

Test Report

Run test by typing this URL address in your browser.

http://localhost:8000/primetest/index_test

And all test results will be output into text file named test_results_with_anja_test_excel_data.py under project directory.

Test id in test result .py file is 6 smaller than the id in test excel file. That means if you add test id by 6, it should lead you to the same line in the test excel file. If python models' total death averted is within (anja testresult-0.01, anja testresult+0.01), then pass the test, otherwise fail.

Total Test Number	Passed Test Number	Failed Test Number
321	231	90

We have two main kinds failed tests.

Case 1: Test 11-70

All veg/fruit exposures discrepancy are caused by different compound calculation methods. In Marissa's Python model, the fruit and veg are taken as different exposures, so the compound results of these two exposures are basically using counter-factual mortality of fruit as baseline mortality of veg, in the same way as other exposures.

So if we can modify Anja's Excel spreadsheets according to this, we should be able to get the same results.

- 1) In Fruit/Veg, Veg CHD death rate first column is using population distribution of fruit, I guess it should be population distribution of veg rather than fruit.
- 2) As initial mortality for Veg CHD is the counter-factual result of Fruit CHD, so, the counter-factual mortality of Veg CHD is compound result of both veg and fruit (Eq.2). But at the end of Fruit CHD, the excel model is using a formulation like (Eq.2), this cause the fruit-averted death is double counted.

$$\text{mortality_averted_fruit_CHD} = \text{counter-factual_mortality_fruit_CHD} - \text{original_mortality_CHD} \text{ (Eq.1)}$$
$$\text{mortality_averted_veg_CHD} = \text{counter-factual_mortality_veg_CHD} - \text{counter-factual_mortality_veg_CHD} \text{ (Eq.2)}$$
$$\text{final_counter-factual_mortality_CHD} = \text{original_mortality_CHD} - (\text{mortality_averted_veg_CHD} + \text{mortality_averted_fruit_CHD}) \text{ (Eq.3)}$$

This affects 60 tests.

Case 2: Test 261-290

Tests for non-rate for alcohol is around 1.2 discrepancy is because:

in the Alcohol sheet, relative risk calculation for Liver Cancer is using E71, like cell F309=\$D309^(F\$289-\$E\$71)/10), E71 should be E289.

This affects 30 tests.

Case 3: Test 280

Test parameters has '0', this causes not-a-number error.

This affects 1 test.

There are some rules to follow to avoid minor discrepancy:

1. Relative risk table's data was 6 digits decimal when it was originally designed, but it might cause discrepancy as big as 2.0 between python results and excel results. Then Relative risk table's data is changed to 19 digits decimal places.
2. Bins table measures table also needs more decimal places as some bins are not manual input but calculated, like bins of Salt.

How to rerun rests when excel model test results are updated.

1. Store Anja's updated test excel in the folder web-django/Run/Excel Model/....csv
2. Clean the database table that stores test parameters
Can do such command:
drop table primemodel_testparameter # in mysql
./manage.py syncdb # This recreate the primemodel_testparameter table
3. update the database prime, table primemodel_testparameter
This can be updated through using the automatic extraction procedure in web-django,
python mysql_from_excel_only_test.py

The file address at the bottom of the python file might be updated if Anja's text excel 's name is changed.

4. rerun Python test module through this url:
http://localhost:8000/primetest/index_test
or through command line, go to primemodel/command
python test_to_run_in_command.py
5. Check test results in primetest folder
Test results are stored in a .csv file and a .txt file. The content of the two files are the same.