Final Exam – DSC 10, Winter 2022

Q1 WNBA 🏀

12 Points

Welcome to the Final Exam for DSC 10 this quarter! In this exam, we will use data from the 2021 Women's National Basketball Association (WNBA) season. In basketball, players score points by shooting the ball into a hoop. The team that scores the most points wins the game.

Kelsey Plum, a WNBA player, attended La Jolla Country Day School, which is adjacent to UCSD's campus. Her current team is the Las Vegas Aces (three-letter code `'LVA'`). **In 2021, the Las Vegas Aces played 31 games, and Kelsey Plum played in all 31.**

The DataFrame `plum` contains her stats for all games the Las Vegas Aces played in 2021. The first few rows of `plum` are shown below (though the full DataFrame has 31 rows, not 5):

	Date	Opp	Home	Won	PTS	AST	TOV
0	2021-05-15	SEA	False	False	11	4	1
1	2021-05-18	SEA	False	True	10	3	0
2	2021-06-03	NYL	False	True	4	2	2
3	2021-06-05	WAS	False	True	2	2	1
4	2021-06-13	DAL	True	True	13	2	2

Each row in `plum` corresponds to a single game. For each game, we have:

- `'Date'` (`str`), the date on which the game was played
- `'Opp'` (`str`), the three-letter code of the opponent team
- `'Home'` (`bool`), `True` if the game was played in Las Vegas ("home") and `False` if it was played at the opponent's arena ("away")
- `'Won'` (`bool`), `True` if the Las Vegas Aces won the game and `False` if they lost
- `'PTS'` (`int`), the number of points Kelsey Plum scored in the game
- `'AST'` (`int`), the number of assists (passes) Kelsey Plum made in the game
- `'TOV'` (`int`), the number of turnovers Kelsey Plum made in the game (a turnover is when you lose the ball – turnovers are bad!)

Q1.1

1 Point

What type of visualization is best suited for visualizing the trend in the number of points Kelsey Plum scored per game in 2021?

- ○ Histogram
- ○ Bar chart
- ○ Line chart
- ○ Scatter plot

Q1.2

3 Points

Fill in the blanks below so that `total_june` evaluates to the total number of points Kelsey Plum scored in June.

```
june_only = plum[__(a)__]
total_june = june_only.__(b)__
```

What goes in blank (a)?

What goes in blank (b)?

Q1.3

1 Point

Consider the function `unknown`, defined below.

```
def unknown(df):
    grouped = plum.groupby('Opp').max().get(['Date', 'PTS'])
    return np.array(grouped.reset_index().index)[df]
```

What does `unknown(3)` evaluate to?

- `'2021-06-05'`
- `'WAS'`
- The date on which Kelsey Plum scored the most points
- The three-letter code of the opponent on which Kelsey Plum scored the most points
- The number 0
- The number 3
- An error

Q1.4

2 Points

For your convenience, we show the first few rows of `plum` again below.

	Date	Opp	Home	Won	PTS	AST	TOV
0	2021-05-15	SEA	False	False	11	4	1
1	2021-05-18	SEA	False	True	10	3	0
2	2021-06-03	NYL	False	True	4	2	2
3	2021-06-05	WAS	False	True	2	2	1
4	2021-06-13	DAL	True	True	13	2	2

Suppose that Plum's team, the Las Vegas Aces, won at least one game in Las Vegas and lost at least one game in Las Vegas. Also, suppose they won at least one game in an opponent's arena and lost at least one game in an opponent's arena.

Consider the DataFrame `home_won`, defined below.

```
home_won = plum.groupby(['Home', 'Won']).mean().reset_index()
```

How many rows does `home_won` have?

How many columns does `home_won` have?

Q1.5

1 Point

Consider the DataFrame `home_won` once again.

```
home_won = plum.groupby(['Home', 'Won']).mean().reset_index()
```

Now consider the DataFrame `puzzle`, defined below. Note that the only difference between `home_won` and `puzzle` is the use of `.count()` instead of `.mean()`.

```
puzzle = plum.groupby(['Home', 'Won']).count().reset_index()
```

How do the number of rows and columns in `home_won` compare to the number of rows and columns in `puzzle`?

- `home_won` and `puzzle` have the same number of rows and columns
- `home_won` and `puzzle` have the same number of rows, but a different number of columns
- `home_won` and `puzzle` have the same number of columns, but a different number of rows
- `home_won` and `puzzle` have both a different number of rows and a different number of columns

Q1.6

3 Points

For your convenience, we show the first few rows of `plum` again below.

	Date	Opp	Home	Won	PTS	AST	TOV
0	2021-05-15	SEA	False	False	11	4	1
1	2021-05-18	SEA	False	True	10	3	0
2	2021-06-03	NYL	False	True	4	2	2
3	2021-06-05	WAS	False	True	2	2	1
4	2021-06-13	DAL	True	True	13	2	2

There is exactly one team in the WNBA that Plum's team did not win any games against during the 2021 season. Fill in the blanks below so that `never_beat` evaluates to a string containing the three-letter code of that team.

```
never_beat = plum.groupby(__(a)__).sum().__(b)__
```

What goes in blank (a)?

What goes in blank (b)?

Q1.7

1 Point

Recall that `plum` has 31 rows, one corresponding to each of the 31 games Kelsey Plum's team played in the 2021 WNBA season.

Fill in the blank below so that `win_bool` evaluates to `True`.

```
def modify_series(s):
    return __(a)__

n_wins = plum.get('Won').sum()
win_bool = n_wins == (31 + modify_series(plum.get('Won')))
```

What goes in blank (a)?

- `-s.sum()`
- `-(s == False).sum()`
- `len(s) - s.sum()`
- `not s.sum()`
- `-s[s.get('Won') == False].sum()`

Q2 Head-to-Head 👊

11 Points

Let's suppose there are 4 different types of shots a basketball player can take – layups, midrange shots, threes, and free throws.

The DataFrame `breakdown` has 4 rows and 50 columns – one row for each of the 4 shot types mentioned above, and one column for each of 50 different players. Each column of `breakdown` describes the distribution of shot types for a single player.

The first few columns of `breakdown` are shown below.

