25	120	4	swei	lol
1	1	2024-04-25	120	4	yashdave	lol lol :smile_cat:
2	2	2024-05-02	75	4	NaN	NaN
3	3	2024-05-08	103	3	NaN	NaN
4	4	2024-05-09	84	4	lillian	looks fun!
5	4	2024-05-09	84	4	swei	nice pic! :smile:
6	5	2024-05-11	95	6	lillian	spotted :open_mouth: :camera:

`merged.head(7)`

- (g) [5 Pts] Fill in the blanks below to create a DataFrame which displays the **longest** `comment_text` for each `comment_user`, as well as the **length** of these comments. Do not worry about ties.

`merged["length"] = _____A_____`  
`merged.____B____.____C____[["length", "comment_text"]].____D_____`

- (i) Fill in blank A:

**Solution:** `merged["comment_text"].str.len()`

- (ii) Fill in blank B:

**Solution:** `sort_values("length")`

students can use `ascending = False` if they aggregate using `.agg("first")`.

- (iii) Fill in blank C:

**Solution:** `groupby("comment_user")`

- (iv) Fill in blank D:

**Solution:** `agg("last")` or `.last()`

or `.agg("first"), .first()` if `.sort_values()` was used in descending order.

## 2 Graduate Descent [6 Pts]

Mir wants to build a model to predict the probability that he will get into each graduate school he applies to. He comes up with the following loss function with the corresponding gradient vector:

$$L(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^n (y_i - \frac{\theta_0}{x_i} - \ln(\theta_1)x_i + \theta_0\theta_1(x_i^2 + 1))$$

$$\nabla_{\theta} L = \begin{bmatrix} \frac{1}{n} \sum_{i=1}^n (-\frac{1}{x_i} + \theta_1 x_i^2 + \theta_1) \\ \frac{1}{n} \sum_{i=1}^n (-\frac{x_i}{\theta_1} + \theta_0 x_i^2 + \theta_0) \end{bmatrix}$$

- (a) [4 Pts] Mir runs a stochastic gradient descent algorithm. He initializes the model's weights as  $\theta_0^{(0)} = 4$  and  $\theta_1^{(0)} = 1$ , then randomly selects the data point  $x_i = 2$  to perform one stochastic gradient descent update. After running one iteration, Mir updates  $\theta_1$  to be  $\theta_1^{(1)} = -5$ . What was Mir's learning rate ( $\alpha$ ) for this update?

$\alpha = \underline{\hspace{2cm}}$

**Solution:** Based on our gradient from above, we can set up our gradient descent update rule for  $\theta_1$  as shown below. Note that we do not need to worry about the summation because we only use one data point in stochastic gradient descent.

$$\theta_1^{(t)} = \theta_1^{(t-1)} - \alpha \left( -\frac{x_i}{\theta_1^{(t-1)}} + \theta_0^{(t-1)} x_i^2 + \theta_0^{(t-1)} \right)$$

Plugging in the values we're given:

$$-5 = 1 - \alpha \left( -\frac{2}{1} + 4(2^2) + 4 \right)$$

This simplifies to

$$-5 = 1 - \alpha(18)$$

From here, we can use some algebra to solve for  $\alpha = \frac{1}{3}$

- (b) [2 Pts] Which of the following statements are true? **Select all that apply.**

- ☐ A. The initial weight(s) chosen for gradient descent can impact whether it converges to the true optimal weights.
- ☐ B. The learning rate  $\alpha$  can impact whether gradient descent converges to the true optimal weights.
- ☐ C. The choice of loss function can impact whether gradient descent converges to the true optimal weights.
- ☐ D. Both batch and stochastic gradient descent compute the true gradient of the loss surface during each iteration.

**Solution:**

Option A is correct. If the loss function has local minima other than the global minima, gradient descent may converge to the local minima depending on where the weights are initialized.

Option B is correct. An inappropriate choice of learning rate can cause gradient descent never to converge or potentially even diverge.

Option C is correct. Like Option A, a non-convex loss function may lead to gradient descent converging to the wrong place.

Option D is incorrect. Batch gradient descent computes the true gradient, but stochastic only uses a small sample to compute this gradient.

### 3 feat. Engineering [14 Pts]

Milad wants to use Ordinary Least Squares (OLS) to predict how many times a song was streamed online. He collects a sample of popular songs from the 2010s in a `DataFrame` called `song_sample`. Its column descriptions and first five rows are shown below:

- `title`: The song title (type = `str`).
- `artist`: The artist(s) performing the song. If there are multiple artists, all guest artists are denoted by "feat. " and separated by commas (type = `str`).
- `genre`: The genre of the song (type = `str`).
- `streams`: The number of times (in millions) the song was streamed online (type = `numpy.float64`).

	title	artist	genre	streams
0	Like a G6	Far East Movement feat. The Cataracs, Dev	hip hop	663.6
1	Love Yourself	Justin Bieber	pop	891.6
2	Telephone	Lady Gaga feat. Beyonce	pop	667.5
3	All I Do is Win	DJ Khaled feat. T-Pain, Ludacris, Snoop Dogg, Rick Ross	hip hop	343.9
4	Need You Now	Lady A	country	252.0

```
song_sample.head()
```

- (a) [2 Pts] Milad posts a thread on the Spring 2024 Data 100 Ed asking students to comment with their favorite song from the 2010s. He then uniformly and randomly selects half of the comments and uses the corresponding songs to form `song_sample`.

