# First Arb Salary Part 1

Allen Ho

# Introduction

In this task, I worked with a subset of batting-level data from players' career, platform year(the year before the year of their arbitration contract), py-1(the year before platform year) and py-2(two years before platform year). Detailed definitions of the variables can be found in the excel file within the same repository. On the basis of this data, the goal of this research was to develop a reliable framework which is capable of predicting a player's first-time eligible arbitration salary(salary_1te). The framework is basically composed of below parts: feature preprocessing, model building, hyperparameter tuning, model evaluation, and first-time eligible arbitration salary prediction.

```
#Main dataframe loading
df <-  read_excel("D:/First Arb Salary/First Arb Salary.xlsx")
#Check the structure of the dataframe
str(df)
```

```
## tibble [279 x 84] (S3: tbl_df/tbl/data.frame)
##  $ player_id       : num [1:279] 203390 985078 576755 232217 847127 ...
##  $ primary_position: chr [1:279] "4" "7" "3" "3" ...
##  $ age             : num [1:279] 30 28 28 30 29 28 27 29 29 29 ...
##  $ platform_year   : num [1:279] 2013 2014 2015 2019 2017 ...
##  $ mls             : num [1:279] 3.17 3.09 3.03 3.08 3.05 ...
##  $ salary_1te      : num [1:279] 849300 2912000 1815000 2626500 1351500 ...
##  $ salary_py       : num [1:279] 577680 570000 598080 663000 611280 ...
##  $ career_pa       : num [1:279] 611 2013 1159 1310 1020 ...
##  $ career_r        : num [1:279] 61 227 123 161 108 76 156 119 134 111 ...
##  $ career_h        : num [1:279] 147 448 299 293 213 195 327 343 217 211 ...
##  $ career_hr       : num [1:279] 6 36 39 63 20 12 28 27 36 11 ...
##  $ career_rbi      : num [1:279] 60 182 156 215 79 60 133 152 147 64 ...
##  $ career_tb       : num [1:279] 216 669 489 539 325 263 481 501 397 285 ...
##  $ career_sb       : num [1:279] 2 29 4 0 12 6 4 15 21 19 ...
##  $ career_avg      : num [1:279] 0.256 0.246 0.276 0.256 0.226 0.276 0.274 0.271 0.227 0.245 ...
##  $ career_obp      : num [1:279] 0.285 0.309 0.316 0.334 0.273 0.298 0.311 0.334 0.318 0.303 ...
##  $ career_slg      : num [1:279] 0.376 0.366 0.451 0.471 0.345 0.373 0.403 0.395 0.414 0.331 ...
##  $ career_ops      : num [1:279] 0.661 0.676 0.767 0.805 0.618 0.671 0.713 0.729 0.732 0.633 ...
##  $ career_war3     : num [1:279] -0.25 8.25 3.2 3.15 1 ...
##  $ car_opt         : num [1:279] 244 284 101 468 38 253 62 462 348 355 ...
##  $ car_out         : num [1:279] 0 0 0 0 0 0 0 0 0 146 ...
##  $ car_il          : num [1:279] 308 0 135 0 168 193 0 127 157 0 ...
