Final Exam - DSC 10, Fall 2023

Full Name:

PID:

Room Number: ◯ PCYNH 106 ◯ PCYNH 109 ◯ PCYNH 122 ◯ Other

Seat Number:

Instructions:

- This exam consists of 14 questions, worth a total of 116 points.
- Write your PID in the top right corner of each page in the space provided.
- Please write **clearly** in the provided answer boxes; we will not grade work that appears elsewhere. Completely fill in bubbles and square boxes; if we cannot tell which option(s) you selected, you may lose points.

  ◯ A bubble means that you should only **select one choice**.

  ☐ A square box means you should **select all that apply**.
- You may refer to the Reference Sheet that was provided to you. Other than that, you may not refer to any other resources or technology during the exam (no notes, no calculators).
- Note: Questions vary in difficulty. You may want to browse through all questions before starting. Be sure to budget your time!

By signing below, you are agreeing that you will behave honestly and fairly during and after this exam.

Signature:

Version A

Please do not open your exam until instructed to do so.

Question 1 (12 pts)

a) (4 pts) Nate's favorite number is 5. He calls a number "lucky" if it's greater than 500 or if it contains a 5 anywhere in its representation. For example, `1000.04` and `5.23` are both lucky numbers.

Complete the implementation of the function `check_lucky`, which takes in a number as a float and returns `True` if it is lucky and `False` otherwise. Then, add a column named `"is_lucky"` to `txn` that contains `True` for lucky transaction amounts and `False` for all other transaction amounts, and save the resulting DataFrame to the variable `luck`.

```
def check_lucky(x):
    return __(a)__

luck = txn.assign(is_lucky = __(b)__)
```

(i) What goes in blank (a)?

(ii) What goes in blank (b)?

b) (4 pts) Fill in the blanks below so that `lucky_prop` evaluates to the proportion of fraudulent `"visa"` card transactions whose transaction amounts are lucky.

```
visa_fraud = __(a)__
lucky_prop = visa_fraud.__(b)__.mean()
```

(i) What goes in blank (a)?

(ii) What goes in blank (b)?

c) (4 pts) Fill in the blanks below so that `lucky_prop` is one value in the Series `many_props`.

```
many_props = (luck.get(["card", "is_fraud", "is_lucky"])
                  .groupby(__(a)__).mean().get(__(b)__))
```

(i) What goes in blank (a)?

(ii) What goes in blank (b)?

Question 2 (6 pts)

Consider the DataFrame `combo`, defined below.

```
combo = (txn.drop(columns=["browser"])
            .groupby(["is_fraud", "method", "card"]).mean())
```

a) (4 pts) What is the maximum possible value of `combo.shape[0]`? Give your answer as an integer.

b) (2 pts) What is the value of `combo.shape[1]`?

◯ 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯ 6

Question 3 (5 pts)

Consider the variable `is_fraud_mean`, defined below.

```
is_fraud_mean = txn.get("is_fraud").mean()
```

Which of the following expressions are equivalent to `is_fraud_mean`? **Select all that apply.**