What sampling frame is used to create `song_sample`?

- ☐ A. All students in the Spring 2024 Data 100 Ed.  
☐ B. All students who left a comment on Milad's Ed post.  
☒ C. All comments left under Milad's Ed post.  
☐ D. The songs that Milad selected to form `song_sample`.

**Solution:** The sampling frame is what Milad selected **from** to form his sample. In this case, it's the commented songs under the Ed post.

(b) [6 Pts] Milad adds two columns: `has_feat` and `num_feats` to `song_sample`.

	title		artist	genre	streams	has_feat	num_feats
0	Like a G6	Far East Movement feat. The Cataracs, Dev	hip hop		663.6	1	2
1	Love Yourself		Justin Bieber	pop	891.6	0	0
2	Telephone	Lady Gaga feat. Beyonce	pop		667.5	1	1
3	All I Do is Win	DJ Khaled feat. T-Pain, Ludacris, Snoop Dogg, Rick Ross	hip hop		343.9	1	4
4	Need You Now		Lady A	country	252.0	0	0

(i) Fill in the blank in the box below to add the column `has_feat` to `song_sample`. A row has a value of 1 in `has_feat` if the artist has the substring "feat." and 0 otherwise.

`song_sample["has_feat"] = _____`

**Solution:** `song_sample["has_feat"] =  
song_sample["artist"].str.contains("feat.").astype(int)`

(ii) Fill in the blank in the box below to add the column `num_feats` to `song_sample`. This column contains the number of featured artists on each song, where each featured artist is listed after the word "feat." and separated by commas (","). You may assume commas are never used for any other purpose.

`song_sample["num_feats"] = _____`

**Hint:** Songs that don't have any featured artists should have a value of 0. You should use `has_feat` and can assume it was implemented correctly.

**Solution:** `song_sample["num_feats"] =  
song_sample["artist"].str.split(",").str.len()  
* song_sample["has_feat"]`

- (c) [4 Pts] Milad creates a DataFrame called `features` and a Series called `target` using the code snippet below. Assume that there is no value in `genre` called "genre".

```
from sklearn.linear_model import LinearRegression

target = song_sample["streams"]
features = song_sample[["genre"]]

for curr_g in features["genre"].unique():
    features[curr_g] = (features["genre"] == curr_g).astype(int)

features = features.drop("genre", axis=1)

model = LinearRegression(fit_intercept=False)
model.fit(features, target)
```

- (i) Which of the following statements are true for this model? **Select all that apply.**

- ☐ **A. There is a unique optimal solution to Milad's model.**
- ☐ **B. The sum of the residuals for Milad's model is 0.**
- ☐ C.  $\mathbb{X}^T \mathbb{X}$  is not invertible.
- ☐ **D. Milad has performed one-hot encoding on one column.**

**Solution:**

Option A is correct. There are no linear dependencies in `features`.

Option B is correct. While there is no intercept column in `features`, by not leaving out one of the one-hot encoded variables, they will sum up to form the intercept term.

Option C is incorrect. There are no linear dependencies in `features`.

Option D is correct. Milad's code creates a Boolean column that checks whether each song matches each unique value in `genre`.

- (ii) If there are  $m$  different genres in `genre`, what is the dimensionality of  $\hat{\theta}$  in Milad's model? Leave your answer in terms of  $m$ .

Dimensionality of  $\hat{\theta} =$    $\times$

**Solution:**  $m \times 1$ . There is originally 1 column in `features`, so you add one column for each unique value of `genre`, then subtract 1 when we drop `genre`.

- (d) [2 Pts] Which of the following are considered linear models with respect to  $\theta$ ? These models are separate from the rest of the question. **Select all that apply.**

☐ **A.**  $\hat{y} = \theta_1 x_1 + \theta_2 x_2$

☐ **B.**  $\hat{y} = \theta_0 + \theta_1 x_1^2$

☐ **C.**  $\hat{y} = \theta_1 + \theta_1^2 x_1$

☐ **D.**  $\hat{y} = \theta_0 + \theta_0 \theta_1 x_1$

**Solution:** To be a linear model with respect to  $\theta$ , each term in  $\theta$  has to correspond to a single term in the model. Options C and D violate this constraint.

## 4 OLS Town Road [9 Pts]

Jessica is a cowgirl who wants to predict how much food she should give to her cows. Every day, she records data from her cattle in a `DataFrame` called `cows`. Its column descriptions and first five rows are shown below:

- `date`: The date a particular data entry was recorded (`type = str`).
- `name`: The name of a cow (`type = str`).
- `weight`: How many pounds the cow weighed that day (`type = numpy.int64`).
- `food`: How many pounds of food the cow ate that day (`type = numpy.int64`).

	date	name	weight	food
0	May 9th	Angus	2210	25
1	May 9th	Butters	2503	28
2	May 9th	C.R.E.A.M.	3024	38
3	May 8th	Angus	2207	26
4	May 8th	Butters	2501	30

`cows.head()`

- (a) [4 Pts] Jessica wishes to create a model to predict how much `food` to give her cows. She calls the following NumPy functions and stores the outputs in the table below:

Function Call	Output
<code>np.mean(cows["weight"])</code>	2500
<code>np.median(cows["weight"])</code>	2700
<code>np.std(cows["weight"])</code>	500
<code>np.mean(cows["food"])</code>	30
<code>np.median(cows["food"])</code>	35
<code>np.std(cows["food"])</code>	10

For each subpart, calculate  $\hat{\theta}_0$  for the specified model.

