Technical Report

Group 6

Abstract

The competitive balance tax (implemented in 2002) is a rule implementing salary caps for baseball players in a given team. This paper attempts to uncover if there is any underlying difference in the number of lost potential players (called "couldabeens") in the pre-rule (1968-2002) and post-rule eras (2002-2018).

Introduction

Major League Baseball's "competitive balance tax," first implemented in 2002, has become a favorite subject of outrage among players and fans alike in recent years. On paper, the policy was meant to make the sport more competitive and boost salaries for players on lower-revenue teams. The owners and the players' union would negotiate the largest reasonable amount a team could spend on its roster, and any team that wanted to spend beyond that cap would pay a "tax" (a share of their excess payroll) to be redistributed to the poorer teams. In practice the policy has effectively become a salary cap, particularly after the 2016 renegotiations. In 2018 only two MLB teams out of thirty went over the salary threshold, and only one of those by any significant margin (the team that did so, the Boston Red Sox, unsurprisingly went on to win the World Series). With leaguewide revenue growth far outpacing growth in the revenue cap and most younger players effectively locked out of salary negotiations by free agency rules, players have understandably chafed at the limitation on their salaries, with mutterings of a player strike unless the rule is altered.

But while the depressive effect on players' salaries is not in dispute, the question of whether the rule *hurts the game* (a far graver sin among baseball fans than merely conspiring to lower salaries) is more open. At least in the conventional wisdom, the tax encourages teams to cultivate a massive pool of recruits and then pay the best of them the minimum permitted salary for up to seven years before they age into free agency. Once these players enter free agency, the narrative goes, the team dumps them for younger players whom they can pay the minimum salary. The remainder of their salary cap goes towards retaining a few older superstars with the name recognition to bring in fans, with many good-but-not-Mike-Trout players pushed into early retirement. The implication, then, is that the salary policy "hurts the game" because the MLB is less likely to be putting the best 30 shortstops in the world on the field at any given time. I would like to test this anecdotal observation empirically.

So, the working hypothesis states that since the advent of the competitive balance tax, the number of better players forced into retirement annually will have increased. A "better player forced into retirement" will be defined as a player under the age of 32 who a) left baseball due to their contract not being renewed, rather than injury or personal circumstances, and b) was outperforming the median rookie in their position at the time of their retirement. In other words, we might expect to find the number of couldabeens to increase after the implementation of the rule.

Methods: The Data

WAR Data

Our data comes from https://stathead.com/baseball/ and we mainly have four sets of data.

1. Rookie pitchers
2. Rookie position players
3. Retired pitchers
4. Retired position players

Payroll data

We also have data on the revenues and payrolls in the MLB for a given year.

1. MLB yearly payroll
2. MLB yearly total revenue

The research question in this paper hinges on classifying retired players of lost potential. To motivate this, we define a classifier for a "couldabeen" in the Classifiers section below. Before proceeding, we clarify the meaning of 'WAR', which is an essential statistic in our research.

What is WAR?

The Wins Above Replacements (WAR) of position players and pitchers are calculated differently.

Position Player's WAR

$$\text{WAR} = \frac{(\text{Player Runs} - \text{Avg Runs}) + (\text{Avg Runs} - \text{Replacement Runs})}{\text{Game Runs to Wins Estimator}}$$

Player Runs = Batting Runs+Baserunning Runs+Double Play Runs+Fielding Runs+Positional Adjustment

Pitcher player's WAR

$$\text{WAR} = \frac{\text{(aARA+aPRA) + (aRRA-aARA)}}{\text{Game Runs to Wins Estimator}}$$

where

Abrev.	Meaning
aARA	Adjusted Average Runs Allowed
aPRA	Adjusted Player Runs Allowed
aRRA	Adjusted Replacement Runs Allowed

Table 1: Abbreviations for Pitcher WAR.

The "replacement-level player" used in WAR is an estimate for the average midseason replacement. Being better than a midseason replacement does not necessarily make you better than the generation of rookies actually replacing you. If we want to argue that a given retiree, on merit, should have kept playing, we require a higher standard.

Methods: Classifiers

We define a few classifers that are useful for our analysis.

The Couldabeen Classifier

For a given year Y, we first compute the mean rookie's WAR, call it μ_Y. Then, we construct the corrosponding classifier for "couldabeen" status C of a given retired player p (from the year Y) to be as follows:

$$C(p) = \begin{cases} True, \text{ WAR}_p \geq \mu_Y \\ False, \text{ WAR}_p < \mu_Y \end{cases}$$

The Moneyball Classifier

To demonstrate that the release of *Moneyball* in 2003 is a fair place to partition the data, we perform a hypothesis test on the `postMoneyball` classifier of a year Y, which we define as follows:

$$\text{postMoneyball}(Y) = \begin{cases} True, Y > 2003 \\ False, Y \leq 2003 \end{cases}$$

Methods: Model Overview

Data Wrangling Procedures

All that being said and done, we briefly describe the data wrangling procedures to obtain our response variable `prop` and our predictor `laborShare`.

(i) `prop`: Our crowning jewel in this research project is the creation of our year-indexed `prop` response variable: the rate of lost potential players ("couldabeens") in a given year. This variable is obtained by utilizing two classes of datasets in parallel: rookie player data and retiree player data. Note that the process for position players and pitcher players is not different, but we split these datasets due to the systemic differences in the WAR statistic for these classes of players.

So, to compute `prop`, we begin by taking our rookie player dataset `player_rkes` and using `dplyr::group_by(Year)` to compute the yearly statistics for the mean rookie WAR. After doing so, we create a dataframe called `player_thresholds` listing these thresholds by year. Once this is done, we use these thresholds and count the number of retirees that exceed that threshold in that given year. Finally, we obtain the number of retirees that year and normalize to obtain the proportion. So, if we denote (for a given year Y) counted couldabeens CB_Y and the number of retirees R_Y, we obtain the proportion or rate of couldabeens by letting $\text{prop}(Y) = \frac{CB_Y}{R_Y}$.

(ii) `laborShare`: From the payroll dataset, we have yearly data for the total revenue and total payrolls in MLB. This is quite convenient since our proportion response variable is similarly sorted by year. As such, we create the `laborShare` variable by using the following formula:

$$\text{laborShare(Y)} = \frac{\text{totalPayroll(Y)}}{\text{totalRevenue(Y)}}$$

Models

Once every retired player is classified appopriately, we run a few models and a hypothesis test:

(i) Linear Model (`prop ~ Year`): After aggregating all the retired players and their classifications, we summarize the data by year to obtain the proportion of retired players that were couldabeens that year, called `prop`. As such, we now have 50 data points (for each year), and a response variable being the proportion of couldabeens. As such, we run a linear model fitting `prop ~ Year` for (i) the unpartitioned data, (ii) the pre-rule era and (iii) the post-rule era. Because there will always be "couldabeens", we do not expect a large effect size and hence a very significant result, however, the **sign** of our coefficient will be essential for our inference. If our research hypothesis is correct (that there is an effect), we expect to see a positive coefficient for β_{Year} on the post-rule era partition.

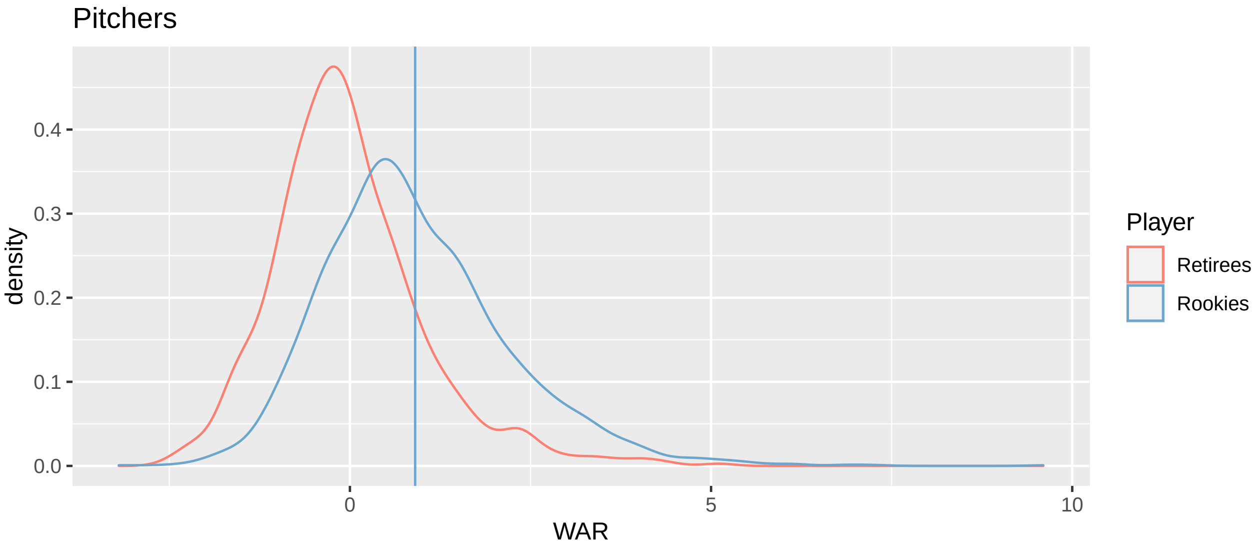
(ii) Hypothesis Test (`prop ~ postMoneyball`): With the dummy variable `postMoneyball` at hand, we perform a hypothesis test to see if the mean proportion of couldabeens in the pre-Moneyball era is different than that of the post-Moneyball era. If the result of our hypothesis test is significant, we proceed to partition our dataset since it affirms our belief that the publication of 'Moneyball' is indeed a confounding variable.

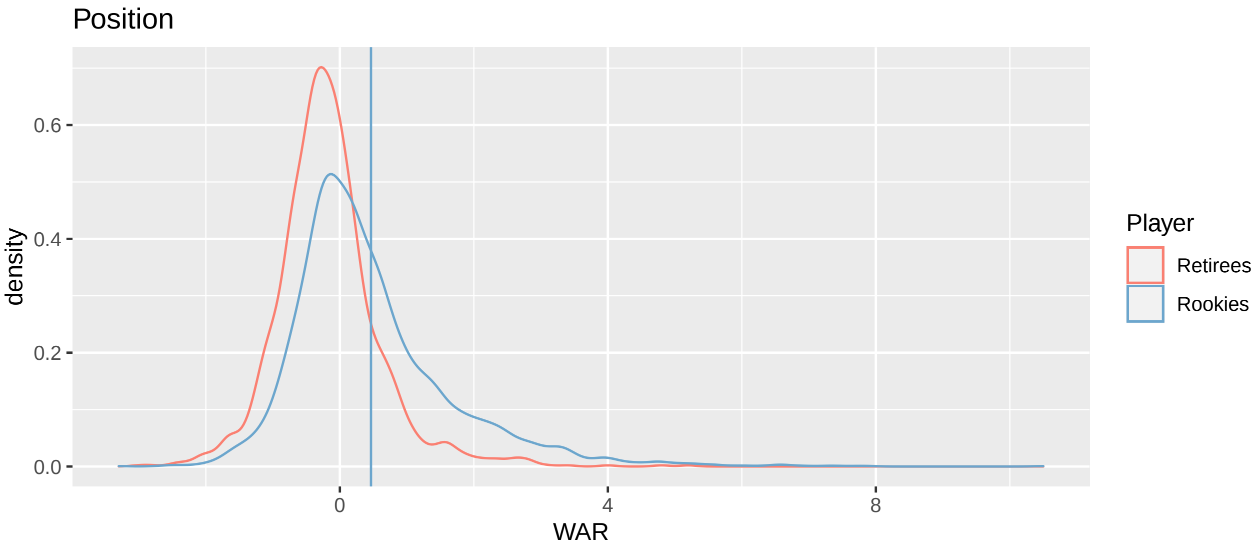
(iii) Linear Model (`prop ~ laborShare`): Lastly, we use the payroll data to obtain the labor share in the MLB for a given year, and fit that data to `prop`. By establishing this relationship, we can utilize the result in Bradbury's paper regarding the relationship of labor share and the luxury tax to attempt to address our research question.

