Question 1

Scenario Setup - The home team begins the bottom of the ninth inning trailing by two runs. Each batter can only produce three outcomes: strikeout, single, or homerun. Runners advance exactly two bases on singles, and homeruns score all runners. No other means of advancement (e.g., walks or steals) are considered.
The goal is to determine the value of p (single probability for Batters 4 and 5) that gives the home team a 20% chance to score at least three runs before recording three outs.
Modeling Approach - This problem is best represented as a random variable simulation evaluated through a Monte Carlo simulation. Each inning is simulated from the start state (bases empty, 0 outs) until one of two terminal conditions occurs:

- The team scores ≥ 3 runs (walk-off win)
- The team makes 3 outs (game over)

Each simulation trial iterates through the batting order sequentially, looping back to the top as needed until the inning ends. On every plate appearance, a random outcome is drawn from the batter's defined probabilities, and the inning state—runs scored, outs, and base occupancy—is updated accordingly.
The win probability function $f(p) = P(score \geq 3;\ before\ 3\ outs)$ was estimated by repeating this inning simulation 500,000 times for candidate values of p between 0.0 and 0.9. Because f(p) increases monotonically with p, a binary search was used to efficiently identify the value of p that yields a target walk-off probability of 20%(0.1997).
Code Summary - The simulation was implemented in Python (q1.py) using only the random module, no external dependencies. Random seeds were fixed for reproducibility. Each inning followed the same probabilistic update rules:

- Strikeout (K): +1 out
- Single (1B): runners advance two bases; batter occupies first
- Homerun (HR): batter and all baserunners score; bases cleared

The code included three core functions:

- simulate_inning(p) — runs one inning and returns True if ≥ 3 runs are scored
- win_prob(p, sims) — estimates walk-off probability over multiple trials
- find_p_for_target(target) — performs binary search to find the value of p that achieves the desired win probability

Results - The simulation output shows a smooth, monotonic increase in walk-off probability as p increases. After 12 binary-search iterations and 500,000 Monte Carlo trials per step, the model converged to $p = 0.3805\ with\ f(p) = 0.1997$
Thus, when Batters 4 and 5 each have approximately a 38.0% chance of hitting a single (while maintaining a 10% homerun probability), the home team's chance of scoring at least three runs—and walking off in the ninth—is about 20%. Monte Carlo sampling error for 500,000 trials is roughly ±0.001, confirming that the result is statistically stable and reproducible across random seeds.
Interpretation - A moderate improvement in single probability for these middle-of-the-order hitters substantially boosts the team's late-inning win probability by extending innings and compounding base-runner advancement. Even small contact-rate gains in high-leverage lineup spots can translate into meaningful differences in overall game outcomes, quantifying the impact of consistent base-hit ability in late-game situations.

Question 2

Scenario Setup - As of September 23, 2025, Milwaukee had already clinched a postseason bye as the National League's top seed. Several teams remained in contention for the Wild Card spots, with the Dodgers, Cubs, Padres, and Mets tightly clustered in the middle of the standings. Each team's remaining regular-season schedule was simulated assuming a 50% chance of winning any given game. The remaining head-to-head series were as follows:

- Dodgers vs Diamondbacks (3 games)
- Cubs vs Mets (3 games)
- Brewers vs Padres (2 games)

Tiebreakers were determined by the order of the standings on September 23.
Division winners occupy seeds 1–3, while the remaining three playoff teams take seeds 4–6 in order of record. The MLB playoff format follows a 12-team structure:

- Wild Card Series (best-of-three): 3 vs 6 and 4 vs 5
- Division Series (best-of-five): 1 vs winner of 4/5, 2 vs winner of 3/6
- The Brewers, as the top seed, enter the Division Series and will face the winner of the 4–5 matchup.

The objective is to determine the probability that Milwaukee's Division Series opponent is the San Diego Padres.