☐ `txn.groupby("is_fraud").mean()`

☐ `txn.get("is_fraud").sum() / txn.shape[0]`

☐ `np.count_nonzero(txn.get("is_fraud")) / txn.shape[0]`

☐ `(txn.get("is_fraud") > 0.8).mean()`

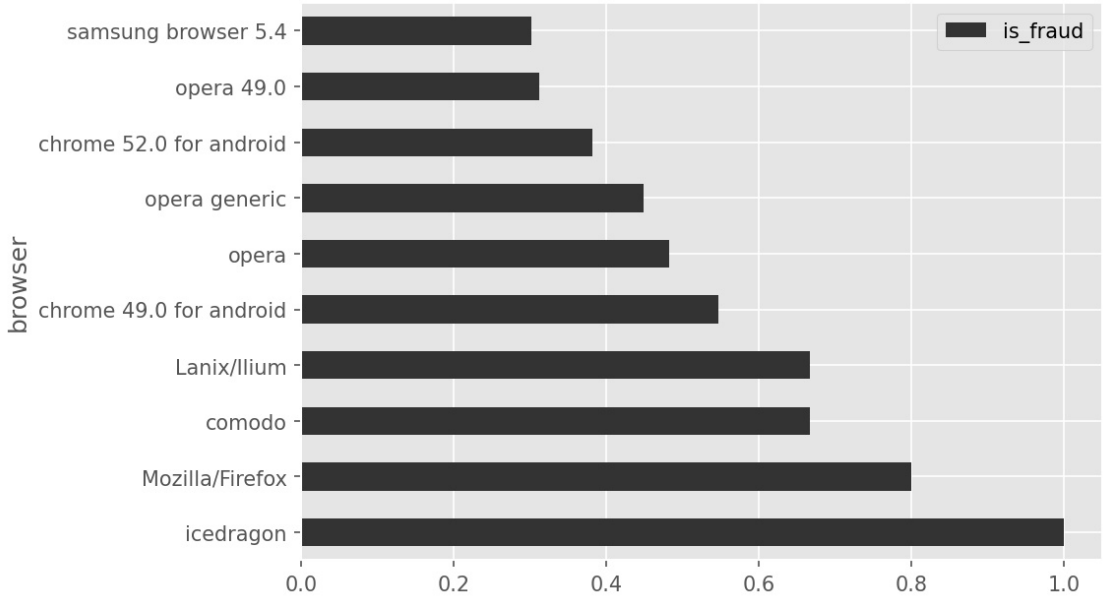
☐ `1 - (txn.get("is_fraud") == 0).mean()`

☐ None of the above.

Question 4 (6 pts)

The following code block produces a bar chart, which is shown directly beneath it.

```
(txn.get(["browser", "is_fraud"]).groupby("browser").mean()
    .sort_values(by="is_fraud", ascending=False)
    .take(np.arange(10))
    .plot(kind="barh", y="is_fraud"));
```



a) (4 pts) Based on the above bar chart, what can we conclude about the browser `"icedragon"`? **Select all that apply.**

- ☐ In our dataset, `"icedragon"` is the most frequently used browser among all transactions.
- ☐ In our dataset, `"icedragon"` is the most frequently used browser among fraudulent transactions.
- ☐ In our dataset, every transaction made with `"icedragon"` is fraudulent.
- ☐ In our dataset, there are more fraudulent transactions made with `"icedragon"` than with any other browser.
- ☐ None of the above.

b) (2 pts) How can we modify the provided code block so that the bar chart displays the same information, but with the bars sorted in the opposite order (i.e. with the longest bar at the top)?

- ◯ Change `ascending=False` to `ascending=True`.
- ◯ Add `.sort_values(by="is_fraud", ascending=True)` immediately before `.take(np.arange(10))`.
- ◯ Add `.sort_values(by="is_fraud", ascending=True)` immediately after `.take(np.arange(10))`.
- ◯ None of the above.

PID: _____

Question 5 (6 pts)

The DataFrame `seven`, shown below to the **left**, consists of a simple random sample of 7 rows from `txn`, with just the `"is_fraud"` and `"amount"` columns selected.

The DataFrame `locations`, shown below to the **right**, is missing some values in its `"is_fraud"` column.

transaction_id	is_fraud	amount
3285401	False	88.21
3115253	False	18.89
3074517	True	25.07
3381236	True	34.92
3020615	False	20.00
3054996	False	100.00
3124460	False	31.13

	is_fraud	location
0	___	San Diego, CA, USA
1	___	New York, NY, USA
2	___	San Diego, CA, USA
3	___	Detroit, MI, USA
4	True	Toronto, ON, Canada

a) (4 pts) Below, fill in the blanks to complete the `"is_fraud"` column of `locations` so that the DataFrame `seven.merge(locations, on="is_fraud")` has **19** rows.