Exploratory Data Analysis

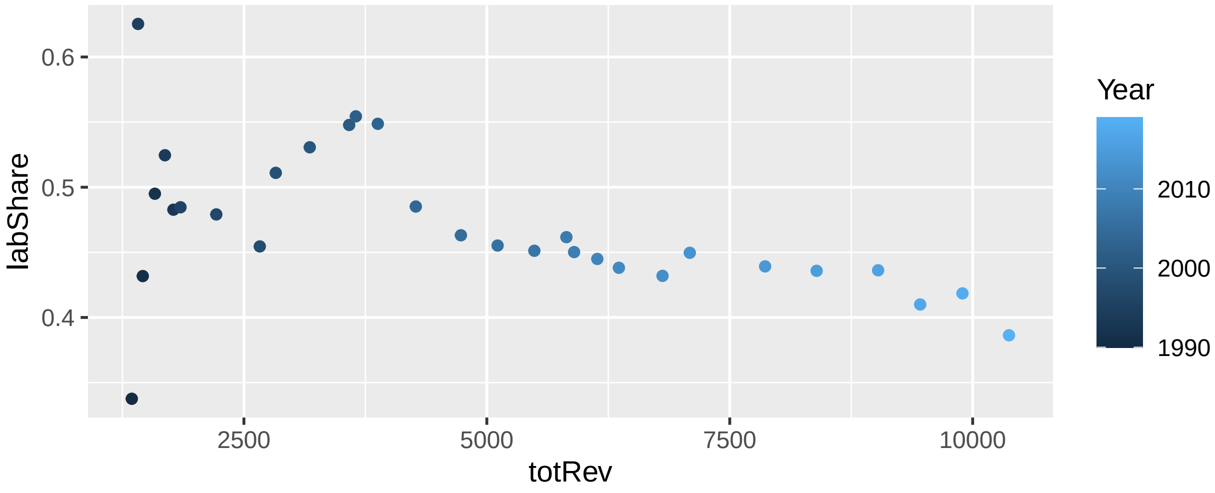
WAR Density Plots

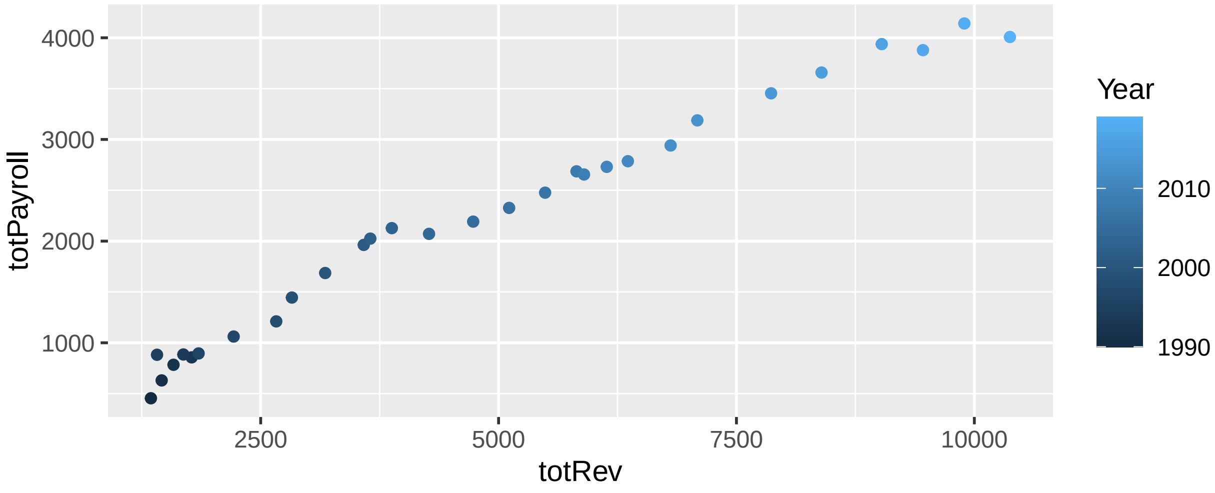
Below, we can visualize the densities of the WAR statistic for each type of player. In blue, we have the rookie players (with the median line denoted), and in red, we have the corrosponding retired players. As such, we can visualize the "couldabeens" as the retired players to the left of the median line (if the threshold $t = 0$), and we corrospondingly count the number of couldabeens based on year, from which we make our inference on the impact of Year.

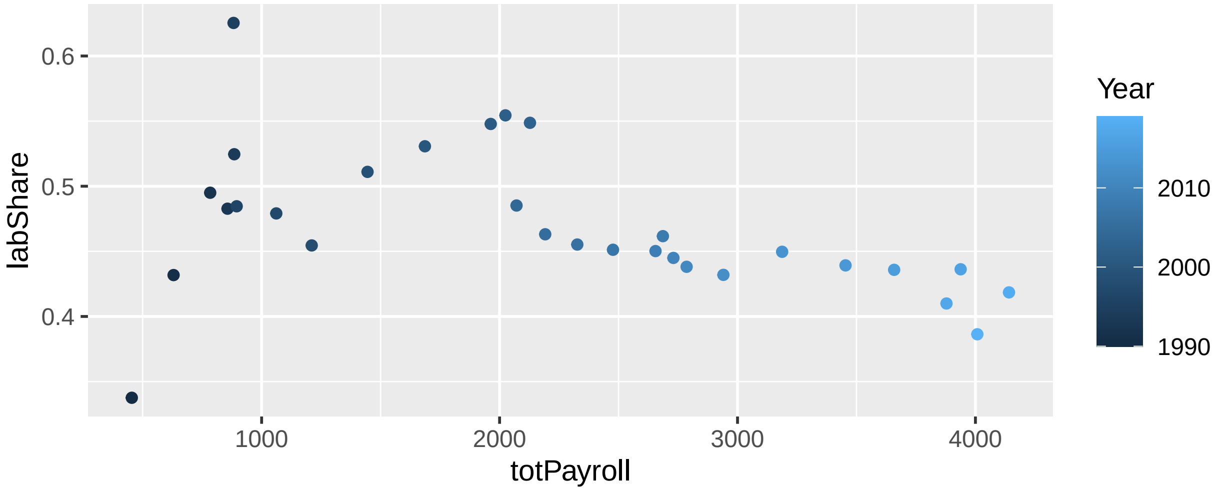




Payroll Predictor Data





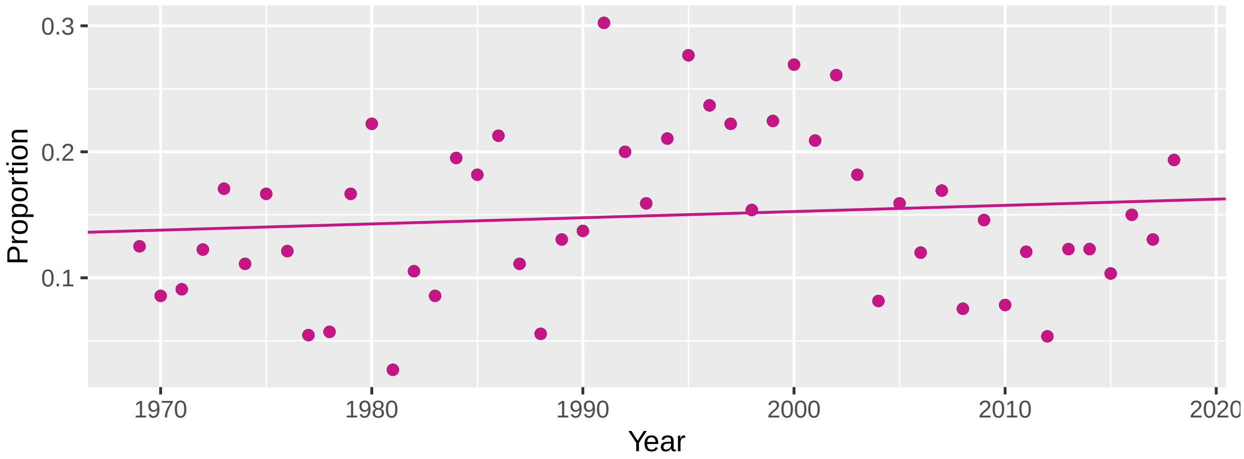


Models

Linear Model: Year

To answer our research question, we believe that we could establish the trend between `prop` by simply fitting the model against the year index `Year`. However, as we can see below, the data was non-linear, and fitting a model `prop ~ Year` lead to a coefficient of $\beta_{Year} \approx 0$ with an insignificant p-value of 0.440. This means the effect is essentially non-existent, and if it is, we cannot be sure this is not random (due to the statistical insignificance of β_{Year}).

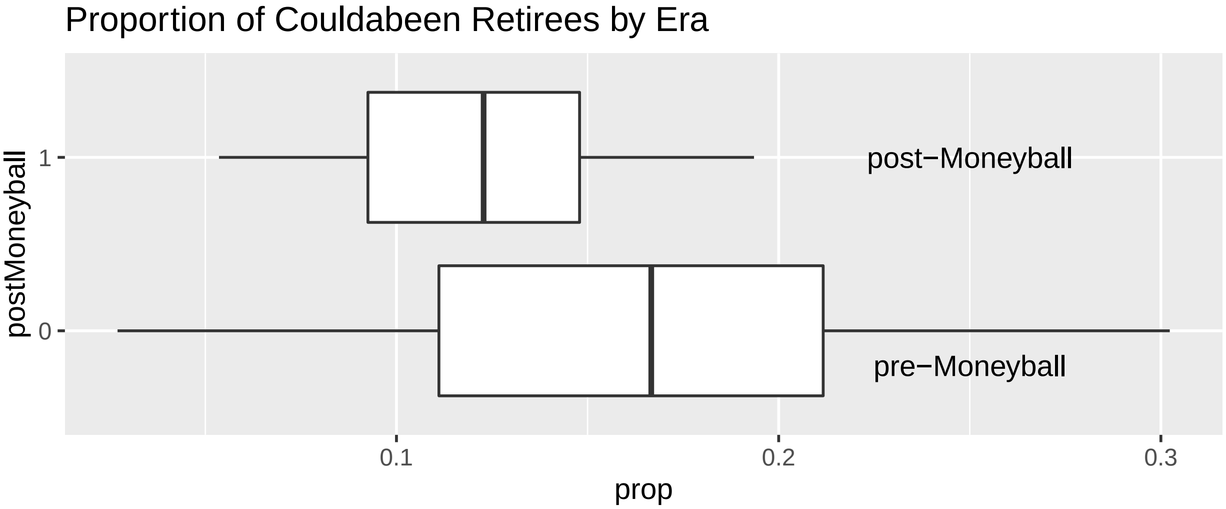
However, there is one particular feature of interest in our scatterplot data. Namely, there is a noticable "quadratic" shape in our `prop` data. Although it is not necessarily helpful to model a quadratic model for our research question, it does seem to indicate that there is an increase, then a drop again near our critical year of 2003. As such, we decide to further investigate this by putting our scope by splitting the data. However, we cannot just "ignore" data from previous years, so a formal investigation of the era partition is dicussed in the form of a hypothesis test.



```
##
## Call:
## lm(formula = prop ~ Year, data = couldabeens_t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.11623 -0.03820 -0.01083  0.04694  0.15415
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.8305575  1.2597256  -0.659    0.513
## Year         0.0004916  0.0006319   0.778    0.440
##
## Residual standard error: 0.06448 on 48 degrees of freedom
## Multiple R-squared:  0.01245,    Adjusted R-squared:  -0.008123
## F-statistic: 0.6052 on 1 and 48 DF,  p-value: 0.4404
```

Hypothesis Test: Why Split the Data?

