R Exercise: Simulating the 2020 MLB Regular Season and Postseason

Jake Singleton

10/07/2020

Introduction

In this notebook, I simulate the 2020 MLB regular season and postseason using the Bradley-Terry model discussed in Chapter 9 of *Analyzing Baseball Data with R* by Max Marchi, Jim Albert, and Benjamin Baumer. I use Pythagorean Win Percentage as an estimate of teams' true talent levels, which, while certainly overly-simplimistic, builds on the foundation presented in the book's chapter and serves as a good starting point for a projection system.

The Math

The math is very quick, and consequently the majority of this report will consist of code. But here's how the model works: suppose Team A is playing Team B in a single game. Then, by the Bradley-Terry model, the probability that Team A wins is:

$P(\text{A wins}) = \frac{exp(T_A)}{exp(T_A)+exp(T_B)}$, where T_A and T_B are the true talent estimates for Team A and Team B respectively. Naturally, we have to decide how to assign a talent to each of the 30 MLB teams, and this is where I've decided to use each team's 2020 Pythagorean Win%. I made this choice because it is a better predictor of a team's W-L record than Win% itself and is easy to find/work with. Once we have each team's talent, it is straightforward to calculate a given team's probability of winning a game.

More complex and better models might use an ELO, Mixed Models, or SRS approach, but that's a project for another time.

The Code

Given we know how to find the relevant probabilities, the main tasks in the code involve the following:

1. Making the 60 game schedule
2. Organizing the teams' talents
3. Playing out a full regular season using the appropriate probabilities, representing each game as the flip of a loaded coin to determine the winner
4. Using the results of the regular season to determine playoff teams, and then simulating the 2020 postseason
5. Repeating the above steps many times in order to be able to compare our model's results to what has actually happened, and to project what will happen in the remainder of the postseason

Most of the code below walks through the above steps in order to simulate a single season, and I'm going to hide it except for a few outputs (standings and a few sample data frames) since it's pretty long. Once we simulate one season, we write a function that simulates one whole season for us, and then we call this function 1,000 times to simulate 1,000 seasons. We then analyze our results graphically!

Note that you can find the R Markdown file and season-simulating R script in their full forms in the code directory of this repository.

```r
library(tidyverse)
library(baseballr)
```

```
## $`NL Overall_up to_2020-09-28`
##     Tm  W  L  W-L%   GB  RS  RA pythW-L%
## 1  LAD 43 17 0.717   -- 349 213    0.712
## 2  SDP 37 23 0.617  6.0 325 241    0.633
## 3  ATL 35 25 0.583  8.0 348 288    0.586
## 4  CHC 34 26 0.567  9.0 265 240    0.545
## 5  MIA 31 29 0.517 12.0 263 304    0.434
## 6  STL 30 28 0.517 12.0 240 229    0.521
## 7  CIN 31 29 0.517 12.0 243 243    0.500
## 8  MIL 29 31 0.483 14.0 247 264    0.470
## 9  SFG 29 31 0.483 14.0 299 297    0.503
## 10 PHI 28 32 0.467 15.0 306 311    0.493
## 11 COL 26 34 0.433 17.0 275 353    0.388
## 12 NYM 26 34 0.433 17.0 286 308    0.466
## 13 WSN 26 34 0.433 17.0 293 301    0.488
## 14 ARI 25 35 0.417 18.0 269 295    0.458
## 15 PIT 19 41 0.317 24.0 219 298    0.363

## $`AL Overall_up to_2020-09-28`
##     Tm  W  L  W-L%   GB  RS  RA pythW-L%
## 1  TBR 40 20 0.667   -- 289 229    0.605
## 2  MIN 36 24 0.600  4.0 269 215    0.601
## 3  OAK 36 24 0.600  4.0 274 232    0.576
## 4  CHW 35 25 0.583  5.0 306 246    0.599
## 5  CLE 35 25 0.583  5.0 248 209    0.578
## 6  NYY 33 27 0.550  7.0 315 270    0.570
## 7  TOR 32 28 0.533  8.0 302 312    0.485
## 8  HOU 29 31 0.483 11.0 279 275    0.507
## 9  SEA 27 33 0.450 13.0 254 303    0.420
## 10 LAA 26 34 0.433 14.0 294 321    0.460
## 11 KCR 26 34 0.433 14.0 248 272    0.458
## 12 BAL 25 35 0.417 15.0 274 294    0.468
## 13 BOS 24 36 0.400 16.0 292 351    0.417
## 14 DET 23 35 0.397 16.0 249 318    0.390
## 15 TEX 22 38 0.367 18.0 224 312    0.353

## # A tibble: 6 x 9
##   Home  Visitor League.x Division.x Talent.Home League.y Division.y
##   <chr> <chr>      <dbl>      <dbl>       <dbl>    <dbl>      <dbl>
## 1 LAD   SFG            1          3      2.46          1          3
## 2 LAD   COL            1          3      2.46          1          3
## 3 LAD   ARZ            1          3      2.46          1          3
## 4 LAD   SDP            1          3      2.46          1          3
## 5 SFG   LAD            1          3      0.0168        1          3
## 6 SFG   COL            1          3      0.0168        1          3
## # ... with 2 more variables: Talent.Visitor <dbl>, prob_Home <dbl>
```

