Data C100/200 - Final

Fall 2025

Write your name BIG and clearly: ______________________________

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 **68 points** spread out over **7 questions**. The exam must be completed in **170 minutes** unless you have accommodations supported by a DSP letter.

- Note that some questions have circular bubbles to select a choice. Please shade in the circle fully to mark your answer.
- **Write clearly and legibly.** We reserve the right to withhold points from answers that are very difficult to read.
- Blank answers and incorrect answers are graded identically, so it's in your best interest to answer every question.
- **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:

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.

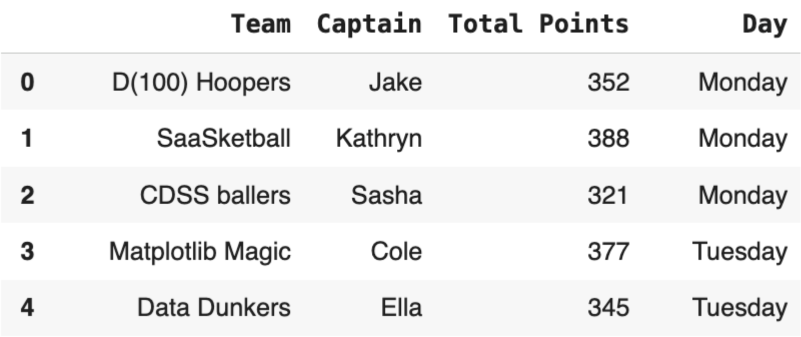
1 D(100) Hoopers [30 Pts]

Jake has been having a rough semester in intramural (IM) basketball. To gain a competitive advantage, he wants to understand the **performance of other IM basketball teams during the Fall 2025 semester**. So, he sends a survey to the **team captain of each IM basketball team** that registered to play during the **Fall 2025 semester** at UC Berkeley.

Jake collects and stores the survey results in a `DataFrame` called `survey`. Here are the columns of `survey`:

- `Team`: the unique name of the team (type = `String`)
- `Captain`: the name of the team captain that completed the survey (type = `String`)
- `Total Points`: the total points scored by the team throughout the semester (type = `int`)
- `Day`: the day of the week that the team plays (type = `String`)

The first five rows of `survey` are shown below:



	Team	Captain	Total Points	Day
0	D(100) Hoopers	Jake	352	Monday
1	SaaSketball	Kathryn	388	Monday
2	CDSS ballers	Sasha	321	Monday
3	Matplotlib Magic	Cole	377	Tuesday
4	Data Dunkers	Ella	345	Tuesday

(a) Answer the following questions about Jake's survey.

(i) [1 Pt] What is the population of interest in Jake's survey?

○ **All** IM sport captains who registered their team during **Fall 2025**

○ **Basketball IM team captains who registered their team during Fall 2025.**

○ **Basketball** IM team captains who registered their team during **Fall 2025** AND who responded to Jake's survey.

○ **Basketball** IM team captains who registered their team during **any** semester

(ii) [1 Pt] What is the sampling frame of Jake's survey?

○ **All** IM sport captains who registered their team during **Fall 2025**

○ **Basketball IM team captains who registered their team during Fall 2025.**

○ **Basketball** IM team captains who registered their team during **Fall 2025** AND who responded to Jake's survey.

○ **Basketball** IM team captains who registered their team during **any** semester

(iii) [1 Pt] What is the sample in Jake's survey?

- ○ **All** IM sport captains who registered their team during **Fall 2025**
- ○ **Basketball** IM team captains who registered their team during **Fall 2025**.
- ○ **Basketball IM team captains who registered their team during Fall 2025 AND who responded to Jake's survey.**
- ○ **Basketball** IM team captains who registered their team during **any** semester

(iv) [1 Pt] Which of the following statements about Jake's survey is true?

○ True ○ **False** Jake only surveyed captains who registered for the Fall 2025 semester, and no one from a prior semester, so there is **selection bias**.

○ **True** ○ False If some team captains exaggerate their team's achievements, there is **response bias**.

○ **True** ○ False If some team captains decide not to complete the survey, there is **non-response bias**.

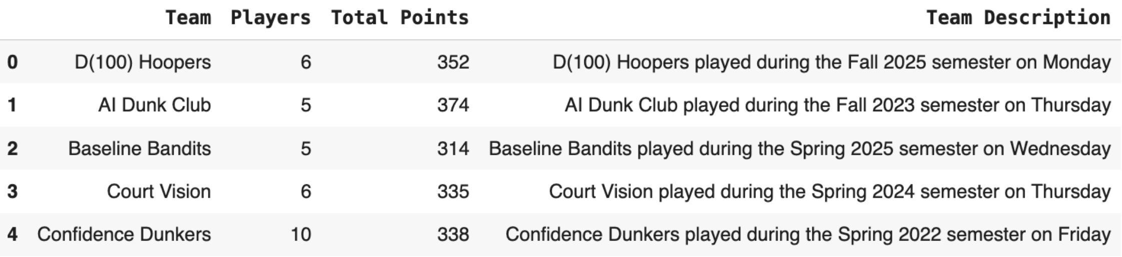
○ **True** ○ False If all Fall 2025 IM basketball team captains complete the survey, there is no **chance error** introduced from sampling.

(b) Jake looks for other sources of information about IM basketball. He finds a `DataFrame`, called `basketball`, that contains information on **all** UC Berkeley IM Basketball teams that have registered from **2020 to the present**.

Here are the columns of `basketball`:

- `Team`: the name of the team. **Teams can only play during one semester. Team names are unique across semesters.** (type = `String`)
- `Players`: the number of players in the team (type = `int`)
- `Total Points`: the total points scored by the team (type = `int`)
- `Team Description`: A description of the team. (type = `String`)

The first five rows of `basketball` are shown below:



	Team	Players	Total Points	Team Description
0	D(100) Hoopers	6	352	D(100) Hoopers played during the Fall 2025 semester on Monday
1	AI Dunk Club	5	374	AI Dunk Club played during the Fall 2023 semester on Thursday
2	Baseline Bandits	5	314	Baseline Bandits played during the Spring 2025 semester on Wednesday
3	Court Vision	6	335	Court Vision played during the Spring 2024 semester on Thursday
4	Confidence Dunkers	10	338	Confidence Dunkers played during the Spring 2022 semester on Friday

(i) [1 Pt] What is the granularity of `basketball`? **Answer in at most one sentence.**

Solution:
Each row represents a team that registered for IM basketball in any semester since 2020.