As mentioned, we expect the Sabermetrics revolution to have reduced the number of couldabeens systemically. If we partition the eras, we find that the difference in means in proportion of couldabeens between the eras is statistically significant. Below, we have a boxplot to help us visualize the systemic drop in the proportion of couldabeens.



To perform this hypothesis test, we perform a fit `prop ~ postMoneyball`. We know that if we fit a response to a dummy variable, the corrosponding parameter returned is the difference in means among the eras. Namely, if we obtain an intercept of β_0 and a parameter value of β_p on the dummy variable, then performing least squares regression tells us that $\mu_{p=0} = \beta_0$ and $\mu_{p=1} = \beta_0 + \beta_p$. That being said, we perform the fit below.

```
##
## Call:
## lm(formula = prop ~ postMoneyball, data = couldabeens_t)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.134209 -0.042568  0.001007  0.045264  0.141090
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.16124    0.01052  15.334   <2e-16 ***
## postMoneyball -0.03944    0.01920  -2.054   0.0454 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06221 on 48 degrees of freedom
## Multiple R-squared:  0.08081,    Adjusted R-squared:  0.06166
## F-statistic:  4.22 on 1 and 48 DF,  p-value: 0.04543
```

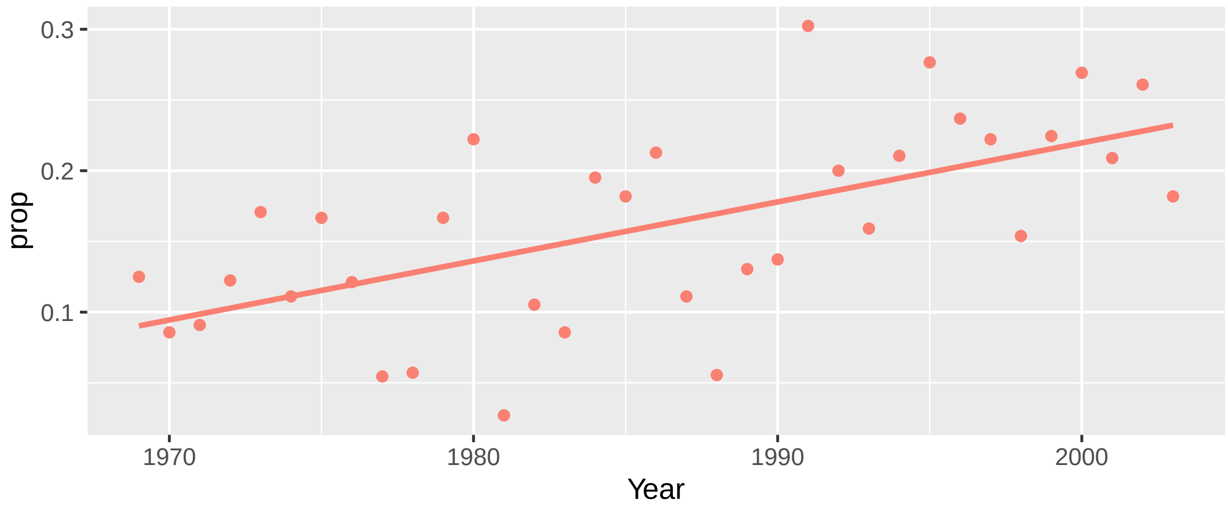
As explained previously, we interpret the parameters as informing us of the difference in means in the corresponding levels of our dummy variable. This means that we have found that the mean of `prop` in the pre-Moneyball era is approximately 0.16 and that of the post-Moneyball era is 0.12. The result of this test is statistically significant, with a p-value of 0.0454. As such, we proceed to partition the dataset to account for the systemic amount of reduction in couldabeen rates that is inherent in the post-Moneyball era as a result of the Sabermetrics revolution.

Models: Splitting the Data

Linear Model: Pre-Moneyball Years

As demonstrated, it may be worthwhile to investigate `prop ~ Year` on the partitioned dataset making our partition based on the `postMoneyball` dummy variable. Once we do so, we fit the two seperate models fiting `prop ~ Year` in both eras. Firstly, we attempt to capture the trendline in the pre-Moneyball years. While this model is not particularly useful to fit this model (the scope of our research question is concerned with the post-Moneyball years), we fit this model to observe the effect size and trend in that era and do so anyway for completeness.

That being said, we observe a $\beta_{Year} = 0.004174$, interpreting this as saying that for every year after 1969 until 2003, we expect to see a 0.4% increase in the rate of couldabeens. Interestingly, this trend is statistically significant with a p-value of approximately 0. Again, we are not concerned with this era, so we go ahead and fit the model of the proportions in the post-Moneyball era.



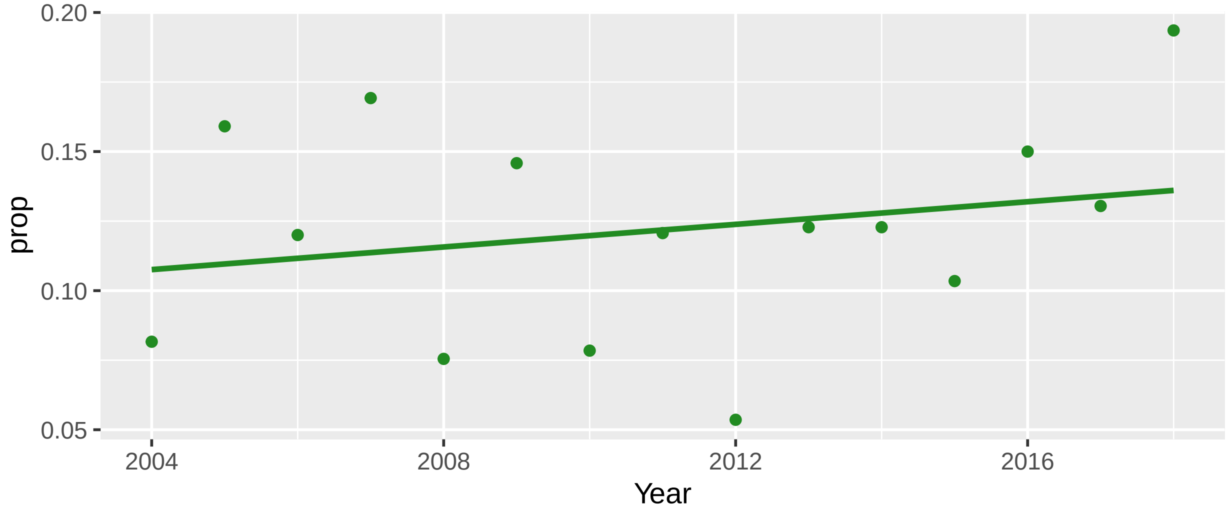
```
##
## Call:
## lm(formula = prop ~ Year, data = couldabeens_pre)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.114028 -0.042000  0.008993  0.034685  0.120220
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.1282589  1.8541569  -4.384 0.000112 ***
## Year         0.0041740  0.0009336   4.471 8.69e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05578 on 33 degrees of freedom
## Multiple R-squared:  0.3772, Adjusted R-squared:  0.3583
## F-statistic: 19.99 on 1 and 33 DF,  p-value: 8.69e-05
```

Linear Model: Post-Moneyball Years

Finally, we come to our last and most important linear model, fitting `prop ~ Year` for data in the post-Moneyball era. Since the release of Moneyball coincides with the luxury tax (2003), we believe that accounting for this confounding variable may allow us to see the trendline in this era and make conclusions about the change in the rate of couldabeens in that era. That being said, we recall that the research hypothesis states that the luxury tax lead to a rise in the rate of couldabeens. To affirm this, one must find that $\beta_{Year} > 0$, since we would want to see that for every year after 2003, there is some percentage increase in the proportion of retirees who are couldabeens.

That being said, we observe a $\beta_{Year} = 0.002034$, interpreting this as saying that for every year after 2003 until 2018, we expect to see a 0.2% increase in the rate of couldabeens. However, before prematurely concluding that the effect has been shown, we unforunately observe that this fit (and hence trendline) is **statistically insignificant** with a p-value of approximately 0.398. To explain this, it may be helpful to observe the sizes of our partitions. Namely, it is possible that we simply do not have as much data in the post-Moneyball era as we did the pre-Moneyball era (15 data points as opposed to 35 data points), so running this a few more years into the future may potentially lead to a more statistically significant result based on `Year` alone.

However, despite the fact, it is assuring that $\beta_{Year} > 0$, indicating that we may be going in the right direction. From this point onwards, there are two possible ways to go: (i) incorporate more data, or (ii) resampling the original data and observe a confidence interval. Next, we take the first strategy (i) and incorporate new data to make another linear model. We explore (ii) resampling in the *Results* section.



```
##
## Call:
## lm(formula = prop ~ Year, data = couldabeens_post)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.07026 -0.02621 -0.00306  0.02307  0.05751
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.967977   4.680332  -0.848    0.412
## Year         0.002034   0.002327   0.874    0.398
##
## Residual standard error: 0.03894 on 13 degrees of freedom
## Multiple R-squared:  0.05548,    Adjusted R-squared:  -0.01718
## F-statistic: 0.7636 on 1 and 13 DF,  p-value: 0.3981
```

Labor Share Model

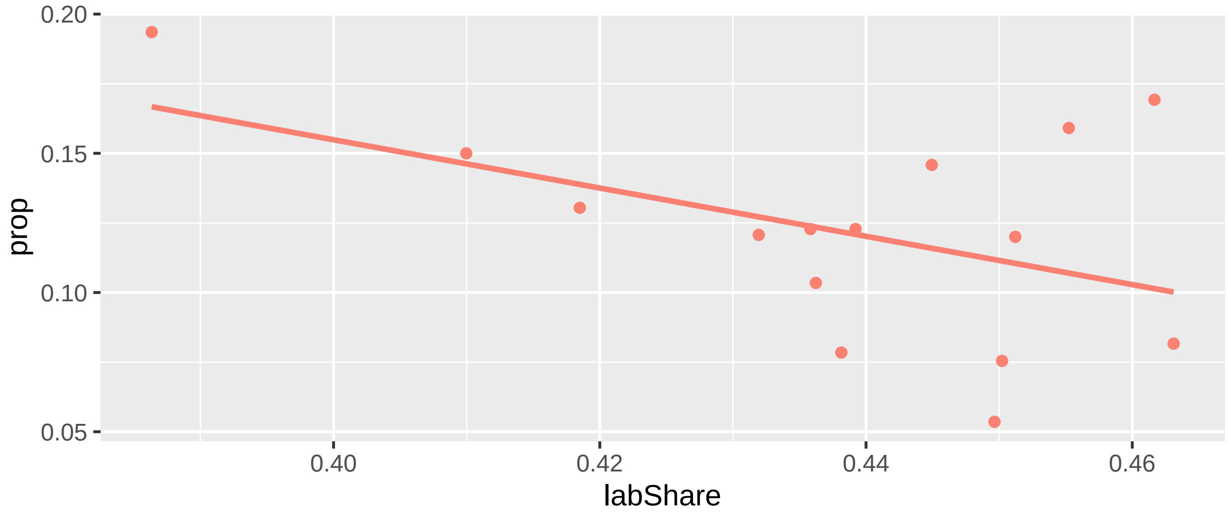
Since we could not establish a relationship between `prop` and `Year` alone in the post-Moneyball era, we seek another path by incorporating labor share data. As such, we obtain the payroll and revenue dataset for the MLB and run our new predictor against the rate of couldabeens; so, we fit `prop ~ labShare` and we obtain the following model.

