Data C100/200 Final

Spring 2025

Name: ____________________

Email: ____________________@berkeley.edu

Student ID: ____________________

Name and SID of the person on your left: ____________________

Name and SID of the person on your right: ____________________

Exam Room: ____________________ Seat Number: ____________________

Instructions:

This exam consists of **51 points** spread out over **4 questions** and the **Honor Code certification**. The exam must be completed in **170 minutes** unless you have accommodations supported by a DSP letter. Note that you should:

- 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 circle/box **fully** to mark your answer.
- Blank answers and incorrect answers are graded identically, so it's in your best interest to answer every question.
- For all coding questions, you may use commas and/or one or more function calls in each blank.
- **You MUST write your Student ID number at the top of each page.**
- You should not use a calculator, scratch paper, or notes you own other than the reference sheets distributed at the beginning of the exam.

For all Python questions, you may assume `Pandas` has been imported as `pd`, `NumPy` as `np`, the Python RegEx library as `re`, `matplotlib.pyplot` as `plt`, and `seaborn` as `sns`.

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 Ordering Chicken Nuggets for the Table [14 Pts]

A chicken nugget fan samples nuggets from various fast food chains across Berkeley and records the data in a `DataFrame` called `nuggets`. The columns of `nuggets` are:

- `chain`: Name of the fast food chain visited by the chicken nugget fan (type = `str`).
- `location`: Full address in Berkeley where the nugget was purchased, formatted as: `"StreetNumber␣StreetName,␣5DigitZIPCode"` (type = `str`).
  Note : The character `"␣"` represents a single space.
- `shape`: Possible nugget shapes: `"circle"`, `"oval"`, `"square"`, or `"crown"` (type = `str`).
- `weight`: Nugget weight in ounces (type = `np.float64`).

The first 5 rows of `nuggets` are shown below:

	chain	location	shape	weight
0	McDonald's	123 San Pablo Ave, 94702	oval	0.60
1	Wendy's	245 Telegraph Ave, 94704	square	0.58
2	Wendy's	88 Shattuck Square, 94709	circle	0.70
3	Chick-fil-A	300 Shattuck Ave, 94705	circle	0.70
4	Burger King	19 Ashby Blvd, 94703	crown	0.60

(a) [1 Pt] The fan only purchased nuggets from fast food locations near their apartment in Berkeley. What type of sampling is this?

> **Solution:** Convenience Sampling

(b) Fill in the blanks to output the `chain` that sold the most nuggets with `weight` less than 0.6 ounces. Your output should be a `String`. There is exactly one chain with the highest count.

```
nuggets[__(i)__][__(ii)__].value_counts()__(iii)__
```

(i) [0.5 Pts] Fill in blank `(i)`:

> **Solution:** `nuggets["weight"] < 0.6`

(ii) [0.5 Pts] Fill in blank `(ii)`:

> **Solution:** `"chain"`

(iii) [0.5 Pts] Fill in blank `(iii)`:

> **Solution:** `.idxmax()`, `.index[0]`, or equivalent.
> `.sort_values(ascending=False)` can precede these.

The ANA (American Nugget Association) releases the expected `price_per_ounce` of chicken nuggets sold at each fast food chain in a `DataFrame` called `expected_prices`.

The first 5 rows of `expected_prices` are shown below:

	chain	price_per_ounce
0	McDonald's	1.20
1	Wendy's	1.10
2	Chick-fil-A	1.35
3	Burger King	1.05
4	Popeyes	1.25

The first 5 rows of `nuggets` are shown again here for your convenience:

	chain	location	shape	weight
0	McDonald's	123 San Pablo Ave, 94702	oval	0.60
1	Wendy's	245 Telegraph Ave, 94704	square	0.58
2	Wendy's	88 Shattuck Square, 94709	circle	0.70
3	Chick-fil-A	300 Shattuck Ave, 94705	circle	0.70
4	Burger King	19 Ashby Blvd, 94703	crown	0.60

(c) Fill in the blanks to create a `DataFrame` with one row per `chain` and a column `avg_price_per_nugget`, showing the average price of a single nugget at each chain, irrespective of `shape`.