(ii) [2 Pts] What is the variable type of the column `Total Points`?

○ Qualitative Nominal

○ Qualitative Ordinal

○ [✓ **Quantitative**]

(iii) [1 Pt] What is the variable type of the column `Team`?

○ [✓ **Qualitative Nominal**]

○ Qualitative Ordinal

○ Quantitative

(iv) [2 Pts] Jake creates a new `Semester` column that contains the semester mentioned in the `Team Description` column. Jake writes the following skeleton code:

```
basketball['Semester'] =
    basketball['Team Description'].str.extract(__(iv)__)
```

Which of the following regex patterns correctly fills the blank `(iv)` to create the `Semester` column? For example, your chosen regex pattern should extract the text "`Fall␣2025`" from the example below:

```
"D(100)␣Hoopers␣played␣during␣the␣Fall␣2025␣semester␣on␣Monday"
```

Note: The special character ␣ refers to a single space.

Assumptions: All entries in the `Team Description` column follow the format below. No team name contains the words `Spring` or `Fall`. IM basketball does not happen during the summer semester.

```
{Team_Name}␣played␣during␣the␣{Fall/Spring}␣{Year}␣semester␣on␣{Day}
```

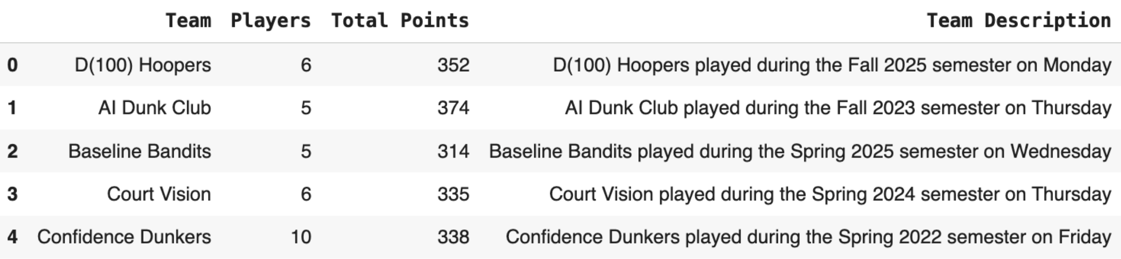
○ True [✓ ○ **False**] `r"([FallSpring]␣\d+)"`
○ True [✓ ○ **False**] `r"(Fall|Spring)\s\d{4}"`
[✓ ○ **True**] ○ False `r"during␣the␣(\w{,6}␣\d{4})␣semester"`
[✓ ○ **True**] ○ False `r"(Fall␣\d*|Spring␣\d*)"`

(v) [2 Pts] Assume that the `Semester` column has been correctly added to `basketball`. Write code that outputs the proportion of rows in `basketball` that refer to a team registered in **Fall 2025**. Your code can return a `DataFrame` with one row, a `Series` with one element, or a single number.

Solution:

```
basketball[basketball['Semester'] == 'Fall 2025']
    .shape[0] / basketball.shape[0]
```

The first five rows of `basketball` are shown below:



	Team	Players	Total Points	Team Description
0	D(100) Hoopers	6	352	D(100) Hoopers played during the Fall 2025 semester on Monday
1	AI Dunk Club	5	374	AI Dunk Club played during the Fall 2023 semester on Thursday
2	Baseline Bandits	5	314	Baseline Bandits played during the Spring 2025 semester on Wednesday
3	Court Vision	6	335	Court Vision played during the Spring 2024 semester on Thursday
4	Confidence Dunkers	10	338	Confidence Dunkers played during the Spring 2022 semester on Friday

(vi) [1 Pt] Jake creates a new `Day` column with the day of the week that each team played. Jake writes the following skeleton code:

```
basketball['Day'] =
    basketball['Team Description'].str.extract(r"__(v)__")
```

Write a regex pattern that correctly fills the blank `(v)` to create the `Day` column. For example, your chosen regex pattern should extract the text "`Monday`" from this text:

`"D(100)␣Hoopers␣played␣during␣the␣Fall␣2025␣semester␣on␣Monday"`

Assumptions: All entries in the `Team Description` *column follow the format below. No team name contains the words* `Spring` *or* `Fall`*. IM basketball does not happen during the summer semester.*

`{Team_Name}␣played␣during␣the␣{Fall/Spring}␣{Year}␣semester␣on␣{Day}`

Write your regex pattern in the box below. **Hint: Make sure your regex works for all cases. For example, consider the team name "Winners on Wednesday".**

Solution: `(\w+$)`

(vii) [1 Pt] Suppose Jake correctly creates the `Day` column described in the previous question. Jake wants to know which **days** of the week have the **most teams** playing. Which of the following line(s) of code returns a `Series` with the number of teams that have played during each day, across all semesters in `basketball`?

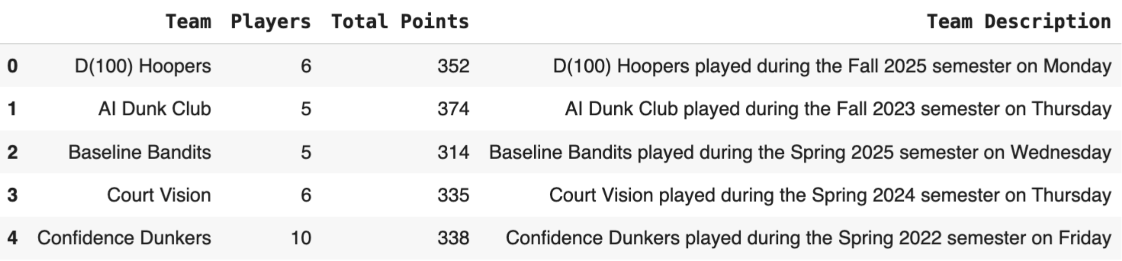
○ **True** ○ False `basketball['Day'].value_counts()`

○ **True** ○ False `basketball.groupby('Day')['Team'].size()`

○ True ○ **False** `basketball.groupby('Team')['Day'].size()`

○ **True** ○ False `basketball.groupby('Day').size()`

The first five rows of the unmodified `basketball DataFrame` are shown below:



	Team	Players	Total Points	Team Description
0	D(100) Hoopers	6	352	D(100) Hoopers played during the Fall 2025 semester on Monday
1	AI Dunk Club	5	374	AI Dunk Club played during the Fall 2023 semester on Thursday
2	Baseline Bandits	5	314	Baseline Bandits played during the Spring 2025 semester on Wednesday
3	Court Vision	6	335	Court Vision played during the Spring 2024 semester on Thursday
4	Confidence Dunkers	10	338	Confidence Dunkers played during the Spring 2022 semester on Friday

(viii) [2 Pts] Jake wants to visualize the frequency of each unique value of the `Semester` column. Recall that the `Semester` column was created in a previous part. Which of the following are appropriate visualizations?

