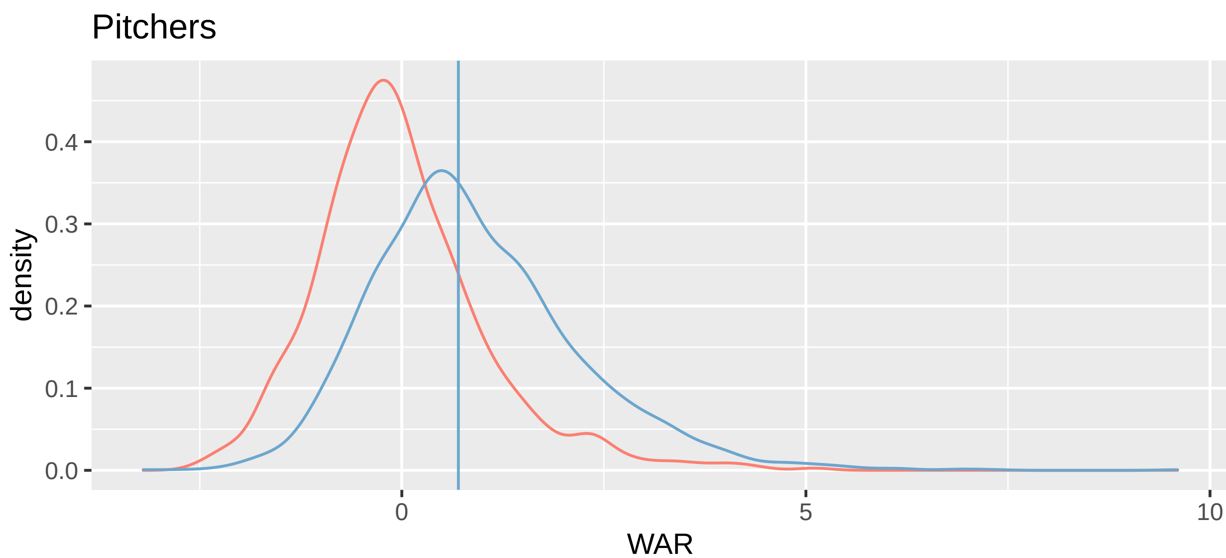
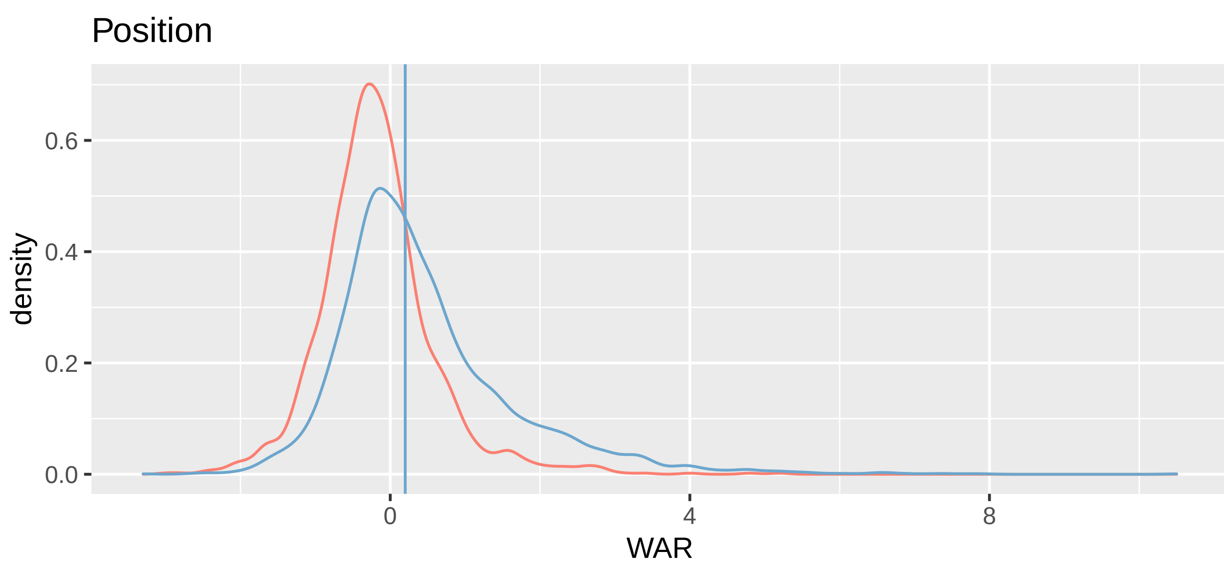
Modeling