Hint: The average price per nugget for a given chain is the product of the average weight per nugget in that chain and the price per ounce of nuggets from `expected_prices`.

```
# Step 1: Compute average weight for each chain.
avg_weights = nuggets____(i)____["weight"].mean()
# Step 2: Convert to DataFrame and rename the column.
avg_weights = pd.DataFrame(avg_weights).reset_index()
avg_weights = avg_weights.rename(columns={"chain" : "chain",
"weight" : "average_weight"})
# Step 3: Merge with expected prices.
merged = avg_weights.____(ii)____
# Step 4: Compute average price per nugget.
merged["avg_price_per_nugget"] = ____(iii)____
```

(i) [0.5 Pts] Fill in blank `(i)`:

Solution: `.groupby("chain")`

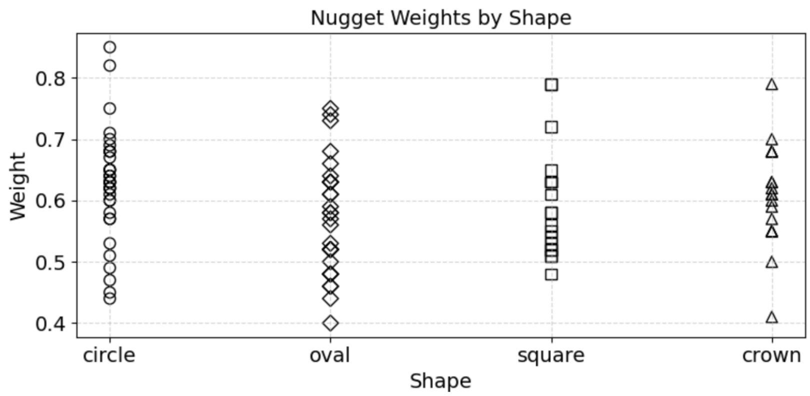
(ii) [1 Pt] Fill in blank `(ii)`:

Solution: `merge(expected_prices, on="chain")`. `.join` CANNOT be used instead with the same syntax. `merge(expected_prices, left_on="chain", right_on="chain")` is also valid.

(iii) [1 Pt] Fill in blank `(iii)`:
Reminder: The average price per nugget for a given chain is the product of the average weight per nugget in that chain and the price per ounce of nuggets from `expected_prices`.

Solution: `merged["average_weight"]*merged["price_per_ounce"]`

(d) [2 Pts] The fan creates the following plot to visualize the distribution of nugget weights aggregated by shape, across all chains in the sample.



Which of the following alternative plot types could be used to determine whether the distribution of nugget weights is bimodal for a given shape?

○ **True** ○ False Overlaid histograms.
○ True ○ **False** Bar chart of mean weight by shape.
○ **True** ○ False Overlaid Kernel Density Estimates (KDEs).
○ True ○ **False** Stacked horizontal boxplots.

(e) [2 Pts] Fill in the blank to extract the **street name** from each `location` in the `nuggets` `DataFrame`. Store the resulting values in a new column called `street`.

```
nuggets["street"] = nuggets["location"].str.extract(r"_______")
```

Note 1: Each `location` is formatted as:

`"StreetNumber␣StreetName,␣5DigitZIPCode"`

The character "␣" represents a single space, and you may use it in your response.

Note 2: `StreetNumber` contains only digits (i.e., no letters, ␣, punctuation, or special characters). `StreetNumber` can start with any digit, including 0. There will always be at least one digit in every `StreetNumber`.

Note 3: `StreetName` only contains uppercase and lowercase letters in the English alphabet and ␣ (i.e., no numbers, punctuation, or special characters). There will always be at least one letter in every `StreetName`.

Example:

- `location`: `"123␣San␣Pablo␣Ave,␣94702"`
- `street`: `"San␣Pablo␣Ave"`

Fill in the blank with the appropriate `RegEx` pattern:

Solution:

Solution: Many possibilities, including:

1. `\d+\s([A-Za-z\s]+)`
2. `\d+\s([\w\s]+)`
3. `\d+\s(.+?),`
4. `\d+\s(.+),\s\d{5}$`

+1 point for `\d+\s` (or equivalent), +1 point for fully correct capture group (everything that comes after `\d+\s`. Alternate solutions verified case-by-case.

(f) The ANA publishes the expected proportion of each nugget shape sold across all restaurants in the United States. They report these values in a `DataFrame` called `official_data`.