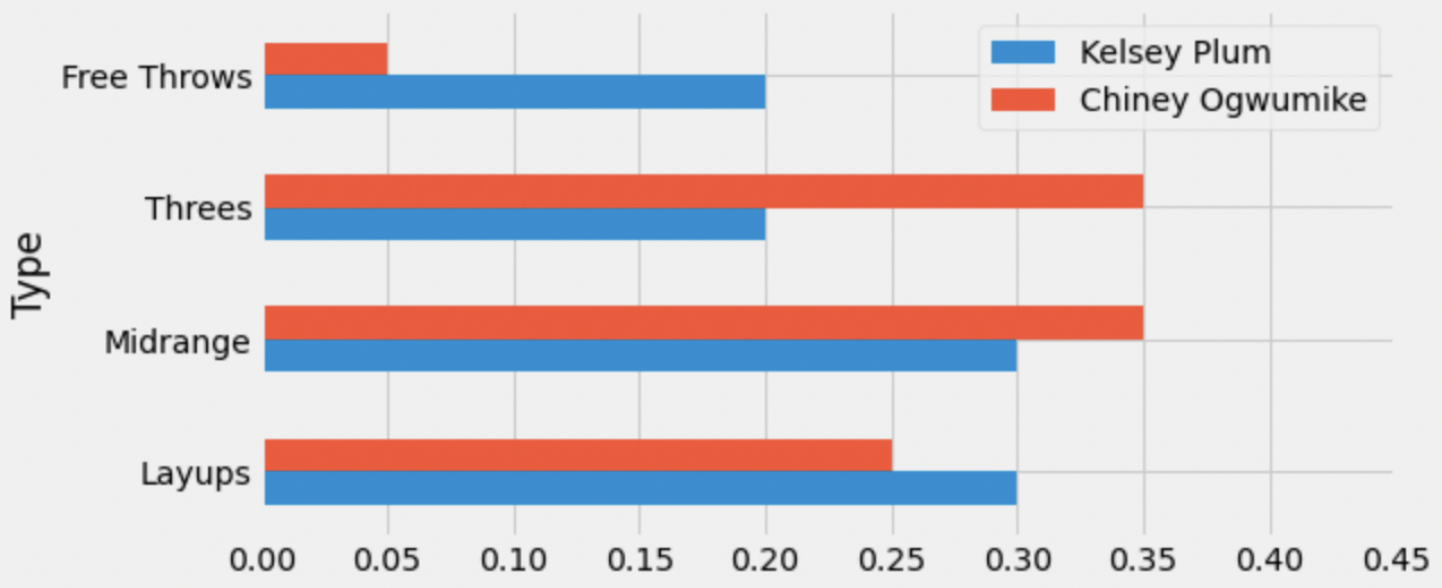
Type	Kelsey Plum	Sabrina Ionescu
Layups	0.3	0.50
Midrange	0.3	0.40
Threes	0.2	0.05
Free Throws	0.2	0.05

For instance, 30% of Kelsey Plum's shots are layups, 30% of her shots are midrange shots, 20% of her shots are threes, and 20% of her shots are free throws.

Q2.1

1.5 Points

Below, we've drawn an overlaid bar chart showing the shot distributions of Kelsey Plum and Chiney Ogwumike, a player on the Los Angeles Sparks.



What is the **total variation distance** (TVD) between Kelsey Plum's shot distribution and Chiney Ogwumike's shot distribution? Give your answer as a **proportion** between 0 and 1 (not a percentage) rounded to three decimal places.

Q2.2

4 Points

Recall, `breakdown` has information for 50 different players. We want to find the player whose shot distribution is the **most similar to Kelsey Plum**, i.e. has the lowest TVD with Kelsey Plum's shot distribution.

Fill in the blanks below so that `most_sim_player` evaluates to the name of the player with the most similar shot distribution to Kelsey Plum. Assume that the column named `'Kelsey Plum'` is the first column in `breakdown` (and again that `breakdown` has 50 columns total).

```
most_sim_player = ''
lowest_tvd_so_far = __(a)__
other_players = np.array(breakdown.columns).take(__(b)__)
for player in other_players:
    player_tvd = tvd(breakdown.get('Kelsey Plum'),
                     breakdown.get(player))
    if player_tvd < lowest_tvd_so_far:
        lowest_tvd_so_far = player_tvd
        __(c)__
```

What goes in blank (a)?

- -1
- -0.5
- 0
- 0.5
- 1
- `np.array([])`
- `''`

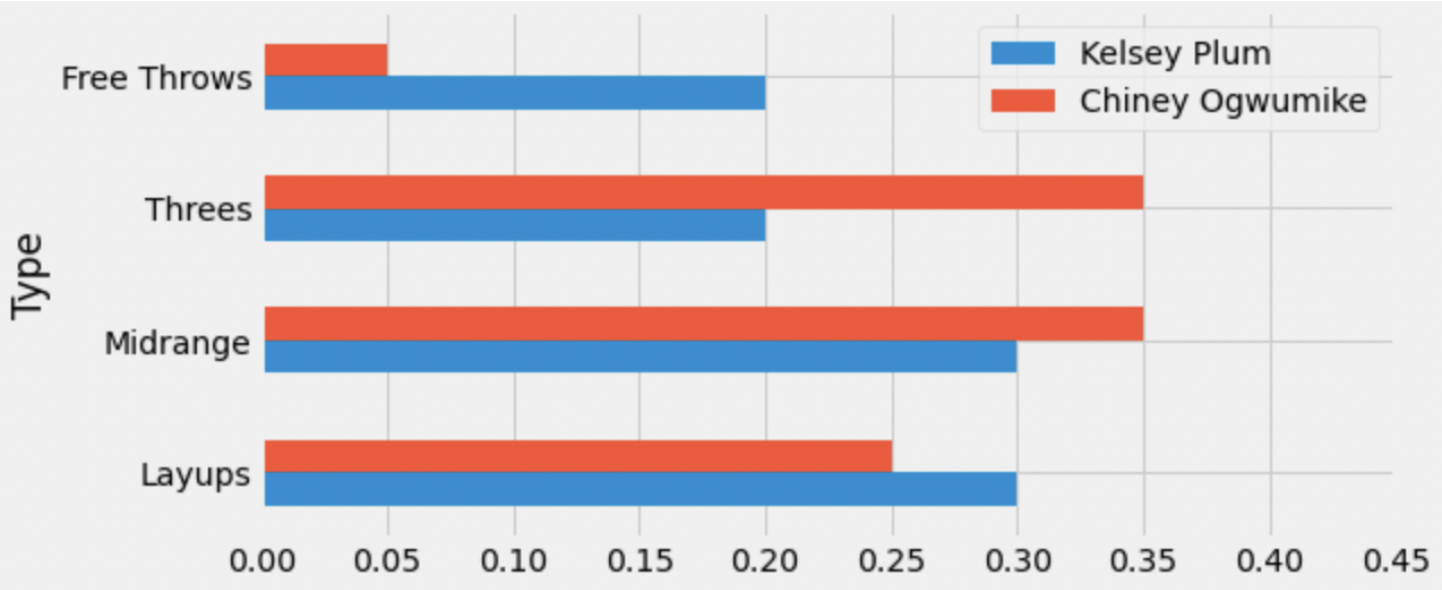
What goes in blank (b)?

