Ryan Zimmerman Career Trajectory

John Benner

2022-05-06

Ryan Zimmerman was a true icon for the Washington Nationals for his entire career. Drafted in the first round of the first year of the National's existance in a new city, Zim swiftly became a fan favorite, reaching the Major Leagues in the same year he was drafted. During a miserable time in D.C with the new expansion team, Zimmerman was a beacon of hope, and was soon surrounded by other stars like Stephen Strasburg and Bryce Harper. Dubbed "Mr. National," Zimmerman lead the Nats to multiple division titles, including their first in 2012, and a World Series in 2019. While his career began quickly and successfully, he soon was derailed by injuries. His body could no longer support him at third base, so he switched to first base. After a few injury-ridden years, Zimmerman resurged in 2017 with a career year, hitting for his highest ever OPS in a full season. He was selected for the All-Star team and led the Nats to yet another division title. Followed by two more injury filled seasons and opting out of the 2020 season, Zimmerman had something to prove in 2021, and signed a final one-year contract with the Nationals. While he did not recreate his 2017 season, he was still a solid piece in the Nationals dreadful lineup and provided great leadership to a team full of young prospects. After a fulfilling 2021 season, Mr. National decided to hang up his cleats in the off season of 2021 and retire. With his extremely early success, injury history, and late resurgence, Zimmerman's career is a fascinating analysis study.

```r
library(tidyverse)
library(Lahman)
```

In this project, I will be using the Lahman baseball database for statistics. I will also be using the Tidyverse package suite.

```r
People %>%
  filter(nameFirst == "Ryan", nameLast == "Zimmerman") %>%
  pull(playerID) -> Zim_id
Batting %>%
  filter(playerID=="zimmery01")
```

```
##     playerID yearID stint teamID lgID   G  AB   R   H X2B X3B HR RBI SB CS BB
## 1  zimmery01   2005     1    WAS   NL  20  58   6  23  10   0  0   6  0  0  3
## 2  zimmery01   2006     1    WAS   NL 157 614  84 176  47   3 20 110 11  8 61
## 3  zimmery01   2007     1    WAS   NL 162 653  99 174  43   5 24  91  4  1 61
## 4  zimmery01   2008     1    WAS   NL 106 428  51 121  24   1 14  51  1  1 31
## 5  zimmery01   2009     1    WAS   NL 157 610 110 178  37   3 33 106  2  0 72
## 6  zimmery01   2010     1    WAS   NL 142 525  85 161  32   0 25  85  4  1 69
## 7  zimmery01   2011     1    WAS   NL 101 395  52 114  21   2 12  49  3  1 41
## 8  zimmery01   2012     1    WAS   NL 145 578  93 163  36   1 25  95  5  2 57
## 9  zimmery01   2013     1    WAS   NL 147 568  84 156  26   2 26  79  6  0 60
## 10 zimmery01   2014     1    WAS   NL  61 214  26  60  19   1  5  38  0  0 22
## 11 zimmery01   2015     1    WAS   NL  95 346  43  86  25   1 16  73  1  0 33
## 12 zimmery01   2016     1    WAS   NL 115 427  60  93  18   1 15  46  4  1 29
## 13 zimmery01   2017     1    WAS   NL 144 524  90 159  33   0 36 108  1  0 44
## 14 zimmery01   2018     1    WAS   NL  85 288  33  76  21   2 13  51  1  1 30
## 15 zimmery01   2019     1    WAS   NL  52 171  20  44   9   0  6  27  0  0 17
## 16 zimmery01   2021     1    WAS   NL 110 255  27  62  16   0 14  46  0  0 16
##     SO IBB HBP SH SF GIDP
## 1   12   0   0  0  1    1
## 2  120   7   2  1  4   15
## 3  125   3   3  0  5   26
## 4   71   1   3  0  4   12
## 5  119   9   2  0  9   22
## 6   98   6   4  0  5   16
## 7   73   4   1  0  3   14
## 8  116   8   2  0  4   20
## 9  133   2   2  0  3   16
## 10  37   0   0  0  4    6
## 11  79   0   1  0 10   13
## 12 104   1   5  0  6   12
## 13 126   1   3  0  5   16
## 14  55   1   3  0  2   10
## 15  39   0   0  0  2    4
## 16  77   0   0  0  2    9
```

The first step in this project is to identify Zimmerman’s unique player ID in the Lahman database and assign it a name. I called his player ID “Zim_id.” I then made sure that there were no data points missing from his batting table, which would be labeled “NA” if there were any.