Modeling Approach - The calculation proceeds in two stages:

1. Regular Season Simulation

Each remaining regular-season game was modeled as an independent Bernoulli trial with a 50 % win probability for either team. For head-to-head matchups (e.g., Cubs vs Mets, Dodgers vs D-backs, Brewers vs Padres), outcomes were paired so that one team's win automatically counted as the opponent's loss. After all games were simulated, win totals were updated and tiebreakers were applied following the existing September 23 standings order to assign playoff seeds 1–6.

2. Postseason Simulation

The Wild Card Round was simulated according to the official bracket format. The higher-seeded team was assumed to win each game with probability 0.55 in a best-of-three series. The Brewers, having secured the No. 1 seed, automatically advanced to the Division Series, where they would face the winner of the 4-vs-5 Wild Card matchup.
Given the interdependent outcomes of multiple concurrent series, the overall probability was estimated using a Monte Carlo simulation across 200,000 to 500,000 complete seasons.

Code Summary - The Python file (q2.py) using only the built-in random module, with fixed random seeds for reproducibility. Each iteration executed the following sequence:

1. Simulate remaining regular-season games and update win–loss records.
2. Apply tiebreakers and assign playoff seeds 1–6.
3. Simulate both Wild Card series (3 vs 6 and 4 vs 5).
4. Record whether Milwaukee (Seed 1) faced San Diego (as winner of 4/5) in the Division Series.

This process was repeated hundreds of thousands of times to approximate the long-run probability distribution of postseason outcomes.

Results - Running 200,000 Monte Carlo trials produced the following frequencies:

1. Padres finish as #5 seed = 1.000

2. Padres vs Cubs in Wild Card = 0.657
3. Padres win Wild Card Series = 0.426
4. Brewers face Padres in NLDS = 0.4258

San Diego remained locked into the #5 seed in all simulations due to its secure standing relative to the Mets, Reds, and Diamondbacks. Consequently, the Cubs–Padres 4–5 Wild Card matchup occurred in roughly 66% of seasons, and San Diego advanced through that best-of-three series 42.6% of the time. Combining these outcomes gives a 42.6% overall probability that Milwaukee’s Division Series opponent is San Diego. Monte Carlo error for 200,000 trials is approximately ±0.003, confirming the stability of these estimates.

Interpretation

The simulation results indicate that the San Diego Padres are effectively locked into the #5 seed, finishing there in every simulated outcome. Their lead over the Mets, Reds, and Diamondbacks is large enough that, under neutral 50/50 game assumptions, no reasonable combination of remaining results alters their seeding. As a result, the Padres face the Chicago Cubs (#4 seed) in the Wild Card Series roughly 66% of the time, reflecting the Cubs’ moderate advantage over other Wild Card contenders for the #4 position. Given a 55% per-game win probability for the higher seed, San Diego advances through the best-of-three Wild Card Series about 43% of the time.

Question 3

To compare the two contract structures, a Monte Carlo simulation was used to model the player's future value under uncertainty. Each season carries a 20% chance of a full-season injury, resulting in 0 WAR for that year and a 1.5 WAR reduction in the following season's projection. The simulation was run for 100,000 trials, using the WAR projections and salary structures provided for both the six-year year-to-year option and the eight-year guaranteed deal. Each trial calculates the team's net value—defined as (WAR × \$8M per WAR) − salary—for both contract paths. The guaranteed contract pays all eight years regardless of injury, while the year-to-year deal allows the team to decide annually whether to continue. The goal was to find the value of X, the guaranteed salary for 2032–2033, that makes the two contracts equally valuable to the team in expectation.

Using a binary search between \$5M and \$20M, the simulation converged to a break-even point of approximately \$11.4 million per year for each of the final two seasons. This means that, given the injury risk and expected WAR decline, offering the pitcher a fully guaranteed 8-year contract with 2032–2033 salaries of around \$11–12 million would yield the same expected team value as continuing year-to-year.