`official_data` is shown below:

	shape	expected_proportion
0	oval	0.2
1	crown	0.3
2	circle	0.1
3	square	0.4

The first 5 rows of `nuggets` are shown below:

	chain	location	shape	weight
0	McDonald's	123 San Pablo Ave, 94702	oval	0.60
1	Wendy's	245 Telegraph Ave, 94704	square	0.58
2	Wendy's	88 Shattuck Square, 94709	circle	0.70
3	Chick-fil-A	300 Shattuck Ave, 94705	circle	0.70
4	Burger King	19 Ashby Blvd, 94703	crown	0.60

The **observed sample proportion** for a given nugget shape is the number of rows in `nuggets` containing that shape, divided by the total number of rows in `nuggets`.

Fill in the blanks to write a SQL query that returns, for each nugget shape, the difference between its **observed sample proportion** and its `expected_proportion`. Store this difference in a column called `proportion_diff`. Your output should look like the table below:

shape	proportion_diff
circle	0.22
crown	-0.05
oval	0.04
square	-0.21

For this SQL question, assume that `duckdb` is imported and `nuggets` and `official_data` can be queried as SQL tables. Treat `DataFrame` names as SQL table names in your query.

```
WITH total AS (
  SELECT ____(i)____ AS total_count
  FROM nuggets),
counts AS (
  SELECT ____(ii)____
  FROM nuggets
  GROUP BY ____(iii)____)
SELECT
  c.shape,
  ( ____(A)____- o.expected_proportion) ____(B)____
FROM
  total AS t,
  counts AS c
  JOIN official_data AS o
  ON ____(C)____;
```

(i) [0.5 Pts] Fill blank `(i)` such that `total` is a table with the total number of nuggets. `total` should look like this:

total_count
108

Solution: `COUNT(*)`

(ii) [1.5 Pts] Fill blank `(ii)` [and `(iii)` in the subpart below], such that `counts` is a table that groups the nuggets by `shape` and counts how many of each shape are in `nuggets`. `counts` should look like this:

shape	count
circle	35
crown	27
oval	26
square	20

Solution: `shape, COUNT(*) AS count`

(iii) [0.5 Pts] Fill blank `(iii)` such that `counts` looks like the output above.

Solution: `shape`

(iv) Fill blanks `(A)`, `(B)`, `(C)`, such that the final output of the SQL query looks like this:

shape	proportion_diff
circle	0.22
crown	-0.05
oval	0.04
square	-0.21

(A) [1 Pt] Fill blank `(A)` with an expression that computes the **observed sample proportion**.

Solution: `c.count/t.total_count`

(B) [0.5 Pts] Fill blank `(B)` to name the column of differences `proportion_diff`.

Solution: `AS proportion_diff`

(C) [1 Pt] Fill blank `(C)` to join relevant tables.

Solution: `c.shape = o.shape`

2 Regula-Raising Cane's [13.5 Pts]

Raising Cane's plans to open in Berkeley! To estimate profits, the owner builds a linear model.

(a) [2.5 Pts] The owner collects $n = 30$ observations and fits an Ordinary Least Squares (OLS) model with 29 features (e.g., foot traffic, temperature, etc.) with an intercept (30 parameters total). The model is:

$$\hat{y}_i = \hat{\theta}_0 + \hat{\theta}_1 x_{i1} + \hat{\theta}_2 x_{i2} + \cdots + \hat{\theta}_{29} x_{i29}$$

Which of the following is **always** true?

- ◯ **True** ◯ False — If the design matrix columns are linearly independent, the OLS solution is unique.
- ◯ **True** ◯ False — The sum of the residuals is 0.
- ◯ True ◯ **False** — Adding more features (i.e., exceeding 30 parameters) yields a unique OLS solution.
- ◯ **True** ◯ False — With $L2$ regularization ($\lambda > 0$), the OLS solution is guaranteed to be unique.
- ◯ True ◯ **False** — With $L1$ regularization ($\lambda > 0$), the OLS solution is guaranteed to be unique.

(b) [1.5 Pts] The owner selects 10 features from part (a) to build a new model, without modifying or transforming the 10 features. The owner observes that the **training error increases, but the validation error decreases**.

Which of the following must be true?

- ◯ True ◯ **False** — The observational variance is lower relative to the model in part (a).
- ◯ True ◯ **False** — The model $(\text{bias})^2$ is lower relative to the model in part (a).
- ◯ **True** ◯ False — The model variance is lower relative to the model in part (a).