Group 6

The Couldabeen Classification Problem





Counting Couldabeens

```r
#====================================#
#        Counting: Couldabeens      #
#====================================#
# Combine the threshold-classified retiree datasets
retirees <- rbind(pit_ret,pos_ret)
# Count couldabeens
couldabeens <- count_cbns(retirees)
```

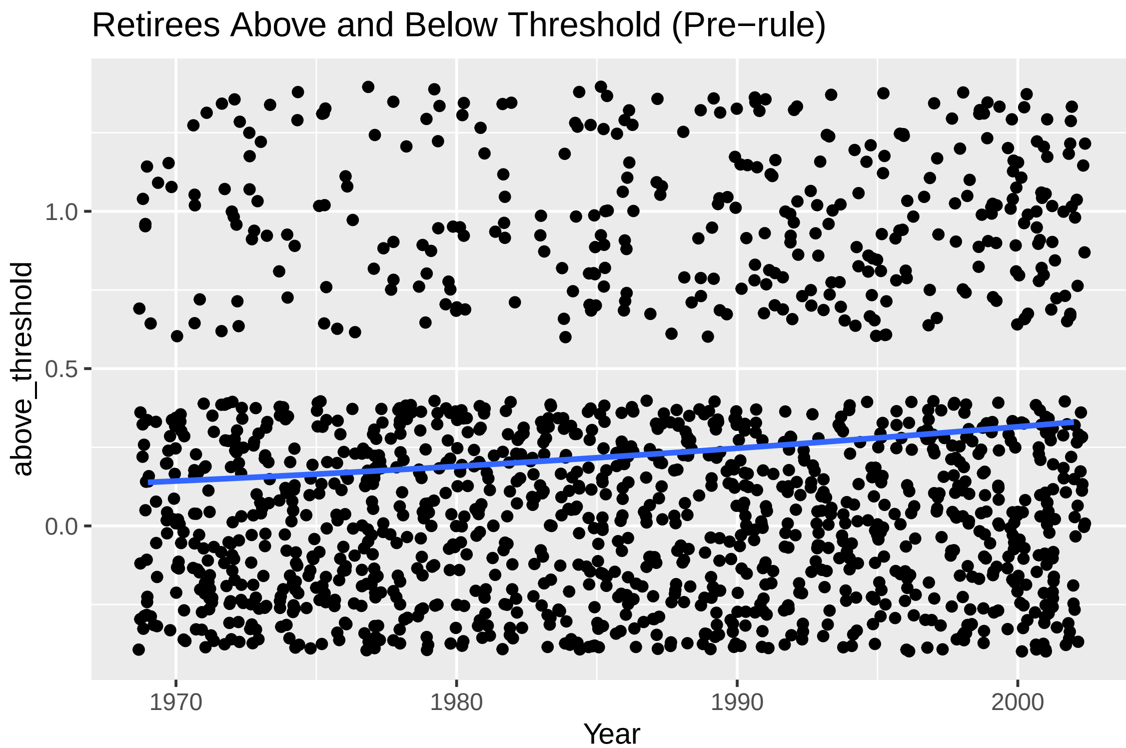
Our retirees dataframe looks like this:

```
##    WAR Year above_threshold
## 1  1.8 1972            TRUE
## 2  0.1 1974           FALSE
## 3  0.3 1976           FALSE
## 4 -0.5 1977           FALSE
## 5  0.4 1977           FALSE
## 6 -1.8 1974           FALSE
```