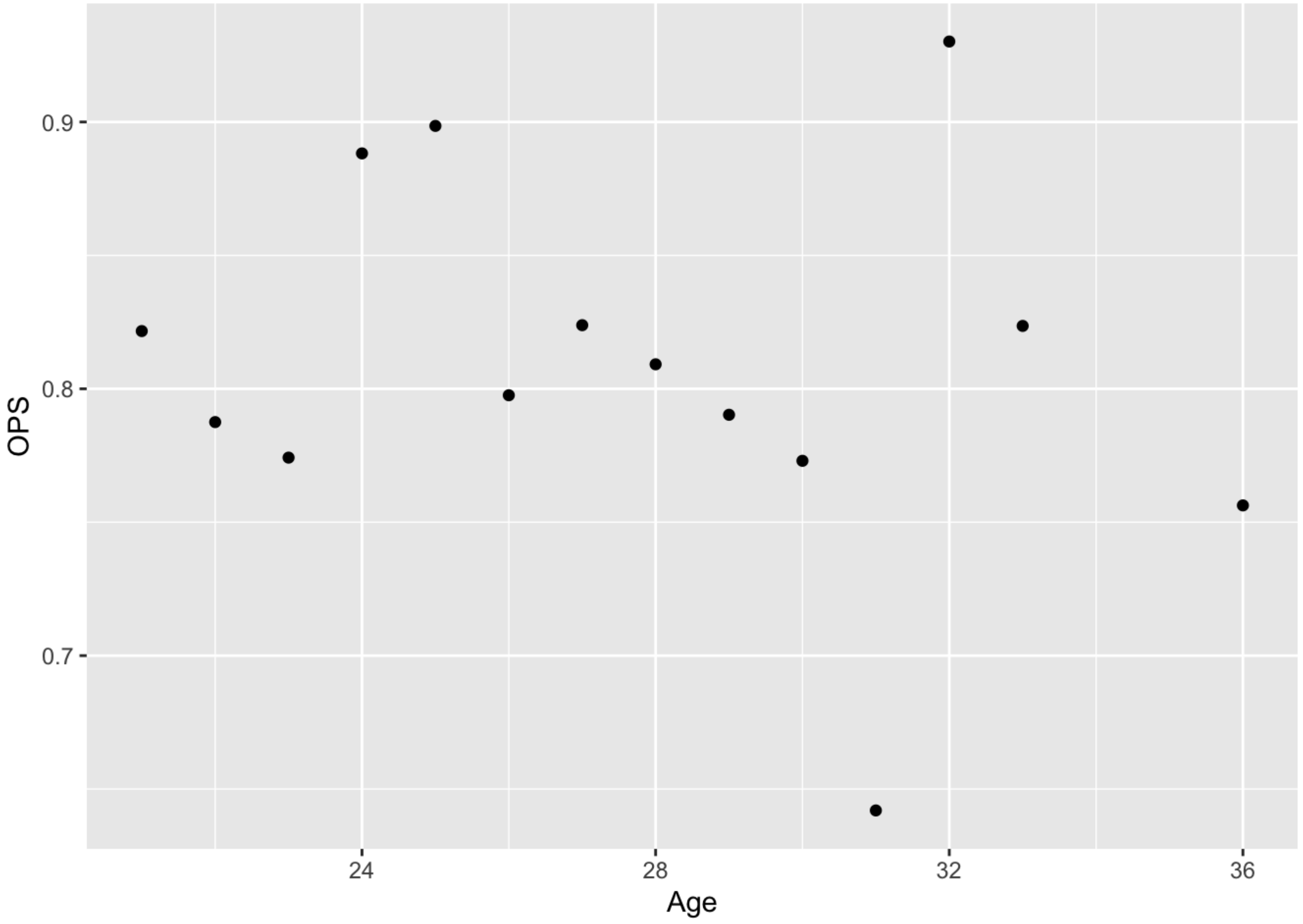
```
batting <- Batting  %>% filter(AB >= 200)
```

In order to properly use the data, I want to filter out any of Zimmerman’s seasons where he had less than 200 at-bats. With less than 200 at-bats, the sample sizes per season are too small and the data would not accurately reflect his career trajectory.

```
get_stats <- function(player.id) {
  batting %>%
    filter(playerID == player.id) %>%
    inner_join(People, by = "playerID") %>%
    mutate(birthyear = ifelse(birthMonth >= 7,
                              birthYear + 1, birthYear),
           Age = yearID - birthyear,
           SLG = (H - X2B - X3B - HR + 2 * X2B + 3*X3B + 4*HR)/AB,
           OBP = (H+BB+HBP)/(AB+BB+HBP+SF),
           OPS = SLG+OBP) %>%
    select(Age, SLG, OBP, OPS)
}
```