(c) To reduce model complexity, the owner computes the optimal $\vec{\theta}$ by minimizing the following regularized loss function:

$$L(\vec{\theta}) = (||Y - X\vec{\theta}||_2)^2 + \lambda g(\vec{\theta})$$

where $g(\vec{\theta})$ is a non-constant function of $\vec{\theta}$ and $\lambda \geq 0$. If the magnitude of any θ_i increases, $g(\vec{\theta})$ will also increase.

(i) [1 pt] Which of the following options reduces model complexity?

- ◯ Decreasing λ.
- ◯ **Increasing λ.**
- ◯ Changing λ has no effect on model complexity.

(ii) [1 pt] Which $g(\vec{\theta})$ is best for feature selection (i.e., eliminating irrelevant features and keeping useful ones)? **Note:** p is the total number of parameters in $\vec{\theta}$.

- ◯ $g(\vec{\theta}) = \sum_{i=1}^{p} \log(\theta_i)$
- ◯ $g(\vec{\theta}) = \sum_{i=1}^{p} \theta_i$
- ◯ $g(\vec{\theta}) = \sum_{i=1}^{p} |\theta_i|$
- ◯ $g(\vec{\theta}) = \sum_{i=1}^{p} \theta_i^2$

(d) [1.5 Pts] The Raising Cane's team uses 4-fold cross-validation for hyperparameter tuning. Which of the following is always true?

◯ True ◯ **False** Cross-validation folds are created by sampling with replacement from the original data.

◯ **True** ◯ False Each data point is in the validation set exactly once across the 4 folds.

◯ True ◯ **False** With 4 folds, each data point is used for training exactly 4 times.

(e) As part of a market expansion, Raising Cane's tests a custom profit forecasting model for Berkeley, using a custom objective function to optimize predicted monthly profits.

The objective function is defined as:

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

$$\nabla L(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} \begin{bmatrix} -\frac{1-\ln(\theta_0)}{\theta_0^2} + \theta_1 \\ (A) \end{bmatrix}$$

(i) [1 pt] Complete the gradient vector by calculating blank (A).

Solution: $-x_i + \theta_0$

(ii) [2 pt] The owner initializes $\theta_0^{(0)} = 1$ and $\theta_1^{(0)} = 2$, and sets $\alpha = 0.5$. Given this information, what is θ_0 after one batch gradient descent step?

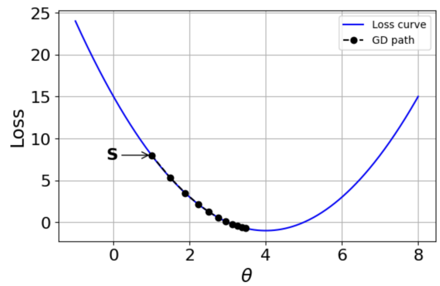
Solution:

$$\theta_0^{(0)} = 1, \quad \theta_1^{(0)} = 2, \quad \alpha = 0.5$$

$$\ln(\theta_0) = \ln(1) = 0 \Rightarrow \frac{\partial L}{\partial \theta_0} = -\frac{1-0}{1^2} + 2 = -1 + 2 = 1$$

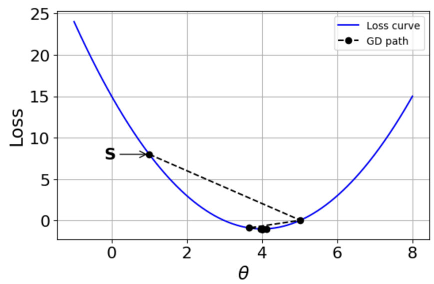
$$\theta_0^{(1)} = \theta_0^{(0)} - \alpha \cdot \frac{\partial L}{\partial \theta_0} = 1 - 0.5 \cdot 1 = \boxed{0.5}$$

(f) [2 Pts] The plots below show Gradient Descent (GD) on a quadratic loss function with a fixed learning rate over 10 updates. The starting point is labeled with an **S** in the plots below.

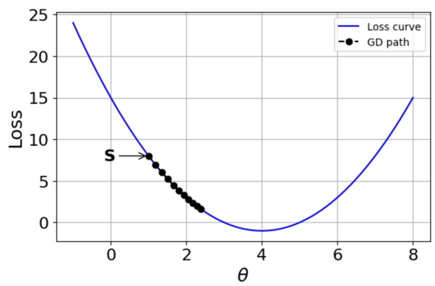


Select all outcomes that could occur if only the learning rate increases, with the loss function and all other gradient descent inputs unchanged.

