Sabermetrics

Alex Bank

3/30/2018

Lecture 1

We can start off this brief examination of Sabermetrics by looking at the fields contained in the "Salaries" table.

```
dbListFields(conn, 'Salaries')
```

```
## [1] "yearID"   "teamID"   "lgID"     "playerID" "salary"
```

From this table, we can extract a dataframe with year, player salary, and league information. We can then get the number of observations in this table, the range of years available, and the different leagues using R.

```
df <- dbGetQuery(conn, '
                 SELECT yearID AS year, salary, lgID AS league
                 FROM Salaries')
nrow(df)
```

```
## [1] 26428
```

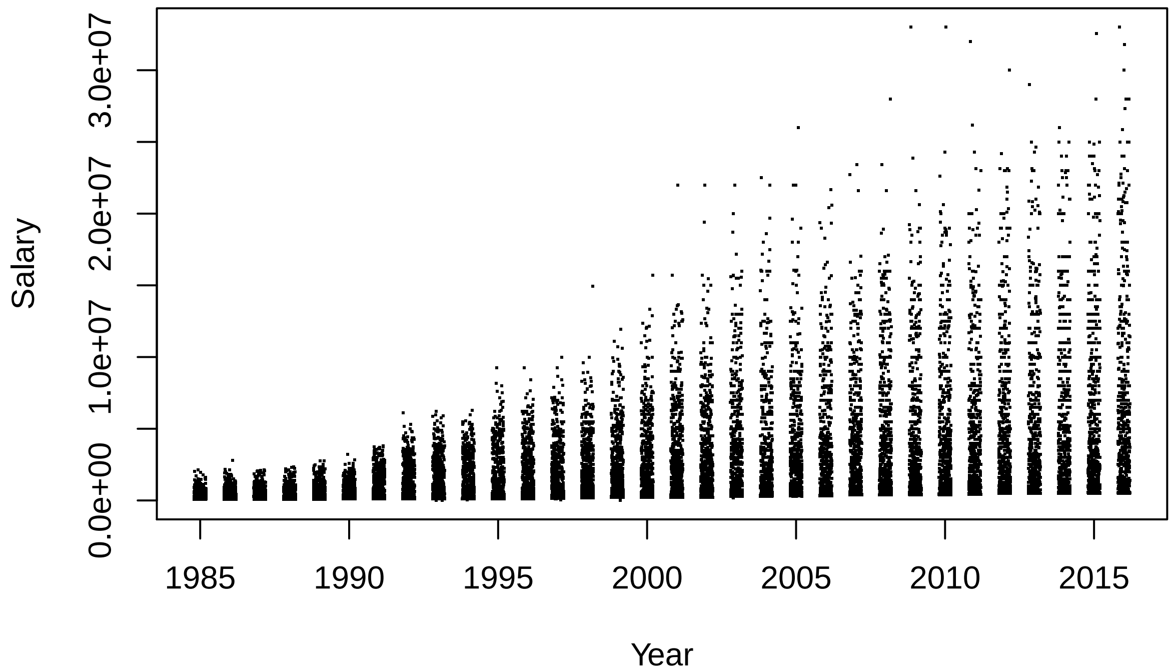
```
range(df$year)
```

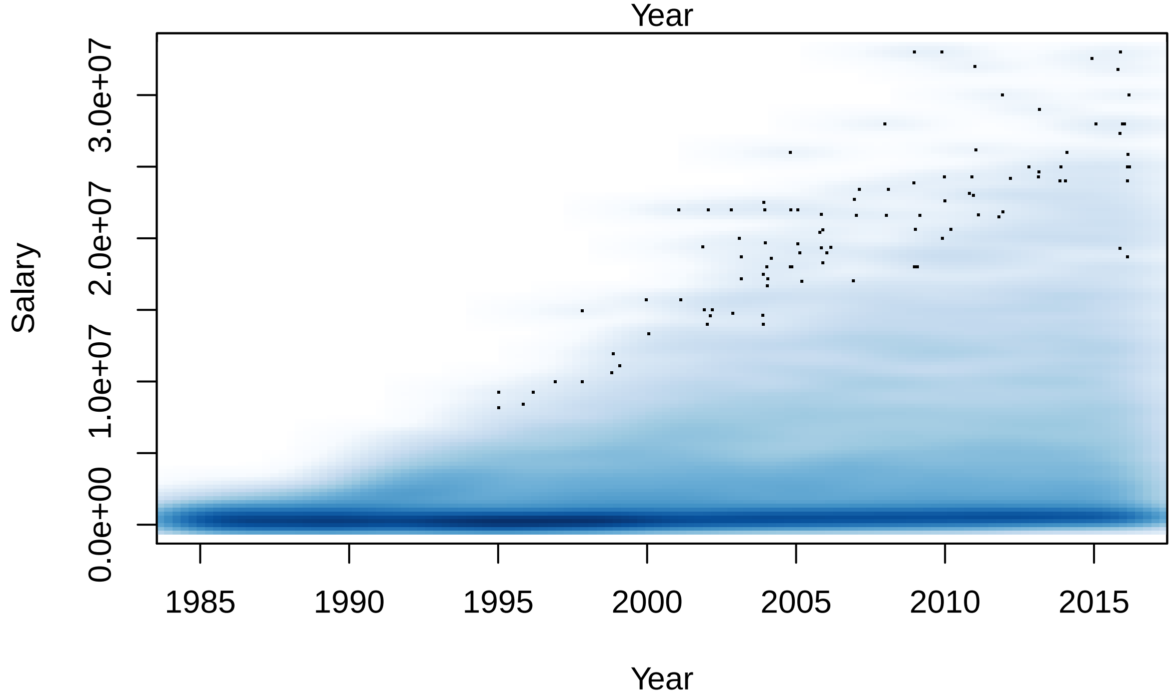
```
## [1] 1985 2016
```

```
unique(df$league)
```

```
## [1] "NL" "AL"
```

We can see that this is a massive set of observations dating back to 1985. We can visualize some of the trends in the data by plotting year vs. salary. The following graph makes it pretty evident that salaries have increased over the years, and have spread out. Below that graph is a nice, smooth representation of the same data.





Now we can use R to fit a linear model to the data, using year and league as predictors. Looking below at the summary of the model, the non-symetric residuals imply that the model predicts salaries far from the actual salaries, which makes sense because there is a huge range of salaries, and we saw that recently salaries have been more spread out than in the 80s. Moving on to the coefficients, it seems that the model is actually capturing a trend; the high t values for the model and in relation to year indicate that there is a true relationship, and that we can reject the null hypothesis. However, the t value for league is much smaller, indicating that the linear fit there may be a product of error. Indeed, the large standard error values with respect to the t values calls into question the validity of all the trends. The multiple R-squared value is only 0.12, so only 12% of the variation in the data is explained by year and league. This is not a very high value, and reconfirms our previous suspicions regarding the validity of these trends (particularly with respect to league since the t value is much smaller).

