# Assignment 4: Evaluating Advanced Quarterback Statistics

Due Friday, November 7 at 11:59 PM

In this assignment, you will analyze quarterback performance using two key advanced statistics: Total Quarterback Rating (QBR) and Expected Points Added (EPA). These metrics go beyond traditional stats like passing yards and touchdowns to provide a more comprehensive evaluation of a quarterback's impact on the game. Through this assignment, you will explore how well these metrics correlate with winning and whether they provide additional insight beyond traditional quarterback stats.

Submit your assignment to Gradescope as a PDF file generated from your R Markdown. Make sure your PDF includes all code, outputs, and written answers to questions. All code and outputs used to answer the written questions must be included in the PDF for full credit.

## Cleaning the data and calcuating QB stats [15 points]

You will use the R package `nflreadr` to obtain weekly data on quarterback and team performance. Make sure to install `nflreadr` first if you have not already.

```
library(nflreadr)
library(tidyverse)
```

1 [5 points]. Use the function `load_schedules()` from `nflreadr` to load the 2024 NFL team performance. Use this dataframe to create a new dataframe called `game_results` with the following columns:

- `season` : season in which the game was played
- `game_id` : the game id
- `game_type` : the game type
- `week`: the week in the season that the game took place
- `team`: one of the teams playing in the game
- `points`: points that `team` scored in the game in week `week`
- `points_against`: points that the opposing team scored in that game
- `win`: whether or not `team` won the game

Each game should have two rows (one for each team that played in the game). Filter the dataset to only include regular season games (based on the `game_type` column).

Print the first few rows of `game_results`. How may rows are in `game_results`?

```
schedules <- load_schedules(seasons = 2024)

# Create game_results dataframe
game_results <- schedules %>%
filter(game_type == "REG") %>%
select(season, game_id, game_type, week, home_team, away_team, home_score, away_score) %>%
mutate(
home_win = ifelse(home_score > away_score, 1, 0),
away_win = ifelse(away_score > home_score, 1, 0)
) %>%
pivot_longer(
cols = c(home_team, away_team),
names_to = "home_away",
```

```
values_to = "team"
) %>%
mutate(
points = ifelse(home_away == "home_team", home_score, away_score),
points_against = ifelse(home_away == "home_team", away_score, home_score),
win = ifelse((home_away == "home_team" & home_win == 1) |
(home_away == "away_team" & away_win == 1), 1, 0)
) %>%
select(season, game_id, game_type, week, team, points, points_against, win)

head(game_results)
```

```
## # A tibble: 6 x 8
##   season game_id         game_type  week team  points points_against   win
##    <int> <chr>           <chr>     <int> <chr>  <int>          <int> <dbl>
## 1   2024 2024_01_BAL_KC  REG           1 KC        27             20     1
## 2   2024 2024_01_BAL_KC  REG           1 BAL       20             27     0
## 3   2024 2024_01_GB_PHI  REG           1 PHI       34             29     1
## 4   2024 2024_01_GB_PHI  REG           1 GB        29             34     0
## 5   2024 2024_01_PIT_ATL REG           1 ATL       10             18     0
## 6   2024 2024_01_PIT_ATL REG           1 PIT       18             10     1
```

```
nrow(game_results)
```

```
## [1] 544
```

2 [2 points]. Take a look at the functions available in `nflreadr`: https://nflreadr.nflverse.com/reference/index.html

Find a function from this list that always you to access the ESPN QBR data. Load in the *weekly* ESPN QBR data for the 2024 season and call the dataframe `qbr_weekly_df`.

```
qbr_weekly_df <- load_espn_qbr(seasons = 2024, summary_type = "week")
head(qbr_weekly_df)
```

```
## # A tibble: 6 x 30
##   season season_type game_id   game_week week_text team_abb player_id name_short
##    <int> <chr>       <chr>         <int> <chr>     <chr>    <chr>     <chr>
## 1   2024 Regular     401671861         1 Week 1    IND      4429084   A. Richar~
## 2   2024 Regular     401671617         1 Week 1    BUF      3918298   J. Allen
## 3   2024 Regular     401671770         1 Week 1    TB       3052587   B. Mayfie~
## 4   2024 Regular     401671617         1 Week 1    ARI      3917315   K. Murray
## 5   2024 Regular     401671696         1 Week 1    SF       4361741   B. Purdy
## 6   2024 Regular     401671628         1 Week 1    NE       2578570   J. Brisse~
## # i 22 more variables: rank <dbl>, qbr_total <dbl>, pts_added <dbl>,
## #   qb_plays <dbl>, epa_total <dbl>, pass <dbl>, run <dbl>, exp_sack <dbl>,
## #   penalty <dbl>, qbr_raw <dbl>, sack <dbl>, name_first <chr>,
## #   name_last <chr>, name_display <chr>, headshot_href <chr>, team <chr>,
## #   opp_id <chr>, opp_abb <chr>, opp_team <chr>, opp_name <chr>,
## #   week_num <int>, qualified <lgl>
```