- (i) What is the value of  $\hat{\theta}_0$  for a constant model which minimizes **Mean Squared Error (MSE)**?
- ☐ A. 2500
 ☐ C. 2700
 ☒ B. 30
 ☐ D. 35



**Solution:** 30. The MSE can be minimized for a constant model when it is set to the mean of the target variable.

(ii) The table is repeated below for your convenience:

Function Call	Output
<code>np.mean(cows["weight"])</code>	2500
<code>np.median(cows["weight"])</code>	2700
<code>np.std(cows["weight"])</code>	500
<code>np.mean(cows["food"])</code>	30
<code>np.median(cows["food"])</code>	35
<code>np.std(cows["food"])</code>	10

What is the value of  $\hat{\theta}_0$  for a constant model which minimizes **Mean Absolute Error** (MAE)?

☐ A. 2500

☐ C. 2700

☐ B. 30

☒ D. 35

**Solution:** 35. The MAE can be minimized for a constant model when it is set to the median of the target variable.

(iii) Jessica finds the correlation,  $r$ , of `weight` and `food` to be 0.5. What is the value of  $\hat{\theta}_0$  for a **Simple Linear Regression** (SLR) model which predicts `food` from `weight`?

$\hat{\theta}_0 = \underline{\hspace{2cm}}$

**Solution:** With SLR, we can find  $\hat{\theta}_0$  by using the equation  $\hat{\theta}_0 = \bar{y} - \hat{\theta}_1 \bar{x}$ .

First, we need to find  $\hat{\theta}_1$ , which is equal to  $r \frac{\sigma_y}{\sigma_x} = 0.5 \frac{10}{500} = 0.01$

From here, we plug this back into the original equation to get  $\hat{\theta}_0 = 30 - 0.01 * 2500 = 30 - 25 = 5$

- (b) [2 Pts] Jessica decides to use LASSO regression. If Jessica has 5 candidate values of regularizer parameter  $\lambda$ , how many validation errors will she calculate if she runs 4-fold cross-validation?

Answer =

**Solution:** In cross-validation, we must calculate one validation error for each fold for each possible parameter ( $\lambda$  in this case). As a result, our final answer is  $5 * 4 = 20$ .

The number of columns in our feature matrix does not impact how many validation errors we need to calculate.

- (c) [2 Pts] Suppose Jessica has a design matrix  $\mathbb{X}$  with an unknown number of rows and features and a target variable  $\mathbb{Y}$  which stores the data in the `food` column. For each subpart, pick the **best** choice for each given scenario.

- (i) Jessica wants to have unimportant features of  $\mathbb{X}$  have a corresponding weight more likely set to 0. What should Jessica do?

- ☐ A. Use OLS to predict  $\mathbb{Y}$  from  $\mathbb{X}$ .  
☐ B. Use ridge regression to predict  $\mathbb{Y}$  from  $\mathbb{X}$ .  
☒ C. Use LASSO regression to predict  $\mathbb{Y}$  from  $\mathbb{X}$ .

**Solution:** LASSO regression is sparse and will set weights to 0 if they are unimportant.

- (ii) Jessica trains a model using LASSO regression and finds that the training error is low, but the validation error is high. What should Jessica do?

- ☒ A. Increase  $\lambda$ .  
☐ B. Decrease  $\lambda$ .  
☐ C. Set  $\lambda = 0$ .

**Solution:** Increasing  $\lambda$  increases the power of regularization on the model, a common way to combat overfitting.

- (d) [1 Pt] Which of the following is **NOT** a use case for cross-validation?

- ☐ A. Determining how well a model generalizes.  
☐ B. Hyperparameter tuning.  
☐ C. Selecting the degree of polynomial models.  
☒ D. Find an optimal solution when the design matrix is not full column rank.

**Solution:** Cross-validation is not able to address linear dependencies in the feature matrix.

## 5 Attention is All You Need [6 Pts]

Brandon plays a weekly Mahjong tournament against his roommates and wants to analyze how much attention he pays to each game. Assume that his attention level for each game is independent.

- (a) [2 Pts] Brandon classifies each game he plays into one of two categories: High Attention and Low Attention. The probability Brandon has high attention for each game is  $p$ .

- (i) Write a **simplified** expression for the probability that Brandon has a **high** attention level for 3 games in a row.

Probability =

**Solution:** The probability of a high attention level for a single game is  $p$ . As a result, the probability of a high attention level for all 3 games is  $p^3$ .

- (ii) Write a **simplified** expression for the probability that Brandon has a **low** attention level for **at least 1** of the first 5 games he plays?

Probability =

**Solution:** The probability that Brandon's attention level will be low for at least one of five games is  $1 - P(\text{no low attention across five games})$ .

The probability of not having low attention for a single game is the same as the probability of having high attention for a single game (we only have 2 categories). As a result, the probability of having no low attention across 5 games is  $p^5$ , leading to our final answer being  $1 - p^5$ .

- (b) [2 Pts] Brandon decides to use a new random variable  $G$  to model his attention levels, where  $E[G] = 2$  and  $E[G^2] = 5$ . What is  $E[(G - 1)^2]$ ?