○ True [✓ ○ **False**] Histogram

○ True [✓ ○ **False**] Box plot

○ True [✓ ○ **False**] Scatter plot

[✓ ○ **True**] ○ False Bar plot

○ True [✓ ○ **False**] KDE plot

○ True [✓ ○ **False**] Hex plot

(ix) [2 Pts] Jake also wants to look at the distribution of the values in the `Total Points` column. Which of the following are appropriate visualizations?

[✓ ○ **True**] ○ False Histogram

[✓ ○ **True**] ○ False Box plot

○ True [✓ ○ **False**] Scatter plot

○ True [✓ ○ **False**] Bar plot

[✓ ○ **True**] ○ False KDE plot

○ True [✓ ○ **False**] Hex plot

(c) Jake wants to figure out which day of the week is the easiest day to compete. He thinks that the easiest day is the day with the **lowest average** `Total Points` scored across all teams.

Complete the code below to return a `DataFrame` that has the average `Total Points` for each `Day` sorted in **ascending** order of the average `Total Points`.

```
basketball.____(i)____(____(ii)____)[["Total Points"]]
    .agg(____(iii)____).____(iv)____(____(v)____)
```

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

Solution: `groupby`

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

Solution: `"Day"`

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

Solution: `"mean"` or `np.mean`

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

Solution: `sort_values`

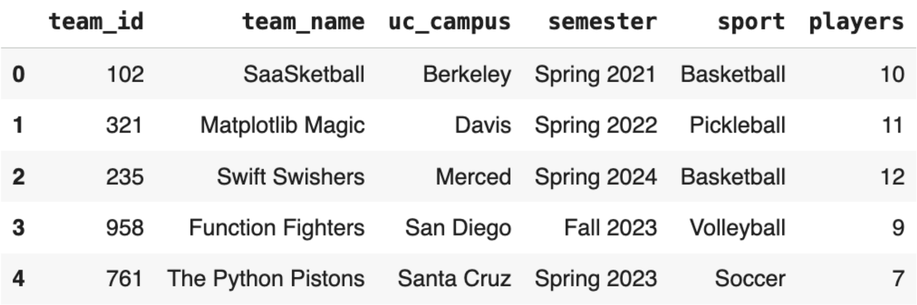
(v) [1 Pt] Fill in blank `(v)`:

Solution: `"Total Points"`

(d) Willy wants to see how Berkeley IM sports compare to IM sports at other UC campuses. He finds a public SQL Database with all the IM teams **across all UC campuses since 2020**. The database contains a table `teams`. Here are the columns of `teams`:

- `team_id`: the primary key of the table (type = `int`)
- `team_name`: the name of the team. At each UC campus, teams can only play for one semester and team names are unique across semesters. **But, team names are not necessarily unique across UC campuses.** (type = `String`)
- `uc_campus`: the UC Campus that the team is from (type = `String`)
- `semester`: the semester that this team played. You can assume that all UC campuses have only the Fall and Spring semester for IM sports (type = `String`)
- `sport`: the sport that this team played (type = `String`)
- `players`: the number of players in the team (type = `int`)

The first five rows of `teams` are shown below:



	team_id	team_name	uc_campus	semester	sport	players
0	102	SaaSketball	Berkeley	Spring 2021	Basketball	10
1	321	Matplotlib Magic	Davis	Spring 2022	Pickleball	11
2	235	Swift Swishers	Merced	Spring 2024	Basketball	12
3	958	Function Fighters	San Diego	Fall 2023	Volleyball	9
4	761	The Python Pistons	Santa Cruz	Spring 2023	Soccer	7

Write an SQL query that returns a table with the **number of IM basketball teams** at each UC campus, sorted by the number of IM basketball teams in **descending order**. There should be two columns (`uc_campus` and `num_teams`).

Fill in the blanks below. Assume that `duckdb` is imported and `teams` can be queried as an SQL table.

```
SELECT _____(i)_____
FROM teams
WHERE _____(ii)_____
GROUP BY _____(iii)_____
ORDER BY _____(iv)_____;
```

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

Solution: `teams.uc_campus, COUNT(*) AS num_teams`

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

Solution: `teams.sport = 'Basketball'`

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

Solution: `teams.uc_campus`

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

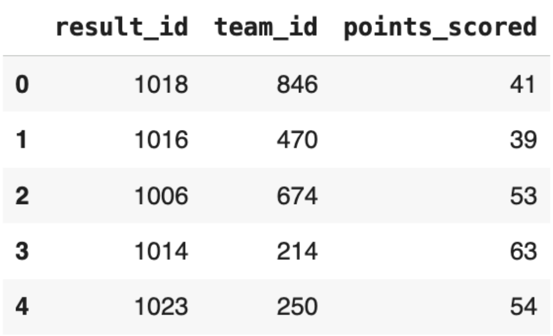
Solution: `num_teams DESC`
(or `COUNT(*) DESC`)

(e) The database also contains a table called results that summarizes the performances of these teams.

Here are the columns of results:

- result_id: the primary key of the table (type = int)
- team_id: a foreign key referring to the team_id in the teams table (type = int)
- points_scored: the amount of points the team scored for that game (type = int).

The first five rows of results are shown below:

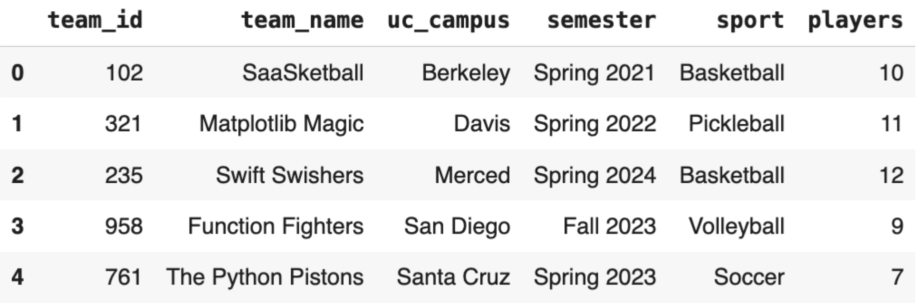


	result_id	team_id	points_scored
0	1018	846	41
1	1016	470	39
2	1006	674	53
3	1014	214	63
4	1023	250	54

Willy wants to determine which UC campus has the weakest IM Basketball teams. Help Willy write an SQL query that returns a table with the following conditions:

1. Each row of the table has the name of a UC Campus and the sum of all points scored across all games played at that UC Campus that meet the conditions in (2). The name of the columns should be uc_campus and total_points, respectively.
2. The table should only include UC campuses where the average points scored by each team in every game is at least 25. The table should only include points scored by IM Basketball teams (i.e., no other sports).
3. The table should be ordered by total_points in ascending order.