Index 0: ____

Index 1: ____

Index 2: ____

Index 3: ____

Index 4: True

b) (2 pts) True or False: It is possible to fill in the four blanks in the `"is_fraud"` column of `locations` so that the DataFrame `seven.merge(locations, on="is_fraud")` has **14** rows.

◯ True ◯ False

Question 6 (11 pts)

Aaron wants to explore the discrepancy in fraud rates between `"discover"` transactions and `"mastercard"` transactions. To do so, he creates the DataFrame `ds_mc`, which only contains the rows in `txn` corresponding to `"mastercard"` or `"discover"` transactions.

After he creates `ds_mc`, Aaron groups `ds_mc` on the `"card"` column using two different aggregation methods. The relevant columns in the resulting DataFrames are shown below.

`ds_mc.groupby("card").sum()`

card	is_fraud
discover	160
mastercard	4000

`ds_mc.groupby("card").count()`

card	is_fraud
discover	2000
mastercard	40000

Aaron decides to perform a test of the following pair of hypotheses:

- **Null Hypothesis**: The proportion of fraudulent `"mastercard"` transactions is **the same as** the proportion of fraudulent `"discover"` transactions.
- **Alternative Hypothesis**: The proportion of fraudulent `"mastercard"` transactions is **less than** the proportion of fraudulent `"discover"` transactions.

As his test statistic, Aaron chooses the **difference in proportion of transactions that are fraudulent**, in the order `"mastercard"` minus `"discover"`.

a) (2 pts) What type of statistical test is Aaron performing?

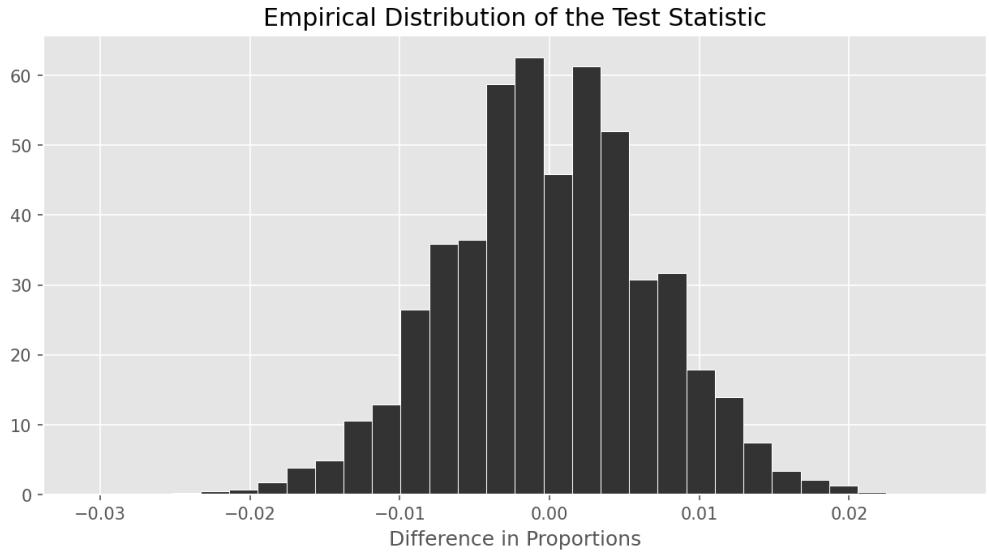
◯ Standard hypothesis test ◯ Permutation test

b) (3 pts) What is the value of the observed statistic? Give your answer either as an exact decimal or simplified fraction.

observed statistic = ____

PID: _____

The empirical distribution of Aaron's chosen test statistic is shown below.



c) (2 pts) Which of the following is closest to the p-value of Aaron's test?

◯ 0.001 ◯ 0.06 ◯ 0.37 ◯ 0.63 ◯ 0.94 ◯ 0.999

d) (2 pts) What is the conclusion of Aaron's test?

