KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING
School of Data Science and Analytics

Applying Self-Referential Grades to NFL Quarterback Play

Macsen Higgins – Class of Dec. 2026
Professor Bowen Eldridge

INTRODUCTION

Across all major sports, numbers lie the most in american football, especially at the quarterback position. Two years ago, Tua Tagovailoa led the NFL in passing yards. Today, his $200M contract is considered one of the worst in the sport. Conversely, Sam Darnold was an unimportant journeyman before being thrust into the starting role for the Vikings in 2024. Despite putting up top-ten counting statistics across the board, Darnold did not receive an offer to return to the team and instead signed in Seattle for only $30M annually, less than 60% of the salary of the league's top QBs. It can be hard to know how good a player is based on their numbers alone at the position.

The goal of this project is to measure QB performance through advanced metrics and filter those raw statistics through a grading system that evaluates how complete a player's skillset is. Factors in the grading system include one's ability to throw accurately across the entire field, maintain success passing even when it is expected, and avoid declining efficiency when their team is close to the endzone. By analyzing the relationship between raw performance and grading results, we can improve our understanding of who is elevating their lackluster situation versus whose situation is elevating their performance metrics.

METHODS

There are three main statistical values that must be understood before further explanation of the methods of this study continues.

- Expected Points Added (EPA): The difference between a team's expected points for a given drive after a play vs. before the play.
- Success Rate (SR): The proportion of plays in a given sample that have a positive EPA value.
- Completion Percentage Over Expected (CPOE): A reward/punishment model for passing based on the difficulty of any given throw. Greater rewards for making hard throws, greater punishments for missing easy throws.

Being compared against a list of qualified QBs, each QB is assigned their raw performance score as the aggregate of their percentiles in EPA/play, SR, and CPOE.

The first grading criteria is cross-field accuracy, which groups the CPOE of all pass attempts by horizontal direction (left, middle, right) and depth (<0, <5, <10, <20 and >20 air yards) to create a grid makeup of a QB's passing ability by location. Given the set of 14 groups (screen passes to the middle are omitted due to their infrequency), a numerical grade is calculated through a formula that most closely resembles a weighted z-score-based normalization and shrinkage model, which weighs each box based on sample size, rewards positive values across the board, and punishes clear "weak points" that demonstrate a QB's inability to throw to certain parts of the field. The numerical output of the grading is then normalized to fit a curve such that 0.50 represents an average grade, and better/worse charts will receive better/worse grades.

The second grading criteria is modeled on SR vs. Expected pass rate (xpass), which is an algorithmically determined value before each play based on time, score, down, distance and past behavior that predicts the likelihood of a passing play. It is generally expected that QBs will have a lower success rate passing the easier it is to predict, but the severity of that decline the point of focus for this grade. Given all plays where xpass is between 0.30 and 1 binned by increments of 0.1, a linear trend can be observed in the change in SR. A model is created for the league as a whole and the QB itself, and the slopes of those two trendlines are compared using linear regression, creating a delta value that is added and scaled to a base score of 0.50.

The third and final grading criteria is a comparison of SR when the offense is 5-20 yards away from their opponents endzone, which denotes situations that are in the "red zone" but not on the goal line. Across the league this season (excluding handoffs), offenses are 18% less successful in the range specified than outside. The grading formula for this statistic is the simplest, and the grade for redzone SR is (redzone SR / non-redzone SR), with a cap at 1.0.

