Lecture 14 – Midterm Review
DSC 10, Fall 2024

Announcements

- Discussion section is **today**.
- The Midterm Exam is **Wednesday during lecture at 9AM, 10AM, or 1PM**. See [this post on Ed](#) for lots of details, including what is covered, what to bring, and how to study.
- The Midterm Project is due on **Sunday at 11:59PM**. Only one partner needs to submit.

Agenda

- We'll work through selected problems from the Winter 2024 Midterm.
- We won't write any code, since you can't run code during the exam. Instead, we'll try to think like the computer ourselves.
- These annotated slides will be posted after lecture is over.
- **Try the problems with us!**

Winter 2024 Midterm

Access the exam here. Make sure to read the data info sheet at the top before starting.

Clue: The Murder Mystery Game

Clue is a murder mystery game where players use the process of elimination to figure out the details of a crime. The premise is that a murder was committed inside a large home, by one of **6 suspects**, with one of **7 weapons**, and in one of **9 rooms**.

The game comes with **22 cards**, one for each of the 6 suspects, 7 weapons, and 9 rooms. To set up the game, one suspect card, one weapon card, and one room card are chosen randomly, without being looked at, and placed aside in an envelope. The cards in the envelope represent the details of the murder: who did it, with what weapon, and in what room.

The remaining 19 cards are randomly shuffled and dealt out to the players (as equally as possible). Players then look at the cards they were dealt and can conclude that any cards they see were **not** involved in the murder. In the gameplay, players take turns moving around to different rooms of the house on the gameboard, which gives them opportunities to see cards in other players' hands and further eliminate suspects, weapons, and rooms. The first player to narrow it down to one suspect, with one weapon, and in one room can make an accusation and win the game!

Suppose Janine, Henry, and Paige are playing a game of Clue. Janine and Paige are each dealt 6 cards, and Henry is dealt 7. The DataFrame `clue` has 22 rows, one for each card in the game. `clue` represents **Janine's knowledge** of who is holding each card. `clue` is indexed by `"Card"`, which contains the name of each suspect, weapon, and room in the game. The `"Category"` column contains `"suspect"`, `"weapon"`, or `"room"`. The `"Cardholder"` column contains `"Janine"`, `"Henry"`, `"Paige"`, or `"Unknown"`.

Since Janine's knowledge is changing throughout the game, the `"Cardholder"` column needs to be updated frequently. At the beginning of the game, the `"Cardholder"` column contains only `"Janine"` and `"Unknown"` values. We'll assume throughout this exam that `clue` contains Janine's current knowledge at an arbitrary point in time, not necessarily at the beginning of the game. For example, `clue` **may look like** the DataFrame at right.

Note: Throughout the exam, assume we have already run `import babypandas as bpd` and `import numpy as np`.

Card	Category	Cardholder
Col. Mustard	suspect	Unknown
Dr. Orchid	suspect	Henry
Miss Scarlett	suspect	Henry
Mr. Green	suspect	Paige
Mrs. Peacock	suspect	Unknown
...	...	
hall	room	Janine
kitchen	room	Janine
library	room	Unknown
lounge	room	Janine
study	room	Unknown

22 rows × 2 columns

About the Game

- 22 cards, in three categories
  - 6 suspects
  - 7 weapons
  - 9 rooms
- Envelope: one card of each category
- Janine: 6 cards
- Paige: 6 cards
- Henry: 7 cards

"suspect" ≥ "this"
False

Question 1 (18 pts)

Each of the following expressions evaluates to an integer. Determine the value of that integer, if possible, or circle "not enough information."

(handwritten: 5-row df)
(handwritten: 5, 8, 11, 14, 17)

a) `clue.get("Cardholder") == "Janine").sum()`

(handwritten: # Trues)

6

not enough information

d) `len(clue.take(np.arange(5, 20, 3)).index)`

5

not enough information

(handwritten: T/F's)

b) `np.count_nonzero(clue.get("Category").str.contains("p"))`

13

not enough information

(handwritten: 7-row df)

e) `len(clue[clue.get("Category") >= "this"].index)`

7

not enough information

c) `clue[(clue.get("Category") == "suspect") & (clue.get("Cardholder") == "Janine")].shape[0]`

(handwritten: query)

not enough information *(circled)*

f) `clue.groupby("Cardholder").count().get("Category").sum()`

(handwritten: ?)

22

not enough information

Question 3 (8 pts)

An important part of the game is knowing when you've narrowed it down to just one suspect with one weapon in one room. Then you can make your accusation and win the game!