◯ The proportion of fraudulent `"mastercard"` transactions is **less than** the proportion of fraudulent `"discover"` transactions.

◯ The proportion of fraudulent `"mastercard"` transactions is **greater than** the proportion of fraudulent `"discover"` transactions.

◯ The test results are inconclusive.

◯ None of the above.

Aaron now decides to test a slightly different pair of hypotheses.

- **Null Hypothesis**: The proportion of fraudulent `"mastercard"` transactions is **the same as** the proportion of fraudulent `"discover"` transactions.
- **Alternative Hypothesis**: The proportion of fraudulent `"mastercard"` transactions is **greater than** the proportion of fraudulent `"discover"` transactions.

He uses the same test statistic as before.

e) (2 pts) Which of the following is closest to the p-value of Aaron's new test?

◯ 0.001 ◯ 0.06 ◯ 0.37 ◯ 0.63 ◯ 0.94 ◯ 0.999

Question 7 (6 pts)

Jason is interested in exploring the relationship between the browser and payment method used for a transaction. To do so, he uses `txn` to create create three tables, each of which contain the distribution of browsers used for credit card transactions and the distribution of browsers used for debit card transactions, but with different combinations of browsers combined in a single category in each table.

Browser	Credit	Debit
Chrome	?	?
Safari	?	?
Firefox	?	?
Edge	?	?
Other	?	?

Table 1

Browser	Credit	Debit
Chrome	?	?
Safari or Firefox	?	?
Edge	?	?
Other	?	?

Table 2

Browser	Credit	Debit
Chrome	?	?
Safari, Firefox, or Edge	?	?
Other	?	?

Table 3

Jason calculates the total variation distance (TVD) between the two distributions in each of his three tables, but he does not record which TVD goes with which table. He computed TVDs of 0.14, 0.36, and 0.38.

a) (2 pts) In which table do the two distributions have a TVD of 0.14?

◯ Table 1 ◯ Table 2 ◯ Table 3

b) (2 pts) In which table do the two distributions have a TVD of 0.36?

◯ Table 1 ◯ Table 2 ◯ Table 3

c) (2 pts) In which table do the two distributions have a TVD of 0.38?

◯ Table 1 ◯ Table 2 ◯ Table 3

Question 8 (6 pts)

Since `txn` has 140,000 rows, Jack wants to get a quick glimpse at the data by looking at a simple random sample of 10 rows from `txn`. He defines the DataFrame `ten_txns` as follows:

```
ten_txns = txn.sample(10, replace=False)
```

Which of the following code blocks also assign `ten_txns` to a simple random sample of 10 rows from `txn`?

Option 1:

```
all_rows = np.arange(txn.shape[0])
perm = np.random.permutation(all_rows)
positions = np.random.choice(perm, size=10, replace=False)
ten_txn = txn.take(positions)
```

Option 2:

```
all_rows = np.arange(txn.shape[0])
choice = np.random.choice(all_rows, size=10, replace=False)
positions = np.random.permutation(choice)
ten_txn = txn.take(positions)
```

Option 3:

```
all_rows = np.arange(txn.shape[0])
positions = np.random.permutation(all_rows).take(np.arange(10))
ten_txn = txn.take(positions)
```

Option 4:

```
all_rows = np.arange(txn.shape[0])
positions = np.random.permutation(all_rows.take(np.arange(10)))
ten_txn = txn.take(positions)
```

Select all that apply.

☐ Option 1

☐ Option 2

☐ Option 3

☐ Option 4

☐ None of the above.

