For a purpose of space I will attach my whole report on this analysis.

4 Bill&Melinda Gates Foundation

For a couple of years now, the Gates Foundation has been putting tremendous efforts to have an impact in the developing countries. In particular, they are collecting huge amount of data through surveys and on-field experiments in poor countries around the world. Data about education, health, finance, etc. led the Foundation to create a unique project, named HBGDki (for Healthy Birth, Growth and Development knowledge integration) aiming at developing knowledge towards all this data. The lab I joined is a member of the latter. The main duties I was assigned to were:

1. Developing new tools to allow program managers at Gates Foundation to access the full potential of Bayesian inference through our tool BayesDB.

2. Analyzing several datasets and show proof of concepts of the tools

4.1 Visualization tools

I started by writing a python script that would allow any user to generate a data sketch, of any dataset, from the command line. The challenges were multiple. First, we had to figure out what kind of information to include in the two pager sketch so that any reader could get a sense of the content of the dataset and even more, understand a bit the field. Second, the user experience was crucial since most of the targeted users for this sketch generation tool were not computer scientists and did not want to deal with lines of codes.

To address the first issue, several iterations with my supervisor were necessary and led to the following template:

1. The title of the dataset with its description (generally the name of the dataset was opaque)

2. A warning message automatically displayed in case the user did not have the codebook. The codebook is the description of each variable name, that are most of the time opaqe (for instance SIFTMM for Supraillac Skinfold Thickness (in mm))

3. A table summarizing the shape of the entire dataset (columns and rows) and the shape of the random subsample to run the analysis on. Indeed, since the analysis takes time, we decided to do the sketch on a subsample of 1000 rows and 20 columns.

4. The number of models (markov chains), iterations (of each chain) and the time of analysis

5. A dependence probability heatmap showing dependence probabilities from pairwise columns of the sample

6. A pairplot of 2 random variables

7. Three first entries for each of the variable to give a sense of the actual data

8. A table summarizing the type (numerical or categorical) of the random variables in the metamodel and the percentage of missing values.

The second issue, with respect to the user experience and the ease of use of the tool, was a bit tricky. Indeed, many choices have to be taken by whoever uses the script to generate it: codebook file to include, how many columns and rows to subsample, include the description of the dataset for external readers, the number of models one should create, the number of iterations, the type of each variable, etc.

To make it super simple and straight forward we decided to let less freedom to the user since the purpose of this tool is not to do a proper statistical study but to have a glance at a dataset and have some insights on a particular field. As a result we limited the number of models to 50 and iterations to 100. Enough for the markov chains to converge and quite reasonable to run on any personal computer. Also, we use the GUESS function of our metamodel that automatically detect categorical and numerical variable (and assigning them broad prior distributions) avoiding the user to select manually each distribution for each variable.

At the end, we came up with a script that can be run from the command line. You have to whether run:

```
python sketch.py file.csv file_col.csv
```

or

```
python sketch.py
```

to run it on all the datasets in your current folder.

Figure 8 showcases an example of a sketch:

Sample Sketch of BngR_Study: data.csv

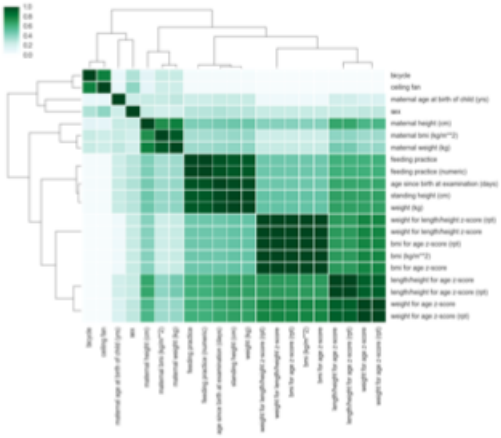
Cross-Sectional Dietary Survey in Children Aged 24-48 months in Bangladesh

Shape:

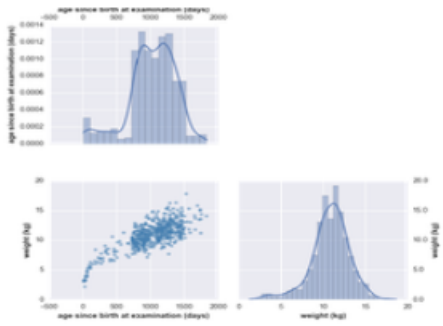
Object	Rows	Columns
Dataset	548	79
Sample	548	21

Initialization of **50** models with **100** iterations for **561** seconds

Sketchmap of dependancies



Pairplot



Data for 10 variables (rows) from 3 records (cols):

Record ID	0	1	2
age since birth at examination (days)	1055.0	1746.0	1245.0
weight (kg)	13.7	13.5	11.7
standing height (cm)	89.1	96.05	88.95
bmi (kg/m**2)	17.256994436	14.633190624	14.787472736
weight for age z-score	-0.25	-2.16	-2.12
length/height for age z-score	-1.67	-2.75	-2.61
weight for length/height z-score	0.96	-0.73	-0.97
bmi for age z-score	1.22	-0.46	-0.56
weight for age z-score (rpt)	-0.25	-2.16	-2.12
length/height for age z-score (rpt)	-1.67	-2.74	-2.61

Figure 4: Sample Sketch of BngR Study

The outcomes were very encouraging and the lab is currently negotiating with a visualization third party to explore new opportunities.