With this code, the function “get_stats” will populate the player’s age in that season, slugging (SLG), on-base percentage (OBP), and on-base plus slugging (OPS) using their playerID. I have also defined the statistics and used the mutate function to add the new measures.

```
Zim <- get_stats(Zim_id)
ggplot(Zim, aes(Age, OPS)) + geom_point()
```



I have now used the “get_stats” function and created a scatter plot of Zimmerman’s OPS numbers per season against his age in that season. As we can see, the data points in the latter part of his career become rather interesting, as he has a career low OPS and then the next year a career high, all while over the age of 30. After the age of 28, his OPS started to decline, but had that late career resurgence in 2017. He then continued to have a rather high OPS even though he dealt with numerous injuries.

```
fit_model <- function(d) {
  fit <- lm(OPS ~ I(Age - 30) + I((Age-30)^2), data = d)
  b <- coef(fit)
  Age.max <- 30 - b[2]/b[3]/2
  Max <- b[1]-b[2]^2/b[3]/4
  list(fit = fit, Age.max = Age.max, Max=Max)
}
```

I now want to create the linear regression model that shows his career trajectory tracking OPS as a function of Age. In order to do so, we estimate that 30 years old is the peak of a player’s career in terms of OPS. Since the player should start and end their career with a lower OPS than their peak, a quadratic curve makes the most sense in this scenario. The function’s input is a data set, marked by data=d. Using the coefficients from the line of best fit, the model creates the variables Age.max and Max.

```
F2 <- fit_model(Zim)
coef(F2$fit)
```

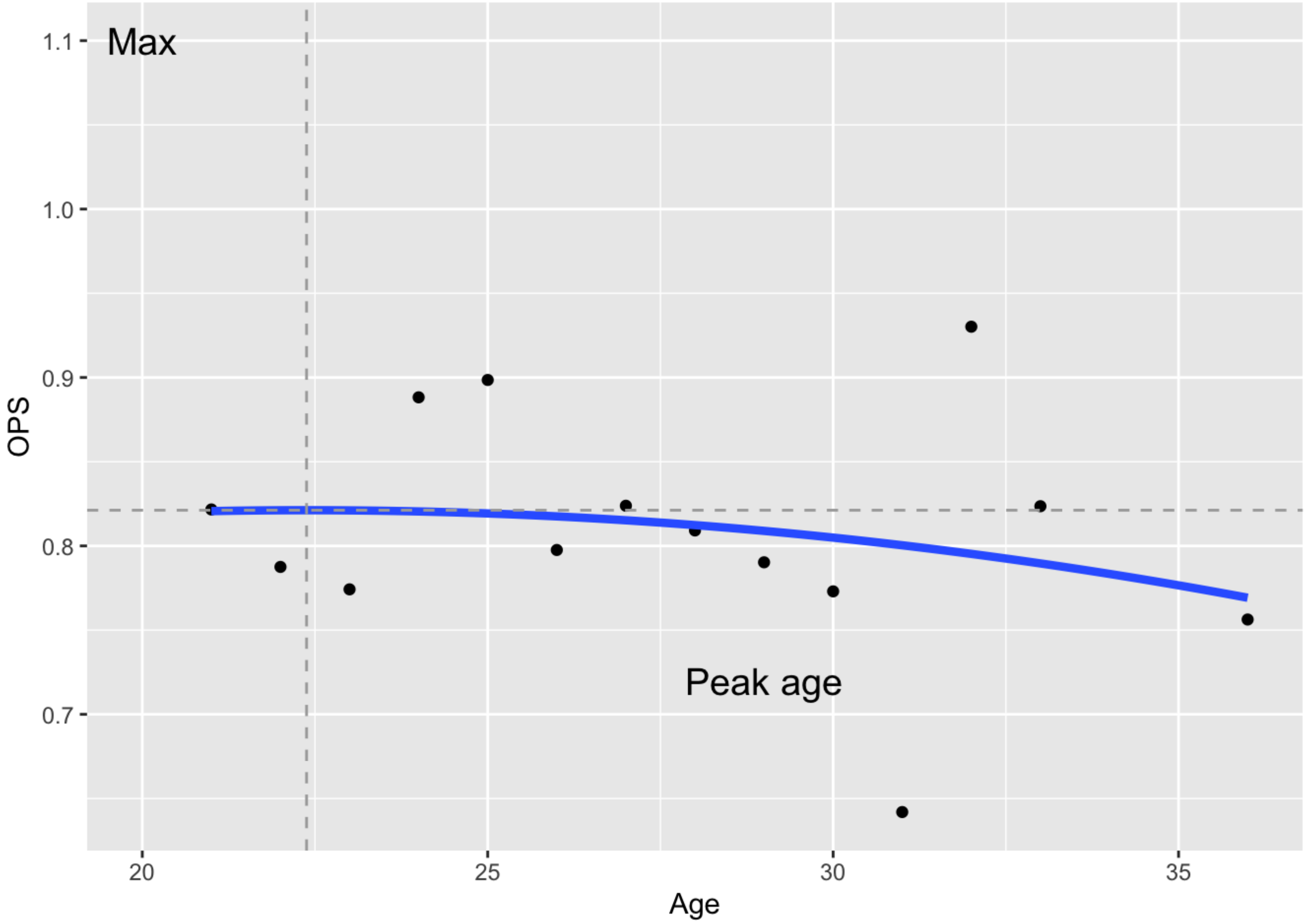
```
##     (Intercept)     I(Age - 30) I((Age - 30)^2)
##    0.8049257202   -0.0042688444   -0.0002800335
```

```
c(F2$Age.max, F2$Max)
```

```
## I(Age - 30) (Intercept)
##  22.3779758   0.8211943
```

