Ian Bolliger

Section AC

3/4/13

CSE 140 Final Project

# Picturing pain: visualizing estimated disease and injury prevalence, incidence, and mortality to evaluate model accuracy

## Research Questions

1. Given estimates of disease prevalence, incidence, and excess mortality for every country-age-sex-year group (with uncertainty intervals), what is the most concise yet informative set of visualizations to generate in order to allow experts to evaluate the plausibility and accuracy of the estimates?
	1. It was determined that age patterns by region and a scatterplot of age-standardized rates in two different years by country were the two most valuable figures for model evaluation.
2. How can the automated creation of these visualizations be incorporated into the existing DisMod III meta-regression program for generating disease estimates? An informative automated set of figures will make the process of evaluating model plausibility more efficient and effective, allowing for improvements to the overall project of mapping disease and injury burden by demographic grouping.
	1. Through the generation of a class called MeanEstimates, data was converted into efficient tabular form from which it is relatively simple to generate figures of one’s choosing. Instances of this class can be easily generated from within the existing DisMod code at the conclusion of the model.

## Motivation and Background

The Institute for Health Metrics and Evaluation (IHME) has recently produced results from the Global Burden of Disease (GBD) 2010 study, an effort to develop coherent estimates of mortality- and disability-related burden of 289 diseases and injuries. The burden for every disease and injury is estimated for each country, for each sex, for each one of 20 age groups, over multiple years. The goal of this project is to create a “global health blueprint” with which disease burdens can be compared across a variety of comparison groups. This overcomes issues relating to missing and incomparable data and allows countries and global health organizations to measure progress over time. Ultimately, this data can be used to better allocate resources toward preventing and curing the diseases responsible for the largest health burden.

One of the most important inputs to the generation of burden estimates is prevalence for each disease. In addition, valuable information can be gained from incidence and mortality results. A meta-regression tool, called DisMod III M-R, has been developed to produce internally consistent estimates of these three parameters. In other words, the incidence, duration, mortality, and prevalence values do not conflict with each other within the results. The model incorporates point estimates and uncertainty intervals gleaned from a meta-analysis of studies reporting any of these values and uses Bayesian techniques to produce its internally consistent country-age-sex-year-specific results. The model produces a vast number of data points for each disease: 3 values (mean, upper uncertainty interval, lower uncertainty interval) x 2 sexes x 20 age groups x 187 countries x 3 years (1990, 2005, 2010) = 67,320 data points. The models for virtually all causes have a much smaller input dataset and thus the value of much of the result dataset can sometimes be affected significantly by model parameters and the presence of outlier data in the input dataset. Because of this, models often need to be evaluated by experts familiar with a particular disease in order to assess the plausibility of results. If anything surprising is found, further investigation is needed to discover whether the unexpected results came from incorrect/non-ideal modeling parameters or simply represent previously unknown aspects of a disease.

Due to the vast quantity of resulting data points and the hundreds of diseases being modeled, it is infeasible for experts to evaluate each point individually. Therefore, the results of the model need to be compiled in such a way that plausibility over a wide set of attributes can be assessed quickly and effectively. Existing visualizations developed for this purpose arose organically as the DisMod model was being developed and have been found to be suboptimal. Therefore, an analysis of which types of figures and tables best convey the most valuable information for model evaluation was necessary. The results of this analysis and the code developed to automate the generation of these figures as a part of the DisMod III process is detailed in the remainder of this report.

## Dataset

The dataset to which the summary figure/table-producing algorithm will be applied consists of all of the results of DisMod III models run on individual diseases and injuries. These files are located internally on the servers within IHME. To facilitate reproduction of the produced figures and tables, the DisMod results for this particular cause (Road Traffic Injuries) are included in a zip file with the Python code developed for this assignment.

## Methodology

First, a brainstorming session with those who most often make decisions on the plausibility of DisMod models helped develop a list of appropriate figures and tables to generate. Because of the time constraints of this project, I decided to focus on automating the generation of two types of figures – plots of age patterns by region and scatterplots of age-standardized rates in 1990 and 2010 by country.

