wOBA for Softball

Matthew Brownsword

4/19/2017

Code appendix

This contains most, if not all, of the set-up code necessary to explain the data, analyze results and generate discussion.

Set-up

```
# Change on your machine
library(mosaic)
```

```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: lattice

## Loading required package: ggplot2

## Loading required package: mosaicData

## Loading required package: Matrix

##
## The 'mosaic' package masks several functions from core packages in order to add additional features.
## The original behavior of these functions should not be affected by this.

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cov, D, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```r
library(knitr)

getwd()
```

```
## [1] "/Users/mbrownsword/Downloads"
```

```r
setwd("/Users/mbrownsword/Downloads")



## Easy way to look for and install missing packages and load them
if (!require("pacman")){ install.packages("pacman") }
```

```
## Loading required package: pacman
```

```r
pacman::p_load("knitr","mosaic","plyr","ggplot2","foreign","gridExtra")

## Set some default options R Markdown
opts_chunk$set(tidy=TRUE,echo=FALSE,results='markup',strip.white=TRUE,cache=T,highlight=TRUE,width.cuto
knitr::opts_chunk$set(echo =TRUE)
library(pander)
install.packages("tables", repos='http://cran.us.r-project.org')
```

```
##
## The downloaded binary packages are in
##  /var/folders/c6/dx184fds50q5pxzjdg65_y240000gn/T//RtmpVlxhrS/downloaded_packages
```

```r
# Default options in R
options(digits=4,width=100,scipen=12)

load("Softball_Stats2016.rda")
load("BaseballOStats2016.rda")
load("Softball_Stats2015.rda")
load("Softball_Stats2014.rda")

Sys.setenv(PATH=paste("/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/local/texlive/2016/bin/x86_64-
Sys.getenv("PATH")
```

```
## [1] "/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/local/texlive/2016/bin/x86_64-darwin"
```

```r
Sys.which("pdflatex")
```

```
##                                              pdflatex
## "/usr/local/texlive/2016/bin/x86_64-darwin/pdflatex"
```

1

```r
Softball_Stats2016[is.na(Softball_Stats2016)] <- 0



# Creating a marker for wOBA
Softball_Stats2016$oneB <- with(Softball_Stats2016, H - twoB - threeB - HR)
```

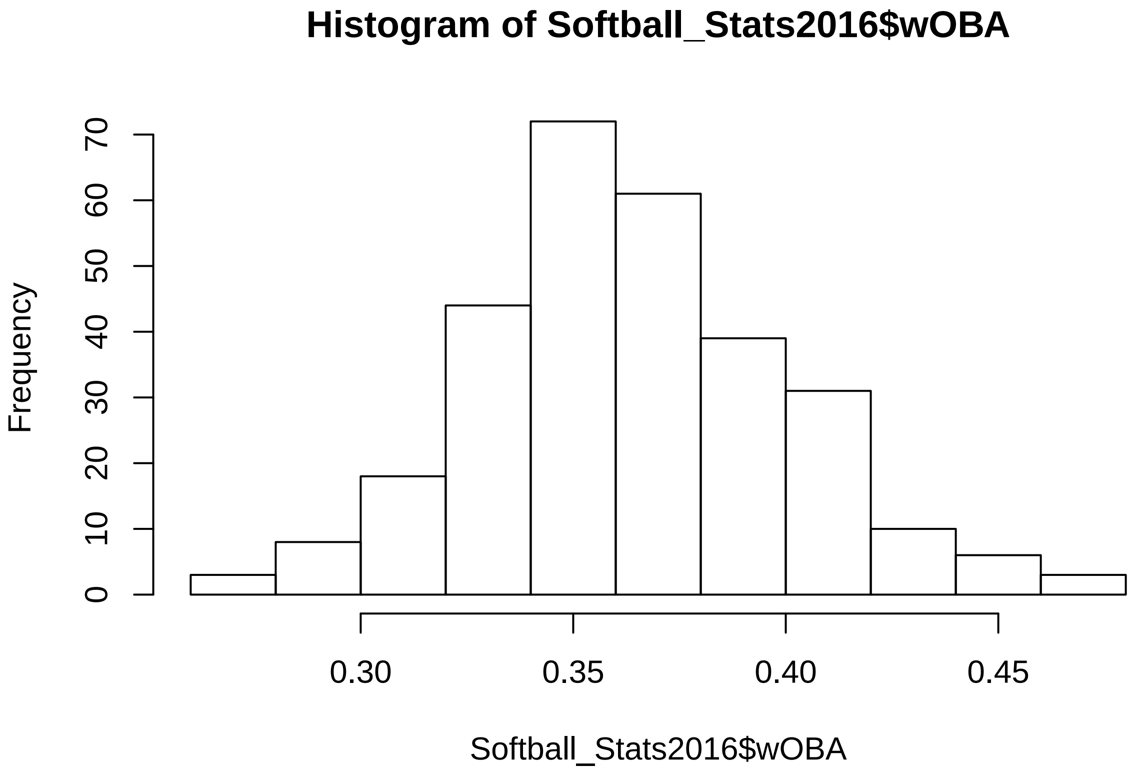
```
Softball_Stats2016$RBOE <- with(Softball_Stats2016, AB - GO - FO - K - H + SF + SH)

Softball_Stats2016$wOBA <- with(Softball_Stats2016, (0.72 * BB + 0.75 * HBP + 0.9 * oneB + 1.24 * twoB +
    1.56 * threeB + 1.95 * HR + 0.92 * RBOE)/(AB + BB + IBB + SH + SF + HBP), na.rm = T)

# Creating a function marking winning % and wOBA
lmwSoftballOBA2016 <- lm(with(Softball_Stats2016, PCT ~ wOBA))
lmwSoftballOBA2016
```

```

Call:
lm(formula = with(Softball_Stats2016, PCT ~ wOBA))

Coefficients:
(Intercept)         wOBA
      -0.37         2.36
```

```
lmwSoftballOBA2016R <- lm(with(Softball_Stats2016, R ~ wOBA))
lmwSoftballOBA2016R
```

```

Call:
lm(formula = with(Softball_Stats2016, R ~ wOBA))

Coefficients:
(Intercept)         wOBA
       -443         1875
```

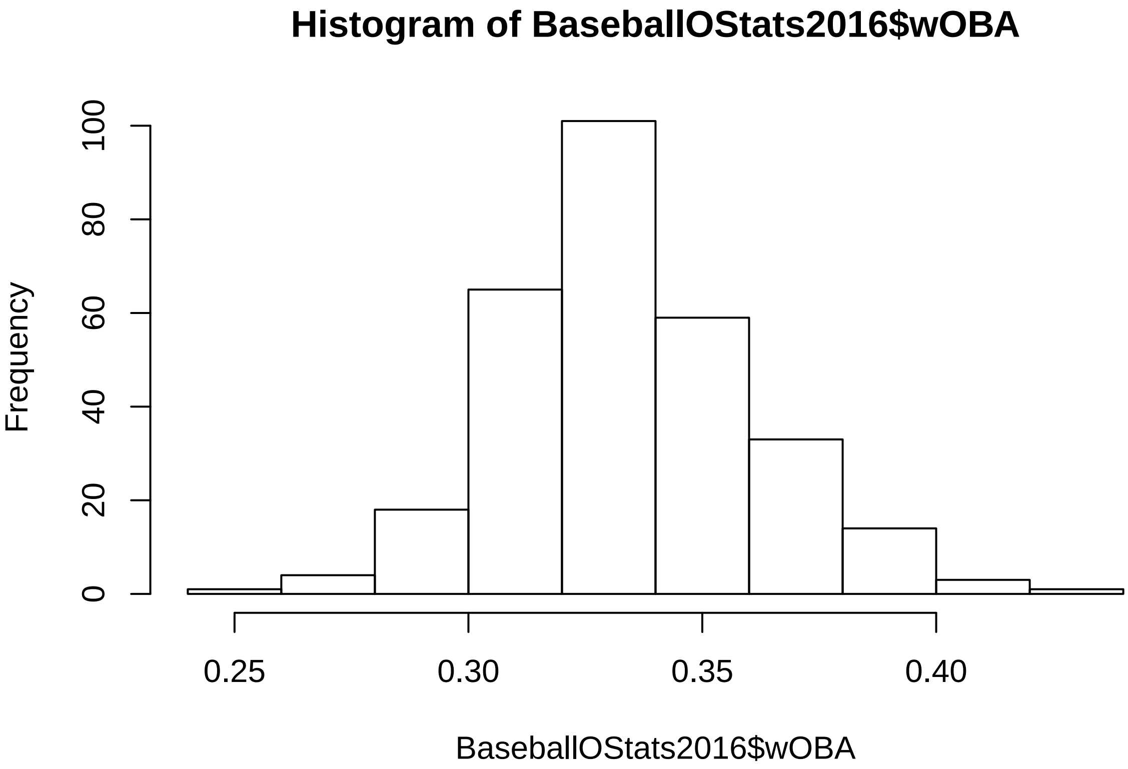
```
# Looking at the distribution of Softball wOBAs, looks normal
hist(Softball_Stats2016$wOBA)
```



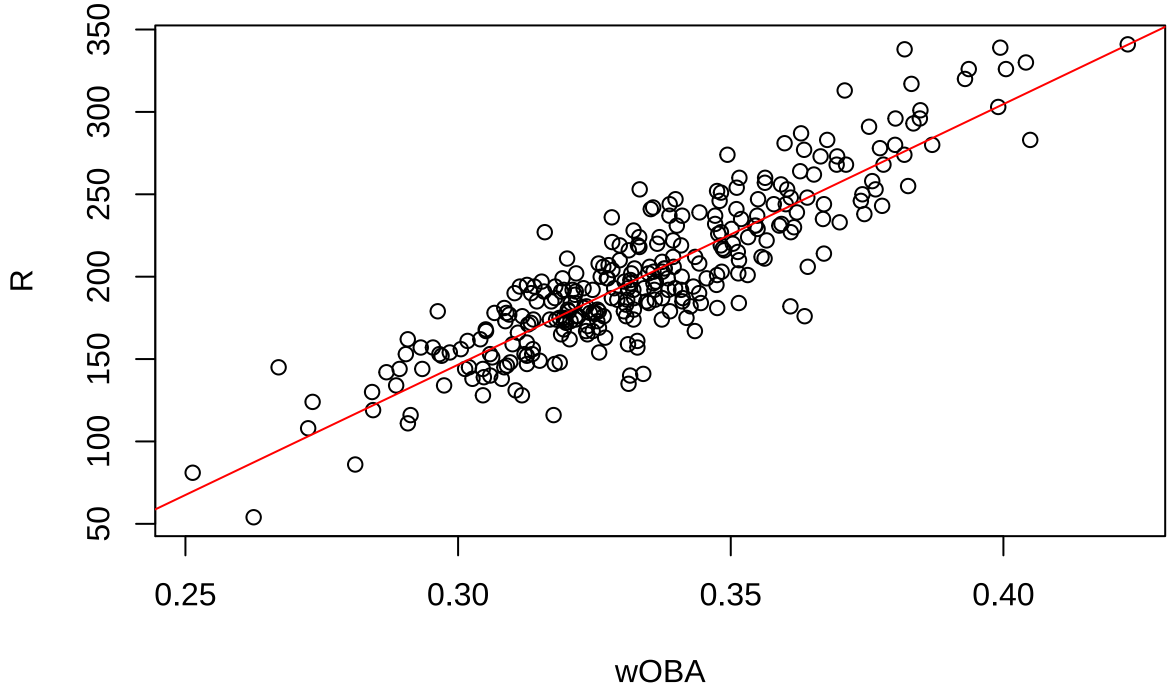
```r
# Repeat the process for 2016 baseball statistics
BaseballOStats2016$oneB <- with(BaseballOStats2016, H - twoB - threeB - HR)

BaseballOStats2016$wOBA <- with(BaseballOStats2016, (0.72 * BB + 0.75 * HBP + 0.9 * oneB + 1.24 * twoB
    1.56 * threeB + 1.95 * HR)/(AB + BB + SH + SF + HBP), na.rm = T)
lmwBaseballOBA2016 <- lm(with(BaseballOStats2016, R ~ wOBA))
lmwBaseballOBA2016
```

```
Call:
lm(formula = with(BaseballOStats2016, R ~ wOBA))

Coefficients:
(Intercept)         wOBA
       -328         1581
```

```r
hist(BaseballOStats2016$wOBA)
```



```r
plot(x = BaseballOStats2016$wOBA, y = BaseballOStats2016$R, xlab = "wOBA", ylab = "R")
abline(lmwBaseballOBA2016, col = "red")
```



```
rsquared(lmwBaseballOBA2016)
```

```
[1] 0.7839
```

```
## 2015

Softball_Stats2015[is.na(Softball_Stats2015)] <- 0


# Creating a marker for wOBA
Softball_Stats2015$oneB <- with(Softball_Stats2015, H - twoB - threeB - HR)
Softball_Stats2015$RBOE <- with(Softball_Stats2015, AB - GO - FO - K - H + SF + SH)

Softball_Stats2015$wOBA <- with(Softball_Stats2015, (0.72 * BB + 0.75 * HBP + 0.9 * oneB + 1.24 * twoB +
    1.56 * threeB + 1.95 * HR + 0.92 * RBOE)/(AB + BB + IBB + SH + SF + HBP), na.rm = T)

# Creating a function marking winning % and wOBA
lmwSoftballOBA2015 <- lm(with(Softball_Stats2015, PCT ~ wOBA))
lmwSoftballOBA2015
```

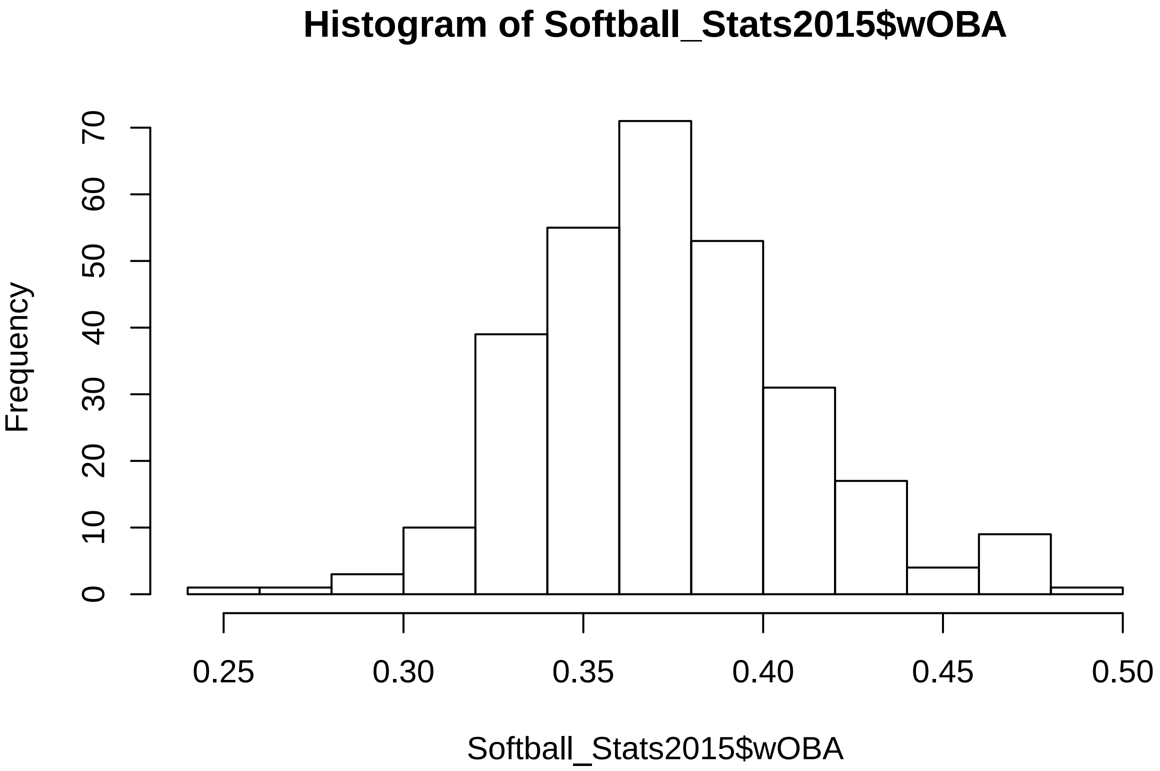
```

Call:
lm(formula = with(Softball_Stats2015, PCT ~ wOBA))

Coefficients:
(Intercept)         wOBA
      -0.06         1.47
```

```
lmwSoftballOBA2015R <- lm(with(Softball_Stats2015, R ~ wOBA))
lmwSoftballOBA2015R
```

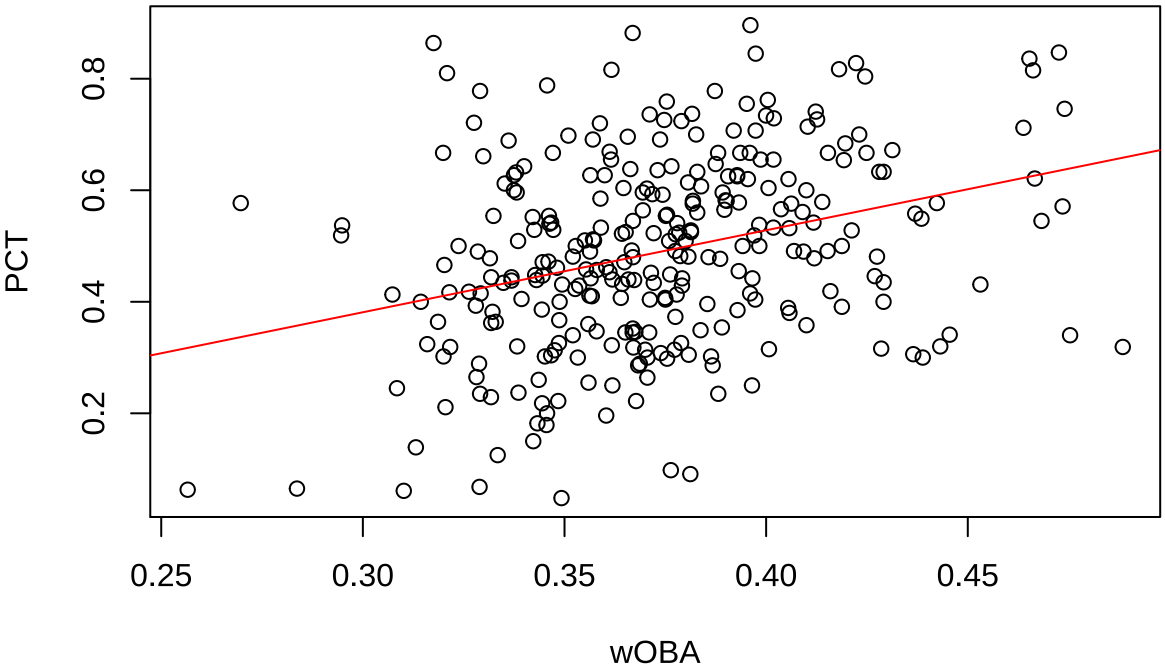
```

Call:
lm(formula = with(Softball_Stats2015, R ~ wOBA))
```

```
Coefficients:
(Intercept)         wOBA
       -501         1999
```

```
# Looking at the distribution of Softball wOBAs, looks normal
hist(Softball_Stats2015$wOBA)
```



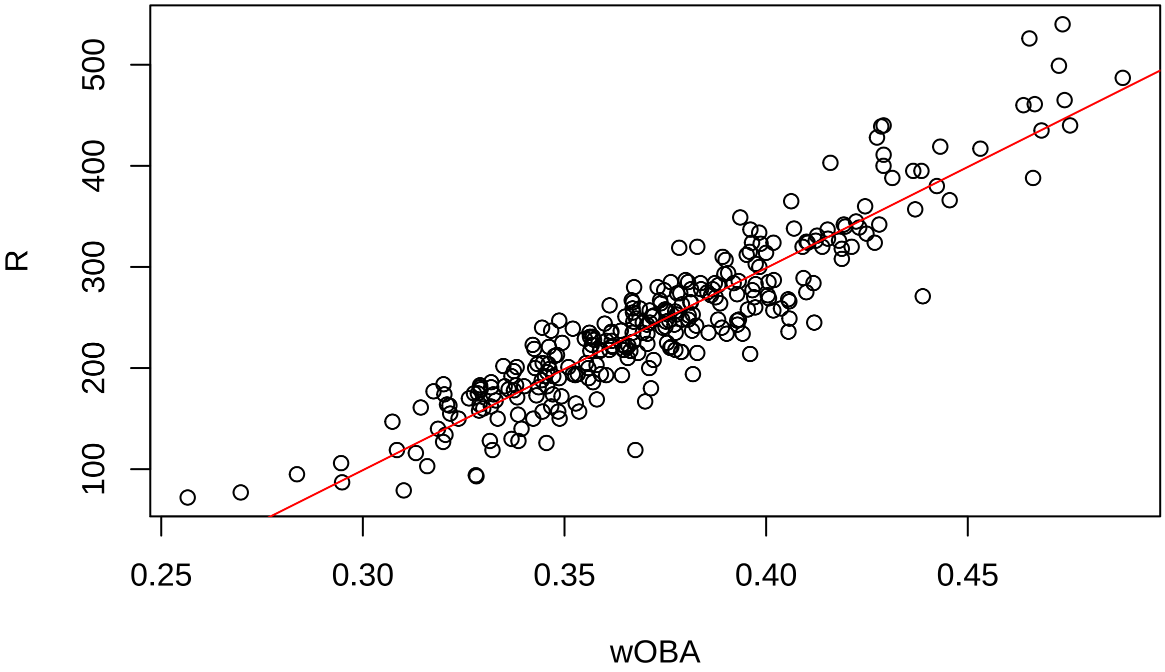
```
# Plot the function with the dataset
plot(x = Softball_Stats2015$wOBA, y = Softball_Stats2015$PCT, xlab = "wOBA", ylab = "PCT")
abline(lmwSoftballOBA2015, col = "red")
```



```
rsquared(lmwSoftballOBA2015)
```

```
[1] 0.1023
```

```
# Plot the function with the dataset
plot(x = Softball_Stats2015$wOBA, y = Softball_Stats2015$R, xlab = "wOBA", ylab = "R")
abline(lmwSoftballOBA2015R, col = "red")
```



```
rsquared(lmwSoftballOBA2015R)
```

```
[1] 0.8501
```

```
## 2014
```

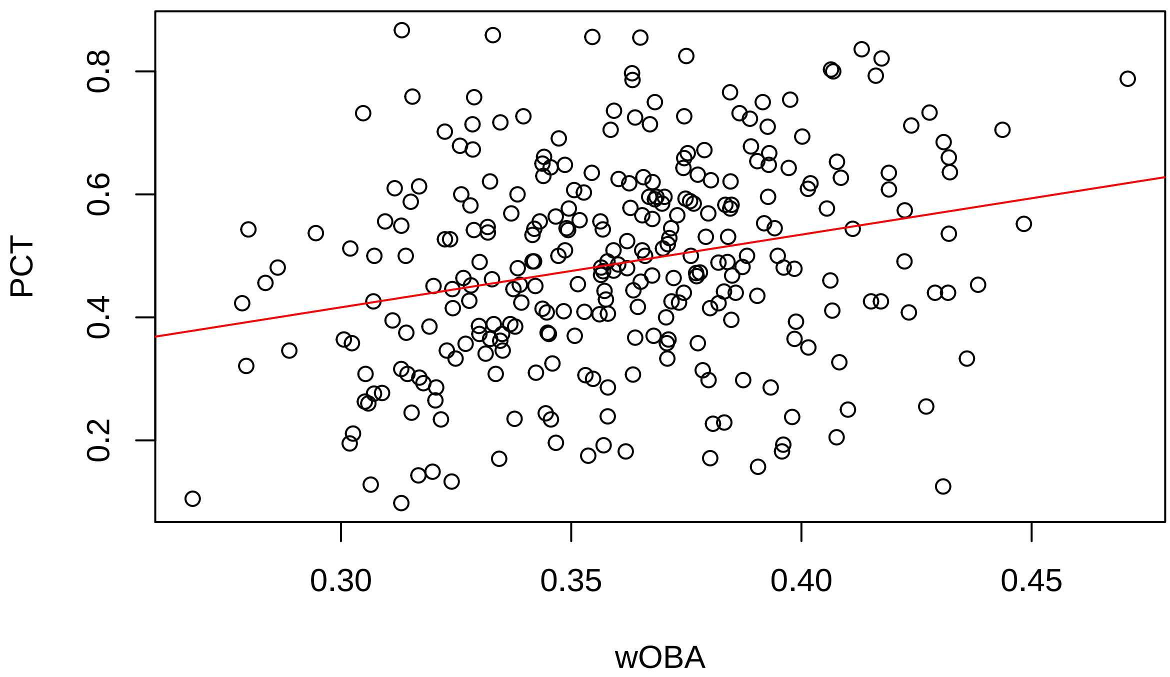
```
Softball_Stats2014[is.na(Softball_Stats2014)] <- 0

# Creating a marker for wOBA
Softball_Stats2014$oneB <- with(Softball_Stats2014, H - twoB - threeB - HR)
Softball_Stats2014$RBOE <- with(Softball_Stats2014, AB - GO - FO - K - H + SF + SH)

Softball_Stats2014$wOBA <- with(Softball_Stats2014, (0.72 * BB + 0.75 * HBP + 0.9 * oneB + 1.24 * twoB +
    1.56 * threeB + 1.95 * HR + 0.92 * RBOE)/(AB + BB + IBB + SH + SF + HBP), na.rm = T)

# Creating a function marking winning % and wOBA
lmwSoftballOBA2014 <- lm(with(Softball_Stats2014, PCT ~ wOBA))
lmwSoftballOBA2014
```

```

Call:
lm(formula = with(Softball_Stats2014, PCT ~ wOBA))

Coefficients:
(Intercept)         wOBA
     0.0616       1.1824
```

```
lmwSoftballOBA2014R <- lm(with(Softball_Stats2014, R ~ wOBA))
lmwSoftballOBA2014R
```

```

Call:
lm(formula = with(Softball_Stats2014, R ~ wOBA))

Coefficients:
(Intercept)         wOBA
       -440         1851
```

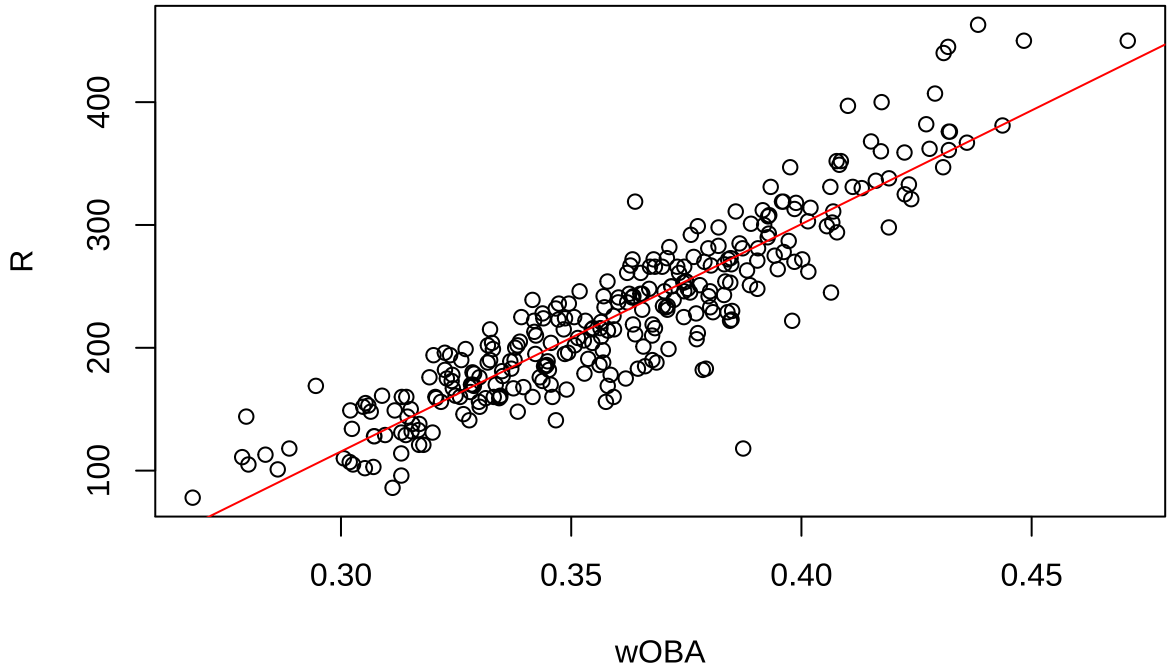
```
# Plot the function with the dataset
plot(x = Softball_Stats2014$wOBA, y = Softball_Stats2014$PCT, xlab = "wOBA", ylab = "PCT")
abline(lmwSoftballOBA2014, col = "red")
```



```
rsquared(lmwSoftballOBA2014)
```

```
[1] 0.06497
```

```
# Plot the function with the dataset
plot(x = Softball_Stats2014$wOBA, y = Softball_Stats2014$R, xlab = "wOBA", ylab = "R")
abline(lmwSoftballOBA2014R, col = "red")
```



```
rsquared(lmwSoftballOBA2014R)
```

```
[1] 0.8305
```

2

```r
save(Softball_Stats2016, file = "Softball_Stats2016.rda")
save(Softball_Stats2015, file = "Softball_Stats2015.rda")
save(Softball_Stats2014, file = "Softball_Stats2014.rda")
```

3

```r
# Load data
load("NCAAPBP.rda")
head(NCAAPBP)
```

```
# A tibble: 6 × 32
   gameID
    <dbl>
1 4008959
2 4008959
3 4008959
4 4008959
5 4008959
6 4008959
# ... with 31 more variables: playID <chr>, inningID <dbl>, sequence <dbl>, `` <dttm>, outs1 <dbl>,
#   doubleplay <dbl>, tripleplay <dbl>, pickedoff <dbl>, outs <dbl>, onfirstsingle <dbl>,
#   onfirstwalk <dbl>, onfirstreached <dbl>, onfirsthitbypitch <dbl>, outatfirst <lgl>,
#   onfirst <dbl>, onthird1 <dbl>, onsecond1 <dbl>, onsecond2 <dbl>, onsecond3 <dbl>,
#   onsecond <dbl>, onthird2 <dbl>, onthird3 <dbl>, onthird <dbl>, runs <lgl>, runs1 <dbl>,
#   runs2 <dbl>, runs3 <dbl>, runs4 <dbl>, totalouts <dbl>, totalruns <dbl>, thrownout <dbl>
```

```r
length(unique(NCAAPBP$gameID))
```

```
[1] 2071
```

```r
# 2071 games Tons of missing data here, not sure what's up
sum(is.na(NCAAPBP$gameID))
```

```
[1] 877063
```

```r
# Getting rid of it for now
season <- NCAAPBP[!is.na(NCAAPBP$gameID), ]
season$play <- trimws(season$playID)
# Create a measure of outs

sum(grepl("[[:space:]]out[[:punct:]]", season$play))
```

```
[1] 182
```

```r
sum(grepl("[[:space:]]out[[:space:]]| popped |[[:space:]]out[:punct:]", season$play))
```

```
[1] 77521
```

```r
sum(grepl("double play", season$play))
```

```
[1] 1175
```

```
# No Triple Plays
sum(grepl("triple play", season$play))
```

```
[1] 0
```

```
sum(grepl("struck out swinging reached", season$play))
```

```
[1] 100
```

```
sum(grepl("[[:space:]]out.*reached", season$play))
```

```
[1] 119
```

```
#
sum(grepl("[[:space:]]out.*reached", season$play))
```

```
[1] 119
```

```
season$play[grepl("[[:space:]]out.*reached", season$play)]
```

```
  [1] "JOHNSON struck out swinging reached first on a passed ball."
  [2] "0 13 0-13 Wartman C struck out swinging reached first on a passed ball; Newbury N advanced to thi
  [3] "Jones Amber struck out swinging reached first on a passed ball (2-2 BKSBFK)."
  [4] "LOPEZ struck out swinging reached first on a passed ball (0-2 FFFFF): BARNES advanced to second.
  [5] "Ogden Krosley struck out swinging reached first on a passed ball."
  [6] "Chamount struck out swinging reached first on a wild pitch."
  [7] "THOMPSON K. struck out swinging reached first on a wild pitch."
  [8] "McKay struck out swinging reached first on a wild pitch (0-2 KKS)."
  [9] "MCGUFFEY K. struck out swinging reached first on a passed ball: MARGAGLIOTTI advanced to third."
 [10] "Sustayta E struck out swinging reached first on a passed ball (1-2 BFFS)."
 [11] "Pruett struck out swinging reached first on a wild pitch (1-2 KFBF)."
 [12] "Hoover K. struck out swinging reached first on a passed ball (1-2 FBKK)."
 [13] "BAKENHUS A. struck out swinging reached first on a wild pitch (1-2); DAVIS Mariah advanced to thi
 [14] "Alderink struck out swinging reached first on a passed ball (2-2): Milligan advanced to third on
 [15] "JAMES H. struck out swinging reached first on a passed ball; BROWN C. advanced to second; WILLIA
 [16] "Anderson T. struck out reached first on a passed ball (1-2 BKFF)."
 [17] "SUGG out at first 3b to ss reached on a fielder's choice (3-2 BBBKF); BELL out on the play."
 [18] "HANCOCK out at first 3b to c reached on a fielder's choice; TARROW advanced to second; MERRELL ad
 [19] "Kimrey struck out swinging reached first on a wild pitch."
 [20] "0 8 0-8 Cyr struck out swinging reached first on a throwing error by c; Salvo scored unearned."
 [21] "Hernandez C struck out swinging reached first on a wild pitch."
 [22] "Messer E. struck out swinging reached first on a passed ball."
 [23] "Schmidt T. struck out swinging reached first on a passed ball; McClain J. advanced to third."
 [24] "VANNOY B. struck out swinging reached first on a passed ball (3-2 BFKBBK)."
 [25] "KENNEDY B struck out swinging reached first on a passed ball (1-2 KSBS)."
 [26] "Allen D struck out swinging reached first on a wild pitch."
 [27] "BUENO M. struck out swinging reached first on a wild pitch (0-2 KKS); DEPIPPO J. advanced to thi
 [28] "Clark struck out swinging reached first on a passed ball advanced to second on an error by c adva
 [29] "Taylor out at first 3b to 2b reached on a fielder's choice; Burnett out on the play."
 [30] "GONZALEZ Desiree struck out swinging reached first on a wild pitch; ANTONACCI Nicole advanced to
 [31] "Keller struck out swinging reached first on an error by c advanced to second on the throw (3-2);
 [32] "Landry N. out at first 1b to ss reached on a fielder's choice: Jones B. out on the play: Kincanno
 [33] "Isenburg K. out at first ss to 2b reached on a fielder's choice: Jones B. out on the play: Rhodes
 [34] "Isenburg K. out at first 2b to ss reached on a fielder's choice: Jones B. out on the play: Kincan
 [35] "Pitek out at first p to 3b reached on a fielder's choice (1-2 KBF): Vieira advanced to second: Do
 [36] "Chambers struck out swinging reached first on a wild pitch (1-2 KBFFFK)."
 [37] "Tinney P. out at first 3b to 2b reached on a fielder's choice (1-0): Vick T. out on the play."
```

```
[38] "Hoover K. struck out swinging reached first on a passed ball (2-2)."
[39] "Holt C struck out reached first on a wild pitch."
[40] "Ogden K. struck out swinging reached first on a wild pitch (2-2 FBFFFBFS)."
[41] "2 5 42771 BARRE AMYE struck out swinging reached first on a wild pitch reached to second on a thi
[42] "GUSTAFSON struck out swinging reached first on a passed ball (1-2 SFBFK); SELLERS advanced to se
[43] "GEARLDS struck out swinging reached first on a passed ball (2-2)."
[44] "Sherlund K struck out swinging reached first on a passed ball (3-2 SBKBBS); Murray L advanced to
[45] "Schanda struck out swinging reached first on a throwing error by c advanced to second on the err
[46] "1 1 42736 JENKINS C struck out swinging reached first on a wild pitch: ECCLES S advanced to seco
[47] "Barrow struck out swinging reached first on a wild pitch (1-2)."
[48] "Gutierrez A struck out swinging reached first on a passed ball (2-2 BKBFS)."
[49] "Corcoran C. struck out swinging reached first on an error by c; Gonzalez P. advanced to third."
[50] "Curtan J struck out swinging reached first on a wild pitch (1-2)."
[51] "Brunck M. struck out swinging reached first on a passed ball."
[52] "Hadley struck out swinging reached first on a wild pitch (3-2); Andolino advanced to third; Fazi
[53] "Brown T. struck out swinging reached first on a passed ball."
[54] "Wenner struck out swinging reached first on a wild pitch (0-2); Houlihan advanced to second."
[55] "Powers A. struck out swinging reached first on a throwing error by c (3-2)."
[56] "DAISS B. struck out swinging reached first on an error by c (2-2 KKFFFBBFS); D'ORAZIO E. advance
[57] "Gabelt struck out swinging reached first on a wild pitch (1-2 BKSFFFFS)."
[58] "SZYMANOWSKI struck out swinging reached first on a wild pitch; CASE advanced to third."
[59] "MAY J. struck out swinging reached first on a passed ball."
[60] "Warren struck out reached first on a wild pitch advanced to second on the throw; Rodgers out on
[61] "0 8 0-8 B. Nakamura struck out swinging reached first on an error by c advanced to second on the
[62] "Collins S struck out swinging reached first on a passed ball."
[63] "Whitley J struck out swinging reached first on a wild pitch (3-2 BBKKFFFBS)."
[64] "VANNOY B. struck out swinging reached first on a passed ball (1-2 KKFB)."
[65] "Cooper struck out swinging reached first on a throwing error by c."
[66] "Bush A. struck out reached first on a throwing error by c; May A. advanced to third; Strub F. sc
[67] "Haney struck out swinging reached first on a wild pitch; Rodriguez out at third c to 2b to 3b."
[68] "HUFF K. struck out swinging reached first on an error by c (1-2 SBSS)."
[69] "BAKENHUS A. out at first 3b to 2b reached on a fielder's choice; DARBY Wanda out on the play."
[70] "DAVIS Mariah out at first 3b to 2b reached on a fielder's choice (0-0); BAKENHUS A. out on the p
[71] "DARBY Wanda out at first ss to 2b reached on a fielder's choice (0-2 FF); RICHARD K. out on the
[72] "Robey struck out swinging reached first on a fielding error by c (3-2 BSBBKS)."
[73] "Walker M. struck out swinging reached first on an error by 1b."
[74] "Kawall A. struck out swinging reached first on a passed ball."
[75] "Johnson A. struck out swinging reached first on a passed ball."
[76] "STRINGER M. struck out swinging reached first on a wild pitch (2-2)."
[77] "Nuccio struck out swinging reached first on a wild pitch."
[78] "WALTERS G. struck out swinging reached first on a wild pitch (3-2 BKBFBS)."
[79] "0 1 0-1 HAYES flied out to cf SAC reached first on the throw RBI; STEVENS advanced to second; ME
[80] "Pierce struck out reached first on a wild pitch; Fagan advanced to third; Nelson scored unearned
[81] "SIDES K struck out swinging reached first on a wild pitch."
[82] "Lemon B. struck out swinging reached first on a wild pitch (3-2 KSBBFB)."
[83] "2 2 42768 YOUNG L. struck out swinging reached first on a wild pitch (0-2 KFS): HALEY P. advance
[84] "STANTON Shersty struck out swinging reached first on a wild pitch; WILSON Jessica advanced to thi
[85] "LEE B. struck out swinging reached first on a passed ball (1-2 KKBFS): STICKROD L advanced to se
[86] "CALLAS Misty struck out swinging reached first on a wild pitch (1-2): HUNTER Makay advanced to s
[87] "JONES C struck out swinging reached first on a throwing error by c RBI (2-2); HARRIS B scored on
[88] "Lucas struck out looking reached first on an error by c."
[89] "LOTZ struck out swinging reached first on a wild pitch; FITZPATRICK advanced to third."
[90] "Montgomery struck out swinging reached first on a wild pitch (1-2)."
[91] "Taynor T. struck out swinging reached first on a wild pitch."
```

```
 [92] "Finley struck out swinging reached first on a fielding error by c RBI: Johnson advanced to third
 [93] "Galovich struck out swinging reached first on a wild pitch (1-2 FKBS)."
 [94] "Belans A J struck out swinging reached first on a wild pitch (1-2 KBSFS)."
 [95] "1 3 42738 Bizzell J. struck out swinging reached first on a wild pitch (3-2 BFKBFBS); Masters L.
 [96] "Anthony M. struck out swinging reached first on a wild pitch (2-2)."
 [97] "BEATTY struck out swinging reached first on an error by c; PACE advanced to second."
 [98] "Hagmeier struck out swinging reached first on a throwing error by c."
 [99] "GIRLIE struck out swinging reached first on a wild pitch; LEE scored. 14 1 14-1"
[100] "HILL struck out reached first on an error by c (1-2)."
[101] "DRAZIN struck out swinging reached first on a wild pitch."
[102] "2 1 42767 Pierce struck out swinging reached first on a wild pitch; Reinhardt advanced to third;
[103] "Reyes L. struck out swinging reached first on a wild pitch (2-2)."
[104] "2 2 42768 Canfield struck out swinging reached first on a wild pitch advanced to third on a thro
[105] "GAMBONE struck out swinging reached first on an error by c."
[106] "CLARK H. struck out swinging reached first on a wild pitch: MURPHY P. advanced to third."
[107] "Brown struck out swinging reached first on a passed ball (1-2); Hansis advanced to second."
[108] "Parsons C. struck out swinging reached first on a wild pitch."
[109] "Convissar struck out swinging reached first on a wild pitch (0-2 KKFK)."
[110] "E. Carosone struck out swinging reached first on a wild pitch (2-2)."
[111] "Wilkinson struck out swinging reached first on a wild pitch (1-2 KKBFS)."
[112] "BRUECK K. struck out swinging reached first on a passed ball (0-2)."
[113] "PIZZANO struck out swinging reached first on a wild pitch."
[114] "Taylor struck out swinging reached first on a passed ball."
[115] "GINGERICH C struck out swinging reached first on a wild pitch (2-2)."
[116] "Ross struck out swinging reached first on a throwing error by c (1-2 SBSS)."
[117] "Celaya B struck out swinging reached first on a throwing error by c advanced to second (2-2 FBBF
[118] "TURNER L. struck out swinging reached first on an error by c."
[119] "EMORY struck out swinging reached first on an error by c."
```

```r
season$play[grepl("outh", season$play)]
```

```
  [1] "South pinch ran for Minnatee."
  [2] "South advanced to second on a passed ball; Stockinger advanced to third on a passed ball."
  [3] "Yarbrough doubled to left center RBI; South scored unearned. 4 5 42830"
  [4] "Southall K. pinch ran for Buckley T.."
  [5] "Bencivenga grounded out to 3b: Southall K. advanced to second."
  [6] "Southall K. out at third c to 3b caught stealing."
  [7] "Southall K. to dh."
  [8] "Southall K. reached on a fielding error by ss."
  [9] "Bencivenga singled through the left side: Southall K. advanced to second."
 [10] "Miller A. hit by pitch: Bencivenga advanced to second: Southall K. advanced to third."
 [11] "1 5 42740 Whitt L. grounded out to 2b RBI: Miller A. advanced to second: Bencivenga advanced to
 [12] "Strouth hit by pitch; Duff advanced to second."
 [13] "Gonzalez reached on a fielder's choice to pitcher; Strouth advanced to second; Saunders out at t
 [14] "Smith walked; Gonzalez advanced to second; Strouth advanced to third."
 [15] "Strouth struck out looking."
 [16] "Mehr pinch hit for Strouth."
 [17] "South pinch ran for Drennan."
 [18] "South to dh."
 [19] "South singled to shortstop."
 [20] "South advanced to second: Edwards advanced to third on a wild pitch."
 [21] "Yarbrough reached on a fielder's choice: Boyd advanced to second: South out at second ss unassis
 [22] "Southall K. pinch ran for Bell M..""
 [23] "2 3 42769 Whitt L. singled to right field RBI (0-0); Southall K. advanced to second; Martinez M
 [24] "Palmer M. walked (3-0); Whitt L. advanced to second; Southall K. advanced to third."
```

```
[25] "Miller A. to c for Southall K.."
[26] "South pinch ran for Drennan."
[27] "Edwards out at first c to 2b SAC bunt: South advanced to second."
[28] "Boyd reached on a fielder's choice: Reid advanced to second: South out at third 3b unassisted."
[29] "South lined out to p."
[30] "Houthoofd pinch hit for Alderink."
[31] "Houthoofd walked (3-0 BBBB)."
[32] "Masek reached on a fielder's choice (2-2 BKBS); Houthoofd out at second 1b to ss."
[33] "Alderink to 2b for Houthoofd."
[34] "Southall K. reached on a fielding error by 2b (0-2)."
[35] "Southall K. out at second c to ss."
[36] "Southall K. flied out to cf; Palmer M. advanced to second; Hogue C. advanced to third."
[37] "Southall K. flied out to rf; Palmer M. advanced to second; Hogue C. advanced to third."
[38] "Southall K. flied out to rf (0-1)."
[39] "Strouth E. pinch hit for Mehr J.."
[40] "Strouth E. reached on an error by 1b (0-0)."
[41] "Houthoofd pinch ran for Rebar."
[42] "7 6 42922 Milligan singled advanced to second on a throwing error by 2b (0-0); Houthoofd scored.
[43] "Southall K. to rf for Rickey D.."
[44] "Southall K. grounded out to ss (1-1)."
[45] "South struck out swinging."
[46] "South singled to center field."
[47] "Yarbrough singled; South advanced to second."
[48] "Scott reached on a fielder's choice; Yarbrough advanced to second; South out at third c to 3b."
[49] "Baltazar pinch hit for South."
[50] "Strouth E. pinch hit for Schoenewald."
[51] "Strouth E. reached on a fielder's choice to shortstop; Johnson K. out at second ss to 2b."
[52] "Davenport B singled to right field; Strouth E. advanced to second."
[53] "Strouth E. to rf."
[54] "Southall K. pinch ran for Buckley T.."
[55] "Buckley T. to p for Southall K.."
[56] "Southall K. pinch ran for Hogue C.."
[57] "Rickey D. struck out swinging (1-2); Southall K. stole second."
[58] "Martinez C doubled down the lf line (1-1); Southall K. advanced to third."
[59] "0 2 0-2 Thompson C singled to center field 2 RBI (0-2); Martinez C scored; Southall K. scored."
[60] "Southall K. to dp."
[61] "Southall K. reached on a fielder's choice to second base (0-0); Miller A. out at second 2b unass
[62] "Hogue C. to dp for Southall K.."
[63] "Strouth E. to rf for Schoenewald."
[64] "Strouth E. grounded out to 3b."
[65] "Strouth E. walked."
[66] "4 2 42827 Davenport B walked RBI; Strouth E. advanced to second; Kowalski A. advanced to third;
[67] "Strouth E. struck out swinging."
[68] "Strouth E. reached on a fielder's choice; Duff L. out at second 3b to 2b."
[69] "Johnson K. doubled to right center; Strouth E. advanced to third."
[70] "6 5 42891 Lattin O. walked; Johnson K. advanced to third on a wild pitch; Strouth E. scored on a
[71] "Strouth E. grounded out to 3b."
[72] "Strouth E. flied out to lf."
[73] "Strouth E. singled to left field."
[74] "Davenport B singled to right field advanced to second; Strouth E. advanced to third on a throwing
[75] "Lattin O. singled to left field advanced to second 2 RBI; Davenport B scored; Strouth E. scored.
[76] "Strouth E. struck out swinging."
[77] "Southall K. pinch ran for Miller A.."
[78] "Whitt L. grounded out to c SAC bunt (0-1); Southall K. advanced to second."
```

```
 [79] "Miller A. to c for Southall K.."
 [80] "Houthoofd pinch hit for Cuyos."
 [81] "Houthoofd walked (3-2 KBKBBB)."
 [82] "0 11 0-11 Rebar singled down the lf line RBI (0-0): Houthoofd advanced to second: Dozier advance
 [83] "0 12 0-12 Russell singled to left field RBI (2-2 BBKF): Rebar advanced to second: Houthoofd advan
 [84] "0 13 0-13 Ober flied out to rf SF RBI (2-1 FBB): Houthoofd scored."
 [85] "Cuyos to ss for Houthoofd."
 [86] "Strouth E. struck out swinging."
 [87] "Strouth E. fouled out to 3b."
 [88] "Strouth E. reached on a fielder's choice; Nolan C. out at second ss to 2b."
 [89] "Houthoofd pinch hit for Cuyos."
 [90] "Houthoofd walked (3-2 BBKSBB): Dozier advanced to second."
 [91] "Cuyos to ss for Houthoofd."
 [92] "South pinch ran for Stracher."
 [93] "Cahill grounded into double play ss to 2b to 1b: South out on the play."
 [94] "Stracher to c for South."
 [95] "South pinch ran for Minnatee."
 [96] "Reid singled: South advanced to second."
 [97] "2 3 42769 Boyd singled RBI: Reid advanced to second: South scored."
 [98] "South pinch hit."
 [99] "South grounded out to p."
[100] "Southall K. pinch ran for Whitt L.."
[101] "Rickey D. advanced to second; Southall K. advanced to third on a passed ball."
[102] "Southall K. to dp."
[103] "Southall K. singled to right field (0-2)."
[104] "Bencivenga singled down the rf line (0-1); Southall K. advanced to second."
[105] "Rickey D. reached on a fielder's choice to first base (0-2); Bencivenga advanced to second; Sout
[106] "3 2 42796 Martinez C reached on a fielder's choice to third base RBI (1-0); Rickey D. advanced t
[107] "South to lf for Yarbrough."
[108] "Southall K. pinch ran for Buckley T.."
[109] "Bencivenga grounded out to 3b SAC bunt; Southall K. advanced to second."
[110] "Whitt L. grounded out to p; Bell M. advanced to second; Southall K. advanced to third."
[111] "Rickey D. singled to left field RBI; Bell M. advanced to third; Southall K. scored. 6 2 42888"
[112] "Buckley T. to p for Southall K.."
[113] "Strouth E. singled to left field (0-0)."
[114] "Strouth E. out at second c to ss caught stealing."
[115] "Strouth E. grounded out to 3b (0-0)."
[116] "Southall K. pinch ran for Miller A.."
[117] "Miller A. to c for Southall K.."
[118] "Southall K. pinch ran for Whitt L.."
[119] "Rickey D. grounded out to p SAC bunt (0-1 K); Southall K. advanced to second."
[120] "Whitt L. to 1b for Southall K.."
[121] "Strouth E. singled to third base (1-1 FB)."
[122] "Smith K. walked (3-1 BBKBB): Strouth E. advanced to second."
[123] "Strouth E. fouled out to 3b (1-1 BF)."
[124] "Strouth E. flied out to cf (0-1 K)."
[125] "Houthoofd pinch hit for Vidales."
[126] "Houthoofd struck out swinging."
[127] "Vidales to 1b for Houthoofd."
```

```r
# How many outs occurred because of play
season$outs_from_play <- as.numeric(grepl("[[:space:]]out[[:space:]]| popped |[[:space:]]out[[:punct:]]
    season$play))
table(season$outs_from_play)
```

```
    0     1 
93820 77692 
```

```
season$outs_from_play[grepl("double play", season$play)] <- 2
# Remove people who reached after striking out
season$outs_from_play[grepl("[[:space:]]out.*reached", season$play)] <- 0
season$outs_from_play[grepl("struck out reached", season$play)] <- 0

table(season$outs_from_play)
```

```
    0     1     2 
93938 76399  1175 
```

```
# Remove inning summaries
season$play[1:11]
```

```
 [1] "ENTZMINGER grounded out to ss."
 [2] "LESTER singled to second base."
 [3] "HUFSTETLER singled down the rf line; LESTER advanced to second."
 [4] "MADRID singled to left center; HUFSTETLER advanced to second; LESTER advanced to third."
 [5] "RODRIGUEZ flied into double play cf to c; LESTER out on the play."
 [6] "Soria Jenn grounded out to 3b."
 [7] "Butterfield grounded out to 2b."
 [8] "Aguirre K. singled through the left side."
 [9] "Gonzales. C. walked; Aguirre K. advanced to second."
[10] "Palacios S. grounded out to p."
[11] "R: 0 H: 3 E: 0 LOB: 2 0 0 0-0 R: 0 H: 1 E: 0 LOB: 2"
```

```
inning_summaries <- season[grepl("R: ", season$play), ]
season <- season[!grepl("R: ", season$play), ]

# Taking out non-plays in the data

season <- season[!((grepl("to[[:space:]][[:alpha:]][[:alpha:]][[:space:]]for", season$play))), ]
season <- season[!(grepl("to[[:space:]][[:digit:]][[:alpha:]][[:space:]]for", season$play)), ]
season <- season[!(grepl("to[[:space:]][[:alpha:]][[:space:]]for", season$play)), ]
season <- season[!(grepl("pinch[[:space:]]ran", season$play)), ]
season <- season[!(grepl("pinch[[:space:]]hit", season$play)), ]
season <- season[!((grepl("\\S[^[:space:]o][^0-9u][^t][[:space:]]to[[:space:]][[:alpha:]][[:alpha:]][[:
    season$play))), ]
season <- season[!(grepl("\\S[^[:space:]o][^u][^t][[:space:]]to[[:space:]][[:digit:]][[:alpha:]][[:punc
    season$play)), ]
season <- season[!(grepl("\\S[^[:space:]o][^u][^t][[:space:]]to[[:space:]][[:alpha:]][[:punct:]]", seas
    ]
season <- season[!((grepl("\\S[^u][^p][[:space:]]to[[:space:]][[:alpha:]][[:alpha:]][[:punct:]]", seaso
    ]
season <- season[!(grepl("\\S[^u][^p][[:space:]]to[[:space:]][[:digit:]][[:alpha:]][[:punct:]]", season
    ]
season <- season[!(grepl("\\S[^u][^p][[:space:]]to[[:space:]][[:alpha:]][[:punct:]]", season$play)),
    ]

season <- season[!(grepl("to[[:space:]]dh[[:punct:]]", season$play)), ]
season <- season[!(grepl("No[[:space:]]play[[:punct:]]", season$play)), ]
```