What goes in blank (c)?

Q2.3

1.5 Points

Let's again consider the shot distributions of Kelsey Plum and Cheney Ogwumike.



We define the **maximum squared distance** (MSD) between two categorical distributions as the **largest squared difference between the proportions of any category**.

What is the MSD between Kelsey Plum's shot distribution and Chiney Ogwumike's shot distribution? Give your answer as a **proportion** between 0 and 1 (not a percentage) rounded to three decimal places.

Q2.4

3 Points

For your convenience, we show the first few columns of `breakdown` again below.

Type	Kelsey Plum	Sabrina Ionescu
Layups	0.3	0.50
Midrange	0.3	0.40
Threes	0.2	0.05
Free Throws	0.2	0.05

In basketball:

- layups are worth 2 points,
- midrange shots are worth 2 points,
- threes are worth 3 points, and
- free throws are worth 1 point

Suppose that Kelsey Plum is guaranteed to shoot exactly 10 shots a game. The type of each shot is drawn from the `'Kelsey Plum'` column of `breakdown` (meaning that, for example, there is a 30% chance each shot is a layup).

Fill in the blanks below to complete the definition of the function `simulate_points`, which simulates the number of points Kelsey Plum scores in a single game. (`simulate_points` should return a single number.)

```
def simulate_points():
    shots = np.random.multinomial(__(a)__, breakdown.get('Kelsey Plum'))
    possible_points = np.array([2, 2, 3, 1])
    return __(b)__
```

What goes in blank (a)?

What goes in blank (b)?

Q2.5

1 Point

True or False: If we call `simulate_points()` 10,000 times and plot a histogram of the results, the distribution will look roughly normal.

- True
- False

Q3 Vegas 🎰

9 Points

ESPN (a large sports news network) states that the Las Vegas Aces have a 60% chance of winning their upcoming game. You're curious as to how they came up with this estimate, and you decide to conduct a hypothesis test for the following hypotheses:

- **Null Hypothesis:** The Las Vegas Aces win each game with a probability of 60%.
- **Alternative Hypothesis:** The Las Vegas Aces win each game with a probability **above** 60%.

In both hypotheses, we are assuming that each game is independent of all other games.

In the 2021 season, the Las Vegas Aces **won 22** of their games and **lost 9** of their games.

Q3.1

3 Points

Below, we have provided the code necessary to conduct the hypothesis test described above.

```
stats = np.array([])
for i in np.arange(10000):
    sim = np.random.multinomial(31, [0.6, 0.4])
    stat = fn(sim)
    stats = np.append(stats, stat)

win_p_value = np.count_nonzero(stats >= fn([22, 9])) / 10000
```

`fn` is a **function** that computes a test statistic, given a list or array `arr` of two elements (the first of which is the number of wins, and the second of which is the number of losses). You can assume that neither element of `arr` is equal to 0.

Below, we define 5 possible test statistics `fn`.

Option 1:

```
def fn(arr):
    return arr[0] / arr[1]
```

Option 2:

```
def fn(arr):
    return arr[0]
```

Option 3:

```
def fn(arr):
    return np.abs(arr[0] - arr[1])
```

Option 4:

```
def fn(arr):
    return arr[0] - arr[1]
```

Option 5:

```
def fn(arr):
    return arr[1] - arr[0]
```

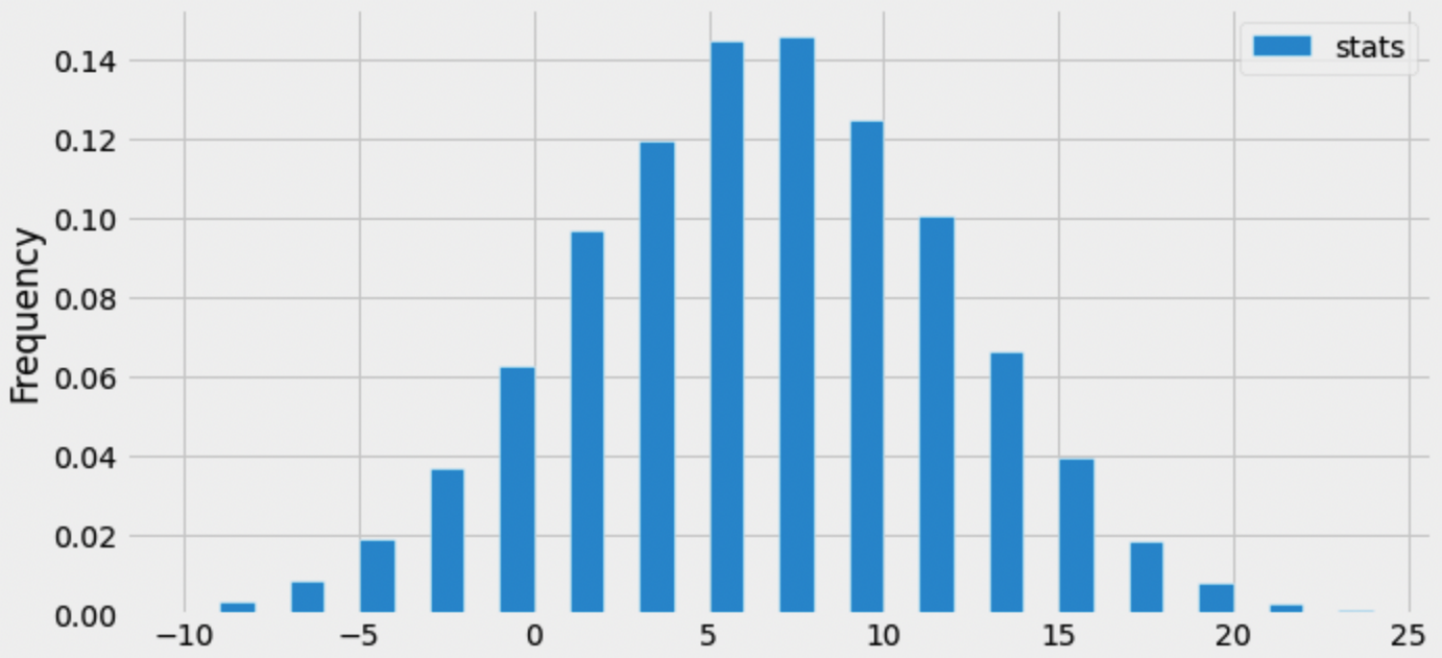
Which of the above functions `fn` would be valid test statistics for this hypothesis test and p-value calculation? **Select all that apply.**