Here's teams again, for reference:



	team_id	team_name	uc_campus	semester	sport	players
0	102	SaaSketball	Berkeley	Spring 2021	Basketball	10
1	321	Matplotlib Magic	Davis	Spring 2022	Pickleball	11
2	235	Swift Swishers	Merced	Spring 2024	Basketball	12
3	958	Function Fighters	San Diego	Fall 2023	Volleyball	9
4	761	The Python Pistons	Santa Cruz	Spring 2023	Soccer	7

Fill in the blanks below. Assume that `duckdb` is imported and `teams` and `results` can be queried as SQL tables.

```
SELECT _____(i)_____
FROM teams AS t
INNER JOIN results AS r
    ON _____(ii)_____
WHERE _____(iii)_____
GROUP BY _____(iv)_____
HAVING _____(v)_____
ORDER BY _____(vi)_____;
```

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

Solution: `t.uc_campus, SUM(r.points_scored) AS total_points`

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

Solution: `t.team_id = r.team_id`

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

Solution: `t.sport = 'Basketball'`

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

Solution: `t.uc_campus`

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

Solution: `AVG(r.points_scored) >= 25`

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

Solution: `SUM(r.points_scored) ASC` or `total_points ASC`

2 This is BEAR TERRITORY! [7 Points]

A team of researchers is studying the football game attendance of UC Berkeley students. In the study, the team collects a simple random sample of **100 students** from all UC Berkeley students. The team collects the following data for each student:

- X_i: The number of home football games attended by student i, for $i = 1, 2, \ldots, 100$.

For the purposes of this question, you can assume that the X_i's are independent and identically distributed. The researchers use $X_1 ... X_{100}$ to estimate properties of the distribution of the number of home football games attended by **all UC Berkeley students.**

(a) We define an estimator $\hat{\alpha}$, which is the sample mean:

$$\hat{\alpha} \ = \ \frac{1}{100}\sum_{i=1}^{100} X_i.$$

Let $\alpha = \mathbb{E}[X_i]$ be the true average number of home games attended by UC Berkeley students.

(i) [2 Pts] **Calculate** Bias$(\hat{\alpha})$. Your answer can be in terms of α. Show your work.
Recall: Bias$(\hat{\alpha}) = \mathbb{E}[\hat{\alpha}] - \alpha$.

Solution: We are given the estimator

$$\hat{\alpha} = \frac{1}{100}\sum_{i=1}^{100} X_i,$$

and we are told that each X_i has population mean $\mathbb{E}[X_i] = \alpha$.

$$\text{Bias}(\hat{\alpha}) = \mathbb{E}[\hat{\alpha}] - \alpha.$$

$$\mathbb{E}[\hat{\alpha}] = \mathbb{E}\left(\frac{1}{100}\sum_{i=1}^{100} X_i\right) = \frac{1}{100}\sum_{i=1}^{100}\mathbb{E}[X_i] = \frac{1}{100}\cdot(100\alpha) = \alpha.$$

Thus, the bias is

$$\text{Bias}(\hat{\alpha}) = \mathbb{E}[\hat{\alpha}] - \alpha = \alpha - \alpha = 0.$$

$$\boxed{\text{Bias}(\hat{\alpha}) = 0}$$

Now suppose that

$$\mathrm{Var}(X_i) = \sigma^2 \quad \text{for all } i = 1, 2, \ldots, 100.$$

Recall that our estimator is

$$\hat{\alpha} \ = \ \frac{1}{100} \sum_{i=1}^{100} X_i.$$

(ii) [2 Pts] **Calculate** $\mathrm{Var}(\hat{\alpha})$. Your answer can be in terms of σ^2. Show your work.

Solution: We have

$$\hat{\alpha} = \frac{1}{100} \sum_{i=1}^{100} X_i.$$

Using the properties of variance and the fact that the X_i's are independent:

$$\mathrm{Var}(\hat{\alpha}) = \mathrm{Var}\left(\frac{1}{100} \sum_{i=1}^{100} X_i \right) = \left(\frac{1}{100} \right)^2 \sum_{i=1}^{100} \mathrm{Var}(X_i).$$

Since each $\mathrm{Var}(X_i) = \sigma^2$,

$$\mathrm{Var}(\hat{\alpha}) = \left(\frac{1}{100} \right)^2 \cdot 100\sigma^2 = \frac{\sigma^2}{100}.$$

$$\boxed{\mathrm{Var}(\hat{\alpha}) = \frac{\sigma^2}{100}}$$

(b) Answer the following questions about bootstrapping. Assume that we are bootstrapping from the sample of 100 students in the previous problem.

(i) [0.5 Pts] The main purpose of bootstrapping is to estimate the **average** of the data points in the **original sample** of 100 students.

○ True ○ **False**

(ii) [0.5 Pts] The main purpose of bootstrapping is to estimate the **variance** of the data points in the **original sample** of 100 students.

○ True ○ **False**

(iii) [0.5 Pts] To generate a bootstrapped confidence interval of the sample mean using 10,000 bootstrapped samples of size 100, we must calculate the variance of the 100 data points in each of the 10,000 bootstrapped samples. So, we calculate 10,000 synthetic values of the sample variance.

○ True ○ **False**

(iv) [0.5 Pts] Suppose we increase the number of bootstrap samples from 10,000 to 100,000, with no changes to our original sample. It is very likely that this action will substantially **reduce** the width of a bootstrapped confidence interval of the average number of games attended by UC Berkeley students.

○ True ○ **False**

(v) [0.5 Pts] Suppose we construct the bootstrap distribution of the **median** number of games attended by UC Berkeley students. We generate 10,000 bootstrap samples of size 100. This distribution would almost certainly be approximately symmetric.