Suppose the DataFrames `grouped` and `filtered` are defined as follows.

```
grouped = (clue.reset_index()
               .groupby(["Category", "Cardholder"])
               .count()
               .reset_index())
filtered = grouped[grouped.get("Cardholder") == "Unknown"]
```

a) (4 pts) Fill in the blank below so that `"Ready to accuse"` is printed when Janine has enough information to make an accusation and win the game.

```
if filtered.get("Card")._____ == 3:
    print("Ready to accuse")
```
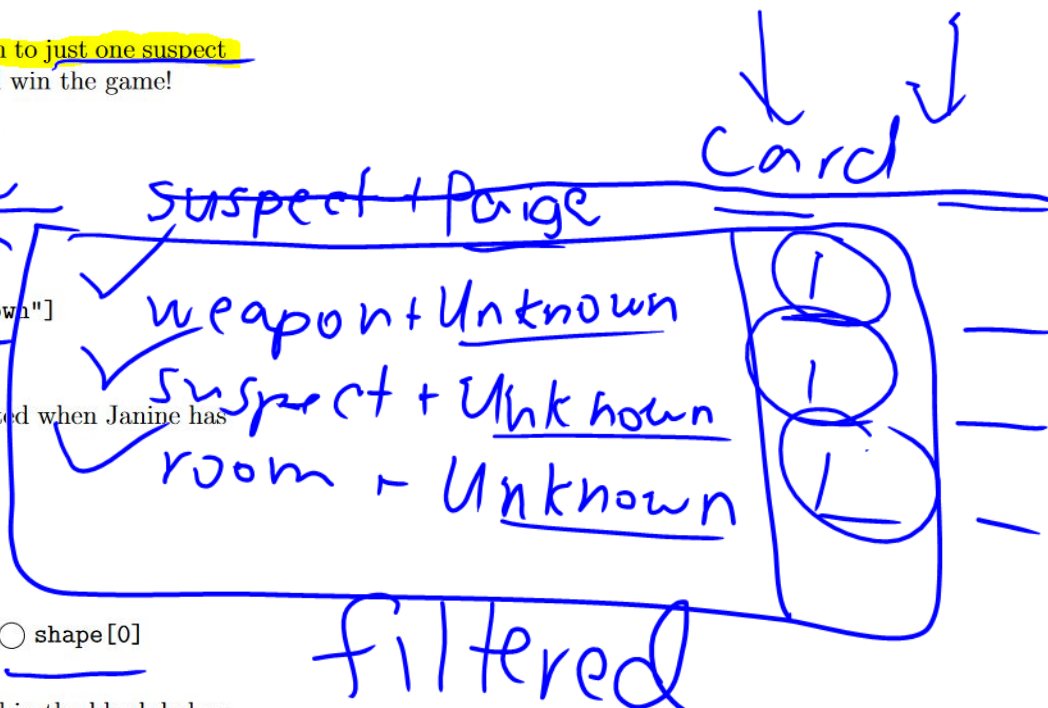
What goes in the blank?

○ count()   ○ sum()   ○ max()   ○ min()   ○ shape[0]

b) (4 pts) Now, let's look at a different way to do the same thing. Fill in the blank below so that `"Ready to accuse"` is printed when Janine has enough information to make an accusation and win the game.

```
if filtered.get("Card")._____ == 1:
    print("Ready to accuse")
```

What goes in the blank?

○ count()   ○ sum()   ○ max()   ○ min()   ○ shape[0]

```
1 grouped = clue.reset_index().groupby(['Category', 'Cardholder']).count().reset_index()
```

	Card	Category	Cardholder
0	Col. Mustard	suspect	Unknown
1	Dr. Orchid	suspect	Henry
2	Miss Scarlett	suspect	Henry
3	Mr. Green	suspect	Paige
4	Mrs. Peacock	suspect	Unknown
...	...	...	...
17	hall	room	Janine
18	kitchen	room	Janine
19	library	room	Unknown
20	lounge	room	Janine
21	study	room	Unknown

		Card
Category	Cardholder	
room	Janine	4
	Unknown	5
suspect	Henry	2
	Paige	1
	Unknown	3
weapon	Janine	2
	Paige	1
	Unknown	4

	Category	Cardholder	Card
0	room	Janine	4
1	room	Unknown	5
2	suspect	Henry	2
3	suspect	Paige	1
4	suspect	Unknown	3
5	weapon	Janine	2
6	weapon	Paige	1
7	weapon	Unknown	4

```
2 filtered = grouped[grouped.get('Cardholder') == 'Unknown']
```