- [ ] Option 1
- [ ] Option 2
- [ ] Option 3
- [ ] Option 4
- [ ] Option 5

Q3.2

1 Point

The empirical distribution of one of the 5 test statistics presented in Question 3.1 is shown below. To draw the histogram, we used the argument `bins=np.arange(-10, 25)`.



Which test statistic does the above empirical distribution belong to?

- Option 1
- Option 2
- Option 3
- Option 4
- Option 5

Q3.3

1 Point

Consider the function `fn_plus` defined below.

```
def fn_plus(arr):
    return fn(arr) + 31
```

True or False: If `fn` is a valid test statistic for the hypothesis test and p-value calculation in Question 3.1, then `fn_plus` is also a valid test statistic for the hypothesis test and p-value calculation in Question 3.1.

- ○ True
- ○ False

Q3.4

2 Points

For your convenience, we show the first few rows of `plum` again below. Recall that `plum` contains information for all 31 games that Kelsey Plum and her team (the Las Vegas Aces) played in 2021.

	Date	Opp	Home	Won	PTS	AST	TOV
0	2021-05-15	SEA	False	False	11	4	1
1	2021-05-18	SEA	False	True	10	3	0
2	2021-06-03	NYL	False	True	4	2	2
3	2021-06-05	WAS	False	True	2	2	1
4	2021-06-13	DAL	True	True	13	2	2

Below, we present the same code that is given at the start of Question 3.1.

```
stats = np.array([])
for i in np.arange(10000):
    sim = np.random.multinomial(31, [0.6, 0.4])
    stat = fn(sim)
    stats = np.append(stats, stat)

win_p_value = np.count_nonzero(stats >= fn([22, 9])) / 10000
```

Below are four possible replacements for the line `sim = np.random.multinomial(31, [0.6, 0.4])`.

Option 1:

```
def with_rep():
    won = plum.get('Won')
    return np.count_nonzero(np.random.choice(won, 31, replace=True))

sim = [with_rep(), 31 - with_rep()]
```

Option 2:

```
def with_rep():
    won = plum.get('Won')
    return np.count_nonzero(np.random.choice(won, 31, replace=True))

w = with_rep()
sim = [w, 31 - w]
```

Option 3:

```
def without_rep():
    won = plum.get('Won')
    return np.count_nonzero(np.random.choice(won, 31, replace=False))

sim = [without_rep(), 31 - without_rep()]
```

Option 4:

```
def perm():
    won = plum.get('Won')
    return np.count_nonzero(np.random.permutation(won))

w = perm()
sim = [w, 31 - w]
```

Which of the above four options could we replace the line `sim = np.random.multinomial(plum.shape[0], [0.6, 0.4])` with and still perform a valid hypothesis test for the hypotheses stated in Question 3.1?

- Option 1
- Option 2
- Option 3
- Option 4

Q3.5

2 Points

Consider again the 4 options presented in Question 3.4.

In which of the four options is it **guaranteed** that `sim[0] + sim[1]` evaluates to `31`? **Select all that apply.**

- [ ] Option 1
- [ ] Option 2
- [ ] Option 3
- [ ] Option 4

Q4 Pass Please!

5 Points

For your convenience, we show the first few rows of `plum` again below. If you'd like a refresher on what the columns all refer to, scroll to the top of the exam.

	Date	Opp	Home	Won	PTS	AST	TOV
0	2021-05-15	SEA	False	False	11	4	1
1	2021-05-18	SEA	False	True	10	3	0
2	2021-06-03	NYL	False	True	4	2	2
3	2021-06-05	WAS	False	True	2	2	1
4	2021-06-13	DAL	True	True	13	2	2

Consider the definition of the function `diff_in_group_means`:

```
def diff_in_group_means(df, group_col, num_col):
    s = df.groupby(group_col).mean().get(num_col)
    return s.loc[False] - s.loc[True]
```

Q4.1

2 Points

It turns out that Kelsey Plum averages 0.61 more assists in games that she wins ("winning games") than in games that she loses ("losing games"). Fill in the blanks below so that `observed_diff` evaluates to `-0.61`.

```
observed_diff = diff_in_group_means(plum, __(a)__, __(b)__)
```

What goes in blank (a)?

What goes in blank (b)?

Q4.2

1 Point

After observing that Kelsey Plum averages more assists in winning games than in losing games, we become interested in conducting a permutation test for the following hypotheses:

- **Null Hypothesis:** The number of assists Kelsey Plum makes in winning games and in losing games come from the same distribution.
- **Alternative Hypothesis:** The number of assists Kelsey Plum makes in winning games is higher on average than the number of assists that she makes in losing games.

