Historical MLB Salary Analaysis

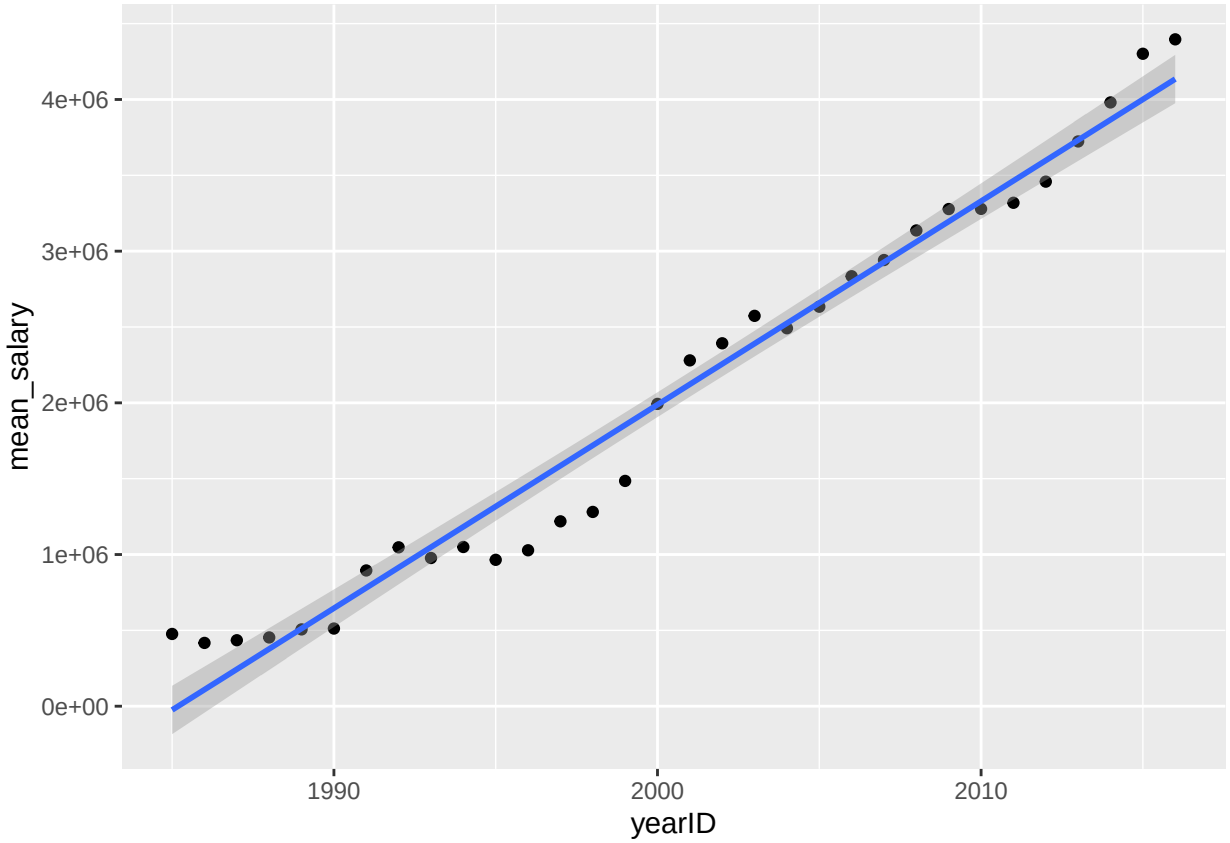
Sean Guglietti

2025-10-22

Retrieving data

```
# This code chunk will retrieve the salary data from github, and transform it
# to a dataframe.

url <- "https://raw.githubusercontent.com/SeanG347/LahmanBaseballDatabase/refs/heads/main/Salaries.csv"
salary.data <- read.csv(url, header = TRUE, sep = ",")
salary.df <- data.frame(salary.data)

# Only including records in the modern era.

salary.df <- salary.df[salary.df$yearID > 1953, ]

# Creating another data-frame which provides the mean salary by year.

mean_salary_by_year <- salary.df %>%
  group_by(yearID) %>%
  summarise(mean_salary = mean(salary))

# Plotting the mean salary by year

ggplot(data = mean_salary_by_year, aes(x=yearID, y = mean_salary)) +
  geom_point() +
  geom_smooth(method = lm) +
  labs(
    xlab = "Year",
    ylab = "Average Salary"
  )
```