```
##
## Call:
## lm(formula = salary ~ year + league, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3731002 -1830699  -720059   570945 29718662
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -271424769    4469223  -60.73  < 2e-16 ***
## year            136738       2234   61.21  < 2e-16 ***
## leagueNL       -167212      39812   -4.20 2.68e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3234000 on 26425 degrees of freedom
## Multiple R-squared:  0.1243, Adjusted R-squared:  0.1242
## F-statistic:  1876 on 2 and 26425 DF,  p-value: < 2.2e-16
```

Looking at a logarithmic model, there is a very different story. The residuals are symetrically distributed, the t values are high relative to the standard error, and the multiple R-squared value increased to 0.21. All of these metrics point to a much better model for the data, meaning that salaries have closer to exponential growth with the years. Again, the t value for league is much smaller than that for year, meaing that the relationship between league and salary is much smaller than that of year and salary.

```
##
## Call:
## lm(formula = log(salary) ~ year + league, data = cleandf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7299 -1.1816 -0.2381  1.0500  3.3054
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.301e+02  1.709e+00 -76.138  < 2e-16 ***
## year         7.184e-02  8.544e-04  84.079  < 2e-16 ***
## leagueNL    -4.953e-02  1.523e-02  -3.253  0.00115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 26423 degrees of freedom
## Multiple R-squared:  0.2111, Adjusted R-squared:  0.211
## F-statistic:  3535 on 2 and 26423 DF,  p-value: < 2.2e-16
```

Lecture 2

```
##   team total_salary
## 1  NYY    222997792
```

```
##    team total_salary
## 30  TBR     57097310
```

In 2016, the New York Yankees had the highest team salary, while the Tampa Bay Rays had the lowest team salary. We can see the same data again but with each team's full name by joining this table with the "Teams" table (only the head of this table is included).