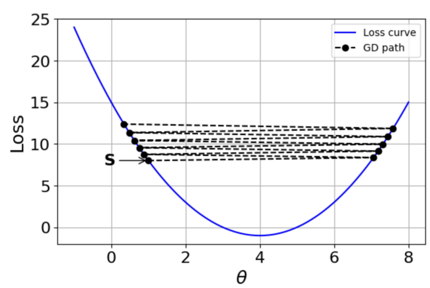
○ **True** ○ False



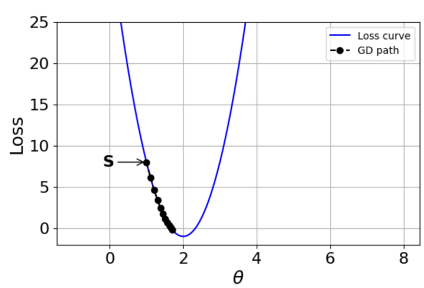
○ True ○ **False**



○ **True** ○ False



○ True ○ **False**



(g) [1 Pt] Three different training paradigms are used to optimize a model. Each runs for **5 epochs**.

Paradigm	Number of Data Points
Paradigm 1 (Batch GD)	120
Paradigm 2 (Mini-batch GD)	100
Paradigm 3 (Stochastic GD)	10

Which ordering correctly ranks the paradigms by total number of gradient computations **across all epochs**?
Note: Paradigm 2 uses a **batch size of 20.**

- ◯ **Paradigm 1 $<$ Paradigm 2 $<$ Paradigm 3**
- ◯ Paradigm 2 $<$ Paradigm 1 $<$ Paradigm 3
- ◯ Paradigm 3 $<$ Paradigm 2 $<$ Paradigm 1
- ◯ Paradigm 2 $<$ Paradigm 3 $<$ Paradigm 1

Solution: **Paradigm 1 (Batch GD):** 1 gradient per epoch $\times$ 5 epochs = $\boxed{5}$ gradients.

Paradigm 2 (Mini-batch GD):

$$\frac{100}{20} = 5 \text{ mini-batches per epoch} \quad \Rightarrow \quad 5 \times 5 = \boxed{25} \text{ gradients.}$$

Paradigm 3 (SGD):

$$10 \text{ points per epoch} \quad \Rightarrow \quad 10 \times 5 = \boxed{50} \text{ gradients.}$$

Thus:

$$\boxed{\text{Paradigm 1} < \text{Paradigm 2} < \text{Paradigm 3}}.$$

3 δrake vs. Kenδrick [13 Pts]

After the Kendrick vs. Drake feud, you want to answer: *Who do UC Berkeley students listen to more?* You collect two independent random samples of UC Berkeley students using their Spotify Data 100 playlist listening data:

- A sample of m students who listened to Drake more than Kendrick (**Drake fans**). For each student, we record their weekly hours listening to Drake as D_i for $i = 1, \dots, m$.
- A sample of n students who listened to Kendrick more than Drake (**Kendrick fans**). For each student, we record their weekly hours listening to Kendrick as K_j for $j = 1, \dots, n$.

Note: Each sampled student listens more to Kendrick or Drake. They never listen to both equally. Assume the following:

- The Drake listening times (in hours), D_i, are independent and identically distributed (i.i.d.) with mean μ_D and variance σ_D^2.
- The Kendrick listening times (in hours), K_j, are i.i.d. with mean μ_K and variance σ_K^2.
- The two samples are independent of each other.

You're interested in whether UC Berkeley students listen to Kendrick and Drake equally, so you decide to estimate the difference in their mean listening times:

$$\delta = \mu_K - \mu_D$$

and use the estimator:

$$\hat{\delta} = \bar{K} - \bar{D}, \quad \text{where } \bar{K} = \frac{1}{n}\sum_{j=1}^{n} K_j, \quad \bar{D} = \frac{1}{m}\sum_{i=1}^{m} D_i.$$

(a) [1.5 Pts] Show that $\hat{\delta}$ is an unbiased estimator for δ. In other words, prove that $\mathbb{E}[\hat{\delta}] = \delta$.

