Question 1

In the DataFrame `songs`, each row is a song. Columns are `"title"`, `"artist"`, `"genre"`, and `"streams"` as an integer (number of times the song has been streamed).

a) To find the song in each genre with the most streams, Elyse writes this code:

```
songs.groupby("genre").max()["title"]
```

In **one sentence**, explain the problem with Elyse's code.

> **Solution:** This will give the song title that is latest alphabetical in each genre, since groupby calculates the max of each column individually.

b) Now, fix Elyse's code. Write **one line of code** that evaluates to a Series indexed by `"genre"` containing the title of the song in that genre with the most streams.

> **Solution:**
> There may be additional possible answers, but two good answers are:
>
> ```
> songs.sort_values("streams", ascending=False)
>      .groupby("genre")["title"].first()
> songs.set_index("title").groupby("genre")["streams"].idxmax())
> ```

c) Assuming song titles are unique, write **one line of code** that evaluates to a Series indexed by `"title"` such that:

- songs with fewer than 1,000,000 streams are omitted, and
- for each remaining song, the Series contains the mean streams of all songs in the same genre with at least 1,000,000 streams.

> **Solution:**
> ```
> songs[songs["streams"]>=1000000].set_index("title")
> .groupby("genre")["streams"].transform(lambda s:  s.mean())
> ```

Question 2

You are given two DataFrames, `A` and `B`, shown below.

`A`

sid	name	age
1	Alice	20
2	Bob	NaN
2	Bob	21
3	NaN	22
5	Eve	23

`B`

sid	course
1	Math
1	Bio
2	NaN
2	Chem
4	CS

We merge these DataFrames as follows:

```
merged = A.merge(B, on="sid", how="outer")
```

a) How many rows does `merged` have?

Solution: 9 rows

sid	name	age	course
1	Alice	20	Math
1	Alice	20	Bio
2	Bob	NaN	NaN
2	Bob	NaN	Chem
2	Bob	21	NaN
2	Bob	21	Chem
3	NaN	22	NaN
5	Eve	23	NaN
4	NaN	NaN	CS

b) How many missing values (`NaN`) does `merged` have in total, across all columns?

Solution: 9 missing values

c) If we replaced `how="outer"` with `how="inner"`, how many **fewer** rows would the result have?

Solution: 3 rows (for `"sid"`s 3, 4, 5 which appear in only one DataFrame)

d) In this example, replacing `how="outer"` with `how="inner"` reduces the number of rows in the result. Is that always the case in general? Justify your answer in **one sentence**.

Solution: It is always the case that the number of rows decreases or stays the same since inner join includes rows that appear in both DataFrames, and outer join includes all of those rows plus possibly more (rows appearing in one DataFrame but not the other).

Question 3

The DataFrame `sessions` has one row per tutoring session, with columns `"tutor"` (`"Annabelle"` or `"Ishayu"`), `"subject"` (`"Math"` or `"English"`), `"length"` (`"Short"` or `"Long"`), and `"score_gain"` (float).

a) Fill in the blank so that `p1` is a pivot table showing the average `"score_gain"`, with one row per `"tutor"` and one column per `"subject"`.

```
p1 = sessions.pivot_table(_________________)
```

> **Solution:** `index="tutor", columns="subject", values="score_gain", aggfunc="mean"`

b) Suppose `p1` evaluates to the table below. Is it possible to observe Simpson's paradox if we don't split by `"tutor"` — i.e., could the overall average `"score_gain"` be higher for `"Math"` than for `"English"`? Choose between "yes", "no", "need more information" and **justify your answer**.

	Math	**English**
Annabelle	5.0	9.0
Ishayu	8.0	8.5

> **Solution: No.** These specific numbers cannot work out to an instance of Simpson's Paradox. The overall average range for Math is a weighted average of 5.0 and 8.0, so whatever the weights happen to be, the overall average must be between 5.0 and 8.0. Therefore this overall average cannot physically exceed the overall average for English, which must be between 8.5 and 9.0 for the same reason.

c) Now suppose `p2` is a pivot table showing the average `"score_gain"` by `"length"` and `"subject"`, shown below. Is it possible to observe Simpson's paradox if we don't split by `"length"`? Choose between "yes", "no", "need more information" and **justify your answer**.

