Data C100/200- Midterm

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 **45 points** spread out over **4 questions** and the **Honor Code certification**. The exam must be completed in **110 minutes** unless you have accommodations supported by a DSP letter. Note that you should:

- Each true/false question and multiple choice question has **exactly one** correct answer. Please **fully** shade in the circle 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 math questions, **please simplify your answer**. Please also **show your work** if a large box is provided.
- 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: ____

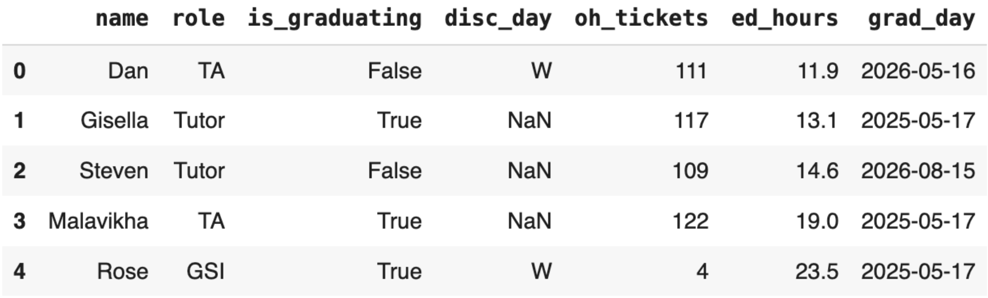
This page has been intentionally left blank.

1 Awesome DATA 100 Staff [19 Pts]

At the end of the semester, Data 100 instructors analyze staff performance using a `DataFrame` called `performance`. The columns of `performance` are described below:

- `name`: Name of the staff member (type = `str`).
  Note: No two staff members have exactly the same `name`.
- `role`: One of three possible staff roles: `"GSI"`, `"TA"`, or `"Tutor"` (type = `str`).
- `is_graduating`: `True` if the staff member is graduating this semester and `False` otherwise (type = `bool`).
- `disc_day`: The staff member's discussion day. One of three values: `"W"` (Wednesday), `"Th"` (Thursday), or `NaN` if the staff member does not hold any discussions (type = `str`).
- `oh_tickets`: Total office hours (OH) tickets the staff resolved during the semester (type = `np.int64`).
- `ed_hours`: Total hours the staff spent resolving questions on EdStem during the semester (type = `np.float64`).
- `grad_day`: The staff member's final day as an enrolled UC Berkeley student, in `"yyyy-mm-dd"` format (type = `str`).

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



	name	role	is_graduating	disc_day	oh_tickets	ed_hours	grad_day
0	Dan	TA	False	W	111	11.9	2026-05-16
1	Gisella	Tutor	True	NaN	117	13.1	2025-05-17
2	Steven	Tutor	False	NaN	109	14.6	2026-08-15
3	Malavikha	TA	True	NaN	122	19.0	2025-05-17
4	Rose	GSI	True	W	4	23.5	2025-05-17

(a) For each `performance` column, choose the best variable type.

(i) [0.5 Pts] The best variable type for the column `name` is:

○ Qualitative ordinal
○ Qualitative nominal
○ Quantitative

(ii) [0.5 Pts] The best variable type for the column `oh_tickets` is:

○ Qualitative ordinal
○ Qualitative nominal
○ Quantitative

(b) [0.5 Pts] What is the granularity of the `performance DataFrame`? Answer with a brief sentence or phrase.

(c) [1.5 Pts] Instructors want to visualize the distribution of `ed_hours`. Mark True if the plot type is appropriate for this visualization task, and False otherwise.