Indeed, we find that labor share is negatively correlated to `prop` with $\beta_{LS} = -0.8670$. We may interpret this as saying that for every 1 increase in labor share, we might expect to find a decrease of -0.8% in the rate of couldabeens. This time around, our result is statistically significant with a p-value of 0.0845. With respect to our research question, we may bring this back by citing the result of Bradbury's research on the effect of the luxury tax on labor share.

If it is established that Bradbury's thesis that labor share is indeed decresaing as a result of the luxury tax, we may use that result in conjunction with this model to say the following:

> Since lower labor shares indicates that the proportion of couldabeens is higher, the luxury tax may have been resulting in this increased rate of retired couldabeens with labor share as the intermediary variable.

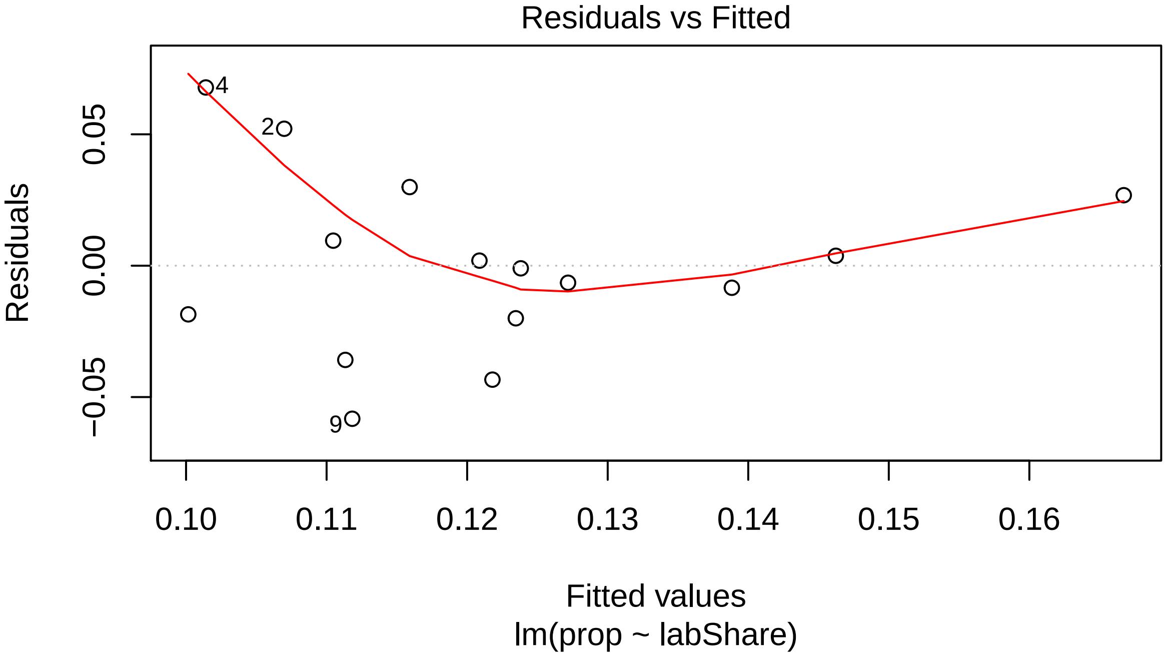
This will be discussed further in the Results and Conclusions sections.

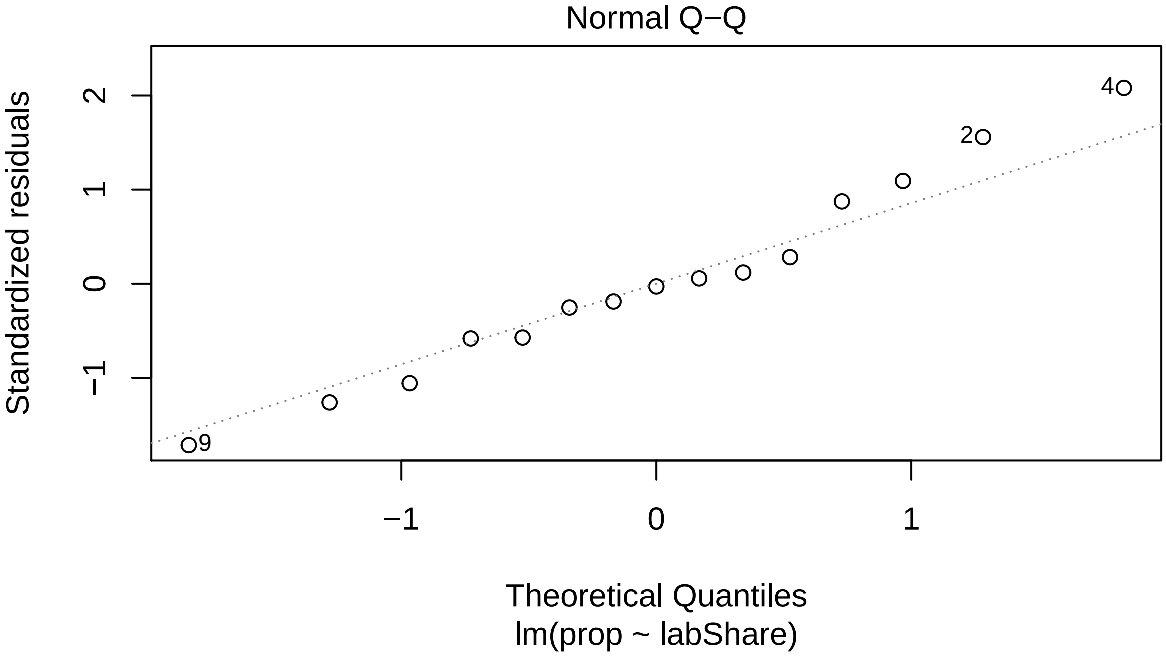


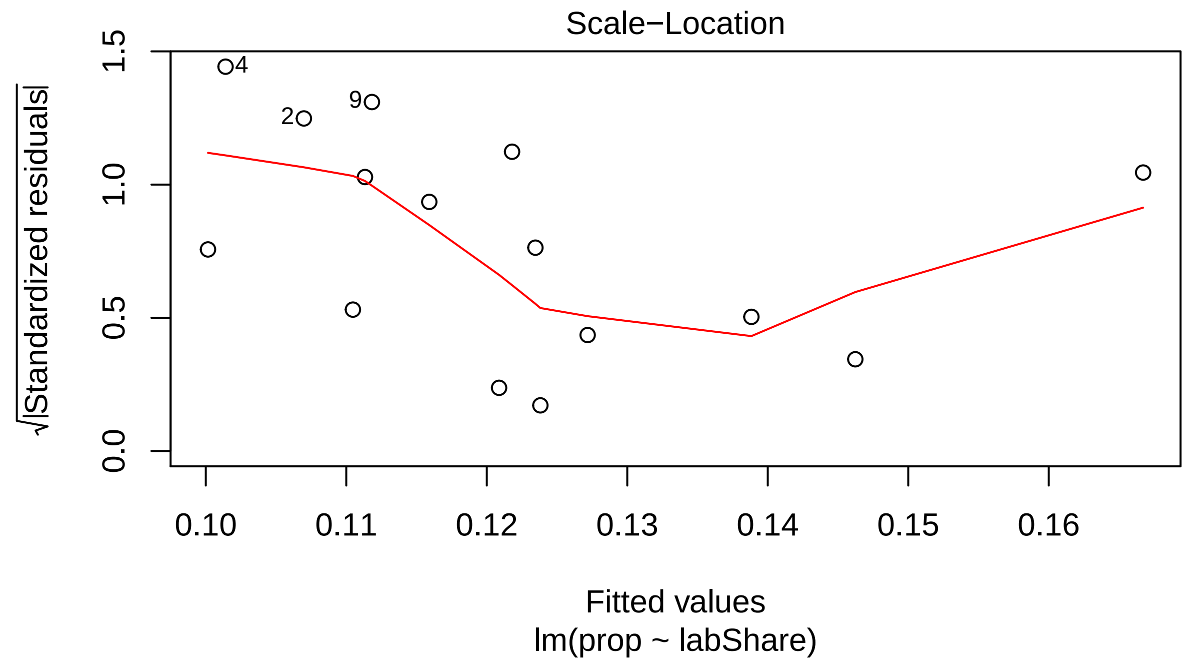
```
##
## Call:
## lm(formula = prop ~ labShare, data = couldabeens_post)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.058253 -0.019268 -0.001008  0.018178  0.067824
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5017     0.2036   2.464   0.0284 *
## labShare     -0.8670     0.4642  -1.868   0.0845 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03558 on 13 degrees of freedom
## Multiple R-squared:  0.2116, Adjusted R-squared:  0.151
## F-statistic:  3.49 on 1 and 13 DF,  p-value: 0.08446
```

