CISC 3225 Midterm Name: _____
Spring 2024
100 points

**1. True/False (10 points)**

**T**___ The result of a mathematical operation on NaN is NaN.

**F**___ Similar to Python lists, appending Series B to Series A will modify Series A in-place.

**T**___ GroupBy objects are iterable (i.e., they can be used in for loops)

**F**___ A Series is the equivalent of a 2-dimensional NumPy array.

**T**___ On a DataFrame, both columns and rows can be accessed with .iloc.

**F**___ Attempting to concatenate two DataFrames with the same index will always raise an exception.

**F**___ Missing values will negatively impact basic DataFrame statistics, and must be removed before calling describe() or making a correlation matrix.

**T**___ A correlation matrix can show whether the data in pairs of columns are related.

**F**___ A header giving column names is required in all CSV files.

**T**___ Despite its name, the Pandas module was not written by actual pandas.

**Multiple Choice (18 points)**

2. (3 points) A dataset of business phone numbers has the columns “name”, “area_code” and “phone_number” sorted by name. Assuming you can create an appropriate index, which of the following will help you find all businesses in the 404 area code? **Select one.**

**a) .loc** b) .iloc

3. (3 points) DataFrame A contains grocery prices in NYC-area grocery stores. DataFrame B contains personal care item prices in NYC-area drug stores. Assuming A and B have similar structures, which is the best way to combine them so you can study the cost of living in the NYC area? **Select one**.

a) Merging

**b) Concatenation**

c) Masking

d) Grouping

4. (3 points) Select the conditions that *must* be present for None to appear in a Series object. **Select all that apply.**

a) The Series is initialized with a list containing a NaN value.

**b) The Series is initialized with a list containing a None value.**

c) The Series is initialized with a list containing an integer value.

d) The Series is initialized with a list containing an floating point value.

**e) The Series is initialized with a list containing a string value.**

For questions 5-9, refer to the DataFrames shown below. When answering each question, base your answer *only* on the contents of each DataFrame as shown.

| | isbn | title |
|---|---|---|
| 0 | 119935 | Jane Eyre |
| 1 | 559332 | Fahrenheit 451 |
| 2 | 660352 | Lord of the Rings |
| 3 | 2118532 | 1984 |

books

| | isbn | genre |
|---|---|---|
| 0 | 119935 | romance |
| 1 | 559332 | dystopian |
| 2 | 660352 | fantasy |
| 3 | 2118532 | dystopian |

genres

| | isbn | name |
|---|---|---|
| 0 | 119935 | Matt |
| 1 | 559332 | Alice |
| 2 | 660352 | Matt |
| 3 | 2118532 | Bob |
| 4 | 119935 | Alice |

read

| | name | genre |
|---|---|---|
| 0 | Matt | sci-fi |
| 1 | Alice | fantasy |
| 2 | Bob | history |
| 3 | Bob | nonfiction |
| 4 | Matt | dystopian |

favorite_genre

5. (3 points) What type is the merge across books and genres?

**a) One-to-one** b) Many-to-one c) Many-to-many

6. (3 points) What type is the merge across genre and favorite_genre?

a) One-to-one **b) Many-to-one** c) Many-to-many

7. (3 points) What type is the merge across genres and read?

a) One-to-one **b) Many-to-one** c) Many-to-many

| | isbn | genre |
|---|---|---|
| 0 | 119935 | romance |
| 1 | 559332 | dystopian |
| 2 | 660352 | fantasy |
| 3 | 2118532 | dystopian |

genres

| | name | genre |
|---|---|---|
| 0 | Matt | sci-fi |
| 1 | Alice | fantasy |
| 2 | Bob | history |
| 3 | Bob | nonfiction |
| 4 | Matt | dystopian |

favorite_genre

**Short Answer and Data Problems**

8. (6 points) Perform a right merge between genres (left) and favorite_genre (right).

| | name | genre | isbn |
|---|---|---|---|
| 0 | Matt | sci-fi | NaN |
| 1 | Alice | fantasy | 660352 |
| 2 | Bob | history | NaN |
| 3 | Bob | nonfiction | NaN |
| 4 | Matt | dystopian | 559332 |
| 5 | Matt | dystopian | 2118532 |

9. (6 points) Perform an inner merge between genres (left) and favorite_genre (right).

| | isbn | genre | name |
|---|---|---|---|
| 1 | 559332 | dystopian | Matt |
| 2 | 660352 | fantasy | Alice |
| 3 | 2118532 | dystopian | Matt |

10. (6 points) Do Python logical operators (and, or, not) work with Series objects? Why or why not?

Python logical operators do not work with Series objects. A logical operator expects to receive Boolean values on both sides. If an object on either side of a logical operator is not a Boolean value, Python will attempt to convert it. However, a Series object cannot be converted to a Boolean value because it is not a singular object: it consists of many objects, one at each index.