##  $ car_mvpvotes    : num [1:279] 0 0 0 7 0 0 0 0 0 0 ...
##  $ car_ssvotes     : num [1:279] 0 0 0 0 0 0 0 0 0 0 ...
##  $ py_pa           : num [1:279] 147 542 186 369 178 550 363 288 285 511 ...
##  $ py_r            : num [1:279] 21 64 14 39 24 62 41 27 28 63 ...
##  $ py_h            : num [1:279] 37 123 42 74 42 160 80 64 44 120 ...
##  $ py_hr           : num [1:279] 2 14 5 12 6 12 12 7 8 5 ...
##  $ py_rbi          : num [1:279] 14 65 24 50 21 51 32 27 38 36 ...
##  $ py_tb           : num [1:279] 61 200 66 122 70 221 129 106 81 157 ...
##  $ py_sb           : num [1:279] 0 8 1 0 3 4 2 6 3 13 ...
##  $ py_avg          : num [1:279] 0.268 0.245 0.24 0.236 0.251 0.305 0.236 0.24 0.181 0.263 ...
##  $ py_obp          : num [1:279] 0.301 0.293 0.28 0.325 0.298 0.329 0.271 0.285 0.295 0.327 ...
##  $ py_slg          : num [1:279] 0.442 0.398 0.377 0.389 0.419 0.422 0.381 0.397 0.333 0.344 ...
##  $ py_ops          : num [1:279] 0.743 0.692 0.657 0.714 0.717 0.751 0.651 0.682 0.628 0.67 ...
##  $ py_war3         : num [1:279] 0.15 1.6 0 -0.3 0.3 2.4 -0.85 0.9 -0.7 2.05 ...
##  $ py_opt          : num [1:279] 0 0 0 0 0 0 16 0 40 0 ...
##  $ py_out          : num [1:279] 0 0 0 0 0 0 0 0 0 0 ...
##  $ py_il           : num [1:279] 103 0 105 0 96 0 0 86 0 0 ...
##  $ py_as           : chr [1:279] "N" "N" "N" "N" ...
##  $ py_mvp          : chr [1:279] "N" "N" "N" "N" ...
##  $ py_mvpvotes     : num [1:279] 0 0 0 0 0 0 0 0 0 0 ...
##  $ py_ss           : chr [1:279] "N" "N" "N" "N" ...
##  $ py_ssvotes      : num [1:279] 0 0 0 0 0 0 0 0 0 0 ...
##  $ py-1 pa         : num [1:279] 74 427 563 566 308 30 479 375 412 66 ...
##  $ py-1 r          : num [1:279] 5 40 55 80 26 0 62 34 58 6 ...
##  $ py-1 h          : num [1:279] 18 97 152 135 62 5 127 94 101 18 ...
##  $ py-1 hr         : num [1:279] 1 4 15 35 4 0 5 6 19 2 ...
##  $ py-1 rbi        : num [1:279] 15 31 68 108 20 0 41 45 65 6 ...
##  $ py-1 tb         : num [1:279] 25 131 241 265 85 7 168 123 192 28 ...
##  $ py-1 sb         : num [1:279] 0 2 3 0 5 0 1 6 5 1 ...
##  $ py-1 avg        : num [1:279] 0.257 0.253 0.288 0.274 0.218 0.185 0.286 0.281 0.272 0.295 ...
##  $ py-1 obp        : num [1:279] 0.284 0.319 0.321 0.352 0.265 0.241 0.323 0.341 0.34 0.333 ...
##  $ py-1 slg        : num [1:279] 0.357 0.341 0.457 0.539 0.299 0.259 0.378 0.368 0.516 0.459 ...
##  $ py-1 ops        : num [1:279] 0.641 0.66 0.779 0.89 0.564 0.501 0.701 0.71 0.856 0.792 ...
##  $ py-1 war        : num [1:279] -0.15 0.95 2 3.15 -0.05 ...
##  $ py-1 opt        : num [1:279] 0 29 0 0 0 79 0 0 0 0 ...
```