4

```
games <- unique(season$gameID)

game <- season[season$gameID == games[1], ]
dim(game)
```

```
[1] 54 34
```

```
# 71 plays Look at Inning ID
table(game$inningID)
```

```

 0  1  2  3  4 
10 13  7 13 11 
```

```
# Only five innings due to run-rule
innings <- unique(game$inningID)
# Look at plays for one inning

game$play[game$inningID == 3]
```

```
 [1] "RODRIGUEZ walked."
 [2] "PINEDO singled to right center; WATT advanced to second."
 [3] "PURDY fouled out to c."
 [4] "LUNA struck out looking."
 [5] "ABACHERLI reached on a fielder's choice; PINEDO out at second 2b to ss."
 [6] "Girard N. doubled to right field."
 [7] "Chilson N. singled to first base bunt; Peralta B. advanced to third."
 [8] "Becerra T. singled to catcher bunt; Chilson N. advanced to second."
 [9] "0 1 0-1 Soria Jenn singled through the right side advanced to second on the throw RBI; Becerra T.
[10] "0 2 0-2 Butterfield singled to catcher RBI advanced to second on the throw; Soria Jenn advanced to
[11] "0 3 0-3 Aguirre K. fouled out to rf RBI SF; Butterfield advanced to third; Soria Jenn scored."
[12] "0 4 0-4 Gonzales. C. singled to center field advanced to second on an error by ss RBI; Butterfield
[13] "Palacios S. grounded out to 2b."
```

```
game$play[game$inningID == 3 & game$outs_from_play == 1]
```

```
[1] "PURDY fouled out to c."
[2] "LUNA struck out looking."
[3] "ABACHERLI reached on a fielder's choice; PINEDO out at second 2b to ss."
[4] "0 1 0-1 Soria Jenn singled through the right side advanced to second on the throw RBI; Becerra T. a
[5] "0 3 0-3 Aguirre K. fouled out to rf RBI SF; Butterfield advanced to third; Soria Jenn scored."
[6] "Palacios S. grounded out to 2b."
```

```
game$outs_from_play
```

```
 [1] 1 0 0 0 2 1 1 0 0 1 0 1 1 0 1 0 0 0 1 0 1 0 1 1 1 1 0 1 1 1 0 0 1 1 1 0 0 0 1 0 1 0 1 0 1 2 1 1
[49] 0 0 0 0 0 0
```

```
# Look at outs in one inning
game$outs1[game$inningID == 0]
```

```
 [1] 1 0 0 0 1 1 1 0 0 1
```

```
game$outs_from_play[game$inningID == 3]
```

```
 [1] 0 0 1 1 1 0 0 0 1 0 1 0 1
```

```r
# Total outs in inning
game$total_outs_in_inning <- NA
for (i in innings) {
    game$total_outs_in_inning[game$inningID == i] <- cumsum(game$outs_from_play[game$inningID == i])
}
# Create an initial indicator of top and bottom
game$top_bottom[game$total_outs_in_inning < 4] <- "top"
```

```
Warning: Unknown or uninitialised column: 'top_bottom'.
```

```r
game$top_bottom[game$total_outs_in_inning > 3] <- "bottom"

# Create empty vector for Total outs before play
game$outs_before_play <- 0

# Total outs before play in top of Inning
for (i in innings) {
    n <- length(game$outs_before_play[game$inningID == i & game$top_bottom == "top"])
    game$outs_before_play[game$inningID == i & game$top_bottom == "top"][2:n] <- cumsum(game$outs_from_p
        i & game$top_bottom == "top"][1:n - 1])
    game$outs_before_play[game$inningID == i & game$top_bottom == "top"][1] <- 0
}
# Total outs before play in bottom of Inning
for (i in innings) {
    n <- length(game$outs_before_play[game$inningID == i & game$top_bottom == "bottom"])
    game$outs_before_play[game$inningID == i & game$top_bottom == "bottom"][2:n] <- cumsum(game$outs_fro
        i & game$top_bottom == "bottom"][1:n - 1])
    game$outs_before_play[game$inningID == i & game$top_bottom == "bottom"][1] <- 0
}

# Fix shift from top to bottom of inning

game$top_bottom[game$outs_before_play == 3] <- "bottom"

# Redo to get correct number of outs for each inning Top of Inning
for (i in innings) {
    n <- length(game$outs_before_play[game$inningID == i & game$top_bottom == "top"])
    game$outs_before_play[game$inningID == i & game$top_bottom == "top"][2:n] <- cumsum(game$outs_from_p
        i & game$top_bottom == "top"][1:n - 1])
    game$outs_before_play[game$inningID == i & game$top_bottom == "top"][1] <- 0
}
# Bottom of Inning
for (i in innings) {
    n <- length(game$outs_before_play[game$inningID == i & game$top_bottom == "bottom"])
    game$outs_before_play[game$inningID == i & game$top_bottom == "bottom"][2:n] <- cumsum(game$outs_fro
        i & game$top_bottom == "bottom"][1:n - 1])
    game$outs_before_play[game$inningID == i & game$top_bottom == "bottom"][1] <- 0
}




# Check

game[game$inningID == 0, c("play", "top_bottom", "outs_from_play", "outs_before_play", "total_outs_in_in
# A tibble: 10 × 5
                                                                                      play
                                                                                     <chr>
1                                                           ENTZMINGER grounded out to ss.
2                                                           LESTER singled to second base.
3                          HUFSTETLER singled down the rf line; LESTER advanced to second.
4  MADRID singled to left center; HUFSTETLER advanced to second; LESTER advanced to third.
5                        RODRIGUEZ flied into double play cf to c; LESTER out on the play.
6                                                           Soria Jenn grounded out to 3b.
7                                                          Butterfield grounded out to 2b.
8                                                Aguirre K. singled through the left side.
9                                      Gonzales. C. walked; Aguirre K. advanced to second.
10                                                          Palacios S. grounded out to p.
# ... with 4 more variables: top_bottom <chr>, outs_from_play <dbl>, outs_before_play <dbl>,
#   total_outs_in_inning <dbl>
```

```
game[game$inningID == 1, c("play", "top_bottom", "outs_from_play", "outs_before_play", "total_outs_in_i
```

```
# A tibble: 13 × 5
                                                                    play top_bottom outs_from_play
                                                                   <chr>      <chr>          <dbl>
1                                          PINEDO singled to left field.        top              0
2                                                 PURDY flied out to lf.        top              1
3                                                  LUNA popped up to ss.        top              1
4             ABACHERLI singled to left field; AMREIN advanced to second.       top              0
5                                         ENTZMINGER grounded out to ss.        top              1
6                                       Stahm M. singled to right field.     bottom              0
7  Girard N. reached on a fielder's choice; Davis A. advanced to second.     bottom              0
8                                                  Davis A. stole third.     bottom              0
9                                        Chilson N. struck out swinging.     bottom              1
10                                               Girard N. stole second.     bottom              0
11                                           Becerra T. flied out to lf.     bottom              1
12                                                    Soria Jenn walked.     bottom              0
13                                         Blair T. struck out swinging.     bottom              1
# ... with 2 more variables: outs_before_play <dbl>, total_outs_in_inning <dbl>
```

```
game[game$inningID == 2, c("play", "top_bottom", "outs_from_play", "outs_before_play", "total_outs_in_i
```

```
# A tibble: 7 × 5
                                                                          play top_bottom
                                                                         <chr>      <chr>
1                                                      LESTER flied out to lf.        top
2                                                  HUFSTETLER flied out to rf.        top
3                                                      MADRID popped up to ss.        top
4                                            Aguirre K. singled up the middle.     bottom
5 Gonzales. C. out at first 3b to 2b bunt SAC; Stankiewicz advanced to second.     bottom
6                                                 Palacios S. popped up to 1b.     bottom
7                                                  Stahm M. grounded out to p.     bottom
# ... with 3 more variables: outs_from_play <dbl>, outs_before_play <dbl>,
#   total_outs_in_inning <dbl>
```

```
game[game$inningID == 3, c("play", "top_bottom", "outs_from_play", "outs_before_play", "total_outs_in_i
```

```
# A tibble: 13 × 5
                                                                                                   play
                                                                                                  <chr>
```

```
1                                                                                     RODRIGUEZ walked.
2                                              PINEDO singled to right center; WATT advanced to second.
3                                                                                PURDY fouled out to c.
4                                                                              LUNA struck out looking.
5                               ABACHERLI reached on a fielder's choice; PINEDO out at second 2b to ss.
6                                                                     Girard N. doubled to right field.
7                                  Chilson N. singled to first base bunt; Peralta B. advanced to third.
8                                    Becerra T. singled to catcher bunt; Chilson N. advanced to second.
9  0 1 0-1 Soria Jenn singled through the right side advanced to second on the throw RBI; Becerra T. ad
10 0 2 0-2 Butterfield singled to catcher RBI advanced to second on the throw; Soria Jenn advanced to t
11        0 3 0-3 Aguirre K. fouled out to rf RBI SF; Butterfield advanced to third; Soria Jenn scored.
12 0 4 0-4 Gonzales. C. singled to center field advanced to second on an error by ss RBI; Butterfield s
13                                                                      Palacios S. grounded out to 2b.
# ... with 4 more variables: top_bottom <chr>, outs_from_play <dbl>, outs_before_play <dbl>,
#   total_outs_in_inning <dbl>
```

```r
game[game$inningID == 4, c("play", "top_bottom", "outs_from_play", "outs_before_play", "total_outs_in_i
```

```
# A tibble: 11 × 5
                                                                                            play
                                                                                           <chr>
1                                                          ENTZMINGER reached on an error by ss.
2                                                                     LESTER struck out looking.
3                    HUFSTETLER hit into double play lf to 2b to 1b; ENTZMINGER out on the play.
4                                                                   Stahm M. grounded out to 3b.
5                                                                     Girard N. flied out to cf.
6                                                                             Spiel Abby walked.
7                                                                       Spiel Abby stole second.
8                                                                             Becerra T. walked.
9          0 7 0-7 Soria Jenn homered to left field 3 RBI; Becerra T. scored; Spiel Abby scored.
10                                                  Butterfield reached on a muffed throw by 1b.
11 0 8 0-8 Aguirre K. reached on an error by rf advanced to second; Butterfield scored unearned.
# ... with 4 more variables: top_bottom <chr>, outs_from_play <dbl>, outs_before_play <dbl>,
#   total_outs_in_inning <dbl>
```

5

```r
games <- unique(season$gameID)
# games<-sample(games,100) Total outs in inning
season$total_outs_in_inning <- NA
for (k in games) {
    innings <- unique(season$inningID[season$gameID == k])
    for (i in innings) {
        season$total_outs_in_inning[season$inningID == i & season$gameID == k] <- cumsum(season$outs_fr
            i & season$gameID == k])
    }
}

season$top_bottom <- NA
season$top_bottom[season$total_outs_in_inning < 4] <- "top"
season$top_bottom[season$total_outs_in_inning > 3] <- "bottom"

# Total outs before play
```

```r
season$outs_before_play <- 0

# Top of Inning
for (k in games) {
    innings <- unique(season$inningID[season$gameID == k])
    for (i in innings) {
        n <- length(season$outs_before_play[season$inningID == i & season$gameID == k & season$top_botto
            "top"])
        if (n == 1) {
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][1] <- 0
        } else {
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][2:n] <- cumsum(season$outs_from_play[season$inningID == i & season$gameID == k &
                season$top_bottom == "top"][1:n - 1])
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][1] <- 0
        }
    }
}
# Bottom of Inning
for (k in games) {
    innings <- unique(season$inningID[season$gameID == k])
    for (i in innings) {
        n <- length(season$outs_before_play[season$inningID == i & season$gameID == k & season$top_botto
            "bottom"])
        if (n == 1) {
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][1] <- 0
        } else {
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][2:n] <- cumsum(season$outs_from_play[season$inningID == i & season$gameID ==
                k & season$top_bottom == "bottom"][1:n - 1])
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][1] <- 0
        }
    }
}

# Fix transition from top to bottom
season$top_bottom[season$outs_before_play == 3] <- "bottom"


# Top of Inning
for (k in games) {
    innings <- unique(season$inningID[season$gameID == k])
    for (i in innings) {
        n <- length(season$outs_before_play[season$inningID == i & season$gameID == k & season$top_botto
            "top"])
        if (n == 1) {
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][1] <- 0
        } else {
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][2:n] <- cumsum(season$outs_from_play[season$inningID == i & season$gameID == k &
                season$top_bottom == "top"][1:n - 1])
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][1] <- 0
        }
    }
}
# Bottom of Inning
for (k in games) {
    innings <- unique(season$inningID[season$gameID == k])
    for (i in innings) {
        n <- length(season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bott
            "bottom"])
        if (n == 1) {
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][1] <- 0
        } else {
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][2:n] <- cumsum(season$outs_from_play[season$inningID == i & season$gameID ==
                k & season$top_bottom == "bottom"][1:n - 1])
            season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][1] <- 0
        }
    }
}

# Check
game[game$inningID == 4, c("play", "top_bottom", "outs_from_play", "outs_before_play", "total_outs_in_i
    season[season$gameID == games[1] & season$inningID == 4, c("play", "top_bottom", "outs_from_play",
        "outs_before_play", "total_outs_in_inning")]
```

```
      play top_bottom outs_from_play outs_before_play total_outs_in_inning
 [1,] TRUE       TRUE           TRUE             TRUE                 TRUE
 [2,] TRUE       TRUE           TRUE             TRUE                 TRUE
 [3,] TRUE       TRUE           TRUE             TRUE                 TRUE
 [4,] TRUE       TRUE           TRUE             TRUE                 TRUE
 [5,] TRUE       TRUE           TRUE             TRUE                 TRUE
 [6,] TRUE       TRUE           TRUE             TRUE                 TRUE
 [7,] TRUE       TRUE           TRUE             TRUE                 TRUE
 [8,] TRUE       TRUE           TRUE             TRUE                 TRUE
 [9,] TRUE       TRUE           TRUE             TRUE                 TRUE
[10,] TRUE       TRUE           TRUE             TRUE                 TRUE
[11,] TRUE       TRUE           TRUE             TRUE                 TRUE
```

```
# Look at Random Game
season[season$gameID == games[24] & season$inningID == 3, c("play", "top_bottom", "outs_from_play", "ou
    "total_outs_in_inning")]
```

```
# A tibble: 13 × 5
                                                                                                  play
                                                                                                 <chr>
1                                                                                Hayes H singled bunt.
2                                           Moran M singled up the middle: Hayes H advanced to second.
```

```
3                 Field E singled up the middle: Moran M advanced to second: Hayes H advanced to third.
4  Tolle M reached on a fielder's choice RBI: Field E out at second ss to 2b: Moran M advanced to third
5                                                                                 Tolle M stole second.
6               Good M grounded out to 2b SAC RBI: Tolle M advanced to third: Moran M scored. 5 0 36647
7                                                                                        Ford J walked.
8                                                                                   Ford J stole second.
9                                                                            Newton T grounded out to p.
10                                                                            Whitley J lined out to 3b.
11                                                                         Hall R doubled to left center.
12                                                                          Maddox S struck out swinging.
13                                                          Slowinski M grounded out to 1b unassisted.
# ... with 4 more variables: top_bottom <chr>, outs_from_play <dbl>, outs_before_play <dbl>,
#   total_outs_in_inning <dbl>
```

```
summary(season$outs_before_play)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   0.000   1.000   0.958   2.000   3.000 
```

```
table(season$outs_before_play)
```

```

    0     1     2     3 
47357 45028 41700    12 
```

```
season$gameID[season$outs_before_play == 3]
```

```
 [1] 4014180 4014830 4017578 4019409 4032124 4032124 4032124 4032124 4032124 4044036 4044036 4044036
```

```
# Games found to be problematic after calculating outs
season <- season[!(season$gameID == 4019409), ]
season <- season[!(season$gameID == 4011385), ]
season <- season[!(season$gameID == 4014180), ]
season <- season[!(season$gameID == 4017578), ]
season <- season[!(season$gameID == 4014830), ]
season <- season[!(season$gameID == 4021185), ]
season <- season[!(season$gameID == 4032124), ]
season <- season[!(season$gameID == 4044036), ]
```

```
save(season, file = "season.rda")
```

6

```
# Create a measure of runs

sum(grepl("RBI", season$play))
```

```
[1] 12986
```

```
sum(grepl("scored", season$play))
```

```
[1] 13663
```

```
# There are 13772-13070 plays in which runs were scored not accounted for just by grepl('RBI')
# because some are scored, unearned
# Show how many runs come from one play
twoRBIs <- sum(grepl("2[[:space:]]RBI[[:space:]]", season$play))
twoRBIp <- sum(grepl("2[[:space:]]RBI[[:punct:]]", season$play))
threeRBIs <- sum(grepl("3[[:space:]]RBI[[:space:]]", season$play))
threeRBIp <- sum(grepl("3[[:space:]]RBI[[:punct:]]", season$play))
fourRBIs <- sum(grepl("4[[:space:]]RBI[[:space:]]", season$play))
fourRBIp <- sum(grepl("4[[:space:]]RBI[[:punct:]]", season$play))

totalRBIs <- sum(grepl("[[:space:]]RBI[[:space:]]", season$play)) + sum(grepl("[[:space:]]RBI[[:punct:]
    season$play))
totalRBIs
```

```
[1] 12986
```

```
totalScoringPlays <- sum(grepl("[[:space:]]scored[[:space:]]", season$play)) + sum(grepl("[[:space:]]sco
    season$play))

miscRuns <- totalScoringPlays - totalRBIs
miscRuns
```

```
[1] 1009
```

```
# How many runs occurred because of play

season$runs_from_play <- as.numeric(grepl("[[:space:]]scored[[:space:]]|[[:space:]]scored[[:punct:]]",
    season$play))


table(season$runs_from_play)
```

```

     0      1 
119928  13663 
```

```
season$runs_from_play[grepl("2[[:space:]]RBI[[:space:]]", season$play)] <- 2
season$runs_from_play[grepl("2[[:space:]]RBI[[:punct:]]", season$play)] <- 2
season$runs_from_play[grepl("3[[:space:]]RBI[[:space:]]", season$play)] <- 3
season$runs_from_play[grepl("3[[:space:]]RBI[[:punct:]]", season$play)] <- 3
season$runs_from_play[grepl("4[[:space:]]RBI[[:space:]]", season$play)] <- 4
season$runs_from_play[grepl("4[[:space:]]RBI[[:punct:]]", season$play)] <- 4

season$runs_from_play[grepl("[^0-9][[:space:]]RBI[[:space:]]", season$play)] <- 1
season$runs_from_play[grepl("[^0-9][[:space:]]RBI[[:punct:]]", season$play)] <- 1
table(season$runs_from_play)
```

```

     0      1      2      3      4 
118801  11559   2520    577    134 
```

7

```
games <- unique(season$gameID)
games <- unique(season$gameID)
```

```r
game <- season[season$gameID == games[1], ]
dim(game)
```

```
[1] 54 38
```

```r
innings <- unique(game$inningID)
# Look at plays for one inning

game$play[game$inningID == 3]
```

```
 [1] "RODRIGUEZ walked."
 [2] "PINEDO singled to right center; WATT advanced to second."
 [3] "PURDY fouled out to c."
 [4] "LUNA struck out looking."
 [5] "ABACHERLI reached on a fielder's choice; PINEDO out at second 2b to ss."
 [6] "Girard N. doubled to right field."
 [7] "Chilson N. singled to first base bunt; Peralta B. advanced to third."
 [8] "Becerra T. singled to catcher bunt; Chilson N. advanced to second."
 [9] "0 1 0-1 Soria Jenn singled through the right side advanced to second on the throw RBI; Becerra T.
[10] "0 2 0-2 Butterfield singled to catcher RBI advanced to second on the throw; Soria Jenn advanced to
[11] "0 3 0-3 Aguirre K. fouled out to rf RBI SF; Butterfield advanced to third; Soria Jenn scored."
[12] "0 4 0-4 Gonzales. C. singled to center field advanced to second on an error by ss RBI; Butterfield
[13] "Palacios S. grounded out to 2b."
```

```r
game$play[game$inningID == 3 & game$runs_from_play == 1]
```

```
[1] "0 1 0-1 Soria Jenn singled through the right side advanced to second on the throw RBI; Becerra T. a
[2] "0 2 0-2 Butterfield singled to catcher RBI advanced to second on the throw; Soria Jenn advanced to
[3] "0 3 0-3 Aguirre K. fouled out to rf RBI SF; Butterfield advanced to third; Soria Jenn scored."
[4] "0 4 0-4 Gonzales. C. singled to center field advanced to second on an error by ss RBI; Butterfield
```

```r
game$runs_from_play
```

```
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0
[49] 0 0 0 3 0 1
```

```r
# Look at runs in one inning
game$runs_from_play[game$inningID == 0]
```

```
 [1] 0 0 0 0 0 0 0 0 0 0
```

```r
game$runs_from_play[game$inningID == 3]
```

```
 [1] 0 0 0 0 0 0 0 0 1 1 1 1 0
```

```r
# Total runs in inning
game$total_runs_in_inning <- NA
for (i in innings) {
    game$total_runs_in_inning[game$inningID == i] <- cumsum(game$runs_from_play[game$inningID == i])
}
# Create an initial indicator of top and bottom
game$top_bottom[game$total_outs_in_inning < 4] <- "top"
game$top_bottom[game$total_outs_in_inning > 3] <- "bottom"

# Create empty vector for Total outs before play
game$runs_before_play <- 0

# Total runs before play in top of Inning
```

```
for (i in innings) {
    n <- length(game$outs_before_play[game$inningID == i & game$top_bottom == "top"])
    game$runs_before_play[game$inningID == i & game$top_bottom == "top"][2:n] <- cumsum(game$runs_from_
        i & game$top_bottom == "top"][1:n - 1])
    game$runs_before_play[game$inningID == i & game$top_bottom == "top"][1] <- 0
}
# Total outs before play in bottom of Inning
for (i in innings) {
    n <- length(game$outs_before_play[game$inningID == i & game$top_bottom == "bottom"])
    game$runs_before_play[game$inningID == i & game$top_bottom == "bottom"][2:n] <- cumsum(game$runs_fro
        i & game$top_bottom == "bottom"][1:n - 1])
    game$runs_before_play[game$inningID == i & game$top_bottom == "bottom"][1] <- 0
}



# Top of Inning
for (i in innings) {
    n <- length(game$outs_before_play[game$inningID == i & game$top_bottom == "top"])
    game$runs_before_play[game$inningID == i & game$top_bottom == "top"][2:n] <- cumsum(game$runs_from_
        i & game$top_bottom == "top"][1:n - 1])
    game$runs_before_play[game$inningID == i & game$top_bottom == "top"][1] <- 0
}
# Bottom of Inning
for (i in innings) {
    n <- length(game$outs_before_play[game$inningID == i & game$top_bottom == "bottom"])
    game$runs_before_play[game$inningID == i & game$top_bottom == "bottom"][2:n] <- cumsum(game$runs_fro
        i & game$top_bottom == "bottom"][1:n - 1])
    game$runs_before_play[game$inningID == i & game$top_bottom == "bottom"][1] <- 0
}



# Manually inseR to make sure things look right

game[game$inningID == 0, c("play", "top_bottom", "runs_from_play", "runs_before_play", "total_runs_in_i
```

```
# A tibble: 10 × 5
                                                                                      play
                                                                                     <chr>
1                                                           ENTZMINGER grounded out to ss.
2                                                           LESTER singled to second base.
3                          HUFSTETLER singled down the rf line; LESTER advanced to second.
4  MADRID singled to left center; HUFSTETLER advanced to second; LESTER advanced to third.
5                        RODRIGUEZ flied into double play cf to c; LESTER out on the play.
6                                                           Soria Jenn grounded out to 3b.
7                                                          Butterfield grounded out to 2b.
8                                                Aguirre K. singled through the left side.
9                                      Gonzales. C. walked; Aguirre K. advanced to second.
10                                                          Palacios S. grounded out to p.
# ... with 4 more variables: top_bottom <chr>, runs_from_play <dbl>, runs_before_play <dbl>,
#   total_runs_in_inning <dbl>
```

```
game[game$inningID == 1, c("play", "top_bottom", "runs_from_play", "runs_before_play", "total_runs_in_i
```

```
# A tibble: 13 × 5
                                                                    play top_bottom runs_from_play
                                                                   <chr>      <chr>          <dbl>
1                                          PINEDO singled to left field.        top              0
2                                                 PURDY flied out to lf.        top              0
3                                                  LUNA popped up to ss.        top              0
4            ABACHERLI singled to left field; AMREIN advanced to second.        top              0
5                                         ENTZMINGER grounded out to ss.        top              0
6                                       Stahm M. singled to right field.        top              0
7  Girard N. reached on a fielder's choice; Davis A. advanced to second.        top              0
8                                                  Davis A. stole third.        top              0
9                                        Chilson N. struck out swinging.     bottom              0
10                                               Girard N. stole second.     bottom              0
11                                           Becerra T. flied out to lf.     bottom              0
12                                                    Soria Jenn walked.     bottom              0
13                                         Blair T. struck out swinging.     bottom              0
# ... with 2 more variables: runs_before_play <dbl>, total_runs_in_inning <dbl>
```

```
game[game$inningID == 2, c("play", "top_bottom", "runs_from_play", "runs_before_play", "total_runs_in_i
```

```
# A tibble: 7 × 5
                                                                          play top_bottom
                                                                         <chr>      <chr>
1                                                      LESTER flied out to lf.        top
2                                                  HUFSTETLER flied out to rf.        top
3                                                      MADRID popped up to ss.        top
4                                            Aguirre K. singled up the middle.        top
5 Gonzales. C. out at first 3b to 2b bunt SAC; Stankiewicz advanced to second.     bottom
6                                                 Palacios S. popped up to 1b.     bottom
7                                                  Stahm M. grounded out to p.     bottom
# ... with 3 more variables: runs_from_play <dbl>, runs_before_play <dbl>,
#   total_runs_in_inning <dbl>
```

```
game[game$inningID == 3, c("play", "top_bottom", "runs_from_play", "runs_before_play", "total_runs_in_i
```

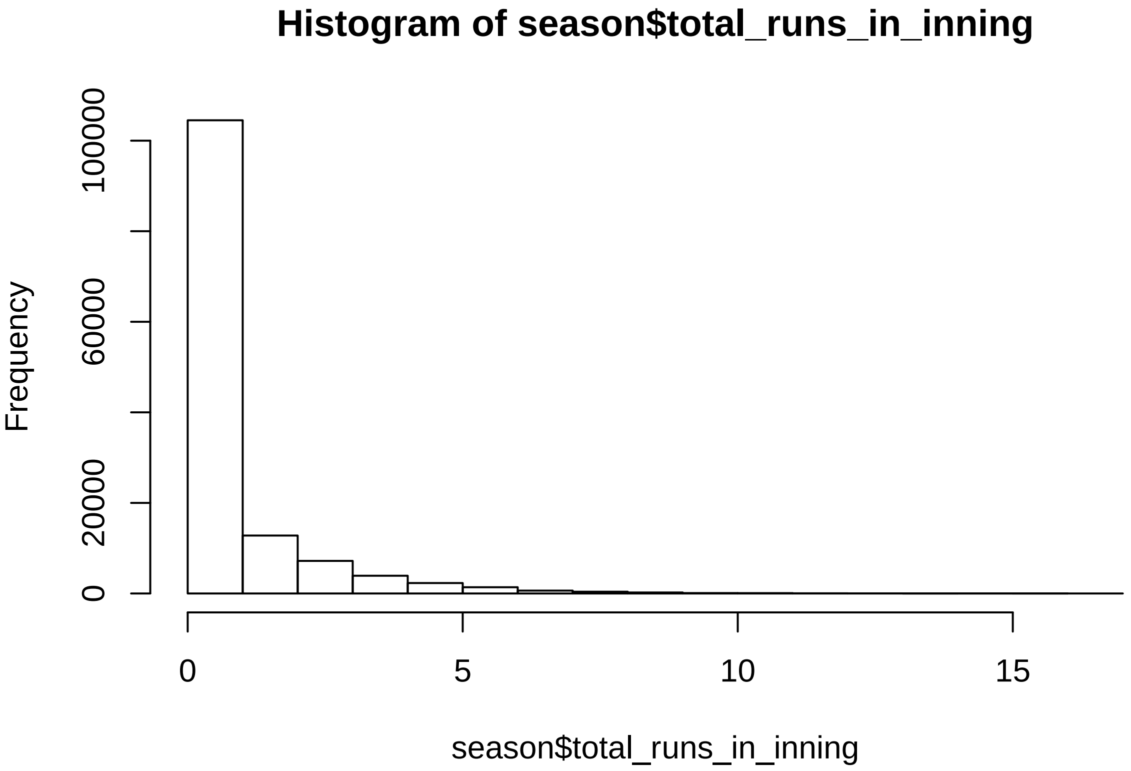
```
# A tibble: 13 × 5
                                                                                                   play
                                                                                                  <chr>
1                                                                                     RODRIGUEZ walked.
2                                              PINEDO singled to right center; WATT advanced to second.
3                                                                                PURDY fouled out to c.
4                                                                              LUNA struck out looking.
5                               ABACHERLI reached on a fielder's choice; PINEDO out at second 2b to ss.
6                                                                     Girard N. doubled to right field.
7                                  Chilson N. singled to first base bunt; Peralta B. advanced to third.
8                                    Becerra T. singled to catcher bunt; Chilson N. advanced to second.
9  0 1 0-1 Soria Jenn singled through the right side advanced to second on the throw RBI; Becerra T. ad
10 0 2 0-2 Butterfield singled to catcher RBI advanced to second on the throw; Soria Jenn advanced to t
11        0 3 0-3 Aguirre K. fouled out to rf RBI SF; Butterfield advanced to third; Soria Jenn scored.
12 0 4 0-4 Gonzales. C. singled to center field advanced to second on an error by ss RBI; Butterfield s
13                                                                      Palacios S. grounded out to 2b.
# ... with 4 more variables: top_bottom <chr>, runs_from_play <dbl>, runs_before_play <dbl>,
#   total_runs_in_inning <dbl>
```

```
game[game$inningID == 4, c("play", "top_bottom", "runs_from_play", "runs_before_play", "total_runs_in_i
```

```
# A tibble: 11 × 5
                                                                                            play
                                                                                           <chr>
1                                                          ENTZMINGER reached on an error by ss.
2                                                                     LESTER struck out looking.
3                    HUFSTETLER hit into double play lf to 2b to 1b; ENTZMINGER out on the play.
4                                                                   Stahm M. grounded out to 3b.
5                                                                     Girard N. flied out to cf.
6                                                                             Spiel Abby walked.
7                                                                       Spiel Abby stole second.
8                                                                             Becerra T. walked.
9          0 7 0-7 Soria Jenn homered to left field 3 RBI; Becerra T. scored; Spiel Abby scored.
10                                                  Butterfield reached on a muffed throw by 1b.
11 0 8 0-8 Aguirre K. reached on an error by rf advanced to second; Butterfield scored unearned.
# ... with 4 more variables: top_bottom <chr>, runs_from_play <dbl>, runs_before_play <dbl>,
#   total_runs_in_inning <dbl>
```

8

```
games <- unique(season$gameID)
# games<-sample(games,100) Total runs in inning
season$total_runs_in_inning <- NA
for (k in games) {
    innings <- unique(season$inningID[season$gameID == k])
    for (i in innings) {
        season$total_runs_in_inning[season$inningID == i & season$gameID == k] <- cumsum(season$runs_fr
            i & season$gameID == k])
    }
}




# Total runs before play
season$runs_before_play <- 0

# Top of Inning
for (k in games) {
    innings <- unique(season$inningID[season$gameID == k])
    for (i in innings) {
        n <- length(season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bott
            "top"])
        if (n == 1) {
            season$runs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][1] <- 0
        } else {
            season$runs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][2:n] <- cumsum(season$runs_from_play[season$inningID == i & season$gameID == k &
                season$top_bottom == "top"][1:n - 1])
            season$runs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][1] <- 0
        }
    }
}
# Bottom of Inning
for (k in games) {
    innings <- unique(season$inningID[season$gameID == k])
    for (i in innings) {
        n <- length(season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bott
            "bottom"])
        if (n == 1) {
            season$runs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][1] <- 0
        } else {
            season$runs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][2:n] <- cumsum(season$runs_from_play[season$inningID == i & season$gameID ==
                k & season$top_bottom == "bottom"][1:n - 1])
            season$runs_before_play[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][1] <- 0
        }
    }
}




# Check
hist(season$total_runs_in_inning)
```



```
summary(season$total_runs_in_inning)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00    0.00    0.88    1.00   17.00
```

```
save(season, file = "season.rda")
```

9

```
# Creating Indicators for baserunners



season$firstbase <- NA
season$firstbase[grepl("[[:space:]]singled", season$play)] <- 1
season$firstbase[grepl("[[:space:]]walked", season$play)] <- 1
season$firstbase[grepl("[[:space:]]reached[[:space:]]on[[:space:]]an[[:space:]]error", season$play)] <-
season$firstbase[grepl("[[:space:]]hit[[:space:]]by[[:space:]]pitch", season$play)] <- 1
season$firstbase[grepl("[[:space:]]catcher's[[:space:]]interference", season$play)] <- 1
season$firstbase[grepl("[[:space:]]reached[[:space:]]first[[:space:]]on[[:space:]]a[[:space:]]passed[[:s
    season$play)] <- 1
season$firstbase[grepl("[[:space:]]advanced[[:space:]]to[[:space:]]first", season$play)] <- 1
season$firstbase[grepl("[[:space:]]out[[:space:]]at[[:space:]]first", season$play)] <- NA
season$firstbase[grepl("[[:space:]]stole[[:space:]]second", season$play)] <- NA
season$firstbase[grepl("[[:space:]]homered", season$play)] <- 0
season$firstbase[grepl("[[:space:]]doubled", season$play)] <- NA
season$firstbase[grepl("[[:space:]]reached[[:space:]]on[[:space:]]a[[:space:]]fielder's[[:space:]]choice
    season$play)] <- 1
season$firstbase[grepl("advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]the[[:space:]]thre
    season$play)] <- NA
season$firstbase[grepl("singled[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]
    season$play)] <- NA
season$firstbase[grepl("base[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an
    season$play)] <- NA
season$firstbase[grepl("field[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an
    season$play)] <- NA
season$firstbase[grepl("center[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]a
    season$play)] <- NA
season$firstbase[grepl("middle[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]a
    season$play)] <- NA
season$firstbase[grepl("side[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an
    season$play)] <- NA
season$firstbase[grepl("choice[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]a
    season$play)] <- NA
season$firstbase[grepl("error[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an
    season$play)] <- NA
season$firstbase[grepl("by[[:space:]][[:alpha:]][[:alpha:]][[:space:]]advanced[[:space:]]to[[:space:]]se
    season$play)] <- NA
season$firstbase[grepl("by[[:space:]][[:digit:]][[:alpha:]][[:space:]]advanced[[:space:]]to[[:space:]]se
    season$play)] <- NA
season$firstbase[grepl("by[[:space:]][[:alpha:]][[:space:]]advanced[[:space:]]to[[:space:]]second", seas
```

```r
season$secondbase <- NA
season$secondbase[grepl("[[:space:]]advanced[[:space:]]to[[:space:]]second", season$play)] <- 1
season$secondbase[grepl("[[:space:]]doubled", season$play)] <- 1
season$secondbase[grepl("[[:space:]]out[[:space:]]at[[:space:]]second", season$play)] <- NA
season$secondbase[grepl("[[:space:]]stole[[:space:]]third", season$play)] <- NA
season$secondbase[grepl("[[:space:]]homered", season$play)] <- 0
season$secondbase[grepl("[[:space:]]tripled", season$play)] <- NA
season$secondbase[grepl("placed[[:space:]]on[[:space:]]second", season$play)] <- 1
season$secondbase[grepl("[[:space:]]stole[[:space:]]second[[:space:]][^a]", season$play)] <- 1
season$secondbase[grepl("[[:space:]]stole[[:space:]]second[^.[:space:]]", season$play)] <- 1


season$thirdbase <- NA
season$thirdbase[grepl("[[:space:]]advanced[[:space:]]to[[:space:]]third", season$play)] <- 1
season$thirdbase[grepl("[[:space:]]tripled", season$play)] <- 1
season$thirdbase[grepl("[[:space:]]out[[:space:]]at[[:space:]]third", season$play)] <- NA
season$thirdbase[grepl("[[:space:]]homered", season$play)] <- 0
season$thirdbase[grepl("scored[[:space:]]on[[:space:]]a[[:space:]]wild[[:space:]]pitch", season$play)] <
season$thirdbase[grepl("scored[[:space:]]on[[:space:]]an[[:space:]]illegal[[:space:]]pitch", season$pla
season$thirdbase[grepl("scored[[:space:]]on[[:space:]]a[[:space:]]passed[[:space:]]ball", season$play)]
season$thirdbase[grepl("[[:space:]]stole[[:space:]]second[^.[:space:]]", season$play)] <- 1
season$thirdbase[grepl("[[:space:]]stole[[:space:]]second[[:space:]][^a]", season$play)] <- 1



# Special Conditions


season$firstbase[grepl("[[:space:]]singled", season$play) & is.na(season$secondbase) & is.na(season$thi
season$firstbase[grepl("[[:space:]]singled", season$play) & season$secondbase == 1 & is.na(season$thirdb
season$firstbase[grepl("[[:space:]]singled", season$play) & season$secondbase == 1 & is.na(season$thirdb

season$firstbase[grepl("[[:space:]]walked", season$play) & is.na(season$secondbase) & is.na(season$thir

season$firstbase[grepl("[[:space:]]walked", season$play) & season$secondbase == 1 & is.na(season$thirdba

season$firstbase[grepl("[[:space:]]reached[[:space:]]on[[:space:]]an[[:space:]]error[[:space:]]", seaso
    is.na(season$secondbase) & is.na(season$thirdbase)] <- NA

season$firstbase[grepl("[[:space:]]reached[[:space:]]on[[:space:]]an[[:space:]]error[[:space:]]", seaso
    season$secondbase == 1 & is.na(season$thirdbase)] <- NA

season$firstbase[grepl("[[:space:]]hit[[:space:]]by[[:space:]]pitch", season$play) & is.na(season$secon
    is.na(season$thirdbase)] <- NA
season$firstbase[grepl("[[:space:]]hit[[:space:]]by[[:space:]]pitch", season$play) & season$secondbase =
    1 & is.na(season$thirdbase)] <- NA

season$firstbase[grepl("[[:space:]]catcher's[[:space:]]interference", season$play) & is.na(season$secon
    is.na(season$thirdbase)] <- NA
season$firstbase[grepl("[[:space:]]catcher's[[:space:]]interference", season$play) & season$secondbase =
    1 & is.na(season$thirdbase)] <- NA

season$firstbase[grepl("[[:space:]]reached[[:space:]]first[[:space:]]on[[:space:]]a[[:space:]]passed[[:
    season$play) & is.na(season$secondbase) & is.na(season$thirdbase)] <- NA


season$secondbase[grepl("[[:space:]]stole[[:space:]]second[.]", season$play)] <- NA

# Indicators for Special Conditions

season$temp <- 0

season$temp[grepl("[[:space:]]singled", season$play) & is.na(season$secondbase) & is.na(season$thirdbas


season$temp[grepl("[[:space:]]walked", season$play) & is.na(season$secondbase) & is.na(season$thirdbase


season$temp[grepl("[[:space:]]reached[[:space:]]on[[:space:]]an[[:space:]]error[[:space:]]", season$pla
    is.na(season$secondbase) & is.na(season$thirdbase)] <- 1


season$temp[grepl("[[:space:]]hit[[:space:]]by[[:space:]]pitch", season$play) & is.na(season$secondbase
    is.na(season$thirdbase)] <- 1


season$temp[grepl("[[:space:]]catcher's[[:space:]]interference", season$play) & is.na(season$secondbase
    is.na(season$thirdbase)] <- 1


season$temp[grepl("[[:space:]]reached[[:space:]]first[[:space:]]on[[:space:]]a[[:space:]]passed[[:space
    season$play) & is.na(season$secondbase) & is.na(season$thirdbase)] <- 1


season$temp[grepl("[[:space:]]stole[[:space:]]second[.]", season$play)] <- 2

season$temp[grepl("[[:space:]]stole[[:space:]]third[.]", season$play)] <- 3

season$temp[grepl("[[:space:]]singled", season$play) & season$secondbase == 1 & is.na(season$thirdbase)

season$temp[grepl("[[:space:]]catcher's[[:space:]]interference", season$play) & season$secondbase ==
    1 & is.na(season$thirdbase)] <- 4

season$temp[grepl("[[:space:]]hit[[:space:]]by[[:space:]]pitch", season$play) & season$secondbase ==
    1 & is.na(season$thirdbase)] <- 4

season$temp[grepl("[[:space:]]reached[[:space:]]on[[:space:]]an[[:space:]]error[[:space:]]", season$pla
    season$secondbase == 1 & is.na(season$thirdbase)] <- 4

season$temp[grepl("[[:space:]]walked", season$play) & season$secondbase == 1 & is.na(season$thirdbase)]
```

```
season$temp[grepl("advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]the[[:space:]]throw",
    season$play) & is.na(season$thirdbase)] <- 5
season$temp[grepl("singled[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an[[
    season$play) & is.na(season$thirdbase)] <- 5
season$temp[grepl("base[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an[[:spa
    season$play) & is.na(season$thirdbase)] <- 5
season$temp[grepl("field[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an[[:sp
    season$play) & is.na(season$thirdbase)] <- 5
season$temp[grepl("center[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an[[:s
    season$play) & is.na(season$thirdbase)] <- 5
season$temp[grepl("middle[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an[[:s
    season$play) & is.na(season$thirdbase)] <- 5
season$temp[grepl("side[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an[[:spa
    season$play) & is.na(season$thirdbase)] <- 5
season$temp[grepl("choice[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an[[:s
    season$play) & is.na(season$thirdbase)] <- 5
season$temp[grepl("error[[:space:]]advanced[[:space:]]to[[:space:]]second[[:space:]]on[[:space:]]an[[:sp
    season$play) & is.na(season$thirdbase)] <- 5
season$temp[grepl("by[[:space:]][[:alpha:]][[:alpha:]][[:space:]]advanced[[:space:]]to[[:space:]]second
    season$play)] <- 5
season$temp[grepl("by[[:space:]][[:digit:]][[:alpha:]][[:space:]]advanced[[:space:]]to[[:space:]]second
    season$play) & is.na(season$thirdbase)] <- 5
season$temp[grepl("by[[:space:]][[:alpha:]][[:space:]]advanced[[:space:]]to[[:space:]]second", season$pl
    is.na(season$thirdbase)] <- 5
season$temp[grepl("caught[[:space:]]stealing", season$play)] <- 6



season$base_state_after <- NA









# Nobody On



season$base_state_after[season$firstbase == 0 & season$secondbase == 0 & season$thirdbase == 0 & season$
    season$outs_from_play == 0] <- "Nobody on, No Outs"
season$base_state_after[season$firstbase == 0 & season$secondbase == 0 & season$thirdbase == 0 & season$
    season$outs_from_play == 1] <- "Nobody on, One out"
season$base_state_after[season$firstbase == 0 & season$secondbase == 0 & season$thirdbase == 0 & season$
    season$outs_from_play == 2] <- "Nobody on, Two outs"


season$base_state_after[is.na(season$firstbase) & is.na(season$secondbase) & season$thirdbase == 0 &
    season$outs_before_play + season$outs_from_play == 0] <- "Nobody on, No Outs"
season$base_state_after[is.na(season$firstbase) & is.na(season$secondbase) & season$thirdbase == 0 &
    season$outs_before_play + season$outs_from_play == 1] <- "Nobody on, One out"
```

```
season$base_state_after[is.na(season$firstbase) & is.na(season$secondbase) & season$thirdbase == 0 &
    season$outs_before_play + season$outs_from_play == 2] <- "Nobody on, Two outs"








# Runners on First
season$base_state_after[season$firstbase == 1 & is.na(season$secondbase) & is.na(season$thirdbase) &
    season$outs_before_play + season$outs_from_play == 0] <- "Runner on 1st, No outs"

season$base_state_after[season$firstbase == 1 & is.na(season$secondbase) & season$thirdbase == 0 & seas
    season$outs_from_play == 0] <- "Runner on 1st, No outs"

season$base_state_after[season$firstbase == 1 & is.na(season$secondbase) & is.na(season$thirdbase) &
    season$outs_before_play + season$outs_from_play == 1] <- "Runner on 1st, One out"

season$base_state_after[season$firstbase == 1 & is.na(season$secondbase) & season$thirdbase == 0 & seas
    season$outs_from_play == 1] <- "Runner on 1st, One out"

season$base_state_after[season$firstbase == 1 & is.na(season$secondbase) & is.na(season$thirdbase) &
    season$outs_before_play + season$outs_from_play == 2] <- "Runner on 1st, Two outs"

season$base_state_after[season$firstbase == 1 & is.na(season$secondbase) & season$thirdbase == 0 & seas
    season$outs_from_play == 2] <- "Runner on 1st, Two outs"




## Runner on 2nd

season$base_state_after[is.na(season$firstbase) & season$secondbase == 1 & is.na(season$thirdbase) &
    season$outs_before_play + season$outs_from_play == 0] <- "Runner on 2nd, No outs"

season$base_state_after[is.na(season$firstbase) & season$secondbase == 1 & season$thirdbase == 0 & seas
    season$outs_from_play == 0] <- "Runner on 2nd, No outs"


season$base_state_after[is.na(season$firstbase) & season$secondbase == 1 & is.na(season$thirdbase) &
    season$outs_before_play + season$outs_from_play == 1] <- "Runner on 2nd, One out"

season$base_state_after[is.na(season$firstbase) & season$secondbase == 1 & season$thirdbase == 0 & seas
    season$outs_from_play == 1] <- "Runner on 2nd, One out"

season$base_state_after[is.na(season$firstbase) & season$secondbase == 1 & is.na(season$thirdbase) &
    season$outs_before_play + season$outs_from_play == 2] <- "Runner on 2nd, Two outs"

season$base_state_after[is.na(season$firstbase) & season$secondbase == 1 & season$thirdbase == 0 & seas
    season$outs_from_play == 2] <- "Runner on 2nd, Two outs"
```

```r
# Runner on third


season$base_state_after[is.na(season$firstbase) & is.na(season$secondbase) & season$thirdbase == 1 &
    season$outs_before_play + season$outs_from_play == 0] <- "Runner on 3rd, No outs"
season$base_state_after[is.na(season$firstbase) & is.na(season$secondbase) & season$thirdbase == 1 &
    season$outs_before_play + season$outs_from_play == 1] <- "Runner on 3rd, One out"
season$base_state_after[is.na(season$firstbase) & is.na(season$secondbase) & season$thirdbase == 1 &
    season$outs_before_play + season$outs_from_play == 2] <- "Runner on 3rd, Two outs"

# Runners on 1st and 2nd

season$base_state_after[season$firstbase == 1 & season$secondbase == 1 & is.na(season$thirdbase) & seas
    season$outs_from_play == 0] <- "Runners on 1st and 2nd, No outs"

season$base_state_after[season$firstbase == 1 & season$secondbase == 1 & season$thirdbase == 0 & season
    season$outs_from_play == 0] <- "Runners on 1st and 2nd, No outs"


season$base_state_after[season$firstbase == 1 & season$secondbase == 1 & is.na(season$thirdbase) & seas
    season$outs_from_play == 1] <- "Runners on 1st and 2nd, One out"

season$base_state_after[season$firstbase == 1 & season$secondbase == 1 & season$thirdbase == 0 & season
    season$outs_from_play == 1] <- "Runners on 1st and 2nd, One out"

season$base_state_after[season$firstbase == 1 & season$secondbase == 1 & is.na(season$thirdbase) & seas
    season$outs_from_play == 2] <- "Runners on 1st and 2nd, Two outs"

season$base_state_after[season$firstbase == 1 & season$secondbase == 1 & season$thirdbase == 0 & season
    season$outs_from_play == 2] <- "Runners on 1st and 2nd, Two outs"

# Runners on 2nd and 3rd

season$base_state_after[is.na(season$firstbase) & season$secondbase == 1 & season$thirdbase == 1 & seas
    season$outs_from_play == 0] <- "Runners on 2nd and 3rd, No outs"
season$base_state_after[is.na(season$firstbase) & season$secondbase == 1 & season$thirdbase == 1 & seas
    season$outs_from_play == 1] <- "Runners on 2nd and 3rd, One out"
season$base_state_after[is.na(season$firstbase) & season$secondbase == 1 & season$thirdbase == 1 & seas
    season$outs_from_play == 2] <- "Runners on 2nd and 3rd, Two outs"


# Runners on 1st and 3rd

season$base_state_after[season$firstbase == 1 & is.na(season$secondbase) & season$thirdbase == 1 & seas
    season$outs_from_play == 0] <- "Runners on 1st and 3rd, No outs"
season$base_state_after[season$firstbase == 1 & is.na(season$secondbase) & season$thirdbase == 1 & seas
    season$outs_from_play == 1] <- "Runners on 1st and 3rd, One out"
season$base_state_after[season$firstbase == 1 & is.na(season$secondbase) & season$thirdbase == 1 & seas
    season$outs_from_play == 2] <- "Runners on 1st and 3rd, Two outs"

# Bases Loaded

season$base_state_after[season$firstbase == 1 & season$secondbase == 1 & season$thirdbase == 1 & season
```

```
    season$outs_from_play == 0] <- "Bases Loaded, No outs"
season$base_state_after[season$firstbase == 1 & season$secondbase == 1 & season$thirdbase == 1 & season
    season$outs_from_play == 1] <- "Bases Loaded, One out"
season$base_state_after[season$firstbase == 1 & season$secondbase == 1 & season$thirdbase == 1 & season
    season$outs_from_play == 2] <- "Bases Loaded, Two outs"


season$base_state_after[season$temp == 1] <- "placeholder"
season$base_state_after[season$temp == 2] <- "placeholder"
season$base_state_after[season$temp == 3] <- "placeholder"
season$base_state_after[season$temp == 4] <- "placeholder"
```