Note: $\bar{K}$ is an unbiased estimator of μ_K, and $\bar{D}$ is an unbiased estimator of μ_D. This result may be used directly in your calculations—you do not need to prove it.

Solution:

$$\mathbb{E}[\hat{\delta}] = \mathbb{E}[\bar{K} - \bar{D}] = \mathbb{E}[\bar{K}] - \mathbb{E}[\bar{D}] = \mu_K - \mu_D = \delta.$$

or

$$\mathbb{E}[\hat{\delta} - \delta] = \mathbb{E}[(\bar{K} - \bar{D}) - (\mu_K - \mu_D)] = \mathbb{E}[\bar{K} - \mu_K - (\bar{D} - \mu_D)] = \mathbb{E}[\bar{K}] - \mu_K - \mathbb{E}[\bar{D}] - \mu_D$$

$$= \mu_K - \mu_K - \mu_D - \mu_D = 0 - 0 = 0$$

(b) [1.5 Pts] Compute the variance of $\hat{\delta}$ in terms of σ_K^2, σ_D^2, n, and m.

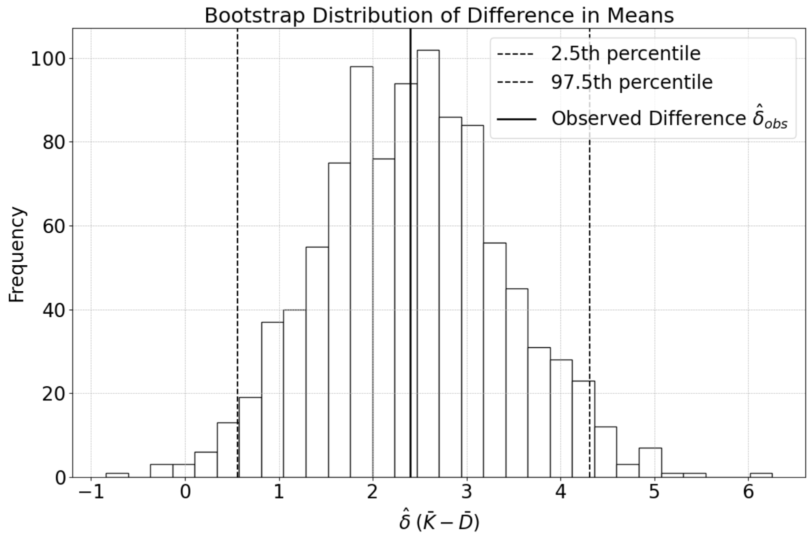
Solution:

$$\begin{aligned}
\mathrm{Var}(\hat{\delta}) &= \mathrm{Var}(\bar{K} - \bar{D}) \\
&= \mathrm{Var}(\bar{K}) + \mathrm{Var}(-\bar{D}) \quad \text{(since samples are independent)} \\
&= \mathrm{Var}(\bar{K}) + (-1)^2 \cdot \mathrm{Var}(\bar{D}) \\
&= \mathrm{Var}(\bar{K}) + \mathrm{Var}(\bar{D}) \\
&= \frac{\sigma_K^2}{n} + \frac{\sigma_D^2}{m}
\end{aligned}$$

Using the estimator in part (a), you compute the difference in sample means:

$$\hat{\delta}_{\text{obs}} = \bar{K} - \bar{D}$$

You estimate the variability in $\hat{\delta}_{\text{obs}}$ by bootstrapping the Kendrick and Drake samples. You record $\bar{K} - \bar{D}$ for each pair of bootstrap samples. Below is the bootstrap distribution. The null hypothesis is $\mu_K = \mu_D$.



(c) [1.5 Pts] Based on the plot, which statements are true?

○ True ○ **False** $\hat{\delta}_{\text{obs}}$ lies in the middle 95% of the bootstrap distribution, so we would fail to reject the null hypothesis at the 0.05 significance level.

○ **True** ○ False The bootstrap distribution shows the variability we might expect in $\bar{K} - \bar{D}$ across different random samples of the population.

○ True ○ **False** Because 0 is outside of the 95% confidence interval, we fail to reject the null hypothesis at the 0.05 significance level.