A portion of the 2020 regular season schedule data frame that includes team talents and probabilities is above.

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 5
##   Team   Seed League outcome[,1] Winner.WS[,1]
##   <chr> <dbl>  <dbl>       <int>         <dbl>
## 1 LAD       1      1           7             1
## 2 MIN       4      2           0             0
```

As we see in the sample data frame above, the Dodgers swept the World Series from Minnesota.

```
# Step 5: Simulate 2020 season 1,000 times
source("one_simulation_20.R")
n_seasons = 1000
Many.Results = suppressMessages(map_df(rep("arg", n_seasons), one_simulation_20))
head(Many.Results)
```

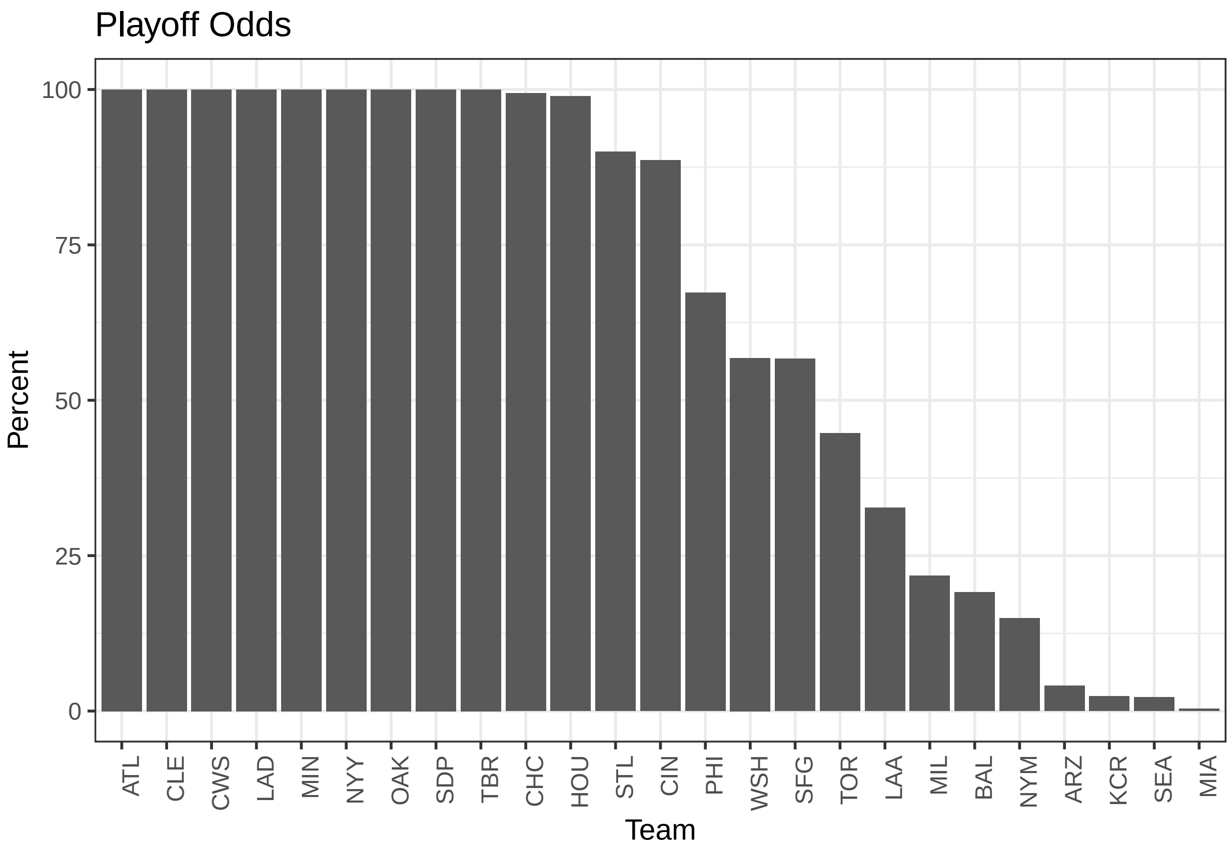
```
## # A tibble: 6 x 13
##   winner Wins.x League.x Division.x Talent.x Winner.Div.x RunnerUp.Div.x  WC.x
##   <chr>   <int>    <dbl>      <dbl>    <dbl>        <dbl>          <dbl> <dbl>
## 1 LAD        56        1          3   2.46              1              0     0
## 2 SDP        46        1          3   1.54              0              1     0
## 3 ARZ        27        1          3  -0.510             0              0     1
## 4 SFG        22        1          3   0.0168            0              0     0
## 5 COL        14        1          3  -1.33              0              0     0
## 6 CHC        39        1          4   0.509             1              0     0
## # ... with 5 more variables: Seed.x <dbl>, Winner.WC.x[,1] <dbl>,
## #   Winner.DS.x[,1] <dbl>, Winner.CS.x[,1] <dbl>, Winner.WS[,1] <dbl>
```

The data frame `Many.Results` contains 30,000 rows = 30 teams * 1,000 seasons and includes the results for each team for each season.

Analysis

```
# Proportion of times making the playoffs for each team
playoff_props = Many.Results %>%
  filter(Winner.Div.x == 1 | RunnerUp.Div.x == 1 | WC.x == 1) %>%
  group_by(winner) %>%
  summarize(Percent = n() / n_seasons * 100)