This model assumes independent injuries, a constant \$8M/WAR market rate, and no discounting or inflation. In practice, a more refined version would incorporate correlated injury risks, discount future payments, and apply an aging curve to WAR projections. Nonetheless, this simulation provides a reasonable first-pass estimate of how much long-term salary risk the team can absorb while remaining financially indifferent between the two contract structures.

Running the simulation files

All three scripts (q1.py, q2.py, and q3.py) are written in Python 3 and require no external libraries beyond the standard library (random, math, numpy for Q3).
All scripts are fully reproducible using fixed random seeds.

Environment Setup

1. Ensure you have Python 3.9+ installed. Check version:
    a. python --version
2. (Optional) Create and activate a virtual environment:
    a. python -m venv .venv
    b. source .venv/bin/activate   # Mac/Linux
    c. .venv\Scripts\activate   # Windows
3. Install NumPy (required for q3.py):
    a. pip install numpy

Running Each Simulation

Each file can be executed directly from the command line while inside the brewers_rnd_questionnaire/ directory.

Q1 – Walk-off Win Probability

Simulates the bottom of the 9th inning scenario and finds the value of p that gives a 20% walk-off win chance. Run with python q1.py in terminal

Q2 – Probability Brewers Face Padres in NLDS

Simulates remaining regular-season games and postseason matchups to estimate how often Milwaukee plays San Diego in the NLDS. Run with python q2.py in terminal

Q3 – Contract Extension Valuation

Uses a Monte Carlo simulation to compare a 6-year year-to-year contract vs. an 8-year guaranteed contract and solve for the break-even salary X. Run with python q3.py in terminal

Question 4

Key challenges in assessing usefulness and validity

The main challenges are data quality, signal extraction, and baseball interpretation.
First, we must confirm the technology's reliability (consistent readings) and validity (accurate measurements). A sensor is useless if "shoulder torque" fluctuates by 10% when nothing has changed. Second, these systems produce massive, high-dimensional datasets, thousands of readings per pitch. Most of those metrics will be noise. The challenge is identifying the handful that truly matter for pitch design, fatigue management, or injury prevention. Finally, this data lacks context and historical baselines. Without multiple seasons of comparable readings, we don't yet know whether certain movement patterns are "healthy" or "risky." Integrating this data with existing systems (Statcast, medical, and workload logs) also requires careful engineering so it can be understood by both analysts and coaches.

Approach to Test and Validate

I would use a two-phase validation pipeline combining engineering accuracy with baseball practicality.

1. Define the Baseball Question:
    a. Start with the goal—can this system spot early signs of elbow stress or fatigue before velocity loss? This keeps the analysis tied to real player-health and development decisions.
2. Technical Validation:
    a. Test the tech in controlled bullpen sessions alongside gold-standard systems like motion capture or force plates.
    b. Measure accuracy, consistency, and calibration drift.
    c. If readings for key metrics (e.g., shoulder torque, hip-shoulder separation) vary by more than ~5%, it's not reliable enough yet.
3. Baseball Validation:
    a. Deploy in minor-league or rehab settings to see if new variables actually explain real outcomes—velocity drops, command issues, or injuries.
    b. Track whether the system provides predictive lift and actionable insights for coaches.

The goal isn't just precision—it's helping pitchers perform better and stay healthier.

Evidence Required for Investment

Before committing financially, I would require:

- Predictive Lift: Quantifiable improvement in model accuracy, for example, if adding biomechanical variables improves an injury risk classifier's AUROC by ≥0.05 or enhances forecast accuracy in pitching effectiveness models.
- Actionable Insights: The data must inform coach-facing tools—like identifying early warning patterns (e.g., release-side drift or reduced hip-shoulder separation) that suggest fatigue before a pitcher's velocity dips.
- Operational Scalability: The system must capture usable data across affiliates, indoors and outdoors, without constant recalibration or full-time data engineers.

