Test plan

UNIT TESTS

geogen::GeoGrid::GeoGrid

Test Title	Test	Test steps	Test data	How?	Expected result
Happy day GeoGen constructor.	Tests if a geogen is constructed properly if the input files are correct.	1) run popgen-sim with default_geogen-config.xml	Default_ geogen-config.xml flanders_cities.csv	We run the Geogen Constructor with correct files.	Variables initialized according to the config file. Cities all loaded into GeoGen with correct data.
Faulty row Geogen	Tests if one of the input's on the csv list is wrong	1) run popgen-sim with default_geoge n-config.xml	default_geogen-config.xml flanders_cities_fau lty.csv	We run the parse_cities function that parses the cities to a list. One row will contain invalid input. (a string where an integer is expected, vice versa)	Skips row of csv file, outputs warning that row is faulty. Continues running GeoGen
Faulty column structure city csv extra column	checks expected result, if we have an extra unwanted column in the csv	1) run popgen-sim with default_geoge n- config.xml	default_geogen-config.xml flanders_cities_f ault y_extra.csv	We use parse_cities again, this time we have an etra unrelevant column in the file. Since we check columns on labels this get's ignored	Extra column is ignored.
Faulty column structure city csv file, lost a column	Checks expected result if we miss a city column.	1) run popgen-sim with default_geoge n- config.xml	default_geogen-config.xml flanders_cities_f ault y_less.csv	Using parse_cities, if we lack one of the necessary labeled columns. Geogen will terminate.	Geogen terminated, missing data.
Happy day row count	Count amount of csv rows, check len map	1) run popgen-sim with default_geogen-config.xml 2) make a unit test in test enviorment and assert equal	Default_ geogen-config.xml flanders_cities.csv	We manually counted how many rows we have in the csv. We check this against how many cities we have. This should be equal.	Assert equal csv rows and map size city's in unit test
Antwerp correct data test	Check if the data filled in for id Antwerp in city's is correct	1) run popgen-sim with default_geogen-config.xml 2) make a unit test in test environment and assert equal to expected value's of antwerp	default_geogen-config.xml flanders_cities.csv	We sample the city of antwerp. Check the data manually with the data we received in our test.	Test should pass all data should be as expected data

Test plan

geogen::GeoGrid::generate_schools

Test Title	Test summary	Test steps	Test	How?	Expected result
Happy day, generate schools	Generate schools with all correct parameters. -schooled fractal: 0.7 -random seed 0	1) run popgen-sim with default_geogen-config.xml	default_geogen-config.xml	Assumption input tests succeed. Run the generate schools. Count the placed schools. Calculate by hand the expected amount of schools to be placed based on pop and fractions.	6058 schools created, placed according to discrete distribution in the city's
Negative total pop	Total population is negative	1) run popgen-sim with default_geogen-config.xml	default_geogen-config.xml	Run generate schools with a negative population	Exception thrown value error, negative pop
schooled fractal is above 100%	schooled fractal is above 100%	1) run popgen-sim with default_geogen-config.xml	default_geogen-config.xml	The % of schooled is abover 100%. Filled in in the XML	Exception thrown value error, schooled fractal above 100%
Negative school size	School size is negative	1) run popgen-sim with default_geogen- config.xml	default_geogen-config.xml	The expected school size is negative, given in de donfig file.	Exception thrown value error, school size is negative
Hand calculated total schools	Calculate total schools to be check against schools generated	1) run popgen-sim with default_geogen-config.xml 2) check school vec size against hand calculated school amount	default_geogen-config.xml	Assumption input tests succeed. Run the generate schools. Count the placed schools. Calculate by hand the expected amount of schools to be placed based	Set boundry in unit test enviroment, equal value's, should be equal
High populated vs low populated school amount	Check if the 5 highest populated cities have more schools then the 5 lowest	1) run popgen-sim with default_geogen- config.xml check in unit test if the 5 highest populated cities have more	default_geogen-config.xml	We sample the 5 highest populated cities and the 5 lowest. We check if the 5 highest cities combined have more cities then the 5 lowest.	Highest populated(5).school_amount >Lowest populated(5).school_amount >