Question 9 (10 pts)

The DataFrame `ten_txns`, **displayed in its entirety below**, contains a simple random sample of 10 rows from `txn`.

transaction_id	is_fraud	amount	method	card	lifetime	browser
3169166	True	100.00	credit	visa	532601.00	chrome 63.0
3093921	False	100.00	debit	mastercard	173276.00	mobile safari 10.0
3137058	False	100.00	debit	visa	120000.00	chrome 63.0
3063164	False	100.00	debit	visa	141342.00	mobile safari 11.0
3051461	False	75.00	credit	visa	153871.00	ie 11.0 for desktop
3171154	False	25.00	debit	visa	182654.00	mobile safari generic
3222420	False	25.00	credit	visa	50199.00	safari generic
3226397	False	22.96	debit	mastercard	122352.00	safari generic
3073572	False	9.33	credit	mastercard	66703.00	mobile safari 11.0
3253371	False	5.00	debit	visa	3007.00	chrome 64.0

a) (3 pts) Suppose we randomly select one transaction from `ten_txns`. What is the probability that the selected transaction is made with a `"card"` of `"mastercard"` or a `"method"` of `"debit"`? Give your answer either as an exact decimal or a simplified fraction.

b) (4 pts) Suppose we randomly select two transactions from `"ten_txns"`, without replacement, and learn that neither of the selected transactions is for an amount of 100 dollars. Given this information, what is the probability that:

- the first transaction is made with a `"card"` of `"visa"` and a `"method"` of `"debit"`, and
- the second transaction is made with a `"card"` of `"visa"` and a `"method"` of `"credit"`?

Give your answer either as an exact decimal or a simplified fraction.

PID: _____

For your convenience, we show `ten_txns` again below.

transaction_id	is_fraud	amount	method	card	lifetime	browser
3169166	True	100.00	credit	visa	532601.00	chrome 63.0
3093921	False	100.00	debit	mastercard	173276.00	mobile safari 10.0
3137058	False	100.00	debit	visa	120000.00	chrome 63.0
3063164	False	100.00	debit	visa	141342.00	mobile safari 11.0
3051461	False	75.00	credit	visa	153871.00	ie 11.0 for desktop
3171154	False	25.00	debit	visa	182654.00	mobile safari generic
3222420	False	25.00	credit	visa	50199.00	safari generic
3226397	False	22.96	debit	mastercard	122352.00	safari generic
3073572	False	9.33	credit	mastercard	66703.00	mobile safari 11.0
3253371	False	5.00	debit	visa	3007.00	chrome 64.0

c) (3 pts) Suppose we randomly select 15 rows, with replacement, from `ten_txns`. What's the probability that in our selection of 15 rows, the maximum transaction amount is less than 25 dollars?

◯ $\frac{3}{10}$
◯ $\left(\frac{3}{10}\right)^3$
◯ $\left(\frac{3}{10}\right)^{10}$
◯ $\left(\frac{3}{10}\right)^{15}$
◯ $\frac{3}{15}$
◯ $\left(\frac{3}{15}\right)^3$
◯ $\left(\frac{3}{15}\right)^{10}$
◯ $\left(\frac{3}{15}\right)^{15}$

Question 10 (12 pts)

As a senior suffering from senioritis, Weiyue has plenty of time on his hands. 1,000 times, he repeats the following process, creating 1,000 confidence intervals:

1. Collect a simple random sample of 100 rows from `txn`.
2. Resample from his sample 10,000 times, computing the mean transaction amount in each resample.
3. Create a 95% confidence interval by taking the middle 95% of resample means.

He then computes the width of each confidence interval by subtracting its left endpoint from its right endpoint; e.g. if $[2, 5]$ is a confidence interval, its width is 3. This gives him 1,000 widths. Let M be the mean of these 1,000 widths.

a) (3 pts) Select the true statement below.

◯ About 950 of Weiyue's intervals will contain the mean transaction amount of all transactions ever.

◯ About 950 of Weiyue's intervals will contain the mean transaction amount of all transactions in `txn`.

◯ About 950 of Weiyue's intervals will contain the mean transaction amount of all transactions in the first random sample of 100 rows of `txn` Weiyue took.