In short: the tech must make a scout, analyst, or coach measurably better at their job.

Modeling Challenges without historical training sets
When there's no historical data, the biggest challenge is avoiding false confidence. With small samples, it's easy to see patterns that don't actually exist. To address this, I'd start simple: group pitchers by mechanical "archetypes" using clustering to find broad trends, then work with coaches to interpret what differentiates those groups. Over time, we can track whether certain movement profiles hold up or break down under workload. As data accumulates, models can evolve — adding statistical regularization and cross-season validation to refine predictions without losing interpretability. The goal is to balance machine precision with baseball sense: the model identifies a flag, and a coach confirms whether it's meaningful.

Question 5

Automated Ball-Strike (ABS) Challenge Strategy
If given access to AAA challenge data, I would focus on identifying optimal challenge timing, personnel tendencies, and situational value to develop a major-league challenge strategy grounded in evidence, not instinct.

1. Quantify Challenge Value
    a. First, I'd measure the run expectancy change for overturned versus upheld calls by count and base-out state. For example, a missed strike three in a 3–2 count with runners on base carries far greater impact than a first-pitch ball.

Modeling this using win probability added (WPA) allows us to prioritize challenges with the highest expected return.

2. Analyze Behavioral Patterns
    a. Using AAA data, I'd study:
        i. Challenge success rate by count and pitch type (e.g., breaking balls low vs. fastballs up).
    b. Catcher and pitcher influence, identifying who frames well enough that overturns are rare.
    c. Hitter tendencies, such as who has the best visual judgment or tends to over-challenge marginal pitches.

These insights can guide which players should make on-field challenge decisions at the MLB level—potentially empowering catchers or bench coaches with the best historical accuracy.

3. Model Optimal Usage
    a. Using Monte Carlo or decision-tree simulation, I'd model expected outcomes under different challenge strategies: aggressive early use, late-game preservation, or targeted high-leverage deployment. The goal is to find a threshold policy (e.g., challenge only when success probability × run value > expected future opportunity).
4. Game Planning Integration
    a. Finally, I'd operationalize these insights into pregame briefings—by umpire zone tendencies, pitch type, and matchup—to guide challenge strategy.

The outcome: a data-driven framework that treats ABS challenges as a limited, leverage-sensitive resource—much like bullpen usage or pinch-hitting decisions.

Question 6

Why increasing bat speed might lower predicted OPS

A negative relationship between bat speed and predicted OPS can arise from both statistical and baseball factors.

Statistically, this may reflect multicollinearity among input variables. Bat speed often correlates strongly with exit velocity and attack angle, so the model may "penalize" bat speed when these features overlap, interpreting isolated increases as noise. This is common in models that lack proper feature regularization or have imbalanced sample distributions.

From a baseball perspective, higher bat speed alone doesn't guarantee better contact quality. Hitters who swing harder might sacrifice contact rate, barrel control, or spray efficiency, leading to more whiffs or mishits despite faster bats. If the training data overrepresents players with high bat speed but low contact skill (e.g., power-oriented minor leaguers), the model could mistakenly learn that "more speed = lower OPS."

To confirm, I'd check partial dependence plots, correlation matrices, and variance inflation factors to see whether bat speed interacts nonlinearly or redundantly with other predictors.

Evaluating Model Accuracy and Utility

To test accuracy, I'd follow these steps:

1. Use cross-validation across multiple seasons and player groups to ensure the model generalizes, not just fits historical noise.
2. Compare predictive metrics (R^2, RMSE, and rank correlation) against simple baseline models—such as prior-year OPS or exit velocity alone.
3. Assess calibration—whether predicted OPS aligns with observed OPS distributions.
4. Interpret residuals by player type (power, contact, switch-hitter) to identify systematic biases.

Utility is proven when the model outperforms simpler alternatives and provides interpretable insights coaches can act on—e.g., showing that efficient swing planes, not raw bat speed, drive consistent offensive production.