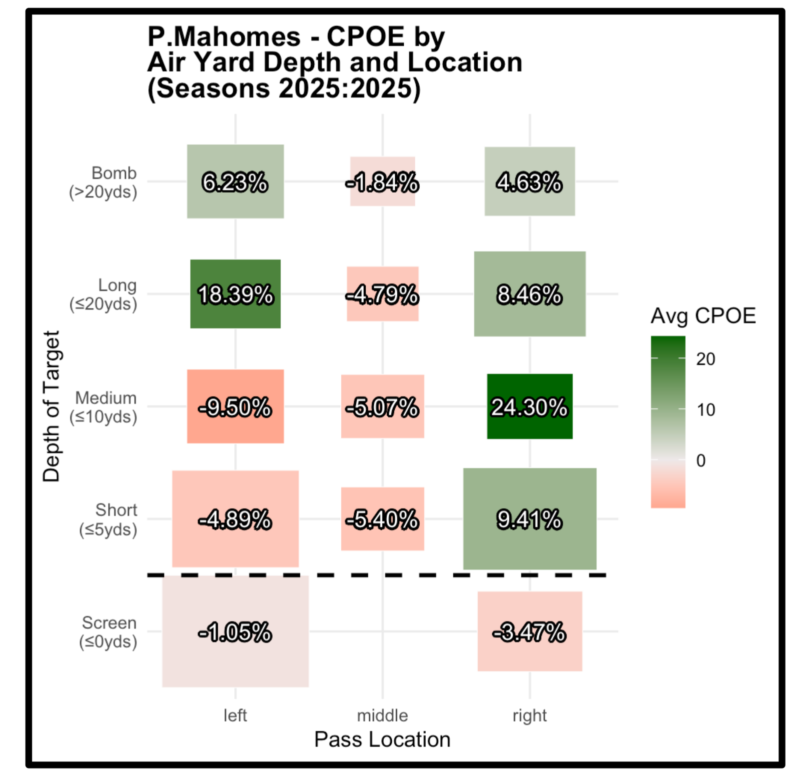
With our grades established, each QB's overall grade is calculated as the mean of their three graded values, and an adjusted final score is calculated by multiplying their raw score by their mean grade, effectively using the grade as a filter through which the raw score is processed.

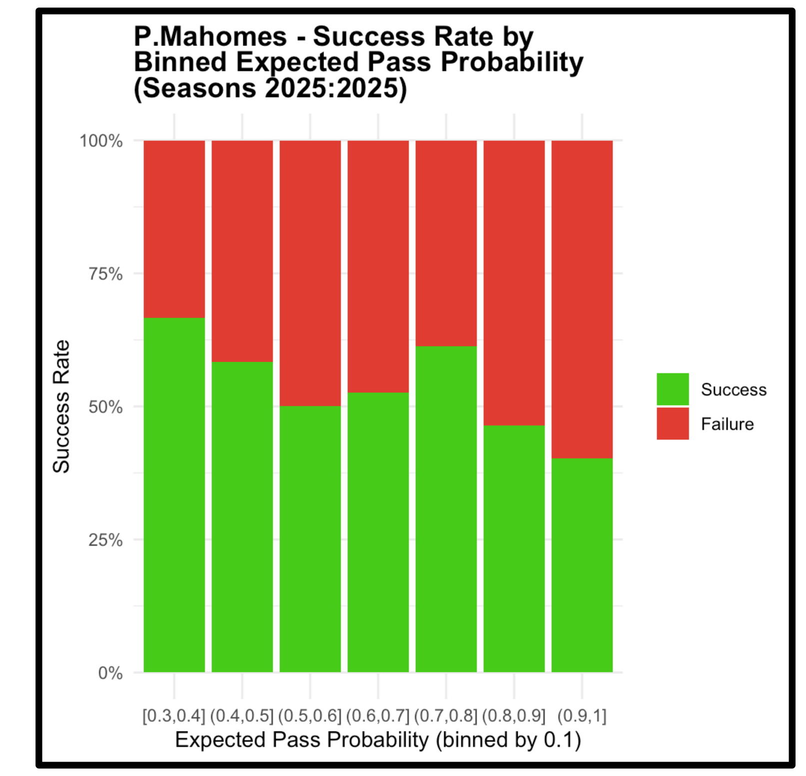
All data for this project was obtained from nflfastr, an open-source NFL play-by-play scraper.

