17/7/2020

Divide and recombine

Advanced topics in Statistics

Brikena Kokalari

ASSIGNMENT II

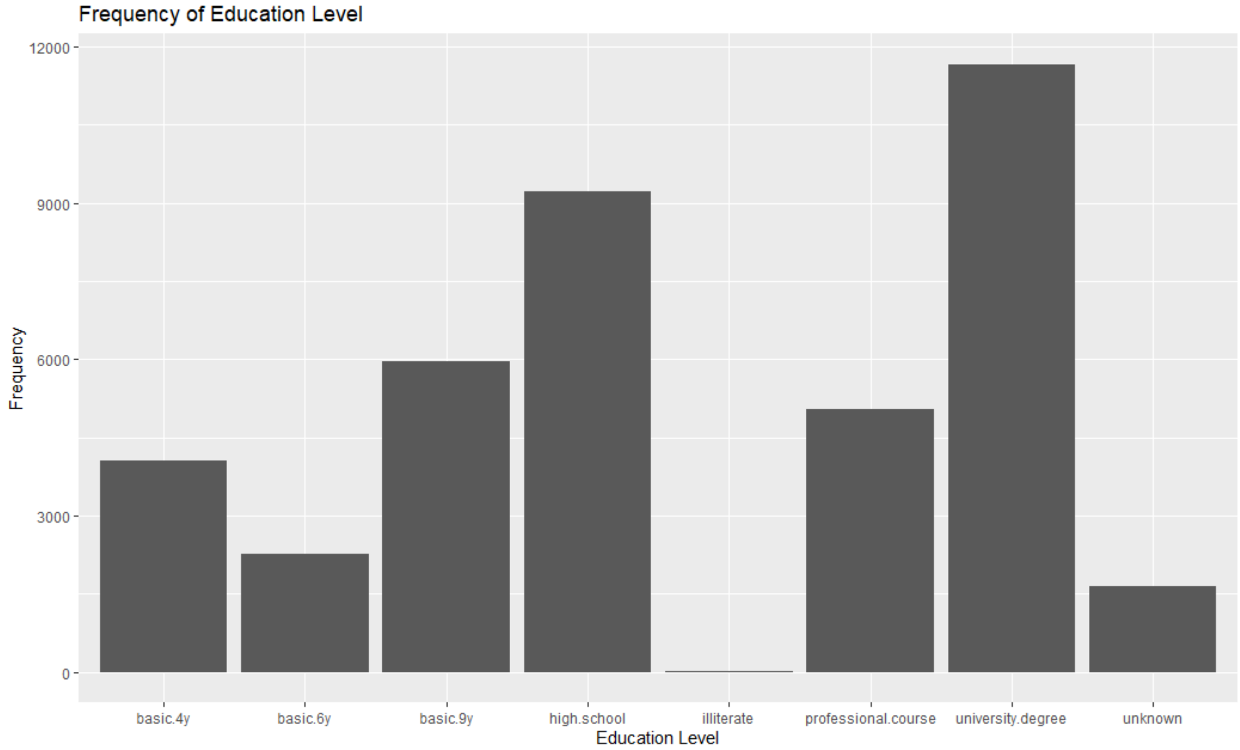
Introduction

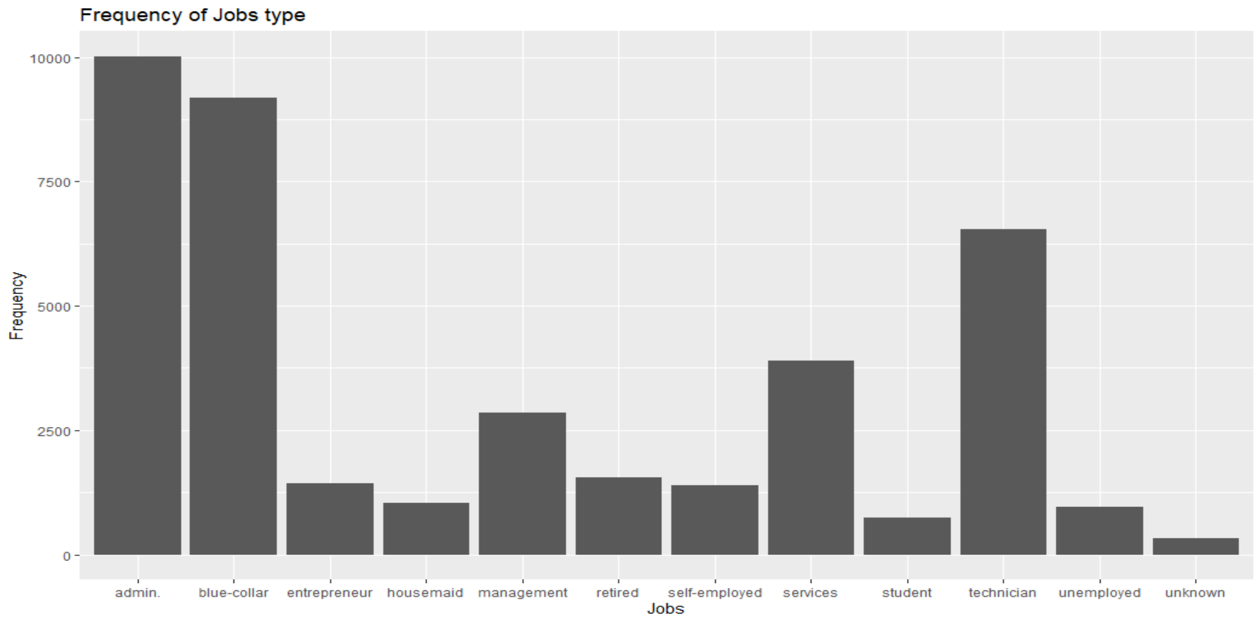
The aim of this project concerns Divide and Recombine method regarding a dataset which relates to telemarketing phone calls to sell long-term deposits. Within a campaign, the agents make phone calls to a list of clients to sell the product (outbound) or, if meanwhile the client calls the contact-center for any other reason, he is asked to subscribe the product (inbound). Thus, the result is a binary unsuccessful or successful contract. Data are collected from one of the retail banks, from May 2008 