3 [3 points]. Join the game results data to the weekly QBR data. Filter the data to only look at regular season games.

*Hints:* One option is to join on team and game week. Note that team and game week columns have different names in each dataset. Also note that team abbreviations may not line up, so you might have to rename some team abbreviations in one of the datasets as part of the data cleaning process.

```r
qbr_weekly_df <- qbr_weekly_df %>%
mutate(team_abb = case_when(
team_abb == "WSH" ~ "WAS",
team_abb == "LA" ~ "LAR",
TRUE ~ team_abb
))

game_qbr_df <- qbr_weekly_df %>%
rename(week = week_num) %>%
inner_join(game_results, by = c("team_abb" = "team", "week" = "week")) %>%
filter(game_type == "REG")


head(game_qbr_df)
```

```
## # A tibble: 6 x 36
##   season.x season_type game_id.x game_week week_text team_abb player_id
##      <int> <chr>       <chr>         <int> <chr>     <chr>    <chr>
## 1     2024 Regular     401671861         1 Week 1    IND      4429084
## 2     2024 Regular     401671617         1 Week 1    BUF      3918298
## 3     2024 Regular     401671770         1 Week 1    TB       3052587
## 4     2024 Regular     401671617         1 Week 1    ARI      3917315
## 5     2024 Regular     401671696         1 Week 1    SF       4361741
## 6     2024 Regular     401671628         1 Week 1    NE       2578570
## # i 29 more variables: name_short <chr>, rank <dbl>, qbr_total <dbl>,
## #   pts_added <dbl>, qb_plays <dbl>, epa_total <dbl>, pass <dbl>, run <dbl>,
## #   exp_sack <dbl>, penalty <dbl>, qbr_raw <dbl>, sack <dbl>, name_first <chr>,
## #   name_last <chr>, name_display <chr>, headshot_href <chr>, team <chr>,
## #   opp_id <chr>, opp_abb <chr>, opp_team <chr>, opp_name <chr>, week <int>,
## #   qualified <lgl>, season.y <int>, game_id.y <chr>, game_type <chr>,
## #   points <int>, points_against <int>, win <dbl>
```

4 [3 points]. Create a new dataframe called `qb_stats` that for each quarterback calculates their average QBR (`qbr_total`), their average EPA (`epa_total`), the total number of team wins (for games that they played in), the average number of team points scored (for games that they played in), and the number of games they played in. *Hint:* I recommend grouping by `name_display` and `team_abb`.

```r
qb_stats <- game_qbr_df %>%
group_by(name_display, team_abb) %>%
summarize(
avg_qbr = mean(qbr_total, na.rm = TRUE),
avg_epa = mean(epa_total, na.rm = TRUE),
total_wins = sum(win, na.rm = TRUE),
avg_team_points = mean(points, na.rm = TRUE),
games_played = n(),
.groups = "drop"
)

head(qb_stats)
```

```
## # A tibble: 6 x 7
##   name_display  team_abb avg_qbr avg_epa total_wins avg_team_points games_played
##   <chr>         <chr>      <dbl>   <dbl>      <dbl>           <dbl>        <int>
## 1 Aaron Rodgers NYJ         45.5    3.04          5            19.9           17
## 2 Aidan O'Conn~ LV          41.8    2.77          2            17.9            7
```

```
## 3 Andy Dalton   CAR         46.1    3.64          1            19.4            5
## 4 Anthony Rich~ IND         45.8    2.47          5            20.4           10
## 5 Bailey Zappe  CLE         17.6   -0.9           0            10              1
## 6 Baker Mayfie~ TB          57.8    4.87         11            29.9           18
```