This code chunk shows the details from the line of best fit. With this line, the model says that Ryan Zimmerman’s peak age is at 22, with a peak OPS being .821. What this model also tells us is that his peak was not necessarily a fluke, as his OPS only drops .00028 points between his peak year and the next year. This makes sense with the career that Zimmerman had up to his 2016 year, as his OPS was rather consistent in his healthy years.

```
ggplot(Zim, aes(Age, OPS)) + geom_point() +
  geom_smooth(method = "lm", se= FALSE, size = 1.5,
              formula = y ~ poly(x, 2, raw = TRUE)) +
  geom_vline(xintercept = F2$Age.max,
             linetype = "dashed", color = "darkgrey") +
  geom_hline(yintercept = F2$Max,
             linetype = "dashed", color = "darkgrey") +
  annotate(geom = "text", x=c(29,20), y=c(.72,1.1),
           label = c("Peak age", "Max"), size = 5)
```



This graph proves the point I made above, as Zimmerman never had any serious fluctuations in OPS over his career and stayed rather consistent even though he battled injuries. The line of best fit stays impressively level even towards the end of his career, with his OPS in his final year only about .07 lower than his peak.

```
F2 %>% pluck("fit") %>% summary()
```

```
##
## Call:
## lm(formula = OPS ~ I(Age - 30) + I((Age - 30)^2), data = d)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.158405 -0.028921 -0.008046  0.027638  0.134890
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.804926   0.027584  29.181 9.01e-12 ***
## I(Age - 30)     -0.004269   0.006142  -0.695    0.501
## I((Age - 30)^2) -0.000280   0.001055  -0.265    0.796
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07426 on 11 degrees of freedom
## Multiple R-squared:  0.04741,    Adjusted R-squared:  -0.1258
## F-statistic: 0.2737 on 2 and 11 DF,  p-value: 0.7656
```

Using the summary function, the statistics for the model are listed above. This summary tells me that the quadratic function does not really represent Ryan Zimmerman's career trajectory too well, as the multiple R-squared value is rather low, which says that 4% of the data is actually represented by the curve. Although the quadratic curve is not entirely accurate, the residual standard error is decently high at .074, which means that the line of best fit is .074 OPS points away from the actual point on average. These statistics make sense for Zimmerman's career as he didn't have the typical parabolic career trajectory that most players have. Zimmerman started as a good player and ended his career well, with two seasons where he really differed from his typical stats. This is an awfully impressive career trajectory, considering the many injuries that he suffered. Not many players have the ability to be great, suffer serious injuries, and then come back to be the same player he was before, and sometimes play even better. He didn't necessarily have the peak that some great players had, but he was consistently impressive from start to finish.

```
batting %>%
  group_by(playerID) %>%
  summarize(Career.AB = sum(AB, na.rm = TRUE)) %>%
  inner_join(batting, by = "playerID") %>%
  filter(Career.AB >= 2000) -> batting_2000
```