◯ About 950 of Weiyue's intervals will contain M.

b) (3 pts) Weiyue repeats his entire process, except this time, he changes his sample size in step 1 from 100 to 400. Let B be the mean of the widths of the 1,000 new confidence intervals he creates.

What is the relationship between M and B?

◯ $M < B$ ◯ $M \approx B$ ◯ $M > B$

c) (3 pts) Weiyue repeats his entire process once again. This time, he still uses a sample size of 100 in step 1, but instead of creating 95% confidence intervals in step 3, he creates 99% confidence intervals. Let C be the mean of the widths of the 1,000 new confidence intervals he generates.

What is the relationship between M and C?

◯ $M < C$ ◯ $M \approx C$ ◯ $M > C$

d) (3 pts) Weiyue repeats his entire process one last time. This time, he still uses a sample size of 100 in step 1, and creates 95% confidence intervals in step 3, but instead of bootstrapping, he uses the Central Limit Theorem to generate his confidence intervals. Let D be the mean of the widths of the 1,000 new confidence intervals he creates.

What is the relationship between M and D?

◯ $M < D$ ◯ $M \approx D$ ◯ $M > D$

PID: _____

Question 11 (8 pts)

On Reddit, Yutian read that 22% of all online transactions are fraudulent. She decides to test the following hypotheses:

- **Null Hypothesis**: The proportion of online transactions that are fraudulent is **0.22**.
- **Alternative Hypothesis**: The proportion of online transactions that are fraudulent is not **0.22**.

To test her hypotheses, she decides to create a **95%** confidence interval for the proportion of online transactions that are fraudulent using the Central Limit Theorem.

Unfortunately, she doesn't have access to the entire `txn` DataFrame; rather, she has access to a simple random sample of `txn` of size n. In her sample, the proportion of transactions that are fraudulent is **0.2** (or equivalently, $\frac{1}{5}$).

a) (4 pts) The width of Yutian's confidence interval is of the form

$$\frac{c}{5\sqrt{n}}$$

where n is the size of her sample and c is some positive integer. What is the value of c? Give your answer as an integer.

Hint: Use the fact that in a collection of 0s and 1s, if the proportion of values that are 1 is p, the standard deviation of the collection is $\sqrt{p(1-p)}$.

$c =$ ☐

b) (4 pts) There is a positive integer J such that:

- If $n < J$, Yutian will fail to reject her null hypothesis at the **0.05** significance level.
- If $n > J$, Yutian will reject her null hypothesis at the **0.05** significance level.

What is the value of J? Give your answer as an integer.

$J =$ ☐

Question 12 (9 pts)

On Reddit, Keenan also read that 22% of all online transactions are fraudulent. He decides to test the following hypotheses at the **0.16 significance level**:

- **Null Hypothesis**: The proportion of online transactions that are fraudulent is **0.22**.
- **Alternative Hypothesis**: The proportion of online transactions that are fraudulent is not **0.22**.

Keenan has access to a simple random sample of `txn` of size **500**. In his sample, the proportion of transactions that are fraudulent is **0.23**.

Below is an incomplete implementation of the function `reject_null`, which creates a bootstrap-based confidence interval and returns **True** if the conclusion of Keenan's test is to **reject** the null hypothesis, and **False** if the conclusion is to **fail to reject** the null hypothesis, all at the **0.16** significance level.

```
def reject_null():
    fraud_counts = np.array([])
    for i in np.arange(10000):
        fraud_count = np.random.multinomial(500, __(a)__)[0]
        fraud_counts = np.append(fraud_counts, fraud_count)

    L = np.percentile(fraud_counts, __(b)__)
    R = np.percentile(fraud_counts, __(c)__)

    if __(d)__ < L or __(d)__ > R:
        # Return True if we REJECT the null.
        return True
    else:
        # Return False if we FAIL to reject the null.
        return False
```

Fill in the blanks below so that `reject_null` works as intended.