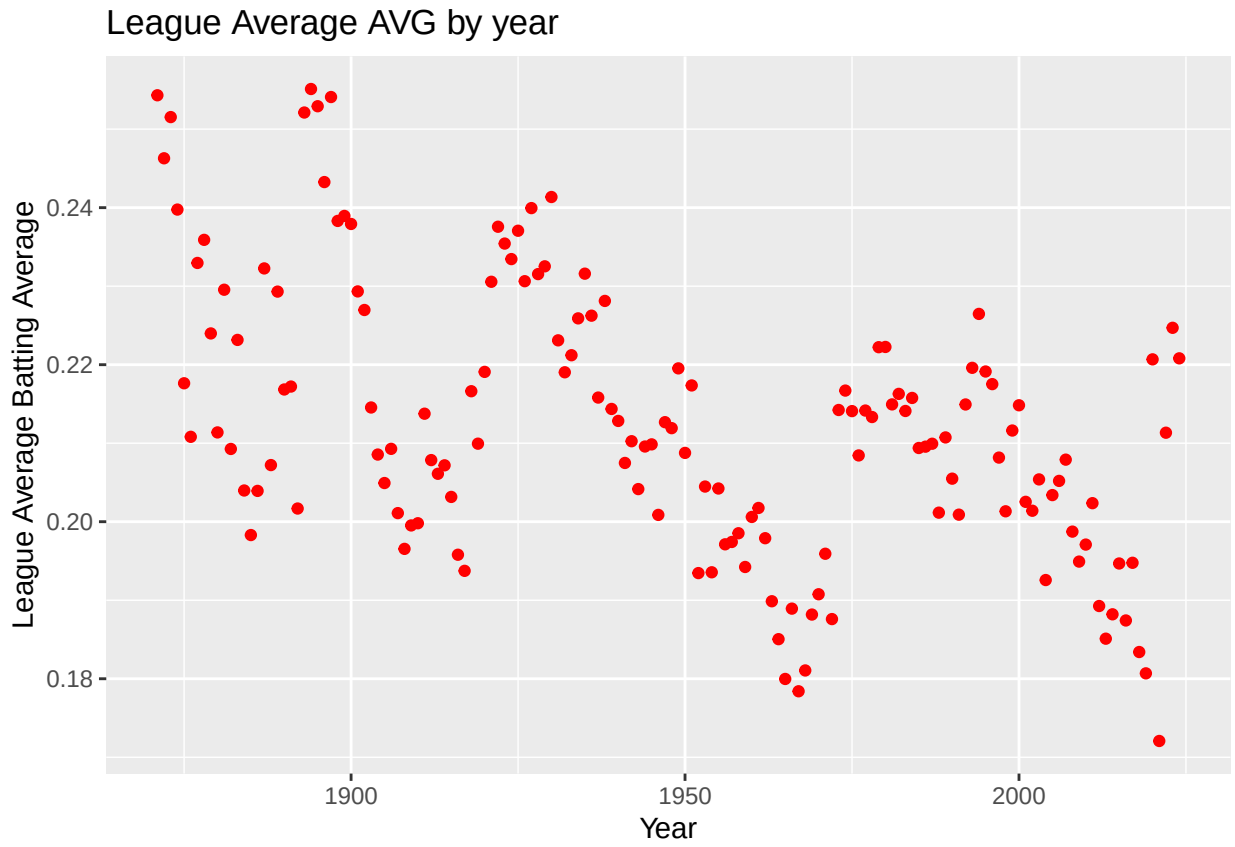
```
## `geom_smooth()` using formula = 'y ~ x'
```