to June 2010 , in a total of 39883 phone contacts. We will run a logistic regression on all the data and then use the divide and recombine approach with 10 and 20 splits. The purpose of the project is to compare the different estimation approaches and see how much they agree.

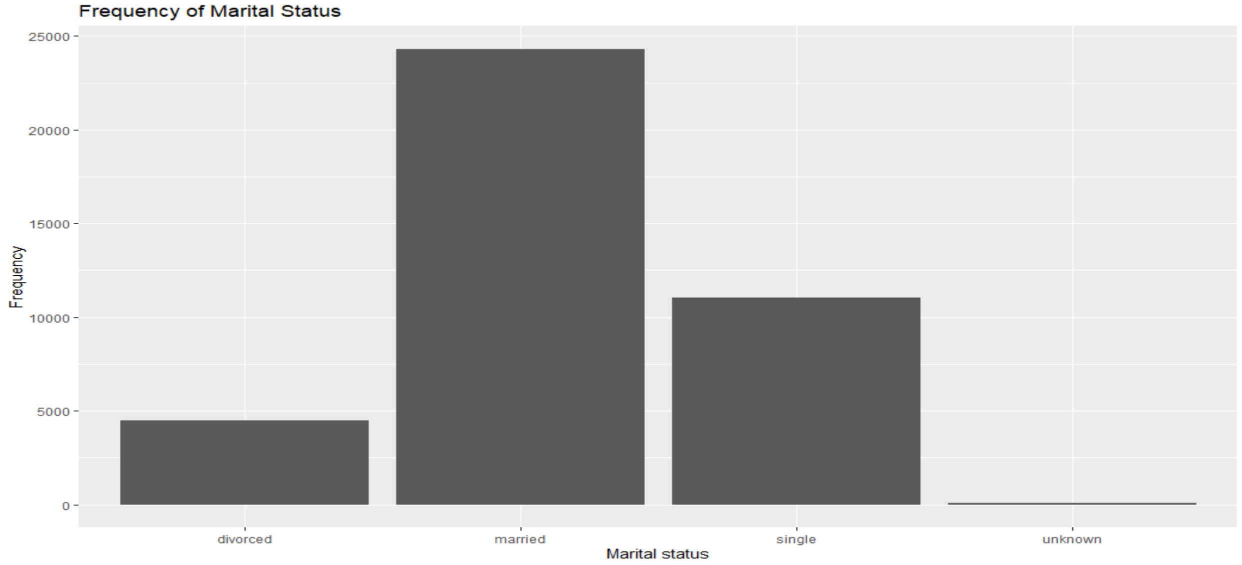
D&R fundamentally a statistical approach to deep analysis of large complex data. In D&R, the data are divided into subsets by a statistical division method. The subsets are stored in objects with the same data structure, either on disk or in memory. A statistical recombination method is applied to the outputs of the of each analytic method to form the final D&R result.