There are 21 specific regions classified within the GBD study, and the regression framework within the DisMod model runs at the regional level (with random effects on countries within each region). Thus, viewing the age-specific results by region gives valuable insight into the overall plausibility of the model’s results. The plots of results by region-age were developed for each year-sex group modeled and for each parameter (prevalence, incidence, and excess mortality). Similar figures are currently viewable via the web interface of DisMod, but greater flexibility in designing and examining these figures was desired. For this reason, code was developed to produce local, more easily modified versions.

The type of summary figure deemed second most valuable was a scatterplot of age-standardized rates in 1990 and 2010 by country. This gives model-evaluators an idea of whether any unexpected results have been generated at the country level in terms of absolute rates or change in rates over time. These were developed for each parameter and for both sexes. Similar figures to these did not yet exist on DisMod’s web interface.

In the future, several more auto-generated figures, as well as tables, will be created for each cause that is run through DisMod. In addition, users will be able to selectively generate customized summary figures if they wish to examine a certain aspect of the results not easily viewed in the auto-generated suite of graphs.

In order to generate these figures, a class was first created that contains the mean estimates at the country and region levels for each age-sex-year group for a given disease parameter, as well as some other disease- and model-specific information. This class, called MeanEstimates, serves several purposes.

1. It standardizes the format of the dataset from which the figures are generated. Currently country-level and region-level DisMod results are stored in different file formats and in different locations. By generating a class that gathers both types of results and stores them in a similar fashion, figure- and table-generating codes can more efficiently generate both country- and region- level figures if desired.
2. It allows for easy manipulation of the data. Both the country and region-level estimates are stored as Panel objects (from the ‘pandas’ package), which are attributes of a MeanEstimates instance. Tabular data manipulation is facilitated by Panel objects, and code can be written to apply similar algorithms to both the country and regional tables within the MeanEstimates object.
3. It can be easily incorporated into the existing DisMod model code, which is written in Python. MeanEstimates objects can be generated at the completion of each model and the figure-generating methods can then be called to produce the desired summary figures.
4. It allows for easy adaptation to changing DisMod result formats. DisMod is a relatively new and constantly updated modeling tool, and it is likely that the format of its results will change in the future. By utilizing the MeanEstimates class, a change in result format necessitates a change only in the MeanEstimates initialization code and not in any of the figure generation methods.

Once this class was created, it was simple to take the region-age-year-sex specific results and plot the age patterns for each region-year-sex. To get the age-standardized rates scatterplot was slightly more difficult. First, weights were incorporated. Since single-age age weights based on the global population were not readily available, the temporary solution implemented for this assignment was to use equal weighting for each age. In other words, to get an age-standardized rate, I simply took the unweighted average of the rates at each age for a given country-sex group. Since there is no “correct” weighting scheme for age-standardization, this simplification should not have a large effect on the conclusions drawn from the summary figures. The value in viewing ASRs is in comparing between countries that could potentially have different population structures. The absolute value of an ASR for a given country has little meaning. Once the ASRs for each country-year-sex group were calculated, each country was then plotted as a point on a scatterplot of 1990 vs. 2010 age-standardized rates. The countries were color-coded according to which region they belonged to.

To avoid cluttering of the graph, country names were omitted from the scatterplots. A simple way to identify individual countries without covering the figure with text would be valuable in the future to quickly identify where any outlier country-specific results are coming from.

## Results

The results of this assignment are represented more by the tool itself than by any concrete results of a specific analysis. To illustrate the benefits of the tool, examples of the summary figures produced for RTIs are depicted in Figure 1 and Figure 2 and a brief discussion of the type of information that can be quickly gleaned from these figures follows.



Figure . Age pattern of RTI incidence for males in 2010.



Figure 2. Scatterplot depicting age-standardized incidence in 1990 and 2010 by country. Each country is represented by a single point and points are grouped by region. A line of equivalence is shown in black.

