

# MSD 2019 Final Project

A replication and extension of Wage disparity and team productivity: evidence from Major League Baseball by Craig Depkin, 1999

*Forrest Hofmann (fhh2112) & Sagar Lal (sl3946)*

*2019-05-03 13:45:24*

## Contents

<b>Introduction</b>	<b>1</b>
Problem Description . . . . .	1
Motivation . . . . .	1
Data Source . . . . .	1
<b>Reproduction</b>	<b>1</b>
Reproduction Code . . . . .	1
Reproduction Notes . . . . .	3
Reproduction Analysis . . . . .	3
<b>Extension</b>	<b>3</b>
Extension Code . . . . .	3
Extension Notes . . . . .	10
Extension Analysis . . . . .	10
<b>Postface</b>	<b>10</b>

## Introduction

### Problem Description

### Motivation

### Data Source

## Reproduction

### Reproduction Code

```
teams <- read_csv(here('data/teams.csv'))
salaries <- read_csv(here('data/salaries.csv'))

teams$WSWin <- as.logical(teams$WSWin == 'Y')
teams <- teams %>%
  filter(1985 <= yearID & yearID <= 2016) %>%
  mutate(winPercentage = W / (W + L) * 1000)

salaries <- salaries %>%
```

```

filter(1985 <= yearID & yearID <= 2016) %>%
mutate(salaryMil = salary / 1000000)

teams <- teams %>%
  inner_join(salaries) %>%
  group_by(yearID, teamID, G, W, L, WSWin, winPercentage) %>%
  summarize(totalSalaryMil = sum(salaryMil))

salaries <- salaries %>%
  inner_join(teams) %>%
  mutate(salaryShare = salaryMil / totalSalaryMil * 100) %>%
  mutate(salaryShareSquared = salaryShare ^ 2) %>%
  select(yearID, teamID, playerID, salary, salaryShare, salaryShareSquared)

teams <- teams %>%
  inner_join(salaries) %>%
  group_by(yearID, teamID, G, W, L, winPercentage, WSWin, totalSalaryMil) %>%
  summarize(HHI = sum(salaryShareSquared))

teams_old <- teams %>%
  filter(1985 <= yearID & yearID <= 1998) %>%
  mutate(normalizedYear = yearID - 1985)

salaries_old <- salaries %>%
  filter(1985 <= yearID & yearID <= 1998)

summary(teams_old$winPercentage)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   327.2   456.8   498.4   500.0   543.2   703.7
sd(teams_old$winPercentage)

## [1] 66.22653
summary(teams_old$totalSalaryMil)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.88   12.76   22.32   25.16   36.29   72.36
sd(teams_old$totalSalaryMil)

## [1] 14.22702
summary(teams_old$HHI)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   427.5   668.6   756.3   815.6   879.1  5300.1
sd(teams_old$HHI)

## [1] 322.5687
linear_fixed_old <- lm(formula = winPercentage ~ totalSalaryMil + HHI +
                        normalizedYear + teamID + 0,
                        data = teams_old)
summary(linear_fixed_old)$coefficients[1:3,]

##              Estimate Std. Error  t value    Pr(>|t|)