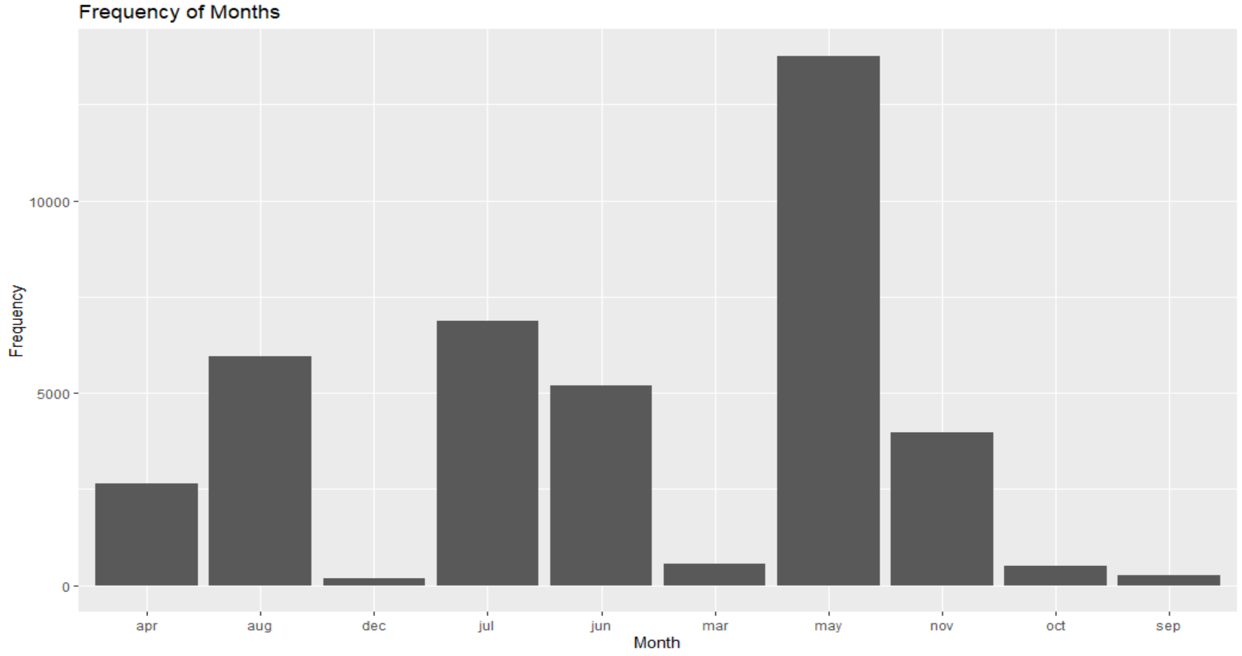
First let's have a look at the data.

Categorical variables :



