MSD Final Project Report

Forrest Hofmann (fhh2112) & Sagar Lal (sl3946)

2019-05-02 13:26:48

Contents

Introduction

Problem Description

Motivation

Data Source

Reproduction

Reproduction Code

```
teams <- read_csv(here('teams.csv'))
```

```
## Parsed with column specification:
## cols(
##   yearID = col_double(),
##   teamID = col_character(),
##   G = col_double(),
##   W = col_double(),
##   L = col_double()
## )
```

```
salaries <- read_csv(here('salaries.csv'))
```

```
## Parsed with column specification:
## cols(
```

```
##   yearID = col_double(),
##   teamID = col_character(),
##   playerID = col_character(),
##   salary = col_double()
## )
```

```
teams_old <- teams %>%
  filter(1985 <= yearID & yearID <= 1998) %>%
  mutate(winPercentage = W / (W + L) * 1000) %>%
  mutate(normalizedYear = yearID - 1985)

salaries_old <- salaries %>%
  filter(1985 <= yearID & yearID <= 1998) %>%
  mutate(salaryMil = salary / 1000000)
```

```
teams_old <- teams_old %>%
  inner_join(salaries_old) %>%
  group_by(yearID, teamID, G, W, L, winPercentage, normalizedYear) %>%
  summarize(totalSalaryMil = sum(salaryMil))
```

```
## Joining, by = c("yearID", "teamID")
```