(d) [2 Pts] You stand outside of Warren Hall and survey UC Berkeley students one by one, recording 1 if they prefer Kendrick and 0 if they prefer Drake. Assume responses are independent and identically distributed (i.i.d.) with probability $p = \frac{1}{3}$ of **preferring Kendrick**.

What is the probability that the **fourth** student surveyed is the **first** Kendrick fan you observe (i.e., the first three students are Drake fans)? Show your work. You do not need to simplify your final answer.

Solution: The first three students must prefer Drake (0), and the fourth must prefer Kendrick (1):

$$(1-p)^3 \times p = \left(\frac{2}{3}\right)^3 \times \frac{1}{3} = \frac{8}{27} \times \frac{1}{3} = \frac{8}{81}$$

You now decide to predict whether a student prefers Kendrick or Drake using design matrix $\mathbb{X}$. This matrix has two features: $\mathbb{X}_{[:,1]}$ (hours studied per day) and $\mathbb{X}_{[:,2]}$ (number of music genres the student listens to).

(e) You build a logistic regression model using $\mathbb{X}_{[:,1]}$ and $\mathbb{X}_{[:,2]}$ to predict whether a student prefers Kendrick ($y_i = 1$) or Drake ($y_i = 0$). The model uses the following parameter vector:

$$\vec{\theta} = \begin{bmatrix} -2 & 2 & 1 \end{bmatrix}^\top$$

and follows the standard logistic regression form.

Suppose that for a particular student s, $\mathbb{X}_{[s,1]} = 1$ and $\mathbb{X}_{[s,2]} = 0$.

(i) [1.5 pts] Compute the probability $\hat{p}$ that student s prefers Kendrick.

Solution:

$$\hat{p} = \frac{1}{1 + e^{-(\theta_0 + \theta_1 \mathbb{X}_{[s,1]} + \theta_2 \mathbb{X}_{[s,2]})}} = \frac{1}{1 + e^{-(-2+2(1)+1(0))}} = \frac{1}{1+e^{-0}} = 0.5$$

(ii) [0.5 pts] Using a probability threshold of 0.3, which rapper would student s be classified as preferring?

- ◯ Drake
- ◯ **Kendrick**
- ◯ On the boundary

(iii) [1.5 pts] Derive the equation of the decision boundary using a probability threshold of 0.5. Express your answer in terms of $\mathbb{X}_{[:,1]}$ and $\mathbb{X}_{[:,2]}$.
Note: You may rearrange the equation so that constants are on either side. For example, $a^2 + b = 5$ would receive the same credit as $a^2 + b - 5 = 0$.

Solution:

$$\hat{p} = \frac{1}{1+e^{-z}} = 0.5 \quad \Rightarrow \quad z = 0, \quad \text{where } z = x^\top \vec{\theta}$$

$$-2 + 2\mathbb{X}_{[s,1]} + \mathbb{X}_{[s,2]} = 0 \quad \Rightarrow \quad 2\mathbb{X}_{[:,1]} + \mathbb{X}_{[:,2]} = 2$$

(f) To evaluate your logistic regression model, you test it on 6 new students and compare predicted versus true preferences. Results are shown below:

Student	$\hat{\mathbb{Y}}$	$\mathbb{Y}$
1	1	1
2	1	0
3	1	1
4	0	0
5	0	1
6	0	0

As a reminder, 1 indicates a preference for Kendrick, and 0 indicates a preference for Drake.

(i) [1 pt] What is the precision of the model on this data? Leave your answer as a fraction.

Solution: Precision = $\frac{\text{TP}}{\text{TP + FP}} = \frac{2}{3}$

(ii) [2 pt] You're considering adjusting the decision threshold of your logistic regression model. Currently, you predict "Kendrick" if the model output is above 0.5.

Which of the following statements is true if you lower the threshold to 0.3?

◯ True ◯ **False** It is possible that the model will predict "Kendrick" less often.

◯ **True** ◯ False The model recall will stay the same or increase.

◯ True ◯ **False** It is possible that the model's ROC-AUC will change.

◯ True ◯ **False** The precision will always stay the same or increase.

Solution: Lowering the threshold makes it easier for the model to predict "Kendrick" (1), so you'll get more positive predictions. This may increase recall (more actual Kendrick fans are correctly identified), but can also increase false positives (predicting Kendrick when it's actually Drake). False negatives are likely to decrease, since fewer Kendrick fans are missed.