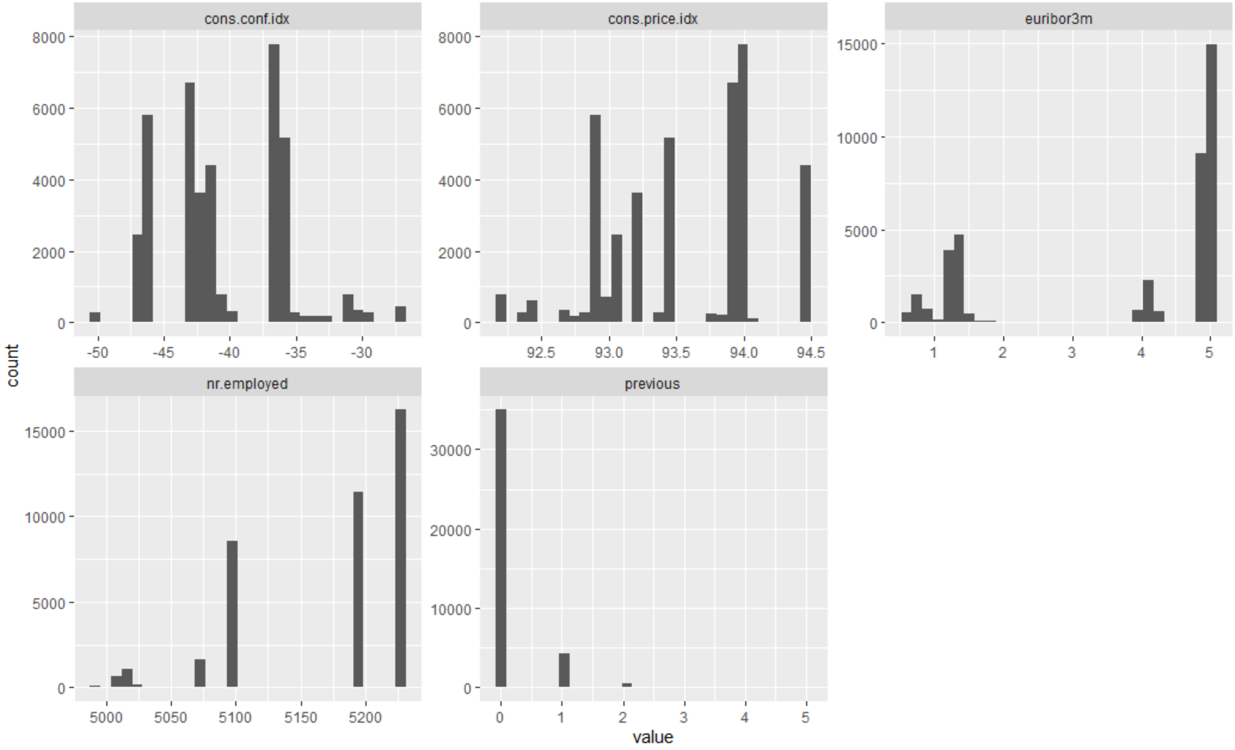






We can observe that some levels in the categorical variables do not appear quite frequently.