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

```
## Joining, by = c("yearID", "teamID")
```

```
teams_old <- teams_old %>%
  inner_join(salaries_old) %>%
  group_by(yearID, teamID, G, W, L, winPercentage, totalSalaryMil, normalizedYear) %>%
  summarize(HHI = sum(salaryShareSquared))
```

```
## Joining, by = c("yearID", "teamID")
```

```
summary(teams_old$winPercentage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   327.2   456.8   498.4   500.0   543.2   703.7
```

```
summary(teams_old$totalSalaryMil)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.88   12.76   22.32   25.16   36.29   72.36
```

```
summary(teams_old$HHI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   427.5   668.6   756.3   815.6   879.1  5300.1
```

```
linear_fixed_old <- lm(formula = winPercentage ~ totalSalaryMil + HHI + normalizedYear + teamID + 0,
                       data = teams_old)
summary(linear_fixed_old)$coefficients
```

```
##                    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
## teamIDANA      520.3412853 46.2484875 11.250990 3.305855e-25
## teamIDARI      415.5845633 64.8210953  6.411255 4.778053e-10
## teamIDATL      500.2874262 20.6822732 24.189190 3.912949e-76
## teamIDBAL      474.6888319 20.3677945 23.305853 1.106243e-72
## teamIDBOS      500.3674770 20.4325763 24.488712 2.689059e-77
## teamIDCAL      478.6676384 20.6569410 23.172242 3.703357e-72
## teamIDCHA      497.3367589 20.2754263 24.529041 1.876375e-77
## teamIDCHN      472.8105792 20.2318161 23.369656 6.216065e-73
## teamIDCIN      504.8712038 19.9756579 25.274322 2.501974e-80
## teamIDCLE      487.3371161 19.2104121 25.368384 1.089483e-80
## teamIDCOL      477.3269106 28.3184575 16.855682 7.284719e-47
## teamIDDET      473.4187213 20.1658328 23.476279 2.374228e-73
## teamIDFLO      454.7556672 29.7189381 15.301881 1.175689e-40
## teamIDHOU      515.7181444 19.7505419 26.111595 1.581432e-83
## teamIDKCA      485.9458762 19.9615669 24.344075 9.787601e-77
## teamIDLAN      493.3318390 20.1372547 24.498466 2.464889e-77
## teamIDMIL      463.0607429 64.2852565  7.203218 3.755772e-12
## teamIDMIN      483.9572756 20.5503113 23.549876 1.222548e-73
## teamIDML4      494.3727955 20.2820858 24.374850 7.433918e-77
## teamIDMON      530.4524240 19.9905472 26.535163 3.921532e-85
## teamIDNYA      509.7065265 20.9672346 24.309669 1.331302e-76
## teamIDNYN      510.0199257 20.5600619 24.806342 1.587344e-78
## teamIDOAK      503.2701041 20.1869933 24.930414 5.270447e-79
## teamIDPHI      463.7557037 19.8160927 23.402984 4.600568e-73
## teamIDPIT      487.6110175 19.2593013 25.318209 1.697337e-80
## teamIDSDN      485.9750617 19.7978035 24.546918 1.599795e-77
## teamIDSEA      472.4450902 20.2362005 23.346531 7.659991e-73
## teamIDSFN      492.8906814 19.9322566 24.728293 3.178580e-78
## teamIDSLN      495.9950092 19.8301500 25.012166 2.551096e-79
## teamIDTBA      413.7955581 64.7841998  6.387291 5.497036e-10
## teamIDTEX      489.5752569 21.9602263 22.293726 1.084455e-68
## teamIDTOR      512.4375949 20.6188893 24.852822 1.050052e-78
```

```
linear_random_old <- lm(formula = winPercentage ~ totalSalaryMil + HHI + normalizedYear,
                        data = teams_old)
summary(linear_random_old)$coefficients