ggplot(playoff_props, aes(reorder(winner, -Percent), Percent)) +
  geom_bar(stat = "identity") +
  xlab("Team") +
  ggtitle("Playoff Odds") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
ggsave("playoff_odds.png")
```

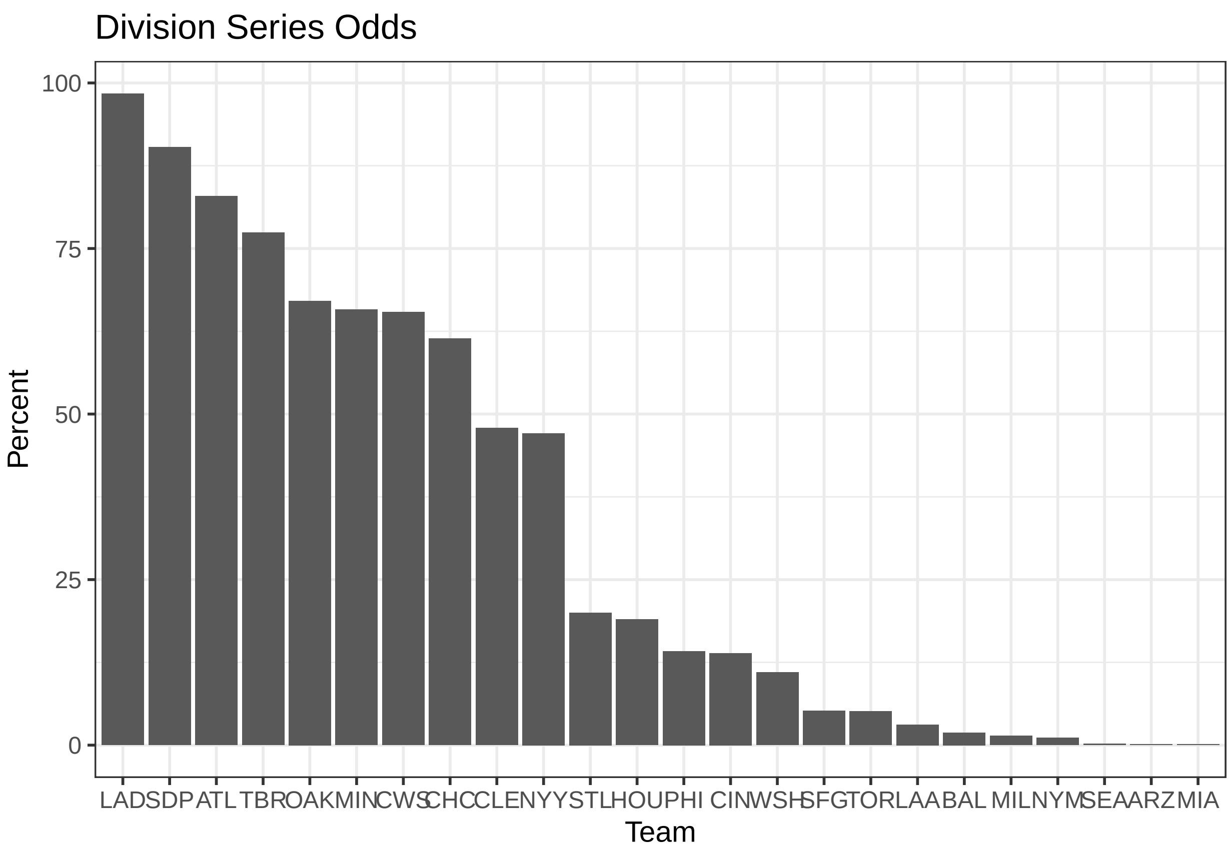
This bar chart shows that our model predicted Atlanta, Cleveland, CWS, LAD, Minnesota, NYY, Oakland, San Diego, and Tampa Bay (9 teams) to make the playoffs 100% of the time (an overestimation, which will be discussed in the conclusion). Next come CHC and Houston at just under 100%, Saint Louis and Cincinnati around 90% (the first substantial dropoff) and then Philadelphia, Washington, and San Francisco rounding out the top 16 teams. It is nice to see Philadelphia and San Francisco in their respective positions, as they came just one win away from making the postseason. As for Washington, well, they underperformed their Pythagorean win expectation by 3 games, tied for most in the league, so it makes sense their playoff odds are overestimated here. In addition, we see that our model predicted Toronto as the 17th most likely to make the playoffs at a bit under 50% odds, which is reassuring since they were one of the AL Wild Card teams. As for Milwaukee, who outperformed Pythagorean win-loss by one game and had the 2nd best record in one-run games at 11-5, seeing them at 19th here isn't too far off. The big miss is Miami, who our model predicted basically had no chance of making the postseason. In fact, Miami oveperformed their Pythagorean win expectation by a Major League-leading 5 games, in part due to having the 6th best record in one-run games in baseball at 11-8. Overall, we see that when using Pythagorean win percentage as teams' talent levels, we predicted 13/16 teams correctly for an accuracy rate of 81%.

Link to relevant stats here.