- Numerical variables

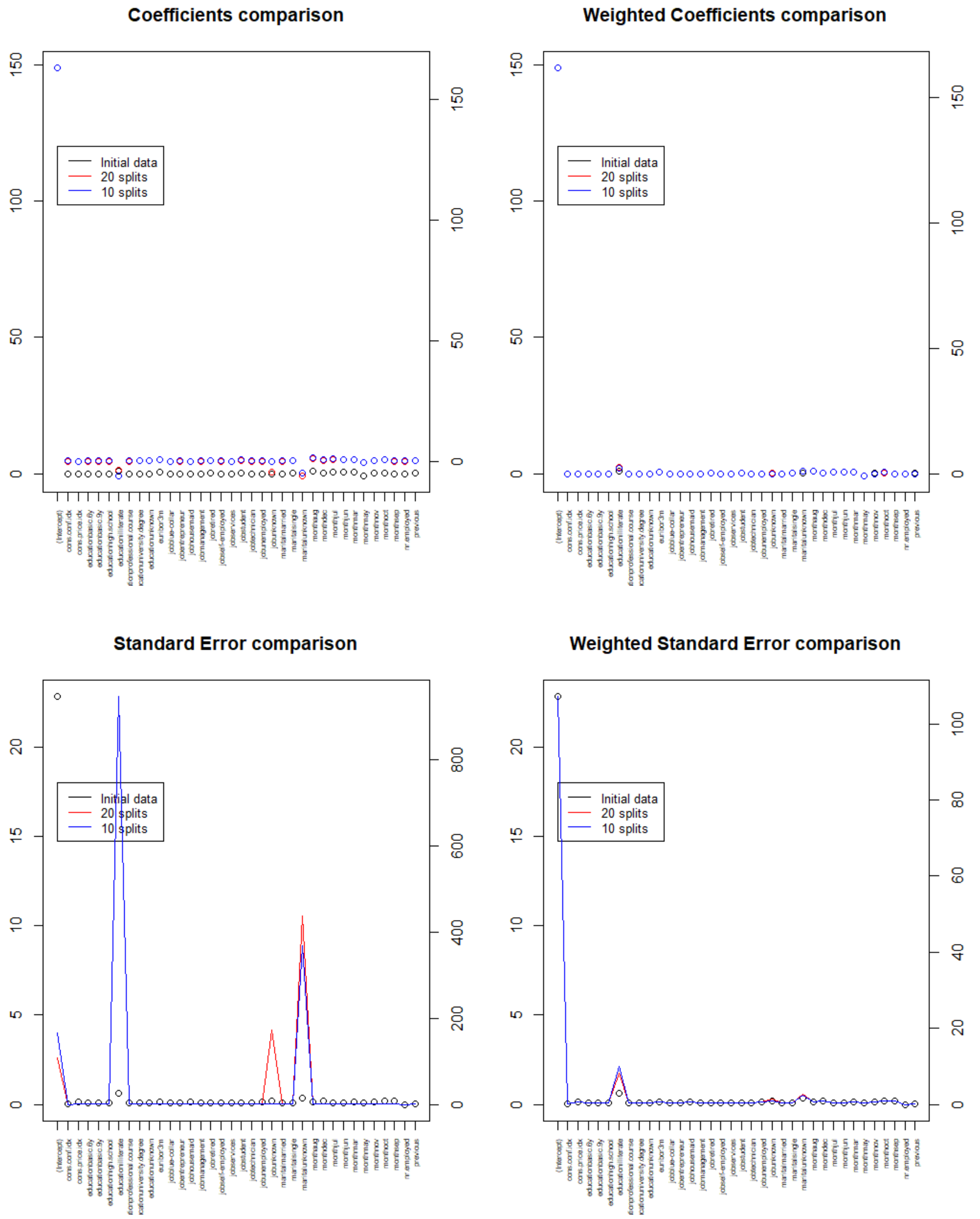


Firstly, we shuffled the data and make a function to split the data in as many splits as we wanted. The fist approach is dividing in 10 subsets randomly. In this phase we may have to re-run the splitting in 10 datasets so as each split contain every factor of the categorical variables at least once. Another way to tackle this problem is by applying dummy encoding on categorical variables so as even if the variable is not present in the bin/split, its label is present.

We run the logistic regression in each split and kept the coefficients of independent variables and the standard errors in a data frame. The next step was to calculate the mean of each coefficient of the 10 split and the mean of standard error.

The same procedure was run for the 20 splits as well. What we observed was that the 10 splits approach gave us coefficients that were closer to the coefficients of the initial model. That make sense since the 10 splits contain more information regarding the data and the standard error is being calculated 10 time whereas in 20 splits approach, we have less data and standard error is higher.