To conduct our permutation test, we place the following code in a `for`-loop.

```
won = plum.get('Won')
ast = plum.get('AST')
shuffled = plum.assign(Won_shuffled=np.random.permutation(won)) \
               .assign(AST_shuffled=np.random.permutation(ast))
```

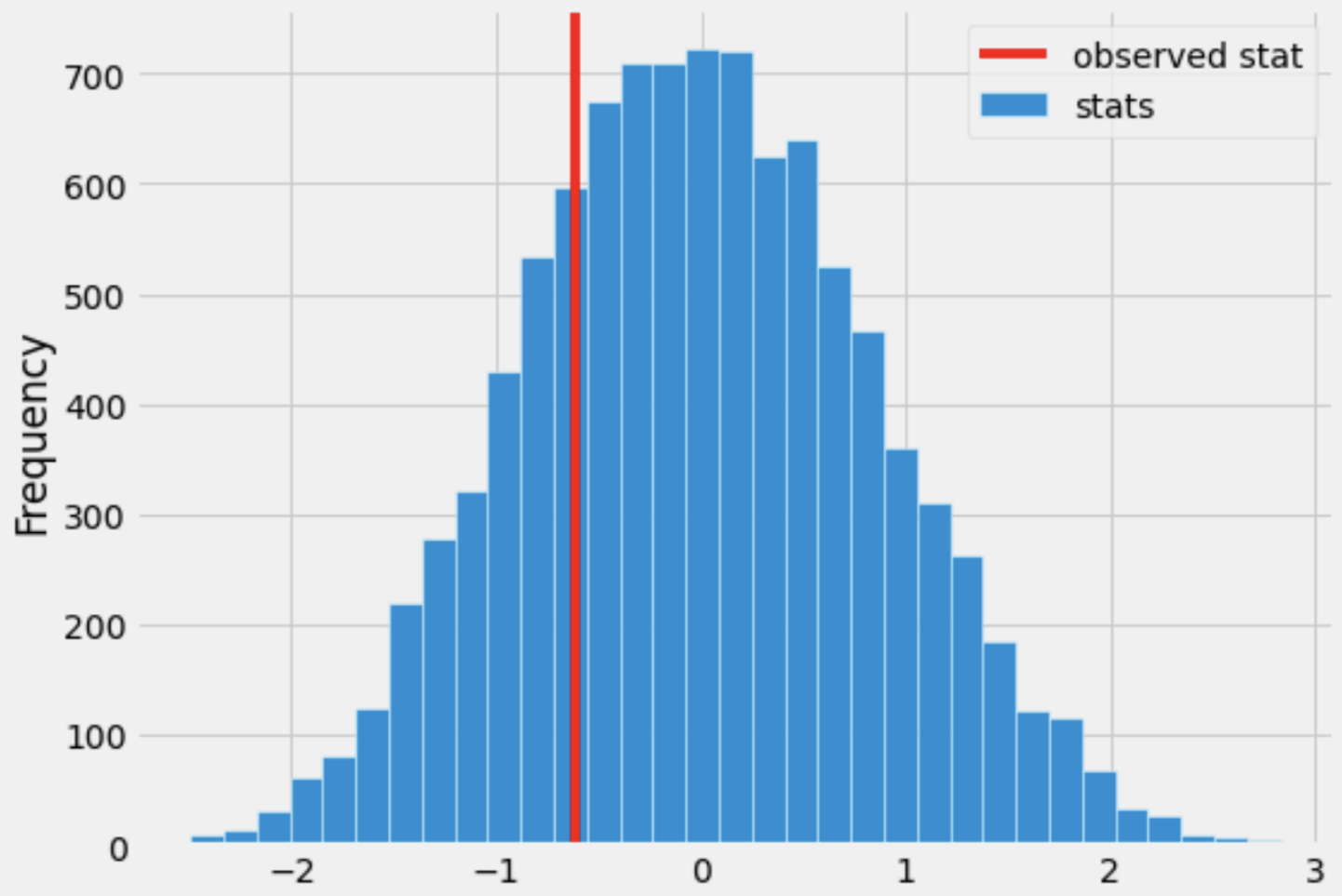
Which of the following options **does not** compute a valid simulated test statistic for this permutation test?

- `diff_in_group_means(shuffled, 'Won', 'AST')`
- `diff_in_group_means(shuffled, 'Won', 'AST_shuffled')`
- `diff_in_group_means(shuffled, 'Won_shuffled, 'AST')`
- `diff_in_group_means(shuffled, 'Won_shuffled, 'AST_shuffled')`
- More than one of these options do not compute a valid simulated test statistic for this permutation test

Q4.3

1 Point

Suppose we generate 10,000 simulated test statistics, using one of the valid options from Question 4.2. The empirical distribution of test statistics, with a red line at `observed_diff`, is shown below.



Roughly one-quarter of the area of the histogram above is to the left of the red line. What is the correct interpretation of this result?

- There is roughly a one quarter probability that Kelsey Plum's number of assists in winning games and in losing games come from the same distribution.
- The significance level of this hypothesis test is roughly a quarter.
- Under the assumption that Kelsey Plum's number of assists in winning games and in losing games come from the same distribution, and that she wins 22 of the 31 games she plays, the chance of her averaging **at least** 0.61 more assists in wins than losses is roughly a quarter.
- Under the assumption that Kelsey Plum's number of assists in winning games and in losing games come from the same distribution, and that she wins 22 of the 31 games she plays, the chance of her averaging 0.61 more assists in wins than losses is roughly a quarter.

Q4.4

1 Point

True or False: The histogram drawn in Question 4.3 is a density histogram.

- ○ True
- ○ False

Q5 Free Samples! 🌮

6 Points

Recall, `plum` has 31 rows.

Consider the function `df_choice`, defined below.

```
def df_choice(df):
    return df[np.random.choice([True, False], df.shape[0], replace=True)]
```

Q5.1

1.5 Points

Suppose we call `df_choice(plum)` once. What is the probability that the result is an empty DataFrame?

- 0
- 1
- $\frac{1}{2^{25}}$
- $\frac{1}{2^{30}}$
- $\frac{1}{2^{31}}$
- $\frac{2^{31}-1}{2^{31}}$
- $\frac{31}{2^{30}}$
- $\frac{31}{2^{31}}$
- None of the above

Q5.2

1.5 Points

Suppose we call `df_choice(plum)` once. What is the probability that the result is a DataFrame with 30 rows?

- 0
- 1
- $\frac{1}{2^{25}}$
- $\frac{1}{2^{30}}$
- $\frac{1}{2^{31}}$
- $\frac{2^{31}-1}{2^{31}}$
- $\frac{31}{2^{30}}$
- $\frac{31}{2^{31}}$
- None of the above

Q5.3

1.5 Points

Suppose we call `df_choice(plum)` once.

True or False: The probability that the result is a DataFrame that consists of **just** row 0 from `plum` (and no other rows) is equal to the probability you computed in Question 5.1.

- True
- False

Q5.4

1.5 Points

Suppose we call `df_choice(plum)` once.