○ True ○ False Contourplot

○ True ○ False Histogram

○ True ○ False KDE Plot

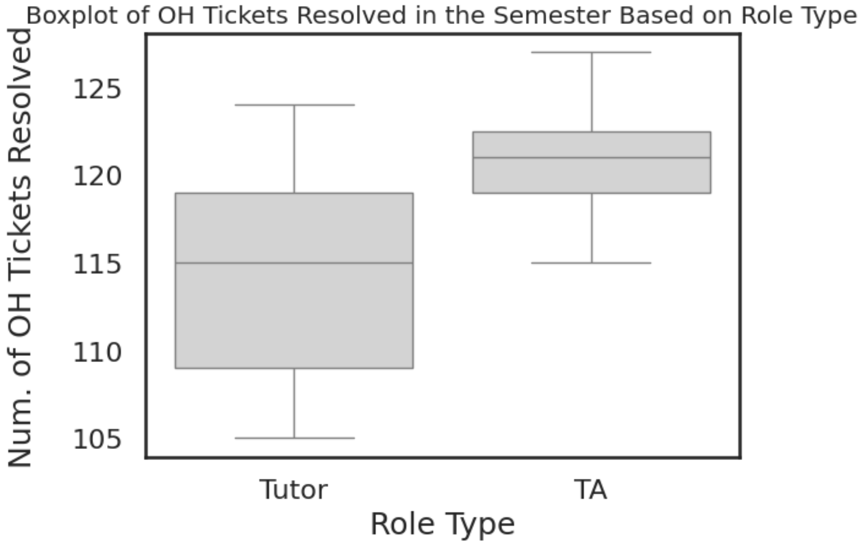
(d) [1.5 Pts] Instructors want to visualize the relationship between `oh_tickets` and `ed_hours`. Mark True if the plot type is appropriate for this visualization task, and False otherwise.

○ True ○ False Jointplot

○ True ○ False Overlaid histograms

○ True ○ False Hexplot

(e) [1 Pt] Instructors compare `oh_tickets` for Tutors and TAs using a boxplot shown below:

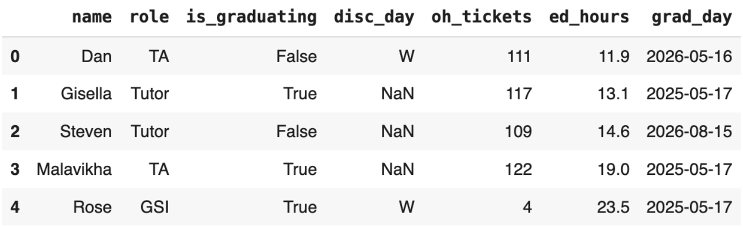


Select True or False for the statements below.

○ True ○ False The median OH tickets resolved is higher for Tutors than for TAs.

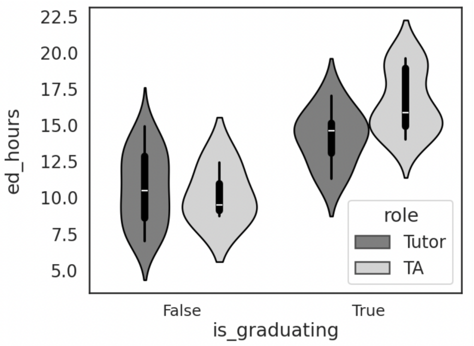
○ True ○ False About 75% of TAs each resolved more OH tickets than about 75% of Tutors.

The first five rows of `performance` are shown again here for your convenience:



	name	role	is_graduating	disc_day	oh_tickets	ed_hours	grad_day
0	Dan	TA	False	W	111	11.9	2026-05-16
1	Gisella	Tutor	True	NaN	117	13.1	2025-05-17
2	Steven	Tutor	False	NaN	109	14.6	2026-08-15
3	Malavikha	TA	True	NaN	122	19.0	2025-05-17
4	Rose	GSI	True	W	4	23.5	2025-05-17

(f) Fill in the blanks to create a side-by-side violin plot to visualize the distribution of `ed_hours` for each combination of role type and graduation status, as shown below:



Note: The order of the arguments does not matter as long as each is named.