```

```
## totalSalaryMil  2.1493337  0.4551468  4.722287  3.404816e-06
## HHI            -0.0120376  0.0114311 -1.053057  2.930560e-01
## normalizedYear -5.4184670  1.5790948 -3.431375  6.738811e-04

linear_random_old <- lm(formula = winPercentage ~ totalSalaryMil + HHI + normalizedYear,
                        data = teams_old)
summary(linear_random_old)$coefficients[1:4,]

##              Estimate Std. Error  t value    Pr(>|t|)
## (Intercept)  494.46265725 10.80464965 45.763877 2.501463e-155
## totalSalaryMil  2.27827992  0.38799272  5.871966  9.513353e-09
## HHI          -0.01402974  0.01077516 -1.302045  1.937025e-01
## normalizedYear -6.05527713  1.38637176 -4.367715  1.627858e-05

log_log_fixed_old <- lm(formula = log(winPercentage) ~ log(totalSalaryMil) + log(HHI) +
                        normalizedYear + teamID + 0,
                        data = teams_old)
summary(log_log_fixed_old)$coefficients[1:3,]

##              Estimate Std. Error  t value    Pr(>|t|)
## log(totalSalaryMil)  0.068481958 0.023048764  2.971177 0.00317594
## log(HHI)            -0.043092478 0.034083173 -1.264333 0.20696902
## normalizedYear      -0.006815679 0.003712079 -1.836081 0.06721103

log_log_random_old <- lm(formula = log(winPercentage) ~ log(totalSalaryMil) + log(HHI) +
                        normalizedYear,
                        data = teams_old)
summary(log_log_random_old)$coefficients[1:4,]

##              Estimate Std. Error  t value    Pr(>|t|)
## (Intercept)        6.336734123 0.228514736 27.730090 9.463580e-93
## log(totalSalaryMil)  0.077748160 0.019984364  3.890450 1.184254e-04
## log(HHI)           -0.046572660 0.031165596 -1.494361 1.359244e-01
## normalizedYear      -0.008653452 0.003294086 -2.626966 8.969280e-03
```

## Reproduction Notes

- original author did not describe how time fixed effects are accounted for (across expansion periods or every year)
- no discussion about limiting to 25 man roster vs 40 man roster
- no discussion of cut players, traded players
- no discussion of signing bonuses