	Category	Cardholder	Card
1	room	Unknown	5
4	suspect	Unknown	3
7	weapon	Unknown	4

(handwritten annotation) .shape[0] = 3 but not ready to accuse

Question 4 (7 pts)

When someone is ready to make an accusation, they make a statement such as:

> *"It was Miss Scarlett with the dagger in the study"*

[handwritten annotation: goal]

While the suspect, weapon, and room may be different, an accusation will always have this form:

> *"It was* _____ *with the* _____ *in the* _____ *"*

[handwritten annotation: answers may be]
[handwritten: answers = np.array(["Miss Scarlett", "dagger", "study"])]

Suppose the array `words` is defined as follows (note the spaces).

```
words = np.array(["It was ", " with the ", " in the "])
```
[handwritten annotations under words: 0, 1, 2]

Suppose another array called `answers` has been defined. `answers` contains three elements: the name of the suspect, weapon, and room that we would like to use in our accusation, in that order. Using `words` and `answers`, complete the `for`-loop below so that `accusation` is a string, formatted as above, that represents our accusation.

```
accusation = ""
for i in ___(a)___:
    accusation = ___(b)___
```

a) (3 pts) What goes in blank (a)?

[handwritten: [0, 1, 2]]
[handwritten: np.arange(len(words))]
[handwritten: range(len(words))]

b) (4 pts) What goes in blank (b)?

[handwritten: accusation + words[i] + answers[i]]

Question 5 (12 pts)

Recall that the game *Clue* comes with 22 cards, one for each of the 6 suspects, 7 weapons, and 9 rooms. One suspect card, one weapon card, and one room card are chosen randomly, without being looked at, and placed aside in an envelope. The remaining 19 cards (5 suspects, 6 weapons, 8 rooms) are randomly shuffled and dealt out, splitting them as evenly as possible among the players. Suppose in a three-player game, Janine gets 6 cards, which are dealt one at a time.

Answer the probability questions that follow. Leave your answers **unsimplified**.

a) (4 pts) Cards are dealt one at a time. What is the probability that the first card Janine is dealt is a weapon card?

$$6/19$$

b) (4 pts) What is the probability that all 6 of Janine's cards are weapon cards?

$$6/19 * 5/18 * 4/17 * 3/16 * 2/15 * 1/14$$

$$6/19 * 5/18 * \ldots$$

P(first W)   P(2ⁿᵈ W given 1ˢᵗ W)

c) (4 pts) Determine the probability that exactly one of the first two cards Janine is dealt is a weapon card. This probability can be expressed in the form

Cases: W, NW
NW, W

$$\frac{k \cdot (k+1)}{m \cdot (m+1)}$$

$$6/19 * \frac{13}{18} +$$

$$\frac{13}{19} * \frac{6}{18}$$

$$= 2 \cdot \frac{6 \cdot 13}{18 \cdot 19}$$

$$= \frac{12 \cdot 13}{18 \cdot 19}$$

where k and m are **integers**. What are the values of k and m?

Hint: There is no need for any sort of calculation that you can't do easily in your head, such as long division or multiplication.

Question 8 (8 pts)

The histogram below shows the distribution of game times in minutes for both two-player and three-player games of *Clue*, with each distribution representing 1000 games played.


Histogram of Clue Board Game Times
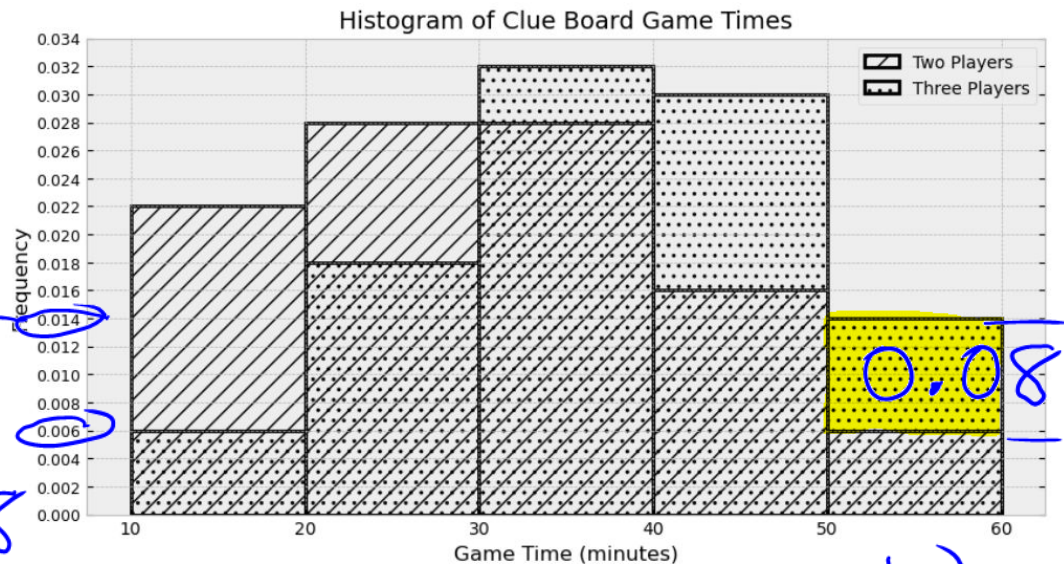
a) (4 pts) How many **more** three-player games than two-player games took at least 50 minutes to play? Give your answer as an ~~integer,~~ **rounded to the nearest multiple of 10**.

