SQL Queries / How to Merge Data for NFL Big Data Bowl 2023

File descriptions

Game data: The `games.csv` contains the teams playing in each game. The *key* variable is **`gameId`**.

Play data: The `plays.csv` file contains play-level information from each game. The *key* variables are **`gameId`** and **`playId`**.

Player data: The `players.csv` file contains player-level information from players that participated in any of the tracking data files. The *key* variable is **`nflId`**.

PFF Scouting data: The `pffScoutingData.csv` file contains player-level scouting information for each game and play. The *key* variables are **`gameId`**, **`playId`**, and **`nflId`**.

Tracking data: Files `week[week].csv` contain player tracking data from season `[week]`. The *key* variables are **`gameId`**, **`playId`**, and **`nflId`**.

Play data

- **`gameId`**: Game identifier, unique (numeric)
- **`playId`**: Play identifier, not unique across games (numeric)
- `passResult`: Dropback outcome of the play (`C`: Complete pass, `I`: Incomplete pass, `S`: Quarterback sack, `IN`: Intercepted pass, `R`: Scramble, text)

Player data

- **`nflId`**: Player identification number, unique across players (numeric)
- `officialPosition`: Official player position (text)
- `displayName`: Player name (text)

PFF Scouting data

- **`gameId`**: Game identifier, unique (numeric)
- **`playId`**: Play identifier, not unique across games (numeric)
- **`nflId`**: Player identification number, unique across players (numeric)
- `pff_role`: The player's role on this play (text)
    - Possible values:

        - `Pass Rush`
        - `Pass`: Offensive player. Player identified as the passer
        - `Pass block`: Offensive player. Anyone fully blocking a defender from the QB, or anyone in a clear pass block stance
- `pff_beatenByDefender`: If player is a blocking offensive player, indicator for whether they are by a defender but was not charged for yielding a hit, hurry or sack (binary)
- `pff_hitAllowed`: If player is a blocking offensive player, indicator for whether they are responsible for a hit on the QB (binary)
- `pff_sackAllowed`: If player is a blocking offensive player, indicator for whether they are responsible for a sack on the QB (binary)
- `pff_nflIdBlockedPlayer`: If player is a blocking offensive player, the `nflId` of the first defender the offensive player blocked (numeric)
- `pff_blockType`: If player is a blocking offensive player, the type of block that the offensive player is executing on the defender (text)
    - NB: No Block - If a blocker executes no block on a play but simply runs his path or takes his pass set then we will note him with one all blocking line with this block type

Tracking data

Files `week[week].csv` contains player tracking data from week `[week]`.

- **`gameId`**: Game identifier, unique (numeric)
- **`playId`**: Play identifier, not unique across games (numeric)
- **`nflId`**: Player identification number, unique across players. When value is NA, row corresponds to ball. (numeric)
- `frameId`: Frame identifier for each play, starting at 1 (numeric)
- `time`: Time stamp of play (time, yyyy-mm-dd, hh:mm:ss)
- `x`: Player position along the long axis of the field, 0 - 120 yards. See Figure 1 below. (numeric)
- `y`: Player position along the short axis of the field, 0 - 53.3 yards. See Figure 1 below. (numeric)
- `event`: Tagged play details, including moment of ball snap, pass release, pass catch, tackle, etc (text)

Merging Tracking Data with PFF Scouting Data into df_tpff

- Merge on following key:
    - gameId
- Type of join:

    - INNER JOIN on gameId, nflId, and playId for tracking data and PFF data

Merging df_tpff with player_data into df_tpffp

- Merge on nflId
-

Merging df_tpffp with play data into df_tpffpp

- Merge on gameId
- This will help get rid of plays that are scrambles