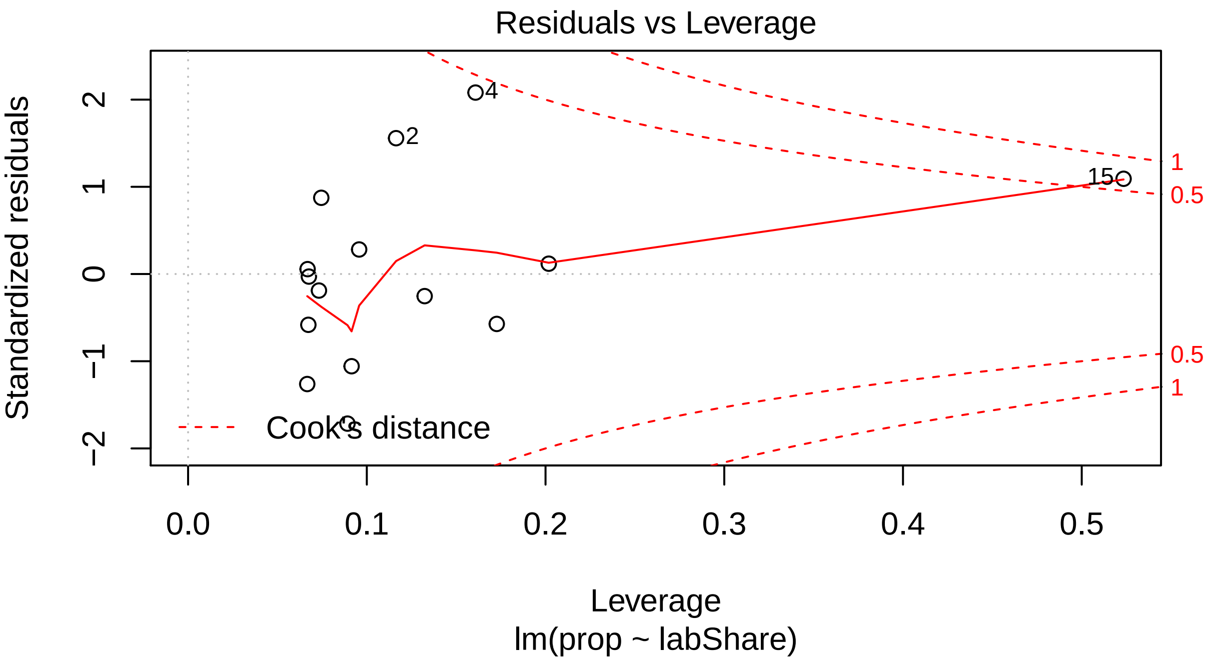
Diagnostic Plots: Labor Share Model

As our best-performing and most relevant model, we analyze the diagnostic plots of our linear model on labor share. At first glance, this model is not without problems. Namely, we notice that our residuals are not uniform as one might notice in the original model plot. In fact, the model has systemically higher residuals for smaller fitted values (near 0.10) indicating that years with higher labor share may be more difficult to predict. However, other than this issue of non-uniform residuals over all fitted values, the model seems to be well-behaved. Our Normal-QQ plot seems to be sufficiently close to the line, with slight deviations on the boundaries. Lastly, the leverage plot seems to indicate that there may be some high leverage points in the model. In fact, this may be confirmed by a quick glance at our model scatterplot. However, since this data is valid, we can say that these leverage points do not necessarily cause decreased confidence in the model output, but rather, may slightly skew the result in the wrong direction.






Scale−Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
4
2
9
0.10
0.11
0.12
0.13
0.14
0.15
0.16
Fitted values
lm(prop ~ labShare)


Residuals vs Leverage
Standardized residuals
2
1
0
−1
−2
4
2
15
1
0.5
0.5
1
Cook's distance
0.0
0.1
0.2
0.3
0.4
0.5
Leverage
lm(prop ~ labShare)

Results

Linear Model on Year

(Unpartitioned)

- $\beta_{Year} \approx 0$.
- If fitting `prop ~ Year` on the entire dataset, we find no linear relationship between proportion of couldabeens along the years.
- β_{Year} is statistically insignificant with a high p-value of $p = 0.440$.
- There seems to be a "quadratic" trend in the data, so we investigate a partition.

Hypothesis Test

- To justify partitioning the dataset, we theorize that `postMoneyball`, a dummy variable indicating if a year is before or after the release of Moneyball may lead us in the right direction.
- Namely, since Sabermetrics might lead to more efficient choices of rookies, we expect to see a drop in the proportion of couldabeens due to this increased efficiency.
- So, we fit `prop ~ postMoneyball` to perform a *difference of means* hypothesis test, measuring the likelihood of observing our data given that the mean `prop` before and after Moneyball is equal.
- In fact, we find that the means may not be equal as we obtain that $\beta_{MB} = -0.039$ with a p-value of 0.045, indicating statistical significance and the rejection of the null hypothesis ($\mu_{Pre} = \mu_{Post}$) if $\alpha = 0.05$. Interpret this as saying that we expect to see an average of 4% less in the rate of couldabeens in the post-Moneyball era.
- Finally, use this result to justify partitioning the dataset.

Linear Model on Year

(Pre-Moneyball era)

- Now that our dataset is partitioned, we may fit `prop ~ Year` in both eras.
- We fit the model and find that $\beta_{Year} = 0.004174$.
- β_{Year} is statistically significant with a very low p-value of $p \approx 0$.
- However, our research question is not concerned with this parameter. We do however find that the rate of couldabeens is increasing between 1968 and 2003.
- That being said, since we observe a $\beta_{Year} = 0.004174$, we interpret this as saying that for every year after 1969 until 2003, we expect to see a 0.4% increase in the rate of couldabeens.
- Next, we fit the post-Moneyball era model for our research question.

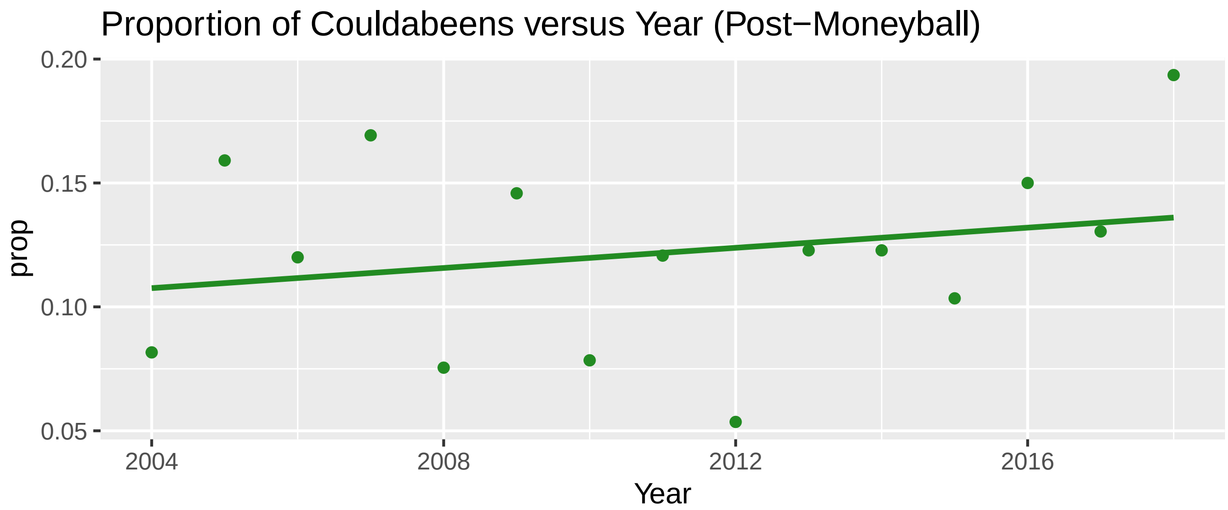
(Post-Moneyball era)

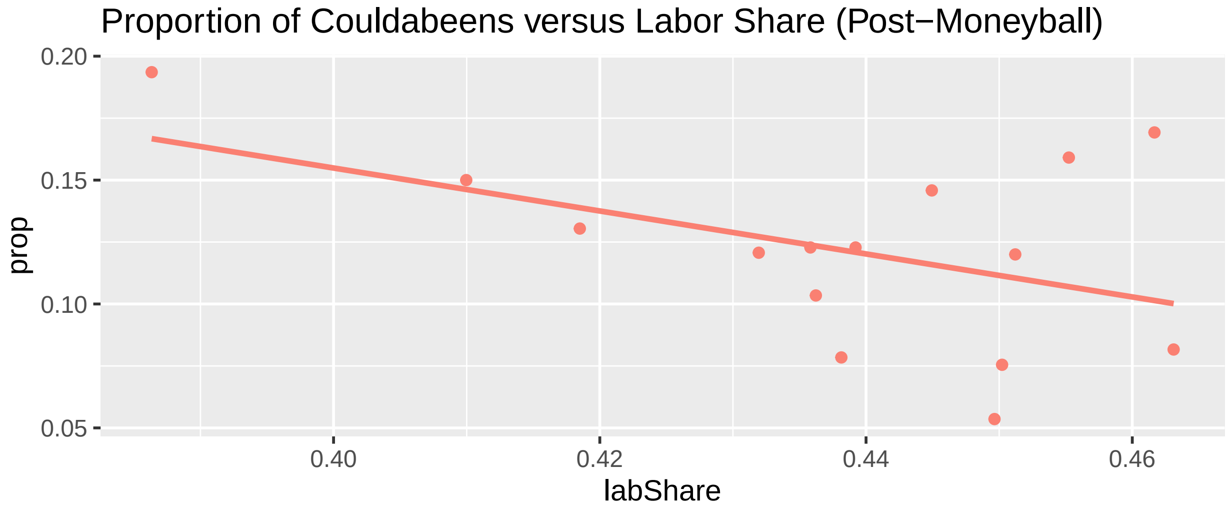
- $\beta_{Year} = 0.002034$.
- $\beta_{Year} > 0$ supports the hypothesis that there is an increasing rate of couldabeens since the luxury tax (as it coincides with the release of Moneyball.
- More precisely, since we observe a $\beta_{Year} = 0.002034$, we interpret this as saying that for every year after 2003 until 2018, we expect to see a 0.2% increase in the rate of couldabeens.
- Interestingly, this is a higher increase than that estimated in the pre-rule era.
- However, β_{Year} is not statistically with a high p-value of 0.398, so we may not base our inference on this parameter.
- Our next step is to either (i) bring in new data, or (ii) use resampling methods to estimate some confidence intervals.

Linear Model on Labor Share

- Since our post-Moneyball linear model was not statistically significant ($p = 0.398$), we cannot base our inference on that model.
- However, along the way, we did learn that partioning the dataset is particularly helpful due to the result of our hypothesis test.
- Additionally, our result that $\beta_{Year} > 0$ in the post-Moneyball era indicates that we may be in the right direction.
- To further investigate, we utilize payroll and revenue data to create a yearly labor share variable. Attaching this column to our proportion data, we have a new predictor.
- If the thesis of Bradbury's research holds, we have cause to believe that the luxury tax is leading to a shrinking labor share.
- So, we fit a model `prop ~ labShare` and find $\beta_{LabShare} = -0.867$ with a p-value of $p = 0.084$. As opposed to the previous models, this is statistically significant and relevant to our research question.
- We interpret this as saying every for every 1% increase in labor share, we might expect to find a decrease of -0.8% in the rate of couldabeens. So labor share is indeed **negatively** correlated with proportion of couldabeens.
- Since lower labor shares indicates that the proportion of couldabeens is higher, the luxury tax may have been resulting in this increased rate of retired couldabeens with labor share as the intermediary variable.

