CMSC320 Final Project

Eric

April 10, 2018

We will be analyzing how effective the video game Out of the Park (OOTP) is at simulating a baseball season. To do this, I have ran a simulation in the game from the years 2000 to 2018. The game has a built-in feature where it will export many of it's important data files to a CSV file with headers on top, allowing for us to easily load the game data into an R dataframe. We will use the Lahman sqlite database as the point of comparison.

First, we will load the CSV files and connect to the Lahman database.

```r
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 3.4.4

## Loading required package: xml2
```

```r
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------

## v ggplot2 2.2.1     v purrr   0.2.4
## v tibble  1.4.2     v dplyr   0.7.4
## v tidyr   0.8.0     v stringr 1.2.0
## v readr   1.1.1     v forcats 0.2.0

## -- Conflicts ----------------------------------------------------------------------------
## x dplyr::filter()         masks stats::filter()
## x readr::guess_encoding() masks rvest::guess_encoding()
## x dplyr::lag()            masks stats::lag()
## x purrr::pluck()          masks rvest::pluck()
```

```r
coaches <- read.csv('csv/coaches.csv')
divisions <- read.csv('csv/divisions.csv')
leagues <- read.csv('csv/leagues.csv')
allstars <- read.csv('csv/league_history_all_star.csv')
players <- read.csv('csv/players.csv')
awards <- read.csv('csv/players_awards.csv')
batting_stats <- read.csv('csv/players_career_batting_stats.csv')
pitching_stats <- read.csv('csv/players_career_pitching_stats.csv')
teams <- read.csv('csv/teams.csv')
team_history_record <- read.csv('csv/team_history_record.csv')
team_history_financials <- read.csv('csv/team_history_financials.csv')

lahman <- DBI::dbConnect(RSQLite::SQLite(), "lahman2016.sqlite")
```

Now we can do some data anlysis. First, let's see how the simulated hit total compares to the real-life values. We will look at the residual for each year for each player with atleast 50 hits in both the simulation and real-life for a given season. This will eliminated players that did not play as many games as expected and also removes pitchers.

```sql
select nameFirst as first_name, nameLast as last_name, H as hits, yearID as year_id
from MASTER, Batting
where MASTER.playerID = Batting.playerID and yearID >= 2000 and H > 50
group by nameFirst, nameLast, yearID
```

```r
library(ggplot2)

hits_by_year <- players %>% inner_join(batting_stats, by = 'player_id') %>% filter(split_id == 1,h > 50)

hits_difference <- hits_by_year %>% inner_join(lahman_hits, on = c('first_name', 'last_name', 'year_id')
```