○ **True** ○ False

(vi) [0.5 Pts] Suppose we construct the bootstrap distribution of the **maximum** number of games attended by any UC Berkeley student. We generate 10,000 bootstrap samples of size 100. This distribution would almost certainly be approximately symmetric.

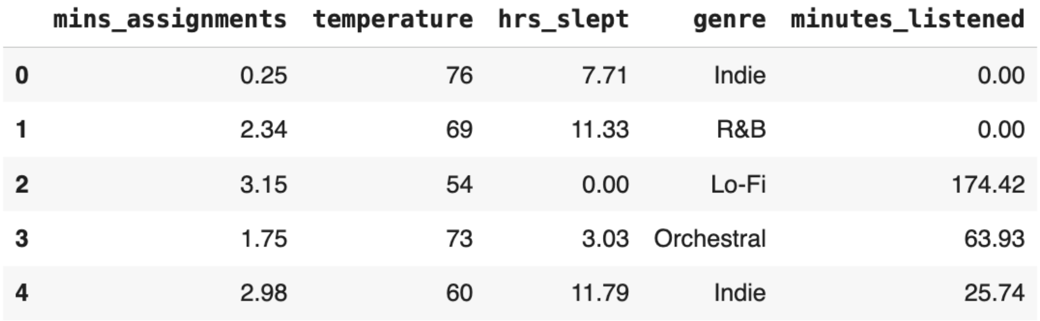
○ True ○ **False**

3 ...and that's a Wrap on Regression [14 Pts]

Rohan has only one Spotify playlist that is 67 (?!) hours long! Despite this, Sarika has reason to believe that Rohan has been fabricating his 2025 Spotify Wrapped! Sarika decides that she wants to predict the number of minutes that Rohan actually listened to music each day. To do this, she uses data that Rohan's roommates collected throughout the Fall semester. She combines the data into the `spotify_stats` DataFrame. The columns of `spotify_stats` are:

- `mins_assignments`: Minutes spent completing assignments per day (type = `float`)
- `temperature`: Daily high temperature (degrees Fahrenheit) in Berkeley (type = `int`)
- `hrs_slept`: Hours spent sleeping each night (type = `float`)
- `genre`: Genre that Rohan planned to listen to today (one of: R&B, Pop, Lo-Fi, Orchestral, Indie) (type = `String`)
- `minutes_listened`: Minutes spent listening to music on Spotify (type = `float`)

The first five rows of `spotify_stats` are shown below:



	mins_assignments	temperature	hrs_slept	genre	minutes_listened
0	0.25	76	7.71	Indie	0.00
1	2.34	69	11.33	R&B	0.00
2	3.15	54	0.00	Lo-Fi	174.42
3	1.75	73	3.03	Orchestral	63.93
4	2.98	60	11.79	Indie	25.74

(a) [3 Pts] Sarika considers several candidate models for predicting Rohan's listening time. **Which of the following models are linear in $\vec{\theta}$?**

[✓ ○ **True**] ○ False $y = \theta_0$

○ True [✓ ○ **False**] $y = \theta_0 + \theta_1\theta_2 x_1$

[✓ ○ **True**] ○ False $y = \theta_0 + \theta_1 x_1^2$

[✓ ○ **True**] ○ False $y = \theta_0 + \theta_1 \sin(x_1)$

○ True [✓ ○ **False**] $y = \theta_0 + \sin(\theta_1) x_1$

○ True [✓ ○ **False**] $y = \theta_0 + \theta_1 \log(x_1) + (\theta_2)^2 x_2$

(b) [2 Pts] Sarika notices that Rohan didn't listen to Spotify at all on several days. These low-valued outliers cause her original model to underpredict the minutes listened. She aims to design a custom loss function that helps address this issue.

Assume Sarika uses this model:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

Which of these loss functions would do the best job of **minimizing the effect of outliers** on the fitted model parameters?

○ $L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$

○ [✓ $L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$]

○ $L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

○ $L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^4$

(c) Sarika decides to one-hot encode the `genre` column. When she tries to compute the optimal OLS parameters for her model via the normal equation, she is unable to get a unique solution!

(i) [2 Pts] Which of the following statements **could explain** why this happened?

○ [✓ **True**] ○ False The matrix $\mathbb{X}$ does not have full column rank. So, its columns are linearly dependent.

○ [✓ **True**] ○ False In the design matrix, Sarika included a bias column and did not remove a reference category from the one-hot encoded columns.

○ [✓ **True**] ○ False There is perfect multicollinearity in the features.

○ True ○ [✓ **False**] The dataset has more rows than columns, so the design matrix is not full column rank.

○ True ○ [✓ **False**] The matrix $\mathbb{X}^T\mathbb{X}$ is invertible, but the dataset is noisy, so there are many possible OLS solutions.

(ii) [1 Pt] After fixing the issue in the previous part, Sarika generates unique OLS predictions $\hat{\mathbb{Y}}$. Which of the following is true?

○ True ○ [✓ **False**] The residual vector $\mathbb{Y} - \hat{\mathbb{Y}}$ has the same dimension as the parameter vector $\boldsymbol{\theta}$.

○ [✓ **True**] ○ False The predictions $\hat{\mathbb{Y}}$ are the projection of $\mathbb{Y}$ onto the span of $\mathbb{X}$.

○ [✓ **True**] ○ False If $\mathbb{X}$ is full column rank, there is exactly one $\hat{\theta}$ vector that minimizes the mean squared error.

○ [✓ **True**] ○ False The prediction vector $\hat{\mathbb{Y}}$ is a linear combination of the columns of $\mathbb{X}$.

(d) [1 Pt] To improve the model, Samion suggests implementing **Regularization**. The model becomes:

$$L(\boldsymbol{\theta}) = \|\mathbb{Y} - \mathbb{X}\boldsymbol{\theta}\|^2 + \lambda R(\boldsymbol{\theta})$$

where $R(\boldsymbol{\theta})$ is a regularization term.

Which of the following statements about regularization are correct?

○ True ○ **False** Regularization helps to reduce overfitting by penalizing parameter values that are too small.

○ True ○ **False** Regularization guarantees a lower training error for any $\lambda > 0$, compared to an unregularized model.

○ **True** ○ False Increasing λ typically increases model bias squared but decreases model variance.

○ True ○ **False** Regularization can only be used if $\mathbb{X}^T\mathbb{X}$ is invertible.

