Accessing and Analyzing MLB Pitch Tracking Data in R

Aaron R. Baggett, Ph.D.

University of Mary Hardin-Baylor
Department of Psychology

March 28, 2017

Resources

- Slides, data, and R code are available at:

bit.ly/austin_r

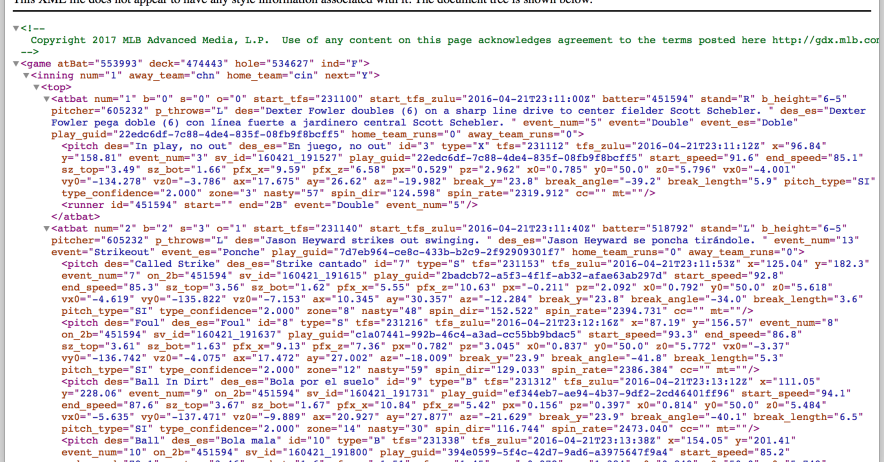
Major League Baseball (MLB) Pitch Tracking Data

MLB Pitch Tracking Data

- Since 2007, MLB has tracked pitch location and play-by-play data for all games
- Source: Sportvision PITCHf/x system
- PITCHf/x data are fed real time to mobile and desktop apps
- All data are stored in XML format on MLB servers

MLB Pitch Tracking Data

- Location: http://gd2.mlb.com/components/game/mlb/



Accessing MLB Pitch Tracking Data

Accessing MLB Pitch Tracking Data

- R packages:
  1. `pitchRx`: Data collection
  2. `dplyr`: Data analysis

pitchRx

- Prior to 2013, researchers had to scrape PITCHf/x data manually
- In 2013, Carson Sievert created the `pitchRx` R package
- `pitchRx` contains tools for accessing play-by-play data

```
pfx_db <- src_sqlite("pfx_db.sqlite3", create = TRUE)
files <- c("inning/inning_all.xml", "players.xml", "miniscoreboard.xml")
scrape(start = "YYY-MM-DD", end = "YYY-MM-DD", suffix = files,
  connect = pfx_db$con)
pfx_db <- src_sqlite("~/your/working/directory/pfx_db.sqlite3")
src_tbls(pfx_16)
```

pitchRx

- Once we set up a PITCHf/x database, we have access to all MLB pitch and gameplay data
- Best to use a small date range for initial setup
- 10 primary tables in the data

PITCHf/x Data Tables

Table Name	Description
`action`	Ball/strike count, result of pitch, ...
`atbat`	Pitcher/batter names, handedness, heights, at bat result, ...
`coach`	Names of manager and staff, ...
`game`	Venue, start time, time zone, TV, win-loss records, ...
`media`	Mobile/TV media assets, ...
`pitch`	Umpire's decision/outcome, strike zone parameters, x-y coordinates, ...
`player`	Players' stats, position, number, ...
`po`	Details about put out attempts (e.g., pickoffs and stolen bases), ...
`runner`	Details about base runner(s) and at bat events, ...
`umpire`	Umpire names and positions, ...

PITCHf/x Data Tables

For most analyses, we usually work with:

Table Name	Description
`action`	Ball/strike count, result of pitch, ...
`atbat`	Pitcher/batter names, handedness, heights, at bat result, ...
`coach`	Names of manager and staff, ...
`game`	Venue, start time, time zone, TV, win-loss records, ...
`media`	Mobile/TV media assets, ...
`pitch`	Umpire's decision/outcome, strike zone parameters, x-y coordinates, ...
`player`	Players' stats, position, number, ...
`po`	Details about put out attempts (e.g., pickoffs and stolen bases), ...
`runner`	Details about base runner(s) and at bat events, ...
`umpire`	Umpire names and positions, ...

dplyr

- Wickham and Francois (2016)
- A grammar of data manipulation
- Provides a set of verbs for lots of tasks
  - `select()`: Selects columns
  - `filter()`: Filters rows (e.g., ==, !=, <=, etc.)
  - `arrange()`: Re-orders and sorts rows
  - `mutate()`: Creates new variables/columns
  - `summarise()`: Summarizes values/output
  - `group_by()`: Allows for by-group operations

Accessing MLB Pitch Tracking Data

- R packages:
  1. `pitchRx`: Data collection
  2. `dplyr`: Data analysis

Analyzing MLB Pitch Tracking Data

Tonight

- There are several ways to analyze PITCHf/x data
  - Ex.: Pitching/batting outcomes, predictive models, et al.
- Tonight though, let's concentrate on home plate umpire decisions
- Specifically:
  1. How many pitches do umpires see during games? Of those, how many require a decision?
  2. How accurate are umpires given pitch count and location?
  3. How accurate are *all* umpires over the season? How accurate are *individual* umpires over the season?