☐ A. 1

☐ C. 3

☒ B. 2

☐ D. None of the Above

**Solution:**

$$\begin{aligned} E[(G - 1)^2] &= E[G^2 - 2G + 1] \\ &= E[G^2] - 2E[G] + E[1] \\ &= 5 - 4 + 1 = 2 \end{aligned}$$

- (c) [2 Pts] Brandon adds a third category: Medium Attention. He uses the random variable  $S$  to represent his score for a given game of Mahjong, such that:

$$S = \frac{1}{4} \times S_{low} + \frac{1}{2} \times S_{medium} + \frac{1}{4} \times S_{high}$$

$Var(S_{low})$	$Var(S_{medium})$	$Var(S_{high})$
800	1000	1600

Each score for each game is generated independently of one another. Given the information from above, what is the **variance** of  $S$ ?

$$Var(S) = \underline{\hspace{2cm}}$$

**Solution:** The variance of  $S$  can be written as the variance of the weighted sum:

$$Var(S) = Var\left(\frac{1}{4} \times S_{Low} + \frac{1}{2} \times S_{Medium} + \frac{1}{4} \times S_{High}\right)$$

This can be split up because each score is independent of each other:

$$\begin{aligned} &= Var\left(\frac{1}{4} \times S_{Low}\right) + Var\left(\frac{1}{2} \times S_{Medium}\right) + Var\left(\frac{1}{4} \times S_{High}\right) \\ &= \left(\frac{1}{4}\right)^2 \times Var(S_{Low}) + \left(\frac{1}{2}\right)^2 \times Var(S_{Medium}) + \left(\frac{1}{4}\right)^2 \times Var(S_{High}) \\ &= \frac{1}{16} \times 800 + \frac{1}{4} \times 1000 + \frac{1}{16} \times 1600 \\ &= 50 + 250 + 100 \\ &= 400 \end{aligned}$$

## 6 I Have the High Ground [6 Pts]

Ishani loves hiking and creates multiple models to predict how long it would take her to complete a given trail. She is interested in studying the bias-variance tradeoff of these models.

- (a) [2 Pts] While adding more features to one of her models, Ishani observes that the testing error rapidly **increases** even though the training error **decreases**. Which of the following statements are **true** for this scenario? **Select all that apply.**

- ☐ A. The model bias increases while the variance decreases.
- ☒ **B. Increasing the regularizer parameter  $\lambda$  should help.**
- ☒ **C. Recollecting more precise data should help.**
- ☐ D. Training the model with fewer data points should help.

**Solution:**

Option A is incorrect. In overfitting, the model bias is too low, and the variance is too high.

Option B is correct. Since we are currently overfitting, we want to increase the power of regularization on our model.

Option C is correct. Having better-quality data reduces our observational variance, which can help reduce the empirical risk.

Option D is incorrect. Using fewer training points is almost never helpful.

- (b) [2 Pts] Ishani has a model with a model risk of 32, a model variance of 8, and an observational variance of 3. Ishani adds more features to this model (without changing the number of data points) and sees the model risk decrease from 32 down to 17.

Which of the following are reasonable values for this newer model? **Select all that apply.**

- ☐ A. Model Variance: 5, Observational Variance: 2
- ☒ **B. Model Variance: 10, Observational Variance: 3**
- ☐ C. Model Variance: 15, Observational Variance: 2
- ☐ D. Model Variance: 15, Observational Variance: 3

**Solution:** As we add more features to a model, we expect bias to decrease and variance to increase. As a result, Option A is incorrect.

Additionally, we are not changing anything about the underlying dataset, so the observational variance should remain the same, eliminating Option C (and Option A again) as

a correct choice.

Lastly, combined model and observational variance cannot be greater than the empirical risk, eliminating Option D.

Option B is the only one which satisfies all these criteria.

- (c) [2 Pts] Ishani trains two models on the **same data with the same features**: Model A and Model B. Information on both models is given below:

Model	Bias Squared	Model Variance
A	2	12
B	4	1

Which of the following statements are true? **Select all that apply.**

- ☐ **A. Model A will overfit more than Model B for this dataset.**
- ☐ **B. If both models used LASSO Regression, Model B had the higher  $\lambda$ .**
- ☐ C. Model A could have used Ridge Regression with  $\lambda > 0$  while Model B used OLS.
- ☐ D. Model B must have larger weights than Model A.

**Solution:**

Option A is correct. Having a very low bias but high variance is a clear warning sign of overfitting.

Option B is correct. A higher  $\lambda$  means regularization has a stronger effect on the model and will have a higher bias but lower variance.

Option C is incorrect. OLS will tend to be more likely to overfit than Ridge Regression.

Option D is incorrect. Models with higher weights could have a lower bias and higher variance.

## 7 SpotQL [12 Pts]

This question involves SQL databases. Where applicable, all code for this question must be written as SQL queries.

All semester, Data 100 staff members have participated in #spotted: a game where staff members take pictures of other staff members they run into in public. Yuerou wants to analyze the results of this game and create the SQL table `spotboard`, where each row represents one of the pictures taken. The full table and column descriptions are displayed below:

- `spotter`: The person who took a picture.
- `caught`: The person who was the subject of the picture.
- `month`: The month the picture was taken in.

