**1. Total Games Played**

* **Prompt Used:** "How many games did Syracuse Women’s Lacrosse play in 2025?"
* **LLM Answer:** 19
* **Python Logic:** games\_df.shape[0]

**Explanation:**This is a foundational query that tests the LLM’s ability to count rows in a dataset. The model returned the correct result by identifying the number of entries in the games dataset. Python verification confirmed that there are indeed 19 rows, representing 19 games. This task required no metric definition or complex filtering, which made it an easy win for the LLM.

**2. Win/Loss Record**

* **Prompt Used:** "What was their overall win/loss record?"
* **LLM Answer:** 10W – 9L
* **Python Logic:**

total\_wins = games\_df[games\_df["Result"] == "W"].shape[0]

total\_losses = games\_df[games\_df["Result"] == "L"].shape[0]

**Explanation:**The LLM filtered the "Result" column to count occurrences of 'W' and 'L'. This is a basic aggregation task involving categorical values. It was handled well, showing the model can manage straightforward classification counts.

**3. Top Goal Scorer**

* **Prompt Used:** "Who scored the most goals in the 2025 season?"
* **LLM Answer:** Emma Muchnick (34 goals)
* **Python Logic:**

top\_goal\_scorer = players\_df.sort\_values(by="Goals", ascending=False).iloc[0]

**Explanation:**The LLM accurately sorted the player stats by "Goals" and selected the top performer. This question involved a simple max operation, and the LLM handled it smoothly.

**4. Highest Points Per Game**

* **Prompt Used:** "Which player had the highest average points per game?"
* **LLM Answer:** Olivia Adamson (5.33 PPG)
* **Python Logic:**

players\_df["Points\_Per\_Game"] = players\_df["Points"] / players\_df["Games\_Played"]

**Explanation:**Initially, the LLM needed help understanding what "Points per Game" meant. After defining the formula (Points = Goals + Assists, then divided by games played), it gave the correct answer. Python validation confirmed the calculated column and result.

**5. Closest Game**

* **Prompt Used:** "What was the closest game?"
* **LLM Answer:** STANFORD on 3/07/25 (14–13)
* **Python Logic:**

games\_df["Goal\_Diff"] = abs(games\_df["SU\_Score"] - games\_df["Opponent\_Score"])

closest\_game = games\_df.sort\_values("Goal\_Diff").iloc[0]

**Explanation:**The model had to compute an absolute difference between Syracuse’s and the opponent’s score to find the game with the smallest margin. While this is not overly complex, LLMs don’t assume abs() unless told. Once the definition of closeness was clarified, the LLM performed well.

**6. Most Goals in a Game**

* **Prompt Used:** "Which team did Syracuse score the most against?"
* **LLM Answer:** UALBANY (21 goals)
* **Python Logic:**

games\_df.sort\_values("SU\_Score", ascending=False).iloc[0]

**Explanation:**This question was handled correctly on the first try. The model found the maximum in the "SU\_Score" column. This type of ranking query is intuitive for LLMs.

**7. Most Consistent Scorer**

* **Prompt Used:** "Who was the most consistent scorer?"
* **LLM Answer:** Olivia Adamson (3.33 goals/game)
* **Python Logic:**

players\_df["Goals\_Per\_Game"] = players\_df["Goals"] / players\_df["Games\_Played"]

**Explanation:**The term “consistent” is subjective. Initially, LLM struggled. Once we clarified that “consistent” meant highest goals per game (GPG), it recalculated and gave the correct answer. Olivia Adamson scored 10 goals in just 3 games.

**8. Top Playmaker**

* **Prompt Used:** "Who was the top playmaker?"
* **LLM Answer:** Emma Ward (46 assists)
* **Python Logic:**

players\_df.sort\_values("Assists", ascending=False).iloc[0]

**Explanation:**“Playmaker” was interpreted correctly as the player with the highest assists. LLM mapped the term semantically to the correct column.

**9. Narrow Losses**

* **Prompt Used: "**How many narrow losses (≤3 goals) did Syracuse have?"
* **LLM Answer:** 5
* **Python Logic:**

games\_df["Margin"] = games\_df["SU\_Score"] - games\_df["Opponent\_Score"]

narrow\_losses = games\_df[(games\_df["Result"] == "L") & (games\_df["Margin"] >= -3)]

**Explanation:**The LLM initially gave an incorrect count because it didn’t combine both conditions (Result == 'L' and Margin >= -3). After being prompted to consider only losses with small negative margins, the correct count of 5 was given.

**10. Largest Win**

* **Prompt Used:** "What was Syracuse’s largest win?"
* **LLM Answer:** UALBANY on 2/07/25 (21–9, margin: 12)
* **Python Logic:**

largest\_win = games\_df[games\_df["Result"] == "W"].sort\_values(by="Margin", ascending=False).iloc[0]

**Explanation:**This involved calculating a custom "Margin" column (positive score difference) and filtering for wins. Once defined, LLM correctly identified the largest winning margin.

**11. Offense vs. Defense Focus**

* **Prompt Used:** *"If Syracuse wanted to win 2 more games next season, should they focus on offense or defense? Consider their 5 narrow losses and average goals scored/allowed."*
* **LLM Answer:** Initially vague. After providing the averages (12.37 goals scored, 11.63 allowed) and narrow loss count (5), the LLM suggested focusing slightly more on offense due to high correlation between shot volume and goals.
* **Python Logic:**

offense\_flips = int((narrow\_losses["Margin"] + 1 >= 0).sum())

defense\_flips = int((narrow\_losses["Margin"] + 1 >= 0).sum())

**Result:** Offense flips 3 games; defense flips 3 games — performance gains could come from improving either area.

* **Explanation:** The model’s recommendation leaned toward offense because the shot-to-goal correlation was high. Python results show both offense and defense could potentially yield the same number of additional wins.

**12. Game-Changer Player Identification**

* **Prompt Used:** *"Which single player, if improved, could flip the most close losses into wins?"*
* **LLM Answer:** Emma Ward — highlighted as a high-impact playmaker with both scoring and assisting ability.
* **Python Logic:**

player\_flip\_count = {player: offense\_flips for player in players\_df["Player"]}

game\_changer\_player = max(player\_flip\_count, key=player\_flip\_count.get)

**Result:** Emma Ward (3 potential flipped games).

* **Explanation:** Emma Ward’s combination of assists (46) and scoring ability positions her as the most likely player to change outcomes in narrow games.

**13. Correlation Between Shots and Goals**

* **Prompt Used:** *"Is there a strong relationship between shots taken and goals scored?"*
* **LLM Answer:** Yes — estimated r ≈ 0.97, indicating a very strong positive relationship.
* **Python Logic:**

players\_df["Shots"].corr(players\_df["Goals"])

**Result:** 0.974 correlation — confirms a strong linear relationship.

* **Explanation:** Strong statistical evidence suggests that increasing shot volume is highly likely to boost goal scoring, which supports focusing on offense for improved performance.

**14. Offensive Catalyst Identification (≥5 GP)**

* **Prompt Used:** *"Among players with at least 5 games played, who has the highest points per game?"*
* **LLM Answer:** Emma Ward — recognized for her high points per game while maintaining consistent participation.
* **Python Logic:**
* eligibility\_mask = players\_df["Games\_Played"] >= 5
* catalyst\_row = players\_df.loc[eligibility\_mask].sort\_values("Points\_Per\_Game", ascending=False).iloc[0]

**Result:** Emma Ward (4.00 PPG).

* **Explanation:** Filtering for players with sustained playtime (≥5 games) ensures the metric reflects consistent season-long performance rather than small-sample spikes.