```
##  $ py-1 out        : num [1:279] 0 0 0 0 0 0 0 0 0 146 ...
##  $ py-1 il         : num [1:279] 0 0 15 0 72 10 0 41 41 0 ...
##  $ py-1 as         : chr [1:279] "N" "N" "N" "1" ...
##  $ py-1 mvp        : chr [1:279] "N" "N" "N" "16" ...
##  $ py-1 mvp votes  : num [1:279] 0 0 0 7 0 0 0 0 0 0 ...
##  $ py-1 ss         : chr [1:279] "N" "N" "N" "N" ...
##  $ py-1 ss votes   : num [1:279] 0 0 0 0 0 0 0 0 0 0 ...
##  $ py-2 pa         : num [1:279] 0 668 319 311 459 0 323 619 227 215 ...
##  $ py-2 r          : num [1:279] 0 84 46 40 49 0 39 47 23 31 ...
##  $ py-2 h          : num [1:279] 0 137 84 74 95 0 89 150 39 44 ...
##  $ py-2 hr         : num [1:279] 0 12 17 16 9 0 8 9 4 2 ...
##  $ py-2 rbi        : num [1:279] 0 50 51 52 34 0 46 62 22 10 ...
##  $ py-2 tb         : num [1:279] 0 199 149 141 151 0 133 216 57 57 ...
##  $ py-2 sb         : num [1:279] 0 13 0 0 4 0 1 3 7 3 ...
##  $ py-2 avg        : num [1:279] 0 0.226 0.284 0.265 0.226 0 0.298 0.273 0.197 0.222 ...
##  $ py-2 obp        : num [1:279] 0 0.294 0.335 0.331 0.275 0 0.338 0.348 0.3 0.274 ...
##  $ py-2 slg        : num [1:279] 0 0.328 0.503 0.505 0.359 0 0.445 0.393 0.288 0.288 ...
##  $ py-2 ops        : num [1:279] 0 0.622 0.839 0.837 0.634 0 0.782 0.741 0.587 0.562 ...
##  $ py-2 war        : chr [1:279] "NA" "2.35" "1.4" "1.1000000000000001" ...
##  $ py-2 opt        : num [1:279] 0 0 0 0 0 0 0 0 0 0 ...
##  $ py-2 out        : num [1:279] 0 0 0 0 0 0 0 0 0 0 ...
##  $ py-2 il         : num [1:279] 0 0 15 0 0 183 0 0 116 0 ...
##  $ py-2 as         : chr [1:279] "N" "N" "N" "N" ...
##  $ py-2 mvp        : chr [1:279] "N" "N" "N" "N" ...
##  $ py-2 mvp votes  : num [1:279] 0 0 0 0 0 0 0 0 0 0 ...
##  $ py-2 ss         : chr [1:279] "N" "N" "N" "N" ...
##  $ py-2 ss votes   : num [1:279] 0 0 0 0 0 0 0 0 0 0 ...
```

```
#Check the summary of the dataframe
summary(df)
```

```
##    player_id      primary_position        age        platform_year 
##  Min.   :100865   Length:279         Min.   :24.00   Min.   :2010  
##  1st Qu.:301704   Class :character   1st Qu.:27.00   1st Qu.:2012  
##  Median :516269   Mode  :character   Median :28.00   Median :2015  
##  Mean   :528612                      Mean   :28.52   Mean   :2015  
##  3rd Qu.:777214                      3rd Qu.:30.00   3rd Qu.:2017  
##  Max.   :995375                      Max.   :35.00   Max.   :2019  
##       mls          salary_1te         salary_py         career_pa   
##  Min.   :1.144   Min.   :  680400   Min.   : 497760   Min.   : 264  
##  1st Qu.:2.167   1st Qu.: 1433400   1st Qu.: 577590   1st Qu.: 976  
##  Median :3.045   Median : 2517500   Median : 592360   Median :1318  
##  Mean   :2.812   Mean   : 2687274   Mean   : 638061   Mean   :1304  
##  3rd Qu.:3.089   3rd Qu.: 3336250   3rd Qu.: 612301   3rd Qu.:1612  
##  Max.   :3.170   Max.   :11730000   Max.   :4200000   Max.   :2590  
##     career_r        career_h       career_hr        career_rbi   
##  Min.   : 18.0   Min.   : 50.0   Min.   :  0.00   Min.   : 20.0  
##  1st Qu.:111.0   1st Qu.:224.0   1st Qu.: 17.00   1st Qu.: 90.0  
##  Median :152.0   Median :303.0   Median : 32.00   Median :137.0  
##  Mean   :155.8   Mean   :303.2   Mean   : 36.15   Mean   :142.1  
##  3rd Qu.:200.0   3rd Qu.:385.5   3rd Qu.: 50.50   3rd Qu.:186.0  
##  Max.   :377.0   Max.   :665.0   Max.   :117.00   Max.   :313.0  
##    career_tb        career_sb        career_avg       career_obp    
##  Min.   :  66.0   Min.   :  0.00   Min.   :0.1830   Min.   :0.2320  
##  1st Qu.: 343.5   1st Qu.:  5.00   1st Qu.:0.2430   1st Qu.:0.3035  
##  Median : 490.0   Median : 14.00   Median :0.2560   Median :0.3180  
##  Mean   : 488.0   Mean   : 24.06   Mean   :0.2557   Mean   :0.3198  
##  3rd Qu.: 626.5   3rd Qu.: 30.50   3rd Qu.:0.2710   3rd Qu.:0.3365  
##  Max.   :1123.0   Max.   :184.00   Max.   :0.3140   Max.   :0.3940  
##    career_slg       career_ops      career_war3        car_opt     
##  Min.   :0.2650   Min.   :0.4970   Min.   :-1.750   Min.   :  0.0  
##  1st Qu.:0.3685   1st Qu.:0.6770   1st Qu.: 2.050   1st Qu.: 47.5  
##  Median :0.4100   Median :0.7310   Median : 3.850   Median :152.0  
##  Mean   :0.4083   Mean   :0.7281   Mean   : 4.940   Mean   :170.1  
##  3rd Qu.:0.4425   3rd Qu.:0.7680   3rd Qu.: 6.875   3rd Qu.:264.5  
##  Max.   :0.5600   Max.   :0.9540   Max.   :22.700   Max.   :598.0  
##     car_out           car_il        car_mvpvotes     car_ssvotes    
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :0.00000  
##  1st Qu.:  0.00   1st Qu.: 15.50   1st Qu.:  0.00   1st Qu.:0.00000  
##  Median :  0.00   Median : 50.00   Median :  0.00   Median :0.00000  
##  Mean   : 22.41   Mean   : 67.78   Mean   : 15.62   Mean   :0.05735  
##  3rd Qu.:  0.00   3rd Qu.:102.50   3rd Qu.:  0.00   3rd Qu.:0.00000  
##  Max.   :858.00   Max.   :357.00   Max.   :581.00   Max.   :2.00000  
##      py_pa            py_r             py_h             py_hr      
##  Min.   : 16.0   Min.   :  0.00   Min.   :  2.00   Min.   : 0.00  
##  1st Qu.:298.0   1st Qu.: 33.00   1st Qu.: 64.00   1st Qu.: 5.00  
##  Median :429.0   Median : 48.00   Median : 96.00   Median :11.00  
```

```
##  Mean   :420.1   Mean   : 51.09   Mean   : 97.55   Mean   :12.41
##  3rd Qu.:562.5   3rd Qu.: 66.50   3rd Qu.:132.00   3rd Qu.:17.00
##  Max.   :745.0   Max.   :129.00   Max.   :192.00   Max.   :47.00
##      py_rbi          py_tb          py_sb            py_avg
##  Min.   :  0.00   Min.   :  2.0   Min.   : 0.000   Min.   :0.1360
##  1st Qu.: 28.00   1st Qu.:101.0   1st Qu.: 1.000   1st Qu.:0.2345
##  Median : 44.00   Median :150.0   Median : 4.000   Median :0.2550
##  Mean   : 46.66   Mean   :159.1   Mean   : 7.728   Mean   :0.2546
##  3rd Qu.: 61.50   3rd Qu.:210.5   3rd Qu.:10.000   3rd Qu.:0.2770
##  Max.   :130.00   Max.   :354.0   Max.   :64.000   Max.   :0.4800
##      py_obp          py_slg          py_ops          py_war3
##  Min.   :0.1830   Min.   :0.154   Min.   :0.3780   Min.   :-1.200
##  1st Qu.:0.2980   1st Qu.:0.370   1st Qu.:0.6750   1st Qu.: 0.525
##  Median :0.3230   Median :0.412   Median :0.7330   Median : 1.250
##  Mean   :0.3217   Mean   :0.412   Mean   :0.7337   Mean   : 1.728
##  3rd Qu.:0.3470   3rd Qu.:0.459   3rd Qu.:0.7970   3rd Qu.: 2.600
##  Max.   :0.5520   Max.   :0.680   Max.   :1.2320   Max.   : 8.850
##      py_opt           py_out            py_il           py_as
##  Min.   : 0.000   Min.   : 0.0000   Min.   :  0.00   Length:279
##  1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.:  0.00   Class :character
##  Median : 0.000   Median : 0.0000   Median :  0.00   Mode  :character
##  Mean   : 4.505   Mean   : 0.2437   Mean   : 21.09
##  3rd Qu.: 0.000   3rd Qu.: 0.0000   3rd Qu.: 32.00
##  Max.   :91.000   Max.   :68.0000   Max.   :159.00
##     py_mvp           py_mvpvotes        py_ss             py_ssvotes
##  Length:279         Min.   :  0.00   Length:279         Min.   :0.00000
##  Class :character   1st Qu.:  0.00   Class :character   1st Qu.:0.00000
##  Mode  :character   Median :  0.00   Mode  :character   Median :0.00000
##                     Mean   :  7.33                      Mean   :0.02867
##                     3rd Qu.:  0.00                      3rd Qu.:0.00000
##                     Max.   :422.00                      Max.   :1.00000
##     py-1 pa          py-1 r           py-1 h          py-1 hr
##  Min.   :  2.0   Min.   :  0.00   Min.   :  0.00   Min.   : 0.00
##  1st Qu.:243.5   1st Qu.: 27.00   1st Qu.: 52.50   1st Qu.: 4.00
##  Median :430.0   Median : 47.00   Median : 97.00   Median :10.00
##  Mean   :389.0   Mean   : 46.89   Mean   : 91.14   Mean   :11.32
##  3rd Qu.:538.5   3rd Qu.: 66.50   3rd Qu.:129.00   3rd Qu.:17.00
##  Max.   :730.0   Max.   :122.00   Max.   :214.00   Max.   :40.00
##     py-1 rbi         py-1 tb         py-1 sb          py-1 avg
##  Min.   :  0.00   Min.   :  0.0   Min.   : 0.000   Min.   :0.0000
##  1st Qu.: 23.00   1st Qu.: 80.5   1st Qu.: 1.000   1st Qu.:0.2345
##  Median : 42.00   Median :154.0   Median : 3.000   Median :0.2550
##  Mean   : 43.57   Mean   :147.7   Mean   : 7.444   Mean   :0.2518
##  3rd Qu.: 62.00   3rd Qu.:209.0   3rd Qu.:10.000   3rd Qu.:0.2750
##  Max.   :113.00   Max.   :359.0   Max.   :70.000   Max.   :0.4000
##     py-1 obp         py-1 slg         py-1 ops         py-1 war
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :-1.700
##  1st Qu.:0.2995   1st Qu.:0.3630   1st Qu.:0.6640   1st Qu.: 0.350
##  Median :0.3190   Median :0.4060   Median :0.7300   Median : 1.400
##  Mean   :0.3178   Mean   :0.4044   Mean   :0.7222   Mean   : 1.591
##  3rd Qu.:0.3430   3rd Qu.:0.4570   3rd Qu.:0.7920   3rd Qu.: 2.475
##  Max.   :0.5000   Max.   :0.6670   Max.   :1.1280   Max.   : 8.900
##     py-1 opt         py-1 out          py-1 il          py-1 as
##  Min.   :  0.00   Min.   :  0.000   Min.   :  0.00   Length:279
##  1st Qu.:  0.00   1st Qu.:  0.000   1st Qu.:  0.00   Class :character
##  Median :  0.00   Median :  0.000   Median :  0.00   Mode  :character
##  Mean   : 14.42   Mean   :  1.986   Mean   : 17.84
##  3rd Qu.:  0.00   3rd Qu.:  0.000   3rd Qu.: 24.00
##  Max.   :178.00   Max.   :152.000   Max.   :166.00
##    py-1 mvp          py-1 mvp votes      py-1 ss          py-1 ss votes
##  Length:279         Min.   :  0.000   Length:279         Min.   :0.00000
##  Class :character   1st Qu.:  0.000   Class :character   1st Qu.:0.00000
##  Mode  :character   Median :  0.000   Mode  :character   Median :0.00000
##                     Mean   :  5.556                      Mean   :0.01792
##                     3rd Qu.:  0.000                      3rd Qu.:0.00000
##                     Max.   :415.000                      Max.   :1.00000
##     py-2 pa          py-2 r           py-2 h          py-2 hr
##  Min.   :  0.0   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:163.0   1st Qu.: 16.00   1st Qu.: 36.00   1st Qu.: 2.000
##  Median :319.0   Median : 36.00   Median : 74.00   Median : 7.000
##  Mean   :322.1   Mean   : 37.98   Mean   : 75.31   Mean   : 8.566
##  3rd Qu.:478.5   3rd Qu.: 57.50   3rd Qu.:112.00   3rd Qu.:13.000
##  Max.   :710.0   Max.   :128.00   Max.   :193.00   Max.   :52.000
##     py-2 rbi         py-2 tb         py-2 sb          py-2 avg
##  Min.   :  0.00   Min.   :  0.0   Min.   : 0.000   Min.   :0.0000
##  1st Qu.: 13.00   1st Qu.: 53.0   1st Qu.: 0.000   1st Qu.:0.2255
##  Median : 31.00   Median :115.0   Median : 3.000   Median :0.2550
##  Mean   : 34.55   Mean   :120.1   Mean   : 5.943   Mean   :0.2389
##  3rd Qu.: 50.00   3rd Qu.:178.5   3rd Qu.: 7.000   3rd Qu.:0.2745
```

```
##  Max.   :114.00   Max.   :340.0   Max.   :56.000   Max.   :0.3420  
##     py-2 obp         py-2 slg         py-2 ops        py-2 war        
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Length:279        
##  1st Qu.:0.2850   1st Qu.:0.3320   1st Qu.:0.6260   Class :character  
##  Median :0.3150   Median :0.3950   Median :0.7130   Mode  :character  
##  Mean   :0.2987   Mean   :0.3754   Mean   :0.6741                     
##  3rd Qu.:0.3385   3rd Qu.:0.4465   3rd Qu.:0.7700                     
##  Max.   :0.4220   Max.   :0.6270   Max.   :1.0490                     
##     py-2 opt         py-2 out          py-2 il          py-2 as         
##  Min.   :  0.00   Min.   :  0.000   Min.   :  0.00   Length:279        
##  1st Qu.:  0.00   1st Qu.:  0.000   1st Qu.:  0.00   Class :character  
##  Median :  0.00   Median :  0.000   Median :  0.00   Mode  :character  
##  Mean   : 31.72   Mean   :  2.756   Mean   : 19.74                     
##  3rd Qu.: 55.50   3rd Qu.:  0.000   3rd Qu.: 25.50                     
##  Max.   :171.00   Max.   :152.000   Max.   :183.00                     
##    py-2 mvp         py-2 mvp votes      py-2 ss          py-2 ss votes    
##  Length:279         Min.   :  0.000   Length:279         Min.   :0.00000  
##  Class :character   1st Qu.:  0.000   Class :character   1st Qu.:0.00000  
##  Mode  :character   Median :  0.000   Mode  :character   Median :0.00000  
##                     Mean   :  2.728                      Mean   :0.01075  
##                     3rd Qu.:  0.000                      3rd Qu.:0.00000  
##                     Max.   :279.000                      Max.   :1.00000
```

# Data Preprocessing

As seen from the summary above, it seems like there are some missing value in the column **py-2 war**.

```
#Change the names of the columns for easier data manipulation
names(df) <- gsub(" ", "", names(df))
names(df) <- gsub("-", "", names(df))
names(df) <- gsub("_", "", names(df))
#Convert the column of interest, py2war, into numeric type
df$py2war <- as.numeric(as.character(df$py2war))
#Take a look at the rows with missing values in py2war
df %>%
  filter(is.na(py2war))
```