Now, in order to compare the career trajectory of Ryan Zimmerman to others, I need to create a data set of hitters with more than 2000 at-bats in order to filter out pitchers and hitters with especially short careers. Since the observations in the batting data frame is player years, I also needed to use the group_by and summarize functions to compile the career at bats for each player and create a new column called Career.ABs. With that new column, I can filter the players out by their number of career at bats.

```
Fielding %>%
  group_by(playerID, POS) %>%
  summarize(Games = sum(G)) %>%
  arrange(playerID, desc(Games)) %>%
  filter(POS == first(POS)) -> Positions
```

Since I want to compare Ryan Zimmerman to other 3rd Baseman (since that’s where he played for the majority of his career), I need to use the the fielding data frame and organize them by position and number of games, so that they are identified with the position they played the most games at.

```
batting_2000 <- batting_2000 %>%
  inner_join(Positions, by = "playerID")
glimpse(batting_2000)
```

```
## Rows: 21,686
## Columns: 25
## $ playerID  <chr> "aaronha01", "aaronha01", "aaronha01", "aaronha01", "aaronha…
## $ Career.AB <int> 12364, 12364, 12364, 12364, 12364, 12364, 12364, 12364, 1236…
## $ yearID    <int> 1954, 1955, 1956, 1957, 1958, 1959, 1960, 1961, 1962, 1963, …
## $ stint     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
## $ teamID    <fct> ML1, ML1, ML1, ML1, ML1, ML1, ML1, ML1, ML1, ML1, ML1, ML1, …
## $ lgID      <fct> NL, NL, NL, NL, NL, NL, NL, NL, NL, NL, NL, NL, NL, NL, NL, …
## $ G         <int> 122, 153, 153, 151, 153, 154, 153, 155, 156, 161, 145, 150, …
## $ AB        <int> 468, 602, 609, 615, 601, 629, 590, 603, 592, 631, 570, 570, …
## $ R         <int> 58, 105, 106, 118, 109, 116, 102, 115, 127, 121, 103, 109, 1…
## $ H         <int> 131, 189, 200, 198, 196, 223, 172, 197, 191, 201, 187, 181, …
## $ X2B       <int> 27, 37, 34, 27, 34, 46, 20, 39, 28, 29, 30, 40, 23, 37, 33, …
## $ X3B       <int> 6, 9, 14, 6, 4, 7, 11, 10, 6, 4, 2, 1, 1, 3, 4, 3, 1, 3, 0, …
## $ HR        <int> 13, 27, 26, 44, 30, 39, 40, 34, 45, 44, 24, 32, 44, 39, 29, …
## $ RBI       <int> 69, 106, 92, 132, 95, 123, 126, 120, 128, 130, 95, 89, 127, …
## $ SB        <int> 2, 3, 2, 1, 4, 8, 16, 21, 15, 31, 22, 24, 21, 17, 28, 9, 9, …
## $ CS        <int> 2, 1, 4, 1, 1, 0, 7, 9, 7, 5, 4, 4, 3, 6, 5, 10, 0, 1, 0, 1,…
## $ BB        <int> 28, 49, 37, 57, 59, 51, 60, 56, 66, 78, 62, 60, 76, 63, 64, …
## $ SO        <int> 39, 61, 54, 58, 49, 54, 63, 64, 73, 94, 46, 81, 96, 97, 62, …
## $ IBB       <int> NA, 5, 6, 15, 16, 17, 13, 20, 14, 18, 9, 10, 15, 19, 23, 19,…
## $ HBP       <int> 3, 3, 2, 0, 1, 4, 2, 2, 3, 0, 0, 1, 1, 0, 1, 2, 2, 2, 1, 1, …
## $ SH        <int> 6, 7, 5, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ SF        <int> 4, 4, 7, 3, 3, 9, 12, 9, 6, 5, 2, 8, 8, 6, 5, 3, 6, 5, 2, 4,…
## $ GIDP      <int> 13, 20, 21, 13, 21, 19, 8, 16, 14, 11, 22, 15, 14, 11, 21, 1…
## $ POS       <chr> "OF", "OF", "OF", "OF", "OF", "OF", "OF", "OF", "OF", "OF", …
## $ Games     <int> 2760, 2760, 2760, 2760, 2760, 2760, 2760, 2760, 2760, 2760, …
```

