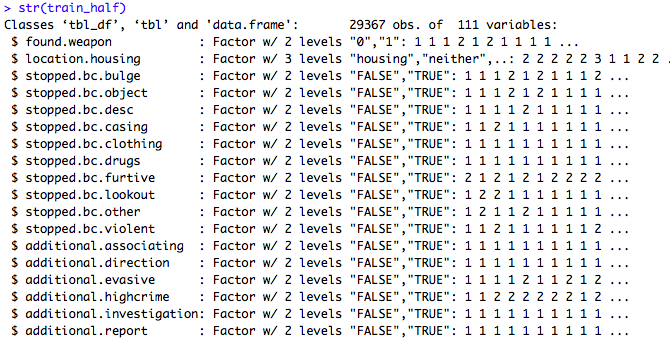
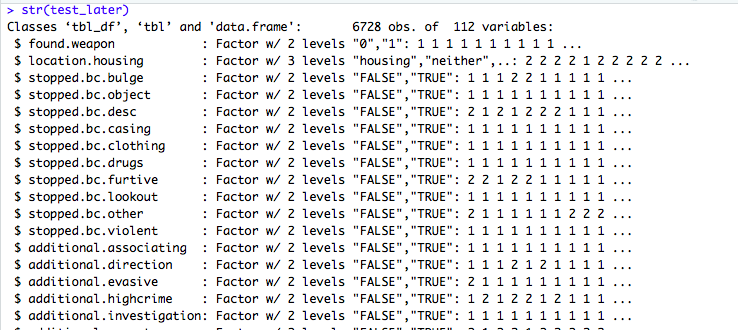
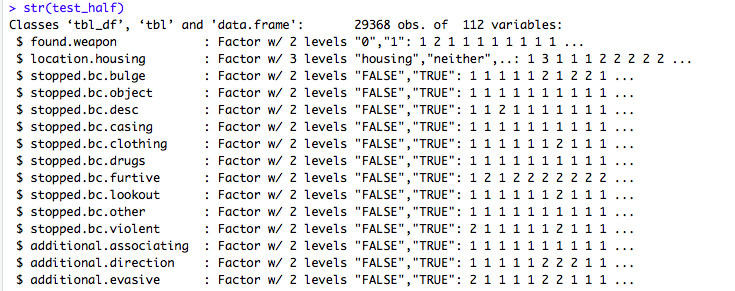
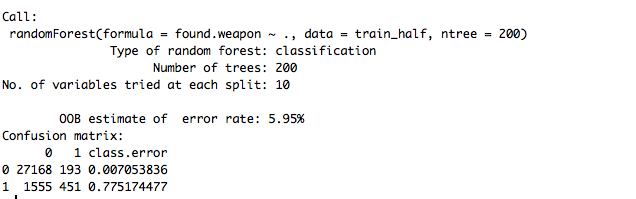
1. Create one training set and two validation sets in the following manner. Restrict **sqf** to years 2013-2014, randomly shuffle the data, and split it in half. Call one half **train\_half**, and the other half **test\_half**. Next, restrict **sqf** to just the year 2015, and call this **test\_later**. Remove the stop id and year columns from **train\_half**, **test\_half**, and **test\_later**.







1. Fit a randomforest model on **train\_half** using the *randomForest* package in R,  predicting found.weapon as a function of all features. Use 200 trees, but all other options  for the model can be the default options.



1. Generate predicted *probabilities* using the model from partB for both **test\_half**  and **test\_later**. Compute the AUC of the model on each test set. Write a paragraph describing and interpreting your results. In particular, discuss the following three questions.

AUC of test\_half: **85.25449**

AUC of test\_later: **75.73225**

1. Why do you think the AUC on **test\_half** is noticeably higher than the AUC on **test\_later**?

The AUC on **test\_half** is noticeably higher than the AUC on **test\_later** because **test\_half** comes from the same subset of data as **train\_half**, therefore are responses from both 2013 and 2014. However, **test\_later** uses responses only from 2015.

1. If you were planning to use this model to guide how officers make stops in the future (e.g., by having officers use the model to compute the probability that an individual suspected of criminal possession of a weapon will have a weapon, and then only making a stop if the model-estimated probability is sufficiently high), would the AUC on **test\_half** or **test\_later** be a better estimate of performance on unseen data?

The AUC from **test\_later** would be a better estimate of performance on unseen data because **test\_later** is based off data from a different year. The **test\_half** AUC is an inferior choice because our model is over fitting the **test\_half** data.

1. More generally, when evaluating a model using a simple training/validation split approach, should you always do the split by shuffling and splitting randomly?   No, because … idk why but I just know no lol.