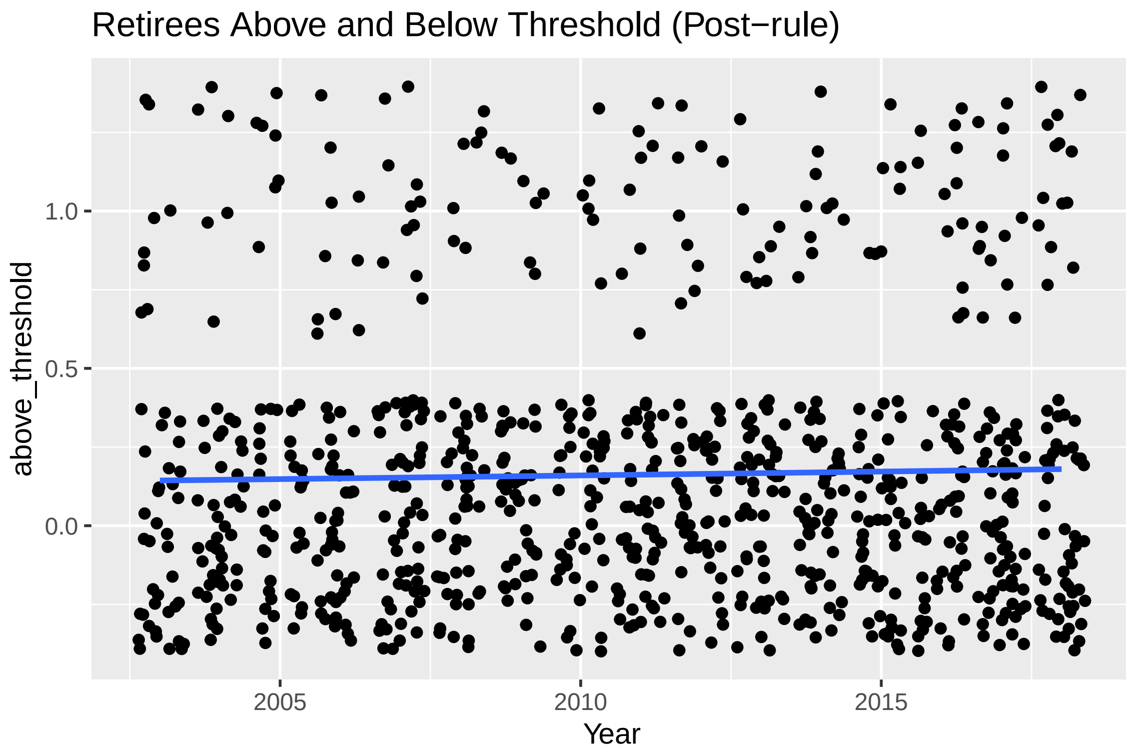
Our couldabeens dataframe looks like this:

```
## # A tibble: 6 x 2
##    Year  cbns
##   <dbl> <int>
## 1  1969     7
## 2  1970     3
## 3  1971     6
## 4  1972    10
## 5  1973     9
## 6  1974     6
```

First Look: A Logistic Model




```
##
## Call:
## glm(formula = above_threshold ~ Year, family = "binomial", data = dataset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8949  -0.7642  -0.6479  -0.5547   1.9883
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -68.617412  12.853177  -5.339 9.37e-08 ***
## Year          0.033921   0.006465   5.247 1.55e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1583.3  on 1470  degrees of freedom
## Residual deviance: 1554.9  on 1469  degrees of freedom
## AIC: 1558.9
##
## Number of Fisher Scoring iterations: 4
```




```
##
## Call:
## glm(formula = above_threshold ~ Year, family = "binomial", data = dataset)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.6297   -0.6096   -0.5900   -0.5617    1.9697
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -37.46198   40.21840  -0.931    0.352
## Year          0.01781    0.02000   0.891    0.373
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 781.43  on 880  degrees of freedom
## Residual deviance: 780.63  on 879  degrees of freedom
## AIC: 784.63
##
## Number of Fisher Scoring iterations: 4
```

Computing Retiree Proportions

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

Here is what the proportion-appended couldabeen dataframe looks like:

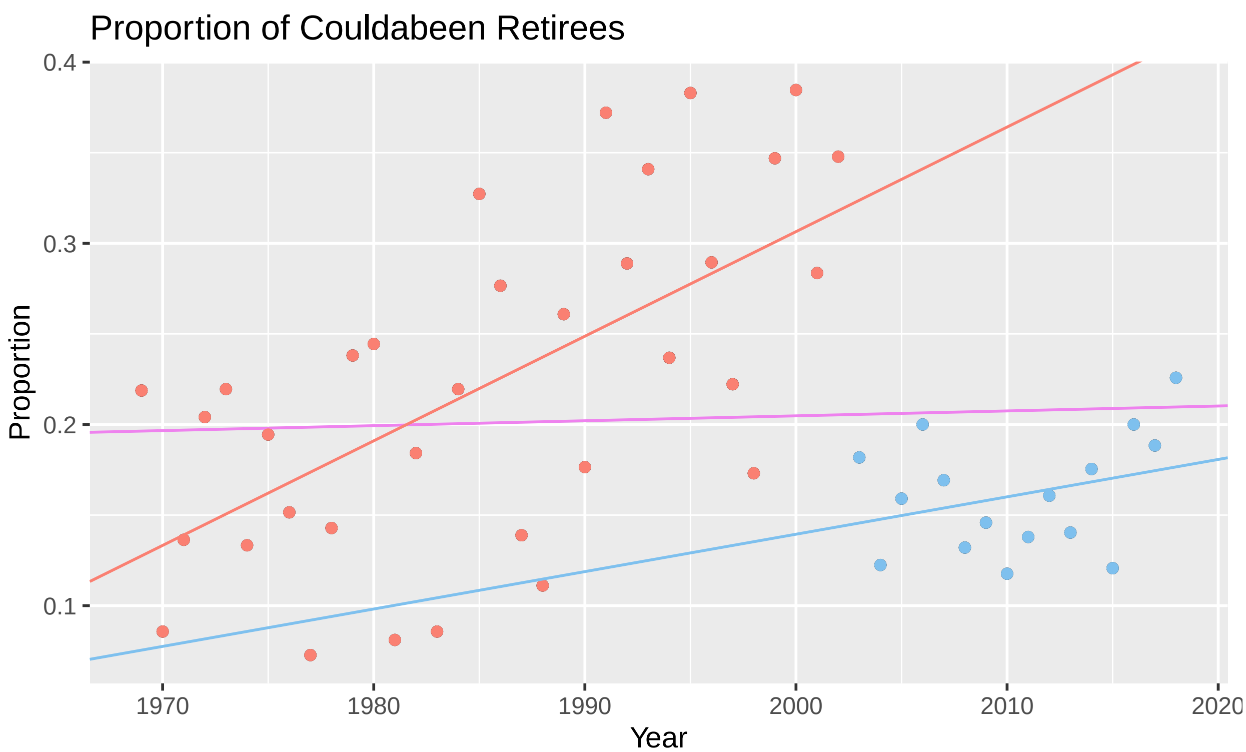
```
##   Year cbns retirees       prop
## 1 1969    7       32 0.21875000
## 2 1970    3       35 0.08571429
## 3 1971    6       44 0.13636364
## 4 1972   10       49 0.20408163
## 5 1973    9       41 0.21951220
## 6 1974    6       45 0.13333333
```