What is the probability that the resulting DataFrame has 0 rows, or 1 row, or 30 rows, or 31 rows?

- 0
- 1
- $\frac{1}{2^{25}}$
- $\frac{1}{2^{30}}$
- $\frac{1}{2^{31}}$
- $\frac{2^{31}-1}{2^{31}}$
- $\frac{31}{2^{30}}$
- $\frac{31}{2^{31}}$
- None of the above

Q6 Turnovers 🥐

5 Points

In addition to the `plum` DataFrame, we also have access to the `season` DataFrame, which contains statistics on all players in the WNBA in the 2021 season. The first few rows of `season` are shown below.

	Player	Team	G	PPG	APG	TPG
0	Natalie Achonwa	MIN	21	3.67	1.19	0.62
1	Bella Alarie	DAL	31	2.61	0.52	0.58
2	Lindsay Allen	IND	32	5.44	3.00	0.97
3	Rebecca Allen	NYL	25	9.16	1.12	0.84
4	Jillian Alleyne	WAS	2	0.00	0.00	0.00

Each row in `season` corresponds to a single player. For each player, we have:

- `'Player'` (`str`), their name
- `'Team'` (`str`), the three-letter code of the team they play on
- `'G'` (`int`), the number of games they played in the 2021 season
- `'PPG'` (`float`), the number of points they scored per game played
- `'APG'` (`float`), the number of assists (passes) they made per game played
- `'TPG'` (`float`), the number of turnovers they made per game played

Note that all of the numerical columns in `season` must contain values that are greater than or equal to 0.

Q6.1

1 Point

Which of the following is the best choice for the index of `season`?

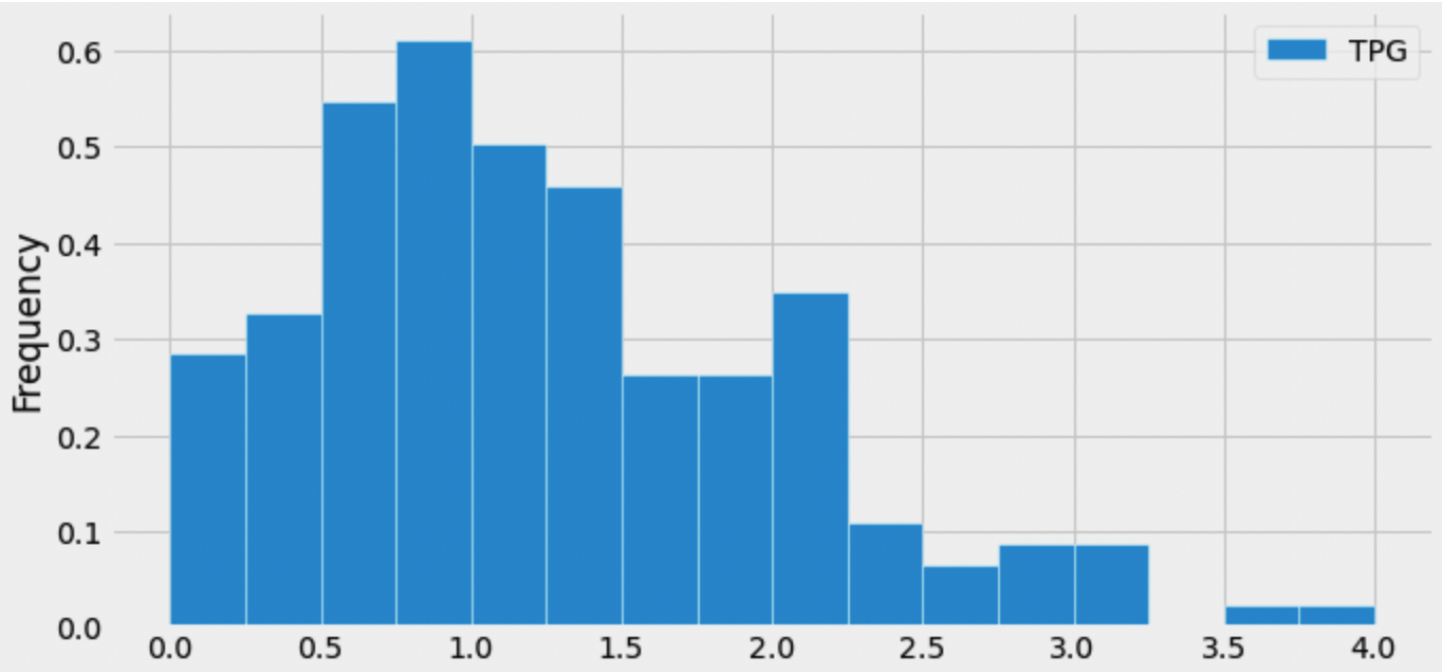
- `'Player'`
- `'Team'`
- `'G'`
- `'PPG'`

Q6.2

1 Point

Note: For the rest of the exam, assume that the index of `season` is still 0, 1, 2, 3, ...

Below is a histogram showing the distribution of the number of turnovers per game for all players in `season`.



Suppose, **throughout this question**, that the mean number of turnovers per game is 1.25. Which of the following is closest to the median number of turnovers per game?

- 0.5
- 0.75
- 1
- 1.25
- 1.5
- 1.75

Q6.3

2 Points

Sabrina Ionescu and Sami Whitcomb are both players on the New York Liberty, and are both California natives.

In "original units", Sabrina Ionescu had 3.5 turnovers per game. In standard units, her turnovers per game is 3.

In standard units, Sami Whitcomb's turnovers per game is -1. How many turnovers per game did Sami Whitcomb have in **original units**? Round your answer to 3 decimal places.

Note: You will need the fact from Question 6.2 that the mean number of turnovers per game is 1.25.

Q6.4

1 Point

What is the **smallest** possible number of turnovers per game, in **standard units**? Round your answer to 3 decimal places.

Q7 Good Point 👆

8 Points

Let's switch our attention to the relationship between the number of points per game and the number of assists per game for all players in `season`. Using `season`, we compute the following information:

- The mean points per game is 7, with a standard deviation of 5
- The mean number of assists per game is 1.5, with a standard deviation of 1.5
- The correlation between points per game and assists per game is 0.65

Q7.1

3 Points

Let's start by using points per game (x) to predict assists per game (y).