```
# Percent Chance of Making Division Series
playoff_props = Many.Results %>%
  filter(Winner.WC.x == 1) %>%
  group_by(winner) %>%
  summarize(Percent = n() / n_seasons * 100)

ggplot(playoff_props, aes(reorder(winner, -Percent), Percent)) +
```

```r
  geom_bar(stat = "identity") +
  xlab("Team") +
  ggtitle("Division Series Odds") +
  theme_bw()
```



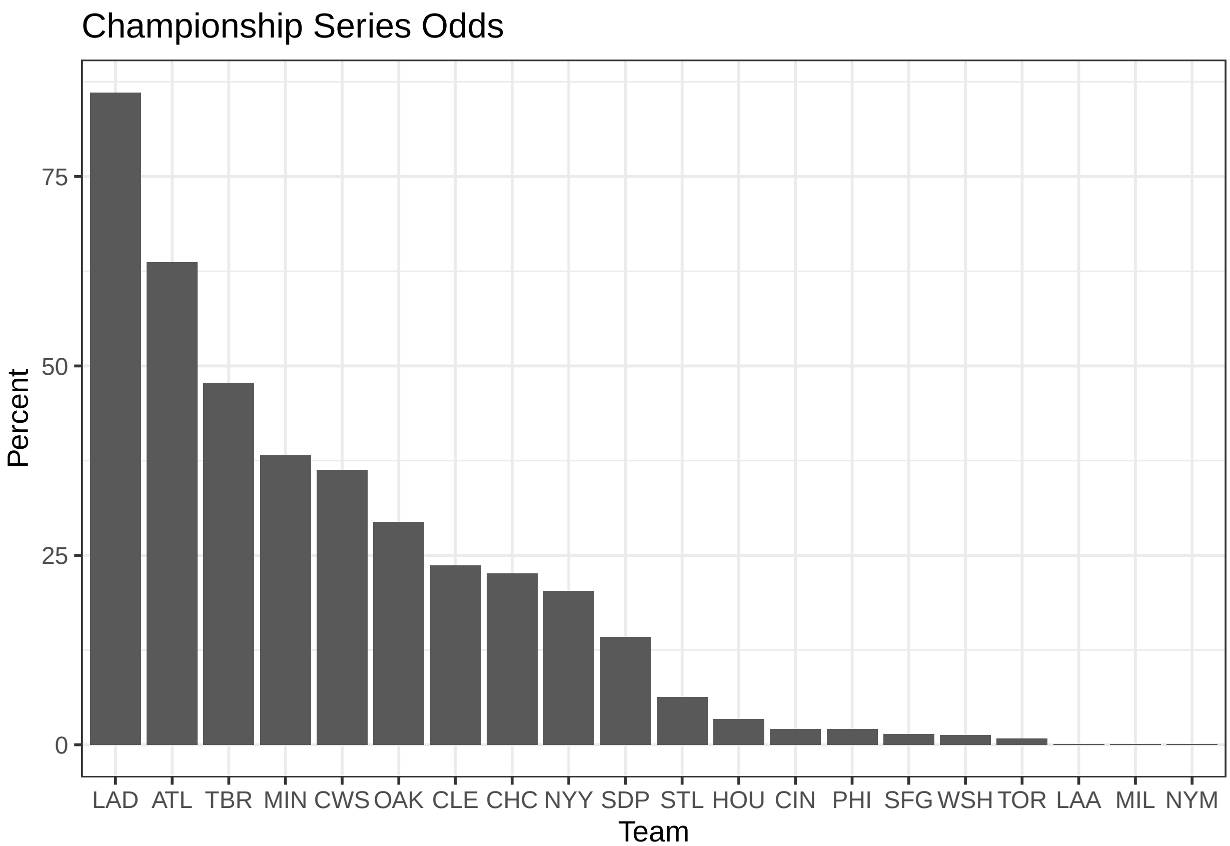

```r
ggsave("divseries_odds.png")
```

Of the 8 teams to make the Division Series this year (Tampa Bay, NYY, Houston, Oakland, LAD, San Diego, Miami, and Atlanta), our model had 5 of them in its top 8 most likely (LAD, San Diego, Atlanta, Tampa Bay, and Oakland) for a 62.5% accuracy rate. The Yankees slotted in at 9th most likely, Houston at 11th, and Miami (as expected based on the plot above) at virtually 0%. Importantly, here we can see a definite weakness of using Pythagorean Win% to estimate a team's talent level. Given the best-of-3 structure of the Wild Card round this year, there was higher-than-usual variance (i.e. a lot of variance) in who would make the Division Series. No team deserves a close-to-100% chance of making the DS, and our model putting the Dodgers in this range missed this. In the concluding section of this report, I'll talk a little bit more about why this is and how the model could be improved.

