

# 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](https://bit.ly/austin_r)

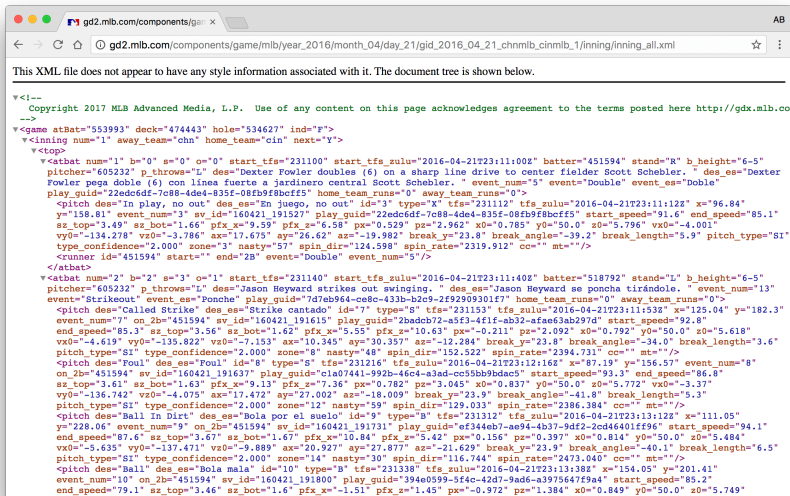
# Major League Baseball (MLB) Pitch Tracking Data

# 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/>



```
<!--
Copyright 2017 MLB Advanced Media, L.P. Use of any content on this page acknowledges agreement to the terms posted here http://gdx.mlb.com
-->
<game atBat="553993" deck="474443" hole="534627" ind="F">
  <inning num="1" away_team="chn" home_team="cin" next="Y">
    <top>
      <catbat num="1" b="0" s="0" o="0" start_tfs="231100" start_tfs_zulu="2016-04-21T23:11:00Z" batter="451594" stand="R" b_height="6-5"
        pitcher="605232" p_throws="L" des="Dexter Fowler doubles (6) on a sharp line drive to center fielder Scott Schebler. " des_es="Dexter
        Fowler pega doble (6) con línea fuerte a jardinero central Scott Schebler. " event_num="5" event="Double" event_es="Doble"
        play_guid="22edc6df-7c88-4de4-835f-08fb9f8bcff5" home_team_runs="0" away_team_runs="0">
        <pitch des="In play, no out" des_es="En juego, no out" id="3" type="X" tfs="231112" tfs_zulu="2016-04-21T23:11:12Z" x="96.84"
          y="158.81" event_num="3" sv_id="160421_191527" play_guid="22edc6df-7c88-4de4-835f-08fb9f8bcff5" start_speed="91.6" end_speed="85.1"
          sz_top="3.49" sz_bot="1.66" pfx_x="9.59" pfx_z="6.58" px="0.529" pz="2.962" x0="0.785" y0="50.0" z0="5.796" vx0="4.001"
          vy0="134.278" vz0="3.786" ax="17.675" ay="26.62" az="-19.982" break_y="23.8" break_angle="-39.2" break_length="5.9" pitch_type="S1"
          type_confidence="2.000" zone="3" nasty="57" spin_dir="124.598" spin_rate="2319.912" cc="" mt="" />
        <runner id="451594" start="3" end="ZB" event="Double" event_num="5"/>
      </catbat>
      <catbat num="2" b="2" s="3" o="1" start_tfs="231140" start_tfs_zulu="2016-04-21T23:11:40Z" batter="518792" stand="L" b_height="6-5"
        pitcher="605232" p_throws="L" des="Jason Heyward strikes out swinging. " des_es="Jason Heyward se poncha tirándole. " event_num="13"
        event="Strikeout" event_es="Ponche" play_guid="7d7eb964-ce8c-433b-b2c9-2f92909301f7" home_team_runs="0" away_team_runs="0">
        <pitch des="Called Strike" des_es="Strike cantado" id="7" type="S" tfs="231153" tfs_zulu="2016-04-21T23:11:53Z" x="125.04" y="182.3"
          event_num="7" on_2b="451594" sv_id="160421_191615" play_guid="2badcb72-a5f3-4f1f-ab32-afae63ab297d" start_speed="92.8"
          end_speed="85.3" sz_top="3.56" sz_bot="1.62" pfx_x="5.55" pfx_z="10.63" px="-0.211" pz="2.092" x0="0.792" y0="50.0" z0="5.618"
          vx0="4.619" vy0="135.822" vz0="-7.153" ax="10.345" ay="30.357" az="-12.284" break_y="23.8" break_angle="-34.0" break_length="3.6"
          pitch_type="S1" type_confidence="2.000" zone="8" nasty="48" spin_dir="152.522" spin_rate="2394.731" cc="" mt="" />
        <pitch des="Foul" des_es="Foul" id="8" type="S" tfs="231216" tfs_zulu="2016-04-21T23:12:16Z" x="87.19" y="156.57" event_num="8"
          on_2b="451594" sv_id="160421_191637" play_guid="c1a07441-992b-46c4-a3ad-cc35b9bdac3" start_speed="93.3" end_speed="86.8"
          sz_top="3.61" sz_bot="1.63" pfx_x="7.36" pfx_z="10.782" px="0.837" pz="3.045" x0="0.837" y0="50.0" z0="5.772" vx0="3.37"
          vy0="136.742" vz0="-4.075" ax="17.472" ay="27.002" az="-18.009" break_y="23.9" break_angle="-41.8" break_length="5.3"
          pitch_type="S1" type_confidence="2.000" zone="12" nasty="59" spin_dir="129.033" spin_rate="2386.384" cc="" mt="" />
        <pitch des="Ball In Dirt" des_es="Bola por el suelo" id="9" type="B" tfs="231312" tfs_zulu="2016-04-21T23:13:12Z" x="111.05"
          y="228.06" event_num="9" on_2b="451594" sv_id="160421_191731" play_guid="ef344eb7-ae94-4b37-9df2-2cd46401ff96" start_speed="94.1"
          end_speed="87.6" sz_top="3.67" sz_bot="1.67" pfx_x="10.84" pfx_z="5.42" px="0.156" pz="0.397" x0="0.814" y0="50.0" z0="5.484"
          vx0="5.635" vy0="137.471" vz0="-9.889" ax="20.927" ay="27.877" az="-21.629" break_y="23.9" break_angle="-40.1" break_length="6.5"
          pitch_type="S1" type_confidence="2.000" zone="14" nasty="30" spin_dir="116.744" spin_rate="2473.040" cc="" mt="" />
        <pitch des="Ball" des_es="Bola mala" id="10" type="B" tfs="231338" tfs_zulu="2016-04-21T23:13:38Z" x="154.05" y="201.41"
          event_num="10" on_2b="451594" sv_id="160421_191800" play_guid="394a0599-3f4c-42d7-9ad6-a39796479a4" start_speed="85.2"
          end_speed="79.1" sz_top="3.46" sz_bot="1.6" pfx_x="1.51" pfx_z="1.45" px="0.972" pz="1.384" x0="0.849" y0="50.0" z0="5.749"
          vx0="3.883" vy0="134.883" vz0="0.814" ax="1.883" ay="1.883" az="1.883" break_y="1.883" break_angle="1.883" break_length="1.883"
          pitch_type="S1" type_confidence="2.000" zone="14" nasty="30" spin_dir="116.744" spin_rate="2473.040" cc="" mt="" />
    </top>
  </inning>
</game>
```