I then want to join the two data frames so that the hitters with more than 2000 at bats are given their positions from the fielding data frame.

```
vars <- c("G", "AB", "R", "H", "X2B", "X3B",
          "HR", "RBI", "BB", "SO", "SB")
```

I now want to create a new vector that only has the statistics I want to use to compare players to Ryan Zimmerman. I used the combine function to create that vector.

```
batting %>%
  group_by(playerID) %>%
  summarize_at(vars, sum, na.rm = TRUE) -> C.totals
```

With the summarize_at function, the statistics in the vars vector are totaled and grouped by playerID, so that each player has their individual career stats. I make sure to tell the code to replace any missing values as a 0 so that the code doesn’t fall apart.

```
C.totals %>%
  mutate(AVG = H/AB,
         SLG = (H - X2B - X3B - HR + 2 * X2B + 3*X3B + 4*HR)/AB) -> C.totals
```

I now use the mutate function to create the AVG and SLG column to C.totals, and I tell the code how to calculate each of these statistics.

```
C.totals %>%
  inner_join(Positions, by = "playerID") %>%
  mutate(Value.POS = case_when(
    POS == "C" ~ 240,
    POS == "SS" ~ 168,
    POS == "2B" ~ 132,
    POS == "3B" ~ 84,
    POS == "OF" ~ 48,
    POS == "1B" ~ 12,
    TRUE ~ 0)) -> C.totals
```

I give the players in C.totals their positions using the inner_join function and create a new column that assigns a certain worth to each position. These values were introduced by baseball scientist and statistician Bill James, which suggests that catcher is the most important and first base is the least important position in baseball.

```
similar <- function(p, number = 10) {
  C.totals %>% filter(playerID == p) -> P
  C.totals %>%
    mutate(sim_score = 1000 -
             floor(abs(G - P$G)/ 20) -
             floor(abs(AB - P$AB)/ 75) -
             floor(abs(R - P$R)/ 10) -
             floor(abs(H - P$H)/ 15) -
             floor(abs(X2B - P$X2B)/ 5) -
             floor(abs(X3B - P$X3B) / 4) -
             floor(abs(HR - P$HR)/ 2) -
             floor(abs(RBI - P$RBI)/ 10) -
             floor(abs(BB - P$BB)/ 25) -
             floor(abs(SO - P$SO)/ 150) -
             floor(abs(SB - P$SB)/ 20) -
             floor(abs(AVG - P$AVG)/ 0.001) -
             floor(abs(SLG - P$SLG)/ 0.002) -
             abs(Value.POS - P$Value.POS)) %>%
    arrange(desc(sim_score)) %>%
    head(number)
}
```

Bill James also created the concept of similarity scores that demonstrated a similarity in statistics between players. This code uses the similar function to find players with the most similar statistics to a given player.

```
similar(Zim_id)
```

```
## # A tibble: 10 × 18
##    playerID      G    AB     R     H   X2B   X3B    HR   RBI    BB    SO    SB
##    <chr>     <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
##  1 zimmery01  1727  6425   937  1779   398    22   278  1028   626  1333    43
##  2 longoev01  1770  6636   935  1769   399    25   310  1061   669  1501    58
##  3 castivi02  1742  6557   875  1812   331    27   314  1073   412  1015    31
##  4 bonilbo01  1869  6631  1016  1881   385    59   273  1102   829  1074    40
##  5 frymatr01  1698  6481   895  1776   345    40   223  1022   602  1369    72
##  6 venturo01  1858  6575   944  1768   317    12   278  1108   993  1096    24
##  7 parrila01  1814  6565   831  1734   350    33   249   962   515  1308    30
##  8 dyeje01    1763  6487   984  1779   363    25   325  1072   597  1308    46
##  9 lowelmi01  1593  5798   770  1615   394     7   223   952   548   816    30
## 10 luzingr01  1785  6393   867  1763   336    24   304  1113   830  1458    35
## # … with 6 more variables: AVG <dbl>, SLG <dbl>, POS <chr>, Games <int>,
## #   Value.POS <dbl>, sim_score <dbl>
```