```

```
##                    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 +
                        data = teams_old)
summary(log_log_fixed_old)$coefficients
```

```
##                         Estimate  Std. Error   t value     Pr(>|t|)
## log(totalSalaryMil)  0.068481958 0.023048764  2.971177 3.175940e-03
## log(HHI)            -0.043092478 0.034083173 -1.264333 2.069690e-01
## normalizedYear      -0.006815679 0.003712079 -1.836081 6.721103e-02
## teamIDANA            6.381277477 0.258802544 24.656935 6.000401e-78
## teamIDARI            6.147474260 0.284859701 21.580709 7.375618e-66
## teamIDATL            6.341780647 0.254054450 24.962289 3.971429e-79
```

```
## teamIDBAL            6.304392674 0.252227238 24.994892 2.973619e-79
## teamIDBOS            6.356680411 0.254042578 25.022106 2.335784e-79
## teamIDCAL            6.296501097 0.251358491 25.049884 1.825754e-79
## teamIDCHA            6.353561607 0.253145391 25.098468 1.186854e-79
## teamIDCHN            6.295614193 0.255035277 24.685268 4.662152e-78
## teamIDCIN            6.363998559 0.252046869 25.249267 3.122677e-80
## teamIDCLE            6.314417826 0.244194538 25.858145 1.458689e-82
## teamIDCOL            6.302669708 0.251774555 25.032989 2.120858e-79
## teamIDDET            6.278537472 0.253687081 24.749142 2.640295e-78
## teamIDFLO            6.239856049 0.264167071 23.620870 6.447698e-74
## teamIDHOU            6.372617406 0.251976225 25.290550 2.167500e-80
## teamIDKCA            6.318245392 0.253724277 24.902014 6.782519e-79
## teamIDLAN            6.336534401 0.251504203 25.194547 5.067834e-80
## teamIDMIL            6.255310674 0.272629757 22.944343 2.918809e-71
## teamIDMIN            6.311256118 0.257376609 24.521483 2.007254e-77
## teamIDML4            6.331026869 0.253671900 24.957541 4.142388e-79
## teamIDMON            6.395786312 0.251422901 25.438360 5.873302e-81
## teamIDNYA            6.374791978 0.252496126 25.247088 3.183433e-80
## teamIDNYN            6.363423835 0.257124220 24.748442 2.656785e-78
## teamIDOAK            6.346043232 0.254205727 24.964202 3.904550e-79
## teamIDPHI            6.268077847 0.253347335 24.741045 2.837549e-78
## teamIDPIT            6.304910264 0.247423029 25.482310 3.985336e-81
## teamIDSDN            6.310660952 0.252946680 24.948582 4.485311e-79
## teamIDSEA            6.303619286 0.251664392 25.047720 1.861126e-79
## teamIDSFN            6.332517546 0.251659418 25.163046 6.698016e-80
## teamIDSLN            6.339364845 0.252096946 25.146536 7.752631e-80
## teamIDTBA            6.126803927 0.282074273 21.720534 2.046862e-66
## teamIDTEX            6.344583695 0.255152738 24.865826 9.354495e-79
## teamIDTOR            6.379838453 0.255083738 25.010761 2.583095e-79
```

```
log_log_random_old <- lm(formula = log(winPercentage) ~ log(totalSalaryMil) + log(HHI) + normalizedYear
                         data = teams_old)
summary(log_log_random_old)$coefficients
```

```
##                         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

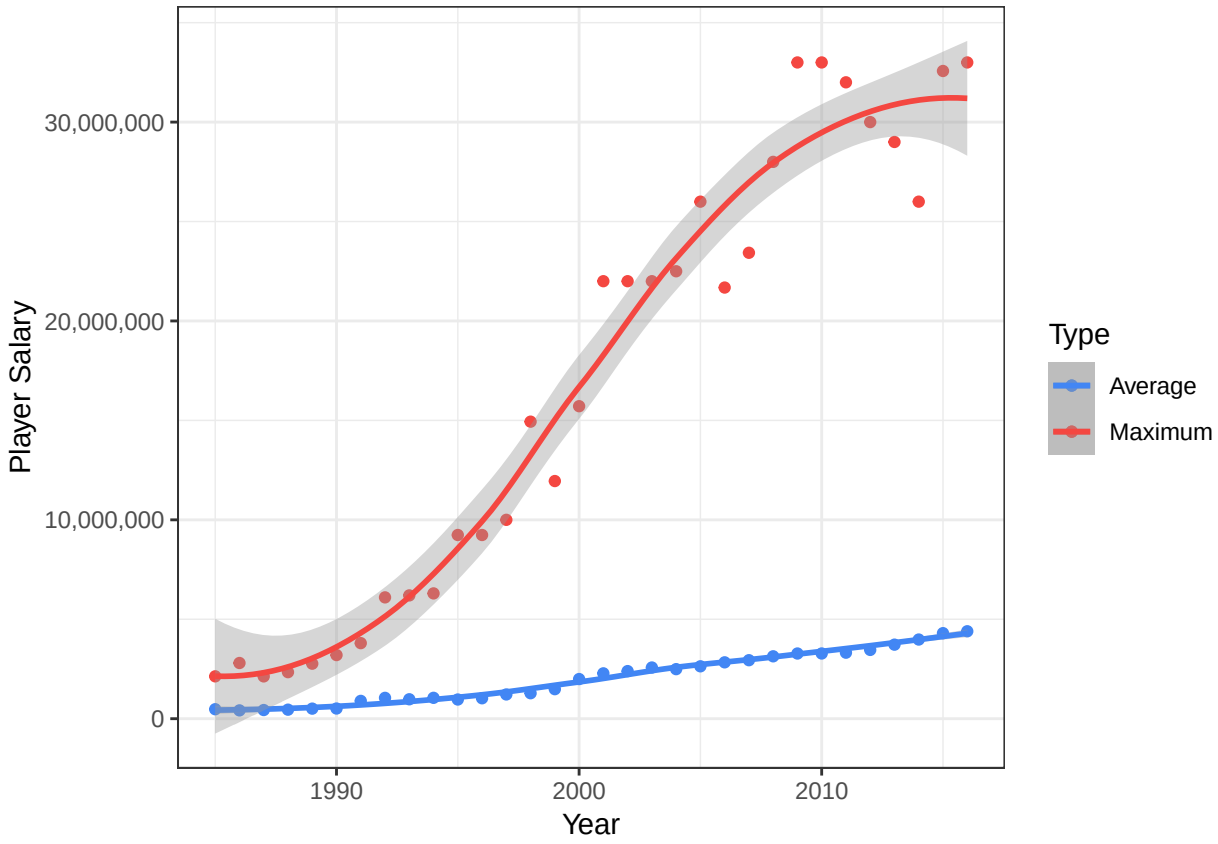
```
teams_new <- teams %>%
  filter(1999 <= yearID & yearID <= 2016) %>%
  mutate(winPercentage = W / (W + L) * 1000) %>%
  mutate(normalizedYear = yearID - 1985)