## 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



# 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 *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
- ▶ 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_bamlb_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_bamlb_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_slmlb_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_bamlb_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_slmlb_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_bamlb_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
```

# Umpire Accuracy

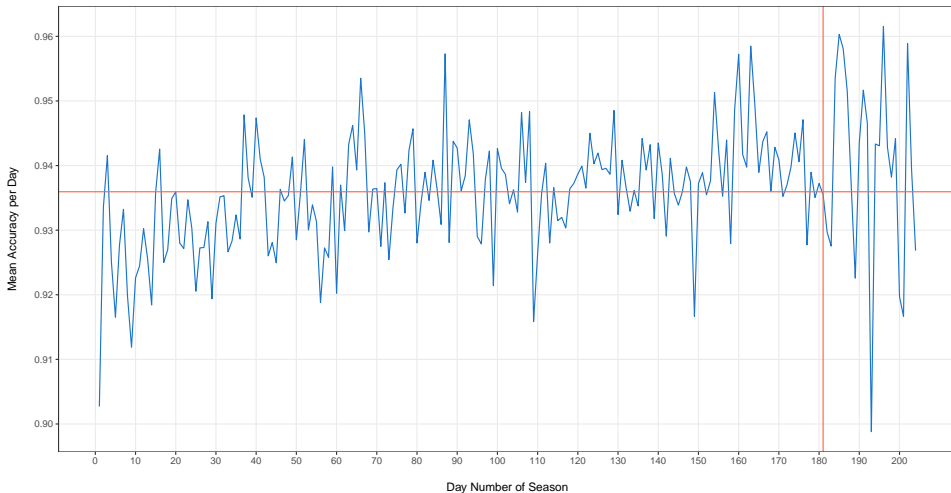
- ▶ Check out `umpire_accuracy.R` in my GitHub Repo for this talk
- ▶ Overall, umpires are quite accurate

<i>M</i>	<i>SD</i>	<i>SEM</i>	95% CI
0.94	0.24	0.05	[0.84, 0.98]



# Umpire Accuracy

- Here's a plot of the cumulative accuracy for MLB umpires over the season




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