This is table that shows the most similar players to Ryan Zimmerman. Obviously he is at the top, but then it shows all the players that have the most similar statistics to Zim based on Bill James's model, with the most similar being Evan Longoria.

```
batting_2000 %>%
  group_by(playerID, yearID) %>%
  summarize(G = sum(G), AB = sum(AB), R = sum(R),
            H = sum(H), X2B = sum(X2B), X3B = sum(X3B),
            HR = sum(HR), RBI = sum(RBI), SB = sum(SB),
            CS = sum(CS), BB = sum(BB), SH = sum(SH),
            SF = sum(SF), HBP = sum(HBP),
            Career.AB = first(Career.AB),
            POS = first(POS)) %>%
  mutate(SLG = (H - X2B - X3B - HR + 2 * X2B +
                  3 * X3B + 4 * HR) / AB,
         OBP = (H + BB + HBP) / (AB + BB + HBP + SF),
         OPS = SLG+OBP)-> batting_2000
```

In this code, the batting_2000 data frame is organized by career stats and uses the sum function and then adds SLG, OBP, and OPS as columns to the data frame of batting_2000

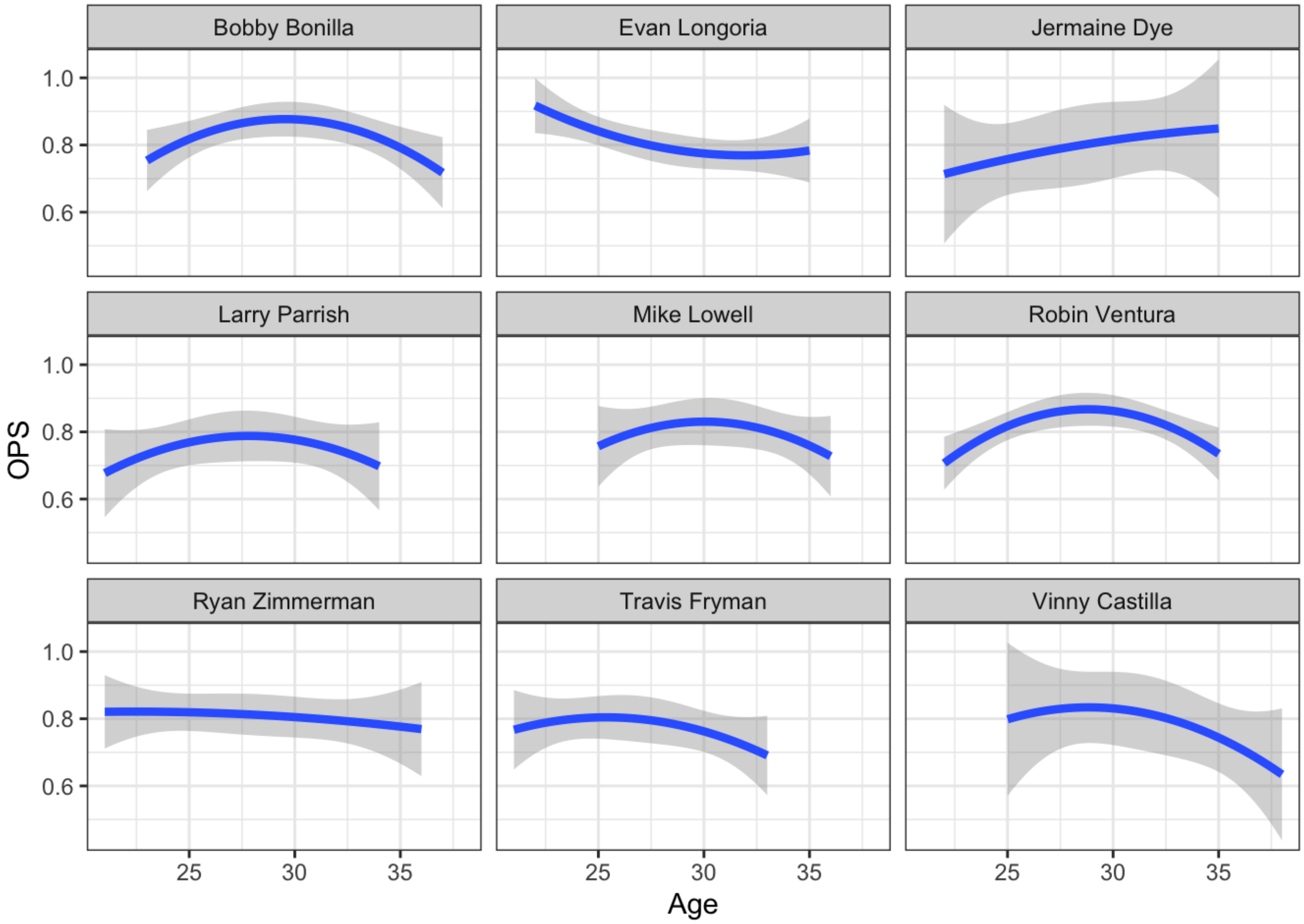
```
batting_2000 %>%
  inner_join(People, by = "playerID") %>%
  mutate(Birthyear = ifelse(birthMonth >= 7,
                            birthYear +1, birthYear),
         Age = yearID - Birthyear) -> batting_2000
batting_2000 %>% drop_na(Age) -> batting_2000
```