(e) [1 Pt] Kelly points out that when she uses regularization, Sarika should pick the λ that produces the lowest mean squared error on validation data. Sarika decides to use **4-fold cross-validation** (CV) to find the best performing value of λ out of **6 possible options**.

To determine the best-performing value of λ out of 6 possible options using 4-fold CV, how many times does Sarika need to calculate the **Mean Squared Error (MSE)** on a held-out validation set? Show your work.

Solution: One validation MSE is computed per fold, so there are **4 total MSE values** per λ.

Since they test 6 different λ values, the total number of MSE values is

$$4 \text{ folds} \times 6 \text{ values of } \lambda = \boxed{24 \text{ total MSE values.}}$$

(f) Collin suggests that Sarika try a completely different loss function to predict minutes listened using two features:

- x_{i1}: minutes spent on assignments on day i
- x_{i2}: temperature on day i

They define the following regularized loss function:

$$L(\theta_1, \theta_2) = \frac{1}{2n} \sum_{i=1}^{n} \Big(y_i - \theta_1 x_{i1} - \theta_2 x_{i2} \Big)^2 + \frac{\lambda}{2} \big(\theta_1^2 + \theta_2^2 \big).$$

Sarika decides to minimize this loss using gradient descent.

She starts from $\theta_1^{(0)} = 0$, $\theta_2^{(0)} = 0$, uses a learning rate $\alpha = 0.1$, and a regularization term $\lambda = 0.5$. **Sarika uses new data to fit this model**:

$$\mathbb{X} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \\ 3 & 1 \end{bmatrix}, \quad \mathbb{Y} = \begin{bmatrix} 25 \\ 10 \\ 5 \end{bmatrix}$$

(i) [2 Pts] Derive $\dfrac{\partial L}{\partial \theta_1}$. Show your work.

Solution:

For θ_1:

$$\frac{\partial L}{\partial \theta_1} = -\frac{1}{n} \sum_{i=1}^{n} \Big(y_i - \theta_1 x_{i1} - \theta_2 x_{i2} \Big) x_{i1} + \lambda \theta_1.$$

(ii) [2 Pts] Use gradient descent to calculate $\theta_1^{(1)}$. Show your work.

Repeated for reference:

$\theta_1^{(0)} = 0$, $\theta_2^{(0)} = 0$, $\alpha = 0.1$, and $\lambda = 0.5$.

$$\mathbb{X} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \\ 3 & 1 \end{bmatrix}, \quad \mathbb{Y} = \begin{bmatrix} 25 \\ 10 \\ 5 \end{bmatrix}$$

Solution: At $t = 0$, we have $\theta_1^{(0)} = 0$, $\theta_2^{(0)} = 0$, so $\hat{y}_i^{(0)} = 0$ and hence $y_i - \hat{y}_i^{(0)} = y_i$. Using the formula from part (i) with $\theta_1^{(0)} = 0$:

$$\left.\frac{\partial L}{\partial \theta_1}\right|_{(0,0)} = -\frac{1}{n}\sum_{i=1}^{n} y_i x_{i1} + \lambda \cdot 0 = -\frac{1}{3}\sum_{i=1}^{3} y_i x_{i1}.$$

Compute the sum using the dataset:

$$\sum_{i=1}^{3} y_i x_{i1} = (25)(1) + (10)(2) + (5)(3) = 25 + 20 + 15 = 60.$$

Therefore,

$$\left.\frac{\partial L}{\partial \theta_1}\right|_{(0,0)} = -\frac{1}{3} \cdot 60 = -20.$$

Now apply the gradient descent update with $\alpha = 0.1$:

$$\theta_1^{(1)} = \theta_1^{(0)} - \alpha \left.\frac{\partial L}{\partial \theta_1}\right|_{(0,0)} = 0 - 0.1(-20) = 2.$$

So after one step of gradient descent,

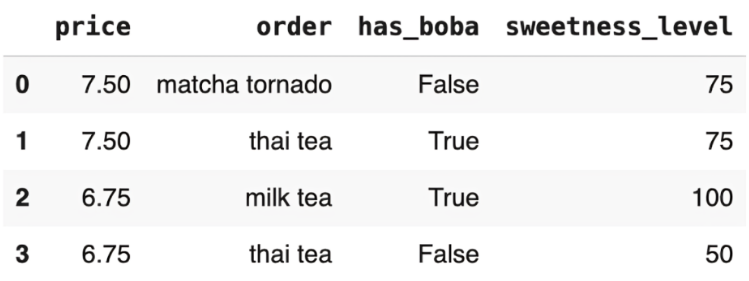
$$\boxed{\theta_1^{(1)} = 2.}$$

4 Spill The Tea [3 Pts]

At the most recent staff meeting, Kevin notices that most of the staff members purchased tea from the Data 100 coffee shop. He compiles his data in `staff_orders`, a `DataFrame` where each row contains:

- `price`: The price of each order. (type = `float`)
- `order`: The name of the beverage ordered. (type = `str`)
- `has_boba`: Whether the drink contains boba. (type = `bool`)
- `sweetness_level`: Sugar percentage added to the drink. (type = `int`)

A sample of `staff_orders` has been pasted below:



	price	order	has_boba	sweetness_level
0	7.50	matcha tornado	False	75
1	7.50	thai tea	True	75
2	6.75	milk tea	True	100
3	6.75	thai tea	False	50

(a) [2 Pts] Kevin first considers **OLS Model 1**, which predicts `price` using `has_boba` and `sweetness_level` as features. He also considers **OLS Model 2**, which predicts `price` using `order`, `has_boba`, and `sweetness_level`. Neither model is regularized.

Assuming Model 1 and Model 2 will be fit to the same data, which of the following is guaranteed to be true?

○ **True** ○ False (squared bias of Model 2) $\leq$ (squared bias of Model 1)

○ True ○ **False** (variance of Model 2) $\leq$ (variance of Model 1)

○ **True** ○ False (irreducible error of Model 2) = (irreducible error of Model 1)

○ True ○ **False** (squared bias of Model 2 + variance of Model 2) $\leq$ (squared bias of Model 1 + variance of Model 1)

(b) [1 Pt] Recall that **OLS Model 2** uses `order`, `has_boba`, and `sweetness_level` as features, and is not regularized. To create **Model 3**, Kevin adds LASSO regularization to **OLS Model 2**, with no other changes. Assuming Model 2 and Model 3 will be fit to the same data, which of the following is guaranteed to be true?