Selection of Model Plots

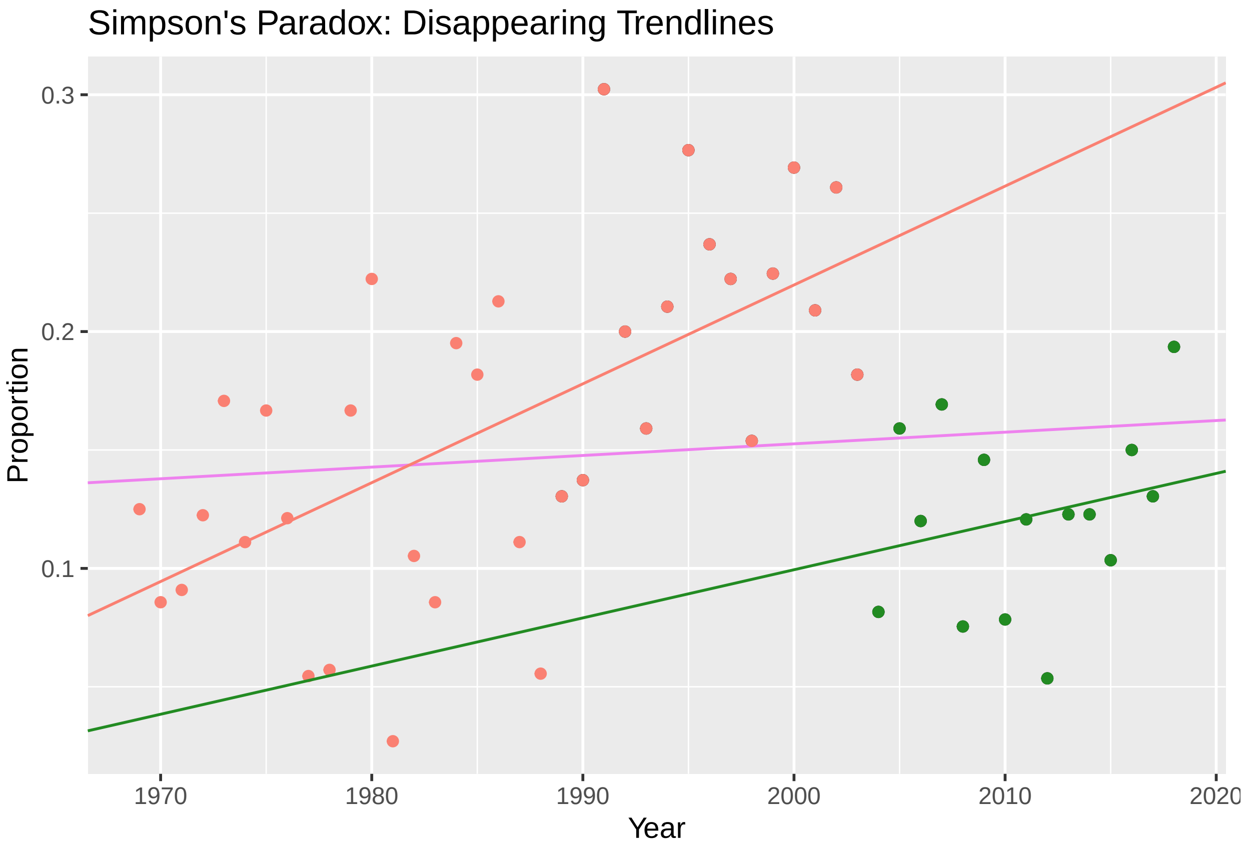

Proportion of Couldabeens versus Year (Post−Moneyball)


Proportion of Couldabeens versus Labor Share (Post−Moneyball)

Discussion

Simpson's Paradox

We now return to discuss the scatterplot obtained when we initially fit `prop ~ Year` on the dataset. Since we observed $\beta_{Year} \approx 0$ on the whole dataset, it was interesting to find that partitioning and fitting linear model with `prop ~ Year` yields $\beta_{Year} > 0$ in both partitions. Consider the plot below, which demonstrates this result.



To explain this, we consider the possible existence of a confounding variable along the years. Indeed, as shown in our hypothesis test, the `postMoneyball` dummy variable indicated that there is a statistically significant difference in means along the eras. This was explained as a result of the Sabermetrics revolution, which made more efficient the selection of better rookies and more cost-effective its selection of players.

So, while Simpson's paradox does not necessarily imply the existence of a confounding variable, the converse is often true. That is, we often find that when we have a confounding variable, we also observe Simpson's paradox. As such, this is reassuring, since we believed that partitioning the dataset into the pre-Moneyball and post-Moneyball eras is indeed the way to go. So to summarize, there is good reason to believe that our lack of trendline is a case of Simpson's paradox; with the appopriate candidate (`postMoneyball`), we found a statistically significant partition of our data from which we will base our inference on.

Generalization: Threshold Stability

Recall, that our couldabeen classifier was constructed by taking the mean rookie's WAR and using that as a threshold of classifying couldabeens. Now, consider the **General Couldabeen classifier**:

For a given year Y, we first compute the mean rookie's WAR, call it μ_Y. Additionally, compute the standard deviation of that year's data and call it σ_Y. Then, given a threshold $t \in \mathbb{R}$, we construct the corresponding classifier for "couldabeen" status C of a given retired player p (from the year Y) to be as follows:

$$C(p) = \begin{cases} True, \text{ WAR}_p \geq \mu_Y + t\sigma_Y \\ False, \text{ WAR}_p < \mu_Y + t\sigma_Y \end{cases}$$

With this at hand, we may generalize our inferential process and zoom out to consider a variety of threshold values $t \in \mathbb{R}$. If we consider our original model, we can say that our model assumes that $t = 0$, so the implied couldabeen threshold is simply the mean rookie WAR. However, this is not the only possibility for couldabeen threshold. In fact, there are an infinite number of thresholds we may consider in theory.

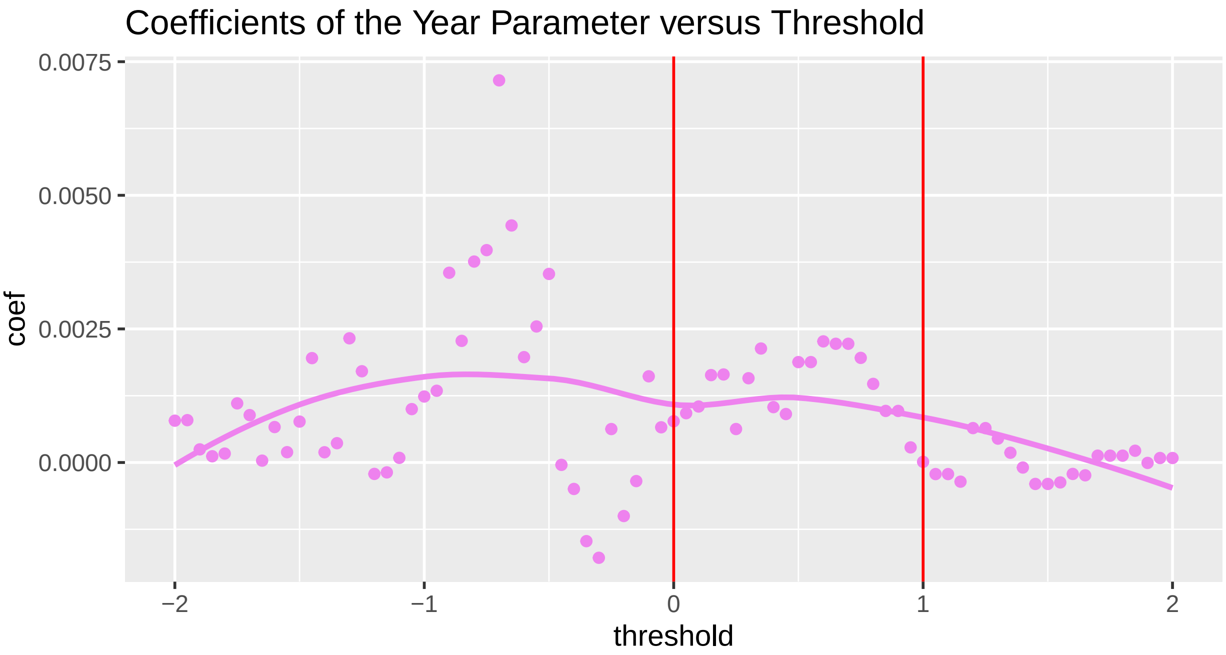
One practical range to consider is $t \in [-2, 2]$ corrosponding to thresholds $\theta \in [\mu_Y - 2\sigma_Y, \mu_Y + 2\sigma_Y]$. We motivate this by utilizing the 95%-rule saying that most data is in that 2σ range. However, note that in practice, the values for t much smaller than 0 are less important, since we are seeking lost potential players and reducing t in turn reduces the bar for a "high potential player". More important is to consider the range of values between $t \in [0, 1]$, where the bar is "just right"; if we tune $t \in [1, 2]$, we may be raising the bar too high and find odd results due to the noise in the data in the tails of our distributions. Regardless, we end up tuning $t \in [-2, 2]$ and observe the relevant results of our models to then connect it to our inferential questions. **More importantly, our priority is to establish that roughly every threshold in the critical $\mathbf{t} \in [\mathbf{0}, \mathbf{1}]$ tuning range indicates that $\beta_{\mathbf{Year}} > \mathbf{0}$ (for the post-rule era partition) and that $\beta_{\mathbf{labShare}} < \mathbf{0}$ in the same partition.**

That being said, we run the model on various thresholds in the range $[-2, 2]$ and observe the parameters for our selected linear models. The results are shown below.

Linear Model: Year

Below, we find our diagnostic plot showing β_{Year} for various thresholds. Highlighted in red is our critical $t \in [0,1]$ region. Upon first inspection, we find that $\beta_{Year} > 0$ for roughly all of the thresholds in the critical region. Although there is no indication that these results are statistically significant (i.e. by studying their p-values), showing invariance of our threshold parameter to our inferential prediction on its sign is beneficial as to generalize our inference beyond the point assumption $t = 0$.

Outside the critical interval, the plot shows that the coefficient is otherwise unstable, which poses an interesting question as to why this is the case. Since this behaviour is replicated in the other plot, this will be discussed later.

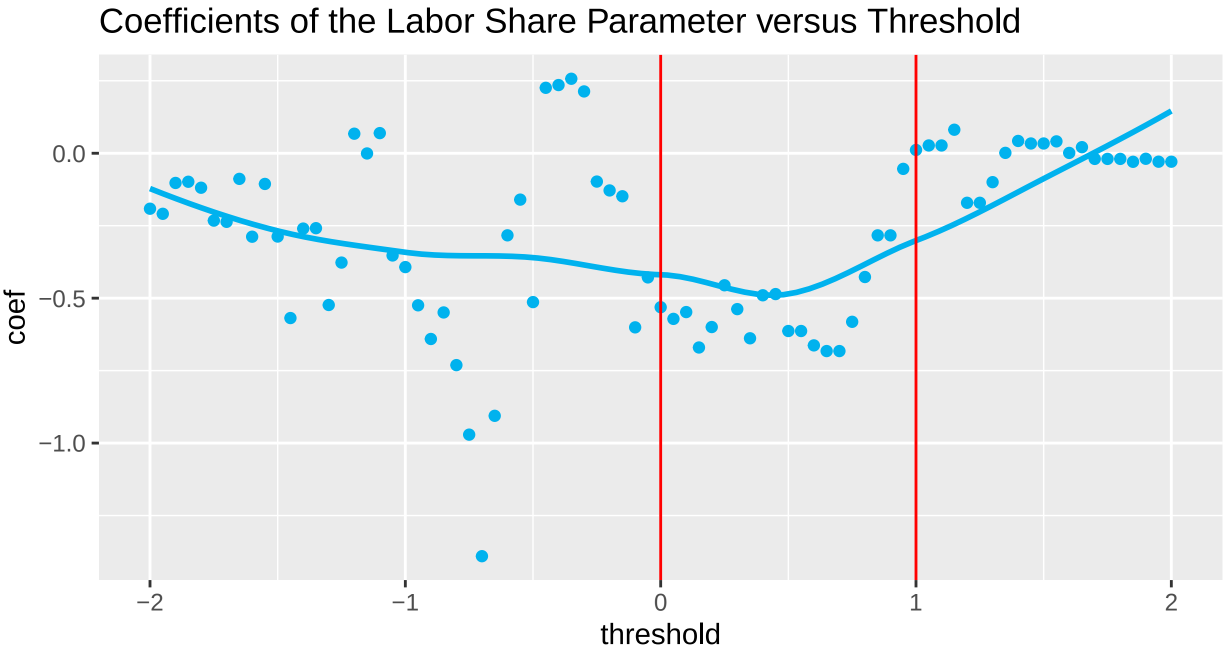


Linear Model: Labor Share

Below, we find our diagnostic plot showing $\beta_{laborShare}$ for various thresholds. Again, we highlight in red the critical $t \in [0, 1]$ region. Upon first inspection, we find that $\beta_{laborShare} < 0$ for roughly all of the thresholds in the critical region. Although there is no indication that these results are statistically significant (i.e. by studying their p-values), showing invariance of our threshold parameter to our inferential prediction on its sign, again, is beneficial as to generalize our inference beyond the point assumption $t = 0$.