10

```
# Setting the table for the base_State_before, including starting each inning with 'Nobody on, No
# Outs'

season$base_state_before <- NA




# Top of inning
for (k in games) {
    innings <- unique(season$inningID[season$gameID == k])
    for (i in innings) {

        n <- length(season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bott
            "top"])
        if (n == 1) {
            season$base_state_before[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][1] <- "Nobody on, No Outs"
        } else {
            season$base_state_before[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][2:n] <- (season$base_state_after[season$inningID == i & season$gameID == k & seas
                "top"][1:n - 1])
            season$base_state_before[season$inningID == i & season$gameID == k & season$top_bottom ==
                "top"][1] <- "Nobody on, No Outs"
        }
    }
}
# Bottom of Inning
for (k in games) {
    innings <- unique(season$inningID[season$gameID == k])
    for (i in innings) {
        n <- length(season$outs_before_play[season$inningID == i & season$gameID == k & season$top_bott
            "bottom"])
        if (n == 1) {
            season$base_state_before[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][1] <- "Nobody on, No Outs"
        } else {
            season$base_state_before[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][2:n] <- (season$base_state_after[season$inningID == i & season$gameID == k &
                season$top_bottom == "bottom"][1:n - 1])
            season$base_state_before[season$inningID == i & season$gameID == k & season$top_bottom ==
                "bottom"][1] <- "Nobody on, No Outs"
        }
    }
}
```

11

```
season$base_state_before[is.na(season$base_state_before)] <- "placeholder"

# Nobody On
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, No Outs" & season$outs_from_play[i - 1] == 1 &
        season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, One out" & season$outs_from_play[i - 1] == 1 &
        season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}


# Runner on 1st

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

# Runner on 2nd
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
```

```
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}

# Runner on 3rd
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, Two outs"
    }
}

# Runners on 1st and 2nd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$outs_from_play[i
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$outs_from_play[i
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}
# Runners on 1st and 3rd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$outs_from_play[i
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$outs_from_play[i
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
```

```
}

# Runners on 2nd and 3rd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, Two outs"
    }
}
# Bases Loaded

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}




# Condition 1 - solo reach first

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, No Outs" & season$temp[i - 1] == 1 & season$base_
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, One out" & season$temp[i - 1] == 1 & season$base_
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, Two outs" & season$temp[i - 1] == 1 & season$bas
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, No outs"
    }
```

```
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$temp[i - 1] == 1 & season$b
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, Two outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
}

## Subset

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 1 & season$b
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 1 & season$b
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}
```

```
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

## 4

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 4 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 4 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}
```

```r
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 4 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, No outs" & season$temp[i - 1] == 4 & season$ba
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, One out" & season$temp[i - 1] == 4 & season$ba
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, Two outs" & season$temp[i - 1] == 4 & season$
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

## 6

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 6 & season
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}
```

```
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}

# Condition 2 - the Stolen Base

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 2 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 2 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 2 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}



# Condition 3 - stolen third

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 3 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 3 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 3 & season$b
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
}
```

11.2

```
season$base_state_before[is.na(season$base_state_before)] <- "placeholder"



# Nobody On
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, No Outs" & season$outs_from_play[i - 1] == 1 &
        season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, One out" & season$outs_from_play[i - 1] == 1 &
        season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}


# Runner on 1st

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

# Runner on 2nd
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}

# Runner on 3rd
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, Two outs"
    }
}

# Runners on 1st and 2nd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}
# Runners on 1st and 3rd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
}

# Runners on 2nd and 3rd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, Two outs"
    }
}
# Bases Loaded

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}




# Condition 1 - solo reach first

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, No Outs" & season$temp[i - 1] == 1 & season$base_
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, One out" & season$temp[i - 1] == 1 & season$base_
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, Two outs" & season$temp[i - 1] == 1 & season$base
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}
```

```
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
}

## Subset

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}
```

```r
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

## 4

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 4 & season$b
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 4 & season$b
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 4 & season
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, No outs" & season$temp[i - 1] == 4 & season$ba
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, One out" & season$temp[i - 1] == 4 & season$ba
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, Two outs" & season$temp[i - 1] == 4 & season$b
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}
```

```
## 6

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 6 & season$b
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 6 & season$b
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}
```

```r
# Condition 2 - the Stolen Base

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 2 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 2 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 2 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}



# Condition 3 - stolen third

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 3 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 3 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 3 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
}
```

11.3

```
season$base_state_before[is.na(season$base_state_before)] <- "placeholder"



# Nobody On
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, No Outs" & season$outs_from_play[i - 1] == 1 &
        season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, One out" & season$outs_from_play[i - 1] == 1 &
        season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}


# Runner on 1st

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

# Runner on 2nd
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}
```

```
for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}

# Runner on 3rd
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, Two outs"
    }
}

# Runners on 1st and 2nd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}
# Runners on 1st and 3rd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
}
```

```r
# Runners on 2nd and 3rd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, Two outs"
    }
}
# Bases Loaded

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}




# Condition 1 - solo reach first

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, No Outs" & season$temp[i - 1] == 1 & season$base_
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, One out" & season$temp[i - 1] == 1 & season$base_
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, Two outs" & season$temp[i - 1] == 1 & season$base
        "placeholder" & season$runs_from_play[i - 1] == 0) {
```

```
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, No outs"
    }
}
```

```
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
}

## Subset

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

## 4

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 4 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 4 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 4 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, No outs" & season$temp[i - 1] == 4 & season$ba
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, One out" & season$temp[i - 1] == 4 & season$ba
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, Two outs" & season$temp[i - 1] == 4 & season$b
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

## 6

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
```

```
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 6 & season
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}
```

```
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 6 & season
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}

# Condition 2 - the Stolen Base

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 2 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 2 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 2 & season
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
```

```r
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}



# Condition 3 - stolen third

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 3 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 3 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 3 & season$b
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
}
```

11.4

```
season$base_state_before[is.na(season$base_state_before)] <- "placeholder"



# Nobody On
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, No Outs" & season$outs_from_play[i - 1] == 1 &
        season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, One out" & season$outs_from_play[i - 1] == 1 &
        season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}


# Runner on 1st

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

# Runner on 2nd
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}

# Runner on 3rd
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, Two outs"
    }
}

# Runners on 1st and 2nd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}
# Runners on 1st and 3rd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
}

# Runners on 2nd and 3rd

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$outs_from_play[i -
        1] == 1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, Two outs"
    }
}
# Bases Loaded

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, No outs" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 3:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, One out" & season$outs_from_play[i - 1] ==
        1 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}



# Condition 1 - solo reach first

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, No Outs" & season$temp[i - 1] == 1 & season$base_
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, One out" & season$temp[i - 1] == 1 & season$base_
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Nobody on, Two outs" & season$temp[i - 1] == 1 & season$base
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}
```

```r
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
}

## Subset

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 1) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, No outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, One out" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 2nd and 3rd, Two outs" & season$temp[i - 1] ==
        1 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}
```

```
for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, No outs" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, One out" & season$temp[i - 1] == 1 & season$
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 3rd, Two outs" & season$temp[i - 1] == 1 & season
        "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 1st, Two outs"
    }
}

## 4

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] == 0) {
        season$base_state_before[i] <- "Bases Loaded, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        4 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 4 & season$b
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 4 & season$b
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 4 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, No outs" & season$temp[i - 1] == 4 & season$ba
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, One out" & season$temp[i - 1] == 4 & season$ba
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Bases Loaded, Two outs" & season$temp[i - 1] == 4 & season$b
        "placeholder" & season$runs_from_play[i - 1] == 2) {
        season$base_state_before[i] <- "Runners on 1st and 2nd, Two outs"
    }
}
```

```
## 6

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, No Outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, Two outs" & season$temp[i - 1] == 6 & season$
        "placeholder") {
        season$base_state_before[i] <- "Nobody on, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, Two outs" & season$temp[i - 1] ==
        6 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}
```

```r
# Condition 2 - the Stolen Base

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, No outs" & season$temp[i - 1] == 2 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, One out" & season$temp[i - 1] == 2 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 1st, Two outs" & season$temp[i - 1] == 2 & season
        "placeholder") {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 2nd and 3rd, Two outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, No outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, One out" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 3rd, Two outs" & season$temp[i - 1] ==
        2 & season$base_state_before[i] == "placeholder" & season$runs_from_play[i - 1] != 0) {
        season$base_state_before[i] <- "Runner on 2nd, Two outs"
    }
}



# Condition 3 - stolen third

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, No outs" & season$temp[i - 1] == 3 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 3 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runner on 2nd, One out" & season$temp[i - 1] == 3 & season$
        "placeholder") {
        season$base_state_before[i] <- "Runner on 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, No outs" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, No outs"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, One out"
    }
}

for (i in 2:length(season$outs_from_play)) {
    if (season$base_state_before[i - 1] == "Runners on 1st and 2nd, One out" & season$temp[i - 1] ==
        3 & season$base_state_before[i] == "placeholder") {
        season$base_state_before[i] <- "Runners on 1st and 3rd, Two outs"
    }
}


table(season$base_state_before)
```

```
           Bases Loaded, No outs            Bases Loaded, One out           Bases Loaded, Two outs 
                            1165                             2502                             2685 
              Nobody on, No Outs               Nobody on, One out              Nobody on, Two outs 
                           27563                            16705                            10864 
                     placeholder           Runner on 1st, No outs           Runner on 1st, One out 
                            8314                             8525                             8446 
         Runner on 1st, Two outs           Runner on 2nd, No outs           Runner on 2nd, One out 
                            7418                             3064                             5503 
         Runner on 2nd, Two outs           Runner on 3rd, No outs           Runner on 3rd, One out 
                            5725                              561                             1872 
         Runner on 3rd, Two outs  Runners on 1st and 2nd, No outs  Runners on 1st and 2nd, One out 
                            2831                             2369                             3576 
Runners on 1st and 2nd, Two outs  Runners on 1st and 3rd, No outs  Runners on 1st and 3rd, One out 
                            3616                              731                             1570 
Runners on 1st and 3rd, Two outs  Runners on 2nd and 3rd, No outs  Runners on 2nd and 3rd, One out 
                            1957                              935                             2378 
Runners on 2nd and 3rd, Two outs 
                            2716 
```

```
save(season, file = "season.rda")
```

12

```
# Max runs per half of an inning

library(dplyr)
runs_df <- season %>% dplyr::group_by(gameID, inningID, top_bottom) %>% dplyr::summarize(runs = max(runs
    na.rm = T))
```

13

```
#'Nobody on, No Outs'

library(dplyr)
nooneon_noout_runs_df <- season[season$base_state_before == "Nobody on, No Outs", ] %>% dplyr::group_by
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

nooneon_noout_runs_df <- merge(nooneon_noout_runs_df, runs_df, all = TRUE, sort = FALSE)
```

```
nooneon_noout_runs_df$runsReal <- nooneon_noout_runs_df$runs - nooneon_noout_runs_df$runsbefore

summary(nooneon_noout_runs_df$runsReal)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   0.000   0.000   0.656   1.000  15.000 
```

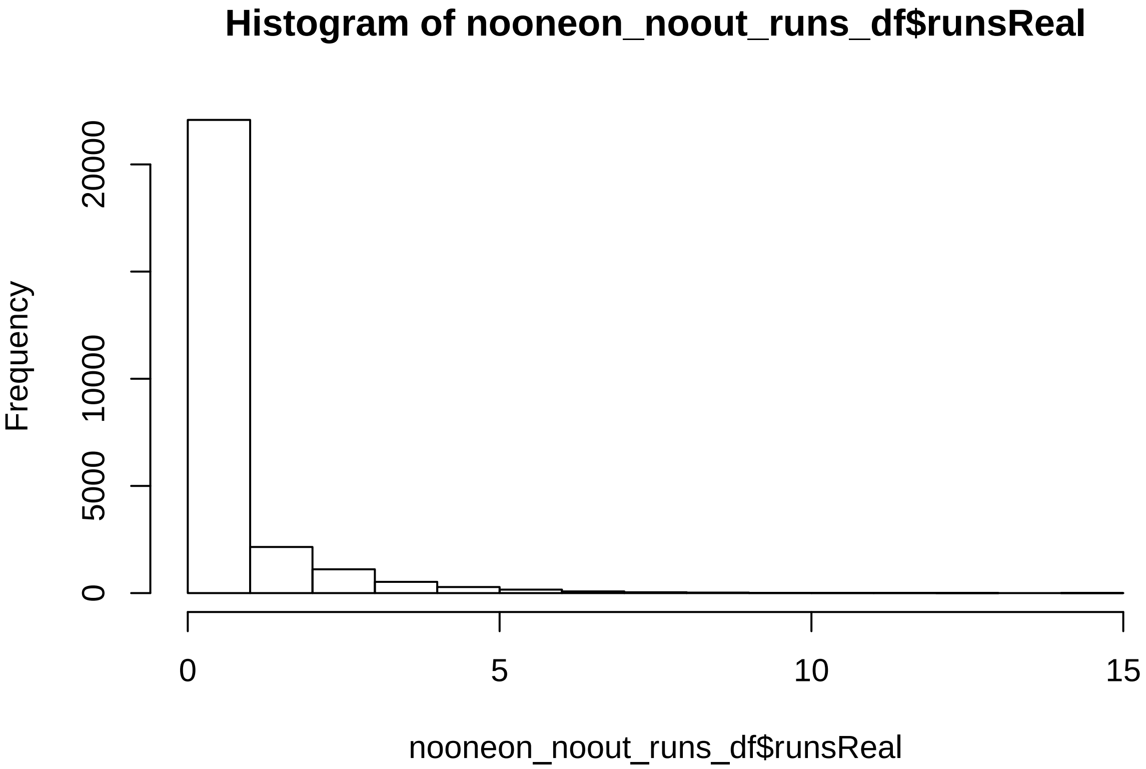
```
length(nooneon_noout_runs_df$runsReal)
```

```
[1] 26472
```

```
sum(nooneon_noout_runs_df$runsReal)/sqrt(length(which(season$base_state_before == "Nobody on, No Outs"))
```

```
[1] 104.5
```

```
hist(nooneon_noout_runs_df$runsReal)
```



```
standarderror1 <- sd(nooneon_noout_runs_df$runsReal)/sqrt(length(which(season$base_state_before == "Nob
standarderror1
```

```
[1] 0.007667
```

```
# 95% confidence intervals of the mean
nooneon_noout_runs_df$ll <- mean(nooneon_noout_runs_df$runsReal) - 1.96 * standarderror1
nooneon_noout_runs_df$ul <- mean(nooneon_noout_runs_df$runsReal) + 1.96 * standarderror1

nooneon_noout_runs_df$ul[1]
```

```
[1] 0.6707
```

```
nooneon_noout_runs_df$ll[1]
```

```
[1] 0.6406
```

```
(mean(nooneon_noout_runs_df$runsReal, na.rm = T) - 0.555)/(standarderror1)
```

```
[1] 13.13
```

14

```
#'Nobody on, One out'

library(dplyr)

nooneon_oneout_runs_df <- season[season$base_state_before == "Nobody on, One out", ] %>% dplyr::group_b
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

nooneon_oneout_runs_df <- merge(nooneon_oneout_runs_df, runs_df, all = TRUE, sort = FALSE)

nooneon_oneout_runs_df$runsReal <- nooneon_oneout_runs_df$runs - nooneon_oneout_runs_df$runsbefore

mean(nooneon_oneout_runs_df$runsReal, na.rm = T)
```

```
[1] 0.3099
```

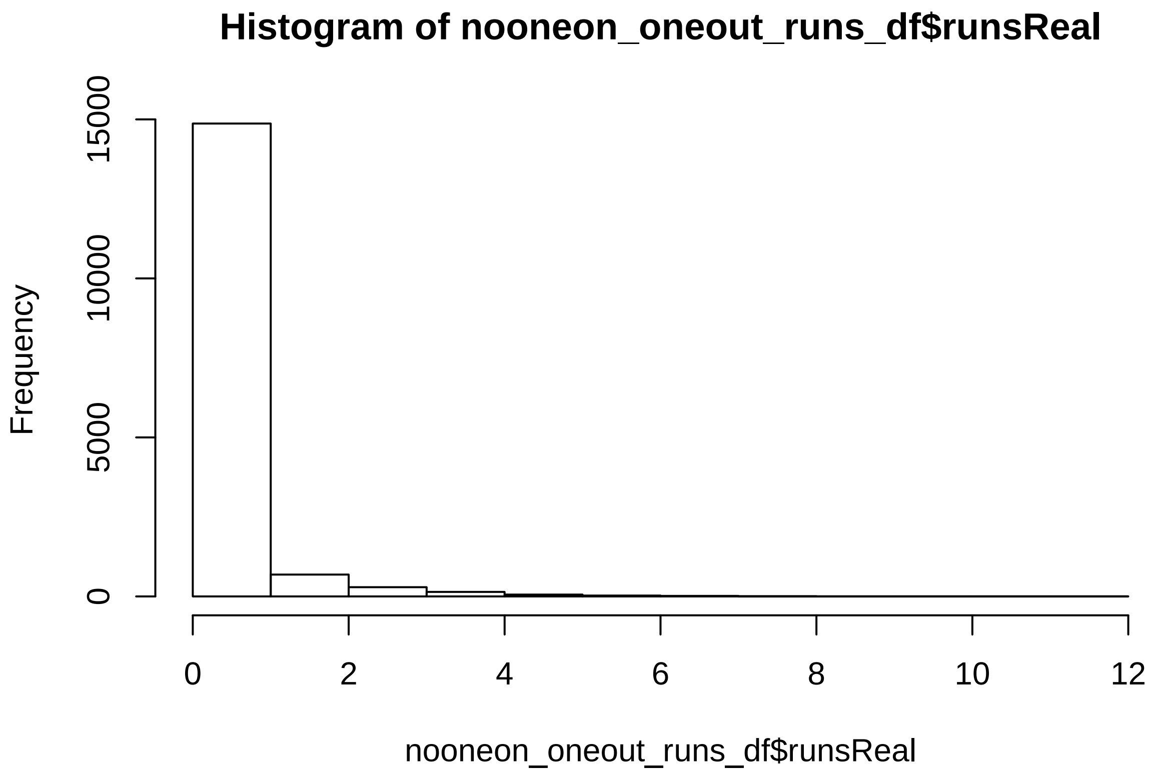
```
table(nooneon_oneout_runs_df$runsReal)
```

```

                0 0.333333333333333               0.5 0.666666666666666 0.666666666666667
            13173                 1               209                 1                 1
                1  1.33333333333333               1.5                 2  2.33333333333333
             1485                 1                68               618                 1
              2.5  2.66666666666667              2.75                 3               3.5
               25                 1                 2               261                12
                4  4.33333333333333               4.5  4.66666666666667                 5
              130                 1                 4                 1                52
 5.33333333333333               5.5                 6               6.5  6.66666666666667
                1                 2                24                 2                 1
                7                 8                 9                10                11
               13                 5                 1                 1                 1
               12
                1
```

```
sum(nooneon_oneout_runs_df$runsReal)/length(nooneon_oneout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(nooneon_oneout_runs_df$runsReal)
```



```
standarderror2 <- sd(nooneon_oneout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_be
    "Nobody on, One out")))
standarderror2
```

```
[1] 0.006478
```

```
# 95% confidence intervals of the mean
nooneon_oneout_runs_df$ll <- mean(nooneon_oneout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror2
nooneon_oneout_runs_df$ul <- mean(nooneon_oneout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror2

nooneon_oneout_runs_df$ul[1]
```

```
[1] 0.3226
```

```
nooneon_oneout_runs_df$ll[1]
```

```
[1] 0.2972
```

```
mean(nooneon_oneout_runs_df$runsReal, na.r = T) - 0.297/(standarderror2)
```

```
[1] NA
```

15

```
#'Nobody on, Two outs'
library(dplyr)

nooneon_twoout_runs_df <- season[season$base_state_before == "Nobody on, Two outs", ] %>% dplyr::group_
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))
```

```
nooneon_twoout_runs_df <- merge(nooneon_twoout_runs_df, runs_df, all = TRUE, sort = FALSE)



nooneon_twoout_runs_df$runsReal <- nooneon_twoout_runs_df$runs - nooneon_twoout_runs_df$runsbefore



mean(nooneon_twoout_runs_df$runsReal, na.rm = T)
```

```
[1] 0.09698
```

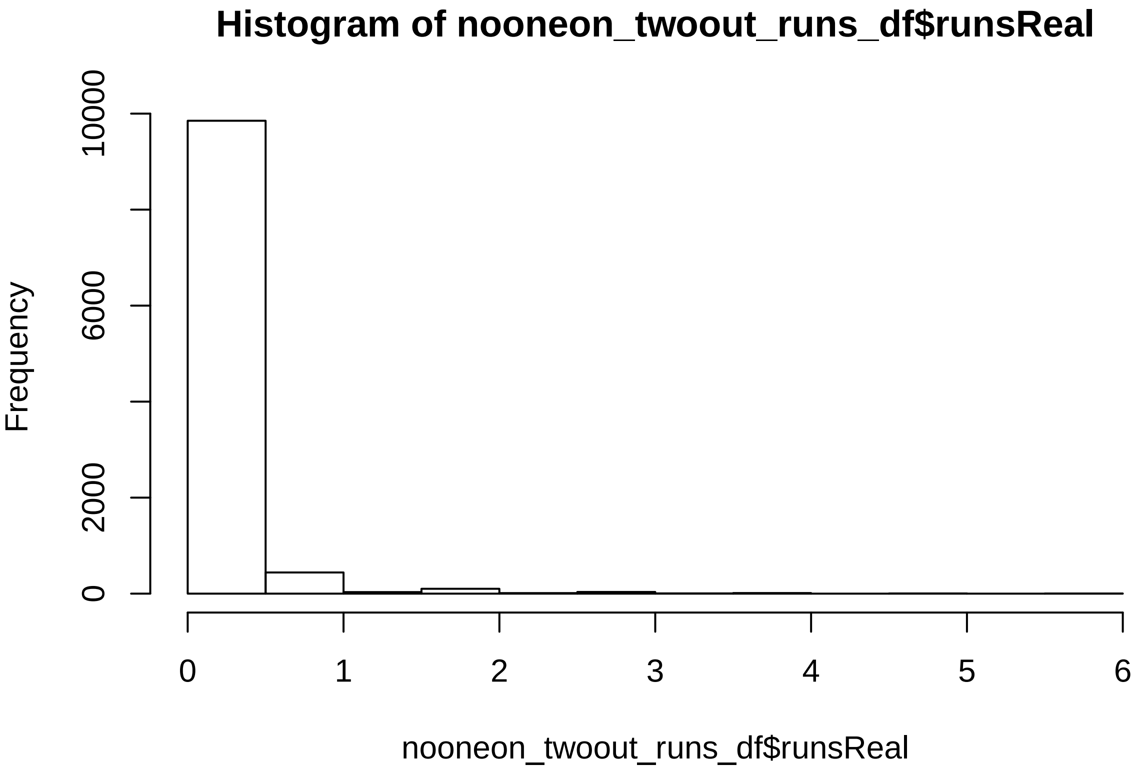
```
table(nooneon_twoout_runs_df$runsReal)
```

```

               0              0.5                1 1.33333333333333              1.5
            9665              187              441                4               28
1.66666666666667             1.75                2 2.33333333333333              2.5
               1                1               99                2                9
2.66666666666667                3              3.5 3.66666666666667                4
               2               33                5                1               11
               5                6
               4                3
```

```
sum(nooneon_twoout_runs_df$runsReal)/length(nooneon_twoout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(nooneon_twoout_runs_df$runsReal)
```



```
standarderror3 <- sd(nooneon_twoout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_be
    "Nobody on, Two outs")))
standarderror3
```

```
[1] 0.003859
```

```
# 95% confidence intervals of the mean
nooneon_twoout_runs_df$ll <- mean(nooneon_twoout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror3
nooneon_twoout_runs_df$ul <- mean(nooneon_twoout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror3

nooneon_twoout_runs_df$ul[1]
```

```
[1] 0.1045
```

```
nooneon_twoout_runs_df$ll[1]
```

```
[1] 0.08942
```

```
mean(nooneon_twoout_runs_df$runsReal, na.r = T) - 0.117/(standarderror3)
```

```
[1] NA
```

16

```
library(dplyr)

one_noout_runs_df <- season[season$base_state_before == "Runner on 1st, No outs", ] %>% dplyr::group_by
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

one_noout_runs_df <- merge(one_noout_runs_df, runs_df, all = TRUE, sort = FALSE)



one_noout_runs_df$runsReal <- one_noout_runs_df$runs - one_noout_runs_df$runsbefore



mean(one_noout_runs_df$runsReal, na.rm = T)
```

```
[1] 1.1
```

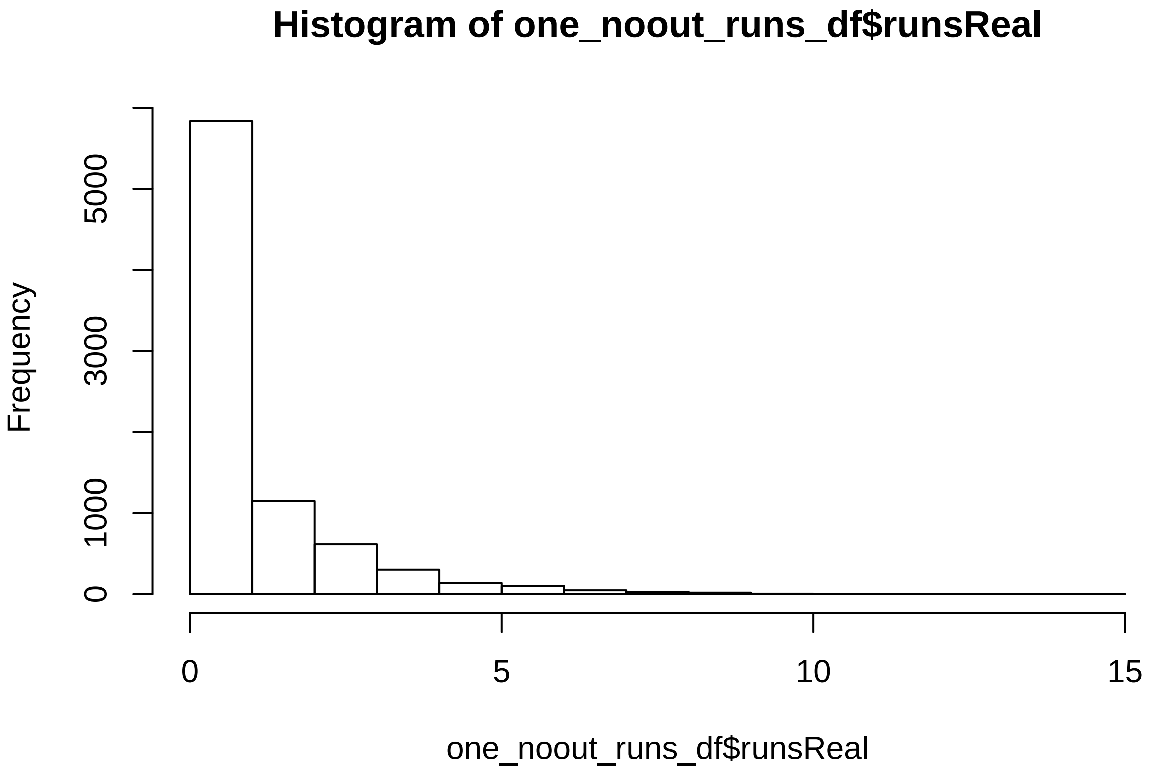
```
table(one_noout_runs_df$runsReal)
```

```

               0              0.5                1              1.5                2
            4233               23             1579               25             1124
             2.5 2.66666666666667                3 3.33333333333333              3.5
              23                1              591                1                8
               4              4.5                5              5.5                6
             293                6              132                8               93
             6.5                7              7.5                8              8.5
               4               43                1               28                2
               9               10               11               12               13
              16                5                2                4                1
              15
               1
```

```
sum(one_noout_runs_df$runsReal)/length(one_noout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(one_noout_runs_df$runsReal)
```



```
standarderror4 <- sd(one_noout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_before
    "Runner on 1st, No outs")))
standarderror4
```

```
[1] 0.01717
```

```
# 95% confidence intervals of the mean
one_noout_runs_df$ll <- mean(one_noout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror4
one_noout_runs_df$ul <- mean(one_noout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror4

one_noout_runs_df$ul[1]
```

```
[1] 1.133
```

```
one_noout_runs_df$ll[1]
```

```
[1] 1.066
```

```
mean(one_noout_runs_df$runsReal, na.r = T) - 0.117/(standarderror4)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

17

```
library(dplyr)

one_oneout_runs_df <- season[season$base_state_before == "Runner on 1st, One out", ] %>% dplyr::group_b
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

one_oneout_runs_df <- merge(one_oneout_runs_df, runs_df, all = TRUE, sort = FALSE)



one_oneout_runs_df$runsReal <- one_oneout_runs_df$runs - one_oneout_runs_df$runsbefore



mean(one_oneout_runs_df$runsReal, na.rm = T)
```

```
[1] 0.6034
```

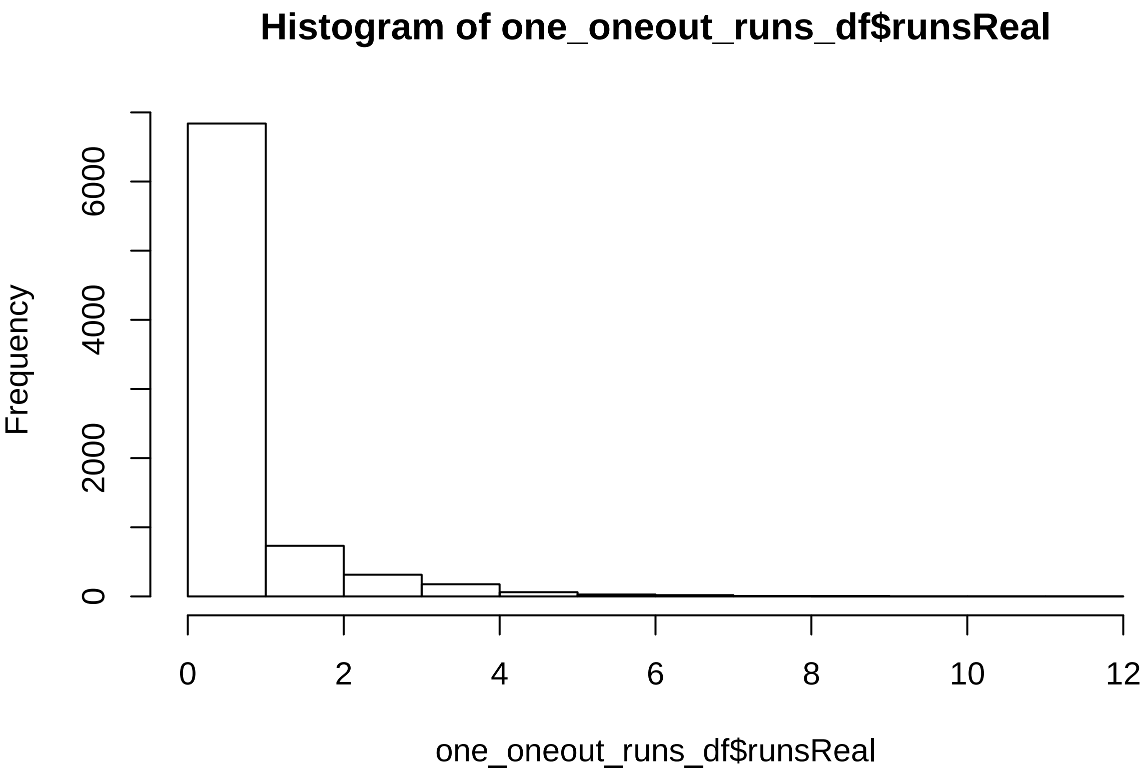
```
table(one_oneout_runs_df$runsReal)
```

```

                0 0.333333333333333               0.5 0.666666666666666                 1 
             5683                 1                40                 1              1114 
              1.5                 2              2.25  2.33333333333333               2.5 
               29               703                 1                 1                12 
                3  3.33333333333333               3.5                 4               4.5 
              300                 1                12               163                 3 
                5               5.5                 6  6.33333333333333               6.5 
               58                 3                25                 1                 1 
                7                 8                 9                10              10.5 
               16                 6                 5                 1                 1 
               12 
                1 
```

```
sum(one_oneout_runs_df$runsReal)/length(one_oneout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(one_oneout_runs_df$runsReal)
```



```
standarderror5 <- sd(one_oneout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_before
    "Runner on 1st, One out")))
standarderror5
```

```
[1] 0.01267
```

```
# 95% confidence intervals of the mean
one_oneout_runs_df$ll <- mean(one_oneout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror5
one_oneout_runs_df$ul <- mean(one_oneout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror5

one_oneout_runs_df$ul[1]
```

```
[1] 0.6282
```

```
one_oneout_runs_df$ll[1]
```

```
[1] 0.5785
```

```
mean(one_oneout_runs_df$runsReal, na.r = T) - 0.573/(standarderror5)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

18

```
library(dplyr)

one_twoouts_runs_df <- season[season$base_state_before == "Runner on 1st, Two outs", ] %>% dplyr::group
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))
```

```
one_twoouts_runs_df <- merge(one_twoouts_runs_df, runs_df, all = TRUE, sort = FALSE)


one_twoouts_runs_df$runsReal <- one_twoouts_runs_df$runs - one_twoouts_runs_df$runsbefore


mean(one_twoouts_runs_df$runsReal, na.rm = T)
```

```
[1] 0.2343
```

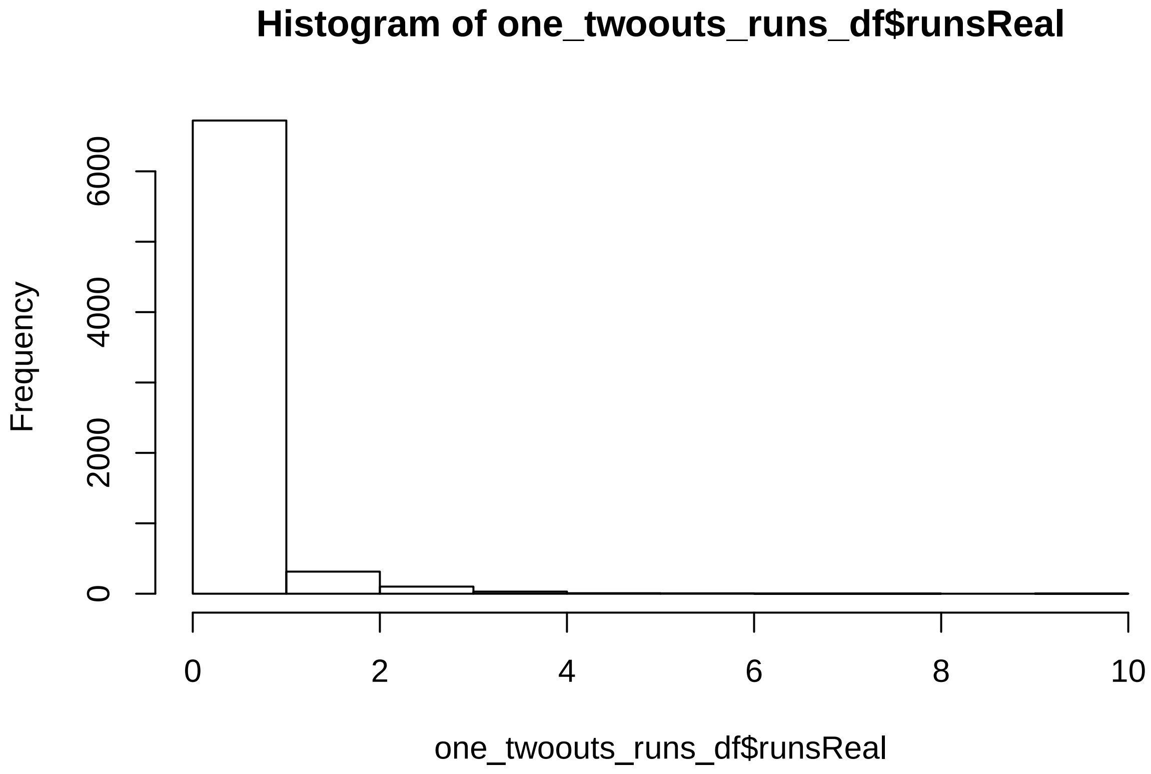
```
table(one_twoouts_runs_df$runsReal)
```

```

               0              0.5                1 1.33333333333333              1.5 
            6124               58              540                1               32 
1.66666666666667                2 2.33333333333333              2.5 2.66666666666667 
               2              279                1               13                1 
               3             3.25              3.5                4              4.5 
              86                1                2               27                1 
               5              5.5                6                7                8 
               6                1                4                1                1 
              10 
               1 
```

```
sum(one_twoouts_runs_df$runsReal)/length(one_twoouts_runs_df$runsReal)
```

```
[1] NA
```

```
hist(one_twoouts_runs_df$runsReal)
```



```
standarderror6 <- sd(one_twoouts_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_before
    "Runner on 1st, Two outs")))
standarderror6
```

```
[1] 0.007809
```

```
# 95% confidence intervals of the mean
one_twoouts_runs_df$ll <- mean(one_twoouts_runs_df$runsReal, na.rm = T) - 1.96 * standarderror6
one_twoouts_runs_df$ul <- mean(one_twoouts_runs_df$runsReal, na.rm = T) + 1.96 * standarderror6

one_twoouts_runs_df$ul[1]
```

```
[1] 0.2496
```

```
one_twoouts_runs_df$ll[1]
```

```
[1] 0.2189
```

```
mean(one_twoouts_runs_df$runsReal, na.r = T) - 0.251/(standarderror6)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

19