<code>spotter</code>	<code>caught</code>	<code>month</code>
Matthew	Shreya	April
Shreya	Matthew	April
Simon	Charlie	May
Shreya	Matthew	May
Simon	Matthew	May

`spotboard`

**NOTE:** There was a typo on the example tables when students took this exam. It has since been fixed (the version you are currently viewing is correct).



spotboard is repeated here for your convenience:

spotter	caught	month
Matthew	Shreya	April
Shreya	Matthew	April
Simon	Charlie	May
Shreya	Matthew	May
Simon	Matthew	May

spotboard

- (a) [4 Pts] Fill in the blanks to write a query showing how many times each `spotter` appears in the table per `month`, then sorts on this number from **highest to lowest**. **Break ties alphabetically by month, then the spotter's name**. Your output should match the table below:

spotter	month	num_spots
Simon	May	2
Matthew	April	1
Shreya	April	1
Shreya	May	1

```
SELECT _____ A
FROM spotboard
_____ B
_____ C _____;
```

- (i) Fill in Blank A:

**Solution:** `spotter, month, COUNT(*) AS num_spots`

- (ii) Fill in Blank B:

**Solution:** `GROUP BY spotter, month`

- (iii) Fill in Blank C:

**Solution:** `ORDER BY num_spots DESC, month, spotter`  
 Note: Students can specify ASC after month and/or spotter.

- (b) [3 Pts] The output from Part (a) was stored in a table called `spots`. Similarly, the number of times each name appeared in the `caught` column per month was stored in the table `caughts`. **Both entire tables are shown below.**

spotter	month	num_spots	caught	month	num_caughts
Simon	May	2	Matthew	May	2
Matthew	April	1	Matthew	April	1
Shreya	April	1	Shreya	April	1
Shreya	May	1	Charlie	May	1

spots                      caughts

For each subpart, select the number of rows in the output of each query:

- (i) `SELECT *`  
`FROM spots AS s`  
`JOIN caughts AS c`  
`ON s.spotter = c.caught AND s.month = c.month;`

☐ A. 2

☐ C. 4

☐ B. 3

☐ D. 5

**Solution:** 2. There are only matches for Matthew in April and Shreya in April.

- (ii) `SELECT *`  
`FROM spots AS s`  
`LEFT JOIN caughts AS c`  
`ON s.spotter = c.caught AND s.month = c.month;`

☐ A. 2

☒ C. 4

☐ B. 3

☐ D. 5

**Solution:** 4. In a left join, all the rows in the left table will remain no matter what.

- (c) [5 Pts] Yuerou aggregates the data from `spots` and `caughts` into a table called `spot_stats` (**full table** shown below).

name	month	num_spots	num_caughts
Matthew	April	1	1
Shreya	April	1	1
Charlie	May	0	1
Matthew	May	0	2
Shreya	May	1	0
Simon	May	2	0

`spot_stats`

First, **exclude** rows where someone was never caught for a given month. Next, find the name and **difference** between `num_spots` and `num_caughts` (stored in the column `diff`) for everybody who was caught **AT LEAST as many times** as they spotted someone else. Your output should match the table below:

name	diff
Matthew	-2
Shreya	0
Charlie	-1

```
SELECT _____A_____
FROM spot_stats
_____B_____
GROUP BY name
_____C_____;
```

- (i) Fill in Blank A:

**Solution:** `name, SUM(num_spots - num_caughts) AS diff`

- (ii) Fill in Blank B:

**Solution:** `WHERE num_caughts > 0 or WHERE num_caughts != 0`

- (iii) Fill in Blank C:

**Solution:** `HAVING diff <= 0`  
`or HAVING SUM(num_spots) <= SUM(num_caughts)`

## 8 Catching Telce [11 Pts]

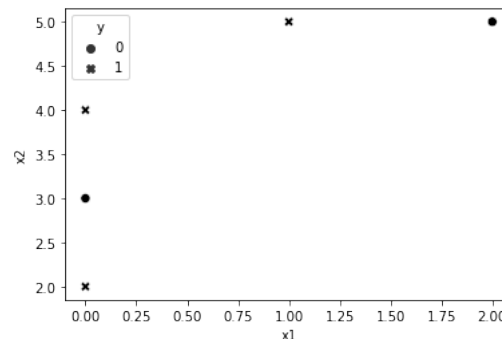
Shiny notices that professional football player Kravis Telce plays much better when superstar singer Saylor Twift is at his game. She collects data from a handful of games this past season, assigning each game a label of  $y_i = 1$  if Saylor Twift was in attendance and  $y_i = 0$  if not.

$\mathbb{X}_{:,1}$	$\mathbb{X}_{:,2}$	$y$
0	3	0
0	4	1
1	5	1
2	5	0
0	2	1

- (a) [2 Pts] What is the maximum accuracy a logistic regression model can achieve for this dataset?

Accuracy =

**Solution:** 0.8. A plot of the data can be seen below:



As you can see, this is not a linearly separable dataset, so perfect accuracy cannot be achieved. However, we can only get one incorrect classification (the lower left datapoint) with a linear boundary, giving us an accuracy of 4/5.

- (b) [4 Pts] Shiny trains a logistic regression model with an intercept term and finds the optimal model parameters to be  $\hat{\theta} = [-3, 1, \frac{1}{2}]^T$ .