	Math	**English**
Short	9.5	6.0
Long	5.0	9.0

> **Solution: No.** Since the Math value is higher in the Short category, and the English value is higher in the Long category, there is no consistent trend across groups that could be reversed when groups are combined. Simpson's Paradox requires a consistent trend across groups that is reversed when groups are combined.

Question 4

The DataFrame `calls` has one row per customer service call, with columns `"day"` (day of week), `"hour"` (integer 0–23), and `"wait_time"` (float, minutes; some values are missing).

For each pair of hypotheses given below,

- Determine whether this is a standard hypothesis test or a permutation test.
- Choose **one** correct simulation procedure among the given options.
- Choose **all** valid test statistics that could be used. **Justify** your choices for the test statistics only.

Simulation procedures:

1. `np.random.multinomial(n, [1/2]*2)`
2. `np.random.multinomial(n, [1/7]*7)`
3. `calls["hour"].sample(n)`
4. `np.random.permutation(calls["hour"])`

Test statistics:

1. Difference in means
2. Absolute difference in means
3. TVD
4. K-S statistic

a) Null: Calls are equally likely to occur on each day of the week.
Alt: Some days are more likely than others.

> **Solution:** Test: Standard hypothesis test.
> Procedure: 2.
> Statistic(s): 3 because day of the week is categorical so we are comparing two categorical distributions, which is the purpose of TVD.

b) Null: The average `"wait_time"` is the same in the morning (`hour` < 12) as in the afternoon.
Alt: The average `"wait_time"` is different.

> **Solution:** Test: Permutation test.
> Procedure: 4.
> Statistic(s): 2 because the statement of the hypotheses is about whether the averages are the same or different (no direction, so we use absolute value). Note that K-S statistic does not work here because the hypotheses are not about whether two distributions are the same, they are specifically about whether two averages are the same.

c) Null: The distribution of `"hour"` is the same for calls with missing `"wait_time"` and calls with non-missing `"wait_time"`.
Alt: The distributions are different.

Solution: Test: Permutation test.

Procedure: 4.

Statistic(s): 3 or 4 depending on if you see `"hour"` as numerical or categorical (could be used either way). 2 may also work if the distributions are noticeably different but without any visualizations, it's impossible to know if this is the case, so K-S statistic is a better choice. 1 does not work because the alternative does not have a direction to it.

Question 5

At a biking event, there are "pit stops" set up at various major intersections throughout the city. Each pit stop is run by a volunteer "site captain" who is on site to record attendance at the pit stop. Attendance, pit stop addresses, and corresponding neighborhoods are recorded in a DataFrame. Some attendance values are missing.

Each part below describes a possible explanation for this missingness. Assuming that the given explanation is **the only force at play**:

- Determine the most likely missingness mechanism (MD, MNAR, MAR, MCAR) and **justify** your answer.
- Determine whether mean imputation would give an overestimate, an underestimate, or an unbiased estimate of the true mean. No justification necessary.

a) Site captains were provided with physical counting devices that they could use to keep track of attendance. Some of the devices were defective.

> **Solution:** MCAR because we have no information about which devices will be defective, it's just completely random.
>
> Mean imputation gives an unbiased estimate for MCAR data because the data we do have is like a random sample of all the data we might have had.

b) Site captains were told about the requirement to keep track of attendance at neighborhood site captain meetings. The leader of the Mira Mesa neighborhood meeting forgot to inform the site captains of this requirement. Some Mira Mesa pit stops did record attendance without being told, but not all of them.

> **Solution:** MAR because there is a dependence on neighborhood and missingness is not completely determined by neighborhood (MD). It is not the case that every Mira Mesa attendance value is missing.
>
> The effect of mean imputation is unclear, as it depends on whether people in Mira Mesa are just as likely people elsewhere in San Diego to participate in a biking event. Y

c) Some sites were so busy that the site captains could not keep track of the number of bikers.

> **Solution:** MNAR because it depends on the missing attendance values themselves - we are missing the largest attendance numbers.
>
> Mean imputation would lead to an underestimate since we're missing the high values.

d) After being informed of the requirement to keep track of attendance, site captains were never reminded. Older site captains were more likely to forget.

Solution: MCAR because we don't have any way of predicting which attendance values will be missing based on the data we do have, which is just address and neighborhood. There is a dependence on age, but since we don't have any age information available, the missingness is MCAR, not MAR.

Mean imputation gives an unbiased estimate for MCAR data because the data we do have is like a random sample of all the data we might have had.