Test plan

geogen::GeoGrid::generate_colleges

Test Title	Test summary	Test steps	Test	How?	Expected result
Happy day scenario for generate_colleges(N) default value for N =10	Tests if colleges are assigned to the correct cities and the number of generated colleges for each of those cities.	Read input Find N largest cities Generate colleges according to fraction of students. Check against expected values	Happy day scenario so considering GeoGrid's constructor read a valid input file which is done by parser::parse_cities()	We search for the X largest cities. We check of those cities all have a college.	N largest cities should have X colleges where X= #students / 3000 #students is the number of students for a particular city
Alternate scenario for generate_colleges(N)	Tests bad input or no input at all.	- Read input - Run as if it was a happy day scenario	Bad input file or no input at all	We search for largest X cities, we should have colleges for x biggest cities, so we check for it, If we have less cities then	Again, largest cities should have X colleges, considering some cities were read correctly. If no cities are left due to bad input, no colleges should be present.
Unit test for adjustLargestCities(...)	Tests function used by generate_colleges() to ensure correct behavior.	- Add N cities - Add the N+1th city and check against expected behavior.	A number of manually created cities as well as a city that should be added.	We add N cities, Then add City N+1 and check if this replaced the smallest city out of our N cities,	The first N cities should be added immediately. The N+1th city should replace the city with the smallest population if that city has a bigger population.
Unit test for assignCollege(...)	Tests function used by generate_colleges() to ensure correct behavior.	- Generate colleges for a particular city. - Check against expected number of colleges.	A number of students or fraction of the population representing the students for the particular city.	We check of the amount of colleges generated are equal to #students/3000	Number of colleges should be equal to #students / 3000

findSmallest(...) will not be tested since this is a trivial function and it's being used by adjustLargestCities(…), therefore if the tests pass for adjustLargestCities(…), findSmallest(…) should be working correctly.

Test plan

geogen::GeoGrid::generate_communities

Test Title	Test summary	Test steps	Test data	How?	Expected result
Happy day generate_communities()	Tests if communities are generated correctly given a certain input.	- Read input - Generate communities - Check against expected values	Any legal input file containing the total population size, the number of cities and the population per city.	We check the highest populated cities community_count is higher then the lowest	Cities with a larger population will have more communities than cities with a smaller population.
Equal_cities	Tests if all cities will contain the same number of communities if they have a population of equal size.	- Read input - Generate communities - Check against expected	Any legal input file containing the total population size, the number of	Check if pop == Equal, Communities_count == Equal	Cities with equal populations have an equal number of communities.
Extreme_cities	If a city has a negligible population it will contain no communities.	- Read input - Generate communities - Check against expected	Any legal input file containing the total population size, the number of	If city_pop < N community_count == 0	Cities with insignificant populations will have no communities.

Test plan

geogen::GeoGrid::generate_workplaces

Test Title	Test summary	Test steps	Test data	How?	Expected result
Happy day generate_workspaces()	Tests if workspaces are generated successfully with given required inputs	- Read input - Calculate the number of people working in a region = active population in the region - out commuters + in commuters - Generate workspaces	Any legal file(s) containing the information of the total population size, the population per region, the population distribution per age, commuting behaviour	city_households =City→pop / household_size	Workspaces are spread according to the number of active population active in that region (active population + in commuters - out commuters) and the average workers for workspace is respected
Commuter vs local active population test	Tests if two regions have the same number of workspaces; if the both regions have the same number of active population but a region has active population without incoming and outgoing commuters and another region where all the active population consists of only incoming commuters (nobody from the own region)	- Read input - Calculate the number of people working in a region = active population in the region - out commuters + in commuters - Generate workspaces - Check if the expected value is met	Any legal file(s) containing the information of the total population size, the population per region, the population distribution per age, commuting behaviour	Check if Workspaces city1 == Workspaces city2	The same number of workspaces for the both regions
Extreme region	Test if a region with no effective active population will have workspaces	- Read input - Calculate the number of people working in a region = active population in the region - out commuters + in commuters - Generate workspaces	Any legal file(s) containing the information of the total population size, the population per region, the population distribution per age, commuting behaviour	region_workspaces == 0	That region will have no workspaces