Year as Predictor: Linear Modeling

```
#===================#
#     Modeling     #
#===================#
# Partition dataset into years before and after rule
couldabeens_pre <- prerule(couldabeens)
couldabeens_post <- postrule(couldabeens)
# Obtain linear model for pre-rule years
model_pre <- linear_model(couldabeens_pre)
coefs_pre <- model_pre$coefficients
# Obtain linear model for post-rule years
model_post <- linear_model(couldabeens_post)
coefs_post <- model_post$coefficients
# Obtain linear model for all years
model_comp <- linear_model(couldabeens)
coefs_comp <- model_comp$coefficients
```

Couldabeens: A Comprehensive Look

```
##
## Call:
## lm(formula = prop ~ I(Year), data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12581 -0.06211 -0.01833  0.04358  0.17984
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.3379371  1.6395066  -0.206    0.838
## I(Year)      0.0002714  0.0008224   0.330    0.743
##
## Residual standard error: 0.08392 on 48 degrees of freedom
## Multiple R-squared:  0.002263,   Adjusted R-squared:  -0.01852
## F-statistic: 0.1089 on 1 and 48 DF,  p-value: 0.7429
```



Couldabeens: Pre-rule Era (1969-2002)

```
##
## Call:
## lm(formula = prop ~ I(Year), data = dataset)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.126056 -0.044806  0.005781  0.053314  0.117608
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.238735   2.549750  -4.408  0.00011 ***
## I(Year)       0.005773   0.001284   4.495 8.56e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07346 on 32 degrees of freedom
## Multiple R-squared:  0.3871, Adjusted R-squared:  0.3679
## F-statistic: 20.21 on 1 and 32 DF,  p-value: 8.56e-05
```