```
library(dplyr)

two_noout_runs_df <- season[season$base_state_before == "Runner on 2nd, No outs", ] %>% dplyr::group_by
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

two_noout_runs_df <- merge(two_noout_runs_df, runs_df, all = TRUE, sort = FALSE)



two_noout_runs_df$runsReal <- two_noout_runs_df$runs - two_noout_runs_df$runsbefore



mean(two_noout_runs_df$runsReal, na.rm = T)
```

```
[1] 1.392
```

```
table(two_noout_runs_df$runsReal)
```

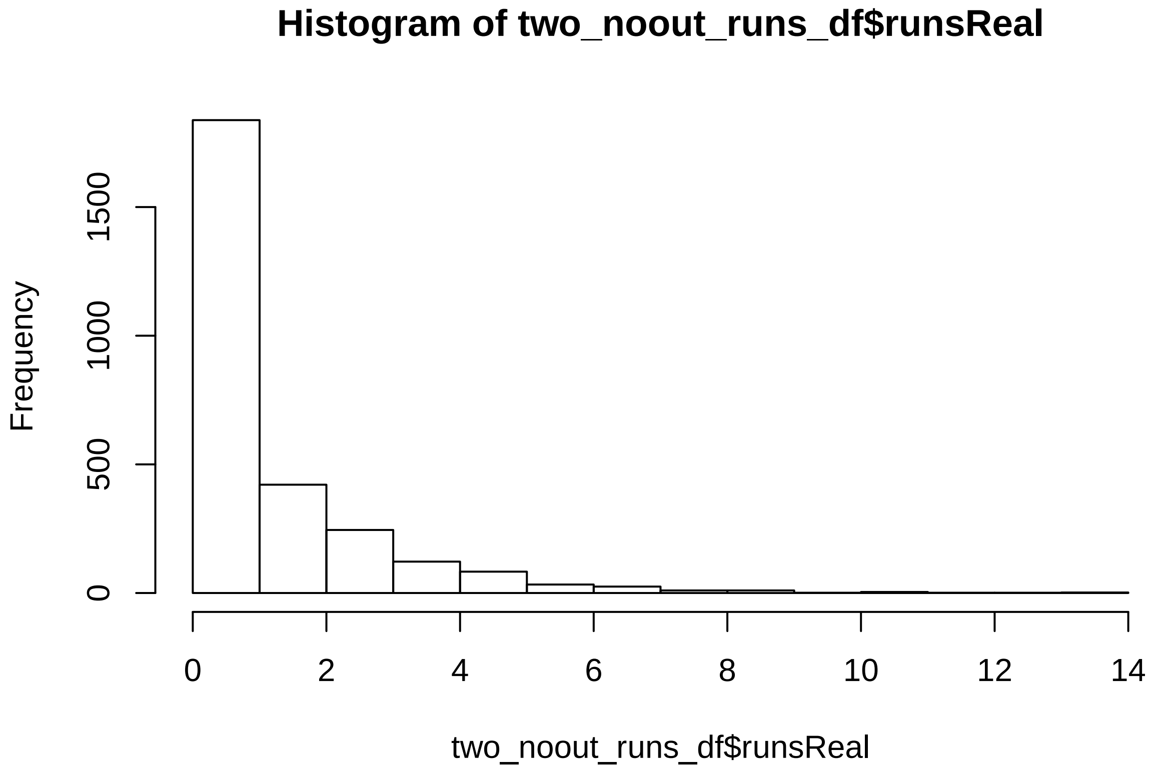
```

               0              0.5                1              1.5 1.66666666666667
             987               44              807               49                2
               2              2.5                3              3.5 3.66666666666667
             370               38              207               16                1
               4              4.5                5              5.5                6
             105               12               71                6               27
             6.5                7              7.5                8              8.5
               1               24                3                7                1
               9               10             10.5               11               12
```

```
           9            1            2            2            1
          13         13.5
           1            2
```

```
sum(two_noout_runs_df$runsReal)/length(two_noout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(two_noout_runs_df$runsReal)
```



```
standarderror7 <- sd(two_noout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_before
    "Runner on 2nd, No outs")))
standarderror7
```

```
[1] 0.03062
```

```
# 95% confidence intervals of the mean
two_noout_runs_df$ll <- mean(two_noout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror7
two_noout_runs_df$ul <- mean(two_noout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror7

two_noout_runs_df$ul[1]
```

```
[1] 1.452
```

```
two_noout_runs_df$ll[1]
```

```
[1] 1.332
```

```
mean(two_noout_runs_df$runsReal, na.r = T) - 0.251/(standarderror7)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

20

```
library(dplyr)

two_oneout_runs_df <- season[season$base_state_before == "Runner on 2nd, One out", ] %>% dplyr::group_b
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

two_oneout_runs_df <- merge(two_oneout_runs_df, runs_df, all = TRUE, sort = FALSE)



two_oneout_runs_df$runsReal <- two_oneout_runs_df$runs - two_oneout_runs_df$runsbefore



mean(two_oneout_runs_df$runsReal, na.rm = T)
```

```
[1] 0.7695
```

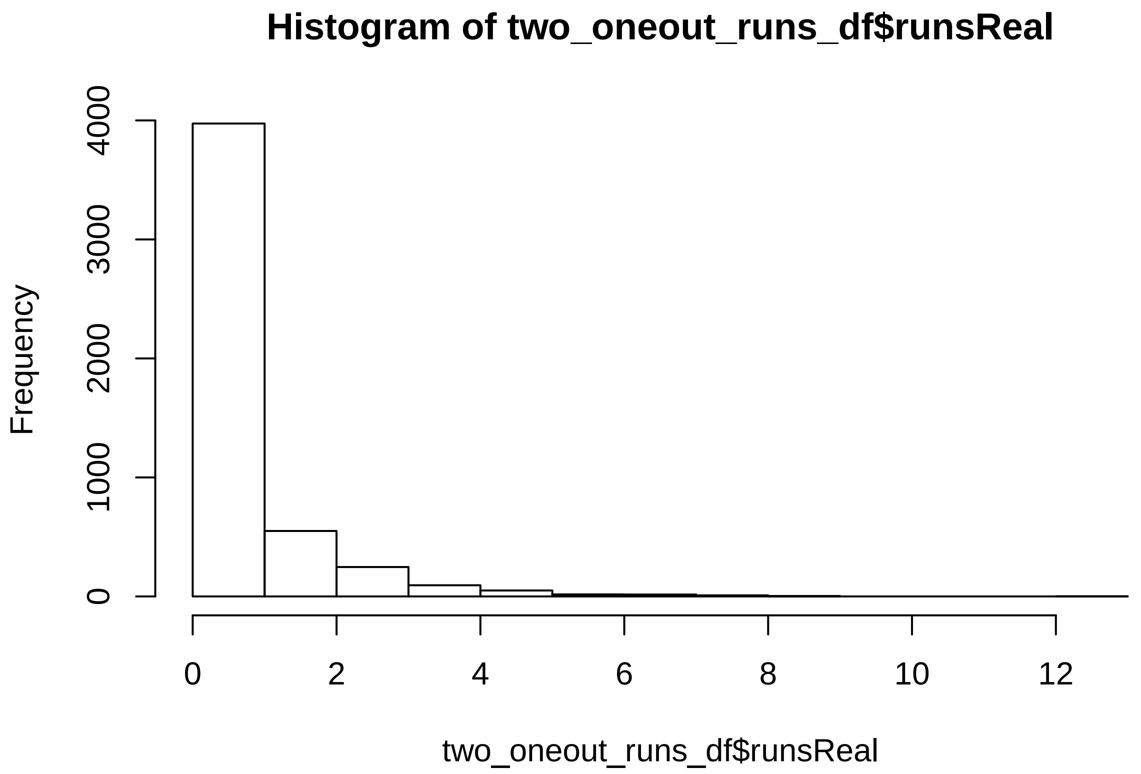
```
table(two_oneout_runs_df$runsReal)
```

```

               0              0.5                1 1.33333333333333              1.5 
            2778              184             1012                3               79 
1.66666666666667             1.75                2 2.33333333333333              2.5 
               6                1              461                2               37 
2.66666666666667                3              3.5              3.6                4 
               5              203               12                1               81 
             4.5                5              5.5                6              6.5 
               9               41                3               14                2 
               7              7.5                8                9               13 
              14                2                7                3                1 
```

```
sum(two_oneout_runs_df$runsReal)/length(two_oneout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(two_oneout_runs_df$runsReal)
```



```
standarderror8 <- sd(two_oneout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_before
    "Runner on 2nd, One out")))
standarderror8
```

```
[1] 0.01632
```

```
# 95% confidence intervals of the mean
two_oneout_runs_df$ll <- mean(two_oneout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror8
two_oneout_runs_df$ul <- mean(two_oneout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror8

two_oneout_runs_df$ul[1]
```

```
[1] 0.8015
```

```
two_oneout_runs_df$ll[1]
```

```
[1] 0.7375
```

```
mean(two_oneout_runs_df$runsReal, na.r = T) - 0.251/(standarderror8)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

21

```
library(dplyr)

two_twoout_runs_df <- season[season$base_state_before == "Runner on 2nd, Two outs", ] %>% dplyr::group_
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))
```

```
two_twoout_runs_df <- merge(two_twoout_runs_df, runs_df, all = TRUE, sort = FALSE)


two_twoout_runs_df$runsReal <- two_twoout_runs_df$runs - two_twoout_runs_df$runsbefore


mean(two_twoout_runs_df$runsReal, na.rm = T)
```

```
[1] 0.3012
```

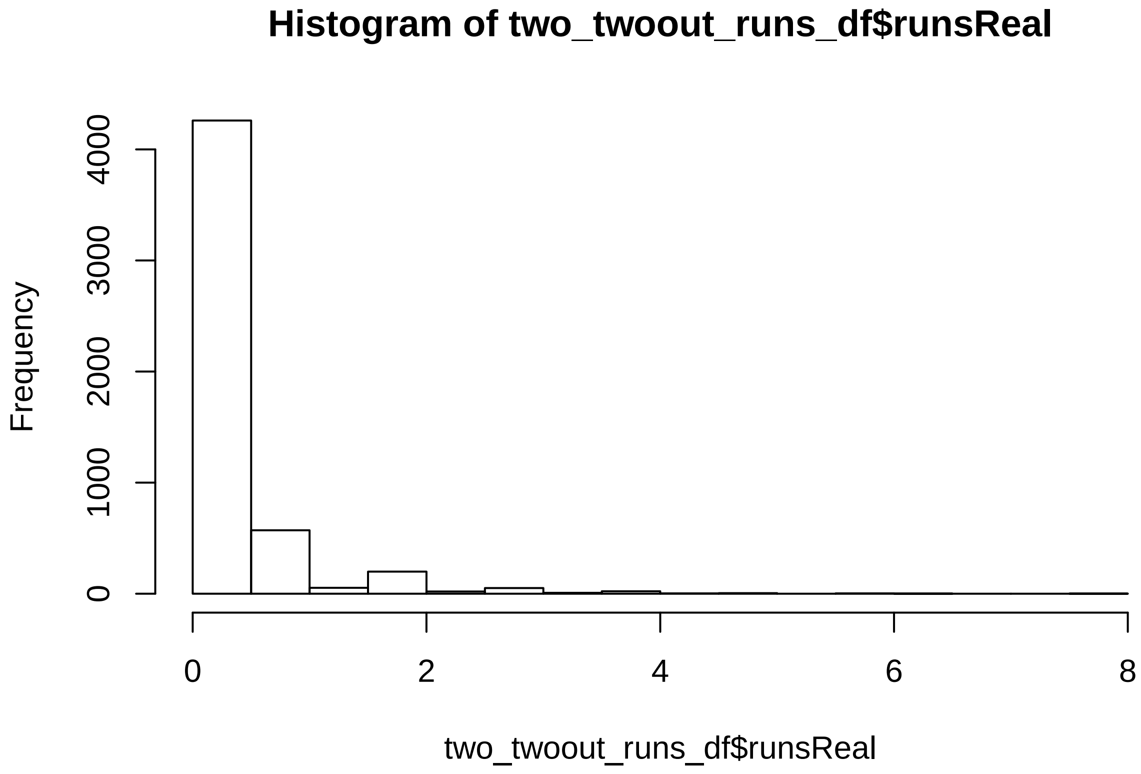
```
table(two_twoout_runs_df$runsReal)
```

```
               0              0.5 0.666666666666667                1 1.33333333333333
            3993              267                 1              570                5
             1.5 1.66666666666667              1.75                2 2.33333333333333
              48                8                 2              189                2
             2.5 2.66666666666667                 3 3.33333333333333              3.5
              18                2                49                2                6
3.66666666666667             3.75                 4              4.4              4.5
               1                1                20                1                2
               5 5.66666666666667                 6              6.5                8
               5                1                 2                1                1
```

```
sum(two_twoout_runs_df$runsReal)/length(two_twoout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(two_twoout_runs_df$runsReal)
```



```
standarderror9 <- sd(two_twoout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_before
    "Runner on 2nd, Two outs")))
standarderror9
```

```
[1] 0.009138
```

```
# 95% confidence intervals of the mean
two_twoout_runs_df$ll <- mean(two_twoout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror9
two_twoout_runs_df$ul <- mean(two_twoout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror9

two_twoout_runs_df$ul[1]
```

```
[1] 0.3191
```

```
two_twoout_runs_df$ll[1]
```

```
[1] 0.2833
```

```
mean(two_twoout_runs_df$runsReal, na.r = T) - 0.251/(standarderror9)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

22

```
library(dplyr)

three_noout_runs_df <- season[season$base_state_before == "Runner on 3rd, No outs", ] %>% dplyr::group_b
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

three_noout_runs_df <- merge(three_noout_runs_df, runs_df, all = TRUE, sort = FALSE)



three_noout_runs_df$runsReal <- three_noout_runs_df$runs - three_noout_runs_df$runsbefore



mean(three_noout_runs_df$runsReal, na.rm = T)
```

```
[1] 1.517
```

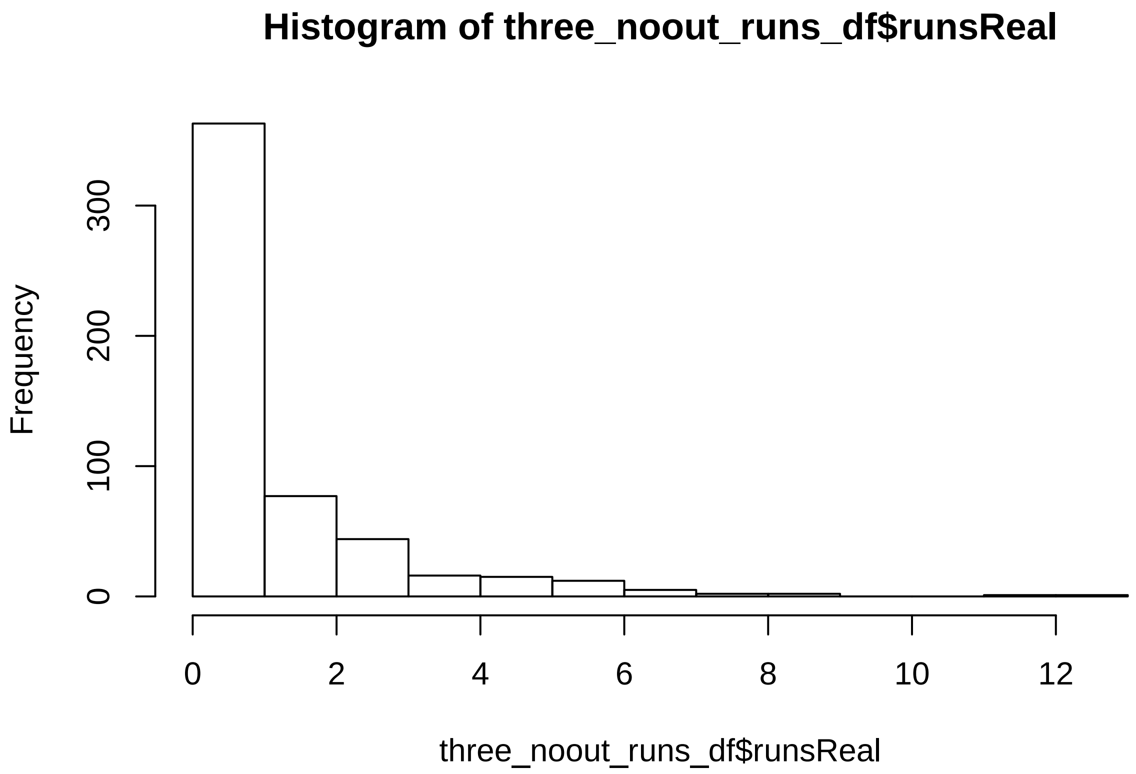
```
table(three_noout_runs_df$runsReal)
```

```

  0 0.5   1 1.5   2 2.5   3   4 4.5   5   6 6.5   7   8   9  12  13
132   1 230   5  72   4  40  16   1  14  12   1   4   2   2   1   1
```

```
sum(three_noout_runs_df$runsReal)/length(three_noout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(three_noout_runs_df$runsReal)
```



```
standarderror10 <- sd(three_noout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_befo
    "Runner on 3rd, No outs")))
standarderror10
```

```
[1] 0.07234
```

```
# 95% confidence intervals of the mean
three_noout_runs_df$ll <- mean(three_noout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror10
three_noout_runs_df$ul <- mean(three_noout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror10

three_noout_runs_df$ul[1]
```

```
[1] 1.659
```

```
three_noout_runs_df$ll[1]
```

```
[1] 1.375
```

```
mean(three_noout_runs_df$runsReal, na.r = T) - 0.251/(standarderror10)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

23

```
library(dplyr)

three_oneout_runs_df <- season[season$base_state_before == "Runner on 3rd, One out", ] %>% dplyr::group_
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))
```

```
three_oneout_runs_df <- merge(three_oneout_runs_df, runs_df, all = TRUE, sort = FALSE)


three_oneout_runs_df$runsReal <- three_oneout_runs_df$runs - three_oneout_runs_df$runsbefore


mean(three_oneout_runs_df$runsReal, na.rm = T)
```

```
[1] 1.028
```

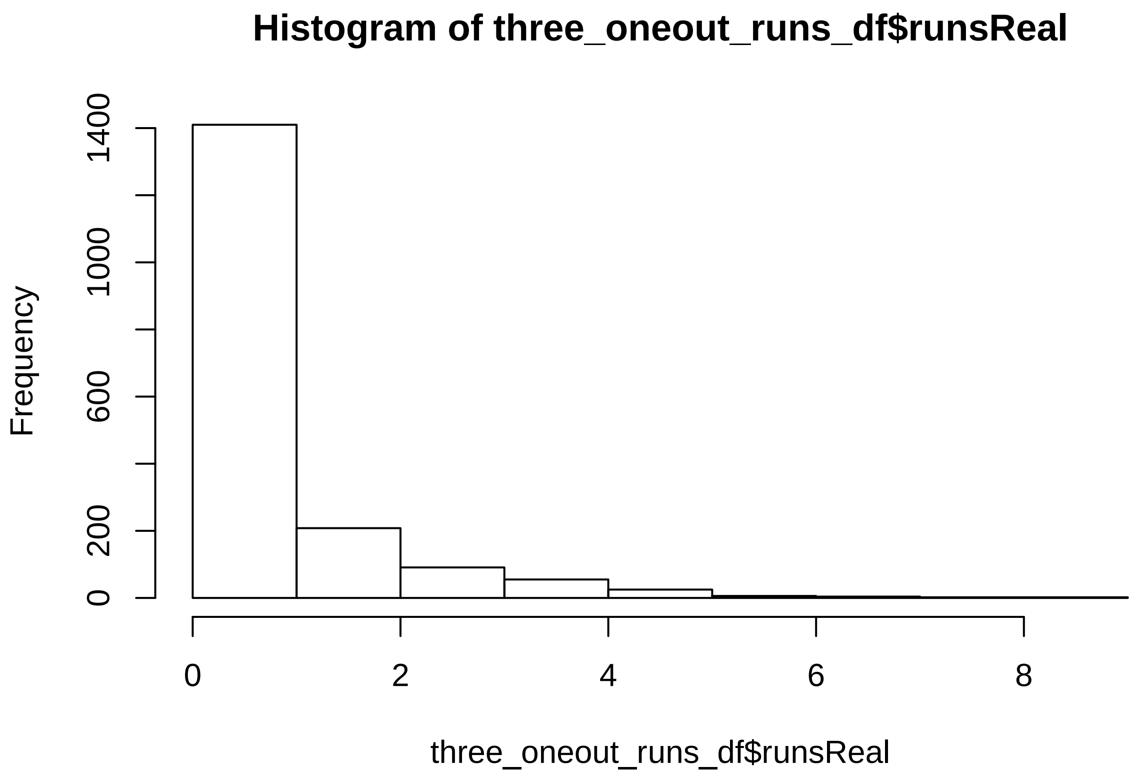
```
table(three_oneout_runs_df$runsReal)
```

```

               0              0.5                1 1.33333333333333              1.5
             671               13              726                1                6
               2              2.5                3              3.5                4
             201                4               87                5               50
4.33333333333333              4.5                5                6                7
               1                4               20                6                4
               8                9
               2                2
```

```
sum(three_oneout_runs_df$runsReal)/length(three_oneout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(three_oneout_runs_df$runsReal)
```



```
standarderror11 <- sd(three_oneout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_befo
    "Runner on 3rd, One out")))
standarderror11
```

```
[1] 0.02784
```

```
# 95% confidence intervals of the mean
three_oneout_runs_df$ll <- mean(three_oneout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror11
three_oneout_runs_df$ul <- mean(three_oneout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror11

three_oneout_runs_df$ul[1]
```

```
[1] 1.083
```

```
three_oneout_runs_df$ll[1]
```

```
[1] 0.9735
```

```
mean(three_oneout_runs_df$runsReal, na.r = T) - 0.251/(standarderror11)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

24

```
library(dplyr)

three_twoout_runs_df <- season[season$base_state_before == "Runner on 3rd, Two outs", ] %>% dplyr::group
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

three_twoout_runs_df <- merge(three_twoout_runs_df, runs_df, all = TRUE, sort = FALSE)



three_twoout_runs_df$runsReal <- three_twoout_runs_df$runs - three_twoout_runs_df$runsbefore



mean(three_twoout_runs_df$runsReal, na.rm = T)
```

```
[1] 0.4105
```

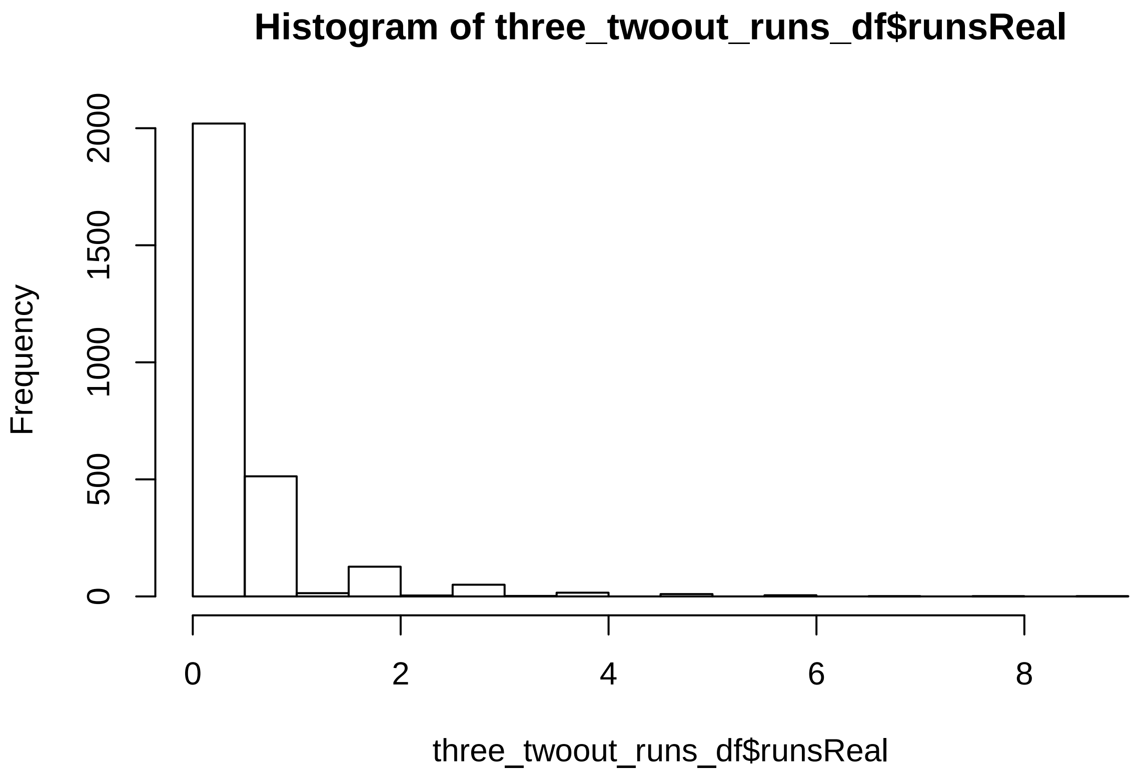
```
table(three_twoout_runs_df$runsReal)
```

```

   0  0.5    1  1.5 1.75    2  2.5    3  3.5    4    5    6    7    8    9 
1996   24  513   14    1  126    4   50    2   16   10    5    1    1    1 
```

```
sum(three_twoout_runs_df$runsReal)/length(three_twoout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(three_twoout_runs_df$runsReal)
```



```
standarderror12 <- sd(three_twoout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_bef
    "Runner on 3rd, Two outs")))
standarderror12
```

```
[1] 0.01587
```

```
# 95% confidence intervals of the mean
three_twoout_runs_df$ll <- mean(three_twoout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror12
three_twoout_runs_df$ul <- mean(three_twoout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror12

three_twoout_runs_df$ul[1]
```

```
[1] 0.4417
```

```
three_twoout_runs_df$ll[1]
```

```
[1] 0.3794
```

```
mean(three_twoout_runs_df$runsReal, na.r = T) - 0.251/(standarderror12)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

25

```
library(dplyr)

onetwo_noout_runs_df <- season[season$base_state_before == "Runners on 1st and 2nd, No outs", ] %>% dpl
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))
```

```
onetwo_noout_runs_df <- merge(onetwo_noout_runs_df, runs_df, all = TRUE, sort = FALSE)


onetwo_noout_runs_df$runsReal <- onetwo_noout_runs_df$runs - onetwo_noout_runs_df$runsbefore


mean(onetwo_noout_runs_df$runsReal, na.rm = T)
```

```
[1] 1.768
```

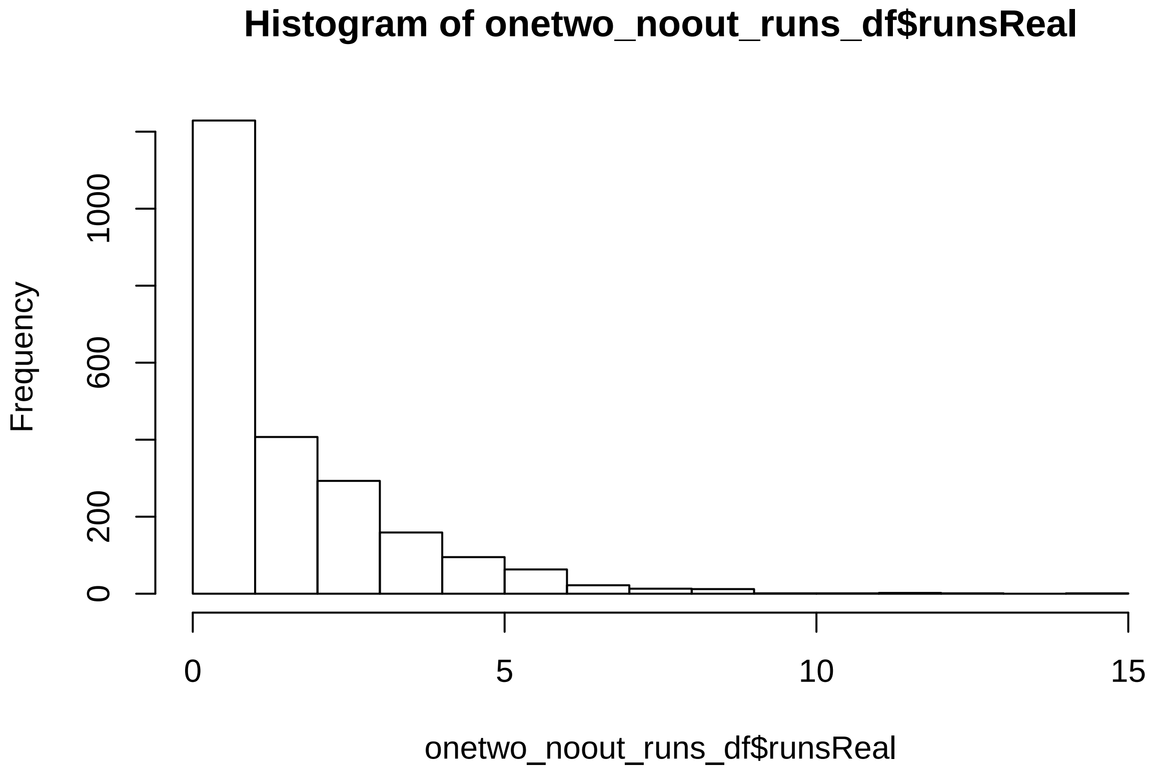
```
table(onetwo_noout_runs_df$runsReal)
```

```

               0              0.5                1              1.5                2
             766                4              459                7              400
             2.5 2.66666666666667                3              3.5                4
               6                1              286                7              152
             4.5                5              5.5 5.66666666666667                6
               4               91                6                1               56
             6.5                7                8             8.75                9
               1               21               13                1               11
              10               11               12               13               15
               1                1                2                1                1
```

```
sum(onetwo_noout_runs_df$runsReal)/length(onetwo_noout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(onetwo_noout_runs_df$runsReal)
```



```
standarderror13 <- sd(onetwo_noout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_bef
    "Runners on 1st and 2nd, No outs")))
standarderror13
```

```
[1] 0.03924
```

```
# 95% confidence intervals of the mean
onetwo_noout_runs_df$ll <- mean(onetwo_noout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror13
onetwo_noout_runs_df$ul <- mean(onetwo_noout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror13

onetwo_noout_runs_df$ul[1]
```

```
[1] 1.845
```

```
onetwo_noout_runs_df$ll[1]
```

```
[1] 1.692
```

```
mean(onetwo_noout_runs_df$runsReal, na.r = T) - 0.251/(standarderror13)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

26

```
library(dplyr)

onetwo_oneout_runs_df <- season[season$base_state_before == "Runners on 1st and 2nd, One out", ] %>%
    dplyr::group_by(gameID, inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_pla
    na.rm = T))

onetwo_oneout_runs_df <- merge(onetwo_oneout_runs_df, runs_df, all = TRUE, sort = FALSE)



onetwo_oneout_runs_df$runsReal <- onetwo_oneout_runs_df$runs - onetwo_oneout_runs_df$runsbefore



mean(onetwo_oneout_runs_df$runsReal, na.rm = T)
```

```
[1] 1.048
```

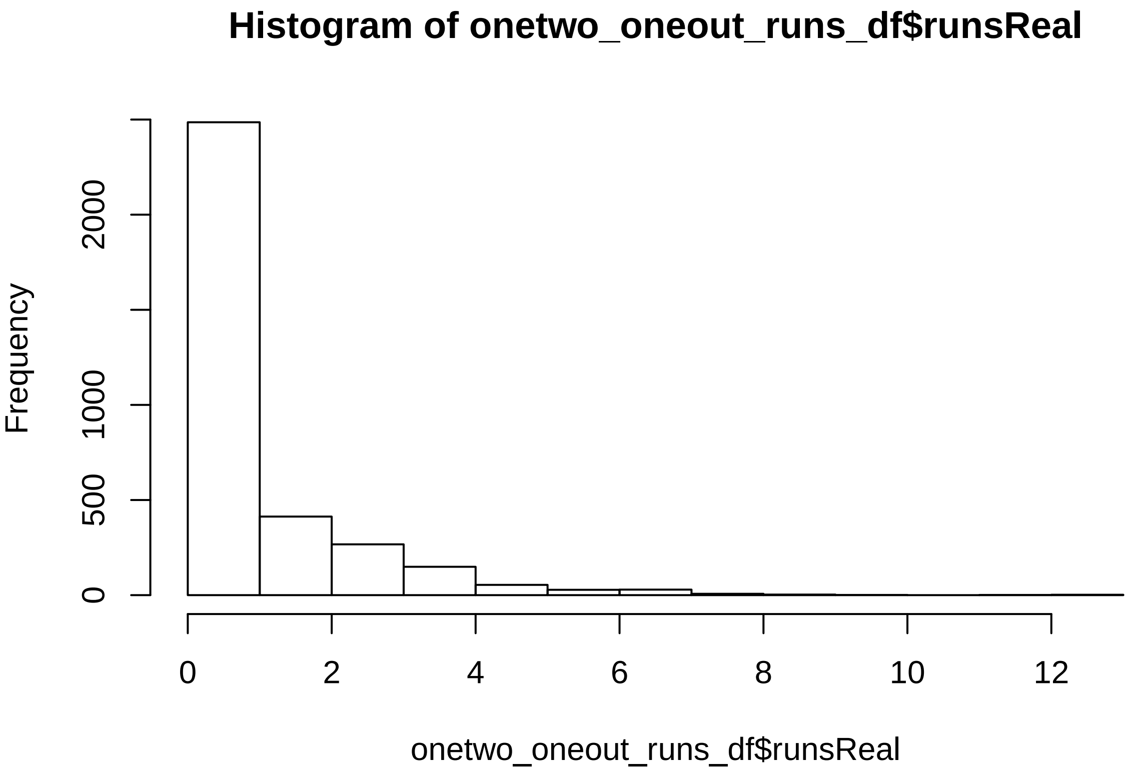
```
table(onetwo_oneout_runs_df$runsReal)
```

```
               0              0.5                1              1.5                2
            1845                7              634               20              393
             2.5                3 3.33333333333333              3.5                4
               7              260                1               14              134
             4.5                5              5.5                6              6.5
               4               50                5               23                2
               7                8                9               10               12
              27                7                3                1                1
             13
              2
```

```r
sum(onetwo_oneout_runs_df$runsReal)/length(onetwo_oneout_runs_df$runsReal)
```

```
[1] NA
```

```r
hist(onetwo_oneout_runs_df$runsReal)
```



```r
standarderror14 <- sd(onetwo_oneout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_be
    "Runners on 1st and 2nd, One out")))
standarderror14
```

```
[1] 0.02578
```

```r
# 95% confidence intervals of the mean
onetwo_oneout_runs_df$ll <- mean(onetwo_oneout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror14
onetwo_oneout_runs_df$ul <- mean(onetwo_oneout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror14

onetwo_oneout_runs_df$ul[1]
```

```
[1] 1.099
```

```r
onetwo_oneout_runs_df$ll[1]
```

```
[1] 0.9977
```

```r
mean(onetwo_oneout_runs_df$runsReal, na.r = T) - 0.251/(standarderror14)
```

```
[1] NA
```

```r
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

27

```
library(dplyr)

onetwo_twoouts_runs_df <- season[season$base_state_before == "Runners on 1st and 2nd, Two outs", ] %>%
    dplyr::group_by(gameID, inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_pla
    na.rm = T))

onetwo_twoouts_runs_df <- merge(onetwo_twoouts_runs_df, runs_df, all = TRUE, sort = FALSE)



onetwo_twoouts_runs_df$runsReal <- onetwo_twoouts_runs_df$runs - onetwo_twoouts_runs_df$runsbefore



mean(onetwo_twoouts_runs_df$runsReal, na.rm = T)
```

```
[1] 0.4529
```

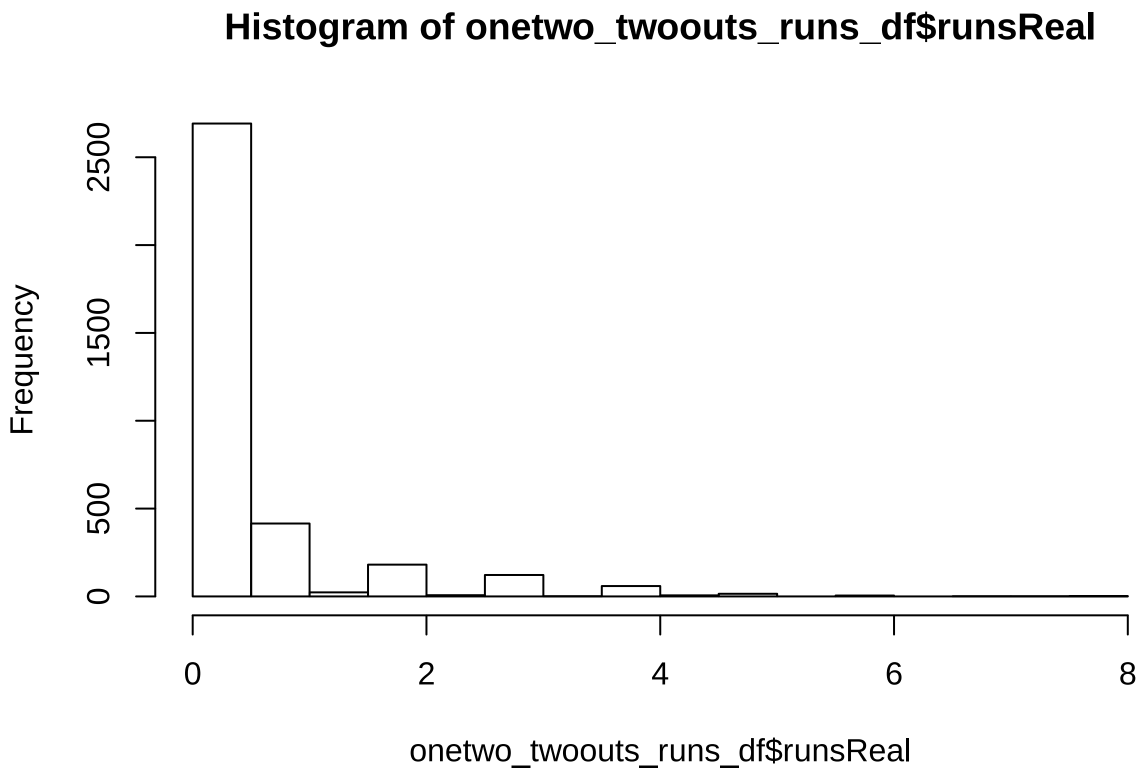
```
table(onetwo_twoouts_runs_df$runsReal)
```

```

               0              0.5                1 1.33333333333333              1.5
            2688                4              415                1               22
               2              2.5                3              3.5                4
             181                7              122                1               59
4.33333333333333              4.5                5                6                7
               1                5               15                5                1
             7.5                8
               1                2
```

```
sum(onetwo_twoouts_runs_df$runsReal)/length(onetwo_twoouts_runs_df$runsReal)
```

```
[1] NA
```

```
hist(onetwo_twoouts_runs_df$runsReal)
```



```
standarderror15 <- sd(onetwo_twoouts_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_be
    "Runners on 1st and 2nd, Two outs")))
standarderror15
```

```
[1] 0.0165
```

```
# 95% confidence intervals of the mean
onetwo_twoouts_runs_df$ll <- mean(onetwo_twoouts_runs_df$runsReal, na.rm = T) - 1.96 * standarderror15
onetwo_twoouts_runs_df$ul <- mean(onetwo_twoouts_runs_df$runsReal, na.rm = T) + 1.96 * standarderror15

onetwo_twoouts_runs_df$ul[1]
```

```
[1] 0.4852
```

```
onetwo_twoouts_runs_df$ll[1]
```

```
[1] 0.4205
```

```
mean(onetwo_twoouts_runs_df$runsReal, na.r = T) - 0.251/(standarderror15)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

28

```
library(dplyr)

onethree_noout_runs_df <- season[season$base_state_before == "Runners on 1st and 3rd, No outs", ] %>%
    dplyr::group_by(gameID, inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_pla
    na.rm = T))

onethree_noout_runs_df <- merge(onethree_noout_runs_df, runs_df, all = TRUE, sort = FALSE)


onethree_noout_runs_df$runsReal <- onethree_noout_runs_df$runs - onethree_noout_runs_df$runsbefore


mean(onethree_noout_runs_df$runsReal, na.rm = T)
```

```
[1] 1.948
```

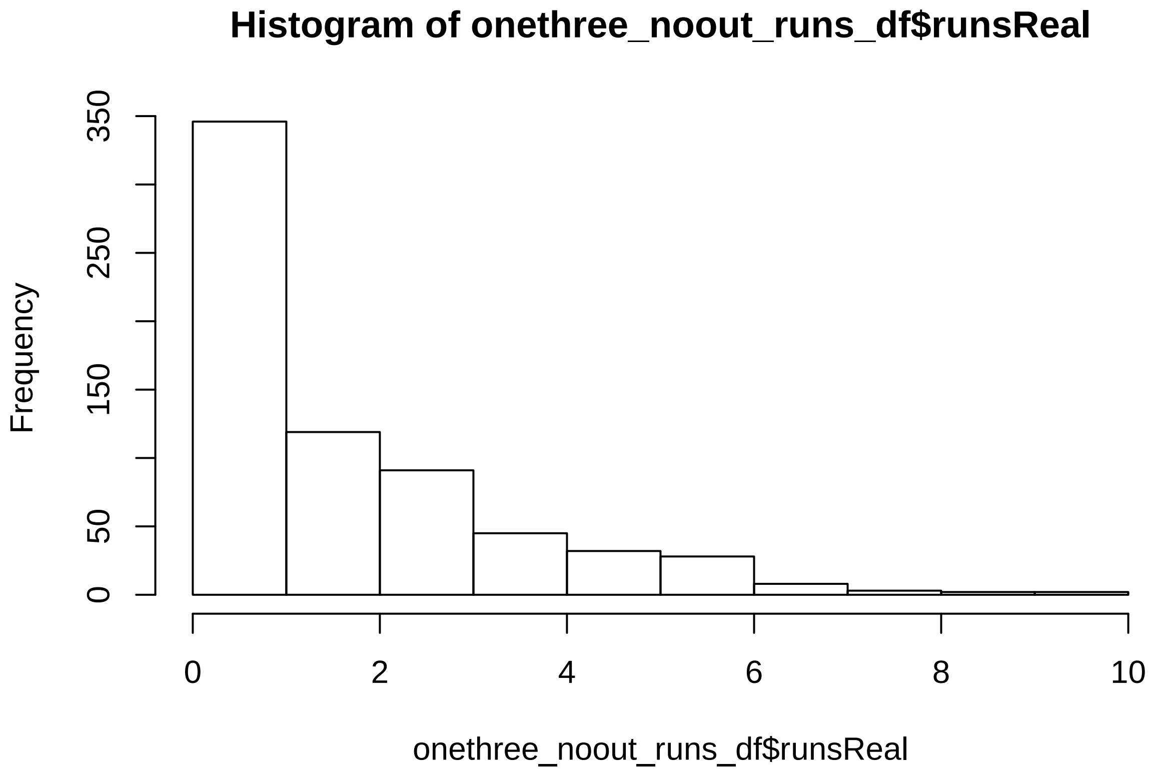
```
table(onethree_noout_runs_df$runsReal)
```

```

               0              0.5                1              1.5 1.66666666666667
             148               10              188               11                1
               2              2.5                3              3.5                4
             107                5               86                5               40
             4.5                5              5.5                6              6.5
               1               31                2               26                1
               7                8                9               10
               7                3                2                2
```

```
sum(onethree_noout_runs_df$runsReal)/length(onethree_noout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(onethree_noout_runs_df$runsReal)
```



```r
standarderror16 <- sd(onethree_noout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_be
    "Runners on 1st and 3rd, No outs")))
standarderror16
```

```
[1] 0.06853
```

```r
# 95% confidence intervals of the mean
onethree_noout_runs_df$ll <- mean(onethree_noout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror16
onethree_noout_runs_df$ul <- mean(onethree_noout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror16

onethree_noout_runs_df$ul[1]
```

```
[1] 2.083
```

```r
onethree_noout_runs_df$ll[1]
```

```
[1] 1.814
```

```r
mean(onethree_noout_runs_df$runsReal, na.r = T) - 0.251/(standarderror16)
```

```
[1] NA
```

```r
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

29

```r
library(dplyr)

onethree_oneout_runs_df <- season[season$base_state_before == "Runners on 1st and 3rd, One out", ] %>%
    dplyr::group_by(gameID, inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_pla
    na.rm = T))

onethree_oneout_runs_df <- merge(onethree_oneout_runs_df, runs_df, all = TRUE, sort = FALSE)



onethree_oneout_runs_df$runsReal <- onethree_oneout_runs_df$runs - onethree_oneout_runs_df$runsbefore



mean(onethree_oneout_runs_df$runsReal, na.rm = T)
```

```
[1] 1.239
```

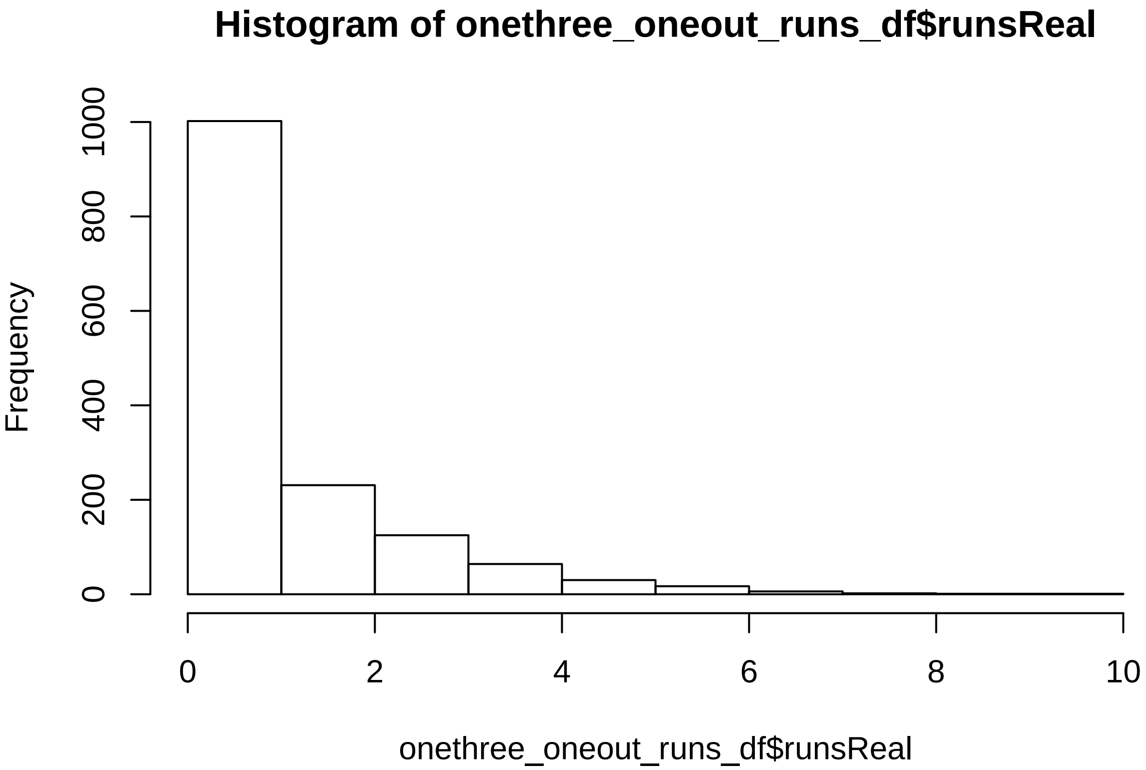
```r
table(onethree_oneout_runs_df$runsReal)
```

```

               0              0.5                1              1.5                2
             564               17              421               20              211
            2.25              2.5                3              3.5 3.66666666666667
               1                8              116                3                1
               4              4.5 4.66666666666667                5                6
              60                2                1               27               17
             6.5                7                8                9               10
               1                5                2                1                1
```

```
sum(onethree_oneout_runs_df$runsReal)/length(onethree_oneout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(onethree_oneout_runs_df$runsReal)
```



```
standarderror17 <- sd(onethree_oneout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_
    "Runners on 1st and 3rd, One out")))
standarderror17
```

```
[1] 0.03653
```

```
# 95% confidence intervals of the mean
onethree_oneout_runs_df$ll <- mean(onethree_oneout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror1
onethree_oneout_runs_df$ul <- mean(onethree_oneout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror1

onethree_oneout_runs_df$ul[1]
```

```
[1] 1.311
```

```
onethree_oneout_runs_df$ll[1]
```

```
[1] 1.168
```

```
mean(onethree_oneout_runs_df$runsReal, na.r = T) - 0.251/(standarderror17)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

30

```
library(dplyr)

onethree_twoouts_runs_df <- season[season$base_state_before == "Runners on 1st and 3rd, Two outs", ] %>
    dplyr::group_by(gameID, inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_pla
    na.rm = T))

onethree_twoouts_runs_df <- merge(onethree_twoouts_runs_df, runs_df, all = TRUE, sort = FALSE)



onethree_twoouts_runs_df$runsReal <- onethree_twoouts_runs_df$runs - onethree_twoouts_runs_df$runsbefor



mean(onethree_twoouts_runs_df$runsReal, na.rm = T)
```

```
[1] 0.4947
```