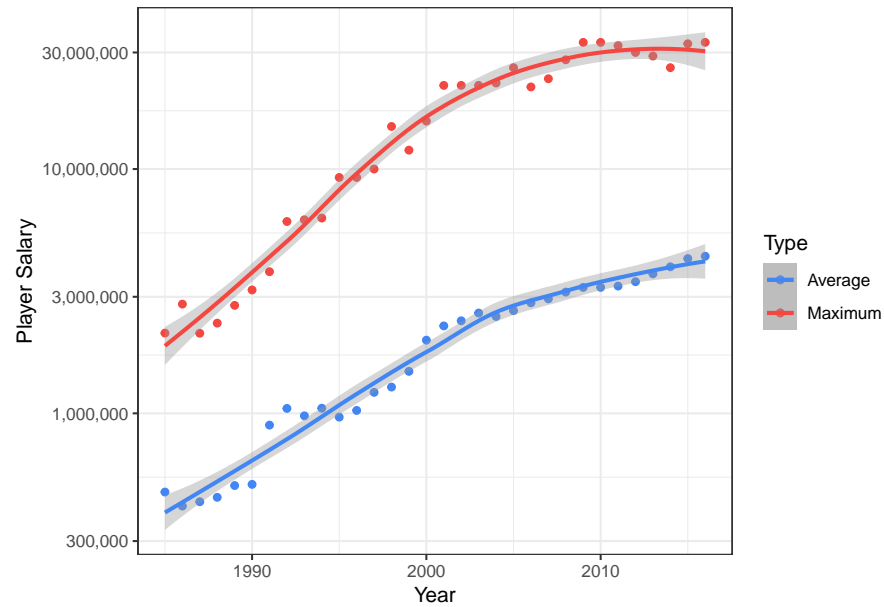
## Reproduction Analysis

## Extension

## Extension Code

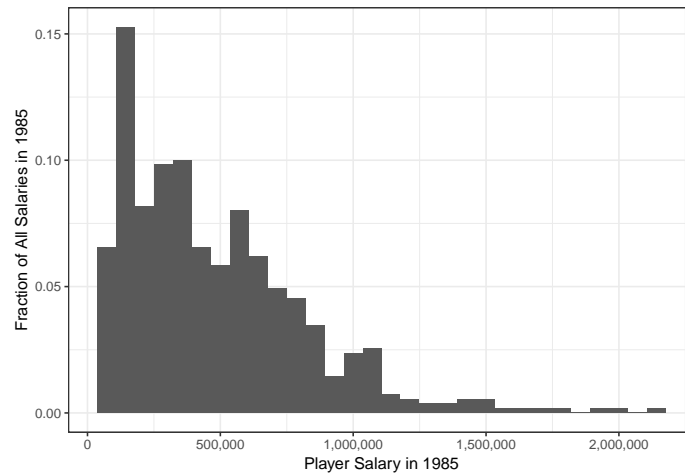
```
salary_vs_time <- salaries %>%
  group_by(yearID) %>%
  summarize(avg = mean(salary), max = max(salary))
```

```
ggplot(data = salary_vs_time) +
  geom_point(aes(x = yearID, y = avg, color = 'Average')) +
  geom_smooth(aes(x = yearID, y = avg, color = 'Average')) +
  geom_point(aes(x = yearID, y = max, color = 'Maximum')) +
  geom_smooth(aes(x = yearID, y = max, color = 'Maximum')) +
  scale_color_manual(values = c('#4286f4', '#f44741')) +
  scale_y_log10(labels = comma) +
  labs(color = 'Type') +
  xlab('Year') +
  ylab('Player Salary')
```

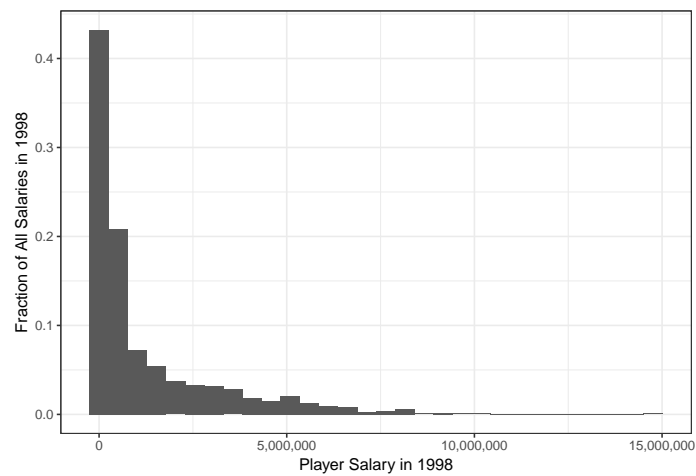


```
salaries_1985 <- filter(salaries, yearID == 1985)
salaries_1998 <- filter(salaries, yearID == 1998)
salaries_2016 <- filter(salaries, yearID == 2016)

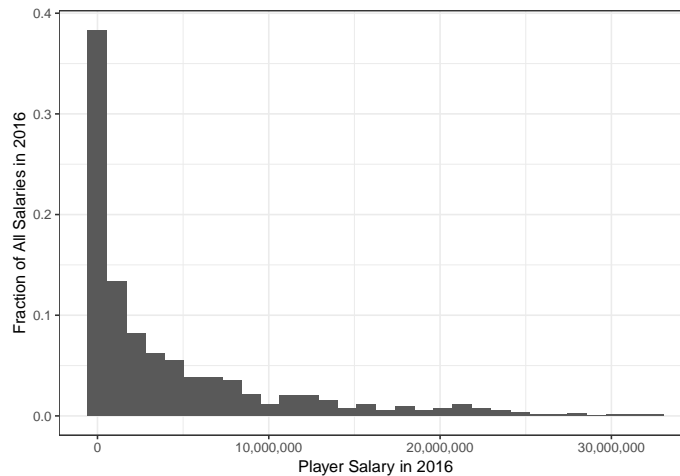
ggplot(data = salaries_1985) +
  geom_histogram(aes(x = salary, y = (..count..) / sum(..count..))) +
  scale_x_continuous(labels = comma) +
  xlab('Player Salary in 1985') +
  ylab('Fraction of All Salaries in 1985')
```



```
ggplot(data = salaries_1998) +
  geom_histogram(aes(x = salary, y = (..count..) / sum(..count..))) +
  scale_x_continuous(labels = comma) +
  xlab('Player Salary in 1998') +
  ylab('Fraction of All Salaries in 1998')
```



```
ggplot(data = salaries_2016) +
  geom_histogram(aes(x = salary, y = (..count..) / sum(..count..))) +
  scale_x_continuous(labels = comma) +
  xlab('Player Salary in 2016') +
  ylab('Fraction of All Salaries in 2016')
```

