Zero Proportion Analysis

L3P3

Friday, November 28, 2014

This document analyzes the model performance based on the proportion of zeros added to the ones in the model and compares it with the model without the zero proportion adjustment phase. We will compare the models in terms of created models, execution time, precision, recall and fscore and decide which zero proportion is the optimal one.

Introduction

We are going to study the influence of the zero proportion added to the ones of the dataset in the dataset split phase. We do so to reduce the amount of zeroes present in our datasets to improve computation time and the performance of the models, based on the paper "Logistic Regression for Rare Events".

Amount of created models

From a total of 78 possible models, this was the amoutn created in each case: 50, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51, 51

The proportion of zeroes added doesn't seem to affect the amount of created models.

```
## Loading required package: glmnet
```

```
## Warning: package 'glmnet' was built under R version 3.0.3
```

```
## Loading required package: Matrix
## Loading required package: lattice
## Loaded glmnet 1.9-8
```

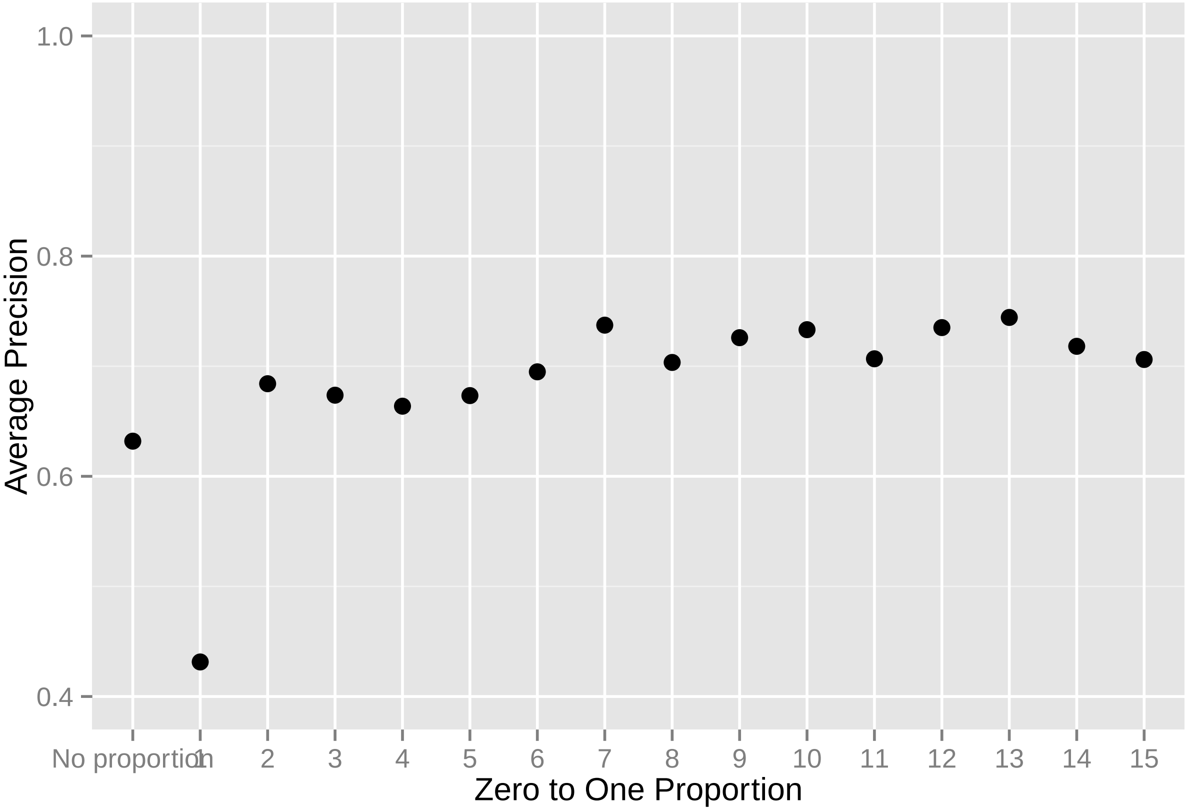
Obtained Precision

Even though precision is not a key deciding parameter, we still can gain some insight by studying it. In average, this is how it varies in the experiments:

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
mean_vec<-apply(prec_df,2,function(x){mean(x,na.rm=TRUE)})
ggplot()+geom_point(aes(x= factor(names(prec_df),levels=names(mean_vec)),y=mean_vec),size=3)+xlab("Zero
```

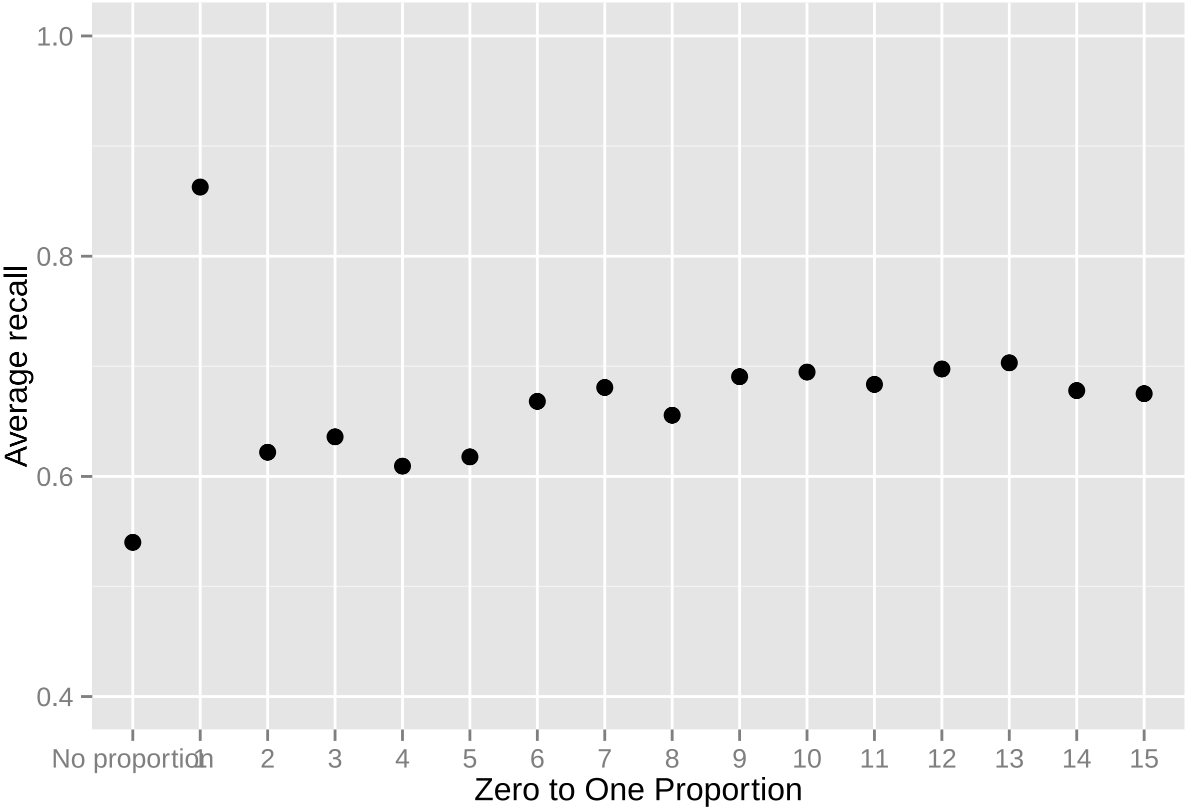


The maximum precision is obtained when the zero proportion is 13.

Obtained Recall

Even though recall is not a key deciding parameter, we still can gain some insight by studying it. In average, this is how it varies in the experiments:

```
require(ggplot2)
mean_vec<-apply(rec_df,2,function(x){mean(x,na.rm=TRUE)})
ggplot()+geom_point(aes(x= factor(names(rec_df),levels=names(mean_vec)),y=mean_vec),size=3)+xlab("Zero
```

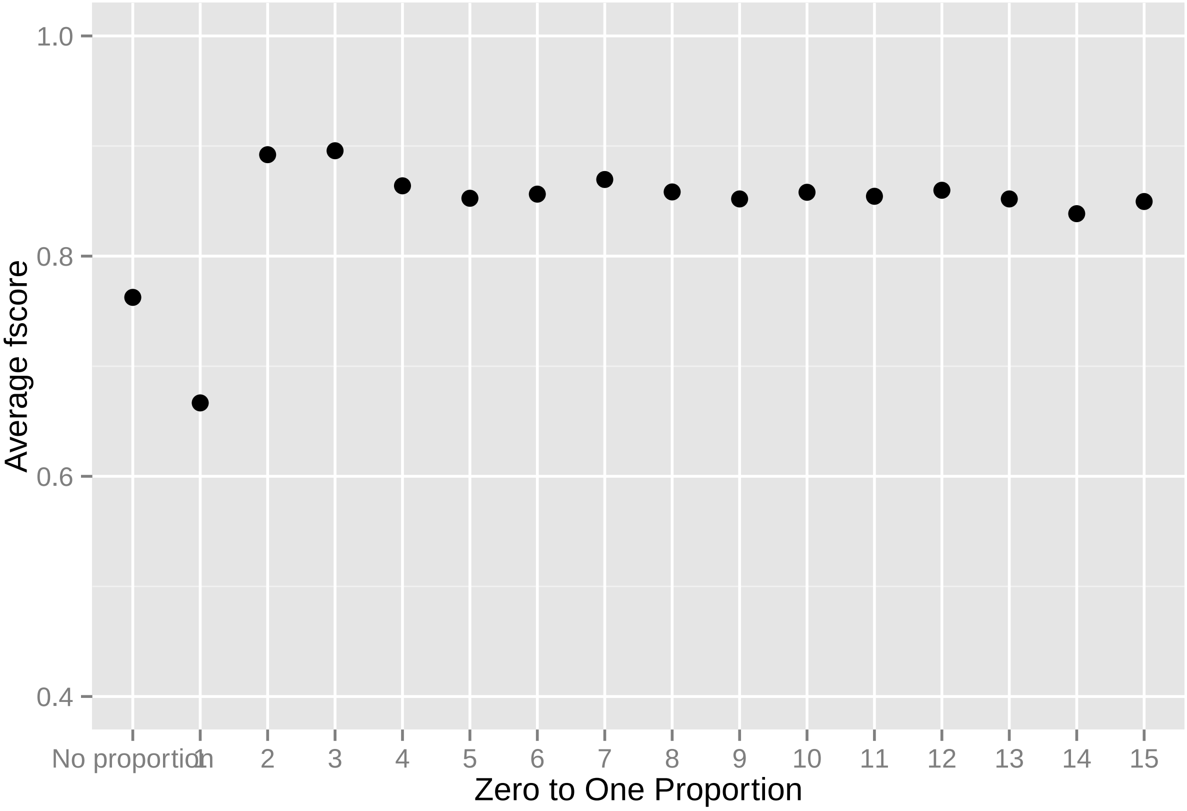


The maximum recall is obtained when the zero proportion is 1, but taking into account that precision was very low in this case, the next one is 13.

Obtained fscore

The fscore is, indeed, a key deciding parameter, so this parameter will make us decide which is the optimal amount of proportion of zeros.