```
## # A tibble: 12 x 84
##    playerid primaryposition   age platformyear   mls salary1te salarypy careerpa
##       <dbl> <chr>           <dbl>        <dbl> <dbl>     <dbl>    <dbl>    <dbl>
##  1   203390 4                  30         2013  3.17    849300  577680       611
##  2   709671 4                  28         2019  3.08   1683000  601120       742
##  3   359676 6                  28         2016  2.13    955800  565950       446
##  4   669563 3                  27         2018  3.05   1248000  616920       659
##  5   397043 2                  27         2013  3.10   2793000  576056      1003
##  6   488655 8                  27         2019  2.16   2524500  582400       921
##  7   175838 7                  31         2015  3.07   1303500  577696       729
##  8   305267 9                  31         2013  3.07   1510500  571300       874
##  9   768972 4                  29         2012  3.11   3393000  604391.     1742
## 10   370455 2                  32         2014  3.08   1344000  577296       673
## 11   834439 8                  34         2010  3.12   2640000  519720      1025
## 12   544603 2                  28         2017  3.03   1537000  605880       730
## # ... with 76 more variables: careerr <dbl>, careerh <dbl>, careerhr <dbl>,
## #   careerrbi <dbl>, careertb <dbl>, careersb <dbl>, careeravg <dbl>,
## #   careerobp <dbl>, careerslg <dbl>, careerops <dbl>, careerwar3 <dbl>,
## #   caropt <dbl>, carout <dbl>, caril <dbl>, carmvpvotes <dbl>,
## #   carssvotes <dbl>, pypa <dbl>, pyr <dbl>, pyh <dbl>, pyhr <dbl>,
## #   pyrbi <dbl>, pytb <dbl>, pysb <dbl>, pyavg <dbl>, pyobp <dbl>, pyslg <dbl>,
## #   pyops <dbl>, pywar3 <dbl>, pyopt <dbl>, pyout <dbl>, pyil <dbl>, ...
```

Judging from the more traditional stats and those votings, it does not seem like those with missing values in **py2war** are players that stand out either way. Thus I'll remove these instances from the dataset. I'll also convert some columns into character types as they should be.

```
#Remove instances with missing values in py2war
df <- df %>%
  filter(!is.na(py2war))

#Convert some columns into character type
i <- c('playerid', 'primaryposition', 'platformyear', 'pyas', 'pymvp', 'pyss', 'py1as', 'py1mvp', 'py1ss', 'py2as
', 'py2mvp', 'py2ss')
df[ , i] <- apply(df[ , i], 2, function(x) as.character(x))
```

# Stepwise Linear Regression Model

Since there are only 267 instances(after data cleaning) in this dataset, I'll focus on linear regression model to predict the first year arbitration salary first. I'll use stepwise feature selection since 80 ish variables are probably just way too many. As from below, a simple linear regression model with stepwise feature selection already gave me **0.98 $R^2$** and **2.041e+05 RMSE**. Not bad I would say. Also from the below table, we can see that the linear regression model focuses a lot on performance in platform year, which is not surprising from a baseball standpoint.

```
#define intercept-only model
intercept_only <- lm(salary1te ~ 1, df)

#define model with all predictors
all <- lm(salary1te ~ . - salary1te - playerid, df)

#perform backward stepwise regression
both <- step(intercept_only, direction='both', scope=formula(all), trace=0)

data.frame(performance(both)) %>%
  kbl() %>%
  kable_classic(full_width = F, html_font = "Cambria")
```

| AIC | BIC | R2 | R2_adjusted | RMSE | Sigma |
|---|---|---|---|---|---|
| 7404.561 | 7616.209 | 0.9857561 | 0.9818714 | 204088.1 | 230674.8 |

```
data.frame(Variable = row.names(anova(both)),
           pvalue = anova(both)$Pr) %>%
  arrange(pvalue) %>%
  kbl() %>%
  kable_classic(full_width = F, html_font = "Cambria")
```

| Variable | pvalue |
|---|---|
| careertb | 0.0000000 |
| py2mvp | 0.0000000 |
| pywar3 | 0.0000000 |
| pymvp | 0.0000000 |
| pyhr | 0.0000000 |
| salarypy | 0.0000000 |
| py1war | 0.0000000 |
| pyrbi | 0.0000000 |
| primaryposition | 0.0000000 |
| py1rbi | 0.0000004 |
| py2sb | 0.0000018 |
| pyslg | 0.0000686 |
| pyil | 0.0002373 |
| py2hr | 0.0016714 |
| pyobp | 0.0020088 |
| carmvpvotes | 0.0059179 |
| pyh | 0.0128064 |
| py1mvp | 0.0128546 |
| pypa | 0.0450678 |
| pyas | 0.0475308 |
| py1as | 0.0478564 |
| py1pa | 0.1036846 |
| py1h | 0.1116442 |
| py1sb | 0.1373192 |
| careerslg | 0.1392772 |
| pymvpvotes | 0.1902540 |
| Residuals | NA |

Now that we now how we should set our baseline expectations for this prediction, I will move on to more potent models like KNN, random forest and lightgbm models. The details of these models can be found in the Jupyter Notebook file. From RMSE, it does not seem like more complex models are helping us with this project though.