As can be seen from Figure 1, the DisMod model predicts significant differences between regions with respect to road traffic injury incidence. There seems to be about twice as many road traffic injuries in Oceania and South Asia than in Western Europe. Additionally, as one might expect, the model estimates the highest RTI incidence in the 20-30 year old age group. Figure 2 shows that, universally among all countries, the model estimates little change in RTI incidence between 1990 and 2010. This seems reasonable as there are likely as many factors increasing RTI incidence (i.e. more cars in developing nations with poor road infrastructure) as there are factors decreasing it (i.e. improving road infrastructure and traffic policies)

## Reproducing Results

In order to run this code, one must first install the pandas, simplejson, and matplotlib packages for Python. With these packages installed, a user should unzip the folder submitted to the Dropbox (bolliger\_final\_project) to a desired location and navigate to that directory in the command line. Typing:

*python produce\_summary\_figures.py 40182 incidence*

in the command line should then run the code to generate the desired figures. The summary figures produced by the code will be found in “*dm-40182/summary\_figures”*. In the command line interface, the argument 40182 represents the model number of Road Traffic Injuries and “incidence” is the parameter displayed in Figure 1 and Figure 2 above. Due to size constraints, only the RTI incidence results are included in the zip file. Unfortunately, this means the user may not use any other model ID or disease parameter when running produce\_summary\_figures.py. These two arguments are included only to show that, when used in conjunction with the IHME server, this figure-producing code will take as an input the any results of any disease-specific model and any disease parameter estimated in that model.

## Collaboration

Abraham Flaxman, PhD – Developer of Dismod III M-R. Provided access to and assistance in interpreting the current DisMod code for use in integrating the figure generation code.

Christopher JL Murray, MD, DPhil – Principal investigator of the GBD 2010 study. Helped in brainstorming the list of figures and tables that should be produced with each DisMod run.

## Reflection

Constructing these figures took much longer than I had anticipated; however, a large part of that was due to the desire to create a data type that would be easily accessible for future plotting. It was challenging yet fun to purposefully develop the ability to add functionality that I knew I was not going to have the time to implement for this particular project. There were many, many bugs that arose when writing this code, largely because I was learning how to navigate the pandas package on-the-go. I chose to use the classes contained in this package because they are used frequently by others at IHME and are already used within the existing DisMod code. While this may have made the implementation time much longer than if I had simply used dictionaries and other standard Python data types with which I am more familiar, I think gaining experience with pandas will make future tabular calculations much more efficient.

Through this assignment, I gained an appreciation for how difficult it can be altering a code to be used in slightly different environments or contexts. For example, it was more complex than I initially thought to integrate my module into the existing DisMod code and then to alter it again to make it able to process a subset of its typical input dataset while living in a different filepath (that of the TFs who will download and run my code). This was made especially difficult because I did not want to include any of the proprietary DisMod code which was called upon in the first version of the code I wrote. Thus, some alterations had to be made so that my code performed some of the functionalities normally performed by subpackages of DisMod.

In an ideal world, I would have spent more time understanding the structure of the existing DisMod code before I began writing my own module to be called by that code. There were a few classes and modules already present within the DisMod code that I only discovered mid-way through my implementation that happened to be very helpful. I wasted time early on in my coding duplicating the functionality of previously-generated modules. However, given the complexity of DisMod, it seemed a daunting task to try to understand the whole of it before writing my module. Presumably, this is how programmers working on other complex programs feel as well. At the very least, I should have spent more time getting familiar with pandas before I started using it. It became a significant part of my code and, as mentioned, I wasted a significant amount of time debugging issues related to pandas-based methods and classes.

One strategy that saved me a significant amount of time was saving objects created early on in my code and then writing temporary code to import those saved objects. Since I was working with several GB of data, it took several minutes to generate my MeanEstimates objects at the beginning of my code; however, once these were created they themselves were not that large. Once I figured out a good way to save the attributes of these objects that took a long time to generate, I was able to more efficiently debug functionality in my code that occurred after the creation of MeanEstimates objects. Before I did this, I would have to wait for these objects to be created every time I reran the code to check whether I had fixed a bug. This was a form of divide and conquer that I had not previously utilized.

Overall, while it was a little frustrating to spend upwards of 40 hours on a project and only end up with essentially two graphs, it is exciting to know that the real benefit of my code lies in the data structure I’ve created from which many more valuable figures can be generated easily and efficiently in the future.