Even if the Series object consists of Boolean values, there are still many of them, and they do not all have to be the same. For example, how should Python convert the series [True, True, False] to a single True/False value? It is not possible to do so.

| | **a** | **b** |
|---|---|---|
| 0 | 9 | 4 |
| 1 | 8 | 4 |
| 3 | 5 | 1 |

| | **c** |
|---|---|
| 2 | 100 |
| 0 | 200 |
| 5 | 300 |

11. (6 points) Concatenate the DataFrames shown above along axis=1.

| | **a** | **b** | **c** |
|---|---|---|---|
| 0 | 9 | 4 | 200.0 |
| 1 | 8 | 4 | NaN |
| 3 | 5 | 1 | NaN |
| 2 | NaN | NaN | 100.0 |
| 5 | NaN | NaN | 300.0 |

12. (6 points) Concatenate the DataFrames shown above along axis=0.

| | **a** | **b** | **c** |
|---|---|---|---|
| 0 | 9 | 4 | NaN |
| 1 | 8 | 4 | NaN |
| 3 | 5 | 1 | NaN |
| 2 | NaN | NaN | 100.0 |
| 0 | NaN | NaN | 200.0 |
| 5 | NaN | NaN | 300.0 |

Use the DataFrame below for questions 13. Note that tip is a percentage, and is not factored into the order total.

| | day | table | waitstaff | party_size | order_total | tip |
|---|---|---|---|---|---|---|
| 0 | 1 | 4 | M | 3 | 74 | .2 |
| 1 | 1 | 2 | L | 2 | 50 | .15 |
| 2 | 1 | 6 | M | 2 | 44 | .17 |
| 3 | 1 | 4 | D | 5 | 104 | .21 |
| 4 | 2 | 5 | D | 4 | 98 | .19 |
| 5 | 2 | 3 | L | 6 | 140 | .21 |
| 6 | 2 | 3 | L | 2 | 25 | .22 |
| 7 | 2 | 5 | M | 2 | 30 | .21 |
| 8 | 3 | 2 | D | 3 | 38 | .19 |
| 9 | 3 | 1 | D | 3 | 40 | .2 |

13. (15 points) With grouping and/or other DataFrame operations as needed, show how to find the following pieces of information. Each operation should output only the requested answer. You do not need to give the final result. Answer each question in 1-2 lines.

How much money (excluding tips) did each table earn on Day 1?

**df[df[‘day’] == 1].groupby(‘table’)[‘order_total’].sum()**

How many customers did each member of the waitstaff interact with over all 3 days of service?

**df.groupby(‘waitstaff’)[‘party_size’].sum()**

What is the total dollar amount of tips earned by each member of the waitstaff?

**df[‘tip_amount’] = df[‘tip’] * df[‘order_total’]**
**df.groupby(‘waitstaff’)[‘tip_amount’].sum()**

What is the total dollar amount of tips earned by M on day 1?

**df.loc[(df[‘waitstaff’] == ‘M’) & (df[‘day’] == 1), ‘tip_amount’].sum()**

If you were to visit this restaurant by yourself, approximately how much would you expect to pay for your meal (excluding tip)?

**Answers vary**
**df[‘order_total’].sum() / df[‘party_size’].sum()**

Below are the shapes of various NumPy arrays. Indicate whether it is possible to broadcast them together for arithmetic operations. If it is possible, give the final shape of both arrays. If it is impossible, show the shape of the arrays with as many broadcasting rules applied as possible, and indicate where Rule 3 applies.

14. (6 points)

a.shape = (3, 3, 1, 3)
b.shape = (3, 1)

**Rule 1 (b):**
**a.shape = (3, 3, 1, 3)**
**b.shape = (1, 1, 3, 1)**

**Rule 2 (a):**
**a.shape = (3, 3, 3, 3)**
**b.shape = (1, 1, 3, 1)**

**Rule 2 (b):**
**a.shape = (3, 3, 3, 3)**
**b.shape = (3, 3, 3, 3)**

**It is possible to broadcast a and b.**

15. (6 points)

a.shape = (1, 2, 3, 4)
b.shape = (2, 1)

