* User-selected output directory (*mothership*)
  + Automatically generated BASSPRO\_output directory (*py\_output\_folder*)
    - Automatically generated BASSPRO\_output\_{timestamp} directory (*output\_dir\_py*)
      * This directory stores the output for a single BASSPRO run.
      * Currently, the metadata and BASSPRO settings files are made when created from the database or saved in their respective subGUI respectively, before BASSPRO is launched. The same is true for the STAGG settings files – they are created when the user saves them in the Config subGUI, before launching STAGG. There was talk of just holding the settings in dictionaries or dataframes until BASSPRO/STAGG is actually launched, but I personally prefer the flexibility and assurance of being able to open the actual csv files and review them and edit them in excel in addition to the subGUIs if I wanted. So I would prefer the csv files be created when the user interacts with the corresponding subGUI and saves. Additionally, if the user wants to dictate the order of categorical variables appearance on the figures produced by STAGG, they currently can only do so manually in graph\_config.csv, after it’s created in the STAGG settings subGUI but before STAGG is launched. So the front end behavior for STAGG settings files in particular would need to remain the same at least until a feature is added to the subGUI that allows the user to dictate variable order directly in the GUI.
      * That said, there’s an outrageous amount of redundancy and bloated dictionaries used to coordinate the communication between the user’s behavior and attribute assignment for those subGUIs, especially the STAGG settings subGUI, so that’s fair game.
      * metadata\_{timestamp}.csv
      * basic\_{timestamp}.csv
      * auto\_sections\_{timestamp}.csv and/or manual\_sections\_{timestamp}.csv
      * And for each signal file given as input to the BASSPRO module:
        + {signal\_file\_name}.json
        + {signal\_file\_name}.txt
        + {signal\_file\_name}\_agg\_auto.csv and/or {signal\_file\_name}\_agg\_man.csv
        + {signal\_file\_name}\_all.csv
  + Automatically generated STAGG\_config directory
    - This directory serves as the default location for saving STAGG settings files. If the user chooses to save as, then the settings files are saved to the user-selected location AND to this directory. When STAGG is launched, the settings files in this folder are timestamped and copied to the timestamped STAGG output folder (*output\_dir\_r*).
    - variable\_config.csv (*variable\_config)*
    - graph\_config.csv (*graph\_config)*
    - other\_config.csv (*other\_config)*
  + Automatically generated STAGG\_output directory (*r\_output\_folder)*
    - Automatically generated STAGG\_output\_{timestamp} directory (*output\_dir\_r*)
      * This directory stores the output for a single STAGG run.
      * variable\_config\_{timestamp}.csv
      * graph\_config\_{timestamp}.csv
      * other\_config\_{timestamp}.csv
      * myEnv\_{timestamp}.RData (this .RData file is saved after STAGG has built a dataframe from concatenated JSON file data but before the model has been run)
      * myEnv\_{timestamp}.RData (this .RData file is saved after the model has been run, so it’s the bigger of the two RData files)
      * {dependent variable}.jpeg/svg (the user can choose the image file format via the main GUI)
      * Automatically generated StatResults directory
        + This directory is generated by STAGG and contains stats output.
        + QQ\_{dependent variable}.jpeg/svg
        + Residual\_{dependent variable}.jpeg/svg
        + stat\_basic.csv
        + stat\_res.csv
        + tukey\_res.csv