In the batting data frame, age is not a column, so we need to write code in order to create it. First, I have to inner_join the people data frame and batting data frame so that birth year can be in the data frame. I then add the age column to the data frame using the yearID - birthyear equation.

```
plot_trajectories <- function(player, n.similar = 8, ncol=3) {
  flnames <- unlist(strsplit(player, " "))

People %>%
    filter(nameFirst == flnames[1],
           nameLast == flnames[2]) %>%
    select(playerID) -> player

 player.list <- player %>%
    pull(playerID) %>%
    similar(n.similar) %>%
    pull(playerID)

 batting_2000 %>%
    filter(playerID %in% player.list) %>%
    mutate(Name = paste(nameFirst, nameLast)) -> Batting.new

ggplot(Batting.new, aes(Age, OPS)) +
    geom_smooth(method = "lm",
                formula = y ~ x + I(x^2),
                size = 1.5) +
    facet_wrap(~ Name, ncol = ncol) + theme_bw()
}
```

I now need a new function that will take the similar players and plot their own career trajectories. I use the similar function to find the players that are similar to Ryan Zimmerman. I chose that I wanted 8 similar players to be displayed in 3 columns and I then plot them with the linear regression model and a geom_smooth line to show the line of best fit. Then I use the facet_wrap function to separate the players onto different graphs.

```
plot_trajectories("Ryan Zimmerman", 9, 3)
```



I then tell the function that I made to take Ryan Zimmerman and show the 8 most similar players and put them in three columns. That plot is shown above.

Conclusion

For the most part, the trajectories of these players are very similar to Ryan Zimmerman. They all hover around that .800 -.900 OPS for the peak of their careers, which proves that they were all good players, but never Hall of Fame caliber players. Evan Longoria was said to be the most similar statistically and looks to have the most similar trajectory, as he started out as a fantastic player and didn't really improve much after his rookie year. The unique aspect of Zimmerman's career, though, is how consistent he stayed throughout his career. Compared to someone like Vinny Castilla, who struggled mightily towards the end of his career, Zimmerman stays surprisingly close to his peak in terms of OPS as his age increases. To have such sustained production from a young age, battle through injuries, and continue to produce is highly impressive and it is what makes him such a unique player.

Citations

Marchi, M., Albert, J., & Baumer, B. (2019). Analyzing baseball data with R. CRC Press, Taylor & Francis Group.

```
citation("Lahman")
```

```
##
## To cite package 'Lahman' in publications use:
##
##   Friendly M, Dalzell C, Monkman M, Murphy D (2022). _Lahman: Sean
##   'Lahman' Baseball Database_. R package version 10.0-1,
##   <https://CRAN.R-project.org/package=Lahman>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {Lahman: Sean 'Lahman' Baseball Database},
##     author = {Michael Friendly and Chris Dalzell and Martin Monkman and Dennis Murph
y},
##     year = {2022},
##     note = {R package version 10.0-1},
##     url = {https://CRAN.R-project.org/package=Lahman},
##   }
```

```
citation("tidyverse")
```

```
##
## To cite package 'tidyverse' in publications use:
##
##   Wickham et al., (2019). Welcome to the tidyverse. Journal of Open
##   Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {Welcome to the {tidyverse}},
##     author = {Hadley Wickham and Mara Averick and Jennifer Bryan and Winston Chang an
d Lucy D'Agostino McGowan and Romain François and Garrett Grolemund and Alex Hayes and L
ionel Henry and Jim Hester and Max Kuhn and Thomas Lin Pedersen and Evan Miller and Step
han Milton Bache and Kirill Müller and Jeroen Ooms and David Robinson and Dana Paige Sei
del and Vitalie Spinu and Kohske Takahashi and Davis Vaughan and Claus Wilke and Kara Wo
o and Hiroaki Yutani},
##     year = {2019},
##     journal = {Journal of Open Source Software},
##     volume = {4},
##     number = {43},
##     pages = {1686},
##     doi = {10.21105/joss.01686},
##   }
```