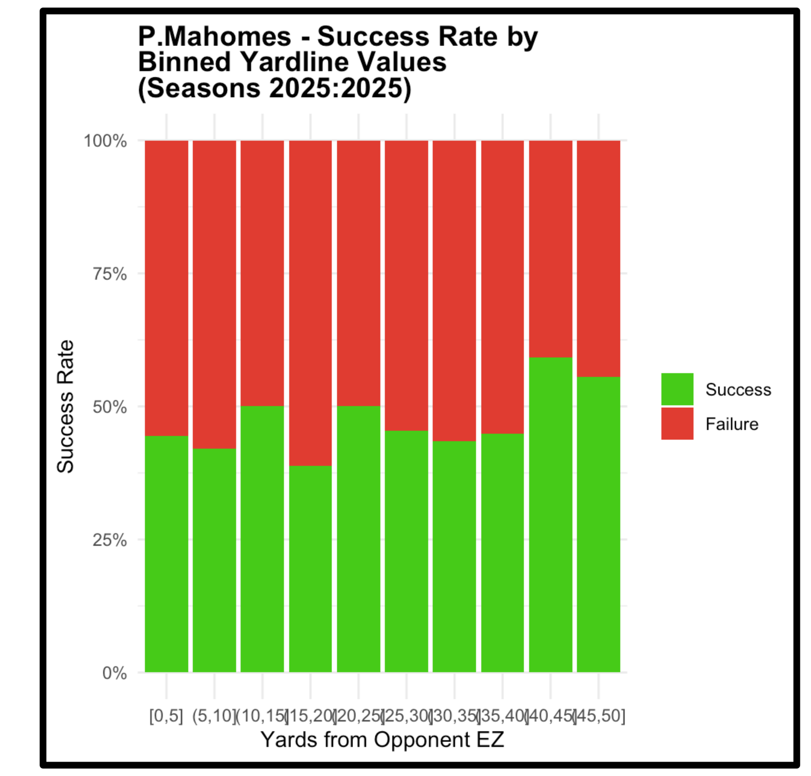
RESULTS

Using Kansas City Chiefs QB Patrick Mahomes as an example, we can illustrate the outcomes of his performance in each of our three grading criteria by running an individual QB grade function.

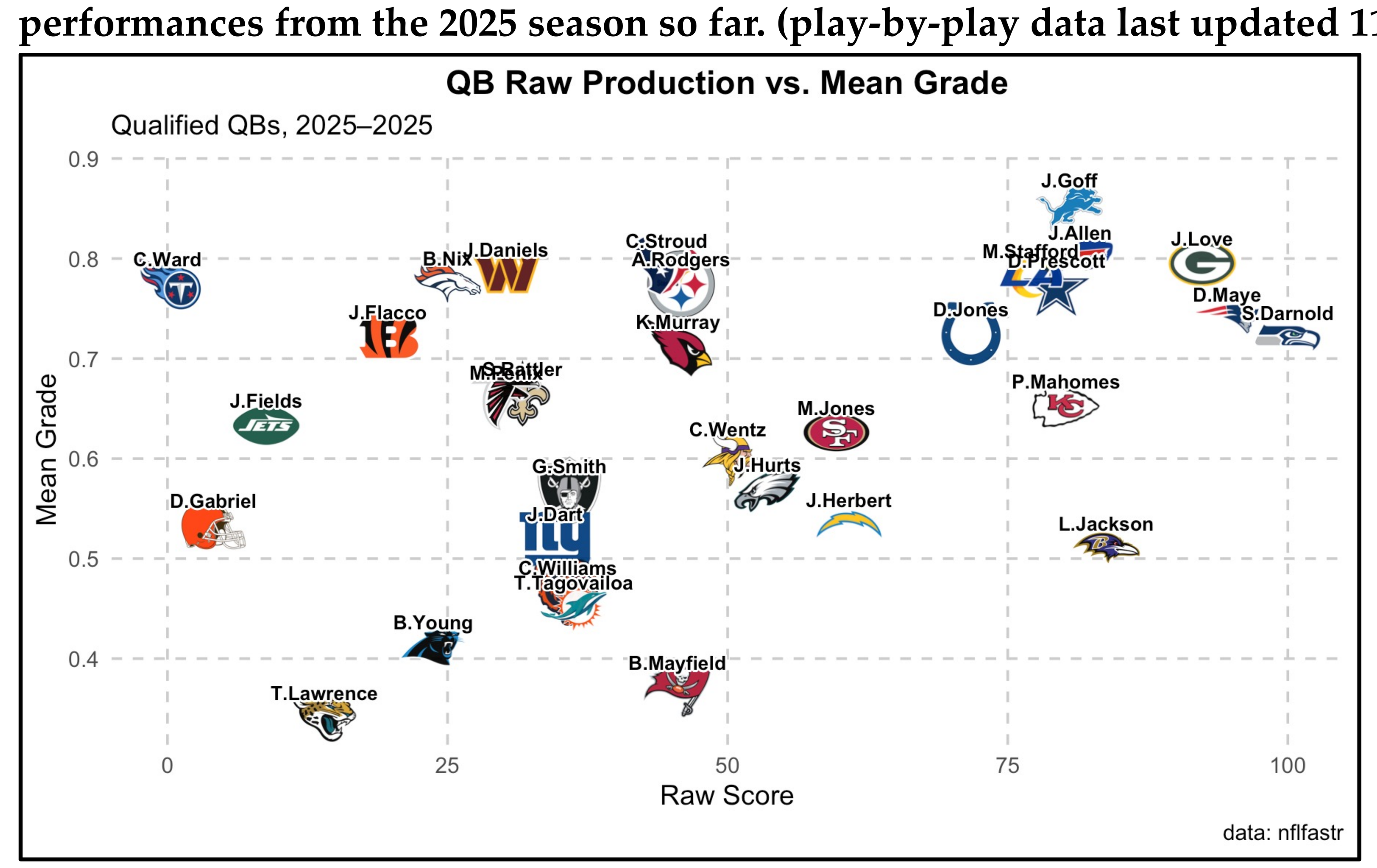
QB Name:	**P. Mahomes**
Years Included:	2025-2025
EPA/Play Percentile:	90.32
SR Percentile:	93.55
CPOE Percentile:	60.00
Raw Score:	**81.29**
CPOE Zones Grade:	0.520
SR vs. Xpass Grade:	0.492
SR vs. Yardline Grade:	0.941
Mean Grade:	**0.651**
Adjusted Score:	**52.91**