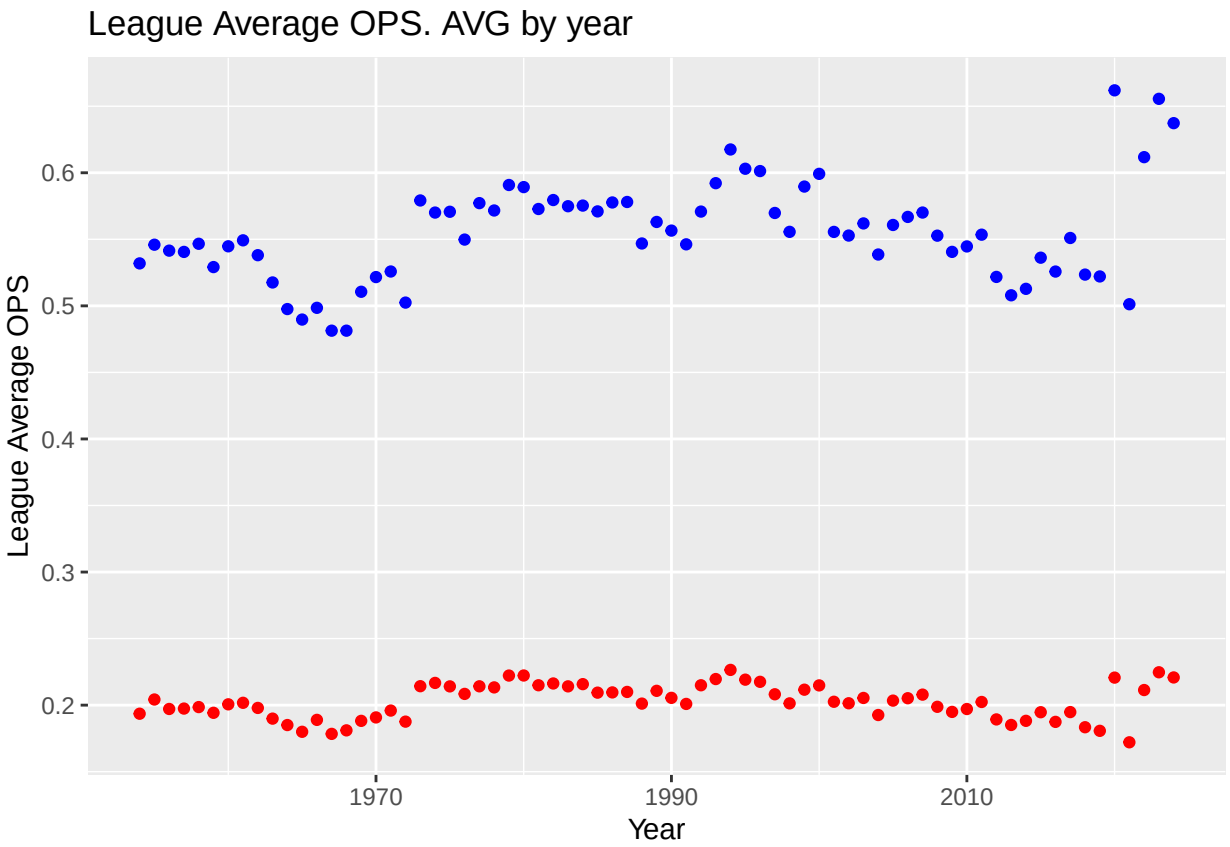

```r
# Creating a dataframe which groups by the teamID and yearID to give the
# yearly salary for each team.

yearly_salary_by_team <- salary.df %>%
  group_by(teamID, yearID) %>%
  summarise(teamSalary = sum(salary), .groups = "drop")
```

```r
# This code chunk retrieves MLB batting data, creates features of interest -
# (Batting average (AVG), on-base percentage (OBP), Slugging percentage (SLG),
# and on-base-plus-slugging (OPS)). This will provide an opportunity to see how
# these statistics have varied historically, but more importantly, will provide
# a way to quantify offensive production - something that has been historically
# more expensive than defensive production in the MLB.

url <- "https://raw.githubusercontent.com/SeanG347/LahmanBaseballDatabase/refs/heads/main/Batting.csv"
batting.data <- read.csv(url, header=TRUE,sep=",")
batting.df <- data.frame(batting.data)

# This ensures no records with 0 AB's, note this is possible as there are players
# who come into only a single game as a defensive substitute and never gets an
# AB. Or a player who only had a few times up and walked (which would be a
# plate appearance, not an at-bat)

batting.df = batting.df[batting.df$AB > 0, ]

# Calculating the AVG, SLG, PA, OBP, OPS statistics from the recorded number of
```

```r
# hits, doubles, triples, home runs, walks, sacrifice flies, hit-by-pitches,
# and at-bats.

batting.df$AVG = batting.df$H/batting.df$AB
batting.df$SLG = with(batting.df, ((H-X2B-X3B-HR)+2*X2B + 3*X3B + 4*HR)/AB)
batting.df$PA = with(batting.df, AB + BB + HBP + SF)
batting.df$OBP = with(batting.df, (H+BB+HBP)/PA)
batting.df$OPS = with(batting.df, OBP + SLG)

# Creating a dataframe which will have the mean batting average of all players
# per year.

mean_ba_by_year = batting.df %>%
  group_by(yearID) %>%
  summarise(mean_ba = mean(AVG))

# Plotting the mean_ba_by_year dataframe.

ggplot(data = mean_ba_by_year, aes(x=yearID, y = mean_ba)) +
  geom_point(color = "red") +
  labs(
    x = "Year",
    y = "League Average Batting Average",
    title = "League Average AVG by year"
  )
```



```r
# Filtering the batting.df dataframe to only include entries after 1953, which
# is when sac flies began actually being counted, and assigning that filtered
# dataframe to a new one, modern.batting.df

modern.batting.df <- batting.df[batting.df$yearID > 1953,]

# Creating two aggregate dataframes which include the mean BA and OPS by year
# from the modern.batting.df dataframe.

mean_ba_by_year_modern = modern.batting.df %>%
  group_by(yearID) %>%
  summarise(mean_avg = mean(AVG))

mean_ops_by_year = modern.batting.df %>%
  group_by(yearID) %>%
  summarise(mean_ops = mean(OPS))

means <- data.frame(year = mean_ops_by_year$yearID,
                    OPS = mean_ops_by_year$mean_ops,
                    AVG = mean_ba_by_year_modern$mean_avg)

# Plotting the means

ggplot(data = means) +
  geom_point(color = "red", data = means, aes(x=year,y=AVG)) +
  geom_point(color = "blue", data = means, aes(x=year,y=OPS))+
  labs(
    x = "Year",
    y = "League Average OPS",
    title = "League Average OPS. AVG by year"
  )
```