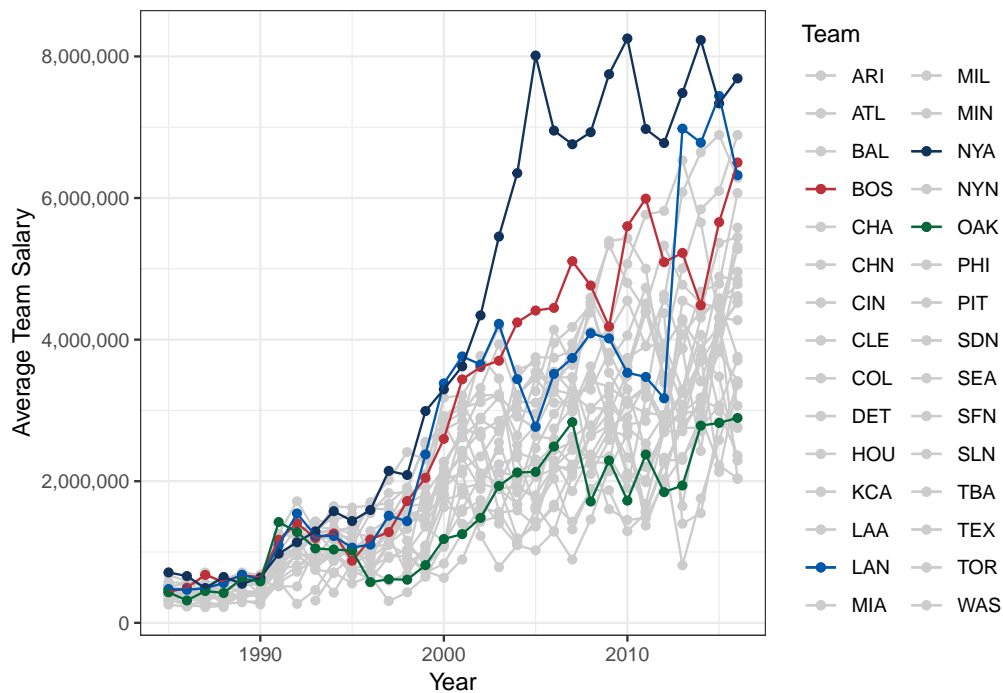


```
current_teamIDs <- c('ARI', 'ATL', 'BAL', 'BOS', 'CHA', 'CHN', 'CIN', 'CLE', 'COL', 'DET',
                    'HOU', 'KCA', 'LAA', 'LAN', 'MIA', 'MIL', 'MIN', 'NYA', 'NYN', 'OAK',
                    'PHI', 'PIT', 'SDN', 'SEA', 'SFN', 'SLN', 'TBA', 'TEX', 'TOR', 'WAS')
team_colors <- c('#cccccc', '#cccccc', '#cccccc', '#BD3039', '#cccccc',
                '#cccccc', '#cccccc', '#cccccc', '#cccccc', '#cccccc',
                '#cccccc', '#cccccc', '#cccccc', '#0157a8', '#cccccc',
                '#cccccc', '#cccccc', '#11325b', '#cccccc', '#04683b',
                '#cccccc', '#cccccc', '#cccccc', '#cccccc', '#cccccc',
                '#cccccc', '#cccccc', '#cccccc', '#cccccc', '#cccccc')
colored_teamIDs <- c('BOS', 'LAN', 'NYA', 'OAK')

team_salary_vs_time <- salaries %>%
  filter(teamID %in% current_teamIDs) %>%
  group_by(yearID, teamID) %>%
  summarize(avg = mean(salary)) %>%
  mutate(flag = teamID %in% colored_teamIDs)

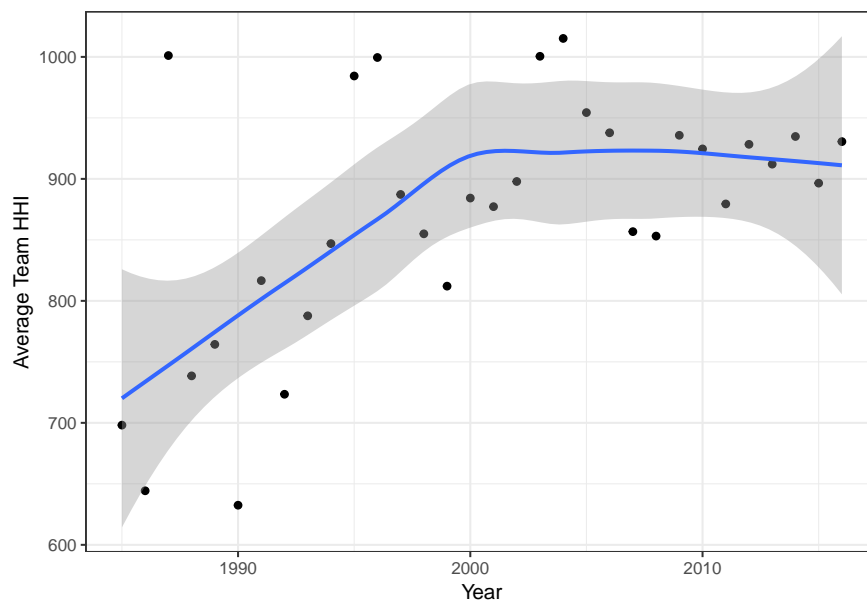
underlay_data <- filter(team_salary_vs_time, !flag)
overlay_data <- filter(team_salary_vs_time, flag)

ggplot() +
  geom_point(data = underlay_data, aes(x = yearID, y = avg, color = teamID)) +
  geom_line(data = underlay_data, aes(x = yearID, y = avg, color = teamID)) +
  geom_point(data = overlay_data, aes(x = yearID, y = avg, color = teamID)) +
  geom_line(data = overlay_data, aes(x = yearID, y = avg, color = teamID)) +
  scale_y_continuous(labels = comma) +
  scale_color_manual(values = team_colors) +
  labs(color = 'Team') +
  xlab('Year') +
  ylab('Average Team Salary')
```



