# Bootstrapping vs crossvalidation

<https://datascience.stackexchange.com/questions/32264/what-is-the-difference-between-bootstrapping-and-cross-validation>

Both cross validation and bootstrapping are *resampling* methods.

* bootstrap resamples with replacement (and usually produces new "surrogate" data sets with the same number of cases as the original data set). Due to the drawing with replacement, a bootstrapped data set may contain multiple instances of the same original cases, and may completely omit other original cases.
* cross validation resamples without replacement and thus produces surrogate data sets that are smaller than the original. These data sets are produced in a systematic way so that after a pre-specified number k of surrogate data sets, each of the n original cases has been left out exactly once. This is called k-fold cross validation or leave-x-out cross validation with x=nk, e.g. leave-one-out cross validation omits 1 case for each surrogate set, i.e. k=n.
* As the name cross validation suggests, its primary purpose is measuring (generalization) performance of a model. On contrast, bootstrapping is primarily used to establish empirical distribution functions for a widespread range of statistics (widespread as in ranging from, say, the variation of the mean to the variation of models in bagged ensemble models).
* The leave-one-out analogue of the bootstrap procedure is called jackknifing (and is actually older than bootstrapping).
* **The bootstrap analogue to cross validation estimates of generalization error is called out-of-bootstrap estimate** (because the test cases are those that were left out of the bootstrap resampled training set).
* In practice there's often not much of a difference between iterated k-fold cross validation and out-of-bootstrap. With a similar total number of evaluated surrogate models, total error [of the model prediction error measurement] has been found to be similar, although oob typically has more bias and less variance than the corresponding CV estimates. There are a number of attempts to reduce [oob bias](https://en.wikipedia.org/wiki/Out-of-bag_error) (.632-bootstrap, .632+-bootstrap) but whether they will actually improve the situation depends on the situation at hand.
* There are fewer combinations possible for CV than for bootstrapping. But the limit for CV is probably higher than you are aware of. For a data set with n cases and k-fold cross validation, you have
  + CV (*n binomimial k*) combinations without replacement (for k < n that are far more than the *k*  possibilities that are usually evaluated) vs.
  + bootstrap/oob (2*n*−1 binomial *n*) combinations with replacement (which are again far more than the, say, 100 or 1000 surrogate models that are typically evaluated)

<https://stats.stackexchange.com/questions/18348/differences-between-cross-validation-and-bootstrapping-to-estimate-the-predictio>

* It comes down to variance and bias (as usual). CV tends to be less biased but K-fold CV has fairly large variance. On the other hand, bootstrapping tends to drastically reduce the variance but gives more biased results (they tend to be pessimistic). Other bootstrapping methods have been adapted to deal with the bootstrap bias (such as the 632 and 632+ rules).
* Two other approaches would be "Monte Carlo CV" aka "leave-group-out CV" which does many random splits of the data (sort of like mini-training and test splits). Variance is very low for this method and the bias isn't too bad if the percentage of data in the hold-out is low. Also, repeated CV does K-fold several times and averages the results similar to regular K-fold. I'm most partial to this since it keeps the low bias and reduces the variance.
* For large sample sizes, the variance issues become less important and the computational part is more of an issues. I still would stick by repeated CV for small and large sample sizes.

Bootstrap bias is not pesimistic, it is optimistic (Simple Bootstrap not .0632). This is because Bootstrap uses a lot of training elements to test the model leading to a lot of weight for in sample erro