```
require(ggplot2)
mean_vec<-apply(fsc_df,2,function(x){mean(x,na.rm=TRUE)})
ggplot()+geom_point(aes(x= factor(names(fsc_df),levels=names(mean_vec)),y=mean_vec),size=3)+xlab("Zero
```

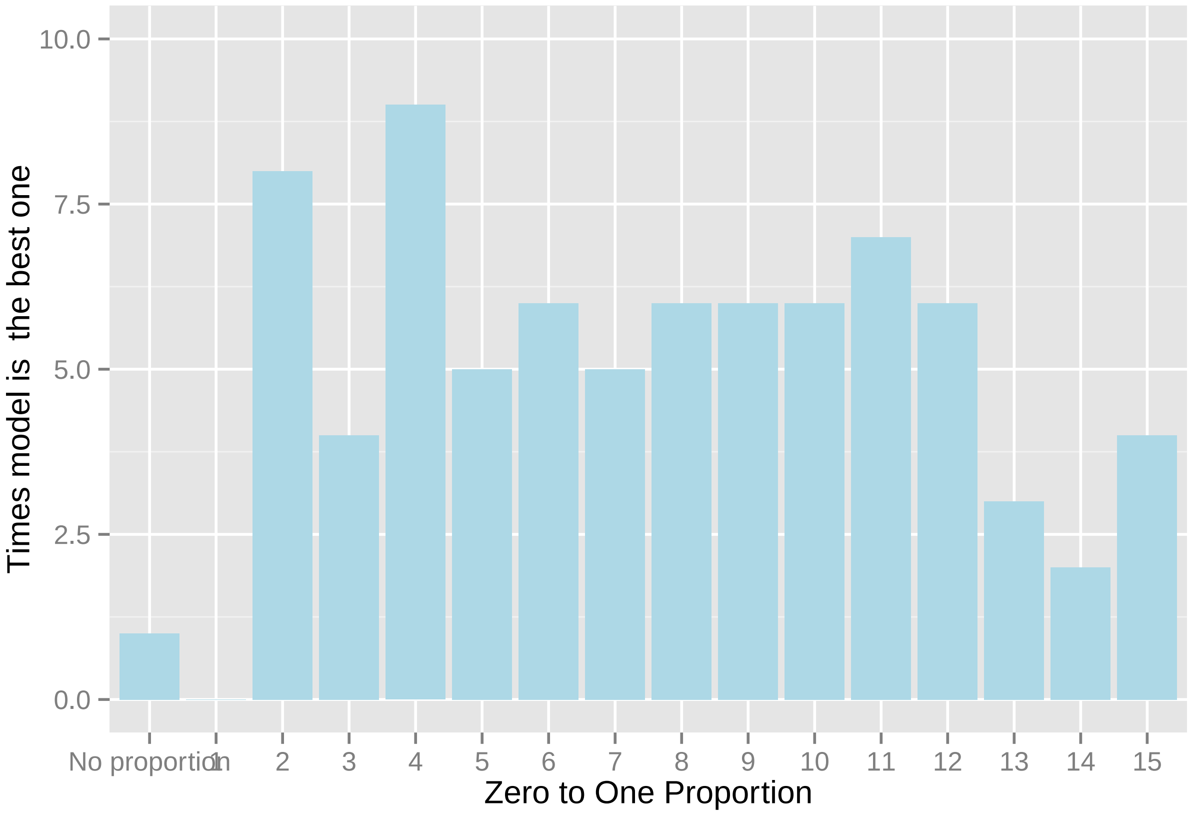


The maximum fscore is obtained when the zero proportion is 3.There is another point to consider: the amount of non-zero models, this is, the amount of models that were succesful. This amount is:

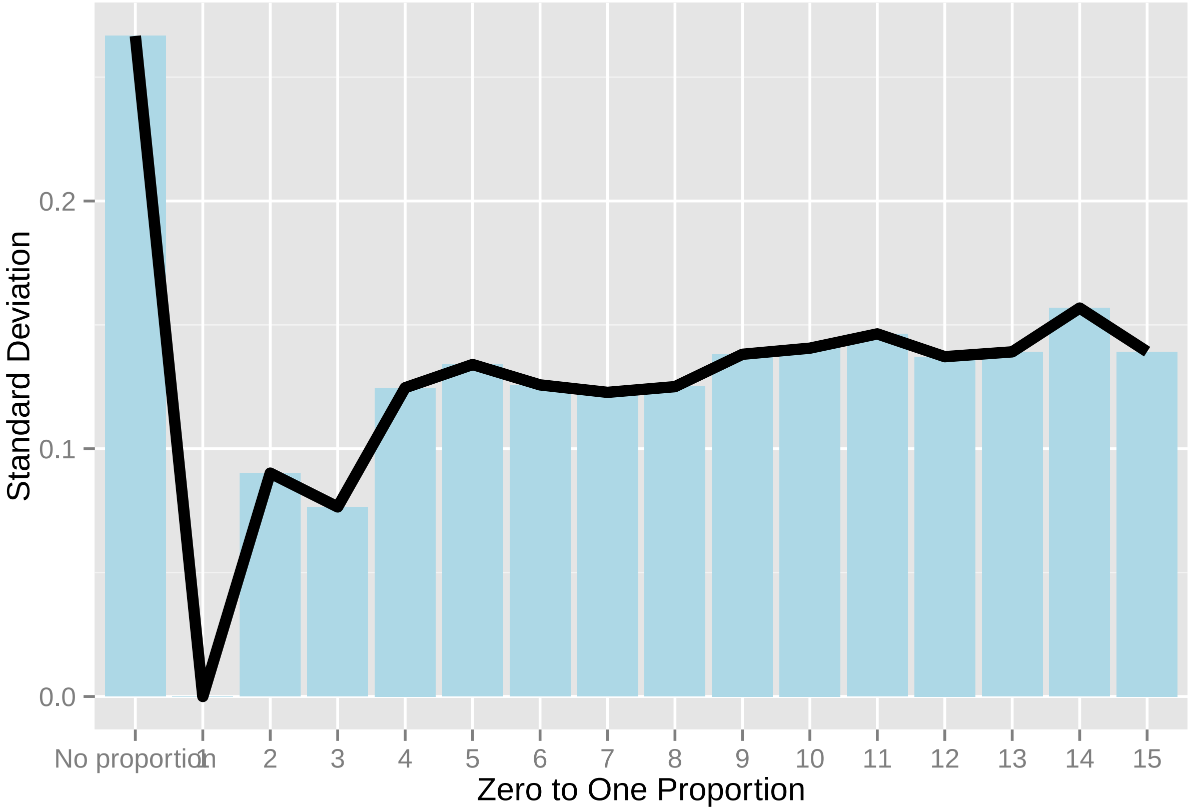
```
##      No proportion  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
## [1,]            36 44 37 37 37 38 40 41 40 42 42 41 42 43 42 41
```

So it is clear that a zero proportion of 3 is, actually, the amount that produces less successful models. So now we take a different approach: we now check, for each event, which zero proportion gave the maximum fscore and summarize it:

```
## Mapping a variable to y and also using stat="bin".
##   With stat="bin", it will attempt to set the y value to the count of cases in each group.
##   This can result in unexpected behavior and will not be allowed in a future version of ggplot2.
##   If you want y to represent counts of cases, use stat="bin" and don't map a variable to y.
##   If you want y to represent values in the data, use stat="identity".
##   See ?geom_bar for examples. (Deprecated; last used in version 0.9.2)
```

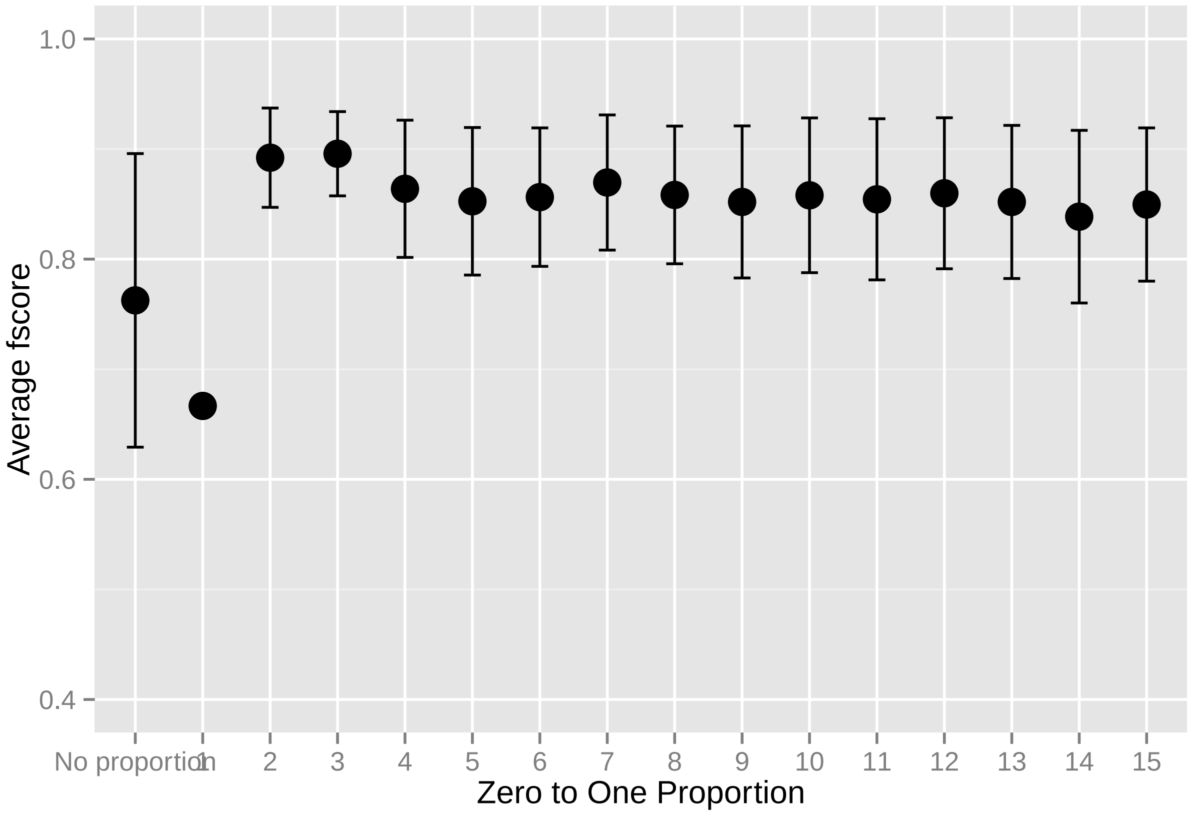


This figure adds an interesting point: even though there are less models created when the proportion is 4 and it is not the option that gets the best average fscore, this is the option that consistently produces best results. To check for possible explanations, we now plot the standard deviation of each option's fscore:



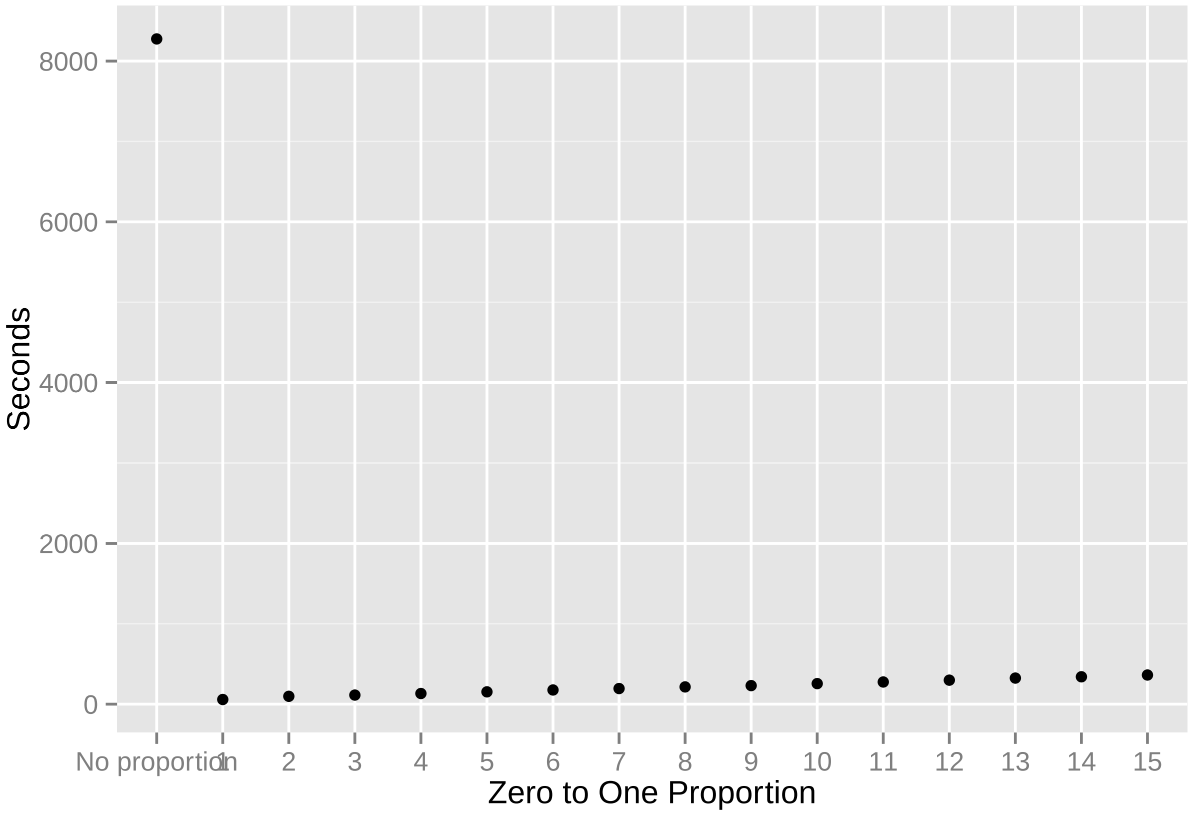
This chart shows how the option 4 can be higher more times than the third one and at the same time have a lower average fscore: it also has more low values.

To add together both informations we now plot each average fscore with its error bars:

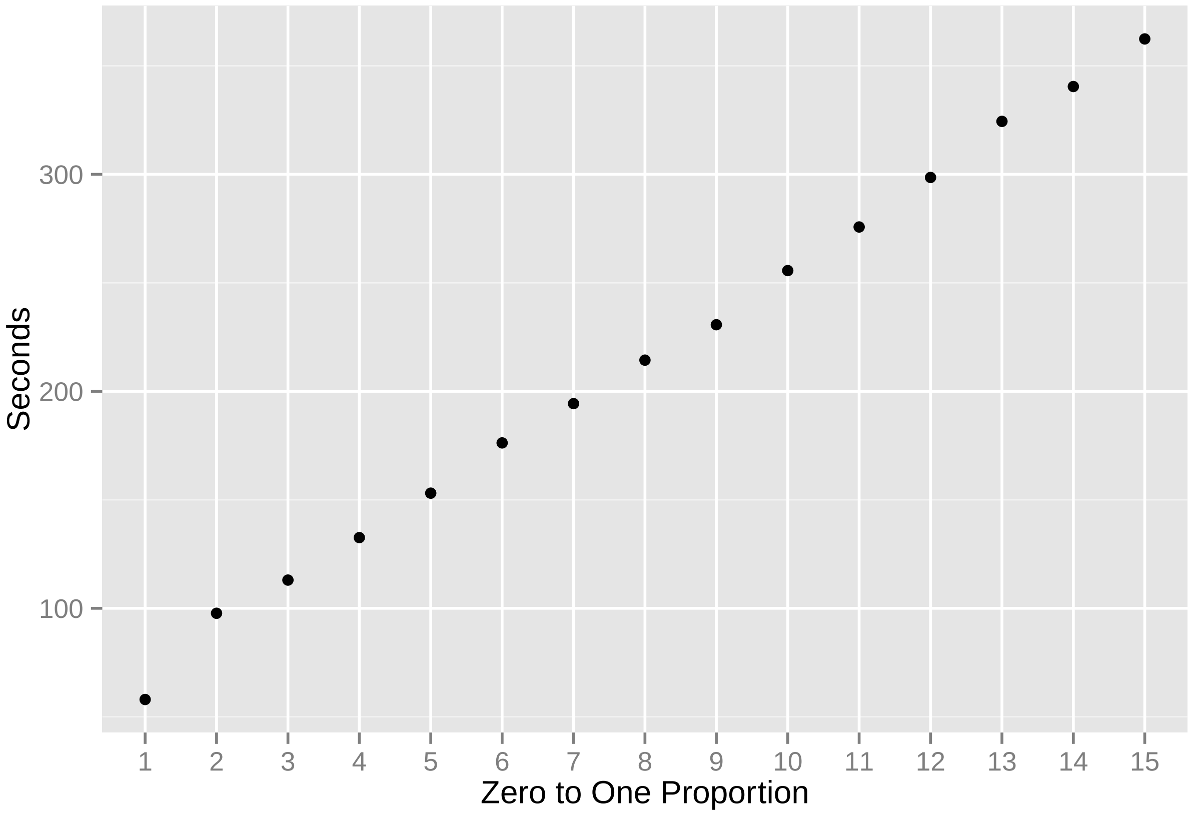


computation Time

As a last criteria, we now compare the computation time for each option:



This figure shows that adding the proportioning phase divides the computation time by, at least, a factor of 22.To further study the effect of the actual proportion of zeroes added, we now plot only that part of the chart:



Unsurprisingly, the computation time for each iteration is a linear function of the amount of zeroes added.In fact, we can model it as:

```
##
## Call:
## lm(formula = time_vec[-1] ~ proportion)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3773  -1.1085   0.1555   2.0139   8.3832
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  47.3777     2.4043    19.7 4.58e-11 ***
## proportion   20.9695     0.2644    79.3  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.425 on 13 degrees of freedom
## Multiple R-squared:  0.9979, Adjusted R-squared:  0.9978
## F-statistic:  6288 on 1 and 13 DF,  p-value: < 2.2e-16
```

And, trying to generalize setting the time for a 1:1 proportion as 1, the model we get is:

```
##
## Call:
```

```
## lm(formula = time_vec_new ~ proportion)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.179011 -0.019123  0.002683  0.034741  0.144613
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.817280   0.041475    19.7 4.58e-11 ***
## proportion  0.361731   0.004562    79.3  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07633 on 13 degrees of freedom
## Multiple R-squared:  0.9979, Adjusted R-squared:  0.9978
## F-statistic:  6288 on 1 and 13 DF,  p-value: < 2.2e-16
```

This model allows us to predict how long it will take for a specific model compared to the time it would take to fit the model with a 1:1 zero proportion, as $times = \alpha + \beta * zero_proportion$, where α = 0.8172799 and β = 0.3617308.

Conclusion

Through this analysis we can conclude that the best option for our experiments is to keep a proportion of 3 samples of zeroes to ones.