Colombia-2016 Peace Referendum

Guillermo Fremd Kanovich

6/27/2020

## WORK IN PROGRESS, THIS IS SIMPLY A DRAFT, WHERE I SAVE MY NOTES ON THE CASE

## Colombia - Intro

The Effect of Political Violence on Affected Communities’ Willingness to Compromise: The Case of the 2016 Peace Referendum in Colombia.

In August 2016, after more than 50 years of violence, the Colombian government reached a final negotiated settlement with the militan organization Revolutionary Armed Forces of Colombia (FARC), which was taken to the polls to be ratified by the Colombian population in October of the same year.

Baseline question:

“How did the level of terrorism in each municipality in Colombia affect the level of support for the 2016 peace agreement with the DARC?”

Were those municipalities that suffered higher levels of violence less willing to support the final peace agreement with the terrorist organization, or more? Can we identify a relationship (maybe non-linear) between the level of violence suffered in each municipality and its level of support for the peace referendum?

The data for the dependent variable “Level of support for the peace agreement referendum” was obtained from the UN Office for the Coordination of Humanitarian Affairs (OCHA) country office in Colombia. (<https://data.humdata.org/dataset/resultados-plebiscito-por-la-paz-en-colombia>). This variable simply indicates what percentage voted in favour of the proposed Peace Agreement with the FARC.

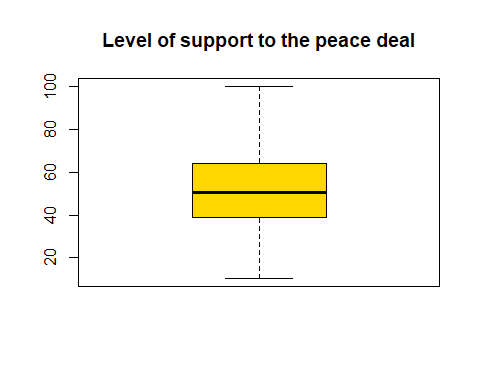
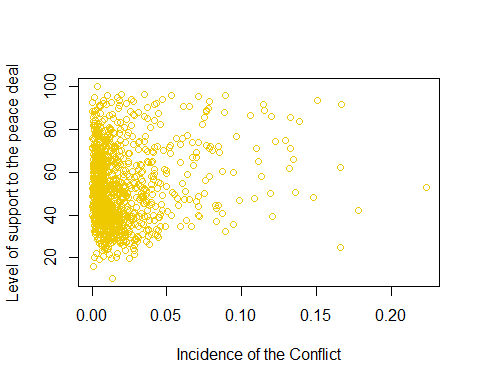
The independent variable is “Armed Conflict Incidence Level Index”, which was developed by Colombia’s National Planning Department. <https://colaboracion.dnp.gov.co/CDT/Poltica%20de%20Vctimas/Construcci%C3%B3n%20de%20Paz/Documento%20%C3%ADndice%20de%20incidencia%20del%20conflicto%20armado.pdf>

My dataset comprises some other variables, including a dicotomic variable which indicates whether prior to the referendum there existed active armed groups in the Municipality.

Let’s divide our data in two subsets: one including those municipalities where armed groups were active, and one comprising those that where free of armed groups:

## Initial exploration

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 10.43 38.66 50.38 52.43 63.92 100.00

Hmm that scatter plot is not very clear, there is no clearcut relatinship between the variables

What is the correlation between my two main variables, Incidence of the Conflict and Level of support for the Peace Agreement?

## [1] 0.1422204

Correlation: 0.14… not very high.

What if we separate those municipalities with armed groups from those without?

## [1] 0.04527594

## [1] 0.2181063

Clearlly, the correlation between the level of incidence of the conflict and the level of support for the Peace Agreement is signficantly higher in those municipalities where armed groups are present (0.22), than that of municipalities without active armed groups (0.05).

## Dividing municipalities in groups by level of incidence of the conflict

My dataset has an additional variable categorising the municipalities in 5 groups (from 1 to 5) according to the level of incidence of the conflict.This division of municipalities in 5 groups was not done by me, it is a classification done by Colombia’s National Planning Department. Let’s see if we can identify a (statistically significant) different level of support for the referendum in each of these groups.

Let’s first visualise the data, and then run a One-way anova test.

## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 3.0.1 v dplyr 0.8.3  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

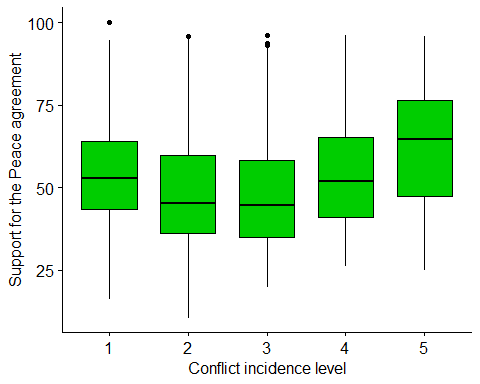
## Warning: package 'tibble' was built under R version 3.6.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

## Warning: package 'rstatix' was built under R version 3.6.3

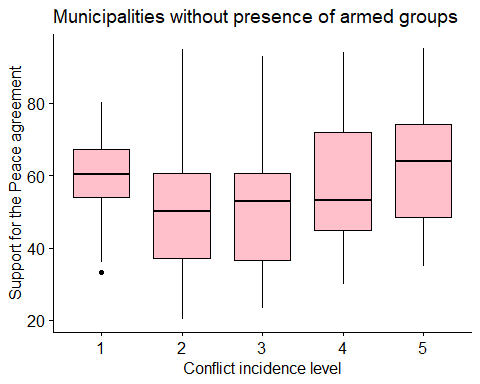
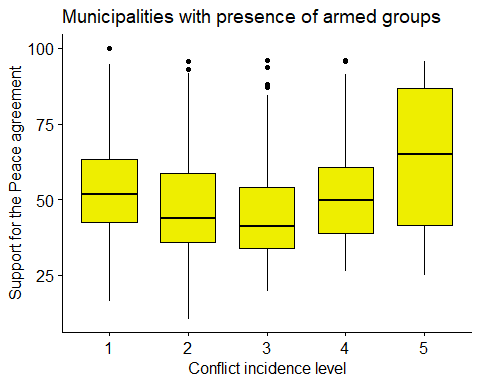
##   
## Attaching package: 'rstatix'

## The following object is masked from 'package:stats':  
##   
## filter



This appears to show something interesting: it seems that those municipalities that had a very high and very low incidence of the concflit, where more supportive for the peace referendum, than those municipalities where the conflict had an intermediate level of incidence.

Let’s now try to see if the same “shame” is visile both if we consider separatelly the municipalities where armed groups are present, and those without presente of armed groups:



Yep..! We observe the same dinamic both in municipalities with and without armed groups.

Let’s see now if the differences that we identify between the groups are statistically significant, using a one-way anova test and a Tukey’s test. (To conduct anovas test we should searh for outliers, and we should make sure that certain assumptions are not violated, but I will not do so in this excercise).

## Df Sum Sq Mean Sq F value Pr(>F)   
## ConflictCategory 1 2004 2003.5 6.848 0.00899 \*\*  
## Residuals 1119 327382 292.6   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## # A tibble: 10 x 8  
## term group1 group2 estimate conf.low conf.high p.adj p.adj.signif  
## \* <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 Conflic~ 1 2 -5.88 -9.11 -2.65 7.62e- 6 \*\*\*\*   
## 2 Conflic~ 1 3 -6.22 -10.7 -1.74 1.47e- 3 \*\*   
## 3 Conflic~ 1 4 0.253 -4.74 5.25 1.00e+ 0 ns   
## 4 Conflic~ 1 5 8.83 3.27 14.4 1.52e- 4 \*\*\*   
## 5 Conflic~ 2 3 -0.341 -4.78 4.10 1.00e+ 0 ns   
## 6 Conflic~ 2 4 6.13 1.18 11.1 6.65e- 3 \*\*   
## 7 Conflic~ 2 5 14.7 9.18 20.2 6.48e-12 \*\*\*\*   
## 8 Conflic~ 3 4 6.48 0.629 12.3 2.13e- 2 \*   
## 9 Conflic~ 3 5 15.1 8.71 21.4 1.31e- 9 \*\*\*\*   
## 10 Conflic~ 4 5 8.58 1.87 15.3 4.52e- 3 \*\*