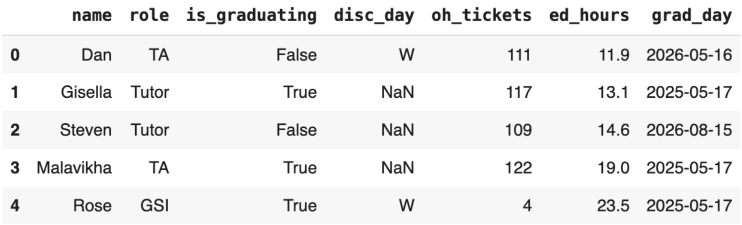
```
sns.violinplot(
data=performance[performance["role"].isin(["TA","Tutor"])],
__________(i)_________,
__________(ii)_________,
__________(iii)_________)
```

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

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

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

The first five rows of `performance` are shown again here for your convenience:



	name	role	is_graduating	disc_day	oh_tickets	ed_hours	grad_day
0	Dan	TA	False	W	111	11.9	2026-05-16
1	Gisella	Tutor	True	NaN	117	13.1	2025-05-17
2	Steven	Tutor	False	NaN	109	14.6	2026-08-15
3	Malavikha	TA	True	NaN	122	19.0	2025-05-17
4	Rose	GSI	True	W	4	23.5	2025-05-17

(g) [2 Pts] Write a single line of code to return a `String` with the **name of the staff member** who has resolved the highest number of `oh_tickets`.

Note: There is exactly one staff member with the highest count.

Answer: (____

____)

(h) Instructors want to assess office hours and EdStem performance by staff role.

Assign `staff_stats` to a `DataFrame` where `role` is the `index`, and the `values` are the **minimum** number of `oh_tickets` and **maximum** number of `ed_hours` for staff members in each role category. Fill in the blanks to achieve this.

Note: The resulting `DataFrame` should contain only two columns. The order of the columns does not matter.

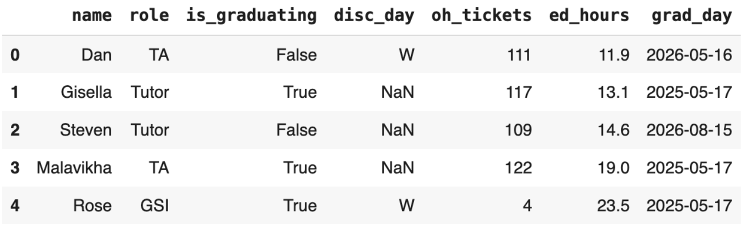
```
staff_stats = performance.groupby(________(i)________)
.agg( {____________(ii)____________, ____________(iii)____________})
```

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

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

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

The first five rows of `performance` are shown again here for your convenience:



	name	role	is_graduating	disc_day	oh_tickets	ed_hours	grad_day
0	Dan	TA	False	W	111	11.9	2026-05-16
1	Gisella	Tutor	True	NaN	117	13.1	2025-05-17
2	Steven	Tutor	False	NaN	109	14.6	2026-08-15
3	Malavikha	TA	True	NaN	122	19.0	2025-05-17
4	Rose	GSI	True	W	4	23.5	2025-05-17

(i) [1.5 Pts] Instructors want to recognize top performers on EdStem who **do not hold discussion sections**. A top-performer is any staff who spent **at least 15 hours** on EdStem.

Note: For staff without discussion sections, `disc_day` is `NaN`.

Fill in the following blank to assign `top_ed` to a modified version of the `performance` `DataFrame` that includes all rows corresponding to these staff members:

```
top_ed = performance[__________(A)____________]
```

Fill in blank `(A)` below.

Answer: (____

____)

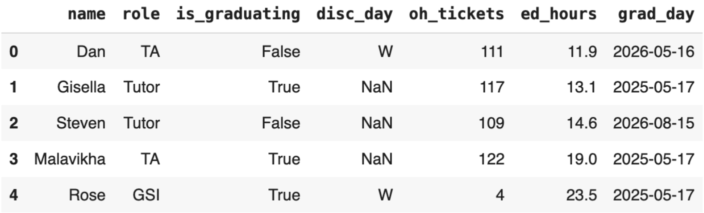
(j) Instructors change their mind about which rows of `performance` to keep. They assign `top_role` to a modified `performance DataFrame` that contains rows where the corresponding `role` is classified as top-performing. A `role` is top-performing if the average `ed_hours` for that `role` is **at least 15**. Fill in the blanks to achieve this.

```
top_role = performance.groupby(_____(A)_____)._____(B)_____
```

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

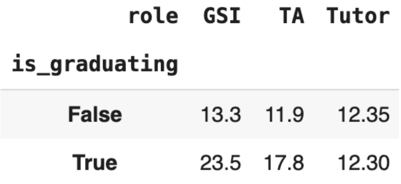
(ii) [1.5 Pts] Fill in blank `(B)`:

The first five rows of `performance` are shown again here for your convenience:



	name	role	is_graduating	disc_day	oh_tickets	ed_hours	grad_day
0	Dan	TA	False	W	111	11.9	2026-05-16
1	Gisella	Tutor	True	NaN	117	13.1	2025-05-17
2	Steven	Tutor	False	NaN	109	14.6	2026-08-15
3	Malavikha	TA	True	NaN	122	19.0	2025-05-17
4	Rose	GSI	True	W	4	23.5	2025-05-17

(k) Instructors want to find out how much time graduating staff members spend on EdStem. To investigate, they created the `DataFrame` shown below, where they aggregated `ed_hours` using the **median** and **imputed Null values with 0.**



role	GSI	TA	Tutor
is_graduating			
False	13.3	11.9	12.35
True	23.5	17.8	12.30

Fill in the blanks to replicate the above `DataFrame`.

```
performance.pivot_table(________(i)________,
                        ________(ii)_______,
                        _______(iii)_______,
                        ________(iv)_______,
                        ________(v)________)
```

Note: The order of the arguments does not matter as long as each is named.

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

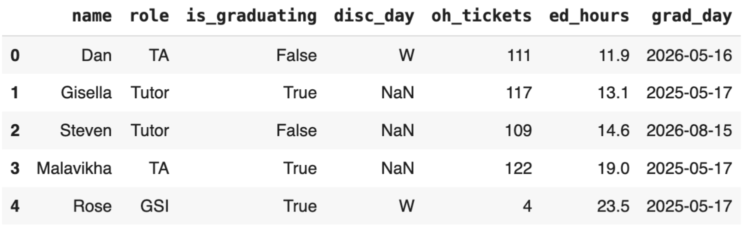
(ii) [0.5 Pt] Fill in blank `(ii)`:

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

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

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

The first five rows of `performance` are shown again here for your convenience:



	name	role	is_graduating	disc_day	oh_tickets	ed_hours	grad_day
0	Dan	TA	False	W	111	11.9	2026-05-16
1	Gisella	Tutor	True	NaN	117	13.1	2025-05-17
2	Steven	Tutor	False	NaN	109	14.6	2026-08-15
3	Malavikha	TA	True	NaN	122	19.0	2025-05-17
4	Rose	GSI	True	W	4	23.5	2025-05-17

(l) Finally, the instructors want to remove all rows where any value is missing and retain only those corresponding to staff members graduating in 2026.

Follow the steps and fill in the blanks to achieve this:

```
# Step 1:  Remove all rows with missing values (NaNs).

perf_no_missing = performance.________(i)________

# Step 2:  Extract year from the "grad_day" column and convert
   to type "int".

perf_no_missing["year"] = ________(ii)________

# Step 3:  Filter the performance DataFrame to keep staff
   members who are graduating in 2026.

filtered_perf = perf_no_missing[_____(iii)_____]
```

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

(ii) [1 Pt] Fill in blank `(ii)`:
Note: For Step 2, you must use string slicing to complete the task.
You may not use the `.dt` accessor.

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

2 CHARprinter's Intro-Spection [6 Pts]

Sabrina Charprinter obtained text from the "Staff" page of the Data 100 website. She wants to use RegEx to extract certain pieces of information (for her upcoming Data 100 parody song).

Note: For all parts, you will only need to consider the example strings given to you.
You may assume that these examples cover all edge cases.

(a) [3 Pts] Sabrina found metadata in each staff introduction to process as strings.

Suppose Sabrina has already created a pattern and runs the code below. The character "␣" represents a single space, and you may use it in your response.

```
 metadata␣=␣"Name:␣Oski;␣Age:␣159;␣Courses␣taken:␣Data8,
␣Data100,␣CS61A,␣Stat134,␣Data88s;␣Phone Number:
␣555-100-5555;␣Likes:␣[Bears,␣Strawberries,␣Data Science]"

pattern = r"\w+:\s([^A-Za-z\s]+)"

re.findall(pattern, metadata)
```

List matches in the order returned by `re.findall(pattern, metadata)`, with the first match next to `Match 1`.