```r
# Percent Chance of Making Championship Series
playoff_props = Many.Results %>%
  filter(Winner.DS.x == 1) %>%
  group_by(winner) %>%
  summarize(Percent = n() / n_seasons * 100)

ggplot(playoff_props, aes(reorder(winner, -Percent), Percent)) +
  geom_bar(stat = "identity") +
  xlab("Team") +
```

```r
  ggtitle("Championship Series Odds") +
  theme_bw()
```



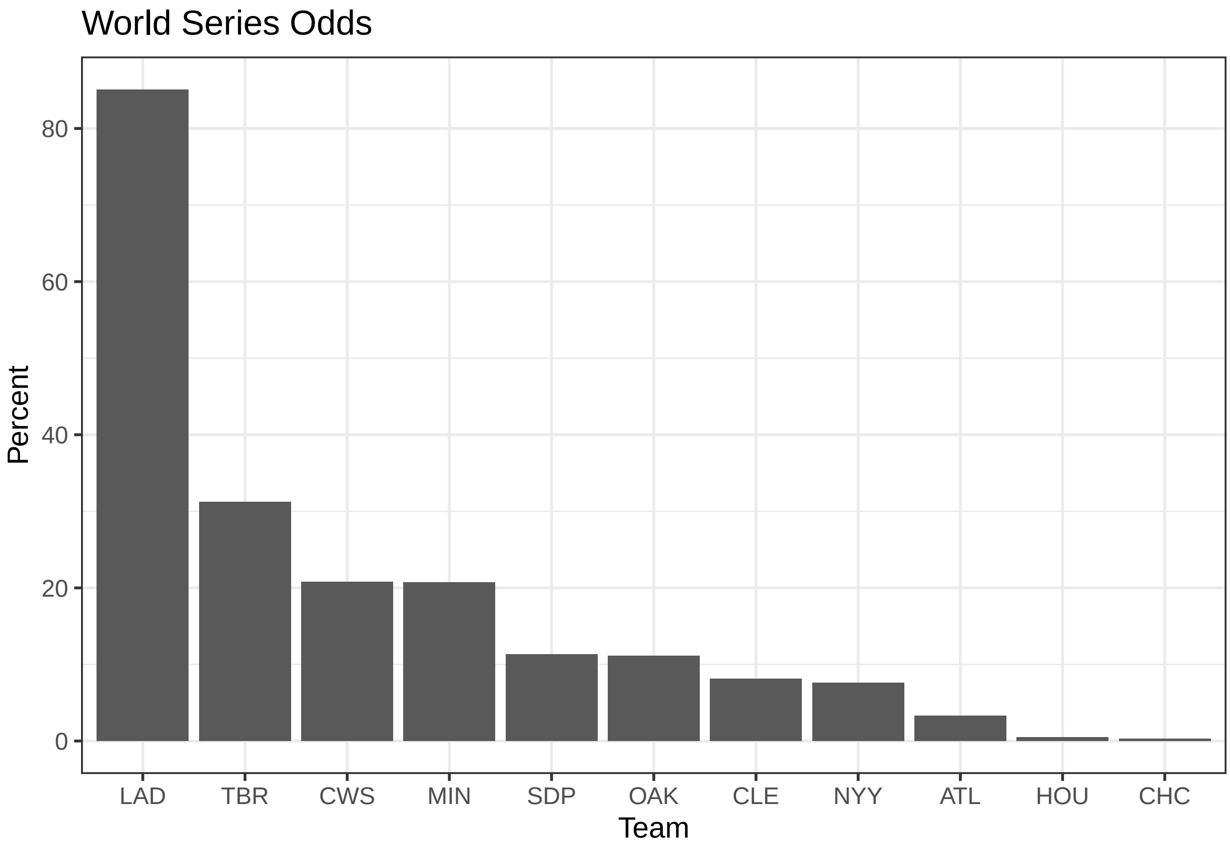