Notes

- I will from now on only use the "modern.batting.df" dataframe, which contains entries only after 1953, as that is when sacrifices were tracked, which allows the plate appearance stat to be accurate.
- Hypothesis:
  - Salary has a bearing on team success (H_0: $\beta_{salary} = 0$, H_1: $\beta_{salary} > 0$)
    - Note: Winning percentage will be used as the response to operationalize "team success"

```
url <- "https://raw.githubusercontent.com/SeanG347/LahmanBaseballDatabase/refs/heads/main/Teams.csv"
team.data <- read.csv(url, header=TRUE,sep=",")
team.df <- data.frame(team.data)

# Reducing data to only include the modern era (since sacrifices were tracked)
team.df <- team.df[team.df$yearID > 1953,]

# Making a new feature for the team's win percentage, this is to operationalize team success.
team.df$winPerc = team.df$W/(team.df$W+team.df$L)

# Making a new dataframe with only the relevant features
relFeatures = c('yearID','teamID','G','W','L','winPerc')
team.df.small <- team.df[relFeatures]
head(team.df.small)
```

```
##   yearID teamID   G  W  L   winPerc
```

```
## 2   1961    LAA 162 70 91 0.4347826
## 3   1962    LAA 162 86 76 0.5308642
## 4   1963    LAA 161 70 91 0.4347826
## 5   1964    LAA 162 82 80 0.5061728
## 6   1965    CAL 162 75 87 0.4629630
## 7   1966    CAL 162 80 82 0.4938272
```

```
# Sorting dataframe by winPerc
head(team.df.small[order(team.df.small$winPerc, decreasing = TRUE),])
```

```
##      yearID teamID   G   W  L   winPerc
## 1025   1954    CLE 156 111 43 0.7207792
## 1591   2020    LAN  60  43 17 0.7166667
## 2532   2001    SEA 162 116 46 0.7160494
## 1977   1998    NYA 162 114 48 0.7037037
## 1066   1995    CLE 144 100 44 0.6944444
## 1593   2022    LAN 162 111 51 0.6851852
```

```
# This chunk will group and summarise the salaries of the individual players into a dataframe containin

team.salary.df <- salary.df %>%
  group_by(teamID, yearID) %>%
  summarise(teamSalary = sum(salary, na.rm=TRUE))
```

```
## `summarise()` has grouped output by 'teamID'. You can override using the
## `.groups` argument.
```

```
team.df.small <- team.df.small %>%
  right_join(team.salary.df, by = c("teamID","yearID"))

head(team.df.small)
```

```
##   yearID teamID   G  W  L   winPerc teamSalary
## 1   1985    CAL 162 90 72 0.5555556   14427894
## 2   1986    CAL 162 92 70 0.5679012   14427258
## 3   1987    CAL 162 75 87 0.4629630   12843499
## 4   1988    CAL 162 75 87 0.4629630   11947388
## 5   1989    CAL 162 91 71 0.5617284   15097833
## 6   1990    CAL 162 80 82 0.4938272   21720000
```

```
# This chunk carries out the analysis between the team winning percentage and their teamsalary.

with(team.df.small,cor.test(teamSalary, winPerc))
```

```
##
##  Pearson's product-moment correlation
##
## data:  teamSalary and winPerc
## t = 6.6332, df = 916, p-value = 5.611e-11
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  0.1514790 0.2749794
## sample estimates:
##       cor
## 0.2140846
```