1. Pitches Seen vs. Decisions Made

- How many pitches do umpires see during games? Of those, how many require an umpire decision?
  - **Pitches seen:** Total number of recorded pitches thrown during game
  - **Decisions made:** Total number of called strikes and called balls during game

1. Pitches Seen vs. Decisions Made

- We'll use the `pitch` table to answer these questions
- Steps:
  1. Create data frame for pitches seen, `observed`
  2. Create data frame for decisions made, `decisions`
  3. Join `observed` and `decisions`
  4. Calculate proportion of pitches requiring decision
  5. Calculate simple descriptive statistics

1. Pitches Seen vs. Decisions Made

- Step 1. Create data frame for pitches seen

```
observed <- pitch %>%
  group_by(gameday_link) %>%
  summarize(seen = n())
```

1. Pitches Seen vs. Decisions Made

- Step 1. Create data frame for pitches seen
  - `pitch`: Current data frame

```
observed <- pitch %>%
  group_by(gameday_link) %>%
  summarize(seen = n())
```

1. Pitches Seen vs. Decisions Made

- Step 1. Create data frame for pitches seen
  - `pitch`: Current data frame
  - `group_by()`, `summarize()`, `n()`: `dplyr` verbs

```
observed <- pitch %>%
  group_by(gameday_link) %>%
  summarize(seen = n())
```

1. Pitches Seen vs. Decisions Made

- Step 1. Create data frame for pitches seen
  - `pitch`: Current data frame
  - `group_by()`, `summarize()`, `n()`: `dplyr` verbs
  - `gameday_link`: Unique date/team label

```
observed <- pitch %>%
  group_by(gameday_link) %>%
  summarize(seen = n())
```

1. Pitches Seen vs. Decisions Made

- Step 1. Create data frame for pitches seen
  - `pitch`: Current data frame
  - `group_by()`, `summarize()`, `n()`: `dplyr` verbs
  - `gameday_link`: Unique date/team label
  - `seen`: New name for variable `n()`

```
observed <- pitch %>%
  group_by(gameday_link) %>%
  summarize(seen = n())
```

1. Pitches Seen vs. Decisions Made

- Step 1. Create data frame for pitches seen
- R code:
  - `pitch`: Current data frame
  - `group_by()`, `summarize()`, `n()`: `dplyr` verbs
  - `gameday_link`: Unique date/team label
  - `seen`: New name for variable `n()`
  - `observed`: Name of new data frame

```
observed <- pitch %>%
  group_by(gameday_link) %>%
  summarize(seen = n())
```

1. Pitches Seen vs. Decisions Made

- Step 1. Create data frame for pitches seen

```
(observed <- pitch %>%
  group_by(gameday_link) %>%
  summarize(seen = n()))
```

```
## # A tibble: 2,468 × 2
##                        gameday_link  seen
##                               <chr> <int>
## 1  gid_2016_04_03_chnmlb_anamlb_1   252
## 2  gid_2016_04_03_nynmlb_kcamlb_1   291
## 3  gid_2016_04_03_slnmlb_pitmlb_1   285
## 4  gid_2016_04_03_tormlb_tbamlb_1   276
## 5  gid_2016_04_04_chamlb_oakmlb_1   292
## 6  gid_2016_04_04_chnmlb_anamlb_1   297
## 7  gid_2016_04_04_colmlb_arimlb_1   362
## 8  gid_2016_04_04_lanmlb_sdnmlb_1   319
## 9  gid_2016_04_04_minmlb_balmlb_1   278
## 10 gid_2016_04_04_phimlb_cinmlb_1   267
## # ... with 2,458 more rows
```

1. Pitches Seen vs. Decisions Made

- Step 2. Create data frame for decisions made
- We need to omit all pitches/outcomes except for called strikes and called balls

```
decisions <- pitch %>%
  group_by(gameday_link) %>%
  filter(des == "Called Strike" | des == "Ball") %>%
  summarize(decisions = n())
```

1. Pitches Seen vs. Decisions Made

- Step 2. Create data frame for decisions made
- We need to omit all pitches/outcomes except for called strikes and called balls
- R code:
  - `filter()`: Returns rows with matching conditions

```
decisions <- pitch %>%
  group_by(gameday_link) %>%
  filter(des == "Called Strike" | des == "Ball") %>%
  summarize(decisions = n())
```