The next week, Shiny records the data point  $x_{new} = [x_{new,1}, x_{new,2}]^T = [1, 4]^T$ .

- (i) What is the probability that Saylor Twift was in attendance at the game corresponding to  $x_{new}$ ?

Probability = \_\_\_\_\_

**Solution:**

$$\hat{P}(y = 1|x_{new}) = \sigma([-3, 1, \frac{1}{2}] \times [1, 1, 4]^T) = \sigma(0) = \frac{1}{1 + e^0} = \frac{1}{2}$$

- (ii) If Saylor Twift attended the corresponding game, what is the cross-entropy loss incurred by the model on  $x_{new}$ ?

Loss = \_\_\_\_\_

**Solution:** We plug our answer from part (i) into the cross-entropy loss formula:

$$\begin{aligned} & -[y_{new} \log(\hat{P}(y = 1|x_{new})) + (1 - y_{new})(P(y = 0|x_{new}))] \\ & = -[(1) \log(\frac{1}{2}) + (0) \log(1 - \frac{1}{2})] \\ & = -\log(\frac{1}{2}) \end{aligned}$$

- (c) [5 Pts] The table below shows a sample of validation data and predictions.

- (i) What is the maximum possible recall attainable while maintaining an accuracy above 0.75? What is the range of classification thresholds that achieves this?

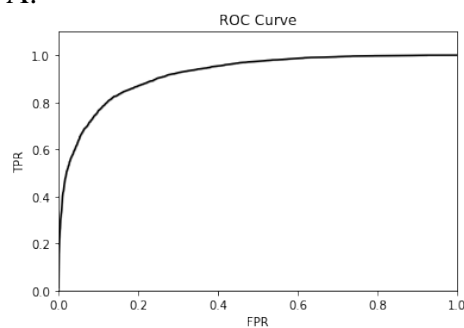
$y_i$	$\hat{P}_\theta(y = 1 x_i)$
1	0.3
0	0.2
0	0.7
0	0.4
1	0.8

Recall = \_\_\_\_\_; Threshold Range = ( \_\_\_\_\_, \_\_\_\_\_ ]

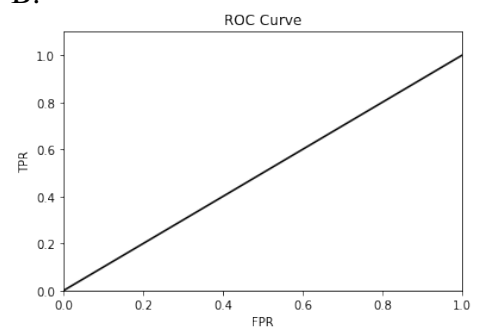
**Solution:** Recall is high if there are few false negatives and more true positives. A maximum recall of 1 can be achieved with a threshold under 0.3 (due to the 1-labeled point with a probability of 0.3), but this falls short of the 0.75 accuracy requirement (it gets 0.6). As such, **the maximum possible recall attainable is 0.5**, because we must have one False Negative (at 0.3) and one True Positive (at 0.8). This can be achieved with a threshold in the range (0.3, 0.8], but to satisfy the 0.75 accuracy requirement, we require **the range to be (0.7, 0.8]**.

- (ii) Suppose we have 4 classification threshold values: 0, 0.25, 0.5, and 0.75. Which ROC curve was generated by these thresholds on the above validation set?

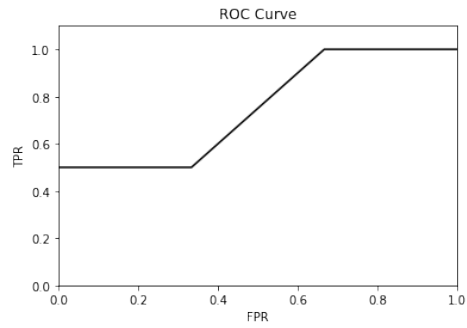
A.



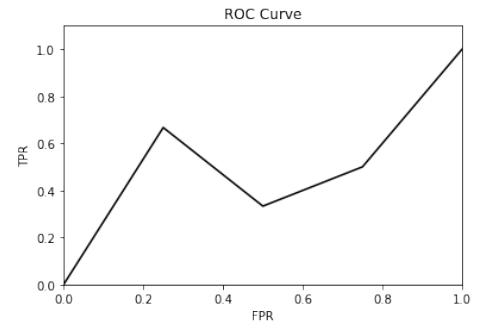
B.



C.

☐ A☐ B

D.