Note: You may have less than 6 total matches. For any unmatched slots, **write** `No Match` **instead.**

Match 1: '________________________________'

Match 2: '________________________________'

Match 3: '________________________________'

Match 4: '________________________________'

Match 5: '________________________________'

Match 6: '________________________________'

(b) Help Sabrina create a RegEx pattern to extract the course IDs for all Data Science courses when running the code below. The IDs appear immediately after the subject name "Data" (with no spaces), and they follow these rules:

- They consist of exactly three digits, **or**
- They consist of one or two digits, optionally followed by a letter from the set ["c", "s", "x"].

For example, the output of running the following code block should be ['8', '100', '88s'].

```
 metadata␣=␣"Name:␣Oski;␣Age:␣159;␣Courses␣taken:␣Data8,
␣Data100,␣CS61A,␣Stat134,␣Data88s;␣Phone Number:
␣555-100-5555;␣Likes:␣[Bears,␣Strawberries,␣Data Science]"


pattern = r"Data(____(i)____|____(ii)____)"


re.findall(pattern, metadata)
```

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

(ii) [2 Pts] Fill in blank (ii):

3 Bay Area Rapid Studies (BARS) [12.5 Pts]

Rachel, a data scientist at the UC Berkeley Transportation Department, oversees the **BayPass Program**. The BayPass Program is a study that analyzes the effects of giving UC Berkeley students free access to all Bay Area transportation services through a BayPass transit card.

(a) Rachel selects 12,000 distinct UC Berkeley students uniformly at random from the UC Berkeley enrollment database and gives them a BayPass transit card. She collects usage data from these cards for analysis.

(i) [0.5 Pts] What is the population of interest?

○ UC Berkeley students

○ BayPass card holders

○ UC Berkeley students who use Bay Area transit services

○ People who use Bay Area transit services

(ii) [0.5 Pts] What is the sampling frame in Rachel's study?

(b) [2 Pts] Rachel invites UC Berkeley students through Data 100 EdStem to take a survey about their experiences with Bay Area public transportation for 5% midterm extra credit. Which statements about her sampling process are true?

○ True ○ False Since Rachel provided a generous incentive, her survey results will no longer be affected by non-response bias.

○ True ○ False Rachel's sampling method may suffer from selection bias.

○ True ○ False Rachel's sampling method may suffer from response bias.

○ True ○ False The respondents are guaranteed to be representative of the target population.

(c) [0.5 Pts] As part of a separate transportation analysis, Rachel wants to survey a sample of UC Berkeley students on how many times they traveled out of Berkeley in the past month. She decides to divide the population into groups based on their year (e.g., freshman, sophomore, junior, senior, and graduate students) and then conduct a simple random sample of size n_g among each group g, where n_g is proportional to the number of enrolled students at UC Berkeley in group g. What type of sample is this?

○ Convenience sample

○ Stratified random sample

○ Uniform random sample with replacement

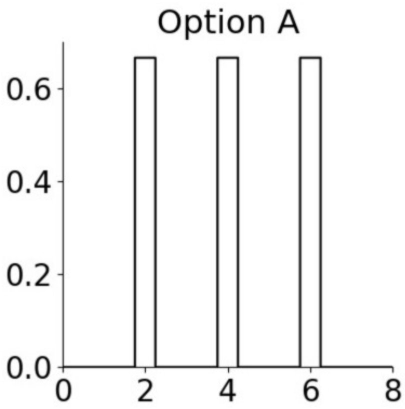
○ Post stratification

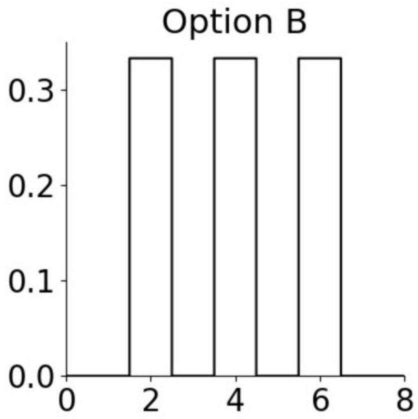
(d) [1 Pt] Rachel is trying to understand how long BayPass users stay on campus. She collects data on `hours` (x_i) for three students: $\{2, 4, 6\}$. She is using a boxcar kernel with bandwidth $\alpha = 0.5$ to estimate the density distribution of the data.