Tina Charles had 27 points per game in 2021, the most of any player in the WNBA. What is her predicted assists per game, according to the regression line? Round your answer to 3 decimal places.

Q7.2

1 Point

Tina Charles actually had 2.1 assists per game in the 2021 season.

What is the error, or residual, for the prediction in Question 7.1? Round your answer to 3 decimal places.

Q7.3

2 Points

Select all true statements below regarding the regression line between points per game (x) and assists per game (y).

- [ ] The point (0, 0) is guaranteed to be on the regression line when both x and y are in standard units.
- [ ] The point (0, 0) is guaranteed to be on the regression line when both x and y are in original units.
- [ ] The point (7, 1.5) is guaranteed to be on the regression line when both x and y are in standard units.
- [ ] The point (7, 1.5) is guaranteed to be on the regression line when both x and y are in original units.
- [ ] None of the above

Q7.4

2 Points

So far, we've been using points per game (x) to predict assists per game (y). Suppose we found the regression line (when both x and y are in original units) to be $y = ax + b$.

Now, let's reverse x and y. That is, we will now use assists per game (x) to predict points per game (y). The resulting regression line (when both x and y are in original units) is $y = cx + d$.

Which of the following statements is guaranteed to be true?

- ○ $a = c$
- ○ $a > c$
- ○ $a < c$
- ○ Using just the information given in Question 7, it is impossible to determine the relationship between a and c.

Q8 Perfectile 💯

6 Points

Q8.1

3 Points

Recall from Question 7 that the mean points per game is 7, with a standard deviation of 5. Also note that for all players, points per game must be greater than or equal to 0.

Using Chebyshev's inequality, we find that at least $p\%$ of players scored 25 or fewer points per game.

What is the value of p? Give your answer as number between 0 and 100, rounded to 3 decimal places.

Q8.2

3 Points

Note: This question uses the mathematical definition of percentile, not `np.percentile`.

The array `aces` defined below contains the points per game scored by all members of the Las Vegas Aces. Note that it contains 14 numbers that are in sorted order.

```
aces = np.array([0, 0, 1.05, 1.47, 1.96, 2, 3.25,
                 10.53, 11.09, 11.62, 12.19,
                 14.24, 14.81, 18.25])
```

As we saw in lab, percentiles are not unique. For instance, the number 1.05 is both the 15th percentile and 16th percentile of `aces`.

There is a positive integer q, between 0 and 100, such that 14.24 is the qth percentile of `aces`, but 14.81 is the $(q+1)$th percentile of `aces`.

What is the value of q? Give your answer as an integer between 0 and 100.

Q9 Sneak Peek 👀

11 Points

For your convenience, we show the first few rows of `season` again below.

	Player	Team	G	PPG	APG	TPG
0	Natalie Achonwa	MIN	21	3.67	1.19	0.62
1	Bella Alarie	DAL	31	2.61	0.52	0.58
2	Lindsay Allen	IND	32	5.44	3.00	0.97
3	Rebecca Allen	NYL	25	9.16	1.12	0.84
4	Jillian Alleyne	WAS	2	0.00	0.00	0.00

In Questions 6-8, we presumed that we had access to the entire `season` DataFrame. Now, suppose we only have access to the DataFrame `small_season`, which is a random sample of **size 36** from `season`. We're interested in learning about the true mean points per game of all players in `season` given just the information in `small_season`.

To start, we want to bootstrap `small_season` 10,000 times and compute the mean of the resample each time. We want to store these 10,000 bootstrapped means in the array `boot_means`.

Here is a broken implementation of this procedure.

```
boot_means = np.array([])
for i in np.arange(10000):
    resample = small_season.sample(season.shape[0], replace=False)  # Line 1
    resample_mean = small_season.get('PPG').mean()                  # Line 2
    np.append(boot_means, new_mean)                                 # Line 3
```

For each of the 3 lines of code above (marked by comments), specify what is incorrect about the line by selecting one or more of the corresponding options below. Or, select "Line _ is correct as-is" if you believe there's nothing that needs to be changed about the line in order for the above code to run properly.

Q9.1
1.5 Points

What is incorrect about Line 1? Select all that apply.

- [ ] Currently the procedure samples from `small_season`, when it should be sampling from `season`
- [ ] The sample size is `season.shape[0]`, when it should be `small_season.shape[0]`
- [ ] Sampling is currently being done without replacement, when it should be done with replacement
- [ ] Line 1 is correct as-is

Q9.2
1.5 Points

What is incorrect about Line 2? Select all that apply.

- [ ] Currently it is taking the mean of the `'PPG'` column in `small_season`, when it should be taking the mean of the `'PPG'` column in `season`
- [ ] Currently it is taking the mean of the `'PPG'` column in `small_season`, when it should be taking the mean of the `'PPG'` column in `resample`
- [ ] `.mean()` is not a valid Series method, and should be replaced with a call to the function `np.mean`
- [ ] Line 2 is correct as-is

Q9.3

1.5 Points

What is incorrect about Line 3? Select all that apply.

- [ ] The result of calling `np.append` is not being reassigned to `boot_means`, so `boot_means` will be an empty array after running this procedure
- [ ] The indentation level of the line is incorrect – `np.append` should be outside of the `for`-loop (and aligned with `for i`)
- [ ] `new_mean` is not a defined variable name, and should be replaced with `resample_mean`
- [ ] Line 3 is correct as-is