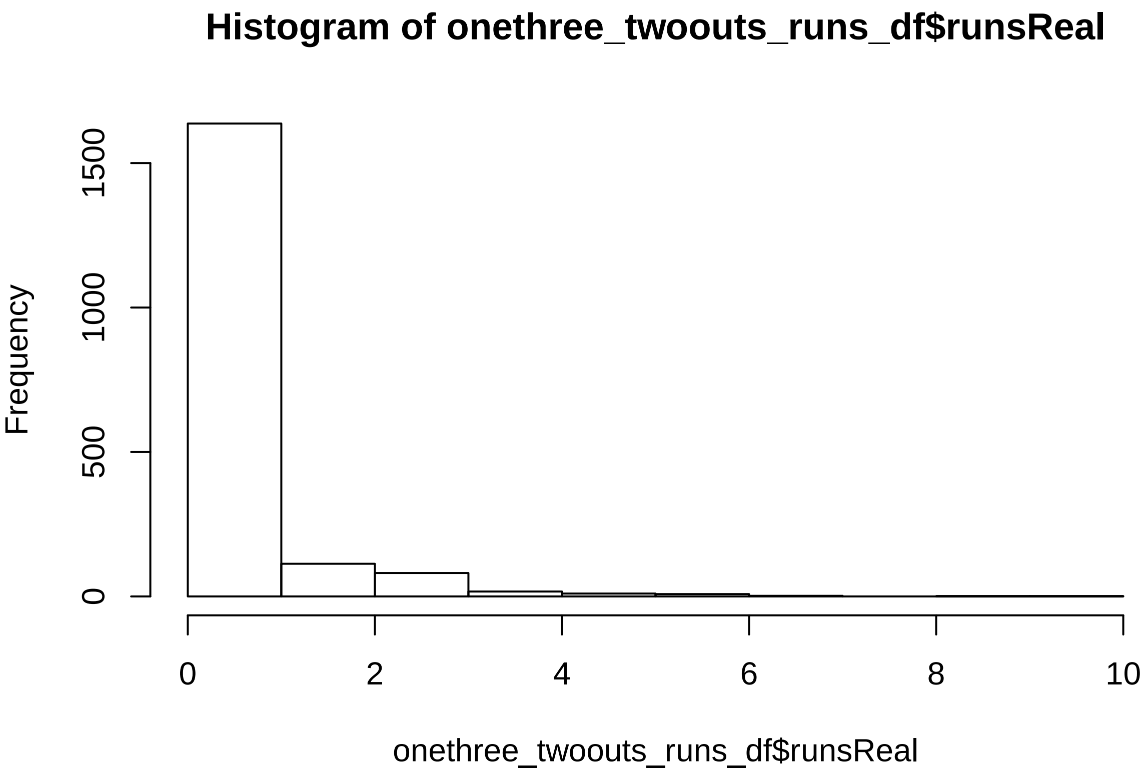
```
table(onethree_twoouts_runs_df$runsReal)
```

```

              0              0.5                1              1.5 1.66666666666667
           1353               32              252               11                2
              2 2.33333333333333              2.5                3              3.5
            100                1                6               74                2
              4                5                6                7                9
             15               10                8                2                1
             10
              1
```

```
sum(onethree_twoouts_runs_df$runsReal)/length(onethree_twoouts_runs_df$runsReal)
```

```
[1] NA
```

```
hist(onethree_twoouts_runs_df$runsReal)
```



```
standarderror18 <- sd(onethree_twoouts_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_
    "Runners on 1st and 3rd, Two outs")))
standarderror18
```

```
[1] 0.02352
```

```
# 95% confidence intervals of the mean
onethree_twoouts_runs_df$ll <- mean(onethree_twoouts_runs_df$runsReal, na.rm = T) - 1.96 * standarderro
onethree_twoouts_runs_df$ul <- mean(onethree_twoouts_runs_df$runsReal, na.rm = T) + 1.96 * standarderro

onethree_twoouts_runs_df$ul[1]
```

```
[1] 0.5408
```

```
onethree_twoouts_runs_df$ll[1]
```

```
[1] 0.4486
```

```
mean(onethree_twoouts_runs_df$runsReal, na.r = T) - 0.251/(standarderror18)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

31

```
library(dplyr)

twothree_noout_runs_df <- season[season$base_state_before == "Runners on 2nd and 3rd, No outs", ] %>%
    dplyr::group_by(gameID, inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_pla
    na.rm = T))

twothree_noout_runs_df <- merge(twothree_noout_runs_df, runs_df, all = TRUE, sort = FALSE)



twothree_noout_runs_df$runsReal <- twothree_noout_runs_df$runs - twothree_noout_runs_df$runsbefore



mean(twothree_noout_runs_df$runsReal, na.rm = T)
```

```
[1] 2.09
```

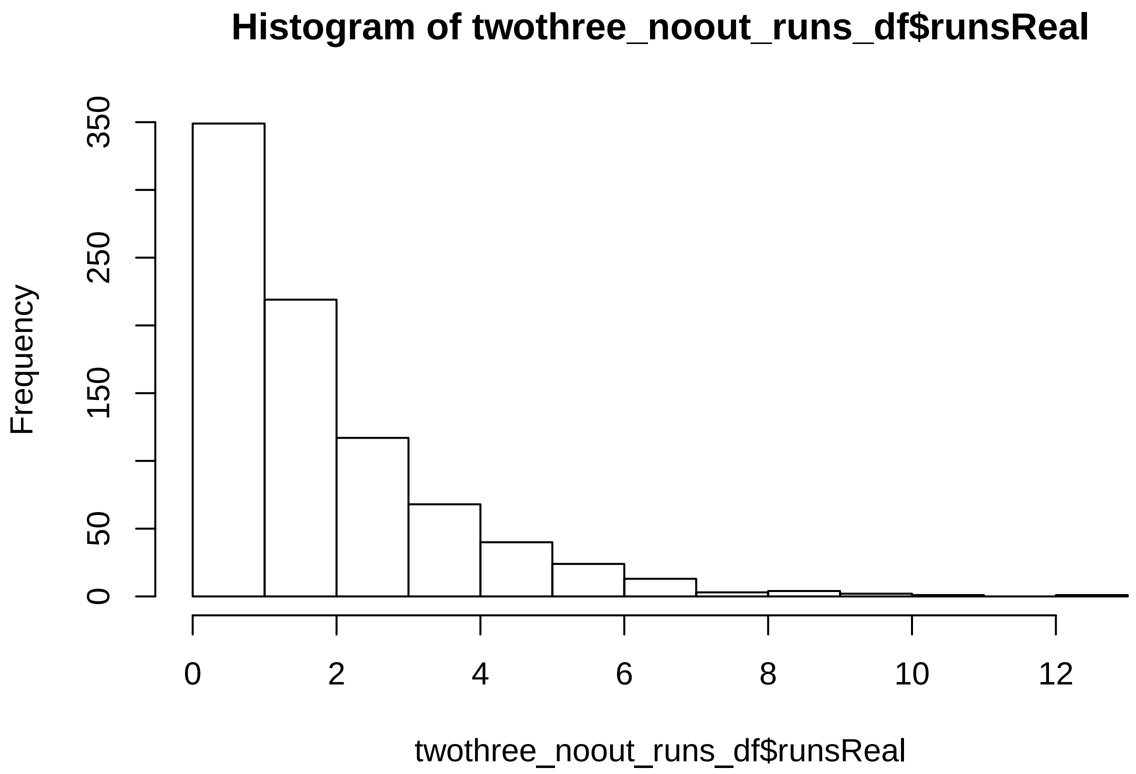
```
table(twothree_noout_runs_df$runsReal)
```

```

               0              0.5                1              1.5                2 
             162               11              176               15              204 
             2.5                3 3.33333333333333              3.5                4 
              16              101                1                7               60 
             4.5                5 5.33333333333333              5.5 5.66666666666667 
               3               37                1                2                1 
               6                7                8              8.5                9 
              20               13                3                1                3 
              10               11               13 
               2                1                1 
```

```
sum(twothree_noout_runs_df$runsReal)/length(twothree_noout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(twothree_noout_runs_df$runsReal)
```



```
standarderror19 <- sd(twothree_noout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_be
    "Runners on 2nd and 3rd, No outs")))
standarderror19
```

```
[1] 0.06057
```

```
# 95% confidence intervals of the mean
twothree_noout_runs_df$ll <- mean(twothree_noout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror19
twothree_noout_runs_df$ul <- mean(twothree_noout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror19

twothree_noout_runs_df$ul[1]
```

```
[1] 2.209
```

```
twothree_noout_runs_df$ll[1]
```

```
[1] 1.971
```

```
mean(twothree_noout_runs_df$runsReal, na.r = T) - 0.251/(standarderror19)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

32

```
library(dplyr)

twothree_oneout_runs_df <- season[season$base_state_before == "Runners on 2nd and 3rd, One out", ] %>%
    dplyr::group_by(gameID, inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_pla
    na.rm = T))

twothree_oneout_runs_df <- merge(twothree_oneout_runs_df, runs_df, all = TRUE, sort = FALSE)



twothree_oneout_runs_df$runsReal <- twothree_oneout_runs_df$runs - twothree_oneout_runs_df$runsbefore



mean(twothree_oneout_runs_df$runsReal, na.rm = T)
```

```
[1] 1.336
```

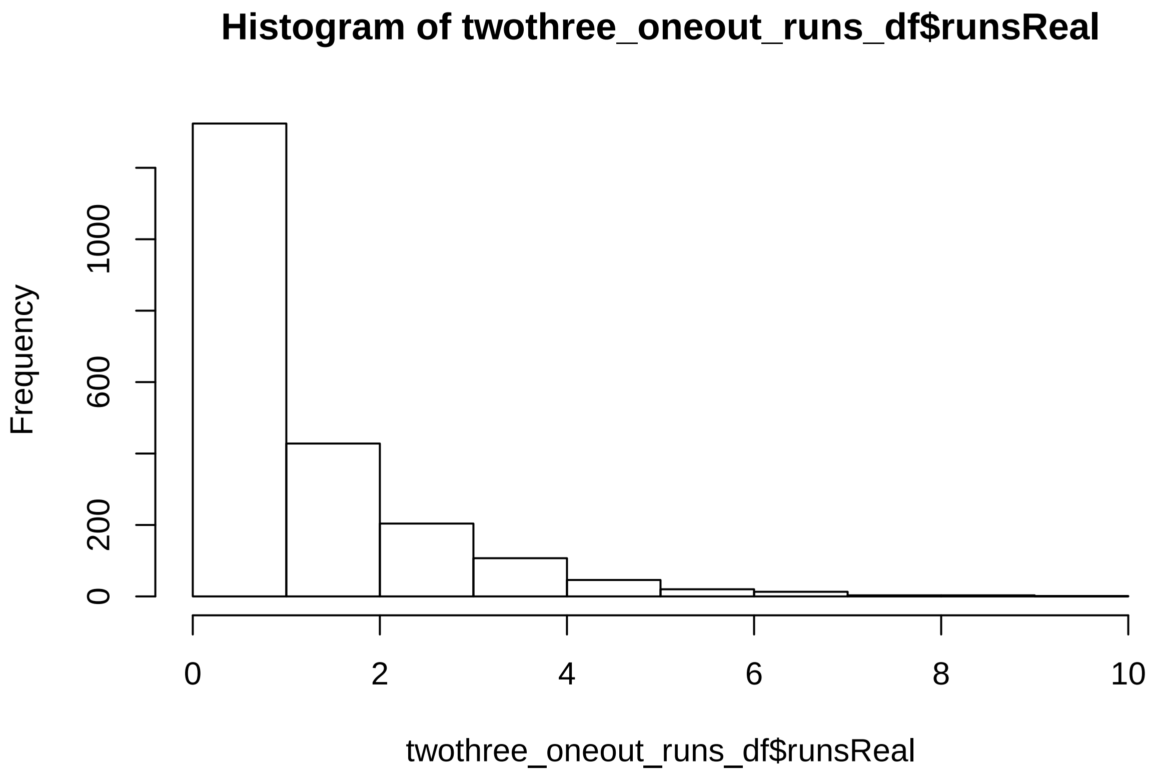
```
table(twothree_oneout_runs_df$runsReal)
```

```

                 0 0.333333333333333               0.5                 1               1.5
               780                 1                40               503                33
  1.66666666666667                 2  2.33333333333333               2.5  2.66666666666667
                 1               394                 1                28                 1
                 3              3.25               3.5  3.66666666666667                 4
               174                 1                12                 3                91
              4.25  4.33333333333333               4.5  4.66666666666667                 5
                 1                 1                 9                 1                34
               5.5                 6               6.5  6.66666666666667                 7
                 4                16                 1                 1                11
  7.33333333333333                 8               8.5                 9                10
                 1                 2                 1                 2                 1
```

```
sum(twothree_oneout_runs_df$runsReal)/length(twothree_oneout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(twothree_oneout_runs_df$runsReal)
```



```
standarderror20 <- sd(twothree_oneout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_
    "Runners on 2nd and 3rd, One out")))
standarderror20
```

```
[1] 0.03022
```

```
# 95% confidence intervals of the mean
twothree_oneout_runs_df$ll <- mean(twothree_oneout_runs_df$runsReal, na.rm = T) - 1.96 * standarderror2
twothree_oneout_runs_df$ul <- mean(twothree_oneout_runs_df$runsReal, na.rm = T) + 1.96 * standarderror2

twothree_oneout_runs_df$ul[1]
```

```
[1] 1.395
```

```
twothree_oneout_runs_df$ll[1]
```

```
[1] 1.277
```

```
mean(twothree_oneout_runs_df$runsReal, na.r = T) - 0.251/(standarderror20)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

33

```
library(dplyr)

twothree_twoouts_runs_df <- season[season$base_state_before == "Runners on 2nd and 3rd, Two outs", ] %>
    dplyr::group_by(gameID, inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_pl
    na.rm = T))

twothree_twoouts_runs_df <- merge(twothree_twoouts_runs_df, runs_df, all = TRUE, sort = FALSE)



twothree_twoouts_runs_df$runsReal <- twothree_twoouts_runs_df$runs - twothree_twoouts_runs_df$runsbefor



mean(twothree_twoouts_runs_df$runsReal, na.rm = T)
```

```
[1] 0.5648
```

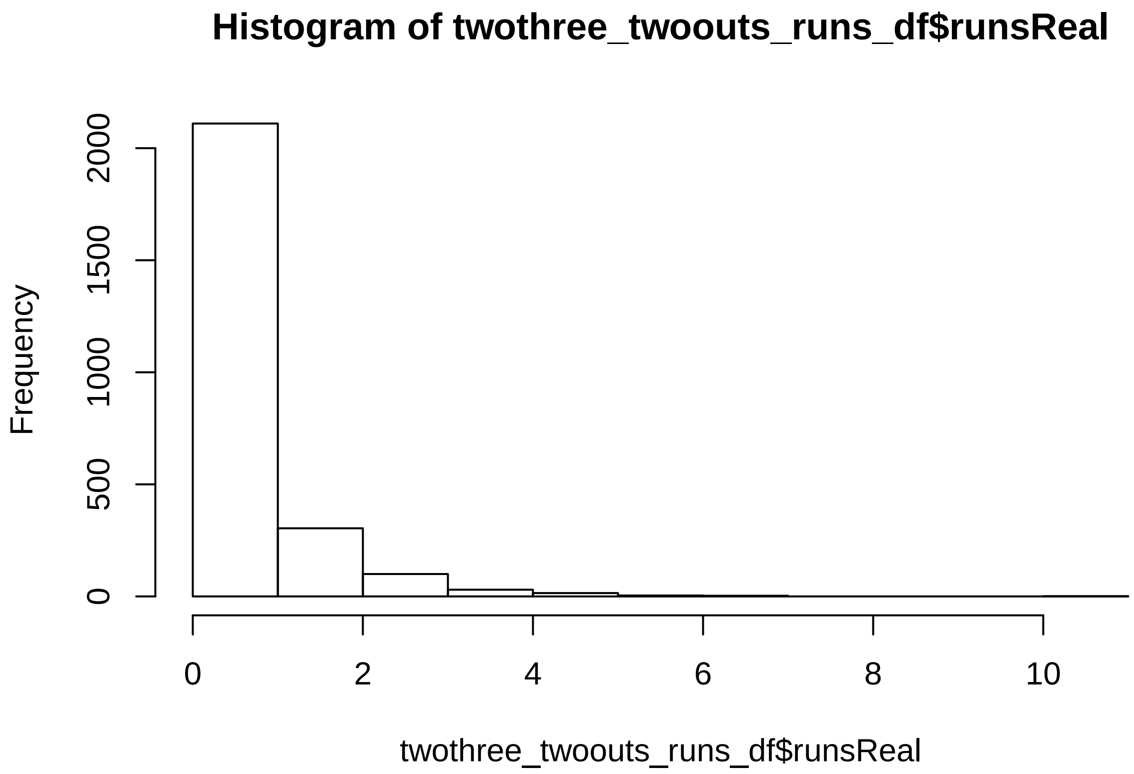
```
table(twothree_twoouts_runs_df$runsReal)
```

```

               0              0.5                1 1.33333333333333              1.5
            1779               36              295                1               19
1.66666666666667                2 2.33333333333333              2.5 2.66666666666667
               2              282                2                9                1
               3             3.25 3.33333333333333              3.5                4
              88                1                1                3               25
             4.5                5                6              6.5                7
               2               13                4                1                2
10.3333333333333
               1
```

```
sum(twothree_twoouts_runs_df$runsReal)/length(twothree_twoouts_runs_df$runsReal)
```

```
[1] NA
```

```
hist(twothree_twoouts_runs_df$runsReal)
```



```
standarderror21 <- sd(twothree_twoouts_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_state_
    "Runners on 2nd and 3rd, Two outs")))
standarderror21
```

```
[1] 0.01984
```

```
# 95% confidence intervals of the mean
twothree_twoouts_runs_df$ll <- mean(twothree_twoouts_runs_df$runsReal, na.rm = T) - 1.96 * standarderro
twothree_twoouts_runs_df$ul <- mean(twothree_twoouts_runs_df$runsReal, na.rm = T) + 1.96 * standarderro

twothree_twoouts_runs_df$ul[1]
```

```
[1] 0.6037
```

```
twothree_twoouts_runs_df$ll[1]
```

```
[1] 0.5259
```

```
mean(twothree_twoouts_runs_df$runsReal, na.r = T) - 0.251/(standarderror21)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

34

```
library(dplyr)

onetwothree_noouts_runs_df <- season[season$base_state_before == "Bases Loaded, No outs", ] %>% dplyr::g
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))
```

```
onetwothree_noouts_runs_df <- merge(onetwothree_noouts_runs_df, runs_df, all = TRUE, sort = FALSE)

onetwothree_noouts_runs_df$runsReal <- onetwothree_noouts_runs_df$runs - onetwothree_noouts_runs_df$runs

mean(onetwothree_noouts_runs_df$runsReal, na.rm = T)
```

```
[1] 2.294
```

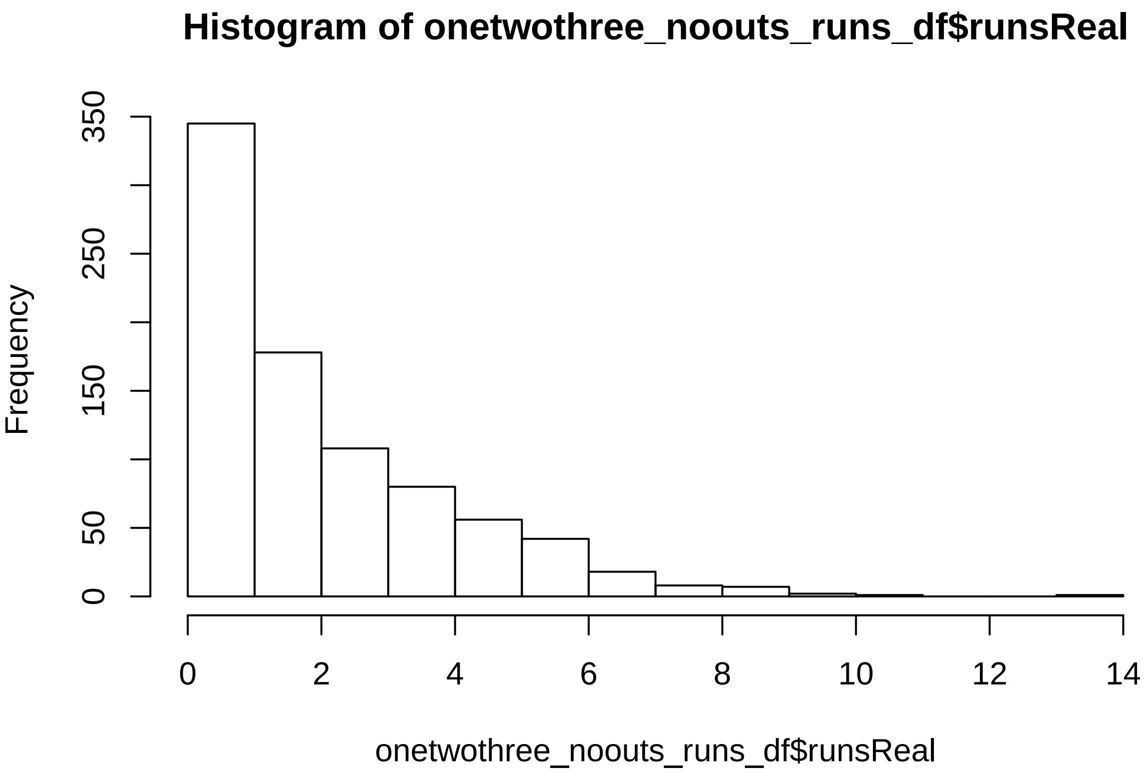
```
table(onetwothree_noouts_runs_df$runsReal)
```

```

               0              0.5                1              1.5 1.66666666666667 
             153               28              164               31                1 
               2 2.33333333333333              2.5             2.75                3 
             146                1               23                1               83 
3.33333333333333              3.5 3.66666666666667             3.75                4 
               1               22                2                1               54 
             4.5              4.6                5 5.11111111111111              5.5 
              19                1               36                1               11 
5.66666666666667             5.75                6              6.5 6.66666666666667 
               1                2               27               10                1 
               7              7.4              7.5                8              8.5 
               7                1                2                5                2 
             8.6                9              9.5               10               11 
               1                4                1                1                1 
13.3333333333333 
               1 
```

```
sum(onetwothree_noouts_runs_df$runsReal)/length(onetwothree_noouts_runs_df$runsReal)
```

```
[1] NA
```

```
hist(onetwothree_noouts_runs_df$runsReal)
```



```
standarderror22 <- sd(onetwothree_noouts_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_sta
    "Bases Loaded, No outs")))
standarderror22
```

```
[1] 0.05988
```

```
# 95% confidence intervals of the mean
onetwothree_noouts_runs_df$ll <- mean(onetwothree_noouts_runs_df$runsReal, na.rm = T) - 1.96 * standarde
onetwothree_noouts_runs_df$ul <- mean(onetwothree_noouts_runs_df$runsReal, na.rm = T) + 1.96 * standarde

onetwothree_noouts_runs_df$ul[1]
```

```
[1] 2.411
```

```
onetwothree_noouts_runs_df$ll[1]
```

```
[1] 2.176
```

```
mean(onetwothree_noouts_runs_df$runsReal, na.r = T) - 0.251/(standarderror22)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

35

```
library(dplyr)

onetwothree_oneout_runs_df <- season[season$base_state_before == "Bases Loaded, One out", ] %>% dplyr::
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))
```

```
onetwothree_oneout_runs_df <- merge(onetwothree_oneout_runs_df, runs_df, all = TRUE, sort = FALSE)



onetwothree_oneout_runs_df$runsReal <- onetwothree_oneout_runs_df$runs - onetwothree_oneout_runs_df$runs



mean(onetwothree_oneout_runs_df$runsReal, na.rm = T)
```

```
[1] 1.504
```

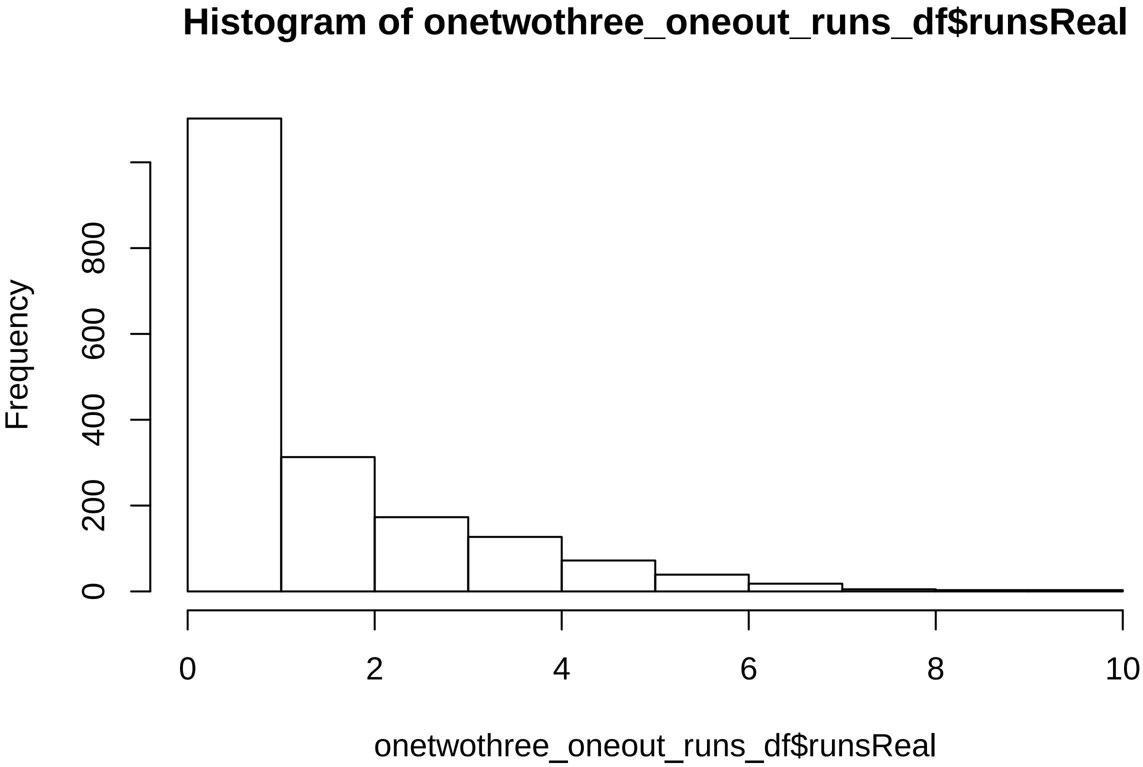
```
table(onetwothree_oneout_runs_df$runsReal)
```

```

                0               0.5 0.666666666666667                 1  1.33333333333333
              618               116                 1               367                 4
              1.5  1.66666666666667                 2               2.2  2.33333333333333
               73                 8               228                 1                 2
              2.4               2.5  2.66666666666667              2.75               2.8
                1                49                 1                 1                 1
                3  3.16666666666667               3.2  3.33333333333333               3.5
              117                 1                 1                 1                22
 3.66666666666667              3.75                 4               4.5  4.66666666666667
                3                 1                98                25                 1
             4.75                 5              5.25               5.5  5.66666666666667
                1                45                 1                 7                 1
                6  6.33333333333333               6.5  6.66666666666667                 7
               30                 1                 3                 1                13
             7.25               7.6                 8                 9               9.5
                1                 1                 3                 3                 1
               10
                2
```

```
sum(onetwothree_oneout_runs_df$runsReal)/length(onetwothree_oneout_runs_df$runsReal)
```

```
[1] NA
```

```
hist(onetwothree_oneout_runs_df$runsReal)
```



```
standarderror23 <- sd(onetwothree_oneout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_sta
    "Bases Loaded, One out")))
standarderror23
```

```
[1] 0.03394
```

```
# 95% confidence intervals of the mean
onetwothree_oneout_runs_df$ll <- mean(onetwothree_oneout_runs_df$runsReal, na.rm = T) - 1.96 * standard
onetwothree_oneout_runs_df$ul <- mean(onetwothree_oneout_runs_df$runsReal, na.rm = T) + 1.96 * standard

onetwothree_oneout_runs_df$ul[1]
```

```
[1] 1.57
```

```
onetwothree_oneout_runs_df$ll[1]
```

```
[1] 1.437
```

```
mean(onetwothree_oneout_runs_df$runsReal, na.r = T) - 0.251/(standarderror23)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

36

```
library(dplyr)

onetwothree_twoout_runs_df <- season[season$base_state_before == "Bases Loaded, Two outs", ] %>% dplyr:
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))
```

```r
onetwothree_twoout_runs_df <- merge(onetwothree_twoout_runs_df, runs_df, all = TRUE, sort = FALSE)



onetwothree_twoout_runs_df$runsReal <- onetwothree_twoout_runs_df$runs - onetwothree_twoout_runs_df$runs



mean(onetwothree_twoout_runs_df$runsReal, na.rm = T)
```

```
[1] 0.6175
```

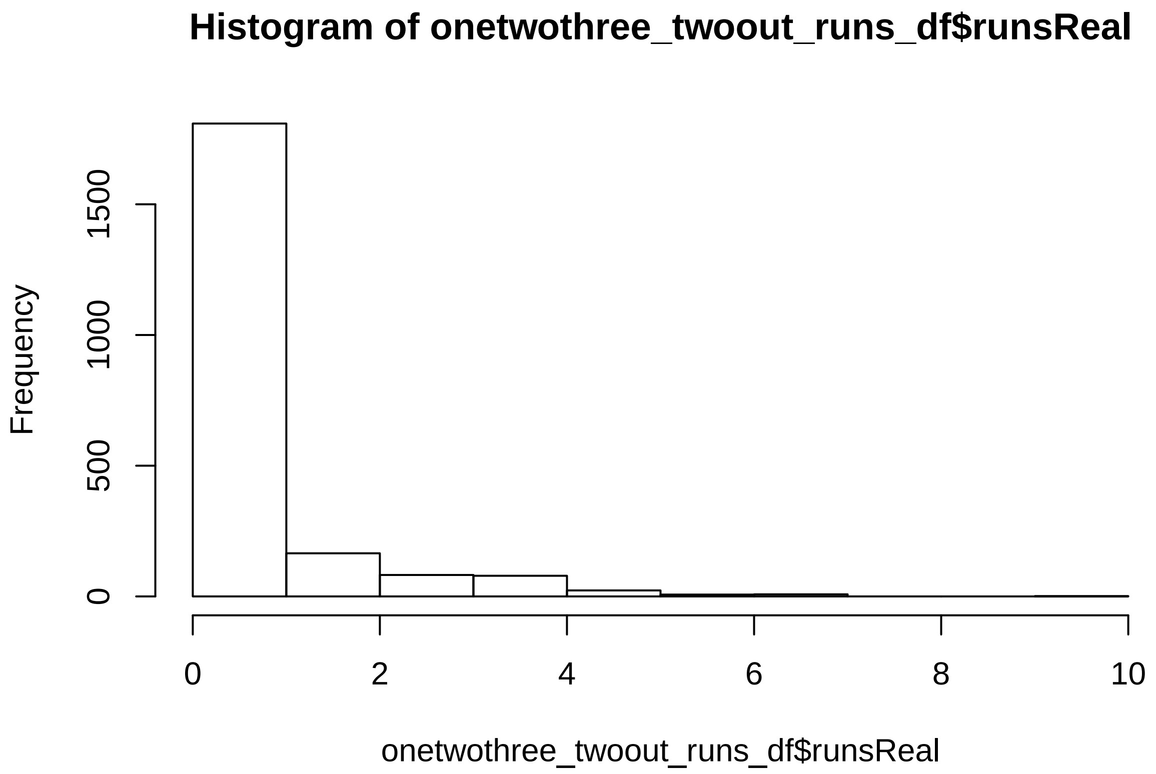
```r
table(onetwothree_twoout_runs_df$runsReal)
```

```

                0               0.5 0.666666666666667                 1  1.33333333333333 
             1433               217                 4               155                 2 
              1.5  1.66666666666667              1.75                 2               2.2 
               17                 3                 3               140                 1 
 2.33333333333333               2.4               2.5               2.8                 3 
                1                 1                12                 1                66 
              3.2               3.5                 4  4.33333333333333               4.5 
                1                15                63                 2                 9 
 4.66666666666667                 5               5.5  5.66666666666667                 6 
                1                11                 3                 1                 3 
 6.33333333333333               6.5                 7               9.5 
                1                 3                 4                 1 
```

```r
sum(onetwothree_twoout_runs_df$runsReal)/length(onetwothree_twoout_runs_df$runsReal)
```

```
[1] NA
```

```r
hist(onetwothree_twoout_runs_df$runsReal)
```



```
standarderror24 <- sd(onetwothree_twoout_runs_df$runsReal, na.rm = T)/sqrt(length(which(season$base_sta
    "Bases Loaded, Two outs")))
standarderror24
```

```
[1] 0.02304
```

```
# 95% confidence intervals of the mean
onetwothree_twoout_runs_df$ll <- mean(onetwothree_twoout_runs_df$runsReal, na.rm = T) - 1.96 * standard
onetwothree_twoout_runs_df$ul <- mean(onetwothree_twoout_runs_df$runsReal, na.rm = T) + 1.96 * standard

onetwothree_twoout_runs_df$ul[1]
```

```
[1] 0.6626
```

```
onetwothree_twoout_runs_df$ll[1]
```

```
[1] 0.5723
```

```
mean(onetwothree_twoout_runs_df$runsReal, na.r = T) - 0.251/(standarderror24)
```

```
[1] NA
```

```
# The second element of the chart is .543 - number cited in 'The Book' is .297 for the MLB
```

37

```
season$playType[grepl("[[:space:]]singled", season$play)] <- "single"
```

```
Warning: Unknown or uninitialised column: 'playType'.
```

```r
season$playType[grepl("[[:space:]]doubled", season$play)] <- "double"

season$playType[grepl("[[:space:]]tripled", season$play)] <- "triple"

season$playType[grepl("[[:space:]]homered", season$play)] <- "home run"
season$playType[grepl("[[:space:]]homered", season$play)] <- "home run"
season$playType[grepl("[[:space:]]walked", season$play)] <- "walk"

season$playType[grepl("[[:space:]]reached[[:space:]]on[[:space:]]an[[:space:]]error", season$play)] <-
season$playType[grepl("[[:space:]]catcher's[[:space:]]interference", season$play)] <- "catcher's interfe
season$playType[grepl("[[:space:]]passed[[:space:]]ball", season$play)] <- "passed ball"
season$playType[grepl("[[:space:]]wild[[:space:]]pitch", season$play)] <- "wild pitch"
season$playType[grepl("[[:space:]]hit[[:space:]]by[[:space:]]pitch", season$play)] <- "hit by pitch"

season$playType[grepl("[[:space:]]stole[[:space:]]", season$play)] <- "stolen base"


season$playType[grepl("[[:space:]]illegal[[:space:]]pitch", season$play)] <- "illegal pitch"

season$playType[grepl("[[:space:]]SAC[[:space:]]bunt", season$play)] <- "SAC bunt"



table(season$playType, season$runs_from_play)
```

```

                             0     1     2     3     4
  catcher's interference    31     4     0     0     0
  double                  2712  1527   650   105     0
  hit by pitch            2358   155     0     0     0
  home run                   0  1116   845   437   134
  illegal pitch            136    64     0     0     0
  passed ball              637   148     0     0     0
  reached on error        1064   300     9     0     0
  SAC bunt                1803    96     0     0     0
  single                 16652  4396   884     8     0
  stolen base             3903   144     0     0     0
  triple                   310   243   117    27     0
  walk                   11188   598     0     0     0
  wild pitch              1729   370     0     0     0
```

38

```r
season$runexp <- NA
season$runexp[season$base_state_before == "Nobody on, No Outs"] <- 0.656
season$runexp[season$base_state_before == "Nobody on, One out"] <- 0.31
season$runexp[season$base_state_before == "Nobody on, Two outs"] <- 0.097

season$runexp[season$base_state_before == "Runner on 1st, No outs"] <- 1.1
season$runexp[season$base_state_before == "Runner on 1st, One out"] <- 0.603
season$runexp[season$base_state_before == "Runner on 1st, Two outs"] <- 0.23
```

```r
season$runexp[season$base_state_before == "Runner on 2nd, No outs"] <- 1.395
season$runexp[season$base_state_before == "Runner on 2nd, One out"] <- 0.77
season$runexp[season$base_state_before == "Runner on 2nd, Two outs"] <- 0.301

season$runexp[season$base_state_before == "Runner on 3rd, No outs"] <- 1.517
season$runexp[season$base_state_before == "Runner on 3rd, One out"] <- 1.029
season$runexp[season$base_state_before == "Runner on 3rd, Two outs"] <- 0.411

season$runexp[season$base_state_before == "Runners on 1st and 2nd, No outs"] <- 1.769
season$runexp[season$base_state_before == "Runners on 1st and 2nd, One out"] <- 1.052
season$runexp[season$base_state_before == "Runners on 1st and 2nd, Two outs"] <- 0.454

season$runexp[season$base_state_before == "Runners on 1st and 3rd, No outs"] <- 1.948
season$runexp[season$base_state_before == "Runners on 1st and 3rd, One out"] <- 1.24
season$runexp[season$base_state_before == "Runners on 1st and 3rd, Two outs"] <- 0.495

season$runexp[season$base_state_before == "Runners on 2nd and 3rd, No outs"] <- 2.089
season$runexp[season$base_state_before == "Runners on 2nd and 3rd, One out"] <- 1.336
season$runexp[season$base_state_before == "Runners on 2nd and 3rd, Two outs"] <- 0.566

season$runexp[season$base_state_before == "Bases Loaded, No outs"] <- 2.294
season$runexp[season$base_state_before == "Bases Loaded, One out"] <- 1.506
season$runexp[season$base_state_before == "Bases Loaded, Two outs"] <- 0.616

mean(season$runexp[season$playType == "home run"], na.rm = T)
```

```
[1] 0.6715
```

```r
mean(season$runexp[season$playType == "single"], na.rm = T)
```

```
[1] 0.667
```

```r
mean(season$runexp[season$playType == "double"], na.rm = T)
```

```
[1] 0.6801
```

```r
mean(season$runexp[season$playType == "triple"], na.rm = T)
```

```
[1] 0.6437
```

```r
mean(season$runexp[season$playType == "walk"], na.rm = T)
```

```
[1] 0.6514
```

```r
mean(season$runexp[season$playType == "reached on error"], na.rm = T)
```

```
[1] 0.6477
```

```r
mean(season$runexp[season$playType == "catcher's interference"], na.rm = T)
```

```
[1] 0.6378
```

```r
mean(season$runexp[season$playType == "passed ball"], na.rm = T)
```

```
[1] 0.7981
```

```r
mean(season$runexp[season$playType == "wild pitch"], na.rm = T)
```

```
[1] 0.8098
```

```r
mean(season$runexp[season$playType == "illegal pitch"], na.rm = T)
```

```
[1] 0.9077
```

```r
mean(season$runexp[season$playType == "stolen base"], na.rm = T)
```

```
[1] 0.8169
```

```r
mean(season$runexp[season$playType == "hit by pitch"], na.rm = T)
```

```
[1] 0.6644
```

```r
mean(season$runexp[season$playType == "SAC bunt"], na.rm = T)
```

```
[1] 1.186
```

```r
mean(season$runexp[season$playType == "bunt"], na.rm = T)
```

```
[1] NaN
```

```r
mean(season$runexp[season$playType == "out"], na.rm = T)
```

```
[1] NaN
```

```r
mean(season$runexp[season$playType == "stolen base"], na.rm = T)
```

```
[1] 0.8169
```

```r
mean(season$runexp[season$playType == "strikeout"], na.rm = T)
```

```
[1] NaN
```

39

```r
# Home Run

homerun_df <- season[season$playType == "home run", ] %>% dplyr::group_by(gameID, inningID, top_bottom)
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

homerun_df <- merge(homerun_df, runs_df, all = TRUE, sort = FALSE)

# Runs to the end of inning after a home run

homerun_df$runsReal <- homerun_df$runs - homerun_df$runsbefore

# RTOE (average) of a HR

mean(homerun_df$runsReal, na.rm = T)
```

```
[1] 2.169
```

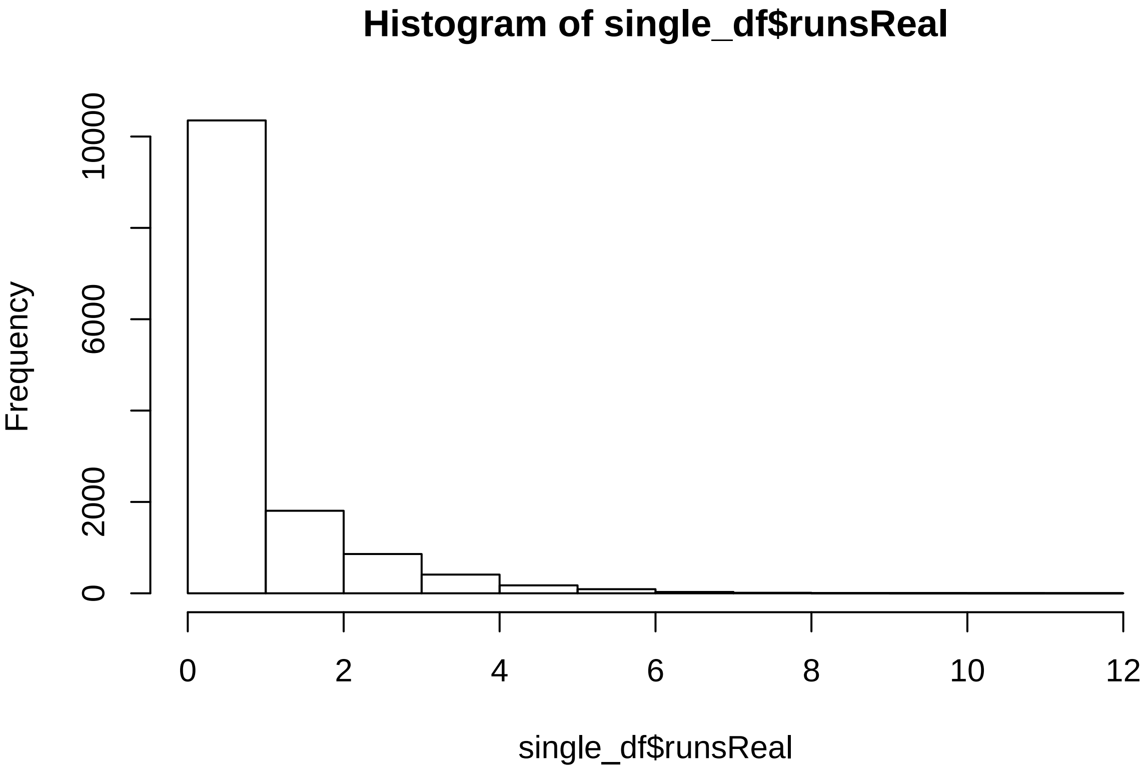
```r
standarderrorhomerun <- sd(homerun_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "home ru
```

40

```
single_df <- season[season$playType == "single", ] %>% dplyr::group_by(gameID, inningID, top_bottom) %>
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

single_df <- merge(single_df, runs_df, all = TRUE, sort = FALSE)



single_df$runsReal <- single_df$runs - single_df$runsbefore



mean(single_df$runsReal, na.rm = T)
```

```
[1] 0.8598
```

```
hist(single_df$runsReal)
```



```
standarderrorsingle <- sd(single_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "single")
```

41

```
double_df <- season[season$playType == "double", ] %>% dplyr::group_by(gameID, inningID, top_bottom) %>
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

double_df <- merge(double_df, runs_df, all = TRUE, sort = FALSE)
```

```
double_df$runsReal <- double_df$runs - double_df$runsbefore



mean(double_df$runsReal, na.rm = T)
```

```
[1] 1.483
```

```
standarderrordouble <- sd(double_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "double")
```

42

```
triple_df <- season[season$playType == "triple", ] %>% dplyr::group_by(gameID, inningID, top_bottom) %>
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

triple_df <- merge(triple_df, runs_df, all = TRUE, sort = FALSE)



triple_df$runsReal <- triple_df$runs - triple_df$runsbefore



mean(triple_df$runsReal, na.rm = T)
```

```
[1] 1.721
```

```
standarderrortriple <- sd(triple_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "triple")
```

43

```
walk_df <- season[season$playType == "walk", ] %>% dplyr::group_by(gameID, inningID, top_bottom) %>%
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

walk_df <- merge(walk_df, runs_df, all = TRUE, sort = FALSE)



walk_df$runsReal <- walk_df$runs - walk_df$runsbefore



mean(walk_df$runsReal, na.rm = T)
```

```
[1] 0.8748
```

```
standarderrorwalk <- sd(walk_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "walk")))
```

44

```r
error_df <- season[season$playType == "reached on error", ] %>% dplyr::group_by(gameID, inningID, top_bo
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

error_df <- merge(error_df, runs_df, all = TRUE, sort = FALSE)



error_df$runsReal <- error_df$runs - error_df$runsbefore



mean(error_df$runsReal, na.rm = T)
```

```
[1] 1.117
```

```r
standarderrorerror <- sd(error_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "reached on
standarderrorerror
```

```
[1] 0.04146
```

45

```r
interference_df <- season[season$playType == "catcher's interference", ] %>% dplyr::group_by(gameID,
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

interference_df <- merge(interference_df, runs_df, all = TRUE, sort = FALSE)



interference_df$runsReal <- interference_df$runs - interference_df$runsbefore



mean(interference_df$runsReal, na.rm = T)
```

```
[1] 0.9714
```

```r
standarderrorinterference <- sd(interference_df$runsReal, na.rm = T)/sqrt(length(which(season$playType =
    "catcher's interference")))
```

46

```r
passedball_df <- season[season$playType == "passed ball", ] %>% dplyr::group_by(gameID, inningID, top_bo
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

passedball_df <- merge(passedball_df, runs_df, all = TRUE, sort = FALSE)
```

```r
passedball_df$runsReal <- passedball_df$runs - passedball_df$runsbefore



mean(passedball_df$runsReal, na.rm = T)
```

```
[1] 1.044
```

```r
standarderrorpassedball <- sd(passedball_df$runsReal, na.rm = T)/sqrt(length(which(season$playType ==
    "passed ball")))
```

47

```r
wildpitch_df <- season[season$playType == "wild pitch", ] %>% dplyr::group_by(gameID, inningID, top_bot
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

wildpitch_df <- merge(wildpitch_df, runs_df, all = TRUE, sort = FALSE)



wildpitch_df$runsReal <- wildpitch_df$runs - wildpitch_df$runsbefore



mean(wildpitch_df$runsReal, na.rm = T)
```

```
[1] 1.077
```

```r
standarderrorwildpitch <- sd(passedball_df$runsReal, na.rm = T)/sqrt(length(which(season$playType ==
    "wild pitch")))
```

48

```r
illegalpitch_df <- season[season$playType == "illegal pitch", ] %>% dplyr::group_by(gameID, inningID,
    top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

illegalpitch_df <- merge(illegalpitch_df, runs_df, all = TRUE, sort = FALSE)



illegalpitch_df$runsReal <- illegalpitch_df$runs - illegalpitch_df$runsbefore



mean(illegalpitch_df$runsReal, na.rm = T)
```

```
[1] 1.363
```

```r
standarderrorillegalpitch <- sd(illegalpitch_df$runsReal, na.rm = T)/sqrt(length(which(season$playType =
    "illegal pitch")))
```

49

```
steal_df <- season[season$playType == "stolen base", ] %>% dplyr::group_by(gameID, inningID, top_bottom
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

steal_df <- merge(steal_df, runs_df, all = TRUE, sort = FALSE)



steal_df$runsReal <- steal_df$runs - steal_df$runsbefore



mean(steal_df$runsReal, na.rm = T)
```

```
[1] 1.011
```

```
standarderrorsteal <- sd(steal_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "stolen bas
```

