Data Cleaning

Tate Huffman

1/31/2022

Overview

This is part of a series of RMarkdown files that will break down the code contained in the document at the end of the fall. This file in particular will redo the code that constructs the variables needed to run the O'Flaherty and Siow model, altering some components that may have been inaccurate in the first pass at this model.

Altering the Variables

The first code chunk adds the necessary variables for signal, exits, promotions, etc. for batters, and the second does the same for pitchers. Key changes in this version of the code include limiting the player pool to those who debuted from 1999 to 2009; altered promotion code to account for second-year promotions; playing time filters; and percentile-based performance standards.

```
n_years <- 10 # max number of seasons for players in the data

bat_performance <- bat %>%
  mutate(Level = case_when(Level == "Rookie" ~ 1,
                           Level == "Short-Season A" ~ 2,
                           Level == "A" ~ 3,
                           Level == "Adv A" ~ 4,
                           Level == "AA" ~ 5,
                           Level == "AAA" ~ 6,
                           Level == "MLB" ~ 7)) %>%  # converts level to scalar for computation
  group_by(Name) %>%
  mutate(yr_unique = cumsum(!duplicated(Year)), # counter of unique year played
         yr_debut = min(Year), # year of debut
         exit = if_else(yr_unique == max(yr_unique), 1, 0)) %>% # whether they exited
  ungroup() %>%
  group_by(Name, Year) %>%
  # one type of promotion:
  # if they played at multiple levels in the same year, we ASSUME this is a promotion
  # will have to lay out more rigorously why this assumption works
  mutate(promotion = if_else(n_distinct(Level) != 1, 1, 0)) %>%
  ungroup() %>%
  group_by(Name, Year) %>%
  mutate(Age = max(Age),
         pa_total = sum(PA), # total PA that season, for filtering purposes
         level_high = max(Level),
         level_low = min(Level), # so we know if next level is greater (for promotions)
         yr_unique = max(yr_unique),
         yr_debut = yr_debut,
```

```
        exit = max(exit),
        promotion = max(promotion)) %>%
  ungroup() %>%
  group_by(Name) %>%
  # promoted if you play at a strictly higher level next year
  mutate(promotion = if_else(level_high < lead(level_low, default = 0), 1, promotion)) %>%
  ungroup() %>%
  filter(pa_total > 20) %>% # filtering out pitchers, for the most part
  group_by(Year, Level) %>%
  # creating OPS threshold for a good signal
  mutate(ops_threshold = quantile(OPS, probs = 1/2, na.rm = TRUE),
        signal = if_else(OPS >= ops_threshold, 1, 0, missing = 0)) %>%
  ungroup() %>%
  arrange(Name, Year, Level) %>% # makes it easier to get lowest-level signal (our signal)
  group_by(Name, Year) %>%
  summarize(age = Age,
            pa_total = pa_total,
            level_high = level_high,
            level_low = level_low,
            yr_unique = yr_unique,
            yr_debut = yr_debut,
            exit = exit,
            promotion = max(promotion),
            signal = first(signal)) %>%
  unique() %>% # gets unique observations
  filter(yr_unique <= n_years, # takes only the first ten years of someone's career
        yr_debut >= 1999, # so we know that the first observation is a debut
        yr_debut <= 2009) # in order to know whether they exit at the end
```

```
pitch_performance <- pitch %>%
  mutate(Level = case_when(Level == "Rookie" ~ 1,
                           Level == "Short-Season A" ~ 2,
                           Level == "A" ~ 3,
                           Level == "Adv A" ~ 4,
                           Level == "AA" ~ 5,
                           Level == "AAA" ~ 6,
                           Level == "MLB" ~ 7)) %>%  # converts level to scalar for computation
  group_by(Name) %>%
  mutate(yr_unique = cumsum(!duplicated(Year)), # counter of unique year played
        yr_debut = min(Year), # year of debut
        exit = if_else(yr_unique == max(yr_unique), 1, 0)) %>% # whether they exited
  ungroup() %>%
  group_by(Name, Year) %>%
  # one type of promotion:
  # if they played at multiple levels in the same year, we ASSUME this is a promotion
  # will have to lay out more rigorously why this assumption works
  mutate(promotion = if_else(n_distinct(Level) != 1, 1, 0)) %>%
  ungroup() %>%
  group_by(Name, Year) %>%
  mutate(Age = max(Age),
        bf_total = sum(BF), # total PA that season, for filtering purposes
        level_high = max(Level),
        level_low = min(Level), # so we know if next level is greater (for promotions)
        yr_unique = max(yr_unique),
```

```
        yr_debut = yr_debut,
        exit = max(exit),
        promotion = max(promotion)) %>%
  ungroup() %>%
  group_by(Name) %>%
  # promoted if you play at a strictly higher level next year
  mutate(promotion = if_else(level_high < lead(level_low, default = 0), 1, promotion)) %>%
  ungroup() %>%
  filter(bf_total > 20) %>% # filtering out pitchers, for the most part
  group_by(Year, Level) %>%
  # creating ERA threshold for a good signal
  # ** if altering this, because lower is better, the quantiles are reversed **
  # i.e., the 75th percentile would actually be probs = 1/4
  mutate(era_threshold = quantile(ERA, probs = 1/2, na.rm = TRUE),
        signal = if_else(ERA <= era_threshold, 1, 0, missing = 0)) %>%
  ungroup() %>%
  arrange(Name, Year, Level) %>% # makes it easier to get lowest-level signal (our signal)
  group_by(Name, Year) %>%
  summarize(age = Age,
            bf_total = bf_total,
            level_high = level_high,
            level_low = level_low,
            yr_unique = yr_unique,
            yr_debut = yr_debut,
            exit = exit,
            promotion = max(promotion),
            signal = first(signal)) %>%
  unique() %>% # gets unique observations
  filter(yr_unique <= n_years, # takes only the first ten years of someone's career
        yr_debut >= 1999, # so we know that the first observation is a debut
        yr_debut <= 2009) # in order to know whether they exit at the end
```

```
# Saving this data, so we can just read it in next time

bat_performance %>%
  write_csv('../Data/Clean/bat/bat_performance_min_20pa.csv')

pitch_performance %>%
  write_csv('../Data/Clean/pitch/pitch_performance_min_20bf.csv')
```