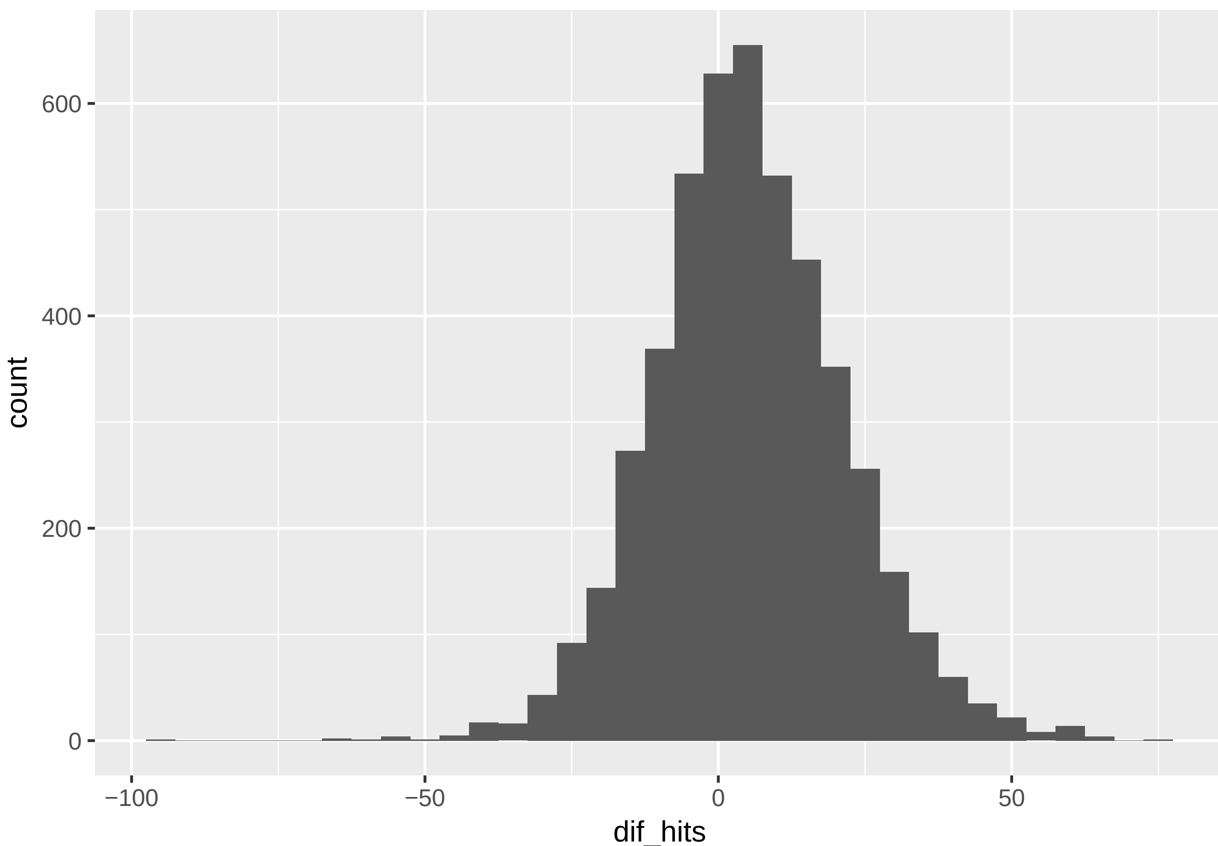
```
## Joining, by = c("first_name", "last_name", "year_id")

## Warning: Column `first_name` joining factor and character vector, coercing
## into character vector

## Warning: Column `last_name` joining factor and character vector, coercing
## into character vector
```

```r
hits_difference['dif_hits'] <- hits_difference['sim_hits'] - hits_difference['hits']

hits_difference %>% ggplot(mapping=aes(x=dif_hits)) + geom_histogram(binwidth = 5)
```



Looking at the above histogram of the residuals, the simulation is quite good at simulating how many hits a player will get. The histogram is slightly skewed to the right, which indicates that was a slight tendency to simulate more hits than a player actually got.

Next lets look at how well the game simulated each season by computing their average positioning in their division

```sql
select avg(Rank) as real_pos, name, teamID, franchID
from Teams
```

```
where yearID >= 2000
group by teamID
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.4.4

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
sim_pos <- team_history_record %>% inner_join(teams, by = 'team_id') %>% select(team_id,historical_id, p

diff_pos <- sim_pos %>% inner_join(lahman_pos, by = c("historical_id"="teamID"))
```

```
## Warning: Column `historical_id`/`teamID` joining factor and character
## vector, coercing into character vector
```

```r
diff_pos['diff'] <- - diff_pos$sim_pos + diff_pos$real_pos

diff_pos %>% select(diff,historical_id) %>% arrange(desc(diff))
```

```
## # A tibble: 30 x 2
##      diff historical_id
##     <dbl> <chr>
##  1 1.72   MIA
##  2 0.859  PIT
##  3 0.794  TOR
##  4 0.755  CHN
##  5 0.716  COL
##  6 0.484  CLE
##  7 0.268  KCA
##  8 0.252  BOS
##  9 0.206  TEX
## 10 0.0915 BAL
## # ... with 20 more rows
```

```r
diff_pos %>% summarize(sd=sd(diff), mean=mean(diff))
```

```
## # A tibble: 1 x 2
##      sd     mean
##   <dbl>    <dbl>
## 1 0.563 -0.00120
```

Above we have the differences between divisional placings. We can see that MIA was the best performer relative to their actual standings and the Yankees were the worst. MIA is a considerable outlier here. With a mean of essentially zero, MIA is over 3 SD away. No other team is over 2 SD away!