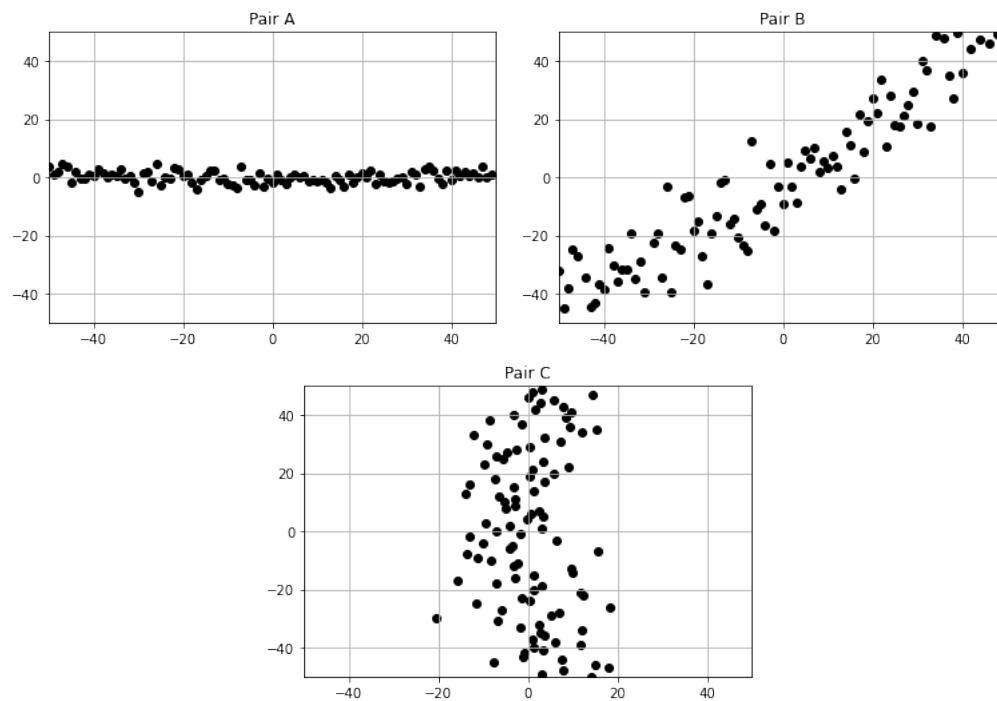
☒ C☐ D

**Solution:** These thresholds yield (FPR, TPR) combinations of  $(1, 1)$ ,  $(\frac{2}{3}, 1)$ ,  $(\frac{1}{3}, \frac{1}{2})$ , and  $(0, \frac{1}{2})$ .

## 9 Is It Principle or Principal? [11 Pts]

Nikhil has a design matrix  $\mathbb{X} \in \mathbb{R}^{n \times d}$ , where  $n > d$  and  $\text{rank}(\mathbb{X}) < d$ . He wishes to perform Principal Component Analysis (PCA) on  $\mathbb{X}$ .

(a) [3 Pts] Nikhil selects 3 unique pairs of columns from  $\mathbb{X}$  and plots them below:



(i) If Nikhil performs PCA on each pair, which pair would have the lowest reconstruction error using its first principal component?

- ☒ A. Pair A
- ☐ B. Pair B
- ☐ C. Pair C
- ☐ D. Not enough information

**Solution:** Pair A has the most variance that lies about a single line.

(ii) Which pair likely has their first principal component pointing along the vertical axis?

- ☐ A. Pair A
- ☐ B. Pair B
- ☒ C. Pair C
- ☐ D. Not enough information



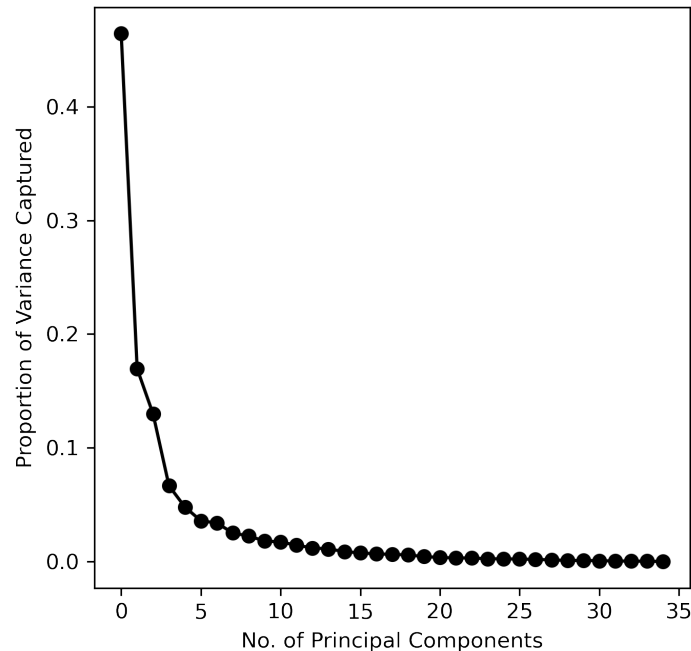
**Solution:** Pair C has the second-most variance along the vertical axis, so that will be the direction of its first PC.

(iii) Which vector approximates the direction of the first principal component of pair B?

- ☐ A. (0, 1)
- ☐ B. (1, 0)
- ☐ C. (0, 0)
- ☒ D. (1, 1)

**Solution:** The most variance lies along a line that goes up and to the right.

- (b) [2 Pts] Nikhil performs PCA and wishes to investigate the number of principal components that may be prominent in his dataset. He generates the following scree plot:



Which of the following statements are true? **Select all that apply.**

- ☐ **A. The intrinsic dimension of this dataset is less than 20.**
- ☐ B. We should keep adding principal components until the explained variance stops increasing.
- ☐ **C. We can retrieve more than half of the explained variance by the first two principal components.**
- ☐ D. There can be a 36th principal component not depicted in this plot, accounting for more than 10% of the variance.

**Solution:**

Option A is correct. The explained variance was already noticeably low well before we reached 20 components.