○ **True** ○ False (squared bias of Model 2) $\leq$ (squared bias of Model 3)

○ True ○ **False** (variance of Model 2) $\leq$ (variance of Model 3)

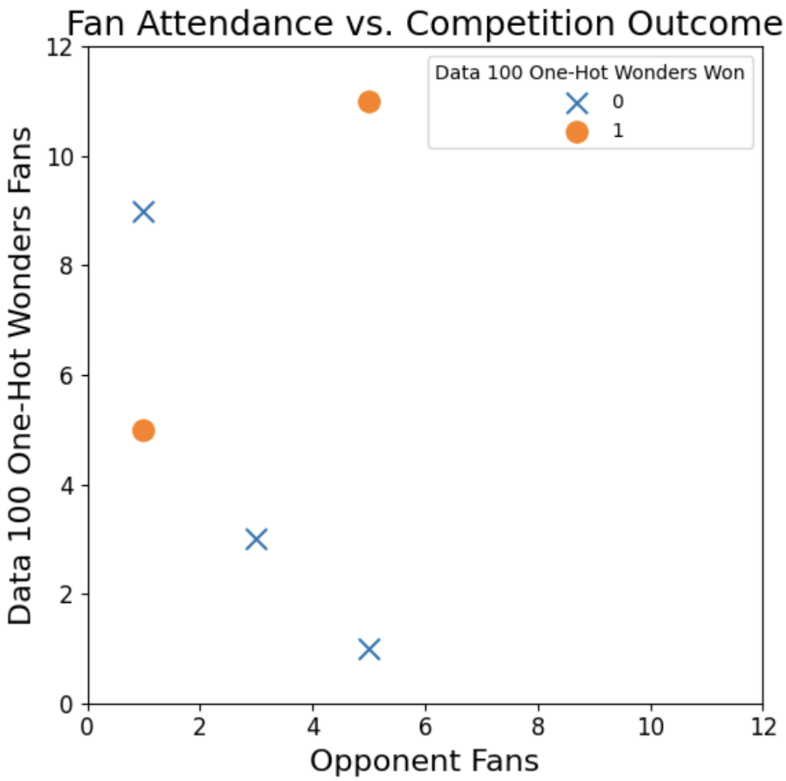
○ **True** ○ False (irreducible error of Model 2) = (irreducible error of Model 3)

○ True ○ **False** (squared bias of Model 2 + variance of Model 2) $\leq$ (squared bias of Model 3 + variance of Model 3)

5 CLASSIFIED: Top Secret Smooth Moves [7 Pts]

The Data 100 Dance Team, the One-Hot Wonders, finished off a historic dance battle season. Gisella suspects that the team's performance is better when there are more fans in attendance at the beginning of the competition.

(a) [2 Pts] Gisella collects data from 5 competitions this past season, logging the number of fans for the One-Hot Wonders, the number of fans for the opposing team, and whether the One Hot Wonders won the competition.



What is the maximum possible accuracy a logistic regression model can achieve on this dataset? **Justify your answer with one sentence.**

Solution: The maximum accuracy we can achieve with this dataset using logistic regression is $4/5 = 80\%$.

The dataset is not linearly separable so we cannot achieve 100% accuracy. The most accurate predictor we can build is a line separating one win from the other four datapoints. This model classifies three of the losses correctly and one win correctly, but incorrectly classifies one win.

(b) [2 Pts] Gisella fits a logistic regression model to the data she collected. Gisella **includes an intercept** in her model and **does not use regularization**. Gisella uses the number of opponent fans as the first feature in her model (x_1) and the number of One-Hot Wonders fans as the second feature (x_2). The bias column is the first column of the the design matrix $\mathbb{X}$. She assigns each dance battle a label of $y_i = 1$ if the One-Hot Wonders won, and $y_i = 0$ if they lost.

Here are the fitted model parameters:

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} 3 \\ 2 \\ 0 \end{bmatrix}$$

Suppose that Gisella attends a new One-Hot Wonders game. At the beginning of the game, she counts **three opponent fans** and **one fan for the One-Hot Wonders**. Use Gisella's fitted logistic regression model to compute the probability p that the One-Hot wonders win the competition.

Solution: The logistic regression model predicts the probability of a win as

$$p = \frac{1}{1 + e^{-z}},$$

where

$$z = \boldsymbol{\theta}^\top \mathbf{x}.$$

$$\mathbf{x}_g = \begin{bmatrix} 1 \\ X[c,1] \\ X[c,2] \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}.$$

Thus,

$$z = \boldsymbol{\theta}^\top \mathbf{x} = (3)(1) + (2)(3) + (0)(1) = 3 + 6 + 0 = 9.$$

Now substitute into the logistic function: $p = \frac{1}{1+e^{-9}}$.

(c) Ella suggests using a different model. Their chosen model is fit to data from the five competitions in part (a). Their model estimates the probabilities listed in the table below.

If Ella uses a **decision threshold of** 0.45, what is the **predicted outcome** of each competition, according to the model? Write your answers in the column labeled $\hat{Y}$ in the table below.

(i) [1 Pt]

$\hat{P}(Y = 1 \mid \mathbf{x})$	Y	$\hat{Y}$
0.95	0	1
0.75	1	1
0.60	0	1
0.40	0	0
0.25	1	0

(ii) [1 Pt] What is the **precision** of Ella's model, assuming we use the same data and threshold as the previous part? Show your work.

Solution:

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN}.$$

Here: $TP = 1$, $FP = 2$, $FN = 1$, so

$$\text{Precision} = \frac{1}{1+2} = \frac{1}{3}, \quad \text{Recall} = \frac{1}{1+1} = \frac{1}{2}.$$

(d) [1 Pt] Recall an ROC curve has the false positive rate (FPR) on the x-axis and true positive rate (TPR) on the y-axis. Which of the following is true?

○ True ○ **False** If you change the decision threshold, it is possible for the area of the ROC curve to increase or decrease.