50

```
HBP_df <- season[season$playType == "hit by pitch", ] %>% dplyr::group_by(gameID, inningID, top_bottom)
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

HBP_df <- merge(HBP_df, runs_df, all = TRUE, sort = FALSE)



HBP_df$runsReal <- HBP_df$runs - HBP_df$runsbefore



mean(HBP_df$runsReal, na.rm = T)
```

```
[1] 1.03
```

```
standarderrorhitbypitch <- sd(HBP_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "hit by
```

51

```
season$playType[grepl("[[:space:]]SAC[[:space:]]bunt", season$play)] <- "SAC bunt"


SAC_df <- season[season$playType == "SAC bunt", ] %>% dplyr::group_by(gameID, inningID, top_bottom) %>%
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

SAC_df <- merge(SAC_df, runs_df, all = TRUE, sort = FALSE)
```

```r
SAC_df$runsReal <- SAC_df$runs - SAC_df$runsbefore



mean(SAC_df$runsReal, na.rm = T)
```

```
[1] 0.912
```

```r
standarderrorSAC <- sd(SAC_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "SAC bunt")))
```

52

```r
season$playType[grepl("bunt", season$play)] <- "bunt"

bunt_df <- season[season$playType == "bunt", ] %>% dplyr::group_by(gameID, inningID, top_bottom) %>%
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

bunt_df <- merge(bunt_df, runs_df, all = TRUE, sort = FALSE)



bunt_df$runsReal <- bunt_df$runs - bunt_df$runsbefore



mean(bunt_df$runsReal, na.rm = T)
```

```
[1] 0.9442
```

```r
standarderrorbunt <- sd(bunt_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "bunt")))
```

53

```r
season$playType[season$outs_from_play == 1] <- "out"

out_df <- season[season$playType == "out", ] %>% dplyr::group_by(gameID, inningID, top_bottom) %>% dply
    na.rm = T))

out_df <- merge(out_df, runs_df, all = TRUE, sort = FALSE)



out_df$runsReal <- out_df$runs - out_df$runsbefore


mean(out_df$runsReal, na.rm = T)
```

```
[1] 0.2957
```

```r
standarderrorout <- sd(out_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "out")))
```

54

```
season$playType[grepl("[[:space:]]struck[[:space:]]out", season$play)] <- "strikeout"


K_df <- season[season$playType == "strikeout", ] %>% dplyr::group_by(gameID, inningID, top_bottom) %>%
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

K_df <- merge(K_df, runs_df, all = TRUE, sort = FALSE)



K_df$runsReal <- K_df$runs - K_df$runsbefore


mean(K_df$runsReal, na.rm = T)
```

```
[1] 0.242
```

```
standarderrorK <- sd(K_df$runsReal, na.rm = T)/sqrt(length(which(season$playType == "strikeout")))
```

55

```
# Creating a dataset for play types (Run Exp based on ~{37})
standarderrorsplays <- c(standarderrorhomerun, standarderrortriple, standarderrordouble, standarderrors
    standarderrorwalk, standarderrorhitbypitch, standarderrorout, standarderrorK, standarderrorinterfer
    standarderrorerror, standarderrorillegalpitch, standarderrorwildpitch, standarderrorpassedball, sta
    standarderrorSAC, standarderrorbunt)
playType <- c("home run", "triple", "double", "single", "walk", "hit by pitch", "out", "strikeout", "ca
    "error", "illegal pitch", "wild pitch", "passed ball", "stolen base", "SAC bunt", "bunt")
RTOE <- c(2.169, 1.721, 1.483, 0.86, 0.875, 1.03, 0.296, 0.242, 0.9714, 1.117, 1.363, 1.077, 1.044, 1.0
    0.912, 0.9442)
run_exp <- c(0.6727, 0.6413, 0.6765, 0.654, 0.6519, 0.6649, 0.6473, 0.6282, 0.6378, 0.643, 0.916, 0.811
    0.8087, 0.814, 1.186, 0.8513)
playvalue <- data.frame(playType, RTOE, run_exp)

# Expected Gain from event

playvalue$ExpectedGain <- playvalue$RTOE - playvalue$run_exp

# Value of an out
playvalue$outvalue <- 0.296


# The base value of an event relative to the out

playvalue$OBAcoeffbase <- with(playvalue, RTOE - outvalue)

# Applying the base value coefficients

Softball_Stats2016$mybaseOBA <- (0.579 * Softball_Stats2016$BB + 0.734 * Softball_Stats2016$HBP + 0.564
    Softball_Stats2016$oneB + 1.187 * Softball_Stats2016$twoB + 1.425 * Softball_Stats2016$threeB + 1.8
    Softball_Stats2016$HR + 0.821 * Softball_Stats2016$RBOE)/(Softball_Stats2016$AB + Softball_Stats201
    Softball_Stats2016$IBB + Softball_Stats2016$SH + Softball_Stats2016$SF + Softball_Stats2016$HBP)

# Discerning the weight needed so the average of myOBA is relatively equal to OBP

mean(Softball_Stats2016$mybaseOBA)
```

```
[1] 0.2845
```

```
Softball_Stats2016$OBP <- (Softball_Stats2016$BB + Softball_Stats2016$HBP + Softball_Stats2016$oneB +
    Softball_Stats2016$twoB + Softball_Stats2016$threeB + Softball_Stats2016$HR)/(Softball_Stats2016$AB
    Softball_Stats2016$BB + Softball_Stats2016$IBB + Softball_Stats2016$SH + Softball_Stats2016$SF +
    Softball_Stats2016$HBP)
mean(Softball_Stats2016$OBP)
```

```
[1] 0.3499
```

```
mean(Softball_Stats2016$OBP)/mean(Softball_Stats2016$mybaseOBA)
```

```
[1] 1.23
```

```
# Creating the OBA coefficients

playvalue$OBAcoeff <- with(playvalue, OBAcoeffbase * 1.23)
```

56

```
# Coding myOBA

Softball_Stats2016$myOBA <- (0.712 * Softball_Stats2016$BB + 0.903 * Softball_Stats2016$HBP + 0.694 *
    Softball_Stats2016$oneB + 1.46 * Softball_Stats2016$twoB + 1.753 * Softball_Stats2016$threeB + 2.30
    Softball_Stats2016$HR + 1.01 * Softball_Stats2016$RBOE)/(Softball_Stats2016$AB + Softball_Stats2016
    Softball_Stats2016$IBB + Softball_Stats2016$SH + Softball_Stats2016$SF + Softball_Stats2016$HBP)


# Creating a function marking winning % and myOBA
lmmwSoftballOBA2016 <- lm(with(Softball_Stats2016, PCT ~ myOBA))
lmmwSoftballOBA2016
```

```

Call:
lm(formula = with(Softball_Stats2016, PCT ~ myOBA))

Coefficients:
(Intercept)        myOBA
      -0.27         2.17
```

```
lmmwSoftballOBA2016R <- lm(with(Softball_Stats2016, R ~ myOBA))
lmmwSoftballOBA2016R
```

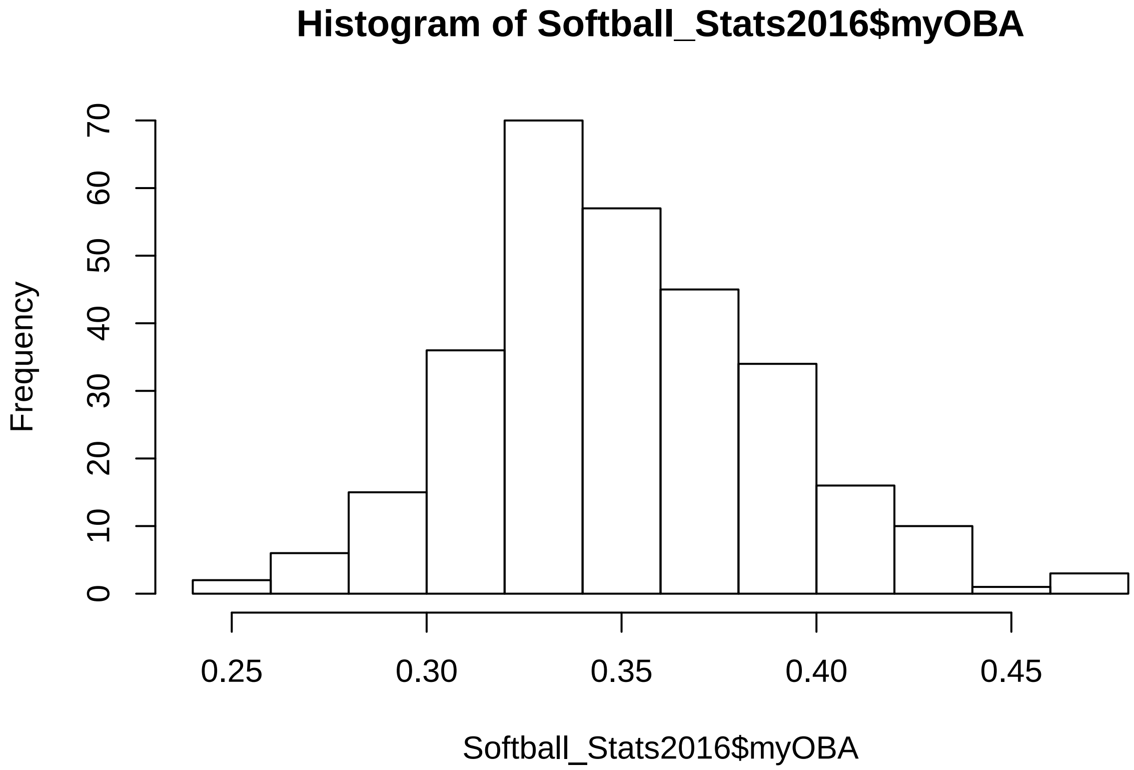
```

Call:
lm(formula = with(Softball_Stats2016, R ~ myOBA))
```

```
Coefficients:
(Intercept)        myOBA
       -373         1749
```

```
rsquared(lmmwSoftballOBA2016R)
```

```
[1] 0.8152
```

```
# Creating predictions based on the function

Softball_Stats2016$fit <- predict(lmmwSoftballOBA2016)
Softball_Stats2016$ul <- Softball_Stats2016$fit + 1.96 * predict(lmmwSoftballOBA2016, se.fit = T)$se
Softball_Stats2016$ll <- Softball_Stats2016$fit - 1.96 * predict(lmmwSoftballOBA2016, se.fit = T)$se







Softball_Stats2016$fit2 <- predict(lmmwSoftballOBA2016)
Softball_Stats2016$ul2 <- Softball_Stats2016$fit2 + 1.96 * predict(lmmwSoftballOBA2016, se.fit = T)$se
Softball_Stats2016$ll2 <- Softball_Stats2016$fit2 - 1.96 * predict(lmmwSoftballOBA2016, se.fit = T)$se



# Looking at the distribution of Softball wOBAs, looks normal
hist(Softball_Stats2016$myOBA)
```



```
# 2015

Softball_Stats2015$myOBA <- (0.712 * Softball_Stats2015$BB + 0.903 * Softball_Stats2015$HBP + 0.694 *
    Softball_Stats2015$oneB + 1.46 * Softball_Stats2015$twoB + 1.753 * Softball_Stats2015$threeB + 2.304
    Softball_Stats2015$HR + 1.01 * Softball_Stats2015$RBOE)/(Softball_Stats2015$AB + Softball_Stats2015$
    Softball_Stats2015$IBB + Softball_Stats2015$SH + Softball_Stats2015$SF + Softball_Stats2015$HBP)



lmmwSoftballOBA2015 <- lm(with(Softball_Stats2015, PCT ~ myOBA))
lmmwSoftballOBA2015
```

```

Call:
lm(formula = with(Softball_Stats2015, PCT ~ myOBA))

Coefficients:
(Intercept)        myOBA
     0.0099       1.3282
```

```
lmmwSoftballOBA2015R <- lm(with(Softball_Stats2015, R ~ myOBA))
lmmwSoftballOBA2015R
```

```

Call:
lm(formula = with(Softball_Stats2015, R ~ myOBA))

Coefficients:
(Intercept)        myOBA
       -419         1842
```

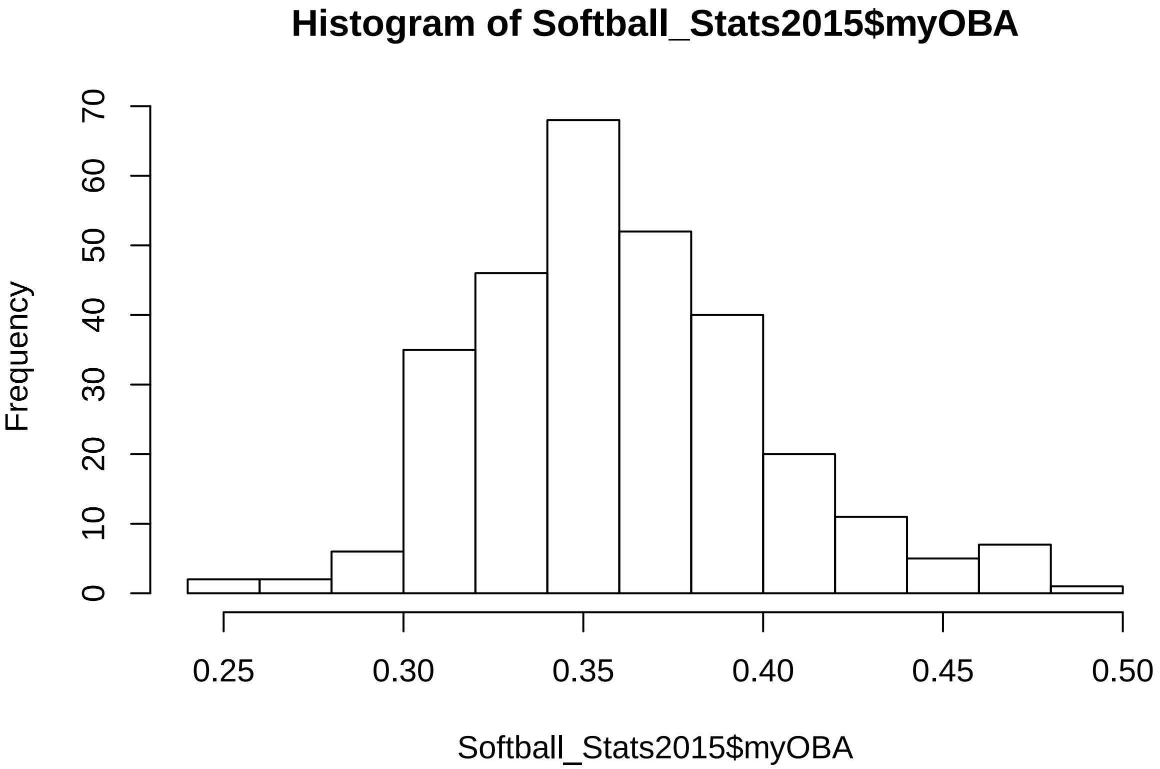
```
rsquared(lmmwSoftballOBA2015R)
```

```
[1] 0.8423
```

```
Softball_Stats2015$fit <- predict(lmmwSoftballOBA2015)
Softball_Stats2015$ul <- Softball_Stats2015$fit + 1.96 * predict(lmmwSoftballOBA2015, se.fit = T)$se
Softball_Stats2015$ll <- Softball_Stats2015$fit - 1.96 * predict(lmmwSoftballOBA2015, se.fit = T)$se






Softball_Stats2015$fit2 <- predict(lmmwSoftballOBA2015R)
Softball_Stats2015$ul2 <- Softball_Stats2015$fit2 + 1.96 * predict(lmmwSoftballOBA2015R, se.fit = T)$se
Softball_Stats2015$ll2 <- Softball_Stats2015$fit2 - 1.96 * predict(lmmwSoftballOBA2015R, se.fit = T)$se





rsquared(lmmwSoftballOBA2015)
```

```
[1] 0.09734
```

```
hist(Softball_Stats2015$myOBA)
```



```
# 2014

Softball_Stats2014$myOBA <- (0.712 * Softball_Stats2014$BB + 0.903 * Softball_Stats2014$HBP + 0.694 *
    Softball_Stats2014$oneB + 1.46 * Softball_Stats2014$twoB + 1.753 * Softball_Stats2014$threeB + 2.304
    Softball_Stats2014$HR + 1.01 * Softball_Stats2014$RBOE)/(Softball_Stats2014$AB + Softball_Stats2014$
    Softball_Stats2014$IBB + Softball_Stats2014$SH + Softball_Stats2014$SF + Softball_Stats2014$HBP)



lmmwSoftballOBA2014 <- lm(with(Softball_Stats2014, PCT ~ myOBA))
lmmwSoftballOBA2014
```

```

Call:
lm(formula = with(Softball_Stats2014, PCT ~ myOBA))

Coefficients:
(Intercept)        myOBA
      0.111        1.084
```

```
lmmwSoftballOBA2014R <- lm(with(Softball_Stats2014, R ~ myOBA))
lmmwSoftballOBA2014R
```

```

Call:
lm(formula = with(Softball_Stats2014, R ~ myOBA))

Coefficients:
(Intercept)        myOBA
```

```
       -374         1732
```

```
rsquared(lmmwSoftballOBA2014R)
```

```
[1] 0.8225
```

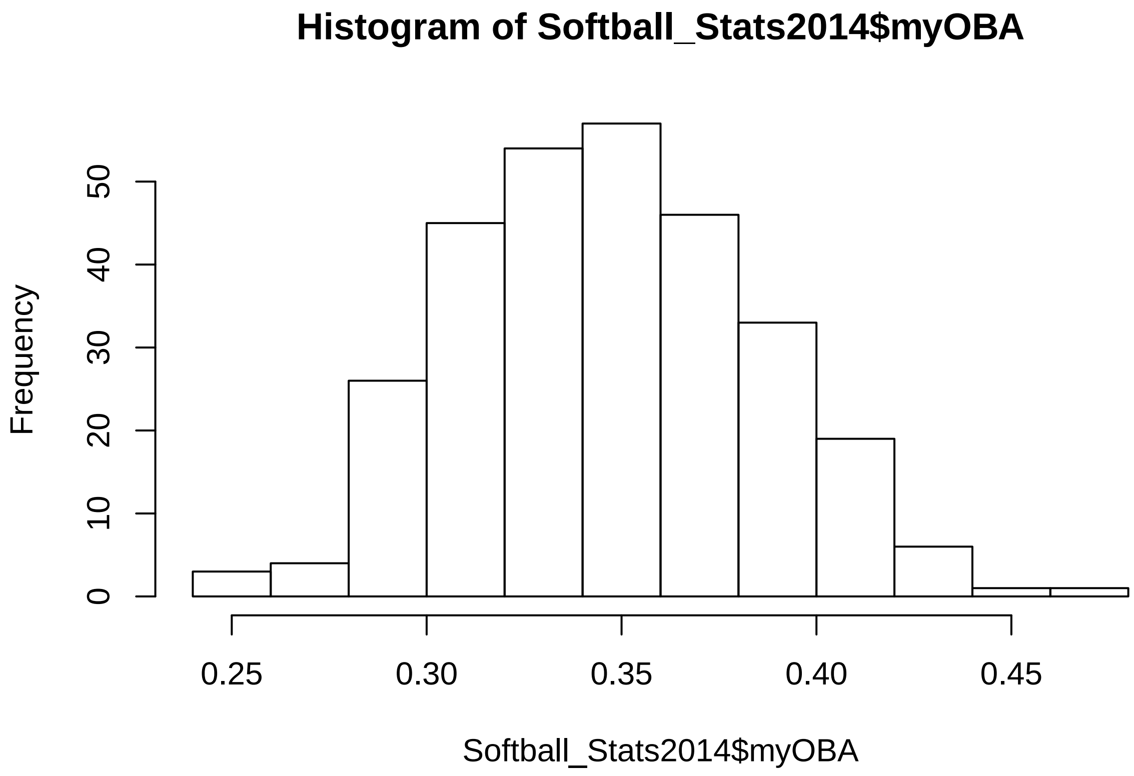
```
anova(lmmwSoftballOBA2014, lmwSoftballOBA2014)
```

```
Analysis of Variance Table

Model 1: PCT ~ myOBA
Model 2: PCT ~ wOBA
  Res.Df  RSS Df Sum of Sq F Pr(>F)
1    293 7.77
2    293 7.74  0    0.0267
```

```
Softball_Stats2014$fit <- predict(lmmwSoftballOBA2014)
Softball_Stats2014$ul <- Softball_Stats2014$fit + 1.96 * predict(lmmwSoftballOBA2014, se.fit = T)$se
Softball_Stats2014$ll <- Softball_Stats2014$fit - 1.96 * predict(lmmwSoftballOBA2014, se.fit = T)$se







Softball_Stats2014$fit2 <- predict(lmmwSoftballOBA2014R)
Softball_Stats2014$ul2 <- Softball_Stats2014$fit2 + 1.96 * predict(lmmwSoftballOBA2014R, se.fit = T)$se
Softball_Stats2014$ll2 <- Softball_Stats2014$fit2 - 1.96 * predict(lmmwSoftballOBA2014R, se.fit = T)$se


rsquared(lmmwSoftballOBA2014)
```

```
[1] 0.06174
```

```
# Looking at the distribution of Softball wOBAs, looks normal
hist(Softball_Stats2014$myOBA)
```



57

```
# Creating MLB datasets
baseoutstates <- (c("Nobody on", "Runner on 1st", "Runner on 2nd", "Runner on 3rd", "Runners on 1st and
    "Runners on 1st and 3rd", "Runners on 2nd and 3rd", "Bases Loaded"))
NoOuts <- c(0.555, 0.953, 1.189, 1.482, 1.573, 1.904, 2.052, 2.417)
OneOut <- c(0.297, 0.573, 0.725, 0.983, 0.971, 1.243, 1.467, 1.65)
TwoOuts <- c(0.117, 0.251, 0.344, 0.387, 0.466, 0.538, 0.634, 0.815)
MLBBOStates <- data.frame(baseoutstates, NoOuts, OneOut, TwoOuts)

playTypeMLB <- c("home run", "triple", "double", "single", "walk", "hit by pitch", "out", "strikeout",
    "catcher's interference", "error", "illegal pitch", "wild pitch", "passed ball", "stolen base", "SA
    "bunt")
RTOEMLB <- c(1.942, 1.616, 1.311, 1.025, 0.849, 0.969, 0.24, 0.207, 1.083, 1.132, 0.949, 1, 1.026, 0.79
    1.031, 0.481)
run_expMLB <- c(0.533, 0.553, 0.547, 0.551, 0.52, 0.584, 0.538, 0.517, 0.655, 0.586, 0.712, 0.716, 0.74
    0.597, 1.058, 0.409)
playvalueMLB <- data.frame(playTypeMLB, RTOEMLB, run_expMLB)
```

Question

"The Book: Playing the Percentages in baseball"_{1} starts with a handy first chapter aptly titled "Toolshed," in which it discusses fundamental tenants of statistical understanding necessary to evaluate trends and values in baseball. For softball, a sport played by women on a different field with different equipment, the accepted theory — as far as I have surmised on my admittedly-less-than-extensive search of the internet for advanced softball statistics — is that the sport follows a pattern similar to baseball.

This affects our understanding of softball. Statistics like wOBA (weighted on-base average), generated from MLB play data, have been used to explain the value of plays as if the sports were similar. This is the wOBA equation, with coefficients derived from the play's — i.e single, double, etc. — relative run value compared to the out.

{MLB's wOBA} ({.90 * Singles} + {1.24* Doubles} + {1.56* Triples} + {1.95* Home runs} + {.75* Hit-By-Pitches} + {.72* Walks} +{.90* Reached Base on Error}) / {Plate Appearances}

In fact, a post_{2} on fastpitchanalytics.com closes with the author's desire to evaluate softball with different coefficients for the wOBA statistic specifically for softball but used MLB's statistic to evaluate the top hitting teams in the country.

Using basic statistics provided by the NCAA website for all teams in 2016, we can see how well wOBA fits teams' ability to score runs — one of the most effective metrics in evaluating a team's offense and its ability to win games.

- **H** hits
- **oneB** singles^{1}
- **RBOE** Reached Base on Error^{1}
- **twoB** doubles
- **threeB** triples
- **HR** home runs
- **BB** walks
- **HBP** hit-by-pitch
- **AB** at-bats
- **IBB** intentional walks
- **SH** sacrifice hits
- **SF** sacrifice flies

$$Runs 1 + Team wOBA$$

{Winning %} 1 + {Team wOBA}

We first create and look at two functions, one modelling the wOBA and runs and the other modelling wOBA and a team's winning percentage. wOBA is the independent variable, and runs and winning percentage are both dependent variables, respectively.

```
lmwSoftballOBA2016 <- lm(with(Softball_Stats2016, PCT ~ wOBA))
lmwSoftballOBA2016
```

```

Call:
lm(formula = with(Softball_Stats2016, PCT ~ wOBA))

Coefficients:
(Intercept)         wOBA
      -0.37         2.36
```

```
rsquared(lmwSoftballOBA2016)
```

```
[1] 0.2667
```

```
lmwSoftballOBA2016R <- lm(with(Softball_Stats2016, R ~ wOBA))
lmwSoftballOBA2016R
```

```
Call:
lm(formula = with(Softball_Stats2016, R ~ wOBA))

Coefficients:
(Intercept)         wOBA
       -443         1875
```

```
rsquared(lmwSoftballOBA2016R)
```

```
[1] 0.8303
```

```
lm1Softball <- lm(with(Softball_Stats2016, R ~ oneB + twoB + threeB + HR + BB + RBOE + HBP))
lm1Softball
```

```

Call:
lm(formula = with(Softball_Stats2016, R ~ oneB + twoB + threeB + 
    HR + BB + RBOE + HBP))

Coefficients:
(Intercept)         oneB         twoB       threeB           HR           BB         RBOE  
    -99.252        0.419        0.679        1.728        1.621        0.417        0.739  
        HBP  
      0.259  
```

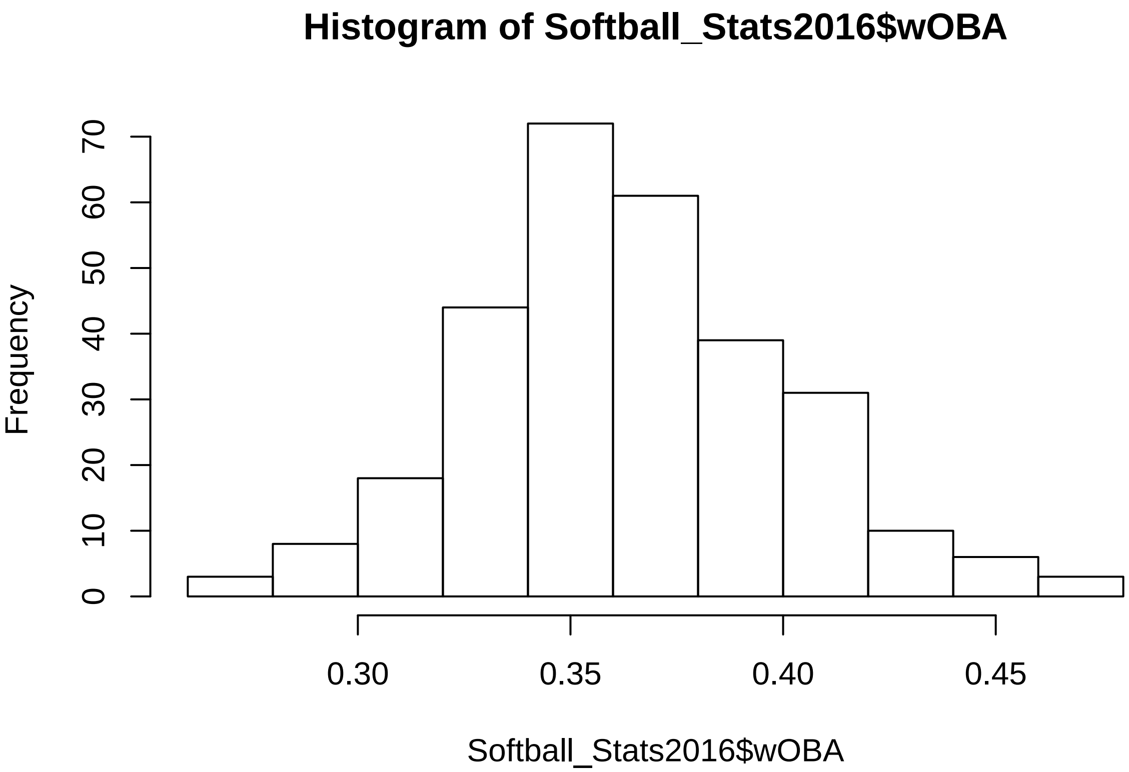
```
rsquared(lm1Softball)
```

```
[1] 0.9396
```

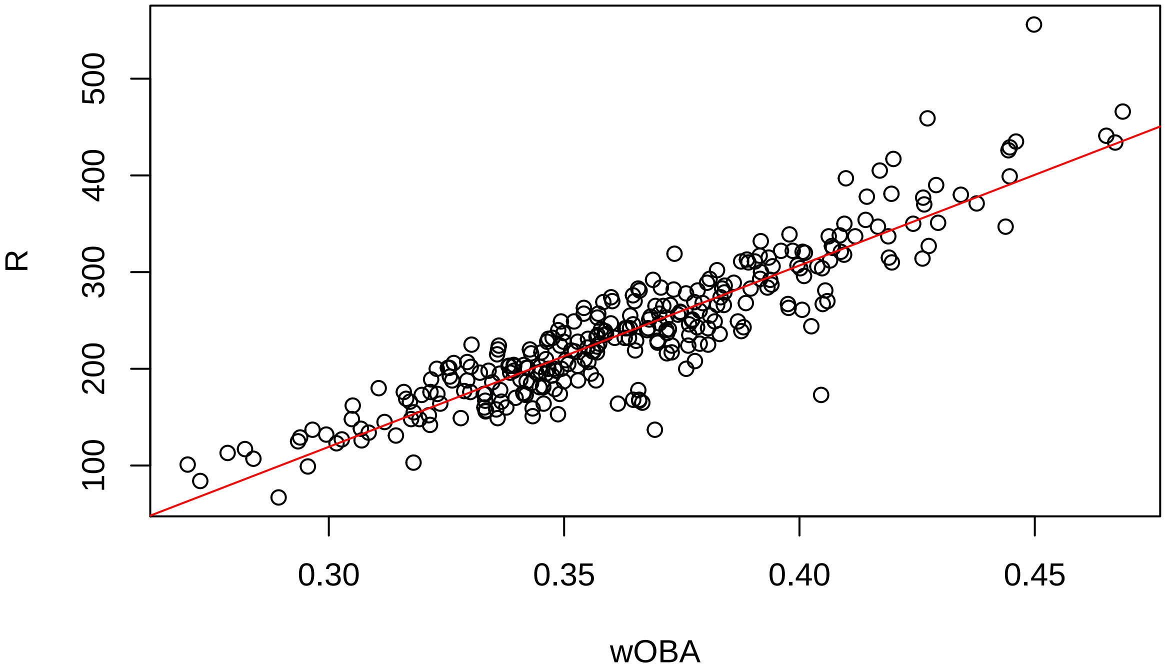
There is a much stronger positive correlation with runs and wOBA, with 83% of the variance in runs explained by the team's wOBA. The distribution of wOBA is normal, as seen below, and there are few outliers — though some on the extremes, as is well-modelled by the confidence intervals of the graphed functions.

I also included an optimization of all the coefficients in wOBA, which may show us a clue of the possible trends we may see in what's to come. Of course, this operates with no premise of how valuable each individual play; it merely shows association between being good at one statistic — for example, triples — can translate into better runs totals. What wOBA also takes into account is the weight of each type of play in respect to its run creation value. So, if a team hits more triples, that's still good, but the coefficient reflects an average amount of runs that are created — which is less than home runs, which obviously makes sense.

```
hist(Softball_Stats2016$wOBA)
```

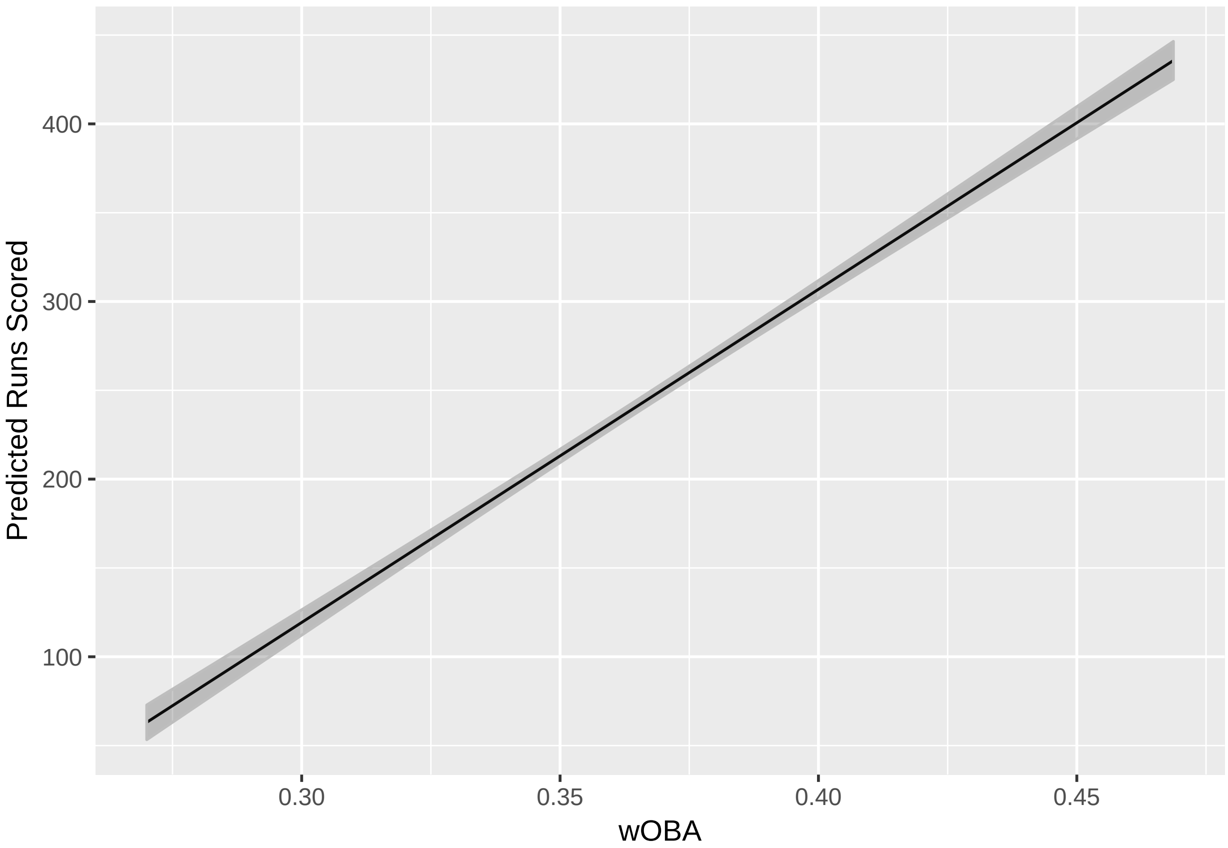


```
# Plot the function with the dataset
plot(x = Softball_Stats2016$wOBA, y = Softball_Stats2016$R, xlab = "wOBA", ylab = "R")
abline(lmwSoftballOBA2016R, col = "red")
```



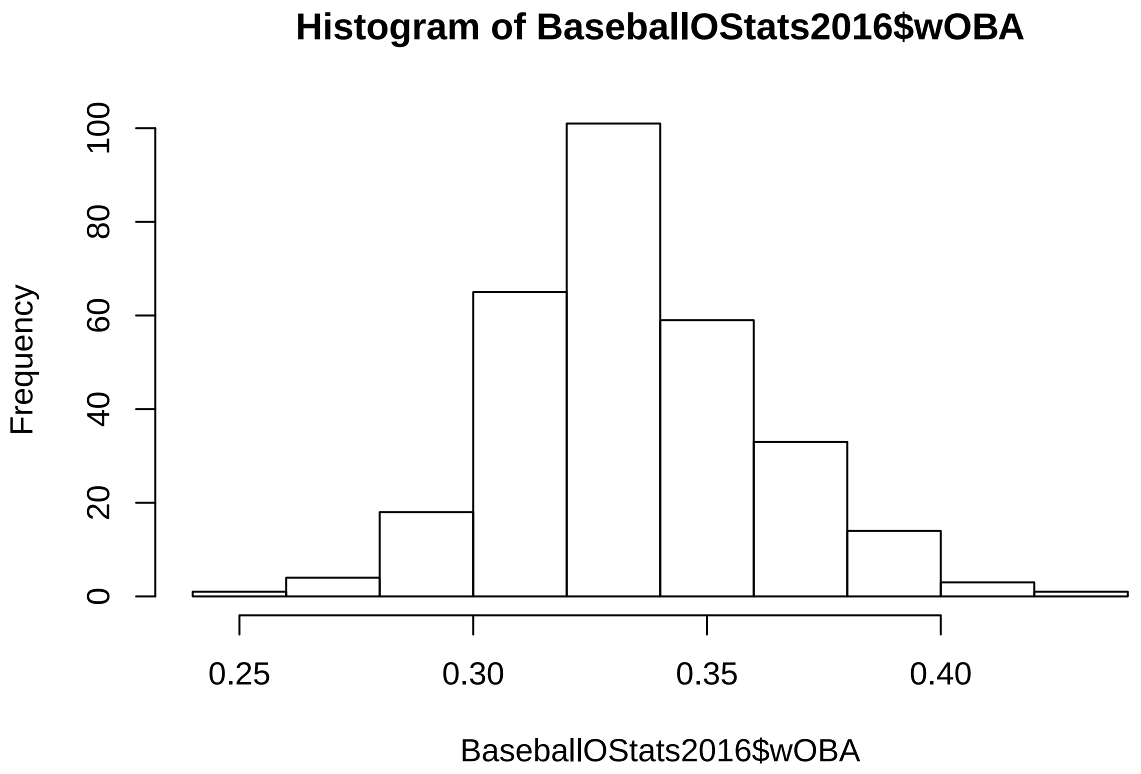
```
Softball_Stats2016$fitwOBA <- predict(lmwSoftballOBA2016R)
Softball_Stats2016$fit2 <- predict(lmwSoftballOBA2016R)
Softball_Stats2016$ul2 <- Softball_Stats2016$fit2 + 1.96 * predict(lmwSoftballOBA2016R, se.fit = T)$se
Softball_Stats2016$ll2 <- Softball_Stats2016$fit2 - 1.96 * predict(lmwSoftballOBA2016R, se.fit = T)$se
```

```
plot <- ggplot(data = Softball_Stats2016, aes(x = wOBA, y = fit2)) + geom_line() + geom_ribbon(aes(ymin
    ymax = ul2), col = "grey", alpha = 0.25) + labs(x = "wOBA", y = "Predicted Runs Scored")
plot
```



Though not perfect comparisons, it is illumnating to look at how wOBA models baseball statistics. Unfortunately, softball's records are better kept and including reached base on error is not possible without knowing certain variables. The distribution of wOBA's for baseball is similarly normal — though shifted left — and, again, the r squared value is significantly better for runs than winning percentage.

```
hist(BaseballOStats2016$wOBA)
```



```
lmwBaseballOBA2016R <- lm(with(BaseballOStats2016, R ~ wOBA))
lmwBaseballOBA2016R
```

```

Call:
lm(formula = with(BaseballOStats2016, R ~ wOBA))

Coefficients:
(Intercept)         wOBA
       -328         1581
```

```
rsquared(lmwBaseballOBA2016R)
```

```
[1] 0.7839
```

```
lmwBaseballOBA2016 <- lm(with(BaseballOStats2016, PCT ~ wOBA))
lmwBaseballOBA2016
```

```

Call:
lm(formula = with(BaseballOStats2016, PCT ~ wOBA))

Coefficients:
(Intercept)         wOBA
     -0.585        3.239
```

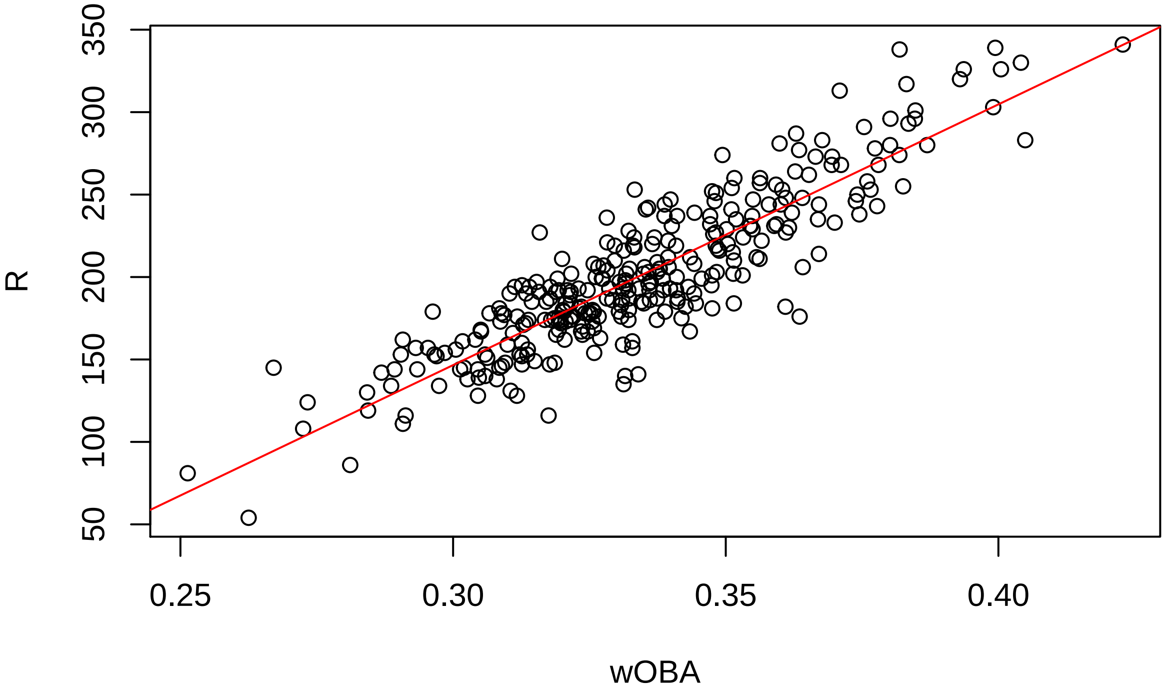
```
rsquared(lmwBaseballOBA2016)
```

```
[1] 0.3195
```

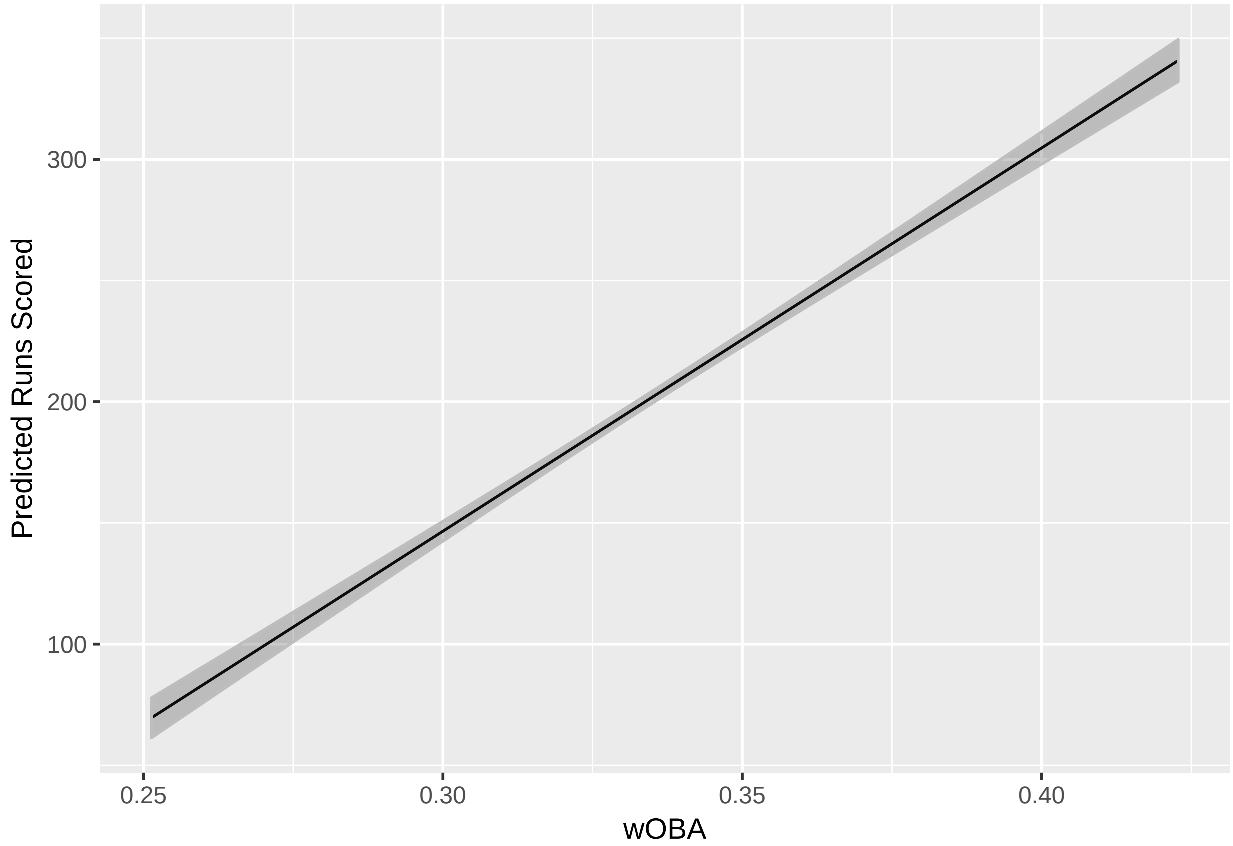
Two notes here: One, softball's r squared function for runs is much higher, which might lend one to think that wOBA works for softball more than baseball. However, we see that PCT is much more correlated for baseball than softball; if anything, this seems to be an indictment of anyone who thinks errors should not be included in datakeeping, as it adds an element of explanation to run totals.