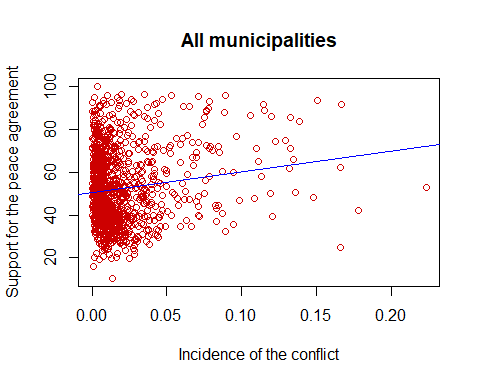
The Anova test shows that there exist statisticall signifcant differences between the groups. In other words, we can rule out the null hypothesis: that all the 5 groups presented the same level of support for the referendum

The Tukey test shows that the only differences that are not statistically significative are the difference between groups 1 and 4, and between 2 and 3. This is coherent with a U-shaped dinamic as the one we saw in the plots before: municipalitites in Groups 2 and 3, which suffered the conflict with the FARC in intermediate levels, presented a lower level of support for the Peace Agreement than municipalitites in Groups 1 and 4, where the incidence of the conflict was Low and High, respectively. Finally, municipalitites in Group 5 (those municipalitiies where the conflict with the FARC had a Very High level of incidence), presented the highest levels of support for the peace deal.

## Regressions

What about trying to fit simple linear regressios between the variables? Let’s do this, first, with the complete dataset.

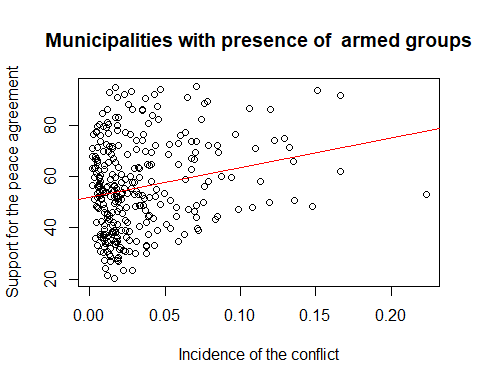
##   
## Call:  
## lm(formula = Votedyes ~ ConflictIndex, data = rawdata)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -41.717 -13.774 -1.821 11.735 49.074   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 50.5804 0.6363 79.493 < 2e-16 \*\*\*  
## ConflictIndex 96.5633 20.0908 4.806 1.75e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 16.98 on 1119 degrees of freedom  
## Multiple R-squared: 0.02023, Adjusted R-squared: 0.01935   
## F-statistic: 23.1 on 1 and 1119 DF, p-value: 1.747e-06



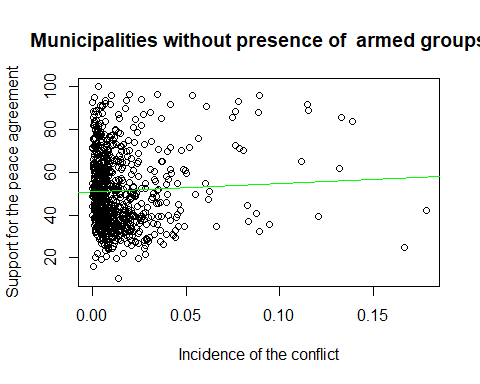
While te regression shows that there is a statistically significative positive relationship between the Incidence of the conflict in each municipality and the level of support for the referendum, the R2 shows that this (too simple) model explains solely a 2% of the variance. It is very poor.

Now let’s try to see what happens if we separate those municipalities were armed groups were present at the time of the referendum, from those municipalitites where there were no armed groups, and see if we have the same results.

##   
## Call:  
## lm(formula = Votedyes ~ ConflictIndex, data = datawitharmed)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -33.661 -15.328 -1.358 12.840 40.837   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 52.057 1.382 37.671 < 2e-16 \*\*\*  
## ConflictIndex 115.248 29.773 3.871 0.000133 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 17.2 on 300 degrees of freedom  
## Multiple R-squared: 0.04757, Adjusted R-squared: 0.0444   
## F-statistic: 14.98 on 1 and 300 DF, p-value: 0.0001331



##   
## Call:  
## lm(formula = Votedyes ~ ConflictIndex, data = datawithoutarmed)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -40.744 -13.071 -2.082 11.237 49.231   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 50.6283 0.7288 69.463 <2e-16 \*\*\*  
## ConflictIndex 39.2550 30.3020 1.295 0.196   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 16.82 on 817 degrees of freedom  
## Multiple R-squared: 0.00205, Adjusted R-squared: 0.0008284   
## F-statistic: 1.678 on 1 and 817 DF, p-value: 0.1955



Interesting! Thanks to these regressions, we can see that actually the level of incidence of the conflict was NOT a statistically significant in those municipalites where, at the time of the referendum, there was no presence of armed groups.