```r
ggsave("championshipseries_odds.png")
```

As of today (October 7), the four Championship Series teams have yet to be decided. However, with Houston owning a 2-0 lead over Oakland, Atlanta also owning a 2-0 lead on Miami, and LAD being up 1-0 on San Diego in a series in which they're already heavy favorites, these three are looking fairly likely. The model likes Atlanta and LAD which provides some nice confirmation, and Tampa, the 3rd most likely, is currently locked at 1-1 with NYY. Minnesota, coming off its shocking 18th straight playoff loss after being upset 2-0 by the Astros in the Wild Card round, was our model's next most popular choice. Note that San Diego, one of our model's favorite teams until this point, checks in at 10th most likely, much lower than earlier because of the playoff format: the winner of the 1 vs. 8 WC series plays the winner of the 4 vs. 5 WC series. The Dodgers were the 1 seed and Wild Card round victors 955 times, and the Padres the 4 seed and WC round victors 879 times, forcing them to face off quite often in the Division Series, in which LAD was a strong favorite and frequently the winning side.

```r
# Percent Chance of Making World Series
playoff_props = Many.Results %>%
  filter(Winner.CS.x == 1) %>%
  group_by(winner) %>%
  summarize(Percent = n() / n_seasons * 100)

ggplot(playoff_props, aes(reorder(winner, -Percent), Percent)) +
  geom_bar(stat = "identity") +
  xlab("Team") +
```

```
  ggtitle("World Series Odds") +
  theme_bw()
```

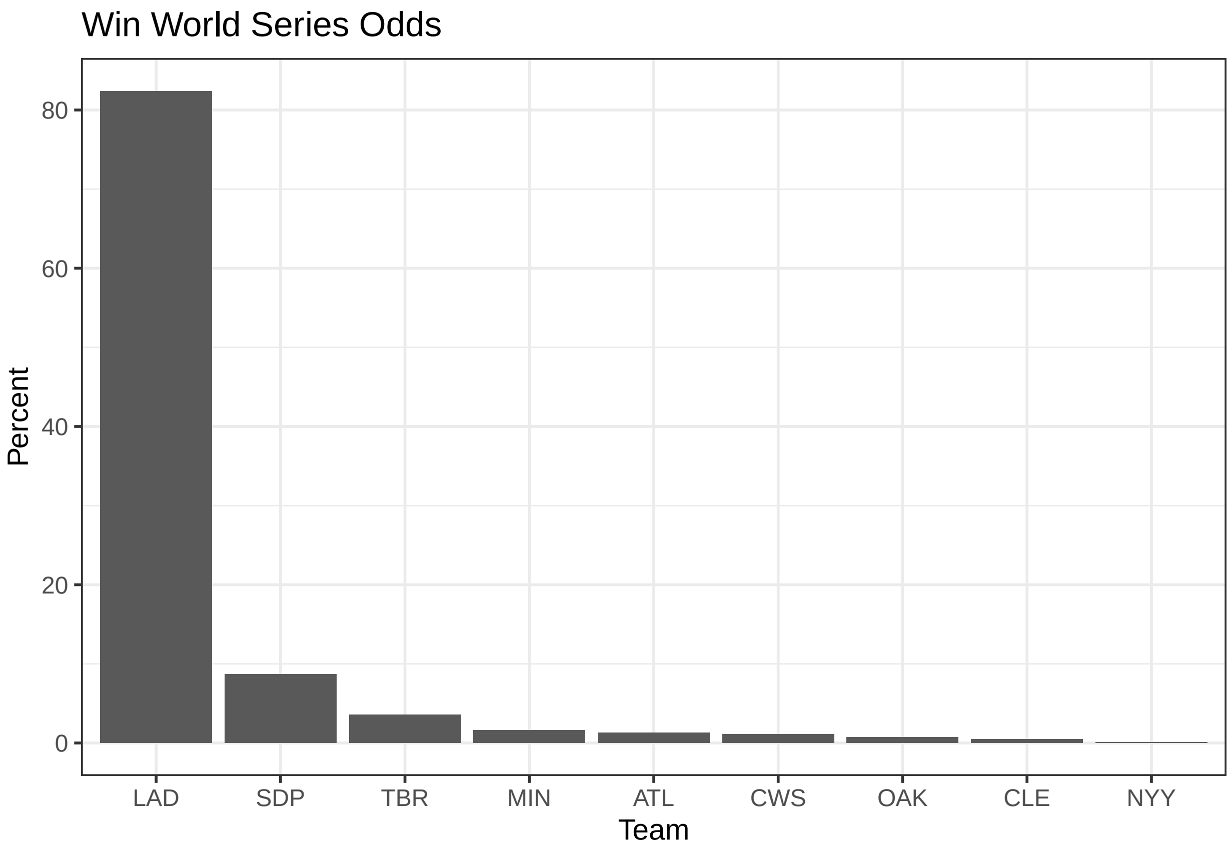