a: _____

b: _____ , c: _____

d: _____

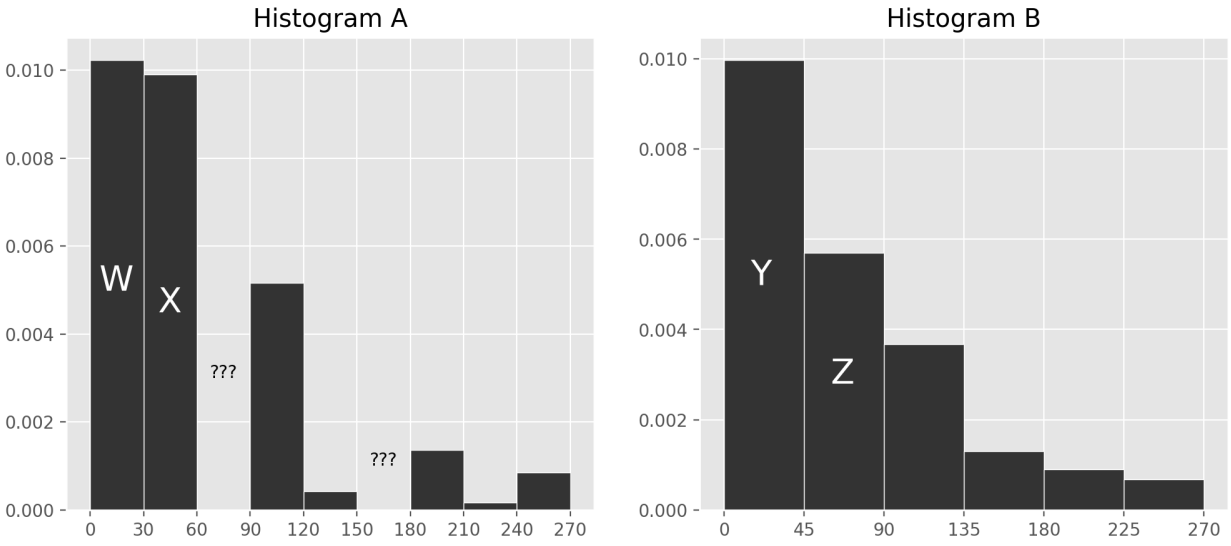
Hint: Your answer to (d) should be an integer greater than 50.

PID: _____

Question 13 (6 pts)

Ashley doesn't have access to the entire `txn` DataFrame; instead, she has access to a simple random sample of **400** rows of `txn`.

She draws two histograms, each of which depicts the distribution of the `"amount"` column in her sample, using different bins.



Unfortunately, DataHub is being finicky and so two of the bars in Histogram A are deleted after it is created.

a) (3 pts) In Histogram A, which of the following bins contains approximately 60 transactions?

◯ $[30, 60)$ ◯ $[90, 120)$ ◯ $[120, 150)$ ◯ $[180, 210)$

b) (3 pts) Let w, x, y, and z be the heights of bars W, X, Y, and Z, respectively. For instance, y is about 0.01.

Which of the following expressions gives the height of the bar corresponding to the $[60, 90)$ bin in Histogram A?

◯ $(y + z) - (w + x)$
◯ $(w + x) - (y + z)$
◯ $\frac{3}{2}(y + z) - (w + x)$
◯ $(y + z) - \frac{3}{2}(w + x)$
◯ $3(y + z) - 2(w + x)$
◯ $2(y + z) - 3(w + x)$
◯ None of the above.

Question 14 (13 pts)

As mentioned in the previous problem, Ashley has sample of 400 rows of `txn`. Coincidentally, in Ashley's sample of 400 transactions, the mean and standard deviation of the `"amount"` column both come out to 70 dollars.