5 [2 points]. Run the following code to create a dataframe with some traditional QB stats.

```
offense_df <- load_player_stats(seasons = 2024,
                                stat_type = "offense")

qb_trad_df <- offense_df %>%
  filter(position == "QB") %>%
  group_by(player_id, player_display_name) %>%
  summarize(comp_perc = sum(completions) / sum(attempts),
            passing_yards = sum(passing_yards),
            rushing_yards = sum(rushing_yards))
```

Join `qb_trad_df` to `qb_stats`. *Hint:* join on `name_display` (in `qb_stats`) and `player_display_name` (in `qb_trad_df`). You can uncomment and run the code below to clean player name in `qb_stats` and `qb_trad_df` to get rid of small differences in the way a player's name might be written (e.g., removes periods).

```
# Clean player name in qb_stats
qb_stats <- qb_stats %>%
  mutate(name_display = clean_player_names(name_display))
# Clean player name in qb_trad_df
qb_trad_df <- qb_trad_df %>%
  mutate(player_display_name = clean_player_names(player_display_name))

qb_combined <- qb_stats %>%
inner_join(qb_trad_df, by = c("name_display" = "player_display_name"))

head(qb_combined)
```

```
## # A tibble: 6 x 11
##   name_display  team_abb avg_qbr avg_epa total_wins avg_team_points games_played
##   <chr>         <chr>      <dbl>   <dbl>      <dbl>           <dbl>        <int>
## 1 Aaron Rodgers NYJ         45.5    3.04          5            19.9           17
## 2 Aidan OConne~ LV          41.8    2.77          2            17.9            7
## 3 Andy Dalton   CAR         46.1    3.64          1            19.4            5
## 4 Anthony Rich~ IND         45.8    2.47          5            20.4           10
## 5 Bailey Zappe  CLE         17.6   -0.9           0            10              1
## 6 Baker Mayfie~ TB          57.8    4.87         11            29.9           18
## # i 4 more variables: player_id <chr>, comp_perc <dbl>, passing_yards <int>,
## #   rushing_yards <int>
```

## Evaluating QB stats [9 points]

6 [3 points]. For each of the following quarterback metrics:

- QBR
- EPA
- Completion percentage
- Passing yards
- Rushing yards

Calculate their correlation with

- Wins

- Win percentages
- Average team points

Interpret these correlation values. Which quarterback stat do you think is best? Explain your reasoning.

```
correlations <- qb_combined %>%
summarize(
cor_qbr_wins = cor(avg_qbr, total_wins, use = "complete.obs"),
cor_epa_wins = cor(avg_epa, total_wins, use = "complete.obs"),
cor_comp_wins = cor(comp_perc, total_wins, use = "complete.obs"),
cor_pass_wins = cor(passing_yards, total_wins, use = "complete.obs"),
cor_rush_wins = cor(rushing_yards, total_wins, use = "complete.obs"),

cor_qbr_pts = cor(avg_qbr, avg_team_points, use = "complete.obs"),
cor_epa_pts = cor(avg_epa, avg_team_points, use = "complete.obs"),
cor_comp_pts = cor(comp_perc, avg_team_points, use = "complete.obs"),
cor_pass_pts = cor(passing_yards, avg_team_points, use = "complete.obs"),
cor_rush_pts = cor(rushing_yards, avg_team_points, use = "complete.obs"))

correlations
```

```
## # A tibble: 1 x 10
##   cor_qbr_wins cor_epa_wins cor_comp_wins cor_pass_wins cor_rush_wins
##          <dbl>        <dbl>         <dbl>         <dbl>         <dbl>
## 1        0.483        0.548         0.343         0.898         0.657
## # i 5 more variables: cor_qbr_pts <dbl>, cor_epa_pts <dbl>, cor_comp_pts <dbl>,
## #   cor_pass_pts <dbl>, cor_rush_pts <dbl>
```

QBR and EPA show the highest correlation with team points since both adjust for game context and situational difficulty. Traditional stats like pass/rush yards can be misleading as shown by the high correlation with wins compared to QBR and EPA, but are less correlated to the points that QBR and EPA. We can safely conclude QBR and EPA are likely the best in terms of predictive accuracy.

7 [3 points]. Due to the way we joined team results with QB stats, if multiple QBs played for the same team in a game, each QB was credited with the team's win (or loss). Do you think this is the best way to attribute wins? Why or why not? If not, what alternative method could provide a more accurate representation?