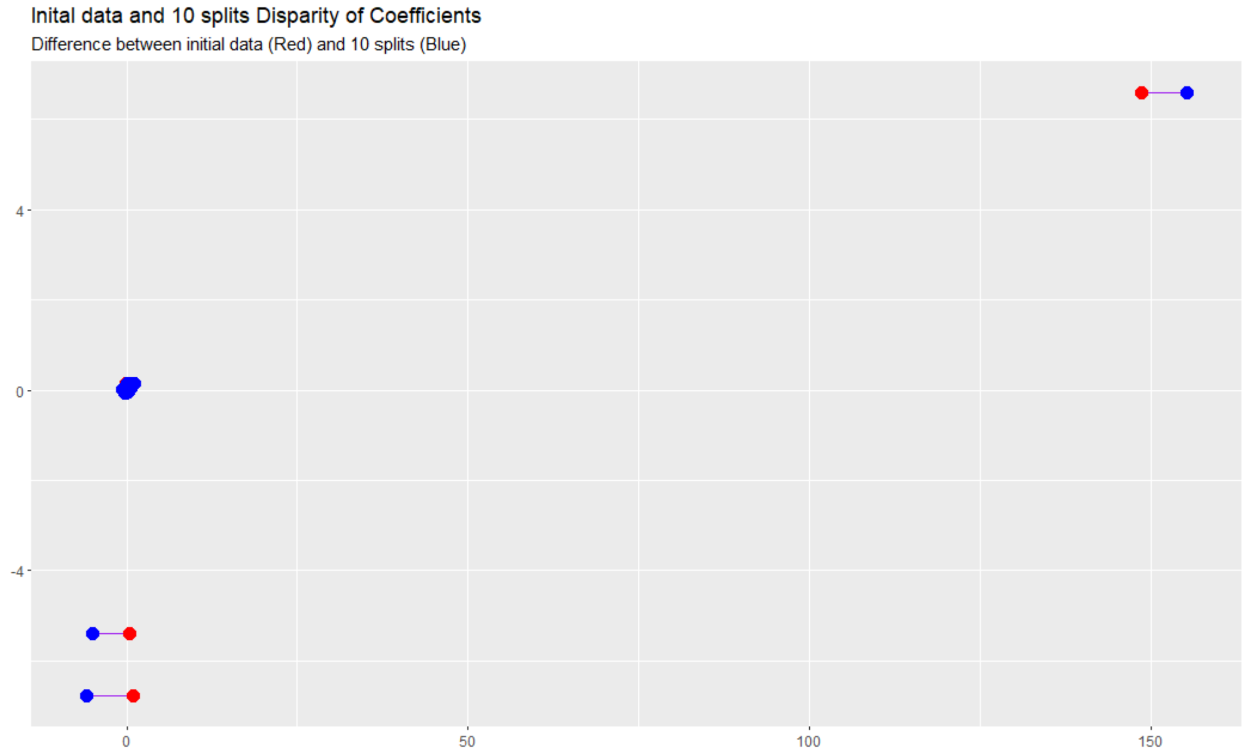
A second approach was to calculate the same coefficients and standard error but with weights. The weights were considered regarding standard error. The higher the standard error, the lower the weight of the coefficient. To be accurate, the weight was calculated as: w=1/St. Error.