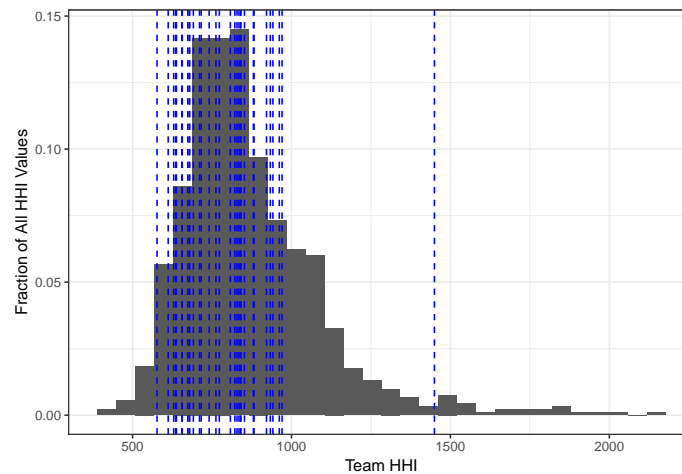
```
hhi_vs_time <- teams %>%
  group_by(yearID) %>%
  summarize(avg = mean(HHI))

ggplot(data = hhi_vs_time) +
  geom_point(aes(x = yearID, y = avg)) +
  geom_smooth(aes(x = yearID, y = avg)) +
  xlab('Year') +
  ylab('Average Team HHI')
```



```
ggplot(data = filter(teams, mean(teams$HHI) - 5 * sd(teams$HHI) <= HHI & HHI <= mean(teams$HHI) + 5 * sd(teams$HHI))) +
  geom_histogram(aes(x = HHI, y = (..count..) / sum(..count..))) +
  geom_vline(data = filter(teams, WSWin), aes(xintercept = HHI), color = 'blue', linetype = 'dashed') +
```

```
xlab('Team HHI') +
ylab('Fraction of All HHI Values')
```