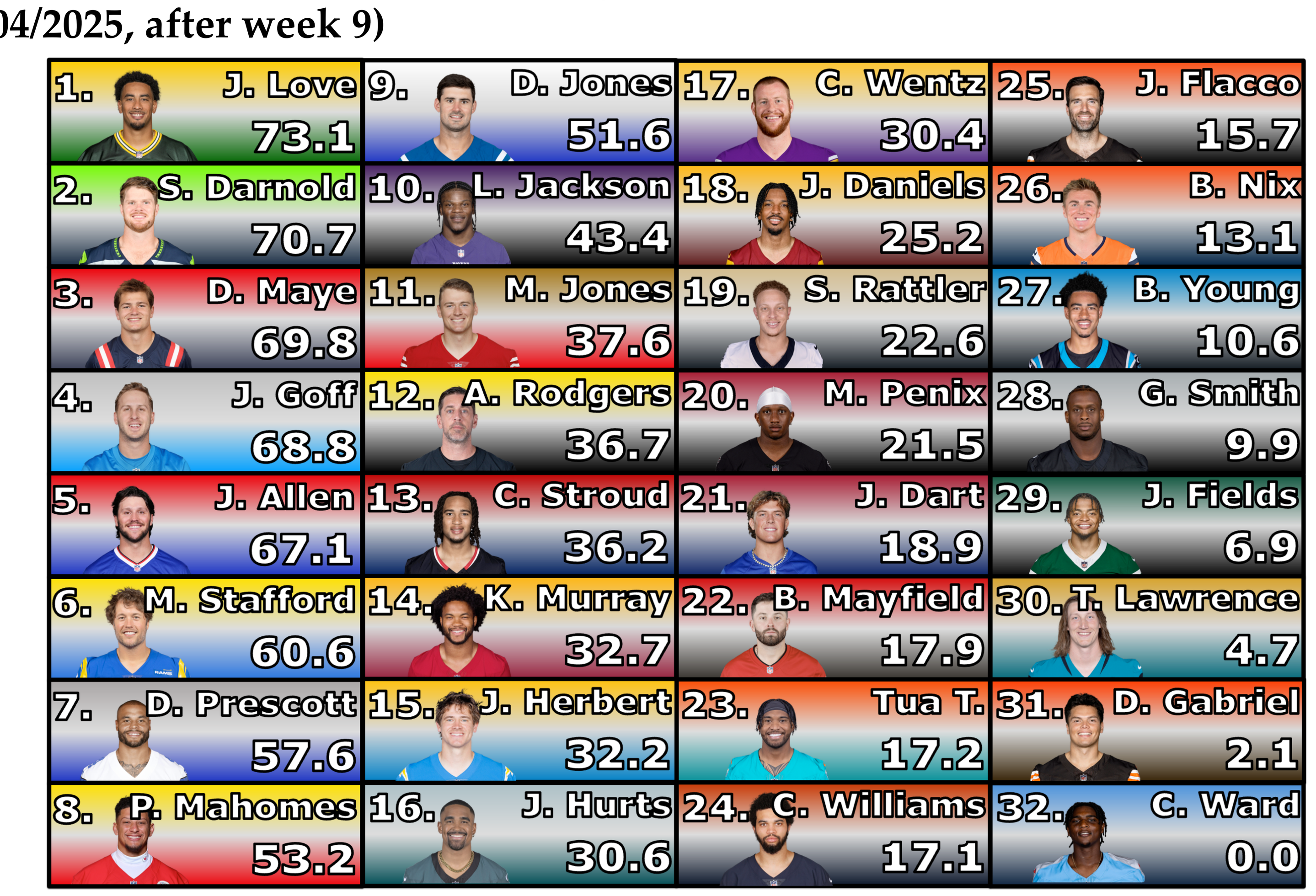






Running the grading function on the top 32 QBs in pass attempts this season, we can plot raw score vs. mean grade to recontextualize individual performances from the 2025 season so far. (play-by-play data last updated 11/04/2025, after week 9)





CONCLUSIONS

Upon reaching the conclusion of this exercise, it seems as though the grading metric is somewhat flawed in its current form, which can be observed through its output as well as its internal structure. For instance, the standard deviation of the SR vs. Yardline grade is much lower than the other two grades, making it inherently less impactful in overall grading. In addition, the system does not capture the abilities of rushing QBs well in its current form, as 2/3 of the grades can only be based on pass plays. If this idea were to be further developed over the course of another semester, I believe that it could turn into a tool that provides reliable insight unavailable to the naked eye, and I have faith in its structure as a balance of raw performance filtered through a series of test results, although the number of tests and weights of each grade would likely need modification. Regardless of the efficacy of the grading system itself, there are still significant takeaways to be made from the tools that emerged from this project. The heatmap for CPOE, which can also be tweaked for EPA or SR if desired, could be a valuable tool for analysts and fans alike for developing an understanding of a QB's "weak spots" or lack thereof on the field.