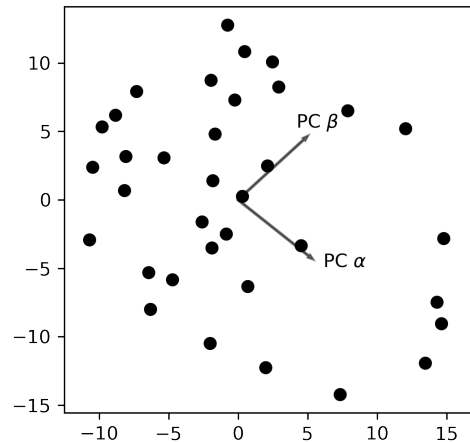
Option B is incorrect. Each PC has a non-zero explained variance and will slowly keep increasing so long as there are more.

Option C is correct. The first PC has an explained variance ratio greater than 0.4; the

second has one greater than 0.1. The sum would be greater than 0.5.

Option D is incorrect. A scree plot must always be decreasing.

- (c) [2 Pts] Nikhil plots each data point along the first two principal components. He doesn't remember which principal component was the first and which was the second, so he randomly names one  $\alpha$  and the other  $\beta$ .



PC  $\alpha$  has a singular value of 48.42, while PC  $\beta$  has a singular value of 36.31. Which of the following statements are true? **Select all that apply.**

- ☐ A. If more data points were added, the results from PCA would never change.
- ☒ B.  $\alpha$  is the first PC, and  $\beta$  is the second PC.
- ☒ C.  $\alpha$  has a higher component score than  $\beta$ .
- ☐ D. The dot product of PC  $\alpha$  and PC  $\beta$  is greater than 0.

**Solution:**

Option A is incorrect. Adding new data points can change the direction of maximum variance.

Option B is correct. X has a higher explained variance than Y.

Option C is correct. This is another term for explained variance ratio.

Option D is incorrect. The two principal components are orthogonal to one another.

- (d) [4 Pts] Suppose Nikhil wishes to decompose  $\mathbb{X}$  using Singular Value Decomposition (SVD), such that  $X = USV^T$ . Which of the following observations must be correct? **Select all that apply.**

**Reminder:**  $\mathbb{X} \in \mathbb{R}^{n \times d}$ , where  $n > d$  and  $\text{rank}(\mathbb{X}) < d$ .

- ☐ A. The dimensions of  $U$  and  $V$  must be the same.
- ☒ **B. There must be at least one 0 along the diagonal of  $S$ .**
- ☐ C.  $S$  and  $V^T$  are both diagonal matrices.
- ☒ **D. The singular values are arranged in non-increasing order.**

**Solution:**

Option A is incorrect. For any original matrix  $X \in \mathbb{R}^{m \times n}$ , the matrix  $U$  would have  $m$  rows, and  $V$  would have  $n$  rows.

Option B is correct. We know this is the case because  $\text{rank}(X) < d$ .

Option C is incorrect. Only  $S$  is diagonal.

Option D is correct. This is why principal components are also in order.

## 10 k-Medians [6 Pts]

Professors Norouzi and Gonzalez wish to analyze the differences between semesters in which at least one of them taught Data 100 and semesters in which other professors taught the course.

- (a) [4 Pts] Suppose the professors have a dataset with one feature,  $x$ , shown in the table below:

$x$	1	1	1	3	4	5	6	8
-----	---	---	---	---	---	---	---	---

The professors want to try a new clustering method called k-Medians, which works the same as k-Means, but with the cluster centers being located at the **median** of their data points instead of the mean. **They initialize 2 clusters centered at 2 and 7.**

- (i) After assigning each point to one of these clusters, what are the  $x$  values of points located in the cluster centered at 7?

$x$  values =

**Solution:** 5, 6, and 8. These points are all closer to 7 than 2.

- (ii) After the first reassignment of cluster centers in k-Medians, where are the new cluster centers located?

Centers =

**Solution:** 1 and 6. These are the median values of their respective clusters.

- (iii) Has k-Medians clustering converged after this first iteration?

- ☐ A. Yes  
☒ **B. No**  
☐ C. Impossible to tell

**Solution:** With the new cluster centers calculated above, the point with an  $x$  value of 4 switches clusters in the next iteration.

- (iv) If the professor were to start over and implement hierarchical agglomerative clustering, what are the  $x$  values of the first two clusters to merge?

$x$  values =

**Solution:** 1 and 1. These are the absolute closest points to one another, as they have the same value.

(b) [2 Pts] Which of the following statements are correct? **Select all that apply.**

- ☐ **A. Where cluster centers are initialized for k-Means clustering can impact the final result.**
- ☐ B. In k-Means clustering, inertia can converge to the true minimum for any starting configuration.
- ☐ C. In k-Means clustering, distortion can converge to the true minimum for any starting configuration.
- ☐ **D. The choice of linkage type in agglomeration clustering can impact the final result.**

**Solution:**

Option A is correct. Initializing clusters at different locations may lead to points converging to different clusters.


Option B is incorrect. Inertia is not able to be optimized for all possible configurations.

Option C is incorrect. Distortion, like inertia, cannot be optimized for all configurations.

Option D is correct. Choosing single, average, or complete linkage can change how the distance between points and clusters is calculated.

**You are done with the final! Congratulations!**

Use this page to draw your favorite Data 100 moment!

A large, empty rectangular box with a thin black border, intended for a student to draw their favorite Data 100 moment.