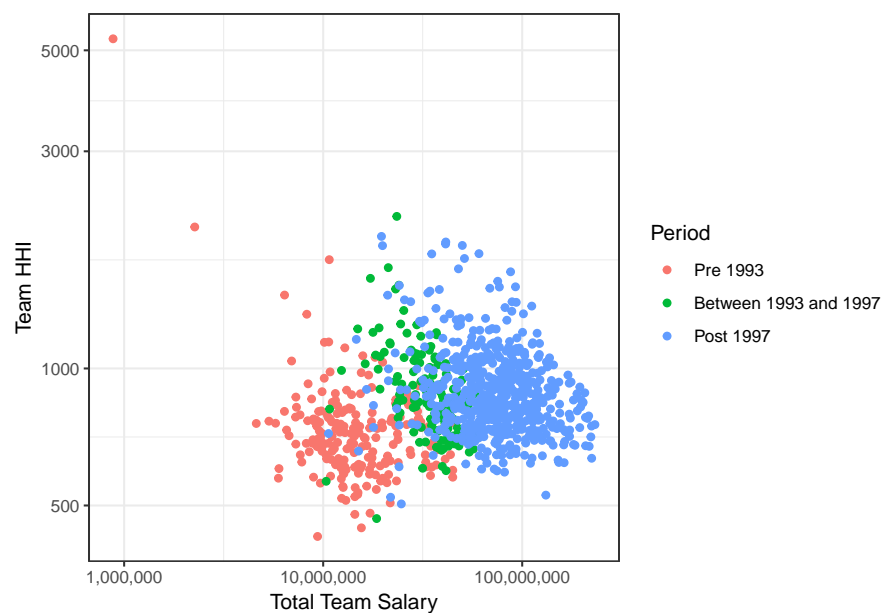


```
year_to_period <- function(year) {
  if (year <= 1992) {
    return('Pre 1993')
  }
  else if (1993 <= year & year <= 1997) {
    return('Between 1993 and 1997')
  }
  else {
    return('Post 1997')
  }
}

hhi_vs_total_salary <- teams %>%
  mutate(period = year_to_period(yearID))
hhi_vs_total_salary$period <- factor(hhi_vs_total_salary$period, levels = c('Pre 1993', 'Between 1993 and 1997', 'Post 1997'))

ggplot(data = hhi_vs_total_salary) +
  geom_point(aes(x = totalSalaryMil * 1000000, y = HHI, color = period)) +
  scale_x_log10(labels = comma) +
  scale_y_log10() +
  labs(color = 'Period') +
  xlab('Total Team Salary') +
  ylab('Team HHI')
```





```
teams_new <- teams %>%
  filter(1999 <= yearID & yearID <= 2016) %>%
  mutate(normalizedYear = yearID - 1999)
```

```
salaries_new <- salaries %>%
  filter(1999 <= yearID & yearID <= 2016)
```

```
linear_fixed_new <- lm(formula = winPercentage ~ totalSalaryMil + HHI +
  normalizedYear + teamID + 0,
  data = teams_new)
summary(linear_fixed_new)$coefficients[1:3,]
```

```
##              Estimate Std. Error  t value    Pr(>|t|)
## totalSalaryMil  0.4631102 0.12923492  3.583476 0.0003719897
## HHI            -0.0553023 0.01424983 -3.880909 0.0001178918
## normalizedYear -1.6956041 0.71841311 -2.360208 0.0186447534
```

```
linear_random_new <- lm(formula = winPercentage ~ totalSalaryMil + HHI + normalizedYear,
  data = teams_new)
summary(linear_random_new)$coefficients[1:4,]
```

```
##              Estimate Std. Error  t value    Pr(>|t|)
## (Intercept)  504.53034280 15.53375065 32.479622 9.998674e-129
## totalSalaryMil  0.68988493 0.08449909  8.164406 2.324053e-15
## HHI           -0.04525471 0.01386628 -3.263651 1.169986e-03
## normalizedYear -2.52975572 0.62573170 -4.042876 6.054218e-05
```

```
log_log_fixed_new <- lm(formula = log(winPercentage) ~ log(totalSalaryMil) + log(HHI) +
  normalizedYear + teamID + 0,
  data = teams_new)
summary(log_log_fixed_new)$coefficients[1:3,]
```

```
##              Estimate Std. Error  t value    Pr(>|t|)
## log(totalSalaryMil)  0.097454270 0.021330706  4.568732 6.177390e-06
## log(HHI)           -0.104976526 0.030167336 -3.479808 5.452894e-04
## normalizedYear     -0.004010857 0.001459855 -2.747436 6.221456e-03
```

```
log_log_random_new <- lm(formula = log(winPercentage) ~ log(totalSalaryMil) + log(HHI) +
                        normalizedYear,
                        data = teams_new)
summary(log_log_random_new)$coefficients[1:4,]
```

```
##              Estimate Std. Error  t value      Pr(>|t|)
## (Intercept)    6.285422459 0.222740979  28.218528 4.704158e-108
## log(totalSalaryMil) 0.125655854 0.014620475   8.594512 9.189347e-17
## log(HHI)         -0.085180977 0.028918094  -2.945594 3.363412e-03
## normalizedYear    -0.005434256 0.001300813  -4.177585 3.439323e-05
```