1. Pitches Seen vs. Decisions Made

- Step 2. Create data frame for decisions made

```
(decisions <- pitch %>%
  group_by(gameday_link) %>%
  filter(des == "Called Strike" | des == "Ball") %>%
  summarize(decisions = n()))
```

```
## # A tibble: 2,468 × 2
##                        gameday_link decisions
##                               <chr>     <int>
## 1  gid_2016_04_03_chnmlb_anamlb_1       124
## 2  gid_2016_04_03_nynmlb_kcamlb_1       150
## 3  gid_2016_04_03_slnmlb_pitmlb_1       145
## 4  gid_2016_04_03_tormlb_tbamlb_1       136
## 5  gid_2016_04_04_chamlb_oakmlb_1       153
## 6  gid_2016_04_04_chnmlb_anamlb_1       135
## 7  gid_2016_04_04_colmlb_arimlb_1       158
## 8  gid_2016_04_04_lanmlb_sdnmlb_1       165
## 9  gid_2016_04_04_minmlb_balmlb_1       136
## 10 gid_2016_04_04_phimlb_cinmlb_1       123
```

1. Pitches Seen vs. Decisions Made

- Step 3. Join `observed` and `decisions` by `gameday_link`

```
pitches <- inner_join(observed, decisions, by = "gameday_link")
```

1. Pitches Seen vs. Decisions Made

- Step 3. Join `observed` and `decisions` by `gameday_link`
- R code:
  - `inner_join()`: Returns observations that match in both x and y

```
pitches <- inner_join(observed, decisions, by = "gameday_link")
```

1. Pitches Seen vs. Decisions Made

- Step 3. Join `observed` and `decisions` by `gameday_link`

```
(pitches <- inner_join(observed, decisions, by = "gameday_link"))
```

```
## # A tibble: 2,468 × 3
##                       gameday_link  seen decisions
##                              <chr> <int>     <int>
## 1  gid_2016_04_03_chnmlb_anamlb_1   252       124
## 2  gid_2016_04_03_nynmlb_kcamlb_1   291       150
## 3  gid_2016_04_03_slnmlb_pitmlb_1   285       145
## 4  gid_2016_04_03_tormlb_tbamlb_1   276       136
## 5  gid_2016_04_04_chamlb_oakmlb_1   292       153
## 6  gid_2016_04_04_chnmlb_anamlb_1   297       135
## 7  gid_2016_04_04_colmlb_arimlb_1   362       158
## 8  gid_2016_04_04_lanmlb_sdnmlb_1   319       165
## 9  gid_2016_04_04_minmlb_balmlb_1   278       136
## 10 gid_2016_04_04_phimlb_cinmlb_1   267       123
## # ... with 2,458 more rows
```

1. Pitches Seen vs. Decisions Made

- Step 4. Calculate proportion of pitches requiring decision

```
pitches <- pitches %>%
  mutate(prop = decisions/seen)
```

1. Pitches Seen vs. Decisions Made

- Step 4. Calculate proportion of pitches requiring decision
- R code: `mutate()`: Adds new variable

```
pitches <- pitches %>%
  mutate(prop = decisions/seen)
```

1. Pitches Seen vs. Decisions Made

- Step 4. Calculate proportion of pitches requiring decision

```
(pitches <- pitches %>%
  mutate(prop = decisions/seen))
```

```
## # A tibble: 2,468 × 4
##                        gameday_link  seen decisions      prop
##                               <chr> <int>     <int>     <dbl>
## 1  gid_2016_04_03_chnmlb_anamlb_1   252       124 0.4920635
## 2  gid_2016_04_03_nynmlb_kcamlb_1   291       150 0.5154639
## 3  gid_2016_04_03_slnmlb_pitmlb_1   285       145 0.5087719
## 4  gid_2016_04_03_tormlb_tbamlb_1   276       136 0.4927536
## 5  gid_2016_04_04_chamlb_oakmlb_1   292       153 0.5239726
## 6  gid_2016_04_04_chnmlb_anamlb_1   297       135 0.4545455
## 7  gid_2016_04_04_colmlb_arimlb_1   362       158 0.4364641
## 8  gid_2016_04_04_lanmlb_sdnmlb_1   319       165 0.5172414
## 9  gid_2016_04_04_minmlb_balmlb_1   278       136 0.4892086
## 10 gid_2016_04_04_phimlb_cinmlb_1   267       123 0.4606742
## # ... with 2,458 more rows
```

1. Pitches Seen vs. Decisions Made

- Step 5. Calculate simple descriptive statistics

```
(pitch_summs <- pitches %>%
  summarize(m_pitches = mean(seen),
    sd_pitches = sd(seen),
    m_calls = mean(decisions),
    sd_calls = sd(decisions),
    m_prop = mean(prop),
    sd_prop = sd(prop)))
```

```
## # A tibble: 1 × 6
##   m_pitches sd_pitches  m_calls sd_calls    m_prop    sd_prop
##       <dbl>      <dbl>    <dbl>    <dbl>     <dbl>      <dbl>
## 1   294.485   40.32311 148.1896  22.9948 0.5028475 0.03240739
```

Contextual Umpire Accuracy

- Text

Cumulative Umpire Accuracy

- Text

Questions?

Contact Details

Aaron R. Baggett, Ph.D.

Assistant Professor of Psychology
University of Mary Hardin-Baylor
abaggett@umhb.edu
(254) 295-4553
@aaron_baggett