Question 8 (8 pts)

The histogram below shows the distribution of game times in minutes for both two-player and three-player games of *Clue*, with each distribution representing 1000 games played.

Histogram of Clue Board Game Times
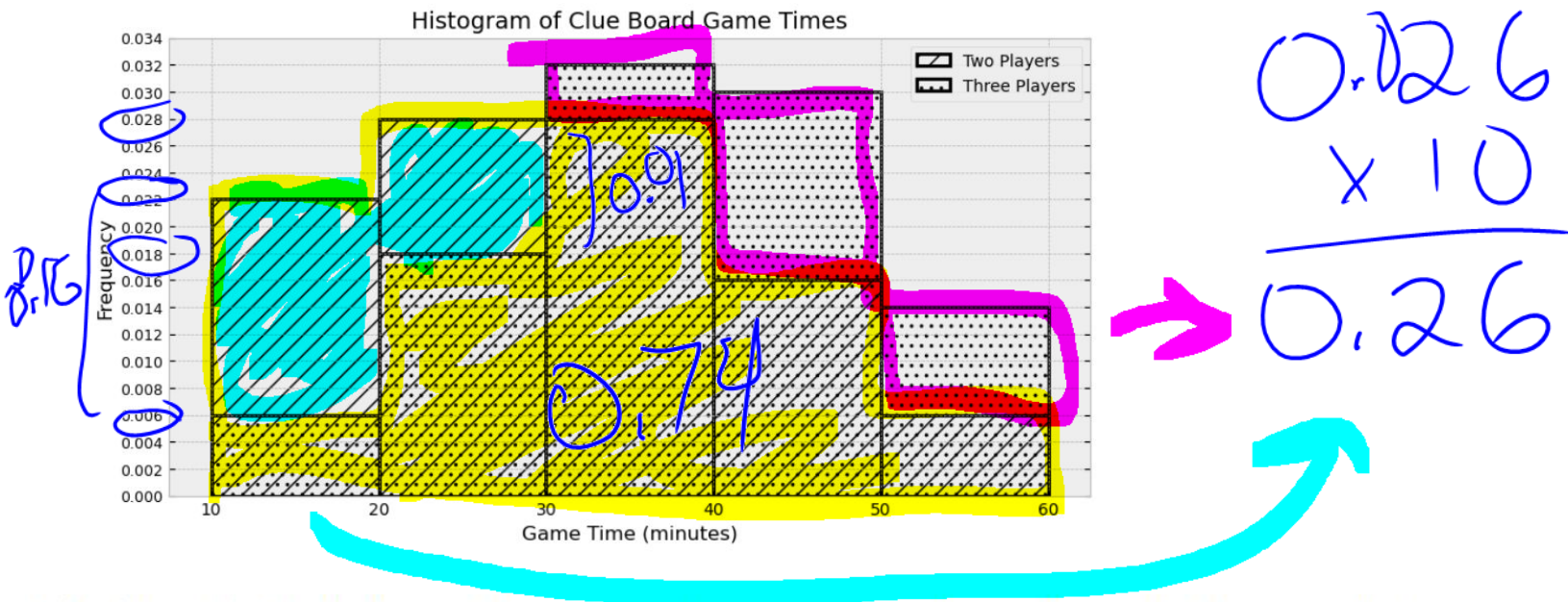
b) (4 pts) Calculate the approximate area of overlap of the two histograms. Give your answer as a **proportion between 0 and 1, rounded to two decimal places**.

0.74