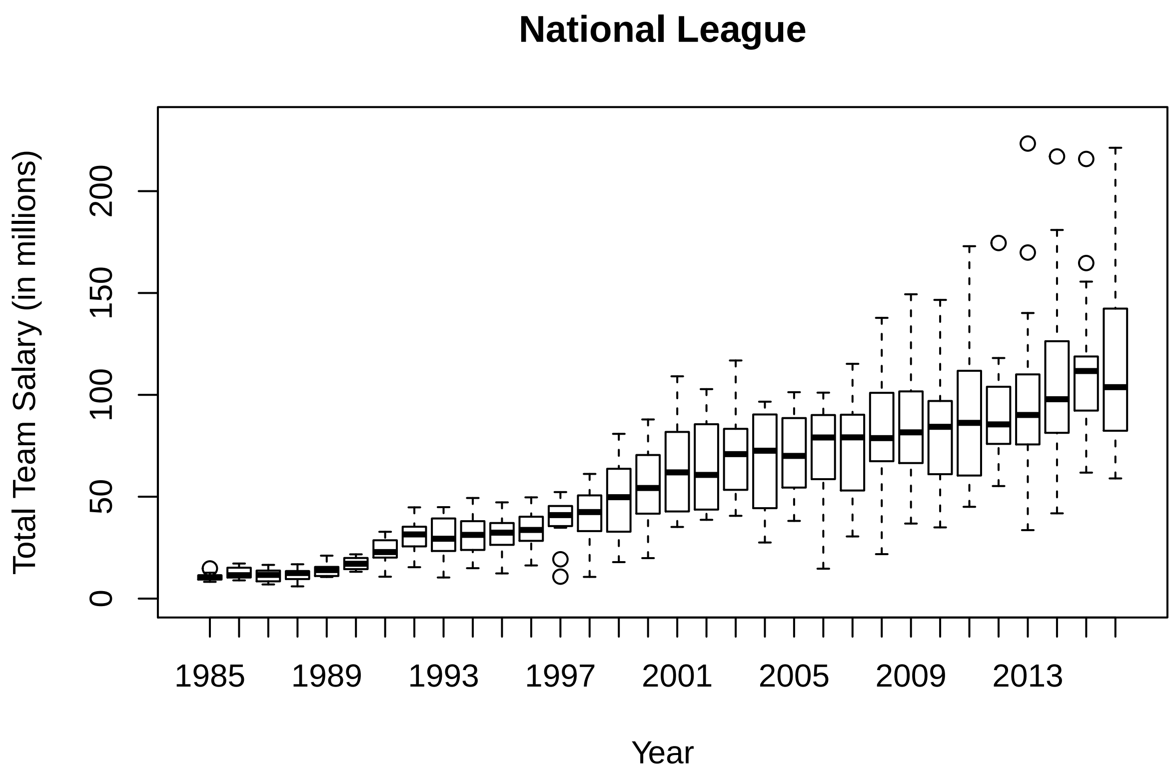
```
##                    team total_salary
## 1     New York Yankees    222997792
## 2  Los Angeles Dodgers    221288380
## 3       Detroit Tigers    194876481
## 4       Boston Red Sox    188545761
## 5        Texas Rangers    176038723
## 6 San Francisco Giants    172253778
```