As mentioned previously, the behaviour of the threshold is interesting outside the critical interval, and should be addressed regardless of its importance to our inference. One reason this instability may arise could be due to the fact that our thresholds for every year were calculated as a function of *only* that year's data. As such, the sample from which we obtain the threshold by each level (corrosponding to each year) is very small compared to the whole dataset; as such, small changes to the threshold cause high levels of noise in our final model.

More specifically, there seems to be very similar behaviour in both coefficient plots that indicate high instability near $t = -1$. Upon further inspection, we find that this instability arises because setting the threshold to $\theta_Y = \mu_Y - \sigma_Y$ means that the threshold is classifying retirees above or below a value very close to its center, where it is most populated. To verify this, consider the visualization of the density plots in the earlier pages of the paper, and find that the 'red' distribution is approximately 1σ to the left of the 'blue' distribution. Fortunately, this is not a weakness of the model, but rather an affirmation that our inferential results only work when we are actually classifying *lost-potential* players, and not the average retiree!



Resampling Results

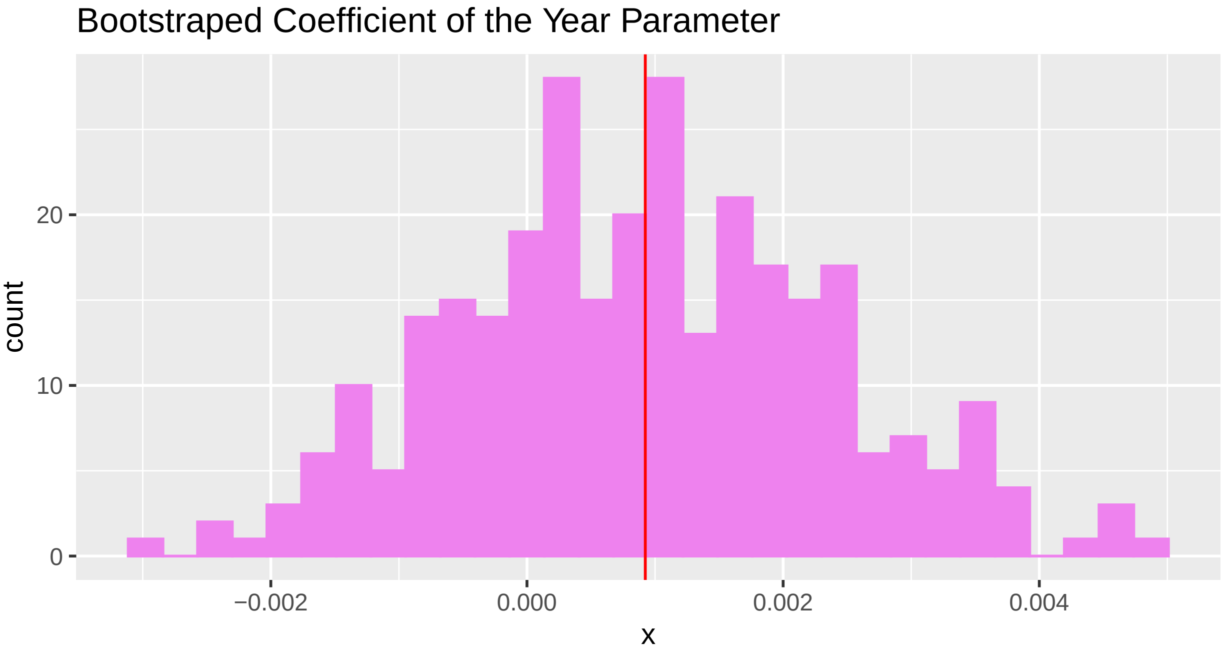
As mentioned previously, resampling is a viable alternative approach to obtaining inferential results when the data is finite and we could not generate new data. By boostrapping the rookie and retiree datasets, we obtain different thresholds and as such we obtain different values of `prop` and as such different parameter values. After bootstrapping 300 times, we obtain the following distributions of our parameters.

The Year Parameter

```
##           5%
## -0.001414456
```

```
##        95%
## 0.00344701
```

Below, we find a bootstrap distribution of β_{Year}. We obtain a mean of $\hat{\beta}_{Year} \approx 0.001$ and a 95% confidence interval of $(-0.0014, 0.0034)$. As a result, we could say that under bootstrapping, we find that our parameter estimate yields an average statistic that supports our inferential prediction ($\beta_{Year} > 0$). and indeed, most of the values in the confidence interval do so too (except on the lower tail, we may have some concerning test statistics).

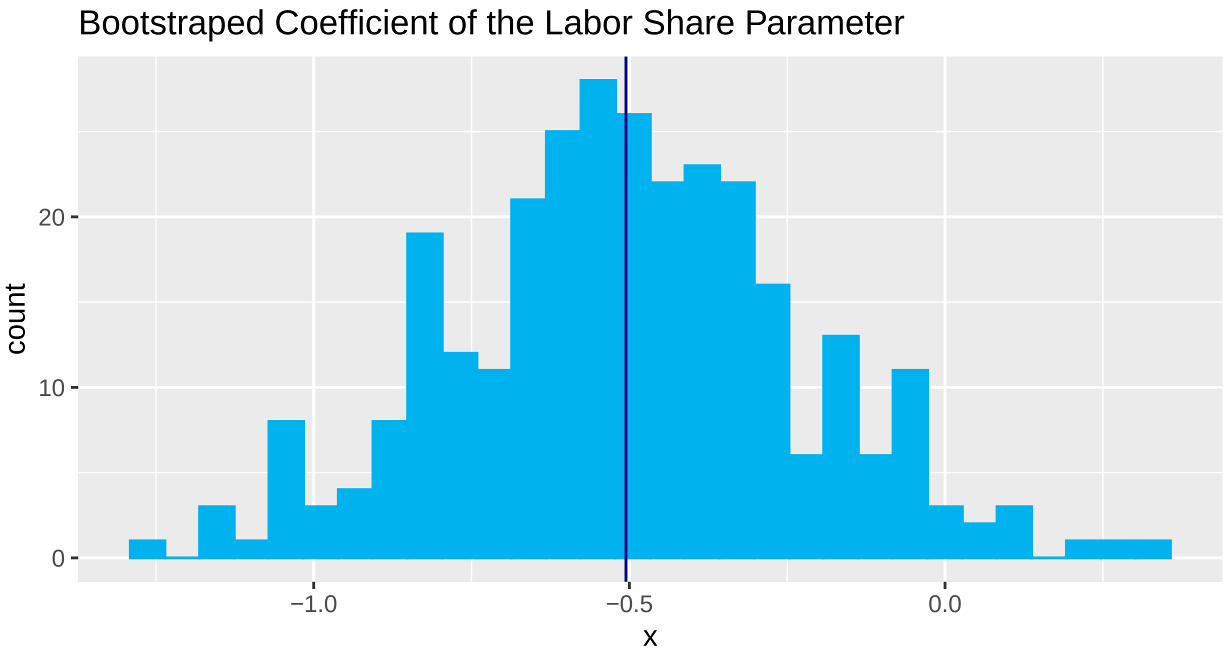


The Labor Share Parameter

```
##         5% 
## -0.9916363
```

```
##          95% 
## -0.05226886
```

Below, we find a bootstrap distribution of β_{LS}. We obtain a mean of $\hat{\beta}_{LS} = -0.5054$ and a 95% confidence interval of $(-0.992, -0.0523)$. As a result, we could say that under bootstrapping, we find that our parameter estimate yields an average statistic that supports our inferential prediction ($\beta_{LS} < 0$), and indeed, all of the values in our confidence interval do so too.



Conclusions

To summarize:

- No correlation between year and proportion of couldabeens in full data set ($\beta_{Year} \approx 0$).
- Positive but non-significant relationship between year and proportion of couldabeens when partitioned into pre- and post-*Moneyball* eras.
- Significant negative relationship between publication of *Moneyball* and proportion of couldabeens. ($p = 0.045$)
- Somewhat significant negative relationship between labor share and proportion of couldabeens in post-*Moneyball* era. ($p = 0.084$)

Previous literature has established that the 2003 CBA, which implemented the luxury tax, led to a decline in labor share. We find *some* evidence for a link between a lower labor share and a higher proportion of "couldabeens" retiring. However, we cannot definitively conclude that the MLB luxury tax has increased the proportion of couldabeens.

Code Appendix

Importing the Datasets

```
# Load rookies datasets
df_pit_rkes <- read_csv("../data/rookie-pitcher.csv")
df_pos_rkes <- read_csv("../data/rookie-position.csv")
# Load retirees datasets
df_pit_ret <- read_csv("../data/retirees-pitcher.csv")
df_pos_ret <- read_csv("../data/retirees-position.csv")
```

Wrangling the Datasets

```
# select appopriate columns from player datasets
wrangle_init <- function(dataset){
  colnames(dataset)[3] <- "WAR"
  dataset %>% select(WAR, Year)
}
```

```
# Obtain wrangled datasets
pit_rkes <- wrangle_init(df_pit_rkes)
pit_ret <- wrangle_init(df_pit_ret)
pos_rkes <- wrangle_init(df_pos_rkes)
pos_ret <- wrangle_init(df_pos_ret)
```

Finding the Couldabeen Thresholds

```
# obtain summary of WAR: median and variance
find_thresholds <- function(dataset, threshold = 0){
  dataset %>%
    group_by(Year) %>%
    summarize(mean_WAR = mean(WAR), sd_WAR = sqrt(var(WAR))) %>%
    mutate(threshold = mean_WAR + threshold*sd_WAR) %>%
    select(Year, threshold)
}
```

```
# Get thresholds in each year
pit_thresholds <- find_thresholds(pit_rkes)
pos_thresholds <- find_thresholds(pos_rkes)
```

```
head(pit_thresholds)
```

Classifying Retiree Couldabeens

```
# Appends above_threshold column to retirees dataset (checks if a retiree exceeds a threshold)
compare_thresholds <- function(dataset, summary_dataset){
  above_threshold <- rep(NA, nrow(dataset))
  for(i in 1:nrow(dataset)){
    year <- as.numeric(dataset[i,2])
    above_threshold[i] <- (dataset[i,1] > summary_dataset[year - 1968, 2])
  }
  dataset <- cbind(dataset,above_threshold)
  colnames(dataset)[3] <- "above_threshold"
  dataset
}
```

```
# See and record which players cross that year's adjusted threshold from rookie players
pit_ret <- compare_thresholds(pit_ret, pit_thresholds)
pos_ret <- compare_thresholds(pos_ret, pos_thresholds)
# Get all retired couldabeens by combining the rows
retirees <- rbind(pit_ret,pos_ret)
```

```
head(retirees)
```