Test plan
geogen::GenPop::AssignHouseholds

Test Title	Test	Test steps	Test data	How?	Expected result
Happy day assign_households	Tests if households are generated successfully with given required inputs	Assumption the Geogrid generation works correctly, We run assign_households on given default input, households are generated	Default input file, default config and csv files	We Loop over our cities and check their households, Households should be #people/household_size	Households are spread according to the number of people and with discrete distributed according to input, We placed as many households as pop/household_size
Household size = 1	Tests what happens if each household has 1 person in it	Run assign households with household size = 1	Any legal file(s) containing the information of the total population size	Check if households = pop_size for each city	The same number of households then people in each household
Household_size = pop_total	Tests extreme that a household is a big as a city	Run households with household size = number_of_people	Any legal file(s) containing the information of the total population size, the population per region, the population distribution per age, commuting behavior,	Households == 1, Placed in a city	One Household is placed in a City

Test plan

geogen::GenPop::AssignToSchools

Test Title	Test	Test steps	Test data	How?	Expected result
Happy day assign_to_schools	Tests if people are added to the schools -	Assumption the Geogrid generation works correctly, We run assign_to_school on given default input, schools are generated	Default input file, default config and csv files	We Loop over our cities and check their schools, attendants of schools should be around average_school_size	School attendants are spread across the schools according the distance radius and discretely distributed,

geogen::GenPop::AssignToColleges

Test Title	Test	Test steps	Test data	How?	Expected result
Happy day assign_to_Colleges	Tests if people are added to the College -	Assumption the Geogrid generation works correctly, We run assign_to_colleges on given default input, colleges are generated	Default input file, default config and csv files	We Loop over our X biggest cities and check their college attendants of schools should be around average_college_size (certain margin)	The colleges attendants are between margin for each college

Test plan

Geogen::GenPop::AssignToWorkPlaces

Test Title	Test	Test steps	Test data	How?	Expected result
Happy day assign_to_Workplaces	Tests if people are added to workplaces	Assumption the Geogrid generation works correctly, We run assign_to_colleges on given default input, workplaces are generated	Default input file, default config and csv files	We Loop over our cities, check if people are placed in workplace by, since it's stochastic, We use a margin	The workplace attendants lie between margins

Geogen::GenPop::AssignToCommunities

Test Title	Test	Test steps	Test data	How?	Expected result
Happy day assign_to_Colleges	Tests if people are added to the Community	Assumption the Geogrid generation works correctly, We run assign_to_colleges on given default input, schools are generated	Default input file, default config and csv files	We Loop over our X biggest cities and check their college attendants of communities should be around average_community_size (certain margin)	The community attendants are between margin for each average community size

Test plan
Scenario Tests

User uses default data and checks one city:

User loads in the default config file with the cities of flanders. The user get’s to see a map with circles for each city. If the user clicks on a city circle he or she gets the pop info of the city.

User uses default data and checks multiple cities:

Same data as the test above. The user now selects multiple cities. The user will see data of the population of all selected cities,

User uses non critical wrong data:

If a city contains invalid input data, the program will still run, The user will get a warning that de data of that city is wrong. The map will load and the user will be able to see all cities except the wrong city data.

User uses Critical wrong data:

If the fractions do not make sense (negative, over 1) the program will not run, instead it will abort and display a error message that the input contains invalid data