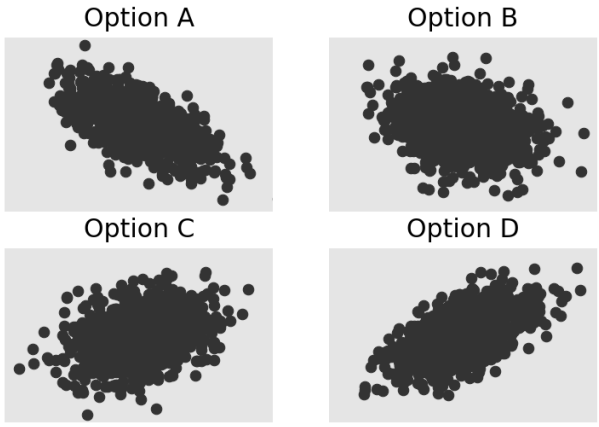
a) (3 pts) Fill in the blank:

"According to Chebyshev's inequality, at most 25 transactions in Ashley's sample are above ____ dollars; the rest must be below ____ dollars."

What goes in the blank? Give your answer as an **integer**. Both blanks are filled in with the same number.

Now, we're given that the mean and standard deviation of the `"lifetime"` column in Ashley's sample are both equal to c dollars. We're also given that the correlation between transaction amounts and lifetime spending in Ashley's sample is $-\frac{1}{4}$.

b) (2 pts) Which of the four options could be a scatter plot of lifetime spending vs. transaction amount?



◯ Option A ◯ Option B ◯ Option C ◯ Option D

Ashley decides to use linear regression to predict the lifetime spending of a card given a transaction amount.

c) (4 pts) The predicted lifetime spending, in **dollars**, of a card with a transaction amount of 280 dollars is of the form $f \cdot c$, where f is a fraction. What is the value of f? Give your answer as a simplified fraction.

$f =$ [ ]

d) (4 pts) Suppose the intercept of the regression line, when both transaction amounts and lifetime spending are measured in **dollars**, is 40. What is the value of c? Give your answer as an integer.

$c =$ [ ]

Feel free to draw us a picture or tell us about your fondest memory from DSC 10 this quarter.

Before turning in your exam, please make sure that your PID is on every page.

Congratulations on finishing DSC 10!

Data Overview: Fraudulent Transactions

Today, we're diving into the high-stakes world of fraud detection. Each row in the DataFrame `txn` represents one online transaction, or purchase. The `txn` DataFrame is indexed by `"transaction_id" (int)`, which is a unique identifier for a transaction. The columns of `txn` are as follows:

- `"is_fraud" (bool)`: `True` if the transaction was fraudulent and `False` if not.
- `"amount" (float)`: The dollar amount of the transaction.
- `"method" (str)`: The payment method; either `"debit"` or `"credit"`.
- `"card" (str)`: The payment network of the card used for the transaction; either `"visa"`, `"mastercard"`, `"american express"`, or `"discover"`.
- `"lifetime" (float)`: The total transaction amount of all transactions made with this card over its lifetime.
- `"browser" (str)`: The web browser used for the online transaction. There are 100 distinct browser types in this column; some examples include `"samsung browser 6.2"`, `"mobile safari 11.0"`, and `"chrome 62.0"`.

The first few rows of `txn` are shown below, though `txn` has 140,000 rows in total. Assume that the data in `txn` is a simple random sample from the much larger population of **all** online transactions.

transaction_id	is_fraud	amount	method	card	lifetime	browser
3061515	False	25.07	debit	visa	69306.00	chrome 63.0
3440724	False	19.80	debit	visa	224403.00	chrome 65.0 for android
3026777	False	25.00	credit	discover	75391.00	chrome 62.0
3122032	True	18.71	debit	visa	535891.00	chrome 63.0 for android
3111095	False	150.00	debit	visa	119037.00	edge 15.0
3041425	False	50.00	debit	visa	46021.00	chrome 62.0
3086600	False	58.42	credit	mastercard	205286.00	ie 11.0 for desktop
3380448	False	160.72	credit	visa	230157.00	chrome 65.0

Throughout this exam, assume that we have already run `import babypandas as bpd` and `import numpy as np`.