Q9.4

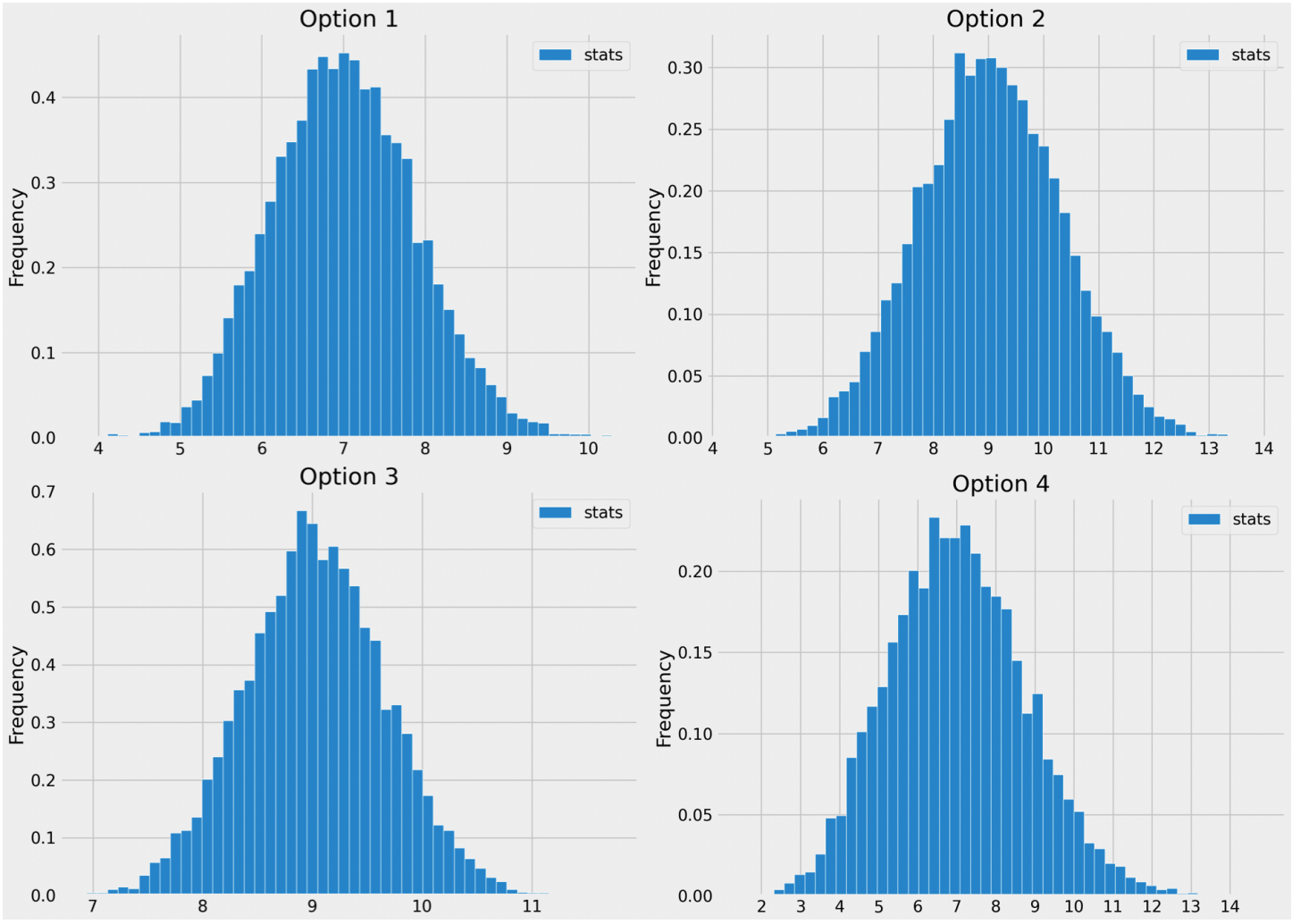
1.5 Points

Suppose we've now fixed everything that was incorrect about our bootstrapping implementation.

Recall from Question 7 that, in `season`, the mean number of points per game is 7, with a standard deviation of 5.

It turns out that when looking at just the players in `small_season`, the mean number of points per game is 9, with a standard deviation of 4. Remember that `small_season` is a random sample of size 36 taken from `season`.

Which of the following histograms visualizes the empirical distribution of the sample mean, computed using the bootstrapping procedure above?



- Option 1
- Option 2
- Option 3
- Option 4

Q9.5

1 Point

We construct a 95% confidence interval for the true mean points per game for all players by taking the middle 95% of the bootstrapped sample means.

```
left_b = np.percentile(boot_means, 2.5)
right_b = np.percentile(boot_means, 97.5)
boot_ci = [left_b, right_b]
```

Select the most correct statement below.

- `(left_b + right_b) / 2` is exactly equal to the mean points per game in `season`.
- `(left_b + right_b) / 2` is not necessarily equal to the mean points per game in `season`, but is close.
- `(left_b + right_b) / 2` is exactly equal to the mean points per game in `small_season`.
- `(left_b + right_b) / 2` is not necessarily equal to the mean points per game in `small_season`, but is close.
- `(left_b _+ right_b) / 2` is not close to either the mean points per game in `season` or the mean points per game in `small_season`.

Q9.6

3 Points

Instead of bootstrapping, we could also construct a 95% confidence interval for the true mean points per game by using the Central Limit Theorem.

Recall that, when looking at just the players in `small_season`, the mean number of points per game is 9, with a standard deviation of 4. Also remember that `small_season` is a random sample of size 36 taken from `season`.

Using only the information that we have about `small_season` (i.e. without using any facts about `season`), compute a 95% confidence interval for the true mean points per game.

What is the left endpoint of your interval? Give your answer as a number rounded to 3 decimal places.

What is the right endpoint of your interval? Give your answer as a number rounded to 3 decimal places.

Q9.7
1 Point

Recall that the mean points per game in `season` is 7, which is not in the interval you found above (if it is, check your work!).

Select the true statement below.

- The 95% confidence interval we created in Question 9.6 did not contain the true mean points per game, which means that the distribution of the sample mean is not normal.
- The 95% confidence interval we created in Question 9.6 did not contain the true mean points per game, which means that the distribution of points per game in `small_season` is not normal.
- The 95% confidence interval we created in Question 9.6 did not contain the true mean points per game. This is to be expected, because the Central Limit Theorem is only correct 95% of the time.
- The 95% confidence interval we created in Question 9.6 did not contain the true mean points per game, but if we collected many original samples and constructed many 95% confidence intervals, then roughly 95% of them would contain the true mean points per game.
- The 95% confidence interval we created in Question 9.6 did not contain the true mean points per game, but if we collected many original samples and constructed many 95% confidence intervals, then exactly 95% of them would contain the true mean points per game.

Q10 Follow for Follow 📲
2 Points

The WNBA is interested in helping boost their players' social media presence, and considers various ways of making that happen.

Which of the following claims can be tested using a randomized controlled trial? Select all that apply.

- [ ] Winning two games in a row causes a player to gain Instagram followers.
- [ ] Drinking Gatorade causes a player to gain Instagram followers.
- [ ] Playing for the Las Vegas Aces causes a player to gain Instagram followers.
- [ ] Deleting Twitter causes a player to gain Instagram followers.
- [ ] None of the above