salaries_new <- salaries %>%
  filter(1999 <= yearID & yearID <= 2016) %>%
  mutate(salaryMil = salary / 1000000)
```

```
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_continuous(labels = comma) +
  labs(color = 'Type') +
  xlab('Year') +
  ylab('Player Salary')
```

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



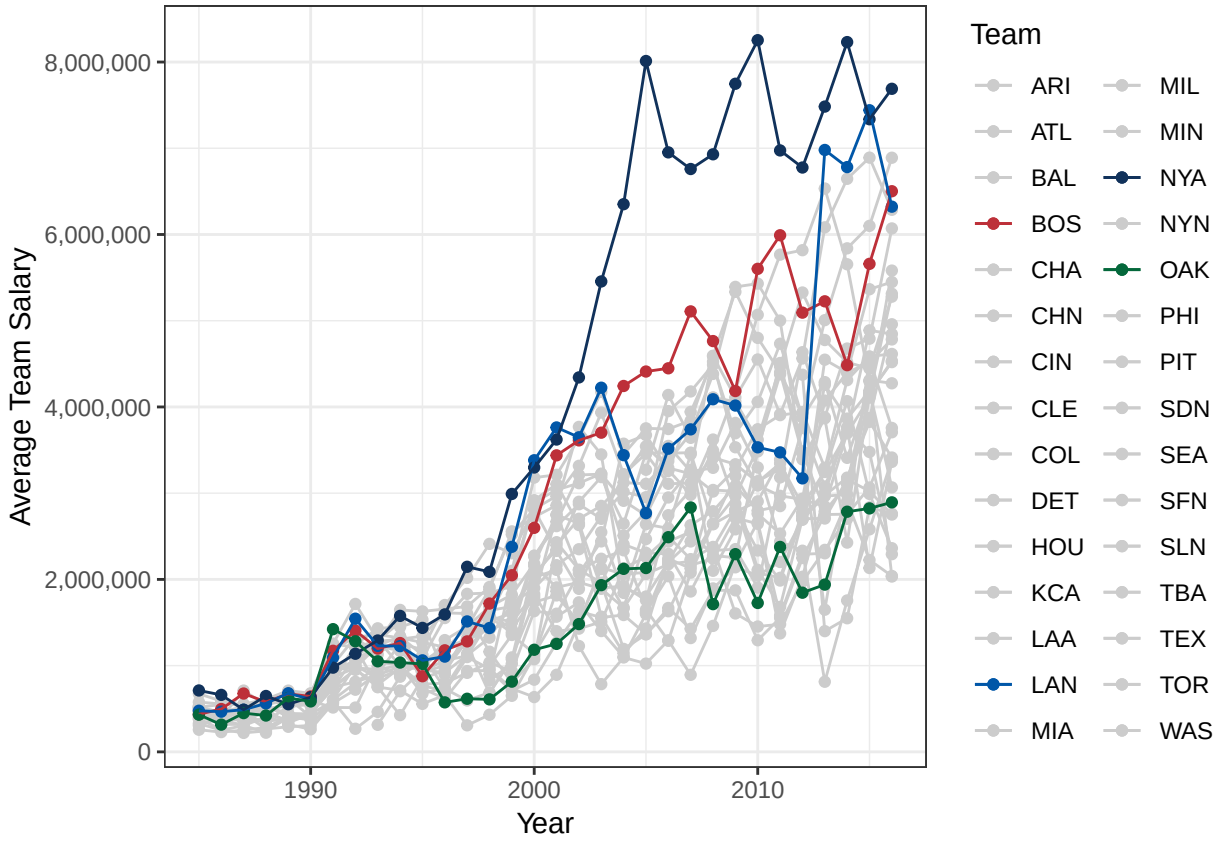

```
current_teamIDs <- c('ARI', 'ATL', 'BAL', 'BOS', 'CHA', 'CHN', 'CIN', 'CLE', 'COL', 'DET',
                     'HOU', 'KCA', 'LAA', 'LAN', 'MIA', 'MIL', 'MIN', 'NYA', 'NYN', 'OAK',
                     'PHI', 'PIT', 'SDN', 'SEA', 'SFN', 'SLN', 'TBA', 'TEX', 'TOR', 'WAS')
team_colors <- c('#cccccc',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




```
# TODO: plot distribution of win percentage, total salary, HHI
```

```
# TODO: plot distribution of win percentage, total salary, HHI
```

Extension Notes

Extension Analysis