fnnr_config_file.py defaults

 # indicates comments in Python.

This is the exact text of fnnr_config_file.py as hosted by default on Github, but is uploaded as a text document so that one can easily read the settings offline

"""
Running the model
"""
numbers should be whole positive integers and strings (text) or floats (decimal numbers)

model settings
run_setting = "normal_run" # "normal_run" or "first_run" (strings with underscores); default is "normal_run"
plot_setting = False # pops up monkey demographic graphs that were exported to the Excel files; default is False
Note: due to time constraints and unfamiliarity with matplotlib, I only have monkey demographic plots in my model.
Plots should usually be generated in Excel.

monkey/human settings
family_setting = 20 # number of monkey families; default/recommended is 20; set to 1 for random walk mapping
year_setting = 20 # number of years the model will run, as an integer multiple of 73 5-day time-steps; default is 10
human_setting = "with_humans" # "with_humans" or "without_humans" (strings with underscores); default is "with_humans"

land settings
PES_span = 8
no_pay_part = 0.25 # chances a household would remain enrolled in GTGP immediately after payment ends
min_threshold = 0.25 # similar to no_pay_part in that it also multiplies with gtgp_part_prob

land scenario settings
scenario = 'flat' # types are 'flat', 'land_type', or 'time' as strings with underscores; default is flat
unit_comp_flat = 270 # only applies if 'flat' scenario is selected; stable compensation; default is ~250-500
unit_comp_dry = 200 # 'land_type' scenario only; compensation for dry conversion

unit_comp_rice = 400 # 'land_type' scenario only; compensation for rice conversion
unit_comp_before = 250 # 'time' scenario only; compensation before scenario_breakpoint year
unit_comp_after = 350 # 'time' scenario only; compensation after scenario_breakpoint year
time_breakpoint = 4 # 'time' scenario only; year that PES ends
land_step_measure = 6 # every 5 days (time-step) * land_count = land time resolution; default is 6 (30-day, monthly)

"""
Configuring randomwalk.py
"""
random_walk_graph_setting = False # generates random walks; set to True only if **family_setting = 1**; default is False
if random_walk_graph_setting == True and family_setting != 1:
    print("Please set the random_walk_graph_setting to False if you are running the model with multiple families.")