*I repeated this process for 2015 and 2014 in the code appendix. The results are similar, though there might be a slight error in the data tallying of win percentages. Either way, the important thing to focus on is the statistics relationship to runs.^{1}

```
plot(x = BaseballOStats2016$wOBA, y = BaseballOStats2016$R, xlab = "wOBA", ylab = "R")
abline(lmwBaseballOBA2016R, col = "red")
```

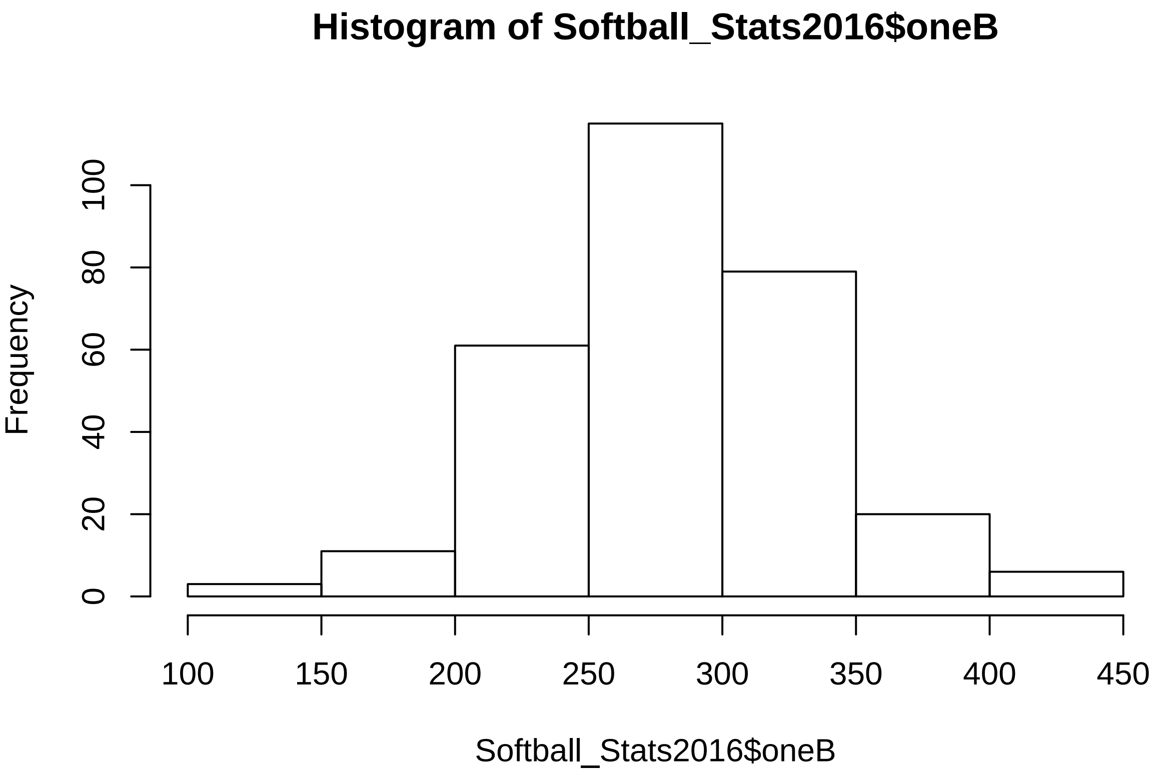


```
BaseballOStats2016$fitwOBA <- predict(lmwBaseballOBA2016R)
BaseballOStats2016$fit2 <- predict(lmwBaseballOBA2016R)
BaseballOStats2016$ul2 <- BaseballOStats2016$fit2 + 1.96 * predict(lmwBaseballOBA2016R, se.fit = T)$se
BaseballOStats2016$ll2 <- BaseballOStats2016$fit2 - 1.96 * predict(lmwBaseballOBA2016R, se.fit = T)$se

plot2 <- ggplot(data = BaseballOStats2016, aes(x = wOBA, y = fit2)) + geom_line() + geom_ribbon(aes(ymi
    ymax = ul2), col = "grey", alpha = 0.25) + labs(x = "wOBA", y = "Predicted Runs Scored")
plot2
```

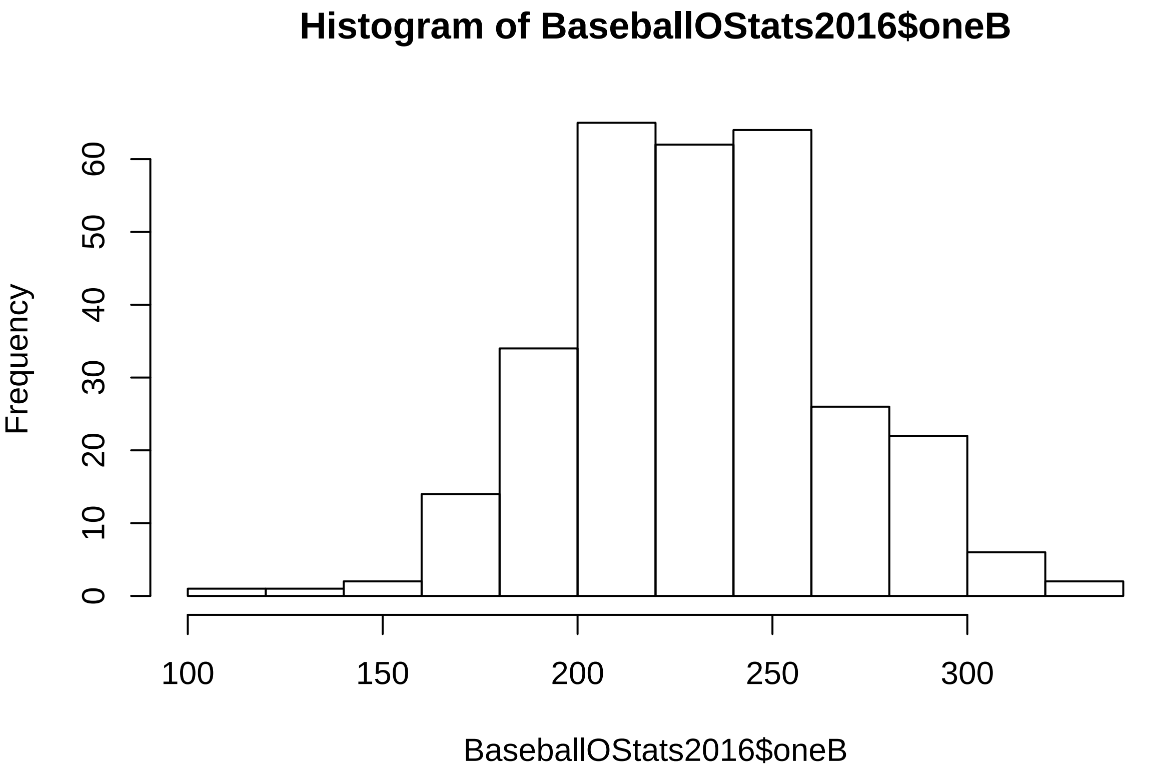


There are a lot of indicators that point to potential differences, including the average of singles hit by each team for each sport. Softball's total numbers are a little inflated because it comes from a larger gameset, so this exercise shows that there is a statistically significant difference in the proportion of singles among the two sports. Both distributions appear fairly normal.

```r
hist(Softball_Stats2016$oneB)
```



```r
hist(BaseballOStats2016$oneB)
```



```r
testA <- Softball_Stats2016$oneB/Softball_Stats2016$AB
testB <- BaseballOStats2016$oneB/BaseballOStats2016$AB
mean(testA)
```

```
[1] 0.2029
```

```
mean(testB)
```

```
[1] 0.1937
```

```
sd(testA)
```

```
[1] 0.02194
```

```
sd(testB)
```

```
[1] 0.01859
```

```
mean(testA) - mean(testB)/sqrt(((sd(testA)^2)/295) + ((sd(testB)^2)/299))
```

```
[1] -115.8
```

These evaluations are a precursor to my question. In order to statistically understand the sport of softball, we must compare the foundations of sabermetrics and advanced statistics between softball and baseball. wOBA is one of the individual statistics highlighted in that first chapter of "The Book" because the statistics combines the positives of on-base percentage and slugging percentage and weights it such that it is equitable to OBP. In order to apply a statistic such as wOBA — the constants of which are derived from play-type run values — we need to make sure that the foundations behind those constants and its application fit the sport to the best of its ability. Comparing the function across both softball and baseball statistics illustrates just this fact. The differences in the data alone is enough to signal that there might be ways to differentiate the two sports.

The underlying principle here is that statistics is fundamental to our understanding of how to evaluate players and teams and in-game decision-making. If there's a person on first and no outs, should a team sacrifice the hitter to move the runner to second? What player is more valuable to a team, the power hitter with low speed or the speedster that can turn doubles into triples and steal bases? When evaluating a team's offense, how can we weigh the importance of singular events?

The statistic at the core of this understanding is run expectancy per base/out state. There are 24 run/out states, comprised of the different combinations of baserunners possible at any given time — such as runner on first, bases loaded, runners on the corners, etc. — and the different possible amount of outs in the inning — 0, 1, 2. These values are critical in determining whether the same situation in softball at all corresponds to the same situation in baseball. For example, these values were able to highlight the inefficiency of a sacrifice bunt in baseball, as run expectancy falls from pre-sacrifice bunt to after it.

Once I can evaluate how many runs are expected to arise from each individual situation, I can dive further into the run values from individual events and various changes in run expectancies across both base-states and events.

Data

The team statistical data was scraped from the NCAA's official statistics website_{3}, which provides almost all of the simplistic stats that necessitate little to no calculation or derivation.

The data for the play-by-play is by far the most important part of this entire project. I got the play-by-play data from user 'octonian' on github_{4}, who had a plethora of softball data and code stored in his account on the website. However, the raw data he provided was relatively unreadable, as the playIDs were the only indicators of what resulted from the specific play.

The raw data needs three things in order for it to be appropriately utilizable for the project — a measure of outs, a measure of runs and a measure of the people on base in a given situation. These are the three variables that can give us run expectancy from each of the identified states as long as I can properly attribute every play with its result.

New variables

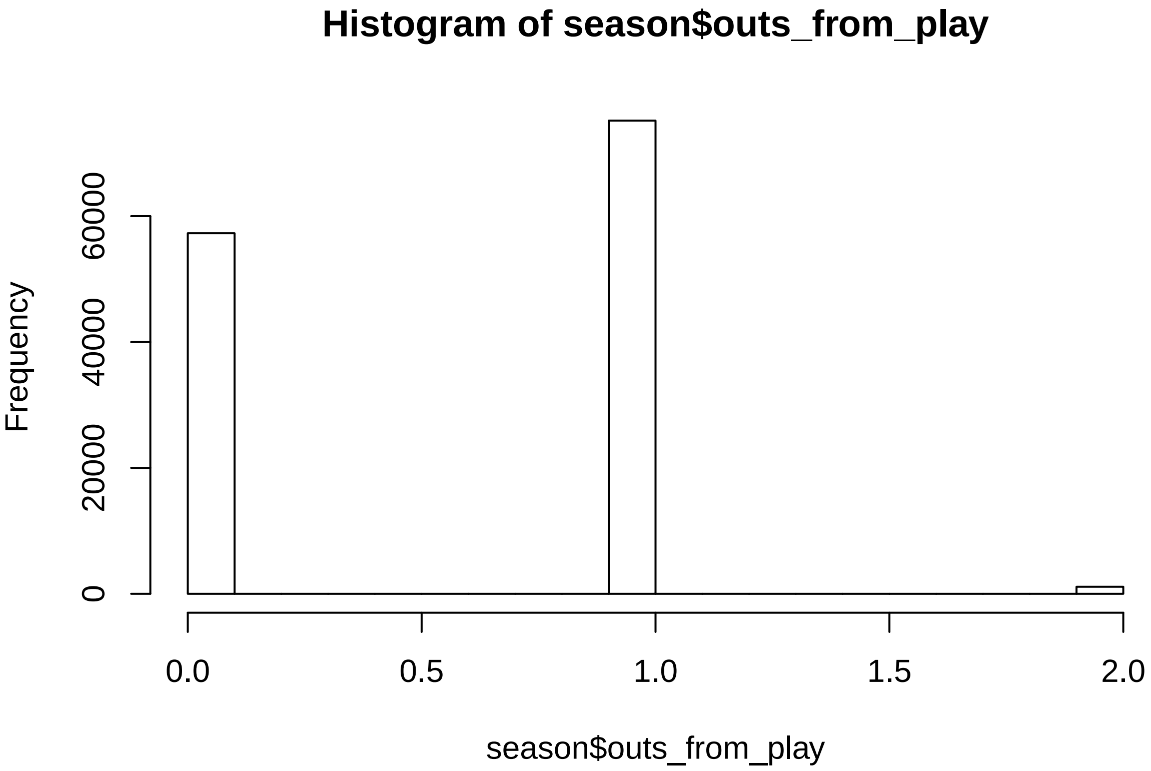
- **gameID** every individual game is given a numeric code labelled as 'game ID'
- **playID** & **play** the description of a play, which includes the event and the movement of baserunners
- **inningID** every inning in every game is given a numeric code labbled as 'inning ID'

Outs

The first step is to create numeric indicators for outs^{3}. There were a few challenges in doing so, as batters reached base despite "striking out". This was also a good time to evaulate data by surveying random games, and I noticed that beyond the inning summaries that were already removed, there were a number of plays — such as substitions and "No play"s — that needed to be removed from the dataset.

Outs_from_play is a category that takes the indicators of outs in the playID and assigns a value of 0 (if there is no out on the play), 1 (if there is an out) and 2 (if there is a double play). There is no value for a triple play because there was no occurence of a triple play in the data.

```
hist(season$outs_from_play)
```



One game

This gives us a metric for the outs created from the play, but — when we are evaluating the base/out states — we need to know how many outs are there while a play is occuring. This requires us to apply the column that we just created across the entire dataset, looping our count of the outs such that we can dilineate between the top and bottom of every inning and how many outs have happened before a certain play. This allows to keep a running tally of the outs in an inning, the outs that have occured before a given play and the outs that occured as a result of a given play.

The next step is to then create those columns and apply it to one game — as compared to a datset with over 2000 games.

Looking at the data, we can see that we have successfully translated^{4} the outs from one play to the next and have the appropriate accumulations for each column. Also, we can now differentiate between the top and bottom of innings based on the amount of outs, which will be important moving forward as we track the runs scored and separate baserunners on either side of an inning.

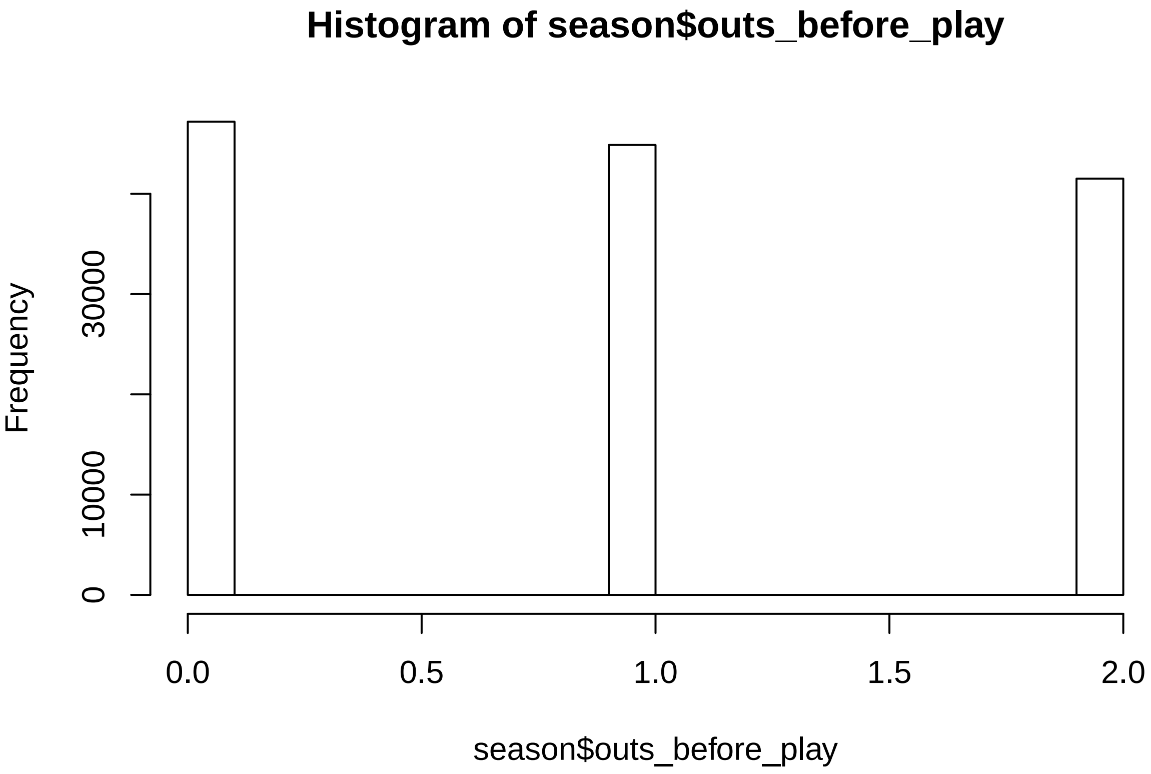
Entire Season

Now, we must repeat the process for the entire season, reapplying the code^{5} that worked for the dataset "game" to the dataset "season". There are not many ways to completely check the dataset for potential errors other than line-by-line checking, but there are some commands toward the end of the code that can functionally check that the code was properly applied.

```
summary(season$outs_before_play)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   0.000   1.000   0.958   2.000   2.000 
```

```
hist(season$outs_before_play)
```



After eliminating the games in which the outs were possibly missapplied indicated by an instance of three outs in an ongoing inning, there is still a wealth of data in the dataset of which to analyze.

New variables

- **total_outs_in_inning** the amount of outs in the inning after the play has occured
- **outs_before_play** the amount of outs in the inning before the play has occured
- **outs_from_play** the amount of outs that occur as a result of the play
- **top_bottom** an all-important variable that differentiates the top and the bottom of an individual inning based on three outs occuring (all plays before out 3 = top, all plays after out 3 = bottom, play in which out 3 occured = top)

Runs

The next step in shaping our data is to create a similarly coded metric for runs that can be identified from the playID, allocate the proper amount of runs to the play, count the runs in an individual inning and establish the amount of runs that have occured before a play.

First, we start with creating a measure of runs similar to the outs beforehand. Instead of searching for words like out, we look for “RBI” and “scored”. There were a few different approaches here: One, there needed to be the proper appropriation of runs according to the amount of RBI listed in the playID. Secondly, since it’s difficult to parse between generic “RBI” and “2 RBI,” I wrote the code such that the character preceding the space before RBI needed to be alphabetic — or non-numeric. That allowed me to create a measure for 1 RBI. Also, not all runs are scored via an RBI — some are from wild pitches, passed balls or random fielding mistakes. I assigned those events one run for the “scored” character string in playID despite the fact that there may have been more than one run scored on the play. I thought about appropriating the distribution of amount of runs scored in plays in which a run was scored, but it’s even more increasingly less likely that multiple runs will scored in these instances. Therefore, I think that any possible error in that assignment of runs will be insignificant when looking at the final results.^{6}

```
length(season$runs_from_play != 0)
```

```
[1] 133591
```

```
table(season$runs_from_play)
```

```

     0      1      2      3      4 
118801  11559   2520    577    134 
```

Creating a measure of runs for one game

We now have created a metric in which all of the runs resulting from one play are assigned to the number of runs created. Moving forward, we need to borrow the pattern for accumulating outs and apply it for accumulating runs — making sure that we are basing runs per inning on the number of outs such that the columns correctly accumulate. Next, code for one game and check to make sure everything works properly. ^{7}

```
pandoc.table(season[season$gameID == 4032168 & season$inningID == 0, c("play", "top_bottom", "runs_befo
```

```

--------------------------------------------------------------
             play               top_bottom   runs_before_play 
------------------------------ ------------ -------------------
S. Venegas grounded out to 3b.     top              0         

 L. Hanna singled to pitcher.      top              0         

    J. Leung reached on a          top              0         
fielder's choice; L. Hanna out                                
     at second ss to 2b.                                      

   C. Seitz homered to left        top              0         
center 2 RBI; J. Leung scored.                                
          2 0 36557                                           

    Y. Park hit by pitch.          top              2         
 Schoenewald tripled to right      top             2
center RBI; Y. Park scored. 3
           0 36586

A. Lotz singled up the middle      top             3
 RBI; Schoenewald scored. 4 0
            36617

 A. McGregor popped up to 2b.      top             4

    Clark popped up to 2b.        bottom           0

   Nino grounded out to 3b.       bottom           0

 Sensi singled up the middle.     bottom           0

    Berg popped up to 1b.         bottom           0
--------------------------------------------------------------
```

```r
pandoc.table(season[season$gameID == 4032168 & season$inningID == 1, c("play", "top_bottom", "runs_befo
```

```
--------------------------------------------------------------
             play               top_bottom   runs_before_play
------------------------------ ------------ ------------------
 Springfield singled to left       top             0
            field.

 S. Venegas popped up to ss.       top             0

 L. Hanna singled to pitcher;      top             0
   Springfield advanced to
           second.

    J. Leung reached on a          top             0
  fielder's choice; L. Hanna
     advanced to second;
Springfield out at third ss to
             3b.

  C. Seitz singled to center       top             0
 field RBI; J. Leung advanced
to second; L. Hanna scored. 5
           0 36647

  Y. Park homered to center        top             1
field 3 RBI; C. Seitz scored;
  J. Leung scored. 8 0 36739

 Schoenewald grounded out to       top             4
             2b.

  Cottrell singled to center      bottom           0
            field.
```

```
Valdes doubled to left center;    bottom            0
 Cottrell advanced to third.

       Crimarco walked.           bottom            0

    Judd popped up to ss.         bottom            0

   Spencer popped up to ss.       bottom            0

    Clark popped up to 1b.        bottom            0
--------------------------------------------------------------
```

```
pandoc.table(season[season$gameID == 4032168 & season$inningID == 2, c("play", "top_bottom", "runs_befo
```

```
--------------------------------------------------------------
             play               top_bottom   runs_before_play
------------------------------ ------------ ------------------
       A. Lotz singled.            top              0

 A. McGregor doubled to left       top              0
 center; A. Lotz advanced to
            third.

     Springfield walked.           top              0

    S. Venegas walked RBI;         top              0
   Springfield advanced to
 second; A. McGregor advanced
to third; A. Lotz scored. 9 0
            36770

  Scarangella walked RBI; S.       top              1
 Venegas advanced to second;
Springfield advanced to third;
A. McGregor scored. 10 0 36800

 J. Leung singled advanced to      top              2
  second on the throw 3 RBI;
Scarangella scored; S. Venegas
scored; Springfield scored. 13
            0 13-0

  C. Seitz singled; J. Leung       top              5
      advanced to third.

   G. Metzger reached on a         top              5
  fielder's choice; C. Seitz
     advanced to second.

 Schoenewald singled RBI; G.       top              5
Metzger advanced to second; C.
Seitz advanced to third out at
home lf to c; J. Leung scored.
          14 0 14-0
```

```
A. Lotz singled to left field;     top              6
   Schoenewald advanced to
second; G. Metzger advanced to
            third.

A. McGregor doubled 2 RBI; A.      top              6
   Lotz advanced to third;
Schoenewald scored; G. Metzger
      scored. 16 0 16-0

 Springfield flied out to lf.      top              8

   S. Venegas reached on a         top              8
fielder's choice; A. McGregor
   out at second ss to 2b.

   Nino grounded out to p.        bottom            0

Sensi doubled to right field.     bottom            0

   Berg grounded out to ss.       bottom            0

 16 2 16-2 Cottrell homered 2     bottom            0
      RBI; Sensi scored.

        Valdes walked.            bottom            2

 Crimarco struck out looking.     bottom            2
--------------------------------------------------------------
```

```r
pandoc.table(season[season$gameID == 4032168 & season$inningID == 3, c("play", "top_bottom", "runs_befo
```

```
--------------------------------------------------------------
             play               top_bottom   runs_before_play
------------------------------ ------------ ------------------
 L. Hanna grounded out to p.       top              0

J. Pinkney grounded out to 3b.     top              0

  C. Seitz flied out to ss.        top              0

 Oruska struck out swinging.      bottom            0

Finnerty struck out swinging.     bottom            0

Clark singled through the left    bottom            0
            side.

 Nino reached on an error by      bottom            0
2b; Clark advanced to second.

Sensi walked; Nino advanced to    bottom            0
  second; Clark advanced to
```

```
            third.

   Berg grounded out to p.        bottom            0
---------------------------------------------------------------
```

```r
pandoc.table(season[season$gameID == 4032168 & season$inningID == 4, c("play", "top_bottom", "runs_befo
```

```
---------------------------------------------------------------
             play               top_bottom   runs_before_play
------------------------------ ------------ ------------------
G. Metzger singled down the lf     top              0
            line.

   S. Syrop singled to left        top              0
field; G. Metzger advanced to
           second.

  A. Lotz singled; S. Syrop        top              0
advanced to second; G. Metzger
      advanced to third.

 A. McGregor singled through       top              0
  the left side RBI; A. Lotz
 advanced to second; S. Syrop
advanced to third; G. Metzger
      scored. 17 2 17-2

A. McGregor advanced to second     top              1
   on a wild pitch; A. Lotz
 advanced to third on a wild
 pitch; S. Syrop scored on a
    wild pitch. 18 2 18-2

    Springfield struck out         top              2
          swinging.

 S. Venegas singled to right       top              2
    field RBI; A. McGregor
  advanced to third; A. Lotz
      scored. 19 2 19-2

   L. Hanna singled RBI; S.        top              3
Venegas advanced to second; A.
  McGregor scored. 20 2 20-2

  J. Leung lined out to cf.        top              4

  C. Seitz popped up to 1b.        top              4

  Cottrell fouled out to lf.     bottom             0

  Swank struck out swinging.     bottom             0

 Crimarco grounded out to 3b.    bottom             0
```

Entire season

Now, just like we did for outs, we apply the same code that we used on one game to the entire season. The distribution of total_runs_in_inning is heavily skewed to one, which makes sense if we consider the distribution of the amount of RBIs per play. The distribution of runs_before_play, which measures the amount of runs before play on either half of an inning tells a good story of the amount of runs scored per inning.^{8}

```
table(season$total_runs_in_inning)
```

```

    0     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
81969 22535 12795  7194  3917  2307  1380   631   390   219    89    77    37    23     8    12
   16    17
    6     2
```

```
table(season$runs_before_play)
```

```

     0      1      2      3      4      5      6      7      8      9     10     11     12     13
105814  13672   6921   3481   1697    939    557    235    128     69     30     24      8      5
    14     15
     4      7
```

A potential source of error could be that we do not account for the runs on a walk-off hit, because total_runs_in_inning is not separated by side and runs_before_play resets after a walk-off hit. If anything, this would slightly depreciate hit values and high order base-out states, but there aren't many games that are walked-off and the amount of base-out states and events make any micro changes fairly insignifcant, which I will discuss later.

New variables

- **total__runs__in__inning** the amount of runs in the inning after the play has occured
- **runs__before__play** the amount of runs in the inning before the play has occured
- **runs__from__play** the amount of runs that occur as a result of the play

Base-Out States

Base-Out states are a component that describe who is on base before and after a certain event, which is fundamental to our understanding of both plays and changes in the game based on what occurs.

Indicators

In order to start extrapolating the base-out states from the data, the first step was to make indicators for the baserunners that moved along the bases. Advanced to second, advanced to third are two of the major ones; hitting a double such that a runner is on second and no one is on first; clearing the bases with a home run, etc. Through trial and error, I coded for five new variables:^{9}

- **firstbase** an indicator describing whether a runner is on first (1 if yes, 0 if no)
- **secondbase** an indicator describing whether a runner is on second (1 if yes, 0 if no)
- **thirdbase** an indicator describing whether a runner is on third (1 if yes, 0 if no)

- **temp** an indicator describing whether the play falls under a special, enumerated condition that requires different coding later (1 if the batter is on first, 2 if a runner stole second, 3 if a runner stole third, 4 if the batter is going to first and a runner is advancing to second, 5 if a batter advances past first on an error or throw, 6 if a runner is caught stealing)
- **base_state_after** an indicator describing who is on base and how many outs there are (Nobody on, No Outs; Nobody on, One out; Nobody on, Two outs; Runner on 1st, No Outs; Runner on 1st, One out; Runner on 1st, Two outs; Runner on 2nd, No Outs; Runner on 2nd, One out; Runner on 2nd, Two outs; Runner on 3rd, No Outs; Runner on 3rd, One out; Runner on 3rd, Two outs; Runners on 1st and 2nd, No Outs; Runners on 1st and 2nd, One out; Runners on 1st and 2nd, Two outs; Runners on 1st and 3rd, No Outs; Runners on 1st and 3rd, One out; Runners on 1st and 3rd, Two outs; Runners on 2nd and 3rd, No outs; Runners on 2nd and 3rd, One out; Runners on 2nd and 3rd, Two outs; Bases Loaded, No outs; Bases Loaded, One out; Bases Loaded, Two outs)

At this point, there's nothing remarkable about the data. No distribution of any of these codes do the final product justice.

Entire Season

Coding for the entire season's base-out states require a process somewhat similar to the code for outs and runs before, in which the states loop about the top and the bottom of innings. However, in this case, we need to make sure that

-**base_state_before** an indicator describing the base-state before the event with the same indicators as base_state_after

reads "Nobody on, No Outs" at the start of every inning.^{10}

Also, we need to make sure that if an out occurs on a given play, the base-state stays the same unless otherwise stated, as in the case of a sacrifice fly. For every out, the next base_state_before retained the baserunners but changed the number of outs. That takes care of the changes in state after an out, which make up a significant portion of the events in a game.

However, the trickiest part — and the way to replace the rest of the "placeholders" — is to code for the previously enumerated special conditions and the resulting changes in state. These are varied based on the condition, but basically the code needed to understand that runners could stay on a base while other runners advanced.

I repeated both of these codes four times because each time the code is replicated, the base-state changes theoretically until there are fewer and fewer placeholders. After doing it four times, I felt that the base_state_before column had significant tallies to move forward. The resulting placeholders represent gaps in this coding; however, there are relatively few compared to the sheer amount of data in the dataset.^{11,11.2,11.3,11.4}

```r
table(season$base_state_before)
```

```
           Bases Loaded, No outs            Bases Loaded, One out           Bases Loaded, Two outs
                            1165                             2502                             2685
              Nobody on, No Outs               Nobody on, One out              Nobody on, Two outs
                           27563                            16705                            10864
                     placeholder           Runner on 1st, No outs           Runner on 1st, One out
                            8314                             8525                             8446
         Runner on 1st, Two outs           Runner on 2nd, No outs           Runner on 2nd, One out
                            7418                             3064                             5503
         Runner on 2nd, Two outs           Runner on 3rd, No outs           Runner on 3rd, One out
                            5725                              561                             1872
         Runner on 3rd, Two outs  Runners on 1st and 2nd, No outs  Runners on 1st and 2nd, One out
                            2831                             2369                             3576
```

```
Runners on 1st and 2nd, Two outs Runners on 1st and 3rd, No outs Runners on 1st and 3rd, One out
                            3616                              731                            1570
Runners on 1st and 3rd, Two outs Runners on 2nd and 3rd, No outs Runners on 2nd and 3rd, One out
                            1957                              935                            2378
Runners on 2nd and 3rd, Two outs
                            2716
```

Play Types

One important part of the data that is relatively easy to code is the playTypes — which correspond to what happened during each at-bat. Because there are a significant amount of possibilities that could have occured, we can just create a column that extrapolates the result of the at-bat and save that for later.^{37} For convenience, I have coded for some of the major plays and tabled them, which gives a value of how many runs on average are created by the event (during the course of the play).

```r
table(season$playType)
```

```
                 bunt catcher's interference                 double           hit by pitch
                 1659                     35                   4871                   2513
             home run          illegal pitch                    out            passed ball
                 2532                    199                  57334                    744
     reached on error                 single            stolen base              strikeout
                 1337                  19938                   3899                  17958
               triple                   walk             wild pitch
                  687                  11773                   2018
```

```r
table(season$playType, season$runs_from_play)
```

```
                             0     1     2     3     4
  bunt                    1488   169     2     0     0
  catcher's interference    31     4     0     0     0
  double                  2657  1484   628   102     0
  hit by pitch            2358   155     0     0     0
  home run                   0  1116   845   437   134
  illegal pitch            135    64     0     0     0
  out                    55317  1944    70     3     0
  passed ball              599   145     0     0     0
  reached on error        1040   289     8     0     0
  single                 15009  4079   842     8     0
  stolen base             3760   139     0     0     0
  strikeout              17930    28     0     0     0
  triple                   305   240   115    27     0
  walk                   11175   598     0     0     0
  wild pitch              1663   355     0     0     0
```

*This table actually helped me detect an error for solo home runs that made me go back and figure out how to properly code for a singular RBI. Portals of discovery.

- **playType** an indicator describing the type of play that occurs
  (home run, single, double, triple, walk, error, interference, out, bunt, SAC bunt, hit by pitch, strike out, illegal pitch, wild pitch, passed ball, steal)

Expectations

I expect to uncover fundamental differences between softball and baseball that will transform the way that current statisticians equate the two sports. The following are the base-out state run expectancy chart for the MLB and the chart for the run value of events in the MLB^{57}, including the run expectancy before each event._{1}

```
pandoc.table(MLBBOStates)
```

```
--------------------------------------------------
    baseoutstates       NoOuts   OneOut   TwoOuts 
---------------------- -------- -------- ---------
      Nobody on         0.555    0.297     0.117  

    Runner on 1st       0.953    0.573     0.251  

    Runner on 2nd       1.189    0.725     0.344  

    Runner on 3rd       1.482    0.983     0.387  

Runners on 1st and 2nd  1.573    0.971     0.466  

Runners on 1st and 3rd  1.904    1.243     0.538  

Runners on 2nd and 3rd  2.052    1.467     0.634  

     Bases Loaded       2.417     1.65     0.815  
--------------------------------------------------
```

```
pandoc.table(playvalueMLB)
```

```
---------------------------------------------
     playTypeMLB         RTOEMLB   run_expMLB 
---------------------- --------- ------------
       home run          1.942      0.533    

        triple           1.616      0.553    

        double           1.311      0.547    

        single           1.025      0.551    

         walk            0.849       0.52    

     hit by pitch        0.969      0.584    

         out             0.24       0.538    

      strikeout          0.207      0.517    

catcher's interference   1.083      0.655    

        error            1.132      0.586    
```

```
    illegal pitch        0.949      0.712

     wild pitch            1        0.716

    passed ball          1.026      0.741

    stolen base          0.792      0.597

      SAC bunt           1.031      1.058

        bunt             0.481      0.409
---------------------------------------------
```

- **run_exp** the run expectancy after a unique event in **playType**
- **RTOE** the average runs to the end of an inning after a unique event in **playType**

Of the listed possibilites during a single at-bat, I expect to see events like stolen bases, errors, walks and bunts to matter significantly more as opposed to the heavy emphasis that baseball places on extra-base hits. Because of these differences in values for certain plays, I expect to find undervalued events and skillsets that baseball may ignore but softball should specifically highlight.

Based on these expectations, I expect to find that there will not only be differences in the constants in the wOBA expression, but also some variables will be weighted more than others differently than the MLB's wOBA. For example, if errors are going to be worth more, than the constant might be well higher than singles, for example.

{MLB's wOBA} ({Singles} + {Doubles} + {Triples} + {Home runs} + {Hit-By-Pitches} + {Walks} + {Reached Base on Error}) / {Plate Appearances}

Overall, I expect to find that softball follows a holistically different pattern than baseball. The dimensions of the field are much different; the bases are closer together; the ball is larger; the outfield is smaller, etc. Whether that will manifest itself in a special reliance on home run hitters, speedy players, great defensive players or pitchers is what I expect to figure out.

Formula wise, this project is based mostly on data-shaping and reading and testing the results, but the observations are inherently formulaic — and the end result is to come up with a statistic based on the wOBA format that properly explains a hitter's individual value and a team's ability to win based on their hitting performance.

{Runs} 1 + {Team wOBA} {Runs} 1 + {Team mywOBA}

{Winning %} 1 + {Team wOBA} {Winning %} 1 + {Team myOBA}

The above two comparisons are the other hypothesis tests of this project, evaluating the differences in the coefficients for both winning percentage and runs.

Analysis

Softball's base-out state chart

The first step was to analyze the maximum amount of runs in each inning, which can be seen in the runs_df data set here.^{12}

- **runs** the maximum amount of runs in each half of an inning

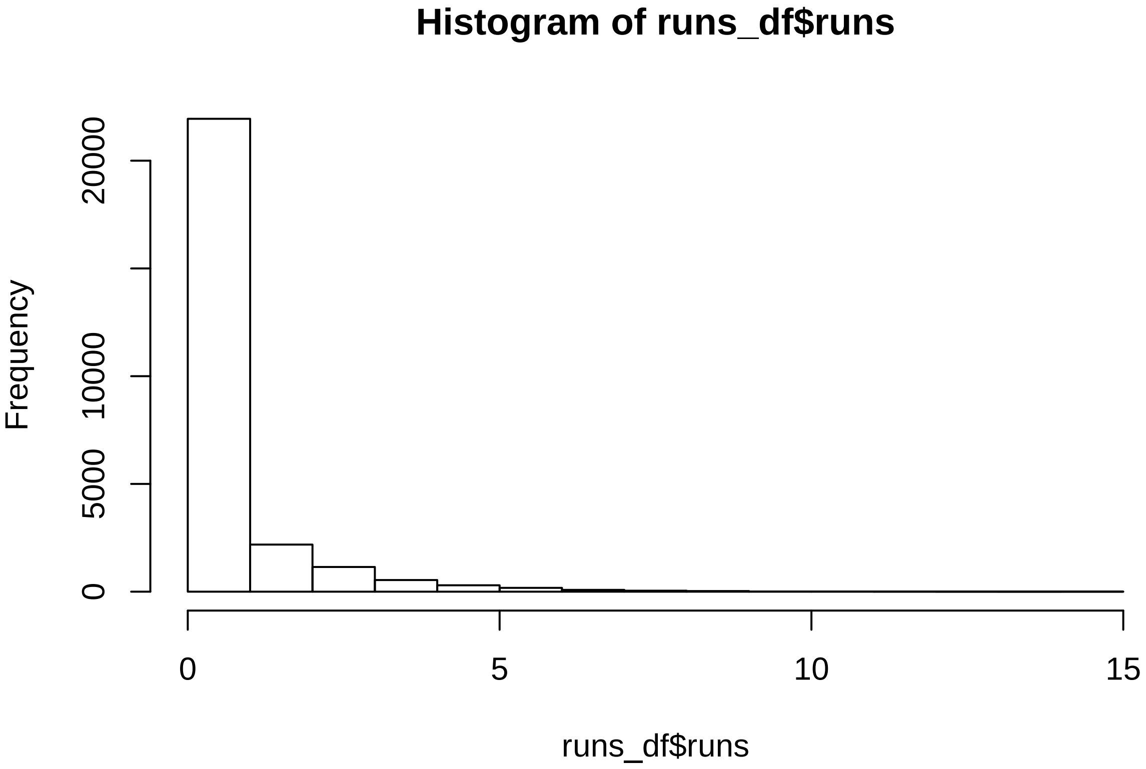
```
library(dplyr)
runs_df <- season %>% dplyr::group_by(gameID, inningID, top_bottom) %>% dplyr::summarize(runs = max(runs
    na.rm = T))

summary(runs_df$runs)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   0.000   0.000   0.684   1.000  15.000 
```

```
length(runs_df$runs)
```

```
[1] 26472
```

```
hist(runs_df$runs)
```



The summary shows us the average amount of runs scored per inning, of which there are 26,472. The distribution of runs is, as expected based on the distribution of runs_before_play in the dataset, heavily skewed towards one, but we can see that it extends to a fairly unfortunate inning — it happens.

The next step is to create a similar dataset tracking the average runs_before_play for each of the 24 base-states in base_state_before for each inning and then subtract that from the max in each inning. That way, the resulting column is the average of the amount of runs after the base-state occurs. For example: ^{13}

```
library(dplyr)
nooneon_noout_runs_df <- season[season$base_state_before == "Nobody on, No Outs", ] %>% dplyr::group_by
    inningID, top_bottom) %>% dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

nooneon_noout_runs_df <- merge(nooneon_noout_runs_df, runs_df, all = TRUE, sort = FALSE)
```

```
nooneon_noout_runs_df$runsReal <- nooneon_noout_runs_df$runs - nooneon_noout_runs_df$runsbefore
mean(nooneon_noout_runs_df$runsReal)
```

```
[1] 0.6557
```

Again, we create a new data set, this one representing the state of "Nobody on, No Outs" in which

- **runsbefore** is the average amount of runs scored before the given state per inning.

Then,

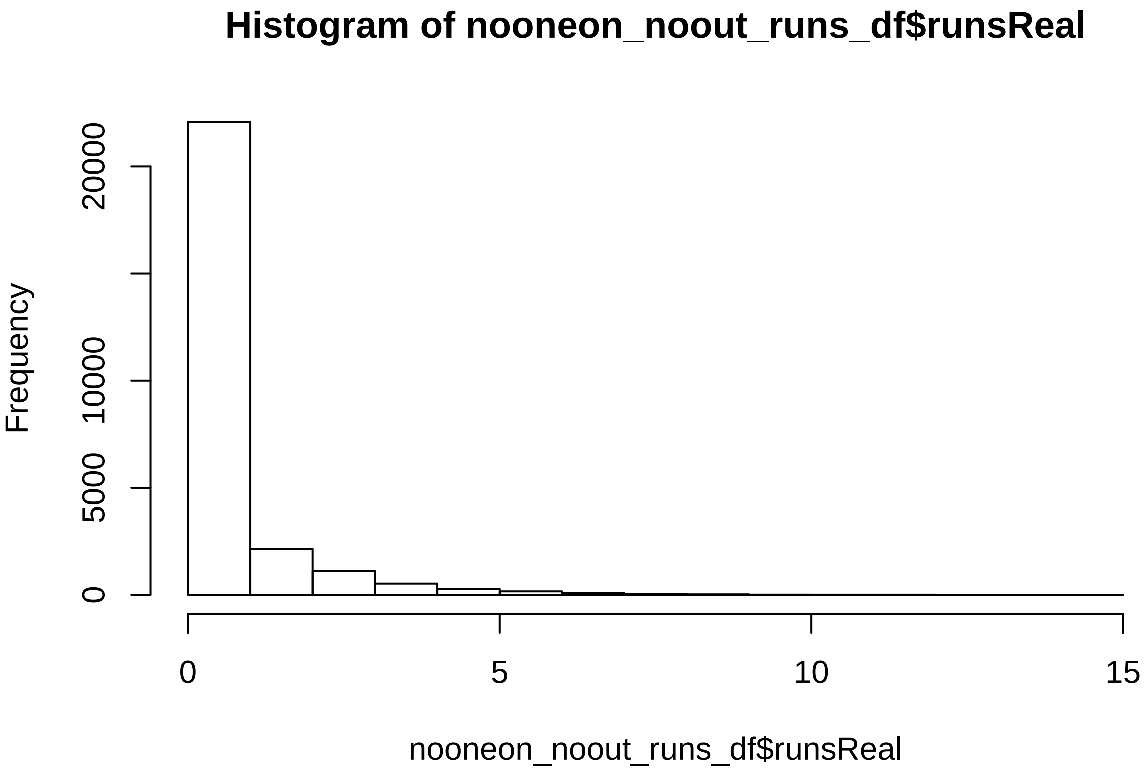
- **runsReal** is the average amount of runs scored after the given state per inning.

The average is .656, very similar to the .684 we calculated for every inning. This is because every state starts with "Nobody on, No Outs" and the only other occurence after would have to be after a home run with no outs. The similarity of these statistics speaks to the functionality of the metric, as it is very close but slightly depreciated, which is expected based on lineup shifts and pitching changes after home runs with no outs in an inning.

```
summary(nooneon_noout_runs_df$runsReal)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   0.000   0.000   0.656   1.000  15.000 
```

```
hist(nooneon_noout_runs_df$runsReal)
```



There is also very similar distributions and quartile readings for the states, which backs up the aforementioned functionality.

I continued to do this for all of the base-out states, until I recorded all of them and created "Softball's Base-Out state" chart.^{13-39}

```
NoOutsS <- c(0.656, 1.1, 1.39, 1.517, 1.769, 1.948, 2.089, 2.294)
OneOutS <- c(0.301, 0.603, 0.77, 1.029, 1.052, 1.244, 1.336, 1.506)
```

```
TwoOutsS <- c(0.097, 0.234, 0.3006, 0.411, 0.454, 0.495, 0.566, 0.616)
baseoutstates
```

```
[1] "Nobody on"                "Runner on 1st"            "Runner on 2nd"
[4] "Runner on 3rd"            "Runners on 1st and 2nd" "Runners on 1st and 3rd"
[7] "Runners on 2nd and 3rd" "Bases Loaded"
```

```
chartS <- data.frame(baseoutstates, NoOutsS, OneOutS, TwoOutsS)

pandoc.table(chartS)
```

```
-----------------------------------------------------
    baseoutstates       NoOutsS   OneOutS   TwoOutsS
---------------------- --------- --------- ----------
      Nobody on          0.656     0.301     0.097

    Runner on 1st         1.1      0.603     0.234

    Runner on 2nd        1.39      0.77      0.3006

    Runner on 3rd        1.517     1.029     0.411

Runners on 1st and 2nd   1.769     1.052     0.454

Runners on 1st and 3rd   1.948     1.244     0.495

Runners on 2nd and 3rd   2.089     1.336     0.566

     Bases Loaded        2.294     1.506     0.616
-----------------------------------------------------
```

```
pandoc.table(MLBBOStates)
```

```
--------------------------------------------------
    baseoutstates       NoOuts   OneOut   TwoOuts
---------------------- -------- -------- ---------
      Nobody on         0.555    0.297     0.117

    Runner on 1st       0.953    0.573     0.251

    Runner on 2nd       1.189    0.725     0.344

    Runner on 3rd       1.482    0.983     0.387

Runners on 1st and 2nd  1.573    0.971     0.466

Runners on 1st and 3rd  1.904    1.243     0.538

Runners on 2nd and 3rd  2.052    1.467     0.634

     Bases Loaded       2.417     1.65     0.815
--------------------------------------------------
```

```r
standarderrorsNoout <- c(standarderror1, standarderror4, standarderror7, standarderror10, standarderror
    standarderror16, standarderror19, standarderror22)
standarderrorsOneout <- c(standarderror2, standarderror5, standarderror8, standarderror11, standarderro
    standarderror17, standarderror20, standarderror23)
standarderrorsTwoout <- c(standarderror3, standarderror6, standarderror9, standarderror12, standarderro
    standarderror18, standarderror21, standarderror24)
Totalchart <- data.frame(baseoutstates, NoOutsS, OneOutS, TwoOutsS, NoOuts, OneOut, TwoOuts, standarder
    standarderrorsOneout, standarderrorsTwoout)
Totalchart$Sig_Noouts <- with(Totalchart, (NoOutsS - NoOuts)/standarderrorsNoout)
Totalchart$Sig_Oneout <- with(Totalchart, (OneOutS - OneOut)/standarderrorsOneout)
Totalchart$Sig_Twoouts <- with(Totalchart, (TwoOutsS - TwoOuts)/standarderrorsTwoout)
Totalchart$Pvalue0 <- with(Totalchart, 2 * pnorm(-abs(Sig_Noouts)))
Totalchart$Pvalue1 <- with(Totalchart, 2 * pnorm(-abs(Sig_Oneout)))
Totalchart$Pvalue2 <- with(Totalchart, 2 * pnorm(-abs(Sig_Twoouts)))
print(Totalchart)
```

```
           baseoutstates NoOutsS OneOutS TwoOutsS NoOuts OneOut TwoOuts standarderrorsNoout
1              Nobody on   0.656   0.301   0.0970  0.555  0.297   0.117            0.007667
2           Runner on 1st   1.100   0.603   0.2340  0.953  0.573   0.251            0.017168
3           Runner on 2nd   1.390   0.770   0.3006  1.189  0.725   0.344            0.030623
4           Runner on 3rd   1.517   1.029   0.4110  1.482  0.983   0.387            0.072337
5 Runners on 1st and 2nd   1.769   1.052   0.4540  1.573  0.971   0.466            0.039239
6 Runners on 1st and 3rd   1.948   1.244   0.4950  1.904  1.243   0.538            0.068527
7 Runners on 2nd and 3rd   2.089   1.336   0.5660  2.052  1.467   0.634            0.060568
8            Bases Loaded   2.294   1.506   0.6160  2.417  1.650   0.815            0.059877
  standarderrorsOneout standarderrorsTwoout Sig_Noouts Sig_Oneout Sig_Twoouts   Pvalue0    Pvalue1
1             0.006478             0.003859    13.1736    0.61746     -5.1825 1.245e-39 0.53693403
2             0.012666             0.007809     8.5625    2.36846     -2.1771 1.105e-17 0.01786252
3             0.016324             0.009138     6.5636    2.75672     -4.7493 5.251e-11 0.00583844
4             0.027836             0.015875     0.4838    1.65254      1.5118 6.285e-01 0.09842442
5             0.025781             0.016499     4.9951    3.14179     -0.7273 5.881e-07 0.00167918
6             0.036533             0.023518     0.6421    0.02737     -1.8284 5.208e-01 0.97816288
7             0.030225             0.019844     0.6109   -4.33420     -3.4268 5.413e-01 0.00001463
8             0.033944             0.023039    -2.0542   -4.24223     -8.6375 3.995e-02 0.00002213
    Pvalue2
1 2.189e-07
2 2.948e-02
3 2.041e-06
4 1.306e-01
5 4.670e-01
6 6.749e-02
7 6.108e-04
8 5.745e-18
```

```r
pandoc.table(Totalchart)
```

```
-----------------------------------------------------------------------
    baseoutstates       NoOutsS   OneOutS   TwoOutsS   NoOuts   OneOut
---------------------- --------- --------- ---------- --------- --------
      Nobody on          0.656     0.301     0.097      0.555    0.297

    Runner on 1st         1.1      0.603     0.234      0.953    0.573
    Runner on 2nd        1.39      0.77      0.3006    1.189    0.725

    Runner on 3rd        1.517     1.029     0.411     1.482    0.983

Runners on 1st and 2nd   1.769     1.052     0.454     1.573    0.971

Runners on 1st and 3rd   1.948     1.244     0.495     1.904    1.243

Runners on 2nd and 3rd   2.089     1.336     0.566     2.052    1.467

     Bases Loaded        2.294     1.506     0.616     2.417     1.65
-----------------------------------------------------------------------

Table: Table continues below


-----------------------------------------------------------------------------
 TwoOuts   standarderrorsNoout   standarderrorsOneout   standarderrorsTwoout
--------- --------------------- ---------------------- ----------------------
  0.117         0.007667               0.006478               0.003859

  0.251          0.01717               0.01267                0.007809

  0.344          0.03062               0.01632                0.009138

  0.387          0.07234               0.02784                0.01587

  0.466          0.03924               0.02578                 0.0165

  0.538          0.06853               0.03653                0.02352

  0.634          0.06057               0.03022                0.01984

  0.815          0.05988               0.03394                0.02304
-----------------------------------------------------------------------------

Table: Table continues below


--------------------------------------------------------------------------------
 Sig_Noouts   Sig_Oneout   Sig_Twoouts      Pvalue0       Pvalue1     Pvalue2
------------ ------------ ------------- ----------------- ---------- ------------
   13.17        0.6175       -5.183        1.245e-39       0.5369   0.0000002189

   8.562        2.368        -2.177        1.105e-17      0.01786      0.02948

   6.564        2.757        -4.749     0.00000000005251  0.005838  0.000002041

   0.4838       1.653         1.512         0.6285        0.09842       0.1306

   4.995        3.142        -0.7273      0.0000005881    0.001679      0.467

   0.6421      0.02737       -1.828         0.5208         0.9782      0.06749
   0.6109       -4.334       -3.427          0.5413      0.00001463  0.0006108

   -2.054       -4.242       -8.638         0.03995      0.00002213  5.745e-18
--------------------------------------------------------------------------------
```

Individual t-tests show us that the difference in the metrics vary, but there is variance in the p-values that arise from the data. For example, runners on first and third with one out elicits a p-value of .9 — which means that we reject the null hypothesis in that the difference between the softball and MLB coefficient in this specific scenario is different. It is one of nine instances in which the p-value is above alpha: .05, so it is clear that same states are much different than others, and others are similar.