## Extension Notes

- note that minimum salary has increased over time: [https://www.baseball-reference.com/bullpen/Minimum\\_salary](https://www.baseball-reference.com/bullpen/Minimum_salary)

## Extension Analysis

## Postface

The following is a list of all packages used to generate these results.

```
sessionInfo()

## R version 3.5.2 (2018-12-20)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.4
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] bindrcpp_0.2.2  forcats_0.3.0  stringr_1.3.1  dplyr_0.7.8
## [5] purrr_0.3.0    readr_1.3.1    tidyr_0.8.2    tibble_2.0.1
## [9] ggplot2_3.1.0  tidyverse_1.2.1 scales_1.0.0    here_0.1
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.0      cellranger_1.1.0 plyr_1.8.4      compiler_3.5.2
## [5] pillar_1.3.1    bindr_0.1.1     tools_3.5.2     digest_0.6.18
## [9] lubridate_1.7.4 jsonlite_1.6     evaluate_0.12    nlme_3.1-137
## [13] gtable_0.2.0    lattice_0.20-38 pkgconfig_2.0.2  rlang_0.3.1
## [17] cli_1.0.1       rstudioapi_0.9.0 yaml_2.2.0       haven_2.0.0
## [21] xfun_0.4        withr_2.1.2     xml2_1.2.0      httr_1.4.0
## [25] knitr_1.21      hms_0.4.2       generics_0.0.2   rprojroot_1.3-2
## [29] grid_3.5.2      tidyselect_0.2.5 glue_1.3.0       R6_2.3.0
```

```
## [33] readxl_1.2.0      rmarkdown_1.11    modelr_0.1.2      magrittr_1.5
## [37] backports_1.1.3   htmltools_0.3.6   rvest_0.3.2       assertthat_0.2.0
## [41] colorspace_1.4-0  labeling_0.3       stringi_1.2.4     lazyeval_0.2.1
## [45] munsell_0.5.0     broom_0.5.1       crayon_1.3.4
```