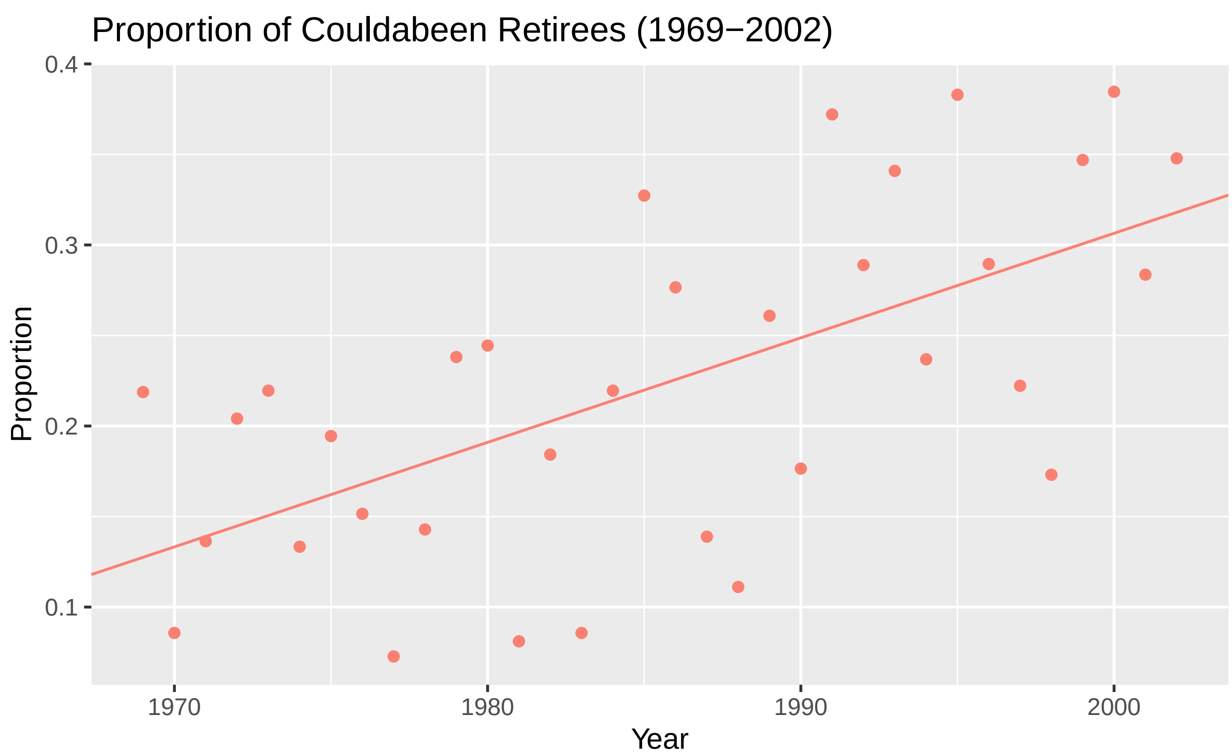


Figure 1: Proportion of Retirees who were Coulabeens prior to the implementation of the Luxury Tax

Couldabeens: Post-rule Era (2003-2018)

```
##
## Call:
## lm(formula = prop ~ I(Year), data = dataset)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.049689 -0.024453  0.001825  0.018410  0.049237
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.987717   3.481582  -1.145    0.271
## I(Year)      0.002064   0.001732   1.192    0.253
##
## Residual standard error: 0.03193 on 14 degrees of freedom
## Multiple R-squared:  0.09209,    Adjusted R-squared:  0.02724
## F-statistic:  1.42 on 1 and 14 DF,  p-value: 0.2532
```

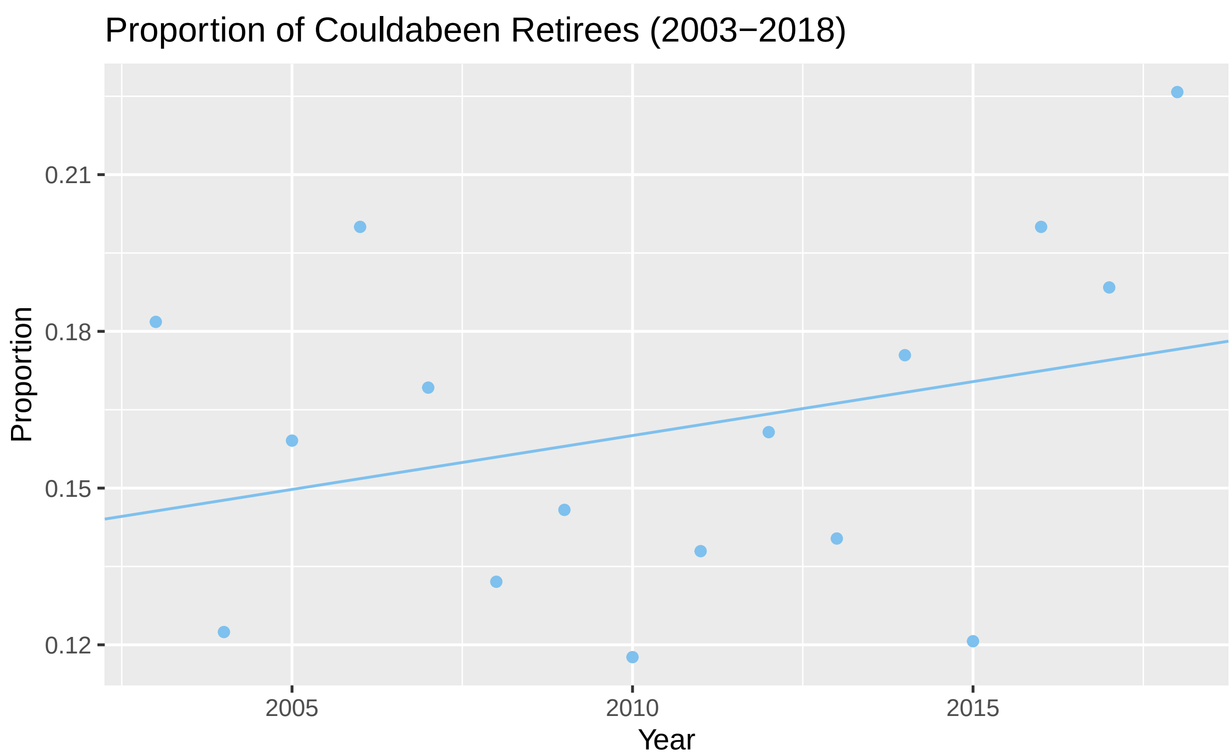


Figure 2: Proportion of Retirees who were Coulabeens after the implementation of the Luxury Tax

