Code organization suggestion:

For each base model, following scripts:

1. Takes fit window range; name of output file; name/loc of configuration file; (‘harness’ that calls ‘grey box script’; can perform a single fit with a given fit window)
2. Takes name of fit (output) file, takes validation fit range; runs prediction rules from output and generates predictions for one validation week (Alt notes: Takes a single validation week; grabs all fits that were freshest as of that week; generates predictions with each model at the varying complexities)
3. Takes names of files with all predictions for all validation weeks in validation window; picks best complexity
4. Looks through each row of blue in the red and generates a summary; pick best

Find recording software for skype calls!

Script 1 – to run the script I need a MM (which is the same for all iterations). For each validation week I must generate an index for to indicate the validation and fit data needed from the MM for model fitting. I will need a configuration file with, say, a matrix where each row is a configuration and each column is a parameter to be varied; I also need to pick which row/configuration is going to be used.

* generates one prediction rule for one validation week, given base model (boosted trees), configuration, and fit window
* takes fit window
* takes configuration file
* takes name of output file
* takes MM
* for given fit window, base model, model configuration, and MM:
  + using training data within fit window, generates a prediction rule
  + saves prediction rule as a .Rout file, with given file name
  + prediction rule corresponds to a single cell of yellow/orange matrix
* loops over validation week; external loop will handle different base models and configurations – one call to this script handles one base model with one configuration
  + after external loops, have a prediction rule for each base model, each configuration, and each validation week

**\*\*\*I am collapsing script 1 and 2 into one script, because most/all of the same information is needed\*\*\***

Script 2

* generates predictions for one validation week using prediction rule generated in script 1
* takes validation week range
* takes output file name
* takes name of prediction rule .Rout file from script 1
* loops over all prediction rules to generate all predictions for a given base model in a given validation week
* loops over all validation weeks to generate all base model predictions
* loads script 1 prediction rule and generates predictions for given validation week
* save all predictions in file
  + results correspond to yellow/orange matrix

Script 3

* for each final prediction week, for each base model configuration:
  + generates summary statistics across validation window; chooses best config of base model for a given final prediction week
* takes matrix of vectors containing predictions from Script 2
* takes validation window size
* takes rule for choosing best configuration given summary statistics
* takes output file name
* loops over all final prediction weeks
* loops over all validation window sizes
  + each validation window size creates one blue strip in red box; each cell in blue strip is the 'best' configuration of base model for one given final prediction week
* stores collection of blue strips, each cell is prediction rule with the best config of base model.

Script 4

* takes collection of blue strips from Script 3
* generates summary statistic/s of each blue strip, picks best strip and thus picks the best

validation window size for the base model.

* stores best blue strip, a list of prediction rules, for a base model, containing # final prediction weeks different models

Script 5

* for each final prediction week; each cell in best blue strip; use the prediction rule to generate final predictions for each final prediction week; each final prediction week then has predictions
* for each game in the week -- rbind predictions together to make one big column that has predictions

Script 6

* collect all final prediction columns from script 5 and cbind to make a matrix where each row is one game and each column is the predictions from a given base model
* cbind actual results for each game
* repeat model selection process, where now the training data is the matrix of final predictions from different base models; will come up with a blended model, bitch.