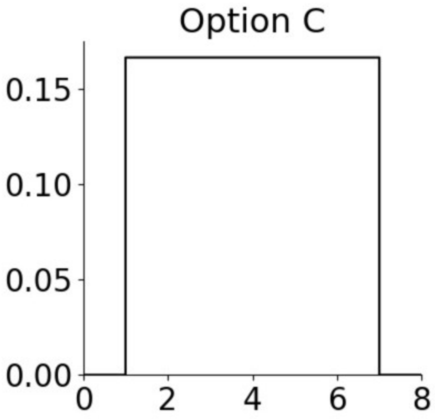
A boxcar kernel is defined as follows:

$$K_\alpha(x, x_i) = \begin{cases} \frac{1}{\alpha}, & \text{if } |x - x_i| \leq \frac{\alpha}{2} \\ 0, & \text{otherwise.} \end{cases}$$

Which of the following KDE plots correctly represents the estimated density?







○ Option A ○ Option B ○ Option C

(e) Rachel is now examining a small sample dataset representing the number of rides taken in a month by three BayPass users: $\{1, 6, 17\}$. She wants to pick a single summary statistic θ to describe the data. Determine the value $\hat{\theta}$ that minimizes each of the following objective functions.

(i) [1 Pt] $R(\theta) = \frac{1}{3}\sum_{i=1}^{3}(x_i - \theta)^2$

○ $\frac{8}{3}$

○ $\frac{6}{3}$

○ 6

○ 8

(ii) [1 Pt] $R(\theta) = \frac{1}{1000}\sum_{i=1}^{3}|x_i - \theta|$

○ $\frac{8}{1000}$

○ 6

○ 8

○ $\frac{6}{1000}$

(f) [1.5 Pts] Rachel is building a model to predict the number of rides taken by UC Berkeley students using their BayPass. Which of the following scenarios would Mean Squared Error (MSE) be preferred over Mean Absolute Error (MAE) for a linear regression task?

○ True ○ False When the model needs to be sensitive to outliers.

○ True ○ False When large errors should be penalized more heavily.

○ True ○ False When a smooth, differentiable loss function is required for finding the minimum average loss.

(g) [0.5 Pts] Rachel fits a Simple Linear Regression (SLR) model to predict the number of rides taken by UC Berkeley students using their BayPass. She uses the hours spent on campus (`hours`) as a predictor and the number of rides taken (`rides`) as a response. The equation for the SLR model is given by:

$$\widehat{\texttt{rides}} = \hat{\theta}_0 + \hat{\theta}_1 \, \texttt{hours}$$

○ Estimated number of rides taken when no hours are spent on campus.

○ Estimated average ride increase per additional hour on campus.

○ Estimated average rides taken by students.

○ Estimated total number of rides by students in the dataset.

(h) Rachel continues analyzing the number of `rides` (y_i) based on the `hours` (x_i). She models the relationship using SLR **without** an intercept:

$$\widehat{\texttt{rides}} = \theta_1 \, \texttt{hours}$$

Instead of using MSE, she decides to minimize the following custom objective function:

$$R(\theta_1) = \frac{1}{n}\sum_{i=1}^{n} x_i(y_i - \theta_1 x_i)^2$$

(i) [2 Pts] Find the derivative of $R(\theta_1)$ with respect to θ_1. Your answer should be in terms of x_i, y_i, θ_1, n. To be eligible for partial credit, show all your work in the box below.

(ii) [2 Pts] Find $\hat{\theta}_1$ that minimizes the objective function. Your answer should be in terms of x_i, y_i, n. To be eligible for partial credit, show all your work in the box below.

Note: Assume that the provided objective function is convex.

4 We Miss Moffitt </3 [6.5 Pts]

Since the closure of Moffitt Library, course staff have struggled to find a study spot. One popular alternative is MLK Student Union. Sarah decides to use Ordinary Least Squares (OLS) to predict the number of students ($\mathbb{Y}$) in MLK Student Union for a given humidity level and time of day.