Quadratic Regression Model

Predicting WAR

```r
# Simpler implementation?
#plot + stat_smooth(mapping = aes(x = Year, y = prop), data = couldabeens_post, method = "lm", formula
```

```r
#pitchers <- df_pit_rkes
#pitchers1 <- drop_na(pitchers)
#pitchers1_trn <- pitchers1 %>% sample_frac(0.7)
#pitchers1_tst <- pitchers1 %>% anti_join(pitchers1_trn)
```

```r
#library(leaps)
#ss1 <- regsubsets(WAR~. - Rk - Player, data = pitchers1_trn, nvmax = 49, method = "forward")
```

```r
# remove troublesome variables
wrangle_lm <- function(dataset){
  dataset[,-c(1,2,5,6,7,8,25,26)] %>% drop_na()
}
dataset <- wrangle_lm(df_pit_rkes)
# select significant variables
select_vars <- function(dataset){
  dataset[,c(1,3,4,12,13,14,19,26,30,32,37,38,40)] %>% drop_na()
}
pitchers <- select_vars(dataset)
```

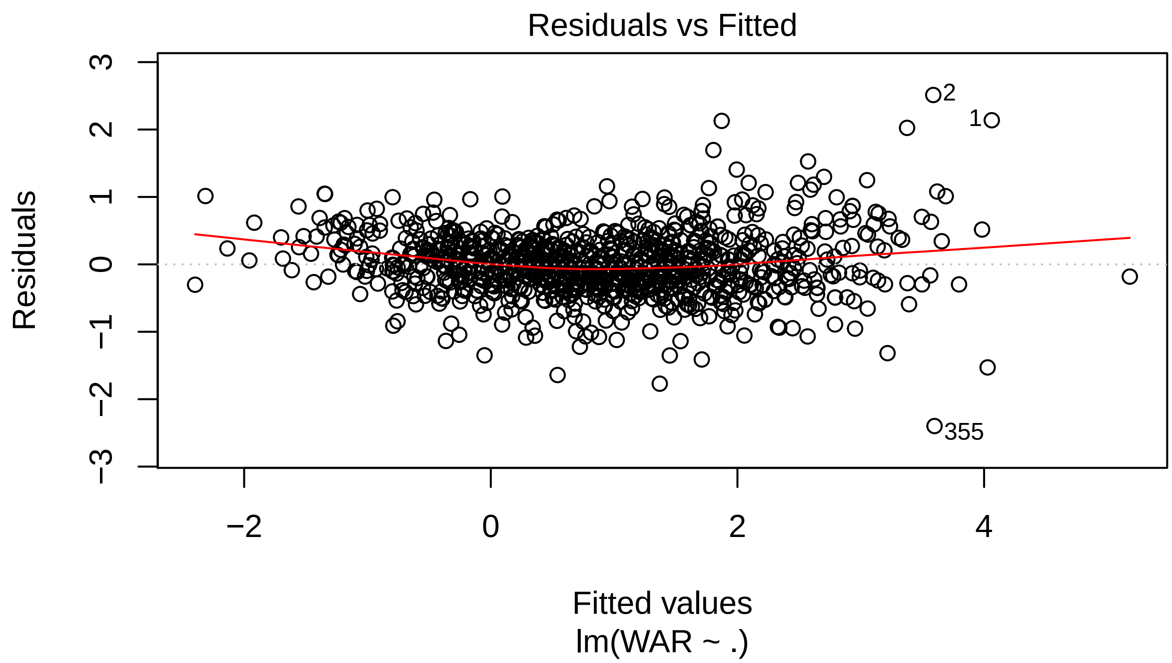
```r
pitchers_trn <-  pitchers %>% sample_frac(0.7)
pitchers_tst <- pitchers %>% anti_join(pitchers_trn)
```