**Rule 1 (b):**
**a.shape = (1, 2, 3, 4)**
**b.shape = (1, 1, 2, 1)**

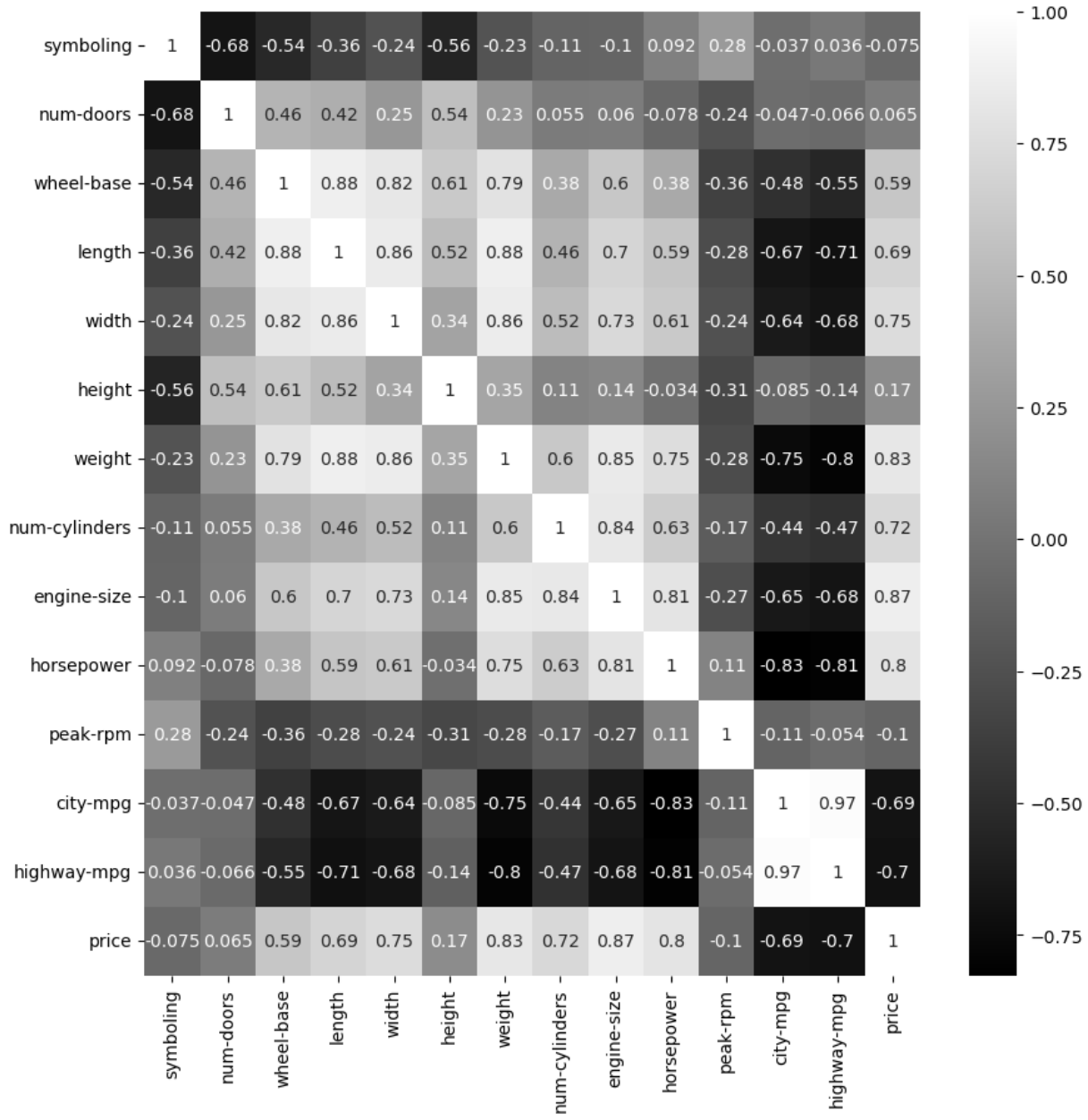
**Rule 2 (a): No change**

**Rule 2 (b):**
**a.shape = (1, 2, 3, 4)**
**b.shape = (1, 2, 2, 4)**

**It is impossible to broadcast a and b. Rule 3 applies because a and b do not match in the 3rd dimension (3 != 2)**

Below is a correlation matrix from a dataset of cars. Most variables relate directly to basic facts about each car, but some require additional explanation:

- wheel-base: The distance between the front and rear wheels.
- symboling: A rating indicating how risky a car is to insure. Positive numbers indicate greater risk, while negative numbers indicate less risk.



16. (5 points) City MPG and Highway MPG refer to how many miles a car can travel on one gallon of gas in each environment (higher is better). In general, what is the difference between at car with high MPG and low MPG? Why?

**Answers vary**

**In general, a car with high MPG will have a lower wheel base, length, width, weight, num-cylinders, engine-size, horsepower, and price. This suggests that cars with high MPG are small, not very powerful, and cheap. In contrast, cars with low MPG are large, powerful, and expensive.**

**This makes sense because large cars have more mass that the engine has to move. It is going to expend more energy moving the car, consuming more gas per mile. In contrast, small cars have less mass to move, and so requires less energy for the same mile.**

17. (5 points) This dataset contains many strong correlations. Are all the strong correlations actually useful? Identify two strong correlations where the pair of variables essentially give the same information.

**Answers vary**

**Wheel base and length: Both are related to the length of the car. If a car has wheels which are very far apart from one another, it is necessarily longer. In contrast, if a car's wheels are very close together, the car cannot be too long.**

**City MPG and Highway MPG: Both measure miles per gallon. It makes sense that MPG would be somewhat different in different driving conditions, but ultimately both are a reflection of the car's performance characteristics – a high-MPG car will have high city/highway MPG, and a low-MPG car will have low city/highway MPG.**

18. (5 points) Explain car pricing. What is different about an expensive car versus a cheap car? Why do you think this difference exists?

**Answers vary**

**In general, a high-priced car has a larger wheel base and larger dimensions, weighs more, has a larger engine (engine size and cylinders), has more horsepower, and is less efficient (low MPG).**

**High-end expensive sports cars might not be designed with fuel efficiency in mind. The main design goals are power and speed, which explains the positive relationship between price and engine characteristics.**

**Another point that could explain the price is that larger cars cost more because more raw materials are required to manufacture the car.**