Attributing team wins equally to all quarterbacks who played inflates totals for backups with less playing time. A better method would be to weight wins by snap count or pass attempts, giving starters the majority share. Additionally, you could also use team-level metrics and connect them to each QB's share of plays.

8 [3 points]. Calculate the correlations between QBR, EPA, completion percentage, passing yards and rushing yards. What do these correlation values tell us about those stats?

```
qb_perf_corrs <- qb_combined %>%
select(avg_qbr, avg_epa, comp_perc, passing_yards, rushing_yards) %>%
cor(use = "complete.obs")

qb_perf_corrs
```

```
##                 avg_qbr   avg_epa comp_perc passing_yards rushing_yards
## avg_qbr       1.0000000 0.9071378 0.4491731     0.4198599     0.3802025
## avg_epa       0.9071378 1.0000000 0.5300902     0.5198260     0.4551437
## comp_perc     0.4491731 0.5300902 1.0000000     0.3412535     0.1699513
## passing_yards 0.4198599 0.5198260 0.3412535     1.0000000     0.6357964
## rushing_yards 0.3802025 0.4551437 0.1699513     0.6357964     1.0000000
```

QBR and EPA are strongly correlated with values over 0.9, as both capture efficiency. Completion percentage correlates moderately with QBR but weakly with passing/rushing yards. Rushing yards are usually very

weakly correlated with the others, providing an indicator that rushing yards require a unique skill set.

## Predicting wins [6 points]

9 [4 points]. Run the following code to get each team's defensive EPA per play:

```
pbp_data <- load_pbp(2024) # load 2024 play by play data
# get defensive epa
defensive_epa <- pbp_data %>%
  filter(is.na(defteam) == FALSE) %>%
  group_by(defteam) %>%
  summarize(def_epa_per_play = mean(epa, na.rm = TRUE))
# Run this code to rename WAS as WSH and LA as LAR to match the other data,
# if that's how you set up the data above
defensive_epa <- defensive_epa %>%
 mutate(defteam = case_when(defteam == "LA" ~ "LAR",                                    defteam == "WAS" ~ "WS
```

The more negative a team's defensive EPA per play, the better its defense (since they are preventing more opponent points per play on average).

Group by team and summarize `game_results` to get total team wins for the 2024 regular season. Group by team and summarize `qb_stats` to calculate the average QBR per team for the season. Join these summarized tables to `defensive_epa`. Note: make sure team abbreviations are consistent across all datasets.

Fit a linear regression regressing team wins on `def_epa_per_play` and average QBR. Interpret the results.

```
team_wins <- game_results %>%
group_by(team) %>%
summarize(total_wins = sum(win))

team_qbr <- qb_stats %>%
group_by(team_abb) %>%
summarize(avg_qbr_team = mean(avg_qbr, na.rm = TRUE))

team_model_df <- team_wins %>%
inner_join(team_qbr, by = c("team" = "team_abb")) %>%
inner_join(defensive_epa, by = c("team" = "defteam"))

model <- lm(total_wins ~ avg_qbr_team + def_epa_per_play, data = team_model_df)
summary(model)
```

```
##
## Call:
## lm(formula = total_wins ~ avg_qbr_team + def_epa_per_play, data = team_model_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4936 -1.7287 -0.1035  1.4529  8.1261
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.21589    2.10605   0.577 0.568499
## avg_qbr_team       0.15056    0.04008   3.757 0.000839 ***
## def_epa_per_play -38.43836   10.55910  -3.640 0.001137 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.541 on 27 degrees of freedom
## Multiple R-squared:  0.5754, Adjusted R-squared:  0.5439
## F-statistic: 18.29 on 2 and 27 DF,  p-value: 9.505e-06
```

A positive coefficient for QBR means that higher offensive efficiency increases wins. A negative coefficient for defensive EPA indicates that stronger defenses, indicated by a more negative EPA, correlate with more wins.

10 [2 points]. League average QBR last year was 55 and league average defensive EPA was 0.014. Predict the number of wins for a team with league average values for QBR and defensive EPA. Does this value make sense to you?

```
avg_qbr_val <- 55
avg_def_epa_val <- 0.014

predicted_wins <- predict(
model,
newdata = data.frame(
avg_qbr_team = avg_qbr_val,
def_epa_per_play = avg_def_epa_val
)
)

predicted_wins
```

```
##        1
## 8.958461
```

The predicted value is close to the average number in wins for teams in the league. This makes sense given that a 0.500 team typically finishes near that mark.