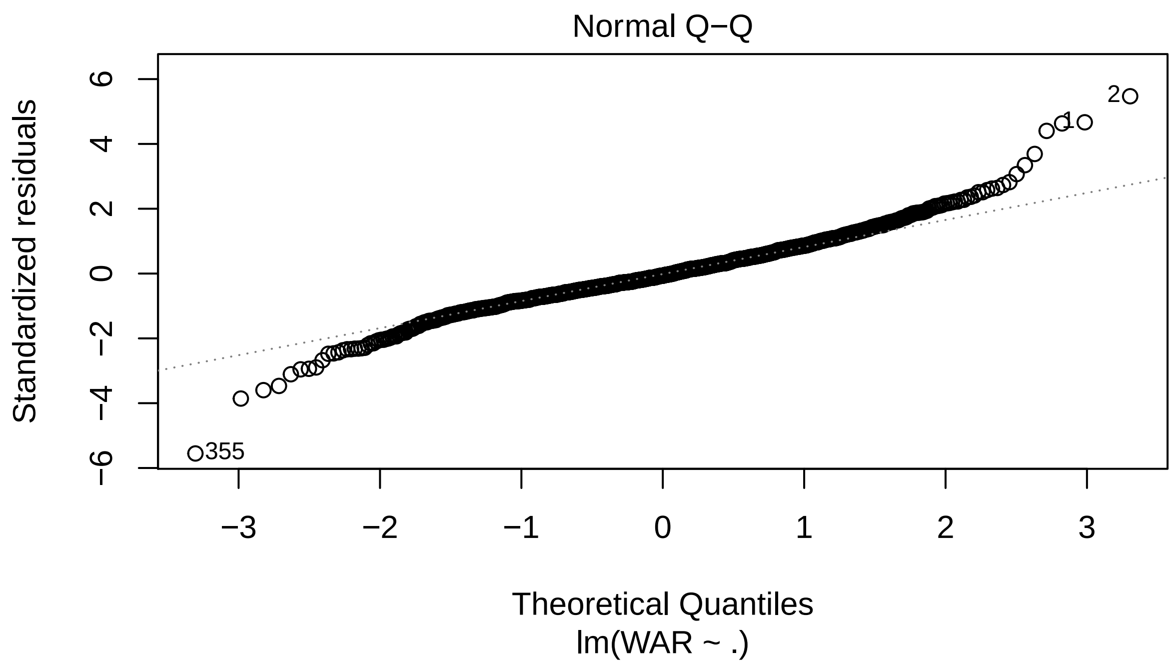
```r
linear_model <- lm(WAR ~ ., data = pitchers)
summary(linear_model)
```

