Maintenance Guide/“Documentation” for the RCR website

Nick Konz

January 6, 2019

1 The Functional Form Webpage

1.1 Where to find these files

All of the needed files

1.2 The C++ Source Code

The RCR source code is made up of the following files:
Source files

1. `RCR.cpp`
2. `NonParametric.cpp`
3. `FunctionalForm.cpp`
4. `MiscFunction.cpp`

Header files

1. `RCR.h`
2. `NonParametric.h`
3. `FunctionalForm.h`
4. `MiscFunction.h`

For usage with the website, I pasted all of the code from these files into one big C++ file, `RCRSWIGFUll.cpp`, **which also includes** the sets of functions and partial derivatives (functions) for each of the six model function types that the website uses: **linear, quadratic, cubic, power law, exponential, logarithmic. Please read through this source code file thoroughly, as well as the RCR paper itself to understand the entire algorithm, as it is quite complex.** These functions and partial derivative functions are used for the Gauss Newton iterator used within `FunctionalForm.BuildModelSpace()`. Specifically, the Functional Form constructor takes in a vector of function pointers to the partial derivative functions, with derivatives taken in the same order as the parameters are specifies. See the RCR source code documentation, as well as the source code itself, for more details. **Also included** within `RCRSWIGFUll.cpp` are two functions, `requestHandlerUnWeighted()` and `requestHandlerWeighted()`, which are directly used with SWIG (more on SWIG in a bit) so that Functional RCR can be called through python (via SWIG) on one of the six model functions, with either weighted or unweighted data. The arguments of these functions are four vectors specifying x data, y data, σ_y data, initial guesses for the model function parameters; also three integers, specifying rejection technique, size of the data (number of x/y datapoints), and an integer specifying the model function type. More on what these function returns in a bit.

The convention is that the aformentioned model functions are numbered 1,2,3,4,5,6, in the order of linear, quadratic, cubic, power law, exponential, logarithmic, respectively. The convention for the numbering of rejection techniques is 1,2,3,4 for `SS_MEDIAN_DL`, `LS_MODE_68`, `LS_MODE_DL`, `ES_MODE_DL`, respectively.

1.3 Using SWIG, and the Python controller

1.4 The Front End

1.5 The Interactive Plot (Bokeh)

1.6 Connecting Everything: The Linux Server and SSH Tap

1.7 Summary