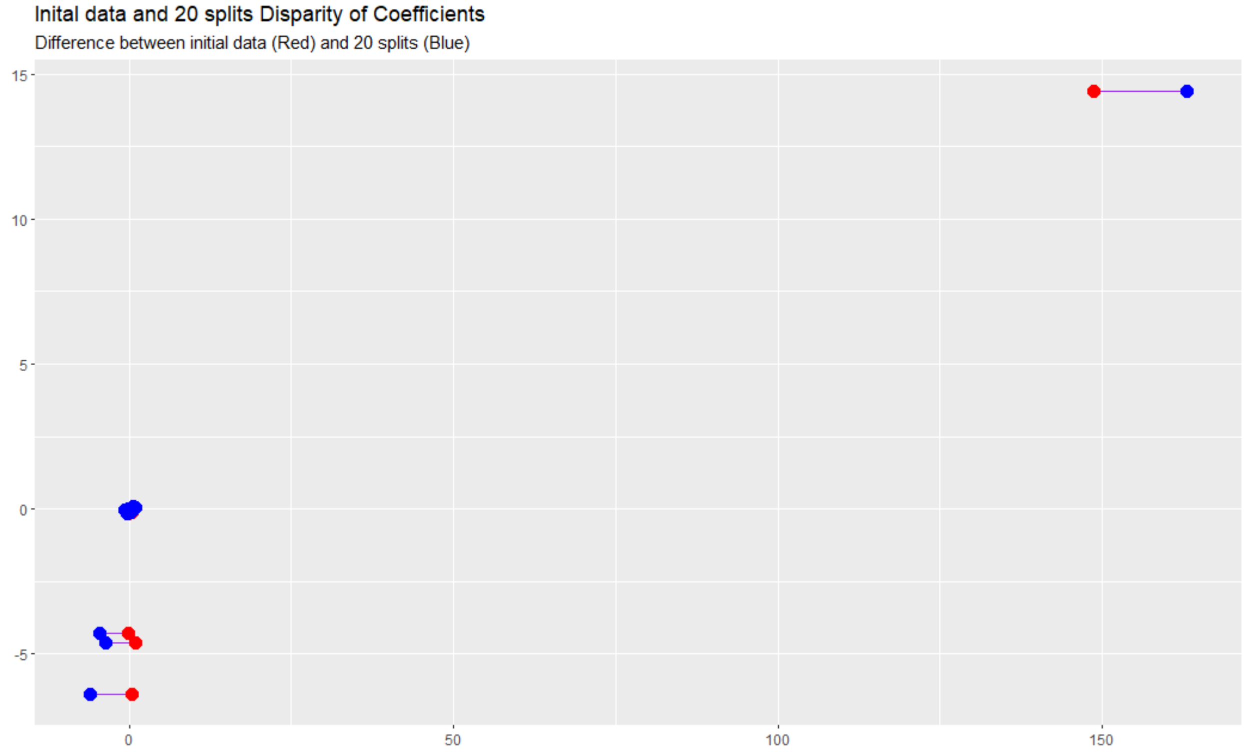


We visualized the results and from the above plots, we observe that weighted mean of coefficients and weighted mean of standard error gave us better results and depict a closer picture of the real data.

We also plotted the difference of the 10 and 20 splits in the below dumbbell, only for visualization reasons.

Comparing 10 splits and 20 splits we can see that the gap of the difference for some variables is higher in the 20 spits approach.

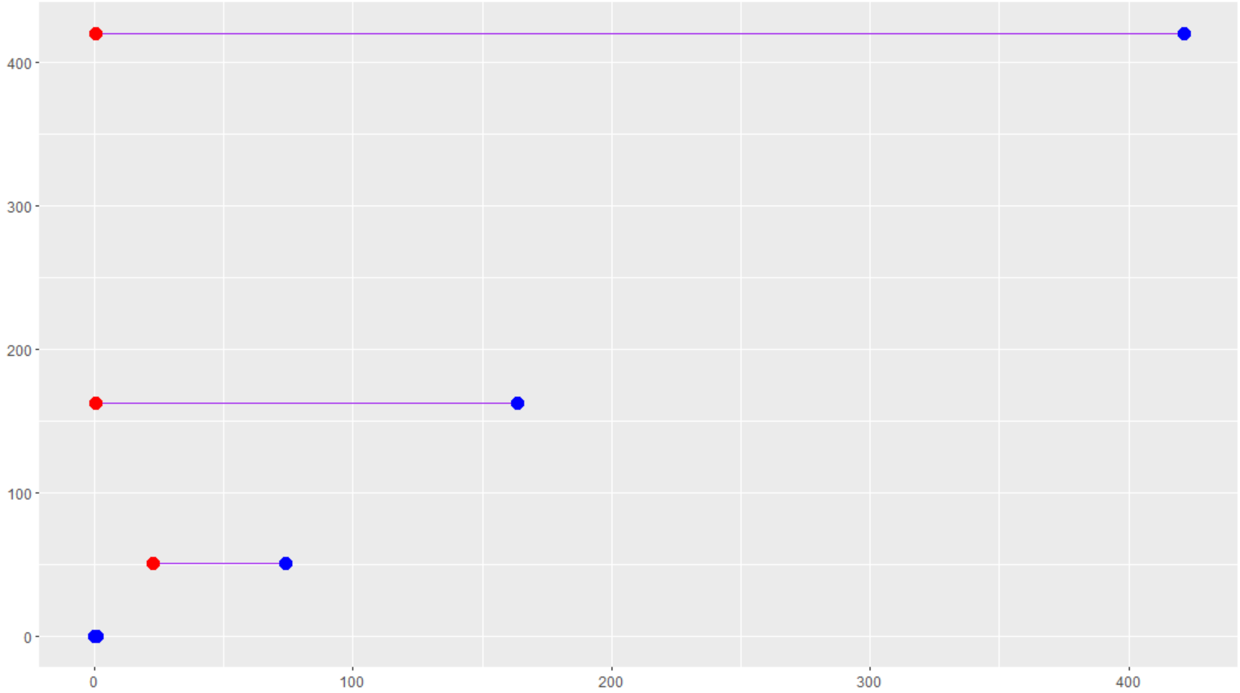




Accordingly, the gap in the standard error estimating seems higher in the 20 splits. A logical explanation of that could be the fact that the frequency of specific levels of a categorical variable (ex. job) is too low that can not be seen in all spits. For instance, Job type illiterate is appearing only 18 times in dataset so by a random split it cannot be seen in every bin. That makes not only the standard error of the specific factor to be higher but the coefficient as well.