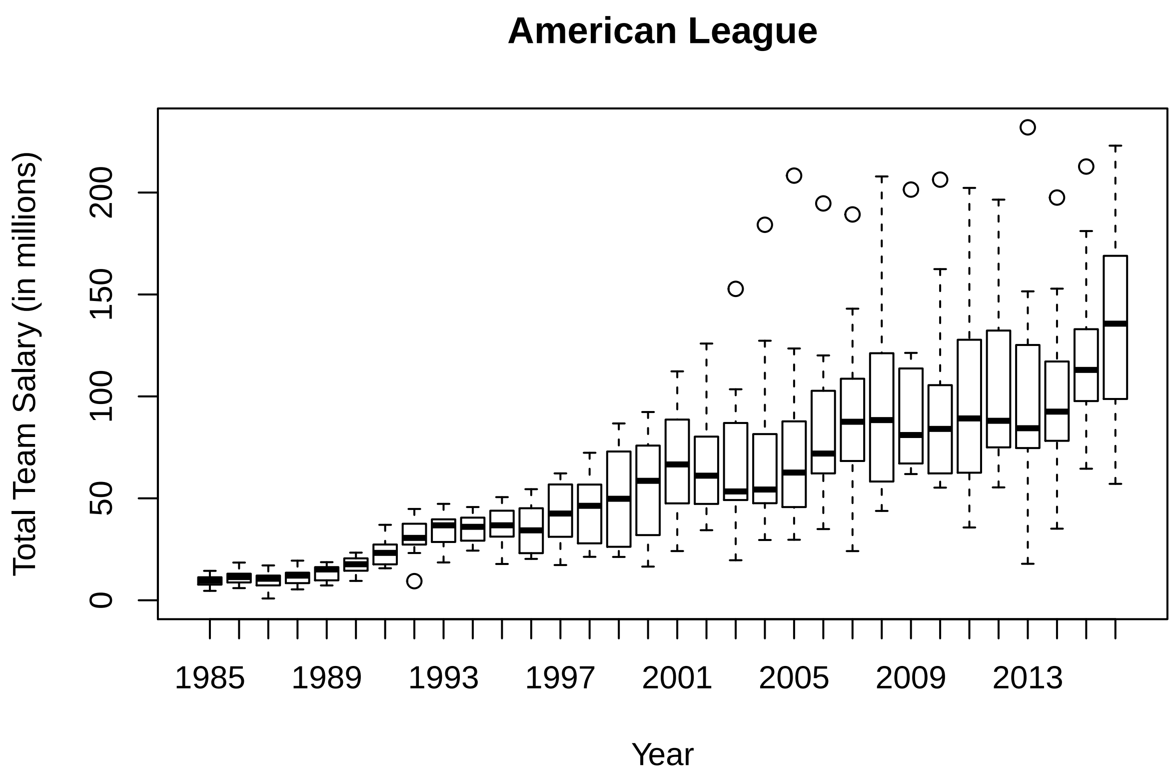
This next data frame contains the total salary and league for every unique combination of year and team. It has 918 observations.

```
## Rows of observations: 918
```

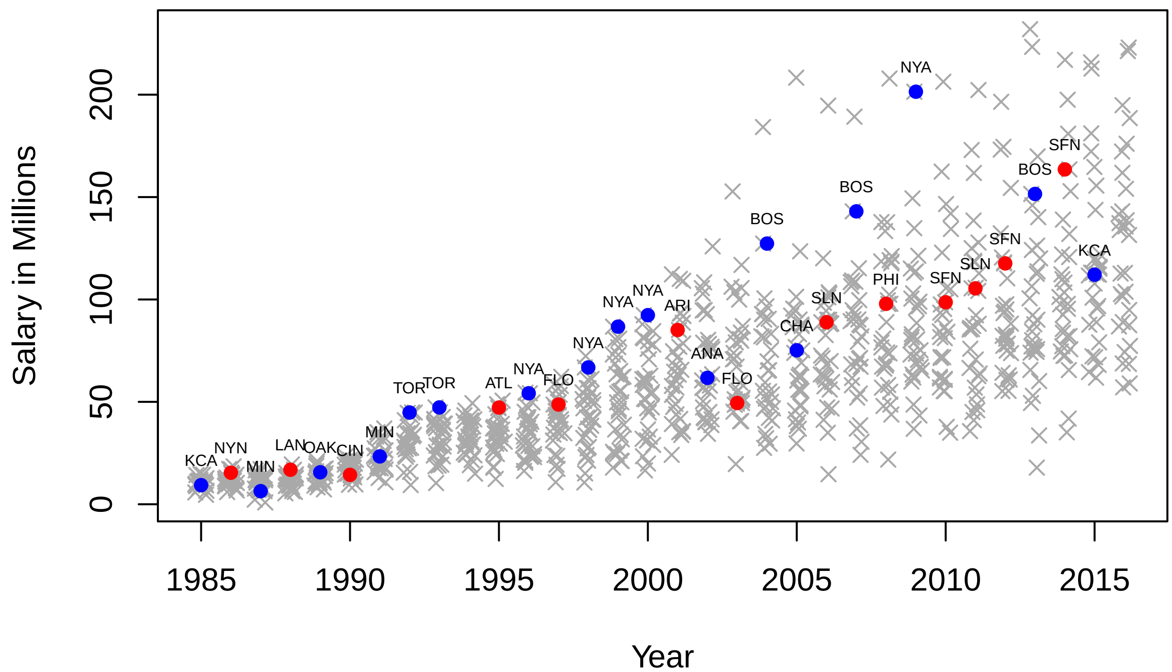
```
## The mean salary (in millions) for the NL in 2016: 115.600044733333
```

```
## The mean salary (in millions) for the AL in 2016: 134.409114733333
```





Interestingly, the barplots show that the American League tends to have a greater spread of team salaries, but both leagues historically have approximately the same median team salary.