## Quadratic regressions?

What happens if we try to fit a quadratic model?

##   
## Call:  
## glm(formula = Votedyes ~ poly(ConflictIndex, 2), data = rawdata)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -44.78 -13.62 -1.71 11.55 48.80   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 52.4268 0.5073 103.353 < 2e-16 \*\*\*  
## poly(ConflictIndex, 2)1 81.6233 16.9837 4.806 1.75e-06 \*\*\*  
## poly(ConflictIndex, 2)2 15.5374 16.9837 0.915 0.36   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 288.4453)  
##   
## Null deviance: 329386 on 1120 degrees of freedom  
## Residual deviance: 322482 on 1118 degrees of freedom  
## AIC: 9536.2  
##   
## Number of Fisher Scoring iterations: 2

The quadratic regression above shows that the quadratic term is not statstically significant. Thus, it appears that the relationship between the variables have a linear shape.

## Logistic regression

Now, let’s try to fit a logistic regression to see if we could predict, using several variables, whether more than 50% of the voters supported the peace agreement in the referendum. We will only use the subset that includes municipalitites with presence of armed groups, because we have seen before that in those municipalitites the level of incidence of the conflict was a statistically significant factor.

Before moving forward, let’s create a dicotomic variable indicing whether more thab 50% of the voters supported the peace deal in the referendum: 0 means that less than 50% voted for the agreement, and 1 means than more than 50% supported it.

We should also divide the data in two subsets: one to fit the quadratic regression, and one to test it:

Once we have divided the data into training and testing datasets, let’s fit the quadratic regression using the training dataset, and let’s see how well it performs with the testing data.

We will use 3 variables to fit the quadratic regression: Level of Incidence of the Conflict (the one we used before), as well as Population and Poverty Index:

Let’s now see how many “correct” answers our predictor returned:

## truth  
## prediction 0 1  
## 0 13 12  
## 1 17 49

##   
## Call:  
## glm(formula = voteabove50 ~ ConflictIndex + Population + PovertyIndex2013,   
## family = binomial, data = traindata)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8579 -1.1765 0.7525 1.0721 1.3025   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.490e+00 8.485e-01 2.934 0.00335 \*\*  
## ConflictIndex 3.835e+00 4.672e+00 0.821 0.41183   
## Population 3.637e-07 3.752e-07 0.969 0.33238   
## PovertyIndex2013 -3.388e+00 1.161e+00 -2.917 0.00353 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 287.33 on 210 degrees of freedom  
## Residual deviance: 275.72 on 207 degrees of freedom  
## AIC: 283.72  
##   
## Number of Fisher Scoring iterations: 4

## [1] 0.6813187

## [1] 0.3186813

Uur model clasiffied more than 68% of the cases correctly. However, from the three variables included in it, only poverty index appears to be statistically significant. The negative term indicates that the higher the Poverty Index in a municipality, the less likely such municipality was to support the refrendum.

## Classification Tree

Let´s now fit a classification tree and see if we could use it as an efficient prediction tool to forecast whether a municipality would have presented a level of support for the peace aggrement above, or below, 50%.

library(tree)

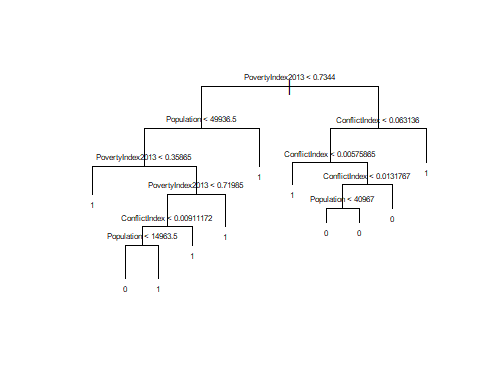
## Warning: package 'tree' was built under R version 3.6.2