(a) [2 Pts] Sarah fits an OLS model to predict the number of students (`num_students`) at MLK Student Union using the humidity level (`humidity`) and time of day (`time`) as predictors. Her fitted model is:

$$\widehat{\texttt{num\_students}} = \hat{\theta}_0 + \hat{\theta}_1 \times \texttt{humidity} + \hat{\theta}_2 \times \texttt{time}$$

After fitting the optimal model, Sarah examines the residual vector $\vec{e} = \mathbb{Y} - \hat{\mathbb{Y}}$. For each statement, indicate whether it is True or False.

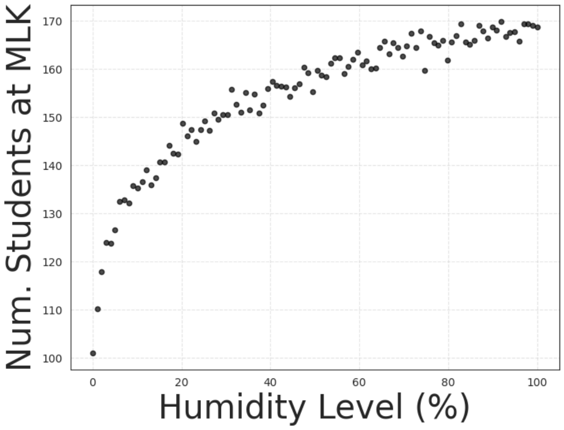
○ True ○ False $\vec{e}$ is orthogonal to all columns of the design matrix $\mathbb{X}$.

○ True ○ False If $\hat{\theta}_0$ is positive, the average of the residuals is also positive.

○ True ○ False Assuming $\mathbb{X}$ is full column rank, the sum of squared residuals is minimized when $\hat{\theta}$ is calculated using the normal equation.

○ True ○ False Residual vector $\vec{e}$ has the same dimension as the parameter vector $\hat{\theta}$.

(b) [0.5 Pts] Sarah observes a non-linear relationship between `humidity` and `num_students`, as shown below:



Let x_i represent `humidity` for the i-th data point. Let h_i represent the **transformed value** of the humidity level used in the model.

Which of the following transformations should Sarah apply to linearize the data?

○ $h_i = \log(x_i)$

○ $\log(h_i) = x_i$

○ $h_i = (x_i)^2$

○ $h_i = (x_i)^3$

(c) Answer the following questions about OLS models:

(i) [2 Pts] We want to fit an OLS model with n observations and $p+1$ features (including the intercept). Select **True** if the dimensions of the following matrices are correct in this task, and **False** otherwise.

○ True ○ False $\mathbb{X} : n \times (p+1)$
○ True ○ False $\mathbb{X}^T\mathbb{X} : (p+1) \times (p+1)$
○ True ○ False $\hat{\theta} : p \times 1$
○ True ○ False $\mathbb{Y} : n \times 1$

(ii) [1 Pt] Which of the following best explains the condition required for the OLS model to produce a unique solution?

○ $\mathbb{X}$ must not be full column rank.
○ $\mathbb{X}$ must be a square matrix.
○ $\mathbb{X}^T\mathbb{X}$ must be invertible.
○ $\mathbb{X}$ must have fewer rows than columns.

(iii) [1 Pt] Which of the following best describes the geometric interpretation of the OLS prediction vector $\hat{\mathbb{Y}}$?

○ $\hat{\mathbb{Y}}$ is the difference between $\mathbb{Y}$ and the orthogonal projection of $\mathbb{Y}$ onto the span of $\mathbb{X}$.
○ $\hat{\mathbb{Y}}$ is the orthogonal projection of $\mathbb{Y}$ onto the span of $\mathbb{X}$.
○ When $\mathbb{X}$ is not full column rank, $\hat{\mathbb{Y}}$ is still uniquely determined because the normal equation always has a single solution.
○ $\hat{\mathbb{Y}}$ is the orthogonal projection of $\mathbb{X}$ onto the span of $\mathbb{Y}$.

You are done with the midterm- Congratulations!

Draw your favorite DATA 100/200 memory so far!