There was a period from about 1993 until 2000 where the teams that spent the most won the World Series. Post-2000, team salaries became more spread out. Only Boston and New York have won the World Series in the past 18 years paying a team significantly more than other teams. All the other World Series winners from that period come from the upper-middle salary range.

Lecture 3

```
number21 <- dbGetQuery(conn, "
                       SELECT year, MAX(total_salary) AS max_salary
                       FROM (
                              SELECT yearID AS year,
                                     teamID AS team,
                                     SUM(salary) AS total_salary,
                                     lgID
                              FROM Salaries
                              GROUP BY year, team
                            ) sub
                       GROUP BY year
                       ORDER BY year")
number21
```

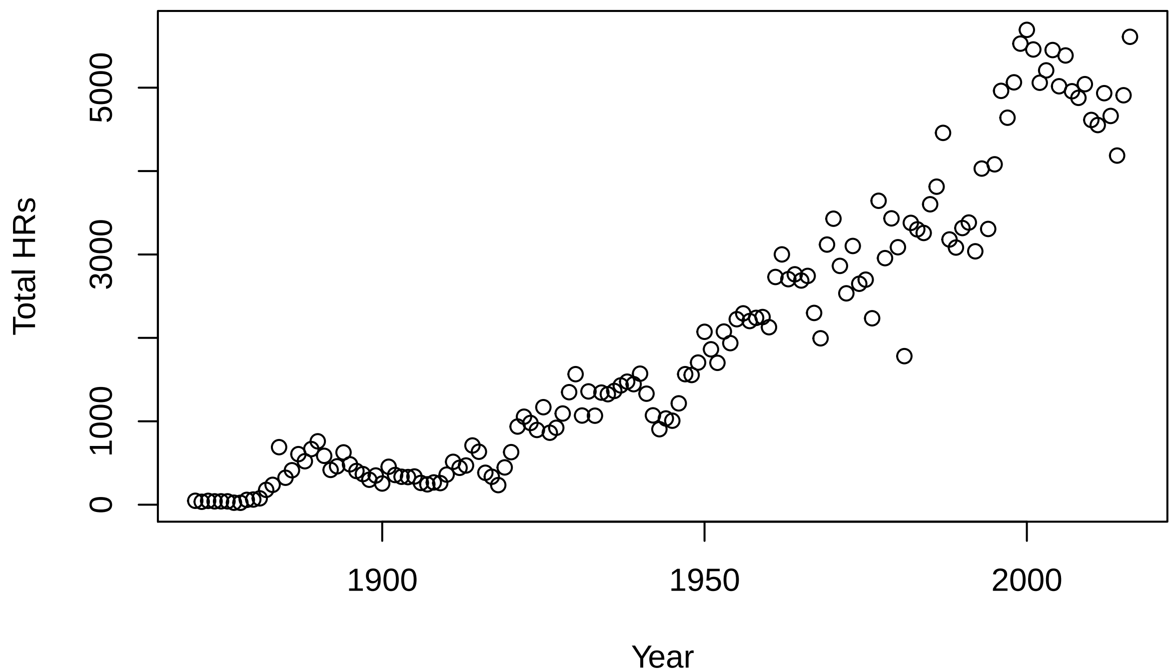
```
##    year max_salary
## 1  1985   14807000
## 2  1986   18494253
## 3  1987   17099714
## 4  1988   19441152
## 5  1989   21071562
## 6  1990   23361084
## 7  1991   36999167
## 8  1992   44788666
## 9  1993   47279166
## 10 1994   49383513
## 11 1995   50590000
## 12 1996   54490315
## 13 1997   62241545
## 14 1998   72355634
## 15 1999   86734359
## 16 2000   92338260
## 17 2001  112287143
## 18 2002  125928583
## 19 2003  152749814
## 20 2004  184193950
## 21 2005  208306817
## 22 2006  194663079
## 23 2007  189259045
## 24 2008  207896789
## 25 2009  201449189
## 26 2010  206333389
## 27 2011  202275028
## 28 2012  196522289
## 29 2013  231978886
## 30 2014  217014600
## 31 2015  215792000
## 32 2016  222997792
```

```
number22 <- dbGetQuery(conn, "
                       SELECT SeriesPost.yearID AS year,
                              SeriesPost.teamIDwinner AS teamID,
                              COUNT(AllstarFull.playerID) AS num_allstars
```