## Registered S3 method overwritten by 'tree':  
## method from  
## print.tree cli

set.seed(1)  
traindata$voteabove50=factor(traindata$voteabove50)  
testdata$voteabove50=factor(testdata$voteabove50)  
  
traintree <- tree(voteabove50~ConflictIndex+Population+PovertyIndex2013,data=traindata)  
summary(traintree)

##   
## Classification tree:  
## tree(formula = voteabove50 ~ ConflictIndex + Population + PovertyIndex2013,   
## data = traindata)  
## Number of terminal nodes: 11   
## Residual mean deviance: 1.095 = 218.9 / 200   
## Misclassification error rate: 0.3365 = 71 / 211

plot(traintree)  
text(traintree, pretty=1, cex=0.5)



set.seed(1)  
tree.pred.test <- predict(traintree, testdata, type="class")  
table(Prediction=tree.pred.test,Reality=testdata$voteabove50)

## Reality  
## Prediction 0 1  
## 0 20 19  
## 1 10 42

mean(tree.pred.test==testdata$voteabove50)

## [1] 0.6813187

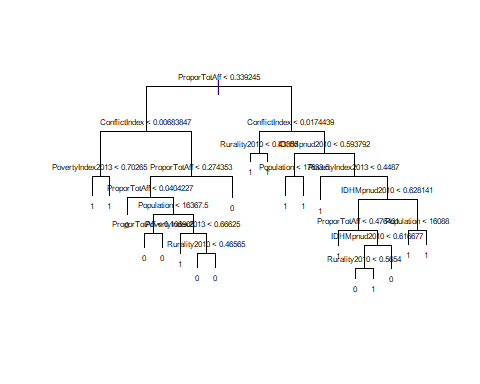
This tree was able to predict efficiently whether the Yes would obtain more than 50% of the votes in a each municipality a 68% of the times, which is not too bad.

However, we must pay attention to the fact that, when our tree predicted “0” (i.e., that the No would obtain more than 50% of the votes) (39 cases), only 20 of those were correct, as in the remainig 19cases the real result was “1” (i.e., that the Yes obtained above 50%). We had a 48% of false negatives, which is very high. We might try prunning the tree to see if we obtain better results (to reduce the overfitting from the training dataset), but let’s start over, adding more variables to build the tree.

newtree <- tree(voteabove50~ConflictIndex+Population+PovertyIndex2013+ArmedGroups+Rurality2010+IDHMpnud2010+ProporTotAff,data=traindata)  
summary(newtree)

##   
## Classification tree:  
## tree(formula = voteabove50 ~ ConflictIndex + Population + PovertyIndex2013 +   
## ArmedGroups + Rurality2010 + IDHMpnud2010 + ProporTotAff,   
## data = traindata)  
## Variables actually used in tree construction:  
## [1] "ProporTotAff" "ConflictIndex" "PovertyIndex2013"  
## [4] "Population" "Rurality2010" "IDHMpnud2010"   
## Number of terminal nodes: 20   
## Residual mean deviance: 0.6712 = 124.8 / 186   
## Misclassification error rate: 0.1699 = 35 / 206

plot(newtree)  
text(newtree, pretty=1, cex=0.5)



set.seed(1)  
newtree.pred.test <- predict(newtree, testdata, type="class")  
table(Prediction=newtree.pred.test,Reality=testdata$voteabove50)

## Reality  
## Prediction 0 1  
## 0 21 16  
## 1 9 45

mean(newtree.pred.test==testdata$voteabove50)

## [1] 0.7252747

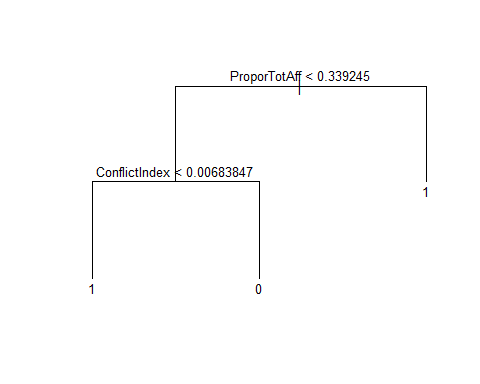
As we can observe, by adding more variables to the tree, it’s prediction power improved a little, from 68% to 72%. But, let’s prune it, hopefully that will help us obtain not only a simpler tree, but also a more acqurate one.