Softball's wOBA

Using the same dataset that had the maximum amount of runs, I did the same thing that I did for the base-states but with the unique events as labelled in **playType**. For example,

```
homerun_df <- season[season$playType == "home run", ] %>% dplyr::group_by(gameID, inningID, top_bottom)
    dplyr::summarize(runsbefore = mean(runs_before_play, na.rm = T))

homerun_df <- merge(homerun_df, runs_df, all = TRUE, sort = FALSE)
homerun_df$runsReal <- homerun_df$runs - homerun_df$runsbefore
```

runsbefore and **runsReal** mean the same thing as before, but now they track the average amount of runs that happen after an event — in this case, HR — per inning.

```
mean(homerun_df$runsReal, na.rm = T)
```

```
[1] 2.169
```

The average for home runs is 2.169, which will be compared to the MLB metric.^{39-54}

```
outvalueMLB <- 0.24
playvalue <- data.frame(playType, RTOE, run_exp, RTOEMLB, run_expMLB, outvalueMLB, standarderrorsplays)

playvalue$ExpectedGain <- playvalue$RTOE - playvalue$run_exp
playvalue$outvalue <- 0.296
playvalue$OBAcoeffbase <- with(playvalue, RTOE - outvalue)
playvalue$OBAcoeffbaseMLB <- with(playvalue, RTOEMLB - outvalueMLB)
```

First, we create a dataset with the runs to the end of the inning and the run_exp based on our calculated base-out states and then find

-**OBAcoeffbase** which is the runs to the end of the inning for an individual event relative to that of an out, unweighted. This can also be read as the run value of the event, as it tracks how many runs it 'creates' compared to if the batter had gotten out.

```
Softball_Stats2016$oneB <- with(Softball_Stats2016, H - twoB - threeB - HR)

Softball_Stats2016$mybaseOBA <- (0.579 * Softball_Stats2016$BB + 0.734 * Softball_Stats2016$HBP + 0.564
    Softball_Stats2016$oneB + 1.187 * Softball_Stats2016$twoB + 1.425 * Softball_Stats2016$threeB + 1.87
    Softball_Stats2016$HR + 0.821 * Softball_Stats2016$RBOE)/(Softball_Stats2016$AB + Softball_Stats2016
    Softball_Stats2016$IBB + Softball_Stats2016$SH + Softball_Stats2016$SF + Softball_Stats2016$HBP)

mean(Softball_Stats2016$mybaseOBA)
```

```
[1] 0.2845
```

```
Softball_Stats2016$OBP <- (Softball_Stats2016$BB + Softball_Stats2016$HBP + Softball_Stats2016$oneB +
    Softball_Stats2016$twoB + Softball_Stats2016$threeB + Softball_Stats2016$HR)/(Softball_Stats2016$AB
    Softball_Stats2016$BB + Softball_Stats2016$IBB + Softball_Stats2016$SH + Softball_Stats2016$SF +
    Softball_Stats2016$HBP)
mean(Softball_Stats2016$OBP)
```

```
[1] 0.3499
```

```
mean(Softball_Stats2016$OBP)/mean(Softball_Stats2016$mybaseOBA)
```

```
[1] 1.23
```

```
playvalue$OBAcoeff <- with(playvalue, OBAcoeffbase * 1.23)
playvalue$OBAcoeffMLB <- with(playvalue, OBAcoeffbaseMLB * 1.15)
print(playvalue)
```

```
                 playType   RTOE run_exp RTOEMLB run_expMLB outvalueMLB standarderrorsplays
1                home run 2.1690  0.6727   1.942      0.533        0.24            0.026709
2                  triple 1.7210  0.6413   1.616      0.553        0.24            0.055378
3                  double 1.4830  0.6765   1.311      0.547        0.24            0.021771
4                  single 0.8600  0.6540   1.025      0.551        0.24            0.008360
5                    walk 0.8750  0.6519   0.849      0.520        0.24            0.012815
6             hit by pitch 1.0300  0.6649   0.969      0.584        0.24            0.032023
7                     out 0.2960  0.6473   0.240      0.538        0.24            0.002272
8               strikeout 0.2420  0.6282   0.207      0.517        0.24            0.005606
9  catcher's interference 0.9714  0.6378   1.083      0.655        0.24            0.230037
10                  error 1.1170  0.6430   1.132      0.586        0.24            0.041464
11          illegal pitch 1.3630  0.9160   0.949      0.712        0.24            0.116364
12             wild pitch 1.0770  0.8115   1.000      0.716        0.24            0.027659
13            passed ball 1.0440  0.8087   1.026      0.741        0.24            0.045228
14            stolen base 1.0110  0.8140   0.792      0.597        0.24            0.022307
15               SAC bunt 0.9120  1.1860   1.031      1.058        0.24            0.029245
16                   bunt 0.9442  0.8513   0.481      0.409        0.24            0.021706
   ExpectedGain outvalue OBAcoeffbase OBAcoeffbaseMLB OBAcoeff OBAcoeffMLB
1        1.4963    0.296       1.8730           1.702  2.30379     1.95730
2        1.0797    0.296       1.4250           1.376  1.75275     1.58240
3        0.8065    0.296       1.1870           1.071  1.46001     1.23165
4        0.2060    0.296       0.5640           0.785  0.69372     0.90275
5        0.2231    0.296       0.5790           0.609  0.71217     0.70035
6        0.3651    0.296       0.7340           0.729  0.90282     0.83835
7       -0.3513    0.296       0.0000           0.000  0.00000     0.00000
8       -0.3862    0.296      -0.0540          -0.033 -0.06642    -0.03795
9        0.3336    0.296       0.6754           0.843  0.83074     0.96945
10       0.4740    0.296       0.8210           0.892  1.00983     1.02580
11       0.4470    0.296       1.0670           0.709  1.31241     0.81535
12       0.2655    0.296       0.7810           0.760  0.96063     0.87400
13       0.2353    0.296       0.7480           0.786  0.92004     0.90390
14       0.1970    0.296       0.7150           0.552  0.87945     0.63480
15      -0.2740    0.296       0.6160           0.791  0.75768     0.90965
16       0.0929    0.296       0.6482           0.241  0.79729     0.27715
```

After, in order to equate our wOBA to a percentage with a similar distribution and league-average as

-**OBP** on-base percentage,

so I created

-OBAcoeff the official wOBA coefficients taken from the Softball season data.

{my wOBA} (.69* {Singles} + 1.46* {Doubles} + 1.75* {Triples} + 2.30* {Home runs} + .90* {Hit-By-Pitches} + .71* {Walks} +1.01*
{Reached Base on Error}) / {Plate Appearances}

```r
playvalue$standerrorsweighted <- with(playvalue, standarderrorsplays * sqrt(1.23))
playvalue$ExpectedGainMLB <- playvalue$RTOEMLB - playvalue$run_expMLB
playvalue$z <- with(playvalue, (ExpectedGain - ExpectedGainMLB)/(standerrorsweighted))
playvalue$pvalue <- with(playvalue, 2 * pnorm(-abs(z)))
print(pandoc.table(playvalue))
```

```
----------------------------------------------------------------------------
       playType         RTOE   run_exp    RTOEMLB    run_expMLB    outvalueMLB
---------------------- ------ --------- --------- ------------ -------------
       home run        2.169   0.6727     1.942      0.533         0.24

        triple         1.721   0.6413     1.616      0.553         0.24

        double         1.483   0.6765     1.311      0.547         0.24

        single          0.86    0.654     1.025      0.551         0.24

         walk          0.875   0.6519     0.849       0.52         0.24

     hit by pitch       1.03   0.6649     0.969      0.584         0.24

         out           0.296   0.6473     0.24       0.538         0.24

      strikeout        0.242   0.6282     0.207      0.517         0.24

catcher's interference 0.9714  0.6378     1.083      0.655         0.24

        error          1.117    0.643     1.132      0.586         0.24

    illegal pitch      1.363    0.916     0.949      0.712         0.24

      wild pitch       1.077   0.8115       1        0.716         0.24

     passed ball       1.044   0.8087     1.026      0.741         0.24

     stolen base       1.011    0.814     0.792      0.597         0.24

       SAC bunt        0.912    1.186     1.031      1.058         0.24

         bunt          0.9442  0.8513     0.481      0.409         0.24
----------------------------------------------------------------------------

Table: Table continues below


--------------------------------------------------------------------------------
 standarderrorsplays   ExpectedGain   outvalue   OBAcoeffbase   OBAcoeffbaseMLB
--------------------- -------------- ---------- -------------- -----------------
```

```
       0.02671            1.496        0.296        1.873            1.702

       0.05538             1.08        0.296        1.425            1.376

       0.02177            0.8065       0.296        1.187            1.071

       0.00836            0.206        0.296        0.564            0.785

       0.01282            0.2231       0.296        0.579            0.609

       0.03202            0.3651       0.296        0.734            0.729

      0.002272           -0.3513       0.296          0                0

      0.005606           -0.3862       0.296        -0.054          -0.033

        0.23              0.3336       0.296        0.6754           0.843

       0.04146            0.474        0.296        0.821            0.892

       0.1164             0.447        0.296        1.067            0.709

       0.02766            0.2655       0.296        0.781            0.76

       0.04523            0.2353       0.296        0.748            0.786

       0.02231            0.197        0.296        0.715            0.552

       0.02925            -0.274       0.296        0.616            0.791

       0.02171            0.0929       0.296        0.6482           0.241
--------------------------------------------------------------------------------

Table: Table continues below


------------------------------------------------------------------------
 OBAcoeff   OBAcoeffMLB   standerrorsweighted   ExpectedGainMLB     z
---------- ------------- --------------------- ----------------- -------
  2.304        1.957            0.02962              1.409        2.947

  1.753        1.582            0.06142              1.063       0.2719

   1.46        1.232            0.02415              0.764        1.76

  0.6937      0.9027           0.009271              0.474       -28.91

  0.7122      0.7003            0.01421              0.329       -7.451

  0.9028      0.8383            0.03552              0.385       -0.5603

    0            0              0.00252             -0.298       -21.15

 -0.06642    -0.03795          0.006217              -0.31       -12.26
```

```
  0.8307      0.9694            0.2551                0.428        -0.37

   1.01        1.026            0.04599               0.546       -1.566

  1.312       0.8153            0.1291                0.237        1.627

  0.9606       0.874            0.03068               0.284       -0.6031

   0.92       0.9039            0.05016               0.285       -0.9908

  0.8794      0.6348            0.02474               0.195       0.08084

  0.7577      0.9096            0.03243              -0.027       -7.615

  0.7973      0.2771            0.02407               0.072       0.8682
-----------------------------------------------------------------------

Table: Table continues below


-------------------
      pvalue
-------------------
     0.003207

      0.7857

      0.07838

    9.928e-184

0.00000000000009262

      0.5753

     2.514e-99

     1.555e-34

      0.7114

      0.1174

      0.1037

      0.5464

      0.3218

      0.9356

0.0000000000000263
```

```
      0.3853
-------------------

NULL
```

As we can see from the standard errors created for each of the playTypes — which were then weighted according to the weight applied to base OBA coefficients — there is a wide variance in the significance in the differences between the MLB and softball runs to the end of an inning relative to the expected run total before the play. These pvalues we treat individually, so pvalue below .05 represents an insignificant change — as we can see for walks. ^{55}

```
playvalue$pvalue[playvalue$playType == "error"]
```

```
[1] 0.1174
```

A pvalue of .1247 is greater than an alpha of .05, so we reject the null hypothesis that we touched on earlier that errors would represent a different amount of runs in softball than in baseball.

```
playvalue$pvalue[playvalue$playType == "single"]
```

```
[1] 9.928e-184
```

However, we see such a small p value for singles that we not only fail to reject the null hypothesis that singles are represented similarly in their effect on runs scored, but I'm taking notice of how significant the difference really is.

Application

- **myOBA** a measure of each individual team's OBA based on the constants and coefficients I calculated

```
lmmwSoftballOBA2016 <- lm(with(Softball_Stats2016, PCT ~ myOBA))
lmmwSoftballOBA2016
```

```

Call:
lm(formula = with(Softball_Stats2016, PCT ~ myOBA))

Coefficients:
(Intercept)        myOBA
      -0.27         2.17
```

```
rsquared(lmmwSoftballOBA2016)
```

```
[1] 0.2541
```

```
lmmwSoftballOBA2016R <- lm(with(Softball_Stats2016, R ~ myOBA))
lmmwSoftballOBA2016R
```

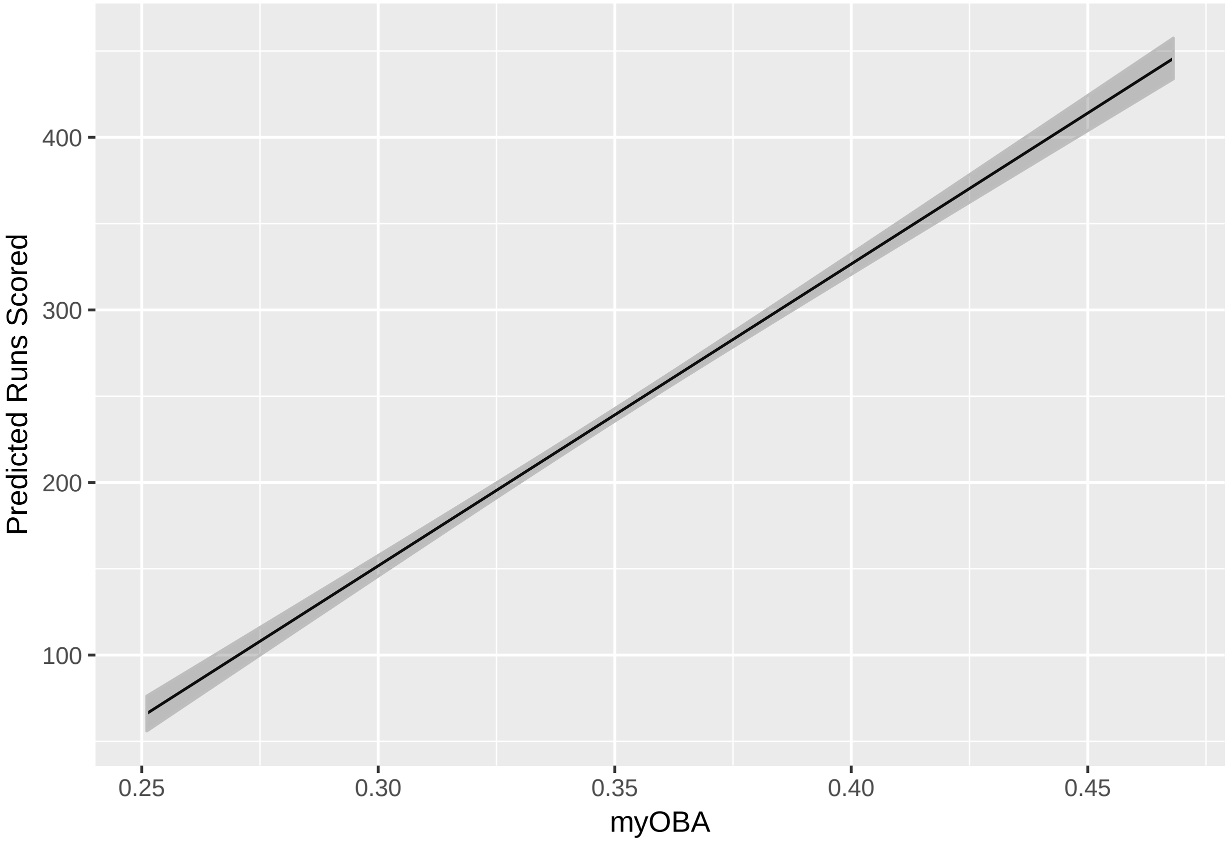
```

Call:
lm(formula = with(Softball_Stats2016, R ~ myOBA))

Coefficients:
(Intercept)        myOBA
       -373         1749
```

```
rsquared(lmmwSoftballOBA2016R)
```

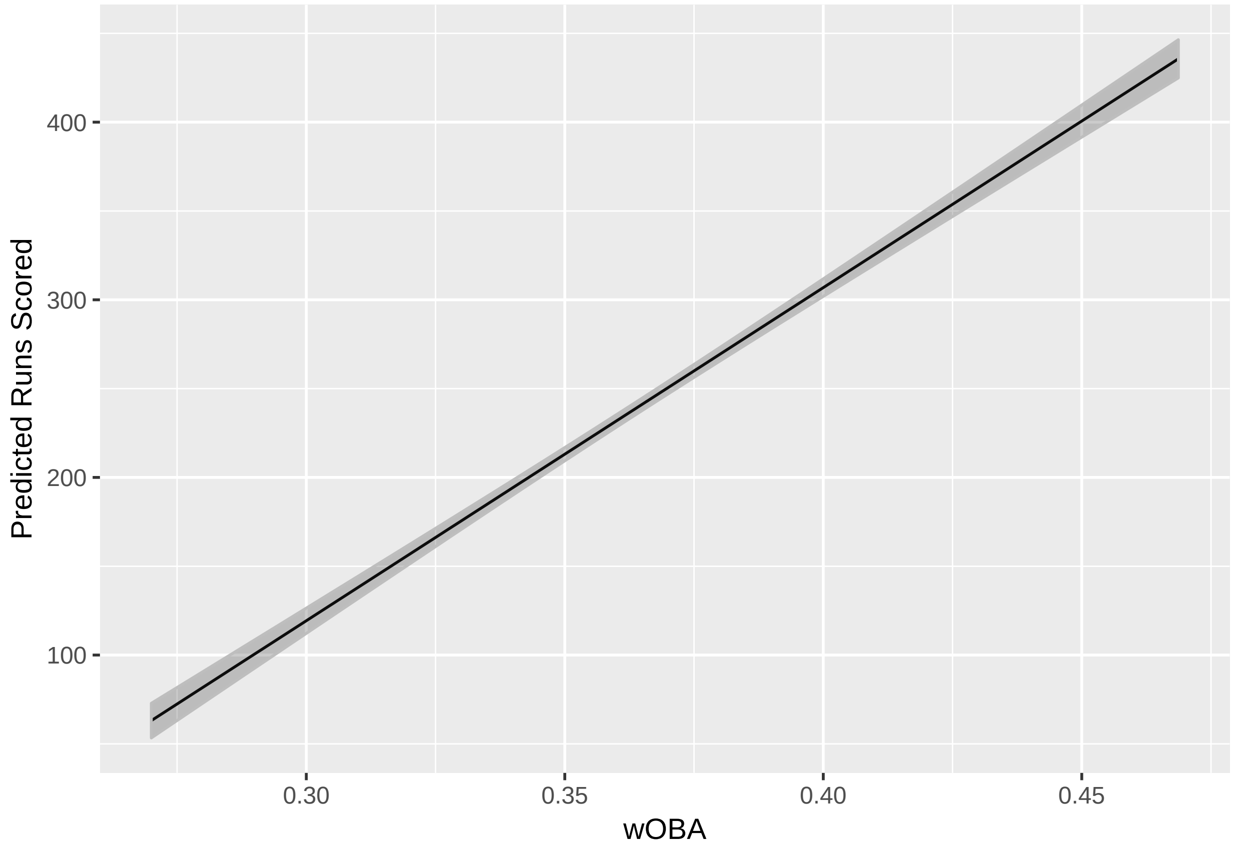
```
[1] 0.8152
```

As we can see, the r-squared values for this new statistic are very close to the original ones, despite the constants being very different for the most part.

```
Softball_Stats2016$fit <- predict(lmmwSoftballOBA2016R)
Softball_Stats2016$ul <- Softball_Stats2016$fit + 1.96 * predict(lmmwSoftballOBA2016R, se.fit = T)$se
Softball_Stats2016$ll <- Softball_Stats2016$fit - 1.96 * predict(lmmwSoftballOBA2016R, se.fit = T)$se

plot <- ggplot(data = Softball_Stats2016, aes(x = myOBA, y = fit)) + geom_line() + geom_ribbon(aes(ymin
    ymax = ul), col = "grey", alpha = 0.25) + labs(x = "myOBA", y = "Predicted Runs Scored", main = "my
plot
```

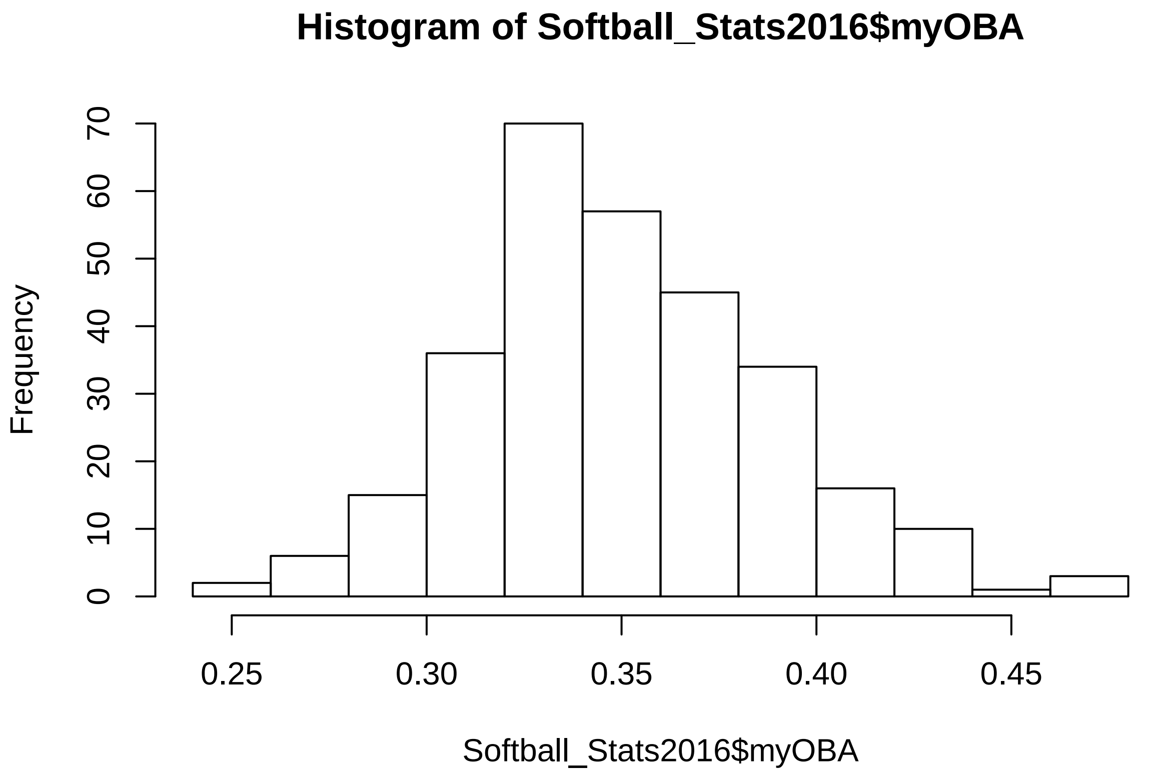


```
Softball_Stats2016$fit2 <- predict(lmwSoftballOBA2016R)
Softball_Stats2016$ul2 <- Softball_Stats2016$fit2 + 1.96 * predict(lmwSoftballOBA2016R, se.fit = T)$se
Softball_Stats2016$ll2 <- Softball_Stats2016$fit2 - 1.96 * predict(lmwSoftballOBA2016R, se.fit = T)$se
plot2 <- ggplot(data = Softball_Stats2016, aes(x = wOBA, y = fit2)) + geom_line() + geom_ribbon(aes(ymi
    ymax = ul2), col = "grey", alpha = 0.25) + labs(x = "wOBA", y = "Predicted Runs Scored", main = "wO
plot2
```



In comparing the two models graphically, including confidence intervals, it is clear that they follow a very similar pattern — as they should. One thing to notice in particular is the tightness of the interval in the middle, which is also observed when evaluating the proximity of the observations to the predictions displayed by the linear equation itself.

```
# Looking at the distribution of Softball wOBAs, looks normal
hist(Softball_Stats2016$myOBA)
```



```
range(Softball_Stats2016$myOBA)
```

```
[1] 0.2511 0.4681
```

```
count(Softball_Stats2016$myOBA >= 0.4)
```

```
      x freq
1 FALSE  265
2  TRUE   30
```

```
count(Baseball0Stats2016$wOBA >= 0.4)
```

```
      x freq
1 FALSE  295
2  TRUE    4
```

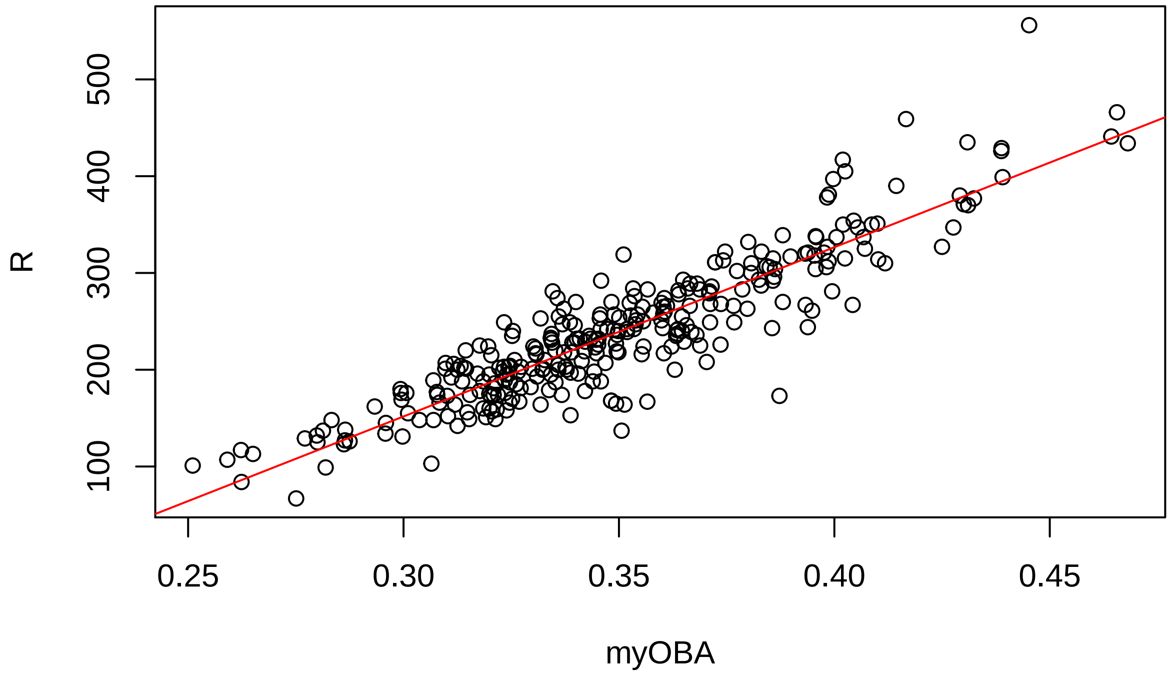
```
count(Baseball0Stats2016$wOBA >= 0.4)/length(Baseball0Stats2016$Institution)
```

```
         x    freq
1 0.000000 0.98662
2 0.003344 0.01338
```

```
count(Softball_Stats2016$myOBA >= 0.4)/length(Softball_Stats2016$Institution)
```

```
        x   freq
1 0.00000 0.8983
2 0.00339 0.1017
```

```
# Plot the function with the dataset
plot(x = Softball_Stats2016$myOBA, y = Softball_Stats2016$R, xlab = "myOBA", ylab = "R")
abline(lmmwSoftball0BA2016R, col = "red")
```



```
rsquared(lmmwSoftballOBA2016R)
```

```
[1] 0.8152
```

Looking at the distribution of myOBAs, it is normal in a very similar fashion to wOBA, which is to be expected. However, as the scales and constants were changed for softball, there is a similar range as well. According to “The Book”_{1}, great MLB teams eclipse .400; however, it seems that it is more common for softball teams than baseball teams at least — 10 percent of teams accrued myOBA percentages above 400, compared to 1.4 percent of baseball teams for wOBA. However, 10 percent of MLB teams would be only three teams, so it seems fair to label a myOBA of over .400 as top-tier.

```
rsquared(lmmwSoftballOBA2016R)
```

```
[1] 0.8152
```

```
rsquared(lmwSoftballOBA2016R)
```

```
[1] 0.8303
```

```
rsquared(lmmwSoftballOBA2016)
```

```
[1] 0.2541
```

```
rsquared(lmwSoftballOBA2016)
```

```
[1] 0.2667
```

```
SEmyModel <- mean(predict(lmwSoftballOBA2016R, se.fit = T)$se)
SEwModel <- mean(predict(lmmwSoftballOBA2016R, se.fit = T)$se)
dfmodel <- (length(Softball_Stats2016$Institution) - 1) + (length(BaseballOStats2016$Institution) - 1)
dfmodel
```

```
[1] 592
```

```
coef(lmmwSoftballOBA2016)[2] - coef(lmwSoftballOBA2016)[2]/((SEmyModel^2) + (SEwModel^2))
```

```
myOBA
1.979
```

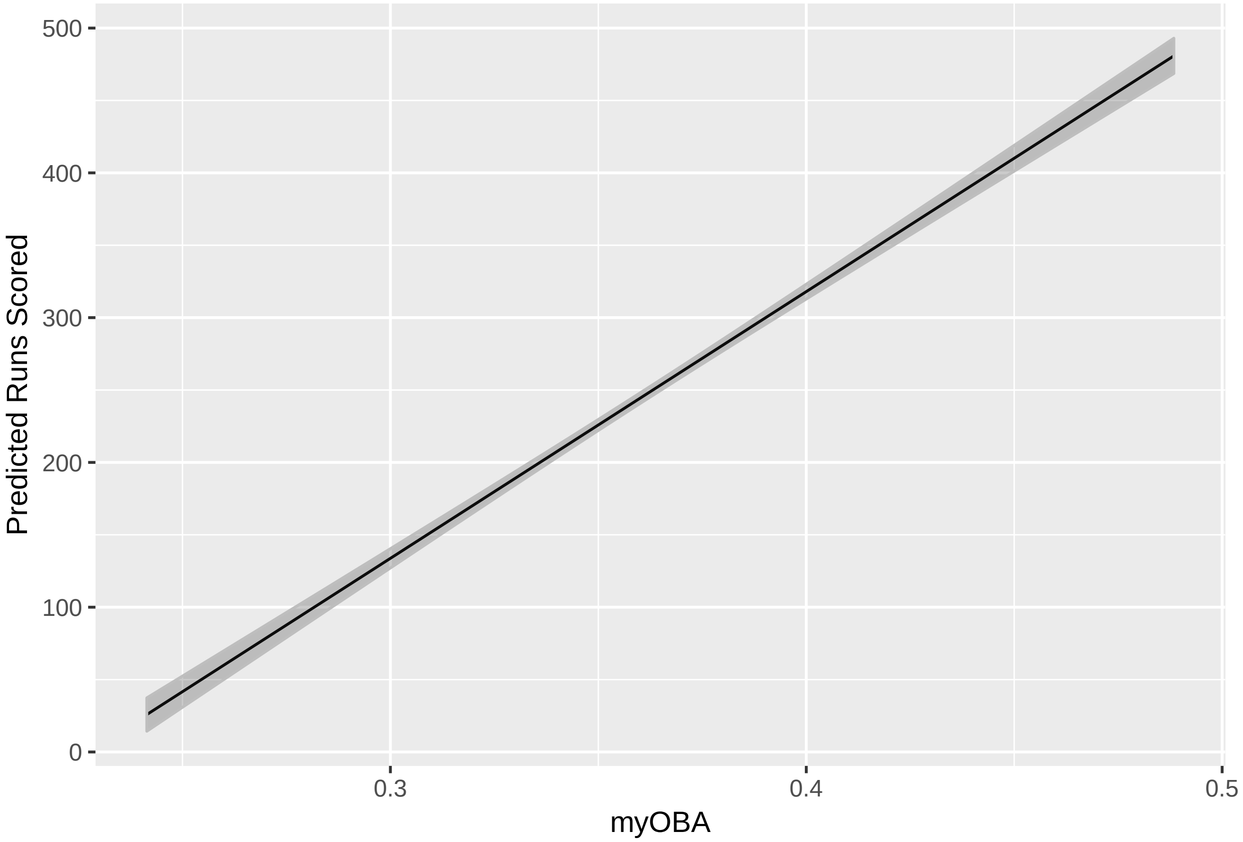
With an alpha set at 1.96, which represents the t-statistic for degrees of freedom 592 in the comparison of these two sample coefficients, we reject the null hypothesis that the slopes similarly describe softball. Of course, any difference in the slopes has to do with the changes in constants; however, this tells us that it is statistically significant that the two are predicting runs scored as a different factor of the given OBA statistic.

*I repeated this process for 2015 and 2014 in the code appendix. The results are eerily similar, and the difference between the rsquared values get even more narrow. In fact, for 2014, the difference between rsquared variables is .0078.^{56}

```
lmmwSoftballOBA2015R
```

```

Call:
lm(formula = with(Softball_Stats2015, R ~ myOBA))

Coefficients:
(Intercept)        myOBA
       -419         1842
```

```
plot3 <- ggplot(data = Softball_Stats2015, aes(x = myOBA, y = fit2)) + geom_line() + geom_ribbon(aes(ym
    ymax = ul2), col = "grey", alpha = 0.25) + labs(x = "myOBA", y = "Predicted Runs Scored", main = "my
plot3
```

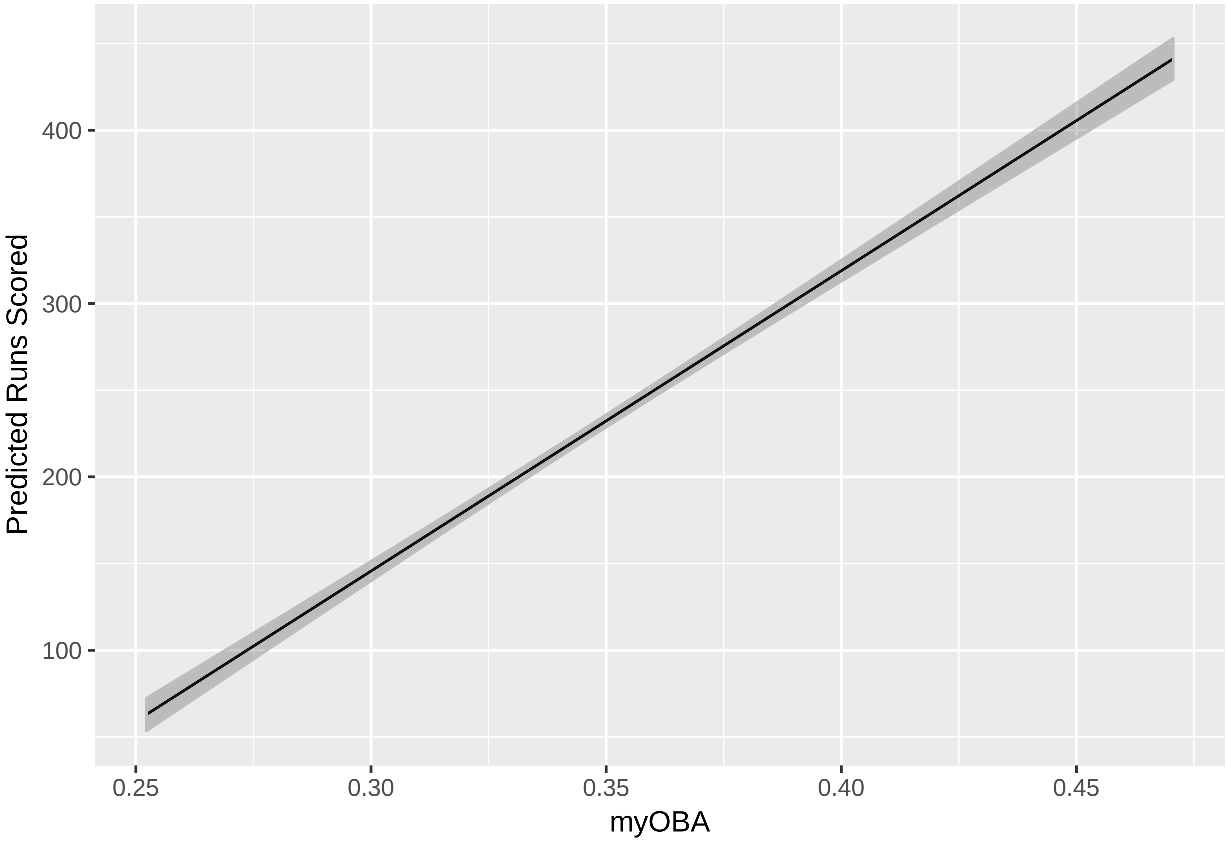


```
lmmwSoftballOBA2014R
```

```

Call:
lm(formula = with(Softball_Stats2014, R ~ myOBA))

Coefficients:
```

```
(Intercept)        myOBA
       -374         1732
```

```
plot4 <- ggplot(data = Softball_Stats2014, aes(x = myOBA, y = fit2)) + geom_line() + geom_ribbon(aes(ymi
    ymax = ul2), col = "grey", alpha = 0.25) + labs(x = "myOBA", y = "Predicted Runs Scored", main = "my
plot4
```



Discussion

There are many takeaways to be had from the results of the analysis. For the three sections, we can extrapolate three important facts for further interpretation:

Run Expectancy Chart

The difference in results among the two charts in Softball and the MLB is both evident and — despite the fact that there is no difference in some cases — it is clear that having two outs is a much more dire circumstance in softball than in baseball. Only one value in Sig_Twoouts is non-negative, “Runner on Third”, and it isn’t significant at alpha: .05, so even that could be the result of error.

However, base-states with zero or one outs have almost uniformly higher metrics and a significant percentage are significantly more. One theory behind this logic is that there are more runs scored per game in softball than in baseball, which seems to be true.

```
Softball_Stats2016$RPG <- Softball_Stats2016$R/Softball_Stats2016$AB
mean(Softball_Stats2016$RPG)
```

```
[1] 0.1704
```

```r
Baseball0Stats2016$RPG <- Baseball0Stats2016$R/Baseball0Stats2016$AB
mean(Baseball0Stats2016$RPG)
```

```
[1] 0.1669
```

However, that didn't stop the two outs metric to be deflated on average, other than if a runner was on third with two outs. Physically, a run only requires moving a runner 60 feet as opposed to 90, so that might explain the statistically non-significant difference in the two metrics.

There are three non-negative z statistics for a non-two out base-state, which are Bases Loaded, No outs; Runners on 2nd and 3rd, One out and Bases Loaded, One out. This may signal decision-makers to place a heavier emphasis on getting the first batter on base, for example, or choosing to intentionally walk good players more often in order to take advantage of common misconceptions about softball's similarity to baseball.

One of the most interesting things to look at here are the differences when the states change. If a runner is on first and second with no one out, how many runs is the team expected to get if a player has a 2% chance of getting a double and a 30% chance of beating out a bunt single? What stolen base percentage would a baserunner have to have and what batting average would the batter have to have in order to justify stealing a base? When do you intentional walk a player? These are just two of the infinite amount of questions that could be statistically evaluated by coaches before the game even starts.

One of the most translatable examples of this is the jump from Runner on 3rd to Runners on 1st and 3rd with no outs. A lot of softball teams rely on the suicide squeeze to get runners from third in and give up an out, reducing a base-state of 1.517 to get a run but create a base-state that only expects an additional .301 runs. $1.517 > 1+.301$, and losing .2 runs a game would add up over a season of 60-plus games for good teams.

If a runner is on first in a one-run game with two outs and the batter can get that runner in from 1st 2% of the time and from 2nd 30% of the time, would you decide to steal?

No. The batter is expected to score the runner from 1st 2 percent of the time, and the change in run expectancy from "Runner on 1st, Two outs" to "Runner on 2nd, Two outs" is .067. The steal gives you a 90% chance to get that gain, so the expected run gain is now .0603. The chance you can capitalize on the expected run gain is 30% * .0603, which is .018 expected run gain — below .02.

Softball's wOBA

The depreciated value of the single steals the headlines in this section, as the pvalue pushes the limits of nonzero. We already demonstrated that softball sees many more singles than baseball; however, that alone cannot account for the difference. In my opinion, many more singles occur in softball that do not leave the infield and do not advance runners and thus do not create runs than in baseball. This principle could signal a loss in value for small-ball players that rely on speed and beating out "slaps", a popular play in softball.

In fact, with double and triple having insignificant differences compared to the MLB's value, the single stands out even more. The strikeout is also much more significant in softball than in baseball, which makes sense given the ease in which runners can advance in softball compared to baseball.

Interestingly, the much-maligned SAC bunt is significantly worse in softball, and the decrease from the initial run expectancy is higher. If anything, this could be the result of human scoring error, as slaps are typically labelled bunts when the batter is thrown out at first. But, if you consider this metric a combination of bunts and slaps that end in sacrifices, then you have to consider reevaluating making the decision to ever slap/bunt sacrifice.

Another theme across all of these sections is the significant weight of a home run. Softball stadiums are typically 200 feet, while MLB and college stadiums can get to over 400 feet in center. The difference in sheer area in outfields between the two sports is significant, and a smaller outfield could contribute in keeping double/triple RTOE rates near the MLB's. But home runs are the great equalizer in that no fielder can do anything different to limit the runs created by a home run, so the difference might be a true reflection of the fact that there are more runs scored in softball than in baseball. That, or softball players hit more home runs:

```
Softball_Stats2016$HRPG <- Softball_Stats2016$HR/Softball_Stats2016$AB
mean(Softball_Stats2016$HRPG)
```

```
[1] 0.02431
```

```
BaseballOStats2016$HRPG <- BaseballOStats2016$HR/BaseballOStats2016$AB
mean(BaseballOStats2016$HRPG)
```

```
[1] 0.02035
```

```
Softball_Stats2016$ABPG <- Softball_Stats2016$AB/Softball_Stats2016$G
mean(Softball_Stats2016$ABPG)
```

```
[1] 26.18
```

A .00096 difference in HRs per AB is fairly significant when considering that there are about 52 ABs per team in a given game, which means that that there is about one more home run hit every twenty games in softball than in baseball. In a dataset with 2000 games, that's 100 more home runs. Either way, it's clear that softball has a greater reliance on home runs for runs than does baseball. Home run hitters are very valuable in softball.

The data on plays can help coaches recruit or coach third base, when they usually inform the baserunner and batter what to do. With this information, I would be less inclined to have a slapper slap as opposed to swing away; I might let a home-run hitter swing at a 3-0 pitch and I might recruit players such as to place a higher value on hitting the ball hard because doubles, triples and home runs are all more valuable in softball than in baseball.

Application

$$mywOBA(.69*Singles+1.46*Doubles+1.75*Triples+2.30*Homeruns+.90*Hit-By-Pitches+.71*Walks+1.01*Reache$$

Looking at my wOBA statistic, the comparison between the changes in the constants from the MLB equation to the softball version aligns with the inflation in runs scored and the depreciation of events like the single and the walk in particular.

The applications of this in the analysis section are well-documented and discussed, and the rsquared values are not only strong but very close to the baseball-optimized wOBA. However, this project is largely about the derivation of those constants and the function's applicability both for evaluating players and making decisions in games and during the course of a season. With that being said, my top three optimization opportunities in the game of softball based on these results:

1) Recruit players that don't just rely on their speed to get on base. Singles that don't leave the infield not only necessitate incredible talent to result in singles in the first place, and those likely don't move non-forced runners, which is a major portion of the game.
2) The differences in the base-out states make a huge difference when evaluating when to intentional walk a player or when to — for example — pinch hit a batter late in a game. The intentional walk can be an important tool when a good player is up in an advantageous base-out state; however, the depreciation for bases loaded states might lend the fielding team an opportunity to create an out with a well-timed intentional walk. Also, being mindful of these base-states on the fly and the expected values based on the batter will always create windows for optimization of hitters during a game. For example, a slapper up with runners on first and second might be disadvantageous compared to a low-average power hitter, which would let a coach make a timely substition.
3) Never SAC bunt. It goes without saying in the big leagues after Moneyball, and that should catch on in softball. The SAC bunt creates an out, which crushes expected run totals when a SAC makes the second out, and, if you followed steps one and two, the batter should be able to get the ball out of the infield at a decent enough rate to justify seining away in nearly every scenario.

Bibliography

— Tango (Author), Mitchel Lichtman (Author),, Tom. “The Book: Playing The Percentages In Baseball Paperback – April 28, 2014.” The Book: Playing The Percentages In Baseball: Tom Tango, Mitchel Lichtman, Andrew Dolphin: 9781494260170: Amazon.com: Books. N.p., n.d. Web. 24 Apr. 2017.

— Nachtigal, Jon. “Softball’s Top 10 Teams and Weighted On-Base Average (wOBA).” Fastpitch Analytics. N.p., 15 May 2013. Web. 15 May 2017.

— NCAA. NCAA Softball Statistics. Raw data. Indianapolis, Indiana.

— Octonion. NCAA softball play-by-play data. 2016. Raw data. Github.com.