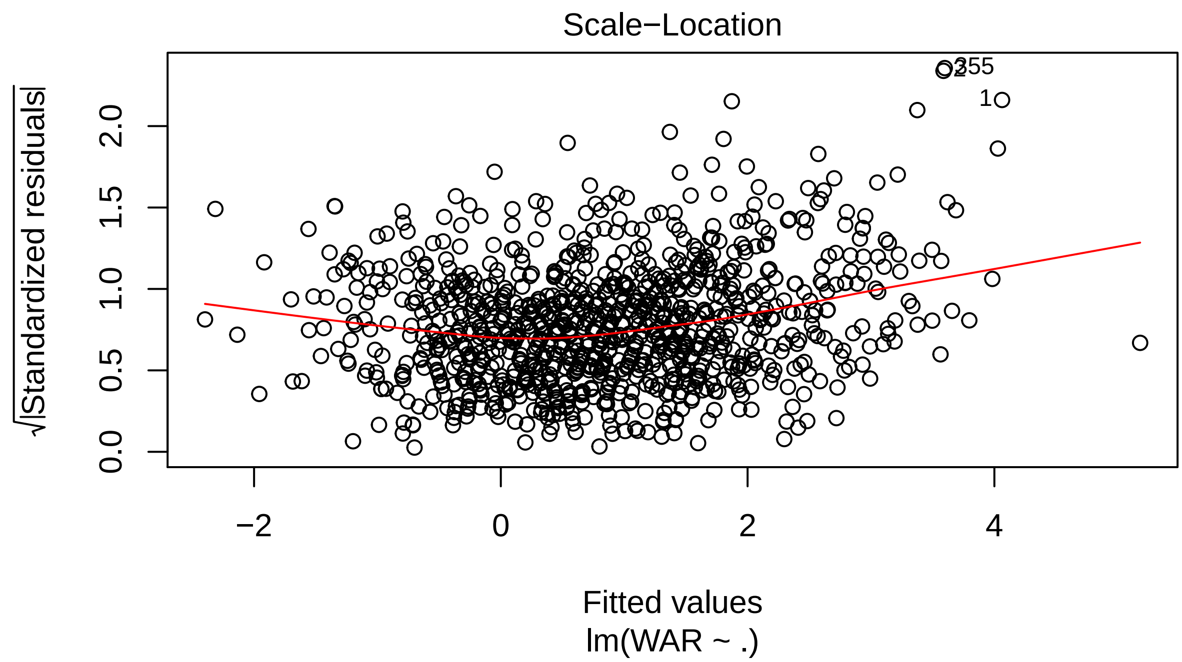
```
##
## Call:
## lm(formula = WAR ~ ., data = pitchers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.39814 -0.26531 -0.03036  0.25217  2.51213
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.4214065  0.3080594  -1.368  0.17163
## G            0.0088687  0.0015426   5.749 1.18e-08 ***
## GS           0.0847422  0.0056713  14.942  < 2e-16 ***
## H            0.0292305  0.0015373  19.014  < 2e-16 ***
## R           -0.1019976  0.0055552 -18.361  < 2e-16 ***
## ER           0.0321035  0.0061658   5.207 2.32e-07 ***
## `ERA+`       0.0047342  0.0007226   6.552 8.93e-11 ***
## IBB          0.0148943  0.0076841   1.938  0.05285 .
## GDP          0.0080564  0.0043426   1.855  0.06385 .
## CS           0.0278851  0.0084695   3.292  0.00103 **
## OBP          5.0432213  0.9027225   5.587 2.95e-08 ***
## SLG          6.7642786  0.6565909  10.302  < 2e-16 ***
## `OPS+`      -0.0440980  0.0024499 -18.000  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

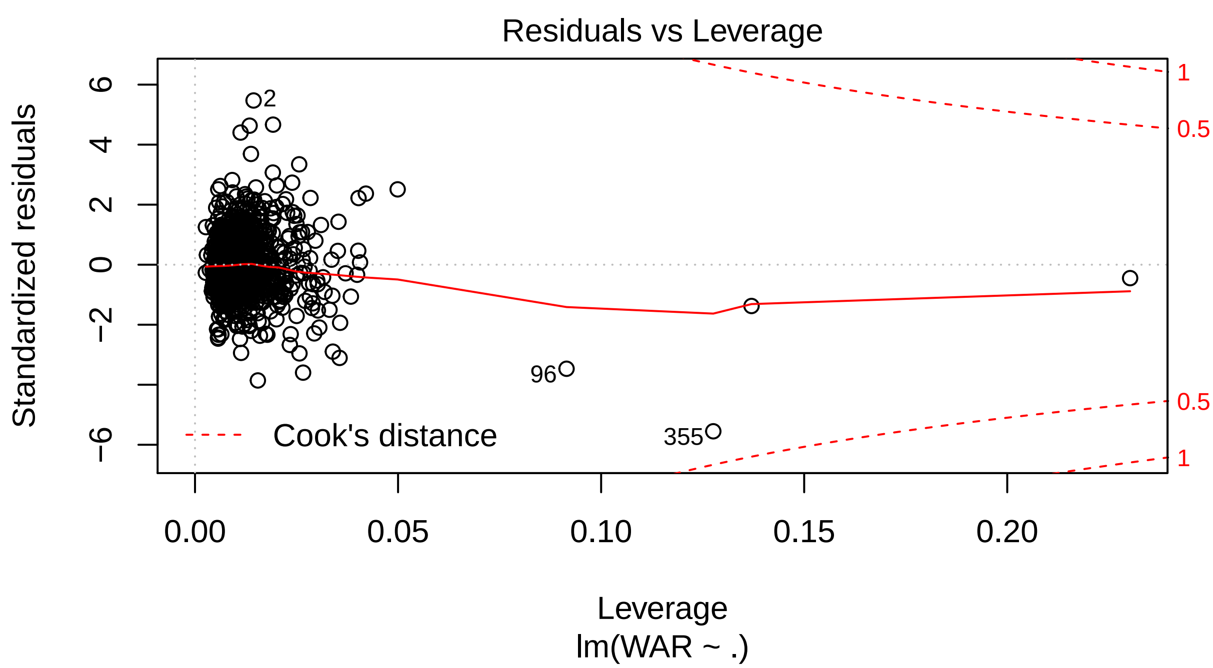
```
##
## Residual standard error: 0.4625 on 1042 degrees of freedom
## Multiple R-squared:  0.8521, Adjusted R-squared:  0.8504
## F-statistic: 500.2 on 12 and 1042 DF,  p-value: < 2.2e-16
```

```r
plot(linear_model)
```










```r
predictions <- predict(linear_model, pitchers_tst)
test_MSE <- mean(predictions - pitchers_tst$WAR)^2
test_MSE
```

```
## [1] 0.0009213926
```

```r
#data.frame(model = 1:50, adjr2 = summary(ss1)$adjr2, rss = summary(ss1)$rss, cp = summary(ss1)$cp)%>%
```