Results

- The Pearson's product-moment correlation test yields a t-value of 6.6332, which means there is sufficient evidence to reject the null hypothesis (which is that $\beta_{salary} = 0$), and due to the fact that the calculated correlation coefficient was found to be 0.2141, there is evidence of a positive correlation between a team's salary and winning percentage (i.e., $\beta_{salary} > 0$).

```
team.salary.df <- salary.df %>%
  group_by(teamID, yearID) %>%
  summarise(teamSalary = sum(salary, na.rm=TRUE))
```

```
## `summarise()` has grouped output by 'teamID'. You can override using the
## `.groups` argument.
```

```
league.salary.mean <- team.salary.df %>%
  group_by(yearID) %>%
  summarise(leagueAvg = mean(teamSalary, na.rm=TRUE))

league.salary.sd <- team.salary.df %>%
  group_by(yearID) %>%
  summarise(leagueSd = sd(teamSalary, na.rm=TRUE))




team.salary.df <- team.salary.df %>%
  left_join(league.salary.mean, by='yearID')
team.salary.df <- team.salary.df %>%
  left_join(league.salary.sd, by='yearID')

team.salary.df['normSalary'] <- (team.salary.df['teamSalary']-team.salary.df['leagueAvg'])/team.salary.

team.salary.df
```

```
## # A tibble: 918 x 6
## # Groups:   teamID [35]
##    teamID yearID teamSalary leagueAvg  leagueSd normSalary
##    <chr>   <int>      <int>     <dbl>     <dbl>      <dbl>
##  1 ANA      1997   31135472 40260210. 13060728.    -0.699
##  2 ANA      1998   41281000 42609429. 15380811.    -0.0864
##  3 ANA      1999   55388166 49807625  20561328.     0.271
##  4 ANA      2000   51464167 55537837. 21416220.    -0.190
##  5 ANA      2001   47535167 65355444. 24707706.    -0.721
##  6 ANA      2002   61721667 67469251. 24692193.    -0.233
##  7 ANA      2003   79031667 70942071. 28011963.     0.289
##  8 ANA      2004  100534667 69022198. 32824114.     0.960
##  9 ARI      1998   32347000 42609429. 15380811.    -0.667
```

```
## 10 ARI      1999   68703999 49807625  20561328.     0.919
## # i 908 more rows
```

```
team.df.small <- team.df.small %>%
  right_join(team.salary.df, by = c("teamID","yearID"))

with(team.df.small, cor.test(winPerc,normSalary))
```

```
##
##  Pearson's product-moment correlation
##
## data:  winPerc and normSalary
## t = 11.971, df = 916, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3104916 0.4224564
## sample estimates:
##       cor
## 0.3678063
```

Results

- After normalizing the team salaries with the mean and standard deviations of the overall team salaries for each year, the magnitude of the correlation coefficient increased by over 75%. This is likely due to how the salary has increased over time dramatically, so it is important to compare the teams relative to the teams in their year, rather than the salaries overall. To conclude, there is very clear evidence that spending more money relative to your opposition does have a positive correlation on winning percentage, but of course, this is contingent on spending money *well*, which is something this model cannot evaluate as of right now.