4 Pandas' Pandas [9.5 Pts]

You and Oski use unsupervised learning to group pandas based on their features.

(a) [2 Pts] Oski considers using Principal Component Analysis (PCA). Select all statements that are true about PCA.

◯ **True** ◯ False PCA requires that Principal Component 2 (PC2) is orthogonal to PC1.

◯ **True** ◯ False The latent features constructed by PCA are linear combinations of the original features.

◯ True ◯ **False** You can compute the variance of the data in the PC1 direction using the U and V matrices from Singular Value Decomposition (SVD).

◯ **True** ◯ False A scree plot shows the proportion of total variance captured by each PC.

(b) Oski performs SVD on the `DataFrame` that contains the data about pandas, denoted X, producing matrices U, S, and V. Due to a corrupted message, you only receive S and the number of rows in X, which is 6.

$$S = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

(i) [0.5 pts] What is the rank of X?

Solution: 3, since there are 3 non-zero singular values.

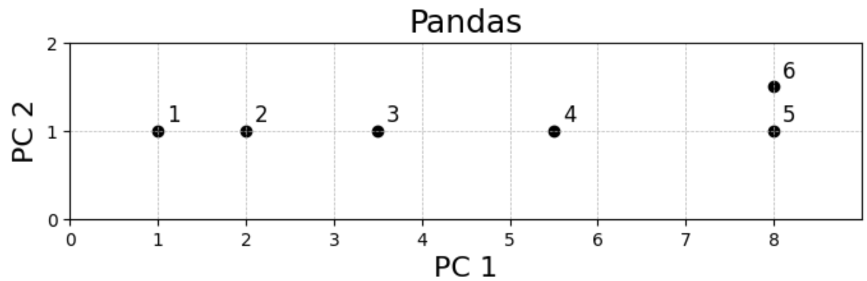
(ii) [1 pt] What **proportion** of the total variance is captured by PC1? You do not need to simplify your answer.

Solution: $\frac{10^2}{10^2+2^2+0.1^2} = \frac{100}{104.01}$

(iii) [1 pt] What is the variance of the data in the PC2 direction? You do not need to simplify your answer.

Solution: $\frac{2^2}{6} = \frac{4}{6} = \frac{2}{3}$

After applying PCA, you keep the top two PCs for clustering. The scatter plot below shows each panda as a point in the two-dimensional PC space:



You want to explore whether meaningful clusters exist in this dataset.

(c) You use hierarchical clustering to group pandas based on their positions in the two-dimensional PC space. Each point (labeled 1 - 6) represents a panda. Start with each panda in its own cluster and use the distance between pandas to measure similarity.

Each subpart below is independent.

(i) [1 pt] **After 3 steps** of agglomerative clustering using **single linkage**, which pandas are in the same cluster as Panda 4? Select all that apply.

☐ 3 ☐ 5 ☐ 6 ☐ **None, Panda 4 remains unmerged with other pandas.**

Solution: After 3 steps using single linkage, these are the clusters: {1, 2, 3}{4}{5, 6}. So Panda 4 is a lonely panda ☹

(ii) [1 pt] **After 3 steps** of agglomerative clustering using **complete linkage**, which panda remains in its own singleton cluster (not merged with any other panda)?

◯ 1
◯ 2
◯ 3
◯ 4
◯ 5
◯ 6
◯ **None; all pandas have at least one other panda in their cluster**

Solution: After 3 steps using complete linkage, these are the clusters: {1, 2}{3, 4}{5, 6}. So there are no lonely pandas ☺

(iii) [1 pt] How many iterations are needed to complete agglomerative clustering using **average linkage** so that all pandas are in one big cluster?

Number of iterations needed: _______________

Solution: If there are $n = 6$ pandas initially, then agglomerative clustering requires exactly $n - 1 = \boxed{5}$ iterations to merge everything into a single cluster.

(d) [2 Pts] Which of the following statements are true about clustering?

○ True ○ **False** For a fixed dataset and a fixed value of K, K-means clustering always produces the same final cluster assignments, regardless of the initial placement of centroids.

○ True ○ **False** K-Means clustering always converges to the global optimum solution.

○ True ○ **False** K-Means optimizes the silhouette score.

○ **True** ○ False K-Means requires specifying the number of clusters in advance.

You are done with the final—Congratulations!

Draw your favorite DATA 100/200 memory!