Counting Couldabeens by Year

```
# Counts couldabeens in a given year
count_cbns <- function(dataset){
  dataset %>%
    group_by(Year) %>%
    summarize(cbns = sum(above_threshold))
}
```

```
# Count the retiree couldabeens by year
couldabeens <- count_cbns(retirees)
```

Counting Retirees in a Given Year

```r
# Yields the sum of retirees in two datasets (pitchers, position commonly)
total_retirees_by_yr <- function(pitchers, position){
  year_count_pitchers <- retirees_by_yr(pitchers)$retirees
  year_count_position <- retirees_by_yr(position)$retirees
  data.frame(Year = 1969:2018, retirees = year_count_position + year_count_pitchers)
}
# Counts the retirees in a single dataset
retirees_by_yr <- function(dataset){
  year_count <- dataset %>%
    group_by(Year) %>%
    summarize(retirees = n())
}
```

```r
# Find number of retirees by year
num_retirees <- total_retirees_by_yr(df_pit_ret, df_pos_ret)
num_retirees <- data.frame(retirees = num_retirees)
```

Retiree Proportion of Couldabeens

```r
# Append number of retirees that year
couldabeens <- cbind(couldabeens, num_retirees)
# Find and append proportion of couldabeens : retirees
couldabeens <- couldabeens %>% mutate(prop = cbns/retirees)
```

Splitting the data

```
couldabeens_pre <- couldabeens_t[which(couldabeens_t$postMoneyball == 0),]
couldabeens_post <- couldabeens_t %>% anti_join(couldabeens_pre)
```

The Models

```
linear_model <- function(dataset){
  linear_model <- lm(formula = prop ~ Year, data = dataset)
  linear_model
}

laborShare_model <- function(dataset){
  linear_model <- lm(formula = prop ~ labShare, data = dataset)
  linear_model
}
```

Threshold Analysis

```
#==========================================================
#                   THRESHOLD ANALYSIS
#==========================================================

# For analyzing single thresholds
isolate_threshold <- function(threshold_stack, value){
  threshold_stack %>% filter(threshold == value)
}

# Creates a stack of arrays yielding couldabeens by varying threshold levels (input a threshold vector)
create_threshold_stack <- function(ls_datasets, threshold_vec, payroll_c = payroll_c){
  # Begin stack by taking initial threshold
  curr_threshold <- as.numeric(threshold_vec[1,])
  threshold_stack <- couldabeens_by_threshold(ls_datasets, threshold = curr_threshold, payroll_c)
  # Recursively stack couldabeens with varying thresholds
  for(i in 2:nrow(threshold_vec)){
    # Obtain current threshold
    curr_threshold <- as.numeric(threshold_vec[i,])
    # Obtain couldabeens under current threshold
    curr <- couldabeens_by_threshold(ls_datasets, threshold = curr_threshold, payroll_c)
    # Recursively stack
    threshold_stack <- rbind(threshold_stack, curr)
  }
  # Standardize row names
  rownames(threshold_stack) <- 1:nrow(threshold_stack)
  # Return stack
  data.frame(threshold_stack)
}
```

```r
#==========================================================
#               COULDABEENS CLASSIFICATION
#==========================================================

# Aggregate function finds couldabeens for a given threshold in standard deviations from the mean rooki
couldabeens_by_threshold <- function(ls_datasets, payroll_c = payroll_c, threshold = 0){
  # Obtain the sd value (esentially renaming variable)
  sd <- threshold
  # Unwind datasets from list
  df_pos_rkes <- as.data.frame(ls_datasets[1])
  df_pos_ret <- as.data.frame(ls_datasets[2])
  df_pit_rkes <- as.data.frame(ls_datasets[3])
  df_pit_ret <- as.data.frame(ls_datasets[4])
  num_retirees <- as.data.frame(ls_datasets[5])
  # Obtain wrangled datasets
  pit_rkes <- wrangle_init(df_pit_rkes)
  pit_ret <- wrangle_init(df_pit_ret)
  pos_rkes <- wrangle_init(df_pos_rkes)
  pos_ret <- wrangle_init(df_pos_ret)
  # Get thresholds in each year, then smooth them
  pit_thresholds <- find_thresholds(pit_rkes, sd)
  pos_thresholds <- find_thresholds(pos_rkes, sd)
  # REMOVE SMOOTHING UNTIL NEEDED, REMEMBER TO ADD ARGUMENTS AGAIN IN FXN CALL
  #pit_thresholds <- smoothed_thresholds(pit_thresholds, w, center_weight)
  #pos_thresholds <- smoothed_thresholds(pos_thresholds, w, center_weight)
  # See and record which players cross that year's adjusted threshold from rookie players
  pit_ret <- compare_thresholds(pit_ret, pit_thresholds)
  pos_ret <- compare_thresholds(pos_ret, pos_thresholds)
  # Get retired couldabeens
  retirees <- rbind(pit_ret,pos_ret)
  couldabeens <- count_cbns(retirees)
  # Append threshold for reference
  threshold_idx <- data.frame(threshold = rep(sd, nrow(couldabeens)))
  couldabeens <- cbind(couldabeens, threshold_idx)
  # Append number of retirees that year
  couldabeens <- cbind(couldabeens, num_retirees)
  # Find and append proportion of couldabeens : retirees
  couldabeens <- couldabeens %>% mutate(prop = cbns/retirees)
  # Return dataframe
  cbind(couldabeens, payroll_c)
}
```

Threshold Analysis: Coefficients

```r
#==========================================================
#                   COEFFICIENT EXTRACTION
#==========================================================

# Find the coefficients with respect to the threshold in a whole stack
coefs_by_stack <- function(threshold_stack, threshold_vec){
  # get first stack
  first_stack <- isolate_threshold(threshold_stack, as.numeric(threshold_vec[1,]))
  # add first coefficients to the coefficientstack
  coef_stack <- coefs_by_threshold(first_stack)
  for(i in 2:nrow(threshold_vec)){
    # get current threshold
    curr_threshold <- as.numeric(threshold_vec[i,])
    # current stack
    curr_stack <- isolate_threshold(threshold_stack, curr_threshold)
    curr_coefs <- coefs_by_threshold(curr_stack)
    # recurisively stack coef arrays in each threshold
    coef_stack <- rbind(coef_stack, curr_coefs)
  }
  coef_stack
}

# Extract the coefficients of the post-rule era coefficients in the current threshold's stack for both
# the year model and the laborShare model.
coefs_by_threshold <- function(curr_stack){
  # extract current threshold
  curr_threshold <- curr_stack[1,"threshold"]
  # fit the linear models for current threshold
  lm_post <- linear_model(postrule(curr_stack)) # `prop ~ Year`
  lm_labShare <- laborShare_model(postrule(curr_stack))
  # coefficients of the respective models
  coef_post <- data.frame(coef = lm_post$coefficients[2],
                          model = "year",
                          threshold = curr_threshold)
  coef_labShare <- data.frame(coef = lm_labShare$coefficients[2],
                              model = "labor",
                              threshold = curr_threshold)
  # stack the coefficient array
  coefs_curr <- rbind(coef_post, coef_labShare)
  # standardize rownames
  rownames(coefs_curr) <- 1:nrow(coefs_curr)
  coefs_curr
}
```

Bootstrapping

```r
# Bootstrap the rookie datasets and return couldabeens
couldabeens_bootstrapped <- function(bootstrap_retirees = F){
  # Obtain bootstrapped rookies
  BOOT_pit_rkes <- sample_n(pit_rkes, replace = T, size = nrow(pit_rkes))
  BOOT_pos_rkes <- sample_n(pos_rkes, replace = T, size = nrow(pos_rkes))
  # Bootstrap the retirees?
  if(bootstrap_retirees){
    BOOT_pit_ret <- sample_n(pit_ret, replace = T, size = nrow(pit_ret))
    BOOT_pos_ret <- sample_n(pos_ret, replace = T, size = nrow(pos_ret))
  } else{
    BOOT_pit_ret <- pit_ret
    BOOT_pos_ret <- pos_ret
  }
  # Get thresholds in each year
  BOOT_pit_thresholds <- find_thresholds(BOOT_pit_rkes)
  BOOT_pos_thresholds <- find_thresholds(BOOT_pos_rkes)
  # See and record which players cross that year's adjusted threshold from rookie players
  if(bootstrap_retirees){
    pit_ret <- compare_thresholds(BOOT_pit_ret, BOOT_pit_thresholds)
    pos_ret <- compare_thresholds(BOOT_pos_ret, BOOT_pos_thresholds)
  } else{
    BOOT_pit_ret <- compare_thresholds(BOOT_pit_ret, BOOT_pit_thresholds)
    BOOT_pos_ret <- compare_thresholds(BOOT_pos_ret, BOOT_pos_thresholds)
  }
  # Get couldabeen retirees
  BOOT_couldabeens <- count_cbns(rbind(BOOT_pit_ret,BOOT_pos_ret))
  # Append number of retirees that year
  BOOT_couldabeens <- cbind(BOOT_couldabeens, num_retirees)
  # Find and append proportion of couldabeens : retirees
  BOOT_couldabeens <- BOOT_couldabeens %>% mutate(prop = cbns/retirees)
  # Return dataframe
  cbind(BOOT_couldabeens[,-c(2,3)], payroll_c)
}
```

```r
# Extract the coefficients of the post-rule era coefficients in the current bootstrap's stack for both
# the year model and the laborShare model.
coefs_bootstrapped <- function(curr_stack){
  # fit the linear models for current bootstrapping
  lm_post <- linear_model(postrule(curr_stack)) # `prop ~ Year`
  lm_labShare <- laborShare_model(postrule(curr_stack))
  # coefficients of the respective models
  coef_post <- data.frame(coef = lm_post$coefficients[2],
                          model = "year")
  coef_labShare <- data.frame(coef = lm_labShare$coefficients[2],
                              model = "labor")
  # stack the coefficient array
  coefs_curr <- rbind(coef_post, coef_labShare)
  # standardize rownames
  rownames(coefs_curr) <- 1:nrow(coefs_curr)
  coefs_curr
}
```

```r
#==========================================================
#                   RUNNING THE BOOTSTRAP
#==========================================================
run_bootstrap <- T
# Number of bootstraps
if(run_bootstrap){
  B <- 100
  bootstrap_coef_YR <- rep(NA, B)
  bootstrap_coef_LB <- rep(NA, B)
  for(i in 1:B){
    # Bootstrap the couldabeens
    curr_couldabeens <- couldabeens_bootstrapped()
    curr_coefs <- coefs_bootstrapped(curr_couldabeens)
    bootstrap_coef_YR[i] <- curr_coefs[1,1]
    bootstrap_coef_LB[i] <- curr_coefs[2,1]
  }
}
```

References

(1) https://stathead.com/baseball/

(2) Bradbury, John Charles. "What Explains Labor's Declining Share of Revenue in Major League Baseball?" (2019).

(3) https://blogs.fangraphs.com/mlbs-evolving-luxury-tax/

(4) Lewis, Michael. *Moneyball: The Art of Winning an Unfair Game.* New York: Norton, 2003.

(5) Hayes, Hannah. "What will Nate Silver do next?" *uchicago.edu.*

(6) Birnbaum, Phil. "Asking the Right Qustions." *SABR.org.*