R CODE

Key code for CPOE Heatmap grading

```
  K <- median(df$plays, na.rm = TRUE)
  df <- df %>%
    mutate(w = plays / (plays + K),
      cpoe_shrunk = w * cpoe_prop)
S <- S_pp / 100
df <- df %>%
    mutate(
      z   = pmax(pmin(cpoe_shrunk / S, 3), -3),
s_z = ifelse(z >= 0, 50 + (50/3)*z, 50 + (60/3)*z),
      s_z = pmin(s_z, 90)
    )
    grade <- weighted.mean(df$s_z, df$w, na.rm = TRUE)
```

Key code for SR vs. Xpass grading

```
qbStats <- SRvsXPass(qbName, pbp, print)
  leagueStats <- SRvsXPass(, pbp, print)
  qbModel <- lm(success_rate ~ bin_mid,
  data = qbStats, weights = plays)
  qb_slope <- coef(qbModel)[2]
  leagueModel <- lm(success_rate ~ bin_mid,
  data = leagueStats)
  league_slope <- coef(leagueModel)[2]
  slope_diff <- qb_slope - league_slope
  scale <- 0.125
  grade <- 50 + (slope_diff / scale) * 50
  grade <- max(min(grade, 100), 0) / 100
```

Key code for SR vs. Yardline grading

```
data <- data |>
    mutate(
      yardline_bin = cut(yardline_100, breaks = seq(0, 50, by = 5),
      include.lowest = TRUE),
      redzone = yardline_100 <= 20 & yardline_100 > 5
    ) |>
    group_by(yardline_bin, redzone) |>
    summarize(success_rate = mean(success, na.rm = TRUE))
returnData <- data |>
    group_by(redzone) |>
    summarize(success_rate = mean(success_rate, na.rm = TRUE), .groups = "drop") |>
    mutate(ratio = redzone_TRUE / redzone_FALSE)
return(max(min(returnData$redzone_TRUE / returnData$redzone_FALSE, 1), 0))
```