```
ggsave("worldseries_odds.png")
```

Our model loves the Dodgers in case you haven't noticed yet. They make the World Series around 85% of the time, and no other team comes close. It likes Tampa Bay, the other 1 seed, next best at 30% odds, and of the other remaining playoff teams, San Diego and Oakland are at a bit more than 10% odds, while New York, Atlanta, Houston, and Miami (not even pictured) are at 6% or lower. Note that Atlanta drops a lot here relative to the previous bar chart for the same reason the Padres did earlier: having to face the Dodgers.

```
# Percent Chance of Winning World Series
playoff_props = Many.Results %>%
  filter(Winner.WS == 1) %>%
  group_by(winner) %>%
  summarize(Percent = n() / n_seasons * 100)

ggplot(playoff_props, aes(reorder(winner, -Percent), Percent)) +
  geom_bar(stat = "identity") +
  xlab("Team") +
  ggtitle("Win World Series Odds") +
  theme_bw()
```




```
ggsave("worldserieschampion_odds.png")
```

Again, we see what we should expect: the Dodgers generally dominate, and if they don't, then San Diego is next most likely, as they were 2nd in baseball in Pythagorean win percentage.

Conclusion

What I mainly want to discuss in this section is how this model could be improved. First of all, it is evident that Pythagorean win percentage is a simple and viable way of estimating teams' true talent levels, as we saw some fairly accurate results. However, there are some key drawbacks, including:

- Lack of strength of schedule. For example, Colorado, San Francisco, and Arizona played very tough schedules this year because of the presence of LA and San Diego in their division. This issue was exacerbated for this season since West teams only played West teams, just as was the case for the Central and Eastern divisions.
- Lack of starting pitcher strength. Win probabilities for single games are heavily dependent on the starting pitcher, and this model does not account for individual players in any way.
- Lack of home-field advantage. Being the home team is advantageous in baseball. While this year without fans was different than normal, home teams still generally have an advantage simply from being more familiar with the ballpark.
- Overconfidence for certain teams despite small sample sizes and new formats. The regular season was only 60 games, and this year's postseason format is subject to high variability. It is unrealistic to predict the Dodgers to win the World Series over the rest of the field, even with their extreme regular season dominance. For a concrete example, consider how the 2019 Nationals upset the favored Dodgers in 5 games in last year's division series on the backs of aces Max Scherzer, Patrick Corbin, and Stephen Strasburg.

- Lack of consideration for strategy changes in the postseason. In the playoffs, aces might pitch on short rest, starters might come out of the bullpen, and managers may be quicker to use key bullpen arms.

A more complete model would take these variables into account.