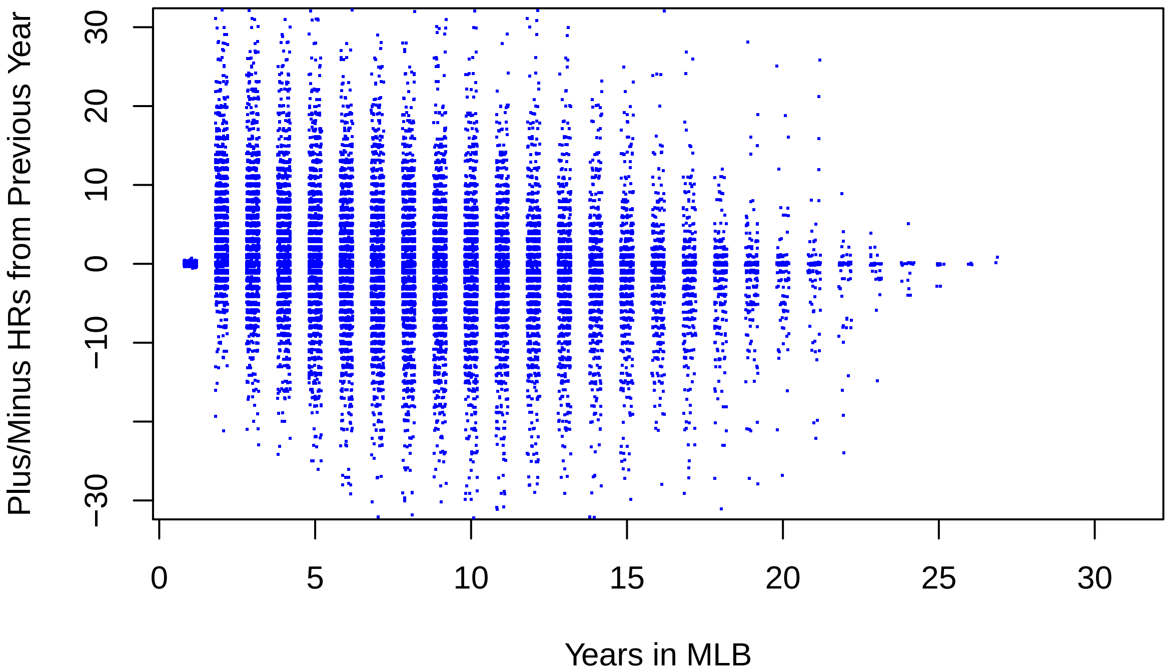
```r
                              FROM SeriesPost
                              JOIN AllstarFull
                              ON SeriesPost.yearID = AllstarFull.yearID AND
                                 SeriesPost.teamIDwinner = AllstarFull.teamID
                              WHERE SeriesPost.round = 'WS'
                              GROUP BY year
                              ORDER BY num_allstars DESC
                              ")
number22 %>% head()
```

```
##   year teamID num_allstars
## 1 1960    PIT           16
## 2 1961    NYA           14
## 3 1962    NYA           13
## 4 1939    NYA           10
## 5 1947    NYA            9
## 6 1958    NYA            9
```

```r
number23 <- dbGetQuery(conn, "
                              SELECT yearID AS year, SUM(HR) AS total_HR
                              FROM Batting
                              GROUP BY year")
plot(number23$year, number23$total_HR, xlab='Year', ylab='Total HRs')
```