set.seed(1)  
cv.newtree=cv.tree(newtree,FUN=prune.misclass)  
cv.newtree

## $size  
## [1] 20 14 12 8 3 2 1  
##   
## $dev  
## [1] 68 70 76 78 73 95 95  
##   
## $k  
## [1] -Inf 0.00 1.00 1.75 3.00 14.00 15.00  
##   
## $method  
## [1] "misclass"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

The cross validation CV.tree function indicates that the optimal level of complexity is 20 (not pruning the tree), given that it minimise the number of misclasified (68). However, and given that this cross validations uses the data from the training dataset, it is possible that it is overfitting and, thus, a tree with level 3, which mislasified a bit more cases (73) but is much simpler, might perform better when we use the testing data. So, let’s prune the tree with complexity 3, and see if it obtains a better prediction;

pruned.tree=prune.misclass(newtree, best=3)  
  
  
plot(pruned.tree)  
text(pruned.tree, pretty=1, cex=0.8)



set.seed(1)  
pruned.tree.pred.test <- predict(pruned.tree, testdata, type="class")  
table(Prediction=pruned.tree.pred.test,Reality=testdata$voteabove50)

## Reality  
## Prediction 0 1  
## 0 19 7  
## 1 11 54

mean(pruned.tree.pred.test==testdata$voteabove50)

## [1] 0.8021978

The predictio of this tree was right 80% of the cases, a significant improvement. This tree uses only two variables: ProporTotAff (a variable computed by me, see explanation below) and ConflictIndex, which I have explained before.

ProporTotAff is simply the proportion of the municipality populationthat was direcctly affected by the conflict between 1984 and 2016. It was calculated in the following way: (sum of the number of individuals that were directly affected in the municipality each year between 1984 and 2016)/(the population of the municipality in 2016).

We had not used the ProporTotAff variables in the previous tests we did before this, but given that we now see that it has a interesting predective power, we might decide to start using it, and even maybe we should incorporate this variable to the tests we did before. We may do that later. Probably, using such variable instead of ConflitIndex might improve our former results.

## Random forest/Bagging

Now, just to practise, let’s fit again a model using all the variables that we used in the tree before pruning it, but let’s try with the random forest. But before going to the random forest, let´s do a bagging (Bootstrap Aggregation).

set.seed(1)  
  
library(randomForest)

## Warning: package 'randomForest' was built under R version 3.6.2

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

tree.bag <- randomForest(voteabove50~ConflictIndex+Population+PovertyIndex2013+ArmedGroups+Rurality2010+IDHMpnud2010+ProporTotAff,data=traindata, na.action=na.roughfix, mtry=7)  
  
bag.pred.test <- predict(tree.bag, testdata,type="class")  
table(Prediction=bag.pred.test,Reality=testdata$voteabove50)

## Reality  
## Prediction 0 1  
## 0 18 13  
## 1 12 48

mean(bag.pred.test==testdata$voteabove50)

## [1] 0.7252747

The tree we had fit with the same variables (named “Newtree”) was also able to predict correctly 72.5% of the testing dataset, i.e., using the bagging method did not lead us to an improvement in the prediction.

What if we go to a random forest? (The difference is that random forest, when building tress on bootstrapped training sample, it also choses a random sample of predictors. We do this by selecting a smaller parameter mtry)

set.seed(1)  
  
library(randomForest)  
tree.rf <- randomForest(voteabove50~ConflictIndex+Population+PovertyIndex2013+ArmedGroups+Rurality2010+IDHMpnud2010+ProporTotAff,data=traindata,na.action=na.roughfix, mtry=3)  
  
rf.pred.test <- predict(tree.rf, testdata,type="class")  
table(Prediction=rf.pred.test,Reality=testdata$voteabove50)

## Reality  
## Prediction 0 1  
## 0 18 9  
## 1 12 52

mean(rf.pred.test==testdata$voteabove50)

## [1] 0.7692308

Now, we obtained a small improvement, as the random forest is able to predict correctly almost 77% of the cases.