15-110 F25 Check6-2 - Written Portion

Name:

AndrewID:

Complete the following problems in the fillable PDF, or print out the PDF, write your answers by hand, and scan the results.

When you are finished, upload your check6-2.pdf to **Check6-2 - Written** on Gradescope. Make sure to upload your Check6-2 work for the Hw6 project as well!

Written Problems

#1 - Visualizations - 8pts

Can attempt after Data Analysis II lecture

Given the description of the data below, what visualization(s) would be **most** appropriate to use? If multiple visualizations are equally valid, select **all** that apply.

Compare quiz averages (85.4, 91.2, etc) to final grades (92, 87, etc).

- ☐ Bar Chart
- ☐ Box and Whiskers Plot
- ☐ Bubble Plot
- ☐ Colored Scatter Plot
- ☐ Histogram
- ☐ Pie Chart
- ☐ Scatter Plot
- ☐ Scatter Plot Matrix

Investigate the counts of letter grades (A, B, C, etc) in a class

- ☐ Bar Chart
- ☐ Box and Whiskers Plot
- ☐ Bubble Plot
- ☐ Colored Scatter Plot
- ☐ Histogram
- ☐ Pie Chart
- ☐ Scatter Plot
- ☐ Scatter Plot Matrix

Compare sports categories (soccer, basketball, etc) to athlete heights (68", 74", etc).

- ☐ Bar Chart
- ☐ Box and Whiskers Plot
- ☐ Bubble Plot
- ☐ Colored Scatter Plot
- ☐ Histogram
- ☐ Pie Chart
- ☐ Scatter Plot
- ☐ Scatter Plot Matrix

Compare athlete heights (68", 74", etc) to ages (24, 26, etc) based on sports categories (soccer, basketball, etc).

- ☐ Bar Chart
- ☐ Box and Whiskers Plot
- ☐ Bubble Plot
- ☐ Colored Scatter Plot
- ☐ Histogram
- ☐ Pie Chart
- ☐ Scatter Plot
- ☐ Scatter Plot Matrix

#2 - Matplotlib - 8pts

Can attempt after Data Analysis II lecture

Assume that you have already set up the variable `data`. This holds a 2D list where each inner list contains three items: a person's gender, height, and weight. Gender is a string (`"female"`, `"male"`, or `"nb"` [non-binary]); height and weight are positive numbers.

For example, this might look like:
`data = [ ['male', 68, 190], ['female', 73, 225], ['nb', 59, 134], ...]`

If you run the following code on `data`, what are the properties of the chart it will produce? Make sure to describe **all** the relevant aspects of the chart.

```
heights = { "female" : [], "male" : [], "nb" : [] }
weights = { "female" : [], "male" : [], "nb" : [] }
for entry in data:
    heights[entry[0]].append(entry[1])
    weights[entry[0]].append(entry[2])

import matplotlib.pyplot as plt
plt.scatter(heights["female"], weights["female"], c="green",  label="female")
plt.scatter(heights["male"],   weights["male"],   c="red",    label="male")
plt.scatter(heights["nb"],     weights["nb"],     c="purple", label="nb")

plt.xlabel("Height (in)")
plt.ylabel("Weight (lb)")
plt.legend()
plt.show()
```

Hint: you probably don't know some of these methods or keyword arguments yet, and that's okay! Use the matplotlib documentation to look them up.

#3 - Simulation Code - 4pts

Can attempt after Simulation II lecture

Recall the moving circle simulation from Check 6-1. We now want to update this simulation so that when the user clicks on the circle or presses 'Enter', the circle moves back to the left side of the screen.

For the additional parts of the simulation (the Event Rules), select the single line of code that needs to be included in that part.

Hint: if you're not sure, try implementing this using the simulation starter code!

How would you check if the user clicked in the circle in `mousePressed(data, event)`?

☐ `((data["left"]+20 - data["x"])**2 + (220 - data["y"])**2)**0.5 <= 20`
☐ `((data["left"]+20 - event.x)**2 + (220 - event.y)**2)**0.5 <= 20`
☐ `(data["x"]**2 + data["y"]**2)**0.5 <= 20`
☐ `(event.x**2 + event.y**2)**0.5 <= 20`

How would you check if the user pressed "Enter" in `keyPressed(data, event)`?

☐ `if (data["char"] == "Return"):`
☐ `if (data["keysym"] == "Return"):`
☐ `if (event.char == "Return"):`
☐ `if (event.keysym == "Return"):`

#4 - Monte Carlo Methods - 9pts

Can attempt after Simulation II lecture

For each of the following questions, use **Monte Carlo methods** to find the answer to the given question. You can use the `monteCarlo(trials)` function from the notes to average results over `100,000` trials; you just need to update the `runTrial()` function for each question.

Please submit your answer as a decimal probability (like 0.45; 100% = 1, 50% = 0.5), and round your answer for each question to have only 2 digits after the decimal point.

A) What is the probability that, if you roll a die twice, the second roll will be either 2 larger or 2 smaller than the first? For example, you could roll a 4 and then a 6, or a 4 and then a 2.

B) Pick a random odd number between 1 and 99 (inclusive on both). What is the probability that that number is a multiple of 7?

Hint: make a list of all odd numbers between 1 and 99, then use `random.choice()`

C) Make a list with six values (two `"red"`, two `"green"`, two `"blue"`) and shuffle it. What is the probability that the first two values in the list are both `"red"`?

Hint: use the destructive function `random.shuffle()`

#5 - Machine Learning Process - 5pts

Can attempt after Machine Learning lecture

Imagine a scenario where Bill wants to train a machine learning algorithm to identify which pictures on the internet have cats in them. He downloads 1,000 pictures of cats and other animals from the internet, decides to use a basic image recognition algorithm which will identify important features, trains on all 1,000 pictures, then tests his on a quarter of that dataset (250 pictures). He finds that his algorithm has a 97% success rate, which he publishes on his blog.

Bill made a few mistakes in this process. **What was his <u>biggest</u> mistake?**

#6 - Game Trees - 11pts

Can attempt after Artificial Intelligence lecture

Nim (https://en.wikipedia.org/wiki/Nim) is a simple game for two players. The game starts with a pot containing some number of marbles on the table. Players take turns removing marbles from the pot. Each player must choose to remove 1, 2, or 3 marbles on their turn. Whoever removes the last marble loses.

Assume you want to build a basic AI agent that can play Nim using a game tree. In this game, the pot will start with 16 marbles, and the **state** of the game is the number of marbles in the pot. On the next page, draw the root node and the first two levels of the game tree (you do not have to draw any levels past that), with the game state (number of marbles) as the value of each node. Annotate your game tree to show which actions are taken by the AI agent vs. the opponent, assuming the agent gets the first turn.

You can do this with a picture of a physical drawing or an online image editing tool (like Google Drawings). To upload the image, use the same approach you used on Check5.

What is the maximum **depth** of the game tree if it was built out fully? (A node with only the root has depth 1).

Assume that the AI uses minimax to find the best action to take on its turn. When comparing the results on the level right below the root, should the AI pass the **maximum** or **minimum** result to the top level with the root?

- ☐ Maximum
- ☐ Minimum

Click here to add your image

If that doesn't work, use a PDF tool to add your image manually.