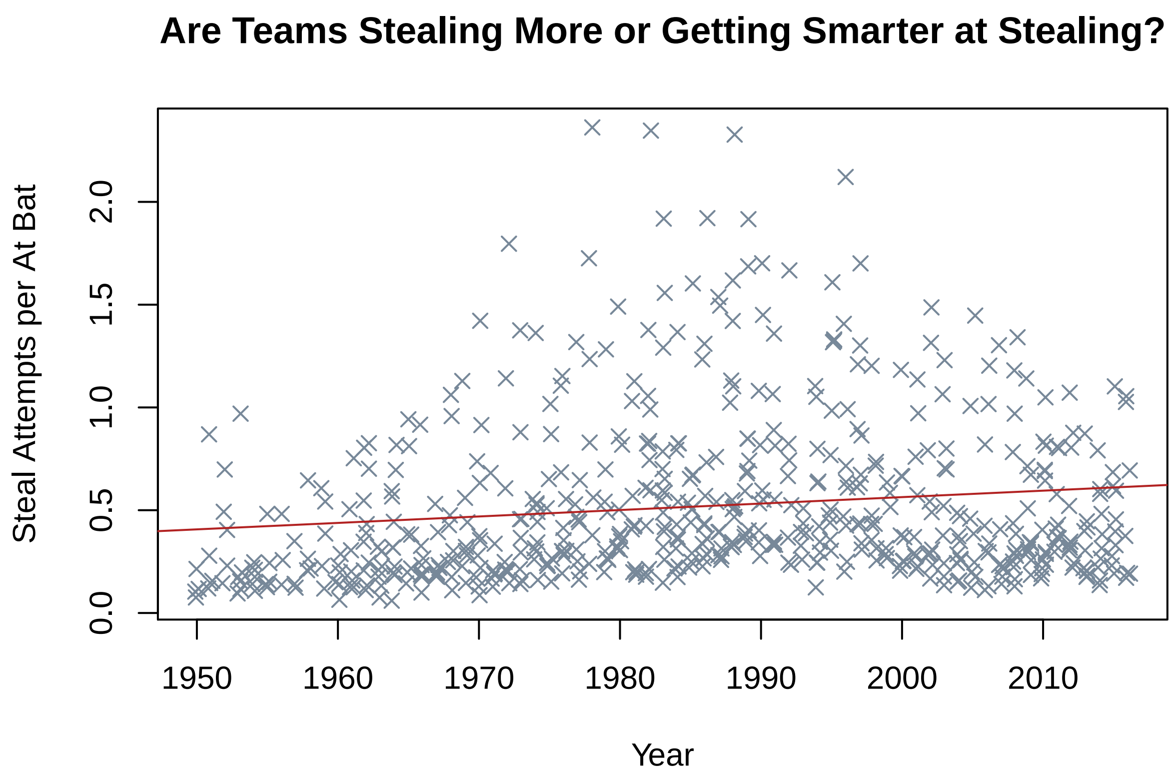
The above graph makes it very evident that home runs have increased in frequency over time, in an almost-exponential fashion. While the general trend in the MLB is to hit more home runs, this does not determine whether individual players hit more home runs during their career. To figure out this question, we can look at a year-to-year plus/minus. Each player's plus/minus for the year is the number of homers they hit that year minus their home runs from the year before. This means that all players start their career with a plus/minus of zero, then the statistic varies from there. The following graph visualizes this metric for players with at least 10 years in the MLB.



The graph shows most players do not hit more home runs as their careers progress in the MLB. The highest concentration of points is at a plus/minus of zero and the distribution is about symetric, meaning players are for the most part consistent. In fact, the best players (or at least the ones with the longest careers) have their plus/minus centered at zero, indicating that home run consistency may lead to longer careers.

One aspect of baseball I wanted to analyze was the evolution of the stolen base. From the begining of our database's record until about 1950, there is no real trend of how *well* people stole bases. Looking at the ratio of stolen bases to runners caught stealing, we can see that around 1950 the pitcher-catcher duos had started to figure out how to keep runners from stealing, keeping them to less than 1.5 steals per out after a period of 25 or so years where runners were getting 2 to 3 bases per out. But after 1950, runners have steadily increased their ratio of completed steals to times caught stealing.

Since runners are getting on base stealing more often now than in the past, the question becomes whether teams are stealing more, or if they have become smarter about stealing. Looking at the data from 1950 until the present, teams have attempted to steal (successful or not) about as much as they have in the past per at-bat. Modeling year to steal attempts per at-bat explains less than two percent of the variation. So teams have gotten smarter at stealing; they are attempting to steal only slightly more than in the past, but have increased their success at getting to the bag.




```
##
## Call:
## lm(formula = steal_attempts_per_AB ~ year, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4719 -0.2731 -0.1427  0.1539  1.8676
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.7132517  1.7952528  -3.182 0.001536 **
## year         0.0031385  0.0009039   3.472 0.000554 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.395 on 597 degrees of freedom
## Multiple R-squared:  0.01979,    Adjusted R-squared:  0.01815
## F-statistic: 12.06 on 1 and 597 DF,  p-value: 0.0005535
```