6.1. Apply the naiveBayes klarR program with *cross-validation* to the classification of your stock **range** (HighRisk or LowRisk), using lagged ranges as x-variables. How well does NB do compared to knn using the kcvSearch to select k from the train data? To make this a fair comparison, you need to run knn on the same train-test split that you gave to the NB program.

6.2 Add 8 more lags to your stock range data - the Hi-risk vs Lo-risk data. Do that in Excel using our best practices method. Try running kNN and klaR with that data and with the same train-test split.

(a) What changes do you observe in the performance?

(b) Did adding more regressors (more lags) effect the choice of k in kNN?

(c) Did that slow down the kcvSearch program? Why would that matter?