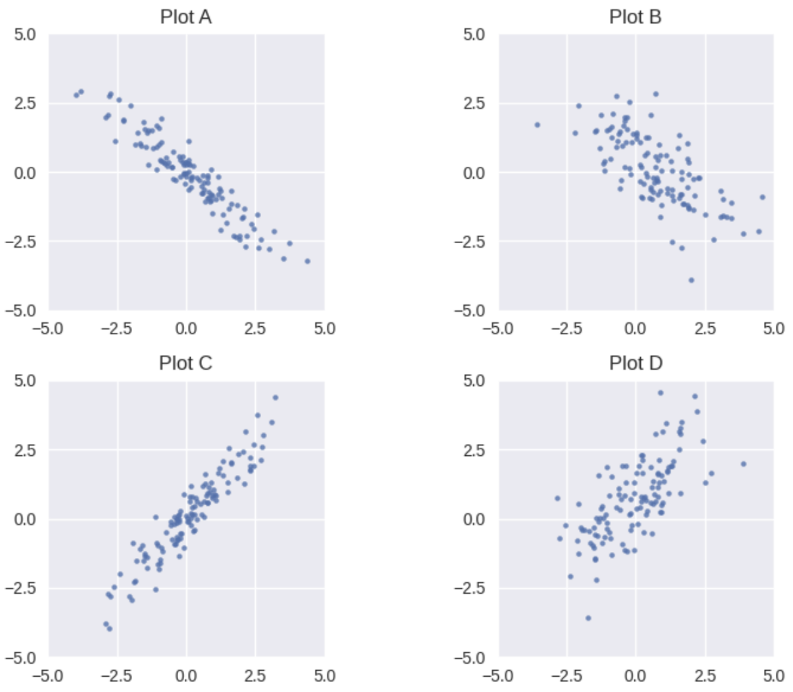
○ True ○ **False** The bottom left of an ROC curve corresponds to a low decision threshold, while the top right corresponds to a high decision threshold.

○ **True** ○ False If you increase the decision threshold, it is possible for the FPR and TPR to both stay the same.

○ True ○ **False** If you decrease the decision threshold, it is possible for recall to go down.

6 Basic Principals [3 Pts]

Sara has four datasets: A, B, C, and D. Each dataset consists of 100 data points in two dimensions. She visualizes the datasets using scatterplots, which are show below and are labeled Plot A, Plot B, Plot C, and Plot D respectively.



(a) [1 Pt] Sara applies PCA to each of the four datasets above. Which of the datasets have a PC2 vector of approximately $[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$?

○ **True** ○ False Dataset A

○ **True** ○ False Dataset B

○ True ○ **False** Dataset C

○ True ○ **False** Dataset D

(b) [1 Pt] Consider Dataset A. Suppose the variance of each feature of Dataset A is 10. What is the variance of Dataset A in the PC1 dimension?

○ Less than 10

○ Exactly 10

○ **Between 10 and 20 (exclusive)**

○ Exactly 20

○ Greater than 20

(c) [1 Pt] Consider Dataset A again. Suppose the variance of each feature of Dataset A is 10. What is the SUM of the variance of Dataset A in the PC1 dimension AND the variance of Dataset A in the PC2 dimension?

○ Less than 10

○ Exactly 10

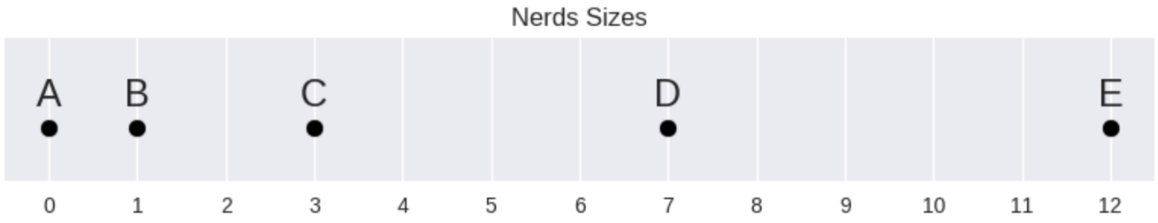
○ Between 10 and 20 (exclusive)

○ **Exactly 20**

○ Greater than 20

7 A Nerds' Clustering [4 points]

Emrie is working with a candy company to analyze the popular candy: Nerds Gummy Clusters. For each piece of candy, he records the size, but the candy production line does not label the flavor of each candy. He does know that the candies come from exactly **two** distinct flavors. He would like to try to identify these two distinct groups from this feature alone. His unlabeled data is shown below:



(a) [2 Pts] Emrie first decides to use agglomerative clustering to merge the data points. Write the steps of agglomerative clustering **with complete linkage** for this dataset, starting from when all points are their own individual clusters, and ending when all the points make one giant cluster. To help you, we have provided a template for you to fill in. We have filled in the first step and part of the last step for you.

1. {A} merges with {B} to form cluster {A,B}

2. {C} merges with {A, B} to form cluster {A, B, C}

3. {D} merges with {E} to form cluster {D,E}

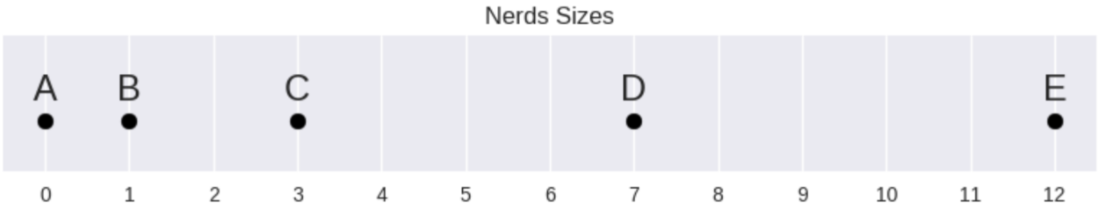
4. {A,B,C} merges with {D, E} to form cluster {A,B,C,D,E}

(b) [2 Pts] Emrie changes his mind and decides to use K-Medians clustering. **K-Medians clustering is exactly the same as K-means clustering, but it uses the median instead of the mean as the cluster centers.** He initializes Cluster 1's center at 1 and Cluster 2's center at 3.

Write the cluster center updates of K-medians clustering for this dataset. Write the updates in order and **until the algorithm converges**. We have provided a template for you to fill in. Each row represents one iteration of K-medians clustering.

Write the location of the updated Cluster 1 center and Cluster 2 center after each assignment step. **You do not have to use all rows. If the algorithm has already converged (i.e, the centers don't change), please write NA in any boxes you do not use.**

Here is the diagram again, for your reference:



Note: Remember that Emrie is using ***K-medians clustering****, not K-means clustering. If you need to find the median of an even number of points, use the average of the two middle points.*

Step	**Cluster 1 center**	**Cluster 2 center**
Initialization	1	3
Step 1	0.5	7
Step 2	1	9.5
Step 3	NA	NA
Step 4	NA	NA

Please state any relevant assumptions in the box below (Optional).

You are done with the final- Congratulations!

Draw your favorite DATA 100/200 memory so far!