4.2 Growth & Development datasets

I've analyzed dozens of datasets from malnutrition to IQ. Here are an example of some of them:

1. CONTENT Study of Growth, Diarrhea and Socioeconomic Status
2. Randomized, community-based trial of the effect of zinc supplementation, with and without other micronutrients, on the duration of persistent childhood diarrhea in Lima
3. Cross-Sectional Dietary Survey in Children Aged 24-48 months in Bangladesh
4. Systems Dynamics Modeling of Growth in Children
5. Dataset of a 1959 cohort with data on IQ at age 1, 4 and 7. Plus some maternal parity and cigarettes use data

The work was basically to find quick and dirty dependencies that could make sense between variables with the minimum amount of expertise, obviously, and of human processing. Some outstanding results came out of an analysis ran on GUSTO (Growing Up in Singapore toward Healthy Outcomes) where simple analysis and plots enabled us to write a convincing stories about the dependencies between level of Zinc, Iron and Magnesium of the mother and some physical characteristics of the baby such as the head circumference, BMI for age at z-score, etc.

Finally, I finished my visit with a work on the satellites example the lab particularly cherish for all the options it gives to showcase our work. The datasets include hundreds of satellites and 23 metrics such as the name, perigee, apogee, period, lifetime, etc. Program managers from the Foundation were dubious about the ability of our metamodel, Crosscat, to cluster the rows and the columns into meaningful clusters. This is indeed a crucial question since the whole dependency matrix is based on this clustering (the estimate of the mutual information between two variables is the number of times they are in the same cluster).

The idea we had was to draw the final state of the markov chain (of one model only) and compare this to pairplots and simulated data.

I wrote a function to draw the state of any model and highlight the clustering of the variables. Figure 9 shows the raw data (none of the rows and columns are clustered). Red color is nan values and the shade of black are the values (dark is low).

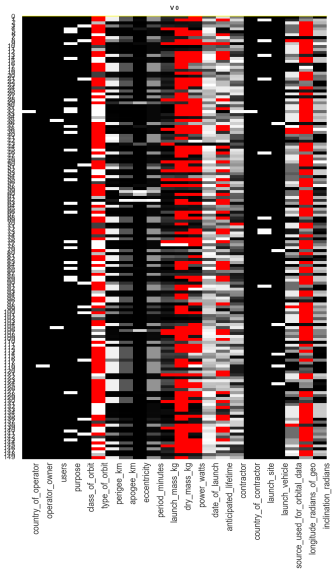


Figure 5: Raw Data

After analysis, the Figure 10 shows the actual clustering Crosscat made. The yellow lines are the row cluster assignments and each variables are organized into views.

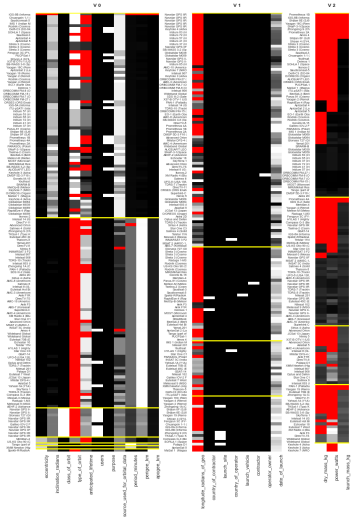
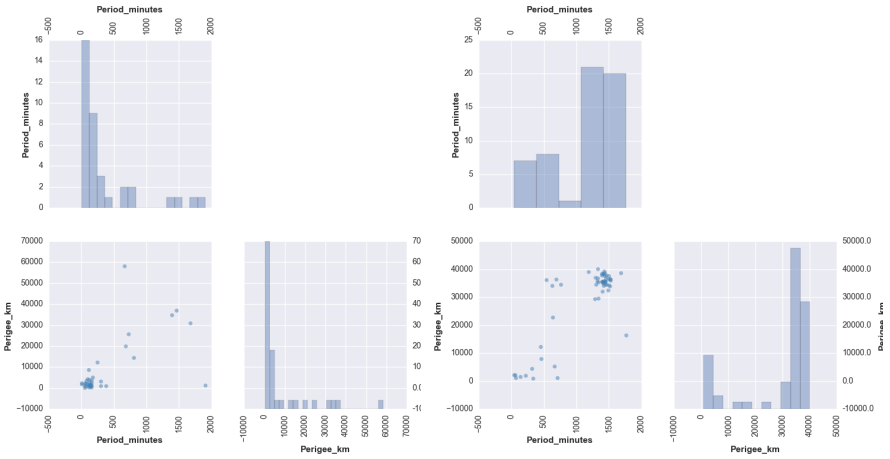


Figure 6: State view: clustering rows and columns in views

We can easily notice two satellites clusters defined by the Anticipated_Lifetime (whether low or high). We then plotted the joint distribution of the perigee and period conditioned on the anticipated lifetime,
$P(period\_minutes, pergigee\_km | Anticipated\_Lifetime = 3)$ and
$P(period\_minutes, pergigee\_km | Anticipated\_Lifetime = 15)$ to assess that the clustering makes sense. The pairplots shown on figure 11 are pretty straight forward.



(a) Cluster conditioned on low lifetime (b) Cluster conditioned on high lifetime

Figure 7: Pairplots showing the conditional clustering

These results were useful to convince the program manager during a call the week after. Some more efforts are being put to extend this to different metamodels and with simulated data.

5 Measuring accuracy of inference-based algorithm

This following section relates to the main research I've been doing during my visit. The next sections will deal with the applications I've pursued whether on high dimensional datasets or even my contribution to the general engineering effort.

Many of the most demanding applications of computing tolerate a spectrum of solutions to a given problem. Examples can be found in fields as diverse as numerical methods, lossy compression, microchip layout, robotics ([5]), computer graphics ([6]), and genetics ([1]). The additional flexibility in these applications makes programming more challenging: There are often several reasonable algorithms, each with its own tunable parameters and tradeoffs between com-