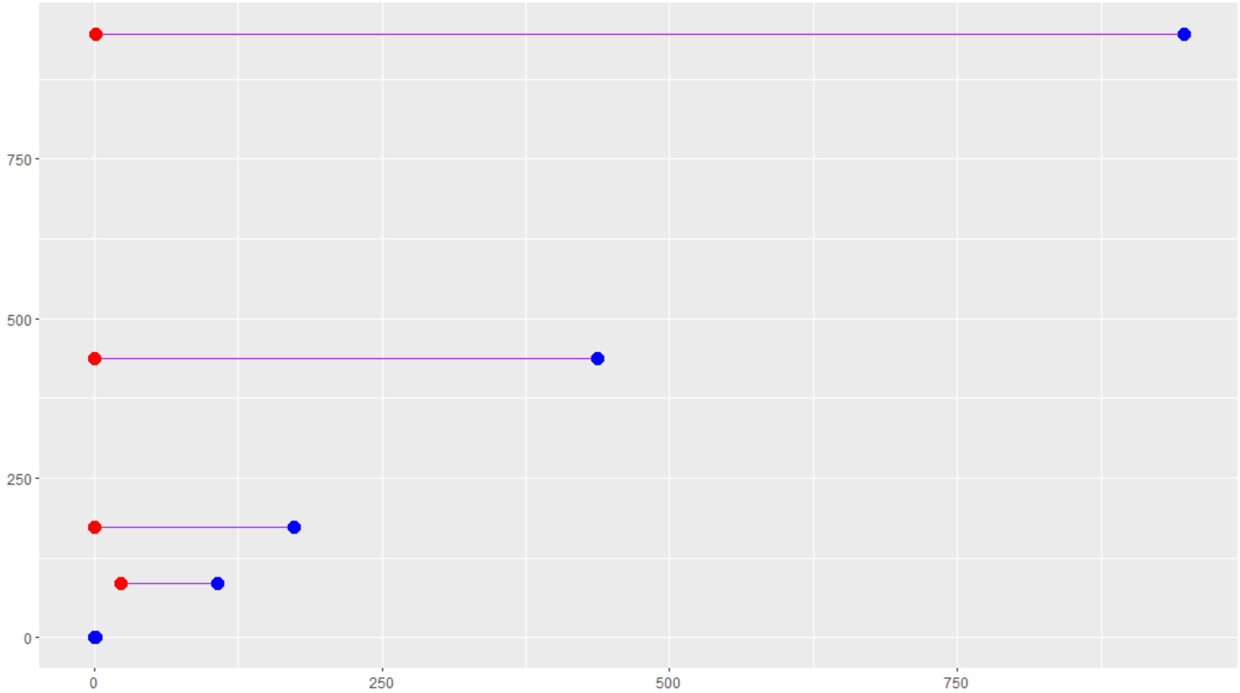
Inital data and 10 splits Disparity of Standard Error

Difference between initial data (Red) and 10 splits (Blue)



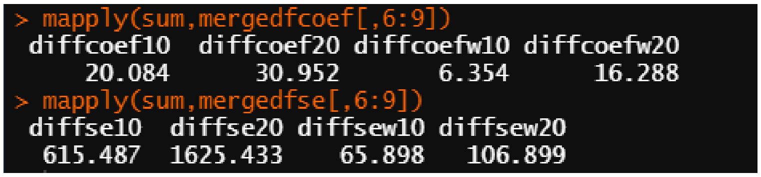
Inital data and 20 splits Disparity of Standard Error

Difference between initial data (Red) and 20 splits (Blue)



But which approach should we choose?

In order to choose the best approach, we calculated the sum of absolute value of the difference of the means of coefficients and St. error of each approach with the real data. What we observed was that the weighted approach of 10 splits gave us sum of 6,35 on coefficients and 65,9 on standard error.



```
> mapply(sum,mergedfcoef[,6:9])
 diffcoef10  diffcoef20 diffcoefw10 diffcoefw20
     20.084      30.952       6.354      16.288
> mapply(sum,mergedfse[,6:9])
 diffse10  diffse20 diffsew10 diffsew20
  615.487  1625.433    65.898   106.899
```